

Índice

0. Introducción	2
1. Comando SELECT	2
2. Cláusula DISTINCT	2
3. Cláusula WHERE	5
AND / OR	9
IN	9
BETWEEN	12
IS NULL	14
LIKE	16
4. Cláusula ORDER BY	17
5. Ejercicios propuestos	20

0. INTRODUCCIÓN

El objetivo de este tema es el de conocer cómo se puede efectuar una de las operaciones más comunes que se realizan sobre una base de datos relacional: la de extraer información de las tablas que la componen. Siempre que se desee información de las tablas que componen una base de datos relacional manipulada con el lenguaje SQL, se utilizará el comando SELECT con una serie de cláusulas, las cuales serán estudiadas en este tema. Un primer formato para el comando SELECT es:

```
SELECT col1, col2, ... , coln
FROM nom_tabla;
```

Con esta sentencia se seleccionan los datos contenidos en las columnas especificadas de la tabla. Solo se seleccionarán las columnas especificadas después de la palabra clave SELECT y en ese mismo orden. Serán devueltas todas las filas de la tabla y en el orden en el que fueron insertadas en la tabla.

Si se desea obtener los datos de todas las columnas de la tabla y en el orden en el que fueron insertadas, se tendrá que sustituir la especificación de las columnas por un asterisco, quedando la sentencia de la siguiente manera:

```
SELECT *
FROM nom_tabla;
```

1. COMANDO SELECT

1.1 Seleccionar las columnas código de departamento, nombre, salario, fecha de entrada y trabajo de la tabla de empleados:

```
SELECT department_id, last_name, job_id, hire_date, salary
FROM employee;
```

	DEPARTMENT_ID	LAST_NAME	JOB_ID	HIRE_DATE	SALARY
1	20	SMITH	667	17/12/84	800
2	30	ALLEN	670	20/02/85	1600
3	13	DOYLE	671	04/04/85	2850
4	23	DENNIS	671	15/05/85	2750
5	14	BAKER	671	10/06/85	2200
6	30	WARD	670	22/02/85	1250
7	13	PETERS	670	31/03/85	1250
8	13	SHAW	670	02/04/85	1250
9	23	DUNCAN	670	31/05/85	1250
10	23	LANGE	670	01/06/85	1250
11	20	JONES	671	02/04/85	2975
12	12	ALBERTS	671	06/04/85	3000
13	13	PORTER	670	15/04/85	1250
14	24	LEWIS	668	16/04/85	1800
15	30	MARTIN	670	28/09/85	1250
16	34	SOMMERS	668	19/04/85	1850
17	30	BLAKE	671	01/05/85	2850
18	10	CLARK	671	09/06/85	2450
19	20	SCOTT	669	09/12/86	3000
20	23	WEST	670	04/04/85	1500
21	12	FISHER	669	12/12/86	3000
22	43	ROSS	670	01/06/85	1300
23	10	KING	672	17/11/85	5000
24	30	TURNER	670	08/09/85	1500
25	20	ADAMS	667	12/01/87	1100
26	30	JAMES	667	03/12/85	950

1.2 . Seleccionar todas las columnas de la tabla de departamentos:

```
SELECT *
FROM department;
```

	DEPARTMENT_ID	NAME	LOCATION_ID
1	10	ACCOUNTING	122
2	20	RESEARCH	124
3	30	SALES	123
4	40	OPERATIONS	167
5	12	RESEARCH	122
6	13	SALES	122
7	14	OPERATIONS	122
8	23	SALES	124
9	24	OPERATIONS	124
10	34	OPERATIONS	123
11	43	SALES	167

1.3 Seleccionar todos los grados (GRADE_ID), salarios mínimos (LOWER_BOUND) y salarios máximos (UPPER_BOUND) correspondientes, de la tabla de grados de salarios:

```
SELECT grade_id, lower_bound, upper_bound
FROM salary_grade;
```

	GRADE_ID	LOWER_BOUND	UPPER_BOUND
1	1	700	1200
2	2	1201	1400
3	3	1401	2000
4	4	2001	3000
5	5	3001	9999

2. CLAUSULA DISTINCT

Si seleccionamos ciertas columnas de la base de datos que contienen datos iguales, las filas devueltas se visualizarán tantas veces como estén en la tabla. Si sólo se desea seleccionar los valores de las filas una sola vez no importando el número de veces que estos se repitan deberemos utilizar la sentencia:

```
SELECT DISTINCT col1, col2, ... , coln
FROM nom_tabla;
```

Seleccionará todas las filas que en las columnas col1, col2, coln, tengan valores distintos. Si existen filas con valores iguales en todas las columnas seleccionadas, solamente aparecerán una vez.

2.1. Seleccionar los distintos departamentos que existen en la tabla de empleados.

```
SELECT DISTINCT department_id
FROM employee;
```

	DEPARTMENT_ID
1	30
2	34
3	43
4	13
5	20
6	14
7	24
8	23
9	12
10	10

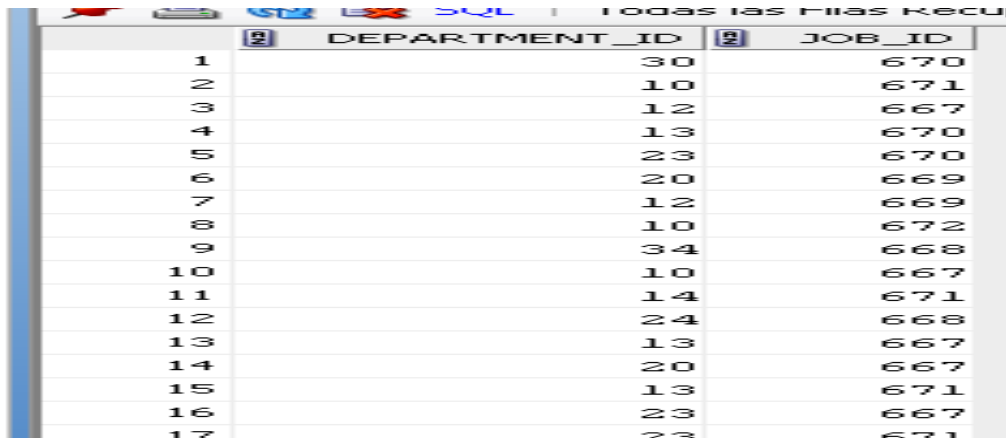
La siguiente sentencia no tendrá ningún sentido ya que si se seleccionan todas las columnas de la tabla, no habrá filas que tengan los mismos valores en todas sus columnas, siempre que se haya diseñado correctamente la base de datos relacional.

```
SELECT DISTINCT *
FROM employee;
```

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	MIDDLE_INITIAL	JOB_ID	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION	DEPARTMENT_ID
1	SMITH	JOHN	Q	667	7902	17/12/84	800	(null)	20
2	ALLEN	KEVIN	J	670	7698	20/02/85	1600	300	30
3	DOYLE	JEAN	K	671	7839	04/04/85	2850	(null)	13
4	DENNIS	LYNN	S	671	7839	15/05/85	2750	(null)	23
5	BAKER	LESLIE	D	671	7839	10/06/85	2200	(null)	14
6	WARD	CYNTHIA	D	670	7698	22/02/85	1250	500	30
7	PETERS	DANIEL	T	670	7505	31/03/85	1250	300	13
8	SHAW	KAREN	P	670	7505	02/04/85	1250	1200	13
9	DUNCAN	SARAH	S	670	7506	31/05/85	1250	(null)	23
10	LANGE	GREGORY	J	670	7506	01/06/85	1250	300	23
11	JONES	TERRY	M	671	7839	02/04/85	2975	(null)	20
12	ALBERTS	CHRIS	L	671	7839	06/04/85	3000	(null)	12
13	PORTER	RAYMOND	Y	670	7505	15/04/85	1250	900	13
14	LEWIS	RICHARD	M	668	7507	16/04/85	1800	(null)	24
15	MARTIN	KENNETH	J	670	7698	28/09/85	1250	1400	30
16	SOMMERS	DENISE	D	668	7507	19/04/85	1850	(null)	34
17	BLAKE	MARTIN	S	671	7839	01/05/85	2850	(null)	30

2.2 Seleccionar los distintos empleos que hay en cada departamento de la tabla de empleados:

```
SELECT DISTINCT department_id, job_id
FROM employee;
```



	DEPARTMENT_ID	JOB_ID
1	30	670
2	10	671
3	12	667
4	13	670
5	23	670
6	20	669
7	12	669
8	10	672
9	34	668
10	10	667
11	14	671
12	24	668
13	13	667
14	20	667
15	13	671
16	23	667
17	23	671

3. CLAUSULA WHERE

Hasta el momento hemos extraído información de todas las filas que componían la tabla, pero si se desea restringir las filas devueltas por una sentencia SELECT especificaremos la cláusula WHERE y a continuación las condiciones que deben cumplir las filas para que sean visualizadas. El comando SELECT con la cláusula WHERE queda de la siguiente manera:

```
SELECT col1, col2, ... , coln
FROM nom_tabla
WHERE condición;
```

Una condición está compuesta por un operador relacional (<, >, =, etc...) uniendo:

- Dos columnas de la tabla.
- Una columna y una constante.

La constante puede ser:

- . Numérica.
- . Carácter, (irá encerrada entre comillas simples).
- . Fecha, (también irá encerrada entre comillas simples).

Los operadores relacionales son:

- Igual, =.
- Distinto, !=.
- Mayor, >.
- Mayor o igual, >=.
- Menor, <.
- Menor o igual, <=.

Las filas que formen parte de las condiciones no tienen porque ser seleccionadas. Pueden seleccionarse unas columnas y restringir las filas extraídas por otras.

3.1 Seleccionar aquellas personas cuyo código de función sea el 670.

```
SELECT last_name, job_id, salary, hire_date
FROM employee
WHERE job_id= 670;
```

	AZ	LAST_NAME	AZ	JOB_ID	AZ	SALARY	AZ	HIRE_DATE
1		ALLEN		670		1600		20/02/85
2		WARD		670		1250		22/02/85
3		PETERS		670		1250		31/03/85
4		SHAW		670		1250		02/04/85
5		DUNCAN		670		1250		31/05/85
6		LANGE		670		1250		01/06/85
7		PORTER		670		1250		15/04/85
8		MARTIN		670		1250		28/09/85
9		WEST		670		1500		04/04/85
10		ROSS		670		1300		01/06/85
11		TURNER		670		1500		08/09/85

3.2 Seleccionar aquellas personas que no trabajen en el departamento 30.

```
SELECT last_name, job_id, salary, hire_date, department_id
FROM employee
WHERE department_id != 30;
```

	AZ	LAST_NAME	AZ	JOB_ID	AZ	SALARY	AZ	HIRE_DATE	AZ	DEPARTMENT_ID
1		SMITH		667		800		17/12/84		20
2		DOYLE		671		2850		04/04/85		13
3		DENNIS		671		2750		15/05/85		23
4		BAKER		671		2200		10/06/85		14
5		PETERS		670		1250		31/03/85		13
6		SHAW		670		1250		02/04/85		13
7		DUNCAN		670		1250		31/05/85		23
8		LANGE		670		1250		01/06/85		23
9		JONES		671		2975		02/04/85		20
10		ALBERTS		671		3000		06/04/85		12
11		PORTER		670		1250		15/04/85		13
12		LEWIS		668		1800		16/04/85		24
13		SOMMERS		668		1850		19/04/85		34
14		CLARK		671		2450		09/06/85		10
15		SCOTT		669		3000		09/12/86		20
16		WEST		670		1500		04/04/85		23
17		FISHER		669		3000		12/12/86		12
18		ROSS		670		1300		01/06/85		43
19		KING		672		5000		17/11/85		10
20		ADAMS		667		1100		12/01/87		20
21		FORD		669		3000		03/12/85		20
22		ROBERTS		669		2875		04/01/87		12
23		DOUGLAS		667		800		04/01/87		12
24		MILLER		667		1300		23/01/86		10
25		JENSEN		667		750		15/01/87		13
26		MURRAY		667		750		16/01/87		23

3.3 Seleccionar aquellas personas que ganen más de 2000 dólares.

```
SELECT last_name, job_id, salary, hire_date
FROM employee
WHERE salary > 2000;
```

	LAST_NAME	JOB_ID	SALARY	HIRE_DATE
1	DOYLE	671	2850	04/04/85
2	DENNIS	671	2750	15/05/85
3	BAKER	671	2200	10/06/85
4	JONES	671	2975	02/04/85
5	ALBERTS	671	3000	06/04/85
6	BLAKE	671	2850	01/05/85
7	CLARK	671	2450	09/06/85
8	SCOTT	669	3000	09/12/86
9	FISHER	669	3000	12/12/86
10	KING	672	5000	17/11/85
11	FORD	669	3000	03/12/85
12	ROBERTS	669	2875	04/01/87

3.4 Seleccionar aquellas personas que hayan entrado en la empresa antes del 1 de Enero de 1985.

```
SELECT last_name, job_id, salary, hire_date
FROM employee
WHERE hire_date < '1-1-85';
```

	LAST_NAME	JOB_ID	SALARY	HIRE_DATE
1	SMITH	667	800	17/12/84

3. 5 CONDICIONES MULTIPLES EN EL WHERE

Dentro de la cláusula WHERE se pueden introducir condiciones múltiples de consulta, las cuales se evalúan antes de devolver las filas correspondientes. Los operadores que se utilizan para unir varias condiciones de consulta son:

AND. Las filas seleccionadas tienen que cumplir todas las condiciones unidas con este operador. Puede haber más de un operador AND.

OR. Las filas seleccionadas tienen que cumplir alguna de las condiciones unidas con este operador. Puede haber más de un operador OR.

IN. El valor de la columna se debe corresponder con alguno de los valores especificados en la lista que sigue al operador IN.

NOT IN. Es la complementaria de la anterior.

El operador AND es más prioritario que el operador OR, por lo que se evalúa antes en las condiciones de consulta.

3.5 Seleccionar aquellos empleados cuya función es 671 y ganan más de 1500 dólares.

```
SELECT last_name, job_id, salary, hire_date
FROM employee
WHERE job_id = 671 AND salary > 1500;
```

	LAST_NAME	JOB_ID	SALARY	HIRE_DATE
1	DOYLE	671	2850	04/04/85
2	DENNIS	671	2750	15/05/85
3	BAKER	671	2200	10/06/85
4	JONES	671	2975	02/04/85
5	ALBERTS	671	3000	06/04/85
6	BLAKE	671	2850	01/05/85
7	CLARK	671	2450	09/06/85

3.6 Seleccionar aquellos empleados cuya función es 671 o que trabajen en el departamento 30.

```
SELECT last_name, job_id, salary, hire_date
FROM employee
WHERE job_id= 671 OR department_id= 30;
```

Todas las Filas Recuperadas: 12 en 0 segundos

	LAST_NAME	JOB_ID	SALARY	HIRE_DATE
1	ALLEN	670	1600	20/02/85
2	DOYLE	671	2850	04/04/85
3	DENNIS	671	2750	15/05/85
4	BAKER	671	2200	10/06/85
5	WARD	670	1250	22/02/85
6	JONES	671	2975	02/04/85
7	ALBERTS	671	3000	06/04/85
8	MARTIN	670	1250	28/09/85
9	BLAKE	671	2850	01/05/85
10	CLARK	671	2450	09/06/85
11	TURNER	670	1500	08/09/85
12	JAMES	667	950	03/12/85

3.7 Seleccionar aquellos empleados que se llamen SMITH, ALLEN, SCOTT o MILLER.

```
SELECT last_name, job_id, salary, hire_date
FROM employee
WHERE last_name IN('SMITH','ALLEN','SCOTT','MILLER');
```

	LAST_NAME	JOB_ID	SALARY	HIRE_DATE
1	SMITH	667	800	17/12/84
2	ALLEN	670	1600	20/02/85
3	SCOTT	669	3000	09/12/86
4	MILLER	667	1300	23/01/86

3.8 Seleccionar aquellos empleados que no se llamen SMITH, ALLEN, SCOTT o MILLER.

```
SELECT last_name, job_id, salary, hire_date
FROM employee
WHERE last_name NOT IN('SMITH','ALLEN','SCOTT','MILLER');
```

	LAST_NAME	JOB_ID	SALARY	HIRE_DATE
1	DOYLE	671	2850	04/04/85
2	DENNIS	671	2750	15/05/85
3	BAKER	671	2200	10/06/85
4	WARD	670	1250	22/02/85
5	PETERS	670	1250	31/03/85
6	SHAW	670	1250	02/04/85
7	DUNCAN	670	1250	31/05/85
8	LANGE	670	1250	01/06/85
9	JONES	671	2975	02/04/85
10	ALBERTS	671	3000	06/04/85
11	PORTER	670	1250	15/04/85
12	LEWIS	668	1800	16/04/85
13	MARTIN	670	1250	28/09/85
14	SOMMERS	668	1850	19/04/85
15	BLAKE	671	2850	01/05/85
16	CLARK	671	2450	09/06/85
17	WEST	670	1500	04/04/85
18	FISHER	669	3000	12/12/86
19	ROSS	670	1300	01/06/85
20	KING	672	5000	17/11/85
21	TURNER	670	1500	08/09/85
22	ADAMS	667	1100	12/01/87
23	JAMES	667	950	03/12/85
24	FORD	669	3000	03/12/85
25	ROBERTS	669	2875	04/01/87
26	DOUGLAS	667	800	04/01/87
27	JENSEN	667	750	15/01/87

Si queremos cambiar la prioridad de evaluación de las condiciones utilizaremos paréntesis.

3.9 Seleccionar aquellos empleados que trabajen en el departamento 30 y su función sea 670 o 671

```
SELECT last_name, job_id, salary, hire_date
FROM employee
WHERE department_id = 30 AND (job_id= 670 OR job_id=671);
```

	LAST_NAME	JOB_ID	SALARY	HIRE_DATE
1	ALLEN	670	1600	20/02/85
2	WARD	670	1250	22/02/85
3	MARTIN	670	1250	28/09/85
4	BLAKE	671	2850	01/05/85
5	TURNER	670	1500	08/09/85

3.10 Seleccionar aquellos empleados cuya función sea 670, o que trabajen en el departamento 20 y su función sea 671.

```
SELECT last_name, job_id, department_id, salary, hire_date
FROM employee
WHERE job_id= 670 OR (department_id=20 AND job_id=671);
```

	LAST_NAME	JOB_ID	DEPARTMENT_ID	SALARY	HIRE_DATE
1	ALLEN	670	30	1600	20/02/85
2	WARD	670	30	1250	22/02/85
3	PETERS	670	13	1250	31/03/85
4	SHAW	670	13	1250	02/04/85
5	DUNCAN	670	23	1250	31/05/85
6	LANGE	670	23	1250	01/06/85
7	JONES	671	20	2975	02/04/85
8	PORTER	670	13	1250	15/04/85
9	MARTIN	670	30	1250	28/09/85
10	WEST	670	23	1500	04/04/85
11	ROSS	670	43	1300	01/06/85
12	TURNER	670	30	1500	08/09/85

Con estos ejemplos se comprueba que el cambio de prioridad de evaluación de las condiciones hace variar la consulta y por lo tanto el resultado.

3.6 . OPERADOR BETWEEN ... AND

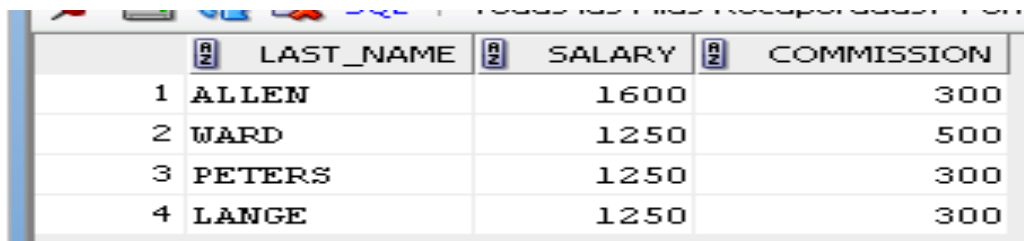
Cuando se desea seleccionar datos de una columna que estén comprendidos entre dos valores se utiliza la cláusula BETWEEN ... AND, en la que se expresan los límites, min_val y max_val. El formato es el siguiente:

WHERE columna **BETWEEN** min_val **AND** max_val.

Matemáticamente es un intervalo cerrado, luego se incluyen los extremos. Por lo tanto su negación será un intervalo abierto, que no comprenderá los extremos.

3.11 Seleccionar aquellos empleados que tengan una comisión mayor de 200 y menor de 600.

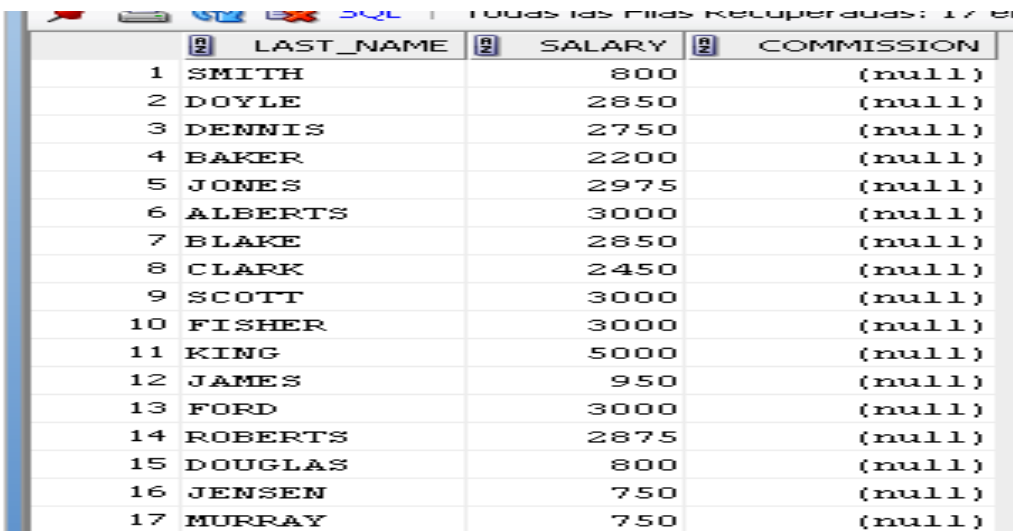
```
SELECT last_name, salary, commission
FROM employee
WHERE commission BETWEEN 200 AND 600;
```



	LAST_NAME	SALARY	COMMISSION
1	ALLEN	1600	300
2	WARD	1250	500
3	PETERS	1250	300
4	LANGE	1250	300

3.12 Seleccionar aquellos empleados que cuyo salario no está comprendido entre 1000 y 2000.

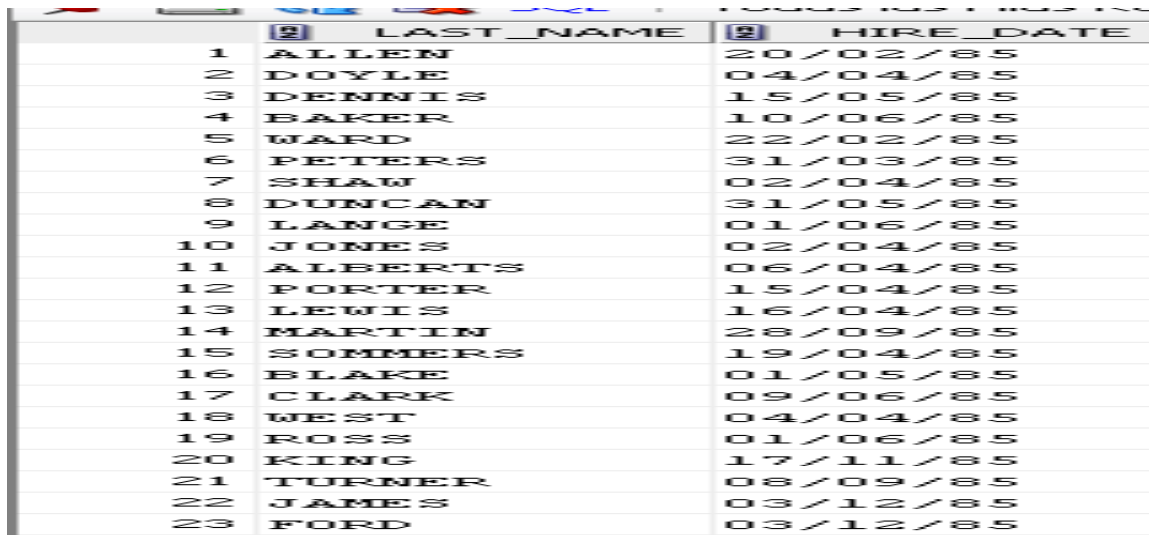
```
SELECT last_name, salary, commission
FROM employee
WHERE salary NOT BETWEEN 1000 AND 2000;
```



	LAST_NAME	SALARY	COMMISSION
1	SMITH	800	(null)
2	DOYLE	2850	(null)
3	DENNIS	2750	(null)
4	BAKER	2200	(null)
5	JONES	2975	(null)
6	ALBERTS	3000	(null)
7	BLAKE	2850	(null)
8	CLARK	2450	(null)
9	SCOTT	3000	(null)
10	FISHER	3000	(null)
11	KING	5000	(null)
12	JAMES	950	(null)
13	FORD	3000	(null)
14	ROBERTS	2875	(null)
15	DOUGLAS	800	(null)
16	JENSEN	750	(null)
17	MURRAY	750	(null)

3.13 Seleccionar aquellos empleados que entraron en la empresa durante el año 1985.

```
SELECT last_name, hire_date
FROM employee
WHERE hire_date BETWEEN '1-1-85' AND '31-12-85';
```



	LAST_NAME	HIRE_DATE
1	ALLEN	20/02/85
2	DOYLE	04/04/85
3	DENNIS	15/05/85
4	BAKER	10/06/85
5	WARD	22/02/85
6	PETERS	31/03/85
7	SHAW	02/04/85
8	DUNCAN	31/05/85
9	LANGE	01/06/85
10	JONES	02/04/85
11	ALBERTS	06/04/85
12	PORTER	15/04/85
13	LEWIS	16/04/85
14	MARTIN	28/09/85
15	SOMMERS	19/04/85
16	BLAKE	01/05/85
17	CLARK	09/06/85
18	WEST	04/04/85
19	ROSS	01/06/85
20	KING	17/11/85
21	TURNER	08/09/85
22	JAMES	03/12/85
23	FORD	03/12/85

3.7 . VALORES NULOS

Hay veces que el valor de una columna es desconocido o inaplicable para una cierta fila. En este caso se le asigna un valor nulo (NULL). Nulo es distinto de cero o blanco. Por ejemplo, la comisión nula significa que ese empleado no tiene comisión - puede que no sea vendedor -, mientras que si un empleado no vende nada su comisión será cero.

Para seleccionar las filas de una tabla que tiene valor nulo en una determinada columna utilizamos la siguiente condición:

WHERE columna **IS NULL**.

Si por el contrario se desea seleccionar aquellas filas de la tabla en las que una columna tiene un valor distinto de nulo usaremos:

WHERE columna **IS NOT NULL**.

3.14 Seleccionar aquellos empleados que tienen comisión:

```
SELECT last_name, commission, department_id, job_id
FROM employee
WHERE commission IS NOT NULL;
```

Todas las Filas Recuperadas: 10 en 0,015 segundos

	LAST_NAME	COMMISSION	DEPARTMENT_ID	JOB_ID
1	ALLEN	300	30	670
2	WARD	500	30	670
3	PETERS	300	13	670
4	SHAW	1200	13	670
5	LANGE	300	23	670
6	PORTER	900	13	670
7	MARTIN	1400	30	670
8	WEST	1000	23	670
9	ROSS	800	43	670
10	TURNER	0	30	670

3.15 Seleccionar aquellos empleados que no tienen comisión:

```
SELECT last_name, commission, department_id, job_id
FROM employee
WHERE commission IS NULL;
```

Todas las Filas Recuperadas: 22 en 0,016 segundos

	LAST_NAME	COMMISSION	DEPARTMENT_ID	JOB_ID
1	SMITH	(null)	20	667
2	DOYLE	(null)	13	671
3	DENNIS	(null)	23	671
4	BAKER	(null)	14	671
5	DUNCAN	(null)	23	670
6	JONES	(null)	20	671
7	ALBERTS	(null)	12	671
8	LEWIS	(null)	24	668
9	SOMMERS	(null)	34	668
10	BLAKE	(null)	30	671
11	CLARK	(null)	10	671
12	SCOTT	(null)	20	669
13	FISHER	(null)	12	669
14	KING	(null)	10	672
15	ADAMS	(null)	20	667
16	JAMES	(null)	30	667
17	FORD	(null)	20	669
18	ROBERTS	(null)	12	669
19	DOUGLAS	(null)	12	667
20	MILLER	(null)	10	667
21	JENSEN	(null)	13	667
22	MURRAY	(null)	23	667

3.8 . CARACTERES DE SUSTITUCIÓN

Para seleccionar una fila que cumpla una cierta condición no es necesario conocer el conjunto exacto de caracteres que forman la restricción. Utilizaremos el operador **LIKE** combinado con unos caracteres especiales que sustituyen cadenas de caracteres. Estos caracteres especiales son:

- Este carácter sustituye a cualquier otro.
- % Este carácter sustituye a cualquier cadena de caracteres.

3.16 Obtener aquellos empleados cuyo nombre comience por A.

```
SELECT last_name  
FROM employee  
WHERE last_name LIKE 'A%';
```

	A Z	LAST_NAME
1		ALLEN
2		ALBERTS
3		ADAMS

3.17 Seleccionar aquellos empleados cuyo nombre contenga como segunda letra una D.

```
SELECT last_name  
FROM employee  
WHERE last_name LIKE '_D%';
```

	A Z	LAST_NAME
1		ADAMS

4. CLAUSULA ORDER BY

Las filas de la tabla se almacenan en el mismo orden en el que han sido insertadas, por lo que cuando las seleccionamos siempre aparecerán en ese orden. Para variar este orden se utiliza la cláusula ORDER BY. Con ella se especifican las columnas por las cuales queremos que se ordenen las filas seleccionadas por una consulta.

Consideraciones sobre la cláusula **ORDER BY**:

- Pueden formar parte de ella cualquier número de columnas hasta un máximo de 16 no importando el tipo de datos de éstas.
- El orden puede ser ascendente (ASC) o descendente (DESC). Por omisión será ascendente.
- Pueden mezclarse ordenaciones ascendentes y descendentes.
- Se puede clasificar por columnas que no han sido seleccionadas.
- Siempre es la última cláusula de la sentencia SELECT.

El formato es:

ORDER BY columna ASC o DESC, columna ASC o DESC

4.1 Seleccionar aquellos empleados que ganan más de 1500 dólares ordenados por id de funcion.

```
SELECT last_name, salary, job_id  
FROM employee  
WHERE salary > 1500  
ORDER BY job_id;
```

	LAST_NAME	SALARY	JOB_ID
1	SOMMERS	1850	668
2	LEWIS	1800	668
3	FORD	3000	669
4	ROBERTS	2875	669
5	FISHER	3000	669
6	SCOTT	3000	669
7	ALLEN	1600	670
8	DOYLE	2850	671
9	BLAKE	2850	671
10	CLARK	2450	671
11	DENNIS	2750	671
12	ALBERTS	3000	671
13	JONES	2975	671
14	BAKER	2200	671
15	KING	5000	672

4.2 Seleccionar los empleados del departamento 20 ordenados descendientemente por el salario.

```
SELECT *  
FROM employee  
WHERE department_id = 20  
ORDER BY salary DESC;
```

	EMPLOYEE_ID	LAST_NAME	FIRST_NAME	MIDDLE_INITIAL	JOB_ID	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION	DEPARTMENT_ID
1	7788	SCOTT	DONALD	T	669	7566	09/12/86	3000	(null)	20
2	7902	FORD	JENNIFER	D	669	7566	03/12/85	3000	(null)	20
3	7566	JONES	TERRY	Y	671	7839	02/04/85	2975	(null)	20
4	7876	ADAMS	DIANE	G	667	7788	12/01/87	1100	(null)	20
5	7369	SMITH	JOHN	Q	667	7902	17/12/84	800	(null)	20

4.3 Seleccionar aquellos empleados que no tienen comisión, ordenados por departamentos y dentro de estos, descendientemente por el salario.

```
SELECT *
FROM employee
WHERE commission IS NULL
ORDER BY department_id, salary DESC;
```

	EMPLOYEE_ID	LAST_NAME	FIRST_NAME	MIDDLE_INITIAL	JOB_ID	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION	DEPARTMENT_ID
1	7839	KING	FRANCIS	A	672	(null)	17/11/85	5000	(null)	10
2	7782	CLARK	CAROL	F	671	7839	09/06/85	2450	(null)	10
3	7934	MILLER	BARBARA	M	667	7782	23/01/86	1300	(null)	10
4	7799	FISHER	MATTHEW	G	669	7569	12/12/86	3000	(null)	12
5	7569	ALBERTS	CHRIS	L	671	7839	06/04/85	3000	(null)	12
6	7916	ROBERTS	GRACE	M	669	7569	04/01/87	2875	(null)	12
7	7919	DOUGLAS	MICHAEL	A	667	7799	04/01/87	800	(null)	12
8	7505	DOYLE	JEAN	K	671	7839	04/04/85	2850	(null)	13
9	7950	JENSEN	ALICE	B	667	7505	15/01/87	750	(null)	13
10	7507	BAKER	LESLIE	D	671	7839	10/06/85	2200	(null)	14
11	7902	FORD	JENNIFER	D	669	7566	03/12/85	3000	(null)	20
12	7788	SCOTT	DONALD	T	669	7566	09/12/86	3000	(null)	20
13	7566	JONES	TERRY	M	671	7839	02/04/85	2975	(null)	20
14	7876	ADAMS	DIANE	G	667	7788	12/01/87	1100	(null)	20
15	7369	SMITH	JOHN	Q	667	7902	17/12/84	800	(null)	20
16	7506	DENNIS	LYNN	S	671	7839	15/05/85	2750	(null)	23
17	7560	DUNCAN	SARAH	S	670	7506	31/05/85	1250	(null)	23
18	7954	MURRAY	JAMES	T	667	7506	16/01/87	750	(null)	23
19	7609	LEWIS	RICHARD	M	668	7507	16/04/85	1800	(null)	24
20	7698	BLAKE	MARION	S	671	7839	01/05/85	2850	(null)	30
21	7900	JAMES	FRED	S	667	7698	03/12/85	950	(null)	30
22	7676	SOMMERS	DENISE	D	668	7507	19/04/85	1850	(null)	34

5. EJERCICIOS PROPUESTOS

- 1.- Mostrar todos los empleados cuyo sueldo es mayor que 100.
- 2.- Mostrar nombre y salario de los empleados cuyo salario es 3000.
- 3.- Mostrar los datos de los empleados que se llamen ALICE.
- 4.- Mostrar nombre, apellido, salario y comisión de los empleados cuyo apellido empieza por M.
- 5.- Mostrar todos los campos de los empleados cuyo salario este entre 1000 y 3000
- 6.- Mostrar el nombre y salario de los empleados cuyo salario no este entre 1000 y 3000
- 7.- Mostrar el nombre, departamento y salario de los empleados que trabajando en el 20 cobren más de 1000, más los del 30.
- 8.- Mostrar el Nombre y apellido de los empleados que no cobran comisión, Ordenado alfabéticamente
- 9.- Mostrar el nombre de los empleados cuya comisión sea superior al 10% de su salario.
- 10.- Visualiza toda la información de los empleados cuyo apellido comienza con D
- 11.- Muestra el nombre de los empleados que tienen un salario anual (sin comisión) menor a 12.000
- 12.- Muestra los distintos nombres de los departamentos ordenados alfabéticamente
- 13.- Muestra la información de los empleados de los departamentos 20 y 30 que cobran comisión
- 14.- Muestra la información de todos los empleados del departamento 20, y de los empleados del departamento 30 que cobran comisión.
- 15.- ¿Cuánto cobraría cada empleado si incrementamos su salario (sin comisión) un 20%?
- 16.- De los empleados de los departamentos 30 y 13, muestra su nombre y salario actual, lo que cobrarían si incrementamos éste un 10% y lo que cobrarían si decrementamos el salario un 5%.
- 17.- Muestra el código del departamento donde trabaja el jefe
- 18.- Muestra el nombre de los empleados que tienen como jefe al empleado con código 7788?
- 19.- Muestra de la tabla de precios todos los precios que se aplican actualmente