

## 2. Sentencias básicas SQL

# Temario

## **2. Sentencias básicas de SQL.**

- 2.1 Selección de los datos a mostrar (Clausula Select).
- 2.2 Operaciones aritméticas
- 2.3 El valor Null.
- 2.4 Uso de Alias.
- 2.5 Concatenación.

## 2.1 Clausula SELECT

- En ocasiones se denomina sentencia de correspondencia o bloque **select-from-where** y tiene la siguiente forma

```
SELECT <lista de atributos o campos>  
FROM <tabla o lista de tablas>  
WHERE <condición>
```

Donde:

- <lista de atributos> es una lista de nombres de atributos cuyos valores van a ser recuperados por la consulta.
- <lista de tablas> es una lista de nombres de las relaciones necesarias para procesar la consulta.
- <condición> es una expresión condicional (booleana) que identifica las tuplas que van a ser recuperadas por la consulta.

## 2.2 Operaciones aritméticas

```
SELECT EMP.ENAME, 'Tarifa anual:', EMP.SAL* 12 FROM EMP WHERE EMP.JOB = 'MANAGER' ;
```

Results Messages

	ENAME	(No column name)	(No column name)
1	JONES	Tarifa anual:	35700.00
2	BLAKE	Tarifa anual:	34200.00
3	CLARK	Tarifa anual:	29400.00

## 2.2 Operaciones aritméticas

```
select ename, SAL, comm, SAL*comm from EMP;
```

<

Results Messages

	ename	SAL	comm	(No column name)
1	SMITH	800.00	NULL	NULL
2	ALLEN	1600.00	300.00	480000.0000
3	WARD	1250.00	500.00	625000.0000
4	JONES	2975.00	NULL	NULL
5	MARTIN	1250.00	1400.00	1750000.0000
6	BLAKE	2850.00	NULL	NULL
7	CLARK	2450.00	NULL	NULL
8	SCOTT	3000.00	NULL	NULL
9	KING	5000.00	NULL	NULL
10	TURNER	1500.00	0.00	0.0000
11	ADAMS	1100.00	NULL	NULL
12	JAMES	950.00	NULL	NULL
13	FORD	3000.00	NULL	NULL
14	MILLER	1300.00	NULL	NULL

## 2.3 El valor NULL

- El valor nulo es la ausencia de valor en un campo o intersección de renglón columna
- Un valor nulo no es lo mismo que cero o espacios en blanco
- Las columnas de cualquier tipo pueden contener valores nulos, a menos que dichas columnas fueran definidas como NOT NULL o PRIMARY KEY

```
SELECT ename, job, comm  
FROM emp;
```

ename	job	comm
KING	PRESIDENT	NULL
ALLEN	SALESMAN	300.00
WARD	SALESMAN	500.00
JONES	MANAGER	NULL
MARTIN	SALESMAN	1400.00
...		
(14 row(s) affected)		

## 2.3 El valor NULL

- En la columna COMM de la tabla EMP, se puede notar que solo aquellos que son SALESMAN(vendedores) pueden tener una comisión
- TURNER, que es un vendedor no gana comisión alguna, pero esta columna tiene cero y no un valor nulo
- Si el contenido de una columna es nulo dentro de una expresión aritmética, el resultado es NULL
- Sin embargo si se divide un número entre un valor nulo, el resultado es nulo

## 2.3 El valor NULL

```
SQL> select ename,comm from EMP WHERE COMM IS NULL;
```

<
Results Messages

	ename	comm
1	SMITH	NULL
2	JONES	NULL
3	BLAKE	NULL
4	CLARK	NULL
5	SCOTT	NULL
6	KING	NULL
7	ADAMS	NULL
8	JAMES	NULL
9	FORD	NULL
10	MILLER	NULL



## 2.3 El valor NULL

- ISNULL(atributo, Valor)
  - Reemplaza el valor del atributo con Valor, si este es NULO

```
SELECT ename, sal, comm, (sal * 12) + ISNULL(comm,0)  
FROM emp;
```

ENAME	SAL	COMM	
SMITH	800.00		9600.00
ALLEN	1600.00	300.00	19500.00
WARD	1250.00	500.00	15500.00
JONES	2975.00		35700.00
MARTIN	1250.00	1400.00	16400.00
....			
14 rows selected.			

## 2.3 El valor NULL

```
select ename,comm, ISNULL(comm,0.0),COMM from EMP WHERE COMM IS NULL;  
select * from EMP;
```

&lt;



Results



Messages

	ename	comm	(No column name)	COMM
1	SMITH	NULL	0.00	NULL
2	JONES	NULL	0.00	NULL
3	BLAKE	NULL	0.00	NULL
4	CLARK	NULL	0.00	NULL
5	SCOTT	NULL	0.00	NULL
6	KING	NULL	0.00	NULL
7	ADAMS	NULL	0.00	NULL
8	JAMES	NULL	0.00	NULL
9	FORD	NULL	0.00	NULL
10	MILLER	NULL	0.00	NULL

## 2.4 Uso de alias

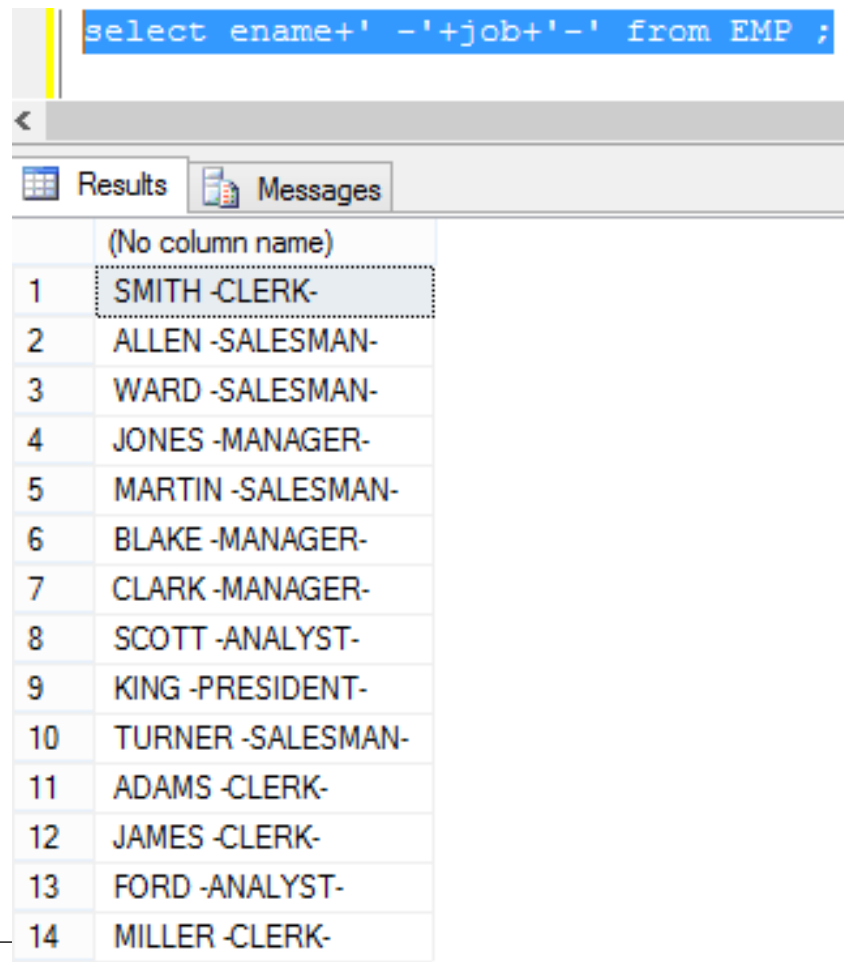
- Calificar las columnas con los nombres de las tablas puede consumir tiempo, sobre todo cuando las tablas tienen nombres largos.
- Se pueden utilizar alias para re-nombrar las tablas
- Sólo un alias por tabla
- Las primeras dos líneas son equivalentes
- La tercer línea esta incorrecta

```
SELECT EMP.ENAME, 'Tarifa anual:', EMP.SAL* 12 FROM EMP WHERE EMP.JOB = 'MANAGER' ;  
SELECT E.ENAME, 'Tarifa anual:', E.SAL* 12 FROM EMP e WHERE E.JOB = 'MANAGER' ;  
SELECT EMP.ENAME, 'Tarifa anual:', E.SAL* 12 FROM EMP e WHERE E.JOB = 'MANAGER' ;
```

## 2.5 Concatenación

- Sintaxis:

`expression + expression`



The screenshot shows a SQL query execution window. The query entered is `select ename+'-'+job+'-' from EMP ;`. The results are displayed in a table with 14 rows, each showing a concatenated string of an employee's name and job title, separated by hyphens. The first row is highlighted.

	(No column name)
1	SMITH -CLERK-
2	ALLEN -SALESMAN-
3	WARD -SALESMAN-
4	JONES -MANAGER-
5	MARTIN -SALESMAN-
6	BLAKE -MANAGER-
7	CLARK -MANAGER-
8	SCOTT -ANALYST-
9	KING -PRESIDENT-
10	TURNER -SALESMAN-
11	ADAMS -CLERK-
12	JAMES -CLERK-
13	FORD -ANALYST-
14	MILLER -CLERK-

## 2.6 Renombrado

- Para renombrar una columna se utiliza AS seguido del nuevo nombre de la columna

```
select ename,comm, ISNULL(comm,0.0) * SAL as nuevaCom,COMM* SAL as ComNull from EMP WHERE COMM IS NULL;  
select * from EMP;
```

	ename	comm	nuevaCom	ComNull
1	SMITH	NULL	0.0000	NULL
2	JONES	NULL	0.0000	NULL
3	BLAKE	NULL	0.0000	NULL
4	CLARK	NULL	0.0000	NULL
5	SCOTT	NULL	0.0000	NULL
6	KING	NULL	0.0000	NULL
7	ADAMS	NULL	0.0000	NULL
8	JAMES	NULL	0.0000	NULL
9	FORD	NULL	0.0000	NULL
10	MILLER	NULL	0.0000	NULL

## 2.6 Renombrado

```
select (ename+'-'+job) as informacion from emp;
```

Results		Messages
	informacion	
1	SMITH-CLERK	
2	ALLEN-SALESMAN	
3	WARD-SALESMAN	
4	JONES-MANAGER	
5	MARTIN-SALESMAN	
6	BLAKE-MANAGER	
7	CLARK-MANAGER	
8	SCOTT-ANALYST	
9	KING-PRESIDENT	
10	TURNER-SALESMAN	
11	ADAMS-CLERK	
12	JAMES-CLERK	
13	FORD-ANALYST	
14	MILLER-CLERK	

## 2.6 Renombrado

```
select (ename+job) as informacion from emp;
```

	informacion
1	SMITHCLERK
2	ALLENSALESMAN
3	WARDSALESMAN
4	JONESMANAGER
5	MARTINSALESMAN
6	BLAKEMANAGER
7	CLARKMANAGER
8	SCOTTANALYST
9	KINGPRESIDENT
10	TURNERSALESMAN
11	ADAMSCLERK
12	JAMESCLERK
13	FORDANALYST
14	MILLERCLERK