

### Índice

1. Introducción, tipos de combinación	2
2. Combinación común (JOIN)	3
3. Combinación no común	6
4. Combinación con más de dos tablas	7
5. Auto combinación	8
6. Combinación exterior	10
7. OPERADORES DE CONJUNTO	11
UNION	
INTERSECT	
MINUS	
8. Ejercicios propuestos	15

## 1. INTRODUCCIÓN

Hasta el momento sólo hemos podido seleccionar datos de una única tabla. Con las consultas anidadas podíamos seleccionar filas de una tabla que cumpliesen una condición almacenada en otra tabla, pero nunca se podía seleccionar y visualizar a la vez, datos que pertenecieran a más de una tabla. Pues bien, por medio de la combinación de tablas se puede **seleccionar datos de tablas diferentes**.

La combinación de dos tablas consiste en realizar el **producto cartesiano de las dos**. El resultado de unir dos tablas será otra que tiene por columnas la unión de las columnas de las dos tablas y por filas el producto cartesiano de las dos, es decir, que **cada fila de la primera tabla se concatena con todas las filas de la otra**.

Siempre que realicemos la combinación de dos tablas tendrá que **haber una condición de combinación por medio de la cual se puedan relacionar los datos de las dos tablas y se restrinja así, aquellas filas del producto cartesiano que realmente nos interesan**. Un simple producto hace aparecer filas que contienen información falsa.

Los distintos tipos de combinación que existen son:

- 1.- **Producto cartesiano**. No tiene restricciones y no daría resultados coherentes.
- 2.- **Combinación común**. La condición de combinación contiene un operador de igualdad.
- 3.- **Combinación no común**. La condición de combinación contiene un operador que no es el de igualdad.
- 4.- **Autocombinación**. Se realiza la combinación de una tabla consigo misma.
- 5.- **Combinación exterior**. Selecciona las filas de una tabla que no tienen correspondencia con alguna de la otra.

## 2. COMBINACIÓN COMUN

La sintaxis de la combinación de tablas es:

```
SELECT tabla1.col1, tabla2.col1, .....  
FROM tabla1, tabla2  
WHERE condición de combinación.
```

Se debe tener en cuenta:

- No importa el orden en el que las columnas sean seleccionadas, sólo las columnas que se seleccionan.
- Sólo es obligatorio poner el nombre de la tabla antes que el de la columna cuando el nombre de ésta es ambiguo, es decir, cuando este nombre de columna aparece en más de una tabla de las seleccionadas.
- El orden en el que se especifican las tablas no influye en el resultado, pero sí en el tiempo de respuesta de la consulta. La segunda tabla es la conductora. Cada fila de ésta se concatena con todas las filas de la primera.
- Para que una combinación esté bien hecha siempre deberá tener una condición de combinación.
- Puede usarse la expresión tabla.\* para obtener todas las columnas de una tabla.

Seleccionar el nombre del empleado, la función, el departamento, código de la localidad y el salario de cada empleado de la tabla empleados. El código de la localidad se encuentra en la tabla de departamentos.

Si hacemos la siguiente consulta

```
SELECT last_name, job_id, employee.department_id, department.department_id, location_id, salary
FROM employee, department;
```

Obtenemos como salida

Resultado de la Consulta x

SQL | Se han recuperado 300 filas en 0,187 segundos

	LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_ID_1	LOCATION_ID	SALARY
4	DENNIS	671	23	10	122	2750
5	BAKER	671	14	10	122	2200
6	WARD	670	30	10	122	1250
7	PETERS	670	13	10	122	1250
8	SHAW	670	13	10	122	1250
9	DUNCAN	670	23	10	122	1250
10	LANGE	670	23	10	122	1250
11	JONES	671	20	10	122	2975
12	ALBERTS	671	12	10	122	3000
13	PORTER	670	13	10	122	1250
14	LEWIS	668	24	10	122	1800
15	MARTIN	670	30	10	122	1250
16	SOMMERS	668	34	10	122	1850
17	BLAKE	671	30	10	122	2850
18	CLARK	671	10	10	122	2450
19	SCOTT	669	20	10	122	3000
20	WEST	670	23	10	122	1500
21	FISHER	669	12	10	122	3000
22	ROSS	670	43	10	122	1300
23	KING	672	10	10	122	5000
24	TURNER	670	30	10	122	1500
25	ADAMS	667	20	10	122	1100
26	JAMES	667	30	10	122	950
27	FORD	669	20	10	122	3000

	LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_ID_1	LOCATION_ID	SALARY
34	ALLEN	670	30	20	124	1600
35	DOYLE	671	13	20	124	2850
36	DENNIS	671	23	20	124	2750
37	BAKER	671	14	20	124	2200
38	WARD	670	30	20	124	1250
39	PETERS	670	13	20	124	1250
40	SHAW	670	13	20	124	1250
41	DUNCAN	670	23	20	124	1250
42	LANGE	670	23	20	124	1250
43	JONES	671	20	20	124	2975
44	ALBERTS	671	12	20	124	3000
45	PORTER	670	13	20	124	1250
46	LEWIS	668	24	20	124	1800
47	MARTIN	670	30	20	124	1250
48	SOMMERS	668	34	20	124	1850
49	BLAKE	671	30	20	124	2850
50	CLARK	671	10	20	124	2450
51	SCOTT	669	20	20	124	3000
52	WEST	670	23	20	124	1500
53	FISHER	669	12	20	124	3000
54	ROSS	670	43	20	124	1300
55	KING	672	10	20	124	5000
56	TURNER	670	30	20	124	1500
57	ADAMS	667	20	20	124	1100

	LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_ID_1	LOCATION_ID	SALARY
70	WARD	670	30	30	123	1250
71	PETERS	670	13	30	123	1250
72	SHAW	670	13	30	123	1250
73	DUNCAN	670	23	30	123	1250
74	LANGE	670	23	30	123	1250
75	JONES	671	20	30	123	2975
76	ALBERTS	671	12	30	123	3000
77	PORTER	670	13	30	123	1250
78	LEWIS	668	24	30	123	1800
79	MARTIN	670	30	30	123	1250
80	SOMMERS	668	34	30	123	1850
81	BLAKE	671	30	30	123	2850
82	CLARK	671	10	30	123	2450
83	SCOTT	669	20	30	123	3000
84	WEST	670	23	30	123	1500
85	FISHER	669	12	30	123	3000
86	ROSS	670	43	30	123	1300
87	KING	672	10	30	123	5000
88	TURNER	670	30	30	123	1500
89	ADAMS	667	20	30	123	1100
90	JAMES	667	30	30	123	950
91	FORD	669	20	30	123	3000
92	ROBERTS	669	12	30	123	2875
93	DOUGLAS	667	12	30	123	800

Y mas datos aún.

Con estos datos que acabo de poner de la salida podemos observar que el único valido que debía haber salido en la primera tabla son CLARK y KING puesto que trabajan en el departamento 10 y el departamento asociado en la tabla department es el 10. Mientras que el resto no coincide.

En la segunda tabla son JONNES, SCOTT y ADAMS por el mismo motivo y en la tercera WARD, MARTIN, BLAKE, TURNER yJAMES. Los demas son datos que no debían haber salido.

Por tanto la select correcta debía ser

```
SELECT last_name, job_id, employee.department_id, location_id, salary
FROM employee, department
WHERE department.department_id=employee.department_id;
```

R	LAST_NAME	R	JOB_ID	R	DEPARTMENT_ID	R	DEPARTMENT_ID_1	R	LOCATION_ID	R	SALARY
1	CLARK		671		10		10		122		2450
2	KING		672		10		10		122		5000
3	MILLER		667		10		10		122		1300
4	DOUGLAS		667		12		12		122		800
5	ROBERTS		669		12		12		122		2875
6	FISHER		669		12		12		122		3000
7	ALBERTS		671		12		12		122		3000
8	JENSEN		667		13		13		122		750
9	DOYLE		671		13		13		122		2850
10	SHAW		670		13		13		122		1250
11	PETERS		670		13		13		122		1250
12	PORTER		670		13		13		122		1250
13	BAKER		671		14		14		122		2200
14	FORD		669		20		20		124		3000
15	JONES		671		20		20		124		2975
16	SMITH		667		20		20		124		800
17	ADAMS		667		20		20		124		1100
18	SCOTT		669		20		20		124		3000
19	DUNCAN		670		23		23		124		1250
20	WEST		670		23		23		124		1500
21	DENNIS		671		23		23		124		2750
22	LANGE		670		23		23		124		1250
23	MURRAY		667		23		23		124		750
24	LEWIS		668		24		24		124		1800

La condición de combinación para saber el código de localidad en la que trabaja cada empleado es que el número de departamento del empleado coincida con el número de departamento de la tabla departamentos.

Puede ocurrir que exista otra condición aparte de la de combinación:

```
SELECT last_name, job_id, employee.department_id, location_id, salary
FROM employee, department
WHERE department.department_id=employee.department_id
AND department.department_id= 20;
```

	LAST_NAME	JOB_ID	DEPARTMENT_ID	LOCATION_ID	SALARY
1	SMITH	667	20	124	800
2	JONES	671	20	124	2975
3	SCOTT	669	20	124	3000
4	ADAMS	667	20	124	1100
5	FORD	669	20	124	3000

### 3. COMBINACIÓN NO COMUN

Además de las combinaciones anteriormente vistas basadas en el operador de igualdad, hay otras basadas en operadores como >, <, <=, >=, !=, BETWEEN...AND. Son las combinaciones no comunes.

Seleccionar el nombre, salario, grado y función de todos los empleados. Un empleado tiene de grado 'grade\_id' si su salario está comprendido entre 'lower\_bound' y 'upper\_bound'.

```
SELECT last_name,salary, grade_id, job_id
FROM employee, salary_grade
WHERE salary BETWEEN lower_bound AND upper_bound;
```

	2	LAST_NAME	2	SALARY	2	GRADE_ID	2	JOB_ID
5		JAMES		950		1		667
6		ADAMS		1100		1		667
7		LANGE		1250		2		670
8		SHAW		1250		2		670
9		PETERS		1250		2		670
10		WARD		1250		2		670
11		DUNCAN		1250		2		670
12		MARTIN		1250		2		670
13		PORTER		1250		2		670
14		ROSS		1300		2		670
15		MILLER		1300		2		667
16		TURNER		1500		3		670
17		WEST		1500		3		670
18		ALLEN		1600		3		670
19		LEWIS		1800		3		668
20		SOMMERS		1850		3		668
21		BAKER		2200		4		671
22		CLARK		2450		4		671
23		DENNIS		2750		4		671
24		BLAKE		2850		4		671
25		DOYLE		2850		4		671
26		ROBERTS		2875		4		669
27		JONES		2975		4		671
28		FORD		3000		4		669
29		FISHER		3000		4		669
30		SCOTT		3000		4		669
31		ALBERTS		3000		4		671
32		KING		5000		5		672

#### 4. COMBINACIÓN DE MÁS DE DOS TABLAS

También se pueden realizar combinaciones de más de dos tablas, resultando como combinación de éstas otra que tendrá por columnas la unión de las columnas de todas las tablas y por filas el producto cartesiano de las filas de todas las tablas.

Se pueden unir tantas tablas como se quiera pero siempre deberá haber (n-1) condiciones de combinación para que la información sea coherente(n=n° de tablas).

Seleccionar el nombre, salario, grado, función y localidad de todos los empleados. Un empleado tiene de grado 'grade\_id' si su salario está comprendido entre 'lower\_bound' y 'upper\_bound'. El informe estará ordenado por el grado.

```
SELECT last_name,salary, grade_id, job_id, regional_group
FROM employee, department, location, salary_grade
WHERE employee.department_id=department.department_id AND department.location_id=location.location_id
and (salary BETWEEN lower_bound AND upper_bound)
ORDER BY grade_id;
```



SQL | Todas las Filas Recuperadas: 32 en 0 segundos

	LAST_NAME	SALARY	GRADE_ID	JOB_ID	REGIONAL_GROUP
1	MURRAY	750	1	667	DALLAS
2	JENSEN	750	1	667	NEW YORK
3	ADAMS	1100	1	667	DALLAS
4	JAMES	950	1	667	CHICAGO
5	SMITH	800	1	667	DALLAS
6	DOUGLAS	800	1	667	NEW YORK
7	PORTER	1250	2	670	NEW YORK
8	ROSS	1300	2	670	BOSTON
9	MARTIN	1250	2	670	CHICAGO
10	DUNCAN	1250	2	670	DALLAS
11	WARD	1250	2	670	CHICAGO
12	PETERS	1250	2	670	NEW YORK
13	MILLER	1300	2	667	NEW YORK
14	LANGE	1250	2	670	DALLAS
15	SHAW	1250	2	670	NEW YORK
16	TURNER	1500	3	670	CHICAGO
17	SOMMERS	1850	3	668	CHICAGO
18	LEWIS	1800	3	668	DALLAS
19	ALLEN	1600	3	670	CHICAGO
20	WEST	1500	3	670	DALLAS
21	BAKER	2200	4	671	NEW YORK
22	CLARK	2450	4	671	NEW YORK
23	DENNIS	2750	4	671	DALLAS
24	BLAKE	2850	4	671	CHICAGO
25	DOYLE	2850	4	671	NEW YORK
26	ROBERTS	2875	4	669	NEW YORK
27	JONES	2975	4	671	DALLAS

## 5. AUTOCOMBINACIÓN

La combinación de una tabla con ella misma se realiza por medio de columnas que contienen la misma información. La combinación será otra tabla que tendrá por columnas dos veces las columnas de la tabla original y por filas el producto cartesiano de la tabla por sí misma.

La operación equivale a que hubiera dos tablas iguales, (no las hay ya que solamente tenemos una copia física de los datos). Todas las columnas son ambiguas, ya que tienen el mismo nombre, por lo que se tendrá que especificar la tabla a la que nos estamos refiriendo. Como la tabla también es igual hay que renombrar una (o las dos), y más tarde hacer mención a las columnas anteponiendo el nuevo nombre de cada tabla.

Seleccionar para cada empleado el número, el nombre, el departamento, el número de su jefe, nombre de su jefe y el departamento al que pertenece

```
SELECT empleado.employee_id as id_empleado, empleado.last_name as nombre_empleado,
empleado.department_id as departamento_empleado, jefe.employee_id as id_jefe,
jefe.last_name as nombre_jefe, jefe.department_id as departamento_jefe
FROM employee empleado, employee jefe
WHERE empleado.manager_id= jefe.employee_id;
```

	ID_EMPLEADO	NOMBRE_EMPLEADO	DEPARTAMENTO_EMPLEADO	ID_JEFE	NOMBRE_JEFE	DEPARTAMENTO_JEFE
1	7950	JENSEN	13	7505	DOYLE	13
2	7820	ROSS	43	7505	DOYLE	13
3	7600	PORTER	13	7505	DOYLE	13
4	7557	SHAW	13	7505	DOYLE	13
5	7555	PETERS	13	7505	DOYLE	13
6	7564	LANGE	23	7506	DENNIS	23
7	7560	DUNCAN	23	7506	DENNIS	23
8	7954	MURRAY	23	7506	DENNIS	23
9	7789	WEST	23	7506	DENNIS	23
10	7676	SOMMERS	34	7507	BAKER	14
11	7609	LEWIS	24	7507	BAKER	14
12	7788	SCOTT	20	7566	JONES	20
13	7902	FORD	20	7566	JONES	20
14	7799	FISHER	12	7569	ALBERTS	12
15	7916	ROBERTS	12	7569	ALBERTS	12
16	7654	MARTIN	30	7698	BLAKE	30
17	7521	WARD	30	7698	BLAKE	30
18	7499	ALLEN	30	7698	BLAKE	30
19	7900	JAMES	30	7698	BLAKE	30
20	7844	TURNER	30	7698	BLAKE	30
21	7934	MILLER	10	7782	CLARK	10
22	7876	ADAMS	20	7788	SCOTT	20
23	7919	DOUGLAS	12	7799	FISHER	12
24	7505	DOYLE	13	7839	KING	10
25	7782	CLARK	10	7839	KING	10
26	7698	BLAKE	30	7839	KING	10
27	7506	DENNIS	23	7839	KING	10
28	7569	ALBERTS	12	7839	KING	10

Seleccionar para cada empleado el número, el nombre, el departamento, el número de su jefe, nombre de su jefe, el departamento al que pertenece y la localidad donde se encuentra la empresa

```
SELECT empleado.employee_id as id_empleado, empleado.last_name as nombre_empleado,
empleado.department_id as departamento_empleado, jefe.employee_id as id_jefe,
jefe.last_name as nombre_jefe, jefe.department_id as departamento_jefe, regional_group
FROM employee empleado, employee jefe, department, location
WHERE empleado.manager_id= jefe.employee_id and jefe.department_id=department.department_id and
department.location_id=location.location_id
```

Es igual que el ejemplo anterior, pero como se utilizan datos de otra tabla más, hay una nueva condición de combinación.

SQL | Todas las Filas Recuperadas: 31 en 0,016 segundos

	ID_EMPLEADO	NOMBRE_EMPLEADO	DEPARTAMENTO_EMPLEADO	ID_JEFE	NOMBRE_JEFE	DEPARTAMENTO_JEFE	REGIONAL_GROUP
1	7369	SMITH	20	7902	FORD	20	DALLAS
2	7499	ALLEN	30	7698	BLAKE	30	CHICAGO
3	7505	DOYLE	13	7839	KING	10	NEW YORK
4	7506	DENNIS	23	7839	KING	10	NEW YORK
5	7507	BAKER	14	7839	KING	10	NEW YORK
6	7521	WARD	30	7698	BLAKE	30	CHICAGO
7	7555	PETERS	13	7505	DOYLE	13	NEW YORK
8	7557	SHAW	13	7505	DOYLE	13	NEW YORK
9	7560	DUNCAN	23	7506	DENNIS	23	DALLAS
10	7564	LANGE	23	7506	DENNIS	23	DALLAS
11	7566	JONES	20	7839	KING	10	NEW YORK
12	7569	ALBERTS	12	7839	KING	10	NEW YORK
13	7600	PORTER	13	7505	DOYLE	13	NEW YORK
14	7609	LEWIS	24	7507	BAKER	14	NEW YORK
15	7654	MARTIN	30	7698	BLAKE	30	CHICAGO
16	7676	SOMMERS	34	7507	BAKER	14	NEW YORK
17	7698	BLAKE	30	7839	KING	10	NEW YORK
18	7782	CLARK	10	7839	KING	10	NEW YORK
19	7788	SCOTT	20	7566	JONES	20	DALLAS
20	7789	WEST	23	7506	DENNIS	23	DALLAS
21	7799	FISHER	12	7569	ALBERTS	12	NEW YORK
22	7820	ROSS	43	7505	DOYLE	13	NEW YORK
23	7844	TURNER	30	7698	BLAKE	30	CHICAGO
24	7876	ADAMS	20	7788	SCOTT	20	DALLAS
25	7900	JAMES	30	7698	BLAKE	30	CHICAGO
26	7902	FORD	20	7566	JONES	20	DALLAS
27	7916	ROBERTS	12	7569	ALBERTS	12	NEW YORK

## 6. COMBINACIÓN EXTERIOR

Si en el resultado de la combinación hay alguna fila que no tiene correspondencia con ninguna de la otra y queremos seleccionarla utilizaremos combinaciones exteriores. Para las columnas de la tabla que no tienen correspondencia toman valor nulo.

Para realizar una combinación exterior se debe poner "(+)" en la condición de combinación. Este signo se pone en la columna de la tabla que quedará vacía en la combinación.

```
SELECT tabla1.col1, tabla2.col1, .....
FROM tabla1, tabla2
WHERE tabla1.col1(+) = tabla2.col1.
```

Seleccionar el nombre del empleado, la función, el departamento, el código de la localidad y el salario de cada empleado de la tabla empleados. También se deberá seleccionar el departamento que no contenga ningún empleado.

```
SELECT last_name, job_id, department.department_id, location_id, salary
FROM employee, department
WHERE department.department_id=employee.department_id(+);
```

	LAST_NAME	JOB_ID	DEPARTMENT_ID	LOCATION_ID	SALARY
1	CLARK	671	10	122	2450
2	KING	672	10	122	5000
3	MILLER	667	10	122	1300
4	DOUGLAS	667	12	122	800
5	ROBERTS	669	12	122	2875
6	FISHER	669	12	122	3000
7	ALBERTS	671	12	122	3000
8	JENSEN	667	13	122	750
9	DOYLE	671	13	122	2850
10	SHAW	670	13	122	1250
11	PETERS	670	13	122	1250
12	PORTER	670	13	122	1250
13	BAKER	671	14	122	2200
14	FORD	669	20	124	3000
15	JONES	671	20	124	2975
16	SMITH	667	20	124	800
17	ADAMS	667	20	124	1100
18	SCOTT	669	20	124	3000
19	DUNCAN	670	23	124	1250
20	WEST	670	23	124	1500
21	DENNIS	671	23	124	2750
22	LANGE	670	23	124	1250
23	MURRAY	667	23	124	750
24	LEWIS	668	24	124	1800
25	JAMES	667	30	123	950
26	TURNER	670	30	123	1500
27	BLAKE	671	30	123	2850
28	MARTIN	670	30	123	1250

## 7. OPERADORES DE CONJUNTO

Son los mismos de la Teoría de Conjuntos, si consideramos a las filas recuperadas de una consulta como elementos que pertenecen a un conjunto. Combinan dos o más consultas en un mismo resultado. Los comandos **SELECT** deben tener el mismo número de columnas y éstas deben corresponderse en tipo, aunque sus longitudes pueden variar.

Para estos operadores está implícita la cláusula **DISTINCT** y son incompatibles con la cláusula **ORDER BY**.

Estos operadores de conjunto son:

- UNION.
- INTERSECT.
- MINUS.

Operador UNION.

El operador UNION devuelve todas las filas que han sido seleccionadas por los comandos SELECT. Si hay dos iguales sólo devuelve una de ellas.





La sintaxis es:

```
SELECT co1, col2, ... , coln
FROM tabla
WHERE condiciones
UNION
SELECT co1, col2, ... , coln
FROM tabla
WHERE condiciones.
```

Si los comandos SELECT se ejecutan sobre la misma tabla este resultado es equivalente a realizar una sola consulta que contenga en la condición un operador lógico OR.

Seleccionar el nombre, salario y trabajo de todos los empleados que trabajen en el departamento 10 ó 20.

```
SELECT last_name, salary, job_id, department_id
FROM employee
WHERE department_id =10
UNION
SELECT last_name, salary, job_id, department_id
FROM employee
WHERE department_id =20;
```

	 LAST_NAME	 SALARY	 JOB_ID	 DEPARTMENT_ID
1	ADAMS	1100	667	20
2	CLARK	2450	671	10
3	FORD	3000	669	20
4	JONES	2975	671	20
5	KING	5000	672	10
6	MILLER	1300	667	10
7	SCOTT	3000	669	20
8	SMITH	800	667	20

### Operador INTERSECT.

El operador INTERSECT devuelve las filas comunes que han sido seleccionadas por los comandos SELECT. Sólo devuelve una de ellas.

La sintaxis es:

```
SELECT co1, col2, ... , coln
FROM tabla
WHERE condiciones
INTERSECT
SELECT co1, col2, ... , coln
FROM tabla
WHERE condiciones.
```

Si los comandos SELECT se ejecutan sobre la misma tabla el resultado es equivalente a realizar una sola consulta que contenga en la condición un operador lógico AND.

Seleccionar el nombre, salario y trabajo de todos los empleados cuya funcion sea la 670 y su departamento sea el 30.

```
SELECT last_name, salary, job_id, department_id
FROM employee
WHERE job_id = 670
INTERSECT
SELECT last_name, salary, job_id, department_id
FROM employee
WHERE department_id = 30;
```

	LAST_NAME	SALARY	JOB_ID	DEPARTMENT_ID
1	ALLEN	1600	670	30
2	MARTIN	1250	670	30
3	TURNER	1500	670	30
4	WARD	1250	670	30

### Operador MINUS.




El operador MINUS devuelve las filas que han sido seleccionadas por el primer comando SELECT y no lo han sido por el segundo.

La sintaxis es:

```
SELECT co1, col2, ... , coln
FROM tabla
WHERE condiciones
MINUS
SELECT co1, col2, ... , coln
FROM tabla
WHERE condiciones.
```

Seleccionar el nombre, salario y trabajo de todos los empleados cuyo empleo esté en el departamento 10 y no lo esté en el departamento 20.

```
SELECT last_name, salary, job_id
FROM employee
WHERE job_id IN ( SELECT job_id
                  FROM employee
                  WHERE department_id =10
                  MINUS
                  SELECT job_id
                  FROM employee
                  WHERE department_id=20);
```

	 LAST_NAME	 SALARY	 JOB_ID
1	KING	5000	672



## 8. EJERCICIOS PROPUESTOS

- 1.- Muestra de cada empleado, su nombre y nombre de función que realiza.
- 2.- Muestra el código y nombre de los departamentos donde trabaja algún empleado, puede que haya algún departamento que no tiene empleados.
- 3.- Visualiza el nombre de los clientes junto al nombre del empleado que es su representante de ventas.
- 4.- Mostrar de cada empleado su nombre y el grado salarial que le corresponde.
- 5.- Nombre de cada empleado y al lado el nombre de la persona que es su jefe.
- 6.- Nombre de cada empleado y al lado el nombre de la persona que es su jefe, y si no tiene que aparezca "No tiene jefe".
- 7.- Nombre de empleados y nombre del depto donde trabajan de aquellos que cobran comisión.
- 8.- Cuantos empleados hay en cada depto de ventas.
- 9.- Muestra el nombre de los clientes cuyo representante de ventas es WARD.
- 10.- Muestra el nombre de los empleados que son vendedores (SALESPERSON)
- 11.- ¿Cuántos empleados son vendedores?
- 12.- ¿Cuál es el valor del pedido de mayor valor que ha realizado el cliente JOCKSPORTS?
- 13.- Muestra de cada producto, su nombre y precio (LIST\_PRICE) que se le aplica actualmente.

- 14.- ¿Cuánto dinero se ha recaudado con la venta de 'ACE TENNIS RACKET I'?
- 15.- Muestra de cada departamento, su nombre y número de empleados que trabajan en él; considera los departamentos con el mismo nombre y en distintas localidades, como un único departamento.
- 16.- Visualiza para cada producto, su nombre y total de unidades vendidas de dicho producto.
- 17.- Visualiza de cada cliente, su nombre y el valor del pedido de mayor valor que ha realizado hasta el momento.
- 18.- Visualiza de cada cliente, su nombre y la suma de los totales de todos los pedidos que ha realizado.
- 19.- Mostrar de cada depto su código, su nombre y el nº de empleados que tiene, pero si no tiene empleados que salga un 0.
- 20.- ¿Cuántos clientes hay que sólo han realizado un pedido?
- 21.- Muestra el nombre del cliente que realizo el pedido de mayor valor.
- 22.- Muestra el nombre del producto más caro.
- 23.- Muestra el nombre del vendedor con más clientes.
- 24.- Muestra el nombre del departamento cuya suma de salarios de empleados es la mayor de todas las sumas.
- 25.- Muestra el nombre del cliente que ha realizado el mayor número de pedidos.
- 26.- Muestra el nombre del producto más vendido.
- 27.- ¿Qué cliente o clientes realizaron el primer pedido a la empresa?

- 28.- Muestra el nombre de cada cliente y el código del pedido (ORDER\_ID) que, entre todos los pedidos realizados por el cliente, es el más caro (el de mayor valor en el campo TOTAL).
- 29.- Muestra de cada cliente y sin repetidos, los códigos de los distintos productos que ha solicitado en sus pedidos.
- 30.- Mostrar el nombre de cada empleado y nombre de la localidad donde trabaja.
- 31.- Seleccionar para cada empleado el nombre, número, número de jefe, departamento y nombre del jefe.
- 32.- Calcula cuántos empleados trabajan en el departamento de ventas (SALES) de CHICAGO.
- 33.- Muestra de cada departamento y de cada localidad, cuántos trabajadores tienen; muestra nombre de localidad, nombre de departamento y número de empleados.
- 34.- Muestra un listado con el nombre de los vendedores y la comisión que obtendrían si se les pagara un 25% del total facturado a sus clientes.
- 35.- Muestra un listado con el nombre de los vendedores y la comisión que obtendrían si se les pagara un 25% del total facturado a sus clientes. Para aquellos vendedores que no han vendido, se debe visualizar cero como comisión.
- 36.- Muestra de cada cliente y sin repetidos, los nombres de los productos que ha solicitado en sus pedidos.
- 37.- Muestra el nombre del cliente que ha comprado más de 100 unidades del producto 'ACE TENNIS RACKET II'.
- 38.- Muestra de cada departamento, su código, nombre, localidad y nombre del empleado más antiguo en ese departamento.