Apuntes SQL

Definición de SQL

El lenguaje de consulta estructurado o SQL es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar a la base de datos los datos que queremos utilizar y nunca como recuperar los datos.

Consultas

Toda consulta debe tener una sintaxis básica, la cual sería la instrucción SELECT y la cláusula FROM. SELECT se utiliza para indicar el campo de la tabla que queremos seleccionar y FROM es necesario ya que indica la tabla sobre la que se aplica el SELECT. Un ejemplo sería :

```
SELECT name,population
FROM world;
```

En esta consulta seleccionamos name y population de la tabla “world”.

Result:

name	population
Afghanistan	25500100
Albania	2821977
Algeria	38700000
Andorra	76098
Angola	19183590
Antigua and Barbuda	86295
Argentina	42669500
Armenia	3017400
Australia	23545500
Austria	8504850
Azerbaijan	9477100
Bahamas	351461
Bahrain	1234571
Bangladesh	156557000
Barbados	285000

La cual nos dará una tabla con dos columnas, en la primera columna nos dará el nombre del país y en la segunda el número de habitantes de ese país.

WHERE

WHERE se utiliza para ejecutar predicados. Es opcional y determina los registros de las tablas enumeradas en la cláusula FROM aparecerán en los resultados de la instrucción SELECT. Ejemplo:

```
SELECT name FROM world
WHERE population >= 200000000;
```

En esta consulta seleccionamos el nombre de los países en los cuales la población es mayor a 200 millones.

Correct answer

name
Brazil
China
India
Indonesia
United States

El resultado será una tabla de una columna con solo los países que superan los 200 millones de habitantes.

Si se necesitan hacer subconsultas, estas nunca deben ir después del WHERE tienen que ir después de lo que queremos compararlas.

```
SELECT name FROM world
  WHERE population >
     (SELECT population FROM world
      WHERE name='Russia')
```

Si al hacer una subconsulta la hacemos al revés nos dará un error.

```
SELECT name FROM world
  WHERE (SELECT population
         FROM world
         WHERE name='Russia') < population;
```

Operadores Lógicos

Los operadores lógicos son AND, OR, IS ,NOT.

AND sirve para seleccionar expresiones que cumplan dos requisitos como :

```
SELECT name, population
FROM world
WHERE population >
    (SELECT population FROM world WHERE name = 'Canada')
AND population <
    (SELECT population FROM world WHERE name = 'Poland')
```

Esta consulta busca los países que tengan una población inferior a la de Canadá y superior a la de Polonia. Dando como resultado una tabla con dos columnas en las que aparece el nombre del país y su población.

OR sirve para seleccionar las expresiones que cumplan un requisito de los pedidos por ejemplo:

```
SELECT name
FROM Empleados
WHERE (Edad > 25 AND Edad < 50) OR Sueldo = 100;
```

Esta consulta busca los nombres de los empleados que tienen una edad entre 25 y 50 años o un sueldo igual a 100.

NOT :

```
SELECT name
FROM Empleados
WHERE NOT Estado = 'Soltero';
```

En esta consulta busca el nombre de los empleados que NO están solteros.

BETWEEN

Es un operador que sirve para indicar que queremos los datos que existen entre dos expresiones como por ejemplo:

```
SELECT name, area
FROM world
WHERE area BETWEEN 250000 AND 300000
```

En esta consulta se busca el nombre y área de los países que tengan un área entre 250000 Km y 300000 Km.

LIKE

LIKE es un operador que se utiliza con expresiones regulares, al contrario de "=" que se utiliza para cadenas.

Una consulta con = es una cadena y con LIKE una expresión regular en la que% se puede cambiar por 0 o n caracteres extra

```
SELECT name FROM world
  WHERE name LIKE 'Y%';
```

Esta consulta selecciona los nombres que empiecen por Y.

Con este operador, "%" significa varios caracteres da igual el número y "_" significa un solo carácter.

```
SELECT name FROM world
 WHERE name LIKE '_t%'
ORDER BY name;
```

En esta consulta selecciona los nombres que tienen una t como segunda letra.

IN

Este operador se utiliza cuando hay más de una expresión y no se puede utilizar "=". Un ejemplo sería:

```
SELECT name, population FROM world
  WHERE name IN ('Sweden', 'Norway', 'Denmark');
```

En esta consulta se busca la población de los países en los que su nombre es Sweden, Norway y Denamark.

OPERADORES DE COMPARACIÓN

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor o igual que
>=	Mayor o igual que
BETWEEN	Intervalo
LIKE	Comparación
In	Especificar

DISTINCT

Se utiliza en el SELECT y sirve para omitir datos duplicados en los campos seleccionados.

```
SELECT DISTINCT region
FROM bbc;
```

ORDER BY

Sirve para ordenar el resultado que obtenemos en una consulta. Podemos ordenar más de un registro de datos. Y podemos especificar el orden en el que queremos que se ordenen con DESC indicando que lo ordene descendentemente y ASC indicando que los ordene de forma ASC. Por defecto los ordenará de manera ascendete.

```
SELECT name, population
  FROM world
  WHERE population > 100000000
  ORDER BY population DESC;
```

Ordena de forma descendente el nombre y población de los países que tengan una población superior a 100000000.

```
SELECT name, population
  FROM world
  WHERE population > 100000000
  ORDER BY population ASC;
```

Hace lo mismo pero de forma ascendente.

AVG

Realiza la media aritmética de valores obtenidos en un campo especifico. No incluye valores NULL.

MAX,MIN

Devuelven el mínimo o el máximo de un conjunto de valores contenidos en un campo de la tabla.

SUM

Devuelve la suma de los valores de un campo.

```
SELECT SUM(population), SUM(gdp)
  FROM bbc
  WHERE region = 'Europe';
```

Suma la población y el gdp de Europa y da como resultado una tabla con dos columnas y una fila con el total de la suma.

GROUP BY

Agrupa los valores obtenidos en los registros que indiquemos por ejemplo.

```
SELECT continent,
    COUNT(name) AS 'NÂº of countries'
FROM world
WHERE population >= 10000000
GROUP BY continent;
```

Agrupa en continentes el número de países que sobre pasan los 10 millones de habitantes.

COUNT

Cuenta los valores de un registro. Es un reductor . Si en una consulta aparece SELECT COUNT(*) el resultado será una tupla y una columna.

```
SELECT continent,
    COUNT(name) AS 'NÂº of countries'
FROM world
WHERE population >= 10000000
GROUP BY continent;
```

Cuenta el número de países que tienen una población superior a 10 millones de habitantes y los agrupa por continentes.

HAVING

Expresa la condición que deben cumplir cada grupo. Al igual que WHERE ejecuta de predicados. Se utiliza con SUM.

```
SELECT continent, SUM(population)
FROM world
WHERE population > 10000000
GROUP BY continent
HAVING SUM(population) >= 100000000;
```

Esta consulta lista los continentes que tienen una población total por encima de 100000000 habitantes.

JOIN

Es una sentencia que sirve para relacionar varias tablas. Cuando utilizamos JOIN debemos poner delante de los registros del SELECT el nombre de la tabla a la que pertenecen para saber que tablas debemos relacionar. Se utiliza en la cláusula FROM y suele ir acompañado de ON.

ON al igual que WHERE y HAVING , ejecuta predicados. No es obligatorio puede ser eliminado y ejecutar el predicado desde el WHERE.

Existen 3 tipos de JOIN que son :

- INNER JOIN o JOIN : Sirve para relacionar tablas.

```
SELECT
  teacher.name AS profes,
  dept.name    AS departamentos
FROM teacher INNER JOIN dept ON dept.id = teacher.dept;
```

- RIGHT JOIN: Dará el resultado y los campos de la tabla de la derecha del JOIN que no tienen coincidencia con la tabla de la izquierda.

```
SELECT
  teacher.name AS profes,
  dept.name    AS departamentos
FROM teacher RIGHT JOIN dept ON dept.id = teacher.dept;
```

- LEFT JOIN: Dará el resultado y los campos de la tabla de la izquierda del JOIN que no tienen coincidencia con la tabla de la derecha.

```
SELECT
  teacher.name AS profes,
  dept.name    AS departamentos
FROM teacher LEFT JOIN dept ON dept.id = teacher.dept;
```

NULL

Es un valor nulo que puede aparecer en las tablas debido a la inexistencia de datos. Podemos utilizar IS NULL para seleccionar campos que tengan valores nulos o IS NOT NULL para seleccionar campos que no tengan valores nulos.

```
SELECT teacher.name
FROM teacher
WHERE teacher.dept IS NULL;
```

Da como resultado los profesores que son nulos para su departamento

```
SELECT teacher.name
FROM teacher
WHERE teacher.dept IS NOT NULL;
```

Da como resultado los profesores que no son nulos para su departamento.

La orden de ejecución en una consulta es FROM , WHERE, GROUP BY, HAVING, SELECT.