

32
2eje.



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ACATLAN**

**DISEÑO Y APLICACION DE UNA BASE DE DATOS
BASADA EN ORACLE**

T E S I S
QUE PARA OBTENER EL TITULO DE:
**LICENCIADO EN MATEMATICAS
APLICADAS Y COMPUTACION**
P R E S E N T A :
CESAR TALAVERA GOMEZ

ASESOR DE TESIS: SERGIO V. CHAPA VERGARA



NAUCALPAN, EDO. DE MEXICO FEBRERO DE 1984



**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.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES "ACATLAN"

DIVISION DE MATEMATICAS E INGENIERIA
PROGRAMA DE ACTUARIA Y M.A.C.

ACATLAN

SR. CESAR TALAVERA GOMEZ
Alumno de la carrera de Matemáticas
Aplicadas y Computación,
P r e s e n t e

De acuerdo a su solicitud presentada con fecha 9 de marzo de 1993, me complace notificarle que esta Jefatura tuvo a bien asignarle el siguiente tema de tesis: "DISEÑO Y APLICACION DE UNA BASE DE DATOS BASADA EN -- ORACLE", el cual se desarrollará como sigue:

INTRODUCCION

CAP. I BASES DE DATOS RELACIONALES

CAP. II ESTUDIOS DE UN CASO: ANALISIS Y DISEÑO

CAP. III ESTUDIO DE UN CASO: IMPLANTACION

CAP. IV APLICACION

CONCLUSIONES

BIBLICGRAFIA

APENDICE--

Asimismo fué designado como Asesor de Tesis el -
Dr. Sergio Chapa Vázquez, Profesor de esta Escuela.

Ruego a usted tomar nota que en cumplimiento de lo especificado en la Ley de Profesiones, deberá presentar servicio social durante un tiempo mínimo de seis meses como requisito básico para sustentar examen profesional, así como de la disposición de la Coordinación de la Administración Escolar en el sentido de que se imprima en lugar visible de los ejemplares de la tesis el título del trabajo realizado. Esta comunicación deberá imprimirse en el interior de la tesis.

EN F. T. ACATLAN
A T E N T A M E N T E
"POR MI PARTE, EL ESPIRITU"
Acatlán, Enero 31 de 1994.

ACT. LAUREN B. BECERRA
Jefe de División de Actuaría
y M.A. de Matemáticas
Aplicadas y Computación

LMRB/cg.



SECRET

SECRET

SECRET

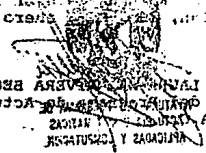
SECRET

SECRET

SECRET

SECRET

SECRET



A MIS TÍOS:

POR QUE SIN SABERLO ME ENSEÑARON MAS DE LO QUE NUNCA SE
IMAGINARON.

A TODA MI FAMILIA EN GENERAL PORQUE SIEMPRE HAN SABIDO
COMPRENDER MIS ERRORES Y ORIENTARME.

A MIS AMIGOS Y MUY ESPECIALMENTE A LOS DE LA UNIVERSIDAD POR
TODAS LAS COSAS QUE PASAMOS JUNTOS.

A MIS MAESTROS:

PORQUE SIEMPRE TENDRÁN UN LUGAR MUY ESPECIAL
EN MI MENTE Y ESTARÁN PRESENTES EN TODOS LOS ACTOS DE MI VIDA.

A MI JURADO Y ESPECIALMENTE A SERGIO CHAPA, POR DEDICARME TIEMPO
TAN VALIOSO.

A MI UNIVERSIDAD:

POR ABRIRME LOS OJOS Y HACER DE MI LO QUE SOY Y POR TODO AQUELLO
QUE CON PALABRAS NUNCA PODRÍA EXPRESAR.

**A MIS AMIGOS DEL PENTATLÓN:
PORQUE MUCHAS DE SUS ENSEÑANZAS LAS HE APLICADO EN MI VIDA.**

**Y FINALMENTE A MI MÉXICO PORQUE SU CULTURA SU GENTE Y SUS
PAISAJES LO HACEN ÚNICO EN EL MUNDO.**

ÍNDICE

INTRODUCCIÓN

CAPITULO 1. BASES DE DATOS RELACIONALES	1
1.1 El modelo relacional	1
1.2 Algebra relacional	3
1.3 Una base de datos mal diseñada	8
1.4 Tipos de asociaciones	11
1.5 Normalización	16
1.6 Diagrama entidad relación	21
CAPITULO 2. ESTUDIO DE UN CASO: ANÁLISIS Y DISEÑO	29
2.1 Propuesta	29
2.2 Análisis del sistema	35
2.3 Diagrama funcional	44
2.4 Diagrama entidad relación	47
2.5 Ambientes integrados	50
CAPITULO 3. ESTUDIO DE UN CASO: IMPLANTACIÓN	55
3.1 Estructuras de archivos ORACLE	55
3.2 Estructuras de memoria ORACLE	62
3.3 Implantación de la base de datos ORACLE	67
CAPITULO 4. APLICACIÓN	70
Conclusiones	78
Bibliografía	79
Apéndice	80

INTRODUCCIÓN

Este trabajo tiene como objetivo presentar el diseño e implementación de una base de datos para una fábrica de automóviles. El problema fue llevado a un sistema manejador ORACLE en una plataforma *Sequent S2000/700*.

El desarrollo del sistema, se plantea desde el control de materiales enviados de Alemania, hasta que son utilizados para el armado de los automóviles en la ciudad de Puebla, pasando por el control del almacén en donde se guardan los materiales.

El primer capítulo, corresponde al marco teórico de las bases de datos relacionales, iniciando con el modelo relacional y su álgebra. Posteriormente se tratan algunos problemas de bases de datos mal diseñadas y como prevenirlos mediante la normalización. En este capítulo también se introduce una sección en donde se tratan los tipos de asociaciones, para comprender el diagrama entidad relación.

En el capítulo dos se presentan los tiempos en horas hombre que llevara la construcción del sistema en cada una de sus etapas, se explica el funcionamiento del mismo, y se muestran los diagramas de funciones y entidad relación. Debido a que para la realización de este sistema, se utilizaron herramientas CASE (Computer Aided Software Engineering), se complementa con una sección sobre tecnologías de ambientes integrados.

En el capítulo tres se explica como el manejador ORACLE manipula sus estructuras internas, (estructuras de archivos y de memoria). Finalmente se explica la implantación del sistema en la fábrica de autos, tomando en cuenta la cantidad de información manejada y los tiempos de respuesta que fueron requeridos.

En el capítulo cuatro se presentan algunos ejemplos de pantallas del sistema, comentando las actualizaciones a la base de datos, así como algunas estadísticas importantes.

El sistema se encuentra actualmente en uso y superando las expectativas que de el se tenían en cuanto a la confiabilidad y seguridad en la información. El tiempo de respuesta para el manejo de transacciones en línea es de aproximadamente 3 segundos para información que involucran un promedio de 500,000 registros, siendo ésta una cifra récord dentro de la fábrica de automóviles.

Después de su liberación, el sistema fue presentado a los directivos de la fábrica de autos en Alemania, ya que lo que implicó con respecto a tecnología, tiempos de respuesta y manejo de información, fueron de gran importancia incluso para el país Alemán.

Debido a que el sistema se encuentra funcionando actualmente en Puebla, los directivos alemanes decidieron adaptarlo en su país, esto representa un ahorro en cuanto a recursos, tanto monetarios como de personal.

La adaptación del sistema, no les será difícil, ya que la forma de trabajo en Alemania es similar a la forma de trabajo mexicana y prácticamente solo se capacitará a los usuarios finales.

Con la afinación de la base de datos ORACLE, se logró reducir el tiempo de los procesos que antes tardaban de 20 a 30 minutos a un promedio de sólo 7 segundos, siendo esta otra característica importante que intereso al país Alemán.

Las bases de datos adquieren más importancia en el mundo actual, ya que ayudan a tomar decisiones más complejas que afectan a un mayor número de personas.

Debido a la competencia de todo tipo no solo a nivel nacional sino internacional, hacen que las personas, las empresas y los gobiernos tengan la creciente necesidad de recuperar información para buscar el liderazgo entre sus competidores, ya que solo los que posean la información más segura de la manera más rápida en todos los niveles de la organización, serán los que tengan las mejores oportunidades.

El objetivo primordial de un manejador de base de datos es precisamente el de crear un ambiente en el que sea posible guardar y recuperar información de forma conveniente y eficiente.

CAPITULO 1

BASES DE DATOS RELACIONALES

En este capítulo se expondrá la teoría de las bases de datos relacionales. Primero se hablará del modelo relacional y los fundamentos matemáticos de éste. En las secciones posteriores se tocan los temas referentes a las bases de datos mal diseñadas y los problemas que se originan, los diferentes tipos de asociaciones que existen en una base de datos, y el concepto de normalización en las bases de datos relacionales. Finalmente, explican las relaciones en un diagrama entidad relación, donde también se muestra cómo obtener las tablas a partir de este diagrama.

1.1 EL MODELO RELACIONAL

Desde una perspectiva histórica, el modelo relacional es relativamente reciente. Los primeros modelos de bases de datos dependen más de la implantación física de la base de datos. Siendo los modelos jerárquico o de red, claros ejemplos de lo anterior.

El Dr. Edgar F. Codd de IBM, propuso un modelo para manipular bases de datos [Codd E.F. *A relational model of data for large shared data* CACM 1970]. El modelo relacional representa la base de datos como un conjunto de tablas, y aunque éstas son un concepto simple e intuitivo, existe una correspondencia directa entre lo que es una tabla y el concepto matemático de relación, de ahí, recibe su nombre el modelo relacional de datos.

Una base de datos relacional está compuesta de tablas o arreglos en dos dimensiones que consisten en renglones y columnas, donde cada columna contiene el mismo tipo de información, y cada renglón el mismo conjunto de información. El campo contiene un valor en la intersección del renglón y la columna.

El nombre de la tabla debe ser único en la base de datos, así como el nombre de la columna debe ser único en la tabla. El orden de las columnas y renglones es arbitrario.

Las columnas de la tabla modelan los atributos que son características o cualidades de la tabla. Para cada atributo existe un conjunto de valores permitidos, llamados dominio de ese atributo; por ejemplo, para una tabla dada con uno de sus atributos llamado MARCA_DE_AUTO el dominio sería las marcas de todos los automóviles.

Una característica del modelo relacional es su gran sencillez, ya que simplifica el diseño del esquema conceptual, puesto que hay sólo una estructura lógica de los datos; es decir, la relación

por considerar. El mecanismo de acceso a la información es por su valor y no por su posición o apuntador.

Cada columna contiene los valores de los atributos y cada renglón es llamado *tupla*, ésta palabra es tomada de la definición de grupos tales como, quintupla, séxtupla, etc. De este modo un grupo de n-elementos es una tupla. El número de tuplas en una tabla es llamado *cardinalidad* de la tabla. La Fig.1.1 muestra las características de una tabla.

En el modelo relacional tenemos lo que se conoce como *entidad*, que es una persona, cosa o lugar de interés para el usuario, acerca del cual el sistema mantiene, relaciona y despliega información. Las entidades son modeladas por medio de tablas (sección 1.6).

Las tuplas de una tabla pueden identificarse en forma única por una o más *llaves*; esto es, la llave asegura que no haya más de una tupla repetida en una tabla en una base de datos. A estas llaves se les conoce como *primarias*.

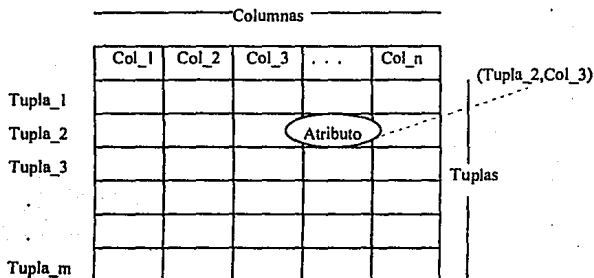


Fig. 1.1 Tuplas, columnas y valores de atributo en una tabla.

Las llaves primarias pueden estar formadas por más de una columna, si éste es el caso, entonces será una *llave primaria compuesta*.

Una *llave foránea* es una columna o grupo de ellas en una tabla, que es (son) llave(s) primaria(s) en otra tabla. Estas indican las asociaciones entre tablas.

Una *tabla prima* es aquella que contiene una llave primaria no compuesta; es decir, la llave primaria consta de sólo una columna.

No está permitido hacer cambios a la llave primaria, y tampoco se permiten valores nulos o duplicados.

Los atributos son una característica o cualidad de una entidad, éstos deben normalizarse (la normalización evita la redundancia de información 'sección 1.5') y son modelados en las columnas de la tabla. Fig. 1.1.

Cuando se tienen dos o más entidades, y éstas tienen relación entre sí, decimos que las entidades están asociadas, y existen tres tipos de asociación, uno a uno, uno a muchos y muchos a muchos (sección 1.4).

Como antes se mencionó, las entidades son modeladas por medio de tablas. Las características de esas entidades son modeladas por las columnas de una tabla (atributos de la entidad). El conjunto de entidades con sus atributos y relaciones entre éstas, debe representarse en algo que permita la percepción del mundo real de una manera completa, el diagrama entidad-relación es la forma como se representa la estructura de la base de datos. En la sección 1.6 explicaremos éste diagrama.

En términos generales, el modelo relacional representa la base de datos como un conjunto de tablas y maneja la información de éstas como operaciones entre relaciones matemáticas, para lo cual utiliza el álgebra y el cálculo relacional.

1.2 ALGEBRA RELACIONAL

Un lenguaje de consulta sirve para que se solicite información a la base de datos, éstos a su vez se clasifican en lenguajes de procedimientos y lenguajes sin procedimientos; en el primero se debe indicar la información requerida y cómo obtenerla, en el segundo se solicita la información sin especificar el modo para obtenerla. El álgebra relacional es un lenguaje de procedimientos.

Una relación matemática esta dada por:

$$R = \{ (a, b, c, \dots, m) / a \in A, b \in B, c \in C, \dots, m \in M \}$$

En donde:

(a, b, c, \dots, m) es una tupla o eneada y A, B, C, \dots, M son los dominios de los atributos.

El orden de una tupla es igual al número de tributos que la conforman.

En una relación se cumple que:

$R \subseteq A \times B \times C \times \dots \times N$ en donde \subseteq es el subconjunto propio.

En el álgebra relacional tenemos las siguientes operaciones:

La unión: Denotada por \cup se define como:

$$R1 \cup R2 = \{ (a,b,c, \dots, m) / (a,b,c, \dots, m) \in R1 \text{ o } \in R2 \}$$

Que indica que si la relación R1 se une con la relación R2. Las tuplas que resulten pertenecen a R1 o pertenecen a R2.

La intersección: Denotada por \cap se define como:

$$R1 \cap R2 = \{ (a,b,c, \dots, m) / (a,b,c, \dots, m) \in R1 \text{ y } \in R2 \}$$

Indica que si la relación R1 se interseca con la relación R2 las tuplas que resulten de la intersección pertenecen a R1 y pertenecen a R2.

La diferencia: Denotada por $-$ se define como:

$$R1 - R2 = \{ (a,b,c, \dots, m) / (a,b,c, \dots, m) \in R1 \text{ no } \in R2 \}$$

La diferencia de la relación R1 - R2 contiene todas las tuplas que están en R1, pero que no están en R2.

El producto cartesiano de R1 con R2 se define como:

$$R1 \times R2 = \{ (a,b,c, \dots, a',b',c', \dots) / (a,b,c, \dots) \in R1 \text{ y } (a',b',c', \dots) \in R2 \}$$

Al número de tuplas de una relación, se le conoce como **cardinalidad**.

En la unión, la intersección y diferencia, se afecta la cardinalidad y se requiere que las relaciones sean del mismo orden y homogéneas.

El producto cartesiano afecta tanto a la cardinalidad como al orden de la relación resultante. Si R1 tiene cardinalidad C1 y orden O1 y R2 cardinalidad C2 y orden O2, el producto cartesiano que resulta tiene cardinalidad C1 X C2 y orden O1 + O2.

En la fig. 1.2 se presenta el ejemplo de una tabla, la cual será utilizada para ejemplificar algunos conceptos y muestra la ubicación de cierto material, el usuario y la fecha de ubicación.

En la fig. 1.2 el nombre de la tabla es SA_MAT_UBICADO, los atributos son FOLIO, AREA, FILA, ESTIBA y NIVEL. El dominio del atributo AREA es N001. Para el atributo NIVEL su dominio es A, B, así sucesivamente.

SA_MAT_UBICADO

FOLIO	AREA	FILA	ESTIBA	NIVEL	USUARIO	FECHA
910629468124	N001	063	007	A	COLIN	12-MAR-93
910628101112	N001	065	007	B	COLIN	13-FEB-93
910906671440	N001	063	005	B	COLIN	14-SEP-92

Fig. 1.2 Ejemplo de la tabla llamada SA_MAT_UBICADO.

La información de una tabla puede estar relacionada con la información de otra tabla. Los operadores relacionales que se explicaron anteriormente son de gran importancia en el álgebra relacional, ya que éstos realizan la operaciones entre conjuntos. En seguida se describirán los operadores relacionales especiales.

La proyección: La denotaremos por el símbolo π . Este sirve para seleccionar ciertos atributos de una relación y es un operador unario. La relación resultante tendrá un orden generalmente menor comparado con el de la relación original.

Por ejemplo, utilizando la tabla de la fig. 1.2, supongamos que cierto material está ubicado en un sitio determinado; para identificar éste material contamos con el número de FOLIO, la FECHA de ubicación y la persona que ubicó ese material.

Q1: Obtener los folios del material.

$\pi_{\{FOLIO\}} : SA_MAT_UBICADO$

Con la proyección anterior se obtiene:

FOLIO
910629468124
910628101112
910906671440

Q2: Obtener las fechas de ubicación del material y el usuario que lo ubicó.

π (FECHA, USUARIO) : SA_MAT_UBICADO

El resultado obtenido de ésta proyección es:

FECHA	USUARIO
12-MAR-93	COLIN
12-FEB-93	COLIN
14-SEP-92	COLIN

La selección: Denotada por σ . Este operador es unario al igual que el operador de selección. Su función es la de eliminar las tuplas que no cumplan con los criterios dados. La cardinalidad que resulta es, por lo general, menor a la de la relación original.

Tomando nuevamente el ejemplo de la tabla de la fig. 1.2.

Q3: Obtener toda la información del material cuya estiba es 007.

σ (ESTIBA=007) : SA_MAT_UBICADO

El resultado de ésta consulta es:

FOLIO	AREA	FILA	ESTIBA	NIVEL	USUARIO	FECHA
910629468124	N001	063	007	A	COLIN	12-MAR-93
910628101112	N001	065	007	B	COLIN	13-FEB-93

Q4: Obtener la información de los folios con nivel B.

π (FOLIO) [σ (NIVEL = B) SA_MAT_UBICADO]

Con esto se obtiene:

FOLIO
910628101112
900906671440

El join: Se denota por el símbolo θ . Es el acoplamiento de las relaciones con dominio común.

Si la columna A1 de la relación A y la columna B1 de la relación B son comunes (esto es, pertenecen al mismo dominio), entonces, el "join" de A y B sobre esas columnas, produce una relación que consta de todas las tuplas de A combinadas con todas las tuplas de B.

La cardinalidad de la relación que resulta es igual al producto del las tuplas de las relaciones originales; es decir, si A tiene cardinalidad C1 y B cardinalidad C2, entonces $A \theta B$ tiene cardinalidad $C1 \times C2$.

De la misma forma, el orden de la relación es igual a la suma de los órdenes de las relaciones originales; es decir, si A tiene orden O1 y B orden O2 entonces $A \theta B$ tiene orden $O1 + O2$.

Por ejemplo para dos relaciones A y B tenemos:

$$\sum_{i=1}^n A(i) + \sum_{j=1}^m B(j) = \text{orden de la relación resultante.}$$

Si consideramos la siguiente tabla SA_DETALLE.

SA_DETALLE

FOLIO	PLANTA	NUMERO PARTE
910629468124	601	N 01024011
910628101112	601	N 01024012

Es posible hacer un "join" entre SA_DETALLE y SA_MAT_UBICADO (fig. 1.2) , ya que se tienen atributos en común, cuyo dominio es el mismo: ese atributo es FOLIO.

Cuando dos o más atributos en dos o más relaciones tienen el mismo nombre, como éste caso, se dice que son homónimos, y es necesario prefijar el atributo con el nombre de la relación a la que pertenece, de ésta manera $SA_DETALLE.FOLIO = SA_MAT_UBICADO.FOLIO$. Algunas veces se le da un nombre temporal a la relación, si es que se utiliza más de una vez la misma, anteponiéndose un nombre temporal al atributo. Al nombre temporal se le conoce como *alias*. Esto se muestra a continuación.

Q5: Obtener el numero de parte y la planta en donde los folios de las tablas SA_DETALLE y SA_MAT_UBICADO sean iguales y ademas el nivel sea igual a A.

π (D.NUMERO PARTE, D.PLANTA)

$[\sigma_{(M.NIVEL = A) (SA_DETALLE D \theta SA_MAT_UBICADO M)}]$ (D.FOLIO = M.FOLIO)

Obsérvese que se dio un alias a las tablas involucradas, D y M respectivamente y se utilizaron para prefijar FOLIO.

La información que resulta es:

NUMERO PARTE	PLANTA
N 01024011	601

Para éste caso sólo se recupera una tupla.

Hasta aquí se han visto los operadores relacionales, en la siguiente sección se tratan las bases de datos mal diseñadas.

1.3 UNA BASE DE DATOS MAL DISEÑADA

Existen algunos problemas que pueden surgir al diseñar bases de datos relacionales, tales como:

- 1.- Repetición de la información
- 2.- No es posible representar cierta información
- 3.- Se pierde la información

En seguida se explica cada uno de ellos detalladamente.

Repetición de la información.

Se consideran la tabla SA_DETALLE (FOLIO, PLANTA, NUMERO PARTE) de la sección 1.2, y una nueva tabla que se llamará SA_DETALLE_REQ (REQ_NUMERO, INDICE, FOLIO, CANTIDAD_SURTIDA). Las dos tablas y sus valores se muestran a continuación.

SA_DETALLE

FOLIO	PLANTA	NÚMERO_PARTE
911220371011	601	N 01024011
910680482111	601	285780
920225495011	601	BAQ049111AB

SA_DETALLE_REQ

REQ_NUMERO	INDICE	FOLIO	CANTIDAD_SURTIDA
112111111	11	911220371011	500
211243001	06	920225495011	606
889789789	09	911220371011	700

Aplicando el operador relacional θ "join" en la columna FOLIO tenemos:

$$\mathcal{T} \{ \text{FOLIO, REQ_NUMERO, INDICE, PLANTA, NUMERO_PARTE, CANTIDAD_SURTIDA} \}$$

$$\{ \text{SA_DETALLE } D \theta \text{ SA_DETALLE_REQ } R \} \{ D.\text{FOLIO} = R.\text{FOLIO} \}$$

La relación que resulta de lo anterior es:

FOLIO	REQ_NUMERO	INDICE	PLANTA	NÚMERO_PARTE	CANTIDAD_SURTIDA
911220371011	112111111	11	601	N 01024011	500
911220371011	889789789	09	601	N 01024011	700
920225495011	211243001	06	601	BAQ049111AB	600

Suponiendo que se tuviera una tabla de éste tipo y fuera necesario agregar información a la misma, como por ejemplo REQ_NUMERO 112111111, INDICE 12, FOLIO 920225495011 y CANTIDAD_SURTIDA 1000. Se tendría que insertar a la tabla una tupla tal como: (920225495011, 112111111, 12, 601, BAQ049111AB, 1000), porque se deben tener valores para todos los atributos, esto no es de ninguna manera conveniente, ya que se necesitaría repetir algunos datos.

En la tabla SA_DETALLE_REQ que se tiene al principio sólo se inserta la tupla: (112111111, 12, 920225495011, 1000), que es más conveniente.

Si cambia el NUMERO_PARTE en la relación anterior, son más las tuplas que deben modificarse, mientras que si se utiliza la tabla SA_DETALLE sólo se modifica una tupla.

Incapacidad para representar cierta información.

Si se consideran las tablas SA_CENTRO_COSTO (CENTRO_COSTO, CLAVE_MOVIMIENTO, DESCRIPCION) y SA_REQ (REQ_NUMERO, PLANTA, NUMERO_PARTE, CENTRO_COSTO), suponiendo que se crea una tabla a partir de éstas dos, que llamaremos SA_COMBINACION (REQ_NUMERO, CENTRO_COSTO, CLAVE_MOVIMIENTO, PLANTA, NUMERO_PARTE, DESCRIPCION), determinamos que:

En SA_COMBINACION no se puede crear un centro de costos que no haya hecho alguna requisición, a no ser que se introduzcan valores vacíos para éstas. En las relaciones originales no es necesario introducir valores vacíos, es decir, un centro de costos puede existir independientemente de que haya o no requisiciones.

Pérdida de información.

Consideremos la tabla SA_DETALLE que se muestra a continuación:

SA_DETALLE

FOLIO	PLANTA	NUMERO_PARTE
911023779170	601	N 01046510
920131976744	601	048105425
920204981091	001	BAC141031C
920219046525	501	026107341A

Ahora consideremos dos tablas mas, SA_DET1(FOLIO, PLANTA) y SA_DET2(PLANTA, NUMERO_PARTE) que se obtuvieron a partir de SA_DETALLE.

Para SA_DET1 se tiene:

FOLIO	PLANTA
911023779170	601
920131976744	601
920204981091	001
920219046525	501

Y para SA_DET2 se tiene:

PLANTA	NUMERO_PARTE
601	N 01046510
601	048105425
001	BAC141031C
501	026107341A

Aplicando el operador de "join" sobre SA_DET1 y SA_DET2 obtenemos el resultado que se muestra en la siguiente tabla.

FOLIO	PLANTA	NUMERO_PARTE
911023779170	601	N 01046510
911023779170	601	048105420
920131976744	601	N 01046510
920131976744	601	048105420
920219046525	501	026107341A
920204981091	001	BAQ141031C

Como se observa, existen diferencias entre SA_DETALLE y SA_DET1 θ SA_DET2, puesto que, para las dos primeras tuplas en ésta última, se tienen diferentes combinaciones PLANTA / NUMERO_PARTE para el mismo FOLIO. Lo mismo sucede para las tuplas 3 y 4, aunque existen más tuplas, hay menos información porque no se puede determinar cual es el FOLIO correcto para la combinación PLANTA / NUMERO_PARTE.

Antes de pasar al tema de normalización, es necesario conocer algunos conceptos importantes, los cuales se explican en la siguiente sección.

1.4 TIPOS DE ASOCIACIONES.

Existen diferentes tipos de asociaciones entre las entidades o entre otros tipos de relaciones. Las asociaciones caen dentro del alcance del sistema, el cual debe mantener, correlacionar y desplegar información.

Las asociaciones requieren de, al menos, dos entidades y se presentan en tres modalidades, las cuales se explican a continuación.

Asociación uno a uno.

La asociación uno a uno (1:1), se presenta cuando cada ocurrencia de A, esta relacionada con a lo más una ocurrencia de B y cada ocurrencia de B esta relacionada con a lo más una ocurrencia de A. La fig. 1.3 describe gráficamente lo anterior.

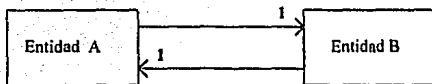


Fig. 1.3 Asociación uno a uno.

Como ejemplo consideremos dos tablas A y B tal como se muestra en seguida:

A	B
PK	PK
a1	b1
a2	b2
a3	b3
a4	
a5	

En donde PK la representa la llave primaria de la tabla (*del Ingles primary key*).

Si se desea establecer una asociación uno a uno (1:1) entre A y B lo podemos hacer de las dos formas siguientes:

A/B		B/A	
PK	FK	PK	FK
a1	b1	b1	a1
a2	b3	b2	a2
a3	---	b3	a3
a4	---		
a5	b2		

En donde FK representa la llave foránea de la tabla (*del Ingles foreign key*).

Tanto en el caso a), como el caso b), se representa la asociación 1:1 entre A y B.

Como se observa, la asociación 1:1 es modelada poniendo la llave primaria (PK) de alguna de las dos tablas como llave foránea (FK) en la otra tabla.

La asociación modelada en el caso b) es más conveniente, ya que se busca minimizar valores nulos, ya que estos son difíciles de manejar. Técnicamente, las dos asociaciones significan lo mismo. Este tipo de asociaciones son simétricas.

Asociación uno a muchos.

La asociación uno a muchos (1:m), se da cuando dos entidades están relacionadas de tal forma, que cada ocurrencia en A esta relacionada con cero, una o más ocurrencias de B; pero cada ocurrencia de B esta relacionada con a lo más una ocurrencia de A. En la fig. 1.4 se describe gráficamente lo anterior.

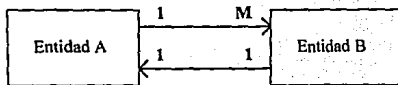


Fig. 1.4 Asociación uno a muchos.

Como ejemplo, se consideran dos tablas A y B con sus respectivos valores, como se muestra en seguida:

A	B
PK	PK
a1	b1
a2	b2
a3	b3
	b4
	b5

Para modelar la asociación uno a muchos (1:m) de A a B, es necesario que la llave primaria (PK) de A sea puesta como llave foránea (FK) en B. El resultado que se obtiene de lo anterior es:

B/A	
PK	FK
b1	a1
b2	a3
b3	a1
b4	a1
b5	a3

Este tipo de relación no es simétrica. En el la tabla siguiente se muestra, que la llave primaria de B fue puesta como llave foránea en A. El resultado que se obtiene es diferente.

A/B	
PK	FK
a1	b1
a2	---
a3	b2

Asociación muchos a muchos.

La asociación muchos a muchos (m:n), se presenta cuando dos entidades están relacionadas de tal forma que cada ocurrencia de A se relaciona con cero, una o más ocurrencias de B, y cada ocurrencia de B se relaciona con cero, una o más ocurrencias de A. Esto se describe en forma gráfica en la fig. 1.5.

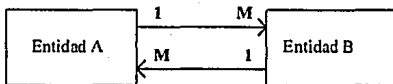


Fig. 1.5 Asociación muchos a muchos.

Como ejemplo, consideremos dos tablas A y B tal como se muestra a continuación:

A
PK
a1
a2
a3

B
PK
b1
b2
b3

Para modelar este tipo de asociaciones, es necesario definir una tabla asociativa. Esta tendrá una llave primaria compuesta, cuyos componentes individuales serán las llaves primarias de A y B. Los componentes individuales de la llave primaria en la tabla asociativa, también son llaves foráneas. A continuación se muestran formas diferentes de modelar las asociaciones muchos a muchos.

a)

A/B	
PK	+
FK	FK
a1	b2
a1	b1
a2	b2
a3	b1

b)

B/A	
PK	+
FK	FK
b1	a1
b1	a3
b2	a2
b2	a2

Como se observa, este tipo de asociación es simétrica.

El tipo de tabla que resulta de las asociaciones m:n es una tabla asociativa, (una tabla no prima), cuyos componentes son todas llaves foráneas (posiblemente compuestas).

Una regla es que una tabla asociativa siempre modela relaciones m:n y las relaciones m:n son siempre modeladas en tablas asociativas.

1.5 NORMALIZACIÓN.

La normalización es un procedimiento mediante el cual es posible prevenir anomalías de inserción, borrado y cambios en la base de datos (sección 1.3). La normalización fue originalmente inventada por el Dr. Codd.

En esta sección se explicarán las tres formas de normalización para las bases de datos relacionales.

Primera forma normal. Una relación esta en primera forma normal, si todos los dominios de sus atributos son atómicos.

La finalidad de la primera forma normal, es la de simplificar la estructura de la base de datos. Si tenemos más de un valor a un tiempo para un atributo, se define una nueva entidad. Lo que se busca es eliminar grupos de repetición.

Por ejemplo, consideremos una entidad llamada SA_VL que representa a una caja con productos X, a la cual debe de darse seguimiento durante su viaje (estatus). SA_VL tiene los atributos siguientes:

VL : clave de la caja (varios estatus), FACTURA : La factura que ampara la caja (o VL).
PROVEEDOR : Quien es el proveedor de la caja, UNIDAD_ENVIO : En donde viene la caja
STATUS : Estatus de la caja (VL) y USUARIO : Responsable del estatus.
El siguiente cuadro, por ejemplo, muestra los posibles valores de la tabla:

vl
factura
proveedor
unidad_envio
status 1
usuario 1
status 2
usuario 2
status 3
usuario 3
.
.
.

En el cuadro anterior puede observarse que existe más de un valor de atributo al mismo tiempo; en otras palabras, para un mismo VL se tienen varios STATUS / USUARIO.

La forma correcta de resolver este problema, es definiendo una nueva entidad, la cual es descrita por ese atributo, es decir, ésta segunda entidad tendrá los atributos STATUS y USUARIO y estará asociada por medio de una llave (VL) con la primera entidad.

Este tipo de relación guarda una asociación 1:m con la relación original. En la fig. 1.6 se describe gráficamente lo anterior.

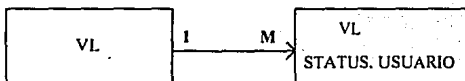


Fig. 1.6 Relación 1:m con SA_VL y la nueva entidad.

La segunda forma normal. Una relación está en segunda forma normal, si todo atributo no primo (no llave) es del todo dependiente de la llave candidata (llave que se elige para identificar en forma única a una tupla). [Deen, 1987, pág. 137]

En otras palabras, cualquier porción de la llave primaria no debe tener asociación 1:1 con otro atributo de la misma relación.

La finalidad de la segunda forma normal, es la de evitar anomalías en la inserción, borrado y modificaciones de los datos.

Como ejemplo de esto, tomaremos la tabla SA_COMBINACION, que fue la relación que resultó de SA_CENTRO_COSTO y SA_REQ en la sección 1.3 con los valores siguientes:

SA_COMBINACION

REQ_NUMERO	CENTRO_COSTO	CLAVE_MOVIMIENTO	PLANTA	NUMERO PARTE
290001297	2288	A	601	N 01024011
290001297	1200	B	501	N 01024011
290001297	3000	C	602	BAQ049111AB
290001299	2288	A	601	5011205AC725
290001299	2288	A	501	N 01024011
290001307	1200	A	600	511205113AC725

En la tabla SA_COMBINACION, REQ_NUMERO y CENTRO_COSTO identifican en forma única a cada atributo, REQ_NUMERO por sí solo no identifica en forma única al centro de costos y éste a su vez no identifica en forma única a PLANTA/NUMERO_PARTE.

CENTRO_COSTO puede identificar en forma única a la clave de movimiento; por ésta razón se dice que la clave de movimiento es parcialmente dependiente.

La solución para esto es definir una nueva relación con la parte de la llave que identifica a el(los) atributo(s) en forma única.

A continuación se muestra un ejemplo, en el cual se aplica lo anterior. Observemos que SA_COMBINACION se separó en dos tablas: una de ellas SA_REQ, y una segunda, a la que para fines del ejemplo llamaremos SA_CENTRO_COSTO.

SA_REQ

REQ_NUMERO	CENTRO_COSTO	PLANTA	NUMERO_PARTE
290001297	2288	601	N 01024011
290001297	1200	501	N 01024011
290001297	3000	602	BAQ049111AB
290001299	2288	601	5011205AC725
290001299	2288	501	N 01024011
290001307	1200	600	511205113AC725

SA_CENTRO_COSTO

CENTRO_COSTO	CLAVE_MOVIMIENTO
2288	A
1200	B
3000	C

La relación que resulta de la separación guarda una asociación m:1 con la original. En fig.1.7 se describe en forma gráfica.

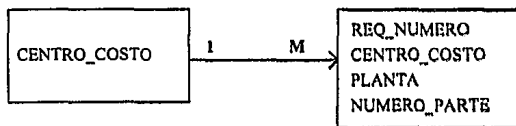


Fig. 1.7 Relación 1:m con SA_CENTRO_COSTO y SA_REQ.

En SA_COMBINACION, podemos decir que PLANTA y NUMERO_PARTE son del todo funcionalmente dependientes de CLAVE_MOVIMIENTO y CENTRO_COSTO, ya que estos últimos son necesarios para identificar en forma única a los primeros.

CLAVE_MOVIMIENTO puede ser identificado en forma única por CENTRO_COSTO, por eso se dice que CLAVE_MOVIMIENTO es parcialmente dependiente.

Haciendo analogía con una función matemática, la segunda forma normal se formaliza de la siguiente manera:

1.- Sea una relación A con dominio $DOM(A)$ y una relación B con dominio $DOM(B)$

$$f1) \quad f: DOM(A) \rightarrow DOM(B)$$

Donde f es un conjunto de pares ordenados.

En $f1$ a cada punto del dominio de A le debe corresponder un punto en el dominio de B, si esto sucede entonces B es funcionalmente dependiente de A y A es funcionalmente determinante de B.

2.- Sean f y g dos funciones definidas como:

$$\begin{aligned} f2) \quad & f: A_1, \dots, A_n \rightarrow B \\ f3) \quad & g: A_1, \dots, A_m \rightarrow B \end{aligned}$$

$$m < n \text{ y } f(a_1, \dots, a_n) = g(a_1, \dots, a_m) \text{ para todo } a_i \text{ en } A_i \quad 1 \leq i \leq n$$

Si existen dos funciones tales que cumplan las condiciones de $f2$ y $f3$, se dice que B es parcialmente dependiente de A_1, \dots, A_n .

Si no existe una g con las propiedades descritas arriba, se dice que B es totalmente dependiente en A_1, \dots, A_n .

La dependencia parcial causa anomalías en la inserción y borrado, dando como resultado problemas de inconsistencia.

La tercera forma normal. Una relación está en tercera forma normal, si está en segunda forma normal y los atributos que no estén en la llave primaria no tienen dependencias transitivas sobre la llave. [Tsichritzis y Lochovsky, 1977, pág. 51].

Cualquier relación en tercera forma normal tiene la propiedad de hacer que los atributos que no están en la llave primaria (atributos no primos) no tienen dependencia parcial ni transitiva sobre la llave. Esto quiere decir que todos los atributos no primos son independientes entre sí; pero sólo los atributos no primos son los que tienen esta propiedad. La tercera forma normal no considera los atributos primos, que también pueden causar problemas de inconsistencia.

La tercera forma normal fue originalmente definida por Codd en 1971. Esta fue posteriormente mejorada por Boyce y es conocida como **forma normal de Boyce-Codd**.

Antes de dar una definición de la forma normal de Boyce-Codd, se explicará lo que es un *determinante*.

Un **determinante**, es un atributo o conjunto de ellos, no redundantes, que pueden actuar como identificador único de otro atributo o conjunto de ellos. Si el conjunto de atributos A es un determinante del conjunto de atributos B, se dice que B es del todo funcionalmente dependiente de A. [Deen, 1987, pág. 135].

Se considera una relación R, cuyo atributo r1 identifica en forma única a la tupla; pero existe otro atributo r2 que también puede identificar en forma única a la tupla. Se dice que r1 y r2 son llaves candidatas, una de ellas será elegida para identificar a la tupla; a ésta se le llamará *llave primaria*.

En la relación R, si r1 es concatenado con cualquier otro atributo de la relación, será también identificador único; pero el segundo atributo es redundante, que va en contra de la definición de llave candidata, puesto que no debe haber atributos redundantes.

Forma normal Boyce-Codd: Una relación está en tercera forma normal si y sólo si, para cada llave candidata, cada atributo que no está en esa llave es total y directamente dependiente de ella. [Deen, 1987, pág. 135].

Por ejemplo si se tiene una relación R

A	B	C	D	E
a1	b1	c1	d1	e1
a2	b1	c2	d1	e1
a3	b2	c3	d2	e2
a4	b3	c4	d3	e3
a5	b3	c5	d3	e3

La llave candidata para esta relación es A, respecto de la cual todos los atributos son del todo dependientes (A B,C,D,E); como A es atributo único, no existe el problema de la dependencia parcial.

Aplicando la forma normal Boyce-Codd a la relación anterior, encontramos que D y E a pesar de ser totalmente dependientes de A (A D,E), no son directamente dependientes de ella porque son totalmente dependientes de B (B D,E), que a su vez es totalmente dependiente de A (A B).

Una segunda definición para la tercera forma normal es la de Sharman (1975).

Una relación está en tercera forma normal, si cada determinante es una llave candidata. [Deen, 1987, pág. 135].

La anterior definición pertenece a Sharman (1977).

Si aplicamos esta definición a la relación R(A,B,C,D,E) tenemos que B determina a D y también a E, es decir, (B D) y (B E), B es un determinante, pero no una llave candidata, y por eso la relación no está en tercera forma normal, el segundo determinante es A que determina a C, D, E (A C,D,E) y es una llave candidata.

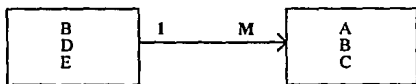


Fig. 1.8 Asociación m:1 entre la entidad original y nueva entidad.

Para que la relación R(A,B,C,D,E) esté en tercera forma normal, se debe descomponer en dos relaciones, en donde la segunda relación tiene una asociación m:1 con la relación original. La fig. 1.8 describe lo anterior en forma gráfica.

1.6 DIAGRAMA ENTIDAD RELACIÓN.

La estructura lógica general de una base de datos, se puede expresar gráficamente por medio de un diagrama entidad relación, el cual está formado por:

- Rectángulos, que representan las entidades.
- Rombos, que representan las asociaciones entre entidades.
- Líneas, conectan entre si las entidades.
- Rectángulo doble, para representar entidades debiles.

Por ejemplo si se consideran las entidades A y B, las cuales tienen una relación $m:n$, la forma de representar gráficamente lo anterior en el diagrama entidad relación es:



Si la relación entre A y B fuera de $1:m$ la representación gráfica quedaría como:



Y finalmente, si la relación entre A y B fuera de $m:1$ el diagrama entidad relación es:



El diagrama entidad relación (E-R), representa: la información de una organización, las entidades, los atributos (propiedades de las entidades) y las relaciones entre los atributos.

Las entidades existen por sí mismas, pueden ser tangibles, tal como edificios, empleados, etc. intangibles como departamentos en una empresa (contabilidad, aseo, etc.) o semi-tangibles, como pedidos, envíos, etc.

Atributo. Es una característica o cualidad de la ENTIDAD, tal como nombre, dirección, teléfono en la entidad PERSONA. Número de pedido, contenido, fecha de embarque en la entidad, PEDIDO.

Técnicamente todas las columnas de una TABLA son atributos.

Relación. Una relación, es una asociación entre una o más entidades, acerca de la cual el sistema debe mantener, correlacionar y desplegar cierta información. (Las relaciones también contienen atributos).

El objetivo del diagrama entidad-relación, es el de proveer un modelo exacto de la información necesaria de la organización, para generar un modelo independiente de los métodos de almacenamiento o acceso.

El diagrama E-R que se explica en esta sección, es el que el diseñador del manejador ORACLE ocupa y difiere un poco al explicado antes en cuanto a la representación gráfica.

El diagrama entidad relación del diseñador de ORACLE, ocupa rectángulos con las esquinas redondeadas para representar las entidades, y no existen los rombos ni los rectángulos dobles. las relaciones m:n también se representan con rectángulos de esquinas redondeadas.

Ya que fue el diagrama E-R del manejador ORACLE el que se uso para la representación gráfica de las entidades y relaciones del sistema que se presenta en el capítulo dos, se explicará con más detalle a continuación. De aquí en adelante cada vez que se haga referencia al diagrama entidad relación, se entenderá que es el manejado por el diseñador de ORACLE, a menos que se indique lo contrario.

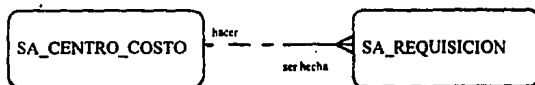
En el diagrama entidad-relación, se modelan las entidades con un nombre, que generalmente son sustantivos en singular, los nombres de las entidades se escriben con letras mayúsculas. Sobre las líneas que unen estas entidades, se escriben con letras minúsculas los verbos para indicar acciones realizadas y hacer más explícita la relación. Esto se verá en los ejemplos siguientes.

Existen los siguientes tipos de líneas:

Opicionalidad		Cardinalidad	
—————	Debe ser	————— <	Una o más
- - - - -	Puede ser	—————	Una y sólo una

Las líneas sirven para representar el tipo de asociación entre las entidades.

Observemos lo siguiente:



Lo anterior se lee como : Cada requisición, *debe ser hecha* por uno y sólo un centro de costos y cada centro de costos *puede hacer* una o más requisiciones. donde centro de costos es un lugar X en donde se hacen ciertas requisiciones Y.

Entre SA_CENTRO_COSTO y SA_REQUISICION existe una línea punteada, que indica que el centro de costos, puede o no hacer una o más requisiciones; ya que la cardinalidad de uno o más se indica por la punta de la flecha que llega hacia SA_REQUISICION. De la misma forma la línea de SA_REQUISICION a SA_CENTRO_COSTO, es continua, lo cual indica obligatoriedad, así, ésta se lee: la requisición debe ser hecha por un centro de costos y la cardinalidad de SA_REQUISICION a SA_CENTRO_COSTO es a lo más de uno, ya que no es posible que dos centros de costos hagan la misma requisición.

En los diagramas entidad-relación, no se permite la intersección entre las líneas, esto con el fin de hacerlo más claro.

Algunas otras relaciones que puede haber en el diagrama entidad relación son como las que muestran a continuación:

opcional a opcional



Esta relación nos dice que tanto A como B pueden existir sin que exista relación entre ellos.

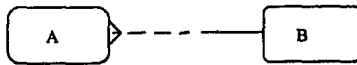
La siguiente relación es muy poderosa y un tanto difícil de implementar; por ello se debe tener cuidado, ya que nos indica que una instancia de B no puede ser creada sin la creación simultánea de al menos una asociación de A.

obligatorio a obligatorio

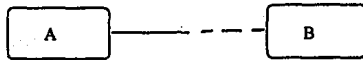


Las siguientes asociaciones son raras:

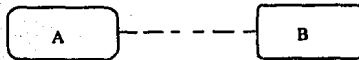
opcional a obligatorio



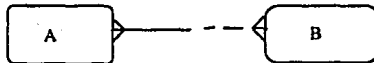
obligatorio a opcional



opcional a opcional

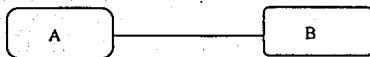


obligatorio a opcional



Una asociación que indica que algo no está bien y que debe ser revisada con más detalle es:

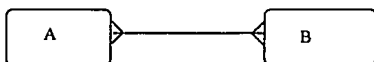
obligatorio a obligatorio



Una relación como la anterior indica que se trata de la misma tabla.

En la siguiente relación se debe tener mucho cuidado, porque si llegara a presentarse, indicaría una inconsistencia en la información.

obligatorio a obligatorio



Esta relación, indica que ninguna instancia en A puede existir sin al menos una instancia en B y viceversa. En la práctica siempre han resultado producto de información errónea.

Una relación muy común es:

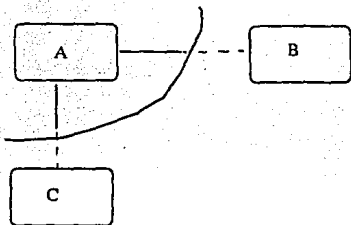
opcional a opcional



Aunque muy parecida a la anterior, indica que una instancia en A puede existir sin necesidad de que exista una relación con B y viceversa.

Es importante mencionar que ésta relación es muy común en las etapas tempranas del análisis, e indica por lo general que la relación entre las entidades no ha sido del todo comprendida. De cualquier manera, en un principio no debe ser motivo de gran preocupación.

Otro tipo de relaciones son las llamadas **relaciones exclusivas**, y se modelan de la siguiente manera:



Esta relación indica que cada "A" debe tener uno y sólo un "B" o debe tener uno y sólo un "C", pero no ambos.

Concluiremos éste capítulo explicando como se obtienen las tablas a partir del diagrama entidad relación.

Los diagramas E-R se reducen a tablas de la base de datos, esto es, una base de datos que se ajuste a un diagrama entidad relación, puede ser representada por medio de un conjunto de tablas. Para cada entidad y cada relación, existe una tabla única. Cada tabla contiene un número de columnas que también tienen nombres únicos.

Dada una entidad E, con su correspondiente identificador de tupla (llave primaria) y con atributos $a_1, a_2, a_3, \dots, a_n$; ésta se representa como la tabla cuyo nombre es E con "n" columnas, cada una de las cuales corresponde a cada uno de los atributos de E.

Al tipo de entidades que poseen una llave primaria, se les denomina entidades fuertes; sin embargo, existen otro tipo de entidades que por no tener suficientes atributos, no tienen una llave primaria; este tipo de entidades se les conoce como entidades débiles.

Las entidades fuertes son dominantes, mientras que las entidades débiles son subordinadas, es decir, dependen de una entidad fuerte.

En caso de tener una entidad E que no cuente con una llave primaria, procederemos de la siguiente forma: se tomará el o los atributos que distingan en forma única a los diferentes tipos de ocurrencia que puede haber en dicha entidad y se unirán con la llave primaria de la entidad de la cual depende.

Por ejemplo, supongamos que se tiene una tabla TABLA1 (VL, FACTURA, PROVEEDOR) en donde VL es la llave primaria; si tomamos otra tabla TABLA2 (STATUS, USUARIO, FECHA), en donde STATUS identifica en forma única a USUARIO, FECHA para una VL determinado, entonces la llave primaria para la tabla TABLA2 es VL,STATUS. El atributo STATUS contiene los distintos tipos de ocurrencia que puede haber para los estatus.

Si tenemos un conjunto de relaciones R que implica las entidades E_1, E_2, \dots, E_n y suponiendo que las llaves primarias de las entidades son: e_1, e_2, \dots, e_n respectivamente, se formará una tabla que se asociará con cada una de las entidades involucradas y cuyos atributos serán las llaves primarias e_1, e_2, \dots, e_n .

En una asociación m:n (muchos a muchos) la relación entre las dos entidades, se modela por medio de una nueva entidad, en donde sus componentes son las llaves primarias de cada una de las entidades involucradas, tal como se vio en la sección 1.4.

Con esto termina la exposición sobre la teoría de las bases de datos relacionales. En los capítulos posteriores, se implementa una aplicación de las bases de datos relacionales.

CAPITULO 2

ESTUDIO DE UN CASO: ANALISIS Y DISEÑO

En éste capítulo se trata el estudio de un caso, que consiste en una base de datos de una fábrica de automóviles compactos.

El problema de la fábrica es el de poder dar seguimiento a los materiales para la construcción de los autos. Los materiales son empacados en Alemania y enviados al estado de Puebla, en donde son recibidos y almacenados para su uso.

Para la fábrica es de gran importancia poder dar seguimiento a los materiales en todo momento, desde que vienen hacia México, entran a territorio mexicano y son procesados en la misma fábrica.

Los resultados obtenidos en el análisis, se muestran en éste capítulo.

Primero se presenta el plan de trabajo, el cual contiene las etapas para el análisis del sistema; después se explica el funcionamiento del mismo, se incluye el diagrama funcional, el diagrama entidad relación, así como las tablas del sistema. Por último, se toca el tema de ambientes integrados, ya que para éste sistema se utilizó CASE (Computer Aided Software Engineering) para facilitar las tareas.

2.1 PROPUESTA.

2.1.1 Identificación del problema.

De acuerdo con las pláticas que se sostuvieron con los directivos de la fábrica de automóviles, se concluyó lo siguiente:

Antecedentes.

La fábrica de automóviles, inicia una nueva etapa en lo que a sistemas se refiere, al plantear la descentralización de las aplicaciones que se manejan actualmente. Pretende optimizar los recursos computacionales en diversas áreas de la empresa. Dentro de éste movimiento de descentralización, están comprendidos los sistemas correspondientes al área de administración de almacenes, con esto se pretende, entre otras cosas:

- 1.- Lograr un tiempo de respuesta de aproximadamente tres segundos para el manejo de transacciones interactivas.
- 2.- Una mejor administración de los recursos informáticos del área.
- 3.- Lograr un alto grado de integración entre los sistemas descentralizados.
- 4.- Lograr un alto nivel de integridad y seguridad en la información.

Las partes que componen el área de administración de almacenes son:

- a) Sistema de Administración de Materiales (S.A.M.).
- b) Partes Seltas (P.S.).
- c) Entrada de Material al Almacén (E.M.A.).

La empresa ORACLE proporcionará a la fábrica de autos, una solución integrada, involucrando los productos de ORACLE para así poder explotar de manera eficiente la base de datos.

A continuación se muestra el plan de trabajo propuesto para analizar el sistema.

2.1.2 Etapas del sistema ORACLE.

Cada una de las etapas, se compone de un número definido de actividades, que deberán realizarse, para considerar que la etapa se ha cumplido.

1.- Análisis del sistema.

En esta etapa se recopilará la información del sistema actual, tomando como base la información contenida en las carpetas de los sistemas ya existentes, así como los esquemas de la base de datos. También se diseñan los cuestionarios que se aplicarán a los usuarios y se realizan las entrevistas correspondientes.

Las actividades para esta etapa son las que se muestran en la tabla 1.

2.- Desarrollo de los servicios de información.

Es la construcción de las estructuras de programación de manera automática, con el empleo de generadores de formas, que permiten portar las aplicaciones a diferentes plataformas.

Las actividades a desarrollar en esta etapa son las que se muestran en la tabla 2.

3.- Construcción de las aplicaciones y documentación.

Durante ésta etapa se construye la totalidad de la aplicación como una especie de prototipo progresivo, es decir, se manejan prototipos enfocados a:

I. Mantenimiento de datos.

II. Pantallas y reportes.

Los prototipos se validan en diversas sesiones con los usuarios, documentando de forma paralela la aplicación.

Las actividades para esta etapa aparecen en la tabla 3.

ACTIVIDADES	RESULTADOS
a) Recopilación de la información a.1) Diagramas del sistema actual. a.2) Documentación general del sistema. a.3) Entrevistas y cuestionarios al personal de sistemas y usuarios.	a) Modelo entidad relación. b) Tablas especificadas en un contexto normalizado. c) Modelo funcional.
b) Análisis de la base de datos b.1) Identificación de entidades. b.2) Identificación de los elementos de datos. b.3) Asignación de los elementos de datos a entidades.	d) Unidades de programación definidas.
c) Diseño del modelo	
d) Evaluación del modelo d.1) Planteamiento de posibles rutas nuevas de navegación de la base de datos. d.2) Evaluación de la asignación de los elementos de datos.	
e) Definición de las unidades de programación para desarrollar.	

Tabla 1. Actividades y resultados de la etapa de análisis del sistema.

ACTIVIDADES	RESULTADOS
a) Especificación de las unidades de programación.	a) Unidades de programación especificadas.
b) Generación de la base de datos ORACLE.	b) Base de datos.
c) Elaboración de un plan de migración.	c) Plan de migración.

Tabla 2. Actividades y resultados de la etapa de desarrollo de los servicios de información.

ACTIVIDADES	RESULTADOS
a) Desarrollo de formas interactivas.	a) Programas en SQL*Forms
b) Desarrollo de los programas procedurales y reportes.	b) Programas en SQL*Menu
c) Afinación del prototipo de presentación.	c) Programas en ReportWriter
	d) Programas en 3GL
	e) Documentación de las aplicaciones a nivel técnico y de usuario

Tabla 3. Actividades y resultados de la etapa de construcción de las aplicaciones y documentación.

4.- Transición y puesta en producción.

En esta etapa se aplica el plan de migración. Es muy importante validar el paso de la información a la base de datos, ya que de esto depende la integridad y la consistencia de los datos.

Otro aspecto que debe cuidarse, es el de entrenamiento a usuarios, ya que estos deberán contar con los conocimientos necesarios en el momento de la liberación del sistema, porque de esto depende en buena parte el éxito en la operación del mismo.

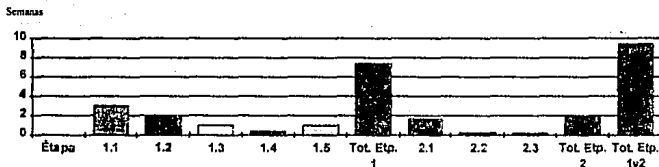
Las actividades a realizar en ésta etapa final se muestran en la tabla 4.

ACTIVIDADES	RESULTADOS
a) Importación de datos históricos.	a) La base de datos con el total de información histórica.
b) Evaluación del trabajo.	b) Reportes de los resultados del proyecto.
c) Entrenamiento a usuarios finales.	c) Material educacional.
d) Ajustes a la plataforma ORACLE.	d) Usuarios entrenados en el empleo de la aplicaciones.
e) Liberación.	e) Posibles ajustes a la aplicación en función del "performance".
f) Supervisión.	f) La aplicación liberada.

Tabla 4. Actividades y resultados de la etapa de transición y puesta en producción.

De acuerdo a las etapas anteriores y las actividades que se realizan en cada una de ellas, se estiman los tiempos para la realización de las mismas.

En la gráfica 2.1 aparecen los tiempos estimados para las etapas 1 y 2.

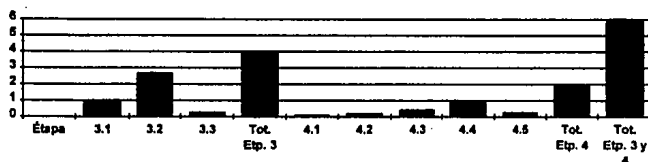


Gráfica 2.1 Tiempos estimados para las etapas 1 y 2.

En la gráfica 2.1, las etapas 1.1, 1.2, 1.3, 1.4 y 2.3 solo se considera una persona, el resto de las etapas tres, por esto los tiempos estimados se reducen a 1/3. El tiempo estimado para concluir las etapas 1 y 2 es de nueve semanas aproximadamente.

La gráfica 2.2 aparecen los tiempos estimados para las etapas 3 y 4.

Semanas



Gráfica 2.2 Tiempos estimados para las etapas 3 y 4.

En la gráfica 2.2 se consideran 3 "consultores junior", por eso los tiempos estimados se reducen a 1/3. En la etapa 4.2 y 4.4 sólo se considera a una persona.

En la tabla 5 están los tiempos estimados en horas hombre para cada actividad.

ACTIVIDAD	TIPO DE RECURSO			
	Gerente	Consultor		Técnico
		Senior	Junior	
1.1 Recopilación de la información.	40	120		
1.2 Análisis de la base de datos.	8	80		
1.3 Diseño del modelo de información	20	40	40	
1.4 Evaluación del modelo.		14	16	
1.5 Definición de la unidades de programación	14	40	120	
2.1 Especificación de las unidades de programación.	10	40	200	
2.2 Generación de la base de datos ORACLE.	2	6	32	
2.3 Elaboración de un plan de migración.	4	8		
3.1 Desarrollo de formas interactivas.	8	24	120	
3.2 Desarrollo de la lógica procedural y reportes.	8	30	320	
3.3 Afinación del prototipo de presentación.	4	10	30	
4.1 Ejecución en la importación de datos.		1	4	
4.2 Evaluación del trabajo.	8	8		
4.3 Educación a usuarios finales.	4	8	50	
4.4 Ajustes a la plataforma ORACLE				40
4.5 Liberación	4	8	30	
Total de horas estimadas	134	437	962	40

Tabla 5. Tiempos estimados por etapa por horas hombre.

2.2 ANÁLISIS DEL SISTEMA.

En la etapa de análisis, se trabajó de acuerdo con el plan propuesto. Se asignaron cinco personas de la fábrica de autos para resolver las dudas y preguntas que surgieron durante la etapa.

Una vez realizadas las actividades correspondientes a la etapa, se concluyó que el sistema trabaja de la siguiente manera:

El proceso inicia cuando los materiales son empacados en Alemania y finaliza cuando los automóviles son terminados en la línea de producción. Diagrama 2.1.

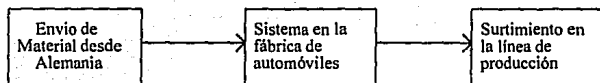


Diagrama 2.1 Inicio y fin del proceso en la fábrica de automóviles.

El sistema se divide a su vez en otros dos subsistemas que son:

- 1.- Subsistema de tránsito.
- 2.- Subsistema de recibo (Almacén).

En tránsito se maneja todo lo relacionado con el material, desde que es enviado de Alemania, hasta que llega a la fábrica. En recibo se controlan todos los materiales desde que llegan a la fábrica, son ubicados y se atienden las requisiciones. Diagrama 2.2.

Subsistema de tránsito.

Desde Alemania el material es enviado hacia puerto mexicano, luego es empacado en cajas que después son puestas dentro de contenedores (especie de cajas grandes parecidas a un vagón de ferrocarril), estos son conocidos también como unidades de envío, registrados en una factura. Los materiales también son registrados en una segunda factura que es la que utiliza la fábrica y es conocida como *factura interna*.

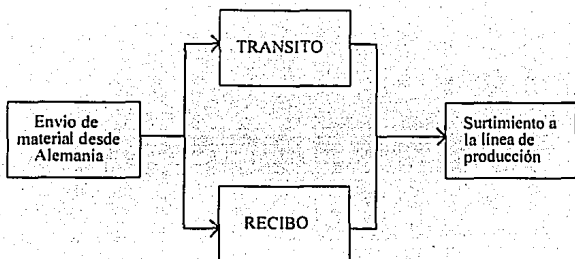


Diagrama 2.2 Subsistemas de tránsito y recibo en la fábrica de automóviles.

Desde Alemania a los contenedores les es asignado un número de identificación, conocido como *número de VL o VL*, simplemente. La fábrica necesita conocer el nombre del barco que transporta los contenedores y algunos datos de la factura, tales como el proveedor y la fecha.

Una vez que las unidades de envío (contenedores) llegan a la fábrica, se les asigna un espacio físico para ser almacenados, descargados y posteriormente ser devueltos a la línea marítima que los rentó.

El espacio físico asignado a los contenedores, es identificado por tres coordenadas, que son área, fila y estiba, en donde las áreas son espacios para la ubicación de los contenedores, estos se dividen filas y estibas (como las parejas ordenadas en un plano cartesiano respectivamente). Es posible que a algunos contenedores se vuelvan a cargar de material antes de ser regresados, a estos se les asigna un nuevo número de identificación (vl) en Puebla.

Para dar seguimiento al contenedor, se tienen los estatus que aparecen en la tabla 6.

En la fig. 2.1 se describe gráficamente los estatus. Cuando el contenedor está cargado de material, es muy común llamarlo *VL*, es decir, un contenedor cargado es sinónimo de *VL* y el material que lleva dentro, también tiene un estatus asignado, estos estatus aparecen en la tabla 7 y se explican a continuación.

Para actualizar los estatus en 1A y 1B la información es enviada por teleproceso desde Alemania.

Ciclo del contenedor.



Patio 8



Cargado.



Vacío.



Salio/planta

Cargado para expor.

Fig 2.1

STATUS	DESCRIPCIÓN
0	Contenedor no existe en la fábrica.
1	Estacionado en patio 8 (lugar dentro de la fábrica).
2	Contenedor lleno de importación (aún sin descargar).
3	Contenedor vacío.
4	Cargado para exportación (algunas veces).
5	Salio de la fábrica.

Tabla 6. Estatus de los contenedores.

Alemania tiene contenedores previamente designados para México, los cuales en raras ocasiones cambian, también los barcos que transportan esos contenedores se tienen ya designados para México. En la fig. 2.2 se describen gráficamente los estatus.

Para el estatus 1A, un contenedor vacío en Alemania comienza a ser llenado para su envío a puerto alemán. El instante en que la primera caja es puesta dentro del contenedor en Puebla se recibe la información referente al material que se está enviando: contenedor (número de vl), factura, proveedor, la orden de compra y el folio, después de que el contenedor fue llenado, se envía a puerto alemán. El diagrama 2.3. resume de forma gráfica la información recibida por el subsistema de tránsito.

STATUS	DESCRIPCIÓN
1A	Empacado en Alemania.
1B	Envío de puerto alemán.
1C	Llegada a puerto mexicano.
1D	Liberado por aduana.
1E	En carretera con destino a Puebla.
1F	Llegada a patio 8 (especie de anexo en la fábrica).
2A	Llegada a la fábrica (vigilancia).
2B	Solicitud de descarga.
3	Entrada de material.
4	Verificación del material.
5A	Inicio de envío de material al almacén.
5B	Envío total de material al almacén.
----	En recibo (status nulo).
5	Enviado al almacén.
6	Material en almacén.

Tabla 7. Estatus del material del contenedor.

Teleproceso



Ciclo del material por estatus.

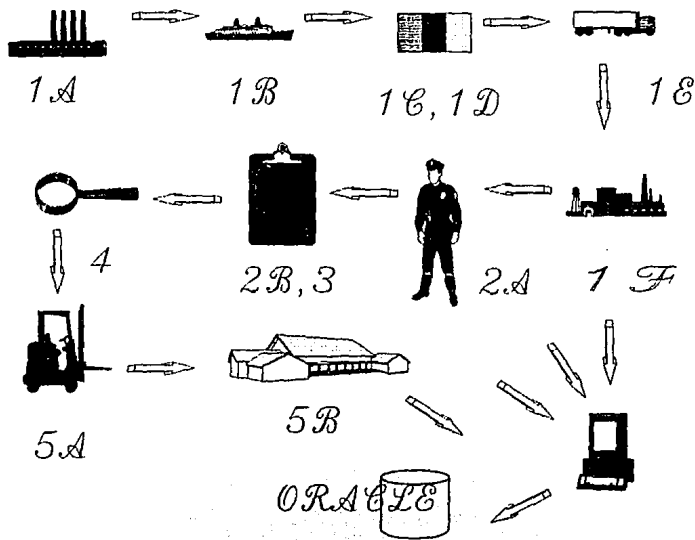


Fig. 2.2

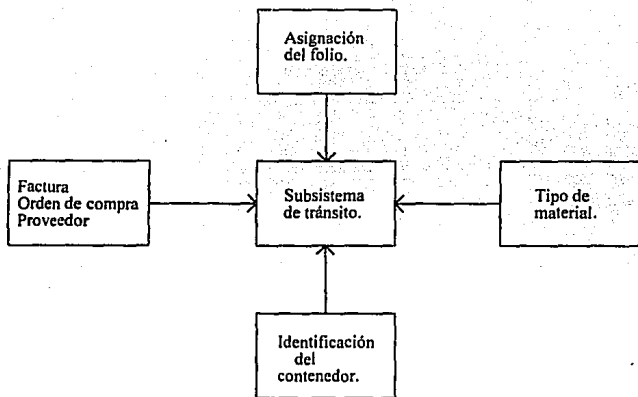


Diagrama 2.3 Información recibida por el subsistema de tránsito.

Cuando los contenedores son enviados a puerto alemán, el estatus es actualizado a 1B. Se dice que el material está en tránsito si está entre los estatus 1A a 2B.

Si el material del contenedor está en estatus 2B, el contenedor es descargado y devuelto a la línea marítima, ya sea vacío o cargado con nuevo material.

Aunque para los estatus posteriores (3 a 5B), el material se encuentra sólo en cajas, éstas son puestas en grupos diferentes si pertenecen a unidades de envío diferentes. A cada una de las cajas le es puesta una etiqueta con el número de contenedor (vl) del que fueron descargadas. Sin embargo, el material es considerado a nivel contenedor de los estatus 1A a 5B, porque la costumbre de los trabajadores es hacer referencia a cada grupo de cajas o a una caja del grupo, como el número de contenedor al que pertenecieron y no como los números asignados a ellas.

Los estatus nulo, 5 y 6 son asociados a las cajas que contiene la unidad de envío, es decir a nivel folio. Después del estatus 4 (verificación del material), se inicia el envío a los almacenes. Con esto se actualiza el estatus del material a 5A, pero a nivel folio se le asigna

el estatus 5; es decir, todas las cajas de un determinado contenedor que están siendo enviadas en ese momento al almacén, tienen el estatus 5.

El estatus no será actualizado a 6, hasta que la totalidad de las cajas pertenecientes al contenedor hayan sido enviadas al almacén. La actualización de el estatus a 6, se hace por lo general en las noches, aunque ya se haya actualizado a nivel folio al estatus 5B (nivel contenedor).

El material a nivel contenedor se considera en recibo si está entre los estatus 3 a 5B. Para que el material a nivel folio se considere en recibo, el estatus debe ser nulo.

Es necesario dar seguimiento de los estatus de la unidad de envío (contenedor) a lo largo de todo su ciclo, por ello se guarda una historia de todo los estatus para cada unidad de envío que esté utilizando, aquí, debe detectarse si existe más de una unidad de envío con el mismo VL con destino a Puebla, es decir VL's duplicados en estatus 1A a 2B (en tránsito); esto no debe permitirse, ya que crearía confusión al no corresponder el material con la orden de compra. En un caso extremo la fábrica tendría lo que se conoce como paro de línea, que es cuando la producción se detiene.

El diagrama 2.4, ejemplifica de forma gráfica la información requerida para el contenedor.

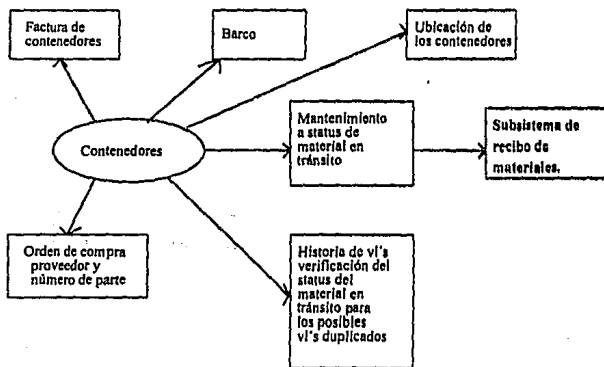


Diagrama 2.4 Información requerida para el contenedor.

Existen algunos puntos importantes, que pertenecen a la liga entre los subsistemas de tránsito y recibo. A continuación se explica lo anterior.

El tipo de material se identifica por un código que está formado por 15 caracteres, éste indica el origen (nacional o extranjero) que se forma con los tres primeros caracteres del código, los caracteres restantes, indican a qué parte del auto debe ser colocado el material (interiores, tablero, puertas, motor etc.); este código es conocido como *planta/número de parte*.

El número de parte es verificado en el estatus 1A cuando, por teleproceso, se envía el tipo de material que se introduce en el contenedor.

Para el número de parte (material), es necesario conocer quién fue el *proveedor* y cuál la *orden de compra*, ésta última: es un compromiso entre la fábrica y el proveedor, en donde se especifica qué material es vendido (número de parte) y la cantidad, ya que uno de los problemas, es que la fábrica no sabe si realmente existe un contrato con ese proveedor por la cantidad que se está surtiendo. En ocasiones los proveedores surten material sin ser realmente proveedores, o la cantidad no es la correcta.

Se contempla en el futuro el lanzamiento de un nuevo modelo de automóvil y para ello deben considerarse ordenes de compra aparte.

El material a nivel folio es revisado para detectar posibles errores, estos son conocidos como *errores-prestop* los cuales pueden ser:

TIPO DE ERROR	DESCRIPCIÓN
1	No existe la combinación: planta/número_parte.
2	No existe proveedor.
3	No existe la combinación : planta/numero_parte/proveedor.
8	La cantidad recibida es mayor que la cantidad autorizada en la orden de compra.

Se le conoce como *RMR* al material que esta libre de errores o que posiblemente los tuvo pero ya fueron corregidos. El diagrama 2.5, muestra gráficamente lo anterior.

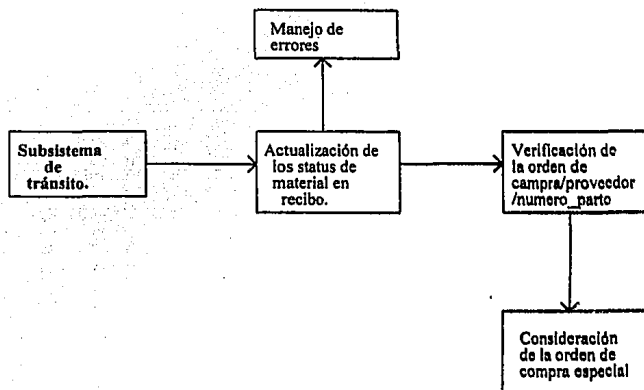


Diagrama 2.5 Liga de los subsistemas de tránsito y recibo.

Subsistema de recibo.

El material a nivel folio es ubicado en cierto lugar; la ubicación se identifica por el área, fila y la estiba (igual que para los contenedores). El área es un lugar determinado que se asigna al material, la fila y la estiba corresponden a las coordenadas X y Y de un plano cartesiano respectivamente; pero además de esto, las cajas del material son puestas una sobre otra y por ello se debe especificar el nivel de estiba.

Es muy importante conocer la ubicación correcta del material para no tener faltantes y/o sobrantes en el momento de hacer el inventario y el máximo y mínimo número de existencias permitido para planta/numero_parte.

Existe un catálogo para las áreas de ubicación validas en la fábrica, cuando el material llega al almacén, se le asigna una ubicación fija; es decir, una área, fila, estiba y nivel, pero algunas veces ésta ya fue ocupada, en éste caso, la persona encargada de ubicar el material en ese lugar (montacarguista), puede colocarlo en un sitio diferente al propuesto, esto no

es problema, pero ocasiona que algunas veces el material esté en un lugar diferente al propuesto.

El material debe poder ser encontrado tanto por su número de parte como por su folio. la ubicación del material por folio es más confiable que la ubicación por número de parte, por lo ya explicado en el parrafo anterior.

Hasta aquí se ha explicado el sistema en cuanto a materiales y su almacenamiento, en seguida se explica el proceso de pedidos de material.

Las requisiciones de material se hacen por medio de unidades administrativas denominadas *centros de costos*. Un centro de costos puede hacer mas de una requisición a la vez.

En la línea de producción existen personas encargadas de verificar si los obreros cuentan con el material necesario para llevar a cabo sus actividades y eliminar la posibilidad de que haya un paro de línea. A esas personas se les conoce como *seguidores*, y el lugar en donde se pone el material en la línea de producción se le conoce como *tacto*; así pues los seguidores son los encargados de revisar los tactos asignados a ellos.

Quando un folio no es suficiente para surtir una requisición de material, se asignan índices a todos los folios necesarios para surtir el pedido, se lleva el control de estos índices para saber cual es el último, y ver si el pedido fue surtido en su totalidad o no. Si el pedido no fue surtido en su totalidad, se conoce como BACKORDER (pedidos no surtidos).

El último índice será utilizado en caso de que la requisición no se haya completado. Este indica la cantidad faltante para surtir, y en caso de requerirse se hará uso del índice inmediato superior.

Una requisición también estará en BACKORDER si no hay material para surtirla. Si hay material en estatus 3 y hubiera un BACKORDER con ese material, éste será utilizado de inmediato para surtir el pedido, sin importar los estatus superiores.

Es posible que una caja de gran tamaño se divida para ser almacenada en cajas más pequeñas, a esto se le llama *traspaleo*, y existen dos tipos de traspaleo:

- 1.- Traspaleo sin salida a producción. Es el traspaleo que se realiza del almacén
- 2.- Traspaleo con salida a producción. Cuando se pide una determinada cantidad de material y ésta cantidad es menor a la que contiene la caja, el material se traspalea para surtir sólo la cantidad requerida.

Los almacenes están comprendidos en otra unidad llamada *nave*; es decir, una nave contiene varios almacenes.

Los movimientos de material realizados en los almacenes por la gente que ahí trabaja, deben ser registrados para posibles aclaraciones.

La fábrica tiene proveedores nacionales y extranjeros, los cuales hacen trabajos de corrección y/o reparación de algunas piezas defectuosas (retrabajo), también se compran llantas, tornillos, y algunos materiales para acabados. A estos materiales se les da seguimiento a partir del estatus 3.

Cada uno de los proveedores identifica su material con su propio número de parte, que no necesariamente es el mismo que utiliza la fábrica; de tal forma que debe realizarse una conversión del número de parte. Los proveedores también tienen unidades de medida diferentes, y por tanto, debe haber una tabla de conversiones para las unidades de medida.

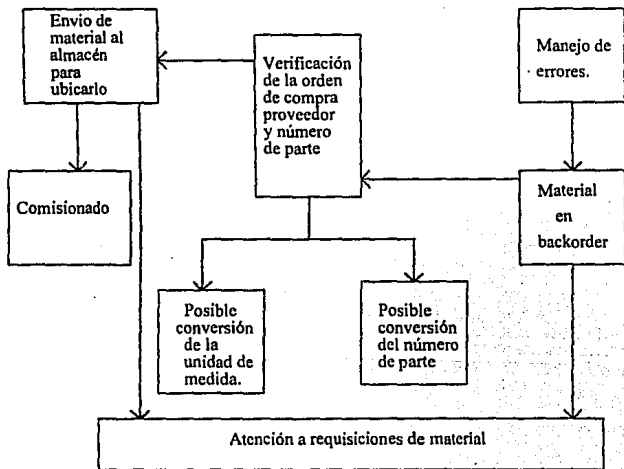


Diagrama 2.6 Subsistema de recibo.

Por otra parte, resulta costoso hacer pedidos al almacén, tal es el caso de piezas pequeñas como tornillos, tuercas, etc. Estos costos se optimizan creando un almacén más pequeño, que contiene las piezas más chicas que se utilizan constantemente, dejando para el "almacén grande" las piezas mayores.

El "almacén chico" hace pedidos regularmente al "almacén grande" para tener existencias suficientes, en éste último se traspalea el material, y cuando esto sucede se dice que el material se comisiona.

El diagrama 2.6 muestra gráficamente todo lo anterior.

Con base en la información de ésta sección se diseñó el diagrama de funciones y el diagrama entidad-relación, los cuales se muestran en la siguiente sección.

2.3 DIAGRAMA FUNCIONAL.

Básicamente el sistema se divide en 9 funciones principales, éstas se describen gráficamente en el diagrama 2.7 y se explican a continuación:

Función 1. Administración de material en tránsito: es donde se da seguimiento al material de los estatus 1A a 2B. Esta función se divide a su vez en:

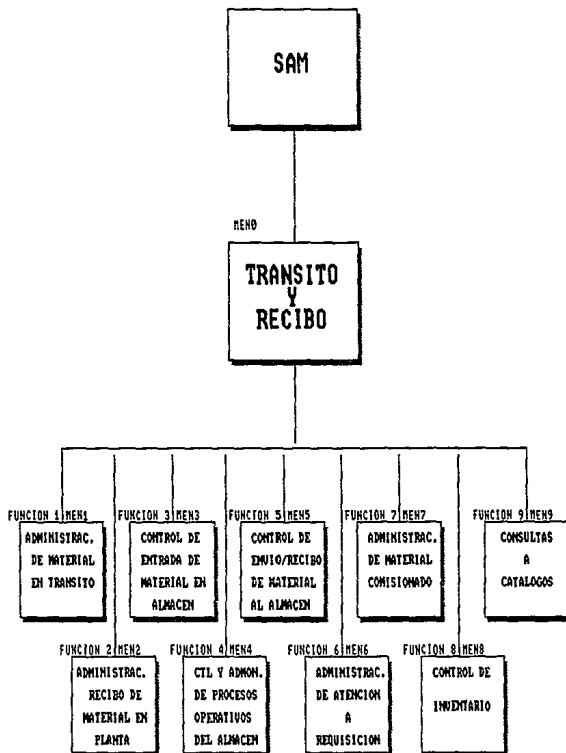
a) Recibo de información de material en tránsito: ésta función tiene el control de la llegada de los contenedores a frontera mexicana, liberación de aduana de los contenedores, salida de los contenedores a carretera, llegada de los contenedores a patio 8, llegada de los contenedores a la planta en la ciudad de Puebla. La información es consultada en todo momento, para la verificación del proceso del contenedor.

b) Reportes de contenedores en tránsito: ésta es básicamente una función para consultas desde el estatus 1A hasta el estatus 1E, se consultan contenedores por factura, en tránsito, etc.

c) Reportes de contenedores a exportación: a diferencia de la función anterior, aquí se realizan las consultas del contenedor, pero como medio de almacenamiento para exportación, importación, etc.

En el diagrama 2.8, aparece de forma gráfica la función 1 y sus divisiones.

FUNCIONES PRINCIPALES DEL SISTEMA S.A.M.



FUNCION 1.

FUNCION 1. RES 1

ADMINISTRACION
DE MATERIAL EN
TRANSITO

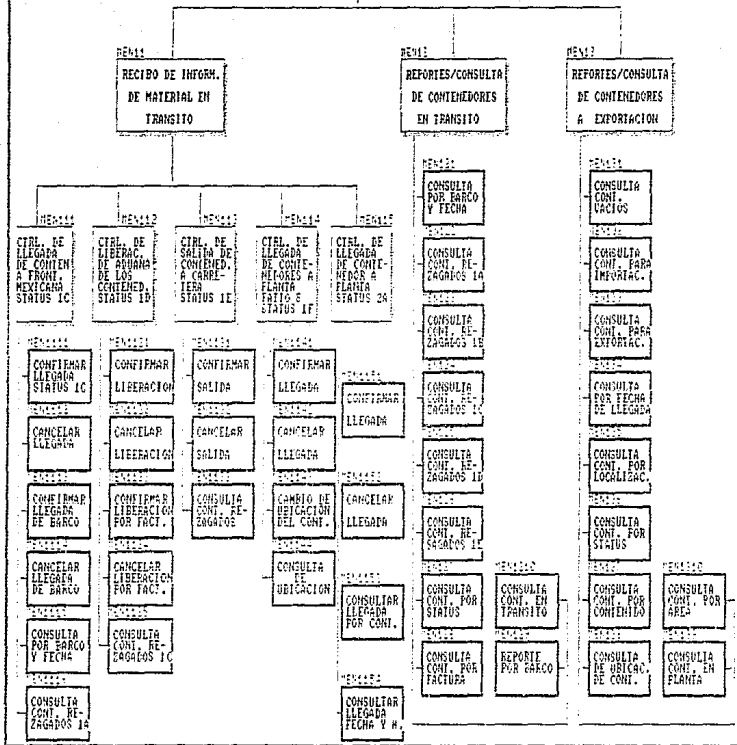


DIAGRAMA 2.8

Función 2. Administración de recibo de material en planta: ésta función, es para el control del material de los estatus 2B a 4, así como para dar seguimiento a los contenedores. La función 2 se divide a su vez en:

a) Control de solicitud de descarga del contenedor: ésta función registra la solicitud de descarga del contenedor, una vez que ha llegado a la planta.

b) Control de entrada a existencias: ésta función es muy importante, ya que es la que sirve para verificar los errores en el material. Se registra el material por contenedor, se consulta el material por folios etc.

c) Control de verificación y ajuste a existencias: aquí se hacen los cambios al número de parte, se revisan las diferencias del material y se hacen las consultas al mismo.

d) Correcciones por contenedor: aquí se hacen los cambios de factura del contenedor, altas y bajas de los contenedores, correcciones a la caja por folio, etc. Se incluye también, la alta y baja de material en tránsito.

e) Control de contenedores de exportación: ésta función, como su nombre lo indica, es para dar seguimiento a los contenedores listos para exportación, con respecto a su salida y ubicación.

En el diagrama 2.9 aparece la función 2 y sus divisiones.

Función 3. Control de entrada de material en almacén: controla el recibo del material en almacén. Se registra la entrada del material procedente de recibo y el que fue devuelto por producción. Se realizan las consultas necesarias para saber las diferencias entre lo enviado y lo recibido y las consultas por número de parte.

El diagrama 2.10a describe gráficamente la función 3.

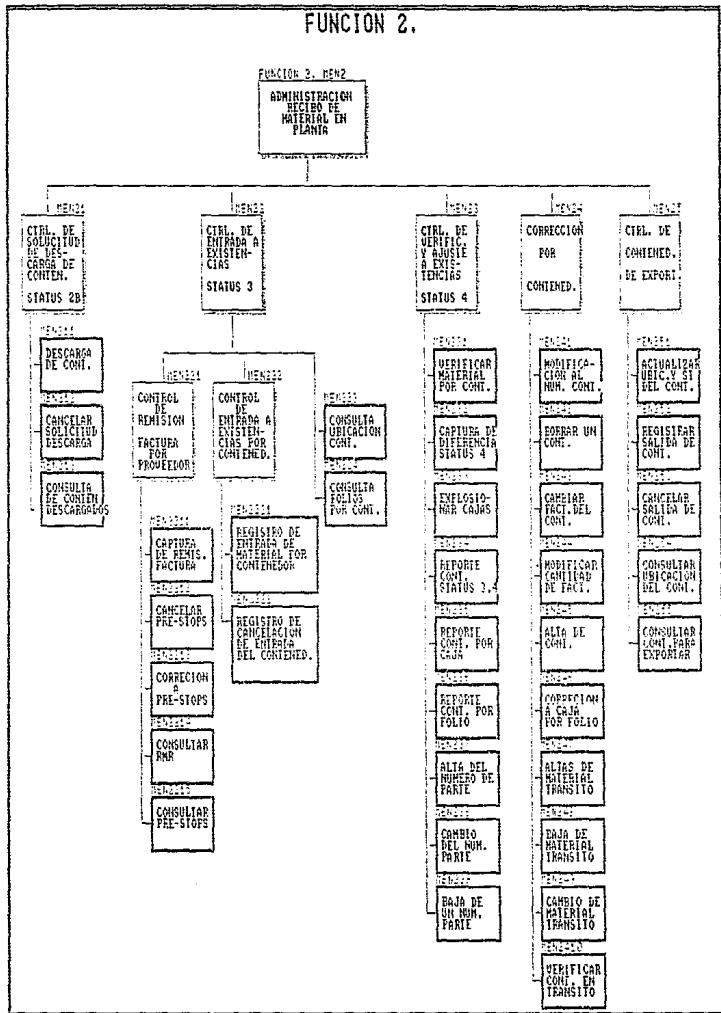
Función 4. Control y administración de los procesos operativos del almacén: se encarga de controlar los movimientos internos en el almacén, se divide a su vez en:

a) Administración de la ubicación fija: para el mantenimiento de las ubicaciones fijas.

b) Movimientos internos de material en el almacén: registra la ubicación del material, reubicación del mismo y traspaleo.

c) Reportes del material en el almacén: se emiten reportes para conocer las existencias en el almacén, tránsito, material con requisición, material sin localización, etc.

FUNCION 2.



FUNCIONES 3 Y 4.

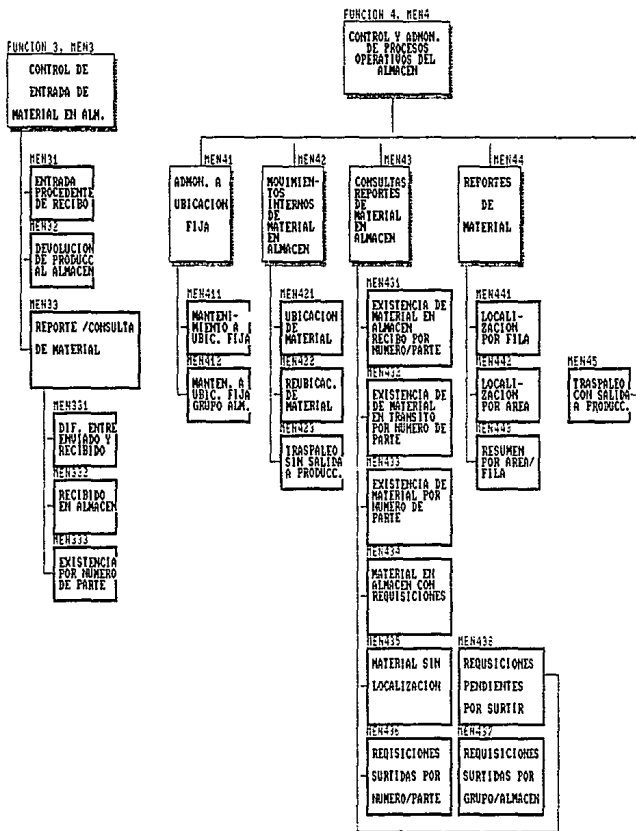


DIAGRAMA 2.10A

DIAGRAMA 2.10B

- d) Reportes de material: se emiten reportes de la localización del material por área, fila.
- e) Traspaleo con salida a producción: se registran los materiales que fueron traspaleados.

El diagrama 2.10b describe de forma gráfica la función 4.

Función 5. Control de envío/recibo de material al almacén: ésta función sirve para registrar las entradas de material que tienen lugar en el almacén, lo que esta pendiente para ser surtido y cancelaciones de envío al almacén.

El diagrama 2.11a describe de forma gráfica la función 5.

Función 6. Administración de atención a requisiciones: controla los pedidos que la línea de producción hace al almacén, las requisiciones surtidas y las no surtidas. Esta función se divide en:

- a) Atención a requisiciones: registro de las requisiciones.
- b) Actualización de requisiciones: se actualizan las requisiciones pendientes por surtir y las que están en BACKORDER.
- c) Consultas y Reportes: se emiten los reportes de las requisiciones surtidas por grupo de almacén, número de parte y las que faltan de surtir.

En el diagrama 2.11b se describe de forma gráfica la función 6.

Función 7. Administración de material comisionado: se encarga de controlar el almacén de piezas chicas y emite los reportes de salida de material, requisiciones, etc. Se divide a su vez en:

- a) Entrada de material en almacén: se registra todo lo que es introducido al almacén.
- b) Salidas de comisionado: actualiza las salidas del material y emite el reporte.
- c) Consultas para comisionado: ésta lleva el control de las requisiciones que faltan de surtir, las que están en BACKORDER (cero existencias) y el reporte de todo el material comisionado (la cantidad de material que tengo).

El diagrama 2.12a describe la función 7 gráficamente.

FUNCIONES 5 Y 6.

FUNCION 5. MEN5

CONTROL DE ENVIO
RECIBO DE MATERIAL
AL ALMACEN

MEN61

CAPTURA DE
ENVIO DEL
FOLIO AL
ALMACEN

MEN62

CANCELAR
ENVIO AL
ALMACEN POR
FOLIO

MEN63

CONSULTA DE
DIFERENCIAS
ENTRE ENVIO
Y RECIBO

MEN64

CONSULTA DE
MATERIAL
PENDIENTE POR
ENVIARSE

MEN65

ENVIO AL
ALMACEN POR
CONTENEDOR

MEN66

CANCELACION
DE ENVIO
DE MATERIAL
AL ALMACEN
POR CONTENEDOR

FUNCION 6. MEN6

ADMINISTRACION
DE ATENCION A
REQUISICIONES

MEN61

ATENCION
A
REQUISIC.

INEN62

ACTUALI-
ZACION
DE
REQUISI-
CIONES

MEN63

CONSULTAS
REPORTE

MEN621

PENDIENTES
POR
SURTIR

MEN624

REQUISIC.
PENDIENTES

MEN622

CERO EN
BACKORDER

MEN623

REQUISICIONES
SURTIDAS POR
GRUPO ALMACEN

MEN625

REQUISICIONES
SURTIDAS POR
NUMERO/PARTE

DIAGRAMA 2.11A

DIAGRAMA 2.11B

FUNCIONES 7, 8 Y 9.

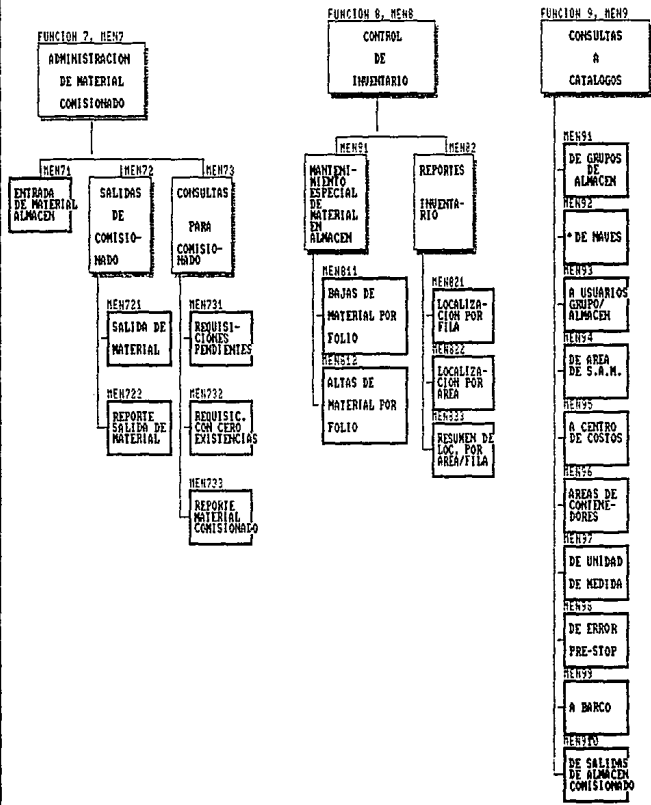


DIAGRAMA 2.12A

DIAGRAMA 2.12B

DIAGRAMA 2.12C

Función 8. Control de inventario: ésta lleva el control de la rectificación del material en almacén con el fin de ubicarlo en el lugar que debe estar, detecta los sobrantes o faltantes. Esta función se divide en:

a) Mantenimiento especial de material en almacén: ésta se utiliza cuando hay inventario, se reorganiza el material por folio. El material que no está registrado se da de alta, y el que no se encuentra físicamente, se da de baja.

b) Reportes de inventario: emite los reportes para la localización del material por área y fila.

La función 8 se describe gráficamente en el diagrama 2.12b.

Función 9. Consulta a catálogos: ésta función es para hacer las consultas a los catálogos que existen en el sistema como almacenes, barcos, áreas, centros de costo, etc.

El diagrama 2.12c describe gráficamente la función 9.

Como podrá observarse, en todo momento en los diagramas de funciones, existe algún tipo de consulta. El sistema es muy transaccional, y es de gran importancia que se conozca en todo momento la disposición del material, es por ello que se hacen tantas consultas.

Con esto finalizamos esta sección, en donde mostré como fueron divididas las funciones para el sistema, las cuales deben ser soportadas por el diagrama entidad relación, que se presenta en la sección siguiente.

2.4 DIAGRAMA ENTIDAD RELACIÓN.

En el capítulo uno se dijo que una base de datos que se ajuste a un diagrama entidad-relación, puede representarse, por medio de un conjunto de tablas. Para cada entidad en el diagrama entidad-relación, existe una tabla única, con un determinado número de atributos, que también son únicos dentro de esa tabla.

A continuación, se hacen algunas observaciones sobre el diagrama entidad relación (diagrama 2.13), y posteriormente se presentan las tablas del sistema en orden alfabético junto con los atributos que las conforman.

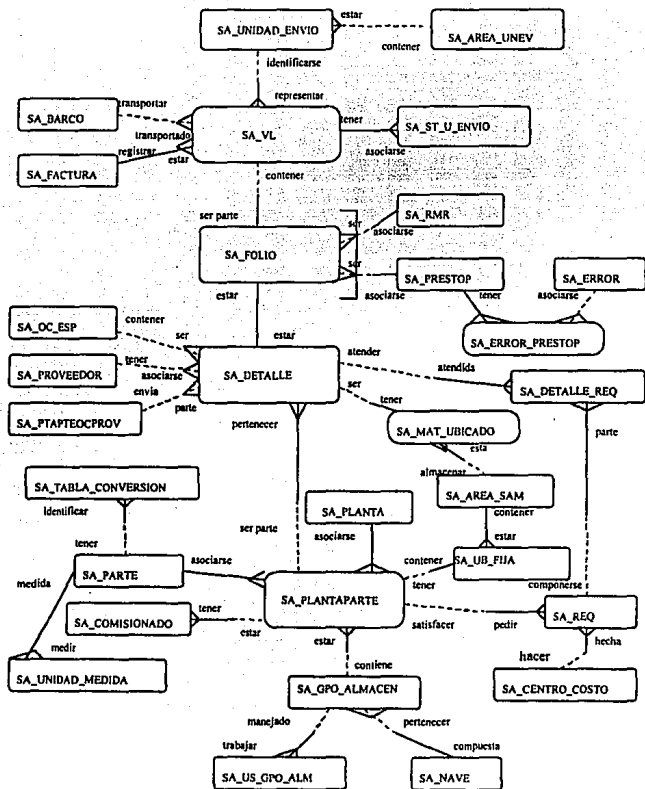


Diagrama 2.13 Diagrama entidad relación.

Observaciones:

Entre las tablas SA_DETALLE y SA_MAT_UBICADO, existe una relación 1:1, sin embargo ésta es opcional, por tanto la relación es perfectamente válida.

Existe un arco para las tablas SA_RMR y SA_PRESTOP relacionadas con SA_FOLIO, que como se señaló en su oportunidad, indica que el FOLIO, puede ser RMR (libre de errores) o puede ser PRESTOP (con errores), pero no ambos a la vez.

Se observa también que existen tablas asociativas, que modelan las relaciones m:n, una de ellas es la de SA_ERR_PRESTOP, que asocia las tablas de SA_ERROR y SA_PRESTOP, ya que un PRESTOP se da cuando existe al menos un ERROR y el ERROR puede estar en mas de un PRESTOP.

La tabla SA_PLANTAPARTE, asocia a cada PLANTA uno o mas NUMERO_PARTE y para cada NUMERO_PARTE, existe mas de una PLANTA.

En el diagrama entidad-relación, la tabla SA_FOLIO y SA_DETALLE se relacionan 1:1; esto según el capítulo 1, viola una de las reglas del diseño de bases de datos relacionales, ya que un FOLIO en SA_FOLIO no puede existir sin la presencia de un FOLIO en SA_DETALLE y viceversa.

Lo anterior indica que SA_FOLIO y SA_DETALLE, son una misma tabla; esto en realidad es cierto, pero ésta decisión fue tomada, porque algunos usuarios sólo trabajan con una parte de los atributos de la tabla; los atributos restantes son ocupados por otro sector de usuarios; además, el volumen de información que se maneja es tan grande que la tabla tuvo que ser dividida en dos para almacenarla en discos distintos, con esto se reduce la contención y se acelera el tiempo de respuesta, con ésto se cumple el primer objetivo que se planteo en la propuesta del sistema.

Con la integración de las áreas de tránsito y recibo en almacén, así como la facilidad del sistema para emitir diversos reportes, la seguridad de la información se garantiza, y por tanto se cumplen en su totalidad, los objetivos restantes que se plantearon en un principio.

Para finalizar el capítulo, se hablará de la herramienta CASE (Computer Aided Software Engineering) el cual jugó un papel decisivo para llevar a buen término la construcción del sistema con el diseño de diagramas y documentación generada en forma paralela.

Nombre de la tabla y Atributos	Llave Primaria	Llave Foranea	DESCRIPCION
SA_AREA_SAM AREA DESCRIPCION NOT NULL CHAR(4) NOT NULL CHAR(30)	AREA	----	ALMACENA TODAS LAS AREAS VALIDAS. CATALOGO DE AREAS PARA MATERIAL A NIVEL FOLIO
SA_AREA_UENV AREA DESCRIPCION NOT NULL CHAR(3) NOT NULL CHAR(30)	AREA	----	CATALOGO DE AREAS VALIDAS PARA LOS CONTENEDORES
SA_BARCO BARCO NOT NULL CHAR(17)	BARCO	----	HOMBRES DE LOS BARCOS VALIDOS
SA_CENTRO_COSTO CENTRO_COSTO DESCRIPCION NOT NULL CHAR(4) NOT NULL CHAR(30)	CENTRO_COSTO	----	CATALOGO DE LOS CENTROS DE COSTOS VALIDOS
SA_CONSISTOMADO PLANTA NUMERO PARTE NOT NULL CHAR(3) NOT NULL CHAR(15) CAJAS PEQUENAS NOT NULL CHAR(12) NOT NULL NUMBER(7) COMI_PORCENTAJE_DEVOLUCION NOT NULL NUMBER(3) COMI_UBICACION CHAR(11) ORIGEN CHAR(3)	PLANTA/NUMERO PARTE /ORIGEN	PLANTA/NUMERO PARTE	CONTROL DE LAS PIERAS PEQUEAS EN EL ALMACEN CONSISTOMADO. (ALMACEN CHICO)
SA_DETALLE FOLIO NOT NULL NUMBER(12) PLANTA NOT NULL CHAR(3) NUMERO PARTE NOT NULL CHAR(15) STATUS CALIDAD NOT NULL CHAR(3) CANT_LOCALIZADA NOT NULL NUMBER(12) CANT_FACTURADA NOT NULL NUMBER(12) CANT_RECIBIDA NOT NULL NUMBER(12) OC_ESPECIAL CHAR(6) OC CHAR(6) PROV NUMBER(6)	FOLIO	PLANTA/NUMERO PARTE FOLIO, OC ESPECIAL, PROV, PARTE/OC/PROV	CONTIENE INFORMACION DE IDENTIFICACION DEL MATERIAL LAS CANTIDADES FACTURADAS Y RECIBIDAS, ASI COMO LOS PROVEEDORES
SA_DETALLE_REQ REQ_NUMERO NOT NULL NUMBER(9) INDICE NOT NULL NUMBER(2) CANTIDAD_SURTIDA NOT NULL NUMBER(12) TRACTOR NOT NULL CHAR(6) FOLIO NOT NULL NUMBER(12) CUE TRASPALEO CHAR(1) FECHA SALIDA DATE ULT_EXISTENCIA CHAR(1) USUARIO SALIDA CHAR(8) ST_RECIBO CHAR(1) TIPO_SALIDA CHAR(1)	REQ_NUMERO/INDICE	FOLIO, REQ_NUMERO	REGISTRA TODAS LAS SALIDAS QUE SE HAN TENIDO EN EL ALMACEN. EL USUARIO QUE DIO SALIDA, LA FECHA, SI EL MATERIAL FUE TRASPALREADO O NO ETC.
SA_ERROR N_ERROR DESCRIPCION NOT NULL CHAR(11) NOT NULL CHAR(30)	N_ERROR	----	CATALOGO DE LOS TIPOS DE ERRORES
SA_ERROR_PRESTOP PRESTOP N_ERROR NOT NULL NUMBER(6) NOT NULL CHAR(1)	PRESTOP/N_ERROR	PRESTOP, N_ERROR	TABLA ASOCIATIVA

Nombre de la tabla Y Atributos	Llave Primaria	Llave Foranea	DESCRIPCION
SA_FACTURA FACTURA FACTURA_INTERNA FECHA PROV NOT NULL CHAR(6) CHAR(6) NOT NULL DATE NOT NULL NUMBER(6)	FACTURA/PROV	----	ALMACENA EL CODIGO DE LA FACTURA, TANTO INTERNA COMO EXTERNA, ASI COMO LA FECHA DE LA MISMA
SA_FOLIO FOLIO TRACTOR CAJA PRESTOP UL RMR OBSERVACIONES ST_DETALLE NOT NULL NUMBER(12) CHAR(4) CHAR(3) NUMBER(6) CHAR(3) NUMBER(6) CHAR(25) CHAR(2)	FOLIO	UL, RMR, PRESTOP	IDENTIFICA EL MATERIAL A NIVEL FOLIO, LAS CAJAS EN DONDE ESTA ESE FOLIO, EL ESTATUS, EL TRACTOR QUE TRANSPORTO EL MATERIAL ETC.
SA_GPO_ALMACEN GRUPO_ALMACEN RESPONSABLE EXTENSION HAVE DESCRIPCION NOT NULL CHAR(2) CHAR(30) NOT NULL NUMBER(4) NOT NULL CHAR(7) CHAR(30)	GRUPO_ALMACEN	HAVE	SOLO ES CON EL FIN DE LLEVAR UN CONTROL SOBRE LOS ALMACENES Y SUS RESPONSA- BLES
SA_MAT_UBICADO FOLIO AREA ETLA ESTIBA NIVEL USUARIO FECHA NOT NULL NUMBER(12) NOT NULL CHAR(4) NOT NULL CHAR(3) NOT NULL CHAR(3) NOT NULL CHAR(1) CHAR(8) NOT NULL DATE	FOLIO	AREA	ES EN DONDE SE ALMACENA LA UBICACION DEFINITIVA DEL MATERIAL
SA_HAVE HAVE DESCRIPCION NOT NULL CHAR(3) NOT NULL CHAR(30)	HAVE	----	CATALOGO DE LAS NAVES VALIDAS
SA_OC_ESP OC ESPECIAL DESCRIPCION NOT NULL CHAR(6) NOT NULL CHAR(30)	OC_ESPECIAL	----	DESCRIPCION DE LAS ORDENES DE COMPRA ESPECIALES. (AUN NO UTILIZADA)
SA_PARTE NUMERO PARTE PESO UNITARIO SM_HUM DESCRIPCION NOT NULL CHAR(15) NOT NULL NUMBER(8, 4) NOT NULL NUMBER(11) NOT NULL CHAR(30)	NUMERO_PARTE	----	ALMACENA LAS CARACTERISTICAS DE LOS MATERIALES, TALES COMO SU PESO, MEDIDA Y DESCRIPCION
SA_PLANTA PLANTA DESCRIPCION NOT NULL CHAR(3) NOT NULL CHAR(30)	PLANTA	----	CONTIENE LAS PLANTAS VALIDAS
SA_PLANTAPARTE PLANTA NUMERO PARTE GRUPO_ALMACEN EXISTENCIA_MAXIMA EXISTENCIA_MINIMA NOT NULL CHAR(3) NOT NULL CHAR(15) NOT NULL CHAR(2) NOT NULL NUMBER(12) NOT NULL NUMBER(12)	PLANTA/NUMERO_PARTE	PLANTA, NUMERO PARTE, GRUPO_ALMACEN	REGISTRA LAS EXISTENCIAS MAXIMAS Y MINIMAS QUE DEBE HABER PARA UNA PLANTA/PARTE ADEMAS DE FUNGIR COMO TABLA ASOCIATIVA

Nombre de la tabla y Atributos	Clase Primaria	Clase Extranera	DESCRIPCION
SA_PRESTOP PRESTOP USUARIO FECHA ENTRADA FECHA FACTURA HAVE RECIBO UVECS_CORREGIDO SU_EMA	NOT NULL NUMBER(6) NOT NULL CHAR(8) NOT NULL DATE NOT NULL DATE NOT NULL CHAR(3) NUMBER(2) CHAR(6)	PRESTOP ----	ALMACENA LA INFORMACION DEL MATERIAL LIBRE DE ERRORES. CONTIENE LA MEDIDA DEL MATERIAL, LUGAR DE RECIBO, RESPONSABLE ETC.
SA_PROVEEDOR PROV PROV_NOMBRE	NOT NULL NUMBER(6) NOT NULL CHAR(54)	PROV ----	CATALOGO DE LOS PROVEEDORES POR SU NUMERO
SA_PTAPIEOPROV NUMERO_PARTE OC PROU CANT_AUTORIZADA	NOT NULL CHAR(15) NOT NULL CHAR(6) NOT NULL NUMBER(6) NOT NULL NUMBER(12)	NUMERO_PARTE/OC/PROU ----	CONTIENE LA INFORMACION QUE SE RELACIONA LA CANTIDAD DE PARTES DE LA FABRICA DE AUTOS, SE CONECTA A EL NUMERO DE PARTE Y LA ORDEN DE COMRA
SA_RED RED_NUMERO PLANTA NUMERO_PARTE CENTRO_COSTO FACTO FECHA_RED ULTIMO_INDICE SEGUIDOR CANT_SOLICITADA CANT_REALIZADA USUARIO CANT_SURT_EVED CLAVE_URGENCIA	NOT NULL NUMBER(9) NOT NULL CHAR(3) NOT NULL CHAR(15) NOT NULL CHAR(4) NOT NULL CHAR(10) NOT NULL DATE NOT NULL NUMBER(2) NOT NULL CHAR(3) NOT NULL NUMBER(12) NUMBER(12) CHAR(8) NUMBER(12) CHAR(1)	RED_NUMERO PLANTA/NUMERO_PARTE CENTRO_COSTO	ALMACENA EL NUMERO DE LA REQUICICION, EN DONDE CADA UNA PUEDE SER SURTIDA CON MAS DE LO QUE SE PIDIO O CON MENOS. SE TIENE LA CANTIDAD SOLICITADA, EL RESP LA FECHA DE REQUICICION ETC.
SA_RMR RMR HAVE RECIBO FECHA ENTRADA USUARIO SU_EMA FECHA FACTURA UVECS_CORREGIDO OBSERVACIONES	NOT NULL NUMBER(6) CHAR(3) DATE CHAR(8) CHAR(6) DATE NUMBER(2) CHAR(25)	RMR ----	ALMACENA LA INFORMACION DEL MATERIAL LIBRE DE ERRORES, FECHA, LUGAR Y USUARIO QUE DIO DE ALTA EL RMR, LA MEDIDA ETC.
SA_ST_U_ENVIO UL STATUS USUARIO FECHA	NOT NULL CHAR(6) NOT NULL CHAR(2) NOT NULL CHAR(8) NOT NULL DATE	UL/STATUS UL	CONTROLA LOS UVECS, HA QUE NO PUEDE HABER MAS DE UN UVECS LA CANTIDAD DE UVECS SE GUARDAN EN LOS UVECS DE LOS DEPENDIENTES ESTADOS QUE ESTA TIENIENDO
SA_TABLA_CONVERSION PARTE_PROV PROU NUMERO_PARTE	NOT NULL CHAR(15) NOT NULL NUMBER(6) NOT NULL CHAR(15)	PARTE_PROV NUMERO_PARTE	CONTIENE EL NUMERO DE PARTE DEL PROVEEDOR CON SU CORRESPONDIENTE NUMERO EN LA FABRICA DE AUTOS
SA_BR_FIJA PLANTA NUMERO_PARTE AREA FIIA ESTIMA NIVEL_ESTIMA	NOT NULL CHAR(3) NOT NULL CHAR(15) NOT NULL CHAR(4) NOT NULL CHAR(3) NOT NULL CHAR(3) NOT NULL CHAR(1)	PLANTA/NUMERO_PARTE AREA	ALMACENA LA UBICACION PROVISIONAL PARA EL MATERIAL (UBICACION PROPUESTA)

Nombre de la tabla y Atributos	Llave Primaria	Llave Foranea	DESCRIPCION
SA_UNIDAD_DNVIO UNIDAD_DNVIO NOT NULL CHAR(13) AREA CHAR(3) STATUS_VE NOT NULL CHAR(1) FILA_VE CHAR(3) FECHA DATE ESTIBA_VE CHAR(3) LINEA_MARITIMA CHAR(30) USUARIO CHAR(8) YZUS NUMBER(1) CONTENIDO CHAR(10)	UNIDAD_DNVIO	AREA	CONTIENE LA INFORMACION DE LA UNICACION DE LA UNIDAD DE ENVIO, SUS MEDIDAS LA LINEA MARITIMA, EL RESPONSABLE ETC.
SA_UNIDAD_MEDIDA SM_INGLES NOT NULL CHAR(1) SM_UMM NOT NULL NUMBER(1) FACTOR_CONVERSION NOT NULL NUMBER(12,6) DESCRIPCION NOT NULL CHAR(30) NUMERO_PARTE NOT NULL CHAR(15)	SM_INGLES/SM_UMM	NUMERO_PARTE	SE UTILIZA PARA HACER LA FACTA LA COMPOSICION DEL NUMERO DE PARTE DEL PRODUCTO A EL CORRESPONDIENTE NUMERO DE PARTE EN LA FABRICA DE AUTOS
SA_US_GPO_ALM USERID NOT NULL CHAR(8) GRUPO_ALMACEN NOT NULL CHAR(2)	USERID	GRUPO_ALMACEN	CONTIENE EL USUARIO EN EL GRUPO DE ALMACEN QUE LE CORRESPONDE
SA_VL VL NOT NULL CHAR(6) FACTURA CHAR(6) PROU NUMBER(6) UNIDAD_DNVIO CHAR(13) AREA CHAR(3) FILA CHAR(3) ESTIBA CHAR(3) BARCO CHAR(17) STATUS_ULT CHAR(2) RESP_DESCARGA NOT NULL CHAR(20) PESO_BRUTO NUMBER(10,2) PESO_NETO NUMBER(10,2)	VL	BARCO, UNIDAD_DNVIO, FACTURA/PROU	ES PARA IDENTIFICAR EL CONTENEDOR, NOS INTERESA EL CONTENIDO DEL MISMO Y EN DONDE SE ENCUENTRA, ASI COMO EL RESPONSABLE

2.5 AMBIENTES INTEGRADOS.

En últimas fecha los analistas se han beneficiado con novedosos instrumentos de productividad, creados para mejorar las tareas mediante el uso de soportes automatizados, a estos instrumentos, se les denomina *tecnologías de ambientes integrados* o de manera alternativa instrumentos CASE (Computer Aided Software Engineering).

Los tres principales enfoques que el analista sigue al adoptar tecnologías de ambientes integrados son: a) Incrementar la productividad, b) Comunicarse con mayor eficacia con los usuarios y c) Integrar el trabajo que se realiza sobre el sistema.

Las tecnologías de ambientes integrados que soportan técnicas estructuradas, tales como los diagramas de flujo de datos, diccionario de datos, diagramas estructurales, diagramas de relación de entidades etc. incrementan la productividad del analista de sistemas.

Los instrumentos CASE facilitan la interacción entre los miembros del grupo, al permitir que la elaboración de diagramas, sea un proceso dinámico e imperativo, además de proveer una comunicación significativa entre los analistas y los usuarios del sistema.

A través del uso de diagramas en la pantalla, los clientes pueden ver como los flujos de datos fueron concebidos, al observarlos, pueden solicitar cambios, que podrían requerir mucho más tiempo en un sistema manual.

La integración de las actividades mediante el uso de tecnologías de ambientes integrados, mejora la comprensión de los usuarios, sobre cómo se encueptran relacionadas cada una de las partes del ciclo de vida y su interdependencia.

En seguida se explican las características de ORACLE*CASE, el cual se utilizó durante el ciclo de vida de desarrollo del sistema en la fábrica de automóviles.

El ORACLE*CASE ayuda a identificar y a articular todas las necesidades de los usuarios; el resultado es un sistema que requiere menor mantenimiento, mejora la comunicación, la calidad e incrementa la productividad.

El ORACLE*CASE provee el medio ambiente necesario para desarrollar las aplicaciones de ORACLE.

La familia ORACLE*CASE está compuesta por:

Case*Dictionary: Principal almacén de información compartida por todas las personas que se involucran en el desarrollo de sistemas.

Contiene la información para el desarrollo de sistemas que toda la organización, maneja todas las definiciones y cubre todo el ciclo de vida de desarrollo de sistemas. **Case*Dictionary**, maneja y mantiene las definiciones a través de un serie de pantallas muy sofisticadas.

Case*Designer: Herramienta gráfica para analizar y diseñar, esta completamente integrada con **Case*Dictionary**.

Rigurosas convenciones de dibujo combinadas con una representación de diagramas, hacen que la comunicación entre los desarrolladores de sistemas y usuarios finales, sea con un alto grado de entendimiento para ambas partes.

Los diagramas que **ORACLE*CASE** permite son:

- a) Diagrama entidad relación.
- b) Diagrama jerárquico de funciones.
- c) Diagrama de flujo de datos.
- d) Diagrama de matriz.

Al momento de añadir un nuevo objeto a un diagrama, se almacena inmediatamente en **Case*Dictionary**, en ese momento se revisa que no se defina algo que antes hubiera estado ya definido.

Case*Generator: Generador de aplicación completa y también está integrado con **Case*Dictionary**.

Todo el conjunto de aplicaciones pueden ser generadas en un sólo paso, con **Case*Generator** toda la información que se encuentra en el diccionario de datos se convierte en aplicaciones con un alto grado de integridad y completamente funcionales.

Con **Case*Generator** se crean las aplicaciones en **SQL*Menu**, **SQL*Forms** y **SQL*ReportWriter**.

Con la generación automática de las aplicaciones, se reducen los riesgos tradicionales en la programación.

Case*Method: Método "top-down" para el desarrollo de sistemas recomendado por **ORACLE**.

ORACLE ha encapsulado en ésta herramienta su experiencia para construir sistemas, ésta es una aproximación a la tecnología "top-down", acompañada con las necesidades de información que el negocio requiere.

Las herramientas de ORACLE*CASE ofrecen un alto nivel de integración con Case*Method, esto redundará en los resultados del sistema. Las etapas en la construcción de sistemas que soporta Case*Method son:

Estrategia: Se trabaja con personas clave, con los altos directivos para conocer las expectativas de la empresa. Se identifican los objetivos de la empresa y hacia donde se dirige, se formula un modelo y se plantean las diferentes alternativas las cuales deben ser aprobadas para continuar con la etapa de análisis.

Análisis: En el análisis debe ratificar lo encontrado en la estrategia y llegar a un nivel con suficiente detalle, para asegurar que se cumplirán los objetivos planteados. Se analizan las funciones y se involucran diagramas, en esta parte se realizan las entrevistas y cuestionarios.

Diseño: Se toman los detalles del análisis y se busca la mejor forma de cumplir con ellos. Es importante tener en mente que el verdadero diseño realmente comienza desde la estrategia. En esta etapa se estima el tamaño de la base de datos y se construye la misma.

Construcción: Se desarrolla el conjunto de programas y reportes con las herramientas adecuadas y se hacen las pruebas de "performance". La etapa de documentación, es paralela a esta etapa, conforme se hacen las correcciones a los programas se actualiza la documentación con el fin de no olvidar detalles importantes durante un posterior mantenimiento a los programas.

Transición: Ayuda a corregir los últimos detalles en el sistema y si todo fue realizado correctamente, las afinaciones serán mínimas. Los usuarios deberán familiarizarse con el sistema y aprender las nuevas funciones.

Producción: Culminación del trabajo realizado en las etapas anteriores. El nuevo sistema debe correr con la mínima intervención del personal de soporte.

En el diagrama 2.14 se presentan de forma gráfica las etapas anteriores.

Case*Exchange: Una facilidad que da el ORACLE*CASE para transferir datos entre CASE no ORACLE y Case*Dictionary.

Con esta herramienta, la información puede ser transferida entre herramientas con un Case no ORACLE en ambas direcciones de manera selectiva.

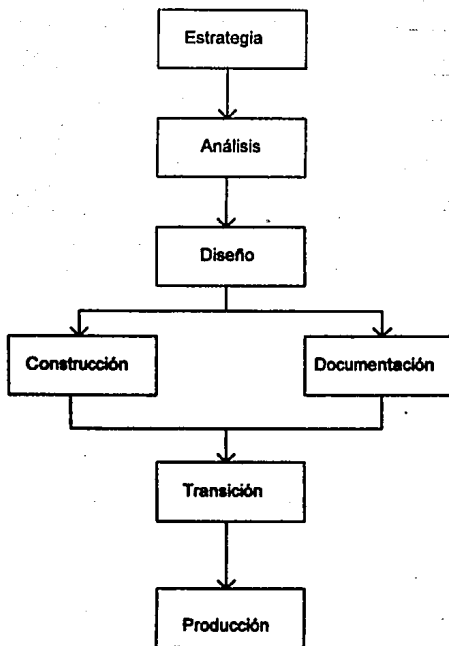


Diagrama 2.14 Etapas de la metodología de desarrollo de sistemas ORACLE.

Al finalizar la transferencia, Case*Exchange cuenta con una serie de reportes para verificar si todos los datos han sido transferidos correctamente.

Con esto termina el capítulo 2 en el cual se habló del funcionamiento del análisis del sistema y se mostraron los diagramas de funciones y de entidad relación, todo ello apoyado en las herramientas ORACLE*CASE.

En el siguiente capítulo, se presenta la arquitectura de la base de datos ORACLE y la implantación del sistema listo para arrancar.

CAPITULO 3

ESTUDIO DE UN CASO: IMPLANTACION

En este capítulo, se describe con detalle el funcionamiento del manejador ORACLE. En la primera sección se explica la estructura de archivos, en segunda sección se explican las estructuras de memoria. Ambas secciones comprenden la arquitectura completa del manejador ORACLE.

La sección tres describe la estructura de la base de datos que fue implantada en la ciudad de Puebla para la fábrica de automóviles.

3.1 ESTRUCTURAS DE ARCHIVOS ORACLE.

Una instancia ORACLE, provee los mecanismos de software para acceder y controlar la base de datos. La instancia puede hacer uso de a lo mas una bases de datos y una base de datos solo puede tener una instancia asociada a ella.

Cada instancia debe tener una área de memoria compartida conocida como SGA (System Global Area) que provee la comunicación entre procesos. Cada uno de los procesos, serán explicados en la siguiente sección.

ORACLE para su funcionamiento, requiere de varios archivos, que son:

- 1.- Data Base Files.
- 2.- Control Files.
- 3.- Redo Logs Files.
- 4.- Rollback Segments.

1.- Data Base Files.

La base de datos ORACLE, debe tener uno o mas archivos de base de datos (Data Base Files), los cuales tienen información de toda la base de datos. Estos archivos tienen las siguientes características:

- a) Un archivo sólo debe estar asociado a una y sólo una base de datos.
- b) Uno o mas archivos físicos forman una unidad lógica, llamada " tablespace".

c) El "performance" de la base de datos, será más alto, si estos archivos se encuentran lógicamente contiguos.

d) Una vez creada la base de datos, los archivos no deberán cambiar de tamaño.

El primer archivo que se crea, es el que pertenece al "tablespace" SYSTEM (tablespace que se crea automáticamente al momento de la creación de la base de datos).

Múltiples archivos pueden ser posteriormente agregados a la base de datos, pero cada uno de esos archivos sólo debe estar asociado a un "tablespace". En un sistema pequeño, podría ser suficiente con un archivo en un "tablespace".

Es importante mencionar que el máximo número de archivos para ORACLE es de 255 y el tamaño de estos archivos debe ser al menos de 500Kbytes.

Los archivos son manejados en unidades llamadas "blocks" (algunas veces llamados páginas), los bloques de la base de datos, se refieren al tamaño de bloque que ORACLE utiliza y depende del sistema operativo. Para todos los sistemas operativos, hay un tamaño de bloque ORACLE que típicamente es de 2 o 4Kbytes, el cual no debe ser alterado al menos que la mayoría de los registros en la base de datos sean mayores o menores a éste tamaño. El tamaño de bloque no puede ser alterado después de la creación de la base de datos.

2.- Control Files.

La base de datos para poder ser accedida por los usuarios, necesita ser montada y después abierta. En el primer paso, es decir, cuando la base de datos es montada, uno o más archivos de control (Control Files) son abiertos. Cuando la base de datos es abierta, los archivos de la base de datos (Data Base Files) son abiertos junto con los archivos de "redo" (Redo Logs Files).

Los "Control Files", son archivos binarios comúnmente nombrados como ORACLE.DCF y estos se asocian con una y sólo una base de datos.

Los archivos de control son creados al crear la base de datos y deberán siempre estar accesibles, ya que son requeridos cuando la instancia de la base de datos es puesta en operación y accede a la base de datos.

Los archivos de control contienen información referente a la base de datos; para que esta última pueda ser accedida por una instancia. Los archivos de control guardan el nombre físico de la base de datos, los nombres de los archivos de "redo", "timestamp" (tiempo de creación de la base de datos) etc.

Los archivos de control verifican que todo en la base de datos esté en orden y sea coherente, si uno de estos archivos llegara a dañarse y no existe respaldo, se corre un gran riesgo de perder la base de datos en su totalidad. Los archivos de control son actualizados automáticamente por ORACLE y no es posible editarlos.

3.- Redo Logs Files.

Los "Redo Logs Files", son archivos externos a la base de datos, en los cuales son grabadas las transacciones cada vez que se ejecuta la instrucción *commit* (es la instrucción para indicar a ORACLE que las actualizaciones deben ser grabadas de forma definitiva en la base de datos).

Los archivos de "redo", son utilizados para la recuperación, para restablecer ciertas transacciones perdidas o dañadas en la base de datos. Si ORACLE no puede escribir en alguno de los archivos de "redo" el RDBMS (Relational Data Base Management System) de ORACLE se detiene con un mensaje de error.

Toda la base de datos deberá tener al menos dos archivos de "redo", estos son de gran importancia, porque son los únicos en donde son escritas las transacciones una vez que se ha indicado la instrucción "commit".

Se considera que una transacción ha sido completada satisfactoriamente después de que fue escrita a los archivos de "redo".

El número máximo de archivos de "redo" es de 256 y no es necesario que estos sean del mismo tamaño, sin embargo se recomienda que lo sean, de ésta manera se garantiza que los "checkpoints" (tiempo en el cual, los Redo Logs Files aplican las transacciones escritas en ellos a los Data Base Files) ocurren a intervalos regulares de tiempo, aumentando el "performance" de la base de datos.

Las siguientes consideraciones se deben tomar en cuenta para estimar el tamaño de los archivos de "redo".

- * Los archivos de "redo" grandes tardan más tiempo en hacer "lockswiches" (acción de continuar escribiendo en el siguiente archivo de "redo" cuando se llena el anterior) aumentando la rapidez en la base de datos.

- * Los archivos de "redo" grandes, hacen que en caso de falla la recuperación sea más tardada.

El mínimo tamaño para estos archivos es de 50Kbytes, pero no hay un tamaño máximo.

Se recomienda que los archivos sean puestos en diferentes discos, para reducir el riesgo de perdida en caso de falla y para reducir la contención en las bases de datos de gran actividad. El contenido de estos archivos no debe ser editado por personas con poca experiencia.

Los archivos de "redo" pueden ser puestos fuera de línea (off-line) o en línea (on-line) para el mantenimiento a la base de datos.

Cuando en un archivo de "redo" se agota la capacidad de almacenamiento, ORACLE comienza a escribir en el siguiente archivo de "redo" en línea y así sucesivamente. Cuando se llega al último archivo de "redo", regresa de nuevo al primer archivo de "redo" para reutilizarlo, el ciclo se mantiene durante todo el tiempo de operación de la base de datos.

Al punto en donde ORACLE termina de escribir en un archivo de "redo" y comienza a escribir en el siguiente, se le conoce como *LOGSWITCH*.

Cada archivo de "redo" (o log) es identificado por ORACLE con un número de secuencia, cuando un predeterminado número de bloques han sido escritos, las transacciones se aplican de los archivos de "redo" a los archivos de la base de datos, esto también sucede cuando ocurre un "logswitch". Al proceso de aplicar las transacciones escritas en los archivos de "redo" a los archivos de la base de datos se le conoce como "checkpoint".

La fig. 3.1 describe gráficamente el proceso.

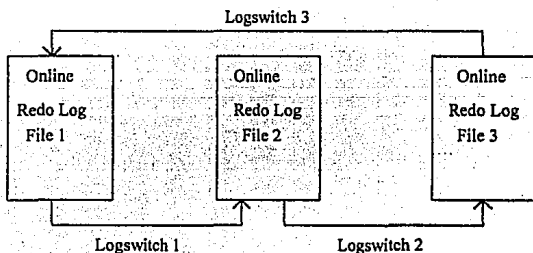


Fig. 3.1 Funcionamiento de los archivos de "redo".

Los "checkpoints" aseguran que las transacciones son aplicadas a la base de datos con cierta frecuencia. En caso de recuperación de la base de datos por falla, se hará a partir del último "checkpoint".

4.- Rollback Segments.

Los Rollback Segments o segmentos de rollback, es espacio en disco que contiene la información necesaria para deshacer cambios a los cuales no se les haya dado "commit", a esta operación se le conoce como "rollback".

Los segmentos de rollback son creados por defecto en el momento de la creación de la base de datos, pero se pueden crear segmentos de rollback adicionales, si así se requiere, esto dependerá del número de transacciones en la base de datos. Al incrementar el número de segmentos de rollback, se reduce la contención en los bloques.

Los Rollback Segments, mantienen la consistencia en las lecturas. Cuando un transacción comienza, esta se empieza a escribirse en los segmentos de rollback y no es borrada de éstos hasta que se indica la instrucción "commit" o "rollback".

Los segmentos se componen de *extents* y cada segmento de rollback debe tener al menos dos.

Si una transacción es demasiado grande y no cabe en los segmentos de rollback asignados, ORACLE puede hacer crecer estos segmentos de rollback dinámicamente.

Para estimar el tamaño de los segmentos de rollback, se recomienda dividir el número de transacciones entre el tamaño del bloque, si por ejemplo suponemos un tamaño de bloque de 4Kbytes tenemos:

Número de transacciones corrientes	Número de segmentos de rollback
Menos de 16 Entre 16 y 32 32 o mas	4 segmentos 8 segmentos $n/4$ donde n es el número de transacciones (no mas de 50 segmentos).

Estas recomendaciones pueden variar dependiendo de la base de datos, de su actividad en horas pico, etc. Si los segmentos de rollback no son suficientes, ORACLE falla con un error.

A continuación se explica como está compuesta la base de datos.

Todos los datos en el "tablespace", son guardados en espacios llamados "segments", los cuales a su vez están constituidos por "extents", que son un conjunto de bloques contiguos.

La fig. 3.2 describe gráficamente lo anterior.

Tipos de segmentos en la base de datos ORACLE.

Un "segment" no puede sobrepasar el espacio del "tablespace" que lo contiene. La base de datos ORACLE requiere de cinco tipos de "segments" los cuales se listan a continuación:

- 1.- Data Segments.
- 2.- Index Segments.
- 3.- Rollback Segments.
- 4.- Temporary Segments.
- 5.- Bootstrap Segments.

1.- Data Segment.

Los Data Segments contienen todos los datos de una tabla o "cluster", éste último es un espacio en el que se obliga a que en el mismo Data Segment se almacene más de una tabla.

2.- Index Segment.

Los Index Segments contienen los índices creados para una tabla o "cluster". Las tablas pueden o no tener índices, los clusters deben tener al menos uno.

3.- Rollback Segments.

Los "Rollback Segments", como explicamos anteriormente, permiten que los cambios a la base de datos puedan ser deshechos bajo ciertas circunstancias, permiten las lecturas de información consistentes a los demás usuarios.

Si por ejemplo una consulta (query) como SELECT contiene DISTINCT, GROUP BY y ORDER BY, tres segmentos temporales podrían ser requeridos.

Estos segmentos son borrados cuando la transacción es completada, se recomienda poner estos segmentos en un diferente "tablespace" (apendice A) para reducir la fragmentación en la base de datos.

5.- Bootstrap Segments.

Los "Bootstrap Segments", contienen información del diccionario de datos, que se cargan cuando la base de datos es abierta. Pertenecen al "tablespace system" y se crean automáticamente al momento de la creación de la base de datos.

Usualmente son pequeños, no cambian de tamaño y son accesibles a los usuarios.

3.2 ESTRUCTURAS DE MEMORIA ORACLE.

La operación de la base de datos ORACLE, depende de las estructuras de memoria y procesos, todas las estructuras de memoria, residen en la memoria principal de la computadora que corre el sistema de la base de datos.

Los procesos son trabajos o tareas que mejoran la operación en memoria, estos incluyen un proceso de usuarios y cinco procesos, llamados procesos de BACKGROUND que son:

- 1.- Data Base Writer (DBWR).
- 2.- Log Writer (LGWR).
- 3.- System Monitor (SMON).
- 4.- Process Monitor (PMON).
- 5.- Archiver (ARCH).

Las estructuras de memoria son usadas para:

- a) Almacenar código de programas ejecutables.
- b) Contener los datos necesarios durante la ejecución del programa.
- c) Compartir información y comunicar los procesos de ORACLE.
- d) Compartir información entre los procesos de usuario.

Existen varias estructuras de memoria, asociadas con el RDBMS de ORACLE y estas son:

- a) System Global Area (SGA).
 - a.1) Database Buffer Pool (memoria).
 - Data Segments Blocks y
 - Rollback Segments Blocks
 - a.2) Redo Logs Buffer (memoria).
- b) Program Global Area (PGA).
- c) Context Areas.

No debe confundirse "Database Buffers Pool" con los "Data Base Files" ni los "Redo Log Buffers" con los "Redo Logs Files" ya que unos están en memoria y los otros en disco. Esto explicará a continuación.

La SGA (System Global Area) es un conjunto de "buffers" de memoria compartida que contiene datos y el control de la información para una instancia de la base de datos ORACLE.

La SGA contiene una imagen de bloques de los archivos de la base de datos y de los archivos de "redo" log" más recientemente usados, así como información del diccionario de datos.

Se recomienda que el SGA sea lo más grande posible, esto con el fin de que mayor número de datos sean accedidos en memoria y no en disco.

Como se mencionó, el SGA contiene la imagen de los datos de los "Data Base Files" que son mayormente utilizados, es decir, el SGA contiene en memoria una copia de los datos de los "Data Files" que son utilizados con mayor frecuencia.

En la fig. 3.3 se puede ver gráficamente, la estructura de memoria que ORACLE utiliza.

En la fig. 3.3 observamos que existen dos tipos de "Data Base Buffers", unos son los "Data Segments Blocks" y otros los "Rollback Segments Blocks", estos últimos se usan por si se desea deshacer algún cambio realizado a la base de datos, esto se conoce como "rollback".

Durante una transacción los datos no son escritos inmediatamente a la base de datos, en vez de ello, tres acciones tienen lugar:

1.- Cada transacción modifica el segmento correspondiente en memoria SGA; cada bloque que reside en el "Data Base Buffer Pool", corresponde a un bloque de los archivos de la

ARQUITECTURA DE LA BASE DE DATOS ORACLE

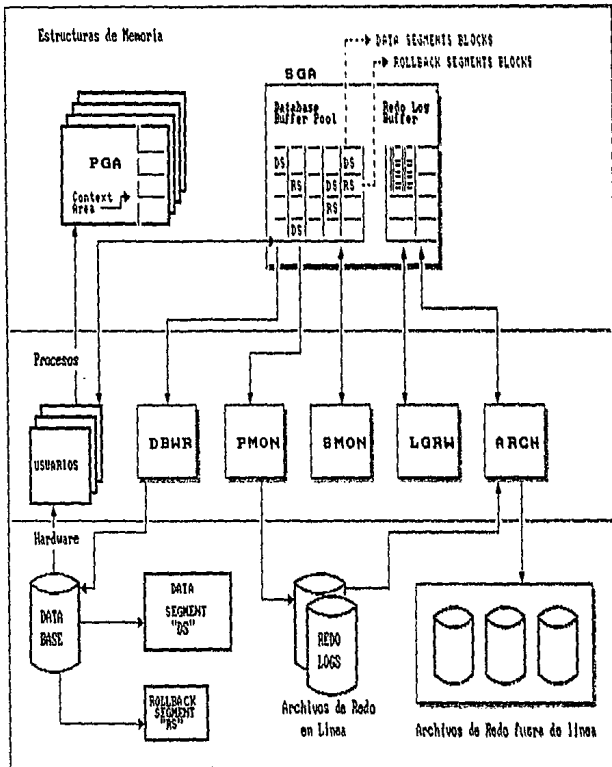


FIG. 3.3

base de datos, sino existe una copia del dato al que se le aplica la transacción en la SGA se accede al disco para obtener la información y hacer la copia en memoria SGA; posteriormente éste bloque es modificado por la transacción.

2.- Mientras lo anterior sucede, en los bloques de "Rollback Segments Blocks", se escriben copias de los bloques modificados, por ello la información puede ser consultada por otros usuarios y producir lecturas consistentes y en caso de ser necesario aplicar "rollback" a la transacción.

3.- Un registro de cada cambio realizado en los bloques de datos y bloques de "rollback" de la SGA es grabado en los "Redo Log Buffers". Cada vez que a una transacción se le aplica "commit", la información de los "Redo Log Buffers" es grabada en los archivos "redo" (Redo Logs Files). A partir de ese momento los archivos de "redo" pueden ser utilizados para la recuperación en caso de falla.

Por medio de un algoritmo el DBWR (Data Base Writer) baja a disco los bloques datos en memoria que tienen cierto tiempo de no ser utilizados (aproximadamente 3 segundos), el algoritmo es conocido con el nombre de LRU (Least Recently Used menos recientemente usado).

Con el algoritmo LRU, se libera espacio en memoria para los datos que requieren ser utilizados.

Generalmente los datos más recientemente usados, son también los más frecuentemente utilizados, el LRU asegura que los datos que más necesitamos están en memoria.

Redo Log Buffers.

Los "Redo Log Buffers", contienen información de los cambios realizados en los "Data Segments Blocks", cada vez que a una transacción se le aplica "commit", la información es inmediatamente escrita en el archivo de "redo" activo en ese momento.

Program Global Area (PGA).

La PGA contiene datos y controla la información para un simple proceso de usuario. Para cada usuario, existe una PGA que le es asignada cuando se conecta al RDBMS ORACLE. La PGA contiene información a cerca de la conexión del usuario con el RDBMS de ORACLE y mantiene esta para que pueda haber comunicación entre ambos.

La PGA es memoria de escritura no compartida. En caso de no haber suficiente espacio en la PGA para que el usuario pueda conectarse, ocurre un error y no se realiza ésta.

La PGA contiene también lo que se conoce como "Context Areas", las cuales se explican a continuación.

Context Areas.

Cada instrucción se SQL (Structured Query Language) requiere de un área de contexto (Context Area) para retenerla.

ORACLE automáticamente asigna una área de contexto cada vez que una instrucción SQL se ejecuta. Cada proceso de usuario, contiene un conjunto de áreas de contexto asociadas al la PGA.

Las áreas de contexto contienen:

- 1.- El texto de la instrucción SQL.
- 2.- La traducción de la instrucción SQL.
- 3.- La información "parseada" (Que el objeto exista y privilegios correctos).

Un proceso es un mecanismo que ejecuta una serie de pasos (algunas ocasiones se utiliza el término "trabajo o tarea").

Un proceso ORACLE, tiene su propio espacio de memoria en el cual corre.

Los procesos son importantes porque checan y resuelven las actividades que ocurren.

El RDBMS de ORACLE, tiene 5 procesos de "background", cuyo propósito es el de incrementar el "performance".

Cada instancia de la base de datos ORACLE, debe tener los cuatro procesos de background obligatorios, DBWR (Data Base Writer), LGWR (Log Writer), SMON (System Monitor) y PMON (Process Monitor). El proceso ARCH (Archiver) es opcional; es decir, la base de datos puede trabajar con o sin este proceso.

Data Base Writer.

El propósito del DBWR, es el de escribir los bloques modificados de los "Data Base Buffers" en la base de datos, según el algoritmo LRU.

El DBWR también se activa cada vez que ocurre un "checkpoint".

Log Writer.

El proceso LGWR escribe las entradas de los "Redo Log Buffers" a el "Redo Log File" activo cuando se ejecuta la instrucción "commit".

Este proceso, asegura el flujo permanente de datos del "buffer" al disco, sin demorar la información que se está generando nuevamente en los "buffers".

Si el LGWR no puede escribir en un "Redo Log File" para con un mensaje de error.

Existe comunicación entre el DBWR y el LGWR cuando ocurre un "checkpoint", LGWR formula una lista de los bloques que tienen que ser pasados a disco y el DBWR los pasa, cuando se completa el "checkpoint" el DBWR manda una señal al LGWR para indicarle que se completo el "checkpoint".

Si un "checkpoint" no ha sido completado y ya es tiempo de hacer otro "checkpoint", el DBWR ignora el primer "checkpoint" y toma el segundo.

La única ocasión el que el LGWR espera al DBWR para completar un "checkpoint" es cuando el LGWR ha usado todos los "Redo Logs Files" y esta a punto de hacer un "lockswitch".

System Monitor.

El propósito del SMON es la recuperación de la instancia de la base de datos; en caso de falla, las transacciones con "commit" se graban en forma definitiva, y a las que no lo tienen les aplica "rollback". Este proceso también se encarga de limpiar los segmentos temporales que no están en uso.

El SMON se encarga se que la base de datos reconozca un "tablespace" que fue puesto fuera de línea y después vuelto a traer en línea.

El SMON se despierta cada cierto tiempo para verificar si existe algún evento en donde sea requerido.

Process Monitor.

El PMON recupera los procesos de usuario, se encarga de liberar los recursos que algún proceso estaba utilizando; por ejemplo si algún usuario aborta un proceso, el PMON libera los candados a las tablas y los bloques que el usuario estaba utilizando.

Al igual que el SMON el PMON se despierta a intervalos de tiempo.

Archiver.

El proceso ARCH copia los "Redo Logs Files" a cinta una vez que éstos se han llenado; como se mencionó antes, este proceso es opcional y se activa al momento de la creación de la base de datos.

Este proceso sólo es con fines de respaldo de información. Si la base de datos esta corriendo en modo Archiver, un "Redo Log File" no puede ser reutilizado hasta que sea completamente respaldado.

De acuerdo los conceptos de la sección 3.1 y 3.2 se implementó la base de datos en la fábrica de automóviles, de esto se habla en la sección siguiente.

3.3 IMPLANTACIÓN DE LA BASE DE DATOS ORACLE.

La base de datos que se presenta en ésta sección, es la que utilizó la fábrica de autos, y con la cual se arrancó el sistema en su etapa de producción.

La base de datos quedó dividida en tres "tablespaces", en cada uno de ellos se distribuyeron las tablas que conforman el sistema, de acuerdo con la cantidad de información manejada.

A continuación se presenta la distribución de las tablas en los tres "tablespaces" que se construyeron.

SAM1

SA_BARCO
 SA_CENTRO_COSTO
 SA_DETALLE_REQ
 SA_ERROR_PRESTOP
 SA_MAT_UBICADO
 SA_NAVES
 SA_OC_ESP
 SA_PLANTAPARTE
 SA_PROVEEDOR
 SA_UNIDAD_ENVIO
 SA_UNIDAD_MEDIDA

SAM2

SA_AREA_UENV
 SA_DETALLE
 SA_ERROR
 SA_PLANTA
 SA_PRESTOP
 SA_RMR
 SA_ST_U_ENVIO
 SA_TABLA_CONVERSION
 SA_UB_FIJA
 SA_VL

SAM3

SA_AREA_SAM
 SA_COMISIONADO
 SA_GPO_ALMACEN
 SA_FACTURA
 SA_FOLIO
 SA_PARTE
 SA_PTAPTEOCPROV
 SA_REQ
 SA_US_GPO_ALMACEN

Se puede observar cada uno de los tres "tablespaces", cada una de las tres tablas más grandes de todo el sistema, SAM1 con SA_PLANTAPARTE (que es la tabla donde están las identificaciones de los materiales), SAM2 con SA_DETALLE (tabla que se dividió con SA_FOLIO para reducir la contención en disco según se vio), SAM3 con SA_FOLIO.

Aunque fue adicionado un "tablespace" exclusivo para segmentos de "rollback", para las transacciones muy grandes, cada "tablespace" de SAM tiene sus propios segmentos de "rollback".

Con el "tablespace" para segmentos de "rollback", se evita que la base de datos se fragmente demasiado. Cada "tablespace" fue puesto en un disco diferente. Los archivos que forman cada uno de los "tablespaces" son:

TABLESPACE	ARCHIVO	TAMAÑO EN MEGABYTES
SAM1	dbssam11.dbf	136
	dbssam12.dbf	21
	dbssam13.dbf	32
SAM2	dbssam21.dbf	157
SAM3	dbssam31.dbf	52
	dbssam32.dbf	52
	dbssam33.dbf	126

Los segmentos de "rollback" para los "tablespaces" son:

SEGMENTO DE ROLLBACK	TAMAÑO EN MEGABYTES	TABLESPACE
ROLLSAM1	21	SAM1
ROLLSAM2	21	SAM2
ROLLSAM3	21	SAM3
ROLLSAM4	52	SAM4

El "tablespace" SAM4 fue creado para asignar exclusivamente, transacciones de gran tamaño; por ejemplo, cuando se tiene que hacer un cambio a un gran número de registros, o para borrarlos, por ésta razón el tamaño de ese segmento de "rollback" es de 52 Megabytes.

También se tienen dos "Redo logs files" los cuales son:.

NOMBRE DE ARCHIVO	TAMAÑO EN MEGABYTES
LOG1_SAM	3
LOG2_SAM	3

El tamaño del SGA (System Global Area), es de 500 Kbytes. En el apéndice A se muestran las instrucciones para la creación de la base de datos, así como para la creación de tablas, índices y secuencias, creación de los "tablespaces" y segmentos de "rollback".

Con esto finaliza este capítulo, en donde se habló del funcionamiento de la base de datos ORACLE y la implementación de la base de datos para la fábrica de automóviles. En el siguiente capítulo se mostrará la aplicación para los usuarios finales.

CAPITULO 4

APLICACION

En este capítulo, se hablará de la aplicación del sistema y se presentan algunas pantallas representativas del mismo, tal y como fueron diseñadas para el usuario final.

Para el diseño y programación de las pantallas de usuario final, se utilizó una herramienta de ORACLE que se conoce SQL*Forms.

SQL*Forms, es una herramienta de propósito general para desarrollar y ejecutar formas para aplicaciones interactivas.

Una forma, es un formato de captura para los usuarios, en la cual están representados los campos y registros a los que el usuario tiene acceso. Las formas pueden estar divididas en lo que SQL*Forms llama bloques, estos bloques se forman por uno o varios campos y por uno o mas registros.

A los bloques que tienen mas de un registro se les llama bloques multiregistro, en otro caso son llamados uniregistro.

Toda forma debe contener al menos un bloque y cada bloque debe estar asociado a lo mas a una tabla (tabla de la base de datos).

Existen dos tipos de formas: de captura y de consulta, en las primeras se pueden hacer actualizaciones a la base de datos; en las segundas, como su nombre lo indica son sólo de consulta y en tales formas no es posible hacer alguna operación de actualización (altas, bajas y cambios).

A continuación se presentan cuatro formas del sistema, en las cuales se tocan los aspectos más importantes del sistema. Como es obvio, no es posible ilustrar todas las formas que componen el sistema.

Las formas que se presentan son:

- 1.- Descarga de contenedores.
- 2.- Entrada de material al almacén procedente de recibo.
- 3.- Atención a requisiciones de material.
- 4.- Consultas de RMR o PRESTOP.

1.- Descarga de contenedores.

La forma de *Descarga de contenedores*, es la que genera la solicitud de descarga de las unidades de envío, es decir, actualiza del estatus 2A a 2B (llegada a la fábrica 'vigilancia' a solicitud de descarga).

El objetivo es actualizar los estatus de la tabla SA_VL de 2A a 2B e insertar un registro en la tabla SA_ST_U_ENVIO incluyendo la fecha y el responsable de la descarga. En la fig. 4.1 se muestra la forma de captura.

SA116A	SISTEMA DE ADMINISTRACION DE MATERIALES	Pag. 1 de 1
USUARIO	TRANSITO Y ALMACENES	DD-MM-YY
DESCARGA DE CONTENEDORES		
(ST=2B)		
IMPRESION DE ETIQUETAS-CAJA (ENGOMADOS) IMPRESORA NORMAL		
IMPRESION DE ORDENES DE DESCARGA (ST=2B) IMPRESORA BARRAS		
CLAVE	CONTENEDOR	RESPONSABLE
<input type="text"/>	<input type="text"/>	<input type="text"/>
Count: *0	<List><Replac>	

Fig. 4.1 Forma de captura para descarga de contenedores.

Como se puede observar en la fig. 4.1. se tiene un bloque multiregistro, cuyos campos son CLAVE, CONTENEDOR y RESPONSABLE. El campo CLAVE, puede ser nulo, no se refiere a la clave del contenedor, sólo se utiliza como un parámetro para ejecutar un reporte de la base de datos, el reporte solo se ejecuta si el valor es "R".

Una vez que el contenedor ha sido capturado, se valida la información en la tabla SA_VL, la unidad de envío capturada (contenedor) debe ser igual a la que existe en SA_VL (campo unidad_envio) y el valor de estatus (campo status_ult) debe ser menor que tres. Si existe mas de un valor con éstas características habrá un error.

Para cada contenedor que es capturado, se accede a la tabla SA_VL y se recupera el vl (campo vl de la tabla); para éste caso, cada contenedor debe recuperar sólo un vl .

En la tabla SA_VL se guarda el responsable de descarga (campo resp_descarga) y se actualiza el estatus (campo status_ult) a 2B, después se inserta un registro en la tabla SA_ST_U_ENVIO, con el valor del vl, el estatus, el usuario y la fecha.

El campo IMPRESORA_NORMAL, puede ser nulo, pero si se le da una dirección de impresora, imprime etiquetas engomadas para las cajas del contenedor. EL campo IMPRESORA_BARRAS manda por impresora la solicitud de descarga para el contenedor. Ninguno de estos campos pertenece a tabla alguna, sólo son campos de trabajo en la forma.

2.- Entrada de material al almacén procedente de recibo.

El objetivo de ésta forma "Entrada de material al almacén procedente de recibo", es la de registrar entradas al almacén procedentes de recibo de materiales. La fig. 4.2 muestra la forma.

SA251A SISTEMA DE ADMINISTRACION DE MATERIALES Pag. 1 de 1
USUARIO TRANSITO Y ALMACENES DD-MM-YY

ENTRADA DE MATERIAL AL ALMACEN PROCEDENTE DE RECIBO

FOLIO	OBSERVACION
[REDACTED]	[REDACTED]

Count: *0 <List><Replace>

Fig. 4.2 Forma de captura para descarga de contenedores.

En la pantalla del usuario, se muestran los campos de FOLIO y OBSERVACION, los cuales pertenecen a la tabla SA_FOLIO. El bloque es multiregistro, se verifica que cada folio capturado cumpla con las condiciones siguientes:

a) El folio debe existir en la tabla SA_FOLIO, en caso contrario se despliega un mensaje en el campo OBSERVACION que dice: 'Folio no existe'.

b) El folio debe tener estatus 5 (campo st_detalle), es decir, enviado al almacén. Si el estatus es nulo, entonces en el campo OBSERVACION aparece: 'Folio no ha sido enviado al almacén'.

c) Si el folio tiene estatus igual a 6, entonces en el campo OBSERVACION aparece: 'Folio en almacén'.

d) Si ninguno de los casos de los incisos anteriores se presentan, entonces se actualiza el estatus (campo st_detalle) a 6, se actualiza la fecha y hora del sistema (campo fecha_st_detalle) y el campo OBSERVACION = 'Folio recibido en almacén'.

3.- Atención a requisiciones de material.

La forma "Atención a requisiciones de material" se implementó para atender las requisiciones de material; primero se valida que toda la información proporcionada por el usuario sea correcta, después se busca el folio más antiguo libre para surtir el material (con el fin de dar salida al material más viejo). La fig. 4.3 muestra la forma de requisiciones.

SA263A	SISTEMA DE ADMINISTRACION DE MATERIALES	Pag. 1 de 1
USUARIO	TRANSITO Y ALMACENES	DD-MM-YY

ATENCIÓN A REQUISICIONES DE MATERIAL

No. REQ.	PLA	NUMERO PARTE	C.C.	TACTO	SE	C U	CANT. SOL	GPO ALM	OBSERV

Count: *0 <List><Replace>

Fig. 4.3 Forma para atención a requisiciones.

Con excepción de los dos últimos campos de la forma de la fig. 4.3 (grupo_almacen y observaciones) que pertenecen a la tabla SA_GPO_ALMACEN todos los demás son campos de la tabla SA_REQ, respectivamente son:

REQ_NUMERO, PLANTA, NUMERO_PARTE CENTRO_COSTO,
TACTO, SEGUIDOR, CLAVE_URGENCIA y CANTIDAD_SOLICITADA.

La requisición se puede surtir sí y sólo sí se cumplen las siguientes condiciones:

- a) Existe un folio disponible y tiene estado de calidad 00 (campo status_calidad de la tabla sa_detalle).
- b) El folio no ha salido como "prestop", es decir, libre de errores.
- c) Está entre los estatus [2B, 6].

Si no se encuentran folios disponibles, la requisición será puesta en "backorder", y se incrementa la cantidad faltante (campo cant_faltante de la tabla sa_req).

Primero deberán atenderse todas las requisiciones con nivel de urgencia 2, después las de nivel de urgencia 1 y al final las de nivel normal.

En la forma se valida que no existan dos requisiciones con el mismo número, que la combinación planta/numero_parte exista, que el centro de costos exista, etc.

No debe haber dos números diferentes de requisición con cantidad faltante mayor que cero, y que tengan el mismo número de parte, planta, centro de costos y tacto; esto indica que se hicieron dos requisiciones con el mismo material para el mismo destino.

4.- Consulta de RMR o PRESTOP.

En la forma "Consulta de RMR o PRESTOP" se consulta toda la información asociada a un RMR o a un PRESTOP. En la forma se introduce el número de RMR o PRESTOP e inmediatamente se despliega la información relacionada con el número tecleado, tal como el número de documento(número secuencial), la fecha de factura, etc.

Los datos se obtienen de las tablas SA_FACTURA, SA_PLANTAPARTE, SA_PTAPTEOCPROV, SA_RMR o SA_PRESTOP, SA_FOLIO y SA_DETALLE.

En la forma no es posible hacer alteraciones a la base de datos, por ser sólo consulta.

La fig. 4.4 muestra la forma de consulta para los RMR's o los PRESTOP's.

Para finalizar éste capítulo, mostraremos las herramientas en las que se trabajó para la realización del sistema, así como los tiempos en que realmente se concluyeron cada una de las etapas.

SA121A	SISTEMA DE ADMINISTRACION DE MATERIALES		Pag. 1 de 1	
USUARIO	TRANSITO Y ALMACENES		DD-MM-YY	
CONSULTA DE RMR / PRESTOP				
NUMERO DE RMR/PRESTOP		[REDACTED]	R / P	[REDACTED]
No. DOCUMENTO	[REDACTED]	FECHA FACTURA	FECHA ENTRADA	PLANTA
FACTURA	[REDACTED]	PROVEEDOR	OC	[REDACTED]
NUMERO_PARTE	[REDACTED]	U.M.	[REDACTED]	
FOLIO	CAJA	CANT. FACT.	CANT. RECIB.	CANTIDAD TOTAL
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	FACTURADA [REDACTED]
				RECIBIDA [REDACTED]

Fig. 4.4 Forma de consulta para RMR o PRESTOP.

Las herramientas CASE utilizadas fueron:

HERRAMIENTA	PORCENTAJE DE CODIGO GENERADO
CASE*DESIGNER	100%
CASE*DICTIONARY	90%
CASE*GENERATOR	40%

Las herramientas para desarrollo fueron:

HERRAMIENTA	DESCRIPCION
SQL*FORMS SQL*REPORTWRITER PRO*C	Para el diseño de formas (pantallas). Para el diseño de reportes. Codigo combinado entre PL/SQL y lenguaje C.

Se realizaron un total de:

- a) Formas en SQL*Forms 102.
- b) Reportes en SQL*ReportWriter 81 y
- c) Programas en Pro*C 36.

Para el control de 130,000 artículos divididos en:

MATERIAL	TOTAL	PORCENTAJE
Tránsito (status 1A ... 2B)	30,800	23.7%
Recepción (status 3 ... 5)	3,000	2.3%
Almacén (status 6)	96,000	74.0%

El número de transacciones promedio para es sistema de la fábrica es de:

TRANSACCIONES	DESCRIPCION
3,000	En recepción.
4,800	Material localizado.
3,500	Material enviado a producción.

Los tiempos reales en semanas para la realización de cada una de las etapas así como el número de personas en cada etapa fue de:

ETAPAS	TIEMPO EN SEMANAS
Análisis 5 personas de la fábrica 1 persona de ORACLE	5
Diseño 6 personas de la fábrica 3 personas de ORACLE	7
Construcción 6 personas de la fábrica 8 personas de ORACLE	16
Liberación 10 personas de la fábrica 3 personas de ORACLE	3
Afinación 11 personas de la fábrica 4 personas de ORACLE	6

Con un total de 300 usuarios y 10 diferentes almacenes con aproximadamente 120,000 mts cuadrados en total.

CONCLUSIONES

Si bien en el modelo relacional y el diagrama entidad relación, existen reglas que debemos seguir para la representación óptima de la información, algunas veces no es posible seguir al pie de la letra éstas reglas, tal es el caso de la fábrica de autos, en donde fue necesario que una misma tabla se dividiera en dos, ya que el objetivo más importante, fue el de el tiempo de respuesta.

Las reglas de normalización y el diseño del diagrama entidad relación no consideran el tiempo de respuesta, o los problemas que existen en los accesos a disco, estos aspectos son de gran importancia en la vida real y son muchas veces la diferencia entre un buen sistema y uno malo.

A medida que se hacen sistemas más complejos, las herramientas para diseño y construcción de sistemas son más necesarias y se requiere que éstas sean más potentes para aminorar el trabajo de los diseñadores y desarrolladores, para estos últimos se busca que su función principal sea la de la afinación de programas preconstruidos por las herramientas CASE.

El modelo relacional, por su gran facilidad con respecto al manejo de la información por medio de tablas, hace que todas las personas puedan entenderlo sin necesidad de que conozcan el álgebra relacional.

Finalmente el diagrama entidad relación sirve de comunicación entre los usuarios y los consultores de sistemas convirtiéndose en un lenguaje común, ayudando de ésta forma a corregir errores de diseño y aportando ideas de gran utilidad para ambas partes, lo cual redundará en un sistema de mayor calidad.

BIBLIOGRAFÍA

1. Senn, J. A., *Análisis y diseño de sistemas de información*, México: Mc Graw Hill, 1987.
2. Kendall, K. E. y Kendall, J. E., *Análisis y diseño de sistemas*, México: Prentice Hall, 1992.
3. Tschritzis, D. C. y Lochovsky, F. H., *Database Management systems*, New York: Academic Press, 1977.
- 4.-S.M. Deen, *Fundamentos de los sistemas de bases de datos*, Barcelona: Gustavo Gili S.A. 1987.
4. ORACLE, *CASE training*. Material para el curso de CASE, USA: Oracle, 1991.
5. ORACLE, *PL/SQL user's guide and reference V1.0*, USA: Oracle, 1991.
6. ORACLE, *Oracle database administrator's guide V6.0*, USA: Oracle, 1990.
7. ORACLE, *Oracle RDBMS V6 architecture*, USA: Oracle 1991.
8. ORACLE, *SQL*Forms designer's reference V3.0*, USA: Oracle, 1991.
- bases de datos Oracle, USA: Oracle, 1990.
9. ORACLE, *SQL*Plus user's guide V3.0*, USA: Oracle, 1990.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

APÉNDICE A

Creación de objetos de la base de datos ORACLE.

a) Creación de la base de datos

```
CREATE DATABASE nombre_database
[CONTROLFILE REUSE]
[LOGFILE log_nombre1 [, log_nombre2] ... ]
[MAXLOGFILES entero]
[DATAFILE data_nombre1 [, data_nombre2] ... ]
[MAXDATAFILES entero]
[MAXINSTANCES entero]
[ARCHIVELOG | NOARCHIVELOG]
[ SHARE | EXCLUSIVE]
```

Para los archivos en LOGFILE y DATAFILE se puede especificar el tamaño del archivo si éste es de reuso. La sintaxis es:

```
nombre_archivo [SIZE entero [K|M] ] [REUSE]
```

En donde SIZE esta en bytes, pero se puede especificar K (1024 bytes) o M (1024 K), es decir, *entero*, se multiplica por 1024 si es K, si es M es multiplicado por 1048576. Si no se especifica ninguno de los anteriores, serán número de bytes.

Los parametros se explican a continuación:

CONTROLFILE RESUSE: Crea al archivo de control.

LOGFILE: Especifica que lo que viene a continuación son los nombres de los archivos "redologs".

MAXLOGFILES: Es el máximo número de archivos "redologs" permitido.

DATAFILE: Especifica que lo que viene a continuación son los nombres de los archivos de datos.

MAXDATAFILES: Es el número máximo de archivos de datos permitidos.

MAXINSTANCES: Es el máximo número de instancias ORACLE, para las cuales se permite levantar la base de datos.

ARCHIVELOG|NOARCHIVELOG: Para activar o desactivar el modo archive.

SHARE|EXCLUSIVE: Si sólo una instancia puede levantar la base de datos, entonces se activa el modo exclusivo.

b) Creación del tablespace

```
CREATE TABLESPACE nombre_tablespace
  DATAFILE [nombre1 [, nombre2] ... ]
  [DEFAULT STORAGE storage]
  [ONLINE | OFFLINE]
```

DATAFILE: Nombre de los archivos que formaran el "tablespace".

ONLINE|OFFLINE: Para indicar que el "tablespace" es traído en línea o fuera de línea.

En donde DEFAULT STORAGE indica que se especificarán los parámetros de *storage* (parámetros de almacenamiento), estos son utilizados también para la creación de las tablas, índices y segmentos de rollback, los parámetros de almacenamiento se explican a continuación.

```
STORAGE (
  [INITIAL entero] [NEXT entero]
  [MINEXTENTS entero] [MAXEXTENTS entero]
  [PCTINCREASE entero] )
```

INITIAL: Tamaño en bytes del primer extent alojado cuando el objeto es creado, el default es 5 extents.

NEXT: Tamaño en bytes del siguiente extent que será alojado cuando el objeto crezca, por default son 5 extents.

MINEXTENTS: Mínimo número de extents que toma el objeto cuando éste es creado.

MAXEXTENTS: Máximo número de extents que abarcará el objeto.

PCTINCREASE: Porcentaje de crecimiento del siguiente extent despues del next.

Por ejemplo si se tiene un NEXT de 20K y un PCTINCREASE de 50 el tercer extent es de $20+(20)(0.5) = 30K$, el cuarto es de $30+(30)(0.5) = 45K$ etc.

Al entero se le puede especificar K o M, como en inciso anterior para los archivos.

Por ejemplo INITIAL 10K, NEXT 1M.

c) Creación de tablas

```
CREATE TABLE nombre
  (campo1 tipo [, campo2 tipo] ... )
  PCTFREE entero
```


PCTUSED *entero*
INTRANS *entero*
MAXTRANS *entero*
STORAGE (*Parametros de storage*)
TABLESPACE *nombre_tablespace*

PCTFREE: Es el porcentaje de espacio en la tabla que debe quedar libre; esto con la finalidad de que registros de la misma tengan menos encadenamiento si es que llegan a crecer.

PCTUSED: Es el porcentaje de espacio en la tabla que debe ser usado, esto permite hacer inserciones a la tabla cuando se cae por debajo de éste limite, es decir bajo, ese limite el espacio de la tabla debe utilizarse de nuevo.

La suma de PCTFREE y PCTUSED debe ser menor o igual a 100.

INTRANS: Especifica el número de transacciones iniciales para la tabla.

MAXTRANS: Especifica el máximo número de transacciones permitidas para la tabla.

STORAGE: Parámetros de almacenamiento (inciso b).

TABLESPACE: Permite guardar una tabla en un "tablespace" determinado.

d) Creación de segmentos de rollback

```
CREATE [PUBLIC] ROLLBACK SEGMENT nombre  
[TABLESPACE nombre_tablespace]  
[ STORAGE parametros]
```

PUBLIC: Permite que otras instancias puedan acceder esos segmentos de rollback.

TABLESPACE: Permite que un segmento de rollback sea asignado a un "tablespace" específico.

STORAGE: Permite especificar los parametros de almacenamiento (inciso b)

e) Creación de índices

```
CREATE [UNIQUE] INDEX nombre_indice ON nombre_tabla  
STORAGE(parametros de storage)  
TABLESPACE nombre_tablespace  
UNIQUE
```

STORAGE: Parámetros de almacenamiento (inciso b).

TABLESPACE: Permite crear el indice en un "tablespace" determinado.

UNIQUE: Asegura que el indice no se repetirá. Este es usado para las llaves primarias.

f) Creación de secuencias

```
CREATE SEQUENCE nombre_secuencia  
  [INCREMENTET BY entero]  
  [START WITH entero]  
  [MAXVALUE entero | NOMAXVALUE]  
  [MINVALUE entero | NOMINVALUE]  
  [CYCLE | NOCYCLE] [CACHE entero | NOCACHE] [ORDER | NOORDER]
```

INCREMENTED BY: Especifica el incremento de los números de la secuencia.

START WITH: Número en que inicia la secuencia.

MAXVALUE|NOMAXVALUE: Permite que la secuencia tenga un valor máximo si así se desea.

CYCLE|NOCYCLE: Permite que la secuencia empiece de nuevo si así se desea.

CACHE|NOCACHE: Guarda cierta cantidad de números en memoria "cache", si así se desea.

ORDER|NOORDER: Permite que la secuencia sea ordenada o desordenada.