

## APÉNDICE

### SQL (Structures Query Language)

SQL se ha establecido como lenguaje de bases de datos relacionales estándar, y se pueden distinguir tres partes:

**DDL** (Data Definition Language) permite creación de tablas, vistas e índices.

**DML** (Data Manipulation Language) permite consultas y actualizaciones.

**DCL** (Data Control Language) permite seguridad, controles de acceso y privilegios.

#### **DDL(Lenguaje de descripción de datos)**

Antes de comenzar a trabajar en SQL hay que activarlo con la orden :

**.SET SQL ON**

Creación de la base de datos :

**CREATE DATABASE <nombre>;**

Activar una base de datos:

**START DATABASE <nombre>;**

Detener la base de datos:

**STOP DATABASE <nombre>;**

Borrar una base de datos:

**DROP DATABASE <nombre>;**

Mostrar una lista de las bases de datos:

**SHOW DATABASE;**

Crear una tabla en la base de datos activa:

**CREATE TABLE <nombre-tabla> (atributo1 tipo-dato, atributo2 tipo-dato,.....);**

Tipos de datos disponibles :

INTEGER : Entero largo.  
 SMALLINT : Entero pequeño.  
 CHAR(n) : Cadena de n caracteres.  
 FLOAT(x,y) : Número de x cifras en coma flotantes con y cifras fraccionarias.  
 DECIMAL(m,n) : Número fraccionario con m dígitos de los cuales n son fraccionarios.  
 TIME : Hora en formato '20:50:35'.  
 DATE : Fecha con formato '15/12/97'.  
 LOGICAL : Valores lógicos.

Modificación de la estructura de una tabla.

**ALTER TABLE <nombre-tabla> ADD (atributo tipo-dato);**

Eliminación de una tabla:

**DROP TABLE <nombre-tabla>;**

Creación de índices:

**CREATE [UNIQUE] INDEX <nombre-índice>  
ON <nombre-tabla> (atributo [ASC | DESC] ,.....);**

Borrar índice:

**DROP INDEX <nombre-índice>;**

## **EJERCICIO.**

Utilizando este ejemplo veamos la órdenes en SQL.

Crear la base de datos HOSPITAL y las siguientes tablas :

Plantas(planta,nombre)

Habita(nhabita,planta)

Camas(nhabita,cama)

Paciente(dni,nhabita,cama,fecha,edad)

Creación de la base de datos:

**CREATE DATABASE hospital;**

Creación de las tablas:

```
CREATE TABLE Plantas (planta smallint, nombre char(20));
CREATE TABLE Habita(nhabita smallint, planta smallint);
CREATE TABLE Camas(nhabita smallint, cama smallint);
CREATE TABLE Paciente(dni char(8), nhabita smallint, cama smallint, fecha
date, edad smallint);
```

Insertar la siguiente filas en la tablas :

```
INSERT INTO paciente VALUES ('11111111',201,2,{01/01/97},45);
```

PLANTAS		HABITA	
planta	nombre	nhabita	planta
1	medicina interna	101	1
2	ginecologia	102	1
3	traumatologia	103	1
		201	2
		202	2
		203	2
		301	3
		302	3

  

CAMAS		PACIENTE				
nhabita	cama	dni	nhabita	cama	fecha	edad
101	1	11111111	201	2	1/1/97	45
102	1	22222222	101	1	1/2/97	50
102	2	33333333	201	1	5/2/97	25
103	1	44444444	102	2	5/8/97	60
201	1	55555555	301	2	6/7/97	40
201	2					
202	1					
203	1					
301	1					
301	2					
302	1					

## DML.(Lenguaje de Manejo de Datos)

Sentencias básicas son:

SELECT : Acceso a los datos.  
 UPDATE : Modificación de datos.  
 INSERT : Inserción de filas.  
 DELETE : Eliminación de filas.

### SELECT

**SELECT campos**  
**FROM tablas**  
**WHERE condiciones;**

Obtener las especialidades del hospital.

```
SELECT nombre FROM plantas;
```

Obtener todos los pacientes del hospital.

```
SELECT * FROM paciente;
```

Obtener las habitaciones de la primera planta.

```
SELECT nhabita FROM habita WHERE planta=1;
```

Obtener los pacientes de ginecología.

```
SELECT paciente.dni FROM paciente, habita, plantas
    WHERE paciente.nhabita=habita.nhabita AND habita.planta=plantas.planta
    AND plantas.nombre='ginecologia';
```

Contar los pacientes que hay en el hospital.

```
SELECT COUNT(*) FROM paciente;
```

Calcular la edad media de los pacientes.

```
SELECT AVG(edad) FROM paciente;
```

Obtener las camas ocupadas ordenadamente.

```
SELECT nhabita,cama FROM paciente ORDER BY nhabita,cama;
```

Habitaciones y camas ocupadas en la primera planta.

```
SELECT nhabita,cama FROM paciente WHERE nhabita IN
    (SELECT nhabita FROM habita WHERE planta=1);
```

Pacientes ingresados en una u otra fecha.

```
SELECT * FROM paciente WHERE fecha IN ({01/01/97},{01/02/97});
```

Pacientes ingresados entre dos fechas.

```
SELECT * FROM paciente WHERE fecha BETWEEN {01/01/97} AND {01/06/97};
```

Obtener las camas no ocupadas.

```
SELECT * FROM camas WHERE NOT EXISTS
    (SELECT * FROM paciente WHERE paciente.nhabita=camas.nhabita
        AND paciente.cama=camas.cama);
```

Encontrar las habitaciones vacías completamente.

```
SELECT DISTINCT nhabita FROM camas WHERE NOT EXISTS
    (SELECT nhabita FROM paciente WHERE nhabita=camas.nhabita);
```

Obtener todos los datos donde se encuentra un paciente.

```
SELECT paciente.nhabita, paciente.cama, paciente.fecha, habita.planta, plantas.nombre
    FROM paciente, habita, plantas
    WHERE paciente.dni='22222222' AND
        paciente.nhabita=habita.nhabita AND
        habita.planta=plantas.planta;
```

Calcular el paciente con más tiempo en el hospital.

```
SELECT * FROM paciente WHERE fecha IN (SELECT MAX(fecha) FROM
paciente);
```

Mostrar la habitaciones y número de camas.

```
SELECT nhabita, COUNT(camas) FROM camas
GROUP BY nhabita ORDER BY nhabita;
```

Habitaciones con dos camas.

```
SELECT nhabita FROM camas GROUP BY nhabita HAVING (((Count(cama))=2));
```

## **INSERT**

```
INSERT INTO tabla
lista_columnas
VALUES (lista_de_valores);
```

Añadir datos de una habitación en al planta 3 con 3 camas.

```
INSERT INTO habita VALUES (303,3);
INSERT INTO camas VALUES (303,1);
INSERT INTO camas VALUES (303,2);
INSERT INTO camas VALUES (303,3);
```

Añadir los datos de los pacientes de la primera planta a otra tabla de traslado.

```
INSERT INTO traslado SELECT dni,fecha,edad FROM paciente
WHERE nhabita IN (SELECT nhabita FROM habita WHERE planta=1);
```

**UPDATE**

**UPDATE tabla SET columna=valor [,columna=valor,...]  
[ WHERE condiciones ];**

Las habitaciones de la segunda planta situarlas en la cuarta.

UPDATE habita SET planta=4 WHERE planta=1;

**DELETE**

**DELETE FROM tabla  
[ WHERE condiciones];**

Eliminar todas las filas de la tabla paciente.

DELETE FROM paciente;

Eliminar las camas de la primera planta.

DELETE FROM camas WHERE nhabita IN  
(SELECT nhabita FROM habita WHERE planta=1);