

7
2ej



UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO Y DESARROLLO DE UN SISTEMA DE BASE
DE DATOS PARA EL ANALISIS BURSATIL.

T E S I S
Que para obtener el Título de
INGENIERO EN COMPUTACION
p r e s e n t a
FRANCISCO ARNAL FUERTES



México, D. F.

1993

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



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 está 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.

I N D I C E

Objetivo	1
Prólogo	2
I Introducción	
1.1 Descripción del problema	4
1.2 Solución propuesta	5
II Análisis Bursátil	
2.1 Definición	8
2.2 Análisis de mercado de dinero	9
2.3 Análisis técnico	11
2.4 Análisis fundamental	12
2.5 Conclusión	14
III Metodología de desarrollo	
3.1 Ambientes de desarrollo de sistemas	15
3.2 Metodologías de desarrollo para cuarta generación	17
3.3 Enfoque de Síntesis	18
3.4 El enfoque analítico	25
IV Diseño lógico de la base de datos	
4.1 Ideas alrededor de la base de datos	27
4.2 Descripción de la base de datos a nivel de tablas	29
V Diseño físico de la base de datos	
5.1 Desnormalización	38
5.2 Indexación	40
VI Implementación e implantación del sistema	
6.1 Características generales	44
6.2 Implementación	45
6.3 Clases de usuarios	49
6.4 Implantación	50
VII Caso de estudio: Sistema de Análisis Fundamental	
7.1 Análisis Fundamental	52
7.2 Rutinas comunes	54
VIII Conclusiones	58
Apéndice A. Modelo relacional de base de datos	
Apéndice B. Manejador de base de datos PROGRESS	
Apéndice C. Base de datos para análisis bursátil	73
Bibliografía	82

OBJETIVO

El objetivo de este trabajo es presentar el diseño y desarrollo de un sistema de base de datos que proporcione eficientemente la información necesaria para el análisis bursátil y con la suficiente flexibilidad para incorporar futuros requerimientos.

PROLOGO

Hoy en día, los grandes avances en el área de telecomunicaciones y sobre todo el ritmo acelerado de vida que llevamos, nos obliga a tener la información necesaria para nuestro trabajo en forma organizada, confiable, oportuna y presentada en forma amigable; esto ha llevado a que muchas empresas consideren la información como su activo más importante.

Lo anterior es también válido en el medio financiero y bursátil de nuestro país, en particular en sus áreas de análisis donde se necesita conocer y procesar grandes volúmenes de información para poder estudiar en forma correcta el comportamiento de la bolsa, las empresas que cotizan en ella, las tasas de interés, la economía, etcétera.

El trabajo aquí presentado muestra el diseño y desarrollo de un sistema de base de datos que proporciona a los analistas bursátiles la información necesaria para poder realizar las actividades mencionadas.

En el capítulo 1 se muestran los requerimientos sobre los cuales se diseñó el trabajo. El capítulo 2 da la definición de análisis bursátil, su importancia dentro del medio financiero y las áreas principales en las cuales se puede dividir; aquí también se dan a conocer sus requerimientos de información, tanto en cuestión a fuentes como volúmenes requeridos. Este capítulo nos servirá para entender el marco conceptual sobre el cual está basado el diseño de la base de datos. Los elementos teóricos de diseño de base de datos usadas en el desarrollo del trabajo son expuestas en el capítulo 3, en el cual veremos los puntos fundamentales de

los ambientes de programación de la llamada cuarta generación y dentro de éstos, los enfoques de síntesis y analítico para el diseño de bases de datos de tipo relacional.

La parte medular de la tesis (capítulo 4) muestra el diseño lógico de la base de datos. En este capítulo se explican las ideas más importantes alrededor de la base, así como su tabla de relaciones. De la misma forma se explica cada tabla a nivel general y se dan ejemplos del uso del modelo en diversas áreas. En el capítulo 5, donde se describe el diseño físico de la base, se muestran las políticas establecidas para la creación y uso de índices con el fin de obtener el rendimiento óptimo del sistema. Las consideraciones especiales usadas en la implementación e implantación del sistema dentro del área de Análisis Bursátil de Operadora de Bolsa son mostradas en el capítulo 6. Aquí se verán todos los ajustes realizados al sistema para adecuarlo a las características del lugar en donde se implantó, como son: ambientes de trabajo, sistemas operativos usados, características de la máquina en disco, memoria, número de usuarios, etcétera. Como caso de estudio se muestra en el capítulo 7 el desarrollo básico del sistema del área de análisis fundamental tomando como soporte la base de datos diseñada. Los resultados obtenidos y las conclusiones del trabajo se dan en el capítulo 8.

Al final del trabajo existen tres apéndices que sirven como complemento; en el apéndice A se mencionan los fundamentos del modelo relacional de base de datos, en el apéndice B una breve introducción al manejador de base de datos y lenguaje de cuarta generación usados en el proyecto y en el apéndice C una descripción más detallada de la base de datos diseñada.

INTRODUCCION

1.1 Descripción del problema.

En los lugares con departamentos multidisciplinarios donde muchas veces se necesita compartir la información de uno con los demás, se suelen dar algunos de los problemas presentados a continuación.

- Núcleos de poder. Este problema se presenta cuando la gente se da cuenta que los demás necesitan los datos que uno tiene; entonces es común que estas personas empiecen a pensar que son los dueños de la información, lo cual ocasiona que ellos decidan a quien se la "prestan" de acuerdo a decisiones muy subjetivas.

- Información redundante e inconsistente. Este problema se presenta por varias razones: 1) Muchas veces cada departamento administra "su" información. 2) Hay mucha información que se captura más de una vez. 3) Cada quien hace las modificaciones que quiere a los datos para "adecuarlos" a sus necesidades. Cuando se dan estos problemas es muy común encontrar que existen los mismos datos con diferentes valores en varios lugares dentro del área.

- Información orientada a los sistemas existentes. Esto se da cuando no existe una independencia entre la información y los programas de aplicación. Muchas veces se diseñan las estructuras pensando en la forma como van a ser accesadas, lo cual ocasiona que éstas presenten muy poca flexibilidad cuando se necesita añadir nueva información al sistema.

- Cada departamento tiene sus propias estructuras de información, sus propios métodos para explotar dicha

información y sus propios programas de aplicación. Esto dificulta las cosas cuando los integrantes de otro departamento tratan de usar dicha información.

1.2 Solución propuesta.

Para tener un mejor control de la información y eliminar los problemas mencionados en la sección anterior, se propone la creación de un sistema de base de datos para el área de análisis bursátil con las siguientes características.

- Base de Datos común al área. Esto implica básicamente 2 cosas. 1) Una sola estructura de información para todos los departamentos dentro del área. 2) La creación de estándares para la forma de almacenar y explotar la información contenida dentro de la Base. Con este punto de diseño se eliminan varios problemas como son: 1) La dependencia del sistema a las personas y por lo tanto se eliminan o al menos se reducen los núcleos de poder. 2) Toda la gente autorizada podrá consultar de la misma manera tanto la información propia de su departamento como la de los demás, contenida dentro de la Base de Datos.

- Sin problemas de redundancia. La información estará almacenada en un solo lugar dentro de la base de datos, de tal forma que todas las personas que tengan la necesidad de consultar la base, obtendrán los mismos valores sin importar a que área pertenecen. Esto acaba con los problemas de redundancia, inconsistencia y reduce en gran parte los tiempos de captura de información.

- Obtención fácil de la información. Cada vez que se tiene que hacer un sistema para algún lugar donde sus necesidades de información son relativamente complejas, se tiene el peligro de caer en el error de diseñar algo que solamente el

creador o gente muy especializada va a poder entender y manejar. Esto trae consigo problemas como el "asustar" al usuario consiguiendo solamente que el sistema no sea usado, o dificultades para seleccionar y capacitar al personal que trabajará con él. Para solucionar este problema se pueden tomar decisiones como:

- Simplificar demasiado el modelado del sistema; esto tiene el problema de que al querer "forzar" la información para que "quepa" dentro de la estructura que diseñamos, tendremos muy probablemente un modelo equivocado o con muy poca flexibilidad para futuros ajustes.

- Crear una estructura tan sofisticada como se necesite, pero con una interfase al usuario lo suficientemente amigable como para que éste no tenga que aprender los detalles del sistema cuando quiera hacer una consulta no prevista en los programas. En nuestro caso optamos por éste último método.

- Flexible a nuevos requerimientos. Este es uno de los puntos más importantes en las consideraciones tomadas para el diseño del sistema ha desarrollar, ya que al tener que modelar un sistema tan dinámico como es el sistema financiero y bursátil de nuestro país, si no se crea una estructura muy flexible para poder modelar muchos tipos de información dentro de ella, en muy poco tiempo se tendrá una base de datos obsoleta o con una cantidad muy grande de "parches" para tratar de adaptarla a los nuevos instrumentos, nuevas políticas de operación, nuevos mercados, etcétera.

- Sin problemas de integridad. Este punto, a pesar de no ser un requerimiento normalmente pedido por los usuarios, tiene que ver con uno de los problemas más importantes que puedan

presentarse en la práctica si no es tomado con la debida importancia. Existen al menos dos tipos de integridad que se deben cuidar:

- Integridad a nivel entidad o tabla. En este punto se prohíbe que algún componente de la llave primaria de las tablas tenga valor nulo. Lo anterior podría llevar al caso de que un conjunto dado de valores dentro de la llave primaria identificara a más de un registro dentro de la tabla, lo cual contradice los fundamentos del modelo relacional de base de datos.

- Integridad referencial. Siempre que mediante la llave foránea de alguna tabla se haga referencia a otra tabla, es decir cuando la llave foránea de la primer tabla no contenga algún valor nulo, en esta última debe existir dicha referencia.

Para mayor información sobre este punto del diseño, consultar el apéndice A, en el cual se habla de los fundamentos de base de datos en general y en particular del modelo relacional de base de datos.

- Plataforma base para futuros desarrollos. El tener la información necesaria bien almacenada y organizada es un requisito forzoso para poder tener desarrollos adecuados de sistemas gráficos, bases de textos, sistemas expertos, etcétera.

- Seguridad de acceso. Aunque la mayor parte de la información necesaria dentro del área de análisis bursátil es información pública, es necesario mantener un cierto nivel de seguridad dentro de la base para evitar que entre, se modifique o sea borrada información de tal forma que afecte a las condiciones básicas de la base de datos (completa y confiable).

ANALISIS BURSATIL

Debido a que el objetivo principal del proyecto es diseñar una base de datos que sirva como plataforma base para el análisis bursátil, en este capítulo daremos las definiciones y explicaciones básicas del análisis bursátil así como de sus componentes principales: el análisis de mercado de dinero, el análisis técnico y el análisis fundamental. Al ir explicando cada uno de ellos iremos conociendo sus principales requerimientos de información, los cuales tomaremos como base para diseñar la estructura de la base de datos.

2.1 Definición.

Definiremos al análisis bursátil como aquella área de una institución financiera que tiene como propósito el estudiar todos aquellos factores legales, económicos, financieros y políticos que permiten explicar el comportamiento en el tiempo de los valores de una empresa o, en general, de todos aquellos instrumentos que se operan en los mercados financieros. El objetivo primario del analista es hacer la evaluación actual de dichos instrumentos y pronosticar su comportamiento en el futuro.

A pesar de que todas las áreas de un departamento de análisis bursátil tienen un mismo objetivo (el pronosticar que instrumento o instrumentos nos proporcionarán una mayor ganancia en un futuro determinado), cada una requiere de información completamente diferente en algunos casos para poder llevar a cabo su labor en forma eficiente.

Aunque cada empresa define generalmente los departamentos que quieren que su área de análisis tenga, podemos decir que

básicamente existen tres tipos de análisis bursátil: El análisis del mercado de dinero, el análisis técnico y el análisis fundamental.

A continuación explicaremos cada una de las áreas del análisis con la profundidad suficiente solamente para que se conozcan y entiendan sus necesidades tanto de información como de sistemas. Así mismo, conforme vayamos estudiando cada área, iremos describiendo las diferentes fuentes donde podemos obtener dicha información.

2.2 Análisis de mercado de dinero.

Definiremos como análisis de mercado de dinero a aquel que estudia los instrumentos que tienen generalmente tanto un plazo como una tasa de interés predefinidos. Como ejemplos de instrumentos de mercado de dinero tenemos los cetes, ajustabonos, etcétera.

Existen algunos instrumentos como son los pertenecientes al mercado de coberturas que aunque no siempre cumplen con la definición dada, por sus características generales de operación se consideran también como dentro de los instrumentos de mercado de dinero. Dentro de estos instrumentos están las divisas, los petrobonos, los pagafes, etcétera.

Generalmente los instrumentos de mercado de dinero son emitidos por el gobierno federal para captar fondos y operados a través del Banco de México mediante subastas en las cuales participan las instituciones financieras. Muchos de estos instrumentos operan a *descuento* (por debajo de su valor nominal) por lo que el interés para las instituciones financieras consiste en obtener el mayor descuento posible para después negociar con sus clientes los mismos

instrumentos a un descuento menor, quedándose con la diferencia. A pesar de la creencia de que la ganancia principal de las instituciones como las casas de bolsa se genera con la operación de acciones, en la realidad sucede que es con la operación en mercado de dinero que dichas empresas obtienen sus principales fuentes de ingresos; de aquí la importancia de este tipo de análisis.

Las características principales de estudio de los instrumentos de mercado de dinero, las cuales se deben ver reflejadas dentro de la base de datos, son los datos de la subasta (montos y tasas tanto en posturas como pactadas, número de postores y posturas, plazos de las emisiones), los datos de las operaciones del Banco de México con los diferentes intermediarios (operaciones en mercado secundario) como son reportos (cuando el cliente compra determinado instrumento y se compromete a venderlo en una fecha futura preestablecida.) y operaciones en directo, etcétera.

Las fuentes principales de información para los instrumentos de mercado de dinero son:

- Diario Oficial
- Periódicos
- Información telefónica
- Reuters
- Publicaciones del Banco de México
- Bases de datos del Banco de México
- Bolsa mexicana de valores

De los mencionados, algunos como Reuters y la Bolsa tienen disponible el servicio de transferencia de información por medios magnéticos, lo que en teoría reduce los problemas de errores al momento de la carga a la base de datos. En algunos otros casos, la información se debe seguir

capturando manualmente, como son los casos de la obtenida telefónicamente o por medio del periódico.

2.3 Análisis técnico.

Definiremos como análisis técnico al seguimiento constante de la oferta y la demanda de las acciones o cualquier otro instrumento del mercado de capitales (definidos como instrumentos sin plazo e interés predeterminados y emitidos casi siempre por las empresas). En este tipo de análisis se ignora la situación financiera y económica de la empresa emisora o del país, ya que el enfoque básico en este tipo de análisis parte del principio económico que la oferta y la demanda son los dos factores que determinan el precio de una acción dentro del mercado. El analista técnico parte del comportamiento de elementos tales como el precio, volumen y paquetes negociados en el piso de remates de la Bolsa. De acuerdo con los métodos utilizados por los analistas técnicos, un nombre que podría ser adecuado para este tipo de análisis es el de análisis estadístico y gráfico ya que utiliza muchos medios gráficos de reconocimiento de patrones gráficos para su estudio.

Los requerimientos básicos de información por parte del análisis técnico son los registros de pujas (ofrecimientos) y hechos (operaciones) realizados dentro del piso de remates.

Aunque dichos requerimientos pueden ser representados dentro de una base de datos con bastante facilidad, generan el principal problema en cuanto al volumen de información dentro de la base, ya que tomando solamente la información de los hechos, se generan de 30 a 40 MB por estos registros cada mes. Esto es especialmente importante en instituciones donde no se cuenta con una gran capacidad de almacenamiento

en disco. En el capítulo donde tocamos la implantación del sistema veremos este problema con un poco más de detalle.

Dentro de los requerimientos de sistemas por parte de este departamento, el más importante será probablemente la facilidad para llevar rápida y eficientemente la información desde la base de datos a los paquetes propios para el análisis como es el MetaStock. Dentro de las consultas básicas encontramos la de la operatividad de cierta emisora durante un período determinado, o la consulta histórica de los precios y volúmenes negociados para una acción en particular. En el momento de migrar a sistemas de quinta generación, será muy interesante simular los métodos de reconocimiento de patrones que utilizan estos analistas.

2.4 Análisis fundamental.

El análisis fundamental es aquel que se especializa en el estudio de la estructura y características de las empresas desde el punto de vista financiero y económico.

Desde el punto de vista cuantitativo, el analista fundamental debe conocer y estudiar los índices de crecimiento de las empresas, su rentabilidad, estructura de capital, flujo de efectivo, estructura del balance y en general, de todos aquellos aspectos cuantitativos que le ayuden a estimar el comportamiento de la empresa en el futuro, ya sea por su propio crecimiento o por su comparación con otras empresas. Como se puede observar, el análisis fundamental tendrá mejores resultados en el mediano y largo plazo, mientras que el técnico analiza mejor el comportamiento de las acciones dentro del corto plazo.

Desde el punto de vista cualitativo el analista fundamental deberá estudiar, además de las empresas, los aspectos políticos y macroeconómicos del país para poder analizar sus

impactos dentro de las empresas con respecto a la penetración y participación de mercado, administración, relaciones laborales, inversiones en tecnología y plantas, importaciones, exportaciones y en general, todos aquellos aspectos cualitativos que le permitan analizar mejor a la empresa con respecto a su medio ambiente.

El sistema presentado en este trabajo proveerá de información al analista fundamental para el estudio de las variables cuantitativas, ya que la representación en sistemas de la mayoría de las variables cualitativas presenta una estructura mucho menos formal, la cual es muy difícil además de poco eficiente de representar dentro de una base de datos. Es para el análisis cualitativo que hemos pensado migrar posteriormente a sistemas de bases de texto, sistemas expertos y en general a todos aquellos sistemas (muchos pertenecientes a ambientes de quinta generación) que nos permitan estudiar en forma adecuada la información no cuantitativa del mercado bursátil.

Los requerimientos principales de información por parte del análisis fundamental son los estados financieros que las empresas que cotizan en bolsa deben presentar a ella. De acuerdo a las diferentes características de las empresas, se tienen varios formatos contables dependiendo si la empresa es de tipo industrial, si es un banco, una aseguradora, una casa de bolsa o un grupo financiero. Como veremos, cada uno de los diferentes formatos contables tendrá una representación especial dentro de la base de datos. Otros requerimientos también básicos son los precios de las acciones y el sector al cual pertenece cada empresa.

Aunque el volumen de información requerida dentro del análisis fundamental no presenta los mismos problemas que el análisis técnico, se deberá tener también cuidado con su administración para cuidar el rendimiento del sistema.

Con respecto a los principales sistemas a desarrollar, se encuentran las consultas de los estados financieros, el reporte de múltiplos de las empresas (relaciones entre el precio de las acciones y el valor en libros o las utilidades esperadas por la compañía), reportes de crecimiento de las empresas, reportes comparativos para empresas dentro de un sector determinado, etcétera.

En el capítulo 7 veremos como ejemplo de aplicación, al sistema básico desarrollado para el área de análisis fundamental dentro de una casa de bolsa.

2.5 Conclusión.

Debemos recordar que aunque el trabajo y propósito de cada área dentro del análisis bursátil va mucho más allá de lo explicado, solamente se dieron los fundamentos necesarios para poder conocer y entender sus necesidades básicas de información y desarrollo de sistemas. Al ser el área de análisis un área de estudio, se deberá dejar al usuario la facilidad de consultar la base de datos de muchas formas diferentes a las establecidas con los programas básicos, por lo que será importante utilizar un lenguaje de fácil entendimiento para ser utilizado por el usuario final.

Como conclusión diremos que mientras el análisis fundamental estudia las empresas para saber que comprar y vender, el análisis técnico determina el momento adecuado para realizar la compra o venta del instrumento. Por otra parte, siempre se deberá tomar en cuenta el riesgo asociado dentro de este mercado y saber cuando y en que proporción estar en él y cuando estar en instrumentos con mucho menos riesgo (pero con menor ganancia potencial) como son los de mercado de dinero.

METODOLOGIA DE DESARROLLO

3.1 Ambientes de desarrollo de sistemas.

Un aspecto muy importante al empezar un diseño de sistemas es el decidir el ambiente en el que vamos a trabajar, ya que existen en el mercado una gran variedad de lenguajes, "paquetes" y en general diferentes ambientes para el desarrollo de aplicaciones. Podemos decir que hay al menos tres ambientes de desarrollo, los cuales podemos clasificar como de tercera, cuarta o quinta generación.

Un ambiente de tercera generación está enfocado a lo que se conoce como *procesamiento de datos* y en él predominan las fórmulas, algoritmos de cálculo, una gran cantidad de programadores desarrollando sistemas con muchas líneas de código, sistemas dependientes de las computadoras donde son desarrollados (es decir, sistemas poco portables en su mayoría), etcétera. Los lenguajes típicos de un ambiente de tercera generación son FORTRAN, COBOL, etcétera. Una característica de los sistemas desarrollados en este ambiente es que tienen una gran cantidad de líneas de código independientes del objetivo del sistema (Overhead). Un ejemplo de esto se da cuando tratamos de usar estos lenguajes para el manejo de grandes volúmenes de información almacenada en archivos indexados u otro medio semejante. En los programas hay mucha programación "de más" para el manejo de este tipo de información. Lo anterior lleva a que los lenguajes pertenecientes a este ambiente sean muy poco amigables para ser usados por los usuarios finales, ya que se requiere un nivel relativamente alto de especialidad para poder desarrollar sistemas en ellos.

Una alternativa muy interesante cuando queremos consultar grandes volúmenes de información se presenta con los

llamados lenguajes de cuarta generación. Las características básicas de un ambiente de cuarta generación son:

- Capacidad de consulta de la información por parte del usuario final sin requerir de algún sistema desarrollado previamente.

- Aumentar la productividad de los desarrolladores de aplicaciones. Esto se debe a que los programas desarrollados en este ambiente tienden a enfocarse en el problema que quieren resolver, disminuyendo el *overhead*.

- Disminuir el costo de mantenimiento a los sistemas. Debido a que hay menos líneas de código, al uso de programas que nos ayudan a diseñar y documentar nuestro sistema (herramientas CASE), al acceso a diccionarios de datos y a otros factores, el tiempo y por lo tanto el costo de mantenimiento a los sistemas desarrollados con lenguajes de cuarta generación se reduce de manera significativa.

- Debido a los grandes volúmenes de información que se manejan hoy en día, este ambiente está orientado al uso de bases de datos. Este punto los hace especialmente interesantes ya que tienen características internas específicas para el manejo de este tipo de estructuras de almacenamiento de información.

Aunque algunos lenguajes como LISP, propios de los ambientes de quinta generación fueron desarrollados hace ya varias décadas (en los 60's), la filosofía de los ambientes de quinta generación tienen relativamente poco tiempo en el mercado. La idea básica alrededor de la quinta generación es el desarrollar sistema que *simulen* de alguna manera a los procesos "*inteligentes*" realizados por el hombre; debido a

lo anterior a estos ambientes se les ha llamado también como lenguajes de *inteligencia artificial*.

Así como generalmente los lenguajes de tercera generación usan como entrada datos aislados y los de cuarta bases de datos, los ambientes de quinta generación utilizan como entrada de información lo que se conoce como bases de conocimiento, en las cuales como su nombre lo indica, no se almacenan datos sino *conocimiento* que será más tarde procesado por los sistemas. Muchos sistemas desarrollados con ambientes de quinta generación están basados a su vez en sistemas de cuarta generación, aprovechándose así tanto la infraestructura de información como la explotación de la misma.

En nuestro caso debido a las características del problema (mucha información, necesidad de que los usuarios pudieran desarrollar sus propias consultas, de tener un mantenimiento fácil de llevar, de tener información compartida, etcétera), optamos por desarrollar el sistema dentro de un ambiente de cuarta generación, sin perder de vista que posteriormente podríamos tomar la estructura del sistema y pasar a desarrollos en ambientes de quinta generación tomando como plataforma a la base de datos aquí desarrollada.

3.2 Metodologías de desarrollo para cuarta generación.

Ya dentro de un ambiente de cuarta generación, observamos que muchos de los diseños actuales de base de datos están basados casi exclusivamente en lo que se conoce como el *enfoque analítico* o de normalización, el cual por si solo para un sistema relativamente complejo, puede llevar a muchas confusiones debido a que va de lo particular (los datos) a lo general (las tablas o entidades), por lo que lo podríamos definir como un proceso *Bottom-Up*.

Otro método que puede ser usado es el que se conoce como *enfoque de síntesis*. Este modelo es recomendable cuando se está comenzando el diseño de la base de datos. Básicamente este método trata de decidir que entidades se tienen que representar dentro de la base, que relaciones existen entre dichas tablas, que propiedades tiene cada tabla y así hasta llegar al diseño detallado de la base de datos. La característica anterior lo define como un proceso *Top-Down*.

Las dos metodologías son complementarias entre sí y muy importantes en la práctica, por lo que para hacer el diseño lógico de la base de datos del trabajo aquí presentado nos basamos primero en el enfoque de síntesis, y una vez teniendo el detalle del diseño, aplicamos las técnicas del enfoque analítico para asegurarnos de que nuestro diseño no rompía con las reglas de normalización.

A continuación se describen los fundamentos tanto del enfoque de síntesis como los del analítico.

Cabe recordar que ninguno de los modelos pretende dar una teoría sobre el diseño de base de datos, sino que sólo dan guías que cada quien debe adaptar a la hora de hacer un diseño en particular.

3.3 Enfoque de síntesis.

El enfoque de síntesis nace a partir del concepto de entidad, a la que podemos definir como cualquier objeto que queramos representar dentro de la base de datos. Como ejemplos de entidades podemos mencionar a las divisas, las empresas, las acciones de las empresas dentro de la bolsa, etcétera.

Tomaremos las siguientes estructuras para los ejemplos usados en el resto del capítulo.

DEPARTAMENTO (#DEP, NOMBRE-DEP, #EMP-JEFE, PRESUPUESTO)

EMPLEADO (#EMP, NOMBRE-EMP, #DEP, SUELDO, BONO)

MATERIA (#MATERIA, NOMBRE-MAT)

HORARIO (#MATERIA, #CURSO, FECHA, LUGAR)

INSCRITOS (#MATERIA, #EMP, #CURSO, CALIFICACION)

PROVEEDOR (#PROV, NOMBRE-PROV, CIUDAD, NIVEL)

PARTE (#PARTE, NOMBRE-PARTE, COLOR, PESO, CIUDAD)

ENVIOS (#PROV, #PARTE, CANTIDAD)

Como ya se dijo, el enfoque de síntesis cae dentro de los procesos llamados *Top-Down*. Los pasos a seguir para el diseño de una Base de Datos usando el enfoque de síntesis son:

1. Identificar las entidades *Kernel*.
2. Identificar las *Asociaciones* entre entidades.
3. Identificar los *Señalamientos* entre entidades.
4. Identificar las entidades de *Características* y las propiedades de cada entidad.

A continuación se describe cada paso del diseño.

3.3.1 Entidades tipo Kernel.

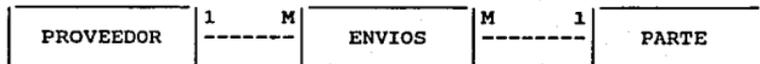
Este tipo de entidades forman el núcleo de la Base de Datos. Cada una de las entidades *kernel* tendrá una existencia propia totalmente independiente de la existencia de las demás entidades dentro de la base de datos. Como ejemplos de entidades *Kernel* tenemos a las tablas de DEPARTAMENTO, EMPLEADO, MATERIA, PROVEEDOR y PARTE.

En la práctica, el punto más importante de cualquier entidad (o de su representación física en forma de tabla) consiste en la forma en que se puede diferenciar una instancia de la entidad de las demás. Dentro del modelo relacional esta distinción entre instancias de entidades se da mediante lo que se conoce como *llave primaria*. Aunque en el apéndice A se dan las principales definiciones de la teoría de bases de datos, podemos decir informalmente que la llave primaria de una tabla está formada por un campo o conjunto de campos de la tabla que pueden ser usados como identificador único para cualquier registro de dicha tabla.

Una característica de la llave primaria que se desprende de la definición anterior es que ésta no puede ser nula, es decir que ninguno de los campos que la forman puede admitir algún valor nulo. Lo anterior es cierto ya que si alguno de los campos de la llave primaria admitiera valores nulos, la llave podría representar a más de un registro dentro de la tabla, lo cual contradice su propia definición.

3.3.2 Asociaciones entre entidades.

Llamaremos entidades de asociación a aquellas que sirven para representar las relaciones "muchos a muchos" entre entidades. Como ejemplo de entidades de asociación tenemos la tabla de ENVIOS.



Nota 1. En las frases "muchos a uno" y "uno a muchos", "muchos" incluye las posibilidades de uno y cero, en cambio "uno" será tomado como "exactamente uno".

Nota 2. Dentro del diagrama anterior, las líneas que conectan a las entidades representan lo que en el siguiente punto del diseño llamaremos *designaciones*. Siempre podremos considerar una asociación como la combinación de al menos dos designaciones, generalmente con la restricción de que para dos registros diferentes de una asociación no pueden darse los mismos valores de las designaciones que forman dicha asociación. Por ejemplo, dos registros de ENVIOS no pueden tener al mismo PROVEEDOR y a las mismas PARTES involucradas.

El punto más importante de una asociación es el identificar las entidades asociadas, es decir a los participantes de la asociación. En el modelo relacional, los participantes de una asociación están definidos con lo que se conoce como *llave foránea* (Foreign Key en su término en inglés). Podemos definir a la llave foránea como el campo o grupo de campos de una tabla cuyos valores identifican a la llave primaria de alguna otra tabla, la cual puede ser en algunos casos la tabla misma.

En el momento de asignar las llaves foráneas de una tabla existen preguntas que debemos contestar como:

1.- ¿ Puede la llave foránea admitir valores nulos ?.

La respuesta a esta pregunta no depende de la persona encargada de hacer el diseño de la base de datos, sino del modelo que se está representando. En el caso de la tabla de ENVIOS, es claro que siempre debe existir tanto el PROVEEDOR como la parte que se envía; en otros casos como el de la tabla de empleados, se podría dar el caso de que existiera un empleado al que no se le hubiera asignado todavía a un departamento. Podemos decir entonces como regla que dada una llave foránea, cada valor de ella es Nulo o apunta a algún valor de la llave primaria correspondiente en otra tabla.

2.- ¿ Qué debe suceder cuando se intenta borrar un registro de una tabla que tenga llaves foráneas en otras tablas ?.

Un ejemplo sería el tratar de borrar un departamento de la tabla de DEPARTAMENTOS cuando existen empleados asociados a él. En general existen 3 posibilidades de acción:

- * Borrar en cascada.
- * Negar el acceso.
- * Borrar el registro y ponerle valor nulo a los datos en la llave foránea. Esto solamente es válido cuando dicha llave foránea admite valores nulos.

3.- ¿ Qué sucede cuando se trata de actualizar la llave primaria de una tabla que tiene llaves foráneas asociadas ?

Como en el caso anterior existen las mismas tres posibilidades :

- * Actualizar en cascada.
- * Negar el acceso.
- * Dejar primero nulos los valores de las llaves foráneas y luego proceder a actualizar la llave primaria. Es muy importante el orden seguido en este último punto para evitar correr el peligro de atentar contra la integridad referencial de la tabla.

Como cada entidad tipo Asociación se representa dentro de la base como una tabla, debe tener su llave primaria para identificar a sus registros. Del mismo modo puede tener sus *Propiedades* y *Características* e incluso puede estar involucrada dentro de otras Asociaciones.

3.3.3 Señalamientos entre entidades.

Sin importar la clasificación que tenga, cualquier entidad puede *señalar* o *apuntar* a otras entidades; esto se da cuando existe una relación de "muchos a uno" entre las dos entidades. Como ejemplo tenemos las tablas de empleados y departamentos, donde cada uno de los registros de la tabla de empleados *apunta* a un solo departamento.

3.3.4 Características y Propiedades.

Llamamos *característica* (por su nombre en inglés) a aquellas entidades cuya única función es describir a alguna otra entidad. Un ejemplo se presenta en la relación INSCRITOS, ya que por cada curso ofrecido existen muchos (aceptándose uno o cero) inscritos en él. Las llaves foráneas para el caso de *características* no deben permitir valores nulos, se deben borrar en cascada y actualizarse también en cascada. La razón de lo anterior es que una *característica* no puede tener una referencia nula a aquella entidad que está describiendo.

Cada entidad, sin importar su clasificación, tendrá un número de *propiedades*. Una propiedad puede servir para identificar la entidad (los campos de la llave primaria), para apuntar a otra entidad (los campos de la llave primaria), o simplemente para mostrar una de las características de la entidad que estamos representando. Como ejemplos de propiedades tenemos : 1) El número de empleado en la tabla EMPLEADO; en este caso la propiedad identifica a cada uno de los registros de la entidad. 2) El número de departamento a la cual pertenece el empleado; aquí la propiedad apunta a otra tabla. 3) Como propiedades que están "por si solas" tenemos el nombre del empleado, su número de seguro social, etcétera. Así como las *entidades*

se representan como *tablas* en la base de datos, las *propiedades* se representan como campos dentro de estas tablas.

Existen cuatro tipos de *propiedades* que son:

1.- Inmediatas, Univaluadas.

Una propiedad es *inmediata* (por su título en inglés) cuando representa una propiedad inherente de la entidad en cuestión. Como ejemplo diremos que el color de parte es una propiedad inmediata de la tabla PARTE y no de EMPLEADO.

Una propiedad es *univaluada* si cada instancia de la entidad que describe tiene solamente un valor de dicha propiedad. El peso de la parte es univaluada en PARTE, pero su color no, ya que una parte puede venir en varios colores.

Este tipo de propiedades se representan como campos dentro de la tabla a la cual pertenecen.

2.- No inmediatas, univaluadas.

Una propiedad *no inmediata* no representa una propiedad inherente de la entidad. Cuando se tiene este caso sin tener alguna otra tabla en la cual dicha propiedad sea inmediata, significa que deberemos crear dicha entidad. Como ejemplo supondremos que el bono de los empleados depende solamente del departamento en el que se encuentra dicho empleado y no del empleado mismo; en este caso tendríamos que quitar el campo de EMPLEADO y pasarlo a DEPARTAMENTO donde sí es inmediato.

3.- Inmediatas, Multivaluadas.

Una propiedad es *multivaluada* en una entidad cuando para cada instancia de la entidad existe más de un valor de dicha propiedad. Un ejemplo se da si las

partes en la tabla PARTE pueden venir en varios colores. En estos casos dichas propiedades formaran parte de tablas separadas, características de la entidad en cuestión.

4.- No inmediatas, multivaluadas.

Este caso es una combinación de los anteriores y por lo tanto dicha propiedad deberá ser representada dentro de otra tabla. En el caso de la tabla MATERIA, existe más de un lugar donde se puede impartir y no es una característica propia de MATERIA.

3.4 El enfoque analítico.

El enfoque analítico, conocido como enfoque de normalización, ha sido ampliamente explicado en muchos libros y artículos, por lo que aquí sólo se dará una pequeña introducción.

El enfoque analítico parte del principio de que una tabla normalizada será aquella que satisface la condición que para cualquier posición renglón-columna dentro de ella existirá siempre un solo valor y no un conjunto de valores. En el caso de sistemas de bases de datos relacionales, sólo se permiten tener tablas normalizadas.

Cualquier tabla normalizada en el sentido descrito se considerará como normalizada en primera forma normal 1NF. Es posible definir diferentes niveles de normalización donde cada nivel estará en el nivel anterior además de poseer una restricción adicional. Esto significa en teoría que cada nivel superior de normalización será más deseable que el anterior ya que tendrá menos probabilidades de generar errores en la base de datos.

El objetivo último de la normalización es eliminar la redundancia de la base de datos. Este enfoque da los medios para descomponer tablas con redundancia en tablas más pequeñas hasta llegar al punto de que cada valor aparezca una sola vez en un solo lugar dentro de la base. Como veremos más adelante, esto no siempre será conveniente en la práctica por razones de rendimiento del sistema. En otras palabras, el objetivo de la normalización será llegar a tener un diseño final con tablas consistentes con una llave primaria que identifique a la relación y cero o más campos mutuamente independientes, cada uno representando una propiedad inmediata univaluada de la entidad representada por la llave primaria y cuya única dependencia sea a esta llave primaria.

En un sentido, la normalización representa un punto a considerar en el rendimiento del sistema referente a la escritura y a la lectura de la información; en otras palabras diremos que *siempre* queremos deshacernos de la redundancia *dañina* en la base, no de cualquier redundancia ya que si en la práctica hay dos o más datos que son leídos juntos en forma frecuente y que serían separados en un proceso de normalización total, se deberá de usar un poco de los que se conoce como *desnormalización* para evitar problemas posteriores en el rendimiento del sistema. Sin embargo siempre es bueno documentar en estos casos la razón por la cual se llegó a dicha desnormalización.

El enfoque analítico involucra el descomponer tablas grandes en pequeñas para evitar o reducir la redundancia, pero nunca explica como llegar a esas "tablas grandes". El enfoque de síntesis ataca precisamente dicho problema, por lo que en la práctica se recomienda que primero se use el enfoque de síntesis para generar las tablas y después, usando el enfoque analítico, comprobar que dichas tablas no violan de una manera sin control las reglas de normalización.

DISEÑO LOGICO DE LA BASE DE DATOS

4.1 Ideas alrededor de la base de datos.

Siempre que se tiene que hacer un diseño nuevo de base de datos se tienen que tomar en cuenta factores que van más allá del modelado del sistema que nos interesa. En nuestro caso, al momento de diseñar la base de datos para el análisis financiero y bursátil de nuestro país, tuvimos que considerar además de los datos puros, que nuestro sistema financiero es muy cambiante y que la base de datos resultante será consultada por gente de diversas especialidades (mercado de dinero, análisis técnico, fundamental, etcétera). Lo anterior nos obliga a crear una base de datos con una estructura lo suficientemente flexible para soportar tanto los cambios que se den dentro del ámbito financiero como las diferentes estructuras de información que necesita cada una de las áreas mencionadas. Dadas estas características consideramos que el núcleo de la estructura de la base de datos debe estar formado alrededor de los conceptos de *objetos* y *grupos* que a continuación se explican.

4.1.1 *Objetos*. Consideraremos como objeto a toda *variable* financiera o bursátil con ciertas características comunes (todos se pueden operar, tienen un precio, etcétera) y sobre la cual queremos llevar un registro dentro de la base de datos.

Cabe mencionar que aunque el concepto de *objeto* dentro de la base de datos tiene algunas características de la "*programación orientada a objetos*", la forma de utilización de ambos conceptos difiere substancialmente, por lo que no se deberán confundir.

Como ejemplos de objetos tenemos las acciones de las empresas, las divisas, los metales, los valores del gobierno como cetes y pagafes, los barriles de petroleo, etcétera. Aunque todos los objetos tienen algunas características en común, es importante anotar que también pueden ser divididos en *tipos de objetos* de acuerdo a ciertos criterios. Existen algunas características que son propias a determinados tipos de objeto como son: si cotiza o no, en que bolsa de valores cotiza, fecha de emisión, fecha de vencimiento, etcétera.

4.1.2 Grupos. Las acciones de las sociedades de inversión están constituidas por una serie de objetos como son acciones de empresas, cetes, ajustabonos, etcétera. Para estas acciones compuestas también aplican las características comunes mencionadas anteriormente para los objetos ya que también tienen un precio, se pueden operar, etcétera. Por casos como el anterior se crea la noción de *grupo*, al cual definiremos como la unión de 2 o más objetos con una característica especial entre ellos. Nótese que los grupos también pueden pertenecer a grupos de un nivel superior lo que permitirá en un momento dado hacer análisis con diferentes grados de profundidad; otra ventaja de la creación de grupos es que podemos formar uno conjuntando varios objetos de acuerdo a un cierto criterio y tratarlos como una sola entidad; gracias a ello, tenemos la flexibilidad de estudiar a las empresas como una sola unidad (a través de un grupo) o a cada una de las acciones que conforman dicha empresa.

Como ejemplos de grupos tenemos los fideicomisos, las sociedades de inversión tanto de renta fija como de renta variable, las agrupaciones de empresas, el agrupar los objetos de acuerdo al mercado al cual pertenecen, etcétera.

Una vez explicados los conceptos básicos dentro de la estructura de la base de datos, procedamos a explicar a grosso modo cada una de las tablas que la constituyen.

4.2 Descripción de la base de datos a nivel de tablas.

En la figura de la página siguiente está el diagrama de relaciones de la base de datos diseñada; en dicha figura se observa claramente que la tabla principal de la base de datos es la de objeto, la cual está ligada a casi todas las demás tablas. La explicación que daremos de las tablas que conforman la estructura de la base de datos, así como del contenido de cada una de las tablas, tomará en cuenta los conceptos de metodología para el diseño de base de datos citados en el capítulo tres.

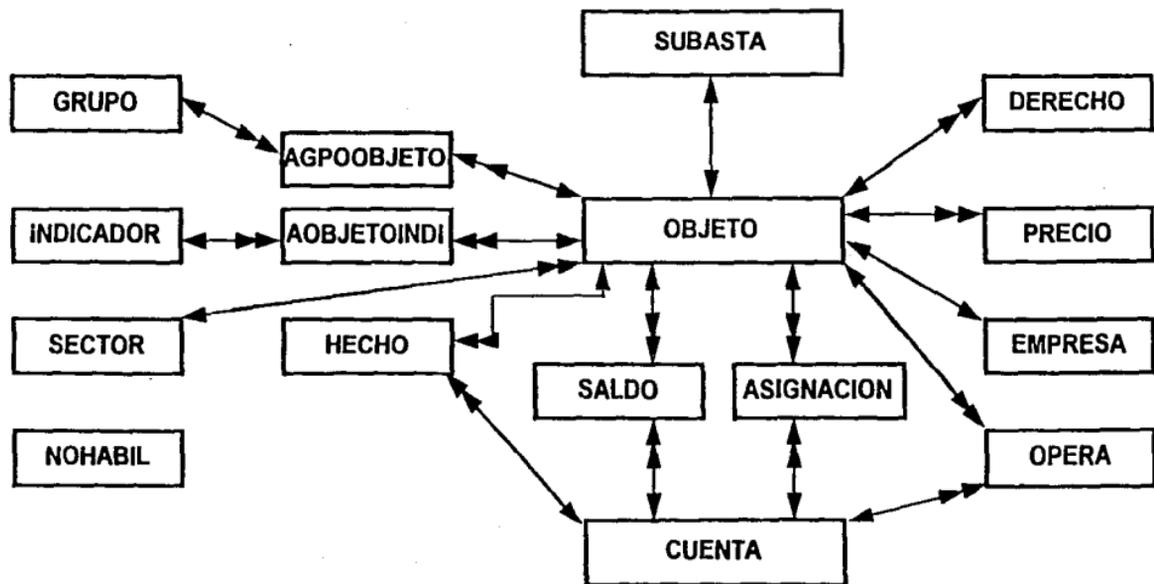
4.2.1 Entidades tipo Kernel. La característica principal de estas entidades es que su existencia no depende de otra tabla; estas tablas por lo general forman el centro de la estructura de la base de datos.

Tabla de cuenta. Esta tabla contiene la clave, el nombre y la descripción de todos aquellos entes que pueden operar a los objetos. Como ejemplos de cuentas están el Banco de México, cada una de las casas de bolsa, los bancos, etcétera. La llave primaria para esta tabla será la clave de cada cuenta.

Tabla de grupo. Aquí se almacenan las claves, nombres y criterios de asociación para formar los diferentes grupos. La llave primaria en este caso es la clave del grupo.

Tabla de empresa. Cada trimestre las empresas que cotizan en la bolsa mexicana de valores tienen la

Diagrama de Relaciones de la B.D.



obligación de enviarle un reporte contable de acuerdo a un formato que esta institución define. La tabla de empresa contiene un campo por cada rubro contable definido en este reporte, así como algunos campos de control para uso interno. Es importante notar que esta tabla sólo tiene los campos contables requeridos a las empresas industriales, comerciales y de servicios, y que habrían que crearse estructuras semejantes para cada una de las instituciones financieras que cotizan en bolsa como: aseguradoras, casas de bolsa, bancos y grupos financieros. Dichas estructuras no están contempladas en este trabajo, ya que su estructura puede ser inferida por el análisis de la tabla de empresa y el formato contable que aplique en cada caso. En el caso de esta tabla, su llave primaria se encuentra compuesta por el nombre de la emisora, el año y el trimestre reportado.

Tabla de indicador. En esta tabla se encuentra un catálogo de indicadores financieros y bursátiles que están relacionados con los distintos objetos. Los datos que se almacenan por cada indicador son su clave, su nombre y una descripción. Como ejemplos de indicadores, tenemos al índice de bursatilidad de las acciones (que tanto se opera cada una de ellas), la relación de precio/utilidad, etcétera.

Tabla de objeto. Esta tabla junto con la de grupo forma el núcleo de la estructura. Aquí se almacena un registro por cada objeto creado dentro de la base de datos. Sus campos principales son la clave del objeto, su nombre, la emisora y serie que lo emiten, el sector al cual pertenece (esta tabla señala a la tabla de sectores), el tipo de objeto (acción industrial, acción financiera, CETE, divisa, etcétera), su fecha de emisión, su fecha de vencimiento, si cotiza o no, una

referencia a la tabla de opera (ver definición de opera), etcétera.

Cada tipo de objeto tiene valores sólo para un determinado número de campos de los existentes en esta tabla, dejando valores nulos a los demás campos. Como ejemplo de lo anterior tenemos a las divisas, las cuales no tienen una fecha de emisión ni de vencimiento. La llave primaria para la tabla de objeto está constituida por la clave del objeto.

Tabla de sector. Existe dentro de esta tabla un registro por cada uno de los sectores que son de interés para el análisis bursátil. Los registros almacenados están constituidos por la clave, el nombre y la descripción de cada uno de los sectores. La llave primaria está formada por la clave de sector.

Como ejemplos de sectores tenemos el sector comercial, el sector de alimentos, bebidas y tabaco y al sector construcción.

Tabla de nohabil. Aquí se encuentran todas las fechas no hábiles de acuerdo a la comisión nacional bancaria y de seguros, ya que la bolsa de valores y las casas de bolsa en general siguen el calendario tomado por los bancos. En esta entidad sólo están considerados los días festivos, sin incluir sábados y domingos, ya que existen otras maneras de averiguar si una fecha dada cae en alguno de estos días. Esta tabla es usada principalmente para saber si la fecha de emisión o de vencimiento de algún instrumento cae en día festivo y que por lo tanto se tenga que hacer alguna adecuación a dicha fecha. Sólo existe un campo dentro de esta tabla el cual formará necesariamente la llave primaria.

4.2.2 Entidades de Asociación. La característica que comparten todas las entidades de asociación es la de unir a dos o más entidades que tienen una relación de "muchos a muchos".

Tabla de agpoobjeto. Esta tabla se encarga de manejar la relación existente entre las tablas de grupo y de objeto. Aquí se indica la constitución de objetos dentro de los grupos; encontramos en esta tabla por ejemplo, cuales instrumentos constituyen cada una de las sociedades de inversión de la casa a una fecha dada, las acciones que forman los diferentes fideicomisos, etcétera. Los campos que constituyen esta tabla son la clave del grupo, la clave del objeto, la indicación de si este objeto es a la vez un grupo, la fecha de alta y la fecha de baja de la relación. La llave primaria de esta tabla está compuesta por la clave del grupo, la del objeto y la fecha de alta.

Tabla aobjetoindi. Este tabla lleva una relación de cuales objetos son afectados por los diferentes indicadores. Sus campos son las claves del indicador y del objeto, la indicación de si dicho objeto es a su vez un grupo y el valor del indicador para dicho objeto. Tomemos como ejemplo al caso del índice de bursatilidad; este índice lo otorga mensualmente la bolsa mexicana de valores a las diferentes acciones que cotizan en ella de acuerdo a que tanto se haya operado dicha acción durante el mes. Los valores que puede tomar el índice son números enteros con dos cifras decimales, siendo el cero y el diez los límites asintóticos de este indicador. Si le damos al índice de bursatilidad una clave de 1 dentro de la tabla de índices, entonces podríamos decir dentro de la tabla de aobjetoindi que una acción X tiene un valor de Y dentro del indicador 1 para una fecha determinada. Cabe hacer

notar que no todos los objetos tendrán valor dentro de esta tabla para todos los indicadores; por ejemplo, no existen índices de bursatilidad para valores del gobierno, ni para las divisas.

Tabla de asignación. Si bien la estructura básica de la tabla de asignación está tomada de la tabla de saldo (siguiente definición), la tabla de asignación se usa específicamente para lo que se conoce como análisis de mercado de dinero. Dentro de esta tabla podemos consultar las asignaciones de valores gubernamentales que hace el Banco de México a los diferentes intermediarios financieros (bancos y casas de bolsa). Los campos básicos dentro de esta tabla son la clave de asignación (llave primaria), la clave de la cuenta asignada, la del objeto asignado, las tasas y montos solicitados y asignados, y el número de posturas y postores tanto de solicitud como de asignación dentro de la subasta.

Tabla de saldo. La tabla de saldo está formada por la relación de muchos a muchos entre las tablas de objeto y cuenta, contestando básicamente a la pregunta "quién (cuenta) tiene qué (objeto) ?.". Esta es una tabla mucho más general que la de asignación, ya que dentro de sus campos sólo están la clave de la cuenta, la del objeto, la fecha del saldo, el monto y las unidades de dicho monto. Como ejemplo podemos tomar al de las sociedades de inversión y almacenar en esta tabla el valor detallado de la composición de dicha sociedad. A primera vista podría parecer que existe alguna redundancia entre esta tabla y la de agpoobjeto, pero no es así, ya que la tabla de agpoobjeto sólo dice como está la composición de objetos dentro de un grupo (ejm. el grupo 'A' contiene a los objetos 'a', 'b' y 'c') mientras que la de saldo indica por fecha el valor

desglosado de dicho grupo (ejm. para un día x, el grupo 'A' tiene tantas acciones de 'a', tantas de 'b' y tantas de 'c'). Cada grupo puede ser un fondo, un fideicomiso, una sociedad de inversión, etcétera.

4.2.3 Entidades de características. La función de estas tablas es la de proporcionar mayor información sobre alguna otra tabla o sobre los componentes de dicha tabla.

Tabla de derecho. Cada determinado tiempo las empresas decretan lo que se conoce como dividendos o derechos sobre sus acciones. Como ejemplo de estos derechos tenemos a las empresas que dan a sus accionistas cierta cantidad de dinero por cada acción que tengan (dividendo en efectivo) o cuando por alguna razón multiplican el número de acciones en circulación (split), teniendo por consiguiente que modificar el precio de dichas acciones de forma tal que el valor total de acciones en el mercado no varíe. Cuando se da este tipo de derechos se tienen que ajustar los valores de los precios de las acciones para que tomen en cuenta dicho cambio. Dentro de la tabla de derecho se guardan los datos necesarios para que se puedan ejercer en forma automática dichos ajustes a las acciones y para poder seguir los diferentes derechos o dividendos que ha dado una empresa durante cierto período de tiempo a sus accionistas. Esta tabla apunta a la tabla de objeto.

Tabla de hecho. En esta tabla se guardan los registros de las operaciones (hechos) de mercado de capitales que se dan en el piso de remates de la Bolsa Mexicana de Valores; sus campos principales son la fecha, la hora, el comprador, el vendedor, el objeto comprado, el precio de la operación, el folio de dicha operación y

el tipo de valor negociado; aunque esta tabla así como la de opera apuntan tanto a la tabla de objeto como a la de cuenta, se consideran más como entidades de característica que como entidades de asociación, ya que su función básica no es la de resolver el problema de la relación muchos a muchos entre esas dos tablas.

Tabla de opera. En opera se guardan los registros de todas las operaciones que nos puedan interesar, exceptuando los hechos de mercado de capitales que se almacenan en la tabla de hecho. A pesar de que las estructuras de estas dos entidades son muy parecidas, se utilizan tablas diferentes dado que en la tabla de hechos se genera el mayor volumen de información de toda la base de datos y manejándola por separado se facilita su mantenimiento. Considerando que cualquier operación se puede descomponer en la unión de una (caso trivial) o más operaciones sencillas, la llave primaria de la tabla de opera se encuentra formada por un número de referencia de la operación (iref) y un número de secuencia (isec) dentro de dicha referencia; por ejemplo, una operación típica de lo que se conoce en el ámbito de mercado de dinero como *reporto* podría tener cierto número de referencia (1000) y estar formada por un depósito (secuencia 1), una compra de títulos (secuencia 2), una venta de los títulos en una fecha posterior (secuencia 3) y un retiro por el valor de la venta (secuencia 4). Para cierto tipo de *Obligaciones* se conoce desde el principio el número y las fechas en que dicho instrumento va a ir amortizando; para casos como estos se puso dentro de la tabla de objeto un número de referencia (iref) a una o varias operaciones que se tendrán que dar con dicho objeto.

Tabla de precio. En esta tabla se guardan las cotizaciones o precios de objetos para una fecha dada. Sus campos principales son la fecha de precio, la clave

del objeto y los precios de compra y de venta de dicho objeto. La razón para poner tanto el precio de compra como el de venta dentro de la tabla se debe a que algunos objetos como las divisas tienen para cada día tanto una cotización de compra como una de venta. La llave primaria de esta tabla está formada por la fecha de cotización y la clave del objeto. En el caso de esta tabla es claro que señala a la tabla de objeto.

Tabla de subasta. Cada semana el banco de México realiza subastas de los diferentes valores gubernamentales como son el Cete, el Pagafe, el Ajustabono, el Bonde, etcétera. Dentro de la convocatoria de dicha subasta se anuncia el objeto a subastar, el monto que se pretende subastar y la tasa (o precio según el caso) aproximada a la cual se hará la subasta. Con estos datos cada intermediario financiero hace una petición de acuerdo al monto y tasa que les interesa. Una vez que el Banco de México (Banxico) tiene todas las propuestas en su poder procede a hacer las asignaciones de acuerdo a ciertas políticas económicas y financieras; después de esto le avisa a cada intermediario por separado y en secreto cuanto le fue asignado de cada instrumento solicitado y a que tasa o precio se dio dicha asignación. Por último Banxico da a conocer los resultados globales de la subasta, anunciando los diferentes montos y tasas tanto solicitados como asignados por grupos de instituciones (bancos casas de bolsa, aseguradoras) sin dar el detalle asignado a cada institución. La tabla de subasta contiene la información global tanto de la solicitud como de la asignación dentro de la subasta. La información por rango de tasas dada a conocer es almacenada en la tabla de asignación. Como un objeto no puede ser subastado o emitido más de una vez, la clave de objeto dentro de la tabla de subasta se convierte en

su llave primaria. Es claro que esta tabla apunta a la tabla de objeto, ya que por cada registro de subasta existe un objeto que se está subastando y por lo tanto debe existir un registro de éste dentro de la tabla objeto; sin embargo, sólo determinado tipo de objetos (valores del gobierno) tendrán un registro asociado dentro de la tabla de subasta.

Las definiciones de las tablas que acabamos de dar pretenden sólo explicar la estructura global de la base de datos; en el apéndice C se da una descripción más detallada de la estructura de la base de datos, ya que ahí se presenta la descripción de la base a nivel de campos incluyendo además los índices creados para cada tabla.

DISEÑO FÍSICO DE LA BASE DE DATOS.

En el capítulo anterior hablamos del diseño lógico de la base de datos; este diseño sólo considera la estructura lógica de almacenamiento de la información, sin tomar en cuenta la forma como ésta va a ser accesada; sin embargo, en la práctica es importante analizar los distintos caminos de acceso a los datos para elegir el más eficiente.

Para lograr lo anterior se tienen que tomar en cuenta factores como son la cantidad de redundancia permitida y por lo tanto el nivel de desnormalización dentro de la base de datos, las políticas de indexación, etcétera. Dentro del diseño físico de la base de datos consideraremos estos y algunos otros factores que afectan el rendimiento (Performance en su término en inglés) de sistema.

5.1. Desnormalización.

Como se habló en el capítulo 3, no podemos considerar a la redundancia como un factor completamente indeseable dentro de la base de datos, ya que si por una parte nos genera muchos problemas al momento de las actualizaciones, por la otra nos facilita las consultas a los datos (suponiendo que tenemos eliminada la inconsistencia de información); por lo tanto siempre trataremos de reducir sólo la redundancia dañina al sistema y no toda la redundancia.

Tomando como referencia lo dicho anteriormente, creamos la política de que si el proceso normal de normalización mantiene separados a datos que se consultan juntos frecuentemente y no se actualizan con frecuencia, aplicaremos entonces cierta desnormalización para agilizar los procesos de consulta de dichos datos.

Como ejemplos de situaciones donde efectuamos desnormalización en la estructura tenemos a las tablas de **derecho**, **opera** y **hecho**.

En la tabla de **derecho** se utilizan tanto los nombres de las emisoras y series anteriores como las posteriores (Por ejemplo, un cambio de serie de Telmex *A a Telmex *B).

Tomando en consideración sólo el diseño lógico, deberíamos de haber utilizado las claves de los objetos equivalentes en vez de utilizar los nombres de las emisora-serie, pero entonces a la hora de las consultas se tendrían que hacer varios accesos (Al menos uno a la tabla de **derecho** y dos a la de **objeto**) para consultar cada derecho; en vez de lo anterior pusimos los nombres de las emisora-serie con la restricción en los campos de que deberían existir las referencias equivalentes dentro de la tabla de **objeto**.

Dentro de la tabla de **opera** existe tanto la clave del objeto (**iobjeto1**) operado como su nombre (**nobjeto1**), de tal forma que al momento de consultar las operaciones no se tenga que hacer un acceso a la tabla de **objeto** para saber que es lo que se está operando. Al igual que en la tabla de **derecho** se tiene, dentro del campo del nombre del objeto (**nobjeto1**), la restricción de que deberá de existir la referencia al objeto asociado dentro de la tabla de **objeto** para permitir la creación del registro en **opera**.

Si analizamos desde el punto de vista lógico las tablas de **hecho** y **opera**, nos daremos cuenta que tienen una estructura muy parecida y que añadiendo uno o dos campos a la tabla de **opera** nos podríamos "evitar" la de **hecho**; sin embargo, al tomar en cuenta el volumen generado por la información dentro de cada tabla decidimos separarlas para facilitar el mantenimiento de la base (ver capítulo de implantación).

5.2. Indexación.

Probablemente el aspecto más importante a considerar al momento del diseño físico de una base de datos es el de definir las políticas de indexación dentro de la base, ya que dependiendo de estas políticas se tendrá un buen o mal rendimiento en las consultas y actualizaciones de la base de datos.

Definimos los índices como archivos en los cuales cada registro consiste de algunos datos más uno o varios apuntadores. Los datos contienen valores de campos del archivo que se está indexando y los apuntadores contienen referencias a los registros de dicho archivo.

El propósito de los índices es el de proveer de métodos de acceso a los registros del archivo indexado. Los índices permiten dos formas de acceder a la información de dicho archivo: Un acceso secuencial mediante el recorrido de los valores dentro del índice y un acceso directo cuando se proporcionan todos los valores dentro del índice para apuntar a un solo registro del archivo.

Podemos clasificar la forma de indexar los archivos en dos grupos: Cuando se indexa el archivo mediante su llave primaria se genera lo que se conoce como índice primario, los cuales nunca pueden ser repetidos. Cuando se crean índices basados en otros campos que no son la llave primaria se llaman índices secundarios; estos índices pueden o no ser repetidos.

En la práctica todas las tablas deben contener al menos su índice primario asociado; sin embargo, es frecuente la creación de uno o más índices para agilizar los procesos de consulta. Esto es especialmente cierto en lugares donde no son muy frecuentes las actualizaciones de la información y

donde hay muchas consultas de diversos tipos a la base de datos.

A continuación presentamos los índices principales de cada tabla de la base de datos, donde P = índice primario, S = índice secundario, U = índice único y M = índice múltiple (no único).

- Agpoobjeto:

- * clave del grupo, del objeto, fecha de alta (P)
- * clave del objeto, del grupo, fecha de alta (S,U)

- Aobjetoindi:

- * clave del objeto, del indicador, fecha de alta (P)
- * clave del indicador, del objeto, fecha de alta (S,U)

- Asignación:

- * Clave de la asignación (P)
- * Clave de la cuenta (S,M)
- * Clave de la emisora, de la serie (S,M)
- * Clave del objeto (S,M)

- Cuenta:

- * Clave de la cuenta (P)
- * Nombre de la cuenta (S,M)

- Derecho:

- * Clave de referencia, de secuencia (P)
- * Fecha del derecho, emisora-serie anterior, cupón que se paga, tipo de derecho (S,M)
- * Emisora anterior (S,M)

- Empresa:

- * Emisora, año, número de trimestre (P)
- * Año, número de trimestre, emisora (S,U)

- Fondo:

- * Clave de grupo, de saldo, fecha del saldo (P)
- * Fecha del saldo, clave del grupo, del objeto (S,U)
- * Clave del objeto, fecha del saldo, c. del grupo (S,U)

- Grupo:

- * Clave del grupo (P)
- * Nombre del grupo (S,U)

- Hecho:

- * Fecha del hecho, número de folio, tipo de valor,
clave de alta (P)
- * Cuenta (1 o 2), emisora, serie, fecha (S,M)
- * Cuenta (1 o 2), fecha, emisora, serie (S,M)
- * Emisora, serie, fecha, número de folio (S,U)

- Indicador:

- * Clave del indicador (P)
- * Nombre corto del indicador (S,M)

- Nohábil:

- * Fecha del día no hábil (P)

- Objeto:

- * Clave del objeto (P)
- * Emisora, serie (S,M)
- * Sector, emisora, serie (S,M)
- * Tipo de objeto, nombre del objeto, fecha de emisión
(S,M)

- Opera:

- * Clave de referencia, de secuencia (P)
- * Tipo de operación, fecha de operación, clave del
objeto operado (S,M)
- * Tipo de operación, fecha de operación, cuenta (1 ó
2), objeto operado (S,M)

- Precio:

- * Fecha de precio, clave del objeto (P)
- * Clave del objeto, fecha del precio (S,U)

- Saldos:

- * Fecha del saldo, clave del objeto, cuenta (P)
- * Cuenta (S,M)
- * Clave del objeto, cuenta, fecha del saldo (S,U)

- Sector:

- * Clave del sector (P)
- * Nombre corto del sector (S,U)

- Subasta:

- * Clave del objeto (P)
- * Clave de la emisión (S,M)
- * Fecha de la subasta (S,M)

Para encontrar una mayor referencia a la estructura completa de la base de datos, consultar el apéndice C.

IMPLEMENTACION E IMPLANTACION DEL SISTEMA.

En este capítulo explicaremos la implementación e implantación del sistema dentro de la dirección de análisis bursátil de Operadora de Bolsa.

Primero veremos las características básicas propias de este departamento como son: tipos de ambientes de trabajo, tamaño y capacidad de las máquinas, sistemas operativos, número de usuarios, etcétera. Después mencionaremos los puntos más importantes de la implementación como son el manejador de base de datos escogido y los diferentes modos de acceso de los usuarios al sistema. Por último analizaremos la implantación del sistema dentro de la dirección, donde veremos las adecuaciones efectuadas al sistema dadas las condiciones de trabajo existentes.

6.1. Características generales.

Al momento de efectuar este trabajo, el equipo de cómputo de la dirección de análisis bursátil está formado por una máquina Digital serie MicroVAX 3400 con 20 Megabytes en RAM, 1.2 Gigabytes en disco duro y sistema operativo VMS.

A la MicroVAX están conectadas 30 computadoras personales mediante una red Ethernet con DECNET, usando Pathworks como sistema operativo de red.

La MicroVAX funciona de dos formas: Como host cuando las PC's se encuentran en modo de emulación y como servidor de archivos cuando las PC's están como máquinas en red.

6.2 Implementación.

6.2.1 Manejador de base de datos.

Para escoger el manejador de base de datos (DBMS) usado en el proyecto se hizo un estudio donde se analizaron los siguientes puntos:

- Existencia de las versiones para VMS y DOS.
- Portabilidad entre sistemas operativos.
- Facilidad de uso.
- Capacidad de ser usado directamente por el usuario final.
- Bajo consumo de memoria.
- Integración de productos (diccionario, manejador de base de datos, lenguaje de cuarta generación).
- Servicio antes y después de la compra.
- Precio.
- Facilidades para respaldo y recuperación de información.
- Capacitación.
- Presencia en el país.
- Tendencias tecnológicas.

Los productos analizados fueron RDB y Rally, DBASE IV, ORACLE, INGRESS SYBASE y PROGRESS. Los resultados globales del análisis fueron los siguientes.

- RDB y Rally. Ambos productos son de la compañía Digital. Al ser productos propietarios su portabilidad entre plataformas es casi nula. Aunque RDB es un manejador de base de datos bastante bueno, existen muchos problemas con Rally, el cual era casi la única opción de lenguaje de cuarta generación que existía entonces.

- DBASE IV. Aunque los productos de la familia de DBASE son bastante usados y no cuesta trabajo encontrar gente capacitada en ellos, tienen muchos problemas cuando se les

quiere usar en sistemas grandes (un muy mal manejo de los índices, lenguaje poco poderoso, diccionario de datos muy ineficiente, la versión de SQL del DBASE IV presenta muchos problemas, etcétera.).

- ORACLE. Aunque el producto cumple con la mayoría de nuestros requerimientos, fue descartado debido a su altísimo consumo de memoria, alto precio y complejidad organizacional para coordinar toda la gama de subproductos que maneja.

- INGRESS. El análisis de INGRESS no llegó a su fin debido a los enormes problemas de soporte que encontramos en la compañía que lo distribuía entonces en México.

- SYBASE. Como producto SYBASE es de los mejores que analizamos pero en su momento tenía una presencia en el país casi nula, lo que se veía reflejado en cursos de capacitación y soporte tanto técnico como de post-venta. Otro problema de SYBASE en aquel momento (ya está superado actualmente) es que no tenía versión para DOS.

- PROGRESS. Este producto fue finalmente el que escogimos ya que en general cumple con todos nuestros requerimientos. Aunque la compañía tenía problemas de presencia dentro del país, demostró que su estructura era suficiente para darnos el soporte necesario. En el apéndice B se da una pequeña descripción de este producto.

6.2.2. Configuración.

De acuerdo a las necesidades planteadas por los usuarios y al esquema de costos de PROGRESS se eligió la siguiente configuración para el sistema: Una versión completa para DOS, una versión RUN-TIME para LAN (entre las dos permiten desarrollar aplicaciones a varias personas a la vez en ambiente DOS) y una versión RUN-TIME para VMS; de esta forma todos los usuarios tienen acceso a la base de datos.

DECNET-DOS permite mediante NETBIOS que, al levantar un servidor de base de datos PROGRESS en VMS, los *clientes* de la red de PC's puedan acceder dicha base de datos. Para poder realizar la conexión se necesita definir en la PC dentro de la utilería NCP (Network Control Program) el nodo VMS donde se encuentra la base de datos y el nombre del servidor de la misma. Esta definición se da mediante el comando `define remote-adapter-name servicio nodo`, donde "servicio" es el nombre del servidor con el que se levantó la base de datos en VMS y "nodo" el nombre del nodo donde reside la base.

Dentro de la VAX el servidor primario (puede haber más de un servidor por base de datos) expande servidores adicionales para los usuarios remotos (los usuarios de VMS accesan directamente la base de datos sin utilizar el servidor), por lo que habrán de definirse todos estos servidores dentro del NCP en la PC para así poder acceder la base de datos mediante cualquiera de ellos.

6.2.3. Acceso a la base de datos.

Dentro de la VAX, la base de datos se levanta con los siguientes parámetros de entrada :

`progress/multi=start_server/buffers=3000/service=bddab bddab`
 donde:

- `multi=start_server`: levanta el servidor y la base de datos en modo de multiusuarios.
- `buffers=3000`: Asigna 3000 bloques para usarse como buffers de memoria compartida. Este parámetro mejora generalmente el tiempo de acceso a los registros de la base de datos.
- `service=bddab`: Asigna bddab como nombre del servicio de la base.
- `bddab`: Nombre del archivo físico que almacena la base de datos.

Cada usuario dentro de VMS utiliza el siguiente comando para entrar a la base de datos:

`progress/multi_user/startup=amp00.p bddab`, donde:

- `multi_user`: Indica que se está en modo multiusuario.
- `startup=amp00.p`: Nombre del programa de entrada a la base. Si no se proporciona este dato y como solamente se tiene una versión RUN-TIME en VMS, el manejador enviará un mensaje de error indicando la necesidad de dicho programa.
- `bddab`: Nombre de la base de datos.

Nótese que en el comando de entrada no existe referencia alguna a un nombre de servicio ya que, como dijimos anteriormente, los usuarios dentro de VMS accesan la base de datos directamente sin necesidad de utilizar el servidor. Además podemos observar que no damos ni el nombre del usuario ni su password, ya que PROGRESS por default toma la clave de usuario que se dio durante el log-in a VMS.

Dentro de DOS, los usuarios entran a la base de datos de la VAX con el comando:

`MPRO BDDAB -H NODE -S SERVICE -U USER -P PASSWORD -p PROGRAM`

donde:

- `MPRO`: Indica que se está en modo multiusuario.
- `BDDAB`: Nombre de la base de datos.
- `-H NODE`: Nombre del nodo donde se encuentra la base; en caso de existir solamente un nodo este parámetro no es necesario.
- `-S SERVICE`: Nombre del servidor de la base de datos.
- `-U`: Clave del usuario.
- `-P`: Password.
- `-p program`: Programa de entrada a la base. En caso de no dar un nombre de programa de entrada y por tener una versión completa de PROGRESS en la red, el usuario entraría al editor.

6.3. Clases de Usuarios: Existen dos clases de usuarios, cada una con características de operación completamente diferentes:

6.3.1. Los analistas de la dirección de análisis bursátil están conectados a la MicroVAX mediante Decnet y Pathworks. Estos usuarios tienen la facilidad de trabajar dentro de dos tipos de ambientes: a) Ambiente de red; estando dentro de la red se tiene el acceso a discos virtuales dentro de la VAX. Estos discos pueden ser Read-Only (para guardar paquetería estandard como LOTUS o WORD) o Read-Write para hacer transferencia de archivos a otros usuarios de la red y para poder almacenar por algún tiempo archivos que les puedan interesar a varios usuarios. Los principales paquetes para PC que se manejan en la dirección de análisis son LOTUS, Microsoft WORD y un programa de análisis técnico llamado Metastock. b) Ambiente de emulación de terminal. Dentro de este ambiente los usuarios tienen el acceso a consultar la base de datos y a algunas otras facilidades que no se encuentran dentro del ambiente de DOS.

6.3.2. Usuarios remotos (aproximadamente 10). Estos usuarios se encuentran en los centros regionales de la compañía: Hermosillo, Monterrey, Guadalajara, Puebla y San Angel; además están otros usuarios como los de la sucursal de Nueva York, la de Chihuahua, etcétera. Todos ellos se conectan a la computadora de análisis (máximo dos a la vez) por línea telefónica conmutada usando BLAST como paquete de comunicaciones. Una vez dentro del sistema tienen el acceso a varias facilidades como son: consulta de la base de datos, correo electrónico con la gente de análisis, transferencia de archivos hacia sus terminales mediante BLAST, etcétera.

6.4. Implantación.

Al momento de la implantación tuvieron que hacer varias adecuaciones al sistema para que su operación fuera más eficiente; los problemas más importantes que había que atacar era la incompatibilidad de formatos entre VMS y DOS y la administración del volumen de información en el disco, el cual se ve afectado básicamente por el crecimiento de la tabla de hechos. Sobre los dos problemas hablaremos a continuación.

6.4.1. Conversión entre VMS y DOS.

La mayoría de los usuarios entran a la base de datos para obtener información y llevársela a sus PC's para poder analizarla en paquetes como LOTUS o EXCEL. El proceso normal es el siguiente: estando en VMS entran a la base de datos, piden información con salida a archivo, la llevan a la PC (por la red o a través de BLAST), entran a LOTUS y por último hacen un File-Import-Numbers del archivo.

Haciendo lo descrito anteriormente se genera el siguiente problema: el formato de archivo en VMS genera sólo un *line-feed* entre cada línea mientras que la PC genera tanto un *line-feed* como un *carriage-return* por lo que un archivo generado en la VAX se verá en la PC de la siguiente forma:

VAX	PC
10	10
20	20
30	30

Para resolver el problema anterior, todos los programas dentro de la base que tienen una salida a archivo cambian el formato de salida para poder ser llevado a la PC. Esto se hace de la siguiente manera:

- Se crea un prototipo de archivo con tipo *stream* (los archivos de la VAX son del tipo *variable length*); esto se puede hacer usando el comando `CREATE/FDL` de VMS.

- Todos los programas con salida a archivo convierten su salida (en formato VMS) a archivo para PC usando el comando `CONVERT/APP programa prototipo`.

6.4.2. Políticas para controlar el volumen de información dentro del sistema.

Cuando hablamos del diseño lógico de la base de datos se mencionó que aunque las tablas de OPERA y HECHO tienen una estructura muy parecida, se mantienen separadas para facilitar la administración del sistema. Esto se debe a que como solamente se tienen 1.2 Gigabytes en disco duro, estamos obligados a mantener la base de datos con un tamaño que no puede exceder de 300 o 400 Megabytes. Siendo la tabla HECHO la que genera el mayor problema de volumen en la base limitamos su tamaño a almacenar solamente tres meses de historia, respaldando en cinta los meses anteriores. Adoptando la política anterior conseguimos por una parte mantener la base de buen tamaño y por la otra se tiene suficiente información para hacer el análisis diario. En caso de necesitarse más información de esta tabla para algún análisis especial se bajan de cinta los respaldos necesarios a un disco de paso, borrándose cuando ya no se necesiten.

CASO DE ESTUDIO: SISTEMA DE ANALISIS FUNDAMENTAL

En de este capítulo estudiaremos a nivel general el módulo desarrollado para el área de análisis fundamental, así como rutinas comunes usadas dentro de este subsistema entre las que se encuentran:

- Búsqueda de la serie más bursátil de una empresa.
- Cálculo del número de acciones totales de una empresa.
- Cálculo de los múltiplos P/U y P/VL de una acción.
- Variaciones semanales, mensuales y anuales en los precios de una acción.

7.1. Análisis fundamental.

Cada trimestre las empresas industriales registradas en la bolsa mexicana de valores deben mandar a esta institución copias de sus estados financieros trimestrales de acuerdo a un formato establecido por la comisión nacional de valores; dichos estados financieros deben tener cierre fiscal a diciembre y deben encontrarse a pesos reales, es decir, inflacionados al mes del reporte. Además deben enviar los estados financieros para el mismo trimestre del año anterior inflacionados también a pesos del mes reportado. Por ejemplo, terminando junio de 1992 se deben enviar los estados financieros tanto de junio de 1991 como de junio de 1992, ambos a pesos de junio de 1992. El enviar información inflacionada permite hacer comparables los números de diferentes períodos.

El objetivo primario del análisis fundamental es el estudiar las empresas que cotizan en bolsa para así poder pronosticar su comportamiento en el futuro; este estudio se hace a través de sus estados financieros, de entrevistas con gente

de la empresa, analizando las noticias que puedan afectar una empresa o un sector determinado, etcétera.

La jerga bursátil maneja términos que ayudan a comprender mejor las características financieras y bursátiles de las empresas y sobre los cuales hemos desarrollado algunas de las rutinas que veremos en este capítulo; algunos de estos términos son:

- Índice de bursatilidad: Las acciones de las empresas se dividen en series, donde cada una de estas serie tiene características especiales de operación (si son o no para extranjeros, si sus tenedores tienen derecho a voto o no, etcétera). El índice de bursatilidad nos dice que tanto se ha movido una serie de una determinada acción dentro del mercado de valores en un período determinado. Es muy importante para el analista conocer este índice, ya que las series o acciones que no se operan en mucho tiempo (acciones con baja bursatilidad o bursatilidad cero) no siempre reflejan las condiciones reales de las empresas. Debido a lo anterior casi toda la información emitida por el analista esta basada en la serie más bursátil de la empresa (la que más se opera).

- Número total de acciones por empresa: A pesar de que las empresas dan a conocer el número de acciones emitidas por serie, es importante conocer en todo momento el número total de acciones de una empresa para así poder hacer operaciones globales a nivel empresa y no sólo a nivel serie. Esto no siempre es tan fácil de calcular debido a que hay algunas empresas que constantemente están moviendo acciones de una serie a otra o que modifican el número total de acciones mediante derechos a sus accionistas (una suscripción, un split, un canje, etcétera).

- Múltiplos P/U y P/VL: Unos de los indicadores más importantes para poder evaluar empresas, sectores o al mercado en general son los llamados múltiplos, los cuales nos muestran que tan cara está una acción con referencia a las utilidades de la empresa (P/U) o a su valor en libros (P/VL). Mientras mayores sean los múltiplos más caras serán las empresas y por lo tanto mayor será el riesgo asociado. Los diferentes niveles en los valores de los múltiplos varían de acuerdo al tipo de empresa, ya que lo que para una empresa es un múltiplo caro para otra puede que no lo sea.

7.2 Rutinas comunes.

Las rutinas presentadas a continuación son usadas en muchos programas de análisis bursátil. La idea de ponerlas aquí es mostrar formas como se puede aprovechar la estructura de la base de datos presentada en este trabajo para obtener información útil al analista bursátil.

7.2.1 Cálculo del número de acciones.

```
def input param emisora like objeto.iemisora.
def output param ctitact like objeto.nacciones initial 0.
def var serie like objeto.iobjeto extent 10.
def var cuenta as int.
def var numseries as int.

find first objeto where objeto.iemisora = emisora.
/* analiza las series vigentes de la empresa */
cuenta = 0.
for each objeto where objeto.iemisora = emisora :
  if objeto.femision > today then next.
  if objeto.fven <= today then next.
  if objeto.itipoobj <> "ACCION-INDUS" and
    objeto.itipoobj <> "ACCION-FINAN" and
    objeto.itipoobj <> "CERT-PART-ORDI" then next.
  cuenta = cuenta + 1.
  serie [cuenta] = objeto.iobjeto.
```

```

end.
numseries = cuenta.
/* calculo del numero de acciones de la emisora */
do cuenta = 1 to numseries :
    find objeto where objeto.iobjeto = serie [cuenta].
    ctitact = ctitact + objeto.nacciones.
end.

```

7.2.2 Cálculo del Índice de bursatilidad.

```

/* definicion de parametros */
def input param emisora like objeto.iemisora.
def input param fecha as date.
def output param obj-bursa like objeto.iobjeto.
def output param bursa as decimal format "->9.9".
def output param nerror as char format "x(20)".

def var serie like objeto.iobjeto extent 10.
def var cuenta as int.
def var numseries as int.
def var pcara like precio.pventa.

find first objeto where objeto.iemisora = emisora.

/* analiza las series vigentes de la empresa */

cuenta = 0.
for each objeto where objeto.iemisora = emisora :
    if objeto.femision > fecha then next.
    if objeto.fven <= fecha then next.
    if objeto.itipoobj <> "ACCION-INDUS"
        objeto.itipoobj <> "ACCION-FINAN" and
        objeto.itipoobj <> "CERT-PART-ORDI" then next.
    cuenta = cuenta + 1.
    serie [cuenta] = objeto.iobjeto.
end.
numseries = cuenta.

/* busca serie mas bursatil de la empresa */

bursa = -1. obj-bursa = 0. /* valores iniciales */
do cuenta = 1 to numseries :
    find objeto where objeto.iobjeto = serie [cuenta].
    find last aobjetoindi of objeto where falta < fecha + 1
        and iindicador = 1 no-error.
    if available aobjetoindi then
do:
    if bursa = - 1 then
do:
        obj-bursa = objeto.iobjeto.
        bursa = cvalor.

```

```

end.
else
if bursa = ? or aobjetoindi.cvalor > bursa then
do:
  obj-bursa = objeto.iobjeto.
  bursa = cvalor.
end.
end.
end.
end.

```

7.2.3 Cálculo de los múltiplos p/u y p/v1 de una acción.

```

/* definicion de variables */
def buffer cuatriaant for empresa. /* 4to trim año ant */
def buffer mistriaant for empresa. /* mismo trim año ant */
def shared var finicial as date. /* fecha del reporte */

/* definicion de parametros */
def input parameter iemisor a like objeto.iemisor a.
def input parameter ctitact like objeto.nacciones.
def output parameter utrepo as decimal format "->>, >>9.9".
def output parameter utl2me as decimal format "->>, >>9.9".
def output parameter vallib as decimal format "->>>, >>9.9".

/* busca ultimos estados financieros de la emisora */
find last empresa where empresa.iemisor a = iemisor a
and empresa.fechregistro <= finicial no-error.
if not available empresa then return.

/* busca el cuarto trimestre del año anterior */
find cuatriaant where cuatriaant.iemisor a =
empresa.iemisor a and
cuatriaant.anio = empresa.anio - 1 and
cuatriaant.numtrimestre = 4 no-error.

/* busca mismo trimestre, año anterior */
find mistriaant where mistriaant.iemisor a =
empresa.iemisor a and mistriaant.anio = empresa.anio - 1
and mistriaant.numtrimestre = empresa.numtrimestre no-error.

/* calcula utilidad por accion (upa) al reporte */
utrepo = empresa.resnetoejer / ctitact.

/* calcula utilidad por accion a 12 meses */
if available cuatriaant and available mistriaant then
utl2mes = (empresa.resnetoejer + cuatriaant.resnetoejer
- mistriaant.resnetoejer) / ctitact.

/* calcula valor en libros por accion */
vallib = empresa.capcontable / ctitact.

```

```

/* calculo del multiplo p/u */
if ut12mes > 0 then mulpu = ppreact / ut12mes.
else mulpu = 0.

```

```

/* calculo del multiplo p/vl */
if vallib > 0 then mulpvl = ppreact / vallib.
else mulpvl = 0.

```

7.2.4. variaciones en precios.

```

/* definicion de variables */
def var emis like objeto.iemisora.
def var serie like objeto.iserie.
def var ppreact like precio.pventa.
def var varsant as dec format "-zz9.99".
def var varmant like varsant.
def var varaant like varsant.

/* obtiene clave del objeto */
find objeto where objeto.iemisora = emis and
objeto.iserie = serie no-error.

/* calcula ultimo precio de la emisora */
find last precio of objeto.
ppreact = precio.pventa.

/* calcula variacion semana anterior */
find last precio of objeto where
precio.fprecio <= today - 7 no-error.
if available precio then
varsant = (ppreact / precio.pventa - 1) * 100.

/* calcula variacion mes anterior */
find last precio of objeto where
precio.fprecio <= today - 30 no-error.
if available precio then
varmant = (ppreact / precio.pventa - 1) * 100.

find last precio of objeto where precio.fprecio <=
date (month (today), day (today),
year (today) - 1) no-error.
if available precio then
varaant = (ppreact / precio.pventa - 1) * 100.

```

CONCLUSIONES.

Desde el inicio del proyecto en la dirección de análisis hasta ahora han habido muchos cambios como: nuevos instrumentos, nuevas políticas económicas, diferentes formas de hacer análisis, etcétera; todos estos cambios se han tenido que ver reflejados dentro de la base de datos para no verse desactualizada.

Con lo expuesto anteriormente es interesante hacer un análisis retrospectivo y comparar los objetivos iniciales del proyecto expuestos en el capítulo de introducción con los resultados obtenidos. Para hacer este análisis escribiremos a continuación cada uno de los objetivos iniciales y los iremos comparando con su equivalente dentro del sistema hoy en día.

- Base de datos común al área.

Actualmente cada persona que necesita cualquier tipo de información que se encuentre dentro de la base de datos la puede consultar (si es que tiene la autorización adecuada) sin depender que los demás analistas se la quieran prestar o no. Esto ha eliminado en gran medida los núcleos de poder, aunque todavía existen personas que tienen su propia información.

- Sin problemas de redundancia.

Aunque existe alguna redundancia dentro de la base de datos (ver capítulo sobre diseño lógico), está controlada por lo que no representa mayor problema.

Al no tener problemas de redundancia de información, tampoco existen problemas de inconsistencia, lo que le da una mayor credibilidad a la información de la base.

Probablemente una de las mayores ventajas obtenidas sea la enorme reducción de tiempos de captura, ya que en vez de que cada analista capture su propia información (muchas veces redundante con la de los demás), ahora toda la información es capturada en forma automática (85%) o capturada por gente especializada en captura de datos (15%), lo que además reduce muchos errores en la entrada de datos al sistema.

- Obtención fácil de la información.

En este punto es quizás donde se tenga que trabajar más todavía, ya que la estructura de la base de datos es demasiado compleja para el usuario medio, lo que ocasiona que se tengan que crear interfases muy amigables y eficientes para que los usuarios puedan obtener toda la información que necesitan sin que la gente de sistemas les tenga que crear un programa especial por cada consulta que quieran hacer al sistema; esto es especialmente importante dentro de un área de análisis, donde cada usuario requiere correlacionar información diferente cada vez para poder hacer los estudios que crea pertinentes en su momento y el sistema de información debe permitirle realizar todo tipo de consulta a la base de datos en la forma más amigable.

El problema de la capacitación sobre la estructura de la base se ha presentado inclusive con la gente de sistemas, ya que en la mayoría de las universidades existen en general muy pocos cursos sobre bases de datos y en particular sobre diseño de las mismas, lo que dificulta que mucha gente le entienda en poco tiempo a estructuras relativamente complejas.

- Flexible a nuevos requerimientos.

Este punto está muy ligado al anterior, ya que si bien por un lado la estructura de la base de datos tiene algunos problemas para su comprensión, por el otro ha resultado sumamente flexible para añadir nueva información al sistema sin tener que hacer cambios fuertes a la estructura de la base.

Cambios en la información operativa como son las diferentes políticas cambiarias del país (banda o desliz en el tipo de cambio), la creación o eliminación de nuevos plazos o instrumentos (como los casos de nuevos plazos en los ajustabonos y tesobonos o la eliminación del pagafe) e inclusive los cambios a la política económica como el plan azteca, representan en su mayoría cambios muy ligeros a la estructura de la base de datos.

- Sin problemas de integridad.

El tener información correcta dentro de la base de datos representa actualmente la principal responsabilidad del área de informática, ya que muchas veces llega información oficial, supuestamente validada, con una gran cantidad de errores, lo que ocasiona que se tengan muchos tipos de validaciones en los programas de captura tanto automática como manual.

Una validación común dentro de los programas de captura manual consiste en la doble captura de los datos; con esto se reduce en gran medida la entrada de datos incorrectos a la base ya que este procedimiento elimina los llamados 'errores de dedo' y otros errores como el de leer mal los datos.

Respecto a la entidad a nivel tabla o la integridad referencial, en el diccionario del manejador de la base de datos (ver apéndice B) están todas o casi todas las reglas de integridad entre las diferentes tablas de la base.

- Plataforma base para futuros desarrollos.

Actualmente existen proyectos para usar bases de textos y sistemas expertos; además se piensa presentar la información al usuario no solamente a través de datos o series de tiempo sino también a través de gráficas, lo que muchas veces ayuda a aumentar la calidad en la presentación de la información.

Otra ventaja de presentar información con gráficas está en que una gráfica adecuada muestra mucha información en poco tiempo y espacio, lo que hace más eficientes a los sistemas.

Todos estos proyectos de gráficas, bases de texto y sistemas expertos no serían posibles sin contar con una base de datos que sirva como plataforma base que proporcione los datos de entrada a este tipo de sistemas.

- Seguridad de acceso.

Como dijimos en un principio, la mayor arte de la información dentro de la base de datos es información pública; a pesar de eso, el tener un buen mecanismo de seguridad en la entrada al sistema mediante claves de usuario, grupos de usuarios y claves de acceso, nos permite proteger el porcentaje de la información que no es pública, además de evitar que la gente no autorizada entre el sistema para cambiar o borrar información.

En general diremos que el sistema ha sido de mucha utilidad para los analistas, sobre todo al reducirles enormemente el tiempo de búsqueda de la información para poder trabajar. Este punto por si solo le da validez suficiente al trabajo desarrollado en el proyecto ya que al verse reducido fuertemente el tiempo de búsqueda de la información, se puede aprovechar mejor el tiempo disponible y hacer por consiguiente un mejor análisis.

Otro punto que ha sido de gran ayuda es el de utilizar programas de captura automática para la entrada de información a la base. Cuando cada quién tenía sus archivos personales, los mismos analistas tenían que capturar su información, lo que conducía a una gran pérdida de tiempo además de prestarse a cometer muchos errores.

Quizás el punto que más beneficios ha traído es el que cualquier persona autorizada tenga acceso a la información que necesita sin tener que depender de los demás para hacer su análisis. El tener una base de datos común agiliza altamente los tiempos de acceso a la información y termina con muchos burocratismos.

Finalmente diremos que dadas las ventajas que hemos citado sobre el resultado del proyecto, el área de análisis ha empezado a hacer sus requerimientos para pasar a la segunda etapa del proyecto, la cual consiste en la incorporación de gráficas y en el manejo libre de textos. El que esté lista y probada la plataforma para este tipo de proyectos muestra que la primer parte del proyecto cumplió con todas las necesidades que se plantearon en su comienzo.

APENDICE A. MODELO RELACIONAL DE BASE DE DATOS

I. Terminología general de bases de datos.

- Base de datos: Llamamos base de datos al conjunto organizado de la información necesaria para poder operar correctamente dentro de una organización.

- Redundancia de la información: En los sistemas carentes de bases de datos cada aplicación accesa sus propios archivos, lo que muchas veces ocasiona que exista información repetida y por consiguiente una gran pérdida de espacio en disco. En una base de datos estos archivos repetidos son generalmente integrados eliminándose así dicha redundancia; el objetivo final es el de *controlar* más que *eliminar* la redundancia de información, ya que algunas veces existen razones técnicas u operativas que requieren tener más de una copia de ciertos datos dentro de la base.

- Inconsistencia de la información: Cuando debido a la redundancia de la información tenemos diferentes valores para los mismos datos dentro del sistema diremos que dicha información es inconsistente.

Es obvio pensar que un sistema que muestra información inconsistente al usuario no tendrá credibilidad y por lo tanto se convierte en un sistema inútil; por esto, es claro que todo sistema deberá tratar en lo posible de evitar cualquier inconsistencia en su información.

Resulta claro que si un dato está representado en un solo lugar dentro de la base de datos no podrá tener problemas de inconsistencia. Por otro lado, cuando se tienen sistemas con redundancia *controlada*, se deberá garantizar que cualquier actualización (altas, bajas o cambios) en un lugar con

información redundante deberá propagarse a todos los demás lugares donde dicha información está repetida.

- Información compartida. El tener la información dentro de una base de datos permite que diferentes aplicaciones puedan acceder simultáneamente los datos que necesitan para su operación.

Otra ventaja importante que se tiene al poder compartir la información se da con la creación de nuevas aplicaciones, ya que éstas no tendrán que crear sus propios archivos y por lo tanto el tiempo de programación y de mantenimiento se verá reducido.

- Integridad de la información: El tener integridad dentro de la base de datos significa tener información correcta dentro de ella. Algunos problemas de falta de integridad son la inconsistencia de la información, el tener datos erróneos (que la edad de una persona es de 60 cuando en realidad es de 25), el que un trabajador tenga una clave de departamento que no existe en la tabla de departamentos, etcétera.

El tener la información centralizada dentro de una base de datos permite poner los procedimientos de validación dentro de la estructura de la base, reduciéndose así los problemas de falta de integridad. En sistemas de bases de datos donde se comparte información, es especialmente importante el cuidar la integridad de los datos, ya que un dato erróneo puede verse reflejado en muchas aplicaciones.

- Independencia física de los datos: En la mayoría de los sistemas sin bases de datos, la forma como los datos están almacenados en memoria secundaria tiene una fuerte relación con la forma como dichos datos son accedidos; esto ocasiona un problema en el mantenimiento a las aplicaciones ya que cada vez que por razones físicas se cambia la organización

de los datos en memoria secundaria, se tienen que modificar las aplicaciones para adecuarlas a los cambios efectuados.

En los sistemas de base de datos existe una independencia física total entre la forma de almacenar y la forma de acceder los datos, ya que es el manejador de la base de datos (y no los programas) el que se encarga de hacer dicha relación.

II Modelo relacional de base de datos.

- Relación: Decimos que R es una relación de una serie de conjuntos D_1, D_2, \dots, D_n si existen dentro de R una serie de tuplas ordenadas de orden ' n ' $\langle d_1, d_2, \dots, d_n \rangle$ tal que d_i pertenece a D_i para toda i menor o igual a n . Los conjuntos D_1, D_2, \dots, D_n son los dominios de R y el valor de ' n ' es el grado de R .

La representación gráfica de una relación se da en forma de tabla, donde los renglones de la tabla están formados por las tuplas de la relación y las columnas (atributos) de la tabla están tomados a partir de los dominios de la relación.

- Llave primaria: Llamaremos llave primaria al conjunto mínimo de atributos que identifican de manera única a cada una de las tuplas dentro de la relación. En algunas ocasiones más de un conjunto de atributos son llaves candidatas a convertirse en la llave primaria. Una llave candidata que no es la llave primaria se conoce como llave alterna.

- Llave foránea: Dadas dos relaciones R_1 y R_2 no necesariamente diferentes, al conjunto de atributos en la relación R_1 que identifican en forma única las tuplas de la relación R_2 (llave primaria en R_2) se le conoce como llave

foránea. La importancia de las llaves foráneas radica en que junto con las llaves primarias y las vistas proveen de formas para relacionar a tuplas de las diferentes tablas de la base, lo cual le da una de sus mayores ventajas al modelo relacional de bases de datos.

- Dependencia funcional: Se dice que el campo B de una tabla T es funcionalmente dependiente del campo A si para todo valor distinto de A, existe un solo valor distinto de B en T. Por ejemplo, si para cada valor distinto de proveedor existe un único valor de ciudad, entonces diremos que el campo de ciudad tiene una dependencia funcional del campo de proveedor.

- Integridad a nivel tabla: Ningún componente de la llave primaria de una tabla puede ser nulo. Al tener la llave primaria el valor nulo en alguno(s) de sus campos estaría 'apuntando' a más de una tupla en la relación, lo que contradice la definición de llave primaria de identificar en forma única los registros de la relación.

- Integridad referencial: Para toda llave foránea en una relación R1, cada valor de dicha llave debe ser: a) nulo o b) apuntar a una tupla existente dentro de la tabla correspondiente R2. Esta regla evita situaciones como la de la tupla en la tabla de empleados donde la clave de la ciudad de nacimiento del empleado no existe en la tabla de ciudades.

- Vista: Una vista es aquella tabla que no tiene una representación física, sino solamente lógica y que se forma tomando como base a otras tablas. La vista es uno de los conceptos más importantes del modelo relacional, ya que les permite a los datos tener una independencia lógica respecto a su estructura (ver regla 9 de Codd en este apéndice).

III Las 12 reglas de Codd.

E. Codd, el inventor del modelo relacional de bases de datos, definió 12 reglas que debería cumplir un manejador de base de datos completamente relacional. Estas reglas son:

- Regla 1 (de la información): Toda la información dentro de una base de datos relacional está representada explícitamente en el nivel lógico y en una sola forma, por valores en las tablas.

- Regla 2 (garantía de acceso): Cada dato (valor atómico) dentro de una base de datos relacional tiene un camino de acceso garantizado tomando el nombre de la tabla, la llave primaria y el nombre de la columna dentro de la tabla.

- Regla 3 (tratamiento sistemático de valores nulos): Los valores nulos (diferentes del carácter blanco, del cero y de cualquier otro número) están soportados dentro de una base de datos relacional en forma sistemática e independiente del tipo de dato, para representar información faltante o no aplicable.

- Regla 4 (catálogo dinámico en línea): La descripción de la base de datos está representada en el nivel lógico de la misma forma que los datos, de tal forma que los usuarios autorizados pueden usar el mismo lenguaje relacional para acceder tanto esta información como los datos comunes.

- Regla 5 (lenguaje comprensible de datos): Aunque un sistema relacional puede aceptar varios tipos de lenguajes y formas de meter nueva información a la base, debe soportar al menos un lenguaje de caracteres con una sintaxis bien definida que contemple la definición de datos y de vistas, la manipulación de los datos en forma interactiva y por programa y las restricciones de integridad y de seguridad.

- Regla 6: Cualquier vista que en teoría es actualizable, deberá ser actualizable por el manejador de la base.
- Regla 7: Un manejador relacional de base de datos debe tener la capacidad para manejar más de una tupla en una relación con una sola operación tanto para la lectura de los registros como para la actualización de los mismos.
- Regla 8 (independencia física de los datos): Los programas de aplicación permanecerán sin cambio cada vez que existan cambios en el almacenamiento físico de la información o en los métodos de acceso.
- Regla 9 (independencia lógica de los datos): Los programas de aplicación permanecerán sin cambio cuando existan modificaciones a la estructura de la base de datos que no afecten la lógica de dichas aplicaciones.
- Regla 10 (independencia de la integridad): Las restricciones de integridad específicas a cierta base de datos relacional deben ser definidas dentro del mismo lenguaje relacional y guardadas dentro del catálogo, no en los programas de aplicación.
- Regla 11: Un manejador de bases de datos relacionales debe ser independiente en su distribución, es decir, permitir el proceso y los datos distribuidos donde datos y aplicaciones puedan residir en diferentes máquinas.
- Regla 12: Aunque un sistema relacional soporte un lenguaje de bajo nivel (registro por registro), éste no podrá pasar por alto las restricciones y reglas de integridad expresadas en el lenguaje de alto nivel (múltiples registros a la vez).

APENDICE B. MANEJADOR DE BASE DE DATOS PROGRESS

Progress consta de tres partes incluidas en el paquete: Un diccionario de datos, un manejador relacional de bases de datos y un lenguaje de cuarta generación. Además cuenta con varias bibliotecas de procedimientos y con la posibilidad de hacer llamadas al lenguaje C para realizar cálculos muy especializados.

I Diccionario de datos.

Dentro del diccionario de datos de PROGRESS se definen las tablas de la base, sus campos, los índices, la seguridad del sistema a nivel de tablas, campos y usuarios, etcétera.

Al definir cada una de las tablas se proporcionan 3 datos: el nombre de la tabla, la descripción y las restricciones de integridad referentes a dicha relación; en este punto se especifican las condiciones que se deberán cumplir para permitir borrar los registros de la tabla.

En la creación de los campos de las tablas se proporciona la siguiente información: nombre del campo, tipo de dato, formato default de despliegue, nombre por default de la etiqueta al momento del despliegue, valor del campo por omisión, restricciones de integridad del campo al momento de la creación o actualización, si es un campo obligado o no, el mensaje de ayuda al usuario al momento de la captura y la descripción del campo.

Es importante notar que el manejador permite tener las restricciones de integridad en su catálogo, lo que soporta la regla 10 de Codd sobre la independencia de la integridad; así mismo contiene al carácter desconocido o nulo '?'

independiente del tipo de carácter, lo que soporta a su vez la regla 3 de Codd sobre tratamiento sistemático de valores nulos.

Al momento de definir los índices se proporcionan los siguientes datos: nombre del índice, si es índice primario o no, si los campos del índice van en ascenso o descenso y si el índice está activo o no.

Relativo a la seguridad de la base, PROGRESS permite definir usuarios dándoles tres tipos de seguridad: de lectura, de creación y de modificación y borrado de registros; estos tipos de seguridad se pueden definir tanto a nivel de tablas como de campos.

II Manejador de base de datos.

Además de tener sus propios comandos de DDL y DML, PROGRESS soporta completamente los comandos de SQL, lo que le da una gran flexibilidad. Se puede definir la estructura de la base tanto a través del diccionario de datos como a través de programación mediante SQL o con el propio lenguaje del manejador. Para los usuarios autorizados están completamente disponibles todas las metatablas de PROGRESS (`_FILE`, `_FIELD`, etcétera) lo que cumple con la regla 4 de Codd acerca del catálogo dinámico en línea.

Las instrucciones básicas de PROGRESS para la actualización de los datos son:

- Insert crea un registro 'lleno' en la tabla.
- Create crea un registro vacío en la tabla.
- Set asigna valores a campos de registros recién creados.
- Assign asigna valores a campos o variables.
- Update actualiza tablas, campos o variables.
- Delete borra registros.

Cada uno de los comandos tiene características especiales de uso, así como una gran cantidad de parámetros opcionales.

Los comandos para la selección de registros son:

- **For each**, el cual selecciona cada uno de los registros de una o varias tablas que cumplen con ciertas condiciones establecidas mediante el parámetro **where**; a diferencia del **select** de SQL, el **for each** permite procesar los registros seleccionados conforme los va accediendo sin tener la necesidad de definir cursor alguno.

- **Find** accesa un solo registro de la base dejándolo disponible para su proceso. De la misma forma que con el **for each**, con el comando **find** se utiliza el **where** para formar el criterio de selección de registros. Al comando de **find** se le pueden añadir parámetros como **first**, **last**, **prev** o **next** para acceder ciertos registros en particular.

Unas opciones muy útiles tanto para el **for each** como para el **find** son **no-error** y **use-index**. Con el **no-error** podemos analizar si se encontró el registro (**if available** parte) y con el **use-index** podemos hacer la búsqueda usando cierto índice en particular. Cabe mencionar que **PROGRESS** tiene su propio algoritmo de búsqueda para utilizar el índice óptimo según el criterio de selección y que el comando **use-index** sólo está por si queremos pasar por alto este algoritmo de default.

Una alternativa muy útil para apuntar simultáneamente a más de un registro en la misma tabla se da con el uso de los llamados **buffers**, los cuales son apuntadores adicionales a las tablas. Si damos una instrucción como **define buffer b1 for empleado** podremos usar arbitrariamente tanto **b1** como **empleado** para acceder los registros de esta tabla.

III Lenguaje de cuarta generación.

PROGRESS tiene un lenguaje de programación estructurado parecido al PASCAL. Las instrucciones de programación básicas son: Do while, If then else y repeat while.

Como mecanismos para la seguridad de los datos, el lenguaje tiene desarrollados instrucciones internas como los shared locks y los exclusive locks, los cuales aseguran los registros al momento de leerlos o actualizarlos para evitar problemas de integridad; así mismo maneja el concepto de transacción y parámetros como undo, retry y next para terminar con commit o rollback las diferentes transacciones.

Para el despliegue de los datos PROGRESS maneja el concepto de **frame** en el cual uno programa todas las características que tendrá el despliegue de información: encabezados, número de columnas, colores, posición de los datos, etcétera.

Por último diremos que a nivel lenguaje también se permite el añadir instrucciones de SQL inmersas en instrucciones de PROGRESS, de tal forma que una instrucción como

```
select clave-parte from parte with frame f1,
```

donde se combinan SQL y PROGRESS presenta una sintaxis adecuada y por lo tanto correría sin ningún problema.

Una de las mayores ventajas de PROGRESS radica en su enorme portabilidad, ya que el mismo código fuente corre sin modificación alguna en sistemas tan distintos como VMS, DOS, BTOS-CTOS y UNIX lo que permite una gran flexibilidad en cuanto a sus plataformas. Usando instrucciones como

```
if opsys = unix then unix silent ls -la
```

podemos escribir los mismos programas para todas las plataformas mencionadas aunque tengamos llamadas a los sistemas operativos.

APENDICE C. BASE DE DATOS PARA ANALISIS BURATIL

Tabla : AGPOOBJETO

Descripcion : Constitucion de los grupos a una fecha dada.

Campos :

- Igrupo : Clave Del Grupo
- Iobjeto : Clave Del Objeto
- Sgrupo : Es O No Grupo
- Falta : Alta De La Asociacion
- Fbaja : Fecha De Baja De La Asociacion

Indices :

- # Igpobjfalt : Unico. Igrupo, Iobjeto, Falta
- Iobjigpofalt : Unico. Iobjeto, Igrupo, Falta

Tabla : AOBJETOINDI

Descripcion : Relaciona Los Objetos Con Los Indicadores

Campos :

- Iobjeto : Clave Del Objeto
- Sgrupo : Es O No Grupo
- Iindicador : Clave Del Indicador
- Falta : Alta De La Asociacion
- Cvalor : Valor Del Indicador

Indices :

- # Objindifecha : Unico. Iobjeto, Iindicador, Falta
- Indiobjfecha : Unico. Iindicador, Iobjeto, Falta

Tabla : ASIGNACION

Descripcion : Asignaciones De Mercado De Dinero.

Campos :

- Iemision : Clave De Emision
- Iobjeto : Clave Del Objeto
- Stipodis : Tipo De Distribucion
- Npostores : Numero De Postores
- Nposturas : Numero De Posturas
- Arangoinf : Limite Inferior De Tasa O Precio
- Arangosup : Limite Superior De Tasa O Precio

Indices :

- # Iasigna : Unico. Iasigna
- Icuenta : No-unico. Icuenta
- Iemisiobj : No-unico. Iemision, Iobjeto
- Iobjeto : No-unico. Iobjeto,
- Stipodis : No-unico. Stipodis

Tabla : CUENTA

Descripcion : Catalogo De Cuentas.
 Campos :
 - Icuenta : Clave De La Cuenta.
 - Ncuenta : Nombre De La Cuenta.
 Indices :
 - # Icuenta : Unico. Icuenta
 - Ncuenta : Unico. Ncuenta

Tabla : DERECHO

Descripcion : Informacion De Derechos Decretados
 Campos :
 - Fderecho : Fecha Del Derecho
 - Sasamblea : Tipo De Asamblea
 - Iemisoraa : Emisora Anterior
 - Iemisorap : Emisora Posterior
 - Sderecho : Tipo De Derecho
 - Fpago : Fecha De Pago Del Derecho
 - Iseriea : Serie Anterior
 - Iseriep : Serie Posterior
 - Icuponpaga : Cupon Que Se Paga
 - Icuponvige : Cupon Que Entra En Vigencia
 - Naccscir : Acciones En Circulacion
 - Vnominal : Valor Nominal
 - Apropor : Tasa De Proporcionalidad
 - Mtotal : Monto Total
 - Pderecho : Precio
 - Pmercado : Precio De Mercado
 - Ndiaspzo : Numero De Dias De Plazo
 - Dobserva : Observaciones
 - Iref : Referencia Del Derecho
 - Isec : Secuencia Del Derecho
 - Naccsproa : Acciones De Proporcion Anteriores
 - Naccsprop : Acciones De Proporcion Posteriores
 Indices :
 - # Refsec : Unico. Iref, Isec
 - Emisora : No-unico. Iemisoraa
 - Femaseacupde : No-unico. Fderecho, Iemisoraa, Iseriea,
 Icuponpaga, Sderecho
 - Sderfder : No-unico. Sderecho, Fderecho

Tabla : EMPRESA

Descripcion : Datos de estados financieros de empresas
 Campos :
 - Iemisora : Emisora
 - Año : Año

- Numtrimestre	: Numero De Trimestre	
- Mes	:	Mes
- Dictaminado	: Dictaminado	
- Fechregistro	: Fecha De Captura	
- Isector	: Clave Sel Sector	
- Activototal	: S01	Activo Total
- Activocircu	: S02	Activo Circulante
- Efeinvtemp	: S03	Efectivo E Inv. Temporales
- Ctesdocxcob	: S04	Clientes Y Doc. X Cobrar
- Otsctasdocco	: S05	Otras Ctas. Y Doc. X Cob.
- Inventarios	: S06	Inventarios
- Otsactivcirc	: S07	Otros Activos Circulantes
- Largoplazo	: S08	Largo Plazo
- Ctasdocxcob	: S09	Ctas. Documentos X Cobrar
- Invaccsnocon	: S10	Inv. Accs. No Consolidadas
- Otrasinversi	: S11	Otras Inversiones
- Inmuplanteg	: S12	Inmuebles Planta Y Equipo
- Nmuebles	: S13	Inmuebles
- Aquequipind	: S14	Maqui. Y Equipo Industrial
- Otrsequipos	: S15	Otros Equipos
- Depamoracum	: S16	Deprecia Amortiza Acum.
- Constenproce	: S17	Construcciones En Proceso
- Activodiferi	: S18	Activo Diferido
- Otrosactivos	: S19	Otros Activos
- Pasivototal	: S20	Pasivo Total
- Pasivocircul	: S21	Pasivo Circulante
- Pprovedores	: S22	Pas. Cir. Proveedores
- Pccredibanca	: S23	Pas. Cir. Credito Bancario
- Pccredibursa	: S24	Pas. Cir. Credito Bursatil
- Pcpmtosxpag	: S25	Pas Cir. Impuestos X Pagar
- Otspasivcirc	: S26	Otros Pasivos Circulantes
- Pasivlplazo	: S27	Pasivo A Largo Plazo
- Plcredibanca	: S28	Pas. L.pzo. Credito Banca.
- Plcredibursa	: S29	Pas. L.pzo. Credito Bursa.
- Plotscredito	: S30	Pas. L.pzo. Otros Creditos
- Credidiferid	: S31	Credito Diferido
- Otrospasivos	: S32	Otros Pasivos
- Capconconsol	: S33	Capital Contable Consolid.
- Particiminor	: S34	Participacion Minoritaria
- Capcontable	: S35	Capital Contable Mayorit.
- Capcontribui	: S36	Capital Contribuido
- Capsocialpag	: S37	Capital Social Pagado
- Actuacapsocp	: S38	Actualiz. Cap. Soc. Pagado
- Primavtaaccs	: S39	Prima En Venta De Acciones
- Aportfuturaum	: S40	Aport. Futuros Aumen. Cap.
- Capitalganad	: S41	Capital Ganado (Deficit)
- Resulacumrva	: S42	Resul. Acum. Y Rva. Cap.
- Rvarecompacc	: S43	Rva. Para Recompra Accs.
- Exc-actcapco	: S44	Exceso Actualiz. Cap. Con.
- Resnetoejer	: S45	Resultado Neto Del Ejer.
- Dispefectivo	: S46	Efectivo Disponible
- Dispinvertem	: S47	Disp. Inversiones Tempo.
- Pascirmonext	: S48	Pasivo Circ. Mon. Extranj.
- Pascirmonnal	: S49	Pasivo Circul. Moneda Nal.

- Crburpapcom : S50 Credito Bursa. Apel Comer.
 - Crburpapcomi : S51 Cre. Burs. Papel Com. Indi
 - Porcicirobl : S52 Porcion Circul. Obliga.
 - Porcicrobli : S53 Porcion Circ. Oblig. Indi.
 - Otpascircos : S54 Otro Pas. Circ. Con Costo
 - Otpascircos : S55 Otro Pasi. Circ. Sin Costo
 - Pasp1plmonext : S56 Pasivo L.pzo Mon. Extranj.
 - Pasp1plmonnal : S57 Pasivo L.pzo Moneda Nal.
 - Plcreburobl : S58 Pas L.pzo Cred. Bur. Obli.
 - Plcreburobli : S59 Pas L.pzo Cred Bur Obl Ind
 - Plotcredccos : S60 Pas L.pzo Otro Cre C Costo
 - Plotcredscos : S61 Pas L.pzo Otro Cre S Costo
 - Otpasreserva : S62 Otros Pasivos Reservas
 - Otpasivotros : S63 Otros Pasivos
 - E-iresacumpm : S64 Exc-ins Res. Acum Pos Mon.
 - E-irestenanm : S65 Exc-ins Tenen. Act No Mon.
 - Capitrabajo : T66 Capital De Trabajo
 - Fdopenspmaan : T67 Fondo Pens. Prima Antigüe.
 - Numaccscircu : T68 Num. De Accs. En Circul.
 - Numaccsrecom : T69 Num. De Accs. Recompradas
 - Valmdoaccion : T70 Valor De Mdo. De La Accion
 - Ventasnetas : R01 Ventas Netas
 - Costoventas : R02 Costo De Ventas
 - Resultbruto : R03 Resultado Bruto
 - Gastosopera : R04 Gastos De Operacion
 - Resultopera : R05 Resultado De Operacion
 - Costintfinan : R06 Costo Integral De Finan.
 - Rescosintfin : R07 Resul. Desp. Cos. Int-fin.
 - Otrasperfin : R08 Otras Operaciones Finan.
 - Resantimpptu : R09 Resul. Antes Imp. Y Ptu.
 - Proviisrptu : R10 Provision Para Imp. Y Ptu.
 - Resdesimpptu : R11 Resul. Despues Imp. Y Ptu.
 - Partressubnc : R12 Parti. Resul. Subsid. N.c.
 - Resnetconape : R13 Res. Neto Cons. Antes P.ex
 - Partixtnetas : R14 Partidas Extraor. Netas
 - Resnetconsol : R15 Resultado Neto Consolidado
 - Partminresul : R16 Particip. Minor. En Resul.
 - Resulnetejer : R17 Resul. Neto Del Ejercicio
 - Vtanetanacio : R18 Ventas Netas Nacionales
 - Vtanetaextra : R19 Ventas Netas Extranjeras
 - Vtanetadolar : R20 Ventas Netas En Dolares
 - Coinfiintpag : R21 Cost Int. Fin. Inter. Pag.
 - Coinfiintgan : R22 Cost Int. Fin. Inter. Gan.
 - Coinfipercam : R23 Cost Int Fin Perdid Cambio
 - Coinfirepomo : R24 Costo Int Fin Res Pos Mon
 - Otsqstosprodu : R25 Otros Gastos (Produc) Neto
 - Perdvtaaccpr : R26 Perd Vta. De Accs. Propias
 - Perdvtainvpe : R27 Perd/util Vta. Inversiones
 - Provisionisr : R28 Provisiones Para Isr.
 - Provisiimpac : R29 Provisiones Para Impac
 - Provisionptu : R30 Provisiones Para Ptu.
 - Perdvtaactffi : R31 Perdida Venta Activo Fijo
 - Perdvtainper : R32 Perd Vta. Inver. C/car Per
 - Reconspasivo : R33 Reconstruccion De Pasivos

- Otrpartextra : R34 Otras Partidas Extraord.
 - Ventastotal : R35 Ventas Totales
 - Resufiscejer : R36 Resultado Fiscal Del Ejer.
 - Ventnet12mes : R37 Ventas Netas 12 Meses
 - Resoper12mes : R38 Resultado Opera. 12 Meses
 - Resneto12mes : R39 Resultado Neto A 12 Meses
 - Resantparext : C01 Res. Neto Antes Part. Ext.
 - Partaplresul : C02 Partidas Aplic. A Result.
 - Fluderresnet : C03 Flujo Deriv. Del Res. Neto
 - Fludercamcap : C04 Flujo Der. Camb. Cap. Tra.
 - Recugenxoper : C05 Recur. Gener. Por Operac.
 - Partiaextraor : C06 Partidas Extraordinarias
 - Recgenoppext : C07 Rec. Gen. Opera. Part.ext.
 - Fluderfinaje : C08 Fluj Der. Por Finan. Ajeno
 - Fluderfinpro : C09 Flujo Deriv. Finan. Propio
 - Recgenfinan : C10 Rec. Genera. Por Financia.
 - Totftesefec : C11 Total Fuentes De Efectivo
 - Recgenactinv : C12 Rec. Gen. Activ. Inversion
 - Incntoefinv : C13 Inc. Neto Efect. Inv. Temp
 - Efinvtempinp : C14 Efec. Inv. Temp. Ini. Per.
 - Efinvtempfip : C15 Efec. Inv. Tempo. Fin Peri
 - Depramorteje : C16 Depre. Y Amort. Del Ejer.
 - Incrvapenant : C17 Increm. En Reserva Pension
 - Otspartidas : C18 Otras Partidas
 - Decctasxcob : C19 Decrem. Cuentas Por Cobrar
 - Decinventari : C20 Decremento En Inventarios
 - Decotasctas : C21 Decremento Otras Cuentas
 - Incproveedor : C22 Incremento En Proveedores
 - Incotospasiv : C23 Incremento Otros Pasivos
 - Finbanctopzo : C24 Finan. Banc. A Corto Plazo
 - Finbanlgopzo : C25 Finan. Banc. A Largo Plazo
 - Finbursatil : C26 Financiamiento Bursatil
 - Divcobrados : C27 Dividendos Cobrados
 - Otsfinancia : C28 Otros Financiamientos
 - Difeintdeven : C29 Diferencia Int. Devengados
 - Efecambnpaga : C30 Efec Cambiarios No Pagados
 - Utilreestpas : C31 Util. Reestructura Pasivos
 - Amorfinanban : C32 Amort. De Finan. Bancarios
 - Amorfinanbur : C33 Amort. De Fianan. Bursatil
 - Amorotosfin : C34 Amort. De Otros Financiam.
 - Inccapsoc : C35 Incremento Capital Social
 - Divdecretado : C36 Dividendos Decretados
 - Camprivtaacc : C37 Prima En Venta De Acciones
 - Apofutaumcap : C38 Aport. Fut. Aumentos Cap.
 - Decinvaccspe : C39 Decr Inv Accs Carac. Perm.
 - Adqinmplaneq : C40 Adq. Inmuebl. Planta Y Eq.
 - Invconstproc : C41 Increm. Const. En Proceso
 - Vtaotasinver : C42 Vta Otras Inver. Car. Per.
 - Vtaactfijtan : C43 Venta Activo Fijo Tangible
 - Otraspardida : C44 Otras Partidas
 - Unidades : Unidad En Que Estan Los Campos
 - Cierre : Mes De Cierre
 - Pesosmesanio : Mes-año A Que Estan Las Cifras

Indices :

- # Emisaniontri : Unico. Iemisora, Anio, Numtrimestre
- Aniotrimemis : Unico. Anio, Numtrimestre, Iemisora
- Fregemiantri : Unico. Fechregistro, Iemisora, Anio, Numtrimestre
- Pesemianitri : Unico. Pesosmesanio, Iemisora, Anio, Numtrimestre

Tabla : HECHO

Descripcion : Informacion De Los Hechos De La Bolsa
Campos :

- Fhecho : Fecha Del Hecho
- Hhecho : Hora Del Hecho
- Salta : Registro De Alta
- Ifolio : Folio
- Stipoval : Tipo De Valor
- Iemisora : Emisora
- Iserie : Serie
- Icupon :
- Cvolumen : Volumen
- Phecho : Precio
- Fliquida : Fecha De Liquidacion
- Shecho : Tipo De Hecho
- Spico : Pico
- Icuental : Cuenta1
- Icuenta2 : Cuenta2
- Iobjeto2 : Objeto2
- Scorto : Tipo De Venta En Corto

Indices :

- # Fechfolvalal : Unico. Fhecho, Ifolio, Stipoval, Salta
- Ctalemiserfe : No-unico. Icuental, Iemisora, Iserie, Fhecho
- Cta1feemiser : No-unico. Icuental, Fhecho, Iemisora, Iserie
- Cta2emiserfe : No-unico. Icuenta2, Iemisora, Iserie, Fhecho
- Cta2feemiser : No-unico. Icuenta2, Fhecho, Iemisora, Iserie
- Emiserfechfo : Unico. Iemisora, Iserie, Fhecho, Ifolio
- Facemiserpre : No-unico. Fhecho, Iemisora, Iserie, Phecho

Tabla : INDICADOR

Descripcion : Catalogo De Indicadores Bursatiles
 Campos :
 - Iindicador : Clave Del Indicador
 - Ncortoindica : Nombre Corto Del Indicador
 - Nlargoindica : Nombre Largo Del Indicador
 Indices :
 - # Iindica : Unico. Iindicador
 - Ncortoindi : Unico. Ncortoindica

Tabla : NOHABIL

Descripcion : Catalogo De Dias Festivos
 Campos :
 - Fhabil : Dia Habil
 Indices :
 - # Fhabil : Unico. Fhabil

Tabla : OBJETO

Descripcion : Catalogo De Objetos
 Campos :
 - Iobjeto : Clave Del Objeto
 - Nobjeto : Nombre
 - Dobjeto : Descripcion
 - Iemisora : Emisora
 - Memision : Monto Emitido
 - Femision : Fecha De Emision
 - Fven : Fecha De Vencimiento
 - Arend : Tasa Rendimiento
 - Adesc : Tasa De Descuento
 - Iref : Ref. A Opera
 - Iobjexp : Objeto Expresado
 - Itipoobj : Tipo De Objeto
 - Iplazo : Plazo
 - Sgrupo : Es O No Grupo
 - Nacciones : Numero De Acciones
 - Isector : Clave Sel Sector
 - Vnominal : Valor Nominal Inicial
 - Vnominalact : Valor Nominal Actual
 Indices:
 - # Iobjeto : Unico. Iobjeto
 - Emitter : No-unico. Iemisora, Iserie
 - Nobjtipofemi : Unico. Nobjeto, Itipoobj, Femision
 - Secemisser : No-unico. Isector, Iemisora, Iserie
 - Sectipoemise : No-unico. Isector, Itipoobj, Iemisora, Iserie
 - Tipofemi : No-unico. Itipoobj, Femision

**ESTA TESIS NO DEBE
 SALIR DE LA BIBLIOTECA**

- Tipofven : No-unico. Itipoobj, Fven
- Tiponobjfemi : Unico. Itipoobj, Nobjeto, Femision
- Tipoplazo : No-unico. Itipoobj, Iplazo

Tabla : OPERA

Descripcion : Tabla De Operaciones

Campos :

- Iref : Numero De Referencia
- Fven : Fecha De Vencimiento
- Iobjeto1 : Clave Del Objeto
- Iobjeto2 : Objeto2
- Adesc : Tasa De Descuento
- Icuenta1 : Cuental
- Icuenta2 : Cuenta2
- Ifolio : Folio
- Sbanxico : Banxico
- Stipoval : Tipo De Valor
- Stasa : Tipo De Tasa

Indices :

- # Refsec : Unico. Iref, Isec
- Sofopeiobi1 : No-unico. Sopera, Fopera, Iobjeto1
- Soffopcta12 : No-unico. Sopera, Fopera, Icuenta1, Iobjeto1
- Soffopcta21 : No-unico. Sopera, Fopera, Icuenta2, Iobjeto1

Tabla : PRECIO

Descripcion : Tabla De Precios De Los Objetos

Campos :

- Fprecio : Fecha Del Precio
- Iobjeto : Clave Del Objeto
- Ctítulos :
- Pcompra : Precio De Compra
- Pventa : Precio De Venta
- Hprecio : Hora Del Precio

Indices :

- # Fpreiobjpcom : Unico. Fprecio, Iobjeto, Pcompra
- Iobjfprepcom : Unico. Iobjeto, Fprecio, Pcompra

Tabla : SALDOS

Descripcion : Tabla Con Los SalDOS Por Fecha.

Campos :

- Iobjeto : Clave Del Objeto
- Ctítulos : Titulos
- Iobjmonto : Unidades
- Fsaldo : Fecha

Indices :

- # Fecobjcta : Unico. Fsaldo, Iobjeto Icuenta
- Icuenta : No-unico. Icuenta
- Iobjctafsal : Unico. Iobjeto, Icuenta, Fsaldo

Tabla : SECTOR

- Descripcion : Catalogo De Sectores
- Campos :
- Isector : Clave Sel Sector
 - Ncortosec : Nombre Corto Del Sector
 - Nlargosec : Nombre Largo Del Sector
 - Stiporta : Tipo De Renta
- Indices :
- # Isect : Unico. Isector
 - Ncsect : Unico. Ncortosec

Tabla : SUBASTA

- Descripcion : Resultados De Las Subastas
- Campos :
- Iemision : Clave De Emision
 - Iobjeto : Clave Del Objeto
 - Fsubasta : Fecha De La Subasta
 - Mconvocado : Monto Convocado
 - Massignado : Monto Asignado
 - Msolicitado : Monto Solicitado
 - Asolimin : Tasa Sol. Min.
 - Asolimax : Tasa Sol. Max.
 - Asoliprom : Tasa Sol. Prom.
 - Aasigmin : Tasa Asig. Min.
 - Aasigmax : Tasa Asig. Max.
 - Aasigprom : Tasa Asig. Prom.
 - Aexcepcion : Tasa Excepcion
 - Npatposnco
- Indices :
- # Iobjeto : Unico. Iobjeto
 - Fsubasta : No-unico. Fsubasta
 - Iemision : No-unico. Iemision

BIBLIOGRAFIA

1. E.F. CODD. *Is your DBMS really relational ?*
Computerworld, Octubre 1985
2. C.J. DATE. *A practical guide to database design.*
IBM Santa Teresa programming center, Diciembre 1982
3. GIO WIEDERHOLD. *Database design.*
McGraw-Hill, 1985
4. C.J. DATE. *Database Management Systems.*
Addisson Wesley, 1980
5. *PROGRESS programming handbook.*
PROGRESS Software Corporation, 1991
6. *DB2 aplicacion design workshop.*
IBM Corporation, 1987
7. Dimitris Chorafas
Fourth and fifth generation programming languages.
McGraw-Hill, 1987