



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"CAMPUS ARAGÓN"

"INTRODUCCIÓN AL LENGUAJE SQL
Y A LAS HERRAMIENTAS ORACLE
CON ALGUNOS EJEMPLOS."

T E S I S
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
FELIPE DE JESUS LÓPEZ ESTALA

ASESOR: ING. SILVA VEGA MUYTOY
SILVIA

SAN JUAN DE ARAGÓN, EDO. DE MÉXICO 1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

TESIS

COMPLETA

TABLA DE CONTENIDO

INTRODUCCIÓN	i
CAPITULO I BASES DE DATOS	1
1.1 Definición	1
1.2 Enfoque Jerárquico	2
1.3 Enfoque de Red	3
1.4 Enfoque Relacional	5
CAPITULO II REDES LAN	15
2.1 Definición	15
2.2 Topología y Topografía	16
2.3 Topología de Estrella	18
2.4 Topología de Bus	18
2.5 Topología de Anillo	19
CAPITULO III REQUERIMIENTOS DE SOFTWARE Y HARDWARE	20
CAPITULO IV COMANDOS Y CONCEPTOS DE SQL/SQL PLUS	22
4.1 Iniciando una sesión	22
4.2 SQL Comandos y Conceptos	23
4.2.1 DESC	24
4.2.2 SELECT	25
4.2.3 ALL	29
4.2.4 DISTINCT	29
4.2.5 FROM	30
4.2.6 WHERE	31
4.2.7 NULL	32
4.2.8 TO_CHAR	34
4.2.9 TO_DATE	34
4.2.10 COMMIT	35
4.2.11 ROLLBACK	35
4.2.12 Querys Multinivel	36
4.2.13 CREATE TABLE	36
4.2.14 CREATE VIEW	36
4.2.15 INSERT INTO	36
4.2.16 ADD	36
4.2.17 ALTER TABLE	39
4.2.18 UPDATE	39
4.2.19 DELETE	40

4.2.20 DROP.....	40
4.3 SQL PLUS.....	42
4.3.1 BUFER	42
4.3.2 Archivos comando	43
4.3.3 Variables	43
4.3.4 Parámetros	44
4.3.5 START LIST	44
4.3.6 Formateo de Reportes a través de parámetros	46
CAPÍTULO V HERRAMIENTAS ORACLE	47
5.1 SQL*Forms	47
5.1.1 Operación interna.....	48
5.1.2 SQL*Forms como compilador.....	48
5.1.3 Pantalla Principal	50
5.1.4 Action	50
5.1.5 Form	51
5.1.6 Block.....	52
5.1.7 Field.....	55
5.1.8 Procedure	
5.1.9 Image	
5.1.10 Help	
5.2 SQLREPORTWRITER.....	87
5.2.1 Query's	89
5.2.2 Fied	94
5.2.3 Summary	99
5.2.4 Text.....	100
5.2.5 Group	102
5.2.6 Report.....	104
5.2.7 Parameter.....	105
5.3 SQL MENU	107
5.3.1 Action	110
5.3.2 Admin	111
5.3.3 Menu.....	112
5.3.4 Aplicacion.....	114
5.3.5 Menu.....	115
5.3.6 Item.....	116
5.3.7 Background	118

5.3.8 Parámetros de Sustitución	118
5.3.9 Macros.....	120
5.3.10 PL/SQL.....	123
5.3.11 Variables Globales.....	128
5.3.12 Procedure.....	128
5.3.13 Help.....	131
5.3.14 Options.....	132
Conclusiones.....	134

Dedico esta obra a todas aquellas personas que de una u otra manera han intervenido en mi vida, enseñándome con esto, que la única manera de poder realizar mis sueños es preparándome física, mental y espiritualmente, por ende doy gracias.

... A mis padres, pues desde que me dieron el ser, han sacrificado gustos y placeres personales con el único objetivo de forjarme un porvenir sin pedir nada a cambio.

... A mis hermanos, para que de igual manera se preparen y forjen su futuro.

... A tí fla-k, porque tu recuerdo me impulsa a seguir adelante, levantándome por cada tropiezo que encuentro en la vida.

... A mis mejores amigas Carmen y Adriana, por su invariable amistad que a través de los años me demostraron, con consejos y afecto, aún en los momentos más difíciles.

... A Silver, mi gran amigo, hermano de escuela y de corazón.

... A Cuy, Memo y Dulce por todo los lazos que nos unen.

... A todos mis maestros por los conocimientos otorgados.

INTRODUCCIÓN

En los inicios del procesamiento de datos, cada departamento creaba programas con archivos y formatos que eran únicos para una aplicación propia, esto provocó el almacenamiento y manipulación de una gran cantidad de información redundante; de ahí que surgiera la necesidad de tener una sola versión de los archivos, que pudieran ser relacionados de manera que a cualquier aplicación, se le permitiera tener acceso a cualquier parte de los archivos sin importar su localización, método de acceso o formato de los registros. A los grandes bancos de información se les denominó Bases de Datos(B.D.), el catálogo de una biblioteca, la lista de operaciones de cajeros automáticos de un banco, la lista de reservaciones de una compañía de transporte de pasajeros, un archivo policiaco, la nómina de una empresa, etc., todos estos son ejemplos de **bases de datos**; y ya que una Base de Datos debe minimizar la cantidad de información redundante, surgieron diversos enfoques para su diseño, de estos enfoques que se mencionan en el Capítulo 1, el enfoque Relacional se explica más a fondo, ya que ORACLE fué diseñado para manipular bases de datos relacionales.

Ahora bien, la creciente cantidad de información manejada en las empresas, provoca el desarrollo de tecnologías para la transmisión y procesamiento de la misma, las redes de computadoras son un ejemplo de este desarrollo y en especial, por su gran aceptación en el Capítulo 2 se habla brevemente de las redes de área local mejor conocidas como redes LAN (de sus siglas en inglés Local Area Network). Tomando en cuenta que

trabajaremos en una red de computadoras tipo LAN, y que desarrollaremos nuestras aplicaciones con ayuda de los productos ORACLE, consultando el Capitulo 3 conoceremos las características tanto de Software como de Hardware que deberá tener nuestro equipo de cómputo.

Por otro lado, las aplicaciones del mundo real demandan el acceso concurrente de los datos criticos; con la mayoría de los tipos de B.D., los sistemas pueden presentar un desempeño limitado, no precisamente por la capacidad del hardware instalado, sino por el número de usuarios que tienen que esperar el acceso a los datos. ORACLE utiliza un bloqueo a nivel registro irrestringido y consultas libres de contención, que minimizan y en algunos casos eliminan completamente tiempos de espera. La filosofía de ORACLE involucra diversos conceptos como son: *Compatibilidad*, que significa que la base de datos debe ser compatible con otras existentes en el mercado. *Portabilidad*, donde una aplicación puede ser transportada de una máquina a otra con diferentes sistemas operativos, y esto no provoca ningún cambio o modificación a dicha aplicación. *Conectividad* a través de una arquitectura que permite que los datos y las aplicaciones residan en diferentes computadoras, plataformas, sistemas operativos y ambientes de red. Aunado a estas características, se tiene la *Capacidad de manejo de grandes volúmenes de información que facilita el uso de grandes sistemas.*

SQL es un lenguaje de consulta estructurado, que por si mismo es capaz de realizar consultas a la información almacenada en cualquier base de datos relacional y generar los reportes necesarios, al igual que en cualquier otro lenguaje de programación, se necesita práctica para poder entender sus alcances y limitaciones, en el Capitulo 4 se mencionan las instrucciones de SQL así como la sintaxis general de estas últimas, se presentan ejemplos que indican al lector cómo empezar a programar y se proporciona la explicación de los campos que deben de estar presentes al momento de utilizar alguna instrucción.

La persona encargada de desarrollar los sistemas, necesita presentar al usuario final la información (menús, reportes o pantallas de captura, etc) en algún formato y, como SQL da un formato simple a los reportes y no facilita la creación de pantallas de captura con una presentación en específico, ORACLE

implemento las herramientas SQLForms, SQLMenu y SQLReportWriter. Estas herramientas proporcionan la ayuda necesaria para poder crear sistemas completos de información, conteniendo pantallas de captura, menus de navegación, reportes, etc. A pesar de que existen en el mercado productos que pueden explotar la información de bases de datos relacionales, estos carecen de la mayor parte de las habilidades de las herramientas ORACLE; la explicación de cada una de estas herramientas se proporciona en los capítulos subsecuentes.

El propósito de este trabajo no es el de enseñar SQL a fondo, sino dar una introducción al manejo de este lenguaje y explicar de forma breve cada una de las herramientas con las que cuenta, preparando al lector para que pueda empezar a desarrollar aplicaciones propias. Es necesario mencionar que SQL y sus herramientas trabajan bajo ambiente Unix, por lo que el lector deberá estar familiarizado con este ambiente ya que algunos de los procesos que ejecuta SQL hacen uso de comandos del sistema operativo.

Conforme practique el lector, se dara cuenta que el lenguaje SQL así como sus herramientas son de fácil entendimiento, cierto es que este trabajo no contiene todo el conocimiento necesario para dominar cada uno de los temas mencionados ,pero si proporciona lo necesario para desarrollar y aprender uno de los lenguajes que debido a su fácil manejo, ha llegado a ser definido como plataforma de desarrollo de grandes empresas gubernamentales como Petroleos Mexicanos.

OBJETIVO

- Explicar los conceptos fundamentales de las Bases de Datos y sus diferentes arquitecturas.
- Definir el entorno necesario que se debe de tener para poder trabajar con SQL y el ambiente que sus herramientas necesitan para operar.
- Proporcionar la sintáxis general del lenguaje SQL, comandos con los que cuenta y los conceptos necesarios para entender el funcionamiento de estos últimos.
- Dar los conocimientos necesarios para entender el funcionamiento de las diferentes herramientas Oracle.
- Dar las bases necesarias para empezar a desarrollar algunas aplicaciones en SQL con ayuda de sus herramientas.

CAP. I: BASES DE DATOS (D.B.)

Antes de la introducción de los DBMS (manejadores de bases de datos), cada unidad de sistemas (informática) contaba con un grupo de programadores que definía e implementaba aplicaciones. Estos programadores debían conocer a fondo la forma en que los diferentes programas de aplicación se relacionaban con el sistema operativo, los datos y el ambiente en que eran usados. Había múltiples copias de los mismos archivos y era difícil saber cual de las copias contenía las últimas modificaciones.

Durante estos primeros años del procesamiento de datos, cada programa de aplicación usaba archivos que eran únicos para esa aplicación, lo cual provocaba el almacenamiento y manipulación de una gran cantidad de datos redundantes; cada departamento dentro de la organización creaba sus propios archivos en el formato requerido para sus actividades únicas.

Los DBMS nacieron bajo el concepto clave de tener una sola versión de los datos, de tal forma que estos pueden ser relacionados de manera que a cualquier aplicación se le permita tener acceso a cualquier parte de los datos, sin importar su localización, método de acceso o formato.

DEFINICION

Algunas veces se define a las B.D. como "una colección integrada de registros en donde cada persona y cada programa autorizado podrá acceder los datos que requiere, dichos datos pueden ser modificados por aquellos autorizados a hacerlo". Nosotros definimos a las B.D. como **un conjunto de datos relacionados entre sí, almacenados, estructurados, no redundantes y de fácil acceso**, es más, podemos imaginar a una B.D. como un arreglo matricial del orden $m \times n$, esto es, que se cuenta con columnas y renglones tal como lo muestra la siguiente figura:

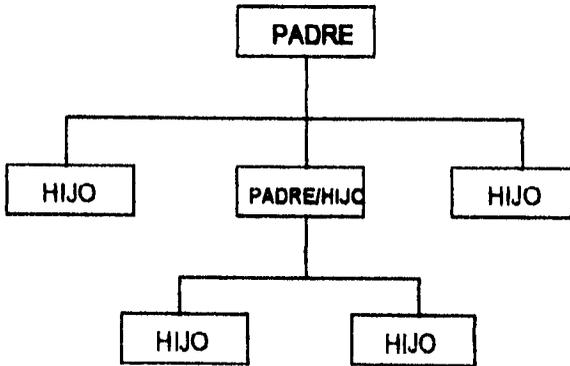
COLUMNA 1	...	COLUMNA N	REGLON 1
			REGLON M

Sin embargo, una B.D. no es más que una pila de información: El catálogo de una biblioteca, el registro de operaciones de cajeros automáticos de un banco, la lista de reservaciones de una compañía de transporte de pasajeros, un archivo policiaco, la nómina de una empresa, etc., todos estos sistemas son **bases de datos**.

Un aspecto importante es que, una B.D. bien diseñada, deberá minimizar la cantidad de información redundante. A continuación se mencionan algunos enfoques de las B.D. así como una breve explicación de cada uno de ellos.

ENFOQUE JERARQUICO

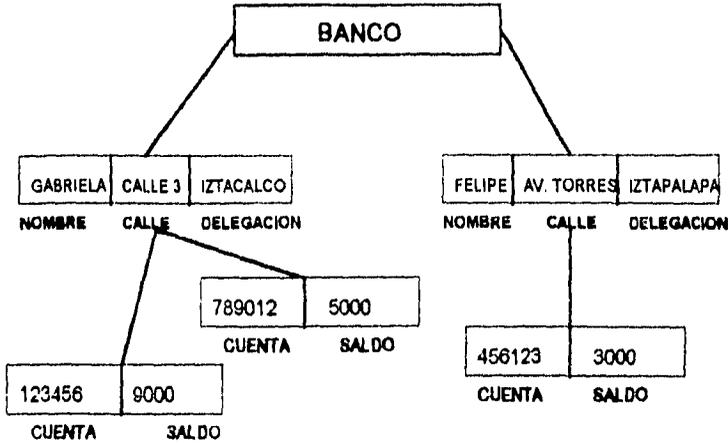
En este caso, los datos se presentan como estructuras de un árbol. Un árbol se compone de un nodo raíz y varios nodos sucesores, ordenados jerárquicamente. Cada nodo representa una entidad (tipo de registro) y las relaciones entre entidades son las conexiones entre los nodos.



El nodo colocado en la parte superior es llamado padre y los nodos inferiores son los hijos. En el sistema jerárquico, las conexiones entre archivos

no dependen de la información contenida en ellos, se definen al principio y son fijos, el procesamiento es del tipo TOP-DOWN.

La característica sobresaliente de este modelo es el manejo de la conexión uno a muchos(U a M), entre un padre y varios hijos ejemplo:

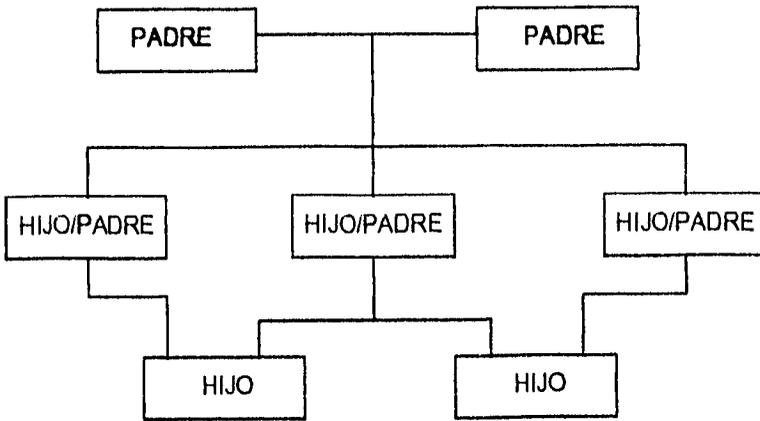


Por ejemplo, en la figura anterior, se tienen dos entidades: cuentahabiente y cuenta, relacionadas entre sí con una relación muchos a muchos(M a M), es decir, un cuentahabiente puede tener varias cuentas y, una cuenta puede pertenecer a varios cuentahabientes. El registro cuentahabiente consiste de 3 campos: nombre, calle y delegación. El registro cuenta tiene dos campos: número y saldo. Sin embargo, este enfoque tiene algunas desventajas como las siguientes:

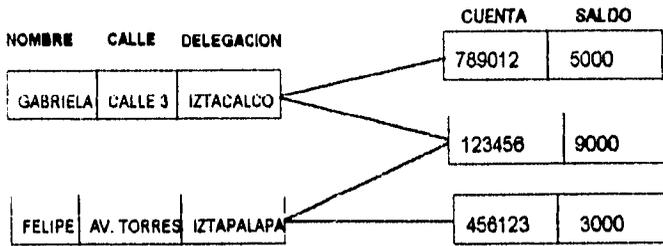
- No modela sencillamente las relaciones M a M.
- Anomalías de inserción.
- Anomalías de borrado.
- Anomalía de actualización.
- Se pueden dar consultas inconsistentes.

ENFOQUE DE RED

En este caso, los datos se representan como registros ligados formando un conjunto de datos intersectados. La base de datos de red, a diferencia de las jerárquicas, permite cualquier conexión entre entidades, es decir, se pueden representar relaciones de muchos a muchos. En una red, un hijo puede tener varios padres a la vez. Ejemplo:



Retomando el ejemplo del cuetahabiente, tenemos lo siguiente:



Sin embargo, existen desventajas en este tipo de enfoque:

- Resulta difícil definir nuevas relaciones.
- Es complicado darle mantenimiento ya que cualquier cambio en la estructura requiere una descarga en los datos.
- Representa desperdicio de recursos.
- Anomalías de inserción.
- Anomalías de borrado.

ENFOQUE RELACIONAL

La estructura lógica de una base de datos relacional esta basada en la representación de entidades mediante tablas, las cuales constan de columnas (campos) y renglones (registros). Las relaciones entre tablas se llevan a cabo a través de un conjunto de columnas que se tengan en común, logrando una conexión dinámica entre un número ilimitado de ellas a través del contenido de esas columnas.

La ventaja de los sistemas relacionales es el poder modificar la relación sin la preocupación de especificar las combinaciones entre registros. Retomando el ejemplo anterior tenemos:

TABLA CUENTA

CUENTA	SALDO
123456	9000
789012	5000
456123	3000

TABLA CUENTAHABIENTE

NOMBRE	CALLE	DELEGACION
GABRIELA	CALLE 3	IZTACALCO
FELIPE	AV. TORRES	IZTAPALAPA

TABLA CUENTA/CUENTAHABIENTE

CUENTA	NOMBRE
123456	GABRIELA
123456	FELIPE
789012	GABRIELA
456123	FELIPE

Para entender mejor este concepto, se explicara que son Tablas, Índices y Vistas.

TABLAS

Una B.D. Relacional, esta compuesta o dividida por tablas, cada tabla tendrá un nombre de 30 caracteres alfanuméricos máximo; una tabla puede ser mejor entendida si la imaginamos como un arreglo bidimensional con sus columnas y renglones tal como lo muestra la sig. figura.

COLUMNA 1		COLUMNA M	REGLON 1
			REGLON N

Por ejemplo, tendremos que usualmente cada renglon de una tabla contiene información relacionada con alguna llave, en función uno a uno, se observa la información perteneciente a un grupo de personas.

Tabla **EMPLEADO**

No_EMP	NOMBRE	No_DEPTO	PUESTO	No_PC	SUELDO	ETX
850	Gonzalez P. Ramón	4	Capturista	07	800.00	524
391	Corona S. Martín	6	Vigilante	03	700.00	254
500	De Valdéz Jorge	1	Jefe_Chof.	09	750.00	324
900	Zapata T. Francisco	4	Capturista	08	800.00	524
.
.
.

De esta forma, en las tablas relacionales, cada renglón contiene información relacionada con alguna llave en función uno a uno. En la tabla anterior llamada EMPLEADO se observa la información perteneciente a los empleados, la llave para esta tabla puede ser *nombre* o *puesto*.

Cada columna contiene información de algún tipo para llaves del mismo tipo, como se observó, las columnas de la tabla EMPLEADO contienen la información referente a NUMERO DE EMPLEADO, NOMBRE, DEPARTAMENTO, PUESTO, NUMERO DE PC, SUELDO y EXTENSIÓN TELEFÓNICA; a su vez, cada columna esta dividida en renglones dependiendo del número de registros con los que cuente la tabla, en este momento, a cada renglón se le conoce como "INSTANCE" (momento) tal como se observa:

...	ETX	
...	524	← 1er instance
...	254	← 2° instance
...	324	
...	524	
...	.	
...	.	
...	.	

Si se necesitan más datos, pueden crearse otras tablas, siempre y cuando esta información no pueda proporcionarse por otro medio. Por ejemplo, pueden solicitar la descripción del departamento para el cual trabaja algún empleado, en este caso se diseña una tabla que contenga tanto el *No de Depto.* como la *descripción* del mismo, las tablas quedarían de la siguiente forma:

TABLA EMPLEADO

No_EMP	NOMBRE	No_DEPTO	PUESTO	No_PC	SUELDO	ETX
.
.
.

No_DEPTO	NOMBRE_DEPTO
4	INFORMATICA
6	SEGURIDAD
1	ENTREGAS
.	.
.	.
.	.

TABLA DEPARTAMENTO

Los diferentes tipos de datos en una tabla son llamados columnas, cada columna tiene un nombre que puede tener de 1 a 30 caracteres alfanuméricos, además de esto, cada columna tiene un formato para especificar el tipo de dato que contendrá para ser utilizado, los datos pueden ser de diversos tipos ya sea FECHAS, CARACTERES ó NUMEROS

Pseudo - columnas.

Este tipo de columnas son creadas automáticamente en cada tabla, los valores son insertados y actualizados por el sistema y no deben ser cambiados; algunas de las pseudo-columnas más importantes son las siguientes:

NOMBRE	FUNCION
ROWID	Identificador único de renglón.
ROWNUM	Número seleccionado con cada renglón en un query que especifica el lugar del renglón en la secuencia de selección.
SYSDATE	Fecha y hora actuales.
UID	Número de identificación de usuario.
USER	Nombre del usuario actual.

Esto significa que en nuestras tablas están presentes estas pseudo - columnas y de poder verlas serían así: como las de la sig. página.

TABLA EMPLEADO

No_EMP	...	ROWID	ROWNUM	SYSDATE	UID	USER
.
.
.

No_DEPTO	NOMBRE_DEPTO	ROWID	ROWNUM	SYSDATE	UID	USER
4	INFORMATICA					
.
6	SEGURIDAD
.
1	ENTREGAS					
.
.
.

TABLA DEPARTAMENTO

INDICES

Al momento de introducir nuevos datos, estos son almacenados en los primeros espacios disponibles, ahora bien, ¿cómo podríamos acceder rápidamente al registro deseado?, esta duda se resuelve indexando cada registro, esta acción se lleva a cabo usando el valor de ciertas columnas o creando otras, las cuales constan de valores únicos, por ejemplo, de la tabla EMPLEADO tendríamos:

TABLA EMPLEADO

No_EMP	NOMBRE	No_DEPTO	PUESTO	No_PC	SUELDO	ETA
850	Gonzalez P. Ramón	4	Capturista	07	800.00	524
391	Corona S. Martín	6	Vigilante	03	700.00	254
500	De Valdéz Jorge	1	Jefe_Chof	09	750.00	324
900	Zapata T. Francisco	4	Capturista	08	800.00	524
.
.
.

No_EMP		No_PC
850	INDICE POR No_PC	07
391		03
500		09
900		08
.		.
.		.
.	INDICE POR No_EMP	.

La ventaja de los índices radica básicamente en que con ellos se logran recuperar los datos de una manera más rápida, sin embargo, en algunas ocasiones el actualizar registros resulta muy lento.

Existe un tipo especial de índices y SQL se asegura de que cada índice creado sea único, a estos índices se les conoce como *índices únicos*.

Para crear un índice se utiliza un comando, del cual se da a continuación la sintaxis general:

```
CREATE [UNIQUE] INDEX Nombre_Indice ON Nombre_Tabla (Nombre_Columna,...),
```

donde:

Nombre_Indice	Es el nombre del índice y contiene de 1 a 240 caracteres alfanuméricos.
Nombre_Tabla	Es la tabla que se este utilizando en ese momento.
Nombre_Columna	Es el nombre de la columna a la cual se le identificara con el índice.

El valor de un índice puede contener hasta 240 caracteres, si un índice es único entonces dos renglones no pueden tener los mismos valores para las columnas en el índice; aparte de esto, sólo es permitido tener un máximo de 16 columnas en un índice.

VISTAS

Una vista es una tabla virtual, una ventana para un grupo de tablas, es decir, es un método para seleccionar información relacionada entre sí sin tener que alterar alguna tabla y utilizarla para obtener los reportes necesarios. Por ejemplo, de las tablas usadas anteriormente tomamos la de EMPLEADO,

No_EMP	NOMBRE	No_DEPTO	PUESTO	No_PC	SUELDO	ETX
850	Gonzalez P. Ramón	4	Capturista	07	800.00	624
391	Corona S. Martín	6	Vigilante	03	700.00	254
500	De Valdéz Jorge	1	Jefe_Chof	09	750.00	324
900	Zapata T. Francisco	4	Capturista	08	800.00	624
.
.
.

de esta tabla podemos obtener algunas vistas que nos sean necesarias como:

LISTA_EMP

NOMBRE	No_DEPTO
Gonzalez P. Ramón	4
Corona S. Martín	6
De Valdéz Jorge	1
Zapata T. Francisco	4
.	.
.	.
.	.

LISTA_DEPTO_EMP

No_EMP	NOMBRE	No_DEPTO	SUELDO	ETX
850	Gonzalez P. Ramón	4	800.00	524
391	Corona S. Martín	6	700.00	254
500	De Valdéz Jorge	1	750.00	324
900	Zapata T. Francisco	4	800.00	524
.
.
.

PRIVILEGIOS

SQL provee seguridad al controlar sus sesiones mediante el uso de identificadores de usuario como nombres (loggin) y passwords, el diseñador de una tabla puede consultar, mantener, modificar y borrar los privilegios de la misma siempre y cuando este diseñador sea el administrador de la base de datos. Los privilegios se dividen en tres: Privilegios del sistema, Privilegios sobre tablas y Privilegios sobre vistas.

Privilegios del sistema.

Existen 3 niveles de privilegios del sistema:

- 1.- DBA Todos los privilegios.
- 2.- RESOURCE Privilegio de conexión y creación de tablas.
- 3.- CONNECT Privilegio de conexión, únicamente consultas.

Unicamente un Administrador de la Base de Datos tiene autoridad para dar de alta nuevos usuarios de la siguiente manera:

```
SQL> GRANT CONNECT TO USUARIO IDENTIFIED BY PASSWORD
```

pero un usuario si puede cambiar su propio password así:

```
SQL>GRANT CONNECT TO USUARIO IDENTIFIED BY NUEVO_PASSWORD
```

Privilegios sobre tablas.

Existen dos maneras de ganar acceso a las tablas:

- 1.- Crear una tabla.
- 2.- Obtener permiso sobre tablas creadas por otros usuarios. Este lo debe otorgar el dueño de la tabla, de la siguiente manera:

```
SQL> GRANT SELECT, INSERT ON TABLA TO OTRO_USUARIO
```

los siguientes privilegios pueden ser otorgados sobre las tablas:

SELECT, INSERT, UPDATE, DELETE, ALTER, INDEX.

para dar a un usuario todos los privilegios sobre una tabla se realiza lo siguiente:

```
SQL> GRANT ALL ON TABLA TO OTRO_USUARIO
```

Sinónimos

Si otro usuario le ha dado privilegio de SELECT sobre alguna de sus tablas, usted la tiene que acceder de la siguiente forma:

```
SQL> SELECT * FROM USUARIO_DUEÑO_DE_LA_TABLA.TABLA;
```

como una alternativa, se pueden crear sinónimos utilizando un nombre más significativo y simple. A continuación se muestra la forma de crear un sinónimo llamado EMP sobre la tabla EMPLEADO del usuario Carmen:

```
SQL>CREATE SYNONYM EMP FOR CARMEN.EMPLEADO;
```

de esta forma se puede consultar la tabla sin gran problema.

Cuando se accesa la tabla de otro usuario, es posible que se reciba el mensaje: "Table or View does not exist", esto significa que:

- a) No se tiene acceso a la tabla o vista
- b) La tabla o vista realmente no existe.

también puede dar a otros usuarios el derecho de otorgar privilegios sobre alguna tabla:

```
SQL> GRANT ALL ON TABLA TO OTRO_USUARIO WITH GRANT OPTION;
```

Eliminando privilegios.

Los privilegios del sistema pueden ser eliminados únicamente por el administrador de la base de datos, los privilegios sobre tablas pueden eliminarse en cualquier momento, los privilegios de tabla se eliminan de la siguiente manera:

```
SQL> REVOKE PRIVILEGIO ON TABLA FROM USUARIO;
```

CAP. II: REDES LAN

Anteriormente, cuando se hablaba de la computadora, se pensaba en un aparato electrónico al que sólo tenían acceso las personas de extrema inteligencia.

Ante la creciente cantidad de información manejada en las empresas, se ha hecho necesario el desarrollo de una tecnología para la transmisión, procesamiento, almacenamiento y análisis de la misma, uno de los frutos de este desarrollo son las redes de computadoras, de las que existen diversos tipos como son Redes Lan, Redes Man, Redes Wan, etc. Sin embargo, debido a su amplia aceptación, hablaremos de las redes de computadoras de área local mejor conocidas como redes LAN (Local Area Network).

Existen diferentes descripciones para el término *Redes de Computadoras*, siendo una de las más aceptadas la siguiente:

"Un grupo de computadoras interconectadas a través de uno o varios caminos o medios de transmisión con una finalidad concreta: Transferir e intercambiar datos entre equipos y personas."

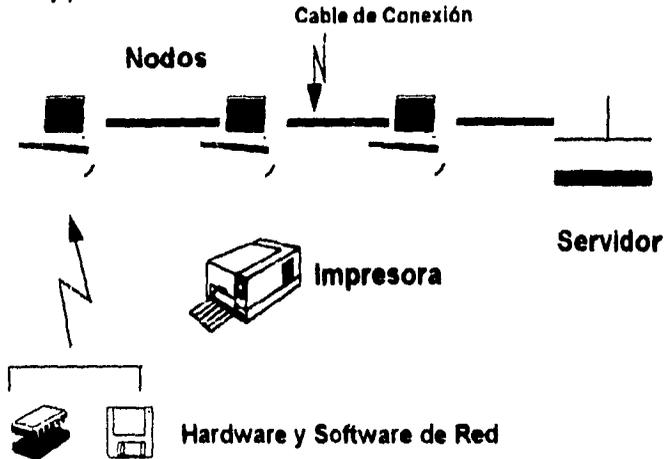
Estos sistemas necesitan dos partes primordiales para su funcionamiento, una de ellas es el *Hardware* (Elementos físicos tales como : Cableado, tarjetas, monitores, etc.) y la otra es el *Software* (Programas, Paquetería, etc.)

Por lo anterior, para formar una Red de Computadoras, se necesitan computadoras con Hardware y Software adecuado para Red. Esto significa que será necesario instalar una tarjeta (Circuito electrónico) a las máquinas que se desee permanezcan en el sistema; así como un programa, que permitirá comunicarse a los equipos. Por el lado de Software existen programas comerciales los cuales permite administrar los recursos de la red. Ejemplo de ellos son: NOVELL, SYNTAX, LAN MANAGER, etc.

Ya que estos equipos son utilizados por usuarios que desarrollan aplicaciones, es necesario que un especialista resuelva los problemas que a

menudo se presentan. A esta persona o grupos de personas se les denomina *Administradores de Red*.

En la siguiente figura se muestran los componentes propios para instalación y puesta en marcha de una Red.



Red de Computadoras y sus Componentes

Los componentes básicos para una red tipo LAN son los siguientes:

- El emisor, en el cual se genera y del que parte la información.
- El codificador, que convierte los datos que se envían en un mensaje; es decir, transforma la información para que se pueda enviar.
- EL medio de transmisión, que proporciona la vía a través de la cual se va a enviar el mensaje.
- El decodificador, que convierte los datos recibidos, dejándolos de forma que el receptor pueda entenderlos.
- El receptor, que es el destinatario de la información enviada y, en definitiva, el que va a utilizarla.

El trabajo fundamental de una red de área local es el de enlazar físicamente varias PC's a una minicomputadora o una mainframe y permitirles compartir recursos como discos duros, impresoras, unidades de disco compacto, etc. Para esto se utilizan gran variedad de materiales como fibra óptica, líneas telefónicas, luz infrarroja, señales de radio, etc., a su vez, también existen diversas maneras o configuraciones de conectar las computadoras una con otra, cada configuración o topología tiene distinta manera de enviar o recibir los datos de las computadoras,

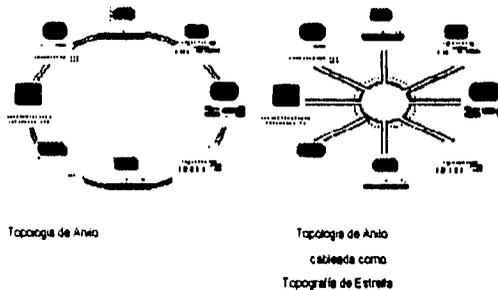
estos mensajes pueden solicitar consultas a los datos, ejecutar programas almacenados en la red y otras más.

Los datos solicitados por la PC pueden ser almacenados en alguna otra computadora perteneciente a la red usada para este propósito, o bien almacenarse en el servidor de la red, este servidor es un computadora especializada (tanto en software como en hardware) que administra y controla los recursos propios y de la red permitiendoles sean compartidos por las computadoras pertenecientes a la misma. Las topologías más usadas son tres: Bus, Token-Ring y Estrella, estas se explican a continuación brevemente con el objeto de dar al lector una idea más clara de como funcionan las redes de computadoras existentes en el mercado.

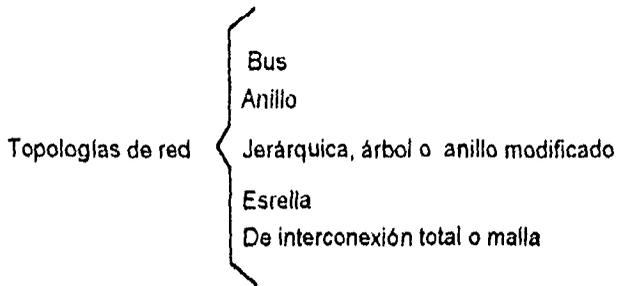
Topologías y Topografías

La topología de una red es la forma lógica de conectar las estaciones de trabajo, utilizada por la persona que diseña la red. Esta topología esta definida por la forma (geométrica) en que se extienden los cables que conectan las distintas estaciones. A la hora de planificar la disposición de los cables la topografía es más importante que la topología. Las estaciones de trabajo se comunican a la red utilizando un método de acceso específico que depende del tipo de red de que se trate. Los métodos de acceso son técnicas utilizadas por las estaciones de trabajo para compartir el canal de comunicación.

La siguiente figura muestra la diferencia entre topología y topografía anteriormente explicadas.



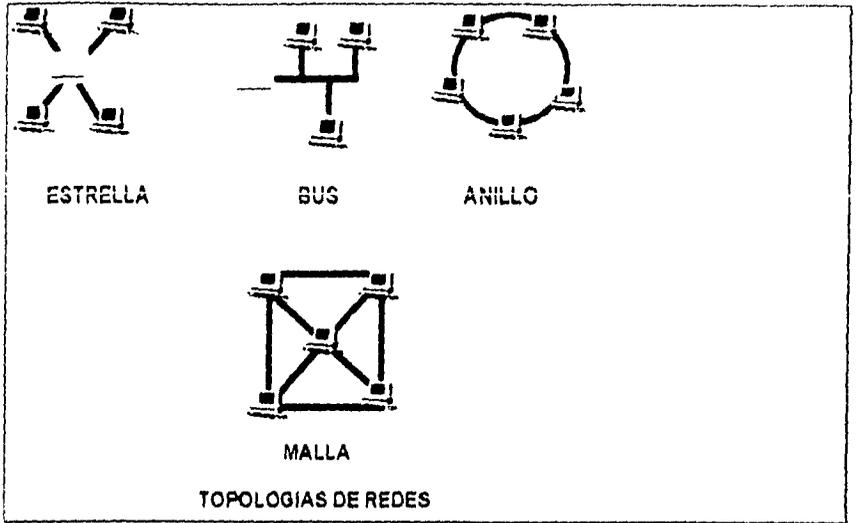
El siguiente cuadro sinóptico muestra las principales topologías de red. Se puede decir que las tres primeras son las más importantes a nivel de redes locales



La elección de una topología determinada, influye en algunas características de la red como por ejemplo:

- 1) La flexibilidad de la red para añadir más estaciones de trabajo.
- 2) El tráfico máximo de información que acepta la red sin que se produzcan interferencias continuas.
- 3) Los tiempos máximos de transmisión-recepción.
- 4) El precio de la red. Una topología mal elegida eleva los costos de la red.

en la siguiente figura, se presentan las topologías de red más usadas en el mercado.

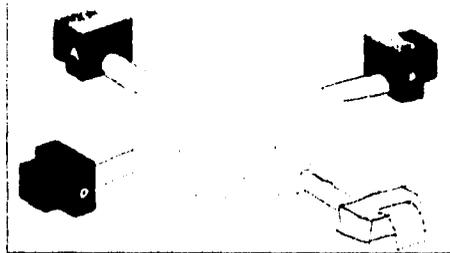


Estrella.

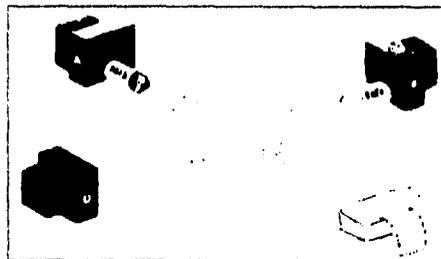
La configuración estrella se instrumenta en esquemas de comunicación punto a punto. Pocos ejemplos de esta configuración son utilizados como el de conmutador telefónico debido al problema que plantea una pieza especializada del equipo para dar servicios a todas las estaciones en los extremos de la líneas.

Esta configuración parece más idónea para un sistema de tipo intercambio en rama privada, en la cual, el principal requisito del servicio es la conmutación de datos de un punto a otro. Los requerimientos de velocidad de los datos son lentos (20 Kb/seg) y las líneas están colocadas en muchas oficinas. Este tipo de red funciona cuando se tienen potentes sistemas multiusuario conectados a ella.

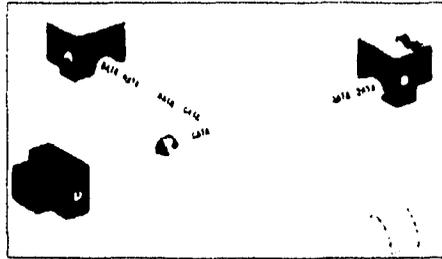
Los nodos en una red tipo estrella están conectados por líneas separadas que conducen a un HUB o estación central, esta última contiene los switches necesarios para conectar cualquier línea con otra.



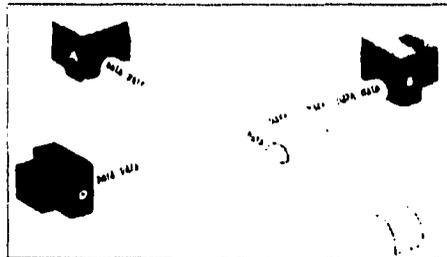
Cuando un nodo envía a la estación central un mensaje, este incluye la dirección del nodo destino, un paquete para verificación de errores y la información del mensaje. Es común que más de un nodo envíe un mensaje al mismo tiempo.



El switcheo de la estación regularmente revisa cada nodo conectado a ella, esto le proporciona la información necesaria para habilitar o deshabilitar los switches necesarios y de esta forma prevenir que un mensaje provoque alguna colisión.



Para prevenir que un nodo monopolice la red, la estación de switcheo le acepta únicamente una pequeña porción del mensaje y atiende inmediatamente una pequeña porción del mensaje del siguiente nodo, dejando en espera el resto del mensaje que tiene el primer nodo y así sucesivamente hasta atender todos los nodos.

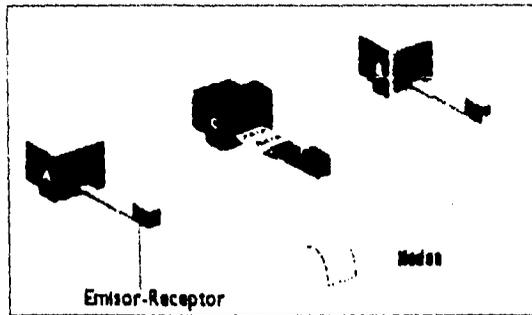


Bus.

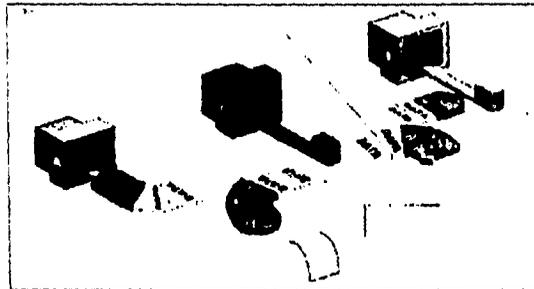
Esta configuración generalmente contiene un cable grande al cual esta conectado cada uno de los nodos, estos nodos pueden estar conectados directamente o a través de un controlador de redes que permita conexiones con estaciones múltiples. Al conectar varios controladores, obtenemos una estructura de árbol de los cables interconectados (una estrella de estrellas), esta configuración sigue siendo un bus en el sentido de que las estaciones pueden comunicarse directamente sin necesidad de una estación intermediaria.

Todas las computadoras en una red tipo BUS, estan conectadas como ramas de una línea en comun. Cada nodo tiene una única dirección, la tarjeta de red instalada en un nodo, que puede ser otra PC, un servidor de archivos o un

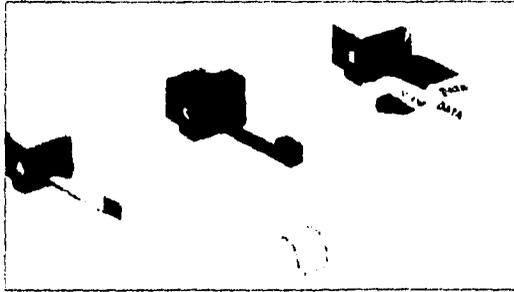
servidor de impresora, procura que ninguna otra señal se transmita por la red. Posteriormente envía un mensaje a otro dispositivo, por medio del emisor-receptor (transceiver). Cada uno de los nodos dispone de un emisor - receptor.



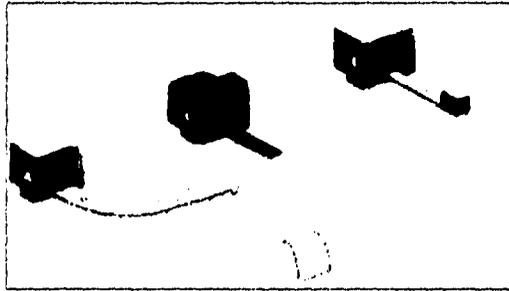
El emisor-receptor transmite el mensaje en ambas direcciones, de tal forma que llegue al resto de los nodos de la Red. El mensaje incluye los domicilios del remitente y del destinatario, paquetes de verificación de errores y la información misma.



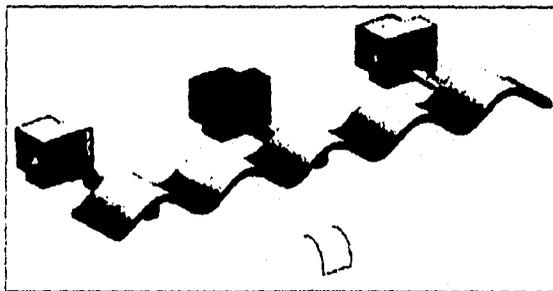
Cada nodo instalado en el bus inspecciona la información contenida en el mensaje, los nodos para los que el mensaje no va dirigido, simplemente lo ignoran. Cuando un nodo detecta su propio domicilio en un mensaje, éste lee la información, inspecciona la posibilidad de errores y transmite un acuse de recibo al emisor, utilizando el domicilio del emisor, que venía incluido en el mensaje.



Cuando dos nodos envían información simultáneamente, la colisión de los dos mensajes crea un patrón de interferencia reconocible que viaja por el bus y es detectado por los emisores.



El primer emisor que detecta la colisión, envía una señal especial que obstruye la red, de tal forma que todos los nodos sepan que esta bloqueada. La transmisión de todos los nodos se paraliza, y cada uno de los nodos espera un lapso de tiempo determinado antes de intentar enviar de nuevo su mensaje. El proceso se repite hasta que uno de los nodos envía su mensaje sin toparse con el mensaje de otro nodo.



CAP. III: REQUERIMIENTOS DE SOFTWARE Y HARDWARE

El ambiente de trabajo de SQL esta dado por el software denominado **Oracle server**, que lo distribuye la propia Corporación Oracle, el precio para la versión 6 de su producto varia, pues depende del sistema operativo y del número de usuarios; esta herramienta trabaja bajo ambiente UNIX, por lo que se debiera tener instalado un sistema operativo que corra desde este ambiente. El precio que se tenia para este producto a principios de 1995 variaba dentro de un rango que iba desde los \$700 por usuario en una pc con sistema operativo UNIX hasta \$800 para usuarios high-end systems.

El tipo de sistemas operativos dentro de los cuales puede trabajarse con las herramientas Oracle es el siguiente:

Para servidores de bases de datos (B.D.) se recomienda utilizar los siguientes Sist. Op.:

- OS/DOS V 1.21 o posterior (MS).
- NetWare V 3.11(S.B.)

para servidores de Red se recomiendan:

- MS Lan Manager.
- IBM Lan Server.
- Novell NetWare.

para estaciones de Trabajo tenemos:

- DOS 3.1 o posterior.
- Windows 3.1 o posterior.
- OS/2 V 1.21 o posterior.

Como supone el lector, tambien existen algunos requerimientos de memoria que debe cumplir cada una de las computadoras anteriormente señaladas:

- Para el caso de un Servidor, el requerimiento mínimo de memoria RAM debe ser de 8MB (MS) o 12 MB (SB).
- Para las estaciones de trabajo, se requiere un mínimo de 512Kb (DOS) o 6MB (OS/DOS).

Por otro lado, el espacio mínimo que se debe de tener en disco duro en el Servidor es de 20 MB.

Ahora bien, Oracle Server cuenta con algunas utilerías que se enlistan a continuación:

- Utilería para la Administración.
- Utilería interactiva con el usuario.

dentro de estas utilerías entran los sistemas operativos que se utilizan y el ambiente de trabajo soportado.

- DOS 3.1 o posterior.
- OS/2 V 1.21 o posterior.
- Windows 3.1 o posterior

El lenguaje del que se origino SQL fue el ANSI SQL. actualmente se tienen algunas innovaciones para el SQL como lo es el SQL Windows, que pretende migrar toda la funcionalidad del SQL original al ambiente ya conocido de Windows.

Para la versión 6 de Oracle Server se tiene la siguiente lista de valores máximos:

Tamaño de la Base de Datos:	2GB (MS), 32GB (SB).
Tamaño de la Columna:	1,962 bytes normalmente.
Tamaño de la Fila:	255 columnas.
No de Columnas en Fila:	255.
No de Filas por Tabla:	Limitado por esp. en disco.
No de Filas por Base de Datos:	Limitado por esp. en disco.
No de Tablas por Base de Datos:	2 Billones.
No de Vistas por Base de Datos:	Ilimitado.
No de tablas por Vista:	250.

CAP. IV: COMANDOS Y CONCEPTOS DE SQL/SQL PLUS

El nombre SQL es una abreviatura de las palabras en inglés Structured Query Language y fue desarrollado en los laboratorios de IBM por el Dr. Tod Codd para crear, mantener y consultar bases de datos relacionales.

SQL se introdujo como un lenguaje de consulta del Sistema R, el cual fué un proyecto de investigación que se desarrollo en 1974. El objetivo del proyecto era demostrar la aplicación práctica del modelo de datos relacional, que en ese entonces se acababa de proponer. Por ser un lenguaje no procedurai, permite que el usuario solicite a la computadora la información que desea ver y no el cómo la computadora obtendrá la información. La forma de construir una consulta a la base de datos se realiza con base en los comandos de SQL, los cuáles son pocos y la facilidad que proporcionan al poder incrementar la complejidad de las consultas que se requieran realizar, hace que la tarea sea más sencilla.

INICIANDO UNA SESION

Al inicio, en la pantalla de nuestro servidor ó computadora observamos que se nos pide un nombre (*login*) con el cual se nos reconoce como usuario, ver figura:

```
SQL Version 3
(C) Copyrights by IBM and by others 1982, 1993.
login: oracle
oracle's Password:
```

una vez introducido este dato, se nos pedirá una clave de acceso (*password*), con esta clave se nos autoriza hacer uso del software de red para ORACLE - SQL, si los datos son correctos se limpiará la pantalla y aparecerán algunas leyendas alucivas a los derechos de autor y de la compañía que lo desarrolló, acto seguido aparecerá una pantalla en la que se mencionan los datos del departamento ó compañía que tiene la licencia de usar ORACLE y los datos del responsable. Una

vez que se visualiza esta pantalla, aparece el cursor y el prompt en forma de "\$", ambos aparecen en la parte inferior izquierda de la pantalla tal como se muestra:

\$ _

Al término de nuestra labor, puede ser necesario terminar la sesión de trabajo, para lograr esto se emplea el siguiente comando.

\$ exit

al término del comando se debe de dar un ENTER, más sin embargo, cada comando estando en el prompt será necesario que se escriba en minúsculas. Para empezar a trabajar con SQL es necesario utilizar los archivos del directorio SQL, el acceso a este directorio es de la siguiente forma:

\$ sqlplus nombre_usuario/password <ENTER>

Con base a la sintaxis anterior, tecleamos lo siguiente:

\$ sqlplus system/manager <ENTER>

en la pantalla aparecerán algunas leyendas alucivas al software, es importante que los comandos se escriban en minúsculas. En el momento en que se entre a SQL aparecerá en prompt de la siguiente forma:

```

Welcome to AIX Version 3.2f
Please see the README file in /usr/lpp/bos for information pertinent to
this release of the AIX Operating System.
*****
Last unsuccessful login: Mon Apr 17 11:22:58 CDT 1995 on tty0
Last login: Tue Apr 25 14:06:58 CDT 1995 on pts/1 from 142.12.2.65
$ sqlplus system/manager
SQL*Plus: Version 3.0.0.1.5 - Production on Tue Apr 25 14:20:42 1995
Copyright (c) Oracle Corporation 1979, 1989. All rights reserved.

Connected to: ORACLE RDBMS V6.0.34.2.1, transaction processing option - Producti
on
PL/SQL V1.0.33.1.1 - Production
SQL> _
    
```

hasta aqui sólo se han tratado algunas instrucciones básicas aparte de los pasos a seguir para iniciar una sesión ORACLE y trabajar con SQL, en los siguientes puntos se hará referencia a los comando de SQL, los cuales sirven para la recuperación, manipulación, definición y seguridad de los datos

COMANDOS Y CONCEPTOS

Para pedir ayuda durante la sesión basta con escribir lo siguiente:

SQL> HELP COMMANDS

si se desea ayuda sobre un comando en específico se teclea:

SQL> HELP COMANDO

Lo anterior es posible gracias a que SQL cuenta con un diccionario de datos, un diccionario de datos es un conjunto de tablas y vistas que contienen información descriptiva acerca de: tablas, privilegios de acceso y otras cualidades de la base de datos. Las siguientes tablas son las más utilizadas para acceder el diccionario de datos:

TAB

Una lista de tablas, vistas y sinónimos que usted ha creado.

DTAB

Tablas que componen al diccionario de datos.

COL

Una lista de las definiciones de columnas de las tablas que Ud ha creado.

CATALOG

Una lista de todas las tablas a las que Ud. tiene acceso.

Si se desea obtener la descripción de una tabla específica se utiliza el comando DESCRIBE o en su forma minimizada DESC. La descripción obtenida contendrá NAME que es el nombre de la columna, NULL? que indica si se permiten nulos en esta columna y TYPE que proporciona el tipo de dato que contiene la columna.

SQL> DESC departamento

SQL cuenta con comandos para la recuperación, manipulación, definición y seguridad de los datos, la siguiente tabla muestra esta clasificación

TIPO DE COMANDO	COMANDO	ACCION
RECUPERACION DE DATOS	SELECT	RECUPERACION DE DATOS
MANIPULACION DE DATOS	INSERT	ALTAS
	DELETE	BAJAS
	UPDATE	CAMBIOS
DEFINICION DE DATOS	CREATE	CREACION DE OBJETOS

	ALTER	MODIFICACION DE OBJETOS
	DROP	BORRADO DE OBJETOS
SEGURIDAD DE DATOS	GRANT	OTORGA PRIVILEGIOS
	REVOKE	ELIMINA PRIVILEGIOS

Para la explicación de los comandos se hará uso de varias tablas que se suponen estan creadas y las cuales se ilustran a continuación:

No_EMP	NOMBRE	No_DEPTO	PUESTO	No_PC	SUELDO	ETX
850	Gonzalez P. Ramón	4	Capturista	07	800.00	524
391	Corona S. Martín	6	Vigilante	03	700.00	254
500	De Valdéz Jorge	1	Jefe_Chof	00	750.00	324
900	Zapata T. Francisco	4	Capturista	08	800.00	524
.
.
.

TABLA EMPLEADO

No_DEPTO	NOMBRE_DEPTO
4	INFORMATICA
6	SEGURIDAD
1	ENTREGAS
.	.
.	.
.	.

TABLA DEPARTAMENTO

CLAUSULA SELECT

Ahora podemos empezar a utilizar la información de las tablas anteriores, para esto empezaremos por crear los que se conoce como **QUERY**(traducido al español como consulta), y que son programas en SQL que usan cláusulas (instrucciones o comandos), de las cuales la más común y más usada es **SELECT**, que como su traducción se emplea para seleccionar información proveniente de tablas o vistas ya existentes, aunado a este comando, se utiliza un juego de instrucciones más, que a medida que se estudie este capítulo se dará cuenta de ellas así como de su función.

Por ejemplo, en base a las tablas anteriores se pide seleccionar todos los campos contenidos en la tabla EMPLEADO; para lograr lo anterior, se escribe el siguiente query:

SQL>SELECT * FROM EMPLEADO

el resultado es el siguiente:

NO_EMPLEADO	EMPLEADO	DEPTO	PUESTO	No_PC	SUELDO	EXT
850	Gonzalez P. Ramón	4	Capturista	07	800.00	524
391	Corona S. Martín	6	Vigilante	03	700.00	254
500	De Váldez Jorge	1	Jefe_Chof.	09	750.00	324
900	Zapata T. Francisco	4	Capturista	08	800.00	524

Ahora, seleccionaremos algunas columnas de la misma tabla y ordenaremos los registros por nombre de empleado:

**SQL>SELECT NOMBRE, PUESTO, SUELDO FROM EMPLEADO
2 ORDER BY NOMBRE;**

NOMBRE	PUESTO	SALARIO
Corona S. Martín	Vigilante	700.00
De Váldez Jorge	Jefe_Chof.	750.00

Gonzalez P. Ramón	Capturista	800.00
Zapata T. Francisco	Capturista	800.00

Si es necesario, se puede hacer uso de condiciones en esta selección con la cláusula **WHERE**.

```
SQL >SELECT * FROM EMPLEADO
2  WHERE No_EMP = 391;
```

No_EMP	NOMBRE	DEPTO	PUESTO	No_PC	SUELDO	EXT
391	Corona S. Martín	6	Vigilante	03	700.00	254

También podemos seleccionar datos de diferentes tablas o vistas, el siguiente ejemplo lo hace pero también damos la condición de desplegar registros que contengan un dato en común en ambas tablas, este dato es en número de departamento.

```
SQL>SELECT No_EMP, NOMBRE, SUELDO, DEPTO
2  FROM EMPLEADO, DEPARTAMENTO
3  WHERE EMPLEADO.No_DEPTO = DEPARTAMENTO.No_DEPTO;
```

No_EMP	NOMBRE	SUELDO	DEPTO
391	Corona S. Martín	700.00	6
850	Gonzalez P. Ramón	800.00	4
900	Zapata T. Francisco	800.00	4
500	De Váldez Jorge	750.00	1

En el modelo relacional, los rengiones no tienen un orden particular, el comando **ORDER BY** es la única forma como Ud. se puede asegurar de que los rengiones van a ser desplegados de acuerdo a cierto criterio.

```
SQL> SELECT No_EMP FROM EMPLEADO
2 ORDER BY No_EMP;
```

```
No_EMP
```

```
-----
```

```
391
```

```
500
```

```
850
```

```
900
```

Por default el orden es ascendente (ASC) de A a Z. Si se desea ordenar de la forma contraria se utiliza el comando **DESC**.

```
SQL> SELECT No_EMP FROM EMPLEADO
2 ORDER BY No_EMP DESC;
```

```
No_EMP
```

```
-----
```

```
900
```

```
850
```

```
500
```

```
391
```

Como se habrá observado, el query en SQL sigue un cierto orden, mismo que no se obtiene de forma aleatoria si no que obedece a la sintaxis general que se presenta a continuación:

```
SELECT    [ ALL|DISTINCT]
           (TABLE.| VISTA) [ * | EXPRESION [ALIAS] ] ]
FROM     [ (| (USER.) TABLE | VIEW || [ALIAS] ] ]
[ WHERE   CONDICION ]
```

[CONNECT BY *CONDICION* [STAR WHIT *CONDICION*]]

[GROUP BY *EXPRESION* ... [HAVING *CONDICION*]]

[[UNION | INTERSECT | MINUS]] SELECT QUERY]

[ORDER BY [[*EXPRESION* | POSN]] [ASC | DESC]]]

[FOR UPDATE OF *COLUMNA*, ... [NOWAIT]]

a continuación, se muestra una tabla en donde se explican algunos de los parámetros que intervienen en la sintaxis general del QUERY que se describió anteriormente.

PARAMETRO	SIGNIFICADO
TABLE	Nombre de la tabla, consta de 1 a 30 caracteres alfanuméricos
VIEW	Nombre de la vista, consta de 1 a 30 caracteres alfanuméricos
EXPRESION	Cualquier expresión válida de SQL.
ALIAS	Nombre que se da a un objeto (Columna, tabla o vista) y que sirve para identificarlo temporalmente
CONDICION	Condición usada para seleccionar renglones, tablas o vista
POSN	Posición de la columna en la cláusula SELECT.
COLUMNA	Nombre de alguna de las columnas pertenecientes a la tabla en uso, este nombre consta de 1 a 30 caracteres alfanuméricos

Entre las muchas instrucciones que se utilizan dentro de los QUERYs de SQL, existen dos que aunque tienen la misma función, entre ellas existe una gran diferencia, estas instrucciones son: ALL y DISTINCT.

ejemplo.

SQL> SELECT ALL No_DEPTO FROM EMPLEADO;

No_DEPTO

4

6

1

4

SQL> SELECT DISTINCT No_DEPTO FROM EMPLEADO;

No_DEPTO

4

6

1

EXPRESIONES

La sintáxis de las expresiones es la siguiente:

[Tabla o Vista | Columna o Alias][operator expression]
 Valor
 Función
 Variable de sustitución

algunas expresiones pueden ser usadas solamente en una condición y pondremos algunas de ellas a continuación.

(Lista) | (subquery)
 operadores como +, -, *, /, concatenation

Ejemplos.

Nombre_ empleado
 Nombre_ empleado.No_ departamento
 20.5
 'José Daniel'
 abs(degrees)
 '21/03/68'
 sueldo + sueldo * vacaciones

Ejemplos del uso de expresiones en queries.

Select distinct No_depto from Empleado;
 Select sueldo + 5 from Empleado;
 Select count(No_depto) from Departamento;

CLAUSULA FROM.

Esta cláusula se usa para especificar la(s) fuente(s) de información para el query, esta(s) fuente(s) suelen ser por lo general tablas o vistas.

ejemplos.

SQL > SELECT NOMBRE FROM EMPLEADO;

SQL > SELECT NOMBRE_DEPTO FROM DEPARTAMENTO;

SQL > SELECT NOMBRE FROM EMPLEADO
 WHERE EMPLEADO.No_DEPTO = DEPARTAMENTO.No_DEPTO

CLAUSULA WHERE

Mediante esta cláusula, se pueden dar condiciones que la información obtenida de una ó varias tablas o vistas debe cumplir, de esta forma uno puede disminuir la cantidad de datos desplegados; en los ejemplos anteriores se puede apreciar perfectamente el uso de esta cláusula. Al igual que la cláusula Select, existe una sintáxis general que debe cumplir la cláusula Where, dicha sintáxis se muestra a continuación:

```
[NOT] expr comp_op [ ANY | ALL ] expr[ || AND | OR || condición ]
```

ejemplo

```
NOMBRE = 'De Váldez Jorge'
```

```
NOMBRE IN ('Jorge', 'Ana', 'Gabriela')
```

usando **SELECT**

```
SQL > SELECT * FROM EMPLEADO
      WHERE SULEDO < 800;
```

No_EMP	NOMBRE	No_DEPTO	PUESTO	No_PC	SUELDO	EXT
391	Corona S. Martin	6	Vigilante	03	700.00	254
500	De Váldez Jorge	1	Jefe_Chof.	09	750.00	324

CONDICIONES

Estas son usadas junto con la cláusula WHERE y al igual que un QUERY existen algunos parámetros que pueden usarse para estructurar de un mejor modo dichas condiciones. ~~Para más información consultar la tabla 4 del apéndice, pag. 324.~~

Los parámetros y operadores comparativos son en cuestión a su funcionamiento, muy similares a los que se usan en aritmética.

Ejemplos:

```
Nombre = 'De Váldez Jorge'
```

Empleado in ('De Váldez', 'Corona')

Empleado like 'C%o_a'

Si usamos Select un ejemplo seria del siguiente modo:

```
SQL > SELECT * FROM EMPLEADO
2 WHERE SALARIO > 1000;
```

CLAUSULA NULL

Esta cláusula se utiliza ya sea para definir el valor de una columna como nulo ó como parte de una condición en un QUERY. Sin embargo, es necesario aclarar que NULL (nulo) no significa que se tenga un valor de cero sino más bien significa que el valor de ese campo no tiene longitud. Para ejemplificar lo anterior tomaremos una de las tablas anteriores, en este caso EMPLEADO, supongamos que introducimos una nueva columna llamada 'Faltas' con sus respectivos valores pero, al valor correspondiente del segundo empleado le asignaremos un nulo y al valor del último se la asignara un cero:

No_EMP	NOMBRE	No_DEPTO	PUESTO	No_PC	SUELDO	ETX	FALTAS
850	Gonzalez P. Ramón	4	Capturista	07	800.00	524	1
391	Corona S. Martin	6	Vigilante	03	700.00	254	
500	De Valdéz Jorge	1	Jefe_Chof	09	750.00	324	2
900	Zapata T. Francisco	4	Capturista	08	800.00	524	0
.	
.	
.	

Haciendo uso de los queries.

```
SQL > SELECT NOMBRE, FALTAS FROM EMPLEADO;
```

```
NOMBRE          FALTAS
-----
Gonzalez P. Ramón 1
```

Corona S. Martín

De Váldez Jorge 2

Zapata T. Francisco 0

si añadimos la unidad en cada valor de la nueva columna y la desplegamos dándole otro nombre tendremos:

SQL > SELECT NOMBRE, FALTAS + 1 "FALTAS_UNO" FROM EMPLEADO;

NOMBRE	FALTAS_UNO
-----	-----
González P. Ramón	2

Corona S. Martín

De Váldez Jorge 3

Zapata T. Francisco 1

es de notar que el valor nulo de la columna faltas_uno, correspondiente al empleado 'Corona S. Martín' no fue alterado y esto se debe a que NULL indica espacio vacío más no valor cero.

FECHAS.

ORACLE permite manipular los campos de fecha (campos *DATE*) con base al siguiente formato:

DD - MM - YY

DD = Dia

MM = Mes

YY = Año

ejemplo.

21 - MAR - 93

para desplegar el formato de fechas usando queries se emplea la siguiente función:

TO_CHAR(*DATE*, Formato)

para introducir datos o modificarlos se emplea lo siguiente:

TO_DATE(*DATE*, Formato)

a continuación se presenta una tabla en la que se muestran los campos utilizados para ambas funciones:

CAMPO	USO/SIGNIFICADO
DATE	Columna de fecha que hace referencia a la fecha que se desea modificar.
FORMAT	Formato específico en el que se desea aparezca el campo DATE
ELEMENTO	SIGNIFICADO
YY	Ultimos dos digitos del año.
YYYY	Los cuatro digitos del año.
Q	Trimestre.
MM	Mes.
MON	Abreviación del mes (tres letras).
WW	Semana del año.
DDD	Día del año.
DD	Día del Mes.
DY	Abreviación del día (tres letras).
HH	Hora del día (1 - 12).
HH24	Hora del día (1 - 24).
MI	Minuto.

SS	Segundo.
text	Texto a desplegar, entre comillas: 'XXX'.

ejemplos:

```
to_char(fecha_actual,'DD/MM/YY HH24:MI')
```

```
to_char(sysdate,'DD/MM/YY')
```

```
to_date(sysdate,'Q')
```

COMMIT Y ROLLBACK

En SQL la función COMMIT tiene como objetivo la de actualizar los cambios en las tablas y hacerlos válidos para todo usuario de la B.D., una vez utilizada esta función las modificaciones no podrán ser alteradas. Para utilizar esta función basta darla de la siguiente manera:

```
SQL> COMMIT;
```

Si, después de modificar la estructura o datos de las tablas de la B.D. desea no tomar en cuenta dichas modificaciones, se puede volver a retomar los anteriores datos utilizando la función ROLLBACK de la siguiente manera:

```
SQL> ROLLBACK;
```

cabe hacer mención que la función anterior sólo podrá ser utilizada siempre y cuando no se haya utilizado la función COMMIT. Si se desea tener siempre actualizada la B.D. basta con utilizar el comando siguiente:

```
SQL> SET AUTOCOMMIT ON
```

En cualquiera que sea el caso, mientras que no se actualice la B.D., los cambios que se realicen sólo podrán ser observados por la persona que realice los cambios, los demás usuarios seguirán con los datos anteriores y toda operación que realicen será almacenada sin problema alguno.

QUERYS MULTINIVEL

Habr  veces se necesite de multiples queries, pero cada uno de ellos puede que sea complemento de otro, por lo tanto, es posible anidar un query dentro de otro y as  sucesivamente mientras sea necesario, a esto se le llama queries multinivel; en este tipo de queries existen niveles, de los cuales el primer query es el que tiene el nivel m s alto, este query utiliza el resultado del siguiente query, que a su vez utiliza el resultado del siguiente y as  sucesivamente. Como ejemplo a esto se tiene el siguiente query:

```
SELECT NOMBRE FROM EMPLEADO
WHERE NOMBRE_DEPTO IN ( SELECT NO_DEPTO
                        FROM EMPLEADO
                        WHERE NOMBRE = 'De Valdez Jorge' );
```

CREANDO UN TABLA

Crear una tabla en SQL es relativamente sencillo, para esto se utiliza el comando CREATE TABLE NOMBRE_TABLA, dicho comando tiene la siguiente sint xis:

```
CREATE TABLE [usuario.] NOMBRE_TABLA
  [( NOMBRE_COLUMNA FORMATO [DEFAULT expr] [columna-
  constraint], ... )
  [tabla-constraint], ...
  [(columna, ...) AS query ]];
```

en la siguiente tabla se proporcionan algunos par metros correspondientes al formato para crear una tabla:

ARGUMENTO	SIGNIFICADO
FORMATO	Tipo de columna y tama�o.
Tipo de columna	Comentario
CHAR(tama�o)	Implica columnas cuyo contenido son caracteres alfanum�ricos y el tama�o m�ximo es de 255.
DATE	Columna con campo tipo fecha, el rango de la fecha esta entre 01/01/4712 AC - 31/12/4712 DC
LONG	Este tipo de columna hace referencia a caracteres alfanum�ricos, sin embargo la longitud en este caso es mucho mayor que las de las columnas tipo char siendo de 65535 caracteres, estas columnas no deben ser incluidas en subqueries, funciones expresiones, where, group by, order by, connect o distinct

NUMBER	Se manejan 38 dígitos como máximo sin incluir el punto decimal y el signo
NUMBER(tamaño)	En esta columna, el parámetro tamaño da la libertad de manejar a voluntad el número de dígitos manejados siempre y cuando este número no exceda la cantidad de 38.
NUMBER(tamaño, dec) Lo mismo que el anterior solo que se incluye también	la posición del punto decimal, cuyos rangos van de -84 a 127.
RAW(tamaño)	Se utiliza para manejar datos binarios cuyo tamaño en bytes no debe de exceder de 255.
LONG RAW	Lo mismo que el anterior con la longitud anteriormente manejada para LONG.

NULL

Como se mencionó anteriormente, el término NULL implica la ausencia de valor, más no el tener valor de cero, todas las columnas creadas tienen la posibilidad de tener valor nulo a menos que se indique lo contrario, esto es, al utilizar el comando NOT NULL.

VISTAS

Las vistas son tablas virtuales que se crean a partir de tablas reales cuya información es necesaria para algunos procesos. Se les dice vistas ya que son únicamente una proyección de los datos contenidos en tablas que no se quieren modificar, pero si se requiere de algunos datos de las mismas, estas vistas se crean de la siguiente forma:

```
SQL> CREATE VIEW [usuario.] Nombre_vista [(alias, ...)] AS query [WITH CHECK OPTION [ CONSTRAINT Nombre]]
```

usuario	Usuario creador de la vista.
Nombre_vista	Nombre de la vista, consta de 1 a 30 caracteres alfanuméricos.
alias	Es el nombre dado temporalmente a una columna, en caso de no darle nombre, toma el nombre correspondiente a la columna del query
Nombre	Nombre tipo Constraint asignado a WITH CHECK OPTION.

WITH CHECK OPTION es usado para especificar un dato que será introducido dentro de la vista

ejemplo:

```
SQL> CREATE VIEW AS
2   SELECT No_EMP, EXT FROM EMPLEADO
```

INSERTANDO RENGLONES DENTRO DE UNA TABLA O VISTA.

Para insertar renglones dentro de una tabla o vista se utiliza el siguiente comando con la sintaxis que se indica:

```
INSERT INTO [usuario] [[tabla|vista]] [(columna,...)] [[VALUES (valor,...)] query]]
```

ARGUMENTO	SIGNIFICADO
valor	Es el valor asignado a la columna en cuestión, las columnas tipo caracter deben ser encerradas por comillas simples. Las columnas tipo fecha (DATE) tienen que ser de la forma DD/MM/YY y deben ser encerradas por comillas simples. Las columnas numéricas pueden ser especificadas dentro de la forma. Las columnas anteriores pueden tener valor nulo.
usuario	Es el creador de la tabla.

ejemplo:

```
SQL>INSERT INTO EMPLEADO (No_EMPLEADO, EXT)
2   VALUES ('JERONIMO GOMEZ S',289);
```

ADICIONANDO RENGLONES A UNA TABLA

Para lo anterior se utiliza lo siguiente:

```
ALTER TABLE [usuario.] NOMBRE_TABLA
ADD (NOMBRE_COLUMNA FORMATO [EXRESION POR DEFAULT=condición] [
COLUMNNA_CONSTRAINT], ... );
```

SQL no prohíbe el adicionar columnas a una tabla ya creada, más sin embargo la(s) columna(s) creada(s) no pueden tener valor nulo (NULL) si existen renglones que correspondan a dicha columna.

ejemplo:

```
SQL>ALTER TABLE EMPLEADO
2   ADD (CURSOS_TECNICOS   NUMBER(3),
        NOMBRE_CURSO      CHAR(30),
        No_COMISION       NUMBER(4),
        LUGAR_COMISION    CHAR(30));
```

MODIFICANDO COLUMNAS DENTRO DE UNA TABLA

El comando que ayuda en este caso es el siguiente:

```
ALTER TABLE [usuario.] NOMBRE_TABLA
MODIFY (NOMBRE_COLUMNA FORMATO [EXPRESION POR DEFAULT = condición]
        [COLUMNA_CONSTRAINT], ... );
```

A una columna se le puede cambiar el tipo de columna que es o decrementar su tamaño únicamente si cada instancia de la columna dentro de la tabla tiene un valor de nulo (NULL). Por otro lado, una columna puede ser no nula (NOT NULL) si cada instancia de la misma tiene valores diferentes de nulo.

Si el tamaño de una columna especificada como no nulo (NOT NULL) es cambiado, la especificación NOT NULL podrá ser borrada si así se especifica.

ejemplo:

```
ALTER TABLE EMPLEADO
MODIFY (NOMBRE CHAR(20), SUELDO NOT NULL);
```

ACTUALIZANDO RENGLONES EN UNA TABLA O VISTA

```
UPDATE [usuario] || NOMBRE_TABLA | NOMBRE_VISTA|| [alias]
SET [COLUMNA = EXPRESION |
    (COLUMNA, COLUMNA, ... ) = (subquery)],...
[WHERE CONDICION];
```

Argumento	Significado
subquery	Un subquery en SQL, este debe retornar el valor de un solo renglón.

CUIDADO.....!

Nunca utilice en comando anterior de la siguiente forma:

```
UPDATE NOMBRE_TABLA SET NOMBRE_COLUMNNA = EXPRESION;
```

ya que de hacerlo, se modificaran todos los renglones pertenecientes a esta columna.

BORRANDO RENGLONES DE UNA TABLA O VISTA

```
DELETE FROM [usuario.] ||NOMBRE_TABLA | NOMBRE_VISTA||
[WHERE CONDICION];
```

ejemplos:

```
SQL>DELETE No_EMP,... FROM EMPLEADO
2 WHERE No_EMP = 32;
```

```
SQL>DELETE FROM EMPLEADO
2 WHERE No_EMP = 32;
```

```
SQL>DELETE FROM EMPLEADO
2 WHERE NOMBRE LIKE 'JERONIMO GOMEZ S';
```

CUIDADO.....!

Nunca utilice el comando de la siguiente forma:

```
DELETE FROM NOMBRE_TABLA;
```

pues se borrarían todos los renglones de la tabla.

BORRANDO CONSTRAINTS DE UNA TABLA

```
ALTER TABLE [usuario] NOMBRE_TABLA
DROP CONSTRAINT NOMBRE_CONSTRAINT ... ;
```

Argumento	Significado
-----	-----
NOMBRE CONSTRAINT	Nombre del constraint previamente especificado.

BORRANDO UNA TABLA

DROP TABLE [usuario] NOMBRE_TABLA;

Es necesario mencionar que Ud. sólo podrá borrar las tablas que haya creado....
ejemplo:

SQL>DROP EMPLEADO;

REMOVIENDO UN INDICE

DROP INDEX [usuario.] NOMBRE_INDICE;

REMOVIENDO UNA VISTA

DROP VIEW NOMBRE_VISTA;

SQLPLUS

Algunas de las capacidades que presenta SQL PLUS son las siguientes:

- Se incluyen herramientas para visualizar, editar y ejecutar comandos en el buffer.
- Se incluyen herramientas para visualizar, editar y ejecutar comandos interactivos.
- Permite el uso de variables.
- Incluye una lista de comandos para formatear el resultado de los "Querys", a su vez, incluye una micelanea de las utilerías y un diccionario de datos.

Buffers.

Los comandos que se utilizan dentro del buffer deben de empezar con una de as siguientes palabras:

ALTER, AUDIT, COMMENT, CREATE, DROP, DELETE,
GRANT, INSERT, LOCK, NOAUDIT, REANME, SELECT,
UPDATE, VALIDATE, REVOKE.

al final de estos comandos debe de ir un punto y coma (;)

Estos comandos del buffer no son otra cosa que los "querys" que se mencionaron anteriormente. Ahora bien, SQL PLUS ofrece algunos comandos que pueden ayudar al usuario en su trabajo:

COMANDO	SIGNIFICADO
/	Corre un comando o query cargado en el buffer
A[ppend] text	Introduce texto al final de la línea disponible
C[hange] .a.b	Reemplaza el contenido del texto a por el b
Del	Borra la línea disponible
Get file	Carga un archivo dentro del buffer
Lis[t] / NoL[ist]	
I[nput] text	Introduce nuevas líneas después de la última línea
L[ist]	Lista las líneas contenidas dentro del buffer
R[un]	Lista y ejecuta el archivo correspondiente
Sav[e] name	Salva el contenido del buffer dentro de un archivo

Ejemplo.

```
SQL>C.JERONIMO.GERARDO
```

Archivos Comando

Un archivo comando puede almacenar comandos de SQL y SQL PLUS, su principal característica radica en que tiene la capacidad de manejar variables cuyo código puede editarse y modificarse desde cualquier editor de texto, ejemplo:

```
accept No_emp prompt " Número de empleado: "
pause press <return> to continue ...
select * from empleado where No_emp = &No_emp;
```

Sugerencia:

Se recomienda el uso del símbolo "-" para continuar una línea de código en el siguiente renglón.

Variables.

Se usan para hacer los archivos comando un poco más flexibles y poderosos, habilitan el uso repetitivo de los comandos para recuperar algunos grupos de datos.

Existen tres tipos de variables:

- 1) Parámetros
- 2) Variables de sustitución
- 3) Variables de sistema

SQL PLUS puede retener valores de parámetros y variables de sustitución, si una de éstas dos variables tiene un valor almacenado se dice que esta definida. Cuando un comando utiliza una de estas variable definidas, puede utilizar tantas veces como desee esta variable, más sin embargo una variable definida no puede utilizar más de un valor , se cuenta con 250 variables que a su vez estarán divididas según un criterio que se vera más adelante.

Para limpiar estas variables se utiliza el comando:

UNDEFINE <nombre_variable>

Al momento de salir de SQL PLUS, las variables se limpian automáticamente pero se recomienda utilizar el anterior comando para evitar posibles errores con líneas en código ascii. Una variable de sustitución es precedida por un ampersand y no podrá ser definida antes de esto por los comandos ACCEPT o DEFINE.

PARAMETROS

Los parámetros se distinguen por ser enteros precedidos por un ampersand tal como se muestra a continuación:

&1,&2,&3,...

La forma de utilizarlos es:

```
rem ejemplo de parámetros
rem select indefinido
SELECT &1,&2 FROM &3
```

Para hacer uso de este select se realiza lo siguiente:

```
SQL> START LIST DISTINCT No_DEPTO EMPLEADO
```

El comando start list inicializa los parámetros &1, &2 y &3 de tal modo que al sustituir los anteriores parámetros por DISTINCT, No_DEPTO y EMPLEADO respectivamente tendremos lo siguiente:

```
SELECT DISTINCT No_DEPTO FROM EMPLEADO
```

LISTADO

Es necesario mencionar que si se requiere utilizar el comando anterior varias veces durante la sesión, no es necesario repetir el comando START tal como se indicó anteriormente, basta con escribir START LIST <enter> y el comando se ejecutara automáticamente.

Podemos cambiar los parámetros del comando anterior de ser necesario:

```
SQL> START LIST ALL
```

en este ejemplo el comando quedaria así:

```
SELECT ALL No_DEPTO FROM EMPLEADO
```

LISTADO

FORMATEO A TRAVES DE COMANDOS

Existen en SQL PLUS comandos que sirven para modificar el despliegue de reportes de tal forma que se ajusten a nuestras necesidades, algunas de las forma en que se presenta la información son:

- Desplegar el titulo del reporte en la parte superior o inferior de la hoja.
- Definir el encabezado y formato de la columnas que integren un reporte.
- Asignar valores de variables a columnas o calculos con las mismas e incluirlos
- Añadir títulos a los reportes en el lugar especificado.
- Saltar líneas o páginas completas entre un valor y otro.
- Suprimir el despliegue de columnas
- Etc.

Los comandos para formatear que veremos son TTITLE, BTITLE, COLUMN, BREAK, COMPUTE, CLEAR.

TTITLE

Con este comando, el título del reporte es desplegado en la parte superior de la página. La sintaxis general de este comando es la siguiente:

```
TTITLE [ON|OFF] TEXTO|VARIABLE ...
        COL n
        S[KIP] n
        TAB n
        LE[FT]
        CE[NTER]
        R[IGHT]
        FORMAT mode
        SQL.PNO
```

solamente el último valor de TTITLE será desplegado. La siguiente tabla proporciona información acerca de los argumentos utilizados en la sintaxis anterior.

ARGUMENTO	SIGNIFICADO
ON	Despliega el título en el reporte.
OFF	No despliega el título en el reporte.
TEXTO	Contenido del título.
COL n	Significa que el siguiente texto se desplegará en el número de columna "n"
SKIP n	Significa que el siguiente texto se desplegará después de saltar "n" líneas
TAB n	El siguiente texto se imprimirá en la posición "n"
LEFT	El siguiente texto tendrá justificación a la izquierda.
CENTER	El siguiente texto tendrá justificación centrada.
RIGHT	El siguiente texto tendrá justificación a la derecha.
FORMAT mode	Los siguientes datos tendrán el formato.
SQL.PNO	Se despliega el número de página.
VARIABLE	

CAP. V: HERRAMIENTAS ORACLE

SQL*FORMS

Una forma es una herramienta por la cual SQL permite visualizar, introducir y/o modificar la información contenida en una o varias tablas. La forma por la cual se permite esto es mediante formatos en pantalla que visualizan campos con información disponible a modificar según lo requiera el usuario. Existen dos formas de ejecutar una forma:

1.- En modo query.

Se le conoce también como cláusulas **WHERE**, permite seleccionar los registros a visualizar y/o modificar en base a un criterio preestablecido.

2.- Modo de actualizar (**UPDATE**).

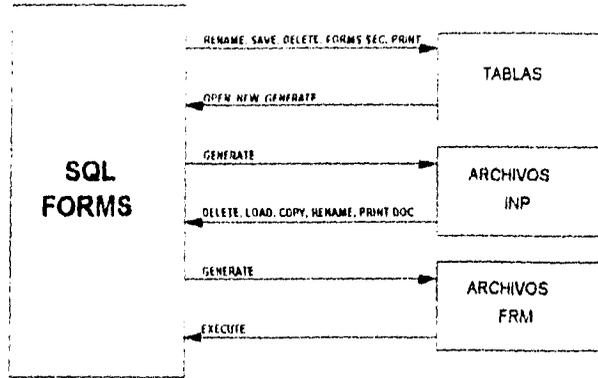
En este modo, la forma se ejecuta mostrando en pantalla el formato, mediante el cual se podrá consultar o alterar la información.

Es necesario mencionar que todo cambio efectuado dentro de la sesión no será reconocido por el sistema a menos que se de el comando **COMMIT**. Este comando se encarga de lograr que el sistema reconozca todo cambio efectuado en la base de datos, aunque los cambios pueden ser reconocidos al momento en que se sale de sesión, se recomienda utilizar el comando anterior para evitar posibles errores durante la sesión.

Antes de dar el comando **COMMIT** asegúrese de que los cambios efectuados sean los correctos, de no ser así, dichos cambios pueden borrarse usando el comando **ROLLBACK**, sin embargo, si se ejecuta el comando **COMMIT**, los cambios quedan de forma permanente.

OPERACION INTERNA DE SQL FORMS

La forma en la que trabaja SQLFORMS internamente es de la siguiente manera:



Una forma se almacena en tablas, y de hecho cada cambio en ella como, borrarla, imprimirla, salvarla o renombrarla se efectúa directamente en estas. Por otra parte, el crear una nueva forma, abrir o generar una ya existente también hace uso de tablas que si no están, son creadas.

SQLFORMS como compilador.

SQLFORMS contiene una utilidad que proporciona un compilador de dos pasos, esta utilidad se encuentra en la opción **GENERATE** que pertenece a la primera opción del menú principal. Cuando SQLFORMS genera un archivo, se crea un archivo de primer paso llamado archivo.INP, acto seguido es generado un archivo de segundo paso llamado archivo.FRM.

Al ejecutar una forma, SQLFORMS hace uso del archivo.FRM con el cual recupera la información necesaria. Los archivos INP son utilizados para llevar la forma de una computadora a otra (origen - destino). Ya con los archivos en la computadora destino, se procede a entrar a SQLFORMS, una vez dentro, use la opción **LOAD** del menú principal para cargar la forma y al salvarla escriba la ruta del subdirectorio donde salvará la forma.

Para hacer uso del ambiente de trabajo de SQLForms, debe tener la clave de acceso que le permita trabajar. Esta clave de acceso consta de un nombre de usuario (username) y una clave alfanumérica (password), en algunos casos, al usuario se le indica dar el comando:

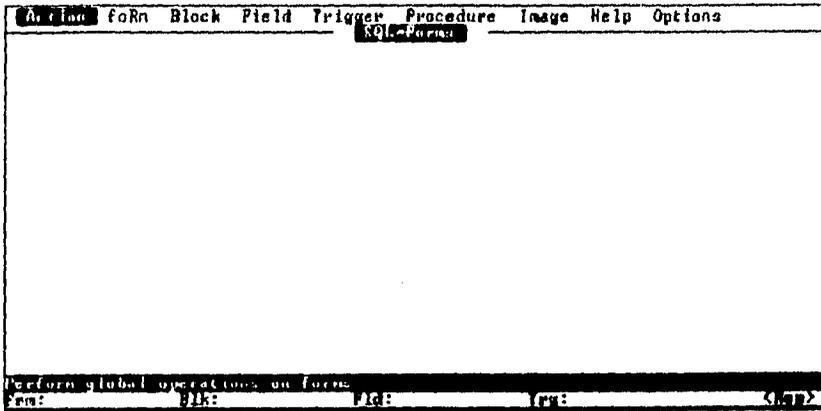
sqlforms <Enter>

y en consecuencia, aparece la pantalla inicial de la aplicación en donde se le pide los anteriores datos. Sin embargo, existe una manera de entrar directamente al ambiente de trabajo y evitar la pantalla inicial; al dar:

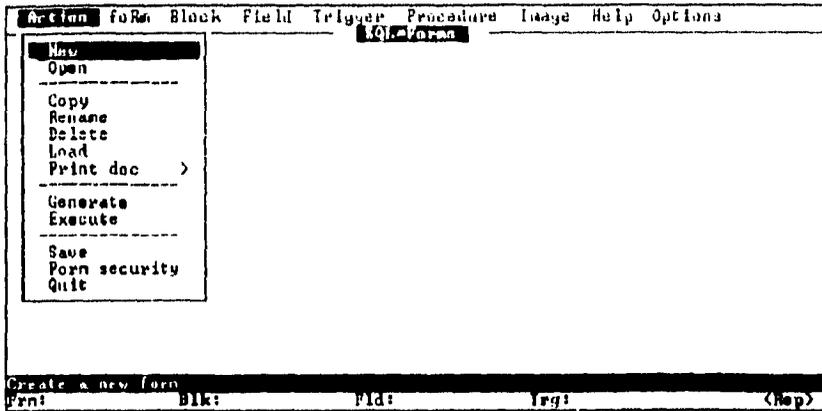
sqlforms30 nombre_de_usuario/clave_alfanumérica <Enter>

El número 30 hace referencia a la versión de SQLFORMS que se este utilizando, en este caso se utiliza la versión 3.

sqlforms reconoce al usuario y a su password. De cualquier forma, pasamos a la pantalla de trabajo en donde se puede apreciar el menú principal que proporciona las herramientas necesarias para el diseño de formas.



Una vez dentro de la aplicación, se procede a cargar o crear el archivo de trabajo, para esto accedamos a la opción Action del menú principal:



Como se observa, se dispone de 12 opciones, mismas que se describen a continuación:

Action	Crea, genera, copia, renombra, borra, carga, salva o ejecuta una forma, también genera los archivos *.JNP y *.FRM mencionados anteriormente, también imprime la documentación relacionada con la forma, incluye una opción de seguridad y la opción de salida de la aplicación.
Form	Almacena las definiciones y form - level triggers.
Block	Conserva la existencia de bloques y triggers, puede crear nuevos triggers y bloques por default and master - detail relationships.
Field	Conserva las definiciones de campo y field level triggers.
Procedure	Conserva el paquete de construcción y los procedimientos de definición de usuario.
Image	Conserva la definición de página y usa el pintado de pantalla para actualizar la página de texto, bloques y campos.
Help	Despliega el texto de ayuda, función de teclas calientes así como una lista de las tablas, columnas, formas y bloques que son accesibles al usuario en sesión.
Options	Muestra una selección de diversas opciones que pueden ser usadas para el diseño y desarrollo de la forma, tanto al correrla como al diseñarla.

El comando o secuencia de teclas a oprimir para visualizar la ayuda varía de acuerdo al tipo de máquina utilizada por lo que corresponde al usuario preguntar esta secuencia al administrador del sistema o bien el comando correspondiente.

A continuación se dará la explicación de cada selección que contiene la opción **Action** del menú principal:

New	Crea una nueva forma.
Open	Modifica una forma ya existente, esta debe estar salvada en la base de datos.
Rename	Renombra una forma.
Delete	Borra una forma.
Load	Pone a disposición una forma proveniente de un archivo *.INP.
Print Doc	Genera la documentación correspondiente a la forma en uso.
Generate	Genera el código ejecutable (archivos INP y FRM) para la forma.
Execute	Hace uso de los archivos INP y FRM generados anteriormente y ejecuta la forma.
Save	Salva la forma en la base de datos.
Form security	Otorga o remueve privilegios a la forma.
Quit	Salida de SQL*Forms

Comment: Breve comentario que describe el funcionamiento de la forma.

Procederemos a crear una pequeña forma, para lo cual introduzca el nombre de la misma(titulo) dentro del recuadro que tiene en esos momentos en pantalla y oprima Enter. Ahora que SQL*Forms tiene definida la forma, crearemos un bloque para desplegar la lista del número que corresponde a cada departamento que tenemos en estos momentos dentro de nuestra tabla DEPARTAMENTO, usemos la opción **Block** del menú principal que consta de tres opciones:

- Modify** Modifica las características del bloque disponible o crea uno nuevo.
- Trigger** Ejecuta algún proceso en específico
- Default** Crea un bloque por default.

una vez seleccionada Block, la pantalla ante Ud. sera semejante a la siguiente:

Action Form Block Field Trigger Procedure Image Help Options						
Block Definition						
Block Name	Base Table	Seq Num	Records Disp	Records Buff	Records Lines	#
depto		1	1			

dentro de esta pantalla daremos el nombre del bloque y de la tabla que usemos con el, la figura anterior muestra estos datos y aparte se pueden observar una serie de campos que despliegan y/o solicitan información relacionada con la forma, la función de estos campos es la siguiente:

- Block Name** Es el nombre del bloque a usar y se recomienda que lleve un nombre relacionado con la tabla usada.
- Base Table** Nombre de la tabla cuyas columnas son seleccionadas para crear los campos de este bloque.

empezar una vez dentro de esta pantalla, el cursor tiene movimiento libre a traves de la pantalla, por lo tanto podremos modificarla a nuestro gusto segun lo permita esta herramienta.

Empezaremos por modificar el texto que se encuentre en la forma, esto se puede realizar de dos formas, la primera consiste en posicionarse en el extremo izquierdo del texto a modificar y presionar la tecla de cortar (Cut), con esto, ese texto se borra y permite que introduzca el texto que desee. La segunda forma consiste en posicionarse en el texto y usando las funciones de suprimir e insertar escriba sobre el texto las modificaciones requeridas, borrando el texto sobrante. Aparte, se puede borrar todo el bloque con lo cual, el texto se borra automáticamente de la forma.

Lo anterior sirve para explicar la opciones de borrar (Cut) y pegar (Paste), que se utiliza para acomodar los campos y el texto dentro de la forma, para borrar basta con repetir los pasos anteriormente descritos en el párrafo anterior, para pegar coloque el cursor en la posición deseada y oprima la tecla de pegar (Paste) que por lo general es Shift-F7.

Otra de las características que tiene esta herramienta es la de poder dibujar margenes (cajas) y líneas con el fin de poder mejorar la presentación de su pantalla. Para dibujar un margen basta con ponerse en la esquina superior izq. del campo seleccionado y oprimir la secuencia de teclas para seleccionar el punto luego de esto, posicionemonos en la esquina inferior der. del campo seleccionado y oprima la secuencia de tecla para seleccionar ese punto; con lo anterior Ud. ha dibujado un margen alderredor del campo.

salgamos ahora de esta opción y retornemos al menú principal.

Opción Field

Ahora procederemos a modificar las características de los campos. Para esto seleccionemos la opción Field del menú principal mismo que cuenta con dos opciones que son:

Modify modifica las características de los campos existentes o
 crea nuevos campos.

Trigger Mantiene los niveles de disparo de campo.

seleccionemos Modify y veremos una pantalla semejante a la siguiente:

Action	Form	Block	Field	Trigger	Procedure	Image	Help	Options	
Field Definition									
Field Name	Seq Num	Data Type	Select Attributes	Fld Len	Qry Len	Dis Len	X		
	1	CHAR	< = >						
								>	
Enter the field name									
Form: depto	Blk: depto	Fld:	Trg:						<Rep>

esta pantalla presenta una serie de columnas en las cuales se puede observar la descripción de los campos que componen la forma, si no comprende la estructura de campos que se muestra, puede cambiar la forma de presentación de la pantalla actual oprimiendo el juego de teclas correspondientes a la opción Change Display Type  y la pantalla se visualizará de la siguiente manera:

Esta pantalla, en cualquiera de sus dos presentaciones sirve para adicionar campos, borrar campos, mantener atributos, ver dentro de la definición del Trigger y Analizar las definiciones de forma y bloque. La pantalla muestra diversos campos por lo cual, a continuación pondremos la definición de cada uno de ellos.

- Field Name** Nombre del campo, generalmente es el nombre de la columna de la tabla y es en ese campo en donde se adicionará información a través del teclado.
- Seq Num** Es el número de secuencia del campo dentro del bloque. Cuando el usuario accesa al siguiente campo o al anterior el cursor toma en cuenta esta secuencia para determinar su posición.
- Data Type** Se refiere al tipo de datos (numérico, carácter, etc.) que recibirá el campo.
- Select Attributes** Son las características del campo.
- Fld Len** Es el número máximo de caracteres que se pueden introducir al adicionar o actualizar un dato.
- Qry Len** Es el número de caracteres que se pueden introducir si se ejecuta un query.

Dis Len	La longitud de caracteres que puede visualizar de un dato un usuario.
X	Coordenada X o posición horizontal de el campo en la pantalla.
Y	Coordenada Y o posición horizontal de el campo en la pantalla.
Page	Número de página en donde el campo esta posicionado.
Editor Attributes	Tomando en cuenta las coordenadas X y Y
Format Mask	Es la forma en la que deseamos se visualice el campo, por ejemplo podemos dar formato a un campo numérico que contenga números telefónicos.
Default	Es el valor por default que se le da a nuevos campos. Este valor puede ser una constante como 'Buen Día', una variable del sistema como \$\$DATE\$\$, una copia de algún otro campo como Número_depto o una secuencia numérica generada automáticamente.
Hint Text	Es el texto alicivo al tipo de dato que deberá introducir dentro del campo como 'Telefono: '
Valid Range Low	Es el menor valor que el usuario puede introducir.
Valid Range High	Es el mayor valor que el usuario puede introducir.
Enforce Foreign Key	Forza a aceptar una llave foránea.
List of Values Title	Es el título a desplegar en la parte superior de la pantalla que corresponde al valor de los campos extraídos de las tablas.
List X	Es la coordenada horizontal correspondiente al campo List of Values Title.
List Y	Es la coordenada vertical correspondiente al campo List of Values Title.
Comment	Comentarios que orienten al personal para futuras modificaciones.

Tipos de campo de datos.

Dentro de la pantalla posicionemonos en la columna "Data Type" y observaremos la siguiente pantalla:

Field Definition							
Field Name	Seq Num	Data Type	Select Attributes	Fld Len	Qry Len	Dis Len	X
	1	CHAR	< * >				

From: depto Bk: depto Fld: Yrg: <List><Ins>

si pedimos la ventana "Show List of Values" se observará una lista de los tipos de datos que puede adquirir el campo.

Data Type Field	
Use the Data Type field to indicate the type of characters that you want SQL*Forms to allow as input in a particular field, and how you want those characters to be displayed on the screen.	
Each field can have only one data type. The valid data types are:	
-- ALPHA	Alphabetic
-- CHAR	Characters (letters, digits, blank spaces, and the special characters \$, %, @, and _)
	Default: DD-MON-YY
-- DATE	Date
	Default: DD-MON-YY
-- DATETIME	Date and time
	Default: DD-MON-YY HH:MI:SS
-- EDATE	European date (obsolete)
	Default: DD/MM/YY

From: depto Bk: depto Fld: Yrg: <Ins>

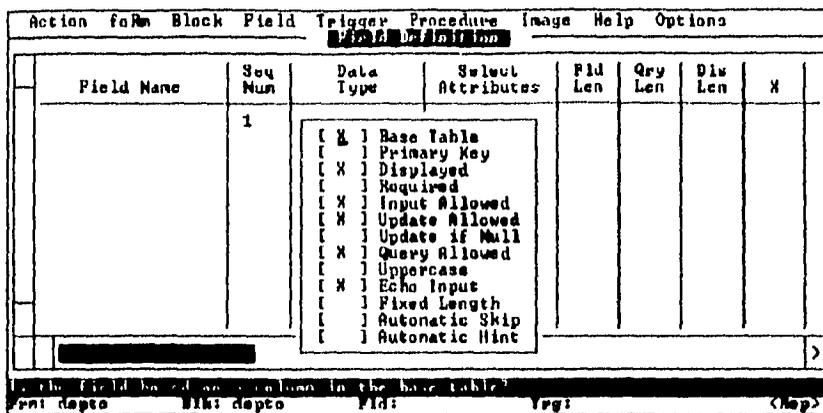
la descripción de las características de estos valores se dan a continuación.

Char Indica valor de caracter, es decir, puede tomar casi cualquier ya sea número, letra o signo, sin embargo este dato carecerá de valor numérico.

- Number** Numérico real, puede ser entero o en notación regular como 543.21 o de notación científica como 789.23 E-5.
- RNumber** Idéntico al anterior pero justificado a la derecha.
- Date** Campo fecha, despliega el día, mes y año según el formato dd/mm/yy.
- JDate/EDate** Despliega la fecha en el formato mm/dd/yy, el tipo EDate lo despliega de la forma dd/mm/yy. Este campo puede ser de tipo numérico almacenado la cantidad de días.
- Time** Despliega la hora en el formato hh:mm:ss. Si el dato es numérico, entonces ese valor sera la hora expresada en segundos.
- Alpha** Caracteres alfanuméricos.
- Int** Entero.
- RInt** Entero justificado a la derecha.
- Money** Numérico con dos decimales.
- RMoney** Numérico con dos decimales pero justificado a la derecha.
- DateTime** Es la fecha y hora expresados en un sólo campo.
- Long** Tipo alfanumérico pero de 65536 caracteres de extensión.

Atributos de Campo

Si nos posicionamos en la siguiente columna estaremos dentro en los atributo de ese campo, seleccionando la lista de estos atributos con la secuencia de teclas adecuada observaremos lo siguiente:



las características de estos atributos se listan a continuación:

- Base Table** Se refiere al nombre de la tabla de la cual se obtiene el contenido del campo seleccionado, el nombre de este campo puede ser idéntico a la columna de la tabla.
- Primary Key** Llave primaria.
- Displayed** Permite al usuario ver el campo en la pantalla final.
- Required** Indica si el usuario deberá de introducir un valor por cada registro nuevo.
- Input Allowed/Aceptar Entrada**
- Update Allowed** Aceptar Actualización
- Update if Null** Actualizar si es nulo
- Query Allowed** Query aceptado
- Uppercase** Toda letra se convertira en mayúscula.

- Echo Input** Indica que el valor del campo usa criterios de entrada que serán desplegados en la pantalla.
- Fixed Length** La longitud de los valores deberá ser igual a la longitud del campo.
- Automatic Skip** El cursores pasará automáticamente al siguiente campo cuando se introduzca un valor en el campo actual.
- Automatic Hint** El texto de ayuda correspondiente al campo en uso será desplegado una vez que el cursores este dentro del campo.

Definición de Bloques

Dentro de la opción Block del menú principal seleccionemos Modify, la pantalla será semejante a la sig.

Action Form Block Field Trigger Procedure Image Help Options						
Block Definition						
Block Name	Base Table	Seq Num	Records Disp	Records Buff	Records Lines	A
depto	departamento	1	1			

Form: depto Buf: depto Vid: Vgr: (F5)

si así o desea, puede cambiar el formato de la pantalla actual con la secuencia de teclas correspondiente, las opciones Block Name, Base Table, Seq Num y Records Disp se explicaron anteriormente, por otro lado las opciones restantes que muestra la pantalla son las siguientes:

- Records Buff** Es el número de registros que puede almacenar el buffer y que SQL*Forms ejecuta como un query.
- Array Size** Es el tamaño del arreglo de datos proveniente de la base de datos en uso.

Primary Key	Llave primaria.
In Menu	Se refiere a la opción de poder o no desplegar las opciones de la forma dentro de un menu.
Menu Description	Es el título que llevará el menu de opciones dentro de una forma.
Default Where/Order By	Condiciones de la selección de registros dentro de un bloque.

Triggers

Un trigger es un paquete de acciones definidas por el desarrollador y que son activadas por un evento o condición. Los Triggers tienen muchas aplicaciones como son:

- Limitan el uso de una forma o bloque a personal autorizado.
- Limitan el uso de una forma o bloque a una hora en específico.
- Incluyen criterios de selección dentro de un query..
- Calculan el valor de un campo basado en otros valores del registro.
- Pueden calcular el promedio, máximo y mínimo, desviación estándar, varianza, suma de campos o registros seleccionados, etc.

- Recuperan o validan información cuando un valor es introducido a un campo.
- Ejecutan algunas acciones si el usuario presiona una tecla.
- Llaman otra forma.
- Pueden controlar el flujo del cursor dentro de un campo si se llama en este a otro campo o forma.
- Pueden actualizar bases de datos, tablas con procesos ya diseñados.
- Controlan el ambiente multiusuario.
- Llaman a procedimientos externos.

Tipos de Triggers

Dentro de SQL*Forms existen tres tipos de triggers, su explicación se presenta en seguida.

- 1.- Event-Triggers: Se presentan como resultado de un evento.
- 2.- Key-Defined Triggers: Triggers definidos para una tecla
- 3.- User-Named Triggers: Triggers usados por el usuario

Activación de Triggers

Las notas anteriores indican que los triggers son enlaces de eventos, por ejemplo, se puede usar una para limitar el acceso de personal a una forma en específico, esto es el enlace de un evento para introducirse a una forma. Entre los eventos que pueden activar un trigger se encuentran:

- Introducirse en una forma, bloque o campo.
- Ejecutar un query.
- Cambiar un valor de campo.
- Agregar, cambiar o borrar un registro de la tabla base.
- Salir de una forma, bloque o campo.

- Procesar funciones o eventos.
- Presionando una tecla especifica.

se puede crear un trigger para que sean activados indirectamente dentro de algún evento.

Niveles

Esto se refiere a que los triggers actuan solo en donde fueron definidos, por ejemplo, si un trigger se definió en el bloque 2 de la forma 1, solo ahí se podrá ejecutar el trigger, si se definió en la forma 5 solo ahí podrá ser ejecutado.

Para adicionar un trigger a nivel forma, seleccione trigger de la opcion Form en el menú principal.

Para adicionar un trigger a nivel bloque, seleccione trigger de la opcion Block en el menú principal.

Para adicionar un trigger a nivel campo, seleccione trigger de la opcion Field en el menú principal.

Select Statements

Se refiere al código que contiene el trigger para realizar su función, por ejemplo:

- Para buscar valores de campo en tablas ...
- Para ejecutar operaciones en campos pertenecientes o no a la base de datos.
- Para copiar registros de una tabla a otra usando un "insert".

SQL*Forms posee una clausula especial para introducir datos en campos o variables llamada "into", por ejemplo:

```
SELECT NOMBRE_DEPTO
INTO :DEPTO.NOMBRE_DEPTO
FROM DEPTO
WHERE NUM_DEPTO = DEPTO.NUM_DEPTO;
```

aquí se ve que se seleccionara el nombre del departamento de la tabla DEPTO y se introducira a la columna NOMBRE_DEPTO de la misma tabla. En otras

palabras el valor seleccionado sera introducido en el campo NOMBRE_DEPTO del bloque DEPTO.

Procedimientos de empaquetado

SQL*Forms dispone de procedimientos de empaquetado que pueden ser invocados por un trigger u operador. Estos procedimientos son usados para controlar el flujo del cursor (navegación) dentro de un bloque o forma, para ejecutar acciones en aplicaciones específicas, para ayudar a mantener los datos en forma eficiente reduciendo valores o procedimientos repetitivos, etc. Algunos de los procedimientos de empaquetado son los siguientes:

Abort_Query	Aborta la operación del query en uso.
Anchor_View	Especifica la posición exacta de la página a ser visualizada en la pantalla.
Bell	Sonido de campana.
Block_Menu	Activa el menu que el usuario desea utilizar.
Break	Ejecuta el procedimiento paso a paso.
Call	Ejecuta una forma siempre y cuando esta exista.
Call_Input	Detiene la ejecución del proceso pero no de la forma.
Call_Query	Ejecuta los querys que contiene la forma, cuando acaba, llama a la forma.
Clear_Block	Limpia el bloque actual.
Clear_Eol	Limpia el valor del campo actual hasta fin de línea.
Clear_Field	Limpia el campo actual.
Clear_Form	Limpia la forma actual.
Clear_Record	Limpia el registro actual.
Commit_Form	Salva las modificaciones hechas a la B.D. desde la forma actual.
Copy	Copia el valor dentro del campo.

Count_Query	Cuenta el número de renglones obtenidos como resultado del query.
Create_Record	Crea un registro en blanco despues de la linea disponible.
Default_Value	Copia un valor dentro de una variable nula, si la variable no existe, se crea y despues se copia el valor dentro de ella.
Delete_Record	Borra el registro actual.
Display_Error	Despliega el error ocurrido.
Display_field	Despliega el valor del campo.
Display_Page	Despliega la página.
Down	Mueve el cursor al siguiente campo.
Duplicate_Field	Copia el registro anterior en la linea actual.
Duplicate_Record	Copia los registros del campo anterior en los del actual.
Edit_Field	Campo de edición.
Enter	Valida el campo actual.
Enter_Query	Pone el bloque en modo query y permite al usuario introducir uno.
Erase	Borra una variable global de la memoria.
Execute_Query	Ejecuta el quey disponible.
Execute_Trigger	Ejecuta el trigger disponible.
Exit_Form	Salda de la forma en uso.
First_Record	Mueve el cursor al primer registro del bloque.
Go Block	Mueve el cursor al bloque especificado.
Go_Field	Mueve el cursor al campo seleccionado.

Go_Record	Mueve el cursor al registro deseado.
Help	Despliega el mensaje de ayuda.
Hide_Menu	Limpia el menu de SQL*Menu.
Hide_Page	Limpia una pagina en particular de la pantalla.
Host	Ejecuta un comando del sistema operativo.
Last_Record	Mueve el cursor al último registro del bloque.
List_Values	Despliega la lista de los valores permitidos para ese campo.
Lock_Record	Permite cerrar el uso de un registro específico.
Message	Despliega un mensaje de 78 caracteres como máximo.
Move_View	Limita a usuarios seleccionados ver una forma completa.
New_Form	Corre una forma, si se sale de esta no se regresa a la anterior o posterior.
Next_Block	Mueve el cursos al siguiente bloque.
Next_Field	Mueve el cursor al siguiente campo.
Next_Record	Mueve el cursor al siguiente registro dentro del bloque.
Next_Set	Siguiente posición.
Pause	Detiene momentáneamente la ejecución.
Post	Procesa las transacciones pero no las valida con un commit.
Previous_Block	Mueve el cursor al bloque anterior.
Previous_Field	Mueve el cursor al campo anterior.
Previous_Record	Mueve el cursor al registro anterior del bloque en uso.
Print	Almacena la forma en un archivo o la imprime.

Redisplay	Refresca la pantalla.
Replace_Menu	Especifica el menu a reemplazar.
Resize_View	Modifica el tamaño de la página en la pantalla.
Scroll_Down	Usado en campos donde se visualizan muchos registros, permite al usuario recorrer cada uno de ellos dentro de una zona determinada en forma descendente.
Scroll_Up	Usado en campos donde se visualizan muchos registros, permite al usuario recorrer cada uno de ellos dentro de una zona determinada en forma ascendente.
Set_Field	Cambia el atributo del campo.
Show_Keys	Despliega la lista de las tecla de funcion.
Show_Menu	Despliega el menu actual.
Show_Page	Especifica la página a desplegar en la pantalla.
Synchronize	Sincroniza
User_Exit	Salida de usuario.

Variables de sistema

Estas variables son empleadas (si así se requiere) por SQL*Forms cada vez que se entra en sesión. Cuando son usadas en un trigger estas deben de tener antepuesta la palabra "SYSTEM" como sigue: SYSTEM.BLOCK_STATUS. Algunas de las variables de sistema se enuncian a continuación:

BLOCK_STATUS	Es el estado del bloque actual, teniendo como posibles resultados: CHANGED (esto es, el último registro ha sido cambiado), NEW (nuevos registros han sido añadidos y ningún registro ha sido cambiado) y QUERY (sólo se tienen registros provenientes de un query, ninguno ha sido cambiado).
CURRENT_FORM	Nombre de la forma actual.
CURSOR_BLOCK	Nombre de la forma actual.

CURSOR_FIELD	Nombre del bloque y campo en el formato block.field.
CURSOR_RECORD	Número del registro actual.
CURSOR_VALUE	Es el valor del campo actual.
FORM_STATUS	Es el estado de la forma actual, sus posibles valores pueden ser CHANGED (el último registro ha sido cambiado) y NEW (nuevos registros han sido añadidos, ningún registro ha sido cambiado).
LAST_QUERY	El último query ha sido ejecutado.
LAST_RECORD	Se refiere a si el registro actual es el último en el bloque, los valores pueden ser: TRUE o FALSE que indican cierto o falso.
MESSAGE_LEVEL	Indica el tipo de mensaje a desplegar, por ejemplo: 0 (despliega todos los mensajes), 5 (no se despliegan mensajes de información), 10 (no se despliegan operator mistakes), 15 (no se despliegan mensajes concernientes al ...), 20 (no se despliegan mensajes que indiquen los problemas que se tienen al usar triggers), 25 (no se despliegan mensajes que indiquen cuando un problema genere otro)
RECORD_STATUS	Es el estado del registro actual, los posibles valores de este son: CHANGED (el último registro ha sido cambiado), INSERT (el registro es nuevo y esta siendo validado), NEW (el registro es nuevo), QUERY (el registro viene de un query y no puede ser cambiado).
TRIGGER_BLOCK	El nombre del bloque estará disponible una vez que el trigger sea ejecutado.
TRIGGER_FIELD	Los nombres del bloque y campo estarán disponibles una vez que el trigger sea ejecutado. El formato a usar el block.field.
TRIGGER_RECORD	Es el número del registro que está siendo procesado por SQL*Forms.
\$\$DATE\$\$	Usado en el atributo de campo FIELD, es la definición de pantalla para la fecha actual.

\$\$DATETIMES\$\$ Usado en el atributo de campo FIELD, es la definición de pantalla para la fecha y hora actual.

\$\$TIMES\$\$ Usado en el atributo de campo FIELD, es la definición de pantalla para la hora actual.

Variables Globales

Una variable global es aquella que el desarrollador crea y pone a disposición en cada sesión de SQL*Forms. Las variables globales tienen el formato:

GLOBAL.nombre_de_la_variable.

Ahora, volvamos a introducirnos a la opción **Trigger** del menú principal y se verá una pantalla semejante a esta:

Action	Form	Block	Field	Trigger	Procedure	Image	Help	Options
Trigger Definition								
		Trigger Name	Block Name	Field Name	Trigger Style	Show Key		
		[REDACTED]	depto		V3	[X]		
>								
<small> From: depto Block: depto Field: Trg: (1/18) (Map) </small>								

la explicación de los campos que se visualizan en esta pantalla es la siguiente:

- Trigger** Nombre del Trigger.
- Block** Nombre del bloque que contiene el trigger a nivel-bloque o nivel-campo.
- Field** Es el nombre del campo, los nombres del campo y bloque deberan estar especificados por el trigger nivel-campo.
- Trigger Style** V2 si el tipo de trigger corresponde a la versión 2 de SQL*Forms, V3 si corresponde a la versión 3 de SQL*Forms.

- Show Keys Indica si se despliegan o no la descripción de teclas en la pantalla que definen cada uno de los triggers.
- Description Es el texto que describe al trigger en la pantalla de definición de teclas mencionada en el punto anterior.
- Trigger Text Es el cuerpo del trigger.
- Comment Nota informativa acerca del funcionamiento del trigger.

ejemplo

Este ejemplo consta de ejecutar un trigger despues de que la tecla de pasar al siguiente campo sea presionada. El texto de este trigger se presenta a continuación:

```

DECLARE
    dummy          CHAR(1);
    already_exists EXCEPTION;
    not_new        EXCEPTION;
BEGIN
    if :system.record_status != 'INSERT' and
       :system.record_status != 'NEW'
    then raise not_new;
    end if;
    SELECT 'a'
           INTO      dummy
           FROM      odemo.departamento
           WHERE     numero_depto = :num_depto;
           RAISED already_exists;
EXCEPTION
    WHEN already_exists
    THEN
        MESSAGE('Ojo!!! El número de departamento ya existe');
        CLEAR_RECORD;
        RAISED FORM_TRIGGER_FAILURE;
    WHEN OTHERS
    THEN
        NEXT_FIELD;
END;
```

Para hacer válido este trigger basta con oprimir **Accept**, en cuanto a la forma, elija de la opción **action** las opciones **generate** y **execute**.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

PROCEDIMIENTOS

Un procedimiento (procedure) es un grupo o paquete de comandos de PL/SQL que son llamados mediante un trigger u otros procedimientos.

Los procedimientos tienen una sintaxis básica que se proporciona a continuación:

```
PROCEDURE definicion_del_procedimiento
{declaracion_válida ... }
BEGIN
    cuerpo_del_programa_en_PL/SQL ...
    [EXCEPTION exception_handler]
END;
```

ejemplo:

```
PROCEDURE chequea_sueldo IS
    sueldo_max      CONSTANT NUMBER(6) :=2000;
    sueldo_depto    odemo.empleado.sueldo%TYPE;
    msg_text        odemo.temp.mensaje%TYPE;
BEGIN
    SELECT SUM(sueldo)
        INTO :global.sueldo_depto
        FROM odemo.empleado
    WHERE numero_depto = :numero_depto;
    IF :global.sueldo_depto > sueldo_max THEN
        msg_text := 'Salario alto' || to_char(sueldo_max);
    ELSE
        msg_text := 'Salario bajo' || to_char(sueldo_max);
    END IF;
    INSERT INTO odemo.temp (numero_depto, mensaje)
        VALUES (:numero_depto, msg_text);
    COMMIT;
    EXCEPTION
    WHEN NO_DATA_FOUND THEN
        msg_text := 'Departamento No:' || :numero_depto || 'No existe';
        mensaje(msg_text);
    WHEN OTHERS THEN
        msg_text := to_char(sqlcode) || sqlerrm;
        mensaje(msg_text);
END;
```

Opción PROCEDURE

La sintaxis de esta declaración es:

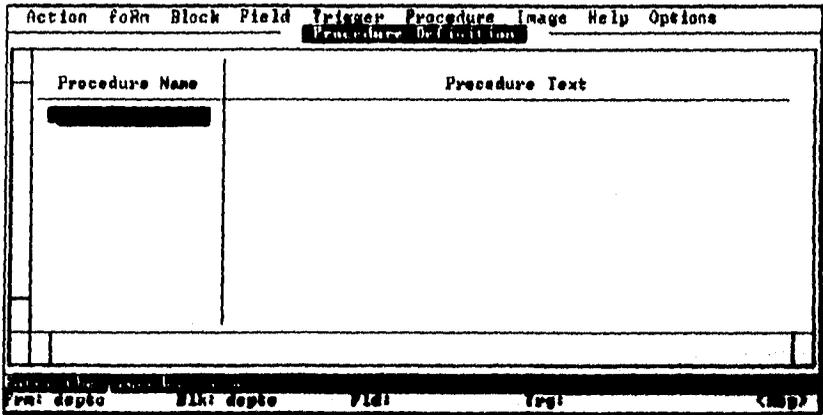
PROCEDURE nombre_procedimiento [(variable [mode] type [:=value],...)] IS

donde:

- nombre_procedimiento Nombre del procedimiento.
- variable Nombre de la variable que recibirá los valores.
- mode Se refiere al modo IN | OUT | IN OUT
- type BOOLEAN | CHAR | NUMBER | DATE
- value Valor que tomara la variable

ejemplo

Para no confundirnos mucho, retomaremos el ejemplo anterior, primero crearemos el procedimiento, seleccione **Procedure** del menu principal, aparecerá una pantalla semejante a la siguiente:



aquí, Procedure Name indica el nombre único que tendrá nuestro procedimiento a nivel forma, este nombre debe ser igual al que se define en el texto de campo. Procedure text es el cuerpo del programa que describe al procedimiento arriba mencionado. Por ejemplo:

Si en nuestra forma queremos introducir el número de departamento e inmediatamente pasar al siguiente campo, podemos hacerlo mediante un procedimiento como sigue:

```

PROCEDURE NUMERO_DEPARTAMENTO IS
  dummy          CHAR(1)
  already_exists  EXCEPTION;
  not_new         EXCEPTION;
BEGIN
  if :system.record_status != 'INSERT' and
     :system.record_status != 'NEW'
  then raise not_new;
  end if;
  SELECT 'a'
         INTO dummy
         FROM odemo.departamento
         WHERE numero_depto = :numero_depto;
  RAISED already_exists
EXCEPTION
  WHEN already_exists
  THEN
    MESSAGE('El numero de depto ya existe');
    CLEAR_RECORD;
    RAISED FORM.TRIGGER_FAILURE;
  WHEN OTHERS
  THEN
    NEXT_FIELD;
END;

```

si presionamos **Accept**, SQL*Forms compila el procedimiento y en caso de no haber errores se visualizara la leyenda 'No Compilation Errors', en caso contrario se visualizara el error encontrado.

Ahora seleccionemos del menu principal la opción **trigger** el **trigger KEY-NXTFLD** de tal forma que se incluya solamente **NUMERO_DEPTO**, presionando **Accept**, se actualizara el archivo. En la opción **Action**, genere y ejecute la forma.

Opción IMAGE

Esta opción es usada para crear y manipular páginas de SQL*Forms, también para invocar el diseñador de pantalla (**screen painter**) manipulando así texto y objetos. Se puede observar que se trata de una interfaz entre la definición de páginas y el diseñador de pantallas.

Para crear y/o modificar páginas, seleccionemos las opción **Image** del menu principal, esta opción consta de dos opciones :

- Modify** Crea y modifica páginas, también permite ver información para más de una pág.
- Painter** Crea, modifica, selecciona, corta y pega objetos; introduce y edita texto, dibuja líneas y cajas.

al seleccionar la opción modify, la pantalla cambia a la siguiente:

Action foRn Block Field Trigger Procedure Image Help Options										
Page Definition										
Page Number	Pop Up	Pg Sz X	Pg Sz Y	Vu Sz X	Vu Sz Y	Vu Lc X	Vu Lc Y	Vu Pg X	Vu Pg Y	U
1	()									

Enter the page number of the page:

Page: Block: Field: Trigger: Procedure: Image: Help: Options:

las opciones que aparecen en la pantalla se explican a continuación:

Page Number Número de una página en particular.

~~**Pop Up** Cuando una página es~~

Page Size Es el tamaño horizontal (X) y vertical (Y) de la página.

View Size Es el tamaño inicial de la página, una vez definido este, la página puede hacerse más grande o más pequeña según se necesite, sin embargo, no puede ser menor que un caracter.

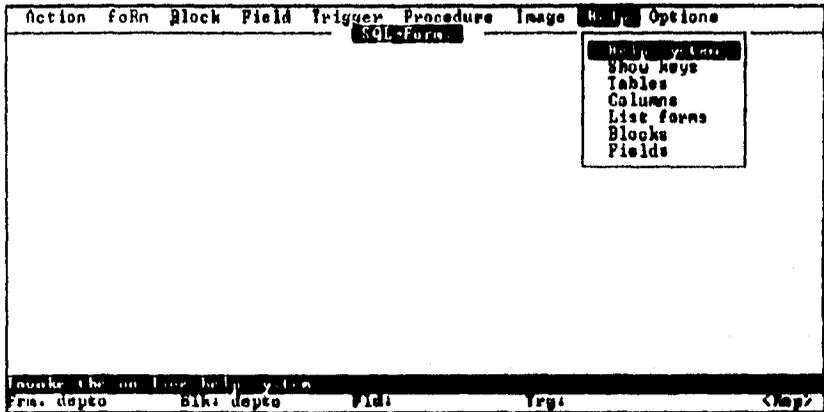
View Loc La localización inicial de la esquina superior izq. de una vista en la pantalla.

View Page La localización inicial de la esquina superior izq. de una vista en una parte determinada de la página que el usuario esta viendo.

- Border** Indica si se visualizara o no un borde en la vista.
- Vertical Scroll Bar** Da la opción de que se pueda visualizar una barra de desplazamiento vertical.
- Horizontal Scroll Bar** Da la opción de que se pueda visualizar una barra de desplazamiento horizontal.
- Title** Indica si se desplegará o no un título en la parte sup. central de la pantalla en una vista.

La opción HELP

En esta opción se proporciona una ayuda general al desarrollador en cuanto a requerimientos del sistema, función de algunas teclas o combinación de ellas, tablas, columnas, listado de formas disponibles, bloques, campos, etc. La pantalla que corresponde a esta opción es la siguiente:



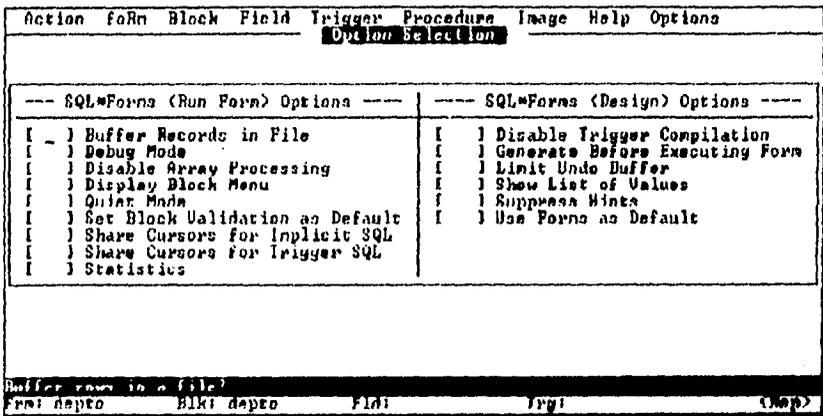
y la explicación de cada opción es la siguiente:

- Help system** Acceso a la documentación correspondiente a SQL*Forms.
- Show Keys** Informa acerca de la función de algunas teclas o combinaciones de ellas.
- Tables** Despliega la lista de tablas disponibles al usuario.
- Columns** Es la lista de las columnas pertenecientes a la tabla base que se este ocupando en ese momento.

- List Forms Proporciona una lista acerca de la formas que pertenecen al usuario y que estan salvadas en la base de datos.
- Blocks Lista de los bloques asociados a la forma actual.
- Fields Lista de los campos asociados al bloque actual.

La opción **SELECTION**

Esta opción es usada para definir el medio ambiente que se trabaja durante una sesión en SQL*Forms. La pantalla perteneciente a esta opción es la siguiente:



como se observa, la pantalla esta dividida en dos categorías, cada una contiene opciones que afectaran a la forma sólo en el ambiente que se define arriba de cada pantalla.

Opciones para SQL*Forms (Run Forms)

Estas opciones afectaran a la forma unicamente si esta se ejecuta desde el menu principal, cabe mencionar que estas opciones regresaran a su estado inicial una vez que se salga de sesión.

Buffer Record in file Indica si se utiliza parte de la memoria para guardar registros.

Debug Mode Ejecuta paso a paso la forma, indicando es estado de las variables, messenger, campos, etc que se vean afectados durante la ejecución.

- Display Block Menu Despliega un menú de bloques en el momento en que se ejecute la forma.
- Quiet Block Inhabilita el beep o sonido de campana.
- Set Block Validation as default Valida la posición del bloque por default.
- Share Cursors for Implicit SQL No toma en cuenta los queries de SQL.
- Share Cursors for Trigger SQL Toma en cuenta los queries de SQL.

- Opciones para SQL*Forms (*Desing*)
 - Disable Trigger Compilation Deshabilita el Trigger o compilación por procedimiento durante la sesión de diseño y espera a que sea generada y que los procedimientos y triggers sean ejecutados.

 - Generate Before Executing Form Esto indica que la forma se generara antes de que sea ejecutada por el usuario.

 - Limit Undo Buffer Limita el número de operaciones a ser almacenadas en el buffer.

 - Show List of Values Se despliega automáticamente una lista de valores.

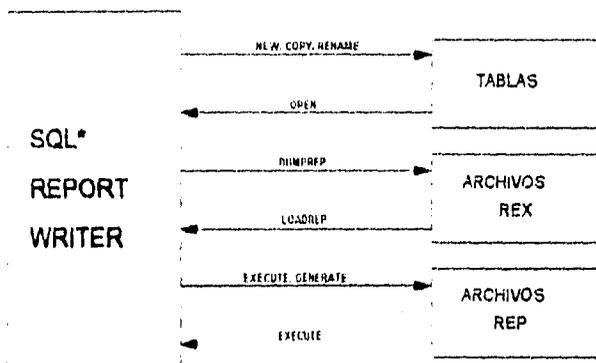
 - Suppress Hints Supresión indirecta.

 - Use Forms as Def Cambia el desplegado inicial de una forma.

SQL*ReportWriter

Esta utilidad es para el usuario final una forma de poder crear, imprimir y/o diseñar reportes, para el diseñador, es una utilidad empleada para crear y/o mantener reportes en el sistema.

Esquema del funcionamiento del SQL*ReportWriter.



Como editor, SQL*ReportWriter se comporta de la siguiente manera:

Para crear o editar un reporte, debemos introducir las especificaciones dentro de celdas que graban esta información en las tablas. Cuando se edita un reporte ya existente, SQL*ReportWriter obtiene toda la información necesaria de estas tablas.

Como compilador, SQL*ReportWriter tiene el siguiente comportamiento:

Se tiene una utilidad que sirve para este efecto, cuando se genera un archivo, se crea el archivo: **nombre_de_archivo.REP**, cuando se corre una forma, se utiliza este archivo.

Se puede transportar el reporte generado de una computadora a otra por medio de los siguientes pasos:

- Se utiliza la utileria DUMPREP para enviar el reporte a un archivo REX
- Se copia el archivo REX al medio de almacenamiento temporal para traspasarlo a la computadora destino o en su defecto se copia directamente este archivo de una computadora a otra.

En la computadora destino se usan las utilerias LOADREP y GENREP para cargar el reporte en las tablas y generar la rutina ejecutable.

Ahora bien, un reporte en SQL*ReportWriter consta de varias partes, mismas que se mencionan a continuación:

- report Se trata del mismo reporte pero con sus comentarios, márgenes, líneas por página, etc.
- query Se especifica un código en SQL y la vía de acceso para el mismo y el reporte.
- page Refiriéndose al número de páginas, estas no están definidas pues dependen del grueso de la información que de por resultado el query.
- panel Este contiene la última parte de un registro ya que un registro puede contener varios paneles.
- group Es un grupo de campos y sumarios.
- record Registro, en otras palabras es un instancia de un grupo.
- field Este contiene un tipo específico de datos como datos pertenecientes a una tabla o a cálculos procedentes de tablas.
- Summary Contiene lo más relevante de la información de un campo.
- text texto alucivo al título o piecera de un reporte.

Query's

Un reporte puede contener varios query's y en cada uno de ellos debe especificar la parte del reporte que despliega. Los query's pueden estar separados o bien trabajar juntos sin problema siempre y cuando se tenga una relación basada en una específico grupo de columnas.

Si un query A esta hecho para ser parent (padre) de un query B, entonces los siguientes datos seran desplegados:

- cada renglón seleccionado del query A

- únicamente los renglones del query B que tienen valores para el grupo de columnas seleccionadas igual a los valores seleccionados en el query A.

Por ejemplo, si tenemos dos tablas T1 y T2 como las siguientes:

T1			T2		
T1A	T1B	T1C	T2A	T2B	T2C
1	A	D	1	G	J
2	B	E	1	H	K
3	C	F	4	I	L

agregamos los query's

```
QA  select * from T1
QB  select * from T2
```

si QA es padre de QB

ejemplo: Crearemos reportes que desplieguen un lista de los departamentos, nombres de los mismos y su número junto con los números de empleados que trabajan ahí, también se tendrá una lista de empleados, en la cual cada empleado contendrá un departamento, sueldo, días de vacaciones y administrador.

CAPITULO V: HERRAMIENTAS ORACLE

Empezaremos con una descripción de las tablas a utilizar en los reportes que se diseñaran:

Tabla Empleado

No_employado	integer not null,
Nombre	char(40),
sueldo	integer,
vacaciones	integer,
No_depto	integer,
Admon	integer

Tabla Departamento

No_depto	integer not null,
Nombre_depto	char(10),
cubículo	integer

Tabla Otros

No_employado	integer not null,
fecha_viaje	date not null,
propósito	char(72),
comidas	number,
airfare	number,
auto	number,
hotel	number,
otros	number

tomaremos en cuenta los siguientes índices:

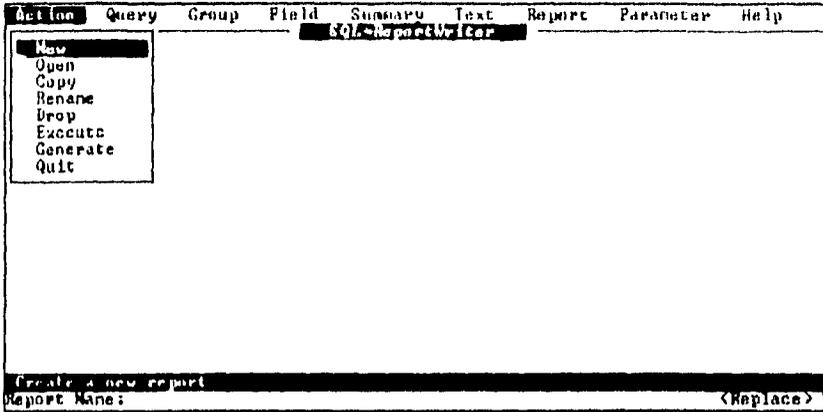
```
depto_index on depto (no_depto) compress
empleado_index on empleado (no_employado) compress
viaje_index on viaje (no_employado, fecha_viaje) compress
```

si no se tienen creadas las tablas el ejemplo no corra como es debido.

Para iniciar una sesión en SQL*ReportWriter se debe de teclear lo siguiente:

```
$ sqlrep cuenta/password
```

donde la cuenta y password son datos que el administrador de la red deberá proporcionar. Al iniciar esta sesión, la pantalla que aparece en su computadora es similar a la siguiente:



si nos equivocamos en la cuenta y/o password, aparecerá una pantalla diferente en donde se nos pedirá introducir los datos antes mencionados. En caso contrario observaremos en la pantalla anterior, en la parte superior, un menú principal y en la parte inferior de la misma estará una barra de estado que informa del reporte que se este utilizando en ese momento. Las opciones que se tienen activadas en el menú principal son las siguientes:

- | | |
|---------|---|
| ACTION | dentro de esta opción se puede crear, abrir, copiar, renombrar, borrar o ejecutar un reporte, también se puede generar un reporte proveniente de algún reporte tipo REX, o en su defecto salir de sesión. |
| QUERY | Graba el programa proveniente de SQL especificando la fuente de información y manteniendo las relaciones entre los programas de SQL. |
| GROUP | Define los grupos de reportes y el formato de los mismos siempre y cuando estén disponibles en ese momento. |
| FIELD | Define los campos en el reporte con definiciones de familias de grupo, formatos y calculos. |
| SUMMARY | Define sumarios. |
| TEXT | Define texto. |
| REPORT | Define el tamaño de página, especificaciones de márgen, opciones con valores por default, comentarios, etc. |

- PARAMETER Define los parámetros de los reportes.
- HELP Habilita la utileria de ayuda.

Las opciones con las que cuenta Action son las siguientes:

- New Crea un nuevo reporte.
- Open Abre un reporte ya existente para poder modificarlo si así se requiere.
- Copy Copia un reporte.
- Rename Renombra un reporte.
- Drop Borra un reporte de las tablas.
- Execute Ejecuta un reporte a la vez que genera los archivos *.REP y *.REX.
- Generate Genera los archivos *.REP y *.REX.
- Quit Opción a salir de SQL*ReportWriter.

De este menú seleccionemos **New** para crear un reporte con el nombre "depto". Una vez creado este reporte, pasemos a la opción **Query** del menu principal dentro de la cual podremos especificar la fuente de información a ser desplegada, la pantalla de esta opción es semejante a la siguiente:

Action	Query	Group	Field	Summary	Tab	Report	Parameter	Help															
<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Query Name:</td> <td style="width: 50%;">SELECT Statement</td> <td style="width: 30%;">Query 1 of 1</td> </tr> <tr> <td colspan="3" style="text-align: center;">Parent-Child Relationships</td> </tr> <tr> <td>Parent Query 1:</td> <td>Parent Query 2:</td> <td></td> </tr> <tr> <td>Child Columns</td> <td>Parent 1 Columns</td> <td>Parent 2 Columns</td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </table>									Query Name:	SELECT Statement	Query 1 of 1	Parent-Child Relationships			Parent Query 1:	Parent Query 2:		Child Columns	Parent 1 Columns	Parent 2 Columns			
Query Name:	SELECT Statement	Query 1 of 1																					
Parent-Child Relationships																							
Parent Query 1:	Parent Query 2:																						
Child Columns	Parent 1 Columns	Parent 2 Columns																					
Report Name: depto (10)1000																							

dentro de esta pantalla se pueden adicionar query's, revisar los que ya existen, cambiar sus atributos y si asi se requiere, borrarlos. Como se puede observar, esta pantalla cuenta con diversas opciones que se explicaran a continuación:

Query Name Cada query que se emplee dentro de un reporte deberá tener un nombre que lo indentifique y es en este campo donde se debe de proporcionar dicho nombre, se recomienda que el nombre de cada query empiece con "Q_".

SELECT statement

Se define el código de SQL en esta área, este código puede tener tantas líneas como se requiera, pero en la pantalla se desplegaran bloques de 8 líneas.

Parent Query 1 and 2

Nombre de los query's que estan relacionados.

Child Columns

Columnas del query hijo.

Parent 1 and 2 Columns

Se especifica el nombre de las columnas de la familia de query's a ser comparadas con las columnas de este query.

Debemos de introducir los datos que se piden en esta pantalla, el query debiera de proporcionar los datos de manera que esten ordenados por numero de depto.

Cuando se acabe de definir el reporte, pasaremos a ejecutar el mismo, para esto pasaremos a la opción **Execute** perteneciente a Action del menu principal. La pantalla correspondiente a este paso es la siguiente:

Parameter Values	
Parameter	Value
Destination Type	Screen
File Name / Spool Device	depto.lis
Printer Description File	dftit
Number of Copies	1

Enter the desired value for each parameter.
Report Name: depto <Replace>

en esta pantalla se pueden observar los parámetros y valores de default, en este paso podemos definir el destino de nuestro reporte segun lo requiera la situación. Si elegimos "screen" (pantalla) podremos ver el reporte y modificarlo segun se desee.

Opción Field

Para realizar nuestro reporte de manera más entendible, podemos cambiar los encabezados acortando la longitud de los campos y centrando el encabezado, para hacer esto pasemos a la opción Field del menu principal teniendo una pantalla semejante a esta:

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Field Sections								
	Field Name	Source	Group	Label				
~	NO_DEPTO NOMBRE_DEPTO	NO_DEPTO NOMBRE_DEPTO	G_Q1 G_Q1	No Depto Nombre Depto				
v								>

Enter a name for the field
Report Name: depto <Replace>

esta opción es usada para adicionar campos, revisar los atributos de los mismos y si es necesario, borrar el campo. Se tiene un orden por default el cual se

Lines Before	Es el numero de líneas a insertar despues del campo, un blanco indica cero líneas.
Spaces Before	Es el número de espacios a insertar despues del campo.

para las siguientes columnas se tiene otra pantalla cuyos datos desplegados en ella son los siguientes:

Aling	Es la alineacion del campo, para valores alfanuméricos y campos fecha se tiene alineación a la izq., para campos numéricos es a la der., por default el valor es un blanco, si se quiere desplegar los datos con este formato basta con marcar este campo con una X.
-------	--

Skip	Si no se quiere desplegar este campo, se marca con una X y el valor del campo es tomado en cuenta para el reporte pero no es desplegado.
Repeat	Se repite el valor en cada pantalla que despliegue el reporte, de no querer esto se marca con una X. campos calculados

Existen otro tipo de campos cuyo valor se necesita calcular en base a datos que se extraen de alguna columna en especial o que son el resultado de otros calculos realizados con anterioridad, los campos calculados son usados para desplegar sumas, máximos, mínimos, promedios, numero de registros contados, porcentajes de un grupo en específico de columnas, etc.

Las funciones disponibles para estos campos son: sum, min, max, count, avg, %total, first, last, r_sum, r_min, r_count, r_avg y r_%total.

Formatos de Despliegue

Es esta sección se proporcionan los formatos disponibles para desplegar datos en un reporte.

9	Despliega un numero en este lugar.
Z	Despliega un numero en este lugar evitando los ceros sobrantes, en cuyo caso pone un espacio en blanco.
,	Despliega una coma en este lugar.
-	Si así se requiere, se despliega este signo con un numero es este es negativo.
+	Si el numero es positivo, se despliega este signo.
\$	Despliega un signo flotante de dolar.
()	Despliega los numeros negativos dentro del paréntesis.
.	Pone un punto decimal en este lugar.
B	Inserta un espacio en este lugar.

b

*Z Reemplaza un blanco por un asterisco.

"str" Inserta la cadena str

a continuacion se ilustran algunos ejemplos para explicar lo anterior.

Numero	Formato	Resultado
3	9	3
	99	03
	Z9	3
	-9	3
	+9	+3
	\$9	\$3
	9.99	3.00
	*Z*Z9	**3
-7124	9	.
	9999	7124
	(99999)	(07124)
	+9,999	-7,124
	-9999CR	-7124CR
	-99"hola"99	-71hola24

si se quiere, se puede modificar el reporte y observar que sucede.

las opciones que restan son:

- Print Group Donde se visualizara o imprimira el sumario cuando este tenga datos.
- Reset Group Donde se visualizara o imprimira el sumario cuando este no tenga datos.

La opción Text

Lo que haremos ahora es poner un encabezado de página y un título a la línea del sumario que tenemos en el reporte, para esto, seleccionemos la opción **Text** y tendremos la siguiente pantalla

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Text Settings								
Object: PAGE			Type: Header			Status: Default		
Relative Position: Lines Before: -- Spaces Before: Width:			Repeat On Page Overflow: Justification: Left Frequency:					
Panel Number:						Panels Defined: 0		
Text								
Choose the position of this text in relation to the previous text								
<input type="checkbox"/> Above <input type="checkbox"/> Next <input type="checkbox"/> After <input type="checkbox"/> Below <input type="checkbox"/> Normal <input type="checkbox"/> Chapter								

esta pantalla tiene al igual que las anteriores, opciones que ayudan al desarrollador a realizar su labor, las opciones son las siguientes:

- Objet Indica con que será asociado el reporte.
- Type Se refiere al tipo de texto que sera desplegado en el reporte, ya sea un encabezado de página, un título, etc.
- Status Indica si el texto ha sido adicionado a el objeto en cuestión. El texto unicamente es desplegado.
- Relative Position Es la posición del texto en relación con un grupo o página previa.

Lines Before	Es el número de líneas que se insertan antes del texto.
Spaces Before	Número de espacio que se insertan antes del texto.
Repeat On Page Overflow	Indica si el texto sera desplegado otra vez en caso de que los datos abarquen más de una hoja.
Justification	Se refiere a la justificación que se le da al texto:Centrado, Izquierda, Derecha.
Frecuency	Se refiere a que si los encabezados de columna sean desplegados por reporte o por página.
Panel Number	Es el número de panel que se tiene en ese momento.
Panels Defined	Es es número de paneles definidos en el reporte.

por ejemplo, podemos modificar el reporte actual, cambiemos en el campo Object hasta seleccionar Page y en la opción Type hasta tener Header. Con estos datos, usando el tabulador pasemos a la ventana principal y escribamos el título para el encabezado. Podemos repetir los pasos anteriores para los campos que se requiera, despues de efectuar dichos cambios, salve sus cambios aceptandolos para despues ejecutarlos con la opción execute.

Después de ejecutar el reporte, podrá observar en la pantalla la información desplegada por el mismo, si desea puede añadir o quitar espacios y líneas, encabezados de pagina y/o columna, etc. de tal modo que la presentación del réporte sea de su agrado.

La opción Group

En SQL*ReportWriter, cada campo pertenece a un grupo, se puede utilizar esta característica para organizar campos dentro de un grupo al que se le definiran determinadas características según se requiera. Si escogemos ahora esta opción del menú principal se tendrá una pantalla similar a la siguiente:

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Group Selection								
	Group Name	Query		Print Direction	Matrix Group	Page Break		
*	G-Q1	Q1		Down				
u								>
Enter a name for this group								
Report Name: dajto <Replace>								

por medio de esta pantalla se pueden adicionar, revisar y borrar grupos y atributos de los mismos. Los campos que aparecen en la pantalla son los siguientes:

- Group Name Es el nombre del grupo.
- Query Es el query al que debe de estar asociado cada grupo, en este caso, nuestro grupo.
- Print Direction Se refiere al sentido de impresion que tendrá el registro con respecto a un registro anterior.
- Matrix Group Se especifica si el grupo es o no parte de una matriz, una X indica que si.
- Page Break Especifica cuando un registro es desplegado en una nueva página.

como en otros casos, las opciones en la pantalla son muchas como para presentarse solo en una, por eso, dichas opciones se presentan en varias pantallas por lo que la segunda pantalla referente a estas opciones es:

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Group Formatting								
	Group Name	Relative Position	Lines Before	Spaces Before	Spacing		Fields Across	
					Record	Field		
^	G_Q1							
v								
<input type="text"/>								
Enter the number of lines to leave blank before printing this group. Report Name: depto <Replace>								

- Relative Position** Es la posición del grupo con respecto a otro.
- Lines Before** Es el número de líneas que se encontrarán entre un grupo y el actual.
- Spaces Before** Es el número de espacios que tendrán entre un grupo y el actual.
- Record Spacing** Es el número de espacios o líneas que estarán entre un registro y otro.
- Field Spacing** Es el número de espacios o líneas que estarán entre un campo y otro.
- Field Across** Es el número máximo de campos que se desplegaran continuamente en el panel.
- Multi-Panel** Esto permite desplegar campos en un panel distinto.
- Label Position** Se refiere a la posición que tendrá el título o etiqueta de cada campo.

Field, Label Highlight Esta opción se utiliza para definir títulos resplandecientes pertenecientes a campos o columnas de suma importancia, en este caso sólo son válidos algunos tipos de letras que son: underline, reverse, bold y combinaciones de estas.

Ahora, seleccionando la opción group, pasemos a su segunda e introduzcamos un valor para Lines Before, por ejemplo 1; al ejecutar el reporte se observa el cambio en la presentación de la información.

La opción Report

La pantalla perteneciente a esta opción es utilizada para especificar el destino que por default tendrá el archivo así como el formato de página que el reporte tendrá, las opciones que se tienen en esta pantalla son:

Top Margin	Se refiere al número de de líneas que se dejan en la parte superior de la página antes de empezar a desplegar el contenido.
Bottom Margin	Es el número de líneas que se dejan antes de terminar la página.
Left Margin	Son los espacios que se dejan a la izquierda de la página.
Right Margin	Espacios que se dejan a la derecha de la página.
Page Height	Número de líneas que contendrá la página.
Page Width	Número de caracteres que contendrá cada línea del reporte.
Margins	Valores con respecto a los márgenes del reporte.
Comments	Comentarios.
Access List	Son los nombres de usuario que pueden ejecutar el reporte.

La opción Parameter

Action	Query	Group	Field	Summary	Text	Report	Parameter	Help
Parameter Settings								
			Parameter Name	Data Type	Width	Default Value	Label	
^			DESTYPE	CHAR	80	Screen	Destination Type	
			DESNAME	CHAR	80		File Name / Spool Dev	
			DESFORMAT	CHAR	80	df1t	Printer Description F	
			COPIES	NUM	2	1	Number of Copies	
			CURRENCY	CHAR	4		Currency Symbol	
			THOUSANDS	CHAR	1		Thousands Separator	
			DECIMAL	CHAR	1		Decimal Indicator	
v								>
Report Name: depto								<Replace>

La pantalla de esta opción es utilizada para adicionar, modificar y borrar parámetros por lo que las opciones de campos que la componen son específicamente para estas acciones.

- Name Nombre del parámetro.
- Data Type Tipos de datos válidos como Char, Number y Date.
- Width Numero de caracteres que pueden ser introducidos en un parámetro.
- Default Value Es el valor que por default se le da al parámetro.
- Label Es el texto que se despliega y que es usado para identificar al parámetro.
- Input Format Es el formato de entrada que el usuario tendrá disponible con este parámetro.
- Out Format Es el formato con el que el parámetro desplegara la información del reporte.
- Skip Indica si el parámetro es desplegado en la pantalla o no.

Ejemplo:

Crearemos un nuevo reporte que sera una copia del anterior, sólo que le pondremos otro nombre, este nombre puede ser el Ud. quiera, por ejemplo "depto2". Este reporte será diferente del primero; seleccionemos la opción query e introduzcamos lo siguiente:

```
where numero_depto = :departamento
```

seleccionando la opción Text posicione en Page, en lugar del valor Header pongamos:

```
&page_title
```

procederemos a ejecutar el reporte. Despues de observar los cambios que se realizaron en la forma de presentar la información es recomendable borrar este reporte para evitar confusiones y tener presente esta modificación para acciones futuras, para borrar el reporte, de la opción Action del menu principal, elija Drop y borrar el menu que se le asigne, en este caso el reclén creado.

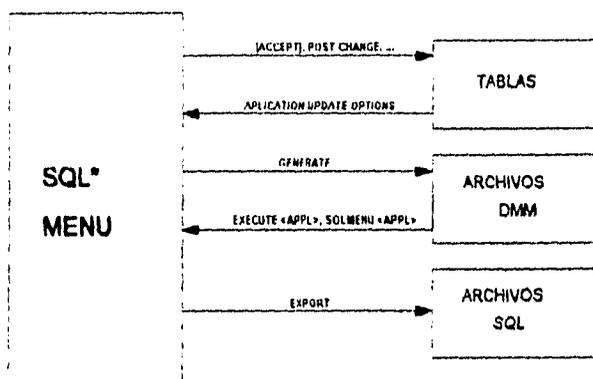
Variables del sistema

SQL*ReportWriter cuenta con una serie de variables de sistema que pueden emplearse dentro del diseño de reportes para facilitar un poco el diseño de los mismos, las variables de sistema con las que se cuenta son las siguientes:

&page	Proporciona el número de página.
&num_pages	Proporciona el número de páginas del reporte.
&date	Fecha actual del sistema.

SQL*Menu

Esta utilidad es para el usuario final una forma de poder acceder a la información ejecutando formas y reportes; para el diseñador, es una utilidad empleada para crear y/o mantener menus que puedan acceder reportes en el sistema. La forma en la que SQL*Menu trabaja es la siguiente:



si analizamos a esta herramienta como un grupo de aplicaciones, veremos que se trata de una forma de acceder a otras aplicaciones de una manera más fácil y comoda.

Analizandola como compilador, contamos con la opción "Generate" que puede crear librerías de archivos *.DMM de nuestra aplicación y hacerlo ejecutable por SQL*Menu. Toda información necesaria es recuperada de la tablas y bases de datos de ORACLE y puesta dentro de los archivos de aplicación DMM. Cuando SQL*Menu corre una aplicación basada en menus, este usa la librería de aplicación de menus.

Las librerías del SQL*Menu no son del todo transportables de una computadora a otra, sin embargo, ORACLE proporciona una opción llamada **Unload** que tiene la habilidad para exportar una aplicación de SQL*Menu en forma de archivo plano y de esta manera transportarlo de una computadora a otra. Si este es el caso, entonces lo que se debe de hacer es lo siguiente:

- a) Seleccionemos Unload de la opción Action.
- b) Introduciremos el nombre de la aplicación y el nombre del archivo en donde queremos que se salve (este nombre debe estar sin extensión, por ejemplo: menu1), demos <enter> o la opción accept. Automáticamente SQL*Menu salva la aplicación en el archivo asignado y le proporciona la extensión "sql".
- c) Traslademos el archivo a la computadora destino.
- d) Una vez salvado el archivo dentro de la computadora destino, entremos en SQL*Plus y ejecutemos desde el prompt de sql lo siguiente: START Nombre del archivo.
- e) Para este momento, se ha generado el archivo de librería que permitira ejecutar la aplicación en el directorio o ruta donde fue grabado.

PRIVILEGIOS

Los privilegios sirven para poder utilizar SQL*Menu, estos privilegios estan divididos en tres niveles que son:

Aplicación de usuario.

A este tipo de usuario se le permite ejecutar las aplicaciones de SQL*Menu a las que sus privilegios le permitan.

Aplicación de diseñador.

Aparte de poseer todos los privilegios del usuario anterior, este usuario tiene la capacidad de poder crear menus de aplicaciones aparte de poder dar mantenimiento a estos y otros menus de aplicaciones.

Administrador de la B.D. de SQL*Menu.

Este usuario tiene la capacidad de los dos anteriores aunada a la de poder otorgar privilegios a usuarios y a cualquier menu de aplicación de SQL*Menu.

Este administrador otorga los privilegios antes mencionados a través de las utilerias de SQL*Menu o bien a través del comando:

GENMENU SYSTEM/clave_de_system -G(privilegio) (usuario)

el privilegio puede ser "e" para aplicación de usuario, "d" para aplicación de diseñador o "a" para aplicación del administrador de la B.D. de SQL*Menu.

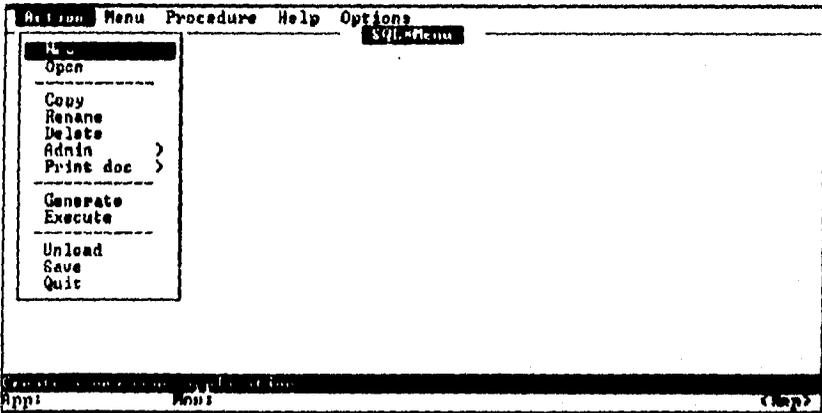
Por otro lado, SQL*Menu implementa el concepto de "Papel", que viene siendo la función o tipo de actividad que realice un empleado, esto se utiliza para definir que aplicaciones pueden ser accesadas por algun empleado.

TRABAJANDO CON SQL*MENU

Para poder trabajar con esta herramienta, es necesario escribir el comando o nombre de archivo adecuado para poder entrar en sesión, generalmente se escribe desde el prompt del sistema operativo:

\$ SQLMENU nombre_usuario/clave_de_acceso <ENTER>

como en las demas herramientas mencionadas, la pantalla inicial sera similar a la siguiente:



como puede observar, la pantalla que tiene a la vista no es muy diferente de las anteriormente, las opciones que se tienen en el menú principal así como la función de cada un de ellas es la siguiente:

Action	Crea, abre, copia, renombra, borra, salva, o ejecuta un menu de aplicación, generando el archivo de librería DMM así como la documentación del contenido del menú de librerías, aparte, puede bajar una aplicación a un archivo plano, otorgar seguridad a los archivos o bien salir de esta aplicación.
Menu	Mantiene las aplicaciones, menus, item's y definiciones de parámetros.
Procedure	Mantiene los procedimientos PL/SQL
Help	Despliega la ayuda en línea, notas acerca de la versión de SQL*Menu y las teclas de función disponibles.
Options	Selecciona la opciones que afectaran el comportamiento de SQL*Menu durante su ejecución o bien durante la sesión de diseño.

si así lo desea, puede desplegar las teclas de función que se tiene disponibles y tomar nota de ello para futuras referencias.

La opción Action

La pantalla, semejante a la anterior, muestra el menú completo perteneciente a esta opción, como puede observar, este menu cuenta con una serie de opciones que le ayudaran mientras se encuentra utilizando SQL*Menu. Las explicación de las opciones disponibles en este menu es la siguiente:

New	Crea un nueva aplicación.
Open	Pone en operación una aplicación ya existente, esta aplicación deba de estar almacenada en la base de datos.
Copy	Copia una aplicación a un archivo con nombre diferente.
Rename	Renombra una aplicación.
Delete	Borra una aplicación.

Admin	Administra y mantiene la seguridad de la aplicación y privilegios de acceso, aparte, permite ejecutar comandos del sistema operativo.
Print Doc	Genera la documentación de la aplicación.
Generate	Genera el código ejecutable (archivo DMM) para la aplicación.
Execute	Corre la aplicación.
Unload	Exporta la aplicación a un archivo plano para poder trasladar la aplicación a otra máquina.
Save	Salva la aplicación en la B.D.
Quit	Salva de SQL*menu.

El funcionamiento de las opciones New, Open, Rename y Delete se sobre entiende, por tal motivo se empezará la explicación de la opción siguiente.

La opción Admin

Al seleccionar esta opción, se visualizará otra pantalla que contendrá las opciones Grant Access, Security y Host, cada una de ella funciona de la siguiente manera:

Grant access	Concede o quita privilegios a usuarios de SQL*Menu.
Security	Mantiene la definición de "Papel" para cada usuario.
Host	Permite ejecutar comandos del sistema operativo.

Por lo que respecta a la definición del papel que cada usuario tendrá, puede observarse una pantalla que aparece al momento de escoger la opción **Security**:

Action Menu Procedure Help Options				
Role Definition				
Role Name	Select Users	Debug Mode	OS Command	Backgr. Menu
DBMS	< * >	[X]	[X]	[X]
DWR	< * >	[]	[]	[]
EC_ADMIN	< * >	[X]	[X]	[X]
EC_CPIEM	< * >	[X]	[X]	[X]
EC_CPIIN	< * >	[X]	[X]	[X]
EC_COMCE	< * >	[X]	[X]	[X]
EC_COMCI	< * >	[X]	[X]	[X]
EC_COMDE	< * >	[X]	[X]	[X]
EC_COMER	< * >	[X]	[X]	[X]
EC_COMPR	< * >	[X]	[X]	[X]
EC_COMPI	< * >	[X]	[X]	[X]
EC_MANCO	< * >	[X]	[X]	[X]
EC_MANCR	< * >	[X]	[X]	[X]

Enter the name of the role:
 App: _____ Menu: _____ (Map)

este tipo de pantalla puede cambiarse con la tecla u opción indicada para ello; como puede darse cuenta, aparecen varios campos en esta pantalla con los siguientes atributos:

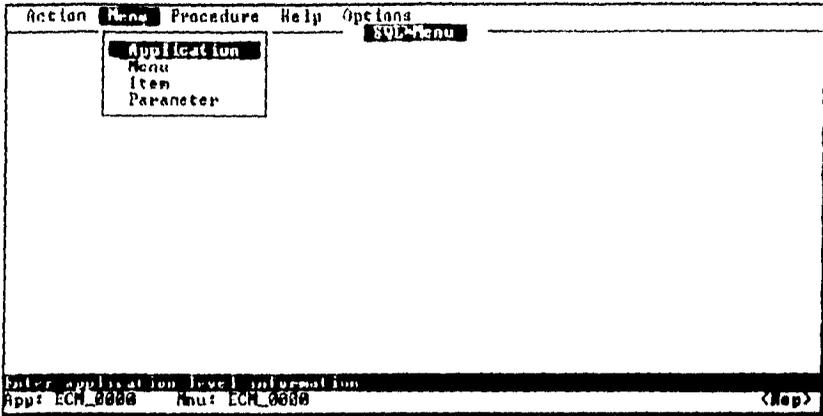
- Role Name Es el nombre del Papel que se otorgará.
- Select Users Es el identificador de usuario ORACLE al cual se le asignara el Papel.
- Debug Mode Indica cuando pueden los usuarios asignados a este papel ejecutar las aplicaciones en modo debug.
- OS Command Indica cuando los usuarios de este papel pueden ejecutar o no comandos del sistema operativo.
- Background Menu Indica cuando los usuarios de este papel pueden acceder al menu de Background en los menus de aplicaciones.
- Comments Descripción del papel.

por ejemplo, para crear un "papel", presione la tecla de insertar e introduzca el nombre del "Papel", ahora pasando al siguiente campo, introduzca el identificador de usuario. Si esto es todo lo que desea introducir deberá presionar la tecla de aceptación y salvar el archivo.

La opción Menu

Empezaremos por definir un nuevo menú, para esto, de la opción **Action** del menú principal, seleccionemos **New** y proporcionemos el nombre de la

opción, "ECM_0000" por ejemplo, después de esto, pasemos a la opción **Menu**, esta opción cuenta con las siguientes opciones:



- Application Conserva información relacionada la aplicación como, el nombre, fecha de creación, autor, número de versión, etc.,
- Menu Conserva la información del menú creado como su nombre, propósito y los títulos a desplegar.
- Item Proporciona información acerca del menu del Item.
- Parameter Proporciona una aplicación de parámetros de sustitución.

Application

En esta opción aparecerá la siguiente pantalla:

```

Action Menu Procedure Help Options
Application Definition
-----
Short Name:      ECM_0000
File Name:      ECM_0000
Creation Date:   05-JUL-93
Creator:        sigac
Version Number:  1
Last Release Date:
Directory:
Identification:  ECM_0000

Short name which will appear on application menu bar
App: ECM_0000  Mnu: ECM_0000  <Key>

```

esta pantalla cuenta con lo siguientes atributos:

Short Name	Es el nombre corto de la aplicación (15 caracteres máximo) y que aparece en la parte inf. izq. de la pantalla, default toma los primeros 15 caracteres del nombre de la aplicación.
File Name	Es el nombre del archivo que SQL*Menu usa para generar la librería.
Creation Date	Es la fecha de la aplicación, por default es la fecha del sistema.
Creator	Es el nombre del creador de la aplicación, por default es el usuario que se tiene para ORACLE.
Version Number	Este campo se utiliza para almacenar el número de veces que se ha modificado el menú, por default es 1.
Last Release Date	Es la última fecha en que la aplicación se libero para su uso.
Directory	Es el directorio en el que se almacena la librería, este debe ser especificado de tal forma que de ser necesario otro usuario pueda utilizar la aplicación. Por default es el directorio actual.

Identification Es la descripción de la aplicación que será desplegada en el menú de aplicación.

Para comprender mejor lo anterior puede presionar la tecla de aceptar lo que tiene y retornar al menú principal, de esta forma al volver a entrar a esta opción, se podrá observar las opciones que por default el sistema le proporciono a su archivo y se así lo desea podrá modificarlas.

La opción Menu

En este caso, entraremos en la opción **menu** del menu principal, las opciones listadas son 4: Application, Menu, Item, Parameter. Escogiendo la segunda opción, la pantalla muestra las opciones Menu Name, Title, Subtitle, Bottom Title y Purpose, cada opción realiza una acción particular relacionada con el menu.

Menu Name	Es el nombre del menu, de preferencia debe ser igual al nombre de la aplicación.
Subtitle	Es el titulo que será desplegado en la parte superior del menú.
Bottom Title	Es el titulo que será desplegado en la parte inferior del menú.
Purpose	Propósito del menú.

Ejemplo: Crearemos el menú principal de la aplicación ECM_0000. Para esto, introduzcamos el nombre del menú y pasemos a los siguientes campos donde pondremos los datos correspondientes a cada uno.

Action Menu Procedure Help Options			
Menu Definition			
Menu Name	Title	Subtitle	Button Title
ECM_0000	SISTEMA DE ENLACE	MENU PRINCIPAL	
ECM_1000	SISTEMA DE ENLACE	SUBGERENCIA DE CO	
ECM_1100	SISTEMA DE ENLACE	PROGRAMA OFICIAL	
ECM_1110	SISTEMA DE ENLACE	REPORTES DEL PROG	
ECM_1200	SISTEMA DE ENLACE	ORDEN DE VENTA	
ECM_1300	SISTEMA DE ENLACE	AUTORIZACION DE C	
ECM_1310	SISTEMA DE ENLACE	REPORTES DE AUTOR	
ECM_1311	SISTEMA DE ENLACE	COMT. DE REPORTES	
ECM_1400	SISTEMA DE ENLACE	RECLAMOS Y DEMORA	
ECM_1410	SISTEMA DE ENLACE	REPORTES DE RECIA	
ECM_1411	SISTEMA DE ENLACE	COMT. DE REPORTES	
ECM_1500	SISTEMA DE ENLACE	INTERORGANISMOS	
ECM_1510	SISTEMA DE ENLACE	REPORTES INTERORG	

Enter the name of the menu
App: ECM_0000 Menu: ECM_0000 <Rep>

Presionando la tecla de aceptar habremos terminado.

La opción Item

La pantalla de esta opción, presenta los siguientes parámetros:

Action Menu Procedure Help Options					
Item Definition					
Item Number	Command Type	Grant Rols	Item Text	Short Item Name	Displ Nn Pr
1	1	< = >	SUBGERENCIA DE CO	RECIBIDO	[]
2	1	< = >	SUBGERENCIA DE CE	CERTIFICACION	[]
3	1	< = >	SUBGERENCIA DE CO	C. FINANCIERO	[]
4	1	< = >	SUBGERENCIA DE MA	MANEJO	[]
5	1	< = >	RESALDOS	RESALDOS	[]
6	7	< = >	SALIR DEL SISTEMA	SALIR	[]

Enter the abbreviated name of the menu item
App: ECM_0000 Menu: ECM_0000 <Rep>

Item Number Cada item en el menú

Command Type Es el tipo de comando a se usado por el usuario al seleccionar esta opción.

Grant Role Access Es el nombre de las actividades o papeles que tengan algunos usuarios y que sólo con esta actividad definida podrán acceder al ítem, de otra forma, todos los papeles que tengan acceso al menú, podrán acceder al ítem.

Item Text Es la descripción del ítem.

Short Item Name Es el nombre del menú del ítem que será desplegado en la pantalla.

Display Without Privilege

Se refiere al momento en que será desplegado este menú cuando un usuario así lo requiera.

Command Line Es el comando o macro a ser ejecutado cuando el usuario seleccione el ítem, máximo 240 caracteres.

Help Text Es el texto de ayuda que será desplegado en caso de que el usuario presione la tecla de ayuda.

Los tipos de comandos que se utilizan al ejecutarse una opción, se muestran en la siguiente lista.

Tipo	Descripción
1	Despliega un menú, la línea del comando deberá contener el nombre del menú.
2	Ejecuta un comando del sistema operativo. Se deberá especificar el comando así como todos los argumentos necesarios.
3	Ejecuta un comando del sistema operativo y hace pausa después de la ejecución.
4	Corre una forma de SQL*Forms.
5	Entra a SQL*Plus.
6	Ejecuta una macro.
7	Ejecuta comandos de PL/SQL.

El Menu Background

Este es un menú muy especial, ya que simplifica la ejecución de algunas funciones. Si este menú existe, los usuarios pueden desplegar de cualquier otro menu presionando una tecla. Un item en el menu Background permite seleccionarlo con una tecla de función, este menú tiene el nombre con la extensión BGM, a excepción de esto último, el menú se crea exactamente igual que un menú sencillo.

Parámetros de Sustitución

Un parámetro de sustitución es una variable que se puede adicionar a una línea de comando de un item para hacerlo más flexible. Si un parámetro de sustitución es usado en una línea de comando que no tenga el tiempo de ejecución el usuario tendrá que hacerlo.

En adición a otras características, el diseñador puede especificar qué parámetro estará ligado a un menú, cuando este menú es especificado toma el nombre de "parámetro de menú".

Cuando se llama un menú de pantalla completa, que contenga parámetros de sustitución, una forma para estos parámetros es desplegada primero, aquí, el usuario deberá introducir los valores de los parámetros de menú. Con los valores introducidos, el usuario podrá ver el menú deseado.

Los parámetros de sustitución se distinguen porque su nombre consta de dos letras, cuando se usa una línea de comando, el nombre deberá de llevar antepuesto un ampersand (&) y cuando se utilice un comando de PL/SQL deberá de llevar dos puntos (:).

SQL*Menu reserva cinco parámetros de sustitución que pueden asociarse a una aplicación.

UN	Es el identificador de Oracle que tiene la persona que usa la aplicación.
PW	Es el password de Oracle que tiene la persona que usa la aplicación.
SO	Es el valor actual del item.
TT	Es el tipo de terminar que el usuario especifica al entrar a SQL*Menu.

AD Es el directorio en donde se tienen los archivos de librería para la aplicación.

Al momento de acceder a esta opción, la pantalla disponible es la siguiente:

Action Menu Procedure Help Options						
Parameter Definition						
Parameter	Size	Prompt	Select Menus	Echo	Fixed Length	Reqd
█			< * >	[X]	[]	[]
Enter the parameter text string (2 character.)						>
App: ECA_0000 Menu: ECA_0000						<Msg>

las opciones de que dispone esta pantalla se explica a continuación:

- Parameter** Es el nombre del parámetro de sustitución (dos letras).
- Size** Es el número máximo de caracteres que este valor puede tener.
- Prompt** Texto que estara a disposición del usuario.
- Select Menus** Son los menus a los que el parámetro de sustitución estara asociado.
- Echo** Especifica cuando se deslegara el nombre de usuario en la pantalla.
- Fixed Length** Esto indica que el número de caracteres introducidos por el usuario debera se igual al máximo número de caracteres permitidos.

Required	Indica que el valor tiene que ser introducido.
Upper Case	Todo valor introducido es con letras mayúsculas.
Default	Indica que el valor es asignado por default al parámetro.
Hint	Es el mensaje de ayuda que se despliega al momento de introducir el parámetro de campo.

Macros

La línea de comando de una macro es de la siguiente forma:

MACRO; [MACRO; ...]

las macros disponibles son las siguientes:

APLMENU	Despliega el menu de aplicación.
APLPARM	Despliega la forma del parámetro de aplicación.
ASSIGN	Asigna un valor a un parámetro de sustitución.
BGMx	Ejecuta
CHRMODE	Inserta/Reemplaza.
CLRFLD	Limpia el campo.
DEBUG	Activa modo de debug.
DELCHR	Borra el caracter a la izq. del cursor.
DISP	Activa la pantalla durante la macro.
DOWN	Se posiciona en el siguiente menu item.
EXIT	Salte de SQL*Menu.
HELP	Despliega el texto de ayuda para el menu item actual.

LEFT	Mueve el cursor un caracter a la izq.
MAINMENU	Despliega el menu principal de la aplicación.
MENUPARM	Despliega el menu actual para la forma del menu de parametros.
NEWAPL	Sale de otras aplicaciones.
NEWUSER	Cambia de usuario Oracle así como de password.
NODISP	Desactiva la pantalla durante la macro.
NXTFLD	Mueve el cursor al siguiente campo.
OSCMD	Ejecuta comandos del sistema operativo.
OSCMD1	Ejecuta un comando del sistema operativo.
PRVFLD	Mueve el cursor un campo antes.
PRVMENU	Despliega el menú anterior.
REDISP	Vuelve a desplegar la pantalla actual.
RIGHT	Mueve el cursor un caracter a la derecha.
SHOWBGM	Despliega el menu de Background.
SHOWKEYS	Despliega la lista de las teclas de función.
SUSPEND	Detiene la ejecución de la macro hasta que el usuario presione alguna tecla que indique el menú anterior o que la Inserción de datos termina.
TRMNATE	Termina la entrada.
UP	Mueve el cursor al menu item previo.
WHERE	Despliega el directorio donde el usuario tenga el menu actual.

Existen macros a las que se deben de añadir argumentos como ASSIGN, NEWAPL y otras a las que puede añadirseles un argumento como EXIT, NEWUSER, OSCMD y OSCMD1.

ASSIGN parámetro = valor;

parámetro Es el parámetro de sustitución al que se le asignara un valor.

valor Este valor que se le asigna al parámetro de sustitución.

EXIT texto;

texto Es el texto a desplegar cuando se salga de SQL*Menu.

NEWAPL appl;

appl Es el nombre de la aplicación de la que se saldrá.

NEWUSER ID/PW

ID Es el identificador de Oracle que se usará.

PW Es el password de Oracle que se usará.

OSCMD [[[TYPE = tipo]] cmd]

tipo Es el tipo de comando que se asignara y puede tener los valores: 2, 3, 4 y 5.

cmd Es un comando del sistema operativo.

OSCMD1 [[[TYPE = tipo]] cmd]

tipo Es el tipo de comando que se asignara y puede tener los valores: 2, 3, 4 y 5.

cmd Es un comando del sistema operativo.

PL/SQL en la líneas de comando del Menu Item

En las líneas de comando se pueden tener bloques de PL/SQL, la sintáxis de estos bloques siguen el estándar de PL/SQL, sin embargo se pueden mencionar cuatro puntos extra que permitan aprovechar un poco más esta capacidad de SQL*Menu:

- 1.- Se pueden definir procedimientos de PL/SQL y almacenarlos como un paquete.
- 2.- Pueden usarse variables de sustitución.
- 3.- Se pueden crear y referenciar variables (comunmente llamadas variables globales) que contengan un valor y que estén disponibles durante la sesión de SQL*Menu.
- 4.- Podemos llamar procedimientos previamente definidos.

Procedimientos Empaquetados

SQL*Menu tiene la capacidad de utilizar paquetes de instrucciones de PL/SQL junto con otros comandos del menu item o en forma aislada. La mayoría de las macros de SQL/Menu tienen por lo menos un procedimiento empaquetado, estos procedimientos son usados para:

- a) Controlar el cursor dentro de una aplicación o menu.
- b) Permite a los usuarios ejecutar funciones propias de sus aplicaciones.
- c) Ayuda a evitar entradas de datos repetidas.

Cuando un menu es utilizado, los procedimientos empaquetados se habilitan automáticamente, los procedimientos empaquetados permitidos son:

APPLICATION_MENU	Despliega el menu de aplicación.
APPLICATION_PARAMETER	Despliega la forma del parámetro de aplicación.
BACKGROUD MENU	Ejecuta el menu que se tiene de fondo.
DEBUG_MODE	Cambia a modo debug.

DISABLE_ITEM	Origina un menu item que se desplegara que será inaccesible para los usuarios.
ENABLE_ITEM	Origina un menu item que será accesible para los usuarios.
EXIT_MENU	Sale de SQL*Menu.
ITEM_ENABLED	Retorna un valor verdadero si el item esta habilitado, es otros casos retorna falso.
MAIN_MENU	Despliega el menú principal de la aplicación.
MENU_CLEAR_FIELD	Limpia el campo.
MENU_HELP	Despliega el texto de ayuda para el menu item actual.
MENU_MESSAGE	Despliega un mensaje definido para el usuario.
MENU_NEXT_FIELD	Mueve el cursor al siguiente campo.
MENU_PARAMETER	Despliega la forma para el parámetro del menu actual.
MENU_PREVIOUS_FIELD	Mueve el cursor al siguiente campo.
MENU_REDISPLAY	Vuelve a presentar en pantalla el menu actual.
MENU_SHOW_KEYS	Despliega la lista de las teclas de función.
NEW_APPLICATION	Sale de la aplicación.
NEW_USER	Cambia el identificador y password de Oracle actual.
NEXT MENU ITEM	Resalta el siguiente menu item.
OS_COMMAND	Ejecuta comandos del sist. operativo.

OS_COMMAND1	Ejecuta un comando del sist. operativo.
PREVIOUS_MENU	Despliega el menu anterior.
PREVIOUS_MENU_ITEM	Resalta el menu item anterior.
QUERY_PARAMETER	Es el query para el valor de los parámetros de sustitución.
SHOW_BACKGROUND_MENU	Despliega el menu background.
TERMINATE	Termina la entrada.
WHERE_DISPLAY	Despliega el directorio del usuario en donde se tiene el menu actual.

Existen argumentos que deben ser adicionados a DISABLE_ITEM, ENABLE_ITEM, ITEM_ENABLED, MENU_MESSAGE, NEW_APPLICATION, NEW_USER, OS_COMMAND, OS_COMMAND1, QUERY_PARAMETER y EXIT_MENU. La sintaxis de estos argumentos es similar a la siguiente:

DISABLE_ITEM ('nombre_menu','nombre_item')

nombre_menu Nombre del menú en el que el item será inhabilitado.

nombre_item Es el nombre corto del item que será inhabilitado.

ENABLE_ITEM ('nombre_menu','nombre_item')

nombre_menu Nombre del menú en el que el item será habilitado.

nombre_item Es el nombre corto del item que será habilitado.

EXIT_MENU ('message')

message Es el message a desplegar.

ITEM_ENABLED ('nombre_menu','nombre_item')

nombre_menu Nombre del menú en el que el ítem será deshabilitado.

nombre_item Es el nombre corto del ítem que será deshabilitado.

MENU_MESSAGE ('message')

message Es el message a desplegar.

NEW_APPLICATION ('nombre_aplicación')

nombre_aplicación Es el nombre de la aplicación.

NEW_USER ('id/pw')

id Es el identificador de usuario Oracle.

pw Es el password de ese usuario.

OS_COMMAND (' (TYPE=tipo) cmd')

tipo Es el tipo de comando que se asignara y puede tener los valores: 2, 3, 4 y 5.

cmd Es un comando del sistema operativo.

OS_COMMAND1 (' (TYPE=tipo) cmd')

tipo Es el tipo de comando que se asignara y puede tener los valores: 2, 3, 4 y 5.

cmd Es un comando del sistema operativo.

QUERY_PARAMETER ('parámetro...')

parametro Es un parámetro de sustitución de SQL*Menu.

Funciones de empaquetado

SQL*Menu tiene funciones de empaquetado que son referenciadas a las líneas de comando de los menu item y a los procedimientos de PL/SQL. Las funciones de empaquetado son usadas para registrar los resultados de la ejecución de los procedimientos de empaquetado, tambien son usadas para cambiar el curso de una aplicación. Las funciones de empaquetado disponibles para ejecutar independientemente un menu son:

MENU_SUCCESS	ERROR_TYPE
MENU_FAILURE	ERROR_CODE
MENU_FATAL	ERROR_TEXT

Cuando un menu de aplicación es llamado desde SQL*Forms, las siguientes funciones de emapquetado se ponen a disposición:

```
FORM_SUCCESS
FORM_FAILURE
FORM_FATAL
```

ejemplo:

Crearemos una opción en el menu de background que permita al usuario moverse al menu principal, para esto usaremos código de PL/SQL.

```
MAIN_MENU;
IF MENU_SUCCESS THEN MENU_MESSAGE ('Realizado');
ELSE MENU_MESSAGE ('Falla al ejecutar');
```

Usando parámetros de sustitución en PL/SQL

Los parámetros de sustitución estan disponibles en PL/SQL, para usarlos dentro de un bloque es necesario utilizar el procedimiento QUERY_PARAMETER dentro del bloque. Cuando el usuario cambia un menu item, el procedimiento anterior se ejecuta, la forma de parámetro es desplegada y el usuario podrá en esos momento introducir los parámetros especificados. En el siguiente ejemplo se muestra cómo incluir parámetros de sustitución en PL/SQL.

```
QUERY_PARAMETER ('&f1 &f2');
OS_COMMAND ('(TYPE=3) copy ' || :f1 || ' ' || :f2);
```

Variables Globales

Las variables globales son variables que conservan el valor asignado durante toda la sesión de SQL*Menu. Una variable global tiene el formato:

```
GLOBAL.nombre_variable
```

donde nombre_variable viene siendo el nombre que se le asigna a la variable global. Como ejemplo, retomaremos el código anterior en donde se copia un archivo existente a otro nuevo, sin embargo aquí el nombre del archivo nuevo será salvado en la variable global, después crearemos un nuevo background menu item para que el usuario pueda editar el nuevo archivo sin introducir el nombre. Después del comando OS_COMMAND incluiremos:

```
:GLOBAL.nombre_archivo := :f2;
```

para crear el item para el menu de background utilizaremos el siguiente código:

```
OS_COMMAND ('(TYPE=3) write sys$output "Editando archivo: " ||
:GLOBAL.nombre_archivo || "'");
OS_COMMAND ('(TYPE=3) edit ' || :GLOBAL.nombre_archivo);
```

ahora salve, genere y ejecute la aplicación y observe los resultados.

La opción Procedure

Un procedimiento es un grupo de comandos de PL/SQL que son llamados por los menu items. La sintaxis de un procedimiento es ligeramente diferente del que tiene el estándar de PL/SQL:

```
PROCEDURE procedure_defn
[declaration ...]
BEGIN
    statement ...
[EXCEPTION exception_handler ...]
END;
```

donde:

procedure_defn	Es la definición del procedimiento.
declaracion	Son declaraciones válidas.
statement	Declaraciones válidas de PL/SQL.
exception_handler	Un manejador de excepciones válido para PL/SQL.

ejemplo:

```

PROCEDURE checa_salario_depto
  IS mayor_salario CONSTANT NUMBER(6):= 2000;
  salario_depto odemo.emp.salary%TYPE;
  msg_text odemo.temp.message%TYPE;
BEGIN
  QUERY_PARAMETER('&DT');
  SELECT SUM (sueldo)
    INTO salario_depto
    FROM odemo.emp
    WHERE no_depto = :dt;
  IF salario_depto > mayor_salario THEN
    msg_text := 'Salario grande' || to_char(mayor_salario);
  ELSE
    msg_text := 'Salario bajo' || to_char(mayor_salario);
  END IF;
  INSERT INTO odemo.temp (no_depto, message)
    VALUES (:dt, msg_text);
  COMMIT;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      msg_text := 'Depto No: ' || :dt || ' No existe';
      menu8_message(msg_text);
    WHEN OTHERS THEN
      msg_text := to_char(sqlcode) || sqlerrm;
      menu_message(msg_text);
END;
```

La declaración PROCEDURE

La sintaxis de esta declaración es la siguiente:

```
PROCEDURE procedimiento [(nombre_variable [modo] type[:=valor],...)] IS
```

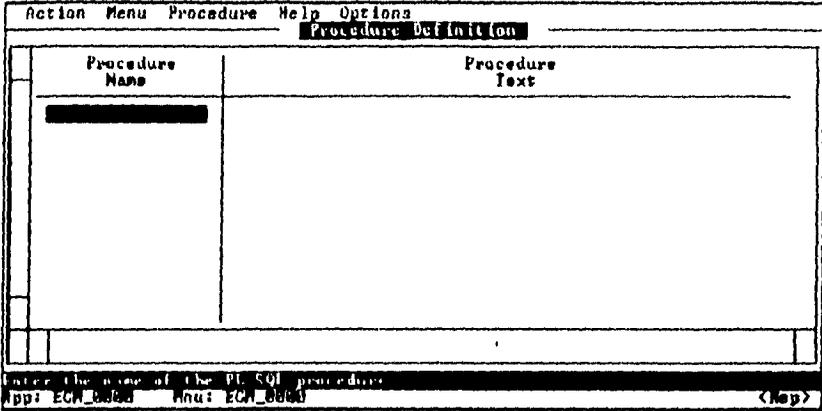
donde:

procedimiento	Es el nombre del procedimiento.
nombre_variable	Es el nombre de la variable que servirá para traspasar el valor.
modo	IN OUT IN OUT

type BOOLEAN | CHAR | NUMBER | DATE

valor Un valor para almacenar en la variable.

cuando seleccionamos la opción Procedure del menu principal, aparecerá la siguiente pantalla:



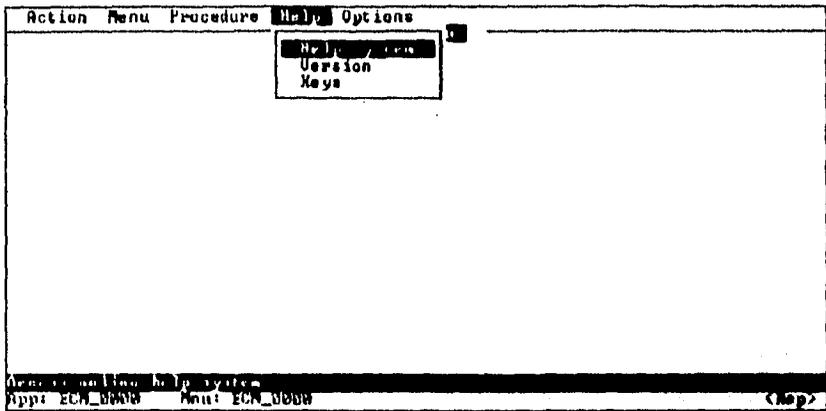
los atributo que contiene esta pantalla son los siguientes:

- procedure name Es un nombre único para el procedimiento, este nombre debe ser igual al nombre del procedimiento definido en el atributo de texto.
- Procedure Text Los comandos de PL/SQL que conforman a este procedimiento.

La opción Help

Esta opción se ocupa para llamar al texto de ayuda que tiene integrado SQL*Menu para la sesión actual. Para desplegar la ayuda de un objeto en particular coloque el cursor en el objeto en cuestión y presione la tecla de ayuda.

Para desplegar la tabla de contenido del sistema de ayuda en línea despliegue el menú de ayuda y presione la letra C. Si oprime la tecla I visualizara el índice. Ahora bien, al momento de seleccionar la opción de ayuda, la pantalla será semejante a la siguiente:



las opciones que aparecen son:

- | | |
|-------------|---|
| Help System | Despliega información acerca de la ayuda en línea de SQL*Menu y de cómo utilizarla. |
| Version | Despliega el número de versión que esta utilizando. |
| Keys | Se visualiza una lista de teclas de función similar a la que se presenta si se oprime la tecla Show Keys. |

La opción Options

Las opciones son usadas para controlar el comportamiento de SQL*Menu y sus aplicaciones, durante la sesión de diseño o durante la ejecución de estas. Para activar estas opciones durante su sesión, seleccione la opción Options del menú principal.

esta pantalla tiene las siguientes opciones:

- | | |
|--------------------------------|---|
| Use Forms as Default | Utiliza por default alguna forma para desplegar la información proveniente de las tablas. |
| Show List of Values | Automáticamente despliega una lista de valores que puede utilizar para definir el campo que este introduciendo en esos momentos. |
| Generate before Executing Menu | Genera el archivo de librería DMM antes de ejecutar la aplicación. |
| Supress Hints | Suprime los mensajes de ayuda que normalmente se despliegan. |
| Disable PL/SQL Compilation | Compila bloques de PL/SQL únicamente cuando el menú de aplicación sea generado. |
| Auto Restrict List Of Values | Restringe la lista de valores a aquellos que sean igual al tipo de campo que se este utilizando, si el campo no tiene valor, entonces se deslignan todos lo valores válidos para este caso. |
| Show Detailed Working Messages | Son los comandos de PL/SQL que desplegaran detalladamente mensajes durante el proceso. |
| Menu Display Style | Controla el tipo de menú que se despliega durante la aplicación. |
| Generate Before Executing Menu | Genera el archivo de librería DMM antes de ejecutar la aplicación. Cuando termina, se |

debe de oprimir la opción Execute y la aplicación se activara.

Supress Hints

Suprime los mensajes de ayuda que normalmente se despliegan.

CONCLUSIONES

Como se ha observado, SQL es un lenguaje de programación que posee características como las de realizar operaciones con los datos directamente, sin tener que utilizar algún tipo de interfaz; proporcionar seguridad al permitir alterar la información a nivel usuario o a nivel B.D.; estructurar de mejor manera la información almacenada, etc. dichas características permiten explotar de manera sencilla la información de una Base de Datos en un ambiente Cliente - Servidor.

La forma en que opera SQL demuestra la aplicación práctica del modelo Entidad - Relación y por ser un lenguaje no procedural permite que el usuario solicite a la computadora los datos que desea ver y no se involucra en el cómo la computadora los va a obtener. Las consultas a la B.D. se basan en comandos propios de SQL, mismos que son pocos y fáciles de utilizar, lo que permite ordenar consultas sumamente complejas de manera sencilla y ordenada.

Este trabajo consta de 5 capítulos que abarcan desde conceptos básicos de bases de datos y redes de computadoras hasta una referencia completa de los comandos y sintaxis de los mismos, también proporciona una explicación paso a paso del cómo funcionan cada una de las herramientas que comúnmente se utilizan dentro del ambiente de trabajo. Por lo anterior, se puede decir que dicho trabajo cumple con su objetivo, que es el de introducir al lector en uno de los lenguajes que por su potencia y sencillez facilita el manejo de grandes volúmenes de información, solo espero que si alguna vez alguien lee con fines de consulta este escrito, sepa aprovechar su contenido, ya que la información aquí recopilada no solo se tomó de algunos libros, sino también de experiencias laborales que se tuvieron mientras se desarrollaban sistemas con este lenguaje en distintas partes de la República Mexicana y que actualmente se encuentran trabajando en forma eficaz a nivel nacional como lo es el Sistema de Transporte Marino realizado para la Región Marina de Petróleos Mexicanos el cual fue el primer sistema implantado a nivel nacional en el que participe como desarrollador y del cual me siento muy orgulloso.