

NULL

La comprobación de si un campo es nulo, o no, debe hacerse con:

- IS NULL
 - SELECT * FROM emp WHERE comm IS NOT NULL
- IS NOT NULL
 - SELECT * FROM emp WHERE mgr IS NULL

¡Nunca comprobarlo con = !

~~■ SELECT * FROM emp WHERE comm = NULL~~

¡No nos devolvería resultados!

Aritmética de NULL

Un valor nulo es un valor desconocido, por tanto las operaciones que involucren a NULL tendrán como resultado un valor NULL

- $1 + \text{NULL} \rightarrow \text{NULL}$

- $1 - \text{NULL} \rightarrow \text{NULL}$

- $1 * \text{NULL} \rightarrow \text{NULL}$

- $1 / \text{NULL} \rightarrow \text{NULL}$

- $\text{NULL} / 1 \rightarrow \text{NULL}$

- $\text{NULL} = \text{NULL} \rightarrow \text{NULL}$

- $\text{NULL LIKE 'x'} \rightarrow \text{NULL}$

- $\text{NULL} = \text{'x'} \rightarrow \text{NULL}$

Predicados compuestos

Son la unión de dos o más predicados simples mediante **AND**, **OR** y **NOT**

Al existir una lógica de tres valores => considerar el efecto del valor **NULL**

NOT X	X	Y	OR	AND
F	T	T	T	T
T	F	F	F	F
	T	F	T	F
	F	T	T	F
?	?	?	?	?
	T	?	T	?
	?	T	T	?
	F	?	?	F
	?	F	?	F

COALESCE

Devuelve el primer valor no nulo en una lista

- `COALESCE(a, b, c, ...)`
- `SELECT COALESCE(comm, 0) FROM emp;`

Aplicación real

- `SELECT sal + comm FROM emp; => NO FUNCIONA!`
- `SELECT sal + COALESCE(comm, 0) FROM emp;`

Predicados compuestos

Las cláusulas WHERE permiten aplicar múltiples condiciones

```
SELECT ename, sal, comm, sal + coalesce(comm, 0)
FROM emp
WHERE
    sal BETWEEN 1000 AND 3000
    AND deptno IN (10, 30)
    AND (
        ename LIKE '%A%'
        OR ename LIKE 'JONES'
    );
```

Funciones

Nos permiten aplicar operaciones sobre los valores almacenados

- ABS --> Valor absoluto
 - `SELECT ABS(sal * -1) FROM emp;`
- SQRT -> Raíz cuadrada
 - `SELECT SQRT(144) FROM emp;`
- POWER(x, y) -> x elevado a y
 - `SELECT ABS(12, 2) FROM emp;`

Funciones - Strings

- UPPER --> Mayúsculas
 - `SELECT UPPER(ename) FROM emp;`
- LOWER --> Minúsculas
 - `SELECT LOWER(ename) FROM emp;`
- SUBSTR(string, start, length) -> Substring
 - `SELECT SUBSTR(job, 2, 5) FROM emp;`
- CONCAT(a, b) -> Concatenar
 - `SELECT CONCAT('Hello', 'World') FROM emp;`

Fechas

Las fechas son un tipo de dato que requiere un tratamiento especial

- DATE → YYYY-MM-DD
- TIME → HH:MM:SS
- DATE_TIME → YYYY-MM-DD HH:MM:SS
- ```
SELECT *
FROM emp e
WHERE e.hiredate > '1982-01-01'
```



# Funciones - Fecha

Nos permiten aplicar operaciones sobre los valores almacenados

- `CURRENT_DATE()`
- `CURRENT_TIME()`
- `CURRENT_TIMESTAMP()`
- ...

[https://www.w3schools.com/sql/sql\\_ref\\_mysql.asp](https://www.w3schools.com/sql/sql_ref_mysql.asp)

# Funciones de agrupamiento

Nos permiten aplicar operaciones sobre toda la columna

- COUNT --> Número de elementos

- SELECT COUNT(\*) FROM emp;

- MIN --> Valor mínimo

- SELECT MIN(sal) from emp;

- MAX --> Valor máximo

- SELECT MAX(sal) from emp;

- AVG --> Valor medio

- SELECT AVG(sal) from emp;