

Tema 5

1. Consultas básicas SQL

Consulta de datos

```
SELECT [ALL | DISTINCT]  
[column1, ..column | *]  
FROM tabla1, .., tablan  
[WHERE condicion]  
[ORDER BY column1 [DESC | ASC],  
column1 [DESC | ASC],...]
```

Consulta de datos

- FROM, obligatoria, se especifican las tablas donde está la información necesaria.
 - SELECT * FROM emp;
- Si el usuario no es propietario de las tablas
 - SELECT * FROM usuario.emp;
- Se puede poner alias a las tablas
 - SELECT * FROM emp e;

Consulta de datos

- WHERE condición
 - expresión **operador** expresión
- Operadores
 - IN, NOT IN, BETWEEN, NOT BETWEEN, LIKE
- Condiciones múltiples
 - AND, OR, NOT y ()
- SELECT ename FROM EMP where salario > 1000;
 - WHERE (salario > 1000) AND (com IS NOT NULL)
 - WHERE (com IS NULL) AND (UPPER(ename) = 'MARY')

Consulta de datos

- ORDER BY colum [ASC |DESC], ...
 - Por defecto es ASC.
 - SELECT ename FROM emp ORDER BY sal * 12;
- si tiene 2 o más criterios de ordenación, el principal es el más a la izquierda y dentro de este los siguientes.

```
SELECT ename, job, sal * 12 AS salario_anual  
FROM emp  
ORDER BY job ASC, sal * 12 DESC;
```

Consulta de datos

- ALL, recupera todas las filas, opción por defecto.
- DISTINCT, recupera las filas distintas.

```
SELECT DISTINCT deptno  
FROM emp;
```

```
SELECT DISTINCT sal  
FROM emp;
```

Proyección

```
SELECT [ALL|DISTINCT]
```

```
column1, ..column
```

```
FROM tabla1,...tablan;
```

Todas las columnas

```
SELECT empno, ename,...
```

```
FROM EMP;
```

```
SELECT * FROM emp;
```

Algunas columnas

```
SELECT deptno, dname from dept;
```

Selección de filas

- Es necesaria la clausula: WHERE condicion.

```
SELECT ename, job  
FROM emp  
WHERE deptno = 10  
ORDER BY job, ename;
```

```
SELECT *  
FROM emp  
WHERE job = 'CLERK' AND deptno = 10  
ORDER BY deptno, ename;
```


ALIAS a columnas

- Si el nombre (aparece como cabecera) de la columna resulta demasiado largo, corto o críptico, se puede hacer uso de los alias.

```
SELECT ename “nombre empleado”,  
       job “trabajo empleado”  
FROM emp
```

Comparación de cadenas

- Usaremos = , para comparar cadenas enteras.

```
SELECT * FROM emp WHERE  
ename='PEPE PEREZ'
```

- Usaremos LIKE para comparar con un patron

```
SELECT * FROM emp WHERE  
ename LIKE 'P%'
```

Comparación de cadenas

LIKE 'ANALYST'

LIKE 'M%'

LIKE '%X%'

LIKE '__M'

LIKE 'N_'

LIKE '_R%'

Se distinguen mayúsculas de las minúsculas

NULL, NOT NULL

NULL, columna vacía

```
SELECT ename
```

```
FROM emp
```

```
WHERE comm IS NULL;
```

NOT NULL, columna llena

```
SELECT ename
```

```
FROM emp
```

```
WHERE comm IS NOT NULL;
```

Pertenencia a un conjunto

- Operador IN. comprueba si una expresión pertenece o no a un conjunto de valores.
- Formato:

Expresion [NOT] IN (lista de valores)

SELECT ename

FROM emp

WHERE deptno IN (10, 20,30);

SELECT ename FROM emp

WHERE deptno NOT IN (30, 40);

Pertenencia a un rango

- **BETWEEN**, comprueba si un valor o no está comprendido en un rango de valores.
- Formato:

**Expresion [NOT] BETWEEN valor_inicial
AND valor_final**

```
SELECT ename FROM emp  
WHERE sal BETWEEN 1000 and 2000;
```

```
SELECT ename FROM emp WHERE sal  
NOT BETWEEN 1000 AND 2000;
```

Combinación AND, OR y NOT

- Orden de prioridad en la evaluación :
NOT AND OR
- Es aconsejable siempre usar paréntesis
SELECT ename, sal, deptno from emp
WHERE sal > 2000
AND (deptno = 10 OR deptno = 20);