TEMA 7: ELABORACIÓN DEL DISEÑO FÍSICO (PARTE 1)

OBJETIVOS:

- **Definir las estructuras físicas de almacenamiento**
- **Crear tablas**
- **Seleccionar los tipos de datos adecuados**
- **Definir los campos clave en las tablas**
- **Implantar todas las restricciones reflejadas en el diseño lógico**
- **Verificar mediante un conjunto de datos de prueba que la implementación se ajusta al modelo**
- **Utilizar asistentes y herramientas gráficas.**
- **Utilizar el lenguaje de definición de datos**

GESTIÓN DE BASE DE DATOS

IES PUERTO DE LA CRUZ

INTRODUCCIÓN:

En un sistema de gestión de bases de datos las operaciones sobre los datos se realizan a través de consultas formuladas con un lenguaje específico declarativo. SQL se considera el lenguaje estándar de bases de datos realcional, es una abreviatura de *structured query language (Lenguaje de Consulta Estructurado).* Pero SQL es algo más que un lenguaje de consulta como su propio indica, puesto que no solo ofrece funciones de recuperación, sino también una diversidad de operaciones de actualización y manipulación de información, así como de definición de los datos. La versión original de SQL fue desarrollada por IBM (San José Research Laboratory) a mediados de los años setenta. A partir de entonces, se han derivado numerosas versiones hasta que en 1986, el American National Standards Institute (ANSI) publicó un SQL estándar.

Breve Historia.

La historia de SQL (que se pronuncia deletreando en inglés las letras que lo componen, es decir "ese-cu-ele" y no "siquel" como se oye a menudo) empieza en 1974 con la definición, por parte de Donald Chamberlin y de otras personas que trabajaban en los laboratorios de investigación de IBM, de un lenguaje para la especificación de las características de las bases de datos que adoptaban el modelo relacional.

Este lenguaje se llamaba SEQUEL (Structured English Query Language) y se implementó en un prototipo llamado SEQUEL-XRM entre 1974 y 1975. Las experimentaciones con ese prototipo condujeron, entre 1976 y 1977, a una revisión del lenguaje (SEQUEL/2), que a partir de ese momento cambió de nombre por motivos legales, convirtiéndose en SQL. El prototipo (System R), basado en este lenguaje, se adoptó y utilizó internamente en IBM y lo adoptaron algunos de sus clientes elegidos.

Gracias al éxito de este sistema, que no estaba todavía comercializado, también otras compañías empezaron a desarrollar sus productos relacionales basados en SQL.
A partir de 1981, IBM comenzó a entregar sus productos relacionales y en 1983 empezó a vender DB2.

En el curso de los años ochenta, numerosas compañías (por ejemplo Oracle y Sybase, sólo por citar algunos) comercializaron productos basados en SQL, que se convierte en el estándar industrial de hecho por lo que respecta a las bases de datos relacionales.

En 1986, el ANSI adoptó SQL (sustancialmente adoptó el dialecto SQL de IBM) como estándar para los lenguajes relacionales y en 1987 se transfomó en estándar ISO.
Esta versión del estándar va con el nombre de SQL/86.

En los años siguientes, éste ha sufrido diversas revisiones que han conducido primero a la versión SQL/89 y, posteriormente, a la actual SQL/92.

El hecho de tener un estándar definido por un lenguaje para bases de datos relacionales abre potencialmente el camino a la intercomunicabilidad entre todos los productos que se basan en él.

Desde el punto de vista práctico, por desgracia las cosas fueron de otro modo. Efectivamente, en general cada productor adopta e implementa en la propia base de datos sólo el corazón del lenguaje SQL (el así llamado Entry level o al máximo el Intermediate level), extendiéndolo de manera individual según la propia visión que cada cual tenga del mundo de las bases de datos.

Actualmente, está en marcha un proceso de revisión del lenguaje por parte de los comités ANSI e ISO, que debería terminar en la definición de lo que en este momento se conoce como SQL3.

Las características principales de esta nueva encarnación de SQL deberían ser su transformación en un lenguaje stand-alone (mientras ahora se usa como lenguaje hospedado en otros lenguajes) y la introducción de nuevos tipos de datos más complejos que permitan, por ejemplo, el tratamiento de datos multimediales.

El lenguaje SQL consta de tres lenguajes específicos: DDL, DML Y DCL.

1. El Lenguaje de Definición de Datos. Una vez diseñado el esquema de la base de datos, hemos de describirlo mediante un conjunto de instrucciones. Esto se realiza mediante un lenguaje específico, denominado Lenguaje de Definición de Datos (DDL - Data Definition Language) y que, como cualquier lenguaje de alto nivel, necesitará un traductor para generar el código objeto a partir del código fuente. El DDL proporciona órdenes para definir, eliminar y modificar tablas, así como para crear índices y vistas. Admite las siguientes sentencias de definición: CREATE, DROP y ALTER.

2. El Lenguaje de Manipulación de Datos. A través del Lenguaje de Manipulación de Datos (DML - Data Management Language) se permite el

acceso a los datos almacenados en una base de datos. El DML puede ser utilizado de dos formas diferentes. Por una parte, se incluyen sentencias DML en programas escritos en lenguajes de alto nivel (COBOL, Pascal, C, etc.)' por lo que al primero se le llama lenguaje huésped y al segundo, lenguaje anfitrión. Inmediatamente se plantea un problema: el lenguaje anfitrión no reconoce las sentencias del lenguaje huésped, por lo que se hace necesario la inclusión en el SGBD de un precompilador que traduzca las instrucciones DML en instrucciones reconocibles por el compilador del lenguaje anfitrión. La otra forma de utilización del DML, es mediante programas que contengan exclusivamente sentencias propias de este lenguaje. El DML está basado en el álgebra relacional e incluye órdenes para insertar, suprimir, y modificar tuplas (filas) de la base de datos.

Admite las siguientes sentencias de definición: Select, Insert, Delete y Update

Clasificación del lenguaje de manipulación de datos

Se clasifican en dos grandes grupos:

- lenguajes de consulta procedimentales

Lenguajes procedimentales. En este tipo de lenguaje el usuario da instrucciones al sistema para que realice una serie de procedimientos u operaciones en la base de datos para calcular un resultado final.

- lenguajes de consulta no procedimentales

En los lenguajes no procedimentales el usuario describe la información deseada sin un procedimiento específico para obtener esa información.

3. El Lenguaje de Control de Datos (DCL - Data Control Language) permite establecer derechos de acceso a los usuarios, comprobaciones de integridad y control de transacciones. Incluye órdenes para dar y quitar privilegios, así como para completar y abortar transacciones. Algunos ejemplos de comandos incluidos en el DCL son los siguientes:

GRANT: Permite dar permisos a uno o varios usuarios o roles para realizar tareas determinadas.

REVOKE: Permite eliminar permisos que previamente se han concedido con GRANT

Características del lenguaje SQL

Las características esenciales del lenguaje SQL son:

- Sencillez. Principalmente derivada de la sencillez conceptual del modelo en el que se basa, el modelo relacional.
- Carácter estándar. Existe una especificación estándar de este lenguaje, la ANSI SQL. No obstante, cada fabricante refleja las peculiaridades propias de su SGBD modificando su motor de SQL mediante la inclusión de un conjunto de sentencias que faciliten la utilización y gestión de la base de datos
- Lenguaje declarativo. SQL es un lenguaje declarativo. Cuando realizamos una consulta, describimos cuál es el conjunto de datos que queremos obtener, sin tener que especificar cuál es la estrategia de recuperación de esos datos. Especificamos QUÉ queremos, sin decir CÓMO conseguirlo. Los lenguajes en los que es necesario decir cómo obtener los datos, se denominan lenguajes imperativos.

Notación

El lenguaje SQL consta de una serie de sentencias cada una de las cuales requiere una acción específica por parte del SGBD (por ejemplo la creación de una tabla, la inserción de datos en una tabla, ...).

Todas las órdenes en SQL comienzan con un verbo que indica la operación a realizar (CREATE, ALTER, UPDATE, ...) seguido de una o más cláusulas que especifican los datos sobre los que actuar o proporcionan más detalles de la operación que hay que llevar a cabo (WHERE, FROM, SET, ...).

Algunas cláusulas son opcionales y otras obligatorias pero todas deben comenzar con una palabra clave. Las palabras clave no pueden ser utilizadas para designar objetos de la base de datos, tales como nombres de tabla o de columna.

La normativa ANSI especifica las palabras claves que se pueden utilizar en las órdenes SQL. Sin embargo, la mayoría de productos comerciales SQL incluyen, además del conjunto de palabras clave estándar, extensiones significativas que permiten la construcción de sentencias más complejas.

A la hora de expresar las distintas construcciones de SQL seguiremos la siguiente notación:

Las palabras clave aparecerán en este texto en mayúsculas.

Los corchetes indican algo opcional.

Las llaves encierran alternativas de las que se debe elegir una.

Tipos de Datos

El estándar ANSI especifica varios tipos de datos que pueden ser almacena dos en un base da datos para ser manipulados con SQL. Cada columna de una tabla debe ser de uno de los siguientes tipos:

• Numéricos

SMALLINTEGER e INTEGER (o SMALLINT e INT). Contienen números enteros. Las columnas de este tipo almacenan típicamente cuentas, cantidades, edades,etc. También se utilizan con frecuencia para contener números identificadores, tales como número de cliente, de empleado o de pedido.

DECIMAL (a,b) o NUMERIC(a,b). Contienen números decimales.

Almacenan datos con parte fraccionaria que deben ser calculados exactamente, tales como porcentajes y tasas.

a = longitud total

b = dígitos a la derecha del punto decimal

FLOAT(a,b), REAL y DOUBLE_PRECISION. Contienen números en coma flotante. Las columnas de este tipo se utilizan para almacenar números científicos que pueden ser calculados aproximadamente, tales como pesos y distancias. Los números en coma flotante pueden representar mayores rangos que los números decimales, pero pueden producir errores de redondeo en los cálculos.

• Alfanuméricos

CHAR(n). Contiene una cadena alfanumérica de longitud n. n suele estar comprendido entre 1 y 255. Almacenan típicamente nombres de personas y empresas, direcciones, descripciones, ...

• Fechas

El nombre de tipo más usual es DATE.

• Lógicos

Los nombres de tipos más comunes son BOOLEAN y LOGICAL.

Constantes

En algunas órdenes SQL, un valor de datos numérico, de carácter o de fecha debe ser expresado de forma textual. En estos casos se hace uso de las constantes.

• Una constante numérica se escribe con números decimales ordinarios dentro de la orden SQL:

-3467

+85.7

• Las constantes de caracteres deben ir encerradas entre comilla simples: 'HOLA'

'esto es una cadena de caracteres constante'

• Los formatos admitidos para constantes de fechas varían de un SGBD a otro. Los formatos más comunes utilizan las constantes de fechas encerradas entre comillas o entre llaves:

{11-5-1997}
“25-2-1998”

Expresiones

Las expresiones se utilizan en el lenguaje SQL para calcular valores a través de la información almacenada en la base de datos. Las operaciones posibles dependerán del SGBD, pero como mínimo se incluirán las siguientes:

• Operaciones aritméticas

a + b: 5+6

a- b: 8.3-5.7

a * b: 45*2

a / b: 20/10

• Operaciones con cadenas

a + b = ab: 'XYZ'+'123' = 'XYZ123'

LEFT(a,n): LEFT('HOLA',2) = 'HO'

RIGTH(a,n): RIGHT('HOLA',2) = 'LA'

- Operaciones con fechas

a + b: {11-6-1998}+20={1-7-1998}

a - b: {20-5-1998}-10={10-5-1998}

- Operaciones booleanas

a OR b: (5<6) OR (5<4) = verdadero
a AND b: (5<6) AND (5<4) = falso
NOT a: NOT (5<4) = verdadero

También se puede hacer uso de los paréntesis para formar expresiones más complejas.

HERRAMIENTAS GRÁFICAS PROPORCIONADAS POR LOS SGBD

Es muy sencillo manipular una base de datos compleja si se dispone de un interfaz gráfica de usuario que ayude a DBA (Database Administrator, a envar comando de administración de forma automática y sin necesidad de conocer su sintaxis:

PhpMyAdmin de MySQL

MySQL dispone de un interfaz basada en páginas web llamada PhpMyAdmin, que a través de un servidor web, por ejemplo Apache, permite administrar las bases de datos de un servidor desde cualquier equipo de la red. Este software dispone de opciones para realizar prácticamente cualquier opción que se pueda realizar vía SQL. Permite gestionar las bases de datos de un servidor, crear, borrar y modificar tablas, lanzar comandos SQL, exportar e importar información, recopilar estadísticas, hacer copias de seguridad, etc. Además, dispone de un pequeño diseñador, tipo MySQL Workbench que permite gestionar las relaciones de las tablas.

Oracle Enterprise Manager y Grid Control

El SGBD Oracle dispone de dos herramientas gráfica, ambas con interfaz web y montadas sobre un servidor web propietario de Oracle.

La herramienta Enterprise Manager la incorpora directamente en el software de Oracle, y es configurada por el asistente de creación de tablas, usuarios, exportación e importación de información, etc

La herramienta Grid Control se ha de instalar aparte del software de Oracle. Permite gestionar múltiples bases de datos en diversos servidores, permitiendo consultar el estado y rendimiento de todas y cada una de ellas.

Ambas herramientas son extremadamente pesadas en términos de consumo de recursos, por lo que en muchas ocasiones se ha de administrar la base de datos sin ayuda.

DB2 Data Studio

Este software sustituirá en un futuro próximo a una herramienta más antigua llamada Control Center. Permite manipular los objetos de bases de datos DB2 tanto en Windows como en Linux, y simplifica la construcción de consultas SQL. Su gran potencia es que facilita la creación de Servicios Web que distribuyen datos de consultas SQL a las aplicaciones cliente.

Muchos administradores sólo conocen las herramientas gráficas de gestión y administración de una base de datos, puesto que es más cómodo y más intuitivo, y además, aprender los lenguajes de programación de bases de datos es una tarea difícil y laboriosa. Sin embargo, conocer los comandos y las instrucciones que proporciona un SGBD, otorga una visión extra que posibilita automatizar tareas rutinarias y permite solucionar problemas que no se pueden solucionar solo con las herramientas gráficas. A un administrador que conoce a la perfección todos estos comandos, le resulta muy sencillo actualizarse en los continuos cambios de versiones en estas herramientas gráficas.

En modo de resumen, están las operaciones más básicas :

Instrucción	Descripción
Show databases;	Muestra el conjunto de bases de datos presentes en el servidor
Use nombre_de_la_base	Determina la base de datos sobre la que vamos a trabajar
Create Database nombre_de_la_base;	Crea una nueva bd con el nombre especificado
Drop Database nombre_de_la_base;	Elimina la base de datos del nombre especificado
Show tables;	Muestra las tablas presentes en la base de datos actual
Describe nombre_de_la_tabla;	Describe los campos que componen la tabla
Drop Table nombre_de_la_tabla;	Borra la tabla de la base de datos
Load Data Local Infile "archivo.txt" Into Table nombre_de_la_tabla;	Crea los registros de la tabla a partir de un fichero de texto en el que separamos por tabulaciones todos los campos de un mismo registro.
Quit	Salir de MySQL

Para evitarnos el tener que editar nuestras tablas directamente sobre archivos de texto, puede resultar muy práctico usar cualquier otra base de datos con un editor y exportar a continuación la tabla en un archivo de texto configurado para dejar tabulaciones entre cada campo. Esto es posible en Access por ejemplo pinchando con el botón derecho sobre la tabla que queremos convertir y eligiendo la opción exportar. Una ventana de dialogo aparecerá en la que elegiremos guardar el archivo en tipo texto. El paso siguiente será elegir un formato delimitado por tabulaciones sin cualificador de texto.

Otra posibilidad que puede resultar muy práctica y que nos evita trabajar continuamente tecleando órdenes al estilo de antaño es servirse de programas en PHP o Perl ya existentes y descargables en la red. El más popular sin duda es phpMyAdmin. Este tipo de scripts son ejecutados desde un navegador y pueden ser por tanto albergados en nuestro servidor o empleados en local para, a partir de ellos, administrar MySQL de una forma menos sufrida.

Asimismo, dentro del directorio bin de MySQL, podemos encontrar una pequeña aplicación llamada MySqlManager. Se trata de una interface windows, más agradable a la vista y al uso que la que obtenemos ejecutando el archivo mysql. En este caso, las sentencias SQL deben realizarse sin el punto y coma final

Intérpretes de comandos de los SGBD

La utilidad principal de un SGBD es su interprete de comandos. ES una aplicación cliente cuya única misión es enviar comandos al SGBD y mostrar los resultados devueltos por el SGBD en pantalla. El cliente del servidor MySQL (mysql-server) se llama mysql, el de oracle se llama sqlplus y el de DB2 se llama db2. Para invocar maria desde el sistema operativo (Windows o Unix), tan sólo hay que escribir en un terminal su nombre con ciertas opciones.

MySQL: El cliente de MySQL-Server

```
mysql [options] [database]

options:
     --help                        Visual iza la ayuda
     {-p |--password}[=frase]      Password con la que se conecta
     {-p |--port} [=numero]        Puerto TCPIP remoto al que se conecte
     {-h |--host}[=numero]         Nombre Host o IP al que se conecta
     {-u |--user}[=usuario]        Usuario con el que se conecta
     {-s |--socket}[=nombre_fich] Fichero socket con el que se conec

#ejemplo típico:
mysql -u root -p
Enter password: *********

Welcome to the MySQL monitor. Commands end with
Your MySQL connection id is 37
Server version: 5.0.75-0ubuntu10.2 (Ubuntu)
or \g.
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

El comando puede ir acompañado de dos tokens opcionales, *options*, que permite especificar una serie de parámetros de conexión y *database*, que específica en qué base de datos se ejecutarán los comandos introducidos. Las opciones que aquí se muestran son sólo algunas de las disponibles, para más información consultar la documentación de MySQL:

- La opción -help muestra un resumen de las opciones disponibles.
- La opción -u o -user permite especificar el usuario de conexión, por ejemplo -u paco.
- La opción -p o -password permite introducir un password para la conexión del usuario. Se puede escribir -p frase_password o sólo -p para que el gestor la solicite y la oculte con asteriscos para que nadie pueda espiar.
- La opción -h o -host permite seleccionar el host donde está el gestor de base de datos. Puede ser localhost o 127.0.0.1 si la base de datos está en el ordenador local y se desea conectar vía tcp-ip u otra dirección IP. Si se omite, por defecto se conecta al servidor por named pipes (o tuberías con nombre) que no salen a la interfaz local de la tarjeta de red y son mecanismos de comunicación más rápidos, pues son internos al ordenador.

- La opción -P o -p permite especificar el número de puerto al que conectarse (si se conecta vía TCP-IP al ordenador remoto). Si no se indica este parámetro, por defecto se conecta al puerto 3306.

- Permite especificar el nombre del socket por defecto, esto será necesario cambiarlo cuando exista más de una versión del gestor de base de datos ejecutándose en un ordenador.

A continuación se muestran unos cuantos ejemplos de conexión:

```
#conexión sin usuario y password (se conecta como anónimo y sin password)
mysql

#conexión con usuario y password (se conecta como root y su password )
mysql -u root -p
Enter password: *********

#conexión con usuario y password en claro a la base de datos jardineria
mysql -u root -pPasswordDelUsuario jardineria

#conexión con usuario y password en claro a la base de datos jardineria
# del host 192.168.3.100
mysql -u root -pPasswordDelUsuario -h 192.168.3.100 jardineria

#conexión con usuario y password en claro a la base de datos jardineria
# del host 192.168.3.100 con puerto 15300
mysql -u root -pPasswordDelUsuario -h 192.168.3.100 jardineria -P 15300
```

Ejecución de consultas en MySQL

Para ejecutar una consulta, tan sólo es necesario arrancar el cliente mysql y onectarse a una base de datos del gestor. A continuación, escribir en la consola el comando SQL que se desea ejecutar y se obtienen los resultados. Cuando una línea es precedida del carácter #, el resto de la línea se ignora. Estas líneas se llaman *comentarios* y son útiles para documentar las acciones realizadas. Por ejemplo:

```
#selecciona la version del gestor y la fecha actual
mysql> select version(),current_date();
+-------------------+---------------+
I versionO                    I current_date() I
+-------------------+---------------+
I 5.0.75-0ubuntu10.2 I 2009-8-20
+-------------------+---------------+
1 row in set (0.00 sec)
```

Este comando ilustra distintas cosas:

- Un comando consiste en una sentencia SQL terminada en punto y coma.

- Cuando se escribe un comando, mysql manda al servidor para que lo ejecute, muestra los

resultados y vuelve al *prompt*, indicando que está listo para recibir más consultas.

- Muestra los resultados en forma de tabla (filas y columnas). La primera fila contiene las etiquetas para las columnas y las filas siguientes muestran los resultados de la consulta.
- Muestra las filas devueltas y cuánto tiempo tardó en ejecutarse, lo cual puede ofrecer una idea de la eficiencia del servidor, aunque estos valores pueden ser imprecisos pues dependen de muchos factores, tales como la carga del servidor, velocidad de comunicación, etc.

Las palabras claves pueden ser escritas en mayúsculas o minúsculas, y los identificadores de tabla, campo, índice, etc son sensitivos a las mayúsculas y minúsculas en las versiones de unix (no así en Windows). Por ejemplo, las siguientes consultas son equivalentes:

```
mysql> SELECT VERSION(),CURRENT_DATE();

mysql> SELect Version(),current_Date();

mysql> select version(),current_date();
```

Es posible escribir más de una consulta por línea, siempre y cuando estén separadas por punto y coma:

```
mysql> select user();select now();
+----------------+
I userO
+----------------+
I root@localhost I
+----------------+
t---------------------+
I nov O
t---------------------+
I 2009-10-20 09:53:29 I
t---------------------+
```

No es necesario escribir un comando en una sola línea, así que los comandos que requieran de varias líneas no son un problema. MySQL determinará donde finaliza la sentencia cuando encuentre el punto y coma, no cuando encuentre el fin de línea:

```
mysql> select user(),
    ->    current_date();
t----------------+----------------+
Iuser()          I current_date()
I--------------+----------------+
I root@localhost   I 2009-10-20
t----------------+----------------+
```

Si no se desea terminar de escribir la consulta, por ejemplo, por haber cometido un error en la sintaxis, se escribe para terminarla sin ejecutar:

```
mysql> select
-> user()
    -> \c
mysql>
```

El prompt se comunica con el usuario dándole información sobre qué ocurre, por ejemplo:

```
-> Espera la siguiente línea del comando
'> Espera que se cierre una comilla sencilla
"> Espera que se cierre una comilla doble
mysql> Listo para una nueva consulta
```

Otra forma de ejecutar comandos SQL es almacenarlos en un fichero de texto y mandarlo a ejecución mediante el comando *source*, que recibe como parámetro un fichero de comandos y ejecuta uno por uno todos los comandos que tenga el fichero:

```
#ejecución del script de creación crear_bbdd_startrek.sql
mysql> source /home/malu/crearbbdd.sql
```

Por último, también es posible ejecutar los comandos de un fichero de texto desde la shell, esto es, en modo batch, redirigiendo al cliente mysql un fichero de entrada. Esto es útil para tareas administrativas donde se ejecutan varios ficheros de comandos, además de otras tareas de mantenimiento del servidor a través de un shell script:

```
#ejecucion en modo batch
-$ mysql -u root -pPassWdUsuario <crear_bbdd.sql

#ejecucion en modo batch almacenando resultados
-$ mysql -u root -pPassWdUsuario <crear_bbdd.sql >resultado
```