

Módulo de adaptación

Master en Business Intelligence y Big Data

PROFESOR/A
Antonio Sarasa Cabezuelo

Consulta de tablas

SELECT FROM

- Para hacer consultas sobre una tabla se utiliza la sentencia **SELECT**:

```
SELECT nombre_columna_a_seleccionar [[AS] col_renombrada]  
[,nombre_columna_a_seleccionar [[AS] col_renombrada]...]  
FROM tabla_a_consultar [[AS] tabla_renombrada];
```

- La palabra clave **AS** permite renombrar las columnas que se quieren seleccionar o las tablas que se quieren consultar. Esta palabra es opcional, y muchas veces se sustituye por un espacio en blanco.

SELECT FROM

- Por ejemplo si queremos seleccionar las columnas código, nombre, dirección y ciudad de la tabla clientes usaríamos la sentencia:

```
SELECT codigo_cli, nombre_cli, direccion, ciudad FROM clientes;
```

- Sin embargo si se quieren recuperar todas las columnas de la tabla se usa el símbolo “*”, en vez de listar todas las columnas:

```
SELECT * FROM clientes;
```

SELECT FROM WHERE

- Si se quieren seleccionar que filas son recuperadas entonces hay que utilizar en la consulta SELECT la palabra reservada WHERE:

```
SELECT [DISTINCT | ALL] nombre_columnas_a_seleccionar  
FROM tabla_a_consultar [WHERE condiciones];
```

- La clausula **WHERE** permite recuperar sólo aquellas filas que cumplen la condición especificada.
- La clausula **DISTINCT** permite indicar que nos muestre las filas resultantes sin repeticiones. La opción por defecto es **ALL** que indica que muestre todas las filas.

SELECT FROM WHERE

- Para construir las condiciones de la clausula **WHERE** es necesario usar operadores de comparación o lógicos: <(menor), >(mayor), =(igual), <=(menor o igual), >=(mayor o igual), <>(distinto), AND(conjunción de condiciones), OR(disyunción de condiciones), NOT(negación).

SELECT FROM WHERE

- Por ejemplo si se quieren recuperar los diferentes sueldos de la tabla empleados:

```
SELECT DISTINCT sueldo FROM empleados;
```

- Y si se quieren recuperar los empleados de la tabla empleados cuyo sueldo es mayor de 1000 euros:

```
SELECT * FROM empleados WHERE sueldo > 1000;
```

Subconsultas

- Una subconsulta es una consulta incluida dentro de otra consulta, y que aparece como parte de una cláusula WHERE o HAVING(se verá a continuación).
- Por ejemplo se quiere obtener los proyectos de la tabla proyectos que se corresponden con un cliente que tiene como NIF el número “444555-E”:

```
SELECT * FROM proyectos  
WHERE codigo_cliente = (SELECT código_cli FROM clientes  
WHERE nif=“444555-E”)
```


Predicados

- En la condición que aparece en la clausula **WHERE** se pueden utilizar un conjunto de predicados predefinidos para construir las condiciones:
 - **BETWEEN**. Expresa que se quiere encontrar un valor entre unos límites concretos:

```
SELECT nombre_columnas_a_seleccionar  
FROM tabla_a_consultar  
WHERE columna BETWEEN límite1 AND límite2;
```

Por ejemplo se quieren recuperar todos los empleados cuyos sueldos están entre 1000 y 2000 euros:

```
SELECT codigo_empl FROM empleados  
WHERE sueldo BETWEEN 1000 and 2000;
```

Predicados

- **IN.** Comprueba si un valor coincide con los elementos de una lista (IN) o no coincide (NOT IN):

```
SELECT nombre_columnas_a_seleccionar  
FROM tabla_a_consultar  
WHERE columna [NOT] IN (valor1, ..., valorN);
```

Por ejemplo se quieren recuperar todos los clientes que viven en Madrid y Zaragoza:

```
SELECT * FROM clientes  
WHERE ciudad IN ('Madrid', 'Zaragoza');
```

Predicados

- **LIKE.** Comprueba si una columna de tipo carácter cumple una condición determinada.

```
SELECT nombre_columnas_a_seleccionar  
FROM tabla_a_consultar  
WHERE columna LIKE condición;
```

Existen un conjunto de caracteres que actúan como comodines:

- El carácter `_` para cada representar un carácter individual.
- El carácter `%` para expresar una secuencia de caracteres incluido la secuencia vacía.

Predicados

Por ejemplo si se quieren recuperar los clientes cuya ciudad de residencia termina por la letra “d”:

```
SELECT * FROM clientes WHERE ciudad LIKE '%d';
```

Y si se quiere refinar la consulta anterior y recuperar los clientes cuya ciudad de residencia termina por la letra “d” y el nombre de la ciudad tiene 6 letras:

```
SELECT * FROM clientes WHERE ciudad LIKE '____d';
```

Predicados

- **IS NULL.** Comprueba si un valor nulo (IS NULL) o no lo es (IS NOT NULL):

```
SELECT nombre_columnas_a_seleccionar  
FROM tabla_a_consultar  
WHERE columna IS [NOT] NULL;
```

Por ejemplo se quieren recuperar todos los clientes que no tienen un número de teléfono:

```
SELECT * FROM clientes WHERE teléfono IS NULL;
```

Predicados

- **EXISTS.** Comprueba si una consulta produce algún resultado(EXISTS) o no(NOT EXISTS):

```
SELECT nombre_columnas_a_seleccionar  
FROM tabla_a_consultar  
WHERE [NOT] EXISTS subconsulta;
```

Por ejemplo se quieren recuperar todos los empleados que están asignados a algún proyecto:

```
SELECT * FROM empleados  
WHERE EXISTS (SELECT * FROM proyectos  
              WHERE codigo_proyec = num_proyec);
```

Predicados

- **ANY/SOME/ALL.** Comprueba si todas(ALL) o algunas(SOME/ANY) de las filas de una columna cumplen las condiciones especificadas:

```
SELECT nombre_columnas_a seleccionar  
FROM tabla_a_consultar  
WHERE columna operador_comparación  
{ALL | ANY | SOME}subconsulta;
```

Predicados

Por ejemplo si se quiere recuperar todos los proyectos en los que los sueldos de todos los empleados asignados son menores que el precio del proyecto:

```
SELECT * FROM proyectos  
WHERE precio > ALL (SELECT sueldo  
FROM empleados WHERE codigo_proyec = num_proyec);
```

Si la condición se relaja, y sólo se pide que la condición sólo ocurra para algunos empleados, entonces sería:

```
SELECT * FROM proyectos  
WHERE precio > SOME (SELECT sueldo  
FROM empleados WHERE codigo_proyec = num_proyec);
```


Order by

- Para ordenar los resultados de una consulta se utiliza la cláusula **ORDER BY**:

```
SELECT nombre_columnas_a seleccionar  
FROM tabla_a_consultar  
[WHERE condiciones]  
ORDER BY columna_según_la_cual_se_quiere_ordenar [DESC]  
[, col_ordenación [DESC]...];
```

- Por defecto los resultados se ordenan de manera ascendente. Así si queremos realizar una ordenación descendente se debe indicar usando la cláusula DESC.

Order by

- Por ejemplo si queremos ordenar los empleados por orden alfabético ascendente de a cuerdo a su nombre y descendente de acuerdo a su sueldo:

```
SELECT * FROM empleados  
ORDER BY nombre_empl, sueldo DESC;
```