

---

## 2. RESTRICCIÓN Y CALIFICACIÓN DE DATOS.

---

- 2.1 La clausula WHERE
- 2.2 Operadores de comparación
- 2.3 Otros operadores de comparacion
  - 2.3.1 BETWEEN
  - 2.3.2 IN
  - 2.3.3 IS NULL
  - 2.3.4 LIKE
- 2.4 Operadores lógicos
- 2.5 Reglas de precedencia
- 2.6 Cláusula ORDER BY

### **2.1.- LA CLAUSULA WHERE.**

Podemos restringir las filas recuperadas usando la cláusula WHERE. Una clausula WHERE contiene una condición que se debe cumplir y se escribe a continuación de la clausula FROM.

Sintaxis:

```
SELECT [ DISTINCT ] { * | column [ alias ] , ... }  
FROM tabla  
[ WHERE condición ];
```

En las condiciones, las fechas y las cadenas de caracteres van entre comillas simples (son sensibles a mayúsculas y minúsculas)

Ejemplo 1:

Realizar una consulta en la que se visualicen el nombre del empleado, el trabajo y el código del departamento para aquellos empleados en los que el JOB = CLERK de la tabla EMP.

```
SELECT ENAME, JOB, DEPTNO  
FROM EMP  
WHERE JOB = 'CLERK';
```

Ejemplo 2:

Realizar una consulta en la que se visualicen los trabajadores que hayan empezado a trabajar el 17-12-1980

```
SELECT ENAME  
FROM EMP  
WHERE HIREDATE = '17-12-1980';
```

El formato de la fecha debe coincidir con el formato de la fecha que tenga establecido el sistema, en este caso el formato es DD-MM-AA

---

## **2.2.- OPERADORES DE COMPARACIÓN**

Los operadores de comparación se utilizan en las condiciones que comparan una expresión con otra. Se suele utilizar con la cláusula WHERE. Estos operadores son:

=	igual a
<	menor que
<=	menor igual a
>	mayor que
>=	mayor igual a
<>	no igual a o distinto a

Ejemplo:

Realizar una consulta en la que se recupere de la tabla EMP, el nombre, el salario y la comisión, cuando el salario es menor o igual que la comisión.

```
SELECT ENAME, SAL, COMM
FROM EMP
WHERE SAL<= COMM;
```

## **2.3.- OTROS OPERADORES DE COMPARACIÓN**

### **2.3.1.- BETWEEN ...AND...**

---

Visualiza filas dentro de un rango de valores, el rango tienen un valor inferior y otro superior.

Los valores especificados por BETWEEN, se incluyen en la operación. Se pueden usar con BETWEEN fechas y cadenas, siempre entre comillas simples.

Ejemplo:

Devolver las filas de empleados cuyo salario esté comprendido entre 1000 y 3000 de la tabla EMP.

```
SELECT ENAME
FROM EMP
WHERE SAL BETWEEN 1000 AND 3000;
```

### **2.3.2.- IN**

---

Selecciona filas de una lista que se asigna. Los literales y las fechas van entre comillas simples.

Ejemplo:

Visualiza todos los trabajadores que sean ANALYST y CLERK de la tabla EMP.

```
SELECT *
```

---

```
FROM EMP
WHERE JOB IN ('ANALYST', 'CLERK');
```

---

### 2.3.3.- IS NULL

---

Recupera aquellas filas en las que el valor que se obtiene es de tipo NULL

Ejemplo 1:

Realizar una consulta de todos los empleados que no tengan comisión.

```
SELECT ENAME
FROM EMP
WHERE COMM IS NULL;
```

Ejemplo 2:

Realizar una consulta de los empleados que no tengan jefe

```
SELECT *
FROM EMP
WHERE MGR IS NULL;
```

---

### 2.2.4.- LIKE

---

Realiza la búsqueda de cadenas de caracteres que incluyen comodines.

Los comodines son:

%	Denota cero o varios caracteres
_	Denota un solo carácter

Ejemplo:

Realizar una consulta que devuelva el nombre de los empleados de la tabla EMP, cuyo tercer carácter sea una S.

```
SELECT * FROM EMP
WHERE ENAME LIKE '__R%';
```

---

## 2.4.- OPERADORES LÓGICOS

---

Combinan la ejecución de dos condiciones. Son:

- AND: requiere que ambas condiciones sean VERDADERAS

AND	VERDADERO	FALSO	DESCONOCIDO
VERDADERO	V	F	D
FALSO	F	F	F
DESCONOCIDO	D	F	D

- OR: requiere que alguna condición sea VERDADERA.

OR	VERDADERO	FALSO	DESCONOCIDO
VERDADERO	V	V	V
FALSO	V	F	D
DESCONOCIDO	V	D	D

- NOT: niega la condición.

NOT	VERDADERO	FALSO	DESCONOCIDO
	F	V	D

Ejemplo 1:

Obtener todos los empleados con su número, nombre, puesto y salario en todos aquellos casos en los que el salario sea mayor de 1100 y su oficio CLERK.

```
SELECT EMPNO, ENAME, JOB, SAL
FROM EMP
WHERE SAL > 1100 AND JOB = 'CLERK';
```

Ejemplo 2:

Obtener el nombre, número, puesto y salario de los empleados que cobren más de 1100 o su oficio sea CLERK.

```
SELECT EMPNO, ENAME, JOB, SAL
FROM EMP
WHERE SAL > 1100 OR JOB = 'CLERK';
```

Ejemplo 3:

Obtener el nombre, número, puesto y salario de los empleados que su oficio no sea CLERK ni ANALYST

---

```
SELECT EMPNO, ENAME, JOB, SAL
FROM EMP
WHERE JOB NOT IN ( 'CLERK', 'ANALYST');
```

## **2.5.- REGLAS DE PREFERENCIA**

Orden evaluación	Operador
1	>, >=, <, <=, =, <>, BETWEEN, LIKE, IN, IS NULL
2	NOT
3	AND
4	OR

Para cambiar las preferencias utilizaremos paréntesis.

Ejemplo 1:

Seleccionar los registros de los empleados cuyo oficio sea PRESIDENT y gane más de 1500 dólares o si su oficio es SALESMAN.

```
SELECT ENAME, JOB, SAL
FROM EMP
WHERE JOB= 'SALESMAN' OR JOB= 'PRESIDENT' AND SAL>1500;
```

Ejemplo 2:

Seleccionar los registros de los empleados cuyo oficio sea PRESIDENT o SALESMAN y gane más de 1500.

```
SELECT ENAME, JOB, SAL
FROM EMP
WHERE (JOB = 'SALESMAN' OR JOB = 'PRESIDENT') AND SAL>1500;
```

## **2.6.- LA CLÁUSULA ORDER BY**

Ordena los registros por un determinado campo. Por defecto sale ordenado por la clave en orden ascendente

Sintaxis:

```
SELECT [ DISTINCT] {*,COLUMNA[ALIAS]...}
FROM tabla
[WHERE condición]
[ORDER BY {columna, expr, alias} [ASC|DESC]];
```

Ejemplo 1:

---

Ordenar la tabla empleado por la fecha, ascendentemente

```
SELECT * FROM EMP  
ORDER BY HIREDATE;
```

Ejemplo 2:

```
SELECT ENAME, SAL*12 SALANUAL  
FROM EMP  
ORDER BY ENAME DESC;
```

```
SELECT ENAME, SAL*12 SALANUAL  
FROM EMP  
ORDER BY SAL DESC;
```

```
SELECT ENAME, SAL*12 SALANUAL  
FROM EMP  
ORDER BY SALANUAL DESC;
```

```
SELECT ENAME, SAL*12 SALANUAL  
FROM EMP  
ORDER BY 2 DESC;
```

```
SELECT ENAME, SAL*12 SALANUAL  
FROM EMP  
ORDER BY ENAME DESC, SAL DESC;
```