

SQL



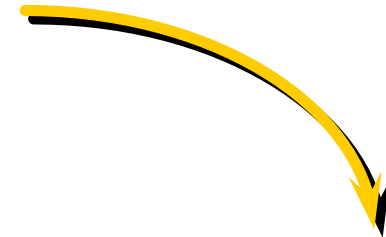
Seleccionando y Ordenando Datos

Seleccionando y ordenando datos

EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7698	BLAKE	MANAGER		30
7782	CLARK	MANAGER		10
7566	JONES	MANAGER		20
...				

Mostrar los empleados
que trabajan en el
departamento 10...



EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7782	CLARK	MANAGER		10
7934	MILLER	CLERK		10

Clausula WHERE

– El **WHERE** va después del **FROM**.

```
SELECT      [DISTINCT] {*, column [alias], ...}  
FROM        table  
[WHERE      condition(s)];
```

Clausula WHERE

```
SQL> SELECT ename, job, deptno  
2 FROM emp  
3 WHERE job='CLERK';
```

ENAME	JOB	DEPTNO
JAMES	CLERK	30
SMITH	CLERK	20
ADAMS	CLERK	20
MILLER	CLERK	10

Cadenas de caracteres y Fechas

- Las cadenas de caracteres y los valores de los campos fechas deben encerrarse entre comillas sencillas,
- Los valores de las cadenas de caracteres son sensibles a las mayúsculas
- Los valores de las fechas son sensibles a los formatos establecidos
- El formato default de un campo fecha es ' DD-MON-YY '

```
SQL> SELECT  ename, job, deptno  
2 FROM      emp  
3 WHERE     ename = 'JAMES';
```

Operadores de Comparación

Operador	
=	Igual a
>	Mayor a
>=	Mayor o igual a
<	Menor que
<=	Menor o igual que
<>	diferente

Ejemplo

```
SQL> SELECT ename, sal, comm  
2   FROM emp  
3   WHERE sal<=comm;
```

ENAME	SAL	COMM
MARTIN	1250	1400

Otros Operadores

Operador	
BETWEEN ...AND...	Entre dos valores(inclusivo)
IN(list)	Incluido en una lista de valores
LIKE	Coincide con un patrón de caracteres
IS NULL	Es un valor Nulo

Operador BETWEEN

Utilícelo para mostrar renglones cuyo valor de algún atributo se encuentre en un rango de valores

```
SQL> SELECT  ename, sal
2  FROM      emp
3  WHERE      sal BETWEEN 1000 AND 1500;
```

ENAME	SAL		
-----	-----	Límite inicial	Límite Final
MARTIN	1250		
TURNER	1500		
WARD	1250		
ADAMS	1100		
MILLER	1300		

1.4 Lenguaje de Consulta Estructurado

Operador IN

Utilízelo para mostrar renglones cuyo valor de algún atributo se pertenezca a una lista de valores

```
SQL> SELECT empno, ename, sal, mgr
2 FROM emp
3 WHERE mgr IN (7902, 7566, 7788);
```

EMPNO	ENAME	SAL	MGR
7902	FORD	3000	7566
7369	SMITH	800	7902
7788	SCOTT	3000	7566
7876	ADAMS	1100	7788

Operador Like

Utilice el operador LIKE para realizar búsquedas con caracteres comodín:

% Denota cero o muchos caracteres

_ Denota un carácter

```
SQL> SELECT  ename  
2  FROM      emp  
3  WHERE     ename LIKE 'S%';
```

Operador LIKE

- Se puede combinar el patrón de búsqueda de los caracteres

```
SQL> SELECT  ename  
2  FROM      emp  
3  WHERE     ename LIKE '_A%';
```

ENAME

JAMES

WARD

Operador IS NULL

Prueba si un valor es nulo

```
SQL> SELECT  ename, mgr  
      2  FROM    emp  
      3  WHERE   mgr IS NULL;
```

ENAME	MGR
-----	-----
KING	

Operadores Lógicos

Operador	
AND	Regresa TRUE si ambos componentes de la condición son verdaderos
OR	Regresa TRUE si cualquiera de los componentes es verdadero
NOT	Regresa TRUE si la condición siguiente es Falsa

Operador AND

Las dos condiciones debe ser verdaderas

```
SQL> SELECT empno, ename, job, sal
2   FROM emp
3   WHERE sal >= 1100
4   AND job = 'CLERK';
```

EMPNO	ENAME	JOB	SAL
7876	ADAMS	CLERK	1100
7934	MILLER	CLERK	1300

Operator OR

```
SQL> SELECT empno, ename, job, sal  
2 FROM emp  
3 WHERE sal >= 1100  
4 OR job = 'CLERK';
```

EMPNO	ENAME	JOB	SAL
7839	KING	PRESIDENT	5000
7698	BLAKE	MANAGER	2850
7782	CLARK	MANAGER	2450
7566	JONES	MANAGER	2975
7654	MARTIN	SALESMAN	1250

...

14 rows selected.

Operator NOT

```
SQL> SELECT ename, job  
2 FROM emp  
3 WHERE job NOT IN ('CLERK', 'MANAGER', 'ANALYST');
```

ENAME	JOB
KING	PRESIDENT
MARTIN	SALESMAN
ALLEN	SALESMAN
TURNER	SALESMAN
WARD	SALESMAN


Reglas de Precedencia

Orden de evaluación	Operador
1	Todos los operadores de comparación
2	NOT
3	AND
4	OR

El uso de paréntesis invalida las reglas de precedencia.

Reglas de Precedencia


```
SQL> SELECT ename, job, sal
2  FROM emp
3  WHERE job='SALESMAN'
4  OR job='PRESIDENT'
5  AND sal>1500;
```



ENAME	JOB	SAL
KING	PRESIDENT	5000
MARTIN	SALESMAN	1250
ALLEN	SALESMAN	1600
TURNER	SALESMAN	1500
WARD	SALESMAN	1250

Reglas de Precedencia

```
SQL> SELECT ename, job, sal
2 FROM emp
3 WHERE job='SALESMAN'
4 OR job='PRESIDENT'
5 AND sal>1500;
```



ENAME	JOB	SAL
-----	-----	-----
KING	PRESIDENT	5000
MARTIN	SALESMAN	1250
ALLEN	SALESMAN	1600
TURNER	SALESMAN	1500
WARD	SALESMAN	1250

Clausula ORDER BY

– Ordena los renglones

- ASC: orden ascendente, default
- DESC: orden descendente

– El ORDER BY va al final de la sentencia SELECT.

```
SQL> SELECT      ename, job, deptno, hiredate  
2  FROM          emp  
3  ORDER BY hiredate;
```

ENAME	JOB	DEPTNO	HIREDATE
SMITH	CLERK	20	17-DEC-80
ALLEN	SALESMAN	30	20-FEB-81
...			

14 rows selected.

Ordenando en orden descendiente

```
SQL> SELECT      ename, job, deptno, hiredate
  2  FROM          emp
  3  ORDER BY hiredate DESC;
```

ENAME	JOB	DEPTNO	HIREDATE
ADAMS	CLERK	20	12-JAN-83
SCOTT	ANALYST	20	09-DEC-82
MILLER	CLERK	10	23-JAN-82
JAMES	CLERK	30	03-DEC-81
FORD	ANALYST	20	03-DEC-81
KING	PRESIDENT	10	17-NOV-81
MARTIN	SALESMAN	30	28-SEP-81
...			

14 rows selected.

Ordenando Columnas con alias

```
SQL> SELECT empno, ename, sal*12 annsal  
2 FROM emp  
3 ORDER BY annsal;
```

EMPNO	ENAME	ANNSAL
7369	SMITH	9600
7900	JAMES	11400
7876	ADAMS	13200
7654	MARTIN	15000
7521	WARD	15000
7934	MILLER	15600
7844	TURNER	18000

...

14 rows selected.

Ordenando columnas múltiples

- Se basa en el orden de la lista después del ORDER BY

```
SQL> SELECT      ename, deptno, sal  
  2  FROM        emp  
  3  ORDER BY deptno, sal DESC;
```

ENAME	DEPTNO	SAL
KING	10	5000
CLARK	10	2450
MILLER	10	1300
FORD	20	3000
...		

14 rows selected.