



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

LIMPIEZA DE UNA BASE DE DATOS

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
M A T E M Á T I C A
P R E S E N T A :
MARÍA DEL CARMEN POO MANTECON

L

DIRECTOR DE TESIS:
Dr. RAFAEL RENÉ DEL RÍO CASTILLO



1998

260333

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

M. en C. Virginia Abrín Batule
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo de Tesis:

Limpieza de una Base de Datos

realizado por María del Carmen Poo Mantecon

con número de cuenta 7693177-6 , pasante de la carrera de Matemáticas

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis

Propietario

Dr. Rafael René del Río Castillo

Propietario

M. en C. María del Carmen Arrillaga Arjona

Propietario

Act. Renata María Villalba Cohen

Suplente

Dr. Raúl Rueda Díaz del Campo

Suplente

Dr. Fernando Brambila Páz

Consejo Departamental de Matemáticas

Mat. Julio César Guevara Bravo

[Handwritten signature: María del Carmen Arrillaga]

[Handwritten signature: Raúl Rueda Díaz del Campo]

[Handwritten signature: Julio César Guevara Bravo]

A la bella Lola que nunca quitó el dedo del renglón.

A mi papá que le hubiera dado tanto gusto.

A Jaime por haberme regalado las 2 cosas mas hermosas del mundo, mis hijos.

A mi vasta familia.

Y por supuesto a mi Javi y a mi Jaime que son la alegría de mi vida.

INDICE

Introducción	1
Plan de Trabajo	5
Diagnóstico	7
Análisis del diagnóstico	22
Propuesta de limpieza	25
Logística del proceso	34
Detalle de la limpieza	41
Limpieza en línea	67
Resultados	71
Anexo	72

Introducción

Este trabajo es parte de un proyecto realizado para el área de sistemas de una organización a la que de aquí en adelante llamaremos **La Compañía**. El objetivo principal de este proyecto era diseñar, desarrollar e implantar un nuevo sistema de cómputo que satisficiera las necesidades de La Compañía y que sustituyera al que actualmente se encuentra en producción cuya principal función es la de cobranza y control de clientes y créditos.

Uno de los puntos mas importantes para la puesta en marcha de la nueva aplicación fue el concebir una estrategia de migración de datos hacia el nuevo desarrollo. Relacionado con este punto está el objetivo de este trabajo, limpiar la base de datos actual con el fin de realizar una migración exitosa hacia el nuevo modelo de datos asociado a esta naciente aplicación.

La complejidad de las aplicaciones (actuales y nuevas), el hecho de que no existe una correspondencia uno a uno entre los módulos del modelo anterior y del nuevo, la distribución geográfica de la operación y el hecho de que el sistema será desarrollado por partes (es decir deberán convivir ambas aplicaciones durante algún tiempo) hacen muy difícil establecer un plan de limpieza y migración que resulte transparente para todos los que están involucrados en este proyecto.

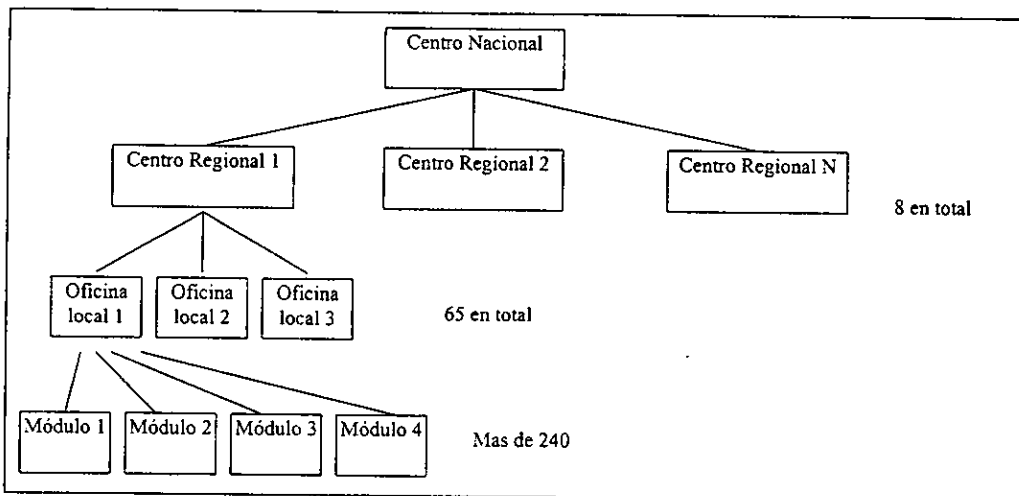
Situación Actual

La Compañía cuenta actualmente con un sistema que se diseñó en los años 1988 - 1989 y sustituyó una serie de procesos que anteriormente eran manuales y representaban serios problemas para ella.

El proceso que realiza este sistema es básicamente de cobranza y control de los clientes que se encuentran en todo el territorio nacional.

Este proceso de cobranza involucra la participación de alrededor de 12,500 empleados y es alimentado por la información de varios millones de clientes en todo el país.

El sistema opera actualmente con terminales "tontas" conectadas a equipos HP
 Las operaciones diarias se realizan a través de diversas entidades pertenecientes a La Compañía y que se representan en el siguiente diagrama:



Este sistema está formado por 8 módulos y cuenta con mas de 1000 programas escritos en 4GL que accesan una base de datos distribuida con aproximadamente 400 tablas en cada nivel (local, regional y nacional).

Uno de los módulos mas importantes de este sistema es el de Registro de Clientes y se prestan múltiples servicios como por ejemplo:

- Registro de clientes y créditos
- Cambios de situación del cliente o sus créditos
- Emisión de documentación.

Se estima que este módulo realiza alrededor de 30,000 movimientos diariamente. En su concepción original este sistema fue orientado para funcionar con procesos tipo batch, por lo cual no cuenta los con esquemas de seguridad, validación y auditoría necesarios.

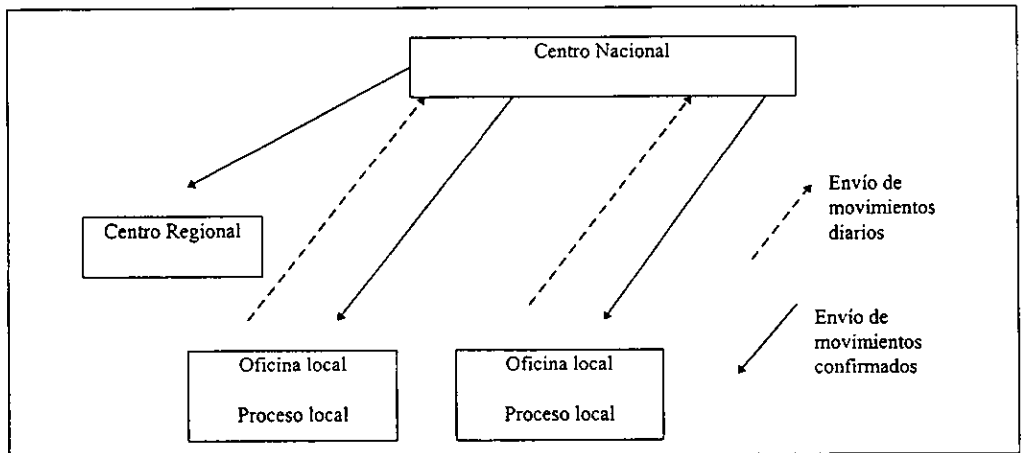
Esquemas de distribución y actualización de datos

Local	En cada oficina local se cuenta con un servidor HP en el que se encuentra la aplicación y la base de datos de los clientes asignados a dicha oficina.
Regional	A nivel de centro regional se concentra la información de clientes pertenecientes a las locales de la regional.
Nacional	A nivel nacional se concentra la información de todas las oficinas locales.
Módulos	A nivel módulo no existe acceso a las bases de datos.

Toda la operación ocurre localmente, es decir sólo se actualizan y consultan datos que se encuentran en la base de datos local, con cada operación se genera un registro en un archivo plano que al finalizar el día es enviado al centro nacional (ftp) para que sea actualizada la base de datos nacional.

En el centro nacional, se espera la llegada de los archivos que envían todas las oficinas locales y se inicia el proceso de actualización nacional. Una vez concluido este proceso se generan archivos de confirmación y complemento de datos que se envían de regreso (ftp) a las oficinas locales.

Las oficinas regionales se actualizan diariamente en base a estos archivos de respuesta que genera el centro nacional.



Problemática

Este tipo de actualización no es confiable y provoca muchas discrepancias entre las bases de datos. Existen varias causas que originan estas discrepancias de información entre el centro nacional y las oficinas locales así como errores e inconsistencias en los datos.

- **Comunicación.**
Si durante la transmisión ocurre una caída de la base de datos o del servidor, tanto el equipo del centro nacional como de la oficina local puede asumir que la información se recibió correctamente dado que no se cuenta con ningún tipo de validación, sin embargo puede haber llegado parte o nada, por lo que el movimiento es parcial o simplemente no es registrado.

- Los operadores.

La información remitida por la nacional o la local no es operada y simplemente se acumula hasta que un operador determina falta de espacio y borra los archivos.

- Funciones del sistema.

Existen varias funciones dentro del módulo de registro de clientes que tienen un alto riesgo de pérdida de información:

- Cambio de nombre. El número de cliente está íntimamente ligado al nombre del cliente. Sin un cliente cambia de nombre deberá cambiar su número de cliente.
- Cambio de domicilio. Los datos de un cliente que cambia de domicilio son enviados a la nueva oficina y eliminados de la anterior.

- Falta de validación.

El sistema tiene grandes carencias de validación ya que fue diseñado para procesos batch.

La falta de integración entre los módulos y el manejo de información, la dificultad para extracción de datos y el inadecuado manejo de los créditos son algunos de los problemas que generaron la necesidad de rediseñar el sistema.

Uno de los beneficios mas importantes que brindaría el nuevo sistema es la integración de aplicaciones e información y una arquitectura que separe datos operativos, datos sobre reglas del negocio y datos de consulta ejecutiva.

Plan de Trabajo

Dado que el objetivo principal de la limpieza de datos era su posterior migración al nuevo modelo, el ideal sería iniciar el trabajo haciendo un análisis del nuevo modelo propuesto para así determinar las necesidades de limpieza de la base de datos original. Sin embargo el diseño del nuevo modelo no se había concluido y se tomó la decisión de realizar la limpieza de datos sin considerar el diseño del nuevo sistema dado que la información sobre clientes es el activo mas valioso para La Compañía.

Aunque no se había concluido el diseño de la base de datos ya se conocían algunos requerimientos que sería necesario considerar:

- Los nombres de los clientes deberían estar separados en campos
- Todos los domicilios deberían estar codificados de acuerdo a los catálogos que maneja SEPOMEX

En la base de datos actual, los nombres de los clientes no se encontraban separados, simplemente se registraban como una cadena de 80 caracteres y se utiliza el carácter “” para hacer la separación, lo mismo ocurría con el domicilio. Por esta razón se decidió contratar el servicio de una empresa especializada en la limpieza de este tipo de datos para que ellos realizaran la separación en campos. A esta actividad se le llamó normalización de datos.

En el Anexo I presento a detalle el plan de trabajo y en este capítulo sólo describiré las actividades de manera general.

Normalización de datos de nombres y apellidos

Definición de la problemática de la base de datos

Definición y obtención de BD muestra

Inventario total (equipos, instancias, dbspace, bases de datos por local, regional, nacional, comunicaciones)

Análisis general de BD muestra

Identificar instancias, dbspace, BD's en local de BD muestra

Inventario de tablas de una local (BD muestra)

Inventario de tablas de una regional (la regional asociada a la BD muestra)

Inventario de tablas de la nacional

Crear el modelo entidad - relación (BD muestra)

Identificar módulos del sistema para actualización de tablas

Inventario de campos (BD muestra)

Análisis de la problemática

Construcción de diccionario de datos

Definición de formatos para captura de diccionario

Llenado de formatos

Documentar diferencias en esquema de BD

Identificar instancias, dbspace, BD's entre varias locales regionales y nacional

Comparación de estructuras entre varias locales

Comparación de estructuras entre varias regionales

- Comparación de estructuras entre local y regional
- Comparación de estructuras entre regional y nacional
- Crear matriz tabla vs nivel (local, regional y nacional)
- Inventario y análisis de catálogos (en código, en papel y reales)
 - Inventario de catálogos reales (locales, regionales, nacionales)
 - Identificar fuentes y métodos de actualización
 - Identificar catálogos por homologar o reestructurar
 - Inventario de catálogos en código y papel
 - Identificar fuentes y métodos de actualización
 - Identificar versiones por homologar o catálogos por crear
- Identificar campos y registros en conflicto (movimientos históricos, multi campos, etc.)
 - Identificar campos que no concuerden con catálogos (tablas locales, regionales y nacional)
 - Identificar registros inconsistentes (padres sin hijos, hijos sin padre)
 - Validaciones numéricas (pagos vs totales etc.)
- Duplicidades o faltantes
 - Identificar registros en local que no existen en regional y viceversa
 - Identificar registros en regional que no existen en nacional y viceversa
 - Pruebas de duplicidad en local
 - Pruebas de duplicidad en regional
 - Pruebas de duplicidad nacional
- Análisis reglas de validación actuales
 - Identificación de causas que originan errores o inconsistencias
 - Identificación de diferencias en interpretación de valores
- Análisis de otros padrones para cruce y/o complemento de datos**
 - Análisis de calidad de información
 - Análisis de campos por comparar y agregar
- Definición de Metas de acuerdo a las necesidades de La Compañía y al resultado del análisis**
 - Presentación de resultados
 - Definición de prioridades en datos y tiempo
- Definición de procedimientos de depuración**
 - Propuesta de Rediseño de BD y catálogos
 - Procedimientos de mantenimiento (nuevas reglas de validación)
 - Procedimiento de sincronización con BD actual
 - Procedimientos para comparación y complemento (otros padrones)
 - Propuesta para homologación de las BD's
 - Impacto a la operación actual (módulos y operación)
 - Procedimientos para limpieza, normalización y verificación
 - Procedimientos de auditoría de movimientos y contingencia
- Definición de procedimientos de integración**
 - Procedimientos de mantenimiento (nuevas reglas de actualización)
 - Propuesta de plan de integración (integración escalonada)
 - Niveles de calidad de la información (inconsistencias, local mata reg, fuentes externas, etc.)
 - Impacto a la operación actual (módulos y operación)
 - Procedimientos para integración
 - Procedimientos de auditoría de movimientos y contingencia
- Limpieza de datos**
 - Programación
 - Pruebas
 - Plan de limpieza
- Compactación de clientes**

Diagnóstico

El objetivo principal de la limpieza de datos es generar una base de datos de clientes que garantice información consistente y confiable en todos los niveles para facilitar la operación y la toma de decisiones.

Dado que el diseño e implantación del nuevo sistema es modular y dada la magnitud del problema que representa la limpieza de una base de datos tan grande, se tomó la decisión de generar diagnósticos y propuestas de limpieza escalonadas y sobre ciertos segmentos de acuerdo a los módulos que se van liberando de manera que la tarea sea realizable con la calidad esperada y en los tiempos requeridos.

El registro de clientes es el primer módulo que se deberá implementar, de modo que las tablas relacionadas con este son las primeras que se depurarán.

El catálogo de clientes es el eje de todas las operaciones que se realizan en el proceso de recuperación.

Depurar el catálogo de clientes, implica afectar de alguna manera, a todas las tablas de la base de datos del sistema.

El proceso de limpieza de una base de datos se forma, básicamente, de 4 grandes tareas:

- Identificar datos incorrectos, inconsistentes e incompletos.
- Identificar las causas que originan dichos errores.
- Definir acciones para garantizar su corrección.
- Definir acciones para garantizar su mantenimiento.

En el caso concreto de este proyecto, tenemos además otras 2 grandes tareas:

- Garantizar la migración a un nuevo modelo de datos.
- Garantizar la convivencia de ambos sistemas y modelos.

Dada la magnitud del problema sabemos que esta limpieza no se puede realizar de un día para el otro, será necesario realizar limpiezas locales y después generales, compactaciones locales y después generales.

Esto implica que en este proceso de limpieza será necesario crear algunos procedimientos que aseguren la convivencia de datos y modificar algunos procesos del sistema actual que aseguren el mantenimiento de la limpieza.

Estructura y procesos

Para preparar este diagnóstico se realizaron, básicamente, 7 tareas:

- Análisis general de las estructuras de datos.
- Identificar tablas y procesos del sistema actual que intervienen en el módulo de registro.
- Preparación y llenado de los formatos para el diccionario de datos.

- Validación con los responsables de sistemas de la información recabada.
- Diagnóstico preliminar con bases de datos de prueba.
- Validación de resultados con funcionarios de La Compañía.
- Diagnóstico final con bases de datos reales.

Análisis general de las estructuras de datos

El objetivo de esta tarea es conocer de manera general las estructuras de datos que utiliza el sistema actual.

Existen 3 diferentes estructuras de datos, una a nivel local que se replica casi idéntica en todas las oficinas locales, otra a nivel regional que es una copia de la estructura nacional mas algunas tablas utilizadas en procesos exclusivamente regionales y otra nacional.

Total de tablas para las 3 estructuras

En la estructura Local se tienen 304 tablas
 En la estructura Regional se tienen 155 tablas
 En la estructura Nacional se tienen 133 tablas

Tablas compartidas por diferentes estructuras

Local y Nacional	14 tablas
Local y Regional	12 tablas
Regional y Nacional	13 tablas
Local, Regional y Nacional	26 tablas

Tablas exclusivas de una estructura

Local	252 tablas
Regional	104 tablas
Nacional	80 tablas

Algunas de estas tablas en realidad son la misma pero tienen distintos nombres y pequeñas variantes en cuanto a sus campos.

Identificar tablas y procesos del sistema actual que intervienen en el módulo de registro

Dentro de las tablas que forman la base de datos del sistema actual se encuentran varias que identifican los procesos y tablas que se utilizan en cada módulo del sistema.

El módulo de registro utiliza (en sus 3 instancias) 46 tablas

- 21 tablas de datos
- 25 catálogos de validación

Son 11 las mas importantes del módulo y el resto son de control, consulta e impresión.

bajas	Datos de clientes que se dan de baja
cliente	Datos que identifican a clientes
cedulas	Control de documentos expedidos
documentos	Tabla de documentos (trámites de un cliente)
externos	Documentos de clientes que se iniciaron en otra oficina local
liquidacion	Datos sobre clientes liquidados y quien los liquidó
nombres	Datos de cliente que cambió de nombre
credito	Tabla de créditos - cliente
representa	Datos que identifican a representantes legales
sucesion	Datos de clientes en sucesión
suspension	Datos de clientes en suspensión

Preparación y llenado de los formatos para el diccionario de datos

Para el diccionario de datos se documentaron 3 grandes tipos de datos

- Tablas
- Indices
- Campos

Para Tablas se documentaron los siguientes datos

- Nombre
- Descripción
- Tipo (datos, paso, catálogo)
- Longitud del registro
- No. de campos
- No. de índices únicos
- No. de otros índices
- Instancia en la que aparece (Nacional, Regional, Local)
- Módulos del sistema actual que la utilizan
- Comentarios generales y problemas
- Propietario
- Responsable de actualización (en caso de catálogos, que área es la responsable del mantenimiento)

Para Indices se documentaron los siguientes datos

- Nombre del índice
- Tabla asociada
- Tipo (único, secundario)
- Campos que lo forman
- Propietario

Para Campos se documentaron los siguientes datos

- Tabla asociada
- Nombre del campo
- Tipo
- Descripción
- Tabla para validación (aunque no se utilice, tabla que se puede usar para validación)
- Problemas
- Se usa (si o no)
- Tipo de validación (catálogo, interna, ninguna, procesos externos)
- Tiene otro uso (si o no)
- Que otro uso
- Valores que toma
- Valor de default

En general se podría decir que la base de datos tiene 5 grandes tipos de problemas

- Datos Inconsistentes
- Datos Incompletos
- Datos Incorrectos
- Clientes Duplicados
- Problemas de estructura

Datos Inconsistentes

Causas

- Campos redundantes en tablas
- Falta de integridad referencial
- Falta de llaves foráneas
- Falta de índices
- Procesos incorrectos e incompletos
- Valores de default inexistentes
- Mecanismos de corrección (modificación manual de los datos)

Ejemplos

- Una oficina local pertenece a varias regionales
- Clientes en bajas que están activos
- Falta de datos históricos de un cliente que se movió o viene de otra oficina local
- Clientes sin créditos y viceversa

Datos Incompletos

Causas

- Falta de validación y derivación de datos
- Falta de requisitos y fuentes confiables de información

- Falta de ayudas al capturista
- Procesos incorrectos e incompletos

Ejemplos

- Código postal no corresponde ni al estado, ni a la regional del domicilio
- Clientes sin domicilio ni teléfono
- Créditos sin fecha de expedición

Datos Incorrectos

Causas

- Falta de validación y derivación de datos
- Validaciones incorrectas
- Mecanismos de corrección (modificación manual de los datos)
- Procesos incorrectos o incompletos

Ejemplos

- Clientes de otra regional o local
- Nombres de representantes, liquidadores, albaceas etc. sin sentido ni forma de localización
- Clientes con créditos que no les corresponden o que les faltan
- Fechas de alta 1999 o posteriores a la fecha actual

Clientes Duplicados

Causas

- Nula validación de clientes por nombre o domicilio
- Procesos incorrectos o incompletos
- Algoritmo deficiente para generación del No. cliente

Ejemplos

- Mismo cliente con diferentes fechas de nacimiento o registro
- Diferencias en algunas letras
- Mismo nombre, mismo domicilio y diferente No. de cliente

Problemas de estructura

Uno de los principales problemas de la base de datos es el diseño de la estructura:

- Campos redundantes
- Falta de índices adecuados
- Falta de valores de default
- Falta de integridad referencial
- Falta de llaves foráneas
- Falta de catálogos

A continuación, presentamos algunos ejemplos de estos problemas

Tabla	Problema
Locales	No tiene índice
Tipo_cte	No contiene los valores que existen en el campo, las descripciones son iguales en un 95%
Municipios	No tiene índice. Sólo marca rangos. No corresponde a los números oficiales de SEPOMEX.
Módulos	En el índice se incluye un campo que se interpreta de diferentes formas en función del valor de otro campo.
Cliente	No todos los números de cliente se generaron con el algoritmo actual, unos por viejos y otros por algoritmo diferente. Sólo tiene un índice. Campos redundantes.
Cliente	No tiene valores de default y al agregar nuevos clientes quedan incorrectamente registrados
Documentos	No están todos los trámites y requiere de tablas adicionales dado que tiene un campo secuencial. Sólo tiene un índice que es extremadamente largo

Ejemplos de campos que no cuentan con un catálogo de validación, las validaciones se realizan por programa o simplemente no se valida.

Tabla	Campo	Tipo	Descripción
cliente	cau_movs	char(1)	Identificación de estatus del cliente
cliente	cau_secgob	integer	Sector gobierno
cliente	cau_tipcont	char(1)	Tipo de cliente
cliente	cau_tipsoc	smallint	Tipo de sociedad

Consecuencias

- Clientes no controlados
- Clientes mal controlados
- Procesos muy tardados
- Poco a poco se deteriora mas la información
- Clientes no localizados
- Gran cantidad de información irrelevante
- Clientes duplicados

Diagnóstico preliminar con bases de datos de prueba

Para este análisis se utilizaron 2 distintas muestras, la primera es una extracción de la base de datos nacional (sólo las tablas principales de Registro de clientes y algunos catálogos) con registros de la oficina local de Oaxaca. La segunda no es una muestra sino una copia de la base de datos de la oficina de Pachuca que utiliza el área de sistemas para pruebas.

Estas muestras presentan los siguientes problemas:

- La muestra de Oaxaca no está completa en todos sus segmentos de manera que las relaciones entre tablas no es real.
- La base de Pachuca es la que se utiliza para pruebas y tiene mucha basura que se incluye para realizar las mismas pruebas.

En este análisis se consideraron todos los clientes, activos y no activos.

TAMAÑO DE LA MUESTRA

Tabla	Oaxaca Regs.	Pachuca Regs.
Bajas	968	19,722
Cliente	10,000	164,827
Cedulas	6,411	102,676
Documentos	26,037	395,468
Liquidación	11	122
Nombres	314	3,207
Credito	27,590	425,356
Representantes	1,485	7,914
Sucesión	1	111
Suspensión	2,645	45,602
Externos	0	9,736

Dado que la Base de Datos de Pachuca es la que se utiliza para pruebas de operación y desarrollo, no pudimos crear índices o tablas adicionales, de manera que algunas de las pruebas que quisimos realizar se hicieron en la base pequeña.

TABLA DE CLIENTES

Esta tabla guarda los datos que identifican a un cliente, uno de sus principales problemas es la falta de índices e integridad pues sólo cuenta con el índice de No. de cliente.

Muchas de las tareas de cobranza se relacionan con el tipo de cliente, en la cobranza se ejecutan búsquedas por tipo de cliente, tipo de sociedad, calidad del cliente y actividad económica, estas búsquedas resultan muy lentas (secuenciales) por la falta de índices.

En esta tabla hay varios campos que se validan internamente, por lo tanto cada vez que exista un nuevo valor será necesario recompilar.

Nombre o Razón social

Personas físicas sin división de apellidos y nombre (sin el separador “/”)

Oaxaca 217 clientes (22.41%) sin división de apellidos

Misma persona con diferente homoclave o nombre

Para este tipo de diagnósticos se emplean métodos muy sofisticados y complejos que no se utilizaron en este momento, se seleccionaron las personas con la misma fecha de nacimiento, nombre parecido y dirección similares, a continuación mostramos algunos ejemplos

No. cliente	Razón Social
3U7	ASOC AGRICOLA LOCAL EL MORRO
842	ASOCIACION AGRICOLA LOCAL EL MORRO

C96	ABAR LA SOLEDAD SA
P39	ABARROTOS LA SOLEDAD SA
3R7	AUTOTRANSPORTES TRANSISTMICA SA DE CV
DK4	AUTOTRANSPS TRANSISTM SA CV
F46	AGRICOLA TOCUELA SOCIEDAD DE PRODUCCION RURAL DE RESPONSABILIDAD ILIMITADA
TF5	AGRICOLA TOCUELA S DE PR DE RI
5L5	BIENES COMUNALES SN SEBASTIAN YUTAMINO CAUACUA
IY4	BIENES COMUNALES SAN SEBASTIAN YUTANINO CAHUACUA SOLA OAX
9E0	COLEGIO DE INGENIEROS CIVILES DE OAXACA A C
9P9	COLEGIO DE INGENIEROS CIVILES DE OAXACA AC
3P4	ESTUDIOS Y SISTEMAS DE INGENIERIA Y ARQU ITECTURA SA DE CV
NT2	ESTUDIO Y SISTEMA DE INGENIERIA Y ARQUITECTURA SA DE CV
B60	ALVARADO/ABAD/GUILLERMO
IF5	ALVARADO/ABAD/GUILLERMO

Misma persona con diferentes fechas de nacimiento o creación

Para este análisis seleccionamos clientes igual, nombre igual y domicilio igual. Aquí presentamos algunos ejemplos.

Nombre	fecha 1	fecha2
APROVECHAMIENTO FORESTAL SAN SEBASTIAN	881209	881219
ASOCIACION GANADERA EJIDAL	870426	870616
ASOCIACION GANADERA EJIDAL	870426	870921
ASOC GAN LOC	550524	550526
AMA INGENIERIA ESPECIALIZADA SC	830104	830311
ARMONT MEXICANA SA CV	790622	830831
ADMINISTRADORA MONTAJES Y MAQUINARIA	720628	800801
AGENCIA MUNICIPAL	840220	871208
ABAR LA SOLEDAD SA	720405	720505
BARDACO SA CV	830202	830209
BIOXON DE MEXICO SA CV	800101	800310
COMITE DE FESTEJOS DEL LUNES DEL CERRO	840715	840730
AAMF ALVARADEJO/MU&OZ/JOSE FRANCISCO	540403	540409

Espacio insuficiente para el nombre, se utilizan abreviaturas, diferentes formas de escribir el tipo de sociedad.

Campo Domicilio, CP, municipio y servicio postal

El principal problema detectado en estos campos es la falta de validación.

C.P. inexistentes en el registro oficial

CP no encontrado en	Oaxaca	Pachuca
	1.88 %	0.77 %
Sin CP en	Oaxaca	Pachuca
	1.71 %	1.5 %

En la base de Oaxaca no se encontró basura sin embargo en la de Pachuca encontramos 67 CP que tienen basura.

El municipio no coincide con el oficial

Municipio no coincide en	Oaxaca	Pachuca
Físicas	59.69 %	0.79 %
Morales	24.86 %	

El C.P. indica que se trata de una colonia y el campo de colonia está vacío

No tienen colonia en	Oaxaca
Físicas	10.82 %
Morales	10.66 %

El nombre de la calle es domicilio conocido

No tienen calle en	Oaxaca
Físicas	3.62 %
Morales	2.29 %

No tiene No. interior ni exterior

Sin número en	Oaxaca
Física	15.25 %
Moral	8.52 %

No tiene teléfono

Sin teléfono en	Oaxaca
Física	66.04 %
Moral	27.42 %

No se incluye el estado en el texto de la dirección por falta de espacio

Sin estado en	Oaxaca
Física	14.87 %
Moral	6.54 %

El estado no coincide con el oficial

Encontramos 16 clientes en Oaxaca que el estado no coincide con el oficial

Campos Clave de Regional, local y módulo

Estos campos tienen muchas inconsistencias, supuestamente el campo de módulo es el que define los otros, pero en las tablas encontramos muchos problemas, de hecho el tipo de campo en cliente no coincide con el del catálogo. Existe cualquier combinación de inconsistencias posible

Campo actividad económica

Existen varios catálogos de actividades económicas en el mercado que permiten segmentar a las personas; este no se relaciona con ninguno de ellos.

Clientes sin actividad económica

Pachuca	54 clientes
Oaxaca	81 clientes

Claves de actividad económica inexistente en el catálogo

Pachuca	1 clientes
Oaxaca	1 clientes

Campo fecha de alta

Este campo incluye fechas de alta desde 01/01/1901 hasta 21/10/1999

Clientes sin fecha de alta

En Pachuca encontramos únicamente 10 cliente sin fecha de alta.

Campo motivo de alta

En Pachuca encontramos únicamente 10 registros sin motivo de alta (los mismos que no tienen fecha)

Campo tipo de sociedad

Este campo identifica a los clientes que sobrepasan cierto límite de ingresos. Este es actualizado por procesos centrales en base a los ingresos que se calculan de los clientes con actividad empresarial y cargado directamente a las tablas. Este campo es fundamental para la cobranza. Problemas con los nulos.

Campo tipo de cliente

Este campo indica el tipo de cliente y su régimen. Este campo también es fundamental para la cobranza y actualizado por procesos centrales en base a los ingresos y a las actividades de los clientes. Tiene problemas muy serios con los nulos, los programas de cobranza esperan que todos los campos estén "llenos" de manera que aquellos que contengan caracteres nulos no entran en el proceso de cobranza. Existen actualmente en la base de Pachuca 32,330 clientes que no se están considerando en el proceso por este problema.

Campo de estatus del cliente

Este campo indica el estatus actual del cliente y es fundamental para todo el sistema de recuperación. Problema con nulos.

Este campo presenta un problema muy particular, si un cliente tiene estatus de activo, efectivamente está activo, pero, si el estatus refleja cualquier otro tipo, en realidad no se sabe cual es, para conocer el estatus real del cliente es necesario buscar en todos los segmentos.

TABLA DE BAJAS

Esta tabla guarda a los clientes dados de baja, y sólo cuenta con un índice (No. de cliente y Determinante) de manera que cualquier búsqueda por motivo o fecha de baja toma mucho tiempo. Aunque existe la tabla de motivos, el campo de motivo de baja se valida internamente por programa esto implica que cada vez que se agregue un nuevo motivo de baja será necesario recompilar el programa que realiza esta validación.

Clientes que se encuentran registrados en bajas y que no existen en cliente.

Esto se debe a varios procesos que eliminan al cliente y no actualizan adecuadamente todas las tablas, por ejemplo, cuando un cliente se mueve de una local a otra, su registro de clientes es borrado y no se borra su registro de bajas, otra causa es la de los homónimos, cuando un cliente se detecta que es homónimo su registro de clientes es borrado para modificar el nombre y su registro de bajas no es actualizado.

Oaxaca	11.98%
Pachuca	0.05 %

Clientes en bajas y estatus diferente en cliente

	Oaxaca
ACTIVO	11.98 %

Fechas de baja 01/01/01

Oaxaca	9
Pachuca	73

Clientes en bajas con cambio de nombre y No. de cliente anterior activo

Oaxaca	2
Pachuca	2

Clientes en bajas con cambio de nombre y el nuevo No. de cliente activo

Oaxaca	7
Pachuca	58

Clientes cuya fecha de baja en menor que la fecha de baja de sus créditos

Oaxaca	167 clientes	17.25% con fechas de baja menor
--------	--------------	---------------------------------

Clientes sin registro de operaciones o que su última operación no es baja

Sólo 543 clientes (56.09%) de Oaxaca tienen registro de operaciones

Clientes en bajas y en liquidación

Pachuca	11 clientes
---------	-------------

Clientes en bajas y en sucesión

Pachuca	1 cliente
---------	-----------

Clientes en bajas y en suspensión

Pachuca	344 clientes
---------	--------------

Clientes en bajas con créditos activos

Oaxaca	6 clientes
--------	------------

Pachuca	9 clientes
---------	------------

Las tablas de LIQUIDACION, SUCESION y SUSPENSION tienen este mismo tipo de problemas además del manejo del nombre y registro del responsable, estos nombres en general presentan los problemas que se tratarán adelante en el análisis de la tabla REPRESENTA

TABLA créditos

En esta tabla se encuentran registrados los créditos de los clientes.

Claves de crédito sin registro en catálogos

	Créditos	Clientes
Oaxaca	1	5
Pachuca	4	48

Personas morales con créditos de físicas y viceversa

	Oaxaca	Pachuca
Morales con créditos de físicas	0	14
Físicas con créditos de morales	4	19

Clientes sin créditos y créditos sin cliente

	Pachuca
Créditos sin cliente	44
Clientes sin crédito	468

Clientes con créditos incompatibles

Oaxaca	170 clientes
Pachuca	1315 clientes

Clientes con créditos en desuso

Oaxaca	8 clientes y todas sus créditos cancelados
Pachuca	9 clientes con 7 créditos activos y 2 cancelados

TABLA CEDULAS

En esta tabla se registra la documentación que se han impreso. En el sistema actual no se permite que un cliente tenga mas de un documento activo asociado al mismo crédito, si algún cliente solicita una reposición, la anterior se cancela automáticamente.

Clientes con mas de 1 documento activo

Oaxaca	13 clientes
Pachuca	176 clientes

Clientes en baja con documentos activos

Oaxaca	2156 clientes con 2223 documentos activos
Pachuca	28558 clientes con 28619 documentos activos

Documentos sin cliente

Oaxaca	0 documentos sin cliente
Pachuca	969 documentos sin cliente 104 canceladas las demás activas.

TABLA REPRESENTANTE

En esta tabla se guardan los datos (nombre y domicilio) del representante legal de un cliente.

Aunque en esta tabla existen los campos de Domicilio y teléfono del representante, no se capturan.

Representantes que no tienen Cliente

Oaxaca	221 representantes sin cliente
Pachuca	41 representantes sin cliente

En general los nombres del representante están escritos como apellido paterno, materno, nombres pero hay muchos casos en los que el nombre está escrito en otro orden o se vuelve a escribir el No. de cliente en su lugar.

TABLA DOCUMENTOS Y EXTERNOS

En esta tabla se registran todas las operaciones de un cliente. Dado que en el índice se utiliza un número serial, para recibir operaciones de clientes que vienen de otra local se emplea una tabla alterna en la que se insertan estos registros, esta tabla es externos. Dado que la cobranza de créditos generalmente ocurre desfasada en el tiempo, aunque un cliente esté activo hay que verificar que estatus tenía en el período que se vigila, para esto se utiliza la tabla de documentos y si existe una reactivación se verifica en que momento ocurrió para definir si debe ser validado. Esto sólo se verifica en la tabla de documentos y no en la de externos.

Documentos sin cliente

Oaxaca	581 clientes con 3049 operaciones
Pachuca	490 clientes con 646 operaciones

Externos sin cliente

Oaxaca	--
Pachuca	112 clientes con 646 operaciones

Clientes sin documentos ni externos

Oaxaca	1806 clientes sin documentos
Pachuca	7354 clientes sin operaciones ni en documentos ni en externos

Clientes en externos que están en documentos

De los 4405 clientes en Pachuca con registro en externos 188 tienen operaciones registradas en documentos.

Error en fechas de operación (especificada por el cliente)

Oaxaca	8 registros indican fecha de 01/01/01 1 registro indica fecha>hoy
Pachuca	5 registros indican fecha de 01/01/01 55 registros indican fecha>hoy

Análisis del diagnóstico

Este trabajo se analizó con 2 grupos de la Compañía, en estas reuniones se aclararon puntos como los que a continuación se mencionan:

La información deberá referirse a datos reales.

La información deberá expresarse en términos del estatus del cliente.

Cuando la información se refiera a clientes en bajas, deberá referirse exclusivamente a este segmento y no a los clientes suspendidos o en inicio de liquidación o en inicio de suspensión.

También se solicitó se agregaran algunas consultas y se eliminaran otras.

Se preparó un nuevo juego de consultas y un programa que se envió a varias oficinas locales para después reunir los resultados y preparar el nuevo diagnóstico. Todo esto se preparó con el fin de tener una primera aproximación de los volúmenes de datos y los tiempos de respuesta esperados.

Diagnóstico final con bases de datos reales

Se prepararon 41 consultas distintas a las principales tablas de REGISTRO

CLIENTES	18 CONSULTAS
BAJAS	6 CONSULTAS
CREDITOS	5 CONSULTAS
CEDULAS	4 CONSULTAS
REPRESENTANTE	3 CONSULTAS
DOCUMENTOS	5 CONSULTAS

Todos los resultados se expresan en función del estatus del cliente. Este campo en realidad no está actualizado adecuadamente, si el estatus indica que el cliente NO ESTA activo en realidad no se puede saber si está en bajas o liquidado o suspendido etc. , lo que si sabemos es que no está activo.

El tiempo aproximado de ejecución del diagnóstico fue de 3 horas.

El programa fue transferido el jueves 3 de julio a las siguientes oficinas locales

SAN LUIS POTOSI
DEL SUR DEL D.F.
PUEBLA
GUADALUPE

TIJUANA
CHIHUAHUA
TLAQUEPAQUE
OAXACA

Dado que las bases de datos no cuentan con índices adecuados para el tipo de consultas que pretendíamos realizar, fue necesario crear algunos que permitieran que el proceso se concluyera en una noche y no interfiriera con el proceso normal del sistema actual. En la oficina de TLAQUEPAQUE el proceso fue interrumpido y las últimas consultas (sobre duplicados) no se concluyeron.

En las oficinas del sur del D.F. y CHIHUAHUA el proceso no fue ejecutado.

En la mayoría de los equipos el programa se ejecutó ese mismo día, en otras se retrasó debido a otros procesos de mayor importancia.

CLIENTES

Se prepararon 18 consultas a esta tabla sobre los campos mas importantes.

- Estatus de clientes
- Personas físicas sin separador “/”
- Clientes con & en el No. de cliente
- Personas morales duplicadas por nombre
- Clientes sin código postal
- Un código postal a diferentes módulos
- DOMICILIO CONOCIDO en la calle
- Inconsistencias de módulo, local y regional
- Totales por local y otras locales
- Totales por regional y otras regionales
- Clientes sin actividad o no en catálogo
- Clientes con actividades no especiales
- Clientes sin fecha de alta o fecha >hoy
- Clientes menores de 18 años
- Clientes con calidad de cliente nula
- Clientes con tipo de sociedad nula
- Clientes con tipo de cliente nulo o mal
- Clientes con sector gobierno nulo

BAJAS

Se prepararon 6 consultas sobre algunos campos de esta tabla y su relación con otras tablas.

- Total de bajas por tipo de baja
- Bajas sin cliente

- Clientes activos en bajas
- Clientes en bajas y liquidación
- Clientes en bajas y sucesión
- Clientes en bajas y suspensión

CREDITOS

Se prepararon 5 consultas sobre la tabla de créditos y su relación con la tabla de clientes

- Total de créditos por clave
- Clientes sin créditos
- Créditos sin cliente
- Créditos activos de clientes en baja
- Créditos con fecha de baja mayor a la baja del cliente

CEDULAS

Se prepararon 4 consultas sobre la tabla de cédulas y su relación con cliente

- Total de cédulas por estatus
- Clientes con mas de una cédula activa
- Clientes en bajas con cédulas activas
- Cédulas sin cliente

REPRESENTANTE

Se prepararon 3 consultas sobre el nombre del representante legal y su relación con cliente

- Total de representantes por tipo de cliente
- Representantes cuyo nombre es un No. de cliente
- Personas morales sin representante legal

DOCUMENTOS Y EXTERNOS

Se prepararon 5 consultas sobre documentos y externos y su relación con cliente

- Total de Documentos por tipo de operación
- Total de externos por tipo de operación
- Documentos sin cliente
- Externos sin cliente
- Clientes sin documentos ni externos

Propuesta de limpieza

Introducción

Una vez analizados los resultados del segundo diagnóstico, en el grupo de limpieza de datos nos dimos a la tarea de preparar una primera versión de los alcances que se pretenden en la limpieza de datos para el módulo de registro de clientes.

La primera tarea será la validar los casos que se deberán atacar para después determinar aproximadamente el número de casos que se limpiarán para cada uno de los puntos acordados. Esto permitirá estimar tiempos de trabajo y personal necesario para la depuración.

Acciones

Nuestra propuesta de limpieza incluye 4 grandes tipos de acciones

Acciones previas a la limpieza

- Creación de tablas necesarias para la limpieza y normalización
- Normalizar y completar nombres y domicilios
- Ajustes al Sistema actual
 - Movimientos de confirmación a y de la nacional
 - Cambio de domicilio
 - Cambio de nombre
 - Captura de domicilios contra catálogos de SEPOMEX
- Confirmación de movimientos no operados por la nacional
- Confronta nacional de paternidad de clientes
- Identificar datos que requieren investigación en expedientes
- Definir y crear base histórica (no depurada) para clientes inactivos

Acciones de limpieza local

- Registro de todas las modificaciones que se realicen en la limpieza
- Procedimiento de limpieza para domicilios
- Reasignación de clientes a la oficina local que realmente le corresponda
- Solución definitiva a clientes con cambio de nombre
- Actualización del estatus del cliente con base en los segmentos
- Limpieza de la oficina local
- Actualización de locales con información de nacional
- Compactación de duplicados a nivel local

Acciones de limpieza nacional

- Solución a conflictos entre locales y nacional
- Actualización de nacional con información de locales
- Compactación de duplicados a nivel nacional

Acciones de validación y carga de limpieza externa

Preparación de programas para carga de datos limpios a la tabla de limpios y su mapeo a la Base de Datos del sistema actual

Validación de datos de acuerdo a los criterios de calidad establecidos por el grupo

Acciones previas a la limpieza

Creación o actualización de tablas necesarias para la limpieza y normalización

Crear

Régimen de capital

CP vs oficina local

Equivalencia para créditos en desuso o reemplazo

Créditos padre

Créditos por tipo de calidad de cliente

Tipo de sociedad (cau_tipsoc)

Tipo de cliente (cau_tipcont)

Estatus del cliente (cau_movs)

Nuevos catálogos (equivalencia para migración)

Equivalencia para módulos con clave anterior

Motivos de cédulas (ced_mot)

Tipo de clientes especiales (cau_calcont)

Actualizar

Créditos incompatibles

Claves de movimiento

Claves de créditos

Catálogos de módulos

Código (SEPOMEX) agregar clave de población y oficina local asociada

Igualar

Cat_aff.- Catálogo de oficina local

Cat_actvi .- Catálogo de actividades preponderante.

Cat_edo .- Catálogo de entidades federativas

Cat_mov .- Catálogo de claves de movimiento

Cat_mpio .- Catálogo de municipios

Normalizar y completar nombres y domicilios

- Tabla con marca de clientes especiales
- Crear programas para validación ocular
- Crear programas para validaciones en lote
- Preparar tablas y programas para carga datos limpios, sucios
- Para nombres que sobrepasan las 80 posiciones se conservará el nombre original en el sistema actual y se migrará el nombre "limpio" siempre y cuando no existan movimientos de cambio para ese registro
- Solucionar problemas de datos incorrectos
- Se migrarán al campo domicilio viejo
- Dejar la información disponible para búsquedas por nombre
- Detección de duplicados

Ajustes al sistema actual

- Configuración de servidores, Informix y terminales para captura de Ñ
- Captura de domicilios con validación en catálogos de SEPOMEX
- Guardar clave de población en tabla alterna para facilitar la migración
- Modificar estructura de base de datos actual
- Incorporar valores de default
- Creación de índices adicionales
- Integridad referencial y llaves foráneas
- Años de 4 dígitos
- Definir espacio y tablas para limpieza y nuevas tablas
- Datos de fusión y escisión
- Tablas de equivalencia
- Bitácoras
- Replicar datos entre oficina local , oficina regional y nacional aprovechando los recursos de INFORMIX
- Modificar procedimientos de confirmación de la nacional para eliminar la pérdida de información
- Errores de lógica
- MC domicilio
- Modificar procedimientos del sistema actual para registro de fusión y escisión
- Modificar procesos del sistema actual que "ensucien" nuevamente la BD
- Cambio de nombre y cambio de domicilio

Confirmación de movimientos no operados por la nacional

Confronta nacional de clientes para detectar duplicados y errores de paternidad

Definir programa para respaldo y restauración de BD locales

Identificar y corregir datos que requieren investigación en expedientes

Generar reportes y programas para actualizar datos de este tipo de clientes por parte de Hacienda

Seleccionar reporteador y front end

Clientes sin créditos y viceversa

Clientes del siglo pasado o fechas mayores a hoy

Clientes con más de una cédula activa

Base histórica (no depurada) para inactivos

Definición de segmentos o tipos de clientes que serán eliminados de la limpieza para crear una base de datos histórica de consulta

Definición de procedimientos para la consulta de dichos segmentos

Propuesta por parte del grupo de limpieza

En todos los procesos de limpieza se podrían eliminar a los clientes en el segmento de bajas con motivo:

DEFUNCION, FUSION, ESCISION

LIQUIDACION TOTAL DEL ACTIVO O CANCELACION

CESACION TOTAL DE OPERACIONES

Siempre y cuando los datos del cliente no tengan inconsistencias entre segmentos o instancias, no tengan créditos, no tengan ningún trámite vigente, no hayan presentado pagos y por lo menos 5 años en el estatus actual.

Todos los clientes que tengan estas características serán eliminados de la BD local y se conservarán en una base histórica en la nacional.

En el proceso de limpieza serán considerados todos los clientes de la base de datos, es decir, el proceso no eliminará a estos clientes, esta actividad deberá realizarse con anterioridad.

Acciones de limpieza local

Registro de todas las modificaciones que se realicen en la limpieza

Crear una bitácora de operaciones realizadas a la base de datos de manera que siempre se pueda "deshacer" cualquier proceso y conocer con exactitud que y como se modificó algún dato.

Procedimientos de limpieza para domicilios

Sincronizar campos relacionados con domicilio en función de CP y calle para clientes poblaciones, municipios, ciudades, estado, módulo, oficina local , oficina regional

Identificar clientes que deben cambiar de oficina local

Identificar clientes con domicilios erróneos

Reasignación de clientes a la oficina local que realmente le corresponda

Preparar programas para migrar clientes a la oficina local que le corresponde

Solución definitiva a clientes con cambio de nombre

Se agregan a la tabla de cambios de nombre todos los nombres de clientes con cambio

Conservamos sólo el último nombre activo

Se eliminan los registros de clientes con baja por cambio de nombre

Se reemplazan todos los registros de todas las tablas con referencia a los No's anteriores por el último

Se elimina el motivo cambio de nombre como motivo de baja

Actualización del estatus del cliente en base a los segmentos

Identificar segmentos a los que pertenece el cliente

Definir prioridades para definir el estatus final

Actualización de estatus

Eliminar o agregar referencias a segmentos

Definición de prioridades de calidad de información y corrección de la base de cada oficina local

Definir reglas y prioridades sobre la calidad de la información a nivel local

Limpieza a nivel local

Actualización de locales con información de nacional

Completar la información de la local con datos de la nacional

Sólo los datos que se diagnostique que la nacional tienen prioridad sobre la local

Compactación de clientes a nivel local

Definir los niveles y prioridades de calidad para la compactación para clientes en la misma local

Procesos individuales para todos los campos de las tablas

Estos procesos se aplicarán para cada campo de cada tabla con el fin de lograr la completa sincronización de la base local con la nacional.

Definición del orden en que se depurarán los campos.

Comparación de local con nacional a nivel registro y campo.

Definición de prioridades y criterios de calidad de información para sincronización en local y nacional.

Identificar todas las posibilidades de inconsistencias y sus posibles correcciones para definir los procesos automáticos o manuales para hacer la corrección.

Valores distintos en local y nacional, nulos o fuera de catálogo.

Registros inexistentes en local o nacional.

Clientes duplicados en otra local.

Acciones de limpieza nacional

Solución a conflictos entre locales y nacional

Corrección de clientes que se encuentran en la nacional y no pertenecen a ninguna local y viceversa

Corrección de valores inconsistentes entre ambas estructuras

Actualización de nacional con información de locales

Completar la información de la nacional con datos de la local

Completa sincronización de locales con nacional

Compactación de clientes

Definir los niveles y prioridades de calidad para la compactación

Acciones de validación y carga de limpieza externa

Normalizar nombres para las personas físicas y morales matrices

Corrección de ortografía

Eliminar abreviaturas

Separar en campos

Separar el régimen de capital

Detección de duplicados

Normalizar domicilios y teléfonos de todos los clientes

Separar en campos

Verificación y corrección de domicilios

Verificación y corrección de oficina local , oficina regional

Eliminar espacios y val(teléfono)=0

Localización de no localizados

Detección de duplicados

Complementar datos de clientes incompletos o no localizados

Domicilios y teléfonos de clientes

Actividades económicas

Datos generales y socioeconómicos

Después de “normalización”

Recálculo del No. de cliente (auxiliar)

Se identifican personas con cambio de No. por cambio de nombre

Se identifican números con dígito verificador mal calculado
Identificar clientes que se mueven de oficina local
Identificar domicilio no correctos
Identificar clientes duplicados

Formato de regreso de normalización

Razón social
Régimen de capital
Ap. paterno
Ap. materno
Nombre
Sexo
Fecha de nacimiento o creación de la empresa
Calle
Entre calle1
Entre calle2
No. interior
No. exterior
Población o colonia
Municipio o delegación
Ciudad
Estado
CP
Oficina local
Clave lada

Criterios de validación

Una vez que se hayan recibido los nombres y domicilios normalizados se aplicarán los siguientes criterios de validación:

Nombre de Personas Físicas

Prueba 1 faltantes en campos

Si el registro entregado estaba dividido:

- Separar el nombre entregado y validar la existencia de Nombre(s), Apellido(s) paterno y Apellido(s) maternos
- Evaluar el registro procesado y deberá contar con los mismos criterios de existencia en Nombre(s), Apellido(s) paterno y Apellido(s) maternos, excepto en los casos que el contenido sea "NT", "NO TIENE", "NE", "NO EXISTE"

Si no tiene Separadores "/"

- Original 1 campo, entrega 3 campos (al menos 2 llenos)

Prueba 2. RFC anterior [4] = RFC nuevo [4]

Físicas con separadores en el original

- Si RFC anterior \diamond RFC nuevo y en el nombre anterior existe "MA" si pasa
 - Si RFC anterior \diamond RFC nuevo y en el nombre anterior no existe "MA" NO pasa
- Físicas sin separador en el nombre original
- Se asume que la primera palabra es Apellido paterno
 - Se asume que la segunda palabra es Apellido materno
 - Se asume que el resto es el nombre
 - Si solo tiene dos elementos se asume que no tiene materno
 - Si RFC anterior \diamond RFC nuevo y en el nombre anterior existe "MA" pasa.
 - Si RFC anterior \diamond RFC nuevo y en el nombre anterior no existe "MA" NO pasa
- Físicas (con marca de morales)
- Igual al caso de personas físicas sin separador en el nombre

Prueba 3. Contar palabras.

Físicas (Marca de Físicas)

- Si cantidad de palabras original (sustituyendo "esp" por /) es igual y el campo de varios esta vacío pasa
- Si no cumple lo anterior
- Si varios esta lleno y la diferencia (No. palabras original – No. palabras nuevo) = 1 pasa
- Si cantidad de palabras es = pero varios esta lleno (no pasa)

Físicas (Marca de morales)

- Si cantidad de palabras original es igual y el campo de varios esta vacío (pasa)
- Si no cumple lo anterior
- Si varios esta lleno y la diferencia de palabras es #original – #nuevo = 1 (pasa)
- Si cantidad de palabras es = pero varios esta lleno con algo diferente de "COPROPIETARIOS" (no pasa)

Nombres de personas morales

Prueba 1

- Se entrega original en 1 campo, regresa con 2 campos razón social, régimen de capital (el primero necesariamente lleno)

Prueba 2 RFC anterior [3] = RFC nuevo [3]

- Si RFC anterior = RFC nuevo pasa
- Si RFC anterior \diamond RFC nuevo y existe en el registro original la palabra "bAbC" o "bSbA" o "bSbC" o "bSbD" o "bSbP" pasa y se marca por que estos no pararán la siguiente prueba

Prueba 3. Contar palabras.

- Los marcados en la prueba 2 no aplicarán para la prueba 3.
- Si el núm. de palabras anterior = núm. de palabras nuevo + num. Palabras de régimen de capital + 1 (pasa)
- El resto de los registros no pasan.

Domicilios

Calle

- Si original esta lleno y nuevo vacío (no pasa)
- Excepto en los casos que el contenido del segmento calle sea "SC"
- Si calle original esta llena y nuevo vacío pero #ext y #int original están vacíos y nuevos llenos tales como MZA., LOTE (pasa)
- Si calle original esta lleno y calle nueva esta vacía pero en varios esta lleno (pasa)
- #EXT #INT
- Si original está lleno y nuevo vacío (no pasa)
- Salvo en caso de "SN"
- Si original está lleno y nuevo vacío y la cadena original está en varios (pasa)

Colonia

- Si el original está lleno y el nuevo vacío (no pasa)
- Salvo en los casos de "SC"
- Si original esta vacío y nuevo lleno y existe mas de 1 colonia asignada al CP no pasa
- Si original diferente al nuevo y CP original diferente al nuevo (no pasa)

Municipio

- Si el original esta lleno nuevo vacío (no pasa)
- Si original vacío y nuevo lleno (pasa)
- El texto nuevo debe ser validado contra catalogo de SEPOMEX utilizando el CP como base

Código Postal

- Si original esta lleno y nuevo vacío a menos de que el original sea 00000 (no pasa)
- Si original es menor a cinco posiciones y al agregar un cero al principio si existe en SEPOMEX y es diferente a nuevo no pasa
- Validar que el CP nuevo este en catalogo

Entre calles

- Si dom. Original con 5 "/" tiene calles y en el nuevo están vacíos ambos campos no pasa.

Genéricas

- Los registros que cambian de estado (entidad federativa) no pasan
- Los que tienen calle, números y colonia vacía no pasan

Logística del proceso de limpieza

Dado que cada oficina local es distinta en cuanto a la cantidad de registros en la base de datos y dado que una parte del proceso requiere tener copias de la base de datos local en el centro de cómputo de CPN será necesario tener varias opciones de solución para este problema.

Presentamos los puntos mas importantes de este proceso

- Instalación de los catálogos actualizados y estandarizados en las 65 oficina locales
- Creación de tablas e índices necesarios para el proceso de comparación en la nacional
- Definición del usuario y permisos para limpiezas locales y nacional
- Recuperación de movimientos operados
- Confronta nacional
- Instalación nacional y local de los programas de limpieza
- Generación de cinta con llaves de las 11 tablas de la local para comparación en la nacional o de las 11 tablas que representan el segmento de la local en nacional para comparación en local.
- Carga y comparación de datos de la local con la nacional
- Generación de cinta con datos faltantes de la local
- Respaldo completo de la base de datos local
- Carga en la local de los datos faltantes
- Proceso de limpieza de la local
- Generación de cinta con el respaldo completo de las 11 tablas de registro de la local
- Carga en la nacional del respaldo de la local

Originalmente se presentaron 3 opciones de solución del problema sin embargo una vez realizadas las pruebas se determinó utilizar la opción I en el proceso de limpieza .

Para todas las opciones se debe iniciar con la recuperación de movimientos operados y esta se realizará una sola vez

Fin de semana 0

Viernes 6:00 PM

- Envío de movimientos del día
- Una vez realizada la transmisión **TODAS LAS OFICINAS LOCALES RELIZAN LA RECUPERACIÓN DE MOVIMIENTOS OPERADOS** hasta un día anterior a la fecha de corte que se les indique
- La nacional debe estar preparada para recibir y actualizar los movimientos
- La nacional procesará los movimientos del día
- La nacional procesará los movimientos atrasados
- La nacional envía los archivos de reingreso

Sábado 8:00 AM

- Los operadores de las oficinas locales iniciarán el proceso de actualización de cuenta única local
- Resto de la semana y durante todo el proceso de limpieza se asegurará que no existan faltantes en los envíos y reingresos

Opción I Copia de nacional a local vía cintas

Fin de semana 1

Viernes 6:00 PM

- La oficina local envía el archivo de movimientos del día
- La nacional procesará los movimientos del día
- La nacional genera los archivos de reingreso
- En la nacional se genera la cinta de comparación para la local, respaldo completo de las 10 tablas (Por oficina local)

Para realizar esta extracción se creará una tabla temporal en la que se insertará No. de cliente y oficina local a la que se enviará (hay que recordar que es probable que se estén realizando varias extracciones simultáneamente) de todos los clientes que se deben extraer. Esta selección se generarán en 3 pasos:

I No. de cliente de todos los clientes con No. de módulo perteneciente a la oficina local que se limpia y motivo de baja diferente de cambio de nombre. Si el campo del módulo del cliente tiene basura, la paternidad quedará determinada por a oficina local, si la oficina local también tiene basura, el cliente no se extraerá y se registrará en una bitácora para generar reporte sobre estos casos

II No. de cliente anterior y posterior de clientes con cambio de No. y que ya estén seleccionados en el paso I.

III No. de cliente de clientes registrados en bitácora especial como asignados a la oficina local que se limpia. En ese momento se marcarán en la bitácora especial como operados.

Esta tabla contará con un índice único por No. de cliente de manera que cada dato se inserte una sola vez

Esto determinará el universo de clientes a extraer.

IV Extracción de todos los datos encontrados en las 10 tablas principales relacionados con los números cargados en esta tabla serán extraídos y enviados para limpieza a la oficina local .

Las 10 tablas principales para registro son:

CLIENTE	BAJAS
LIQUIDACION	SUCESION
SUSPENSION	BAJAS
CREDITOS	CEDULAS
DOCUMENTOS (EXTERNOS)	NOMBRES

Dado que tenemos problemas de espacio en la nacional, la extracción, compactación y generación de archivos se realizará por tabla, es decir :

- Se genera extracción de una tabla en archivo
- Se comprime el archivo
- Se almacena en cinta (respaldo multivolumen) o envía el archivo FTP
- Se elimina archivo

- Se manda la cinta a la oficina local o FTP de archivos

Sábado 6:00 AM

- El operador de la oficina local inicia el proceso de actualización de cuenta local
- El operador iniciará el proceso de limpieza utilizando el programa PROCESO
- Se activará el proceso que enviará y recibirá bitácoras de consulta especial
- El operador respalda las 11 tablas de la local utilizando el programa
- El programa de limpieza no podrá ser ejecutado si el operador no ha realizado el respaldo.
- En caso de transferencia vía cinta se carga la cinta con información de la nacional
- En caso de transferencia de archivos el operador verificará la llegada de los 11 archivos

Iniciar proceso de limpieza que tiene varias etapas:

1. Comparación "QUE EXISTE EN LA NACIONAL Y NO EXISTE EN LA LOCAL" para completar tablas de la local
2. Generación de bitácora de clientes en conflicto y datos faltantes, en esta bitácora se registrarán Números de clientes de los que se duda su pertenencia en la oficina local dado que no se recibieron datos en la extracción, no estaban registrados en la local, números de cliente de nombres anteriores no encontrados.
3. El proceso de envío manda los archivos a la nacional y espera respuestas.
4. Se inicia la limpieza de la base de datos local considerando que todos los clientes pertenecen a la local
5. En la nacional se inicia el proceso de complemento de información de conflictos y faltantes
6. Recepción de datos faltantes de la nacional y confirmación de la paternidad de clientes en conflicto
7. Complemento de información de datos faltantes
8. Eliminación de clientes no confirmados en la nacional
9. Ultima etapa de limpieza de albaceas, representantes y liquidadores
10. Se genera bitácora de llaves a respaldar de la oficina local y se envía por FTP al centro nacional
11. En la nacional se inicia el respaldo de los clientes que serán borrados
12. En la local se genera la cinta de respaldo completo de las 11 tablas
13. Dado que tenemos problemas de espacio en la nacional la generación de la cinta se realizará en multivolumen, es decir tabla por tabla
14. Se genera extracción de una tabla en archivo
15. Se comprime el archivo
16. Se almacena en cinta (respaldo multivolumen) o envía el archivo FTP a la nacional
17. Se elimina archivo
18. Se manda la cinta a la nacional o FTP de archivos
19. En la nacional se inicia el proceso de descarga de los 11 archivos
20. Se lee el primer archivo
21. Se descomprime y carga en tabla
22. Se actualiza tabla equivalente en nacional

23. Se eliminan tabla y archivo

En el caso de oficinas locales pequeñas y medianas el proceso de limpieza local durará un fin de semana.

En el caso de oficinas locales grandes será necesario mantener cerrada la oficina local un día más.

Dos de los puntos más complejos de este proceso de limpieza es el determinar la pertenencia de un cliente a determinada oficina local y el asegurar limpieza para todos los clientes de la base de datos (local y nacional).

Con este fin se utilizarán varias bitácoras de paternidad y limpieza que se generarán y conservarán en la base de datos nacional hasta que se concluya la limpieza de las 65 oficinas locales.

Bitácora especial:

En esta bitácora se registrarán aquellos clientes cuya paternidad está en duda, las oficinas locales que se disputan la paternidad y el estatus de integración a la oficina local que le corresponde, es decir, si ya ha sido integrado a la oficina local correcta.

Los campos de esta bitácora son:

No. del cliente en conflicto

oficina local en la que se encontró originalmente

oficina local a la que debe pertenecer

Estatus de operación (si el cliente ya ha sido tomado por la oficina local propietaria)

Clientes limpios:

En esta bitácora se registrarán los clientes que ya han sido reemplazados en la nacional y la oficina local a la que pertenecen. Esta bitácora se crea con el fin de detectar clientes con paternidad en más de una oficina local o sin paternidad.

Los campos de esta bitácora son:

No. del cliente limpio y reemplazado en la base de datos nacional

oficina local a la que quedó asignado

Clientes en conflicto:

En esta bitácora se registrarán aquellos clientes con paternidad duplicada. Tanto en la bitácora especial como en clientes limpios sólo se registrará una sola vez al cliente, el resto de intentos de inserción se realizarán en esta bitácora y no tendrá efecto en la BD. El objetivo de esta bitácora es que se realice una búsqueda en expediente que determine la verdadera paternidad del cliente.

En el caso de inserciones en la bitácora especial, la paternidad del cliente la determinará la primera inserción, el resto se registrará en esta tabla.

En el caso de inserciones en la bitácora clientes en conflicto, cada vez que aparezca una nueva paternidad, esta será la que prevalezca y la anterior se registrará en esta tabla, de lo contrario sería necesario editar el archivo en el que se encuentra la extracción de la local y eliminar al cliente y todos sus datos, lo cual representa muchas horas de proceso sin sentido.

Los campos de esta bitácora son:

- No. del cliente en conflicto
- oficina local a la que debería quedar asignado (pero no ocurrió)
- oficina local en la que se encontró originalmente (sólo para bitácora especial)
- Tipo de duplicidad (especial o limpios)

En los casos de limpieza local se considerará que todos los clientes pertenecen a la local y una vez confirmada o cancelada la paternidad se procederá a borrar o confirmar a los clientes de la base de datos local.

Para facilitar la tarea de limpieza a los operadores de la oficina local se preparará un proceso que automatice los envíos y recepciones de archivos.

Una vez determinados los casos de consulta nacional el programa de limpieza generará un archivo con los siguientes campos:

Caso, variante, subvariante del caso de limpieza
No. de cliente a consultar

Este archivo llevará el nombre oficina_local_nn . una vez que el archivo se haya concluido, se generará un nuevo archivo con nombre yamandann.

En la oficina local antes de iniciar el proceso de limpieza se levantará un proceso que estará monitoreando la creación del archivo yamandann.

El demonio enviará el archivo oficina_local_nn y después el archivo yamandann que indica que el archivo original ya se encuentra completo.

En la nacional también se encontrará activo un proceso que monitorea la llegada de los archivos y lanza el proceso de comparación faltante. Una vez concluido el proceso de comparación faltante, se generan dos archivos, Roficina_local_nn con los resultados de la comparación y el otro yaacabenn que indica el final de la generación del archivo de resultados.

El mismo proceso que envió el archivo a la nacional, está monitoreando la generación de los archivos antes mencionados y en el momento que los detecta los transfiere de la nacional a la local.

Mientras tanto en la local se está llevando a cabo la limpieza de la base de datos, para facilitar el proceso, TODOS los clientes de la local se limpiarán sin considerar la

paternidad, cuando hayan regresado los archivos de resultado de la nacional, aquellos clientes que no pertenecen a la local serán eliminados.

El proceso de limpieza local buscará el regreso de los archivos de consulta nacional, en caso de que ya estén en la local, el proceso continuará y concluirá la limpieza, si no han regresado, el programa se detendrá y avisará al operador que requiere de los archivos y esperará la orden de continuar.

Notas finales

Es muy importante que durante los fines de semana en que se realicen las limpiezas los equipos estén libres para realizar los procesos en un solo fin de semana.

Esto significa que se deberán suspender tanto en local como en nacional otro tipo de procesos que no se relacionen con la limpieza.

Detalle de limpieza

A continuación presento la propuesta de limpieza definitiva que se realizó en una oficina piloto.

Para cada caso se insertaron los siguientes datos:

- Tabla y campos a limpiar
Nombre de la tabla y campos relacionados con la limpieza
- No de casos y descripción
No. de caso (y en su caso variante y subvariante) y descripción del problema
Los números de caso no son consecutivos
Los números de caso no se realizan en el orden en que se presentan
- Solución de limpieza
Descripción de la solución
- Bitácoras de registro
Datos que se registran en las bitácoras de control

Comentario:

- En todos los casos que se indica que es necesario identificar la paternidad de un cliente, se identificarán los cambios de nombre y domicilio.

Tabla CLIENTES

cau_rfc RFC del cliente
cau_deter Determinante del cliente

Caso 1 RFC nulo
Caso 2 Determinante nulo
• Borra

Bitácora:

Caso 1|0|0|Registro de cliente | Fecha y hora|Oficina local
Caso 2|0|0|Registro de cliente | Fecha y hora|Oficina local

Caso 4 RFC menor a 13 posiciones para personas físicas y 12 para personas morales.
Caso 5 Dígito verificador mal calculado
• Para el dígito mal calculado sólo se buscarán registros con último carácter <>[0-9,A]
• Para RFC menor a 13 posiciones sólo se contarán caracteres <> de espacio

Bitácora:

Caso 4|0|0|RFC|DETER||Fecha y hora|Oficina local
Caso 5|0|0|RFC|DETER||Fecha y hora|Oficina local

cau_ofh Módulo del cliente

- Caso 7 Módulo nulo, fuera de catálogo e inconsistencias
- Var 3. Si Módulo es menor o igual a 4 posiciones, se completa con ceros a la derecha hasta 5
 - Var 1. Si el Módulo está fuera de catálogo y existe en el catálogo de equivalencias para claves anteriores, se sustituye
 - Si existe inconsistencia entre los campos Módulo, Oficina local y Oficina regional
 - Se valida en extracción nacional
 - Var 2|1. Si es consistente se utilizan datos de nacional (todo lo relacionado al domicilio del cliente)
 - Cau_domi, cau_codpos, cau_mpio, cau_afr, cau_entfed, cau_ofh, cau_mescie, cau_zondil, cau_telefo
 - Si en extracción nacional tampoco es consistente se quedan valores de local
 - Var 2|2. Si el Módulo no existe en el catálogo se asigna el primer Módulo de la Oficina local que se está limpiando
 - Var 2|3. Se hace consistente en función de Módulo

Bitácora:

Caso 7|1|0|RFC|DETER|Módulo,Oficina local,Oficina regional|Fecha y hora|Oficina local

Caso 7|2|1|RFC|DETER|Datos de domicilio|Fecha y hora|Oficina local

Caso 7|2|2|RFC|DETER|Módulo,Oficina local,Oficina regional|Fecha y hora|Oficina local

Caso 7|2|3|RFC|DETER|Módulo,Oficina local,Oficina regional|Fecha y hora|Oficina local

Caso 7|3|0|RFC|DETER|Módulo,Oficina local,Oficina regional|Fecha y hora|Oficina local

cau_rsoc Razón social o nombre del cliente

- Caso 10 Clientes sin nombre
- Var 1. Búsqueda en la extracción nacional y si existe nombre diferente se copia
 - Si no existe en la extracción nacional o tampoco tiene nombre
 - Var 2. Buscar un cambio de nombre, si existe se copia el último nombre del cliente
 - Var 3. Si tampoco existe cambio de nombre se borra

Bitácora:

Caso 10|1|0|RFC|DETER|Fecha y hora|Oficina local

Caso 10|2|0|RFC|DETER|Nombre Nacional|Fecha y hora|Oficina local

Caso 10|3|0|Registro de cliente |Fecha y hora|Oficina local

Caso 15 Paternidad de un cliente

Para determinar la correcta paternidad de un cliente será necesario asignar una calificación a cada uno de ellos:

- 1 Se encontró en la extracción y también existe en la local
- 2 Se encuentra en la extracción de la nacional y no existía en la local
- 3 Se encuentra en la local pero no llegó en la extracción
 - 3.1 En la extracción nacional si existe con otra paternidad
 - 3.2 En la extracción nacional no existe

Para los escenarios 2 y 3 se registrará la paternidad de los clientes en las bitácoras especiales de la nacional.

Para decidir la correcta paternidad analizaremos cada escenario:

Escenario 1 Se encuentra en la extracción y en la local (se resuelve en la local y se confirma en nacional)

- Identificar si se trata de una baja por cambio de nombre
- Buscar el último nombre asignado al cliente
- Var 1|1. Si no existe el último nombre pero si se trata de una baja por cambio de nombre, se registra en la nacional como sin paternidad (el y sus nombre relacionados) y sin confirmar y se elimina de la local con todo e historia.
- El resto de los casos no se registran en bitácora especial puesto que tanto la local como la nacional coinciden.

Causas :
Movimientos no confirmados
Cambios de nombre (se perdió la liga)
Cambio de domicilio

Escenario 2 Se encuentra en la extracción y no existe en la local (se resuelve en la local y se confirma en nacional)

- Identificar si se trata de una baja por cambio de nombre
- Buscar el último nombre asignado al cliente
- Var 2|1. Si el último nombre se encuentra en la local o se trata de un cambio de nombre, se registrará con paternidad de la Oficina local y operado (el todos los nombres relacionados)
- Var 2|2. Si no existe el último nombre pero si se trata de una baja por cambio de nombre, se registra en la extracción nacional como sin paternidad y sin operar y se elimina de la local con todo e historia.

Causas :
Movimientos no confirmados y cambio de domicilio
Reactivación de cambio de nombre (no llegó orden de borrado a nacional)

Escenario 3 No se encuentra en la extracción y si en la local (se resuelve en la local y se confirma en nacional)

- Identificar si se trata de una baja por cambio de nombre
- Buscar el último nombre asignado al cliente
- Var 3|1. Si no se encuentra el último cambio de nombre se registra como sin paternidad y no operado.
- Var 3|2. Si el último nombre sí se encuentra en la local, se registrará con paternidad de la Oficina local y operado (el y todos los nombres relacionados)
- Si no se trata de un cambio de nombre, identificar si existe el registro de inscripción en la local
- Si no existe registro de inscripción, verificar si tiene datos completos (actividad, fecha de inicio, motivo de alta).
 - Var 3|3. Si se encuentra en alguno de estos casos se marca como inscrito para consulta nacional
 - Var 3|4. Identificar si el documento de inscripción no está confirmado
- Identificar un cambio de domicilio sin confirmar en la tabla de documentos de la Oficina local.
- Si no existe un registro de cambio de domicilio sin confirmar identificar si tiene datos nulos(actividad, fecha de inicio, motivo de alta).
 - Var 3|5. Si se encuentra en alguno de estos casos se marca como cambio de domicilio no confirmado
 - Si queda marcado como 3|3 o 3|4 y 3|5 se queda como 3|3
- Se envían dudas a la nacional
- Si en la nacional se detecta que no existe el cliente:
 - Para los casos 3|2, 3|3 y 3|4 se registra a nombre de la local y operado
 - Para los casos 3|1, 3|5 se registra sin paternidad y sin confirmar y se borra de la local

Causas : Movimientos no confirmados
Inscripciones sin confirmar
Cambio de domicilio de clientes sin registro

- Si en la nacional se detecta que si existe pero el cliente pertenece a otra Oficina local:
 - Para caso 3|2 se registra con paternidad de la local y operado
 - Para los casos 3|1, 3|3, 3|4 y 3|5 se registra con paternidad que indica la nacional y sin confirmar y se borra de la local

Causas : Movimientos no confirmados, cambios de domicilio
No llegó orden de borrado por cambio de domicilio
Inscripciones no confirmadas de clientes que ya existen.

Bitácora:

Caso 15|1|1|RFC|Determinante| Fecha y hora|Oficina local

Caso 15|2|1|RFC|Determinante| Fecha y hora|Oficina local

Caso 15|2|2|RFC|Determinante| Fecha y hora|Oficina local
 Caso 15|3|1|RFC|Determinante| Fecha y hora|Oficina local
 Caso 15|3|2|RFC|Determinante| Fecha y hora|Oficina local
 Caso 15|3|3|RFC|Determinante| Fecha y hora|Oficina local
 Caso 15|3|4|RFC|Determinante| Fecha y hora|Oficina local
 Caso 15|3|5|RFC|Determinante| Fecha y hora|Oficina local
 Caso 15|4|1|Registro Bajas| Fecha y hora|Oficina local
 Caso 15|4|2|Registro Sucesión| Fecha y hora|Oficina local
 Caso 15|4|3|Registro Nombres| Fecha y hora|Oficina local
 Caso 15|4|4|Registro Liquidación| Fecha y hora|Oficina local
 Caso 15|4|5|Registro Suspensión| Fecha y hora|Oficina local
 Caso 15|4|6|Registro Clientes| Fecha y hora|Oficina local
 Caso 15|4|7|Registro Cédulas| Fecha y hora|Oficina local
 Caso 15|4|8|Registro Documentos| Fecha y hora|Oficina local
 Caso 15|4|9|Registro Representantes| Fecha y hora|Oficina local
 Caso 15|4|10|Registro Externos| Fecha y hora|Oficina local
 Caso 15|4|11|Registro Créditos| Fecha y hora|Oficina local

cau_fecalt Fecha de alta del cliente

Caso 16 Clientes sin fecha de alta

- Var 0. Busco fecha en extracción nacional
- Var 1. Si no existe o es nula asignar fecha de constitución para morales
- Var 2. Para físicas asignar la fecha del documento de inscripción o la mas antigua de los aumentos de créditos registrada en documentos o externos
- Var 3. Si no tienen registro en documentos o externos o la fecha de constitución no es válida asignar 01/01/88

Bitácora:

Caso 16|0|0|RFC y Determinante| Fecha y hora|Oficina local
 Caso 16|1|0|RFC y Determinante| Fecha y hora|Oficina local
 Caso 16|2|0|RFC y Determinante| Fecha y hora|Oficina local
 Caso 16|3|0|RFC y Determinante| Fecha y hora|Oficina local

cau_motalt Motivo de alta del cliente

Caso 25 Clientes sin motivo de alta
 Caso 26 Clientes con motivo fuera de catálogo
 Caso 27 Clientes matriz con alta 10 (apertura de establecimiento)

- Buscar en la extracción nacional el motivo
- Si no se encuentra el motivo o está fuera de catálogo se asigna alta por inscripción (00)

Bitácora:

Caso 25|0|0|RFC y Determinante| Fecha y hora|Oficina local
 Caso 26|0|0|RFC y Determinante| Fecha y hora|Oficina local|Motivo
 Caso 27|0|0|RFC|DETER| Fecha y hora|Oficina local

cau_calcont Calidad del cliente

- Caso 29 Clientes sin calidad
- Var 0. Busco en extracción nacional
 - Var 1. Se queda 0
- Caso 30 Clientes con calidad (quitando los últimos 3 caracteres) <> 1, 2, 3, 4, 5, 6, 7, 8 y 10
- Var 1. Se queda 0

Bitácora:

Caso 29|0|0|RFC| Deter |Fecha y hora|Oficina local

Caso 29|1|0|RFC| Deter |Fecha y hora|Oficina local

Caso 30|0|0|RFC| Deter|Calidad|Fecha y hora|Oficina local

- Caso 31 Clientes con grupo económico fuera de catálogo
- Sólo con los últimos 3 caracteres si está fuera de catálogo se genera archivo para validar
 - Sólo se validarán los sectores 3, 4, 5, 7, 8 y 10

Bitácora:

Caso 31|0|0|RFC|DETER| Fecha y hora|Oficina local

- Caso 32 Calidad en local<>Calidad en extracción nacional
- Se queda calidad en extracción nacional siempre y cuando no sea nulo.

Bitácora:

Caso 32|1|0|RFC| Deter|Calidad|Fecha y hora|Oficina local

- Caso 33 Clientes con calidad <>0 sin créditos adecuadas (sólo matrices, sólo créditos activas)

- Var 1. Tiene crédito 123 y no tiene marca de controladora (1000 – 1999)
- Var 2. Tiene crédito 154 y no tiene marca de controlada (2000 – 2999)
- Var 3. No tiene crédito 123 y tiene marca de controladora (1000 – 1999)
- Var 4. No tiene crédito 154 y tiene marca de controlada (2000 – 2999)
- Var 5. Tiene créditos 123 y 154 y tiene marca de controlada (2000 – 2999)
 - Sólo registro en bitácora y reporte
- Var 6. Tiene créditos 123 y 154 y tiene marca de controladora (1000 – 1999)
 - Eliminar crédito 154

Bitácora:

Caso 33|1|0|RFC| Fecha y hora|Oficina local

Caso 33|2|0|RFC| Fecha y hora|Oficina local

Caso 33|3|0|RFC| Fecha y hora|Oficina local

Caso 33|4|0|RFC| Fecha y hora|Oficina local

Caso 33|5|0|RFC| Fecha y hora|Oficina local

Caso 33|6|0|Registro de crédito| Fecha y hora|Oficina local

cau_tipsoc Tipo de sociedad del cliente (vigilancia mensual)

- Caso 34 Clientes sin tipo de sociedad (sólo para matrices)
- Var 0. Busco valor en extracción nacional
 - Var 1. Se queda 0 que indica no vigilar mensualmente

- Caso 35 Clientes con tipo ≤ 0 y ≤ 5
- Sólo para matrices se queda 0 que indica no vigilar mensualmente

Bitácora:

Caso 34|0|0|RFC| Deter| Fecha y hora|Oficina local *

Caso 34|1|0|RFC| Deter| Fecha y hora|Oficina local

Caso 35|0|0|RFC| Deter|Tipo sociedad|Fecha y hora|Oficina local

cau_tipcont Tipo de cliente

- Caso 36 Clientes con tipo fuera de catálogo
- Var 0. Se queda "espacio" que indica normal

- Caso 37 Clientes sin tipo
- Var 0. Busco valor en extracción nacional
 - Var 1. Se queda "espacio" que indica normal

Bitácora:

Caso 36|0|0|RFC| Deter|Tipo cliente|Fecha y hora|Oficina local

Caso 37|0|0|RFC| Deter|Fecha y hora|Oficina local

Caso 37|1|0|RFC| Deter|Fecha y hora|Oficina local

- Caso38 Tipo en local \leq tipo en extracción nacional
- Para valores 1 y 2 se queda tipo nacional
 - Para otros valores se queda tipo local

Bitácora:

Caso 38|0|0|RFC|DETER|Tipo nal.|Fecha y hora|Oficina local

Caso 38|1|0|RFC|DETER|Tipo borré|Fecha y hora|Oficina local

cau_cveact Actividad preponderante

- Caso 41 Clientes sin actividad
- Var 0. Busco valor en extracción nacional
 - Var 1. Se queda actividad desconocida

- Caso 43 Clientes con actividad fuera de catálogo
- Se queda actividad desconocida

Bitácora:

Caso 41|0|0|RFC y Determinante| Fecha y hora|Oficina local

Caso 41|1|0|RFC y Determinante| Fecha y hora|Oficina local

Caso 43|0|0|RFC y Determinante|Actividad|Fecha y hora|Oficina local

cau_movs Estátus del cliente

- Caso 45 Actualizar estatus de cliente de acuerdo a su segmento

Nota: para todos los casos sólo se registrará en bitácora aquellos casos en los que el estatus se modificó.

Estatus cliente = '0'+

Si está en bajas 00001 1+ Si está en liquidación 00010 2+

Si está en sucesión 00100 4+ Si está en suspensión 01000 8+

Si está en nombres 10000 16

- Si cau_movs=' ' y estatus cliente=0 OK
- Var 1|0. Si cau_movs<>' ' y estatus cliente=1 o 2 o 4 o 8 o 17 se mapea estatus a cau_movs (ojo con motivo de baja)
 - Var 6|0. Si es baja con motivo 14 o 33 y no existe registro en nombres, se inventa el cambio de nombre
 - Si es 17 y motivo de baja no es 14 o 33
 - Var 1|1. Se actualiza el motivo por 33 y se actualiza la fecha de baja por la fecha de cambio de nombre
 - Var 1|2. Se actualizan las fechas de baja de las créditos (con fecha de baja igual a la del segmento baja) por la fecha del cambio de nombre
- Si cau_movs=' ' y estatus cliente=1 o 2 o 4 o 8 o 17
 - Buscar en documentos una reactivación (16,22,54,56) posterior o igual al tipo de movimiento que origina el estatus del cliente
 - Fecha op >= fecha de segmento y no existen movimientos de baja posteriores del mismo día.
 - Var 2|0. Si existe se queda como activo y se borra del segmento (o los segmentos)
 - Var 2|1. Se reactivan las créditos con fecha de baja igual a la del segmento.
 - Var 3|0. Si no existe se mapea el estatus a cau_movs
 - Si es 17 y motivo de baja no es 14 o 33
 - Var 6|0. Si es baja con motivo 14 o 33 y no existe registro en nombres, se inventa el cambio de nombre

- Var 3|1. Se actualiza el motivo por 33 y se actualiza la fecha de baja por la fecha de cambio de nombre
- Var 3|2. Se actualizan las fechas de baja de las créditos (con fecha de baja igual a la del segmento baja) por la fecha del cambio de nombre
- Si cau_movs<>' ' y estatus de cliente <> (0,1,2,4,8,17) está en mas de 1 segmento o cambio de nombre sin bajas.
 - Si es 16 o tiene 16 (cambio de nombre sin baja) se registra en el segmento de bajas con la fecha de cambio de nombre y el motivo 33.
 - Se elimina del resto de los segmentos (sin incluir nombres)
 - Var 4|1. Se mapea estatus de acuerdo a lo especificado en Var1
 - Si es 17 y motivo de baja no es 14 o 33
 - Var 5|0. Si está en el segmento de bajas (si es non) y el motivo es diferente de 50 o 06 (no localizado) se marca como baja y se elimina del resto de los segmentos. Se mapea estatus.
 - Var 6|0. Si es baja con motivo 14 o 33 y no existe registro en nombres, se inventa el cambio de nombre
 - Var 5|1. Si está en bajas y el motivo es 50 o 06 se elimina de bajas
 - Var 5|2 Se reactivan las créditos que coincidan con fecha de baja.
 - Se recalcula el estatus del cliente para determinar si continúa en mas de un segmento (restar 1)
 - Si no está en bajas y no es (o no tiene) 16
 - Var 7|1. Si está en suspensión y liquidación, se elimina de suspensión.
 - Var 7|2. Buscar documento o externos el registro que asocia el movimiento con fecha mayor, si no existe se busca en los segmentos; en ambos casos se mapea estatus de fecha mayor y se eliminan registros de fechas anteriores
 - Var 10|0. Si están en liquidación y sucesión manda el tipo de persona
 - Var 10|1. Se mapea estatus
- Si cau_movs<>' ' y estatus=0
 - Var 8|0 Se activa
- Si cau_movs=' ' y estatus cliente<>(0,1,2,4,8,17) (activo y en varios segmentos o cambio de nombre sin baja)
 - Buscar en documentos una reactivación posterior al tipo de movimiento que origina el estatus del cliente
 - Var 11|0. Si existe se queda como activo y se borra del segmento
 - Var 11|1. Se realcivan las créditos de los segmentos borrados
 - Si no existe se mapea el estatus de acuerdo a los criterios de los puntos (4-10)
 - Si es 16 o tiene 16 (cambio de nombre sin baja) se registra en el segmento de bajas con la fecha de cambio de nombre y el motivo 33.
 - Se elimina del resto de los segmentos (sin incluir nombres)
 - Var 6|0. Si es baja con motivo 14 o 33 y no existe registro en nombres, se inventa el cambio de nombre
 - Var 12|1. Se mapea estatus de acuerdo a lo especificado en Var1
 - Si es 17 y motivo de baja no es 14 o 33

- Var 12|3. Si está en el segmento de bajas (si es non) y el motivo es diferente de 50 o 06 (no localizado) se marca como baja y se elimina del resto de los segmentos. Se mapea estatus.
- Var 12|4. Si está en bajas y el motivo es 50 o 06 se elimina de bajas
- Var 12|5 Se reactivan las créditos que coincidan con fecha de baja.
- Se recalcula el estatus del cliente para determinar si continúa en mas de un segmento (restar 1)
- Si no está en bajas y no es (o no tiene) 16
- Var 12|6. Si está en suspensión y liquidación, se elimina de suspensión.
- Var 12|7. Buscar documento o externos el registro que asocia el movimiento con fecha mayor, si no existe se busca en los segmentos; en ambos casos se mapea estatus de fecha mayor y se eliminan registros de fechas anteriores
- Var 12|8. Si están en liquidación y sucesión manda el tipo de persona
- Var 12|9. Se mapea estatus

Nota. Es necesario realizar primero la limpieza de físicas en liquidación y morales en sucesión

Bitácora:

Caso 45|1|0|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|1|1|RFC| bajas | Fecha y hora|Oficina local|
 Caso 45|1|2|RFC| créditos | Fecha y hora|Oficina local|
 Caso 45|2|0|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|2|1|RFC| créditos | Fecha y hora|Oficina local|
 Caso 45|3|0|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|3|1|RFC| bajas | Fecha y hora|Oficina local|
 Caso 45|3|2|RFC| créditos | Fecha y hora|Oficina local|
 Caso 45|4|1|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|5|0|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|5|1|RFC| Fecha y hora|Oficina local|
 Caso 45|5|2|RFC| créditos | Fecha y hora|Oficina local|
 Caso 45|6|0|RFC| Fecha y hora|Oficina local|
 Caso 45|7|1|RFC| Fecha y hora|Oficina local|
 Caso 45|7|2|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|8|0|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|10|0|RFC| Fecha y hora|Oficina local|
 Caso 45|10|1|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|11|0|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|11|1| créditos | Fecha y hora|Oficina local|
 Caso 45|12|1|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|12|3|RFC| estatus | Fecha y hora|Oficina local|
 Caso 45|12|4|RFC| Fecha y hora|Oficina local|
 Caso 45|12|5|RFC| créditos | Fecha y hora|Oficina local|
 Caso 45|12|6|RFC| Fecha y hora|Oficina local|

Caso 45|12|7|RFC| estatus | Fecha y hora|Oficina local|
Caso 45|12|8|RFC| Fecha y hora|Oficina local|
Caso 45|12|9|RFC| estatus | Fecha y hora|Oficina local|

Caso 45|13|1|Registro Bajas| Fecha y hora|Oficina local
Caso 45|13|2|Registro Sucesión| Fecha y hora|Oficina local
Caso 45|13|3|Registro Nombres| Fecha y hora|Oficina local
Caso 45|13|4|Registro Liquidación| Fecha y hora|Oficina local
Caso 45|13|5|Registro Suspensión| Fecha y hora|Oficina local

Caso 47 Clientes sin estatus
Caso 48 Clientes con estatus fuera de catálogo
 • Se queda “espacio” (activo)
Caso 49 Clientes en extracción nacional que no están en local
 • Insertar toda la información del cliente de la nacional que no exista en la local.

Bitácora:

Caso 47|0|0|RFC| Fecha y hora|Oficina local
Caso 48|0|0|RFC|cau_movs| Fecha y hora|Oficina local
Caso 49|0|0|Registro de cliente nal.| Fecha y hora|Oficina local

cau_mescie Oficina local del cliente

Caso 46 Oficina local nula
 • Var 1. Si el Módulo es correcto, se asigna la Oficina local asociada
 • Var 2. Si el Módulo no es correcto se asigna la Oficina local que se está limpiando

Bitácora:

Caso 46|1|0|RFC|DETER| Fecha y hora|Oficina local
Caso 46|2|0|RFC|DETER| Fecha y hora|Oficina local

Tabla BAJAS

baj_rfc **RFC del cliente en segmento bajas**
baj_deter **Determinante del cliente en segmento bajas**

Caso 50 RFC nulo
Caso 51 Determinante nulo
 • Se borra

Bitácora:

Caso 50|0|0|Registro bajas| Fecha y hora|Oficina local
Caso 51|0|0|Registro bajas| Fecha y hora|Oficina local

Caso 52 Bajas sin cliente

- Se borra

Bitácora:

Caso 52|1|0|Registro bajas|Fecha y hora|Oficina local

baj_cvebaj Motivo de baja del cliente

Caso 54 Clientes sin motivo de baja

Caso 55 Clientes con motivo fuera de catálogo

- Se busca motivo en documentos o externos.
 - Var 0|0. Si existe, se copia la clave.
 - Var 1|0. Si no existe registro
 - Para físicas se pone 04 (defunción)
 - Para morales se pone 01 (liquidación total)

Bitácora:

Caso 54|0|0|RFC|DETER| Fecha y hora|Oficina local

Caso 55|0|0|RFC|DETER|motivo baja|Fecha y hora|Oficina local

Caso 54|1|0|RFC|DETER| Fecha y hora|Oficina local

Caso 55|1|0|RFC|DETER|motivo baja|Fecha y hora|Oficina local

baj_fec Fecha de baja del cliente

Caso 56 Clientes sin fecha

- Var 1. Si existe en bajas nacional se queda fecha de nacional
- Var 2. Buscar fecha en documentos de la local el documento de baja
- Var 3. Si no existe registro se inicializa con 99/01/01

Bitácora:

Caso 56|1|0|RFC| Fecha y hora|Oficina local

Caso 56|2|0|RFC| Fecha y hora|Oficina local

Caso 56|3|0|RFC| Fecha y hora|Oficina local

Caso 57 Clientes en bajas de la nacional que no existen en la local

- Insertar toda la información de bajas del cliente en la extracción nacional que no exista en mi local.

Bitácora:

Caso 57|0|0|Registro bajas| Fecha y hora|Oficina local

Tabla LIQUIDACION

liq_rfc RFC del cliente con inicio de liquidación

Caso 58 RFC nulo

- Se borra

Bitácora:

Caso 58|0|0|Registro liquidación| Fecha y hora|Oficina local

Caso 59 Liquidación sin cliente

- Se borra

Bitácora:

Caso 59|0|0|Registro liquidación| Fecha y hora|Oficina local

Caso 60 Personas físicas en liquidación

- Var 1. Si cliente no existe en sucesión, mover a sucesión y borrar de liquidación
- Var 2. Si cliente ya existe en sucesión, borra en liquidación

Bitácora:

Caso 60|1|0|Registro liquidación| Fecha y hora|Oficina local

Caso 60|2|0|Registro liquidación| Fecha y hora|Oficina local

liq_fechi Fecha de inicio de la liquidación del cliente

Caso 64 Clientes sin fecha

- Var 1. Si existe en liquidación nacional se queda fecha de nacional
- Var 2. Buscar fecha en documentos de la local el inicio de liquidación
- Var 3. Si no existe registro se inicializa con 99/01/01

Bitácora:

Caso 64|1|0|RFC| Fecha y hora|Oficina local

Caso 64|2|0|RFC| Fecha y hora|Oficina local

Caso 64|3|0|RFC| Fecha y hora|Oficina local

liq_rfcliq RFC del liquidador

Caso 65 Limpieza del RFC del liquidador

- Var 1. Borrar (limpiar todos los datos del liquidador) a los que son ellos mismos
- Var 2. Completar con datos de cliente a los que si se encontraron en la local
- Var 3|0. Generar archivo para completar con datos de nacional (sólo para RFC de 13 posiciones)
- Var 3|1. Completar con datos de cliente a los que si se encontraron en la extracción nacional
- Var 4. Borrar a los que no se encontraron y el nombre es otro RFC o está vacío
- Es necesario realizar primero la limpieza de personas físicas en liquidación y morales en sucesión

Bitácora:

Caso 65|0|0|RFC|liq_nomliq|liq_rfcliq|liq_dom|liq_tel|Oficina local liquidador|Fecha y hora|Oficina local (nal.)

Caso 65|1|0|RFC|liq_nomliq|liq_rfcliq|liq_dom|liq_tel| Fecha y hora|Oficina local (borré)

Caso 65|2|0|RFC|liq_nomliq|liq_rfcliq|liq_dom|liq_tel| Fecha y hora|Oficina local (borré)

Caso 65|3|0|RFC|liq_nomliq|liq_rfcliq|liq_dom|liq_tel| Fecha y hora|Oficina local (borré)

Caso 65|4|0|RFC|liq_nomliq|liq_rfcliq|liq_dom|liq_tel| Fecha y hora|Oficina local (borré)

Caso 66 Clientes en liquidación de la nacional que no existen en la local

- Insertar toda la información del cliente de la nacional que no exista en la local que se esta procesando

Bitácora:

Caso 66|0|0|Registro liquidación| Fecha y hora|Oficina local

Tabla SUCESION

suc_rfc RFC del cliente con inicio de sucesión

Caso 67 RFC nulo

Bitácora:

Caso 67|0|0|Registro sucesión| Fecha y hora|Oficina local

Caso 68 Sucesión sin cliente

- Se borra

Bitácora:

Caso 68|1|0|Registro sucesión| Fecha y hora|Oficina local

Caso 69 Personas morales en sucesión

- Var 1. Si cliente no existe en liquidación, mover a liquidación y borrar de sucesión
- Var 2. Si cliente ya existe en liquidación, borra

Bitácora:

Caso 69|1|0|Registro sucesión| Fecha y hora|Oficina local

Caso 69|2|0|Registro sucesión| Fecha y hora|Oficina local

suc_fecha Fecha de inicio de la sucesión del cliente

Caso 70 Clientes sin fecha

- Var 1. Si existe en sucesión nacional se queda fecha de nacional
- Var 2. Buscar fecha en documentos de la local el inicio de sucesión
- Var 3. Si no existe registro se inicializa con 99/01/01

Bitácora:

Caso 70|1|0|RFC| Fecha y hora|Oficina local

Caso 70|2|0|RFC| Fecha y hora|Oficina local

Caso 70|3|0|RFC| Fecha y hora|Oficina local

suc_rfcalb RFC del albacea

Caso 71 Limpieza del RFC del albacea

- Marca de encontrado, 0 el mismo, 1 en la local, 2 en la extracción nacional con diferente Oficina local, 3 no encontrado
- Var 1. Borrar (limpiar el campo) a los que son ellos mismos
- Var 2. Completar con datos de cliente a los que si se encontraron en local
- Var 3|0. Generar archivo para completar con datos de nacional (sólo para RFC de 13 posiciones)
- Var 3|1. Completar con datos de cliente a los que si se encontraron en nacional
- Var 4. Borrar a los que no se encontraron y el nombre es otro RFC o está vacío.

Bitácora:

Caso 71|0|0|RFC| Fecha y hora|Oficina local| RFC albacea| Oficina local albacea| suc_nomalb|suc_rfc|suc_domalb (de la nal.)

Caso 71|1|0|RFC| Fecha y hora|Oficina local| suc_nomalb|suc_rfc|suc_domalb

Caso 71|2|0|RFC| Fecha y hora|Oficina local| RFC albacea|

suc_nomalb|suc_rfc|suc_domalb

Caso 71|3|0|RFC| Fecha y hora|Oficina local| suc_nomalb|suc_rfc|suc_domalb

Caso 71|4|0|RFC| Fecha y hora|Oficina local| suc_nomalb|suc_rfc|suc_domalb

Caso 72 Clientes en sucesión de la nacional que no existen en la local

- Insertar toda la información del cliente en sucesión de la nacional que no exista en la local que se esta procesando

Bitácora:

Caso 72|0|0|Registro sucesión| Fecha y hora|Oficina local

Tabla SUSPENSION

sus_rfc RFC en suspensión
sus_deter Determinante del RFC en suspensión

Caso 73 RFC y/o determinante nulo

Var 1: RFC nulo

Var 2: Determinante nulo

- Se borra

Bitácora:

Caso 73|1|0|Registro suspension| Fecha y hora|Oficina local

Caso 73|2|0|Registro suspensión| Fecha y hora| Oficina local

Caso 74 Suspensión sin cliente

- Se borra

Bitácora:

Caso 74|1|0|Registro suspensión| Fecha y hora|Oficina local

sus_fecha Fecha de inicio de la suspensión del cliente

Caso 75 Clientes sin fecha

- Var 1. Si existe en suspensión nacional se queda fecha de nacional
- Var 2. Buscar fecha en documentos de la local el inicio de suspensión
- Var 3. Si no existe registro se inicializa con 99/01/01

Bitácora:

Caso 75|1|0|RFC|DETER| Fecha y hora|Oficina local

Caso 75|2|0|RFC|DETER| Fecha y hora|Oficina local

Caso 75|3|0|RFC|DETER| Fecha y hora|Oficina local

Caso 77 Clientes en suspensión de la nacional que no existen en la local

- Insertar toda la información del cliente en suspensión de la nacional que no existe en la local que se esta procesando

Bitácora:

Caso 77|0|0|Registro suspensión| Fecha y hora|Oficina local

Tabla REPRESENTANTE

rep_rfccau RFC del cliente con representante

Caso 78 RFC nulo

Caso 79 Determinante nulo

- Borrar

Bitácora:

Caso 78|0|0|Registro representante| Fecha y hora|Oficina local

Caso 79|0|0|Registro representante| Fecha y hora|Oficina local

Caso 80 Representante sin cliente

- Se borra

Bitácora:

Caso 80|1|0|Registro representante| Fecha y hora|Oficina local

rep_rfcrep RFC del representante

Caso 81 Limpieza especial para representantes

- Marca de encontrado, 0 el mismo, 1 en la local, 2 en la extracción nacional con diferente Oficina local, 3 no encontrado
- Var 1. Borrar (registro) a los que son ellos mismos
- Var 2. Completar con datos de cliente a los que si se encontraron en local
- Var 3|0. Generar archivo para completar con datos de nacional (sólo para RFC de 13 posiciones)
- Var 3|1. Completar con datos de cliente a los que si se encontraron en nacional
- Var 4. Borrar a los que no se encontraron y el nombre es otro RFC o está vacío.

Bitácora:

Caso 81|0|0|RFC|DETER| Fecha y hora|Oficina local| RFC albacea| Oficina local albacea| rep_nomrep|rep_domrep|rep_telef (nal.)

Caso 81|1|0|Registro representa| Fecha y hora|Oficina local|

Caso 81|2|0|RFC|DETER| Fecha y hora|Oficina local|rep_nomrep|rep_domrep|rep_telef

Caso 81|3|0|RFC|DETER| Fecha y hora|Oficina local| rep_nomrep|rep_domrep|rep_telef (borré)

Caso 81|4|0|Registro representa| Fecha y hora|Oficina local

Caso 82 Clientes con representante de la nacional que no existen en la local

- Insertar toda la información del cliente de representante en la extracción nacional que no exista en la local

Bitácora:

Caso 82|0|0|Registro representante| Fecha y hora|Oficina local

Tabla CEDULAS

ced_rfc RFC del cliente con cédula

Caso 83 RFC nulo

- Borra y se genera registro para validación

Bitácora:

Caso 83|0|0|Registro cédulas| Fecha y hora|Oficina local

Caso 84 Cédula sin cliente

- Se borra

Bitácora:

Caso 84|1|0|Registro cédulas| Fecha y hora|Oficina local

ced_let Letra de la cédula

ced_num Número de la cédula

Caso 85 Cédulas con letra o número nulo

- Var 1. Cédulas con letra nula, se borran
- Var 2. Cédulas con número nulo, se borran

Bitácora:

Caso 85|1|0|Registro cédulas| Fecha y hora|Oficina local

Caso 85|2|0|Registro cédulas| Fecha y hora|Oficina local

Caso 86 Cédulas de la nacional que no existen en la local

- Insertar toda la información del cliente de cédulas en la extracción nacional que no exista en la local

Bitácora:

Caso 86|0|0|Registro cédula| Fecha y hora|Oficina local

ced_fecalt Fecha de alta de la cédula

Caso 90 Cédulas sin fecha alta

- Var 1: Si existe se guarda la fecha nacional
- Var 2: Si no existe se inicializa con 99/01/01

Bitácora:

Caso 90|1|0|RFC|Letra y número de cédula|Fecha y hora|Oficina local|

Caso 90|2|0|RFC| Letra y número de cédula|Fecha y hora|Oficina local|

Caso 142 Cédulas con estatus que indica activo y fecha de baja
Cédulas con estatus de baja y sin fecha de baja

- Var 1. Cédulas con fecha de baja y estatus no acorde, se cancelan "C"
- Var 2. Cédulas sin fecha de baja y estatus no acorde, se activan "R"

Bitácora:

Caso 142|1|0|Registro cédulas| Fecha y hora|Oficina local

Caso 142|2|0|Registro cédulas| Fecha y hora|Oficina local

Tabla DOCUMENTOS y EXTERNOS

Doc_estcap Estación de captura

Ext_estcap Estación de captura

Caso 92, 93 Documentos y externos sin estación de captura

- Se marca con clave 99

Bitácora:

Caso 92|0|0| No. de operación|Módulo | Fecha y hora|Oficina local|

Caso 93|0|0| No. de operación|Módulo | Fecha y hora|Oficina local|

doc_rfc RFC del cliente

doc_deter Determinante del cliente

Caso 96, 97 RFC nulo

Caso 98, 99 Determinante nulo

- Borra

Bitácora:

Caso 96|0|0|Registro documentos| Fecha y hora|Oficina local|

Caso 97|0|0|Registro externos| Fecha y hora|Oficina local|

Caso 98|0|0|Registro documentos| Fecha y hora|Oficina local|

Caso 99|0|0|Registro externos| Fecha y hora|Oficina local|

Caso 100,101 Documentos sin cliente

- Se borra

Bitácora:

Caso 100|1|0|Registro documentos| Fecha y hora|Oficina local|

Caso 101|1|0|Registro externos| Fecha y hora|Oficina local

doc_nofb Módulo del documento

Caso 102,103 Módulo nulo

- Var 0. Buscar en extracción de nacional
- Var 2. Si en menor a 5 posiciones se completa con ceros a la derecha
- Var 2|1. Si se duplica el registro se elimina
- Var 1. Si no existe se queda la clave "99900" Módulo desconocido
- Agregar clave al catálogo

Bitácora:

Caso 102|0|0|RFC|DETER| No. de operación|Módulo | Fecha y hora|Oficina local

Caso 102|1|0|RFC|DETER| No. de operación|Módulo | Fecha y hora|Oficina local

Caso 102|2|0|RFC|DETER| No. de operación|Módulo | Fecha y hora|Oficina local

Caso 102|2|1|Registro de documentos | Fecha y hora|Oficina local

Caso 103|0|0|RFC|DETER| No. de operación|Módulo | Fecha y hora|Oficina local

Caso 103|1|0|RFC|DETER| No. de operación|Módulo | Fecha y hora|Oficina local

Caso 103|2|0|RFC|DETER| No. de operación|Módulo | Fecha y hora|Oficina local

Caso 103|2|1|Regiastro de externos | Fecha y hora|Oficina local

Caso 104,105 Módulo fuera de catálogo

- Var 4. Si en menor a 5 posiciones se completa con ceros a la derecha
- Var 4|1. Si se duplica el registro se elimina
- Validar con catálogo de equivalencia para claves anteriores
- Var 1. Si existe equivalencia y no se duplica el registro se reemplaza con el nuevo valor
- Var 2. Si existe equivalencia y se duplica el registro, se elimina
- Var 3. Si no existe equivalencia se queda la clave "999" Módulo desconocido
- Var 3|1. Si se duplica el registro se elimina

Bitácora:

Caso 104|1|0|RFC|DETER|No. de operación|Módulo borré|Fecha y hora|Oficina local

Caso 105|1|0|RFC|DETER|No. de operación|Módulo|Fecha y hora|Oficina local|Módulo borré

Caso 104|2|0|Registro documentos|Fecha y hora|Oficina local

Caso 105|2|0|Registro externos|Fecha y hora|Oficina local

Caso 104|3|0|RFC|DETER|No. de operación|Módulo borré|Fecha y hora|Oficina local

Caso 105|3|0|RFC|DETER|No. de operación|Módulo|Fecha y hora|Oficina local|Módulo borré

Caso 104|4|0|RFC|DETER| No. de operación|Módulo | Fecha y hora|Oficina local

Caso 104|4|1|Registro de documentos | Fecha y hora|Oficina local

Caso 104|3|1|Registro de documentos | Fecha y hora|Oficina local

Caso 105|4|0|RFC|DETER| No. de operación|Módulo | Fecha y hora|Oficina local

Caso 105|4|1|Regiastro de externos | Fecha y hora|Oficina local

Caso 105|3|1|Registro de documentos | Fecha y hora|Oficina local

doc_fechop Fecha de operación
doc_mofech Fecha del motivo del trámite

Caso 106,107 Documentos sin fecha de operación

Caso 108,109 Documentos sin fecha de trámite

- Var 0. Buscar en extracción nacional.
- Var 1. Si cualquiera está lleno se copia la fecha al otro campo
- Var 2. Si no existe ninguno se inicializa con 99/01/01

Bitácora:

Caso 106|0|0|RFC|DETER|No. de operación| Fecha y hora|Oficina local

Caso 108|0|0|RFC|DETER|No. de operación|Módulo| Fecha y hora|Oficina local

Caso 107|0|0|RFC|DETER|No. de operación| Fecha y hora|Oficina local

Caso 109|0|0|RFC|DETER|No. de operación|Módulo| Fecha y hora|Oficina local

Caso 106|1|0|RFC|DETER|No. de operación| Fecha y hora|Oficina local

Caso 106|2|0|RFC|DETER|No. de operación| Fecha y hora|Oficina local|fecha de operación
nal.

Caso 107|1|0|RFC|DETER|No. de operación|Módulo| Fecha y hora|Oficina local

Caso 107|2|0|RFC|DETER|No. de operación|Módulo| Fecha y hora|Oficina local|fec.
operación nal.

Caso 108|1|0|RFC|DETER|No. de operación| Fecha y hora|Oficina local

Caso 108|2|0|RFC|DETER|No. de operación| Fecha y hora|Oficina local|fecha trámite nal.

Caso 109|1|0|RFC|DETER|No. de operación|Módulo| Fecha y hora|Oficina local

Caso 109|2|0|RFC|DETER|No. de operación|Módulo| Fecha y hora|Oficina local|fecha
trámite nal.

Caso 110 Documentos que están tanto en externos como en documentos.

- Var 1. Si el Módulo de documentos pertenece a la Oficina local se borra de externos
- Var 2. Si el Módulo de documentos no pertenece a la Oficina local se borra de documentos.
- Var 3. Si el Módulo no es identificable se borra de documentos

Bitácora:

Caso 110|1|segmento de externos |Fecha y hora |Oficina local

Caso 110|2|segmento de documentos | Fecha y hora |Oficina local

Caso 110|3|segmento de documentos | Fecha y hora |Oficina local

Caso 111 Registros duplicados en la tabla de externos.

- Eliminar registros duplicados de acuerdo a la llave de la nacional (documentos)

Bitácora:

Caso 111|1|segmento de externos |Fecha y hora |Oficina local

Doc_nope No. de operación

Ext_nope No. de operación

Caso 114, 115 Documentos y externos sin número de operación

- Se borran

Bitácora:

Caso 114|0|0|Registro documentos| Fecha y hora|Oficina local

Caso 115|0|0|Registro externos| Fecha y hora|Oficina local

Caso 116,117 Documentos de la nacional que no existen en la local

- Insertar toda la información de documentos ó externos de la nacional que no están en la local que se está procesando

Bitácora:

Caso 116|0|0|Registro documentos| Fecha y hora|Oficina local

Caso 117|0|0|Registro externos| Fecha y hora|Oficina local

doc_motmov Clave del movimiento

Caso 118,119 Motivo nulo

- Var 0. Se busca en extracción nacional
- Var 1. Si no existen datos se marcan con "ZZ" (Motivo indefinido)

Caso 120,121 Motivo fuera de catálogo

- Se marcan con "ZZ" (Motivo indefinido)

Bitácora:

Caso 118|0|0|RFC|DETER|No. operación| Fecha y hora|Oficina local

Caso 119|0|0|RFC|DETER|No. operación|Módulo| Fecha y hora|Oficina local

Caso 118|1|0|RFC|DETER|No. operación| Fecha y hora|Oficina local

Caso 119|1|0|RFC|DETER|No. operación|Módulo| Fecha y hora|Oficina local

Caso 120|0|0|RFC|DETER|No. operación| Fecha y hora|Oficina local|motivo

Caso 121|0|0|RFC|DETER|No. operación|Módulo| Fecha y hora|Oficina local|motivo

Tabla CRÉDITOS

obl_rfc RFC del cliente

obl_deter Determinante del cliente

Caso122 RFC nulo

- Borrar

Bitácora:

Caso 122|0|0|Registro créditos| Fecha y hora|Oficina local

Caso 123 Determinante nulo

- Var 1. Si la matriz existe en la local y ya tiene esa crédito se borra
- Var 2. Si la matriz existe en la local y no tiene esa crédito se marca como matriz
- Var 3. Si la matriz no existe en la local se borra

Bitácora:

Caso 123|1|0|Registro crédito| Fecha y hora|Oficina local

Caso 123|2|0|RFC|DETER|Crédito| Fecha y hora|Oficina local

Caso 123|3|0|Registro crédito| Fecha y hora|Oficina local

Caso 124 Créditos sin cliente

- Se borra

Bitácora:

Caso 124|1|0|Registro crédito| Fecha y hora|Oficina local

Caso 125 Clientes matriz sin créditos

- Si cliente no cuenta con créditos en extracción nacional
 - Generar registro para validación

Bitácora:

Caso 125|0|0|RFC| Fecha y hora|Oficina local

obl_clave Clave de la crédito

Caso 126 Clave nula o fuera de catálogo

- Var 0. Se borra
- Var 1. En caso de que el cliente quede sin créditos, se indicará en el campo de marca de unicidad.

Bitácora:

Caso 126|0|0|Registro crédito| Fecha y hora|Oficina local

Caso 126|1|0|RFC y Determinante| Fecha y hora|Oficina local

Caso 127 Clave en desuso, personas físicas con créditos de morales y viceversa

- Se actualiza según catálogo de equivalencias
- Var 1.Si catalogo especifica borrar
- En caso de que el cliente quede sin créditos, se indicará en el campo de marca de unicidad.
- Var 2.Si catálogo especifica actualizar
- Var 3.El cliente ya contaba con la crédito

Bitácora:

Caso 127|1|0|RFC|DETER| Marca de unicidad de crédito|crédito| Fecha y hora|Oficina local

Caso 127|2|0|RFC|DETER|Crédito|Fecha y hora|Oficina local

Caso 127|3|0|RFC|DETER|Crédito|Fecha y hora|Oficina local

- Caso 129 Clientes sin crédito padre
- Se validará contra el catálogo de Créditos Padre
 - Si no tiene Crédito Padre registro para validación

Bitácora:

Caso 129|0|0|RFC|DETER| Fecha y hora|Oficina local

- Caso 131 Créditos en nacional que no existen en local
- Insertar toda la información de créditos de la nacional que no están en la local que se está procesando

Bitácora:

Caso 131|0|0|Registro crédito nal.| Fecha y hora|Oficina local

- Caso 132 Clientes con créditos incompatibles (sólo entre créditos activos)

- Utilizando el catálogo de incompatibles para limpieza
- Var 1. Se eliminará la crédito que tenga la fecha mas antigua
- Var 2. En caso de que ambas fechas sean iguales, se eliminará la de número menor

Bitácora:

Caso 132|1|0|Registro crédito| Fecha y hora|Oficina local

Caso 132|2|0|Registro crédito| Fecha y hora|Oficina local

obl_fecalt Fecha de alta de la crédito

- Caso 133 Créditos sin fecha
- Var 1. Búsqueda en extracción nacional para completar, si existe se queda la fecha nacional
 - Var 2. Si no hay registro en la extracción nacional se copia la fecha de inicio de operaciones del cliente

Bitácora:

Caso 133|1|0|RFC|DETER|Clave Obl|Fecha de alta| Fecha y hora|Oficina local

Caso 133|2|0|RFC|DETER|Clave Obl| Fecha y hora|Oficina local

obl_fecbaj Fecha de baja de la crédito

- Caso 134 Clientes en baja con créditos activos
- Var 1. Se marcan como baja con la fecha de baja en que se encuentra
 - Se generará registro para validación

Bitácora:

Caso 134|1|0|RFC|DETER|Cve. crédito| Fecha y hora|Oficina local

Tabla NOMBRES

nom_rfca RFC anterior

Caso 135 RFC anterior nulo

- Se borra

Bitácora:

Caso 135|0|0|Registro nombres| Fecha y hora|Oficina local

Caso 136 RFC anterior no existe en local

- Si RFC posterior si existe en la local
- Se genera archivo de dudas a la nacional y se registrará con paternidad confirmada de la Oficina local
 - Var 1. Si en la nacional si existe el cliente, traer e insertar datos de las 11 tablas y registro con paternidad de la local
 - Var 2. Si en la nacional no existe, se registra en bitácora especial con paternidad de la Oficina local y en la local se inventará el registro con los datos del RFC posterior.
 - Var 3. Si el posterior (último cambio de nombre) no es de la Oficina local se borra de la local y se registra con paternidad de la otra Oficina local
 - Var 4. Si el posterior (último cambio de nombre) no existe de borra de la local y se registra sin paternidad

Bitácora:

Caso 136|1|0|RFCA|RFCP|Fecha y hora|Oficina local

Caso 136|2|0|RFCA|RFCP|Fecha y hora|Oficina local

Caso 136|3|0|Registro nombres| Fecha y hora|Oficina local

Caso 136|4|0|Registro nombres| Fecha y hora|Oficina local

Caso 136|5|1|Registro Bajas| Fecha y hora|Oficina local

Caso 136|5|2|Registro Sucesión| Fecha y hora|Oficina local

Caso 136|5|3|Registro Nombres| Fecha y hora|Oficina local

Caso 136|5|4|Registro Liquidación| Fecha y hora|Oficina local

Caso 136|5|5|Registro Suspensión| Fecha y hora|Oficina local

Caso 136|5|6|Registro Clientes| Fecha y hora|Oficina local

Caso 136|5|7|Registro Cédulas| Fecha y hora|Oficina local

Caso 136|5|8|Registro Documentos| Fecha y hora|Oficina local

Caso 136|5|9|Registro Representantes| Fecha y hora|Oficina local

Caso 136|5|10|Registro Externos| Fecha y hora|Oficina local

Caso 136|5|11|Registro Créditos| Fecha y hora|Oficina local

nom_rfcpc RFC nuevo

Caso 137 RFC nuevo nulo

- Busco en la extracción nacional (por RFC anterior)

- Var 1. Si no existe o también es nulo se marca en la nacional para borrar y sin paternidad
- Var 2. Si en la extracción no es nulo se reemplaza

Bitácora:

Caso 137|1|0|RFC anterior| Fecha y hora|Oficina local

Caso 137|2|0|RFC anterior| Fecha y hora|Oficina local

Caso 138 RFC nuevo no existe en local

- Búsqueda en nacional
 - Var 1. Si es de otra Oficina local se borran el y sus cambios de nombre y se asignan a otra Oficina local en la bitácora especial
 - Var 2. Si no existe en nacional se borra el registro de cambio de nombre, de bajas con motivo 14 o 33 y se reactiva el RFC anterior y las créditos asociadas a la fecha del cambio de nombre.

Bitácora:

Caso 138|1|0|Registro nombres| Fecha y hora|Oficina local

Caso 138|2|0|Registro nombres| Fecha y hora|Oficina local

Caso 138|2|1|Registro bajas| Fecha y hora|Oficina local

Caso 138|2|2|Clave crédito|Fecha de baja| Fecha y hora|Oficina local

Caso 138|3|1|Registro Bajas| Fecha y hora|Oficina local

Caso 138|3|2|Registro Sucesión| Fecha y hora|Oficina local

Caso 138|3|3|Registro Nombres| Fecha y hora|Oficina local

Caso 138|3|4|Registro Liquidación| Fecha y hora|Oficina local

Caso 138|3|5|Registro Suspensión| Fecha y hora|Oficina local

Caso 138|3|6|Registro Clientes| Fecha y hora|Oficina local

Caso 138|3|7|Registro Cédulas| Fecha y hora|Oficina local

Caso 138|3|8|Registro Documentos| Fecha y hora|Oficina local

Caso 138|3|9|Registro Representantes| Fecha y hora|Oficina local

Caso 138|3|10|Registro Externos| Fecha y hora|Oficina local

Caso 138|3|11|Registro Créditos| Fecha y hora|Oficina local

nom_fecha Fecha de cambio de RFC

Caso 139 Nombres sin fecha

- Búsqueda en extracción nacional para completar
 - Var 1. Si existe se queda la fecha nacional
 - Var 2. Si no existe fecha se busca en segmento de bajas y se copia fecha
 - Var 3. Si no existe se inicializa con 99/01/01

Bitácora:

Caso 139|1|0|RFC|Fecha y hora|Oficina local

Caso 139|2|0|RFC| Fecha y hora|Oficina local

Caso 139|3|0|RFC| Fecha y hora|Oficina local

- Caso 140 Cambio de nombre en la extracción nacional que no existen en la local
- Insertar toda la información del cliente en nombres de la nacional que no exista en la local que se está procesando
 - En caso de llaves duplicadas se eliminará el registro de la nacional.

Bitácora:

Caso 140|0|Registro nombres nal.| Fecha y hora|Oficina local

Caso 141 Procedimiento de limpieza especial para sucursales

- Registro de sucursales en local sin registro en extracción nacional y viceversa (Caso 49)
- Consolidar estatus de acuerdo al segmento de bajas (Caso 45)

Nota: para todos los casos sólo se registrará en bitácora aquellos casos en los que el estatus se modificó.

Estatus cliente = '0'+
Si está en bajas 00001 1+

- Si cau_movs=' ' y estatus cliente=0 OK
- Var 1|0. Si cau_movs<>' ' y estatus cliente=1 se mapea estatus a cau_movs (ojo con motivo de baja)
- Si cau_movs=' ' y estatus cliente=1
 - Buscar en documentos una reactivación (16,22,54,56) posterior o igual al tipo de movimiento que origina el estatus del cliente
 - Fecha op >= fecha de segmento y no existen movimientos de baja posteriores del mismo día.
 - Var 2|0. Si existe se queda como activo y se borra del segmento (o los segmentos)
 - Var 3|0. Si no existe se mapea el estatus a cau_movs
 - Var 4|0. Si cau_movs<>' ' y estatus cliente=0 se activa

Caso 141|1|0|RFC y determinante|estatus|Fecha y hora|Oficina local

Caso 141|2|0|RFC y determinante|estatus|Fecha y hora|Oficina local

Caso 141|3|0|RFC y determinante|estatus|Fecha y hora|Oficina local

Caso 141|4|0|RFC y determinante|estatus|Fecha y hora|Oficina local

Caso 141|5|0|Segmento bajas| Fecha y hora|Oficina local

Propuesta para limpieza y migración en línea

Esta propuesta se realizará para Oficinas locales con la nueva aplicación en las que se solicita un cambio de domicilio de un cliente que proviene de una oficina local con el sistema anterior.

Esto significa que la información del cliente que se encuentra en la nacional no es confiable.

Es importante recordar que la mayor parte de la información confiable de un cliente se encuentra en la local y no en la nacional.

Uno de los puntos de la limpieza consiste en reemplazar la nacional con información de la local.

Dado que en el sistema anterior se enviará la orden de borrado a la antigua oficina local la información que prevalecerá en el sistema será la que exista en la nacional igual que como ocurre actualmente.

Es importante recalcar que tanto el proceso anterior de confirmación de la nacional como el nuevo proceso, no cuentan con los mecanismos de control necesarios para garantizar el envío y recepción de archivos completos. Dado que no se planea modificar este procedimiento, en el proceso de limpieza en línea se asumirá que en la nacional no existe nada más ni menos que lo que se recibió en el archivo y que si algún dato está incompleto y requiere validación nacional se eliminará la inconsistencia localmente sin consulta a la nacional.

Caso	Descripción	Tabla
Caso 1	RFC nulo	CLIENTE
Caso 2	Determinante nulo	CLIENTE
Caso 7	Módulo nulo, fuera de catálogo e inconsistencias	CLIENTE
Caso 10	Clientes sin nombre	CLIENTE
Caso 16	Clientes sin fecha de alta	CLIENTE
Caso 29	Clientes sin calidad	CLIENTE
Caso 30	Clientes con calidad \diamond 1, 2, 3, 4, 5, 6, 7, 8 y 10	CLIENTE
Caso 33	Clientes con calidad con créditos 123 y 154 marca de controladora	CLIENTE
Caso 34	Clientes sin tipo de sociedad (sólo para matrices)	CLIENTE
Caso 35	Clientes con tipo de sociedad \diamond 0 y \diamond 5	CLIENTE
Caso 36	Clientes con tipo de cliente fuera de catálogo	CLIENTE
Caso 37	Clientes sin tipo de cliente	CLIENTE
Caso 41	Clientes sin actividad	CLIENTE
Caso 43	Clientes con actividad fuera de catálogo	CLIENTE
Caso 45	Actualizar estatus de cliente de acuerdo a su segmento	CLIENTE
Caso 47	Clientes sin estatus	CLIENTE
Caso 48	Clientes con estatus fuera de catálogo	CLIENTE

Caso 46	Oficina local nula	CLIENTE
Caso 50	RFC nulo	BAJAS
Caso 51	Determinante nulo	BAJAS
Caso 52	Bajas sin cliente	BAJAS
Caso 54	Clientes sin motivo de baja	BAJAS
Caso 55	Clientes con motivo fuera de catálogo	BAJAS
Caso 58	RFC nulo	LIQUIDACION
Caso 59	Liquidación sin cliente	LIQUIDACION
Caso 60	Personas físicas en liquidación	LIQUIDACION
Caso 64	Clientes sin fecha	LIQUIDACION
Caso 67	RFC nulo	SUCESION
Caso 68	Sucesión sin cliente	SUCESION
Caso 69	Personas morales en sucesión	SUCESION
Caso 70	Clientes sin fecha	SUCESION
Caso 73	RFC y/o determinante nulo	SUSPENSION
Caso 74	Suspensión sin cliente	SUSPENSION
Caso 75	Clientes sin fecha	SUSPENSION
Caso 78	RFC nulo	REPRESENTANTE
Caso 79	Determinante nulo	REPRESENTANTE
Caso 80	Representante sin cliente	REPRESENTANTE
Caso 83	RFC nulo	CEDULAS
Caso 84	Cédula sin cliente	CEDULAS
Caso 85	Cédulas con letra o número nulo	CEDULAS
Caso 90	Cédulas activas sin fecha alta	CEDULAS
Caso 142	Cédulas con estatus que indica activo y fecha de baja y viceversa	CEDULAS
Caso 92	Documentos sin estación de captura	DOCUMENTOS
Caso 93	Externos sin estación de captura	EXTERNOS
Caso 96	RFC nulo	DOCUMENTOS
Caso 97	RFC nulo	EXTERNOS
Caso 98	Determinante nulo	DOCUMENTOS
Caso 99	Determinante nulo	EXTERNOS
Caso 100	Documentos sin cliente	DOCUMENTOS
Caso 101	Externos sin cliente	EXTERNOS
Caso 102	Módulo nulo	DOCUMENTOS
Caso 103	Módulo nulo	EXTERNOS
Caso 104	Módulo fuera de catálogo	DOCUMENTOS
Caso 105	Módulo fuera de catálogo	EXTERNOS
Caso 106	Documentos sin fecha de operación	DOCUMENTOS
Caso 107	Externos sin fecha de operación	EXTERNOS
Caso 108	Documentos sin fecha de trámite	DOCUMENTOS
Caso 109	Externos sin fecha de trámite	EXTERNOS
Caso 110	Documentos que están tanto en externos como en documentos.	DOCUMENTOS

Caso 111	Registros duplicados en la tabla de externos.	DOCUMENTOS
Caso 114	Documentos sin número de operación	DOCUMENTOS
Caso 115	Externos sin número de operación	EXTERNOS
Caso 118	Motivo nulo	DOCUMENTOS
Caso 119	Motivo nulo	EXTERNOS
Caso 120	Motivo fuera de catálogo	DOCUMENTOS
Caso 121	Motivo fuera de catálogo	EXTERNOS
Caso 122	RFC nulo	CRÉDITOS
Caso 123	Determinante nulo	CRÉDITOS
Caso 124	Créditos sin cliente	CRÉDITOS
Caso 126	Clave nula o fuera de catálogo	CRÉDITOS
Caso 127	Clave en desuso, personas físicas con créditos de morales y viceversa	CRÉDITOS
Caso 132	Clientes con créditos incompatibles (sólo entre créditos activos)	CRÉDITOS
Caso 133	Créditos sin fecha	CRÉDITOS
Caso 134	Clientes en baja con créditos activos	CRÉDITOS
Caso 135	RFC anterior nulo	NOMBRES
Caso 136	RFC anterior no existe en local	NOMBRES
Caso 137	RFC nuevo nulo	NOMBRES
Caso 138	RFC nuevo no existe en local	NOMBRES
Caso 139	Nombres sin fecha	NOMBRES

- En todos los casos la limpieza se registrará en las mismas bitácoras que se generaron para la limpieza local.
- La migración se realizará de acuerdo a las especificaciones generales de migración de la Oficina local.
- La limpieza y migración al nuevo sistema se realizará mediante un proceso general que se ejecutará después de recibir y procesar los archivos de confirmación de la nacional.
- Dicho proceso general invocará, entre otros procesos, el procedimiento llamado **limpia_migra.sh** que será el responsable de limpiar y migar los nuevos cambios de domicilio que provienen de otra Oficina local.
- El procedimiento de limpieza y migración **limpia_migra.sh** es el siguiente:
 - Recibirá como parámetro la fecha del archivo de órdenes de inserción **aaaaddd**
 - Se verifica la existencia de las 11 tablas de limpieza temporal de Sistema actual – RFC

- En caso de que si existan serán borradas
- Se verifica la existencia de las 24 tablas de migración temporal de Nuevo sistema
 - En caso de que existan serán borradas
- Se crean las 11 tablas temporales de Sistema actual y las 24 de Nuevo sistema
- Se consulta el archivo de RFC's a limpiar **dario_oi. aaaaddd**
 - Si no existe se cancela el proceso
- Se carga a una tabla temporal
- Se extraen datos de las 11 tablas de RFC a las tablas temporales de limpieza de datos sobre los RFC's del archivo.
- Se realiza la limpieza sobre tablas temporales
- Si la limpieza no concluye correctamente
 - Se cancela el proceso
- Si la limpieza concluye correctamente
 - Se migra la información de las tablas temporales ya limpias
- Si la migración es correcta
 - Se genera el archivo **todoook** que indica el fin correcto de la migración
 - Se eliminan archivos y tablas de paso
- Si la migración no es correcta
 - Se cancela el proceso
- El proceso general realizará el borrado y reemplazo de todas las bases de datos locales, regionales y nacionales (Sistema actual y Nuevo sistema)

Resultados

Para la limpieza local se prepararon 19 programas

menu1.4gl	Menú principal de limpieza local
verieje.c	Verifica que el equipo cuente con espacio suficiente para iniciar la limpieza
cero.4gl	Respaldo de la base de datos antes de la limpieza y creación de bitácoras
dudas.4gl	Envío de dudas al equipo nacional y espera respuestas
fin3_loc.4gl	Descarga de la base de datos local para reemplazo en la nacional
bitagral.4gl	Crea la bitácora general de casos de limpieza
limpcau.4gl	Limpieza de clientes
limpced.4gl	Limpieza de cédulas
limpdye.4gl	Limpieza de documentos y externos
limpliq.4gl	Limpieza de clientes en liquidación
limpobl.4gl	Limpieza de créditos
limprep.4gl	Limpieza de representantes
limpsles.4gl	Limpieza de sucursales
limpsuc.4gl	Limpieza de clientes en sucesión
nalvsloc.4gl	Compara clientes y datos de la extracción nacional contra la local
resduda.4gl	Con base en el archivo de respuestas se determinan paternidades
tablasfc.4gl	Prepara tablas para carga de la extracción nacional
dud136.4gl	Envío de dudas al equipo nacional y espera respuestas para caso 136
resdudas136.4gl	Con base en el archivo de respuestas se determinan paternidades para caso 136

Para la limpieza nacional se prepararon 12 programas

monito1.4gl	Programa monitor que espera la llegada de archivos de dudas y envía las respuestas
borracomp.4gl	Actualiza las bitácoras de control de dudas
compalr.4gl	Actualiza la bitácora de paternidad especial
act_fin3.4gl	Realiza el borrado de registros en base a la tabla de llaves para la sustitución en nacional
descalrs.4gl	Descarga de datos de la nacional que se compararán en la local
carga_loc.4gl	Carga datos de extracciones de la local en la base de datos nacional
compalr136.4gl	Validaciones para caso 136
creaproc.4gl	Define el trigger para borrado en cadena de un cliente en la base de datos nacional
nacional.4gl	Programa principal para iniciar el proceso de limpieza
restaura.4gl	Restaura la base de datos
tabnal.4gl	Prepara bitácoras y catálogos para limpieza nacional
prepnal.4gl	Respalda datos antes de borrar

Casos por programa de limpieza local

Cero.4gl Respaldo de la base de datos y preparación de bitácoras y catálogos de validación

Inicia detección de casos de paternidad dudosa

Navisloc.4gl Complemento de datos entre nacional y local

49	Cau. en Nal que no existe en Loc	67	RFC nulo (Sucesión)	102	Documentos con CRH nula
57	Bajas en Nal que no existe en Loc	73	RFC nulo (Suspensión)	103	Externos con CRH nula
131	Cred. En Nal que no existe en Loc	78	RFC nulo (Representante)	137	RFC nuevo nulo
116	Docos en Nal que no existe en Loc	79	Determinante nulo (Representante)	111	Registros duplicados en externos
117	Ext. en Nal que no existe en Loc	83	RFC nulo (Cédulas)	10	Cliente sin nombres
86	Cédulas en Nal que no existe en Loc	85	Cédulas con letra o número nulo	52	Bajas sin cliente
140	Nombres en Nal que no existe en Loc	96	RFC nulo (Documentos)	59	Liquidación sin cliente
82	Repres. en Nal que no existe en Loc	92	Documentos con estación de captura	68	Sucesión sin cliente
72	Sucesión en Nal que no existe en Loc	114	Documentos con número de operación nulo	74	Suspensión sin Cliente
77	Suspensión en Nal que no existe en Loc	97	RFC nulo (Externos)	80	Representante sin cliente
66	Liq. en Nal que no existe en Loc	93	Externos con estación de captura	84	Cédulas sin cliente
136	RFC anterior en nombres que no existe en Loc	115	Externos con número de operación nula	100	Documentos sin cliente
1	RFC nulo (Clientes)	98	Determinante nulo (Documentos)	101	Externos sin cliente
2	Determinante nulo (Clientes)	99	Determinante nulo (Externos)	138	RFC posterior en nombres q. no existe en Loc
50	RFC nulo (Bajas)	122	RFC nulo (Créditos)	15	Paternidad de Contribuyente
51	Determinante nulo (Bajas)	135	RFC anterior nulo	124	Créditos sin cliente
58	RFC nulo (Liquidación)	46	Cliente con ALB nula		

Limpli.4g1

Limpieza de liquidadores

60	Personas físicas en liquidación
65	Limpieza del RFC del liquidador

Limpsuc.4g1

Limpieza de Sucesores

69	Personas morales en sucesión
71	Limpieza del RFC del albacea

Limprep.4g1

Limpieza de representantes

81	Limpieza especial para representantes
----	---------------------------------------

Dudas.4g1

Envío de dudas a la nacional y espera resultados de paternidades y datos incompletos

Resduda.4g1

Con base en el archivo de respuestas se determinan paternidades y se complementan los datos

138	RFC posterior en nombres q' no existe en Loc
15	Domicilios local <-> Domicilios nacional (Cliente)
65	Limpieza del RFC del liquidador
71	Limpieza del RFC del albacea
81	Limpieza especial para representantes
38	Tipo en local <-> tipo nacional (Cliente)
54	Cientes sin motivo de baja
55	Cientes sin motivo fuera de catálogo
64	Cliente Sin fecha de inicio de liquidación
70	Cliente sin fecha de inicio de sucesión
75	Cliente sin fecha de inicio de suspensión
56	Cliente sin fecha de baja
90	Cédulas activas sin fecha de alta
106	Doctos. sin fecha de operación
108	Documentos sin fecha de trámite
107	Externos sin fecha de operación
109	Externos sin fecha de trámite
139	Nombres sin fecha de cambio

Inicia limpieza de todos los clientes que quedaron asignados en esa oficina local

Limpcau.4g| Limpieza de clientes

32	Calidad en local <-> Calidad nacional (Cliente)
33	Ciudad con calidad <-> 0 sin obligaciones adecuadas
4	RFC menor a 13 posiciones para personas físicas y 12 para morales
5	Dígito verificador mal calculado
7	Ciudad con crh nula o cambio por su equivalente
16	Ciudad sin fecha de alta
27	Ciudad matriz con alta 10 (apertura de establecimiento)
25	Ciudad sin motivo de alta

26	Ciudad con motivo fuera de catálogo
29	Ciudad sin calidad
30	Ciudad con calidad (quitando los últimos 3 caracteres) <0 y >10
31	Ciudad con grupo económico fuera de catálogo
34	Ciudad sin tipo de sociedad
35	Ciudad con tipo <0 y <->5
37	Ciudad sin tipo
36	Ciudad con tipo fuera de catálogo

41	Ciudad sin actividad preponderante
43	Ciudad con actividad preponderante fuera de catálogo
47	Ciudad sin estatus
48	Ciudad con estatus fuera de catálogo
45	Actualizar estatus de cliente de acuerdo a su segmento
133	Ciudad sin fecha de alta

Limpccd. 4g| Limpieza de cédulas

142	Cédulas cuyo estatus se obtiene de la fecha de alta y baja
-----	--

Limpdye.4g| Limpieza de documentos y externos

104	CRH fuera de catálogo en Documentos
105	CRH fuera de catálogo en Externos
118	Motivo nulo en Documentos
119	Motivo nulo en Externos
120	Motivo fuera de catálogo en Documentos
121	Motivo fuera de catálogo en Externos
110	Documentos que están en externos y viceversa

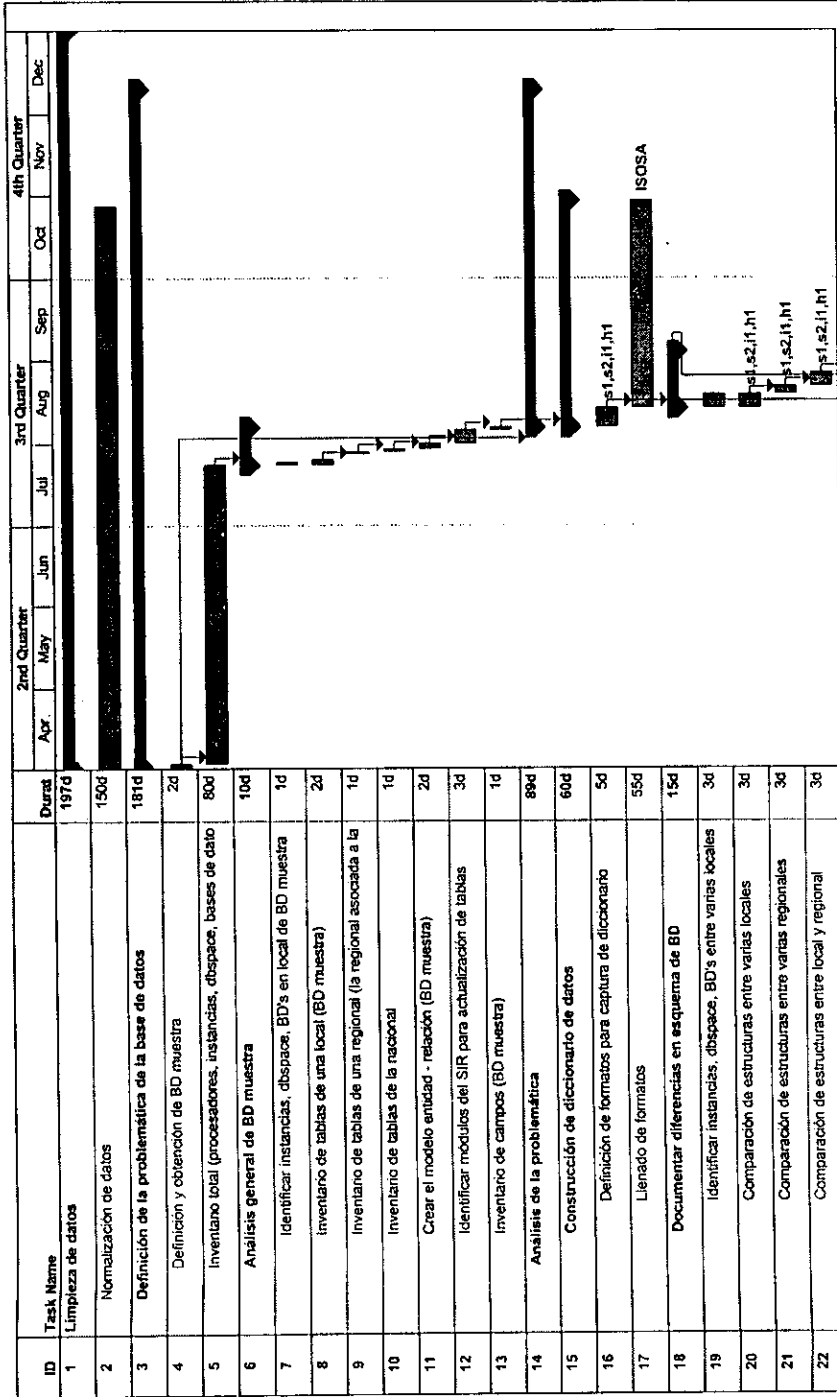
Limpobl.4g| Limpieza de créditos

123	Determinante nulo (Créditos)
125	Ciudad sin créditos
126	Clave nula o fuera de catálogo
127	Clave en desuso
129	Ciudad sin crédito padre
132	Ciudad con créditos incompatibles
134	Ciudad en baja (sin incluir no localizados) con créditos activos

Limppls.4g| Limpieza de sucursales

141	Limpieza especial para sucursales
-----	-----------------------------------

Anexo



Task
 Progress
 Milestone

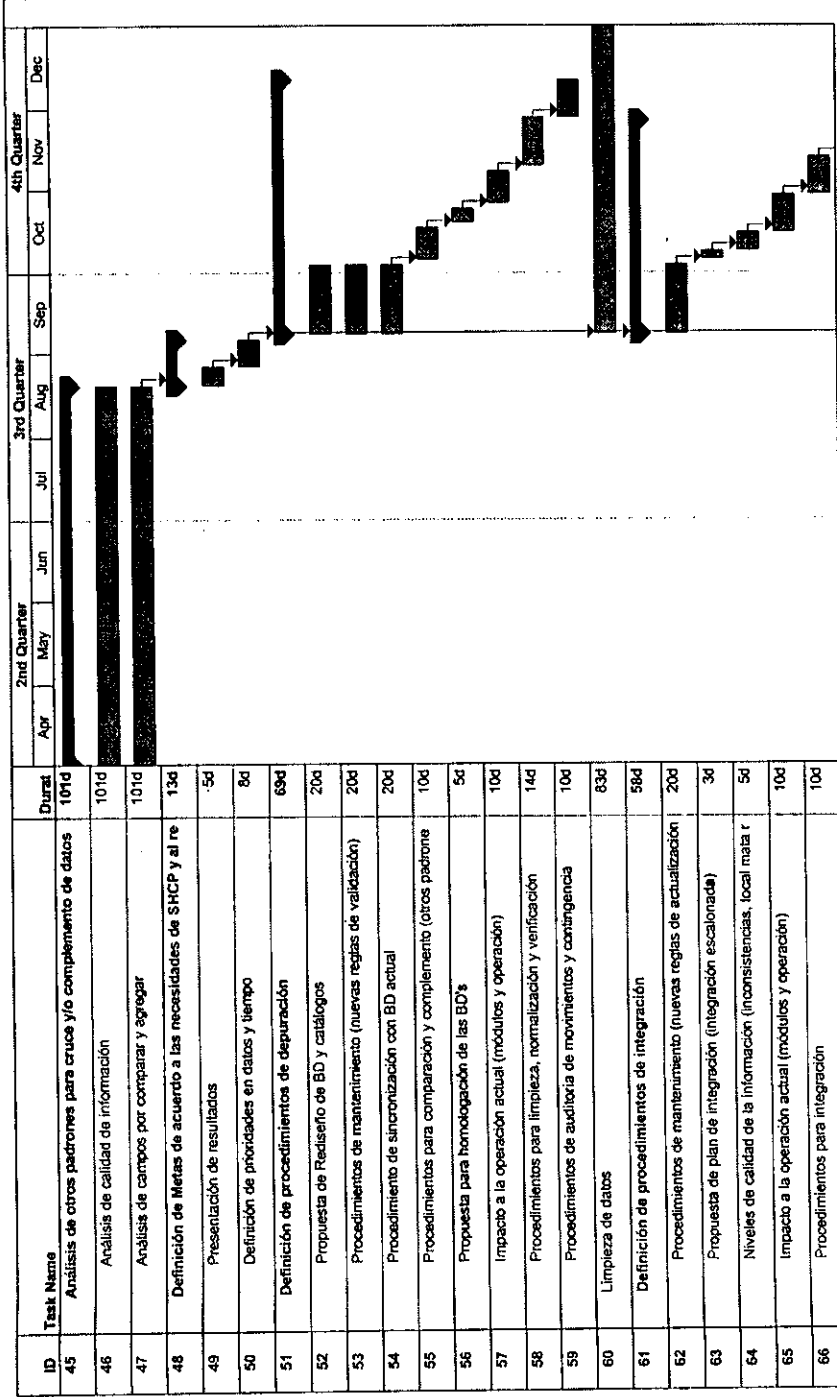
Summary
 Rolled Up Task
 Rolled Up Milestone
 Rolled Up Progress

Project: Limpieza de datos I
 Date: Mon 23/11/98

ID	Task Name	Durat	2nd Quarter			3rd Quarter			4th Quarter				
			Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec		
23	Comparación de estructuras entre regional y nacional	3d											
24	Crear matriz tabla vs nivel (local, regional y nacional)	3d											
25	Inventario y análisis de catálogos (en código, en papel y	24d											
26	Inventario de catálogos reales (locales, regionales, nabo	5d											
27	Identificar fuentes y métodos de actualización	2d											
28	Identificar catálogos por homologar o reestructurar	5d											
29	Inventario de catálogos en código y papel	5d											
30	Identificar fuentes y métodos de actualización	2d											
31	Identificar versiones por homologar o catálogos por crea	5d											
32	Identificar campos y registros en conflicto (movimientos	15d											
33	Identificar campos que no concuerden con catálogos (la	5d											
34	Identificar registros inconsistentes (padres sin hijos, hijo	5d											
35	Validaciones numéricas (pagos vs totales etc.)	5d											
36	Duplicidades o faltantes	15d											
37	Identificar registros en local que no existen en regional y	3d											
38	Identificar registros en regional que no existen en nacion	3d											
39	Pruebas de duplicidad en local	3d											
40	Pruebas de duplicidad en regional	3d											
41	Pruebas de duplicidad nacional	3d											
42	Análisis reglas de validación actuales	84d											
43	Identificación de causas que originan errores o inconsist	68d											
44	Identificación de diferencias en interpretación de valores	15d											

Project: Limpieza de datos I
 Date: Mon 23/11/98

Task: [Bar] Summary: [Bar] Rolled Up Progress: [Bar]
 Progress: [Bar] Rolled Up Task: [Bar]
 Milestone: [Diamond] Rolled Up Milestone: [Diamond]

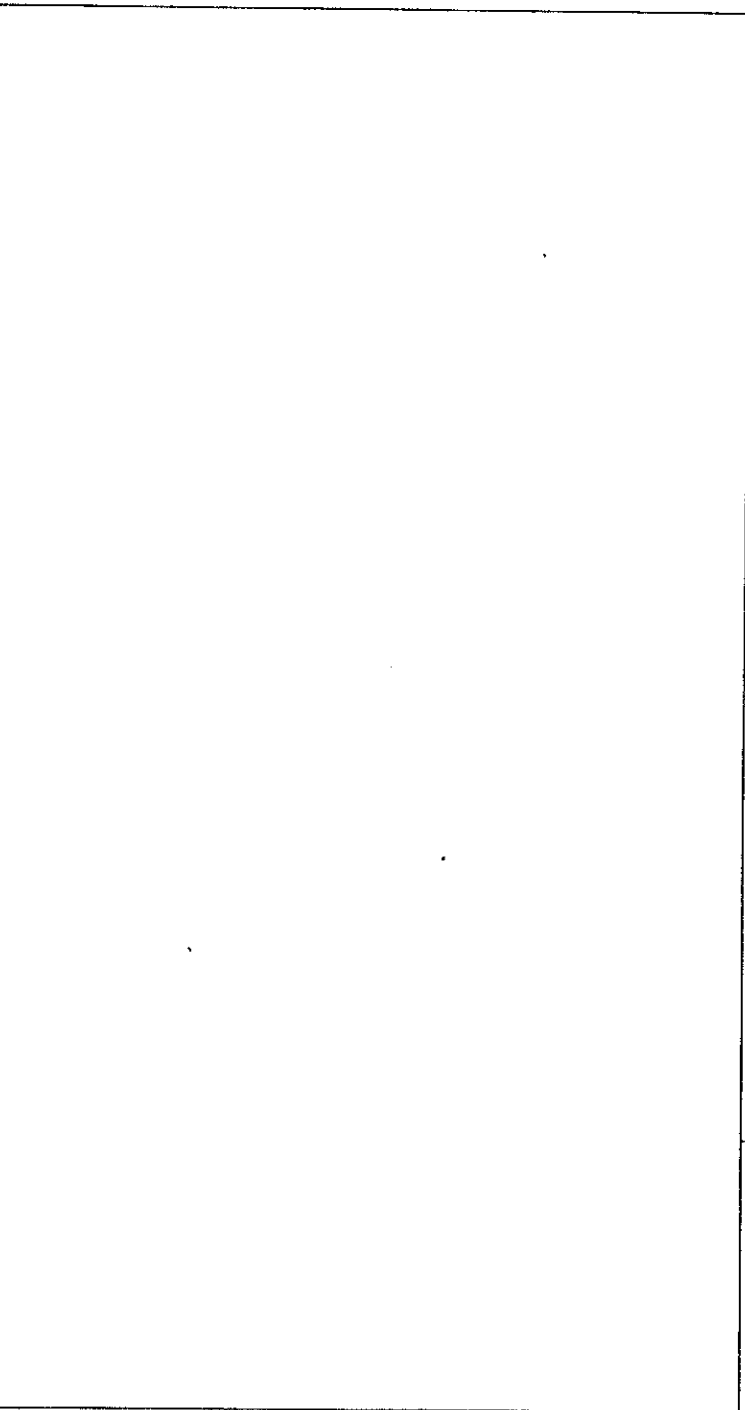


Project: Limpieza de datos I
Date: Mon 23/11/98

Task
 Progress
 Milestone
 Summary
 Rolled Up Task
 Rolled Up Milestone
 Rolled Up Progress

Page 3

ID	Task Name	Durat	2nd Quarter			3rd Quarter			4th Quarter			
			Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
67	Procedimientos de auditoría de movimientos y contingencia	10d										
68	Compatibilización de contribuyentes	35d										



Project: Limpieza de datos I
Date: Mon 23/11/98

Task: [Task Name]

Progress: [Progress Bar]

Milestone: [Milestone Symbol]

Summary: [Summary Bar]

Rolled Up Task: [Rolled Up Task Bar]

Rolled Up Milestone: [Rolled Up Milestone Symbol]

Rolled Up Progress: [Rolled Up Progress Bar]

Programas de limpieza local

VERIEJE.C

```
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#define TAMMAX 50
char linea[TAMMAX];
char varch [40];

pasa ()
{
    char vargv[40];
    char *pespacio;
    popquote(vargv,40);
    pespacio = strchr ( vargv, ' ');
    if ( pespacio )
        *pespacio = 0;

    strcpy(varch,vargv);
    lee_linea();
    if ( linea[0]!='\0' || strstr(linea,"No procede") )
    {
        retint (1);
        return(1);
    }
    retint (0);
    return(1);
} /* end main */
/* ----- funcion lee ----- */
lee_linea()
{
    FILE *fd;
    if ((fd=fopen(varch,"rb"))==NULL)
        linea[0]='\0';
    else
    {
        if ((fgets(linea,TAMMAX,fd))==NULL)
            linea[0]='\0';
        fclose (fd);
    }
}

TABLASRFC:4GL
database shcp

main

select tablename from systables
where tablename='hist_det'

if status <> NOTFOUND then
    drop table hist_det
end if

create table hist_det
(
    cau_rfc char(13),
    cau_deter smallint,
    cau_rsoc varchar(80),
    cau_domi varchar(80),
    cau_codpos char(5),
    cau_zonpos char(2),
    cau_mpio smallint,
    cau_afr smallint,
    cau_entfed smallint,
    cau_ofh char(5),
    cau_fecall date,
```

```
cau_motall char(2),
cau_calcont smallint,
cau_tipsoc smallint,
cau_tipcont char(1),
cau_segjob integer,
cau_cveact char(6),
cau_mesclie smallint,
cau_zondil char(2),
cau_serpos char(1),
cau_telefo varchar(10),
cau_estable smallint,
cau_movs char(1)
) IN dbsir
revoke all on hist_det from "public"

select tablename from systables
where tablename='bajnal'

if status <> NOTFOUND then
    drop table bajnal
end if

create table bajnal
(
    baj_rfc char(13),
    baj_deter smallint,
    baj_fec date,
    baj_cvebaj char(2)
) IN dbsir

revoke all on bajnal from "public"

select tablename from systables
where tablename='obnal'

if status <> NOTFOUND then
    drop table obnal
end if

create table obnal
(
    obl_rfc char(13),
    obl_deter smallint,
    obl_clave smallint,
    obl_fecall date,
    obl_fecbaj date
) IN dbsir

revoke all on obnal from "public"

select tablename from systables
where tablename='docnal'

if status <> NOTFOUND then
    drop table docnal
end if

create table docnal
(
    doc_rfc char(13),
    doc_deter smallint,
    doc_nofh char(5),
    doc_fechop date,
    doc_mofech date,
    doc_estcap smallint,
    doc_nope serial not null,
    doc_tipfor char(3),
```

```

doc_motmov char(2),
doc_claope smallint,
doc_nord integer,
doc_resver char(1),
doc_movi char(1)
) IN dbsir
revoke all on docnal from "public"

```

```

select tabname from systables
where tabname='extrnal'

```

```

if status <> NOTFOUND then
    drop table extrnal
end if

```

```

create table extrnal
(
    ext_rfc char(13),
    ext_deter smallint,
    ext_nofh char(5),
    ext_fechop date,
    ext_mofech date,
    ext_estcap smallint,
    ext_nope integer,
    ext_tipfor char(3),
    ext_motmov char(2),
    ext_claope smallint,
    ext_nord integer,
    ext_resver char(1)
) IN dbsir
revoke all on extrnal from "public"

```

```

select tabname from systables
where tabname='cednal'

```

```

if status <> NOTFOUND then
    drop table cednal
end if

```

```

create table cednal
(
    ced_rfc char(13),
    ced_let char(1),
    ced_num integer,
    ced_fecall date,
    ced_fecbaj date,
    ced_mot char(1)
) IN dbsir
revoke all on cednal from "public"

```

```

select tabname from systables
where tabname='nomnal'

```

```

if status <> NOTFOUND then
    drop table nomnal
end if

```

```

create table nomnal
(
    nom_rfca char(13),
    nom_rfcj char(13),
    nom_fecha date
) IN dbsir
revoke all on nomnal from "public"

```

```

select tabname from systables
where tabname='repnal'

```

```

if status <> NOTFOUND then
    drop table repnal
end if
create table repnal
(
    rep_rfcau char(13),
    rep_detcau smallint,
    rep_rfcj char(13),
    rep_nomrep varchar(80),
    rep_domrep varchar(80),
    rep_telef varchar(10)
) IN dbsir
revoke all on repnal from "public"

```

```

select tabname from systables
where tabname='sucnal'

```

```

if status <> NOTFOUND then
    drop table sucnal
end if
create table sucnal
(
    suc_rfc char(13),
    suc_fechi date,
    suc_nomalb varchar(80),
    suc_rfcalb char(13),
    suc_domalb varchar(80)
) IN dbsir
revoke all on sucnal from "public"

```

```

select tabname from systables
where tabname='susnal'

```

```

if status <> NOTFOUND then
    drop table susnal
end if
create table susnal
(
    sus_rfc char(13),
    sus_deter smallint,
    sus_fecsus date,
    sus_normaut varchar(80),
    sus_domaut varchar(80)
) IN dbsir
revoke all on susnal from "public"

```

```

select tabname from systables
where tabname='liqnal'

```

```

if status <> NOTFOUND then
    drop table liqnal
end if
create table liqnal
(
    liq_rfc char(13),
    liq_fechi date,
    liq_nomialq varchar(80),
    liq_rfdliq char(13),
    liq_domliq varchar(80),
    liq_telliq varchar(10)
) IN dbsir
revoke all on liqnal from "public"

```

```

end main

```

RESUDA136.4GL

```
database shcp
globals
  define x_alr char(2), hora char(30)
  define xx_rfc char(13), xx_deter smallint,
        xx_caso smallint, xx_var smallint
  define g_alr smallint, g_ofh char(5)
  define cau0 record like causante.*
  define cau136 record
    variante,
    subvar,
    cau_rfc char(13),
    cau_deter smallint
  end record
  define x_caso char(300)
  define f_mayor date
  define rfcpaux char(13)
  define cau1, cau2 char(13)
end globals
main
  --- obtiene la ALR al ejecutar estcau.4ge '/u/sirair/alr' ---
  let x_alr= arg_val(1)
  defer quit
  defer interrupt

  declare c_ofh scroll cursor for
  select cve_ofh, afr
  from cat_ofh where edo=x_alr
  open c_ofh
  fetch first c_ofh into g_ofh, g_alr
  close c_ofh
  let g_ofh=g_ofh clipped, '00'

  update casos set inicio=CURRENT where caso=208
  call caso_136()
  call verinvo()
  update casos set fin=CURRENT where caso=208

end main

function caso_136()
  declare c136 cursor for
  select variante, subvar, rfc, deter
  into cau136.*
  from res_casos1
  where num_caso=136
  foreach c136
    if cau136.variante=1 then
      insert into bita_3
      values(136, 1, 0, cau136.cau_rfc, 0, current, x_alr, "", "")
      call
      cartablas(136, cau136.variante, cau136.cau_rfc, cau136.cau_deter
      )
      else
      if cau136.variante=2 then
      select cau_rfc from causante where
      cau_rfc=cau136.cau_rfc
      and cau_deter=0
      if status=NOTFOUND then
      #crea causante
      insert into bita_3
      values(136, 2, 0, cau136.cau_rfc, 0, current, x_alr, "", "")
      initialize rfcpaux to null
      select nom_rfc into rfcpaux
      from nombres where nom_rfc=cau136.cau_rfc
      if rfcpaux is not null then
      select * into cau0.* from causante
```

```
where cau_rfc=rfcpaux and cau_deter=0
if status<>NOTFOUND then
  let cau0.cau_rfc=cau136.cau_rfc
  insert into causante
  values(cau0.*)
else
  initialize f_mayor to null
  select baj_fec into f_mayor
  from bajas
  where baj_rfc=cau136.cau_rfc and baj_deter=0
  and (baj_cvebaj='14' or baj_cvebaj='33')
  if f_mayor is NULL then
    let f_mayor='88/01/01'
  end if
  insert into causante values (cau136.cau_rfc, 0,
  'CAUSANTE GENERADO POR LA
  LIMPIEZA;', "", '00',
  "", g_alr, "", g_ofh, f_mayor, '00', 0, 0, '8', 0, '999999',
  x_alr, '00', "", '0000000', 0, '')
  end if
else # else de que no encontro el rfc
  initialize f_mayor to null
  select baj_fec into f_mayor
  from bajas
  where baj_rfc=cau136.cau_rfc and baj_deter=0
  and (baj_cvebaj='14' or baj_cvebaj='33')
  if f_mayor is NULL then
    let f_mayor='88/01/01'
  end if
  insert into causante values (cau136.cau_rfc, 0,
  'CAUSANTE GENERADO POR LA LIMPIEZA;', "", '00',
  "", g_alr, "", g_ofh, f_mayor, '00', 0, 0, '8', 0, '999999',
  x_alr, '00', "", '0000000', 0, '')
  end if
end if # end de que ya se encontraba en la local
else
  # variante 3 se mando borrar desde la nacional
  # por que pertenecia a otra ALR
  insert into bita_3
  values(136, 3, 0, cau136.cau_rfc, 0, current, x_alr, "", "")

  let xx_rfc=cau136.cau_rfc
  let xx_deter=0
  let xx_caso=136
  let xx_var=5
  call bor_causante()
  call bor_obligacion()
  call bor_cedulas()
  call bor_representa()
  call bor_bajas()
  call bor_liquid()
  call bor_suce()
  call bor_suspen()
  call bor_nombre()
  call bor_documentos()
  call bor_extemos()
  end if
end if
end foreach
close c136
end function

function cartablas(casox, varx, rfcx, deterx)
define casox, varx, deterx smallint, rfcx char(13)

select a.cau_rfc, a.cau_deter, a.cau_rsoc, a.cau_domi,
a.cau_codpos, a.cau_zonpos, a.cau_mpio, a.cau_afr, a.cau_entfed,
```

```

a.cau_ofh,a.cau_fecalt,a.cau_motalt,a.cau_calcont,
a.cau_tipsoc,a.cau_tipcont,a.cau_secgob,a.cau_cveact

a.cau_mescie,a.cau_zondil,a.cau_serpos,a.cau_telefo,
a.cau_estable,a.cau_movs from bita_1 a
  where a.num_caso=casox and a.variante=varx
  and a.cau_rfc=rfcx and a.cau_deter=deterx
  and not exists( select b.cau_rfc from causante b
  where b.cau_rfc=a.cau_rfc and
b.cau_deter=a.cau_deter)
  into temp cauxx with no log
  insert into causante select * from cauxx
  drop table cauxx

select
a.obl_rfc,a.obl_deter,a.obl_clave,a.obl_fecalt,a.obl_fecbaj
from bita_16 a where a.num_caso=casox and
a.variante=varx
  and a.obl_rfc=rfcx and a.obl_deter=deterx
  and not exists( select b.obl_rfc from obligacion b
  where b.obl_rfc=a.obl_rfc and b.obl_deter=a.obl_deter
  and b.obl_clave = a.obl_clave)
  into temp obbxx with no log
  insert into obligacion select * from obbxx
  drop table obbxx

select a.ced_rfc,a.ced_let,a.ced_num,a.ced_fecalt,
a.ced_fecbaj,a.ced_mot,a.ced_cti from bita_13 a
  where a.num_caso=casox and a.variante=varx
  and a.ced_rfc=rfcx
  and not exists( select b.ced_rfc from cedulas b
  where b.ced_let=a.ced_let and b.ced_num=a.ced_num)
  into temp cedxx with no log
  insert into cedulas select * from cedxx
  drop table cedxx

select a.rep_rfcau,a.rep_detcau,a.rep_rfcrep,
a.rep_nomrep,a.rep_domrep,a.rep_telef from bita_12 a
  where a.num_caso=casox and a.variante=varx
  and a.rep_rfcau=rfcx and a.rep_detcau=deterx
  and not exists( select b.rep_rfcau from representa b
  where b.rep_rfcau=a.rep_rfcau and
b.rep_detcau=a.rep_detcau)
  into temp repxx with no log
  insert into representa select * from repxx
  drop table repxx

select a.baj_rfc,a.baj_deter,a.baj_fec,a.baj_cvebaj from
bita_6 a
  where a.num_caso=casox and a.variante=varx
  and a.baj_rfc=rfcx and a.baj_deter=deterx
  and not exists(select b.baj_rfc from bajas b
  where b.baj_rfc=a.baj_rfc and b.baj_deter=a.baj_deter)
  into temp bajxx with no log
  insert into bajas select * from bajxx
  drop table bajxx

select a.liq_rfc,a.liq_fechi,a.liq_nomliq,a.liq_rfcliq,
a.liq_domliq,a.liq_telliq from bita_9 a
  where a.num_caso=casox and a.variante=varx
  and a.liq_rfc=rfcx
  and not exists( select b.liq_rfc from liquidacion b
  where b.liq_rfc=a.liq_rfc)
  into temp liqxx with no log
  insert into liquidacion select * from liqxx
  drop table liqxx

select a.suc_rfc,a.suc_fechi,a.suc_nomalb,
a.suc_rfcalb,a.suc_domalb from bita_7 a
  where a.num_caso=casox and a.variante=varx
  and a.suc_rfc=rfcx
  and not exists( select b.suc_rfc from sucesion b
  where b.suc_rfc=a.suc_rfc)
  into temp sucxx with no log
  insert into sucesion select * from sucxx
  drop table sucxx

select a.sus_rfc,a.sus_deter,a.sus_fecsus,a.sus_nomaui,
a.sus_domaut from bita_10 a
  where a.num_caso=casox and a.variante=varx
  and a.sus_rfc=rfcx and a.sus_deter=deterx
  and not exists( select b.sus_rfc from suspension b
  where b.sus_rfc=a.sus_rfc and
b.sus_deter=a.sus_deter)
  into temp susxx with no log
  insert into suspension select * from susxx
  drop table susxx

select a.nom_rfca,a.nom_rfcp,a.nom_fecha from bita_8 a
  where a.num_caso=casox and a.variante=varx
  and a.nom_rfcp=rfcx
  and not exists( select b.nom_rfcp from nombres b
  where b.nom_rfcp=a.nom_rfcp)
  and not exists( select c.nom_rfca from nombres c
  where c.nom_rfca=a.nom_rfca)
  into temp nomxx with no log
  insert into nombres select * from nomxx
  drop table nomxx

select a.doc_rfc,a.doc_deter,a.doc_nofh,a.doc_fechop,
a.doc_mofech,a.doc_estcap,a.doc_nope,a.doc_tipfor,a.doc_mot
mov,
a.doc_claope,a.doc_nord,a.doc_resver,a.doc_movi from
bita_14 a
  where a.num_caso=casox and a.variante=varx
  and a.doc_rfc=rfcx and a.doc_deter=deterx
  and not exists( select b.doc_rfc from documentos b
  where b.doc_rfc=a.doc_rfc and
b.doc_deter=a.doc_deter and
b.doc_nofh=a.doc_nofh and
b.doc_fechop=a.doc_fechop and
b.doc_nope=a.doc_nope)
  into temp docxx with no log
  insert into documentos select * from docxx
  drop table docxx

select a.ext_rfc,a.ext_deter,a.ext_nofh,a.ext_fechop,
a.ext_mofech,a.ext_estcap,a.ext_nope,a.ext_tipfor,
a.ext_motmov,a.ext_claope,a.ext_nord,
a.ext_resver from bita_15 a
  where a.num_caso=casox and a.variante=varx
  and a.ext_rfc=rfcx and a.ext_deter=deterx
  and not exists( select b.ext_rfc from externos b
  where b.ext_rfc=a.ext_rfc and b.ext_deter=a.ext_deter)
  and
  b.ext_nofh=a.ext_nofh and b.ext_fechop=a.ext_fechop)
  and
  b.ext_nope=a.ext_nope)
  into temp extxx with no log
  insert into externos select * from extxx
  drop table extxx

end function

function bor_obligacion()
  define y_inst char(300)

```



```

let y_inst=insert into bita_16 select
'xx_caso','xx_var',11,current,
x_alr,'* from obligacion where obl_rfc="" ,xx_rfc clipped,
"" and obl_deter=xx_deter
prepare y_sus5 from y_inst
execute y_sus5
-- elimina
delete from obligacion
where obl_rfc=xx_rfc
and obl_deter=xx_deter
end function #

```

```

function bor_externos()
define y_inst char(300)
let y_inst=insert into bita_15 select
'xx_caso','xx_var',10,current,
x_alr,'* from externos where ext_rfc="" ,xx_rfc clipped,
"" and ext_deter=xx_deter
prepare y_sus4 from y_inst
execute y_sus4
-- elimina
delete from externos
where ext_rfc=xx_rfc
and ext_deter=xx_deter
end function #

```

```

function bor_representa()
define y_inst char(300)
let y_inst=insert into bita_12 select
'xx_caso','xx_var',9,current,
x_alr,'* ,"" from representa where rep_rfcca="" ,xx_rfc
clipped,
"" and rep_detcau=xx_deter
prepare y_sus3 from y_inst
execute y_sus3
-- elimina
delete from representa
where rep_rfcca=xx_rfc
and rep_detcau=xx_deter
end function #

```

```

function bor_documentos()
define y_inst char(300)
let y_inst=insert into bita_14 select
'xx_caso','xx_var',8,current,
x_alr,'* from documentos where doc_rfc="" ,xx_rfc
clipped,
"" and doc_deter=xx_deter
prepare y_sus2 from y_inst
execute y_sus2
-- elimina
delete from documentos
where doc_rfc=xx_rfc
and doc_deter=xx_deter
end function #

```

```

function bor_cedulas()
define y_inst char(300)
let y_inst=insert into bita_13 select
'xx_caso','xx_var',7,current,
x_alr,'* from cedulas where ced_rfc="" ,xx_rfc clipped, ""
prepare y_sus1 from y_inst
execute y_sus1
-- elimina
delete from cedulas
where ced_rfc=xx_rfc
end function #

```

```

function bor_causante()
define y_inst char(300)

```

```

let y_inst=insert into bita_1 select
'xx_caso','xx_var',6,current,
x_alr,'* from causante where cau_rfc="" ,xx_rfc clipped,
"" and cau_deter=xx_deter
prepare y_sus from y_inst
execute y_sus
-- elimina
delete from causante
where cau_rfc=xx_rfc
and cau_deter=xx_deter
end function

```

```

function bor_bajas()
define y_inst char(300)
let y_inst=insert into bita_6 select 'xx_caso','xx_var,
',1,current,'x_alr,'* from bajas where baj_rfc="" ,
xx_rfc clipped, "" and baj_deter=xx_deter
prepare y_baj from y_inst
execute y_baj
---- Elimina la baja
delete from bajas
where baj_rfc=xx_rfc and
baj_deter=xx_deter
end function

```

```

function bor_liquid()
define y_inst char(300)
let y_inst=insert into bita_9 select
'xx_caso','xx_var',4,current,'x_alr,
',' ,"" ,"" from liquidacion where liq_rfc="" ,
xx_rfc clipped, ""
prepare y_liq from y_inst
execute y_liq
-- elimina la liquidacion
delete from liquidacion
where liq_rfc=xx_rfc
end function

```

```

function bor_suce()
define y_inst char(300)
let y_inst=insert into bita_7 select
'xx_caso','xx_var',2,current,'x_alr,
',' ,"" ,"" from sucesion where suc_rfc="" ,
xx_rfc clipped, ""
prepare y_suc from y_inst
execute y_suc
-- elimina la sucesion
delete from sucesion
where suc_rfc=xx_rfc
end function

```

```

function bor_suspen()
define y_inst char(300)
let y_inst=insert into bita_10 select
'xx_caso','xx_var',5,current,
x_alr,'* from suspension where sus_rfc="" ,xx_rfc clipped,
"" and sus_deter=xx_deter
prepare y_sus9 from y_inst
execute y_sus9
-- elimina la suspension
delete from suspension
where sus_rfc=xx_rfc
and sus_deter=xx_deter
end function

```

```

function bor_nombre()
define y_inst char(300)
let y_inst=insert into bita_8 select
'xx_caso','xx_var',3,current,'x_alr,
',' * from nombres where nom_rfcp="" ,xx_rfc clipped, ""
prepare y_nom from y_inst

```

```

execute y_nom
-- elimina la supencion
delete from nombres
where nom_rfc=xx_rfc
end function
function verinv()
select ln,nom_rfca,ln.nom_rfc from nombres ln
where not exists (select cau_rfc from causante
where cau_rfc=ln.nom_rfca and cau_deter=0)
into temp tb136u with no log
declare c136u cursor for select * from tb136u
foreach c136u into cau1,cau2
insert into bita_3 values(136,2,0,cau1,0,current,x_alr,"")
select * into cau0.* from causante
where cau_rfc=cau2 and cau_deter=0
if status <> NOTFOUND then
let cau0.cau_rfc=cau1
insert into causante
values(cau0.*)
else
initialize f_mayor to null
select baj_fec into f_mayor
from bajas where baj_rfc=cau1 and baj_deter=0
and (baj_cvebaj='14' or baj_cvebaj='33')
if f_mayor is NULL then
let f_mayor='88/01/01'
end if
insert into causante values (cau1,0,
'CAUSANTE GENERADO POR LA
LIMPIEZA',"", '00',
",g_alr,"g_ofh,f_mayor','00',0,0,'8',0,'999999',
x_alr,'00',"',0000000',0,'')
end if
end foreach
close c136u
drop table tb136u
end function

```

RESUDA.4GL

```

database shcp
globals
define x_alr char(2), hora char(30)
define cauDn record
num_caso smallint,
variante smallint,
subvar smallint,
alr smallint,
rfc char(13),
deter smallint,
rsoc char(80),
domi char(80),
telefono char(10),
rfc_dupli char(13),
ofh char(5),
afr smallint
end record
define bitrec record
rfc char(13),
deter smallint,
ofh_loc char(5),
alr_loc smallint,
arr_loc smallint,
ofh_nal char(5),
alr_nal smallint,
arr_nal smallint,
ofh_queda char(5),
alr_queda smallint,
arr_queda smallint,
alr_origen smallint
end record
define xx_rfc char(13), xx_deter smallint,

```

```

xx_caso smallint, xx_var smallint
define g_alr smallint,g_ofh char(5)
end globals
main
-- obtiene la ALR al ejecutar estcau.4ge `u/siralr/alr` --
let x_alr=arg_val(1)
defer quit
defer interrupt

declare c_ofh scroll cursor for
select cve_ofh,afr
from cat_ofh where edo=x_alr
open c_ofh
fetch first c_ofh into g_ofh,g_alr
close c_ofh
let g_ofh=g_ofh clipped,'00'

update casos set inicio=CURRENT where caso=208
call caso_138()
call caso_15(x_alr)
call caso_65_REG()
call caso_71_REG()
call caso_81_REG()
update casos set inicio = current where caso = 38
call caso_38()
update casos set fin = current where caso = 38
update casos set inicio = current where caso = 54 or caso = 55
Call caso_54_55()
update casos set fin = current where caso = 54 or caso = 55
update casos set inicio = current where caso = 64
call caso_64()
update casos set fin = current where caso = 64
update casos set inicio = current where caso = 70
call caso_70()
update casos set fin = current where caso = 70
update casos set inicio = current where caso = 75
call caso_75()
update casos set fin = current where caso = 75
update casos set inicio = current where caso = 90
call caso_90()
update casos set fin = current where caso = 90
update casos set inicio = current where caso = 106 or caso =
108
call caso106_108()
update casos set fin = current where caso = 106 or caso =
108
update casos set inicio = current where caso = 107 or caso =
109
call caso107_109()
update casos set fin = current where caso = 107 or caso =
109
update casos set inicio = current where caso = 139
call caso_139()
update casos set fin = current where caso = 139
update casos set fin=CURRENT where caso=208
end main

```

----- INSERTA CASO 106,108 DOCUMENTOS -----

```

function caso106_108()
define fechop date
define mofech date
define x_doctos record like documentos.*
id float
declare c106108 cursor for
select *,rowid into x_doctos.*,id
from documentos where doc_fechop is null or doc_mofech
is null
foreach c106108

```

```

if x_doctos.doc_fechop is null and x_doctos.doc_mofech is null
then
insert into bita_17 values(106,0,0,current,x_alr,
x_doctos.doc_rfc,x_doctos.doc_deter,x_doctos.doc_estcap,
x_doctos.doc_nope,"","")
insert into bita_17 values(108,0,0,current,x_alr,
x_doctos.doc_rfc,x_doctos.doc_deter,x_doctos.doc_estcap,
x_doctos.doc_nope,"","")
select doc_fechop,doc_mofech into fechop,mofech from docnal
where doc_rfc=x_doctos.doc_rfc and
doc_deter=x_doctos.doc_deter
and doc_nofh=x_doctos.doc_nofh and
doc_nope=x_doctos.doc_nope and
(doc_fechop is not null or doc_mofech is not null)
if status <> notfound then
if fechop is not null and mofech is not null then
update documentos set doc_fechop = fechop,
doc_mofech =
mofech where rowid=id
insert into bita_17 values(106,1,0,current,x_alr,
x_doctos.doc_rfc,x_doctos.doc_deter,x_doctos.doc_estcap,
x_doctos.doc_nope,"","")
insert into bita_17 values(108,1,0,current,x_alr,
x_doctos.doc_rfc,x_doctos.doc_deter,x_doctos.doc_estcap,
x_doctos.doc_nope,"","")
else
if fechop is null then
update documentos set doc_fechop =
mofech ,
doc_mofech = mofech where rowid=id
insert into bita_17 values(108,1,0,current,x_alr,
x_doctos.doc_rfc,x_doctos.doc_deter,x_doctos.doc_estcap,
x_doctos.doc_nope,"","")
else
update documentos set doc_fechop =
fechop ,
doc_mofech = fechop where rowid=id
insert into bita_17 values(106,1,0,current,x_alr,
x_doctos.doc_rfc,x_doctos.doc_deter,x_doctos.doc_estcap,
x_doctos.doc_nope,"","")
end if
end if
else
update documentos set doc_fechop = '99/01/01' ,
doc_mofech = '99/01/01' where rowid=id
insert into bita_17
values(106,2,0,current,x_alr,x_doctos.doc_rfc,
x_doctos.doc_deter,x_doctos.doc_estcap,x_doctos.doc_nope,"",
"")
insert into bita_17
values(108,2,0,current,x_alr,x_doctos.doc_rfc,
x_doctos.doc_deter,x_doctos.doc_estcap,x_doctos.doc_nope,"",
"")
end if
else
if x_doctos.doc_fechop is null then
update documentos set doc_fechop =
x_doctos.doc_mofech
where rowid=id
insert into bita_17
values(108,1,0,current,x_alr,x_doctos.doc_rfc,
x_doctos.doc_deter,x_doctos.doc_estcap,x_doctos.doc_nope,"",
"")
else

```

```

update documentos set doc_mofech =
x_doctos.doc_fechop
where rowid=id
insert into bita_17 values(106,1,0,current,x_alr,
x_doctos.doc_rfc,x_doctos.doc_deter,x_doctos.doc_estcap,
x_doctos.doc_nope,"","")
end if
end if
end foreach
end function

```

----- INSERTA CASO 107,109 Externos -----

```

function caso107_109()
define fechop date
define mofech date
define x_extnos record like externos. ,
id float
declare c107109 cursor for
select *,rowid into x_extnos.," id
from externos
where ext_fechop is null or ext_mofech is null
foreach c107109
if x_extnos.ext_fechop is null and x_extnos.ext_mofech is null
then
insert into bita_17
values(107,0,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"","")
insert into bita_17
values(109,0,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"","")
select ext_fechop,ext_mofech into fechop,mofech from
external
where ext_rfc=x_extnos.ext_rfc and
ext_deter=x_extnos.ext_deter
and ext_nofh=x_extnos.ext_nofh and
ext_nope=x_extnos.ext_nope and
(ext_fechop is not null or ext_mofech is not null)
if status <> notfound then
if fechop is not null and mofech is not null then
update externos set ext_fechop = fechop,
ext_mofech =
mofech where rowid=id
insert into bita_17
values(107,1,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"","")
insert into bita_17
values(109,1,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"","")
else
if fechop is null then
update externos set ext_fechop = mofech ,
ext_mofech =
mofech where rowid=id
insert into bita_17
values(109,1,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"","")
else

```

```

update externos set ext_fechop = fechop ,
ext_mofech =
fechop where rowid=id
insert into bita_17
values(107,1,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"",",")
end if
else
update externos set ext_fechop = '99/01/01' ,
ext_mofech = '99/01/01' where rowid=id
insert into bita_17
values(107,2,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"",",")
insert into bita_17
values(109,2,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"",",")
end if
else
if x_extnos.ext_fechop is null then
update externos set ext_fechop =
x_extnos.ext_mofech where rowid=id
insert into bita_17
values(109,1,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"",",")
else
update externos set ext_mofech =
x_extnos.ext_fechop where rowid=id
insert into bita_17
values(107,1,0,current,x_alr,x_extnos.ext_rfc,
x_extnos.ext_deter,x_extnos.ext_estcap,
x_extnos.ext_nope,"",",")
end if
end if
end foreach
end function

```

----- INSERTA CASO 139 NOMBRES -----

```

function caso_139()
define x_inst char(430),id float,
l_mx record like nombres.*,
fecalt date,x_vari smallint
-- trae las fechas de alta de nombres que falten en la local de la
nacional
declare crs_139 cursor for
select *,rowid into i_mx.*,id from nombres
WHERE nom_fecha IS NULL
foreach crs_139
select nom_fecha into fecalt
from nominal where nom_rfcp = l_mx.nom_rfcp
if status <> notfound and fecalt is not null then
let x_vari=1
else

```

```

select baj_fec into fecalt from bajas
where baj_rfc=l_mx.nom_rfc and
baj_deter=0
and (baj_cvebaj='14' or baj_cvebaj='33')
if status<>NOTFOUND then
let x_vari=2
else
let x_vari='3'
let fecalt='99/01/01'
end if
end if
insert into bita_8 values(139,x_vari,0,current,x_alr,
l_mx.nom_rfc,l_mx.nom_rfc,l_mx.nom_fecha)
update nombres set nom_fecha = fecalt
where rowid=id
end foreach
close crs_139
end function # fin de la funcion

function caso_75()
define rec_sus record
rfc char(13),
deter smallint,
fec date
end record
define x_que char(500)
let x_que="insert into
Bita_10(num_caso,variante,subvar,sus_rfc,sus_deter,"
" fecha_hora,sus_fecsus,alr)",
* select
75,1,0,a.sus_rfc,a.sus_deter,CURRENT,b.sus_fecsus,"x_alr,
" from suspension a, susnal b",
" where a.sus_rfc=b.sus_rfc and a.sus_deter=b.sus_deter and ",
" a.sus_fecsus IS NULL and b.sus_fecsus is Not Null"
prepare sus_sfch from x_que
execute sus_sfch

declare crs_suspen cursor for
select sus_rfc,sus_deter,sus_fecsus
into rec_sus."
from bita_10
where num_caso=75 and variante=1 and subvar=0

foreach crs_suspen
update suspension
set sus_fecsus=rec_sus.fech
where sus_rfc=rec_sus.rfc
and sus_deter=rec_sus.deter
end foreach
call suspenloc()

end function

function suspenloc()
define strfc char(13),fechsus date
define intdeter smallint

declare crs_suspen1 cursor for
select sus_rfc,sus_deter
into strfc,intdeter
from suspension
where sus_fecsus is null

foreach crs_suspen1
initialize fechsus to null

```

-- a qui cual es fecha de liquidacion y cual es la clave de la operacion

```
select max(doc_fechop) into fechsus
from documentos a
where a.doc_rfc=strffc and a.doc_deter=intdeter and
(a.doc_motmov='21'
or a.doc_motmov='43') and a.doc_fechop is not null
if fechsus is not null then
--variante 2.2
--actualiza la fecha de inicio de suspension
update suspension
set sus_fecsus=fechsus
where sus_rfc=strffc
and sus_deter=intdeter
--graba a bitacora que si se encontro
insert into
Bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
values(75,2,0,strffc,intdeter,CURRENT,x_alr)
else
select max(ext_fechop) into fechsus
from externos a
where a.ext_rfc=strffc and a.ext_deter=intdeter and
(a.ext_motmov='21' or a.ext_motmov='43') and
a.ext_fechop is not null
if fechsus is not null then
update suspension
set sus_fecsus=fechsus
where sus_rfc=strffc
and sus_deter=intdeter
--graba a bitacora que si se encontro
insert into
Bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
values(75,2,0,strffc,intdeter,CURRENT,x_alr)
else
insert into
Bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
values(75,2,0,strffc,intdeter,CURRENT,x_alr)
update suspension set sus_fecsus='99/01/01'
where sus_rfc=strffc and sus_deter=intdeter
end if
end if
end foreach
end function
```

```
function caso_56()
define rec_baj record
rfc char(13),
deter smallint,
fech date
end record
define x_que char(500)
let x_que="insert into
Bita_6(num_caso,variante,subvar,baj_rfc,baj_deter,",
" fecha_hora,baj_fec,alr)",
" select 56,1,0,a.baj_rfc,baj_deter,CURRENT,b.baj_fec,",x_alr,
" from bajas a, bajnal b",
" where a.baj_rfc=b.baj_rfc and a.baj_deter=b.baj_deter and ",
" a.baj_fec IS NULL and b.baj_fec is Not Null"
prepare baj_sfech from x_que
execute baj_sfech
```

```
declare crs_baj cursor for
select baj_rfc,baj_deter,baj_fec
into rec_baj.*
from bita_6
where num_caso=56 and variante=1 and subvar=0
```

```
foreach crs_baj
update bajas set baj_fec=rec_baj.fech
```

```
where baj_rfc=rec_baj.rfc and baj_deter=rec_baj.deter
end foreach
call bajloc()
```

```
end function
function bajloc()
define strffc char(13),fechbaj date
define intdeter smallint,b_cve char(2)
```

```
declare crs_baj1 cursor for
select baj_rfc,baj_deter,baj_cvebaj
into strffc,intdeter,b_cve
from bajas where baj_fec is null
```

```
foreach crs_baj1
initialize fechbaj to null
select max(doc_fechop) into fechbaj
from documentos a
where a.doc_rfc=strffc and a.doc_deter=intdeter and
a.doc_motmov =b_cve and a.doc_fechop is not null
if fechbaj is not null then
```

```
--actualiza la fecha de inicio de bajas
update bajas set baj_fec=fechbaj
where baj_rfc=strffc and baj_deter=intdeter and
baj_cvebaj=b_cve
```

```
--graba a bitacora que si se encontro
insert into
Bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
values(56,2,0,strffc,intdeter,CURRENT,x_alr)
```

```
else
select max(ext_fechop) into fechbaj
from externos a
where a.ext_rfc=strffc and a.ext_deter=intdeter and
a.ext_motmov=b_cve and a.ext_fechop is not null
if fechbaj is not null then
```

```
update bajas set baj_fecbaj=fechbaj
where baj_rfc=strffc and baj_deter=intdeter and
baj_cvebaj=b_cve
```

```
--graba a bitacora que si se encontro
insert into
Bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
values(56,2,0,strffc,intdeter,CURRENT,x_alr)
else
insert into
```

```
Bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
values(56,2,0,strffc,intdeter,CURRENT,x_alr)
update bajas set baj_fecbaj='99/01/01'
where baj_rfc=strffc and baj_deter=intdeter
and baj_cvebaj=b_cve
end if
end if
end foreach
```

```
end function
```

```
function caso_90()
define x_inst char(500)
define rw_b13 record
num_caso smallint ,
variante smallint,
subvar smallint,
fecha_hora datetime YEAR TO SECOND ,
alr smallint ,
ced_rfc char(13),
ced_le1 char(1),
ced_num integer,
```

```

        ced_fecalt    date,
        ced_fecbaj   date,
        ced_mot      char(1),
        ced_ctl      char(1)
    end record
----- INSERTA CASO 90 CEDULAS -----
--
let x_inst = "insert into bita_13 ",
"select
90,1,0,current,"x_alr","lm.ced_rfc,lm.ced_let,lm.ced_num,lm.ced
_fecalt,""",
" from cednal cn, cedulas lm where cn.ced_rfc = lm.ced_rfc ",
" and cn.ced_let = lm.ced_let and cn.ced_num = lm.ced_num
and ",
" lm.ced_fecalt IS NULL ",
" and cn.ced_fecalt is not null "
prepare x_eje2 from x_inst
execute x_eje2
declare c90s cursor for
select * from bita_13
where num_caso = 90 and variante=1
foreach c90s into rw_b13.*
update cedulas set ced_fecalt = rw_b13.ced_fecalt
where ced_rfc=rw_b13.ced_rfc and
ced_let=rw_b13.ced_let and
ced_num=rw_b13.ced_num
end foreach
close c90s
-- inserto en bita 13 las cedulas sin fecha de alta
let x_inst="insert into bita_13 select 90,2,0,current,"x_alr,
" from cedulas where ced_fecalt is NULL "
prepare x_90 from x_inst
execute x_90
update cedulas set ced_fecalt = '99/01/01'
where ced_fecalt is NULL
end function # fin de la funcion

function caso_70()
define rec_suc record
    rfc char(13),
    fechi date
end record
define x_que char(350)
let x_que="insert into
Bita_7(num_caso,variante,subvar,suc_rfc,fecha_hora,suc_fechi,a
lr)",
" select 70,1,0,a.suc_rfc,CURRENT,b.suc_fechi,"x_alr,
" from sucesion a,sucnal b",
" where a.suc_rfc=b.suc_rfc and a.suc_fechi IS NULL and
b.suc_fechi is Not Null"
prepare suc_sfech1 from x_que
execute suc_sfech1

declare crs_sucesion cursor for
select suc_rfc,suc_fechi
into rec_suc.*
from bita_7
where num_caso=70 and variante=1 and subvar=0

foreach crs_sucesion
update sucesion
set suc_fechi=rec_suc.fechi
where suc_rfc=rec_suc.rfc

end foreach
call sucesionloc()

end function

```

```

function sucesionloc()
define strffc char(13),fechsuc date

declare crs_sucesion1 cursor for
select suc_rfc
into strffc
from sucesion
where suc_fechi is null

foreach crs_sucesion1
initialize fechsuc to null

-- a qui cual es fecha de liquidacion y cual es la clave de la
operacion

select max(doc_fechop) into fechsuc
from documentos a
where a.doc_rfc=strffc and a.doc_deter=0 and
(a.doc_motmov='18' or a.doc_motmov='42') and a.doc_fechop is
not null
if fechsuc is not null then
--variante 2.2
--actualiza la fecha de inicio de liquidacion
update sucesion
set suc_fechi=fechsuc
where suc_rfc=strffc
--graba a bitacora que si se encontro
insert into
Bita_3(num_caso,variante,subvar,rfc,fecha_hora,alr)
values(70,2,0,strffc,CURRENT,x_alr)
else
select max(ext_fechop) into fechsuc
from externos a
where a.ext_rfc=strffc and a.ext_deter=0 and
(a.ext_motmov='18' or a.ext_motmov='42') and
a.ext_fechop is not null
if fechsuc is not null then

update sucesion
set suc_fechi=fechsuc
where suc_rfc=strffc
--graba a bitacora que si se encontro
insert into
Bita_3(num_caso,variante,subvar,rfc,fecha_hora,alr)
values(70,2,0,strffc,CURRENT,x_alr)
else
insert into
Bita_3(num_caso,variante,subvar,rfc,fecha_hora,alr)
values(70,3,0,strffc,CURRENT,x_alr)
update sucesion
set suc_fechi='99/01/01'
where suc_rfc=strffc
end if

end if

end foreach

end function

function caso_64()
define rec_liq record
    rfc char(13),
    fechi date
end record

```

```

define x_que char(350)
let x_que="insert into
Bit_9(num_caso,variante,subvar,liq_rfc,fecha_hora,liq_fechi,alr)
"
* select 64,1,0,a,liq_rfc,CURRENT,b,liq_fechi,"x_alr,
* from liquidacion a,liqnal b",
* where a.liq_rfc=b.liq_rfc and a.liq_fechi is null and b.liq_fechi is
Not Null"
prepare liq_sfech from x_que
execute liq_sfech

declare crs_liquida cursor for
select liq_rfc,liq_fechi
into rec_liq.*
from bita_9
where num_caso=64 and variante=1 and subvar=0

foreach crs_liquida
update liquidacion
set liq_fechi=rec_liq.fechi
where liq_rfc=rec_liq.rfc

end foreach

call liquidaloc()

end function
function liquidaloc()
define strfc char(13),fechliq date

declare crs_liquida1 cursor for
select liq_rfc
into strfc
from liquidacion
where liq_fechi is null

foreach crs_liquida1
initialize fechliq to null

-- a qui cual es fecha de liquidacion y cual es la clave de la
operacion

select max(doc_fechop) into fechliq
from documentos a
where a.doc_rfc=strfc and a.doc_deter=0 and
(a.doc_motmov='20' or a.doc_motmov='41')
and a.doc_fechop is not null
if fechliq is not null then
--actualiza la fecha de inicio de liquidacion
update liquidacion
set liq_fechi=fechliq
where liq_rfc=strfc
--graba a bitacora que si se encontro
insert into
Bit_3(num_caso,variante,subvar,rfc,fecha_hora,alr)
values(64,2,0,strfc,CURRENT,x_alr)
else
select max(ext_fechop) into fechliq
from externos a
where a.ext_rfc=strfc and a.ext_deter=0 and
(a.ext_motmov='20' or a.ext_motmov='41') and
a.ext_fechop is not null
if fechliq is not null then
update liquidacion
set liq_fechi=fechliq
where liq_rfc=strfc
--graba a bitacora que si se encontro
insert into
Bit_3(num_caso,variante,subvar,rfc,fecha_hora,alr)
values(64,2,0,strfc,CURRENT,x_alr)
else

```

```

insert into
Bit_3(num_caso,variante,subvar,rfc,fecha_hora,alr)
values(64,3,0,strfc,CURRENT,x_alr)
update liquidacion set liq_fechi='99/01/01'
where liq_rfc=strfc
end if

end function

function caso_54_55()
define cont integer,temp char(2)
define cau0 record like bajas.*, id float
define motivorar record
num_caso smallint,
variante smallint,
subvar smallint,
rfc char(13),
deter smallint,
fecha_hora datetime YEAR TO SECOND,
alr smallint,
rfc_nuevo char(13),
nombre varchar(80)
end record
define motivoraro record
num_caso smallint,
variante smallint,
subvar smallint,
rfc char(13),
deter smallint,
fecha_hora datetime YEAR TO SECOND,
alr smallint,
cau_motall char(2),
cau_calcont smallint,
cau_tipsoc smallint,
cau_tipcont char(1),
cau_secgob integer,
cau_cveact char(6),
cau_movs char(1),
baj_cvebaj char(2),
calcont_nal smallint,
tipcont_nal char(1)
end record
declare c54 cursor for
select *,rowid
into cau0.*,id
from bajas where baj_rfc[1,4]<>'YYYY'

initialize motivorul.* to NULL
initialize motivoraro.* to NULL
foreach c54
if cau0.baj_cvebaj is null then
Select doc_motmov into temp from documentos d,
cat_mov c
where d.doc_motmov = c.cve_mov and
doc_rfc = cau0.baj_rfc and
doc_deter = cau0.baj_deter and
doc_fechop = cau0.baj_fec
if Status <> notfound then
let motivorul.rfc=cau0.baj_rfc
let motivorul.deter=cau0.baj_deter
let motivorul.fecha_hora=current
let motivorul.alr=x_alr
let motivorul.num_caso=54
let motivorul.variante=0
let motivorul.subvar=0

```

```

insert into bita_3 values (motivonul.*)
update bajas set baj_cvebaj = temp where
rowid=id
    let cau0.baj_cvebaj = temp
Else
    Select ext_motmov into temp from externos
e, cat_mov c
    where e.ext_motmov = c.cve_mov and
ext_rfc = cau0.baj_rfc and
    ext_deter = cau0.baj_deter and ext_fechop =
cau0.baj_fec
If Status <> notfound then
    let motivonul.rfc=cau0.baj_rfc
    let motivonul.deter=cau0.baj_deter
    let motivonul.fecha_hora=current
    let motivonul.alr=x_alr
    let motivonul.num_caso=54
    let motivonul.variante=0
    let motivonul.subvar=0
insert into bita_3 values (motivonul.*)
update bajas set baj_cvebaj = temp where
rowid=id
    let cau0.baj_cvebaj = temp
Else
    let motivonul.rfc=cau0.baj_rfc
    let motivonul.deter=cau0.baj_deter
    let motivonul.fecha_hora=current
    let motivonul.alr=x_alr
    let motivonul.num_caso=54
    let motivonul.variante=1
    let motivonul.subvar=0
insert into bita_3 values (motivonul.*)
if cau0.baj_rfc[1,1]=' ' then
    let temp='01'
else
    let temp='04'
end if
update bajas set baj_cvebaj = temp where
rowid=id
    let cau0.baj_cvebaj = temp
End If
End If
Else
    Select cve_mov from cat_mov where cve_mov =
cau0.baj_cvebaj
If Status = notfound then
    Select doc_motmov into temp from documentos d,
cat_mov c
    where d.doc_motmov = c.cve_mov and doc_rfc =
cau0.baj_rfc and
    doc_deter = cau0.baj_deter and doc_fechop =
cau0.baj_fec
If Status <> notfound then
    let motivonul.rfc=cau0.baj_rfc
    let motivonul.deter=cau0.baj_deter
    let motivonul.fecha_hora=current
    let motivonul.alr=x_alr
    let motivonul.num_caso=55
    let motivonul.variante=0
    let motivonul.subvar=0
insert into bita_3 values (motivonul.*)
update bajas set baj_cvebaj = temp where
rowid=id
    let cau0.baj_cvebaj = temp
Else
    Select ext_motmov into temp from externos
e, cat_mov c
    where e.ext_motmov=c.cve_mov and
ext_rfc=cau0.baj_rfc and

```

```

    ext_deter = cau0.baj_deter and ext_fechop
= cau0.baj_fec
If Status <> notfound then
    let motivonul.rfc=cau0.baj_rfc
    let motivonul.deter=cau0.baj_deter
    let motivonul.fecha_hora=current
    let motivonul.alr=x_alr
    let motivonul.num_caso=55
    let motivonul.variante=0
    let motivonul.subvar=0
insert into bita_3 values (motivonul.*)
update bajas set baj_cvebaj = temp where
rowid=id
    let cau0.baj_cvebaj = temp
Else
    let motivoraro.rfc=cau0.baj_rfc
    let motivoraro.deter=cau0.baj_deter
    let motivoraro.fecha_hora=current
    let motivoraro.alr=x_alr
    let motivoraro.num_caso=55
    let motivoraro.variante=1
    let motivoraro.subvar=0
    let
    motivoraro.baj_cvebaj=cau0.baj_cvebaj
(motivoraro.*)
    insert into bita_11 values
    if cau0.baj_rfc[1,1]=' ' then
        let temp='01'
    else
        let temp='04'
    end if
    update bajas set baj_cvebaj =
temp where rowid=id
    let cau0.baj_cvebaj = temp
End If
End If
End If
end foreach
end function

function Caso_38()
    define x_caso char(550)
    let x_caso= insert into bita_11
(num_caso,variante,subvar,rfc,deter,',
' fecha_hora,alr,cau_tipcont,tipcont_nal) select
38,0,0,lc.cau_rfc,',
' lc.cau_deter,current,',x_alr,',lc.cau_tipcont,c.cau_tipcont',
' from causante lc, hist_det c',
' where lc.cau_rfc = c.cau_rfc and lc.cau_deter = c.cau_deter
and',
' lc.cau_deter = 0 and lc.tipcont <> c.cau_tipcont and
c.cau_tipcont',
' is not null'

    update causante set cau_tipcont =
(select b.tipcont_nal from bita_11 b
where b.rfc = causante.cau_rfc and b.deter =
causante.cau_deter)
    where cau_rfc in (select rfc from bita_11
where num_caso = 38 and variante = 0 and
(tipcont_nal = 1 or tipcont_nal = 2))

    select num_caso,1
a.subvar,rfc,deter,fecha_hora,alr,cau_tipcont
from bita_11 where num_caso = 38 and variante = 0 and
(tipcont_nal = 1 or tipcont_nal = 2) into temp hola
with no log

```



```

insert into bita_11
(num_caso,variante,subvar,rfc,deter,fecha_hora,
alr,cau_tipcont)
select * from hola

drop table hola

End Function

function caso_81_REG()
define cau0 record
num_caso smallint,
variante smallint,
subvar smallint,
alr smallint,
rfc char(13),
deter smallint,
ofn char(5),
afn smallint
end record
define cau1 record like representa.*
define nombre char(80),domicilio char(80),telefono char(10)
define reprec record
num_caso          smallint ,
variante          smallint,
subvar            smallint,
fecha_hora        datetime YEAR TO SECOND ,
alr                smallint ,
rep_rfcca        char(13),
rep_detcau        smallint,
rep_rfcrcp        char(13),
rep_nomrep        varchar(80),
rep_domrep        varchar(80),
rep_telef         varchar(10),
rfc_rep           char(13),
alr_rep           smallint
end record
declare c81 cursor for
select comp_casos.*,representa.*
into cau0.*,cau1.*
from comp_casos,representa
where rep_rfcrcp = rfc and num_caso = 81 and rep_detcau
= 0
foreach c81
initialize reprec.* to NULL

Select distinct rsoc,domi,telefo into
nombre,domicilio,telefono
from res_casos where num_caso = 81 and rfc = cau0.rfc
If Status <> notfound then
let reprec.num_caso=81
let reprec.variante=3
let reprec.subvar=1
let reprec.fecha_hora=current
let reprec.alr=x_alr
let reprec.rep_rfcca=cau1.rep_rfcca
let reprec.rep_detcau=cau1.rep_detcau
let reprec.rep_nomrep=cau1.rep_nomrep
let reprec.rep_rfcrcp=cau1.rep_rfcrcp
let reprec.rep_domrep=cau1.rep_domrep
let reprec.rep_telef=cau1.rep_telef
insert into bita_12 values (reprec.*)
update representa set rep_nomrep = nombre,
rep_domrep = domicilio,
rep_telef = telefono
where rep_rfcrcp=cau0.rfc
Eise

```

```

If 13 >= length(cau1.rep_nomrep) or cau1.rep_nomrep is
null then
let reprec.num_caso=81
let reprec.variante=4
let reprec.subvar=0
let reprec.fecha_hora=current
let reprec.alr=x_alr
let reprec.rep_rfcca=cau1.rep_rfcca
let reprec.rep_detcau=cau1.rep_detcau
let reprec.rep_nomrep=cau1.rep_nomrep
let reprec.rep_rfcrcp=cau1.rep_rfcrcp
let reprec.rep_domrep=cau1.rep_domrep
let reprec.rep_telef=cau1.rep_telef
insert into bita_12 values (reprec.*)
delete from representa
where rep_rfcrcp=cau0.rfc
End if
End if
end foreach
end function

```

```

function Caso_71_REG()
define cau0 record
num_caso smallint,
variante smallint,
subvar smallint,
alr smallint,
rfc char(13),
deter smallint,
ofn char(5),
afn smallint
end record
define cau1 record like sucesion.*
define nombre char(80),domicilio char(80)
define sucrec record
num_caso          smallint ,
variante          smallint,
subvar            smallint,
fecha_hora        datetime YEAR TO SECOND ,
alr                smallint ,
suc_rfc           char(13),
suc_fechi         date,
suc_nomalb        varchar(80),
suc_rfcalb        char(13),
suc_domalb        varchar(80),
rfc_alb           char(13),
alr_alb           smallint
end record
declare c71 cursor for
select comp_casos.*,sucesion.*
into cau0.*,cau1.*
from comp_casos,sucesion
where suc_rfcalb = rfc and num_caso = 71
foreach c71
initialize sucrec.* to NULL

Select distinct rsoc,domi into nombre,domicilio
from res_casos where num_caso = 71 and rfc = cau0.rfc
If Status <> notfound then
let sucrec.num_caso=71
let sucrec.variante=3
let sucrec.subvar=1
let sucrec.fecha_hora=current
let sucrec.alr=x_alr
let sucrec.suc_rfc=cau1.suc_rfc
let sucrec.suc_nomalb=cau1.suc_nomalb
let sucrec.suc_rfcalb=cau1.suc_rfcalb
let sucrec.suc_domalb=cau1.suc_domalb
insert into bita_7 values (sucrec.*)

```

```

update sucesion set suc_nomalb = nombre,
suc_domalb = domicilio
where suc_rfcalb=cau0.rfc
Else
If 13 >= length(cau1.suc_nomalb) or
cau1.suc_nomalb is null then
let sucrec.num_caso=71
let sucrec.variante=4
let sucrec.subvar=0
let sucrec.fecha_hora=current
let sucrec.alr=x_alr
let sucrec.suc_rfc=cau1.suc_rfc
let sucrec.suc_nomalb=cau1.suc_nomalb
let sucrec.suc_rfcalb=cau1.suc_rfcalb
let sucrec.suc_domalb=cau1.suc_domalb
insert into bita_7 values (sucrec.*)
update sucesion set suc_nomalb =
",suc_rfcalb = ",
suc_domalb = "
where suc_rfcalb=cau0.rfc
End If
End If
end foreach
end function

```

```

function caso_65_REG()
define cau0 record
num_caso smallint,
variante smallint,
subvar smallint,
alr smallint,
rfc char(13),
deter smallint,
ofh char(5),
afr smallint
end record
define cau1 record like liiquidacion.*
define nombre char(80),domicilio char(80),telefono char (10)
define liqrec record
num_caso smallint ,
variante smallint,
subvar smallint,
fecha_hora datetime YEAR TO SECOND ,
alr smallint ,
status char(1),
rfc char(13),
liq_rfc char(13),
liq_fechi date,
liq_nomliq varchar(80),
liq_rfcliq char(13),
liq_domliq varchar(80),
liq_telliq varchar(10),
alr_liq smallint
end record
declare c65 cursor for
select comp_casos.*,liiquidacion.*
into cau0.*,cau1.*
from comp_casos,liquidacion
where liq_rfcliq = rfc and num_caso = 65
foreach c65
initialize liqrec.* to NULL

Select distinct rsoc,domi,telefo into
nombre,domicilio,telefono
from res_casos where num_caso = 65 and rfc = cau0.rfc
If Status <> notfound then
let liqrec.num_caso=65
let liqrec.variante=3
let liqrec.subvar=1

```

```

let liqrec.fecha_hora=current
let liqrec.alr=x_alr
let liqrec.liq_rfc=cau1.liq_rfc
let liqrec.liq_nomliq=cau1.liq_nomliq
let liqrec.liq_rfcliq=cau1.liq_rfcliq
let liqrec.liq_domliq=cau1.liq_domliq
let liqrec.liq_telliq=cau1.liq_telliq
insert into bita_9 values (liqrec.*)
update liiquidacion set liq_nomliq = nombre,
liq_domliq = domicilio,
liq_telliq = telefono
where liq_rfcliq=cau0.rfc
Else
If 13 >= length(cau1.liq_nomliq) or cau1.liq_nomliq is
null then
let liqrec.num_caso=65
let liqrec.variante=4
let liqrec.subvar=0
let liqrec.fecha_hora=current
let liqrec.alr=x_alr
let liqrec.liq_rfc=cau1.liq_rfc
let liqrec.liq_nomliq=cau1.liq_nomliq
let liqrec.liq_rfcliq=cau1.liq_rfcliq
let liqrec.liq_domliq=cau1.liq_domliq
let liqrec.liq_telliq=cau1.liq_telliq
insert into bita_9 values (liqrec.*)
update liiquidacion set liq_nomliq = ",liq_rfcliq
= ",
liq_domliq = ",liq_telliq
= "
where liq_rfcliq=cau0.rfc
End If
End If
end foreach
end function

```

```

function caso_15(n_alr)
define n_alr smallint, x_caso char(300)
define cau10 record
variante,
subvar,
rfc char(13),
deter smallint
end record

```

```

declare c1 cursor for
select variante,subvar,rfc,deter
into cau10.*
from res_casos
where num_caso=15

```

```

foreach c1
call borrsseg(cau10.rfc,15,4)
end foreach

```

```

close c1
end function # fin de la funcion

```

```

function caso_138()
define fecha date, cau2aux,cau2aux1 char(13),y_inst char(300)
define cau138 record
variante smallint,
subvar smallint,
rfc char(13),
deter smallint
end record
define nom_rfcaux char(13)

```

```

declare c138 cursor for
select variante,subvar,rfc,deter
into cau138.*
from res_casos
where num_caso=138

foreach c138
  if cau138.variante=1 then
    # si lo encuentre en la nacional

    insert into bita_3
    values(138,1,0,cau138.rfc,0,current,x_alr,")
    call borseg(cau138.rfc,138,3)
    else
      # no lo encuentre en la nacional o es mio
    if cau138.variante=2 then
      select nom_rfca into nom_rfcaux
      from nombres
      where nom_rfc=cau138.rfc

      insert into bita_3
      values(138,2,0,cau138.rfc,0,current,x_alr,")

      let xx_rfc=nom_rfcaux
      let xx_deter=0
      let xx_caso=138
      let xx_var=3

      let y_inst=insert into bita_8 select
      ',xx_caso,',xx_var,',3,current,'x_alr,
      ',* from nombres where nom_rfca=',xx_rfc clipped, "'
      prepare y_nom2 from y_inst
      execute y_nom2

      delete from nombres
      where nom_rfca=xx_rfc

      select baj_fec into fecha from bajas
      where baj_rfc=nom_rfcaux and baj_deter=0
      and (baj_cvebaj='14' or baj_cvebaj='33')
      if status<>NOTFOUND then
        # si existe
        let y_inst=insert into bita_6 select
        ',xx_caso,',xx_var,
        ',1,current,'x_alr,',' from bajas where baj_rfc=',
        xx_rfc clipped, "' and baj_deter=',xx_deter,
        ' and (baj_cvebaj='14" or baj_cvebaj='33")'
        prepare y_baj138 from y_inst
        execute y_baj138
        ---- Elimina la baja
        delete from bajas
        where baj_rfc=xx_rfc and
        baj_deter=xx_deter
        and (baj_cvebaj='14" or baj_cvebaj='33")

        -- se activa al contribuyente
        let xx_var=2

        let y_inst=insert into bita_1 select
        ',xx_caso,',xx_var,
        ',1,current,'x_alr,',' from causante where cau_rfc=',
        xx_rfc clipped, "' and cau_deter=',xx_deter
        prepare y_cau138 from y_inst
        execute y_cau138

        update causante
        set cau_movs='
        where cau_rfc=xx_rfc
        and cau_deter=0

```

```

-- reactiva obligaciones iguales a la fecha
de camb de nom.

```

```

let y_inst=insert into bita_16 select
',xx_caso,',xx_var,
',2,current,'x_alr,',' from obligacion where obl_rfc=',
xx_rfc clipped, "' and obl_deter=',xx_deter,
' and obl_fecalt=',fecha
prepare y_obl138 from y_inst
execute y_obl138

update obligacion
set obl_fecbaj=NULL
where obl_rfc=xx_rfc and obl_deter=0
and obl_fecalt=fecha

---- Borrar al posterior de las 10 tablas restantes
call borseg(cau138.rfc,138,3)

end if
end if
end foreach
close c138

end function

```

```

function borseg(causante, caso, vari)
define causante char(13), caso, vari smallint

```

```

let xx_rfc=causante
let xx_deter=0
let xx_caso=caso
let xx_var=vari
call bor_bajas()
call bor_suce()
call bor_nombre()
call bor_liquid()
call bor_suspen()
call bor_obligacion()
call bor_externos()
call bor_representa()
call bor_documentos()
call bor_cedulas()
call bor_causante()

```

```
end function
```

```
function bor_obligacion()
```

```

define y_inst char(300)
let y_inst=insert into bita_16 select
',xx_caso,',xx_var,',11,current,'
x_alr,',' from obligacion where obl_rfc=',xx_rfc clipped,
"' and obl_deter=',xx_deter
prepare y_sus5 from y_inst
execute y_sus5
-- elimina
delete from obligacion
where obl_rfc=xx_rfc
and obl_deter=xx_deter
end function #

```

```
function bor_externos()
```

```

define y_inst char(300)
let y_inst=insert into bita_15 select
',xx_caso,',xx_var,',10,current,'
x_alr,',' from externos where ext_rfc=',xx_rfc clipped,
"' and ext_deter=',xx_deter
prepare y_sus4 from y_inst

```

```

execute y_sus4
  -- elimina
delete from externos
  where ext_rfc=xx_rfc
  and ext_deter=xx_deter
end function #

function bor_representa()
  define y_inst char(300)
  let y_inst="insert into bita_12 select
',xx_caso,';',xx_var;',9,current,'
  x_alr',';','' from representa where rep_rfccau='',xx_rfc
  clipped,
  "" and rep_detcau=',xx_deter
  prepare y_sus3 from y_inst
  execute y_sus3
  -- elimina
  delete from representa
    where rep_rfccau=xx_rfc
    and rep_detcau=xx_deter
end function #

function bor_documentos()
  define y_inst char(300)
  let y_inst="insert into bita_14 select
',xx_caso,';',xx_var;',8,current,'
  x_alr',';','' from documentos where doc_rfc='',xx_rfc
  clipped,
  "" and doc_deter=',xx_deter
  prepare y_sus2 from y_inst
  execute y_sus2
  -- elimina
  delete from documentos
    where doc_rfc=xx_rfc
    and doc_deter=xx_deter
end function #

function bor_cedulas()
  define y_inst char(300)
  let y_inst="insert into bita_13 select
',xx_caso,';',xx_var;',7,current,'
  x_alr',';','' from cedulas where ced_rfc='',xx_rfc clipped, ""
  prepare y_sus1 from y_inst
  execute y_sus1
  -- elimina
  delete from cedulas
    where ced_rfc=xx_rfc
end function #

function bor_causante()
  define y_inst char(300)
  let y_inst="insert into bita_1 select
',xx_caso,';',xx_var;',6,current,'
  x_alr',';','' from causante where cau_rfc='',xx_rfc clipped,
  "" and cau_deter=',xx_deter
  prepare y_sus from y_inst
  execute y_sus
  -- elimina
  delete from causante
    where cau_rfc=xx_rfc
    and cau_deter=xx_deter
end function

function bor_bajas()
  define y_inst char(300)
  let y_inst="insert into bita_6 select
',xx_caso,';',xx_var,'
',1,current,'x_alr',';','' from bajas where baj_rfc='',
  xx_rfc clipped, "" and baj_deter=',xx_deter
  prepare y_baj from y_inst
  execute y_baj

```

```

  -- Elimina la baja
  delete from bajas
    where baj_rfc=xx_rfc and
    baj_deter=xx_deter
end function

function bor_liquid()
  define y_inst char(300)
  let y_inst="insert into bita_9 select
',xx_caso,';',xx_var;',4,current,'x_alr,
',,,liq_rfc,','' from liquidacion where liq_rfc='',
  xx_rfc clipped, ""
  prepare y_liq from y_inst
  execute y_liq
  -- elimina la liquidacion
  delete from liquidacion
    where liq_rfc=xx_rfc
end function

function bor_suce()
  define y_inst char(300)
  let y_inst="insert into bita_7 select
',xx_caso,';',xx_var;',2,current,'x_alr,
',,,suc_rfc,','' from sucesion where suc_rfc='',
  xx_rfc clipped, ""
  prepare y_suc from y_inst
  execute y_suc
  -- elimina la sucesion
  delete from sucesion
    where suc_rfc=xx_rfc
end function

function bor_suspen()
  define y_inst char(300)
  let y_inst="insert into bita_10 select
',xx_caso,';',xx_var;',5,current,'
  x_alr',';','' from suspension where sus_rfc='',xx_rfc clipped,
  "" and sus_deter=',xx_deter
  prepare y_sus9 from y_inst
  execute y_sus9
  -- elimina la supencion
  delete from suspension
    where sus_rfc=xx_rfc
    and sus_deter=xx_deter
end function

function bor_nombre()
  define y_inst char(300)
  let y_inst="insert into bita_8 select
',xx_caso,';',xx_var;',3,current,'x_alr,
',,,nom_rfc,','' from nombres where nom_rfcp='',xx_rfc
  clipped, ""
  prepare y_nom from y_inst
  execute y_nom
  -- elimina la supencion
  delete from nombres
    where nom_rfcp=xx_rfc
end function

NALVSLOC.4GL

Database shcp
globals
  define vpm integer
  define x_alr char(2),
  alr_1 smallint,disp136 char(20)
  define stat_cau integer,s_char,x_bajmov char(2)
  define f_baj,f_liq,f_suce,f_sus,f_nom,f_mayor date,x_numnon
  char(1),

```

```

        x_vari char(10),x_caso char(2)
define cau0n record like causante.*
define cau10 record
    rfc char(13),
    deter smallint,
    domi char(80),
    alr smallint,
    arr smallint,
    crh char(5)
end record
define hora char(30), xx_rfc char(13), xx_deter smallint,
    xx_caso smallint, xx_var smallint
define nom1 record
    num_caso smallint,
    variante smallint,
    subvar smallint,
    fecha_hora datetime YEAR TO SECOND,
    alr smallint,
    nom_rfca char(13),
    nom_rfc char(13),
    nom_fecha date
end record
define caurec record like causante.*
define g_alr smallint,g_ofh char(5)
end globals
main
    define x_instj,c_comm char(300)
    let x_alr=arg_val(1)
    update casos set inicio = current where caso = 205
    defer quit
    defer interrupt
    if length(x_alr)=0 or x_alr is null then
        display " error, no se proporciono la atr"
        exit program(1)
    end if

    declare c_ofh scroll cursor for
    select cve_ofh,afr
    from cat_ofh where edo=x_alr
    open c_ofh
        fetch first c_ofh into g_ofh,g_afr
    close c_ofh
    let g_ofh=g_ofh clipped,'00'

    let hora=current
    display 'inicio a las ',hora,' alr = ',x_alr
    select * from sysindexes
    where idxname='j_mescie_i'
    if status=notfound then
        let hora=current
        display ' Creando indice por alr ',hora
        create index j_mescie_i on hist_det(cau_mescie)
        let hora=current
        display ' Indice creado ',hora
    end if

    let hora=current
    display ' Inicia comparacion de lo que la local no tiene
',hora

    call Nainoloc(x_alr)
    let hora=current
    display ' Termina comparacion de lo que la local no
tiene ',hora

    update casos set inicio = current where caso = 136
    let hora=current
    display 'caso 136 ',hora
    call caso_136()
    select count(*) into vpm from comp_casos1
    where num_caso = 136
    if vpm > 0 then

```

```

        let disp136='dud136 ',x_alr clipped
    run disp136
    let disp136='sh cargres136.sh ',x_alr clipped
    run disp136
end if
update casos set fin = current where caso = 136

    let hora=current
    display ' Inicia identificacion de nulos ',hora
-----
-- ejecuta programa de nulos --
-----
let c_comm=" nulos ",x_alr
run c_comm
let hora=current
display ' Termina identificacion de nulos ',hora

let hora=current
update casos set inicio = current where caso = 137
display 'caso 137 ',hora
call caso_137()
update casos set fin = current where caso = 137

let hora=current
update casos set inicio = current where caso = 111
display 'caso 111 ',hora
call extdup()
update casos set fin = current where caso = 111

update casos set inicio = current where caso = 10
let hora=current
display 'caso 10 ',hora
call caso_10()
update casos set fin = current where caso = 10
update casos set inicio = current where caso = 52
let hora=current
display 'caso 52 ',hora
call caso_52()
update casos set fin = current where caso = 52
update casos set inicio = current where caso = 59
let hora=current
display 'caso 59 ',hora
call caso_59()
update casos set fin = current where caso = 59
update casos set inicio = current where caso = 68
let hora=current
display 'caso 68 ',hora
call caso_68()
update casos set fin = current where caso = 68
update casos set inicio = current where caso = 74
let hora=current
display 'caso 74 ',hora
call caso_74()
update casos set fin = current where caso = 74

update casos set inicio = current where caso = 80
let hora=current
display 'caso 80 ',hora
call caso_80()
update casos set fin = current where caso = 80
update casos set inicio = current where caso = 84
let hora=current
display 'caso 84 ',hora
call caso_84()
update casos set fin = current where caso = 84
update casos set inicio = current where caso = 100
let hora=current
display 'caso 100 ',hora
call caso_100()
update casos set fin = current where caso = 100
update casos set inicio = current where caso = 101

```

```

let hora=current
display 'caso 101 ',hora
call caso_101()
update casos set fin = current where caso = 101

update casos set inicio = current where caso = 138
let hora=current
display 'caso 138 ',hora
call caso_138()
update casos set fin = current where caso = 138

update casos set inicio = current where caso = 15
let hora=current
display 'caso 15 ',hora
call caso_15(x_alr)
update casos set fin = current where caso = 15

update casos set inicio = current where caso = 124
let hora=current
display 'caso 124 ',hora
update casos set inicio = current where caso = 124
call caso_124()
update casos set fin = current where caso = 124
let hora=current
display 'Termino todo a las ',hora

display 'alr ',x_alr
end main

function Nalnoloc(n_alr)
define n_alr char(2)
define x_que char(580)

select a.cau_rfc,a.cau_deter from causante a, hist_det b
where a.cau_rfc=b.cau_rfc and
a.cau_deter=b.cau_deter and a.cau_deter=0
into TEMP tmpcau1 with no log

-- caso 49
-- Query que selecciona los causantes que están en la nacional
que no existe en
-- la Local
initialize x_que to null
update casos set inicio = current where caso = 49
let x_que="insert into bita_1",
" select 49,0,0,CURRENT,"x_alr clipped","a.",
" from hist_det a",
" where not exists(select * from causante b ",
"where b.cau_rfc=a.cau_rfc and b.cau_deter=a.cau_deter)"
prepare x_caso49 from x_que
let hora=current
display ' Inicia comparacion de causantes que la local
no tiene ',hora
execute x_caso49
insert into causante
select
cau_rfc,cau_deter,cau_rsoc,cau_domi,cau_codpos,cau_zonpos,
cau_mpio,cau_afr,cau_entfed,cau_ofh,cau_fecalt,cau_
motall,
cau_calcont,cau_tipsoc,cau_tipcont,cau_secgob,cau_c
veact,
cau_mescie,cau_zondil,cau_serpos,cau_telefo,cau_est
able,cau_movs
from Bita_1
where num_caso=49
update casos set fin = current where caso = 49
let hora=current

```

```

display ' Termina comparacion de causantes que la
local no tiene ',hora
display 'Listo causante...'

-- caso 57
-- Query que selecciona los bajas que están en la nacional que
no existe en
-- la Local
initialize x_que to null
update casos set inicio = current where caso = 57
let x_que="insert into bita_6",
" select 57,0,0,CURRENT,"x_alr clipped","a.",
" from bajnal a",
" where not exists(select * from bajas c ",
"where c.baj_rfc=a.baj_rfc and
c.baj_deter=a.baj_deter)"
prepare x_caso57 from x_que
let hora=current
display ' Inicia comparacion de bajas que la local no
tiene ',hora
execute x_caso57
insert into bajas
select baj_rfc,baj_deter,baj_fec,baj_cvebaj
from Bita_6
where num_caso=57
let hora=current
update casos set fin = current where caso = 57
display ' Termina comparacion de bajas que la local no
tiene ',hora
display 'Listo bajas...'

-- caso 131
-- Query que selecciona las Obligaciones que están en la
nacional que no existe en
-- la Local
initialize x_que to null
update casos set inicio = current where caso = 131
let x_que="insert into bita_16",
" select 131,0,0,CURRENT,"x_alr clipped","a.",
" from obnal a where not exists(select * from
obligacion c ",
" where c.obl_rfc=a.obl_rfc ",
" and c.obl_deter=a.obl_deter and c.obl_clave=a.obl_clave)"
prepare x_caso131 from x_que
let hora=current
display 'Inicia comparacion de obl. que la local no tiene
',hora
execute x_caso131
insert into obligacion
select obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj
from Bita_16
where num_caso=131
update casos set fin = current where caso = 131
let hora=current
display ' Termina comparacion de obl. que la local no
tiene ',hora
display 'Listo obligacion...'

-- caso 116
-- Query que selecciona los Documentos que están en la
nacional que no existe en
-- la Local
initialize x_que to null
update casos set inicio = current where caso = 116
let x_que="insert into bita_14",
" select 116,0,0,CURRENT,"x_alr clipped","a.",
" from docnal a",
" where not exists(select * from documentos c where
a.doc_rfc=c.doc_rfc",

```

```

" and a.doc_deter=c.doc_deter and a.doc_nofh=c.doc_nofh
",
"and a.doc_fechop=c.doc_fechop and
a.doc_nope=c.doc_nope)"
prepare x_caso116 from x_que
let hora=current
display 'Inicia comparacion de doctos que la local no
tiene ',hora
execute x_caso116
insert into documentos
select
doc_rfc,doc_deter,doc_nofh,doc_fechop,doc_mofech,
doc_estcap,doc_nope,doc_tipfor,doc_motmov,doc_claope,
doc_nord,doc_resver,doc_movi
from Bita_14
where num_caso=116
update casos set fin = current where caso = 116
let hora=current
display ' Termina comparacion de docs. que la local no
tiene ',hora
display 'Listo documentos...'

-- caso 117
-- Query que selecciona los Externos que están en la nacional
que no existe en
-- la Local

initialize x_que to null
update casos set inicio = current where caso = 117
let x_que="insert into bita_15",
" select 117,0,0,CURRENT,"x_alr clipped","a.",
" from extnal a where not exists(select * from
externos c",
" where c.ext_rfc=a.ext_rfc and c.ext_deter=a.ext_deter and
",
" c.ext_nofh=a.ext_nofh",
" and c.ext_fechop=a.ext_fechop and
c.ext_nope=a.ext_nope)"
prepare x_caso117 from x_que
let hora=current
display 'Inicia comparacion de externos que la local no
tiene ',hora
execute x_caso117
insert into externos
select ext_rfc,ext_deter,ext_nofh,ext_fechop,ext_mofech,
ext_estcap,ext_nope,ext_tipfor,ext_motmov,ext_claope,
ext_nord,ext_resver
from Bita_15
where num_caso=117
update casos set fin = current where caso = 117
let hora=current
display ' Termina comparacion de ext. que la local no
tiene ',hora
display 'Listo externos...'

-- caso 86
-- Query que selecciona las Cedula que están en la nacional
que no existe en
-- la Local

initialize x_que to null
update casos set inicio = current where caso = 86
let x_que="insert into bita_13",
" select 86,0,0,CURRENT,"x_alr clipped","a.",",",
" from cednal a",
" where not exists( select * from cedulas c",
" where c.ced_rfc=a.ced_rfc and",
" c.ced_let=a.ced_let and c.ced_num=a.ced_num )"
prepare x_caso86 from x_que
let hora=current

```

```

display ' Comparando cedulas que la local no tiene
',hora
execute x_caso86
insert into cedulas
select
ced_rfc,ced_let,ced_num,ced_fecalt,ced_fecbaj,ced_mot,ced_ct
from Bita_13
where num_caso=86
update casos set fin = current where caso = 86
let hora=current
display ' Termina comparacion de cedulas que la local
no tiene ',hora
display 'Listo cedulas...'

-- caso 140
-- Query que selecciona los Nombres que están en la nacional
que no existe en
-- la Local

initialize x_que to null
update casos set inicio = current where caso = 140
let x_que="insert into bita_8",
" select 140,0,0,CURRENT,"x_alr clipped","a.",
" from nomnal a",
" where not exists(select * from nombres c",
" where c.nom_rfc=a.nom_rfc)"
prepare x_caso140 from x_que
let hora=current
display 'Inicia comparacion de nombres que la local no
tiene ',hora
execute x_caso140

declare crs_nom cursor for
select *
into nom1.*
from bita_8 where num_caso=140
foreach crs_nom
select nom_rfca from nombres
where nom_rfc=nom1.nom_rfc
or nom_rfca=nom1.nom_rfca
if status<>NOTFOUND then
let nom1.variante=1
insert into bita_8
values(nom1.*)
else
insert into nombres
values(nom1.nom_rfca,nom1.nom_rfc,nom1.nom_fec
ha)
end if
end foreach
close crs_nom
update casos set fin = current where caso = 140
let hora=current
display ' Termina comparacion de nom. que la local no
tiene ',hora
display 'Listo nombres...'

-- caso 82
-- Query que selecciona los Representantes que están en la
nacional que no existe en
-- la Local

initialize x_que to null
update casos set inicio = current where caso = 82
let x_que="insert into bita_12",
" select 82,0,0,CURRENT,"x_alr clipped","a.",",",
" from repnal a",
" where not exists( select * from representa c",
" where c.rep_rfcau=a.rep_rfcau and
c.rep_detcau=a.rep_detcau)"

```

```

prepare x_caso82 from x_que
let hora=current
display ' Comparando rep. que la local no tiene ',hora
execute x_caso82
insert into representa
select
rep_rfcau,rep_detcau,rep_rfc,rep_nomrep,rep_domrrep,rep_t
elef
from Bita_12
where num_caso=82
update casos set fin = current where caso = 82
let hora=current
display ' Termine de comparar los rep que la local no
tiene ',hora
-- caso 72
-- Query que selecciona las Sucesiones que están en la nacional
que no existe en
-- la Local
initialize x_que to null
update casos set inicio = current where caso = 72
let x_que="insert into bita_7",
" select 72,0,0,CURRENT,"x_alr clipped","a.",",",",",
" from sucnal a",
" where not exists(select * from sucesion c",
" where c.suc_rfc=a.suc_rfc)"
prepare x_caso72 from x_que
let hora=current
display ' Comparando sucesiones que la local no tiene
',hora
execute x_caso72
insert into sucesion
select
suc_rfc,suc_fechi,suc_nomalb,suc_rfcaib,suc_domalb
from Bita_7
where num_caso=72
update casos set fin = current where caso = 72
let hora=current
display ' Termina comparacion de sucesiones que la
local no tiene ',hora
display 'Listo sucesion...'
-- caso 77
-- Query que selecciona las Suspensiones que están en la
nacional que no existe en
-- la Local
update casos set inicio = current where caso = 77
initialize x_que to null
let x_que="insert into bita_10",
" select 77,0,0,CURRENT,"x_alr clipped","a.",",",
" from susnal a",
" where not exists(select * from suspension c",
" where c.sus_rfc=a.sus_rfc and c.sus_deter=a.sus_deter)"
prepare x_caso77 from x_que
let hora=current
display ' Comparando suspensiones que la local no
tiene ',hora
execute x_caso77
insert into suspension
select
sus_rfc,sus_deter,sus_fecsus,sus_nomaut,sus_domaut
from Bita_10
where num_caso=77
update casos set fin = current where caso = 77
let hora=current
display ' Termina comparacion de suspensiones que la
local no tiene ',hora
display 'Listo suspension...'

```

```

-- caso 66
-- Query que selecciona las Liquidaciones que están en la
nacional que no existe en
-- la Local
update casos set inicio = current where caso = 66
initialize x_que to null
let x_que="insert into bita_9",
" select 66,0,0,CURRENT,"x_alr clipped","a.",",",",",
" from liqnal a",
" where not exists(select * from liquidacion c",
" where c.liq_rfc=a.liq_rfc)"
prepare x_caso66 from x_que
let hora=current
display ' Comparando liquidaciones que la local no
tiene ',hora
execute x_caso66
insert into liquidacion
select
liq_rfc,liq_fechi,liq_nomliq,liq_rfdciq,liq_domliq,liq_telliq
from Bita_9
where num_caso=66
update casos set fin = current where caso = 66
let hora=current
display ' Termina comparacion de liquidaciones que la
local no tiene ',hora
display 'Listo liquidacion...'
end function # fin function Nalnoloc
function caso_137()
define cau1,cau2 char(13)
declare c137 cursor for
select nom_rfca from nombres
where nom_rfc is null or length(nom_rfc)=0
foreach c137 into cau1
select nom_rfc into cau2
from normal where nom_rfca=cau1
and nom_rfc is not null
if status<>NOTFOUND then
insert into bita_8
values(137,2,0,current,x_alr,cau1,",")
#meter a bitacora
update nombres
set nom_rfc=cau2
where nom_rfca=cau1
else
insert into bita_8
values(137,1,0,current,x_alr,cau1,",")
call borr137(cau1,137,1,x_alr)
end if
end foreach
end function
function borr137(cauP,cas,var,alr)
define cauP,cau2aux,cau2aux1 char(13)
define var,cas,alr smallint
let cau2aux=cauP
while TRUE
select nom_rfca into cau2aux1 from nombres
where nom_rfc=cau2aux
if status <> NOTFOUND then
call borseq(cau2aux,cas,var)
insert into comp_casos

```



```

        values(cas,var,0,alr,cau2aux,0,,"")
        let cau2aux=cau2aux1
        continue while
    else
        call borrseg(cau2aux,cas,var)
        insert into comp_casos
        values(cas,var,0,alr,cau2aux,0,,"")
        exit while
    end if
end while
end function

```

bajas sin causante

```

function caso_52()
define x_r52 record like bajas.*
define x_inst char(430),
        x_tabla,x_tabla2 varchar(10)
select a.* from bajas a where not exists ( select b.cau_rfc
from causante b where b.cau_rfc=a.baj_rfc
and b.cau_deter=a.baj_deter ) into temp sincau with no log
declare x_c52 cursor for
select * from sincau
foreach x_c52 into x_r52.*
-- El causante no existio en la extraccion de la
nacional
-- por lo que se borra de la local
-- se almacena el registro que se borrara en bita_6
insert into bita_6 values(52,1,0,current,x_alr,x_r52.*)
-- borra el registro de bajas
delete from bajas where baj_rfc=x_r52.baj_rfc
and baj_deter= x_r52.baj_deter
end foreach
drop table sincau
end function # caso 52

```

Liquidaciones sin causante

```

function caso_59()
define x_r59 record like liquidacion.*
define x_inst char(430),
        x_tabla,x_tabla2 varchar(10)
select a.* from liquidacion a where not exists ( select
b.cau_rfc
from causante b where b.cau_rfc=a.liq_rfc and b.cau_deter=0 )
into temp sincau with no log
declare x_c59 cursor for
select * from sincau
foreach x_c59 into x_r59.*
-- El causante no existio en la extraccion de la
nacional
-- por lo que se borra de la local
-- se almacena el registro que se borrara en bita_9
insert into bita_9
values(59,0,0,current,x_alr,"",x_r59.liq_rfc
,x_r59."x_alr)
-- borra el registro de liquidacion
delete from liquidacion where liq_rfc=x_r59.liq_rfc
end foreach
drop table sincau
end function # caso 59

```

Sucesiones sin causante

```

function caso_68()
define x_r68 record like sucesion.*

```

```

define x_inst char(430),
        x_tabla,x_tabla2 varchar(10)
select a.* from sucesion a where not exists ( select b.cau_rfc
from causante b where b.cau_rfc=a.suc_rfc and b.cau_deter=0
)
into temp sincau with no log
declare x_c68 cursor for
select * from sincau
foreach x_c68 into x_r68.*
-- El causante no existio en la extraccion de la
nacional
-- por lo que se borra de la local
-- se almacena el registro que se borrara en bita_7
insert into bita_7
values(68,1,0,current,x_alr,x_r68."")
-- borra el registro de sucesion
delete from sucesion where suc_rfc=x_r68.suc_rfc
end foreach
drop table sincau
end function # caso 68

```

suspension sin causante

```

function caso_74()
define x_r74 record like suspension.*
define x_inst char(430),
        x_tabla,x_tabla2 varchar(10)
select a.* from suspension a where not exists ( select
b.cau_rfc
from causante b where b.cau_rfc=a.sus_rfc
and b.cau_deter=a.sus_deter ) into temp sincau with no log
declare x_c74 cursor for
select * from sincau
foreach x_c74 into x_r74.*
-- El causante no existio en la extraccion de la
nacional
-- por lo que se borra de la local
-- se almacena el registro que se borrara en bita_10
insert into bita_10 values(74,1,0,current,x_alr,x_r74.*)
-- borra el registro de suspension
delete from suspension where sus_rfc=x_r74.sus_rfc
and sus_deter= x_r74.sus_deter
end foreach
drop table sincau
end function # caso 74

```

Representantes sin causante

```

function caso_80()
define x_r80 record like representa.*
define x_inst char(430),
        x_tabla,x_tabla2 varchar(10)
select a.* from representa a where not exists ( select
b.cau_rfc
from causante b where b.cau_rfc=a.rep_rfccau
and b.cau_deter=a.rep_detcau ) into temp sincau with no log
declare x_c80 cursor for
select * from sincau
foreach x_c80 into x_r80.*
-- El causante no existio en la extraccion de la
nacional
-- por lo que se borra de la local
-- se almacena el registro que se borrara en bita_12
insert into bita_12
values(80,1,0,current,x_alr,x_r80."")
-- borra el registro de representa
delete from representa where
rep_rfccau=x_r80.rep_rfccau

```

```

and rep_detcau = x_r80.rep_detcau
end foreach
drop table sincau
end function # caso 80

```

Cedulas sin causante

```

function caso_84()
define x_r84 record like cedulas.*
define x_inst char(430),
      x_tabla,x_tabla2 varchar(10)
select a.* from cedulas a where not exists ( select b.cau_rfc
from causante b where b.cau_rfc=a.ced_rfc and b.cau_deter=0
)
into temp sincau with no log
declare x_c84 cursor for
select * from sincau
foreach x_c84 into x_r84.*
-- El causante no existio en la extraccion de la
nacional
-- por lo que se borra de la local
-- se almacena el registro que se borrara en bita_13
insert into bita_13 values(84,1,0,current,x_alr,x_r84.*)
-- borra el registro de cedulas
delete from cedulas where ced_rfc=x_r84.ced_rfc
end foreach
drop table sincau
end function # caso 84

```

Documentos sin causante

```

function caso_100()
define x_r100 record like documentos.*
define x_inst char(430),
      x_tabla,x_tabla2 varchar(10)
select a.* from documentos a where not exists ( select
b.cau_rfc
from causante b where b.cau_rfc=a.doc_rfc
and b.cau_deter=a.doc_deter ) into temp sincau with no log
declare x_c100 cursor for
select * from sincau
foreach x_c100 into x_r100.*
-- El causante no existio en la extraccion de la
nacional
-- por lo que se borra de la local
-- se almacena el registro que se borrara en bita_14
insert into bita_14
values(100,1,0,current,x_alr,x_r100.*)
-- borra el registro de documentos
delete from documentos where
doc_rfc=x_r100.doc_rfc
and doc_deter= x_r100.doc_deter
end foreach
drop table sincau
end function # caso 100

```

Externos sin causante

```

function caso_101()
define x_r101 record like externos.*
define x_inst char(430),
      x_tabla,x_tabla2 varchar(10)
select a.* from externos a where not exists ( select b.cau_rfc
from causante b where b.cau_rfc=a.ext_rfc
and b.cau_deter=a.ext_deter ) into temp sincau with no log
declare x_c101 cursor for
select * from sincau
foreach x_c101 into x_r101.*

```

```

-- El causante no existio en la extraccion de la
nacional
-- por lo que se borra de la local
-- se almacena el registro que se borrara en bita_15
insert into bita_15
values(101,1,0,current,x_alr,x_r101.*)
-- borra el registro de externos
delete from externos where ext_rfc=x_r101.ext_rfc
and ext_deter= x_r101.ext_deter

```

```

end foreach
drop table sincau
end function # caso 101

```

Obligaciones sin causante

```

function caso_124()
define x_r124 record like obligacion.*
define x_inst char(430),
      x_tabla,x_tabla2 varchar(10)
select a.* from obligacion a where not exists ( select b.cau_rfc
from causante b where b.cau_rfc=a.obl_rfc
and b.cau_deter=a.obl_deter ) into temp sincau with no log
declare x_c124 cursor for
select * from sincau
foreach x_c124 into x_r124.*
-- El causante no existio en la obraccion de la
nacional
-- por lo que se borra de la local
-- se almacena el registro que se borrara en bita_16
insert into bita_16
values(124,1,0,current,x_alr,x_r124.*)
-- borra el registro de obligacion
delete from obligacion where obl_rfc=x_r124.obl_rfc
and obl_deter= x_r124.obl_deter
end foreach
drop table sincau
end function # caso 124

```

– Funcion Sin_Nomre

```

function caso_10()
define x_inst char(400), x_busnom char(1)
define x_rfc,x_rfcz char(13),x_nom,x_nom2,x_nom3,x_nom4
varchar(80)
declare x_f cursor for
select cau_rfc,cau_rsoc into x_rfc, x_nom
from causante where
cau_deter=0
foreach x_f
let x_busnom='N'
let x_nom2 = null
let x_nom2=quita_blanco(x_nom)
let x_nom2=x_nom2 clipped
if length(x_nom2)=0 or x_nom2 is null
or x_nom2 matches "/" then
select cau_rsoc into x_nom3 from hist_det where
cau_rfc=x_rfc and cau_deter=0
if status <> notfound then
let x_nom4=quita_blanco(x_nom3)
let x_nom4=x_nom4 clipped
if length(x_nom4)=0
or x_nom4 is null or
x_nom4 matches "/" then
let x_busnom='S'
else
insert into bita_3 values
(10,1,0,x_rfc,0,current,
x_alr,x_nom)
update causante set cau_rsoc = x_nom3
where cau_rfc = x_rfc and cau_deter =0

```

```

        end if
    else
        let x_busnom='S'
    end if
    if x_busnom='S' then
        let x_busnom='N'
        select nom_rfca into x_rfcz from nombres where
nom_rfcp=x_rfc
        if status <> notfound then
            select cau_rsoc into x_nom3 from causante
where
            cau_rfc=x_rfcz and cau_deter=0
            if status <> notfound then
                let x_nom4=quita_blanco(x_nom3)
                let x_nom4=x_nom4 clipped
                if length(x_nom4)=0
                    or x_nom4 is null or
                    x_nom4 matches "/" then
                    let x_busnom='S'
                else
                    let x_inst="insert into bita_1 select
10,2,0,current,"
                    ,x_alr,," from causante where
cau_rfc="",x_rfc,
                    "" and cau_deter = 0"
                    prepare x_1010 from x_inst
                    execute x_1010
                    update causante set cau_rsoc = x_nom3
                    where cau_rfc = x_rfc and cau_deter = 0
                    end if
                else
                    let x_busnom='S'
                end if
            else
                let x_busnom='S'
            end if
            if x_busnom='S' then
                select nom_rfcp into x_rfcz from nombres where
nom_rfca=x_rfc
                if status <> notfound then
                    select cau_rsoc into x_nom3 from causante
where
                    cau_rfc=x_rfcz and cau_deter=0
                    if status <> notfound then
                        let x_nom4=quita_blanco(x_nom3)
                        let x_nom4=x_nom4 clipped
                        if length(x_nom4)=0
                            or x_nom4 is null or
                            x_nom4 matches "/" then
                            let x_inst="insert into bita_1 select
10,3,0,current,"
                            ,x_alr,," from causante where
cau_rfc="",x_rfc,
                            "" and cau_deter = 0"
                            delete from causante where cau_rfc =
x_rfc and
                            cau_deter = 0
                        else
                            let x_inst="insert into bita_1 select
10,2,0,current,"
                            ,x_alr,," from causante where
cau_rfc="",x_rfc,
                            "" and cau_deter = 0"
                            prepare x_1010b from x_inst
                            execute x_1010b
                            update causante set cau_rsoc = x_nom3
                            where cau_rfc = x_rfc and cau_deter = 0
                            end if
                        else
                            let x_inst="insert into bita_1 select
10,3,0,current,"

```

```

        ,x_alr,," from causante where
cau_rfc="",x_rfc,
        "" and cau_deter = 0"
        delete from causante where cau_rfc =
x_rfc and
        cau_deter = 0
    end if
else
    let x_inst="insert into bita_1 select
10,3,0,current,"
    ,x_alr,," from causante where cau_rfc="",x_rfc,
    "" and cau_deter = 0"
    delete from causante where cau_rfc =
x_rfc and
    cau_deter = 0
end if
end if
end if
end foreach
End Function

```

```

function caso_15(n_alr)
define cau10 record
    cau_rfc char(13),
    cau_deter smallint
end record
define n int, edonal,n_alr smallint
    define crhnal char(5),crhnal1 char(3)
define x_caso char(300)
define fecall date,molalt char(2),cveact char(6)
define cau1,cau2aux,cau2aux1,nom_rfcaux char(13),ofh_mia
char(3)
define bv3,bv4,bv5,deter1 smallint

```

```

#####
# Escenario I para la solución del caso 15#
#####

```

```

create unique index pt_tmp1 on
tmpcau1(cau_rfc,cau_deter)

```

```

create table ptcausante( cau_rfc char(13),cau_deter
smallint)

```

```

insert into ptcausante
select distinct baj_rfc,baj_deter from bajas , tmpcau1
where baj_rfc=cau_rfc and baj_deter=cau_deter
and (baj_cvebaj='14' or baj_cvebaj='33')

```

```

insert into ptcausante
select a.cau_rfc,a.cau_deter
from tmpcau1 a, nombres b
where a.cau_rfc=b.nom_rfca and a.cau_deter=0

```

```

create index pt_tmp2 on ptcausante(cau_rfc,cau_deter)
drop table tmpcau1

```

```

declare c15 cursor for
select distinct cau_rfc,cau_deter from ptcausante

```

```

foreach c15 into cau1,deter1
select * from nombres where nom_rfca=cau1
if status<>NOTFOUND then
let cau2aux=cau1
while TRUE
select nom_rfcp into cau2aux1 from nombres
where nom_rfca=cau2aux

```

```

if status <> NOTFOUND then
  if cau2aux1=cau1 then
    let cau2aux=cau2aux1
    exit while
  else
    let cau2aux=cau2aux1
    continue while
  end if
else
  exit while
end if
end while
select cau_ofh[1,3] into ofh_mia from causante
where cau_rfc=cau2aux and cau_deter=0
if status=NOTFOUND then
  # se registra como sin paternidad
  call regnal(cau2aux,15,1,1,n_alr)
end if
else
  select cau_ofh[1,3] into ofh_mia from causante
  where cau_rfc=cau1 and cau_deter=0
  if status=NOTFOUND then
    # se registra como sin paternidad
    select num_caso from comp_casos where
    num_caso=15 and rfc=cau1 and deter=0
    if status=NOTFOUND then
      insert into
      bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
      values(15,1,1,cau1,0,CURRENT,n_alr)
      insert into comp_casos
      values(15,1,1,n_alr,cau1,0,"")
      call borseg(cau1,15,4)
      #HAY QUE BORRAR DE LA LOCAL
    end if
  end if
end if
end foreach
drop table ptcausante
close c15

#####
# Escenario 2 para la solucion del caso 15#
#####

declare c15_2 cursor for
select cau_rfc,cau_deter from bita_1
where num_caso=49 and cau_deter=0
foreach c15_2 into cau1,deter1

initialize nom_rfcaux to null
select nom_rfca into nom_rfcaux
from nombres
where nom_rfca=cau1

select distinct baj_rfc from bajas
where baj_rfc=cau1 and baj_deter=0
and (baj_cvebaj='14' or baj_cvebaj='33')
if status<>NOTFOUND OR nom_rfcaux is not null
then
select * from nombres where nom_rfca=cau1
if status<>NOTFOUND then
let cau2aux=cau1
while TRUE
select nom_rfc into cau2aux1 from nombres
where nom_rfca=cau2aux
if status <> NOTFOUND then
if cau2aux1=cau1 then
let cau2aux=cau2aux1
exit while
else
let cau2aux=cau2aux1

```

```

continue while
end if
else
exit while
end if
end while

select cau_ofh[1,3] into ofh_mia from causante
where cau_rfc=cau2aux and cau_deter=0
if status<>NOTFOUND then
# Se registra con paternidad de la ALR y
operado
call regnal(cau2aux,15,2,1,n_alr)
else
#Se registra como sin paternidad
#No operado
call regnal(cau2aux,15,2,2,n_alr)
end if
else
select cau_ofh[1,3] into ofh_mia from causante
where cau_rfc=cau1 and cau_deter=0
if status<>NOTFOUND then
select num_caso from comp_casos where
num_caso=15 and rfc=cau1 and deter=0
if status=NOTFOUND then
insert into
bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
values(15,2,1,cau1,0,CURRENT,n_alr)
insert into comp_casos
values(15,2,1,n_alr,cau1,0,"")
end if
else
select num_caso from comp_casos where
num_caso=15 and rfc=cau1 and deter=0
if status=NOTFOUND then
insert into
bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
values(15,2,2,cau1,0,CURRENT,n_alr)
insert into comp_casos
values(15,2,2,n_alr,cau1,0,"")
# SE BORRA DE LA LOCAL
call borseg(cau1,15,4)
end if
end if
end if
end foreach
close c15_2

#### FIN ESCENARIO 2

#####
# Escenario 3 para la solucion del caso 15#
#####

declare c15_1 cursor for
select a.cau_rfc,a.cau_deter from causante a
where a.cau_deter=0 and not exists(select b.cau_rfc
from hist_det b
where b.cau_rfc=a.cau_rfc and
b.cau_deter=a.cau_deter)

foreach c15_1 into cau1,deter1

```

```

initialize nom_rfcaux to null
select nom_rfca into nom_rfcaux
from nombres
where nom_rfca=cau1

select distinct baj_rfc from bajas
where baj_rfc=cau1 and baj_deter=deter1
and (baj_cvebaj='14' or baj_cvebaj='33')

if status<>NOTFOUND or nom_rfcaux is not null then
  select * from nombres where nom_rfca=cau1
  if status<> NOTFOUND then
    let cau2aux=cau1
    while TRUE
      select nom_rfc into cau2aux1 from nombres
      where nom_rfca=cau2aux
      if status <> NOTFOUND then
        if cau2aux1=cau1 then
          let cau2aux=cau2aux1
          exit while
        else
          let cau2aux=cau2aux1
          continue while
        end if
      else
        exit while
      end if
    end while
  end if
  select cau_ofh[1,3] into ofh_mia from causante
  where cau_rfc=cau2aux and cau_deter=0
  if status=NOTFOUND then
    call regnal(cau2aux,15,3,1,n_alr)
    # registrar como sin paternidad y no
    confirmado
  else
    # se registra con paternidad de la ALR y
    confirmados
    call regnal(cau2aux,15,3,2,x_alr)
    # ( en todos los nombres relacionados )
  end if

  else # else de que no se encontro en nombres por
  primera vez
    # se registra como sin paternidad

    select cau_ofh[1,3] into ofh_mia from causante
    where cau_rfc=cau1 and cau_deter=0
    if status=NOTFOUND then

      select num_caso from comp_casos where
      num_caso=15 and rfc=cau1 and deter=0
      if status=NOTFOUND then
        insert into
        bita_3(num_caso, variante, subvar, rfc, deter, fecha_hora, alr)
        values(15,3,1,cau1,0,CURRENT,n_alr)
        insert into comp_casos
        values(15,3,1,n_alr,cau1,0,"")
        # LA NACIONAL DECIDE SI SE BORRA O
      SE QUEDA
      end if
    else
      select num_caso from comp_casos where
      num_caso=15 and rfc=cau1 and deter=0
      if status=NOTFOUND then

        insert into
        bita_3(num_caso, variante, subvar, rfc, deter, fecha_hora, alr)
        values(15,3,2,cau1,0,CURRENT,n_alr)
        insert into comp_casos
        values(15,3,2,n_alr,cau1,0,"")

```

```

end if
end if
end if # no se encontro por primera vez
else # No se trata de un cambio de nombre
  initialize fecalt,motalt,cveact to null
  let bv3=0
  let bv4=0
  let bv5=0
  select cau_fecalt,cau_motalt,cau_cveact into fecalt,
  motalt,cveact from causante
  where cau_rfc=cau1 and cau_deter=0
  # identificar si existe un registro de inscripcion a la local
  select distinct doc_rfc from documentos
  where doc_rfc=cau1 and doc_deter=deter1
  and (doc_motmov='00' or doc_motmov='03')
  if status=NOTFOUND then
    #verificar si tiene datos completos (actividad,
    #fecha de inicio,motivo de alta.
    if fecalt is not null and motalt is not null and cveact is
    not null then
      # es caso 3|3
      let bv3=1
    end if
    else
      #se queda co 3|3
      let bv3=1
    end if
    # identificar un cambio de domicilio sin confirmar
    select distinct doc_rfc from documentos
    where doc_rfc=cau1 and doc_deter=deter1
    and (doc_motmov='00' or doc_motmov='03') and
    doc_movi=""

    if status<>NOTFOUND then
      # se queda co 3|4
      let bv4=1
    end if

    # Identificar un cambio de domicilio sin confirmar

    select distinct doc_rfc from documentos
    where doc_rfc=cau1 and doc_deter=0 and
    (doc_motmov='17' or doc_motmov='31') and
    doc_movi=""
    if status<>NOTFOUND then
      # se queda como 3|5
      let bv5=1
    else
      if fecalt is null and motalt is null and cveact is
      null then
        # se queda como 3|5
        let bv5=1
      end if
    end if
    if bv5=1 then
      if bv4=1 or bv3=1 then
        # inscripcion local para consulta nacional
        call regnal(cau1,15,3,3,x_alr)
      else # Cambio de domicilio no confirmado
        call regnal(cau1,15,3,5,x_alr)
      end if
    else
      if bv4=1 then
        # inscripcion no confirmada
        call regnal(cau1,15,3,4,x_alr)
      end if
      if bv3=1 then
        # No inscripcion pero datos llenos
        call regnal(cau1,15,3,3,x_alr)
      end if
    end if

```

```

end if
end foreach
close c15_1

end function

function regnal(cauP,cas,var,svar,alr)
  define cauP,cau2aux,cau2aux1 char(13)
  define var,cas,alr,svar smallint
  let cau2aux=cauP
  select distinct num_caso from comp_casos where
  num_caso=15 and rfc=cau2aux and deter=0
  # ya se habian metido en comp_casos.
  if status=NOTFOUND then
  # meter a bita_padre
  insert into comp_casos
  values(cas,var,svar,x_alr,cau2aux,0,"")
  while TRUE
  select nom_rfca into cau2aux1 from nombres
  where nom_rfc=cau2aux
  if status <> NOTFOUND then

      if cauP=cau2aux1 then
      insert into bita_3

values(cas,var,svar,cau2aux,0,current,x_alr,"")
      if (var=1 and svar=1) or (var=2 and
svar=2) then
          # borrar segmento
          call borraseg(cau2aux,15,4)
          exit while
          end if
          end if

          if cas=15 then
          # se registra archivo de
          select num_caso from
          comp_hijos where
          num_caso=15 and
          rfchij=cau2aux1
          if status=NOTFOUND then

              insert into bita_3

values(cas,var,svar,cau2aux,0,current,x_alr,"")
              insert into comp_hijos

          values(cas,var,svar,x_alr,cauP,cau2aux1)
          if (var=1 and svar=1) or (var=2
and svar=2) then
              # borrar segmento
              call
              borraseg(cau2aux,15,4)
              end if
              end if
              end if
              let cau2aux=cau2aux1
              continue while
          else
              insert into bita_3

values(cas,var,svar,cau2aux,0,current,x_alr,"")
          if (var=1 and svar=1) or (var=2 and svar=2) then
          # borrar segmento
          call borraseg(cau2aux,15,4)
          end if
          exit while

```

```

end if
end while
end if # fin de que ya se encontraba
end function

function caso_136()
  define x_que char(255),ofh_mia char(3),n_alr smallint
  define cau1 , cau2,cau2aux,cau2aux1 char(13)
  define xxy integer,x_tbi char(10),f_mayor date
  select ln.nom_rfca,ln.nom_rfc from nombres ln
  where not exists (select cau_rfc from causante
  where cau_rfc=ln.nom_rfca and cau_deter=0)
  into temp tb136 with no log
  declare c136 cursor for select * from tb136
  let n_alr=x_alr
  foreach c136 into cau1,cau2
  select * from nombres where nom_rfca=cau1 and
nom_rfc=cau2
  if status<>NOTFOUND then # si existe en nombres
  let cau2aux=cau2
  while TRUE
  select nom_rfc into cau2aux1 from nombres
  where nom_rfca=cau2aux
  if status <> NOTFOUND then
      if cau2aux1=cau1 then
      let cau2aux=cau2aux1
      exit while
      else
      let cau2aux=cau2aux1
      continue while
      end if
  else
  exit while
  end if
  end while
  select cau_ofh[1,3] into ofh_mia from causante
  where cau_rfc=cau2aux and cau_deter=0
  if status<>NOTFOUND then
      call borrrgal(cau2aux,136,1,n_alr)
  else
  # si no lo encuentro se marca para borrar
  call borrrgal(cau2aux,136,4,x_alr)
  end if
  end if
end foreach
close c136
end function

function borrrgal(cauP,cas,var,alr)
  define cauP,cau2aux,cau2aux1 char(13)
  define var,cas,alr smallint
  let cau2aux=cauP
  select distinct num_caso from comp_casos1 where
  num_caso=136 and rfc=cau2aux and deter=0
  # ya se habian metido en comp_casos1.
  if status=NOTFOUND then
  # meter a bita_padre
  insert into comp_casos1
  values(cas,var,0,x_alr,cau2aux,0,"")
  while TRUE
  select nom_rfca into cau2aux1 from nombres
  where nom_rfc=cau2aux
  if status <> NOTFOUND then
      if cas=136 then
      # se registra archivo de
      select num_caso from
      comp_hijos1 where

```

```

num_caso=136 and
rfchij=cau2aux1
if status=NOTFOUND then
    insert into comp_hijos1
        values(cas,var,0,x_alr,cauP,cau2aux1)
    end if
    if var!=1 then
        # borrar segmento
        insert into bita_3
            values(cas,var,0,cau2aux,0,current,x_alr,"")
        call
    end if
    end if
    let cau2aux=cau2aux1
    continue while
else
    if var!=1 then
        # borrar segmento
        insert into bita_3
            values(cas,var,0,cau2aux,0,current,x_alr,"")
        call borseg(cau2aux,136,5)
    end if
    end if
    end while
end if # fin de que ya se encontraba
end function

```

```

function borseg(causante,caso,vari)
define causante char(13), caso,vari smallint

```

```

    let xx_rfc=causante
    let xx_deter=0
    let xx_caso=caso
    let xx_var=vari
    call bor_bajas()
    call bor_suce()
    call bor_nombre()
    call bor_liquid()
    call bor_suspen()
    call bor_obligacion()
    call bor_extemos()
    call bor_representa()
    call bor_documentos()
    call bor_cedulas()
    call bor_causante()
end function

```

```

function caso_138()
define x_que char(255)
define cau0 record like nombres.*
define xxy integer,x_tbl char(10)
select * from nombres ln where not exists (select cau_rfc from
causante
where cau_rfc=ln.nom_rfc and cau_deter=0) into temp tb138
with no log
declare c138 cursor for select * from tb138
foreach c138 into cau0.*
    call reg138(cau0.nom_rfc,138,1,x_alr)
end foreach
End function

```

```

function reg138(cauP,cas,var,alr)
define cauP,cau2aux,cau2aux1 char(13)
define var,cas,alr smallint
let cau2aux=cauP
select distinct num_caso from comp_casos where
num_caso=138 and rfc=cau2aux and deter=0

```

```

# ya se habian metido en comp_casos.
if status=NOTFOUND then
# meter a bita_padre
insert into comp_casos
values(cas,var,0,x_alr,cau2aux,0,"")
while TRUE
select nom_rfca into cau2aux1 from nombres
where nom_rfc=cau2aux
if status <> NOTFOUND then
if cauP=cau2aux1 then
exit while
end if
# se registra archivo de
dudas y ALR confirmada
select num_caso from
comp_hijos where
num_caso=138 and
rfchij=cau2aux1
if status=NOTFOUND then
insert into comp_hijos
values(cas,var,0,x_alr,cauP,cau2aux1)
end if
let cau2aux=cau2aux1
continue while
else
exit while
end if
end if
end while
end if # fin de que ya se encontraba
end function

```

```

function bor_obligacion()
define y_inst char(300)
let y_inst="insert into bita_16 select
'xx_caso','xx_var',11,current,'
x_alr',' from obligacion where obl_rfc="xx_rfc clipped,
" and obl_deter="xx_deter
prepare y_sus5 from y_inst
execute y_sus5
-- elimina
delete from obligacion
where obl_rfc=xx_rfc
and obl_deter=xx_deter
end function #

```

```

function bor_extemos()
define y_inst char(300)
let y_inst="insert into bita_15 select
'xx_caso','xx_var',10,current,'
x_alr',' from extemos where ext_rfc="xx_rfc clipped,
" and ext_deter="xx_deter
prepare y_sus4 from y_inst
execute y_sus4
-- elimina
delete from extemos
where ext_rfc=xx_rfc
and ext_deter=xx_deter
end function #

```

```

function bor_representa()
define y_inst char(300)
let y_inst="insert into bita_12 select
'xx_caso','xx_var',9,current,'
x_alr',' from representa where rep_rfcau="xx_rfc
clipped,
" and rep_detcau="xx_deter
prepare y_sus3 from y_inst
execute y_sus3

```

```

-- elimina
delete from representa
where rep_rfcau=xx_rfc
and rep_detcau=xx_deter
end function #

function bor_documentos()
define y_inst char(300)
let y_inst=insert into bita_14 select
',xx_caso,',xx_var,',8,current,',
x_alr,',* from documentos where doc_rfc="",xx_rfc
clipped,
" and doc_deter=',xx_deter
prepare y_sus2 from y_inst
execute y_sus2
-- elimina
delete from documentos
where doc_rfc=xx_rfc
and doc_deter=xx_deter
end function #

function bor_cedulas()
define y_inst char(300)
let y_inst=insert into bita_13 select
',xx_caso,',xx_var,',7,current,',
x_alr,',* from cedulas where ced_rfc="",xx_rfc clipped,"
prepare y_sus1 from y_inst
execute y_sus1
-- elimina
delete from cedulas
where ced_rfc=xx_rfc
end function #

function bor_causante()
define y_inst char(300)
let y_inst=insert into bita_1 select
',xx_caso,',xx_var,',6,current,',
x_alr,',* from causante where cau_rfc="",xx_rfc clipped,
" and cau_deter=',xx_deter
prepare y_sus from y_inst
execute y_sus
-- elimina
delete from causante
where cau_rfc=xx_rfc
and cau_deter=xx_deter
end function

function bor_bajas()
define y_inst char(300)
let y_inst=insert into bita_6 select
',xx_caso,',xx_var,
',1,current,',x_alr,',* from bajas where baj_rfc="",
xx_rfc clipped," and baj_deter=',xx_deter
prepare y_baj from y_inst
execute y_baj
-- Elimina la baja
delete from bajas
where baj_rfc=xx_rfc and
baj_deter=xx_deter
end function

function bor_liquid()
define y_inst char(300)
let y_inst=insert into bita_9 select
',xx_caso,',xx_var,',4,current,',x_alr,
',,,liq_rfc,',* from liquidacion where liq_rfc="",
xx_rfc clipped,"
prepare y_liq from y_inst
execute y_liq
-- elimina la liquidacion
delete from liquidacion

```

```

where liq_rfc=xx_rfc
end function

function bor_suce()
define y_inst char(300)
let y_inst=insert into bita_7 select
',xx_caso,',xx_var,',2,current,',x_alr,
',,,suc_rfc,',* from sucesion where suc_rfc="",
xx_rfc clipped,"
prepare y_suc from y_inst
execute y_suc
-- elimina la sucesion
delete from sucesion
where suc_rfc=xx_rfc
end function

function bor_suspen()
define y_inst char(300)
let y_inst=insert into bita_10 select
',xx_caso,',xx_var,',5,current,',
x_alr,',* from suspension where sus_rfc="",xx_rfc clipped,
" and sus_deter=',xx_deter
prepare y_sus9 from y_inst
execute y_sus9
-- elimina la suspension
delete from suspension
where sus_rfc=xx_rfc
and sus_deter=xx_deter
end function

function bor_nombre()
define y_inst char(300)
if xx_caso=137 then
let y_inst=insert into bita_B select
',xx_caso,',xx_var,',3,current,',x_alr,
',* from nombres where nom_rfca="",xx_rfc clipped,"
prepare y_nom from y_inst
execute y_nom
-- elimina la suspension
delete from nombres
where nom_rfca=xx_rfc
else
let y_inst=insert into bita_B select
',xx_caso,',xx_var,',3,current,',x_alr,
',* from nombres where nom_rfcp="",xx_rfc
clipped,"
prepare y_nom1 from y_inst
execute y_nom1
-- elimina la suspension
delete from nombres
where nom_rfcp=xx_rfc
end if
end function

function quita_blanco(s)
define s,r char(80),
i,j,longi smallint
let i=1
let r=""
while i<= length(s)
if s(i,i)>' ' then
let r=r clipped,s(i,i)
end if
let i=i+1
end while
return r
end function # quita_blanco

```



```

function extdup()
define mayor float
define casos char(400)
define rec record
    cuantos float,
    ext_rfc char(13),
    ext_deter smallint,
    ext_nofh char(5),
    ext_estcap smallint,
    ext_nope integer
end record

select count(*)
cuantos,ext_rfc,ext_deter,ext_nofh,ext_estcap,ext_nope
from externos
group by ext_rfc,ext_deter,ext_nofh,ext_estcap,ext_nope
into temp temptext with no log;

declare c_ext cursor for
select * from temptext
where cuantos > 1
foreach c_ext into rec.*
select max(rowid) into mayor
from externos where ext_rfc=rec.ext_rfc and
ext_deter=rec.ext_deter
and ext_nofh=rec.ext_nofh and ext_estcap=rec.ext_estcap
and ext_nope=rec.ext_nope
let casos = 'insert into bita_15 select 111,1,0,current,'
,x_alr, ',' from externos where ext_rfc=',rec.ext_rfc,
' and ext_deter=',rec.ext_deter,' and ext_nofh=',rec.ext_nofh,
' and ext_estcap=',rec.ext_estcap,' and
ext_nope=',rec.ext_nope clipped,
'' and rowid < ', mayor clipped, ''
prepare pext from casos
execute pext
delete from externos where ext_rfc=rec.ext_rfc and
ext_deter=rec.ext_deter and ext_nofh=rec.ext_nofh and
ext_estcap=rec.ext_estcap and ext_nope=rec.ext_nope and
rowid < mayor
end foreach

End Function

```

MENU1.4GL

```

database shcp
globals
define x_alr char(2), x_inst char(400),
x_fin,x_fin2,x_fin3 datetime year to second,
x_pau char(1)
end globals
main
define ans,ans1,ans2 char(1),archivo char(40),
x_regreso smallint
defer quit
defer interrupt
let x_alr=arg_val(1)
call gen_casos()
open window x_menu at 2,5 with 21 rows,72 columns
attribute(border, yellow)
while True
call texto()
prompt "" for char ans
display ans at 21,45
case
when ans='1'
select fin into x_fin from casos where
caso=202
if x_fin is not null then

```

```

display "Esta opcion no puede ejecutarse, porque
ya se ejecuto la opcion 2" at 20,5 attribute (reverse)
display " =====> Presiona cero (0) para continuar
<===== " at 21,15 attribute (reverse)
let x_pau=' '
while x_pau<>'0'
prompt "" for char x_pau
end while

else
let x_inst="sh fin1.sh"
run x_inst
end if

when ans='2'
select fin into x_fin from casos where
caso=202
select fin into x_fin2 from casos where
caso=200
select fin into x_fin3 from casos where
caso=205
if x_fin2 is not null and x_fin is null and
x_fin3 is null then
open window x_menu1 at 14,15 with 5 rows,55
columns
attribute(border, green)
while True
call texto1()
prompt "" for char ans1
let ans1=upshift(ans1)
display ans1 at 4,35
if ans1='C' or ans1='F' then
exit while
end if
end while
close window x_menu1
let ans2=0
if ans1='C' then
open window x_menu2 at 14,15 with 5 rows,55
columns
attribute(border, green)
while True
call texto2()
prompt "" for char ans2
display ans2 at 4,35
if ans2 >= 0 then
exit while
end if
end while
close window x_menu2
end if
display 'Iniciando la carga de datos '
let x_inst="sh des_nal.sh ",x_alr,',ans1,',ans2
run x_inst
else
if x_fin3 is not null then
display "Esta opcion no puede ejecutarse, porque
ya se ejecuto la opcion 3" at 20,5 attribute (reverse)
else
if x_fin is null then
display "Esta opcion no puede ejecutarse,
porque no se ha pasado por la opcion 1" at 20,2 attribute
(reverse)
else
display "Esta opcion no puede ejecutarse, porque
ya se ejecuto la opcion 2" at 20,5 attribute (reverse)
end if
end if
display " =====> Presiona cero (0) para continuar
<===== " at 21,15 attribute (reverse)
let x_pau=' '

```

```

        while x_pau<>'0'
            prompt " for char x_pau
        end while
    end if

    when ans='3'
        display ans at 20,45
        select fin into x_fin from casos where
        caso=205
        select fin into x_fin2 from casos where
        caso=202
        select fin into x_fin3 from casos where
        caso=203
        if x_fin2 is not null and x_fin is null and
        x_fin3 is null then
            display "Verificando las características del equipo.

Espere"

            let x_inst="nohup sh checaalr.sh >> hola.txt"
            run x_inst
            let archivo="resck'x_alr clipped,'.txt"
            let x_regreso=pasa(archivo)
            run 'rm hola.txt'
            if x_regreso=1 then
                display "Los parametros del equipo son

incorrectos"

                display "la limpieza no se puede continuar."
                exit while
            end if
            let x_inst="sh limpia.sh"
            run x_inst
        else
            if x_fin3 is not null then
                display "Esta opcion no puede ejecutarse, porque
ya se ejecuto la opcion 4" at 20,5 attribute (reverse)
            else
                if x_fin2 is null then
                    display "Esta opcion no puede ejecutarse,
porque no se ha pasado por la opcion 2" at 20,2 attribute
(reverse)
                else
                    display "Esta opcion no puede ejecutarse,
porque ya se ejecuto la opcion 3" at 20,5 attribute (reverse)
                end if
            end if
            display " =====> Presiona cero (0) para continuar
<===== " at 21,15 attribute (reverse)
            let x_pau=' '
            while x_pau<>'0'
                prompt " for char x_pau
            end while
        end if

        when ans='4'
            select fin into x_fin from casos where
            caso=205
            if x_fin is not null then
                open window x_menu1 at 14,15 with 5 rows,55

columns
                attribute(border, green)
                while True
                    call texto3()
                    prompt "" for char ans1
                    let ans1=upshift(ans1)
                    display ans1 at 4,35
                    if ans1='C' or ans1='F' then
                        exit while
                    end if
                end while
                close window x_menu1
                if ans1='C' then

```

```

                open window x_menu2 at 14,15 with 5 rows,55

columns
                attribute(border, green)
                while True
                    call texto2()
                    prompt "" for char ans2
                    display ans2 at 4,35
                    if ans2 >= 0 then
                        exit while
                    end if
                end while
                close window x_menu2
            end if
            let x_inst="sh fin3.sh ",ans1,' ',ans2
            run x_inst

            call drop_tablas()
            open window x_menu1 at 14,15 with 5 rows,55

            attribute(border, green)
            let x_pau=' '
            while x_pau not matches "[NnSs]"
                clear window x_menu1
                display "Va a realizar otro respaldo <S/N>" at
2,10
            end while
            if x_pau='n' or x_pau='N' then
                run 'touch borrado.out'
            end if
            close window x_menu1
        else
            display "Esta opcion no puede ejecutarse, " at
19,19 attribute (reverse)
            display " porque 'NO' se ha ejecutado la opcion 3 "
at 20,15 attribute (reverse)
            display " =====> Presiona cero (0) para continuar
<===== " at 21,15 attribute (reverse)
            let x_pau=' '
            while x_pau<>'0'
                prompt " for char x_pau
            end while
        end if

        when ans='5'
            select fin into x_fin from casos where
            caso=300
            select fin into x_fin3 from casos where
            caso=203
            if x_fin is null and x_fin3 is not null then
                let x_inst="sh migracion.sh"
                run x_inst
            else
                if x_fin3 is null then
                    display "Esta opcion no puede ejecutarse, porque
no se ha realizado la limpieza" at 20,2 attribute (reverse)
                else
                    display "Esta opcion no puede ejecutarse, porque
ya se ejecuto la Migracion" at 20,5 attribute (reverse)
                end if
            end if
            display " =====> Presiona cero (0) para continuar
<===== " at 21,15 attribute (reverse)
            let x_pau=' '
            while x_pau<>'0'
                prompt " for char x_pau
            end while
        end if
        when ans='6'
            select fin into x_fin from casos where
            caso=300
            if x_fin is not null then

```

```

open window x_menu1 at 14,15 with 5 rows,55
columns
  attribute(border, green)
  while True
    call texto3()
    prompt "" for char ans1
    let ans1=upshift(ans1)
    display ans1 at 4,35
    if ans1='C' or ans1='F' then
      exit while
    end if
  end while
  close window x_menu1
  if ans1='C' then
    open window x_menu2 at 14,15 with 5 rows,55
    columns
      attribute(border, green)
      while True
        call texto2()
        prompt "" for char ans2
        display ans2 at 4,35
        if ans2 >= 0 then
          exit while
        end if
      end while
      close window x_menu2
    end if
    let x_inst="sh fin3m.sh ",ans1,',',ans2
    run x_inst
  end if
  open window x_menu1 at 14,15 with 5 rows,55
  columns
    attribute(border, green)
    let x_pau=""
    while x_pau not matches "[NnSs]"
      clear window x_menu1
      display "Va a realizar otro respaldo <S/N>" at 4,10
      prompt "" for char x_pau
      end while
      if x_pau='n' or x_pau='N' then
        run 'touch borratodo.out'
      end if
      close window x_menu1
    else
      display "Esta opcion no puede ejecutarse, " at
19,19 attribute (reverse)
      display " porque 'NO' se ha realizado la migracion "
at 20,15 attribute (reverse)
      display " ===== Presiona cero (0) para continuar
<===== " at 21,15 attribute (reverse)
      let x_pau=""
      while x_pau<>'0'
        prompt "" for char x_pau
      end while
    end if
  when ans='9'
    display "9" at 20,10
    exit while
  otherwise
    display "Opcion Invalida" at 21,50
    attribute (white,reverse, blink)
    sleep 2
    display " " at 21,50
  end case
  clear window x_menu
end while
exit program
end main
function texto()
  define hora char(49)

```

```

  let hora=current
  let hora=hora[1,10]
  display "PROCESO MASIVO: Limpieza de Datos en la ALR" at
2,15
  attribute (white)
  display "A.L.R.: ",x_alr using '&&', " Fecha: ",hora
at 3,12
  attribute (white)
  display "-----" at 4,8
  attribute (white)
  display " 1) Respaldo de las 11 tablas de RFC antes de limpiar
" at 6,10
  attribute (white)
  display " 2) Carga de Datos provenientes de CPN " at
8,10
  attribute (white)
  display " 3) <<<<< Limpieza de la ALR >>>>> " at
10,10
  attribute (white)
  display " 4) Descarga total de informacion por enviar a CPN "
at 12,10
  attribute (white)
  display " 5) <<<<< Migracion de la ALR >>>>> " at
14,10
  attribute (white)
  display " 6) Descarga de informacion Migrada por enviar a CPN
" at 16,10
  attribute (white)
  display " 9) Salida " at 18,10
  attribute (white)
  display " Teclee su opcion ..." at 21,25
  attribute (white)
end function
function texto1()
  display "La carga de datos sera por <C>inta o <F>tp ?" at 2,5
  attribute (white)
  display " Teclee su opcion ..." at 4,10
  attribute (white)
end function
function texto2()
  display "Proporcione la unidad fisica de cinta?" at 3,5
  attribute (white)
end function
function gen_casos()
  SELECT * FROM systables WHERE tablename = 'casos'
  if status = NOTFOUND then
    CREATE TABLE casos (
      caso smallint,
      descripcion varchar(75),
      listo char(1),
      programa char(10),
      nivel varchar(8),
      orden smallint,
      resp varchar(2),
      requiere smallint,
      inicio DATETIME YEAR TO FRACTION(3),
      fin DATETIME YEAR TO FRACTION(3),
      termino varchar(1),
      tablas varchar(200),
      comenta varchar(100)
    ) in dbsir;
  CREATE UNIQUE INDEX caso ON casos
  (
    caso ASC
  )
  LOAD FROM "casos.unl" INSERT INTO casos;
end if
end function
function texto3()

```

```

display 'El envio de datos sera por <C>inta o <F>tp o <N>o se
?' at 2,2
attribute (white)
display * Teclee su opcion ...' at 4,17
attribute (white)
end function

function drop_tablas()

SELECT * FROM systables WHERE tabname = 'res_casos'
if status <> NOTFOUND then
drop table res_casos
end if
SELECT * FROM systables WHERE tabname = 'res_casos1'
if status <> NOTFOUND then
drop table res_casos1
end if
SELECT * FROM systables WHERE tabname = 'bita_matriz'
if status <> NOTFOUND then
drop table bita_matriz
end if
SELECT * FROM systables WHERE tabname = 'cat_cedmot'
if status <> NOTFOUND then
drop table cat_cedmot
end if
SELECT * FROM systables WHERE tabname = 'cat_contesp'
if status <> NOTFOUND then
drop table cat_contesp
end if
SELECT * FROM systables WHERE tabname = 'cat_obl_obs'
if status <> NOTFOUND then
drop table cat_obl_obs
end if
SELECT * FROM systables WHERE tabname = 'cat_oblissr'
if status <> NOTFOUND then
drop table cat_oblissr
end if
SELECT * FROM systables WHERE tabname = 'cat_eqcrh'
if status <> NOTFOUND then
drop table cat_eqcrh
end if
SELECT * FROM systables WHERE tabname = 'cat_tip_per'
if status <> NOTFOUND then
drop table cat_tip_per
end if
SELECT * FROM systables WHERE tabname = 'cat_tipcont'
if status <> NOTFOUND then
drop table cat_tipcont
end if
SELECT * FROM systables WHERE tabname = 'cat_tipemp'
if status <> NOTFOUND then
drop table cat_tipemp
end if
SELECT * FROM systables WHERE tabname = 'cat_tipsoc'
if status <> NOTFOUND then
drop table cat_tipsoc
end if
SELECT * FROM systables WHERE tabname = 'cat_status'
if status <> NOTFOUND then
drop table cat_status
end if
SELECT * FROM systables WHERE tabname = 'cat_comp2'
if status <> NOTFOUND then
drop table cat_comp2
end if
select tabname from systables where tabname='hist_det'
if status <> NOTFOUND then
drop table hist_det
end if
select tabname from systables where tabname='bajnal'

```

```

if status <> NOTFOUND then
drop table bajnal
end if
select tabname from systables where tabname='obinal'
if status <> NOTFOUND then
drop table obinal
end if
select tabname from systables where tabname='docnal'
if status <> NOTFOUND then
drop table docnal
end if
select tabname from systables where tabname='extrnal'
if status <> NOTFOUND then
drop table extrnal
end if
select tabname from systables where tabname='cednal'
if status <> NOTFOUND then
drop table cednal
end if
select tabname from systables where tabname='nomnal'
if status <> NOTFOUND then
drop table nomnal
end if
select tabname from systables where tabname='repnal'
if status <> NOTFOUND then
drop table repnal
end if
select tabname from systables where tabname='sucnal'
if status <> NOTFOUND then
drop table sucnal
end if
select tabname from systables where tabname='susnal'
if status <> NOTFOUND then
drop table susnal
end if
select tabname from systables where tabname='liqnal'
if status <> NOTFOUND then
drop table liqnal
end if
SELECT * FROM systables WHERE tabname = 'comp_casos1'
if status <> NOTFOUND then
drop table comp_casos1
end if
SELECT * FROM systables WHERE tabname = 'comp_casos'
if status <> NOTFOUND then
drop table comp_casos
end if
SELECT * FROM systables WHERE tabname = 'comp_hijos'
if status <> NOTFOUND then
drop table comp_hijos
end if
SELECT * FROM systables WHERE tabname = 'comp_hijos1'
if status <> NOTFOUND then
drop table comp_hijos1
end if
SELECT * FROM systables WHERE tabname = 'bita_caudel'
if status <> NOTFOUND then
drop table bita_caudel
end if
end function

```

LIMPSUC.4GL

Database shcp

globals

```

define x_alr char(2),
rw_liq record like liquidacion.*,
rw_suc record like sucesion.*,
w_x1 smallint,
id2 float

```

end globals

main

```
-- obtiene la ALR al ejecutar estcau.4ge /opt/siralr/alr --  
let x_alr= arg_val(1)  
defer quit  
defer interrupt
```

update casos set inicio = current where caso = 69

```
declare c1 cursor for  
select *.rowid into rw_suc.*, id2  
from sucesion  
where sucesion.suc_rfc[1,1] = ''
```

```
foreach c1  
call caso_69()  
end foreach
```

close c1

```
update casos set fin = current where caso = 69  
update casos set inicio = current where caso = 71  
call Caso_71_REG()  
update casos set fin = current where caso = 71
```

end main

-- Funcion tipo

function Caso_71_REG()

```
define cau0 record like sucesion.*,id float  
define nombre char(80),domicilio char(80),telefono char (10)  
define sucrec record  
num_caso smallint ,  
variante smallint,  
subvar smallint,  
fecha_hora datetime YEAR TO SECOND ,  
alr smallint ,  
suc_rfc char(13),  
suc_fechi date,  
suc_normalb varchar(80),  
suc_rfcalb char(13),  
suc_domalb varchar(80),  
rfc_alb char(13),  
alr_alb smallint
```

end record

```
define sucrec2 record  
num_caso smallint,  
variante smallint,  
subvar smallint,  
alr smallint,  
rfc char(13),  
deter smallint,  
ofh char(5),  
alr smallint
```

end record

```
declare c71 cursor for  
select *.rowid into cau0.*,id from sucesion
```

```
foreach c71  
initialize sucrec.* to NULL  
initialize sucrec2.* to NULL
```

if length(cau0.suc_rfcalb) >= 13 then

```
if cau0.suc_rfc = cau0.suc_rfcalb then  
let sucrec.num_caso=71  
let sucrec.variante=1  
let sucrec.subvar=0  
let sucrec.fecha_hora=current  
let sucrec.alr=x_alr
```

```
let sucrec.suc_rfc=cau0.suc_rfc  
let sucrec.suc_normalb=cau0.suc_normalb  
let sucrec.suc_rfcalb=cau0.suc_rfcalb  
let sucrec.suc_domalb=cau0.suc_domalb  
insert into bita_7 values (sucrec.*)  
update sucesion set suc_normalb = ",suc_rfcalb = ",  
suc_domalb = " where
```

rowid=id

Else

```
Select cau_rsoc,cau_domi into nombre,domicilio  
from causante where cau_rfc =
```

cau0.suc_rfcalb

```
and cau_deter = 0  
if Status <> notfound then  
let sucrec.num_caso=71  
let sucrec.variante=2  
let sucrec.subvar=0
```

```
let sucrec.fecha_hora=current  
let sucrec.alr=x_alr  
let sucrec.suc_rfc=cau0.suc_rfc  
let sucrec.suc_normalb=cau0.suc_normalb  
let sucrec.suc_rfcalb=cau0.suc_rfcalb  
let sucrec.suc_domalb=cau0.suc_domalb  
insert into bita_7 values (sucrec.*)  
update sucesion set suc_normalb = nombre,  
suc_domalb = domicilio where rowid=id
```

Else

```
let sucrec.num_caso=71  
let sucrec.variante=3  
let sucrec.subvar=0
```

```
let sucrec.fecha_hora=current  
let sucrec.alr=x_alr  
let sucrec.suc_rfc=cau0.suc_rfc  
let sucrec.suc_normalb=cau0.suc_normalb  
let sucrec.suc_rfcalb=cau0.suc_rfcalb  
let sucrec.suc_domalb=cau0.suc_domalb  
insert into bita_7 values (sucrec.*)
```

```
let sucrec2.num_caso=71  
let sucrec2.variante=3
```

```
let sucrec2.subvar=0  
let sucrec2.alr=x_alr  
let sucrec2.rfc=cau0.suc_rfcalb  
let sucrec2.deter=0  
insert into comp_casos values (sucrec2.*)
```

End If

End If

end if

end foreach

end function

function caso_69() ##sustituir por funcion

```
select * into rw_liq.*  
from liquidacion  
where liq_rfc = rw_suc.suc_rfc  
if status = notfound then  
let w_x1 = 1 ##causante no existe en
```

liquidacion

```
call grabit_9(w_x1)  
return
```

end if

```
let w_x1 = 2 ##causante existe en las dos
```

```
call grabit_9(w_x1)
```

end function

function grabit_9(w_x1)

```
define w_x1 smallint  
if w_x1 = 1 then
```

```
insert into liquidacion values(rw_suc.suc_rfc,  
rw_suc.suc_fechi,
```

```

nw_suc.suc_nomalb,
                                rw_suc.suc_rfcalb,

nw_suc.suc_domalb,
                                null
                                )
end if

insert into bita_7 values(69,
                                w_x1,
                                0,
                                current,
                                x_alr,
                                rw_suc.suc_rfc,
                                rw_suc.suc_fechi,
                                rw_suc.suc_nomalb,
                                rw_suc.suc_rfcalb,
                                rw_suc.suc_domalb,
                                null,
                                null
                                )

delete from sucesion
where rowid = ld2 #####rw_liq.liq_rfc =
rw_suc.suc_rfc and

end function

LIMPSLES.4GL
database shcp
globals
define p_correr char(20)
define id,conta float, stat_cau integer,s_char,x_bajmov,x_alr
char(2)
define f_baj_f_liq,f_suca,f_sus,f_nom,f_mayor
date,x_numnon,stat_ori char(1),
x_vari char(10),x_caso char(2),alr_1,x_svar
smallint,x_num integer
define cau0 record like causante.*
define bit11 record
num_caso smallint ,
variante smallint,
subvar smallint,
rfc char(13) ,
deter smallint,
fecha_hora datetime YEAR TO SECOND ,
alr smallint ,
cau_motait char(2),
cau_calcont smallint,
cau_tipsoc smallint,
cau_tipcont char(1),
cau_secgob integer,
cau_cveact char(6),
cau_movs char(1),
baj_cvebaj char(2),
calcont_nal smallint,
tipcont_nal char(1)
end record
define w_doc, w_ext, w_obliga smallint, w_fecfis char(8),
w_ini,w_fin char(25), hora char(30),y_stat varchar(2),
r_stat integer,q_mayor integer,g_mov char(2)
define g_afir smallint,g_ofh char(5),g_fechop date, g_nope
integer,

```

```

x_rfcnvo char(13),c_num char(3),nulos
char(1),x_reactiva char(1)
end globals
main
lock table causante in exclusive mode
lock table bajas in exclusive mode
let x_num=0
-- obtiene la ALR al ejecutar estcau.4ge /u/siralr/alr
let x_alr= arg_val(1)
defer quit
defer interrupt
update casos set inicio = current where caso =141
call caso_141()
update casos set fin = current where caso =141
unlock table causante
unlock table bajas
let p_correr = "bitagral ",x_alr
run p_correr
update casos set fin = current where caso = 205
call llena_rfc_del()
end main
-- funcion comige estatus del contribuyente
function caso_141()
declare c1 cursor for
select *,rowid
into cau0.*,id
from causante where cau_deter>0
foreach c1
let nulos='N'
if cau0.cau_movs is null then
let nulos='S'
let cau0.cau_movs=' '
end if
let stat_on=cau0.cau_movs
let x_caso='141'
let x_numnon='N'

initalize f_baj_x_vari to null
let stat_cau=0
let q_mayor=0
---- Busca en bajas ----
select baj_fec,baj_cvebaj into f_baj,x_bajmov from bajas
where baj_rfc=cau0.cau_rfc and
baj_deter=cau0.cau_deter
if status<> notfound then
let stat_cau=stat_cau+1
else
let x_bajmov="
end if
let r_stat=stat_cau
let y_stat=stat_cau
case
-- Contribuyente en OK
when cau0.cau_movs=' ' and stat_cau=0
if nulos='S' then
insert into bita_11 values(141,0,0,cau0.cau_RFC,
cau0.cau_deter,current,x_alr,null,null,null,null,null,null,null,
null,null)
-- Actualiza el casuante con el estatus
update causante set cau_movs=' ' where rowid=id
end if
-- Si el contribuyente no esta activo
-- asigna el estatus encontrado
when cau0.cau_movs<>' ' and stat_cau=1
----- variante .. 141,1,0 --

let x_vari='1'
let x_svar=0
call estabaj()

```

```

call asig_stat()

-- Si el contribuyente esta activo, pero
-- existe informacion en
bajas,liquidacion
-- sucesion o suspension, se busca una
-- reactivacion
when cau0.cau_movs=' ' and stat_cau=1
if busca()=N' then
-----
- variante .. 141,3,0 -
-----
let x_vari='3'
let x_svar=0
call estabaj()
call asig_stat()
else
-----
- variante .. 2 --
-----
let x_vari='2'
if nulos='S' then
insert into bita_11 values(141,2,0,cau0.cau_RFC,
cau0.cau_deter,current,x_alr,null,null,null,null,null,
null,null, null,null)
-- Actualiza el casuante con el estatus
update causante set cau_movs=' ' where
rowid=id
else
insert into bita_11
values(141,2,0,cau0.cau_RFC,cau0.cau_deter,
current,x_alr,null,null,null,null,null,
cau0.cau_movs,null,null,null)
end if
call bor_bajas() # Si existe se deja
activo y se borra # el segmento.
end if
-- Esta en mas de un segmento o en la
-- tabla de nombres sin registro en
bajas
When cau0.cau_movs<>' ' and stat_cau=0
-----
- Grabar en archivos que van a ir a la
nacional
- para su correccion. V A R I A N T E ..... 9
-----
insert into bita_11
values(141,4,0,cau0.cau_RFC,cau0.cau_deter,current,
x_alr,null,null,null,null,null,cau0.cau_movs,null,
null,null)
-----
.-Se activa al causante, porque ya se
verificaron los --
-- segmentos
-----
update causante set cau_movs=' ' where rowid=id
let cau0.cau_movs=' '

end case

end foreach
close c1
end function # fin de la funcion corrige status del contribuyente
----- Establece si se asigna el nuevo status
function asig_stat()
if stat_ori <> cau0.cau_movs then
-- Almacena es status del causante original a bita 11

```

```

insert into bita_11
values(141,x_vari,x_svar,cau0.cau_RFC,
cau0.cau_deter,current,x_alr,null,null,null,null,null,stat_ori,nul
l,
null,null)
-- Actualiza el casuante con el estatus
update causante set cau_movs=cau0.cau_movs where
rowid=id
end if
end function # signa status
-- Establece el tipo de baja
function estabaj()
define
r_id float ,
x_inst char(100),
x_rfcno char(13),
c_num char(3)
if x_bajmov<>'06' and x_bajmov<>'50' then
let cau0.cau_movs='2'
else
let cau0.cau_movs='1'
end if
end function # estabaj

function busca()
define x_mov char(2) ,
x_inst char(200),
p_mot char(50),
fec date,
x_tantos_x_hay float,
xenco char(1),
x_ope integer

initialize fec,x_tantos to null
let fec=f_baj
-- Genera la fusion de documento y externos
call fusdoctos(fec)

let x_inst=0
-- Localiza una reactivacion, reanudacion de
actividades,
-- MC reactivacion de actividades y MC reactivacion
por
-- cambio de nombre o una baja del mismo dia.
declare c_rea cursor for
select doc_nope,doc_motmov from doc_ext2
where doc_fechop >= fec order by doc_nope
let xenco=' '
let x_tantos=0
foreach c_rea into x_ope,x_mov
if x_mov='16' or x_mov='22' or
x_mov='54' or x_mov='56'
then
if xenco=' ' then
let x_tantos=4
let xenco='R' # encuentre una reactivacion
# o reanudacion
else
continue foreach
end if
else
if xenco=' ' then
continue foreach
else
let x_tantos=-1
exit foreach
end if
end if
end foreach
drop table doc_ext2

```

```

if x_tantos > 0 then
    return "S"
else
    return "N"
end if
end function #busca

function bor_bajas()
    define y_inst char(400)
    -- Guarda la informacion de bajas en bita_9
    let y_inst=insert into bita_6 select 'x_caso','S',
    ',1,current','x_alr',' from bajas where baj_rfc="',
    cau0.cau_rfc,'" and baj_deter=',cau0.cau_deter
    prepare y_baj from y_inst
    execute y_baj
    -- Elimina la baja
    delete from bajas
    where baj_rfc=cau0.cau_rfc and
    baj_deter=cau0.cau_deter
end function # (borra bajas)
----- Fusiona documentos y externos
function fusdoctos(fec)
    define fec date
    select doc_nope,doc_motmov,doc_fechop from
documentos
    where doc_RFC=cau0.cau_rfc and
doc_deter=cau0.cau_deter and
doc_fechop >= fec and doc_motmov in ("16","22","54","56",
"01","02","04","05","06","07","08","09","18","20","21","29",
"41","42","43","46","47","48","49","50","51","52","53")
    into temp doc_ext2 with no log

insert into doc_ext2
select ext_nope,ext_motmov,ext_fechop from externos
where ext_RFC=cau0.cau_rfc and
ext_deter=cau0.cau_deter and
ext_fechop >= fec and ext_motmov in ("16","22","54","56",
"01","02","04","05","06","07","08","09","18","20","21","29",
"41","42","43","46","47","48","49","50","51","52","53")
create index x_doc_ext on
doc_ext2(doc_fechop,doc_motmov)
end function

function liena_rfc_del()
insert into bita_caudel
select num_caso,variante,subvar,cau_rfc,cau_deter,cau_rsoc
from bita_1
where num_caso = 15 and variante = 4 and subvar = 6

insert into bita_caudel
select num_caso,variante,subvar,cau_rfc,cau_deter,cau_rsoc
from bita_1
where num_caso = 138 and variante = 3 and subvar = 6

insert into bita_caudel
select num_caso,variante,subvar,cau_rfc,cau_deter,cau_rsoc
from bita_1
where num_caso = 137 and variante = 1 and subvar = 6

insert into bita_caudel
select num_caso,variante,subvar,cau_rfc,cau_deter,cau_rsoc
from bita_1
where num_caso = 136 and variante = 5 and subvar = 6

insert into bita_caudel
select num_caso,variante,subvar,cau_rfc,cau_deter,cau_rsoc
from bita_1
where num_caso = 10 and variante = 3 and subvar = 0

insert into bita_caudel

```

```

select num_caso,variante,subvar,cau_rfc,cau_deter,cau_rsoc
from bita_1
where num_caso = 1 and variante = 0 and subvar = 0

```

```

insert into bita_caudel
select num_caso,variante,subvar,cau_rfc,cau_deter,cau_rsoc
from bita_1
where num_caso = 2 and variante = 0 and subvar = 0
end function

```

LIMPREP.4GL

```

database shcp
globals
    define x_alr char(2), hora char(30)
end globals
main
    -- obtiene la ALR al ejecutar estcau.4ge 'u/siratr/alr' --
    let x_alr= arg_val(1)
    defer quit
    defer interrupt

update casos set inicio = current where caso = 81
    Call caso_81_REG()
update casos set fin = current where caso = 81

end main

```

-- Funcion tipo

```

function caso_81_REG()
define cau0 record like representa.*,id float
define nombre char(80),domicilio char(80),telefono char (10)
define reprec record
    num_caso          smallint ,
    variante          smallint,
    subvar           smallint,
    fecha_hora       datetime YEAR TO SECOND ,
    alr              smallint ,
    rep_rfccau      char(13),
    rep_detcau      smallint,
    rep_rfcrcp      char(13),
    rep_nomrep      varchar(80),
    rep_domrep      varchar(80),
    rep_telef       varchar(10),
    rfc_rep         char(13),
    alr_rep         smallint

end record
define reprec2 record
    num_caso smallint,
    variante smallint,
    subvar smallint,
    alr smallint,
    rfc char(13),
    deter smallint,
    ofn char(5),
    alr smallint
end record
declare c81 cursor for
select *,rowid
into cau0.*,id
from representa where rep_detcau=0
foreach c81
initialize reprec2.* to NULL
initialize reprec.* to NULL

if length(cau0.rep_rfcrcp) >= 13 then

    If cau0.rep_rfccau = cau0.rep_rfcrcp then
        let reprec.num_caso=81
        let reprec.variante=1
        let reprec.subvar=0
    
```



```

let reprec.fecha_hora=current
let reprec.alr=x_alr
let reprec.rep_rfccau=cau0.rep_rfccau
let reprec.rep_detcau=cau0.rep_detcau
let reprec.rep_nomrep=cau0.rep_nomrep
let reprec.rep_rfcrc=cau0.rep_rfcrc
let reprec.rep_domrep=cau0.rep_domrep
let reprec.rep_telef=cau0.rep_telef
insert into bita_12 values (reprec.*)
delete from representa where rowid=id
Else
Select cau_rsoc,cau_domi,cau_telefo into
nombre,domicilio,telefono
from causante where cau_rfc = cau0.rep_rfcrcp and
cau_deter = 0
If Status <> notfound then
let reprec.num_caso=81
let reprec.variante=2
let reprec.subvar=0
let reprec.fecha_hora=current
let reprec.alr=x_alr
let reprec.rep_rfccau=cau0.rep_rfccau
let reprec.rep_detcau=cau0.rep_detcau
let reprec.rep_nomrep=cau0.rep_nomrep
let reprec.rep_rfcrc=cau0.rep_rfcrcp
let reprec.rep_domrep=cau0.rep_domrep
let reprec.rep_telef=cau0.rep_telef
insert into bita_12 values (reprec.*)
update representa set rep_nomrep = nombre,
rep_domrep = domicilio,
rep_telef = telefono where rowid=id
Else
let reprec.num_caso=81
let reprec.variante=3
let reprec.subvar=0
let reprec.fecha_hora=current
let reprec.alr=x_alr
let reprec.rep_rfccau=cau0.rep_rfccau
let reprec.rep_detcau=cau0.rep_detcau
let reprec.rep_nomrep=cau0.rep_nomrep
let reprec.rep_rfcrc=cau0.rep_rfcrcp
let reprec.rep_domrep=cau0.rep_domrep
let reprec.rep_telef=cau0.rep_telef
insert into bita_12 values (reprec.*)
let reprec2.num_caso=81
let reprec2.variante=3
let reprec2.subvar=0
let reprec2.alr=x_alr
let reprec2.rfc=cau0.rep_rfcrcp
let reprec2.deter=0
insert into comp_casos values (reprec2.*)
End If
End If
end if
end foreach
end function

```

LIMPOBL.4GL

database shcp

globals

```

define x_alr char(2),
nw_b16 record
num_caso          smallint ,
variante          smallint,
subvar            smallint,
fecha_hora        datetime YEAR TO SECOND ,
alr                smallint ,
obl_rfc            char(13),
obl_deter          smallint,

```

```

obl_clave          smallint,
obl_fecalt         date,
obl_fecbaj         date
end record,
rw_obli record like obligacion.*,
w_fec char(8),
x_fec,
w_fecau date
end globals

main
let x_alr=arg_val(1) # Recibe parametro No. ALR
'op/siralr/alr'
defer quit
defer interrupt

update casos set inicio = current where caso = 123
call caso_123()
update casos set fin = current where caso = 123
update casos set inicio = current where caso = 125
call caso_125()
update casos set fin = current where caso = 125

update casos set inicio = current where caso = 126
call caso_126()
update casos set fin = current where caso = 126

update casos set inicio = current where caso = 127
call caso_127()
update casos set fin = current where caso = 127
update casos set inicio = current where caso = 129
call caso_129()
update casos set fin = current where caso = 129
update casos set inicio = current where caso = 132
call caso_132()
update casos set fin = current where caso = 132

update casos set inicio = current where caso = 134
call Caso_134()
update casos set fin = current where caso = 134

end main

- Funcion
function Caso_134()
define bita16 record
num_caso          smallint ,
variante          smallint,
subvar            smallint,
fecha_hora        datetime YEAR TO SECOND ,
alr                smallint ,
obl_rfc            char(13),
obl_deter          smallint,
obl_clave          smallint,
obl_fecalt         date,
obl_fecbaj         date
end record

define casos char(400)

let casos =
'insert into bita_16
(num_caso,variante,subvar,fecha_hora,alr,obl_rfc,obl_deter,obl_c
lave,obl_fecbaj) ',
'select
134,1,0,current,'x_alr','obl_rfc,obl_deter,obl_clave,baj_fec',
'from obligacion o,bajas b',
'where o.obl_rfc = b.baj_rfc and o.obl_deter = b.baj_deter and
';

```

```

'o.obl_deter = 0 and obl_fecbaj is null '

prepare x_caso134 from casos
execute x_caso134
declare crs_bita16 cursor for
select *
into bita16.*
from bita_16
where num_caso=134
foreach crs_bita16
    update obligacion
    set obl_fecbaj=bita16.obl_fecbaj
    where obl_rfc=bita16.obl_rfc and
obl_deter=bita16.obl_deter
    and obl_clave=bita16.obl_clave
end foreach
close crs_bita16

update bita_16 set obl_fecbaj=null where num_caso=134

End Function

function caso_123()
define id float
define cont integer
define Obl1 record like obligacion.*

declare crs_obl1 cursor for
select *,rowid
into Obl1.*,id
from obligacion
where obl_deter is null
foreach crs_obl1
select *
from causante
where cau_rfc=Obl1.obl_rfc and cau_deter=0
if status<>notfound then
    let cont=0
    select count(*) into cont from obligacion
    where obl_rfc=Obl1.obl_rfc and obl_deter=0
    and obl_clave=Obl1.obl_clave
    if cont>0 then
        -- se marca para borrar
        --graba variante1
        insert into bita_16

        values(123,1,0,CURRENT,x_alr,Obl1.*)

        --borra el reg de obligacion
        delete from obligacion
        where rowid=id

    else
        --graba variante2
        insert into bita_16

        values(123,2,0,CURRENT,x_alr,Obl1.*)

        --actualiza el registro como matriz

con el rowid
        update obligacion
        set obl_deter=0
        where rowid=id

    end if
else
    -- se marca para borrar
    --graba variante3
    --graba variante4

    insert into bita_16

```

```

values(123,3,0,CURRENT,x_alr,Obl1.*)

--borra el reg de obligacion
delete from obligacion
where rowid=id

end if
end foreach
end function

function caso_125()
define strcaso125 char(300)
let strcaso125=
"insert into
bita_3(num_caso,variante,subvar,rfc,deter,fecha_hora,alr)",
" select 125,0,0,a.cau_rfc,a.cau_deter,CURRENT,"x_alr,
" from causante a",
" where cau_deter = 0 and not exists(select b.obl_rfc",
" from obligacion b where b.obl_rfc=a.cau_rfc",
" and b.obl_deter=a.cau_deter)"
prepare strexe from strcaso125
execute strexe

end function # fin de la funcion corrige status del contribuyente

function caso_126()
define strcaso126, strcaso126X char(350)
define casos char(450)
let strcaso126=
"insert into bita_16",
" select 126,0,0,CURRENT,"x_alr clipped ","a.",
" from obligacion a",
" where a.obl_clave is null and a.obl_deter = 0"
prepare strexe1 from strcaso126
execute strexe1

delete from obligacion
where obl_clave is null and obl_deter = 0

let strcaso126X="insert into bita_16",
" select 126,0,0,CURRENT,"x_alr clipped ","a.",
" from obligacion a",
" where a.obl_deter = 0 and not exists( select cve_oblig from
cat_oblig b",
" where b.cve_oblig=a.obl_clave)"
prepare strexe2 from strcaso126X
execute strexe2

delete from obligacion
where obl_deter = 0 and not exists(select cve_oblig from
cat_oblig b
where b.cve_oblig=obligacion.obl_clave)

select * from bita_16 where num_caso = 126 and variante = 0
and alr = x_alr
into temp tempbita16 with no log

let casos =
'insert into bita_16 (num_caso,variante,subvar,fecha_hora,alr,
'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj) ',
'select num_caso,1,0,fecha_hora,alr, ',
'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj ',
'from tempbita16 a where not exists(select b.obl_rfc from
obligacion b',
'where b.obl_rfc=a.obl_rfc and b.obl_deter=',
'a.obl_deter)'

```

```
prepare caso126 from casos
execute caso126
```

```
drop table tempbita16
end function
```

```
function caso_127()
define casos char(400)
define recb16 record
    num_caso          smallint ,
    variante          smallint,
    subvar            smallint,
    fecha_hora        datetime YEAR TO SECOND ,
    alr               smallint ,
    obl_rfc           char(13),
    obl_deter         smallint,
    obl_clave         smallint,
    obl_fecalt       date,
    obl_fecbaj       date
```

```
end record
define poner smallint
define conta float
let casos =
    'insert into bita_16 (num_caso,variante,subvar,fecha_hora,alr,
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj) ',
    'select 127,1,0,current',x_alr,
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj ',
    'from obligacion ',
    'where obl_deter = 0 and obl_rfc[1,1] = "" and ',
    'obl_clave in (select oblig_obs from cat_obl_obs ',
    'where oblig_equiv is null and persona = 1)
    prepare x_caso127bis from casos
    execute x_caso127bis
```

```
let casos =
    'insert into bita_16 (num_caso,variante,subvar,fecha_hora,alr,
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj) ',
    'select 127,1,0,current',x_alr,
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj ',
    'from obligacion ',
    'where obl_deter = 0 and obl_rfc[1,1] <> "" and ',
    'obl_clave in (select oblig_obs from cat_obl_obs ',
    'where oblig_equiv is null and persona = 2)
    prepare x_caso127bis2 from casos
    execute x_caso127bis2
```

```
delete from obligacion where obl_rfc[1,1] = ""
and obl_deter = 0 and obl_clave in (select oblig_obs from
cat_obl_obs
where oblig_equiv is null and persona = 1)
```

```
delete from obligacion where obl_rfc[1,1] <> ""
and obl_deter = 0 and obl_clave in (select oblig_obs from
cat_obl_obs
where oblig_equiv is null and persona = 2)
```

```
select * from bita_16 where num_caso = 127 and variante=1
and subvar = 0 and alr = x_alr
into temp tempbita 16 with no log
```

```
let casos =
    'insert into bita_16 (num_caso,variante,subvar,fecha_hora,alr,
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj) ',
    'select num_caso,variante,1,fecha_hora,alr, ',
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj ',
    'from tempbita 16 a where not exists(select b.obl_rfc from
obligacion b ',
```

```
'where b.obl_rfc=a.obl_rfc and b.obl_deter=a.obl_deter)
prepare caso127z from casos
execute caso127z
```

```
drop table tempbita16
```

```
let conta = 0
```

```
let casos =
    'insert into bita_16 (num_caso,variante,subvar,fecha_hora,alr,
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj) ',
    'select 127,2,0,current',x_alr,
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj ',
    'from obligacion ',
    'where obl_deter = 0 and obl_rfc[1,1] = "" and ',
    'obl_clave in (select oblig_obs from cat_obl_obs ',
    'where oblig_equiv is not null and persona = 1)
    prepare x_caso127a from casos
    execute x_caso127a
```

```
let casos =
    'insert into bita_16 (num_caso,variante,subvar,fecha_hora,alr,
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj) ',
    'select 127,2,0,current',x_alr,
    'obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj ',
    'from obligacion ',
    'where obl_deter = 0 and obl_rfc[1,1] <> "" and ',
    'obl_clave in (select oblig_obs from cat_obl_obs ',
    'where oblig_equiv is not null and persona = 2)
    prepare x_caso127abis from casos
    execute x_caso127abis
```

```
declare c127 cursor for
select bita_16.*,oblig_equiv
into recb16.*,poner
from bita_16,cat_obl_obs
where num_caso = 127 and variante = 2 and obl_clave =
oblig_obs
and obl_rfc[1,1] = "" and persona = 1
foreach c127
select distinct obl_rfc from obligacion
where obl_rfc = recb16.obl_rfc and obl_deter =
recb16.obl_deter and
obl_clave = poner
if status = NOTFOUND then
update obligacion set obl_clave = poner
where obl_rfc = recb16.obl_rfc and obl_deter =
recb16.obl_deter and
obl_clave = recb16.obl_clave
else
let recb16.variante=3
insert into bita_16 values (recb16.*)
delete from obligacion
where obl_rfc = recb16.obl_rfc and obl_deter =
recb16.obl_deter and
obl_clave = recb16.obl_clave
end if
end foreach
```

```
declare c127bis cursor for
select bita_16.*,oblig_equiv
into recb16.*,poner
from bita_16,cat_obl_obs
where num_caso = 127 and variante = 2 and obl_clave =
oblig_obs
and obl_rfc[1,1] <> "" and persona = 2
foreach c127bis
select distinct obl_rfc from obligacion
```

```

where obl_rfc = recb16.obl_rfc and obl_deter =
recb16.obl_deter and
  obl_clave = poner
if status = NOTFOUND then
  update obligacion set obl_clave = poner
  where obl_rfc = recb16.obl_rfc and obl_deter =
recb16.obl_deter and
  obl_clave = recb16.obl_clave
else
  let recb16.variante=3
  insert into bita_16 values (recb16.*)
  delete from obligacion
  where obl_rfc = recb16.obl_rfc and obl_deter =
recb16.obl_deter and
  obl_clave = recb16.obl_clave
end if
end foreach
End function

```

```

function caso_129()
define reccau record like causante.*
define cant integer

  declare c129 cursor for
  select *
  into reccau.*
  from causante
  where cau_deter = 0
foreach c129

if reccau.cau_rfc[1,1] <> "" then
  select count(*) into cant from obligacion o,cat_oblisc r
  where o.obl_clave = c.cve_oblig
  and cve_tiper = 2 and o.obl_rfc = reccau.cau_rfc and
o.obl_deter = 0
  if cant = 0 then
    insert into bita_3
    (num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
    values
    (129,0,0,reccau.cau_rfc,reccau.cau_deter,current,x_alr)
  end if
else
  select count(*) into cant from obligacion o,cat_oblisc r
  where o.obl_clave = c.cve_oblig
  and cve_tiper = 1 and o.obl_rfc = reccau.cau_rfc and
o.obl_deter = 0
  if cant = 0 then
    insert into bita_3
    (num_caso,variante,subvar,rfc,deter,fecha_hora,alr)
    values
    (129,0,0,reccau.cau_rfc,reccau.cau_deter,current,x_alr)
  end if
end foreach
end Function

```

```

function caso_132()
define strobli char(500)
define recobl record like obligacion.*

select obl_rfc,obl_deter,obl_clave,obl_fecalt,obl_fecbaj
from obligacion where obl_deter = 0 and obl_fecbaj is null
into TEMP tmpobli1 with no log

select * from tmpobli a
where exists( select * from obtgacion b

```

```

where b.obl_rfc=a.obl_rfc and b.obl_deter=a.obl_deter
and obl_deter = 0 and obl_fecbaj is null
and b.obl_fecalt > a.obl_fecalt and exists(
select * from cat_comp2 c where
(c.cve_obl=a.obl_clave and c.incompat=b.obl_clave) or
(c.cve_obl=b.obl_clave and c.incompat=a.obl_clave) ))
into TEMP tmpobli1 with no log

```

```

delete from tmpobli1
where obl_clave in (select cve_obl from cat_comp2
  where cve_calcont = 0)
and obl_rfc in (select cau_rfc from causante
  where cau_calcont <> 0)

```

```

let strobli="insert into bita_16",
* select 132,1,0,CURRENT,"x_alr",obl_rfc,obl_deter,obl_clave,*,
"obl_fecalt,obl_fecbaj from tmpobli1"
prepare xobli from strobli
execute xobli

```

```

declare c132 cursor for
select *
into recobl.*
from tmpobli1
foreach c132
  delete from obligacion
  where obl_rfc=recobl.obl_rfc and
  obl_deter=recobl.obl_deter and
  obl_clave=recobl.obl_clave
end foreach

```

```
drop table tmpobli1
```

```

select * from tmpobli a
where exists( select * from obligacion b
  where b.obl_rfc=a.obl_rfc and b.obl_deter=a.obl_deter
  and obl_deter = 0 and obl_fecbaj is null
  and b.obl_clave > a.obl_clave and exists(
  select * from cat_comp2 c where
  (c.cve_obl=a.obl_clave and c.incompat=b.obl_clave) or
  (c.cve_obl=b.obl_clave and c.incompat=a.obl_clave) ))
into TEMP tmpobli1 with no log

```

```

let strobli="insert into bita_16",
* select 132,2,0,CURRENT,"x_alr",obl_rfc,obl_deter,obl_clave,*,
"obl_fecalt,obl_fecbaj from tmpobli1"
prepare xobli9 from strobli
execute xobli9

```

```

declare c132a cursor for
select *
into recobl.*
from tmpobli1
foreach c132a
  delete from obligacion
  where obl_rfc=recobl.obl_rfc and
  obl_deter=recobl.obl_deter and
  obl_clave=recobl.obl_clave
end foreach

```

```
drop table tmpobli1
drop table tmpobli
end function

```

LIMPLIQ.4GL

```

database shcp
globals
  define x_alr char(2)
  define rw_l record like liquidacion.*
  define rw_suc record like sucesion.*
  define w_x1 smallint

```

```

define id float
end globals
main

-- obtiene la ALR al ejecutar estcau.4ge 'u/siralr/alr' --
let x_alr=arg_val(1)
defer quit
defer interrupt
update casos set inicio = current where caso = 60
Call Caso_60()
update casos set fin = current where caso = 60

update casos set inicio = current where caso = 65
Call Caso_65_REG()
update casos set fin = current where caso = 65

end main

-----
function caso_60() ##sustituir por function

declare c1 cursor for
select *,rowid into rw_1., id
from liquidacion
where liquidacion.liq_rfc[1,1] <> ''

foreach c1
select * into rw_suc.* from sucesion
where sucesion.suc_rfc = rw_1.liq_rfc
if status = notfound then
insert into sucesion values(rw_1.liq_rfc, rw_1.liq_fechi,
rw_1.liq_nomliq,rw_1.liq_rfcliq,rw_1.liq_domliq)
let w_x1 = 1 ##causante no existe en
sucesion
else
let w_x1 = 2 ##causante existe en las dos
end if
insert into bita_9 values(60, w_x1,0,current,x_alr,null,
rw_1.liq_rfc,rw_1.liq_rfc,rw_1.liq_fechi,rw_1.liq_nomliq,
rw_1.liq_rfcliq,rw_1.liq_domliq,rw_1.liq_telliq,x_alr)

delete from liquidacion where rowid = id
end foreach
end function

-----
-- Funcion liq_rfcliq
function Caso_65_REG()
define cau0 record like liquidacion.*,id float
define nombre char(80),domicilio char(80),telefono char (10)
define liqrec record
num_caso smallint ,
variante smallint,
subvar smallint,
fecha_hora datetime YEAR to SECOND .
alr . smallint ,
status char(1),
rfc char(13),
liq_rfc char(13),
liq_fechi date,
liq_nomliq varchar(80),
liq_rfcliq char(13),
liq_domliq varchar(80),
liq_telliq varchar(10),
alr_liq smallint

end record
define liqrec2 record
num_caso smallint,
variante smallint,
subvar smallint,
alr smallint,

```

```

rfc char(13),
deter smallint,
ofh char(5),
alr smallint

end record
declare c65 cursor for
select *,rowid into cau0.*,id from liquidacion
foreach c65
initialize liqrec.* to NULL
initialize liqrec2.* to NULL

if length(cau0.liq_rfcliq) >= 13 then

If cau0.liq_rfc = cau0.liq_rfcliq then
let liqrec.num_caso=65
let liqrec.variante=1
let liqrec.subvar=0
let liqrec.fecha_hora=current
let liqrec.alr=x_alr
let liqrec.liq_rfc=cau0.liq_rfc
let liqrec.liq_nomliq=cau0.liq_nomliq
let liqrec.liq_rfcliq=cau0.liq_rfcliq
let liqrec.liq_domliq=cau0.liq_domliq
let liqrec.liq_telliq=cau0.liq_telliq
insert into bita_9 values (liqrec.*)
update liquidacion set liq_nomliq = ",liq_rfcliq = ",
liq_domliq = ",liq_telliq = " where rowid=id

Else
Select cau_rsoc,cau_domi,cau_telefo into
nombre,domicilio,telefono
from causante where cau_rfc = cau0.liq_rfcliq
and cau_deter = 0

If Status <> notfound then
let liqrec.num_caso=65
let liqrec.variante=2
let liqrec.subvar=0
let liqrec.fecha_hora=current
let liqrec.alr=x_alr
let liqrec.liq_rfc=cau0.liq_rfc
let liqrec.liq_nomliq=cau0.liq_nomliq
let liqrec.liq_rfcliq=cau0.liq_rfcliq
let liqrec.liq_domliq=cau0.liq_domliq
let liqrec.liq_telliq=cau0.liq_telliq
insert into bita_9 values (liqrec.*)
update liquidacion set liq_nomliq = nombre,
liq_domliq =
domicilio,
liq_telliq = telefono where rowid=id

Else
let liqrec.num_caso=65
let liqrec.variante=3
let liqrec.subvar=0
let liqrec.fecha_hora=current
let liqrec.alr=x_alr
let liqrec.liq_rfc=cau0.liq_rfc
let liqrec.liq_nomliq=cau0.liq_nomliq
let liqrec.liq_rfcliq=cau0.liq_rfcliq
let liqrec.liq_domliq=cau0.liq_domliq
let liqrec.liq_telliq=cau0.liq_telliq
insert into bita_9 values (liqrec.*)

let liqrec2.num_caso=65
let liqrec2.variante=3
let liqrec2.subvar=0
let liqrec2.alr=x_alr
let liqrec2.rfc=cau0.liq_rfcliq
let liqrec2.deter=0
insert into comp_casos values (liqrec2.*)

End If
End If

end if

```

```

end foreach
end function

```

LIMPDYE.4GL

```

database shcp

```

```

globals

```

```

define x_alr char(2)

```

```

define doc0 record like documentos.*

```

```

define ext0 record like externos.*

```

```

end globals

```

```

main

```

```

let x_alr= arg_val(1)

```

```

defer quit

```

```

defer interrupt

```

```

update casos set inicio = current where caso = 104

```

```

call caso_104()

```

```

update casos set fin = current where caso = 104

```

```

update casos set inicio = current where caso = 105

```

```

call caso_105()

```

```

update casos set fin = current where caso = 105

```

```

--update casos set inicio = current where caso = 112 or caso =
113

```

```

- call caso_112_113()

```

```

--update casos set fin = current where caso = 112 or caso = 113

```

```

update casos set inicio = current where caso = 118

```

```

call caso_118()

```

```

update casos set fin = current where caso = 118

```

```

update casos set inicio = current where caso = 119

```

```

call caso_119()

```

```

update casos set fin = current where caso = 119

```

```

update casos set inicio = current where caso in (120,121)

```

```

call caso_120_121()

```

```

update casos set fin = current where caso in (120,121)

```

```

update casos set inicio = current where caso =110

```

```

call caso110()

```

```

update casos set fin = current where caso =110

```

```

end main

```

```

-----caso 104

```

```

function caso_104()

```

```

define id float, c_equiv,x_temp char(5),x_tres char(3)

```

```

define strcaso char(400),x_long smallint

```

```

define doctos record like documentos.*

```

```

declare c104 cursor for

```

```

select a.*,a.rowid from documentos a

```

```

foreach c104 into doctos.*,id

```

```

--- Probando si lo encontro

```

```

let x_long=length(doctos.doc_nofh)

```

```

if x_long<=3 then

```

```

--- Si es menor de 3 posiciones

```

```

insert into bita_17(num_caso,variante,subvar,fecha_hora,

```

```

alr,rfc,deter,no_oper) values (104,4,0,CURRENT,x_alr,

```

```

doctos.doc_rfc,doctos.doc_deter,doctos.doc_nope)

```

```

let x_temp='00000'

```

```

let doctos.doc_nofh=doctos.doc_nofh clipped,x_temp[1,5-

```

```

x_long]

```

```

----- Busco duplicidad

```

```

select doc_nofh from documentos where

```

```

doc_rfc=doctos.doc_rfc and

```

```

doc_deter=doctos.doc_deter and

```

```

doc_nofh=doctos.doc_nofh and

```

```

doc_fechop=doctos.doc_fechop and

```

```

doc_estcap=doctos.doc_estcap and

```

```

doc_nope=doctos.doc_nope

```

```

if status <> notfound then

```

```

-----Si lo encontro duplicidad

```

```

insert into bita_14

```

```

values (104,4,1,CURRENT,x_alr,doctos.*)

```

```

delete from documentos where rowid=id

```

```

continue foreach

```

```

else

```

```

let x_tres=doctos.doc_nofh[1,3]

```

```

select cve_ofh from cat_ofh b

```

```

where b.cve_ofh=x_tres

```

```

if status <> notfound then

```

```

----- Ya no esta fuera de catalogo

```

```

insert into

```

```

bita_17(num_caso,variante,subvar,fecha_hora,

```

```

alr,rfc,deter,no_oper,crh)

```

```

values (104,4,2,CURRENT,x_alr,doctos.doc_rfc,

```

```

doctos.doc_deter,doctos.doc_nope,doctos.doc_nofh)

```

```

update documentos set doc_nofh=doctos.doc_nofh

```

```

where rowid=id

```

```

continue foreach

```

```

end if

```

```

end if

```

```

end if

```

```

let x_tres=doctos.doc_nofh[1,3]

```

```

select cve_ofh from cat_ofh b where b.cve_ofh=x_tres

```

```

if status = notfound then

```

```

select crh_equiv into c_equiv from cat_eqcrh

```

```

where crh_ante= doctos.doc_nofh

```

```

if status <> notfound then

```

```

select doc_rfc from documentos where

```

```

doc_rfc=doctos.doc_rfc and doc_deter=doctos.doc_deter

```

```

and doc_nofh=c_equiv and

```

```

doc_fechop=doctos.doc_fechop

```

```

and doc_estcap=doctos.doc_estcap and

```

```

doc_nope=doctos.doc_nope

```

```

if status <> notfound then

```

```

insert into bita_14

```

```

values (104,2,0,CURRENT,x_alr,doctos.*)

```

```

delete from documentos where rowid=id

```

```

else

```

```

insert into

```

```

bita_17(num_caso,variante,subvar,fecha_hora,

```

```

alr,rfc,deter,no_oper,crh)

```

```

values (104,1,0,CURRENT,x_alr,doctos.doc_rfc,

```

```

doctos.doc_deter,doctos.doc_nope,doctos.doc_nofh)

```

```

update documentos set doc_nofh=c_equiv where

```

```

rowid=id

```

```

end if

```

```

else

```

```

select doc_nofh from documentos where

```

```

doc_rfc=doctos.doc_rfc and

```

```

doc_deter=doctos.doc_deter and doc_nofh='99900' and

```

```

doc_fechop=doctos.doc_fechop and

```

```

doc_estcap=doctos.doc_estcap and

```

```

doc_nope=doctos.doc_nope

```

```

if status = notfound then

```

```

insert into

```

```

bita_17(num_caso,variante,subvar,fecha_hora,

```

```

alr,rfc,deter,no_oper,crh)

```

```

values (104,3,0,CURRENT,x_alr,doctos.doc_rfc,

```

```

doctos.doc_deter,doctos.doc_nope,doctos.doc_nofh)

```

```

update documentos set doc_nofh='99900' where

```

```

rowid=id

```

```

else

```

```

----- Encontro duplicidad

```

```

insert into bita_14

```

```

values (104,3,1,CURRENT,x_alr,doctos.*)

```

```

delete from documentos where rowid=id

```

```

end if

```

```

end if

```

```

end foreach

```

```

end function #104

```

```

-----caso 105

```

```

function caso_105()

```

```

define id float, c_equiv_x_temp char(5), x_tres char(3)
define strcaso char(400), x_long smallint
define r_exter record like externos.*
declare c105 cursor for
  select a.*, a.rowid from externos a
foreach c105 into r_exter.*id
  --- Probando si lo encontro
  let x_long=length(r_exter.ext_nofh)
  if x_long<=3 then
    --- Si es menor de 3 posiciones
    insert into bita_17(num_caso,variante,subvar,fecha_hora,
alr,rfc,deter,no_opero) values (105,4,0,CURRENT,x_alr,
r_exter.ext_rfc,r_exter.ext_deter,r_exter.ext_nope)
let x_temp='00000'
let r_exter.ext_nofh =r_exter.ext_nofh clipped,x_temp[1,5-
x_long]
    ----- Busco duplicidad
    select ext_nofh from externos where
ext_rfc=r_exter.ext_rfc and
ext_deter=r_exter.ext_deter and
ext_nofh=r_exter.ext_nofh and
ext_fechop=r_exter.ext_fechop and
ext_estcap=r_exter.ext_estcap and
ext_nope=r_exter.ext_nope
if status <> notfound then
    -----Si lo encontro duplicidad
    insert into bita_15
values (105,4,1,CURRENT,x_alr,r_exter.*)
delete from externos where rowid=id
continue foreach
else
  let x_tres=r_exter.ext_nofh[1,3]
  select cve_ofh from cat_ofh b
  where b.cve_ofh=x_tres
  if status <> notfound then
    ----- Ya no esta fuera de catalogo
    insert into
bita_17(num_caso,variante,subvar,fecha_hora,
alr,rfc,deter,no_opero,crh)
values (105,4,2,CURRENT,x_alr,r_exter.ext_rfc,
r_exter.ext_deter,r_exter.ext_nope,r_exter.ext_nofh)
update externos set ext_nofh=r_exter.ext_nofh where
rowid=id
    continue foreach
  end if
end if
end if
let x_tres=r_exter.ext_nofh[1,3]
select cve_ofh from cat_ofh b where b.cve_ofh=x_tres
if status = notfound then
  select crh_equiv into c_equiv from cat_eqcrh
  where crh_ante= r_exter.ext_nofh
  if status <> notfound then
    select ext_rfc from externos where
ext_rfc=r_exter.ext_rfc and ext_deter=r_exter.ext_deter
and ext_nofh=c_equiv and
ext_fechop=r_exter.ext_fechop
and ext_estcap=r_exter.ext_estcap and
ext_nope=r_exter.ext_nope
if status <> notfound then
  insert into bita_15
values (105,2,0,CURRENT,x_alr,r_exter.*)
delete from externos where rowid=id
else
  insert into
bita_17(num_caso,variante,subvar,fecha_hora,
alr,rfc,deter,no_opero,crh)
values (105,1,0,CURRENT,x_alr,r_exter.ext_rfc,
r_exter.ext_deter,r_exter.ext_nope,r_exter.ext_nofh)
update externos set ext_nofh=c_equiv where rowid=id
end if

```

```

else
  select ext_nofh from externos where
ext_rfc=r_exter.ext_rfc and
ext_deter=r_exter.ext_deter and ext_nofh='99900' and
ext_fechop=r_exter.ext_fechop and
ext_estcap=r_exter.ext_estcap
and ext_nope=r_exter.ext_nope
if status = notfound then
  insert into
bita_17(num_caso,variante,subvar,fecha_hora,
alr,rfc,deter,no_opero,crh)
values (105,3,0,CURRENT,x_alr,r_exter.ext_rfc,
r_exter.ext_deter,r_exter.ext_nope,r_exter.ext_nofh)
update externos set ext_nofh='99900' where rowid=id
else
  insert into bita_15
values (105,3,1,CURRENT,x_alr,r_exter.*)
delete from externos where rowid=id
end if
end if
end foreach
end function #105

```

Function caso_112_113()

```

define casos char(400)
let casos =
'insert into bita_17 (rfc,deter,fecha_hora,alr,
num_caso,variante,subvar,no_opero) ',
'select doc_rfc,doc_deter,current,'x_alr','112,0,0,doc_nope ',
'from documentos ',
'where doc_claope IS NULL or ',
'doc_claope not in (select usu_clave from usuarios)'

prepare x_caso112 from casos
execute x_caso112

```

```

update documentos set doc_claope = 99
where doc_claope IS NULL or
doc_claope not in (select usu_clave from usuarios)

```

```

let casos =
'insert into bita_17 (rfc,deter,fecha_hora,alr,
num_caso,variante,subvar,no_opero) ',
'select ext_rfc,ext_deter,current,'x_alr','113,0,0,ext_nope ',
'from externos ',
'where ext_claope IS NULL or ',
'ext_claope not in (select usu_clave from usuarios)'

```

```

prepare x_caso113 from casos
execute x_caso113

```

```

update externos set ext_claope = 99
where ext_claope IS NULL or
ext_claope not in (select usu_clave from usuarios)

```

end function

```

function caso_118()
define x_doctos record like documentos.*
id float,x_motivo char(2)
declare c118 cursor for
select *,rowid into x_doctos.*,id
from documentos where doc_motmov is null
foreach c118
let x_motivo=' '
select doc_motmov into x_motivo from docnal
where doc_rfc=x_doctos.doc_rfc and
doc_deter=x_doctos.doc_deter

```

```

        and doc_nofh=x_doctos.doc_nofh and
doc_nope=x_doctos.doc_nope
        and doc_motmov is not null
        if status <> notfound then
            insert into bita_17
values(118,0,0,current,x_alr,x_doctos.doc_rfc,
        x_doctos.doc_deter,x_doctos.doc_estcap,
x_doctos.doc_nope,"",x_doctos.doc_motmov)
        update documentos set doc_motmov = x_motivo where
rowid=id
        else
            insert into bita_17
values(118,1,0,current,x_alr,x_doctos.doc_rfc,
        x_doctos.doc_deter,x_doctos.doc_estcap,
x_doctos.doc_nope,"",x_doctos.doc_motmov)
        update documentos set doc_motmov = 'ZZ' where
rowid=id
        end if
    end foreach
end function # caso 118
function caso_119()
    define x_ext record like externos.*
    id float,x_motivo char(2)
    declare c119 cursor for
        select *,rowid into x_ext.*,id
        from externos where ext_motmov is null
    foreach c119
        let x_motivo='
        select ext_motmov into x_motivo from extnal
        where ext_rfc=x_ext.ext_rfc and
ext_deter=x_ext.ext_deter
        and ext_nofh=x_ext.ext_nofh and
ext_nope=x_ext.ext_nope
        and ext_motmov is not null
        if status <> notfound then
            insert into bita_17
values(119,0,0,current,x_alr,x_ext.ext_rfc,
        x_ext.ext_deter,x_ext.ext_estcap,
x_ext.ext_nope,"",x_ext.ext_motmov)
        update externos set ext_motmov = motmov where
rowid=id
        else
            insert into bita_17
values(119,1,0,current,x_alr,x_ext.ext_rfc,
        x_ext.ext_deter,x_ext.ext_estcap,
x_ext.ext_nope,"",x_ext.ext_motmov)
        update externos set ext_motmov = 'ZZ' where rowid=id
        end if
    end foreach
end function # caso 119
function caso_120_121()
    define casos char(600)

    let casos = 'insert into bita_17 (rfc,deter,fecha_hora,alr,
        'num_caso,variante,subvar,no_opera,crh,motmov)',
        'select doc_rfc,doc_deter,current,'x_alr,
        ',120,0,0,doc_nope,doc_nofh,doc_motmov from documentos ',
        'where doc_motmov not in (select cve_mov from cat_mov)'
    prepare x_caso120 from casos
    execute x_caso120

    update documentos set doc_motmov = 'ZZ'
    where doc_motmov not in (select cve_mov from cat_mov)

    let casos = 'insert into bita_17 (rfc,deter,fecha_hora,alr,
        'num_caso,variante,subvar,no_opera,crh,motmov)',
        'select ext_rfc,ext_deter,current,'x_alr,
        ',121,0,0,ext_nope,ext_nofh,ext_motmov from externos ',
        'where ext_motmov not in (select cve_mov from cat_mov)'
    prepare x_caso121 from casos

```

```

execute x_caso121

update externos set ext_motmov = 'ZZ'
where ext_motmov not in (select cve_mov from cat_mov)

end function

function caso110()

define rec_doc record like documentos.*,rec_ext record like
externos.*
define y_alr smallint, x_crh char(3)

select documentos.*,externos.*
from documentos , externos
where doc_rfc=ext_rfc and doc_deter=ext_deter and
doc_nofh=ext_nofh
and doc_fechop=ext_fechop and doc_nope=ext_nope
into TEMP docext with no log

declare crs_ext cursor for
select *
into rec_doc.*,rec_ext.*
from docext

foreach crs_ext
    let x_crh=rec_doc.doc_nofh[1,3]
    select edo into y_alr
    from cat_ofh
    where cve_ofh=x_crh
    if status<>notfound then
        if y_alr=x_alr then
            insert into bita_15
values(110,1,0,CURRENT,x_alr,rec_ext.*)
            delete from externos
            where ext_rfc=rec_ext.ext_rfc and
ext_deter=rec_ext.ext_deter
            and ext_nofh=rec_ext.ext_nofh and
ext_fechop=rec_ext.ext_fechop
            and ext_estcap=rec_ext.ext_estcap and
ext_nope=rec_ext.ext_nope
        else
            insert into bita_14
values(110,2,0,CURRENT,x_alr,rec_doc.*)
            delete from documentos
            where doc_rfc=rec_doc.doc_rfc and
doc_deter=rec_doc.doc_deter
            and doc_nofh=rec_doc.doc_nofh and
doc_fechop=rec_doc.doc_fechop
            and doc_estcap=rec_doc.doc_estcap and
doc_nope=rec_doc.doc_nope
        end if
        --checa contra doctos y externos
    else

--variante 3
        insert into bita_14
values(110,3,0,CURRENT,x_alr,rec_doc.*)

        delete from documentos
        where doc_rfc=rec_doc.doc_rfc and
doc_deter=rec_doc.doc_deter
        and doc_nofh=rec_doc.doc_nofh and
doc_fechop=rec_doc.doc_fechop
        and doc_estcap=rec_doc.doc_estcap and
doc_nope=rec_doc.doc_nope

    end if

end foreach
close crs_ext

```



```
drop table docext
```

```
end function
```

LIMPCED.4GL

```
database shcp
globals
```

```
define x_alr char(2), hora char(30), RfcAnt char(13), activo
smallint
```

```
end globals
```

```
main
```

```
— obtiene la ALR al ejecutar estcau.4ge '/u/siralr/alr' —
```

```
let x_alr= arg_val(1)
```

```
defer quit
```

```
defer interrupt
```

```
update casos set inicio = current where caso = 142
```

```
call caso_142()
```

```
update casos set fin = current where caso = 142
```

```
end main
```

```
function caso_142()
```

```
define query char(400)
```

```
let query = 'insert into bita_13 '
```

```
'select 142,1,0,current,'x_alr','*'
```

```
'from cedulas '
```

```
'where ced_fecbaj is not null and (ced_mot <> "C" or
```

```
ced_mot is null)'
```

```
prepare cur142 from query
```

```
execute cur142
```

```
update cedulas
```

```
set ced_mot = 'C'
```

```
where ced_fecbaj is not null and (ced_mot <> "C" or ced_mot is
null)
```

```
let query = 'insert into bita_13 '
```

```
'select 142,2,0,current,'x_alr','*'
```

```
'from cedulas '
```

```
'where ced_fecbaj is null and '
```

```
'(ced_mot not in ("N","R","A") or ced_mot is null)'
```

```
prepare cur142a from query
```

```
execute cur142a
```

```
update cedulas
```

```
set ced_mot = 'R'
```

```
where ced_fecbaj is null and (ced_mot not in ("N","R","A") or
ced_mot is null)
```

```
end function
```

LIMPCAU.4GL

```
database shcp
```

```
globals
```

```
define id,conta float,
```

```
stat_cau_alr_1,x_svar,x_svar2,x_svar3,x_svar4,x_svar5,x_svar6,
```

```
x_svar7 smallint,
```

```
x_num integer,s_ana smallint,
```

```
s_char,x_bajmov,x_alr,x_caso char(2)
```

```
define f_baj,f_liq,f_suce,f_sus,f_nom,f_mayor date,
```

```
x_numnon,stat_ori char(1),
```

```
x_vari,x_vari2,x_vari3 char(10)
```

```
define cau0 record like causante.'
```

```
define bit11 record
```

```
num_caso smallint ,
```

```
variante smallint,
```

```
subvar smallint,
```

```
rfc char(13) ,
```

```
deter smallint,
```

```
fecha_hora datetime YEAR TO SECOND ,
```

```
alr smallint ,
```

```
cau_motalt char(2),
```

```
cau_calcont smallint,
```

```
cau_tipsoc smallint,
```

```
cau_tipcont char(1),
```

```
cau_segobob integer,
```

```
cau_cveact char(6),
```

```
cau_movs char(1),
```

```
baj_cvebaj char(2),
```

```
calcont_nal smallint,
```

```
tipcont_nal char(1)
```

```
end record
```

```
define w_doc, w_ext, w_obliga smallint, w_fecfis char(8),
```

```
w_ini,w_fin char(25), hora char(30),y_stat varchar(2),
```

```
r_stat integer,q_mayor integer,g_mov char(2)
```

```
define g_afs smallint,g_ofh char(5),g_fechop date, g_nope
integer,
```

```
x_rfcno char(13),c_num char(7),nulos
```

```
char(1),x_reactiva char(1),
```

```
numERICA integer
```

```
end globals
```

```
main
```

```
lock table causante in exclusive mode
```

```
lock table documentos in exclusive mode
```

```
lock table externos in exclusive mode
```

```
lock table nombres in exclusive mode
```

```
lock table bajas in exclusive mode
```

```
lock table sucesion in exclusive mode
```

```
lock table suspension in exclusive mode
```

```
lock table liquidacion in exclusive mode
```

```
select max(cau_rfc[5,13]+0)/100 into numerica from causante
```

```
where cau_rfc[1,4]='XXXX'
```

```
if numerica is not null then
```

```
let x_num=numERICA
```

```
else
```

```
let x_num=0
```

```
end if
```

```
— obtiene la ALR al ejecutar estcau.4ge '/u/siralr/alr'
```

```
let x_alr= arg_val(1)
```

```
defer quit
```

```
defer interrupt
```

```
declare c_ofh scroll cursor for
```

```
select cve_ofh,afs
```

```
from cat_ofh where edo=x_alr
```

```
open c_ofh
```

```
fetch first c_ofh into g_ofh,g_afs
```

```
close c_ofh
```

```
let g_ofh=g_ofh clipped,'00'
```

```
update casos set inicio = current where caso = 32
```

```
call caso_32()
```

```
update casos set fin = current where caso = 32
```

```
update casos set inicio = current where caso = 33
```

```
call caso_33()
```

```
update casos set fin = current where caso = 33
```

```
— corrige status del contribuyente
```

```
update casos set inicio = current where caso in (4,5,7,16,
```

```
27,25,26,29,30,31,34,35,36,37,
```

```
41,43,45,47,48)
```

```
call cau_stat()
```

```
update casos set fin = current where caso in (4,5,7,16,
```

```
27,25,26,29,30,31,34,35,36,37,
```

```
41,43,45,47,48)
```

```

unlock table causante
unlock table documentos
unlock table externos
unlock table nombres
unlock table bajas
unlock table sucesion
unlock table suspension
unlock table liquidacion
update casos set inicio = current where caso = 133
call caso_133()
update casos set fin = current where caso = 133
end main
--          funcion corrige estatus del contribuyente
function cau_stat()
declare c1 cursor for
select *,rowid
into cau0.*,id
from causante where cau_rfc[1,4]<>'XXXX'
let conta=0
foreach c1
let w_obliga=0
let nulos='N'
let x_reactiva='N'
let stat_ori=cau0.cau_movs
--Checa la longitud del RFC y digito verificador caso
4 y 5--
if cau0.cau_deter=0 then
call rfcver()
end if
call caso_7()
-- Causantes sin fecha de alta --
if cau0.cau_fecalt is null then
call caso_16()
end if
-- Causante matriz con alta 10 caso 27 --
if cau0.cau_deter=0 and cau0.cau_motalt = 10 then
call caso_27()
end if
-- Causantes sin mitivo de alta caso 25 --
call caso_25()
-- Causante con motivo fuera de catalogo caso 26 --
call caso_26()
-- Causantes sin calidad caso 29 --
-- Causantes con calidad <0 y mayor 10
caso_30 --
-- Causantes con grupo economico fuera de catalogo
caso_31 --
call caso_29_30_31()
-- Causantes sin tipo de sociedad --
-- Causantes con tipo <0 y <= 5 --
if cau0.cau_deter=0 then
call caso_34_35()

```

```

end if
-- Causantes sin tipo caso(37)
-- Causantes con tipo fuera de catalogo caso 36
call caso_37_36()
-- Causantes si actividad preponderante caso 41
call caso_41()
-- causantes con actividad fuera del catalogo caso 43
call caso_43()
-- Causantes sin estus
if cau0.cau_movs is null then
call caso_47()
end if
-- Causantes con fuera de catalogo
call caso_48()
-- Comienza caso 45
let stat_ori=cau0.cau_movs
if cau0.cau_deter=0 then
let s_ana=0
let x_caso='45'
let x_numnon='N'
initialize f_baj_f_liq,f_sus,f_suce,f_nom,x_vari to null
let stat_cau=0
let q_mayor=0
---- Busca en bajas ----
select baj_fec,baj_cvebaj into f_baj,x_bajmov from bajas
where baj_rfc=cau0.cau_rfc and
baj_deter=cau0.cau_deter
if status<> notfound then
let stat_cau=stat_cau+1
else
let x_bajmov=""
end if
---- Busca en liquidacion ----
select liq_fechi into f_liq from liquidacion
where liq_rfc=cau0.cau_rfc
if status<> notfound then
let stat_cau=stat_cau+2
end if
-- Busca en sucesion ----
select suc_fechi into f_suce from sucesion
where suc_rfc=cau0.cau_rfc
if status<> notfound then
let stat_cau=stat_cau+4
end if
-- Busca en suspension ----
select sus_fecsus into f_sus from suspension
where sus_rfc=cau0.cau_rfc
and sus_deter=cau0.cau_deter
if status<> notfound then
let stat_cau=stat_cau+8
end if
-- Busca en nombres ----
select nom_fecha into f_nom from nombres
where nom_rfc=cau0.cau_rfc
if status<> notfound then

```

```

let stat_cau=stat_cau+16
end if
let r_stat=stat_cau
let y_stat=stat_cau
case
- Contribuyente en OK
when cau0.cau_movs=' ' and stat_cau=0
let conta=0
--ok
- Si el contribuyente no esta activo
asigna el estatus encontrado
when cau0.cau_movs<>' ' and (y_stat matches
*[1248]*
or stat_cau=17)
-----
- variante .. 45,1,0 -
-----
let x_vari='1'
let x_svar=0
let x_svar2=1
let x_svar3=2
call map_stat() #mapea el estatus
- Si el contribuyente esta activo, pero
- existe informacin en
bajas,liquidacion
- sucesion o suspension, se busca una
- reactivacion
when cau0.cau_movs=' ' and (y_stat matches
*[1248]*
or stat_cau=17)
if busca()=N' then
-----
- variante .. 45,3,0 -
-----
let x_vari='3'
let x_svar=0
let x_svar2=1
let x_svar3=2
call map_stat() # de no existir se
mapea el status
else
-----
- variante .. 2 -
-----
let x_vari='2'
insert into bita_11 values(45,2,0,cau0.cau_RFC,
0,current,x_alr,null,null,null,null,null,
stat_ori,null,null,null)
call borraseg() # Si existe se deja activo
y se borra # el segmento.
end if
- Esta en mas de un segmento o en la
- tabla de nombres sin registro en
bajas
when cau0.cau_movs<>' ' and (y_stat not matches
*[01248]*
and stat_cau<>17)
let s_ana=1
call anatipo()
When cau0.cau_movs<>' ' and stat_cau=0
-----
- Grabar en archivos que van a ir a la
nacional
- para su correccion. V A R I A N T E ..... 8
-----
insert into bita_11
values(45,8,0,cau0.cau_RFC,0,current,

```

```

x_alr,null,null,null,null,null,cau0.cau_movs,null,
null,null)
-----
--Se activa al causante, porque ya se
verificaron los ---
-- segmentos ---
-----
update causante set cau_movs=' ' where rowid=id
let cau0.cau_movs=' '
when cau0.cau_movs=' ' and (y_stat not matches
*[01248]*
and stat_cau<>17)
let f_mayor = null
let q_mayor=0
if busca()=N' then
-----
- Variante ... 45,12,0 -
-----
let s_ana=2
call anatipo()
else
-----
- Variante ... 11 --
-----
insert into bita_11 values(45,11,0,cau0.cau_RFC,
0,current,x_alr,null,null,null,null,null,
null,null,null,null)
let x_vari='11'
let x_svar2=1
let x_bajmov='00'
let x_reactiva='S'
call bor_segnon(16) # borra los demas
segmentos call bor_segnon(8) # borra los demas
segmentos call bor_segnon(4) # borra los demas
segmentos call bor_segnon(2) # borra los demas
segmentos call bor_segnon(1) # borra los demas
segmentos
let x_reactiva='N'
end if
end case
end if # deter=0 caso 45
end foreach
close c1
end function # fin de la funcion corrige status del contribuyente
----- mapea estatus
function map_stat()
define y_inst char(400)
let y_stat=stat_cau
----- verifica la baja por cambio de nombre
case
when stat_cau=17 and (x_bajmov<>'14' and
x_bajmov<>'33')
- Guarda el contenido del registro de la tabla de bajas
let y_inst=insert into bita_6 select 45,'x_vari
clipped,
',x_svar2,'current',x_alr,'*' from bajas where baj_rfc="",
cau0.cau_rfc," and baj_deter=",cau0.cau_deter
prepare a_baj from y_inst
execute a_baj
- cambia la la cve de la baja

```

```

update bajas set (baj_cvebaj,baj_fec)=(‘33’,f_nom) where
baj_rfc=cau0.cau_rfc
and baj_deter=cau0.cau_deter
-- Guarda los datos que tenia la obligacion
let y_inst='insert into bita_16 select 45,x_vari
clipped,
',x_svar3,current,x_alr,' from obligacion where
obl_rfc=',
cau0.cau_rfc,'" and obl_deter=',cau0.cau_deter,' and
obl_fecbaj=',
f_baj,'"
prepare a_obl from y_inst
execute a_obl
update obligacion set obl_fecbaj=f_norn where
obl_rfc=cau0.cau_rfc
and obl_deter=cau0.cau_deter and obl_fecbaj =f_baj
when stat_cau=1 and (x_bajmov='14' or x_bajmov='33')
let x_numnon='S'
end case
case
when stat_cau=1 and y_stat matches "[12]"
call estabaj()
when stat_cau=2
let cau0.cau_movs='5'
when stat_cau=4
let cau0.cau_movs='4'
when stat_cau=8
let cau0.cau_movs='3'
when stat_cau=17
let cau0.cau_movs='2'
end case
call asig_stat()
end function
----- Establece si se asigna el nuevo status
function asig_stat()
if stat_ori <> cau0.cau_movs then
-- Almacena es status del causante original a bita 11
insert into bita_11
values(45,x_vari,x_svar,cau0.cau_RFC,
0,current,x_alr,null,null,null,null,null,stat_ori,null,
null,null)
-- Actualiza el casuante con el estatus
update causante set cau_movs=cau0.cau_movs where
rowid=id
end if
end function # signa status
-- Establece el tipo de baja
function estabaj()
define
r_id float ,
x_inst char(100),
x_rfcnvo char(13),
c_num char(7)
if x_bajmov <> '06' and x_bajmov <> '50' then # **corregido** se
incluyo 50
let cau0.cau_movs='2'
else
let cau0.cau_movs='1'
end if
if x_numnon='S' and (x_bajmov='14' or x_bajmov='33') then
if (stat_cau-16) < 0 then
-----
variante 6 del 45
-----
insert into bita_11 values(45,6,0,cau0.cau_RFC,0,current,
x_alr,null,null,null,null,null,null,cau0.cau_movs,null,null,null)
let x_num=x_num+1
let c_num=x_num
let x_rfcnvo='XXXXX',c_num clipped,x_alr clipped
insert into nombres values(cau0.cau_rfc,x_rfcnvo,f_baj)

```

```

insert into causante values (x_rfcnvo,0,
'CAUSANTE GENERADO POR LA
LIMPIEZA',' ','00','g_afc','g_oh,
f_baj','00',0,0,'8',0,'999999',x_alr,'00','0000000',0,')
end if
end if
end function # estabaj

function busca()
define x_mov char(2) ,
x_inst char(200),
p_mot char(50),
fec date,
x_tantos,x_hay float,
xenco char(1),
xope integer

initialize fec,x_tantos to null
case
when stat_cau=1
let fec=f_baj
when stat_cau=2
let fec=f_liq
when stat_cau=4
let fec=f_suce
when stat_cau=8
let fec=f_sus
when stat_cau=17
let fec=f_norn
otherwise
let fec=fmay()
end case
-- Genera la fusion de documento y externos
call fusdoctos(fec)

let x_inst=0
-- Localiza una reactivacion, reanudacion de
actividades,
-- MC reactivacion de actividades y MC reactivacion
por
-- cambio de nombre o una baja del mismo dia.
declare c_rea cursor for
select doc_nope,doc_motmov from doc_ext1
where doc_fechop >= fec order by doc_nope
let xenco=' '
let x_tantos=0
foreach c_rea into xope,x_mov
if x_mov='16' or x_mov='22' or
x_mov='54' or x_mov='56'
then
if xenco=' ' then
let x_tantos=4
let xenco='R' # encuentre una reactivacion
# o reanudacion
else
continue foreach
end if
else
if xenco=' ' then
continue foreach
else
let x_tantos=-1
exit foreach
end if
end if
end foreach
drop table doc_ext1
if x_tantos > 0 then
return "S"
else
return "N"

```

```

end if
end function #busca
-- Borra el segmento que esta de mas.
function borrareg()
define f_reac date
case
when stat_cau=1 # elimina el registro de bajas
let f_reac=f_baj
call bor_bajas()
when stat_cau=2
let f_reac=f_liq
call bor_liquid()
when stat_cau=4
let f_reac=f_suce
call bor_suce()
when stat_cau=8
let f_reac=f_liq
call bor_suspen()
when stat_cau=17
if f_baj=f_nom then
let f_reac=f_baj
else
let f_reac=f_nom
end if
call bor_bajas()
call bor_nombre()
end case
if x_vari='2' then
let x_svar2=1
call reac_oblig(f_reac)
end if
end function # borra segmento
function reac_oblig(f_okreac) # reactiva obligaciones
define f_okreac date,
y_inst char(400)
-- Guarda los datos que tenia la obligacion
let y_inst=insert into bita_16 select 45,'x_vari clipped,
','x_svar2','current','x_alr','* from obligacion where obl_rfc="",
cau0.cau_rfc," and obl_deter=",cau0.cau_deter," and
obl_fecbaj="",
f_okreac,"
prepare a_obl1 from y_inst
execute a_obl1
update obligacion set obl_fecbaj = null where
obl_rfc=cau0.cau_rfc
and obl_deter=cau0.cau_deter and obl_fecbaj=f_okreac
end function # fin reactiva obligaciones
--
function bor_segnon(x_val)
define x_val smallint,
f_mov date
let stat_cau=stat_cau-x_val
if stat_cau<0 then
let stat_cau=stat_cau+x_val
return
end if
case
when x_val=16
if x_bajmov<>'14' and x_bajmov<>'33' then
let f_mov=f_nom
call bor_nombre()
end if
when x_val=8
let f_mov=f_sus
call bor_suspen()
when x_val=4
let f_mov=f_suce
call bor_suce()
when x_val=2
let f_mov=f_liq
call bor_liquid()

```

```

when x_val=1
let f_mov=f_baj
call bor_bajas()
end case
if x_reactiva='S' then
call reac_oblig(f_mov)
end if
end function # (borra el segmento non)

function bor_bajas()
define y_inst char(400)
-- Guarda la informacion de bajas en bita_9
let y_inst=insert into bita_6 select 45,13',
',1,current','x_alr','* from bajas where baj_rfc="",
cau0.cau_rfc," and baj_deter=",cau0.cau_deter
prepare y_baj from y_inst
execute y_baj
-- Elimina la baja
delete from bajas
where baj_rfc=cau0.cau_rfc and
baj_deter=cau0.cau_deter
let f_baj = null
end function # (borra bajas)

function bor_liquid()
define y_inst char(400)
-- Guarda la informacion de liquidacion en bita_9
let y_inst=insert into bita_9 select 45,13',
',4,current','x_alr','',liq_rfc,'" from liquidacion ',
' where liq_rfc="",cau0.cau_rfc,'"
prepare y_liq from y_inst
execute y_liq
-- elimina la liquidacion
delete from liquidacion
where liq_rfc=cau0.cau_rfc
let f_liq = null
end function # (borra liquidacion)

function bor_suce()
define y_inst char(400)
-- Guarda la informacion de sucesion en bita_7
let y_inst=insert into bita_7 select 45,13',
',2,current','x_alr','',suc_rfc,'" from sucesion where suc_rfc="",
cau0.cau_rfc,'"
prepare y_suc from y_inst
execute y_suc
-- elimina la sucesion
delete from sucesion
where suc_rfc=cau0.cau_rfc
let f_suce = null
end function # (borra sucesion)

function bor_suspen()
define y_inst char(400)
-- Guarda la informacion de supencion en bita_10
let y_inst=insert into bita_10 select 45,13',
',5,current','x_alr','',sus_rfc,'" from suspension where sus_rfc="",
cau0.cau_rfc,'" and sus_deter=",cau0.cau_deter
prepare y_sus from y_inst
execute y_sus
-- elimina la supencion
delete from suspension
where sus_rfc=cau0.cau_rfc
and sus_deter=cau0.cau_deter
let f_sus = null
end function # (borra suspension)

function bor_nombre()
define y_inst char(400)
-- Guarda la informacion de supencion en un
registro

```

```

    let y_inst='insert into bita_8 select 45,13',
    ',3,current','x_alr,',' from nombres where nom_rfca=',
    cau0.cau_rfc,'"
prepare y_nom from y_inst
execute y_nom
-- elimina la suspencion
delete from nombres
where nom_rfca=cau0.cau_rfc
let f_nom = null
end function # (borra suspencion)

-- existe en mas de un segmento
function analipo()
case
when -1 ** stat_cau=-1 # si es non esta en bajas
let x_nuonnon='S'
if s_ana =1 then
let x_vari='5'
let x_svar=1
let x_svar2=2
else
let x_vari='12'
let x_svar=4
let x_svar2=5
end if
call estabaj() # Establece la baja
let stat_cau=r_stat
let stat_cau=stat_cau-1
if cau0.cau_movs='1' then
insert into bita_11
values(45,x_vari,x_svar,cau0.cau_RFC,
0,current,x_alr,null,null,null,null,null,null,null,
null,null)
call reac_oblig(f_baj)
call bor_bajas()
let y_stat = stat_cau
-----
-- Vuelve a comenzar ---
-----
if y_stat not matches "[01248]" and stat_cau<>17 then
if s_ana=1 then
if (stat_cau-16)< 0 then
let x_vari3='7'
let x_svar6=1
let x_vari2='10'
let x_svar7=0
let x_svar4=1
let x_svar3=2
else
let x_vari='4'
let x_svar=1
let x_svar3=2
end if
else
if (stat_cau-16)< 0 then
let x_vari3='12'
let x_svar6=6
let x_vari2='12'
let x_svar7=8
let x_svar4=9
let x_svar3=7
else
let x_vari='12'
let x_svar=1
let x_svar3=2
end if
end if
call espar()
else
let x_vari='1'
let x_svar=0

```

```

let x_svar2=1
let x_svar3=2
call map_stat()
end if
else
if s_ana=1 then
let x_vari='5'
let x_svar=0
else
let x_vari='12'
let x_svar=3
end if
call asig_stat()
let x_reactiva='N'
call bor_segnon(16) # borra los demas
segmentos
call bor_segnon(8) # borra los demas
segmentos
call bor_segnon(4) # borra los demas
segmentos
call bor_segnon(2) # borra los demas
segmentos
end if

when -1 ** stat_cau=1 # es par
if s_ana =1 then
-----
- variante .. 4 --
-----
let x_vari='4'
let x_svar4=1
let x_vari2='10'
let x_vari3='7'
let x_svar6=1
let x_svar7=0
let x_svar3=2
else
let x_vari='12'
let x_svar4=9
let x_vari2='12'
let x_vari3='12'
let x_svar6=6
let x_svar7=8
let x_svar3=7
end if
let x_svar=1
call espar()
end case
end function # (Analiza tipo)

-----
Funcion es par
function espar()

define fec date

if (stat_cau-16)< 0 then
call q_gana()
if y_stat matches "[0248]" then
let x_vari=x_vari2
let x_svar= x_svar4
call map_stat()
return
end if
select max(doc_fechop) fechop,doc_nope,doc_motmov
from documentos where doc_rfc=cau0.cau_RFC
and doc_deter=cau0.cau_deter group by
doc_nope,doc_motmov
into temp hola
insert into hola select
max(ext_fechop),ext_nope,ext_motmov
from externos where ext_rfc=cau0.cau_RFC

```

```

and ext_deter=cau0.cau_deter group by
ext_nope,ext_motmov
declare c_may scroll cursor for
select fechop,doc_nope,doc_motmov
from hola order by 1 desc,2 desc,3 desc
open c_may
fetch first c_may into g_fechop,g_nope,g_mov
close c_may
if g_fechop is not null then
let x_vari=x_vari2
let x_svar= x_svar4
case
when g_mov='20' or g_mov='41' #liquidacion
let q_mayor=2
if f_liq is null then
let x_vari=x_vari3
let x_svar= x_svar3
let fec=fmay()
end if
when g_mov='18' or g_mov='42' #sucesion
let q_mayor=4
if f_suce is null then
let x_vari=x_vari3
let x_svar= x_svar3
let fec=fmay()
end if
when g_mov='21' or g_mov='43' #suspension
let q_mayor=8
if f_sus is null then
let x_vari=x_vari3
let x_svar= x_svar3
let fec=fmay()
end if
otherwise
let x_vari=x_vari3
let x_svar= x_svar3
let fec=fmay()
end case
end case
else
let x_vari=x_vari3
let x_svar= x_svar3
let fec=fmay() # busca fecha mayor de los status
end if
drop table hola
let stat_cau=stat_cau-q_mayor
let x_reactiva='N'
call bor_segnon(16) # borra los demas segmentos
call bor_segnon(8) # borra los demas segmentos
call bor_segnon(4) # borra los demas segmentos
call bor_segnon(2) # borra los demas segmentos
call bor_segnon(1) # borra los demas segmentos
let stat_cau=q_mayor
call map_stat()
else
# Esta en nombres pero no en bajas
insert into bita_11 values(45,x_vari,x_svar,cau0.cau_RFC,
0,current,x_alr,null,null,null,null,null,stat_orf,null,
null,null)
insert into bajas values (cau0.cau_rfc,cau0.cau_deter,
f_nom,'33')
# mapeo status a 2 (causante en baja)
update causante set cau_movs='2' where rowid=id
let cau0.cau_movs='2'
let stat_cau=stat_cau-16
-----
-- Guarda los datos que tenia la obligacion
let y_inst=insert into bita_16 select 45,'x_vari clipped,
',x_svar3,current,'x_alr,' from obligacion where
obl_rfc=','.
cau0.cau_rfc,'" and obl_deter=',cau0.cau_deter,' and
obl_fecbaj =','.
f_nom,'"

```

```

prepare a_obl34 from y_inst
execute a_obl34
update obligacion set obl_fecbaj=f_nom where
obl_rfc=cau0.cau_rfc
and obl_deter=cau0.cau_deter and obl_fecbaj =f_nom
-----}
let x_reactiva='N'
call bor_segnon(8) # borra los demas segmentos
call bor_segnon(4) # borra los demas segmentos
call bor_segnon(2) # borra los demas segmentos
end if
end function # es par
-----
quien gana
function q_gana()
let stat_cau=r_stat
if f_sus is not null and f_liq is not null then
insert into bita_11
values(45,x_vari3,x_svar6,cau0.cau_RFC,0,
current,x_alr,null,null,null,null,null,
null,null,null,null)
call bor_suspen()
let stat_cau=stat_cau -8
end if
if f_liq is not null and f_suce is not null then
insert into bita_11
values(45,x_vari2,x_svar7,cau0.cau_RFC,0,
current,x_alr,null,null,null,null,null,
null,null,null,null)
if cau0.cau_rfc[1,1]=' ' then # gana liquidacion
call bor_suce()
let stat_cau=stat_cau -4
else
call bor_liquid()
let stat_cau=stat_cau -2
end if
end if
let y_stat=stat_cau
end function # quien gana
-----
Checa RFC y digito verificador
function rfcver()
define rfcvno,x_rfcresp char(13)
let x_rfcresp=quita_blanco(cau0.cau_rfc)
-----
-- caso 4 RFC menor a 13 posiciones ----
-----
if (cau0.cau_rfc[1,1]=' ' and length(x_rfcresp)< 12) or # P.
Moral
(cau0.cau_rfc[1,1]<>' ' and length(x_rfcresp)< 13) then # P.
Fisica.
insert into bita_3 values (4,0,0,cau0.cau_rfc,0,current,x_alr,
'', '')
end if
-----
-- caso Digito verificador mal calculado ----
-----
if cau0.cau_rfc[13,13] not matches "[0-9]" and
cau0.cau_rfc[13,13]<>'A' then
insert into bita_3 values
(5,0,0,cau0.cau_rfc,cau0.cau_deter,current,
x_alr,'"')
end if
end function

function fmay()
define x_tipper char(1)
let x_tipper='F'
let f_mayor='01/01/01'
case
when f_liq is not null and f_suce is not null

```

- variante .. 10

```
insert into bita_11
values(45,x_vari2,0,cau0.cau RFC,current,
      x_ air,null,null,null,null,null,null,
      null,null,null)
      let x_vari='10'
      if cau0.cau_rfc[1,1]=' ' then
        call bor_suce()
        let x_tipper='M'
      else
        call bor_liquid()
      end if
      when cau0.cau_rfc[1,1]=' '
        let x_tipper='M'
      end case
      if x_tipper='F' then          # Persona fisica
        case
          when f_baj is not null
            if f_baj>f_suce or f_suce is null then
              if (x_bajmov='14' or x_bajmov='33') and
                f_nom>f_baj and f_nom is not null then
                let f_mayor=f_nom
                let q_mayor=17
              else
                if f_nom is not null and f_nom>f_baj then
                  let f_mayor=f_nom
                  let q_mayor=16
                else
                  let f_mayor=f_baj
                  let q_mayor=1
                end if
              end if
            else
              let f_mayor=f_suce
              let q_mayor=4
            end if
          when f_nom is not null
            if f_nom>f_suce or f_suce is null then
              let f_mayor=f_nom
              let q_mayor=16
            else
              let f_mayor=f_suce
              let q_mayor=4
            end if
          end case
        else
          # Persona moral
          case
            when f_baj is not null
              if f_baj>f_liq or f_liq is null then
                if (x_bajmov='14' or x_bajmov='33') and
                  f_nom>f_baj and f_nom is not null then
                  let f_mayor=f_nom
                  let q_mayor=17
                else
                  if f_nom is not null and f_nom>f_baj then
                    let f_mayor=f_nom
                    let q_mayor=16
                  else
                    let f_mayor=f_baj
                    let q_mayor=1
                  end if
                end if
              else
                let f_mayor=f_liq
                let q_mayor=2
              end if
            when f_nom is not null
              if f_nom>f_liq or f_liq is null then
                let f_mayor=f_nom
```

```
        let q_mayor=16
      else
        let f_mayor=f_liq
        let q_mayor=2
      end if
    end case
  end if
  if f_sus > f_mayor and f_sus is not null then
    let f_mayor=f_sus
    let q_mayor=8
  end if
  return f_mayor
end function
```

```
function quita_blanco(s)
  define s,r char(80),
  i,j,longi smallint
  let i=1
  let r=""
  while i<= length(s)
    if s[i,i]<' ' then
      let r=r clipped,s[i,i]
    end if
    let i=i+1
  end while
  return r
end function # quita_blanco
```

```
##### comienza el rfc en
function caso_16()
```

```
  if cau0.cau_rfc[1,1]=' ' then
    initialize w_fecfis to null
    select cau_fecalt into w_fecfis
    from hist_det where cau_rfc=cau0.cau_rfc
    and cau_deter=cau0.cau_deter
    call graba(w_fecfis,1)
  else
    # personas fisicas
    call val_fechfis()
  end if

  let w_fin = current
end function
```

```
function val_fechfis()
  define
  w_fedoc date
```

```
  initialize w_fedoc to null
  select cau_fecalt into w_fedoc
  from hist_det where cau_rfc=cau0.cau_rfc
  and cau_deter=cau0.cau_deter
  if w_fedoc is null then
    ## ----- var 2, busca en documentos
    select min(doc_fechop) into w_fedoc from documentos
    where doc_rfc = cau0.cau_rfc and
    doc_deter = cau0.cau_deter and
    doc_motmov='00'
    if w_fedoc is not null then
      call graba(w_fedoc,2)
    else
      initialize w_fedoc to null
      select min(doc_fechop) into w_fedoc
```

```
##var 2,
  from documentos
```



```

where doc_rfc = cau0.cau_rfc and
doc_deter = cau0.cau_deter and
doc_motmov='11'
if w_fedoc is not null then
    call graba(w_fedoc,2)
else
    initialize w_fedoc to null
    select min(ext_fechop) into
w_fedoc    ##var 2,
    from extemos
    where ext_rfc= cau0.cau_rfc and
    ext_deter = cau0.cau_deter and
    ext_motmov='11'
    if w_fedoc is not null then
        call graba(w_fedoc,2)
    else
        select min(ext_fechop) into w_fedoc
        from extemos
        where ext_rfc = cau0.cau_rfc and
        ext_deter = cau0.cau_deter and
        ext_motmov='00'
        if w_fedoc is not null then
            call graba(w_fedoc,2)
        else
            initialize w_fedoc to null
            call graba(w_fedoc,3)
        end if
    end if
end if
end if
else # Si lo encuentre en la extraccion
    call graba(w_fedoc,2)
end if
end function    ##fin de documentos

function graba(w_fech1,x)
define
w_fecha    char(8),
w_fech1    date,
x          smallint
if w_fech1 is null and cau0.cau_rfc[1,1]=' ' then
let w_fecha = "",cau0.cau_rfc[5,6],',',cau0.cau_rfc[7,8],
',',cau0.cau_rfc[9,10],""
let w_fech1 = w_fecha
if w_fech1 is null or length(w_fecha)<8 then
let x=3
let w_fech1='88/01/01'
end if
else
if w_fech1 is null then
let x=3
let w_fech1='88/01/01'
end if
end if
insert into bita_3 values(16,x,0,cau0.cau_rfc,cau0.cau_deter,
current,x_alr,null,null)
update causante set cau_fecalt = w_fech1 where rowid = id
let cau0.cau_fecalt=w_fech1
end function

function caso_27()
-- Inserta en bitacora 3 el caso 27
insert into bita_3 values('27',
'0',
'0',
cau0.cau_rfc,

```

```

cau0.cau_deter,
current,
cau0.cau_mescie,
"")
update causante set cau_motalt = '00' where cau_rfc =
cau0.cau_rfc and
cau_deter=0
let cau0.cau_motalt = '00'
end function

function caso_25()
define cmotalt char(2)
if length(cau0.cau_motalt) = 0 or cau0.cau_motalt is null then
-- Inserta en Bitacora_3 el caso 25
# busca en nacional motivo de alta
select cau_motalt into cmotalt from hist_det
where cau_rfc=cau0.cau_rfc and
cau_deter=cau0.cau_deter
and cau_motalt is not null and cau_motalt!=' '
if status<>NOTFOUND then
let cau0.cau_motalt=cmotalt
else
let cau0.cau_motalt='00'
end if
insert into bita_3 values('25',
'0',
'0',
cau0.cau_rfc,
cau0.cau_deter,
current,
cau0.cau_mescie,
"")
update causante set cau_motalt = cau0.cau_motalt where
cau_rfc = cau0.cau_rfc
and cau_deter=cau0.cau_deter
end if
end function

function caso_26()
define clave_mov char(2)
select cve_mov
into clave_mov
from cat_mov
where cve_mov = cau0.cau_motalt
if status = notfound then
-- Graba en bitacora 11 el caso 26
insert into bita_11 values('26',
'0',
'0',
cau0.cau_rfc,
cau0.cau_deter,
current,
cau0.cau_mescie,
cau0.cau_motalt,
"","","","","")
update causante set cau_motalt = '00' where
cau_rfc = cau0.cau_rfc and cau_deter=cau0.cau_deter
let cau0.cau_motalt = '00'
end if
end function

-- Funcion Calidad
function caso_29_30_31()
define query char(500),x_calcont smallint
define cont integer,temp char(5), Graba integer,varobli integer
define calidadnul record

```

```

num_caso      smallint ,
variante     smallint,
subvar       smallint,
rfc          char(13) ,
deter       smallint,
fecha_hora   datetime YEAR TO SECOND ,
alr          smallint ,
rfc_nuevo    char(13),
nombre       varchar(80)
end record
define calidadraro record
num_caso      smallint ,
variante     smallint,
subvar       smallint,
rfc          char(13) ,
deter       smallint,
fecha_hora   datetime YEAR TO SECOND ,
alr          smallint ,
cau_motalt   char(2),
cau_calcont  smallint,
cau_tipsoc   smallint,
cau_tipcont  char(1),
cau_secgob  integer,
cau_cveact   char(6),
cau_movs     char(1),
ba_cvebaj   char(2),
calcont_nal smallint,
tipcont_nal  char(1)
end record
initialize calidadnul.* to NULL
initialize calidadraro.* to NULL
let Graba = 0
let varobli = 0
If cau0.cau_calcont is null then
# Busca extraccion nacional
initialize x_calcont to null
select cau_calcont into x_calcont
from hist_det
where cau_rfc=cau0.cau_rfc
and cau_deter=cau0.cau_deter

let calidadnul.variante=0
if x_calcont is null then
let x_calcont=0
let calidadnul.variante=1
end if
let calidadnul.rfc=cau0.cau_rfc
let calidadnul.deter=cau0.cau_deter
let calidadnul.fecha_hora=current
let calidadnul.alr=x_alr
let calidadnul.num_caso=29
let calidadnul.subvar=0
insert into bita_3 values (calidadnul.*)
update causante set cau_calcont = x_calcont where
rowid=id
let cau0.cau_calcont = x_calcont
end if
let temp = cau0.cau_calcont
If cau0.cau_calcont <> 0 Then
If length(temp) < 4 Then
let Graba = 1
Else
let Graba = 0
let cont = length(temp) - 3
let temp = temp[1,cont]
Select cve_tipemp from cat_tipemp where
cve_tipemp = temp
If Status = notfound then
let Graba = 1
end if

```

```

if graba=0 and (temp matches "[3,4,5,7,8]"* or
temp='10') then
Select cve_contesp from
cat_contesp
where cve_contesp = cau0.cau_calcont
If Status = notfound then
let calidadnul.rfc=cau0.cau_rfc
let
calidadnul.deter=cau0.cau_deter
let calidadnul.fecha_hora=current
let calidadnul.alr=x_alr
let calidadnul.num_caso=31
let calidadnul.variante=0
let calidadnul.subvar=0
insert into bita_3 values (calidadnul.*)
End If
End If
End If
If Graba = 1 then
let calidadraro.rfc=cau0.cau_rfc
let calidadraro.deter=cau0.cau_deter
let calidadraro.fecha_hora=current
let calidadraro.alr=x_alr
let calidadraro.num_caso=30
let calidadraro.variante=0
let calidadraro.subvar=0
let
calidadraro.cau_calcont=cau0.cau_calcont
insert into bita_11 values (calidadraro.*)
update causante set cau_calcont = 0
where rowid=id
let cau0.cau_calcont = 0
End If
end if
end function

```

```

function caso_33()
define query char(500)
define cont integer,temp char(5), Graba integer,varobli integer
---caso 33
insert into bita_5
select 33,1,0,cau_rfc,cau_deter,current,cau_mescie,current
from causante,obligacion
where cau_rfc=obl_rfc and cau_deter=obl_deter
and cau_deter = 0
and obl_fecbaj is null
and obl_clave = 123
and not ( cau_calcont >= 1000
and cau_calcont <= 1999 )

insert into bita_5
select 33,2,0,cau_rfc,cau_deter,current,cau_mescie,current
from causante,obligacion
where cau_rfc=obl_rfc and cau_deter=obl_deter
and cau_deter = 0
and obl_fecbaj is null
and obl_clave = 154
and not ( cau_calcont >= 2000
and cau_calcont <= 2999 )

insert into bita_5
select 33,3,0,cau_rfc,cau_deter,current,cau_mescie,current
from causante,obligacion
where cau_rfc=obl_rfc and cau_deter=obl_deter
and cau_deter = 0
and obl_fecbaj is null
and not obl_clave = 123
and ( cau_calcont >= 1000
and cau_calcont <= 1999 )

```

```

insert into bita_5
select 33,4,0,cau_rfc,cau_deter,current,cau_mescie,current
from causante,obligacion
where cau_rfc=obl_rfc and cau_deter=obl_deter
and cau_deter = 0
and obl_fecbaj is null
and not obl_clave = 154
and ( cau_calcont >= 2000
and cau_calcont <= 2999 )

```

```

let query = 'insert into bita_5 ',
'select 33,5,0,cau_rfc,0,current,',x_alr,',current ',
'from causante,obligacion ',
'where cau_rfc=obl_rfc and cau_deter=obl_deter ',
'and cau_deter = 0 ',
'and obl_fecbaj is null ',
'and ( cau_calcont >= 2000 ',
'and cau_calcont <= 2999 ) ',
'and ( obl_clave = 123 or obl_clave = 154 ) ',
'group by cau_rfc ',
'having count(*) > 1 '

```

```

prepare c33x from query
execute c33x

```

```

let query = 'insert into bita_5 ',
'select 33,6,0,cau_rfc,0,current,',x_alr,',current ',
'from causante,obligacion ',
'where cau_rfc=obl_rfc and cau_deter=obl_deter ',
'and cau_deter = 0 ',
'and obl_fecbaj is null ',
'and ( cau_calcont >= 1000 ',
'and cau_calcont <= 1999 ) ',
'and ( obl_clave = 123 or obl_clave = 154 ) ',
'group by cau_rfc ',
'having count(*) > 1 '

```

```

prepare c33xx from query
execute c33xx

```

```

delete from obligacion
where obl_clave = 154
and obl_rfc in (select rfc from bita_5 where num_caso = 33
and variante = 6)

```

```
end function
```

```

function caso_34_35()
define cont integer,temp char(5),x_tipsoc smallint
define sociedadnul record
num_caso      smallint ,
variante      smallint,
subvar        smallint,
rfc           char(13) ,
deter         smallint,
fecha_hora    datetime YEAR TO SECOND ,
alr           smallint ,
rfc_nuevo     char(13),
nombre        varchar(80)
end record
define sociedadraro record
num_caso      smallint ,
variante      smallint,
subvar        smallint,
rfc           char(13) ,
deter         smallint,
fecha_hora    datetime YEAR TO SECOND ,
alr           smallint ,

```

```

cau_motalt    char(2),
cau_calcont   smallint,
cau_tipsoc    smallint,
cau_tipcont   char(1),
cau_segob     integer,
cau_cveact    char(6),
cau_movs      char(1),
baj_cvebaj    char(2),
calcont_nal   smallint,
tipcont_nal   char(1)
end record
initialize sociedadnul.* to NULL
initialize sociedadraro.* to NULL
if cau0.cau_tipsoc is null then

# Busca extraccion nacional
initialize x_tipsoc to null
select cau_tipsoc into x_tipsoc
from hist_det
where cau_rfc=cau0.cau_rfc
and cau_deter=cau0.cau_deter
let sociedadnul.variante=0
if x_tipsoc is null then
let x_tipsoc=0
let sociedadnul.variante=1
end if
let sociedadnul.rfc=cau0.cau_rfc
let sociedadnul.deter=cau0.cau_deter
let sociedadnul.fecha_hora=current
let sociedadnul.alr=x_alr
let sociedadnul.num_caso=34
let sociedadnul.subvar=0
insert into bita_3 values (sociedadnul.*)
update causante set cau_tipsoc = x_tipsoc where
rowid=id
let cau0.cau_tipsoc = x_tipsoc
Else
if cau0.cau_tipsoc <> 0 and cau0.cau_tipsoc <> 5

Then
let sociedadraro.rfc=cau0.cau_rfc
let sociedadraro.deter=cau0.cau_deter
let sociedadraro.fecha_hora=current
let sociedadraro.alr=x_alr
let sociedadraro.num_caso=35
let sociedadraro.variante=0
let sociedadraro.subvar=0
let sociedadraro.cau_tipsoc=cau0.cau_tipsoc
insert into bita_11 values (sociedadraro.*)
update causante set cau_tipsoc = 0 where
rowid=id
let cau0.cau_tipsoc = 0
End If
End If
end function

-- Funcion tipo
function caso_37_36()
define x_tipnal char(1)
define tiponul record
num_caso      smallint ,
variante      smallint,
subvar        smallint,
rfc           char(13) ,
deter         smallint,
fecha_hora    datetime YEAR TO SECOND ,
alr           smallint ,
rfc_nuevo     char(13),
nombre        varchar(80)
end record
define tiporaro record
num_caso      smallint ,

```

```

    variante      smallint,
    subvar       smallint,
    rfc          char(13) ,
    deter       smallint,
    fecha_hora   datetime YEAR TO SECOND ,
    alr         smallint ,
    cau_motalt  char(2),
    cau_calcont smallint,
    cau_tipsoc  smallint,
    cau_tipcont char(1),
    cau_seggob  integer,
    cau_cveact  char(6),
    cau_movs    char(1),
    baj_cvebaj  char(2),
    calcont_nal smallint,
    tipcont_nal char(1)
end record
initialize tiponul.* to NULL
initialize tiporaro.* to NULL
if cau0.cau_tipcont is null then
    let x_tipnal=''
        let tiponul.variante=1
    select cau_tipcont into x_tipnal from hist_det
    where cau_rfc=cau0.cau_rfc and
cau_deter=cau0.cau_deter
    if x_tipnal is null then
        let x_tipnal=''
    else
        let tiponul.variante=0
    end if
        let tiponul.rfc=cau0.cau_rfc
        let tiponul.deter=cau0.cau_deter
        let tiponul.fecha_hora=current
        let tiponul.alr=x_alr
        let tiponul.num_caso=37
        let tiponul.subvar=0
    insert into bita_3 values (tiponul.*)
    update causante set cau_tipcont = x_tipnal where
rowid=id
        let cau0.cau_tipcont = x_tipnal
    Else
        Select cve_tipcont from cat_tipcont where
cve_tipcont = cau0.cau_tipcont
        If Status = notfound then
            let tiporaro.rfc=cau0.cau_rfc
            let tiporaro.deter=cau0.cau_deter
            let tiporaro.fecha_hora=current
            let tiporaro.alr=x_alr
            let tiporaro.num_caso=36
            let tiporaro.variante=0
            let tiporaro.subvar=0
            let tiporaro.cau_tipcont=cau0.cau_tipcont
            insert into bita_11 values (tiporaro.*)
            update causante set cau_tipcont = '' where
rowid=id
                let cau0.cau_tipcont = ''
        End If
    End If
end function

function caso_41() # Causantes sin actividad
define x_cveact char(6)
define c41 record
    num_caso      smallint ,
    variante      smallint,
    subvar       smallint,
    rfc          char(13) ,
    deter       smallint,
    fecha_hora   datetime YEAR TO SECOND ,
    alr         smallint ,
    rfc_nuevo    char(13),

```

```

    nombre      varchar(80)
end record
initialize c41.* TO NULL
if cau0.cau_cveact IS NULL then

# Busca extraccion nacional
initialize x_cveact to null
select cau_cveact into x_cveact
from hist_det
where cau_rfc=cau0.cau_rfc
and cau_deter=cau0.cau_deter
let c41.variante=0
if x_cveact is null then
    let x_cveact='999999'
    let c41.variante=1
end if
let c41.alr=x_alr
let c41.num_caso=41
let c41.subvar=0
let c41.fecha_hora=CURRENT
let c41.rfc=cau0.cau_rfc
let c41.deter=cau0.cau_deter
insert into bita_3 values (c41.*) # inserta bitacora
update causante # Actualiza cve_actividad
desconocida
        set cau_cveact = x_cveact
        where cau_rfc=cau0.cau_rfc and
cau_deter=cau0.cau_deter
        let cau0.cau_cveact = x_cveact
    end if
end function

function caso_43() # Causantes con actividad fuera de catalogo

define c43 record
    num_caso      smallint ,
    variante      smallint,
    subvar       smallint,
    rfc          char(13) ,
    deter       smallint,
    fecha_hora   datetime YEAR TO SECOND ,
    alr         smallint ,
    cau_motalt  char(2),
    cau_calcont smallint,
    cau_tipsoc  smallint,
    cau_tipcont char(1),
    cau_seggob  integer,
    cau_cveact  char(6),
    cau_movs    char(1),
    baj_cvebaj  char(2),
    calcont_nal smallint,
    tipcont_nal char(1)
end record
initialize c43.* TO NULL

SELECT cve_activ FROM cat_activ
WHERE cat_activ.cve_activ = cau0.cau_cveact
if status = NOTFOUND then

let c43.alr=x_alr
let c43.num_caso=43
let c43.variante=0
let c43.subvar=0
let c43.fecha_hora=CURRENT
let c43.rfc=cau0.cau_rfc
let c43.deter=cau0.cau_deter
let c43.cau_cveact=cau0.cau_cveact

insert into bita_11 values (c43.*) # inserta bitacora

        update causante # Actualiza
cve_actividad sin catalogo

```

```

        set cau_cveact = 999999
        where cau_rfc=cau0.cau_rfc and
cau_deter=cau0.cau_deter
        let cau0.cau_cveact = '999999'
        end if
end function

-- caso 47 reporta los caso_s en que cau_movs del causante
-- sea igual a NULL

function caso_47()
--Inserto a bitacora 3
insert into bita_3
values(47,0,0,cau0.cau_rfc,cau0.cau_deter,
CURRENT,x_alr,null,null)
-- Actualiza a la tabla de causante con cau_movs=' '
update causante set cau_movs=' ' where rowid=id
-- inicializo el cau0.cau_movs a activo
let cau0.cau_movs=' '
end function

function caso_48()
--busca si el estatus de cau_movs esta dentro de
catalogo
if cau0.cau_movs<>' ' and cau0.cau_movs<>'1' and
cau0.cau_movs<>'2' and cau0.cau_movs<>'3' and
cau0.cau_movs<>'4' and cau0.cau_movs<>'5' then
insert into bita_11
values(48,0,0,cau0.cau_rfc,
cau0.cau_deter,CURRENT,x_alr,null,null,null,
null,null,cau0.cau_movs,null,null,null)
-- Actualiza a la tabla de causante con
cau_movs=' '
update causante set cau_movs=' ' where
rowid=id
-- inicializo el cau0.cau_movs a activo
let cau0.cau_movs=' '
end if
end function

function fusdoctos(fec)
define fec date
select doc_nope,doc_motmov,doc_fechop from
documentos
where doc_RFC=cau0.cau_rfc and
doc_deter=cau0.cau_deter and
doc_fechop >= fec and doc_motmov in ("16","22","54","56",
"01","02","04","05","06","07","08","09","18","20","21","29",
"41","42","43","46","47","48","49","50","51","52","53")
into temp doc_ext1 with no log

insert into doc_ext1
select ext_nope,ext_motmov,ext_fechop from externos
where ext_RFC=cau0.cau_rfc and
ext_deter=cau0.cau_deter and
ext_fechop >= fec and ext_motmov in ("16","22","54","56",
"01","02","04","05","06","07","08","09","18","20","21","29",
"41","42","43","46","47","48","49","50","51","52","53")
create index x_doc_ext on
doc_ext1(doc_fechop,doc_motmov)
end function

function caso_46()
define query char(400)
define poner char(2)
select edo into poner
from cat_ofh
where cve_ofh = cau0.cau_ofh
if status = NOTFOUND then

```

```

let query = 'insert into Bitas_1 ',
'select 46,1,0,current,'x_alr','* from causante where
rowid=id'
prepare c46 from query
execute c46
update causante set cau_mesclie = x_alr
where rowid=id
else
let query = 'insert into Bitas_1 ',
'select 46,0,0,current,'x_alr','* from causante where
rowid=id'
prepare c46a from query
execute c46a
update causante set cau_mesclie = poner
where rowid=id

end if
end function

function caso_7()
define bCons smallint
define query char(400)
define poner char(2)
define ofh3 char(5)
define ofhnal char(5)
define mesclial smallint
define afmal,x_long smallint
define dominal char(80),x_temp,codposnal char(5),mpional
smallint,
entfednal smallint,zondinal char(2),telefonar char(10)

if cau0.cau_ofh is not null then
let x_long=length(cau0.cau_ofh)
if x_long<=3 then
let query = 'insert into Bitas_1 ',
'select 7,3,0,current,'x_alr','* from causante where rowid=',id
prepare c7fg from query
execute c7fg
let x_temp='00000'
let cau0.cau_ofh =cau0.cau_ofh clipped,x_temp[1,5-x_long]
end if
select crh_equiv into poner
from cat_eqcrh
where crh_ante = cau0.cau_ofh
if status <> NOTFOUND then
let query = 'insert into Bitas_1 ',
'select 7,1,0,current,'x_alr','* from causante where rowid=',id
prepare c7a from query
execute c7a
update causante set cau_ofh = poner
where rowid=id
let cau0.cau_ofh = poner
end if

let ofh3 = cau0.cau_ofh[1,3]
select cve_ofh
from cat_ofh
where edo = cau0.cau_mesclie and cve_ofh = ofh3 and afr =
cau0.cau_afr
if status = NOTFOUND then
let bCons = 1
end if
end if

if cau0.cau_ofh is null or bCons = 1 then
select
cau_ofh,cau_mesclie,cau_afr,cau_domi,cau_codpos,cau_mpio,cau
u_entfed,cau_zondinal,cau_telefo

```

```

into
ofhnal,mescienal,afrnal,dornal,codposnal,mpional,entfednal,zon
dlinal,telefonl
from hist_det
where cau_rfc = cau0.cau_rfc and cau_deter =
cau0.cau_deter
let ofh3 = cau0.cau_ofh[1,3]
select cve_ofh
from cat_ofh
where edo = mescienal and cve_ofh = ofh3 and afr = afrnal
if status <> NOTFOUND then
let query = 'insert into Bitas_1 ',
'select 7,2,1,current,'x_alr','* from causante where
rowid='id
prepare c7d from query
execute c7d
update causante
set cau_ofh = ofhnal,cau_mescie = mescienal,cau_afr =
afrnal,
cau_domi = dornal,cau_codpos = codposnal,cau_mpio
= mpional,
cau_entfed = entfednal,cau_zondil =
zondinal,cau_telefo = telefonl
where rowid=id
else
select edo,afr
into mescienal,afrnal
from cat_ofh
where cve_ofh = ofh3
if status <> NOTFOUND then
let query = 'insert into Bitas_1 ',
'select 7,2,3,current,'x_alr','* from causante where
rowid='id
prepare c7b from query
execute c7b
update causante
set cau_mescie = mescienal,cau_afr = afrnal
where rowid=id
else
let query = 'insert into Bitas_1 ',
'select 7,2,2,current,'x_alr','* from causante where
rowid='id
prepare c7c from query
execute c7c
select min(cve_ofh),afr into ofhnal,afrnal
from cat_ofh
where edo = x_alr
group by afr
update causante
set cau_ofh = ofhnal,cau_afr = afrnal
where rowid=id
end if
end if
end function

```

– Funcion CalCont <>

```

function caso_32()
define x_caso char(550)

let x_caso= 'insert into bita_11
(num_caso,variante,subvar,rfc,deter,'
' fecha_hora,alr,cau_calcont,calcont_nal)',
'select 32,1,0,lc.cau_rfc,lc.cau_deter,current,'x_alr,
',lc.cau_calcont,c.cau_calcont from causante lc, hist_det c'
where lc.cau_rfc = c.cau_rfc and lc.cau_deter =
c.cau_deter and',
'lc.cau_deter = 0 and lc.calcont < c.cau_calcont',
'and c.cau_calcont is not null'

```

```

update causante set cau_calcont =
(select b.calcont_nal from bita_11 b
where b.rfc = causante.cau_rfc and b.deter =
causante.cau_deter)
where cau_rfc in (select rfc from bita_11
where num_caso = 32 and variante = 0)

```

End Function

```

----- Caso 133
function caso_133()
define rw_obli record like obligacion.*
define w_fec,x_fec,w_fecau date, id float
declare c133 cursor for
select *,rowid from obligacion
where obl_fecalt is null and obl_deter = 0

initialize w_fec,x_fec,w_fecau to null

foreach c133 into rw_obli.,id
select a.obl_fecalt into w_fecau from oblnal a
where a.obl_rfc = rw_obli.obl_rfc and a.obl_deter = 0 and
a.obl_clave = rw_obli.obl_clave
and a.obl_fecalt is not null
if status <> NOTFOUND then
insert into bita_16 values (133,1,0,current,x_alr,rw_obli.*)
update obligacion set obl_fecalt = w_fecau
where rowid=id
else
select cau_fecalt into w_fecau from causante
where cau_rfc = rw_obli.obl_rfc and
cau_deter=rw_obli.obl_deter
and cau_fecalt is not null
if status <> notfound then
insert into bita_16 values(133,2,0,CURRENT,x_alr,
rw_obli.obl_rfc,rw_obli.obl_deter,
rw_obli.obl_clave,")
update obligacion set obl_fecalt = w_fecau where
rowid=id
end if
end if
end foreach
end function

```

FIN3_LOC.4GL

{ Programa fin3: Descarga la B.D. Local Completa para reemplazo

en Nacional.

}

database shcp

globals

define x_alr char(2)

end globals

main

```

define sql_inst varchar(200), via
char(1),unidad_x_regreso smallint,
control datetime YEAR TO SECOND,archivo char(30)
defer quit
defer interrupt

```

--- obtiene la ALR al ejecutar fin3_loc.4ge /opt/siralr/alr'

```

let x_alr=arg_val(1)
let via=arg_val(2)
let unidad=arg_val(3)

```

```

-- VALIDA QUE RECIBA LA ALR
if length(x_alr) = 0 OR x_alr IS NULL OR x_alr = ""
then
    DISPLAY "!!! Parámetro de ALR
(opt/siralr/alr) Incorrecto. Ejecute de nuevo."
    exit program(1)
end if

-- Actualiza tabla casos para descarga total (caso 203)
= 203
UPDATE casos SET inicio = CURRENT WHERE caso

let control=CURRENT
DISPLAY control

.....
-- ***** RUTA DE ARCHIVOS DE PASO EN LOCAL *****

let sql_inst= 'Creando Directorios requeridos ...'
DISPLAY sql_inst

run "sh crea_fin3.sh"

-- ***** INICIA PROCESO DE DESCARGA

let sql_inst= 'Iniciando Proceso de Descarga de
Informacion Local.... Por Favor Espere'
DISPLAY sql_inst

UNLOAD TO "/var/spool/limpieza/fin3/caus.unl"
SELECT * FROM causante
run "compress /var/spool/limpieza/fin3/caus.unl"

UNLOAD TO "/var/spool/limpieza/fin3/baja.unl"
SELECT * FROM bajas
run "compress /var/spool/limpieza/fin3/baja.unl"

UNLOAD TO "/var/spool/limpieza/fin3/susp.unl"
SELECT * FROM suspension
run "compress /var/spool/limpieza/fin3/susp.unl"

UNLOAD TO "/var/spool/limpieza/fin3/docu.unl"
SELECT * FROM documentos
run "compress /var/spool/limpieza/fin3/docu.unl"

UNLOAD TO "/var/spool/limpieza/fin3/exte.unl"
SELECT * FROM externos
run "compress /var/spool/limpieza/fin3/exte.unl"

UNLOAD TO "/var/spool/limpieza/fin3/repr.unl"
SELECT * FROM representa
run "compress /var/spool/limpieza/fin3/repr.unl"

UNLOAD TO "/var/spool/limpieza/fin3/suce.unl"
SELECT * FROM sucesion
run "compress /var/spool/limpieza/fin3/suce.unl"

UNLOAD TO "/var/spool/limpieza/fin3/obli.unl" SELECT
* FROM obligacion
run "compress /var/spool/limpieza/fin3/obli.unl"

UNLOAD TO "/var/spool/limpieza/fin3/nomb.unl"
SELECT * FROM nombres
run "compress /var/spool/limpieza/fin3/nomb.unl"

```

```

UNLOAD TO "/var/spool/limpieza/fin3/liqu.unl" SELECT
* FROM liquidacion
run "compress /var/spool/limpieza/fin3/liqu.unl"

UNLOAD TO "/var/spool/limpieza/fin3/cedu.unl"
SELECT
ced_rfc,ced_let,ced_num,ced_fecalt,ced_fecbaj,ced_mot FROM
cedulas
run "compress /var/spool/limpieza/fin3/cedu.unl"

unload to "/var/spool/limpieza/fin3/cau_key.unl"
SELECT cau_rfc,cau_deter FROM causante

run "compress /var/spool/limpieza/fin3/cau_key.unl"

let sql_inst="touch /var/spool/limpieza/fin3/local_",x_alr
clipped
run sql_inst

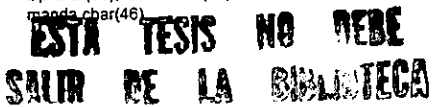
if via='C' or via='c' then
    .. ***** Respaldar archivos a cinta default *****
    let sql_inst = "sh boxadmin.sh ",x_alr
    run sql_inst
    let sql_inst = 'Respaldo de Información en cinta ....'
    DISPLAY sql_inst
    let sql_inst ="sh resp_fin3.sh ",unidad
    run sql_inst
    UPDATE casos SET fin = CURRENT WHERE caso =
203
    let control=CURRENT
    DISPLAY control
    let sql_inst = 'Proceso de Descarga Local Terminado.'
    DISPLAY sql_inst
else
    display ' Inicia transferencia (FTP)'
    let sql_inst=" sh transf.sh ",x_alr
    run sql_inst
    let archivo='ftp.txt'
    let x_regreso=pasa(archivo)
    if x_regreso=1 then
        display 'Termina transferencia sin exito'
        exit program
    else
        display 'Termina transferencia con exito'
        UPDATE casos SET fin = CURRENT WHERE caso
= 203
        let control=CURRENT
        DISPLAY control
        let sql_inst = 'Proceso de Descarga Local
Terminado.'
        DISPLAY sql_inst
    end if
end if

-- Actualiza tabla casos para descarga total (caso 203)

end main

DUDAS.4GL
database shcp
globals
define x_alr char(2), archivo,archivo1,x_tar,mueve char(400),
zip char(39), touch char(50),
manda char(46)

```



```

end globals
main
--- obtiene la ALR al ejecutar estcau.4ge `u/siralr/alr` ---
let x_alr=arg_val(1)
defer quit
defer interrupt
let archivo = "comp_casos",x_alr clipped
let archivo1= "comp_hijos",x_alr clipped

unload to archivo select * from comp_casos
unload to archivo1 select * from comp_hijos

let x_tar="tar cvf alr",x_alr clipped,
" comp_casos",x_alr clipped,
" comp_hijos",x_alr
run x_tar

let zip = "compress alr",x_alr
run zip

let mueve=" mv alr",x_alr clipped,"Z /var/spool/limpieza/ftps/"
run mueve

let touch = "echo > /var/spool/limpieza/ftps/yamanda",x_alr
display touch
run touch
let manda = "sh boxadmind.sh ",x_alr
run manda

end main

```

DUD136.4GL

```

database shcp
globals
  define x_alr char(2), archivo char(100),archivo1 char(100),
  zip char(100), touch char(100),
  x_tar char(250),mueve char(150),
  manda char(46)
end globals
main
--- obtiene la ALR al ejecutar estcau.4ge `u/siralr/alr` ---
let x_alr=arg_val(1)
defer quit
defer interrupt

let archivo = "comp_casos1",x_alr clipped
let archivo1= "comp_hijos1",x_alr clipped

unload to archivo select * from comp_casos1
unload to archivo1 select * from comp_hijos1

let x_tar="tar cvf caso136",x_alr clipped,
" comp_casos1",x_alr clipped,
" comp_hijos1",x_alr
run x_tar

let zip = "compress caso136",x_alr
run zip
let mueve=" mv caso136",x_alr clipped,"Z
/var/spool/limpieza/ftps/"
run mueve

let touch = "echo > /var/spool/limpieza/ftps/envia136",x_alr
display touch
run touch
let manda = "sh boxadmind.sh ",x_alr
run manda

```

end main

CERO.4GL

```

{ Programa 0(cero): Descarga la B.D. Local y prepara algunos
casos
de limpieza para traslado a B.D. Nacional.
}

database shcp

globals
  define x_alr char(2)
end globals

main

define sql_inst char(350),
control datetime YEAR TO SECOND
defer quit
defer interrupt
let x_alr=arg_val(1)

-- VALIDA QUE RECIBA LA ALR
if length(x_alr) = 0 OR x_alr IS NULL OR x_alr = "" then
  DISPLAY "!!! Parámetro de ALR (/opt/siralr/alr) Incorrecto.
Ejecute de nuevo."
  exit program(1)
end if

update casos set inicio = current where caso = 200
display "
display '-----'
display ' Inicia la descarga de la local para ser respaldada'
display ' El directorio donde se encontrara la informacion es'
display ' /var/spool/limpieza/respalr'
display '-----'
display ' Espere un Momento por favor.....'
display '-----'
unload to "var/spool/limpieza/respalr/causante.unl" select *
from causante
run "compress /var/spool/limpieza/respalr/causante.unl"
unload to "var/spool/limpieza/respalr/sucesion.unl" select * from
sucesion
run "compress /var/spool/limpieza/respalr/sucesion.unl"
unload to "var/spool/limpieza/respalr/liquidacion.unl" select *
from liquidacion
run "compress /var/spool/limpieza/respalr/liquidacion.unl"
unload to "var/spool/limpieza/respalr/nombres.unl" select * from
nombres
run "compress /var/spool/limpieza/respalr/nombres.unl"
unload to "var/spool/limpieza/respalr/bajas.unl" select * from
bajas
run "compress /var/spool/limpieza/respalr/bajas.unl"
unload to "var/spool/limpieza/respalr/cedulas.unl" select * from
cedulas
run "compress /var/spool/limpieza/respalr/cedulas.unl"
unload to "var/spool/limpieza/respalr/documentos.unl" select *
from documentos
run "compress /var/spool/limpieza/respalr/documentos.unl"
unload to "var/spool/limpieza/respalr/suspension.unl" select *
from suspension
run "compress /var/spool/limpieza/respalr/suspension.unl"
unload to "var/spool/limpieza/respalr/representa.unl" select *
from representa
run "compress /var/spool/limpieza/respalr/representa.unl"
unload to "var/spool/limpieza/respalr/externos.unl" select * from
externos
run "compress /var/spool/limpieza/respalr/externos.unl"

```



```
unload to "/var/spool/limpieza/respalr/obligacion.unl" select *
from obligacion
run "compress /var/spool/limpieza/respalr/obligacion.unl"
```

```
-- ***** LLAMA A LA FUNCION QUE CREA BITACORAS
DISPLAY 'Creando Bitacoras .....'
call borra_bitas()
DISPLAY 'Proceso de creacion de bitacoras Terminado'
```

```
display ***> ya puede respaldar el dir
/var/spool/limpieza/respalr <***
update casos set fin = current where caso = 200
end main
```

```
function borra_bitas()
```

```
SELECT * FROM systables WHERE tabname = 'bita_17'
if status <> NOTFOUND then
drop table bita_17
end if
```

```
CREATE TABLE Bitas_17 (
num_caso smallint NOT NULL,
variante smallint,
subvar smallint,
fecha_hora datetime YEAR TO SECOND
```

```
NOT NULL,
```

```
alr smallint NOT NULL,
rfc char(13),
deter smallint,
estcap smallint,
no_opera integer,
crh char(5),
crh_borrada char(5),
fecha date,
motmov char(2)
```

```
) in dbsir;
```

```
CREATE INDEX caso_bita17 ON Bitas_17
```

```
(
num_caso ASC,
variante ASC,
subvar ASC
)
```

```
SELECT * FROM systables WHERE tabname = 'bita_16'
if status <> NOTFOUND then
drop table bita_16
end if
```

```
CREATE TABLE Bitas_16 (
num_caso smallint NOT NULL,
variante smallint,
subvar smallint,
fecha_hora datetime YEAR TO SECOND
```

```
NOT NULL,
```

```
alr smallint NOT NULL,
obl_rfc char(13),
obl_deter smallint,
obl_clave smallint,
obl_fecalt date,
obl_fecbaj date
```

```
) in dbsir;
```

```
CREATE INDEX caso_bita16 ON Bitas_16
```

```
(
num_caso ASC,
variante ASC,
subvar ASC
)
```

```
SELECT * FROM systables WHERE tabname = 'bita_15'
if status <> NOTFOUND then
drop table bita_15
end if
```

```
CREATE TABLE Bitas_15 (
num_caso smallint NOT NULL,
variante smallint,
subvar smallint,
fecha_hora datetime YEAR TO SECOND
```

```
NOT NULL,
```

```
alr smallint NOT NULL,
ext_rfc char(13),
ext_deter smallint,
ext_nofh char(5),
ext_fechop date,
ext_mofech date,
ext_estcap smallint,
ext_nope integer,
ext_tipfor char(3),
ext_motmov char(2),
ext_claope smallint,
ext_nord integer,
ext_resver char(1)
```

```
) in dbsir;
```

```
CREATE INDEX caso_bita15 ON Bitas_15
```

```
(
num_caso ASC,
variante ASC,
subvar ASC
)
```

```
SELECT * FROM systables WHERE tabname = 'bita_14'
if status <> NOTFOUND then
drop table bita_14
end if
```

```
CREATE TABLE Bitas_14 (
num_caso smallint NOT NULL,
variante smallint,
subvar smallint,
fecha_hora datetime YEAR TO SECOND
```

```
NOT NULL,
```

```
alr smallint NOT NULL,
doc_rfc char(13),
doc_deter smallint,
doc_nofh char(5),
doc_fechop date,
doc_mofech date,
doc_estcap smallint,
doc_nope integer,
doc_tipfor char(3),
doc_motmov char(2),
doc_claope smallint,
doc_nord integer,
doc_resver char(1),
doc_movi char(1)
```

```
) in dbsir;
```

```
CREATE INDEX caso_bita14 ON Bitas_14
```

```
(
num_caso ASC,
variante ASC,
subvar ASC
)
```

```
SELECT * FROM systables WHERE tabname = 'bita_13'
if status <> NOTFOUND then
drop table bita_13
```

```

end if
CREATE TABLE Bit13 (
    num_caso          smallint NOT NULL,
    variante          smallint,
    subvar            smallint,
    fecha_hora        datetime YEAR TO SECOND
NOT NULL,
    alr               smallint NOT NULL,
    ced_rfc           char(13),
    ced_let           char(1),
    ced_num           integer,
    ced_fecalt        date,
    ced_fecbaj        date,
    ced_mot           char(1),
    ced_ctl           char(1)
) in dbsir;

CREATE INDEX caso_bit13 ON Bit13
(
    num_caso          ASC,
    variante          ASC,
    subvar            ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_12'
if status <> NOTFOUND then
drop table bita_12
end if

```

```

CREATE TABLE Bit12 (
    num_caso          smallint NOT NULL,
    variante          smallint,
    subvar            smallint,
    fecha_hora        datetime YEAR TO SECOND
NOT NULL,
    alr               smallint NOT NULL,
    rep_rfccau        char(13),
    rep_datcau        smallint,
    rep_rfcrcp        char(13),
    rep_nomrep        varchar(80),
    rep_domrep        varchar(80),
    rep_telef         varchar(10),
    rfc_rep           char(13),
    alr_rep           smallint
) in dbsir;

CREATE INDEX caso_bit12 ON Bit12
(
    num_caso          ASC,
    variante          ASC,
    rep_rfccau        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_18'
if status <> NOTFOUND then
drop table bita_18

```

```

end if
create table bita_18
(
    num_caso smallint not null,
    variante smallint,
    subvar smallint,
    fecha_hora datetime year to second not null,
    alr smallint not null,
    cau_rfcloc char(13),
    cau_rfc char(13),
    cau_deter smallint,
    cau_rsoc varchar(80),

```

```

    cau_domi varchar(80),
    cau_codpos char(5),
    cau_zonpos char(2),
    cau_mpio smallint,
    cau_afr smallint,
    cau_entfed smallint,
    cau_ofh char(5),
    cau_fecalt date,
    cau_motalt char(2),
    cau_calcont smallint,
    cau_tipsoc smallint,
    cau_tipcont char(1),
    cau_secgob integer,
    cau_cveact char(6),
    cau_mescie smallint,
    cau_zondil char(2),
    cau_serpos char(1),
    cau_telefo varchar(10),
    cau_estable smallint,
    cau_movs char(1)
) IN dbsir;

```

revoke all on bita_18 from "public";

```

create index i_bit18 on bita_18(num_caso,variante,subvar,alr);
create index i_bit182 on bita_18(cau_rfc,cau_deter);

```

```

SELECT * FROM systables WHERE tabname = 'res_casos1'
if status <> NOTFOUND then
drop table res_casos1

```

```

end if
create table res_casos1

```

```

(
    num_caso smallint,
    variante smallint,
    subvar smallint,
    alr smallint,
    rfc char(13),
    deter smallint,
    rsoc char(80),
    domi char(80),
    telefo char(10),
    rfc_dupli char(13),
    ofh char(5),
    afr smallint
) IN dbsir;

```

```

create index i_rescas1 on
res_casos1(num_caso,variante,subvar)
create index c_rescas1 on res_casos1(rfc,deter,alr)

```

```

SELECT * FROM systables WHERE tabname = 'res_casos'
if status <> NOTFOUND then
drop table res_casos

```

```

end if
create table res_casos

```

```

(
    num_caso smallint,
    variante smallint,
    subvar smallint,
    alr smallint,
    rfc char(13),
    deter smallint,
    rsoc char(80),
    domi char(80),
    telefo char(10),
    rfc_dupli char(13),
    ofh char(5),
    afr smallint
) IN dbsir ;

```

```

create index i_rescas on res_casos(num_caso,variante,subvar)
create index c_rescas on res_casos(rfc,deter,alr)

```

```

SELECT * FROM systables WHERE tabname = 'bita_esp'
if status <> NOTFOUND then
    drop table bita_esp
end if

```

```

create table bita_esp

```

```

(
    rfc char(13),
    deter smallint,
    ofh_loc char(5),
    alr_loc smallint,
    arr_loc smallint,
    ofh_nal char(5),
    alr_nal smallint,
    arr_nal smallint,
    ofh_queda char(5),
    alr_queda smallint,
    arr_queda smallint,
    alr_origen smallint
) IN dbsir ;

```

```

create index j_bitaesp on bita_esp (rfc,deter,alr_origen)

```

```

SELECT * FROM systables WHERE tabname = 'bita_1'
if status <> NOTFOUND then
    drop table bita_1
end if

```

```

CREATE TABLE Bita_1 (
    num_caso      smallint NOT NULL,
    variante      smallint,
    subvar        smallint,
    fecha_hora    datetime YEAR TO SECOND
NOT NULL,

```

```

    alr            smallint NOT NULL,
    cau_rfc        char(13),
    cau_deter      smallint,
    cau_rsoc       varchar(80),
    cau_domi       varchar(80),
    cau_codpos     char(5),
    cau_zonpos     char(2),
    cau_mpio       smallint,
    cau_afr        smallint,
    cau_entfed     smallint,
    cau_ofh        char(5),
    cau_fecall     date,
    cau_motall     char(2),
    cau_calcont    smallint,
    cau_tipsoc     smallint,
    cau_tipcont    char(1),
    cau_secgob     integer,
    cau_cveact     char(6),
    cau_mascie     smallint,
    cau_zondil     char(2),
    cau_serpos     char(1),
    cau_telefo     varchar(10),
    cau_estable    smallint,
    cau_movs       char(1)
) in dbsir;

```

```

CREATE INDEX caso_bita1 ON Bita_1
(
    num_caso      ASC,
    variante      ASC,
    subvar        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_3'
if status <> NOTFOUND then
    drop table bita_3
end if

```

```

CREATE TABLE Bita_3 (
    num_caso      smallint NOT NULL,
    variante      smallint,
    subvar        smallint,
    rfc           char(13) NOT NULL,
    deter         smallint,
    fecha_hora    datetime YEAR TO SECOND
NOT NULL,

```

```

    alr            smallint NOT NULL,
    rfc_nuevo     char(13),
    nombre        varchar(80)
) in dbsir;

```

```

CREATE INDEX caso_bita3 ON Bita_3
(
    num_caso      ASC,
    variante      ASC,
    subvar        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_2'
if status <> NOTFOUND then
    drop table bita_2
end if

```

```

CREATE TABLE Bita_2 (
    num_caso      smallint NOT NULL,
    variante      smallint,
    subvar        smallint,
    rfc           char(13) NOT NULL,
    deter         smallint,
    fecha_hora    datetime YEAR TO SECOND
NOT NULL,

```

```

    atr            smallint NOT NULL,
    crh_ant       char(5),
    alr_ant_liq   smallint,
    arr_ant       smallint
) in dbsir;

```

```

CREATE INDEX caso_bita2 ON Bita_2
(
    num_caso      ASC,
    variante      ASC,
    subvar        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_5'
if status <> NOTFOUND then
    drop table bita_5
end if

```

```

CREATE TABLE Bita_5 (
    num_caso      smallint NOT NULL,
    variante      smallint,
    subvar        smallint,
    rfc           char(13) NOT NULL,
    deter         smallint,
    fecha_hora    datetime YEAR TO SECOND
NOT NULL,

```

```

    alr            smallint NOT NULL,
    fecha_movimiento date
) in dbsir;

```

```

CREATE INDEX caso_bita5 ON Bita_5
(

```

```

        num_caso      ASC,
        variante      ASC,
        subvar        ASC
    )

SELECT * FROM systables WHERE tabname = 'bita_6'
if status <> NOTFOUND then
drop table bita_6
end if
CREATE TABLE Bita_6 (
    num_caso      smallint NOT NULL,
    variante      smallint,
    subvar        smallint,
    fecha_hora    datetime YEAR TO SECOND
NOT NULL,
    alr           smallint NOT NULL,
    baj_rfc       char(13),
    baj_deter     smallint,
    baj_fec       date,
    baj_cvebaj    char(2)
) in dbsir;

CREATE INDEX caso_bita6 ON Bita_6
(
    num_caso      ASC,
    variante      ASC,
    subvar        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_7'
if status <> NOTFOUND then
drop table bita_7
end if
CREATE TABLE Bita_7 (
    num_caso      smallint NOT NULL,
    variante      smallint,
    subvar        smallint,
    fecha_hora    datetime YEAR TO SECOND
NOT NULL,
    alr           smallint NOT NULL,
    suc_rfc       char(13),
    suc_fechi     date,
    suc_nomalb    varchar(80),
    suc_rfcalb    char(13),
    suc_domalib   varchar(80),
    rfc_alb       char(13),
    alr_alb       smallint
) in dbsir;

CREATE INDEX caso_bita7 ON Bita_7
(
    num_caso      ASC,
    variante      ASC,
    subvar        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_8'
if status <> NOTFOUND then
drop table bita_8
end if
CREATE TABLE Bita_8 (
    num_caso      smallint NOT NULL,
    variante      smallint,
    subvar        smallint,
    fecha_hora    datetime YEAR TO SECOND
NOT NULL,
    alr           smallint NOT NULL,

```

```

        nom_rfca    char(13),
        nom_rfc     char(13),
        nom_fecha   date
    ) in dbsir;

CREATE INDEX caso_bita8 ON Bita_8
(
    num_caso      ASC,
    variante      ASC,
    subvar        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_9'
if status <> NOTFOUND then
drop table bita_9
end if
CREATE TABLE Bita_9 (
    num_caso      smallint NOT NULL,
    variante      smallint,
    subvar        smallint,
    fecha_hora    datetime YEAR TO SECOND
NOT NULL,
    alr           smallint NOT NULL,
    status        char(1),
    rfc           char(13),
    liq_rfc       char(13),
    liq_fechi     date,
    liq_nomialq   varchar(80),
    liq_rfclicq   char(13),
    liq_domalq    varchar(80),
    liq_telliq    varchar(10),
    alr_liq       smallint
) in dbsir;

CREATE INDEX caso_bita9 ON Bita_9
(
    num_caso      ASC,
    variante      ASC,
    subvar        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_10'
if status <> NOTFOUND then
drop table bita_10
end if
CREATE TABLE Bita_10 (
    num_caso      smallint NOT NULL,
    variante      smallint,
    subvar        smallint,
    fecha_hora    datetime YEAR TO SECOND
NOT NULL,
    alr           smallint NOT NULL,
    sus_rfc       char(13),
    sus_deter     smallint,
    sus_fecsus    date,
    sus_nomaout   varchar(80),
    sus_domaut    varchar(80)
) in dbsir;

CREATE INDEX caso_bita10 ON Bita_10
(
    num_caso      ASC,
    variante      ASC,
    subvar        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_11'
if status <> NOTFOUND then

```

```

drop table bita_11
end if
CREATE TABLE Bita_11 (
  num_caso      smallint NOT NULL,
  variante      smallint,
  subvar        smallint,
  rfc           char(13) NOT NULL,
  deter         smallint,
  fecha_hora    datetime YEAR TO SECOND
NOT NULL,
  alr           smallint NOT NULL,
  cau_motallt   char(2),
  cau_calcont   smallint,
  cau_tipsoc    smallint,
  cau_tipcont   char(1),
  cau_secgob    integer,
  cau_cveact    char(6),
  cau_movs      char(1),
  baj_cvebaj    char(2),
  calcont_nal   smallint,
  tipcont_nal   char(1)
) in dbsir;

CREATE INDEX caso_bita11 ON Bita_11
(
  num_caso      ASC,
  variante      ASC,
  subvar        ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_matriz'
if status <> NOTFOUND then
drop table bita_matriz
end if
CREATE TABLE bita_matriz (
  num_caso      smallint,
  variante      smallint,
  subvar        smallint,
  fecha_hora    datetime YEAR TO SECOND,
  alr_origen    smallint,
  rfc           char(13),
  deter         smallint,
  rsoc          varchar(80),
  movs          char(1),
  alr_destino   smallint,
  afr           smallint,
  ofh           char(5)
) in dbsir;

CREATE INDEX j_bita_matriz ON bita_matriz
(
  num_caso      ASC,
  variante      ASC,
  subvar        ASC,
  rfc           ASC
)

```

```

SELECT * FROM systables WHERE tabname = 'bita_18'
if status <> NOTFOUND then
drop table bita_18
end if
CREATE TABLE Bita_18 (
  num_caso      smallint NOT NULL,
  variante      smallint,
  subvar        smallint,
  fecha_hora    datetime YEAR TO SECOND
NOT NULL,
  alr           smallint NOT NULL,

```

```

  cau_rfcloc    char(13),
  cau_rfc       char(13),
  cau_deter     smallint,
  cau_rsoc      varchar(80),
  cau_domi      varchar(80),
  cau_codpos    char(5),
  cau_zonpos    char(2),
  cau_mpio      smallint,
  cau_afr       smallint,
  cau_entfed    smallint,
  cau_ofh       char(5),
  cau_fecalt    date,
  cau_motallt   char(2),
  cau_calcont   smallint,
  cau_tipsoc    smallint,
  cau_tipcont   char(1),
  cau_secgob    integer,
  cau_cveact    char(6),
  cau_mesocie  smallint,
  cau_zondill   char(2),
  cau_serpos    char(1),
  cau_telefo    varchar(10),
  cau_estable   smallint,
  cau_movs      char(1)
) in dbsir;

```

```

create index i_bit18 on bita_18(num_caso,variante,subvar,alr);
create index i_bit182 on bita_18(cau_rfc,cau_deter);

```

----- CREACION DE CATALOGOS NUEVOS -----

```

SELECT * FROM systables WHERE tabname = 'cat_cedmot'
if status <> NOTFOUND then
drop table cat_cedmot
end if
CREATE TABLE cat_cedmot (
  cve_mot       char(1) NOT NULL,
  desc_mot      varchar(30)
) in dbsir;

CREATE UNIQUE INDEX XPKcat_cedmot ON
cat_cedmot
(
  cve_mot       ASC
)

LOAD FROM "cat_cedmot.uni" INSERT INTO
cat_cedmot;

```

```

SELECT * FROM systables WHERE tabname = 'cat_contesp'
if status <> NOTFOUND then
drop table cat_contesp
end if
CREATE TABLE cat_contesp (
  cve_contesp   smallint NOT NULL,
  desc_sector   varchar(30),
  desc_grupo    varchar(50)
) in dbsir;

CREATE UNIQUE INDEX XPKcat_secesp ON
cat_contesp
(
  cve_contesp   ASC
)

LOAD FROM "cat_contesp.uni" INSERT INTO
cat_contesp;

```

```

SELECT * FROM systables WHERE tabname = 'cat_obl_obs'
if status <> NOTFOUND then
drop table cat_obl_obs
end if
CREATE TABLE cat_obl_obs (
oblig_obs smallint NOT NULL,
persona char(1),
oblig_equiv smallint
) in dbsir;

CREATE UNIQUE INDEX XPKcat_obl_obs ON
cat_obl_obs
(
oblig_obs ASC,
persona ASC
)

LOAD FROM "cat_obl_obs.uni" INSERT INTO
cat_obl_obs;

```

```

SELECT * FROM systables WHERE tabname = 'cat_oblisr'
if status <> NOTFOUND then
drop table cat_oblisr
end if
CREATE TABLE cat_oblisr (
cve_tiper char(1) NOT NULL,
cve_oblig smallint NOT NULL
) in dbsir;

CREATE UNIQUE INDEX XPKcat_oblisr ON cat_oblisr
(
cve_tiper ASC,
cve_oblig ASC
)

LOAD FROM "cat_oblisr.uni" INSERT INTO cat_oblisr;

```

```

SELECT * FROM systables WHERE tabname = 'cat_tip_per'
if status <> NOTFOUND then
drop table cat_tip_per
end if
CREATE TABLE cat_tip_per (
cve_tiper char(1) NOT NULL,
desc_tiper varchar(30)
) in dbsir;

CREATE UNIQUE INDEX XPKcat_tip_per ON
cat_tip_per
(
cve_tiper ASC
)

LOAD FROM "cat_tip_per.uni" INSERT INTO
cat_tip_per;

```

```

SELECT * FROM systables WHERE tabname = 'cat_tipcont'
if status <> NOTFOUND then
drop table cat_tipcont
end if
CREATE TABLE cat_tipcont (
cve_tipcont char(1) NOT NULL,
desc_tipcont varchar(30)
) in dbsir;

CREATE UNIQUE INDEX XPKcat_tipcont ON
cat_tipcont
(
cve_tipcont ASC

```

```

)

LOAD FROM "cat_tipcont.uni" INSERT INTO
cat_tipcont;

```

```

SELECT * FROM systables WHERE tabname = 'cat_tipemp'
if status <> NOTFOUND then
drop table cat_tipemp
end if
CREATE TABLE cat_tipemp (
cve_tipemp smallint NOT NULL,
desc_tipemp varchar(50)
) in dbsir;

CREATE UNIQUE INDEX XPKcat_tipemp ON
cat_tipemp
(
cve_tipemp ASC
)

LOAD FROM "cat_tipemp.uni" INSERT INTO
cat_tipemp;

```

```

SELECT * FROM systables WHERE tabname = 'cat_tipsoc'
if status <> NOTFOUND then
drop table cat_tipsoc
end if
CREATE TABLE cat_tipsoc (
cve_tipsoc smallint NOT NULL,
desc_tipsoc varchar(30)
) in dbsir;

CREATE UNIQUE INDEX XPKcat_tipsoc ON
cat_tipsoc
(
cve_tipsoc ASC
)

LOAD FROM "cat_tipsoc.uni" INSERT INTO
cat_tipsoc;

```

```

SELECT * FROM systables WHERE tabname = 'cat_status'
if status <> NOTFOUND then
drop table cat_status
end if
create table cat_status (
cve_status char(1) not null,
desc_status varchar(30)
) in dbsir;
create unique index xpkcat_status on cat_status
(cve_status);

LOAD FROM "cat_status.uni" INSERT INTO
cat_status;

```

```

SELECT * FROM systables WHERE tabname = 'comp_casos'
if status <> NOTFOUND then
drop table comp_casos
end if

SELECT * FROM systables WHERE tabname = 'comp_casos1'
if status <> NOTFOUND then
drop table comp_casos1
end if

create table comp_casos1

```

```
(
  num_caso smallint,
  variante smallint,
  subvar smallint,
  alr smallint,
  rfc char(13),
  deter smallint,
  ofh char(5),
  afr smallint
) in dbsir;
revoke all on comp_casos1 from "public";
create index idx_c_casos2 on comp_casos1(num_caso,variante);
create index idx_c_casos3 on comp_casos1(rfc,deter);
```

```
create table comp_casos
```

```
(
  num_caso smallint,
  variante smallint,
  subvar smallint,
  alr smallint,
  rfc char(13),
  deter smallint,
  ofh char(5),
  afr smallint
) in dbsir;
```

```
revoke all on comp_casos from "public";
```

```
create index it_comp_cas on comp_casos(num_caso,variante);
create index it_comp_cas1 on comp_casos(rfc,deter);
```

```
SELECT * FROM systables WHERE tabname = 'cat_eqcrh'
```

```
if status <> NOTFOUND then
```

```
  drop table cat_eqcrh
```

```
end if
```

```
create table cat_eqcrh
```

```
(
  crh_ante char(5) not null,
  crh_equiv char(5) not null
) in dbsir;
```

```
revoke all on cat_eqcrh from "public";
```

```
create unique index xpkcat_eqcrh on cat_eqcrh (crh_ante);
```

```
LOAD FROM "cat_eqcrh.uni" INSERT INTO cat_eqcrh;
```

```
SELECT * FROM systables WHERE tabname = 'cat_comp2'
```

```
if status <> NOTFOUND then
```

```
  drop table cat_comp2
```

```
end if
```

```
create table cat_comp2
```

```
(
  cve_obl          smallint ,
  incompat        smallint ,
  cve_catcont     smallint
) in dbsir;
```

```
revoke all on cat_comp2 from "public";
```

```
create index xpkcat_comp2 on cat_comp2 (cve_obl);
```

```
LOAD FROM "cat_comp2.uni" INSERT INTO cat_comp2;
```

```
SELECT * FROM systables WHERE tabname = 'comp_hijos1'
```

```
if status <> NOTFOUND then
```

```
  drop table comp_hijos1
```

```
end if
```

```
create table comp_hijos1
```

```
(
  num_caso smallint,
  variante smallint,
  subvar smallint,
```

```
  alr smallint,
  rfcpad char(13),
  rfchij char(13)
```

```
) in dbsir;
```

```
revoke all on comp_hijos1 from "public";
```

```
create index idx_c_hijos1 on comp_hijos1(num_caso,variante);
```

```
create index idx_c_hijos2 on comp_hijos1(rfcpad,rfchij);
```

```
SELECT * FROM systables WHERE tabname = 'comp_hijos'
```

```
if status <> NOTFOUND then
```

```
  drop table comp_hijos
```

```
end if
```

```
create table comp_hijos
```

```
(
  num_caso smallint,
  variante smallint,
  subvar smallint,
  alr smallint,
  rfcpad char(13),
  rfchij char(13)
) in dbsir;
```

```
revoke all on comp_hijos from "public";
```

```
create index ix_hijo3 on comp_hijos(num_caso,variante);
```

```
create index ix_hijo4 on comp_hijos(rfcpad,rfchij);
```

```
select * from sysindexes where idxname = 'iob_clave'
```

```
if status = NOTFOUND then
```

```
  create index iob_clave on obligacion(obl_clave)
```

```
end if
```

```
SELECT * FROM systables WHERE tabname = 'bita_caudel'
```

```
if status <> NOTFOUND then
```

```
  drop table bita_caudel
```

```
end if
```

```
create table bita_caudel
```

```
(
  num_caso smallint,
  variante smallint,
  subvar smallint,
  rfc char(13),
  deter smallint,
  rsoc char(80)
) in dbsir;
```

```
revoke all on bita_caudel from "public";
```

```
end function
```

BITAGRAL.4GL

```
database shcp
```

```
globals
```

```
  define x_alr char(2)
```

```
  define baja char(700)
```

```
  define donde char(100)
```

```
end globals
```

```
main
```

```
  --- obtiene la ALR al ejecutar estcau.4ge '/u/siralr/alr' --
```

```
  let x_alr= arg_val(1)
```

```
-- busca la tabla bitagral--
```

```
select * from systables
where tabname ="bitagral"
```

```
--Crea la tabla bitagral--
```

```
if status <> NOTFOUND then
```

```
  drop table bitagral
```

```

end if

-- Creandó la tabla bitagral

create table bitagral
(
    caso            smallint,
    variante        smallint,
    subvar          smallint,
    rfc             char(13),
    deter          smallint,
    bita           smallint
)

--insertando las bitas

Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,cau_rfc,cau_deter,01
from bita_1;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,rfc,deter,02
from bita_2;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,rfc,deter,03
from bita_3;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,rfc,deter,05
from bita_5;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,baj_rfc,baj_deter,06
from bita_6;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso, variante,subvar,suc_rfc,0,07
from bita_7;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,nom_rfca,0,08
from bita_8;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,rfc,0,09
from bita_9 where num_caso<>60 and num_caso<>65 and
num_caso<>66;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,liq_rfc,0,09
from bita_9 where num_caso=60 or num_caso=65 or
num_caso=66;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,sus_rfc,sus_deter,10
from bita_10;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,rfc,deter,11
from bita_11;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,rep_rfccau,rep_detcau,12
from bita_12;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,ced_rfc,0,13
from bita_13;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,doc_rfc,doc_deter,14
from bita_14;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,ext_rfc,ext_deter,15
from bita_15;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,obl_rfc,obl_deter,16
from bita_16;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,rfc,deter,17
from bita_17;
Insert into bitagral (caso,variante,subvar,rfc,deter,bita)
select num_caso,variante,subvar,cau_rfcloc,cau_deter,18

```

```

from bita_18

let donde = 'opt/siratr/limpieza/borrados',x_alr,'out'

let baja =
' select caso,variante,subvar,rfc,deter,cau_rsoc',
' from bitagral, causante',
' where rfc=cau_rfc and deter=cau_deter and ('
' (caso = 1 and variante = 0 and subvar = 0) or',
' (caso = 2 and variante = 0 and subvar = 0) or',
' (caso = 10 and variante = 3 and subvar = 0) or',
' (caso = 15 and variante = 4 and subvar = 6) or',
' (caso = 136 and variante = 5 and subvar = 6) or',
' (caso = 137 and variante = 1 and subvar = 6) or',
' (caso = 138 and variante = 3 and subvar = 6) )'

unload to donde baja

end main

```


BIBLIOGRAFIA

- Bell D 1992 "Distributed Database Systems" Addison Wesley
- Chorafas, Dimitris N. 1993 "Object-oriented databases" PTR Prentice-Hall
- Hawryszkiewycz I.T. "Database Analysis and Design" Maxwell Macmillan International
- Kemper.A.H.1994 "Object-oriented database management: applications in engineering and computer science" Prentice-Hall
- Khoshafian S. 1993 "Object-oriented databases" Willey Professional Computing
- Kim W. 1995 "Modern database systems: the object model, interoperability, and beyond" ACM press
- Renaud P.E. 1996 "Introduction to Client/Server Systems" 2nd edition Willey Revistas:
- Codd E.F. 1990 "The Relational Model for Database Management Version 2" Addison Wesley
- Date C.J. 1995 "An Introduction to Database Systems" sixth edition Addison Wesley
- Date C.J. 1985 "An introduction to Database Systems vol II" Addison Wesley
- Korth, H.F. 1991 "Database System Concepts" 2nd edition McGraw Hill
- Wiederhold G. 1987 "File Organization for Database Design" McGraw Hill