

# *Práctica 1*



## TERMINOLOGÍA

### SQL

- Structured Query Language o Lenguaje de Consultas Estructurado.
  - ▶ Es el lenguaje que permite la comunicación con el Sistema Gestor de Bases de Datos (Oracle en nuestro caso).
- El SQL es un lenguaje unificado
  - ▶ Lo utilizan todo tipo de usuarios, desde el administrador de la base de datos, DBA, hasta el usuario final.
- El SQL es un lenguaje no procedimental.
  - ▶ El usuario especifica Qué quiere, no Cómo ni Dónde conseguirlo.
- El SQL es relacionalmente completo.
  - ▶ Permite la realización de cualquier consulta de datos.



## SQL= DDL + DML

Las sentencias del SQL se clasifican como parte del DDL o del DML.

### ■ Lenguaje de Definición de Datos, DDL

- ▶ sentencias del SQL que permiten definir los objetos de la Base de Datos (create, revoke, grant, alter, etc.). Cuando se definen dichos objetos se almacenan en el diccionario de datos.

### ■ Lenguaje de Manipulación de Datos, DML

- ▶ sentencias del SQL que se utilizan para manejar los datos de la base de datos (select, insert, update, delete, etc).



```
SELECT [ALL | DISTINCT | UNIQUE ] <lista de selecciones>
FROM <nombre de tabla> [alias de tabla] [,...]
[WHERE <condición>]
[GROUP BY <lista-de columnas> [HAVING <condición>] ]
[ORDER BY <nombre de columna> [ASC | DESC] [,...] ]
```



## CLAÚSULAS: SELECT

- **SELECT:** indica qué atributos o funciones se van a recuperar.

**SELECT [DISTINCT | UNIQUE ] < lista-de selecciones | \* >**

- ▶ **DISTINCT:**

Elimina las filas duplicadas en el resultado de la consulta.

- ▶ **UNIQUE:**

Es igual que **DISTINCT**.

- ▶ **<lista-de selecciones>**

Lista de nombre de columnas o expresiones separadas por comas.

- ▶ **\***

Refiere a todas las columnas de todas las tablas



## CLAÚSULAS: FROM y WHERE

- **FROM:** especifica todas las relaciones (tablas) que se necesitan para la consulta

**FROM { <nombre de tabla> } [...]**

- **WHERE:** especifica las condiciones, si es que son necesarias, para seleccionar las filas de esas tablas, incluyendo las condiciones de reunión.

**WHERE <condición>**

*“Una **condición** está formada por una o varias expresiones condicionales conectadas por los operadores lógicos*

***AND, OR y NOT”***



## CLAÚSULAS: WHERE (expresiones)

- Una *EXPRESIÓN CONDICIONAL* tiene alguna de las formas siguientes:

- ▶ **<expresión1> <operador> <expresión2>**

*Verifica si las dos expresiones satisfacen la comparación*

- ▶ **<expresión1> [NOT] BETWEEN <expresión2> AND <expresión3>**

*Verifica si la expresión1 tiene un valor comprendido entre los valores de la expresión2 y la expresión3*

- ▶ **<expresión> [NOT] IN (<lista-de valores>)**

*Verifica si la expresión tiene un valor de los indicados en la lista de valores.*



## CLAÚSULAS: WHERE (expresiones)

- ▶ **<nombre de columna> [NOT] LIKE "<string>"**

*Verifica si el valor de la columna se adapta al patrón de búsqueda (string).  
Se admiten caracteres comodín*

*"%" representa cero o más caracteres*

*"\_" representa un único carácter.*

*'/' El carácter de escape permite referirse a los caracteres comodín como caracteres y no como comodines.*

- ▶ **<nombre de columna> IS [NOT] NULL**

*Verifica si el valor de la columna es nulo.*



## Operadores

$+$ $-$ $  $	SUMAR RESTAR CONCATENAR
$*$ $/$	MULTPLICAR DIVIDIR
$=$ $<>$ $!=$ $<$ $,$ $>$ $<=$ $>=$	COMPARADORES CLASICOS
IS NULL BETWEEN IN LIKE	COMPARADORES ESPECIALES
NOT AND OR	OPERADORES LOGICOS CLASICOS



## CLAÚSULAS: ORDER BY

- ORDER BY: especifica un orden para presentar el resultado de la consulta
  - ▶ ORDER BY <nombre de columna> [ASC | DESC] [,...]
  - ASC
    - los valores de la columna afectada estarán en orden descendente
  - DESC
    - los valores de la columna afectada estarán en orden descendente.



## RESUMEN

- **SELECT:** indica qué atributos o funciones se van a recuperar.
- **FROM:** especifica todas las relaciones (tablas) que se necesitan para la consulta (no las de las consultas “anidadas”)
- **WHERE:** especifica las condiciones, si es que son necesarias, para seleccionar las filas de esas tablas, incluyendo las condiciones de reunión.
- **GROUP BY:** -
- **ORDER BY:** especifica un orden para presentar el resultado de la consulta.



## EJEMPLOS

- “Obtenga el curso y nombre de las asignaturas que empiecen por A y que tengan entre 3 y 9 créditos. Mostrar el resultado ordenado ascendentemente por nombre y descendentemente por curso”

```
SELECT NOMBRE, CURSO
FROM ASIGNATURAS
WHERE NOMBRE LIKE 'A%' AND
                                CREDITOS IN (3,6,9)
ORDER BY NOMBRE, CURSO DESC;
```



## EJEMPLOS

- “Obtenga el código de docencia, las siglas de la asignatura y la anualidad de la misma”.

```
SELECT DOCENCIA.ID, DOCENCIA.PROFESOR,  
       ASIGNATURAS.ANUALIDAD  
FROM DOCENCIA, ASIGNATURAS  
WHERE  
       DOCENCIA.SIGLAS=ASIGNATURAS.SIGLAS;
```



## EJEMPLOS

- “Obtenga el código de docencia, las siglas de la asignatura y la anualidad de la misma”.

```
SELECT D.id, D.profesor, anualidad  
FROM docencia D, asignaturas A  
WHERE D.siglas=A.siglas;
```



## Base de datos para ejemplos: semántica

- Disponemos de una base de datos para la gestión de la docencia en una universidad. En dicha BD se almacena información correspondiente a profesores, asignaturas, departamentos, áreas, locales, la docencia que se imparte y los grupos de alumnos.



## Base de datos para ejemplos: tablas

