

EJERCICIOS DE SQL

1°.- Sacar el contenido de la tabla EMPLE

```
SELECT * FROM EMPLE;
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_AL	SALARIO
COMISION	DEPT				
-----	-----	-----	-----	-----	-----
7369	SANCHEZ	EMPLEADO	7902	17/12/80	104000
20					
7499	ARROYO	VENDEDOR	7698	20/02/80	208000
39000	30				
7521	SALA	VENDEDOR	7698	22/02/81	162500
65000	30				
7566	JIMENEZ	DIRECTOR	7839	02/04/81	386750
20					
7654	MARTIN	VENDEDOR	7898	28/09/81	182000
182000	30				
7698	NEGRO	DIRECTOR	7839	01/05/81	370500
30					
7782	CEREZO	DIRECTOR	7839	09/06/81	318500
10					
7788	GIL	ANALISTA	7566	30/03/87	390000
20					
7839	REY	PRESIDENTE		17/11/81	650000
10					
7844	TOVAR	VENDEDOR	7698	08/09/81	195000
0	30				
7876	ALONSO	EMPLEADO	7788	03/05/87	143000
20					
7900	JIMENO	EMPLEADO	7698	03/12/81	123500
30					
7902	FERNANDEZ	ANALISTA	7566	03/12/81	390000
20					
7934	MUÑOZ	EMPLEADO	7782	23/01/82	169000
10					

2°.- Sacar los números de departamento, nombres y oficios de los empleados de la tabla

```
SELECT DEPT_NO, APELLIDO, OFICIO  
FROM EMPLE;
```

DEPT_NO	APELLIDO	OFICIO
-----	-----	-----
20	SANCHEZ	EMPLEADO
30	ARROYO	VENDEDOR
30	SALA	VENDEDOR
20	JIMENEZ	DIRECTOR
30	MARTIN	VENDEDOR
30	NEGRO	DIRECTOR

10	CEREZO	DIRECTOR
20	GIL	ANALISTA
10	REY	PRESIDENTE
30	TOVAR	VENDEDOR
20	ALONSO	EMPLEADO
30	JIMENO	EMPLEADO
20	FERNANDEZ	ANALISTA
10	MUÑOZ	EMPLEADO

3°.- Seleccionar el nombre y salario de todos los empleados de la tabla

```
SELECT APELLIDO,SALARIO
FROM EMPL;
```

APELLIDO	SALARIO
-----	-----
SANCHEZ	104000
ARROYO	208000
SALA	162500
JIMENEZ	386750
MARTIN	182000

NEGRO	370500
CEREZO	318500
GIL	390000
REY	650000
TOVAR	195000
ALONSO	143000
JIMENO	123500
FERNANDEZ	390000
MUÑOZ	169000

4°.-Seleccionar los distintos tipos de empleo de toda la tabla

```
SELECT DISTINCT OFICIO
FROM EMPL;
```

OFICIO

ANALISTA
DIRECTOR
EMPLEADO
PRESIDENTE
VENDEDOR

5°.-Seleccionar los diferentes empleos para cada departamento

```
SELECT DISTINCT DEPT_NO,OFICIO
FROM EMPL;
```

DEPT_NO	OFICIO
10	DIRECTOR
10	EMPLEADO
10	PRESIDENTE
20	ANALISTA
20	DIRECTOR
20	EMPLEADO
30	DIRECTOR
30	EMPLEADO
30	VENDEDOR

6°.- Seleccionar todos los datos de los que sean directores

```
SELECT *
FROM EMPL
WHERE OFICIO = 'DIRECTOR';
```

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_AL	SALARIO
7566	JIMENEZ	DIRECTOR	7839	02/04/81	386750
7698	NEGRO	DIRECTOR	7839	01/05/81	370500
7782	CEREZO	DIRECTOR	7839	09/06/81	318500

7°.- Seleccionar los nombres y salarios de los empleados del departamento 20

```
SELECT APELLIDO, SALARIO
FROM EMPL
WHERE DEPT_NO = 20;
```

APELLIDO	SALARIO
SANCHEZ	104000
JIMENEZ	386750
GIL	390000
ALONSO	143000
FERNANDEZ	390000

8°.- Seleccionar los nombres,oficios,salarios y retención del 13% solo de los EMPLEADOS de la tabla

```
SELECT APELLIDO, SALARIO, SALARIO*0.13 RETENCION
FROM EMPL
WHERE OFICIO = 'EMPLEADO' ;
```

APELLIDO	OFICIO	SALARIO	RETENCION
SANCHEZ	EMPLEADO	104000	13520
ALONSO	EMPLEADO	143000	18590
JIMENO	EMPLEADO	123500	16055
MUÑOZ	EMPLEADO	169000	21970

9°.- Seleccionar los nombres, el mensaje "tiene un salario de " y salario de los directores de la tabla

```
SELECT APELLIDO, 'tiene un salario de' , SALARIO
FROM EMPL
WHERE OFICIO = 'DIRECTOR';
```

APELLIDO	'TIENEUNSALARIODE'	SALARIO
JIMENEZ	tiene un salario de	386750
NEGRO	tiene un salario de	370500
CEREZO	tiene un salario de	318500

10°.- Seleccionar los nombres y trabajo de los que no tengan por oficio EMPLEADO

```
SELECT APELLIDO, OFICIO
FROM EMPL
WHERE OFICIO != 'EMPLEADO'
```

APELLIDO	OFICIO
ARROYO	VENDEDOR
SALA	VENDEDOR
JIMENEZ	DIRECTOR
MARTIN	VENDEDOR
NEGRO	DIRECTOR
CEREZO	DIRECTOR
GIL	ANALISTA
REY	PRESIDENTE
TOVAR	VENDEDOR
FERNANDEZ	ANALISTA

11°.- Seleccionar los nombres, salario y oficio de las personas que ganen mas de 100000 pts y su oficio sea vendedor

```
SELECT APELLIDO, SALARIO, OFICIO
FROM EMPLE
WHERE SALARIO > 100000 AND OFICIO = 'VENDEDOR';
```

APELLIDO	SALARIO	OFICIO
ARROYO	208000	VENDEDOR
SALA	162500	VENDEDOR
MARTIN	182000	VENDEDOR
TOVAR	195000	VENDEDOR

12°.- Seleccionar los nombres, salario y departamento de las personas que ganen mas de 100000 pts o su departamento sea 10.

```
SELECT APELLIDO, SALARIO, DEPT_NO
FROM EMPLE
WHERE SALARIO > 100000 AND DEPT_NO = 10;
```

APELLIDO	SALARIO	DEPT_NO
CEREZO	318500	10
REY	650000	10
MUÑOZ	169000	10

13°.- Seleccionar los nombres, oficio y número de departamento de los empleados que sean analistas o que sean directores y del departamento n° 10

```
SELECT APELLIDO, OFICIO, DEPT_NO
FROM EMPLE
WHERE OFICIO = 'ANALISTA' OR ( OFICIO = 'DIRECTOR' AND DEPT_NO = 10);
```

APELLIDO	OFICIO	DEPT_NO
CEREZO	DIRECTOR	10
GIL	ANALISTA	20
FERNANDEZ	ANALISTA	20

14°.- Seleccionar los nombres, oficios y número de departamento de los empleados que sean analistas o directores y su departamento sea el 10

```
SELECT APELLIDO, OFICIO, DEPT_NO
FROM EMPLE
WHERE (OFICIO = 'ANALISTA' OR OFICIO = 'DIRECTOR') AND DEPT_NO = 10;
```

APELLIDO	OFICIO	DEPT_NO
----------	--------	---------

15°.- ! CUIDADO SI NO PONEMOS PARENTESIS EL AND SE EVALUA ANTES DEL OR ;

```
SELECT APELLIDO, OFICIO, DEPT_NO
FROM EMPL
WHERE OFICIO = 'ANALISTA' OR OFICIO = 'DIRECTOR' AND DEPT_NO =
10;
APELLIDO          OFICIO          DEPT_NO
-----
CEREZO            DIRECTOR          10
GIL               ANALISTA          20
FERNANDEZ         ANALISTA          20
```

16°.- Seleccionar el apellido, oficio y salario de los analistas, directores y presidente

```
SELECT APELLIDO, OFICIO, SALARIO
FROM EMPL
WHERE OFICIO IN ('DIRECTOR', 'ANALISTA', 'PRESIDENTE')
```

APELLIDO	OFICIO	SALARIO
-----	-----	-----
JIMENEZ	DIRECTOR	386750
NEGRO	DIRECTOR	370500
CEREZO	DIRECTOR	318500
GIL	ANALISTA	390000
REY	PRESIDENTE	650000
FERNANDEZ	ANALISTA	390000

17°.- Seleccionar el apellido, oficio y salario de los que no sean analistas directores ni el presidente

```
SELECT APELLIDO, OFICIO, SALARIO
FROM EMPL
WHERE OFICIO NOT IN ('DIRECTOR', 'ANALISTA', 'PRESIDENTE');
```

APELLIDO	OFICIO	SALARIO
-----	-----	-----
SANCHEZ	EMPLEADO	104000
ARROYO	VENDEDOR	208000
SALA	VENDEDOR	162500
MARTIN	VENDEDOR	182000
TOVAR	VENDEDOR	195000
ALONSO	EMPLEADO	143000
JIMENO	EMPLEADO	123500
MUÑOZ	EMPLEADO	169000

18°.- Seleccionar los nombres y comisiones de los empleados que no tienen comisiones (es nulo)

```
SELECT APELLIDO, COMISION
FROM EMPLE
WHERE COMISION IS NULL;
```

APELLIDO	COMISION
-----	-----
SANCHEZ	
JIMENEZ	
NEGRO	
CEREZO	
GIL	
REY	
ALONSO	
JIMENO	
FERNANDEZ	
MUÑOZ	

19°.- Seleccionar los nombres y comisiones de los empleados que si tienen comisiones

```
SELECT APELLIDO, COMISION
FROM EMPLE
WHERE COMISION IS NOT NULL;
```

APELLIDO	COMISION
-----	-----
ARROYO	39000
SALA	65000
MARTIN	182000
TOVAR	0

20°.- Seleccionar los números de departamento, nombres y salarios de los que su salario sea menor ó igual a 150000, 200000

```
SELECT DEPT_NO, APELLIDO, SALARIO
FROM EMPLE
WHERE SALARIO <= ALL (150000,200000)
```

DEPT_NO	APELLIDO	SALARIO
-----	-----	-----
20	SANCHEZ	104000
20	ALONSO	143000
30	JIMENO	123500

21°.- Seleccionar los nombres y salarios de los empleados que su nombre empiece por A

```
SELECT APELLIDO, SALARIO
FROM EMPLE
```

WHERE APELLIDO LIKE 'A%';

APELLIDO	SALARIO
ARROYO	208000
ALONSO	143000

22°.- Seleccionar los nombres y SALARIOS de los empleados que la tercera letra de su nombre sea R

**SELECT APELLIDO, SALARIOS
FROM EMPLE
WHERE APELLIDO LIKE '__R%';**

APELLIDO	SALARIO
ARROYO	208000
MARTIN	182000
CEREZO	318500
FERNANDEZ	390000

23°.- Seleccionar los nombres y salarios de los empleados que su nombre tenga una E y una O

**SELECT APELLIDO, SALARIO
FROM EMPLE
WHERE APELLIDO LIKE '%E%O%';**

APELLIDO	SALARIO
NEGRO	370500
CEREZO	318500
JIMENO	123500

EJERCICIOS DE SQL

24°.-Sacar los nombres salarios y fechas de alta de los vendedores y directores ordenados en orden ascendente de empleo y descendente de fecha de alta

```
SELECT SALARIO, FECHA_ALTA, OFICIO
FROM EMPLE
WHERE OFICIO IN ( 'VENDEDOR', 'DIRECTOR' )
ORDER BY OFICIO ASC, FECHA_ALTA DESC;
```

SALARIO	FECHA_AL	OFICIO
318500	09/06/81	DIRECTOR
370500	01/05/81	DIRECTOR
386750	02/04/81	DIRECTOR
182000	28/09/81	VENDEDOR
195000	08/09/81	VENDEDOR
162500	22/02/81	VENDEDOR
208000	20/02/80	VENDEDOR

25°.-Sacar los nombres salarios y fechas de alta de los vendedores y directores ordenados en orden ascendente de empleo y descendente de fecha de alta PERO USANDO POSICION

```
SELECT SALARIO, FECHA_ALTA, OFICIO
FROM EMPLE
WHERE OFICIO IN ( 'VENDEDOR', 'DIRECTOR' )
ORDER BY 3 DESC, 2;
```

SALARIO	FECHA_AL	OFICIO
208000	20/02/80	VENDEDOR
162500	22/02/81	VENDEDOR
195000	08/09/81	VENDEDOR
182000	28/09/81	VENDEDOR
386750	02/04/81	DIRECTOR
370500	01/05/81	DIRECTOR
318500	09/06/81	DIRECTOR

26°.-Sacar el sueldo y la fecha de alta de aquellos que su salario sea mayor de 200000 pts ordenados por el campo APELLIDO utilizando un alias para tabla con el que después nos referimos a sus campos y alias para las cabeceras

```
SELECT SALARIO Sueldo, FECHA_ALTA "Fecha de alta", APELLIDO
```

```
FROM EMPLE E
WHERE E.SALARIO > 200000
ORDER BY E.APELLIDO
```

cuidado el literal se corta!!

```

SUELDO Fecha de APELLIDO
-----
208000 20/02/80 ARROYO
318500 09/06/81 CEREZO
390000 03/12/81 FERNANDEZ
390000 30/03/87 GIL
386750 02/04/81 JIMENEZ
370500 01/05/81 NEGRO

```

27°.-Probar las funciones aritméticas y mandar la ejecución sobre la tabla

DUAL

```
SELECT SYSDATE
FROM DUAL;
```

```

SYSDATE
-----
28/10/99

```

```
SELECT ABS (-18)
FROM DUAL;
```

```

ABS (-18)
-----
18

```

```
SELECT MOD (7,2)
FROM DUAL;
```

```

MOD (7,2)
-----
1

```

```
SELECT CEIL(13.4)
FROM DUAL;
```

```

CEIL(13.4)
-----
14

```

```
SELECT FLOOR (11.3)
FROM DUAL;
```

```

FLOOR(11.3)
-----

```

```
SELECT POWER (4,3)
FROM DUAL;
```

```
POWER (4,3)
-----
          64
```

```
SELECT ROUND (12.19,1)
FROM DUAL;
```

```
ROUND (12.19,1)
-----
          12,2
```

```
SELECT SIGN (18)
FROM DUAL;
```

```
SIGN (18)
-----
          1
```

```
SELECT SQRT (24)
FROM DUAL;
```

```
SQRT (24)
-----
4,8989795
```

```
SELECT TRUNC (12.49,1)
FROM DUAL;
```

```
TRUNC (12.49,1)
-----
          12,4
```

28°.-Probar las funciones alfanuméricas hacia la tabla DUAL

```
SELECT CHR (65)
FROM DUAL;
```

```
C
-
A
```

```
SELECT ASCII (A)
FROM DUAL;
```

```
ASCII ('A')
-----
          65
```

```
SELECT INITCAP('la casa era verde'  
FROM DUAL;
```

```
INITCAP('LACASAER  
-----  
La Casa Era Verde
```

```
SELECT LOWER('CASA')  
FROM DUAL;
```

```
LOWE  
----  
casa
```

```
SELECT LPAD('CASA',7,'*')  
FROM DUAL;
```

```
LPAD('C  
-----  
***casa
```

```
SELECT LTRIM('ABABCASA','AB')  
FROM DUAL;
```

```
LTRI  
----  
CASA
```

```
SELECT REPLACE('roma re','r','t')  
FROM DUAL;
```

```
REPLACE  
-----  
toma te
```

```
SELECT RPAD('casa',7,'*')  
FROM DUAL;
```

```
RPAD('C  
-----  
casa***
```

```
SELECT RTRIM('casaxyxy','yx')  
FROM DUAL;
```

```
RTRI  
----  
casa
```

```
SELECT SUBSTR('ABCDEF',2,3)
FROM DUAL;
```

```
SUB
---
BCD
```

```
SELECT UPPER('casa')
FROM DUAL;
```

```
UPPE
----
CASA
```

```
SELECT INSTR('MISSISSIPI','SI',5,2)
FROM DUAL
```

```
INSTR('MISSISSIPI','SI',5,2)
-----
0
```

busca la 2ª ocurrencia de SI dentro de MISSISSIPI desde la posición 5ª y no existe

```
SELECT LENGTH ('Pepito se fue')
FROM DUAL;
```

```
LENGTH('PEPITOSEFUE')
-----
13
```

29°.-Visualizar número de departamento y nombre de los empleados así como los caracteres 3°,4° y 5° de los nombres siempre que la longitud del nombre sea mayor de 6

```
SELECT APELLIDO, DEPT_NO, SUBSTR(APELLIDO,3,3)
FROM EMPL
WHERE LENGTH(APELLIDO)>6;
```

```
APELLIDO          DEPT_NO SUB
-----
SANCHEZ           20 NCH
JIMENEZ           20 MEN
FERNANDEZ         20 RNA
```

30°.-Visualizar numero de cada empleado y longitud de su nombre siempre que sea mayor de 2 y menor de 10

```
SELECT EMP_NO, APELLIDO, LENGTH(APELLIDO)
FROM EMPLE
WHERE LENGTH(APELLIDO) BETWEEN 2 AND 10;
```

EMP_NO	APELLIDO	LENGTH(APELLIDO)
7369	SANCHEZ	7
7499	ARROYO	6
7521	SALA	4
7566	JIMENEZ	7
7654	MARTIN	6
7698	NEGRO	5
7782	CEREZO	6
7788	GIL	3
7839	REY	3
7844	TOVAR	5
7876	ALONSO	6
7900	JIMENO	6
7902	FERNANDEZ	9
7934	MUÑOZ	5

31°.-Averiguar la fecha del empleado más antiguo y la fecha de ingreso del más nuevo

```
SELECT MIN(FECHA_ALTA), MAX(FECHA_ALTA)
FROM EMPLE;
```

MIN(FECHA)	MAX(FECHA)
20/02/80	03/05/87

32°.-Calcular el salario máximo y mínimo de cada departamento

```
SELECT MIN(SALARIO), MAX(SALARIO), COUNT(*)
FROM EMPLE
WHERE DEPT_NO = 10;
```

MIN(SALARIO)	MAX(SALARIO)	COUNT(*)
169000	650000	3

Y ASI PARA CADA DEPARTAMENTO

33°.-Contar el número de empleos que contiene la tabla

```
SELECT COUNT(OFICIO)
FROM EMPLE;
```

COUNT(OFICIO)
14

34°.-¿Cuántos empleados trabajan en el departamento 30?

```
SELECT COUNT(*)
FROM EMPLE
WHERE DEPT_NO = 10;
```

```
COUNT(*)
-----
3
```

35°.-¿Cuántos empleados distintos aparecen en la tabla de empleados

```
SELECT COUNT(DISTINCT OFICIO)
FROM EMPLE;
```

```
COUNT(DISTINCTOFICIO)
-----
5
```

36°.-Calcular la media del salario de los trabajadores poniendo como

cabecera del resultado ➔ Media de sueldo

```
SELECT AVG(SALARIO) "Media de sueldo"
FROM EMPLE;
```

```
Media de sueldo
-----
270910,71
```

37°.-Seleccionar el número de departamento, el mínimo, el máximo, la media

de los salarios de cada departamento así como los empleados de cada uno

```
SELECT DEPT_NO, MIN(SALARIO), MAX(SALARIO), AVG(SALARIO),
COUNT(*)
FROM EMPLE
GROUP BY DEPT_NO;
```

DEPT_NO	MIN(SALARIO)	MAX(SALARIO)	AVG(SALARIO)	COUNT(*)
10	169000	650000	379166,67	3
20	104000	390000	282750	5
30	123500	370500	206916,67	6

38°.-Calcular el número de personas que desempeñen el mismo empleo en cada

departamento sacando también los trabajos y el n° de departamentos

```
SELECT COUNT(EMP_NO), OFICIO, DEPT_NO
```

```
FROM EMPLE
GROUP BY OFICIO, DEPT_NO;
```

COUNT(EMP_NO)	OFICIO	DEPT_NO
-----	-----	-----
2	ANALISTA	20
1	DIRECTOR	10
1	DIRECTOR	20
1	DIRECTOR	30
1	EMPLEADO	10
2	EMPLEADO	20
1	EMPLEADO	30
1	PRESIDENTE	10
4	VENDEDOR	30

39°.-Calcular el numero de personas que tengan distintos empleos, excepto el presidente. Listar por departamento y tipo de empleo

```
SELECT DEPT_NO, OFICIO, COUNT(*)
FROM EMPLE
WHERE OFICIO != 'PRESIDENTE'
GROUP BY DEPT_NO, OFICIO;
```

DEPT_NO	OFICIO	COUNT(*)
-----	-----	-----
10	DIRECTOR	1
10	EMPLEADO	1
20	ANALISTA	2
20	DIRECTOR	1
20	EMPLEADO	2
30	DIRECTOR	1
30	EMPLEADO	1
30	VENDEDOR	4

40°.-Selecciona todos los oficios que tengan mas de dos empleados en ese oficio

```
SELECT DEPT_NO, OFICIO, COUNT(*)
FROM EMPLE
GROUP BY DEPT_NO, OFICIO
HAVING COUNT (*) >2;
```

DEPT_NO	OFICIO	COUNT(*)
-----	-----	-----
30	VENDEDOR	4

41.-Selecciona el salario anual medio por empleo en cada departamento que tenga más de dos personas con dicho empleo

```
SELECT DEPT_NO, OFICIO, COUNT(*), AVG(SALARIO)*12
```



```

FROM EMPLE
GROUP BY DEPT_NO, OFICIO
HAVING COUNT (*) > 2

```

DEPT_NO	OFICIO	COUNT (*)	AVG(SALARIO)*12
30	VENDEDOR	4	2242500

42°.-Selecciona el trabajo desempeñado y el número de personas que lo realizan siempre y cuando el trabajo sea realizado por mas de una persona

```

SELECT OFICIO, COUNT(OFFICIO)
FROM EMPLE
GROUP BY OFICIO
HAVING COUNT(OFFICIO) > 1;

```

OFICIO	COUNT(OFFICIO)
ANALISTA	2
DIRECTOR	3
EMPLEADO	4
VENDEDOR	4

43.-Selecciona el numero de empleado, el nombre y el número de director de los trabajadores cuyo jefe sea NEGRO pero sin que salga el
➔ RECUPERACION JERARQUICA

```

SELECT EMP_NO, APELLIDO, DIR
FROM EMPLE
WHERE APELLIDO <> 'NEGRO'
CONNECT BY PRIOR EMP_NO = DIR
START WITH APELLIDO = 'NEGRO';

```

EMP_NO	APELLIDO	DIR
7499	ARROYO	7698
7521	SALA	7698
7844	TOVAR	7698
7900	JIMENO	7698

ENUNCIADO 1 SQL 2 EVA CON SOLUCIÓN(EXAMEN 2022 PRÁCTICA CLASE)

1.- Cuantas asignaturas hay?

```
SELECT IdAsignatura, Nombre
FROM ASIGNATURA
```

2.- Coste máximo, mínimo y medio de las asignaturas, utilizar alias para las columnas

```
SELECT MAX(CosteBasico) AS MAXIMO,
       MIN(CosteBasico) AS MINIMO,
       AVG(CosteBasico) AS MEDIA
FROM ASIGNATURA;
```

3.- Nombre de las asignaturas del segundo cuatrimestre que no sean de 6 créditos.

```
SELECT Nombre
FROM ASIGNATURA
WHERE Cuatrimestre = "2"
AND Creditos <> 6;
```

4.- Mostrar el id de los alumnos matriculados en la asignatura '150212' o bien en la '130113'. Es obligatorio que uses IN

```
SELECT IdAlumno
FROM ALUMNO_ASIGNATURA
WHERE IdAsignatura IN ( "150212", "130113");
```

5.- Mostrar el nombre y apellido de las personas cuyo apellido comience por 'G'

```
SELECT Nombre, Apellido
FROM PERSONA
WHERE Apellido LIKE "G*";
```

6.-Mostrar el nombre de las asignaturas cuyo coste por crédito sea de mayor de 8 euros

```
SELECT Nombre
FROM ASIGNATURA
WHERE CosteBasico/Credito > 8;
```

7.- Mostrar el nombre de las personas para las que se desconoce la fecha de nacimiento

```
SELECT Nombre
FROM PERSONA
WHERE FechaNacimiento = NULL;
```

8.- Listado de personas ordenadas por apellido descendente y nombre ascendente

```
SELECT Nombre, Apellido
FROM PERSONA
```

```
ORDER BY Apellido desc, Nombre asc;
```

Utilizar select anidados

9.- Coste global de cursar la titulación de Matemáticas si el coste de cada asignatura se incrementa un 7%

```
SELECT SUM(CosteBasico*1.07) AS NUEVO_COSTE
FROM ASIGNATURA
WHERE IdTitulacion = (SELECT IdTitulacion
FROM TITULACION
WHERE Nombre = "Matemáticas");
```

10.- Listado de asignaturas que tengan más créditos que "seguridad vial"

```
SELECT ASIGNATURA.Nombre AS ASIGNATURA_  
FROM ASIGNATURA  
WHERE Creditos > (SELECT Creditos  
FROM ASIGNATURA  
WHERE Nombre = "Seguridad Vial");
```

11.- Lista de las asignaturas en las que no se ha matriculado nadie

```
SELECT IdAsignatura
FROM ASIGNATURA
WHERE IDAsignatura NOT IN (SELECT DISTINCT(IdAsignatura)
                           FROM ALUMNO ASIGNATURA);
```

12.- Listado de las asignaturas cuyo coste es superior al coste medio de las asignaturas que no pertenecen a ninguna titulación

```
SELECT Nombre
FROM ASIGNATURA
WHERE COsteBasico > (SELECT AVG(CosteBasico)
FROM ASIGNATURA
WHERE IdTitulacion IS NULL);
```

13.- Listado de personas que viven en la ciudad que vive más gente.

[illegible]

Utilizar relación entre tablas y si es necesario combinar con select anidados y/o los conectores de condición, agrupamiento, ordenación...etc..

14.- Listado de los profesores que además son alumnos

```
SELECT Nombre, Apellido
  FROM PERSONA, PROFESOR, ALUMNO
 WHERE PERSONA.DNI = PROFESOR.DNI
 AND PERSONA.DNI = ALUMNO.DNI;
```

Otra forma de escribir la misma consulta:

```
SELECT Nombre, Apellido
  FROM PERSONA, PROFESOR, ALUMNO
 WHERE PERSONA.DNI = PROFESOR.DNI
 AND PROFESOR.DNI = ALUMNO.DNI;
```

15.- Listados de alumnos que son más viejos que el profesor de mayor edad

```
SELECT Nombre&" "Apellido AS ALUMNO
  FROM ALUMNO, PERSONA
 WHERE PERSONA.DNI = ALUMNO.DNI
 AND FechaNacimiento < (SELECT MIN(FechaNacimiento)
                        FROM PROFESOR, PERSONA
                        WHERE PROFESOR.DNI = PERSONA.DNI);
```

16.- Listado de asignaturas impartidas por profesores de Logroño (el teléfono de esta comunidad autónoma empieza por 941)

```
SELECT ASIGNATURA.Nombre
  FROM ASIGNATURA, PROFESOR, PERSONA
 WHERE ASIGNATURA.IdProfesor = PROFESOR.IdProfesor
 AND PROFESOR.DNI = PERSONA.DNI
 AND Telefono LIKE "941*";
```

17.- Clasifica las titulaciones por orden descendente en función del total de créditos necesarios para cursarlas.

```
SELECT TITULACION.Nombre
  FROM TITULACION, ASIGNATURA
 WHERE TITULACION.IdTitulacion = ASIGNATURA.IdTitulacion
 GROUP BY ASIGNATURA.IdTitulacion
 ORDER BY SUM(Creditos) DESC;
```

18.- Cuanto paga cada alumno por su matricula

```
SELECT IdAlumno, SUM(CosteBasico) AS COSTE_MATRICULA
FROM ALUMNO_ASIGNATURA, ASIGNATURA
WHERE ALUMNO_ASIGNATURA.IdAsignatura = ASIGNATURA.IdAsignatura
GROUP BY IdAlumno;
```

19.- visualizar la asignatura con más créditos, la media de créditos, la suma de créditos y la titulación a la que pertenecen, para titulaciones con más de una asignatura

```
SELECT MAX(Creditos) AS MAXIMO,
       AVG(Creditos) AS MEDIA,
       SUM(Creditos) AS TOTAL,
       TITULACION.Nombre AS TITULACION_
FROM ASIGNATURA, TITULACION
WHERE ASIGNATURA.IdTitulacion = TITULACION.IdTitulacion
GROUP BY Titulacion.Nombre
HAVING COUNT (ASIGNATURA.Nombre) > 1;
```

20.-Que alumnos está matriculado al menos en dos asignaturas. Ordenar el resultado alfabéticamente.

```
SELECT PERSONA.Nombre
FROM PERSONA, ALUMNO, ALUMNO_ASIGNATURA
WHERE PERSONA.DNI = ALUMNO.DNI
AND ALUMNO_ASIGNATURA.IdAlumno = ALUMNO.IdAlumno
GROUP BY PERSONA.Nombre, PERSONA.Apellido
HAVING COUNT (PERSONA.DNI) > 1
ORDER BY PERSONA.Nombre
```

21.- nombre de las titulaciones que contengan al menos una asignatura de más de 4,5 créditos

```
SELECT DISTINCT (TITULACION.Nombre) AS TITULACION
FROM TITULACION, ASIGNATURA
WHERE ASIGNATURA.IdTitulacion = TITULACION.IdTitulacion
AND Creditos > 4.5;
```

22.- listado de alumnos que nacieron antes que el profesor más joven

```
SELECT *
FROM PERSONA
WHERE FechaNacimiento < (SELECT MAX (FechaNacimiento)
                        FROM PERSONA, PROFESOR
                        WHERE PROFESOR.DNI = PERSONA.DNI);
```

23.-Listado de ciudades en las que ha nacido algún profesor y también algún alumno.

```
SELECT DISTINCT (Ciudad)
FROM PERSONA, PROFESOR
```

```
WHERE PERSONA.DNI = PROFESOR.DNI
AND Ciudad IN (SELECT Ciudad
               FROM PERSONA, ALUMNO
               WHERE PERSONA.DNI = ALUMNO.DNI);
```

Ejercicios de SQL sencillo 2

Se dispone de la tabla ALUMNOS2 con los datos de aquellos alumnos que están matriculados en el Instituto.

Lo primero que tenemos que hacer es modificar el sistema de tipo de variable a number , varchar 2 y date si queremos a varchar2

En caso de que no cambiemos el formato de date tendremos que modificar la carga de datos ya que por sistema es

AAAA/dd/mm

tendríamos que pasarlo a

DD/mm/aaaa

Código SQL para la creación de las tablas...

```
create table alumnos2

(expediente number(10) primary key,
 nombre varchar2(50),
 localidad varchar2(50),
 fecha_nac date,
 direccion varchar2(50),
 curso number(2),
 nivel varchar2(10),
 faltas number(3)
);

insert into alumnos2 values(123456,'Juan Miguel Soler
Bakero','Murcia','10/10/1995','Gran Vía 2 4a',1,'ESO',15);
insert into alumnos2 values(654321,'Laura Gómez
Fernández','Lorca','05/10/1995','Junterones 10
5B',2,'ESO',25);
insert into alumnos2 values(765432,'Beatriz Martínez
Hernández','Murcia','05/05/1993','Plaza Mayor 6
3B',3,'ESO',5);
insert into alumnos2 values(987654,'Diego Martín
Llorente','Alhama de Murcia','06/03/1990','Diego de la
Cierva 5 7A',1,'BACHILLER',34);
insert into alumnos2 values(445544,'Juan Francisco Cano
Riquelme','Murcia','07/01/1992','Plaza de Belluga 3
```

```

4A',4,'ESO',13);
insert into alumnos2 values(223322,'Raquel Riquelme
Rubio','Lorca','11/12/1990','San Juan 14
3B',1,'BACHILLER',7);
insert into alumnos2 values(9988877,'Cristina Sánchez
Bermejo','Murcia','03/09/1995','Torre de Romo 7',
1,'ESO',1);
insert into alumnos2 values(334455,'Pedro Jesús Rodríguez
Soler','Alhama de Murcia','03/10/1994','Camino de Badel
4',2,'ESO',11);
insert into alumnos2 values(334400,'Javier Ramáñez
Rodríguez','Murcia','05/07/1993','Gran Vía 4 3A',3,'ESO',0);
insert into alumnos2 values(993322,'Gema Rubio
Colero','Lorca','09/09/1992','Plaza Fuensanta 5
7A',1,'BACHILLER',19);
insert into alumnos2 values(554411,'Joaquín Hernández
González','Lorca','12/12/1991','Junterones 4
5A',2,'BACHILLER',14);

```

Realizar las siguientes consultas:

Mostrar/Ocultar todas las soluciones

1. Muestra todos los datos de todos los alumnos.

```

SELECT *
FROM alumnos2;

```

2. Muestra el nombre , localidad y fecha de nacimiento de todos los alumnos.

```

SELECT nombre, localidad, fecha_nac
FROM alumnos2;

```

3. Muestra el nombre de todos los alumnos. En el resultado de la consulta, la columna "Nombre" debe aparecer con la etiqueta "Nombre y apellidos" (ALIAS DE COLUMNAS).


```
SELECT nombre "Nombre y apellidos", fecha_nac
FROM alumnos2;
```

4. Obtén el nombre y el número de faltas multiplicado por 2 de todos los alumnos (COLUMNAS CALCULADAS). En el resultado de la consulta, la columna faltas debe aparecer con la etiqueta "Faltas de asistencia" (ALIAS DE COLUMNAS)

```
SELECT nombre, faltas * 2 "Faltas de asistencia"
FROM alumnos2;
```

5. Consulta los datos de los alumnos que son de Lorca.

```
SELECT *
FROM alumnos2
WHERE localidad = 'Lorca';
```

6. Obtén los datos de los alumnos que son de Murcia o Alhama de Murcia.

```
SELECT nombre, localidad
FROM alumnos2
WHERE localidad = 'Alhama de Murcia'
    OR localidad = 'Murcia';

SELECT nombre, localidad
FROM alumnos2
WHERE localidad IN ('Murcia', 'Alhama de Murcia')

SELECT nombre, localidad
FROM alumnos2
WHERE localidad LIKE '%murcia%';
```

7. Obtén los datos de los alumnos que son de Murcia y están en el primer curso de E.S.O.

```
SELECT *
FROM alumnos2
WHERE localidad = 'Murcia'
```

```
AND nivel = 'ESO'  
AND curso = 1;
```

8. Muestra los alumnos que son de Lorca, están en segundo curso de Bachillerato y tienen más de 10 faltas.

```
SELECT *  
FROM alumnos2  
WHERE localidad = 'Lorca'  
AND nivel = 'BACHILLER'  
AND curso = 2  
AND faltas > 10;
```

9. Obtén los datos de aquellos alumnos que son de Murcia ordenados por nombre.

```
SELECT *  
FROM alumnos2  
WHERE localidad = 'Murcia'  
ORDER BY nombre;
```

10. Obtén los datos de todos los alumnos ordenados por nivel, y dentro de cada nivel por curso.

```
SELECT nivel, nombre, curso  
FROM alumnos2  
ORDER BY nivel, curso;
```

11. Muestra los datos de aquellos alumnos que tengan más de 10 faltas en primer o segundo curso.

```
SELECT nombre, faltas, curso  
FROM alumnos2  
WHERE faltas > 10  
AND (curso = 1 OR curso = 2);
```

```
SELECT nombre, faltas, curso
FROM alumnos2
WHERE faltas > 10
AND curso IN (1, 2);
```

12. Muestra los datos de todos aquellos alumnos que tengan menos de 10 faltas en 3 o 4 curso y además sean de Murcia.

```
13.      SELECT nombre, faltas, curso, localidad
FROM alumnos2
WHERE faltas < 10 AND localidad = 'Murcia' AND
(curso = 3 or curso = 4);
```

```
SELECT *
FROM alumnos
WHERE faltas < 10 AND localidad = 'Murcia' AND curso
IN (3, 4);
```

14. Obtén con una consulta todos los cursos que hay sin repeticiones (DISTINCT).

```
SELECT DISTINCT curso
FROM alumnos2;

SELECT DISTINCT nivel
FROM alumnos2;

SELECT DISTINCT localidad
FROM alumnos2;
```

15. Obtén los datos de los alumnos que no tengan 10 faltas en 1o de E.S.O.

```
SELECT nombre, faltas, curso, nivel
FROM alumnos2
WHERE faltas <> 10
```

```
AND nivel = 'ESO'  
AND curso = 1;
```

16. Muestra los datos de aquellos alumnos cuyo nombre empiece por la letra 'B'.

```
SELECT nombre  
FROM alumnos2  
WHERE nombre LIKE 'B%';
```

17. Con una consulta obtén los alumnos que son de Murcia y cuyo nombre termina con una letra 'O'.

```
SELECT nombre, localidad  
FROM alumnos2  
WHERE nombre LIKE '%o'  
AND localidad = 'Murcia';
```

18. Muestra los datos de todos aquellos alumnos que están en 1o curso de E.S.O. Y tienen una letra 'U' en la segunda posición del nombre.

```
SELECT nombre, curso, nivel  
FROM alumnos2  
WHERE curso = 1  
AND nivel = 'ESO'  
AND nombre LIKE '_u%';
```

19. Obtén los datos de los alumnos cuya columna "Faltas" es nula.

```
SELECT nombre, faltas  
FROM alumnos2  
WHERE faltas IS NULL ;
```

20. Muestra los datos de aquellos alumnos que tienen entre 10 y 20 faltas (BETWEEN). Ordena el resultado por nombre.

```
21.      SELECT nombre, faltas
        FROM alumnos2
        WHERE faltas BETWEEN 10 AND 20
```

```
ORDER BY nombre;
```

22. Muestra los datos de los alumnos que tienen entre 10 y 20 faltas y además son de Murcia.

```
SELECT nombre, faltas, localidad
FROM alumnos2
WHERE faltas BETWEEN 10 AND 20
    AND localidad = 'Murcia';
```

23. Muestra los datos de los alumnos que tienen entre 10 y 20 faltas, son de Murcia y están matriculados en 1o de E.S.O.

```
24.      SELECT nombre, faltas, localidad, curso, nivel
        FROM alumnos2
        WHERE faltas BETWEEN 10 AND 20
            AND localidad = 'Murcia' AND curso = 1 AND nivel =
            'ESO';
```

25. Con una consulta muestra los datos de los alumnos cuyas faltas sean menores que 10 y mayores que 20.

```
SELECT nombre, faltas
FROM alumnos2
WHERE faltas NOT BETWEEN 10 AND 20;

SELECT nombre, faltas
FROM alumnos2
WHERE faltas < 10
    OR faltas > 20;
```

26. Muestra los datos de los alumnos cuya fecha de nacimiento comprenda los años 1993 y 1994. Ordena el resultado por nombre.

```
SELECT nombre, fecha_nac
FROM alumnos2
WHERE fecha_nac BETWEEN '1993-01-01' AND '1994-12-31'
ORDER BY nombre;
```

27. Muestra los datos de los alumnos que sean de primer o segundo curso (no importa que sean de E.S.O. O Bachiller). Utiliza el operador IN.

```
SELECT nombre, curso, nivel
FROM alumnos2
WHERE curso IN (1, 2);
```

28. Obtén los datos de aquellos alumnos que sean de tercer o cuarto curso y sean de Murcia. Utiliza el operador IN.

```
SELECT nombre, curso, localidad
FROM alumnos2
WHERE curso IN (3, 4)
AND localidad = 'Murcia';
```

29. Muestra los datos de los alumnos que no sean de E.S.O, ordenados por curso y por nombre descendentemente. Utiliza el operador IN.

```
SELECT nombre, curso, nivel
FROM alumnos2
WHERE nivel NOT IN ('ESO')
ORDER BY curso DESC, nombre DESC;
```

30. Muestra los datos de los alumnos que sean de primer o segundo curso y no sean de Bachiller. Ordena el resultado por nombre.

```
SELECT nombre, curso, nivel
FROM alumnos2
WHERE nivel <> 'BACHILLER'
AND (curso = 1 OR curso = 2)
ORDER BY nombre;
```

```
SELECT nombre, curso, nivel
FROM alumnos2
WHERE nivel NOT IN ('BACHILLER')
      AND curso IN (1, 2)
ORDER BY nombre;
```

31. Obtén los datos de los alumnos cuyo nombre empiece por la letra 'J', que tengan más de 10 faltas y no sean de Bachiller. Ordena el resultado por curso, y dentro de cada curso, por nombre.

```
SELECT nombre, faltas, curso, nivel
FROM alumnos2
WHERE nivel <> 'BACHILLER'
      AND faltas > 10
      AND nombre LIKE 'J%'
ORDER BY curso, nombre;

SELECT nombre, faltas, curso, nivel
FROM alumnos2
WHERE nivel <> 'BACHILLER'
      AND faltas > 10
      AND nombre >= 'J' AND nombre < 'K'
ORDER BY curso, nombre;
```

32. Con una consulta obtén el expediente, nombre, curso y nivel de todos los alumnos ordenado por curso, nivel ascendentemente y nombre descendentemente cuyo nivel no sea 'BACHILLER'.

```
SELECT expediente, nombre, curso, nivel
FROM alumnos2
WHERE nivel <> 'Bachiller'
ORDER BY curso, nivel, nombre DESC;
```

33. Con una consulta obtén el nombre de cada alumno en mayúscula cuya localidad sea Murcia.

```
SELECT upper(nombre)
FROM alumnos2
WHERE localidad = 'Murcia';
```

34. Muestra el nombre de cada alumno en mayúscula y la localidad a la que pertenecen en minúscula, ordenados por localidad.

```
SELECT upper(nombre), lower(localidad)
FROM alumnos2
ORDER BY localidad;
```

35. Con una consulta obtén el nombre de cada alumno y la longitud de la cadena "Nombre".

```
SELECT nombre, length(nombre)
FROM alumnos2;

SELECT nombre,
       length(nombre) numero_caracteres,
       replace(nombre, ' ', '') nombre_sin_espacios,
       length(replace(nombre, ' ', '')) numero_letras
FROM alumnos2;
```


TABLA ENUNCIADO Y SOLUCIÓN EJERCICIOS GLOBAL COMPLETO SQL

rem **tablas depart y emple**

```
CREATE TABLE DEPART (  
  DEPT NO    NUMBER(2) primary key,  
  DNOMBRE    VARCHAR2(14),  
  LOC        VARCHAR2(14) ) ;
```

```
INSERT INTO DEPART VALUES (10,'CONTABILIDAD','SEVILLA');  
INSERT INTO DEPART VALUES (20,'INVESTIGACIÓN','MADRID');  
INSERT INTO DEPART VALUES (30,'VENTAS','BARCELONA');  
INSERT INTO DEPART VALUES (40,'PRODUCCIÓN','BILBAO');  
INSERT INTO DEPART VALUES (50,'otros',null)
```

```
CREATE TABLE EMPLE (  
  EMP NO      NUMBER(4) primary key,  
  APELLIDO    VARCHAR2(10) ,  
  OFICIO      VARCHAR2(10) ,  
  DIR         NUMBER(4) ,  
  FECHA ALT   DATE ,  
  SALARIO     NUMBER(10),  
  COMISION    NUMBER(10),  
  DEPT NO     NUMBER(2),  
  foreign key (DEPT_NO) references depart;
```

```
INSERT INTO EMPLE VALUES  
(7369,'SÁNCHEZ','EMPLEADO',7902,'17/12/1980',104000,NULL,20);  
INSERT INTO EMPLE VALUES  
(7499,'ARROYO','VENDEDOR',7698,'20/02/1980',208000,39000,30);  
INSERT INTO EMPLE VALUES  
(7521,'SALA','VENDEDOR',7698,'22/02/1981',162500,65000,30);  
INSERT INTO EMPLE VALUES  
(7566,'JIMÉNEZ','DIRECTOR',7839,'02/04/1981',386750,NULL,20);  
INSERT INTO EMPLE VALUES  
(7654,'MARTÍN','VENDEDOR',7698,'29/09/1981',162500,182000,30);  
INSERT INTO EMPLE VALUES  
(7698,'NEGRO','DIRECTOR',7839,'01/05/1981',370500,NULL,30);  
INSERT INTO EMPLE VALUES  
(7782,'CEREZO','DIRECTOR',7839,'09/06/1981',318500,NULL,10);  
INSERT INTO EMPLE VALUES  
(7788,'GIL','ANALISTA',7566,'09/11/1981',390000,NULL,20);  
INSERT INTO EMPLE VALUES  
(7839,'REY','PRESIDENTE',NULL,'17/11/1981',50000,NULL,10);  
INSERT INTO EMPLE VALUES  
(7844,'TOVAR','VENDEDOR',7698,'08/09/1981',195000,0,30);  
INSERT INTO EMPLE VALUES  
(7876,'ALONSO','EMPLEADO',7788,'23/09/1981',143000,NULL,20);  
INSERT INTO EMPLE VALUES  
(7900,'JIMENO','EMPLEADO',7698,'03/12/1981',123500,NULL,30);  
INSERT INTO EMPLE VALUES  
(7902,'FERNÁNDEZ','ANALISTA',7566,'03/12/1981',390000,NULL,20);  
INSERT INTO EMPLE VALUES  
(7934,'MUÑOZ','EMPLEADO',7782,'23/01/1982',169000,NULL,10);
```

REM ** TABLA NOTAS_ALUMNOS: **

Drop table notas_alumnos cascade constraints;

```

create table notas_alumnos
(
    NOMBRE ALUMNO VARCHAR2(25) NOT NULL ,
    nota1 number(2),
    nota2 number(2),
    nota3 number(2)
) ;

insert into NOTAS ALUMNOS VALUES ('Alcalde García, M. Luisa',5,5,5);
insert into NOTAS ALUMNOS VALUES ('Benito Martín, Luis',7,6,8);
insert into NOTAS ALUMNOS VALUES ('Casas Martínez, Manuel',7,5,5);
insert into NOTAS ALUMNOS VALUES ('Corregidor Sánchez, Ana',6,9,8);
insert into NOTAS_ALUMNOS VALUES ('Díaz Sánchez, María',NULL,NULL,7);

```

REM ** TABLA LIBRERIA: **

```
Drop table LIBRERIA cascade constraints;
```

```

create table LIBRERIA
(TEMA CHAR(15) NOT NULL ,
  Estante CHAR(1),
  EJEMPLARES NUMBER(2)
);

```

```

INSERT INTO LIBRERIA VALUES ('Informática', 'A',15);
INSERT INTO LIBRERIA VALUES ('Economía', 'A',10);
INSERT INTO LIBRERIA VALUES ('Deportes', 'B',8);
INSERT INTO LIBRERIA VALUES ('Filosofía', 'C',7);
INSERT INTO LIBRERIA VALUES ('Dibujo', 'C',10);
INSERT INTO LIBRERIA VALUES ('Medicina', 'C',16);
INSERT INTO LIBRERIA VALUES ('Biología', 'A',11);
INSERT INTO LIBRERIA VALUES ('Geología', 'D',7);
INSERT INTO LIBRERIA VALUES ('Sociedad', 'D',9);
INSERT INTO LIBRERIA VALUES ('Labores', 'B',20);
INSERT INTO LIBRERIA VALUES ('Jardinería', 'E',6);

```

REM ** TABLAS ALUMNOS, ASIGNATURAS, NOTAS: **

```
DROP TABLE ALUMNOS cascade constraints;
```

```

CREATE TABLE ALUMNOS
(
    DNI VARCHAR2(10) NOT NULL,
    APENOM VARCHAR2(30),
    DIREC VARCHAR2(30),
    POBLA VARCHAR2(15),
    TELEF VARCHAR2(10)
);

```

```
DROP TABLE ASIGNATURAS cascade constraints;
```

```

CREATE TABLE ASIGNATURAS
(
    COD NUMBER(2) NOT NULL,
    NOMBRE VARCHAR2(25)
);

```

```
DROP TABLE NOTAS cascade constraints;
```

```
CREATE TABLE NOTAS
```

```
(
  DNI VARCHAR2(10) NOT NULL,
  COD NUMBER(2) NOT NULL,
  NOTA NUMBER(2)
);

INSERT INTO ASIGNATURAS VALUES (1,'Prog. Leng. Estr. ');
INSERT INTO ASIGNATURAS VALUES (2,'Sist. Informáticos ');
INSERT INTO ASIGNATURAS VALUES (3,'Análisis ');
INSERT INTO ASIGNATURAS VALUES (4,'FOL ');
INSERT INTO ASIGNATURAS VALUES (5,'RET ');
INSERT INTO ASIGNATURAS VALUES (6,'Entornos Gráficos ');
INSERT INTO ASIGNATURAS VALUES (7,'Aplic. Entornos 4ªGen ');

INSERT INTO ALUMNOS VALUES
('12344345','Alcalde García, Elena', 'C/Las Matas,
24','Madrid','917766545');

INSERT INTO ALUMNOS VALUES
('4448242','Cerrato Vela, Luis', 'C/Mina 28 - 3A', 'Madrid','916566545');

INSERT INTO ALUMNOS VALUES
('56882942','Díaz Fernández, María', 'C/Luis Vives 25',
'Móstoles','915577545');

INSERT INTO NOTAS VALUES('12344345', 1,6);
INSERT INTO NOTAS VALUES('12344345', 2,5);
INSERT INTO NOTAS VALUES('12344345', 3,6);

INSERT INTO NOTAS VALUES('4448242', 4,6);
INSERT INTO NOTAS VALUES('4448242', 5,8);
INSERT INTO NOTAS VALUES('4448242', 6,4);
INSERT INTO NOTAS VALUES('4448242', 7,5);

INSERT INTO NOTAS VALUES('56882942', 4,8);
INSERT INTO NOTAS VALUES('56882942', 5,7);
INSERT INTO NOTAS VALUES('56882942', 6,8);
INSERT INTO NOTAS VALUES('56882942', 7,9);
```

PRIMEROS EJEMPLOS DE CONSULTAS

=====

```
REM 1.- Seleccionar los empleados con salario superior a 2000
      select apellido from emple where salario > 2000;

REM 2.- Seleccionar los empleados con salario superior a 2000 y con el
trabajo de director
      select apellido from emple where salario > 2000 and OFICIO like
'DIRECTOR';

REM 3.- Seleccionar el nº de departamento que tengan distinto nombre
      select distinct apellido, dept_no from emple;

REM 4.- Seleccionar todos los empleados del departamento nº 20 y ademas
debe de estar ordenado por el apellido y tienen que aparecer en la
consulta el nº del empleado, el apellido, el oficio y el nº de
departamento.

      select emp no, apellido, oficio, dept_no from emple where dept_no =
20 order by apellido;
```

REM 5.- Empleados cuyo oficio sea analista, ordenado por el numero del empleado.

```
select apellido from emple where oficio like 'analista' order by emp_no;
```

REM 6.-Seleccionar las filas del departamento 10 cuyo oficio no sea analista. La consulta debe de estar ordenada descendientemente por el apellido y también descendientemente por el n° empleado.

```
select apellido, dept_no from emple where (oficio<>'analista'and dept_no = 10) order by emp_no desc , apellido desc;
```

REM 7.-Seleccionar la nota_media de los alumnos de la tabla notas_alumnos.

```
select nombre_alumno, (nota1+nota2+nota3)/3 from notas_alumnos;
```

REM 8.-Seleccionar los alumnos que de la nota1 hayan sacado un 7 y que la media de las 3 notas sea mayor que 6

```
select nombre alumno from notas_alumnos where nota1=7 and (nota1+nota2+nota3)/3>6;
```

REM 9.-Seleccionar los empelados que sean analistas

```
select apellido from emple where oficio like 'analista';
```

REM 10.- Seleccionar los empleados cuyo apellido tenga una A en la 2ª posición.

```
select apellido from emple where apellido like '_A%';
```

REM 11.- Seleccionar el apellido de los empleados cuyo oficio sea vendedor,analista o empleado

```
select apellido from emple where oficio like 'vendedor' or oficio like 'analista' or oficio like 'empleado';
```

REM 12.- Seleccionar el apellido, el salario y el n° de departamento de los empleados cuyo salario sea mayor o igual a 125000 en los departamentos 10 o 20.

```
select apellido, salario, dept_no from emple where salario > 125000 and dept_no in (10,20);
```

REM 13.- Seleccionar la lista de empleados indicando para cada uno su apellido, oficio, fecha de alta y el salario con un aumento del 16%.

```
select apellido, oficio, fecha_alt, salario* 1.16 aumento from emple;
```

REM 14.- De cada departamento saber el nombre y la localidad.

```
select dnombre , loc from depart;
```

REM 15.- seleccionar los departamentos agrupados por el n° departamento.

```
select * from depart order by dept_no asc;
```

REM 16.- seleccionar los empleados ordenados alfabéticamente por el nombre y por el oficio.

```
select * from emple order by apellido asc,oficio asc;
```

REM 17.- seleccionar los empleados que no tienen comisión.

```
select apellido from emple where comision = 0;
```

REM 18.- Seleccionar los empleados de apellido Muñoz.

```
select apellido from emple where apellido like 'MUÑOZ';
```

```

REM22.- Seleccionar los departamentos cuyo nombre acabe en 'on';
        select dnombre from depart where dnombre like '%on';

REM 19.- Seleccionar los empleados cuyo n° de departamento no sea ni 30,
ni 20 ni 40.
        b) select apellido from emple where dept_no not in(30,20,40);

REM 20.- Seleccionar los empleados cuya fecha de alta este entre el
8/9/61 y el 27/2/82.
        select apellido from emple where fecha_alt between '8/9/61' and
'27/2/82';

REM 21.- seleccionar los empleados cuyo salario sea mayor de 20000 y
menor que 50000.
        select apellido from emple where salario > 20000 and salario < 50000;

```

SEGUNDOS EJEMPLOS DE CONSULTAS

```

=====
REM 1.- Seleccionar el apellido de los empleados que trabajen en Madrid o
Barcelona.
        select apellido from emple where dept_no in
(select dept no from depart where
loc in ('MADRID','BARCELONA'));

REM 2.- Seleccionar el apellido y el oficio de todos los empleados del
departamento 20 cuyo trabajo sea idéntico al de los empleados del
departamento ventas.

        select apellido, oficio
from emple where dept_no = 20 and oficio in
( select oficio
from emple where dept_no =
(select dept no
from depart where dnombre = 'VENTAS'));

REM 3.- Obtener el departamento de los empleados con el mismo oficio y el
salario de GIL.

        select dept_no, apellido, salario
from emple
where (oficio, salario) =
(select oficio, salario
from emple where apellido = 'GIL');

REM 4.- Obtener el nombre de alumno, su asignatura y su nota.
        select apenom, nombre, nota
from alumnos
where alumnos.dni = notas.dni
and notas.cod = asignaturas.cod;

REM 5.- Nombre de los alumnos matriculados en FOL.
        select apenom from alumnos, asignaturas, notas
where alumnos.dni = notas.dni
and notas.cod = asignaturas.cod
and nombre = 'FOL';

```

REM 6.- seleccionar de la tabla emple aquellas filas cuyo apellido empiece por A y el oficio tenga una e en cualquier posición.

```
select apellido,oficio
from emple where
apellido like '%A' or oficio like '%E%';
```

REM 7.- Seleccionar el apellido, el oficio y la localidad de los departamentos donde trabajan los analistas.

```
select apellido,oficio,loc
from emple,depart
where oficio='ANALISTA';
```

REM 8.- Mostrar los empleados (nombre, oficio ,salario y fecha de alta) que desempeñen el mismo oficio que JIMENEZ o que tengan el salario mayor o igual a FERNANDEZ.

```
select apellido,oficio,salario,fecha_a_alt
from emple where
oficio = (select oficio
from emple where apellido = 'JIMENEZ')
or salario >=
(select salario
from emple where apellido = 'FERNANDEZ');
```

REM 9.- Mostrar por pantalla el nombre, el oficio y el salario de los empleados del departamento de FERNANDEZ que tengan su mismo salario.

```
select apellido,salario,oficio
from emple where (dept_no, salario)in
(select dept_no,salario from emple
where apellido = 'FERNANDEZ' );
```

REM 10.- Presentar los nombres y oficios de los empleados que tienen el mismo trabajo que JIMENEZ.

```
select apellido,oficio
from emple where OFICIO =
(select oficio
from emple where apellido = 'JIMENEZ');
```

REM111.- Visualizar los nombres de los alumnos que tengan una nota entre 7 y 8 en la asignatura de FOL.

```
select apenom,nombre,nota
from alumnos ,asignaturas,notas
where nombre ='FOL' and nota between 7 and 8 and
asignaturas.cod=notas.cod and notas.dni=alumnos.dni;
```

REM 12.- Visualizar los nombres de asignaturas que no tengan suspensos.

```
select nombre
from asignaturas, notas
where nota between 5 and 10
and asignaturas.cod=notas.cod;
```

REM 13.- Visualizar los nombres de alumnos de Madrid que tengan alguna asignatura suspendida.

```

        select apenom,direc,nota
from alumnos , notas where direc='Madrid'
and nota between 3 and 4;

```

REM 14.- Mostrar los nombres de alumnos que tengan la misma nota que tiene "Díaz Fernández, María" en FOL en alguna asignatura.

```

        select apenom from alumnos where dni in (select dni from notas
where nota =
(select nota from notas  where dni
= (select dni from alumnos
where apenom = 'Díaz Fernández. María')
and cod = (select cod
from asignaturas where nombre = 'FOL')));

```

TERCEROS EJEMPLOS DE CONSULTAS

=====

REM 1.- Visualizar los departamentos en los que el salario medio es mayor o igual que la media de todos los salarios;

```

        select dept_no, avg(salario)
from emple group by dept_no
having avg(salario) >=
(select avg(salario) from emple);

```

REM 2.- A partir de la tabla emple, visualizar el número de vendedores del departamentos 'VENTAS';

```

        select count(*) from emple
where dept_no =
(select dept_no
from depart where dnombre ='VENTAS'
and oficio = 'VENDEDOR');

```

REM 3.- Partiendo de la tabla EMPLE, visualizar por cada oficio de los empleados del departamento 'VENTAS' la suma de salarios.

```

        select sum(salario), oficio
from emple where dept_no in
(select dept_no from depart where dnombre like 'VENTAS')
group by oficio;

```

REM 4.- seleccionar aquellos apellidos de la tabla EMPLE cuyo salario sea igual a la media de su salario en su departamento.

```

        select apellido, salario
from emple where (salario,dept_no) in
(select avg(salario),dept_no
from emple group by dept_no);

```

REM 5.- A partir de la tabla emple, visualizar el numero de empleados de cada departamento cuyo oficio sea 'EMPLEADO'

```

        select dept_no,count(*)
from emple where oficio like 'EMPLEADO'
group by dept_no;

```

REM 6.- Desde la tabla EMPLE, visualizar el departamento que tenga más empleados cuyo oficio sea 'EMPLEADO'

```
select dept no, count(*)
from emple where oficio like 'EMPLEADO'
group by dept no
having count(*)=
(select max(count(*))
from emple where oficio
like 'EMPLEADO' group by dept_no);
```

REM 7.- A partir de las tablas EMPLE y DEPART, visualizar el número de departamento y el nombre de departamento que tenga más empleados cuyo oficio sea 'EMPLEADO'

```
select dept no,dnombre
from depart where dept no=
(select dept no from emple
where oficio= 'EMPLEADO'
group by dept_no having
count(*)=(select max(count(*))
from emple where oficio='EMPLEADO'
group by dept_no));
```

REM 8.- Buscar los departamentos que tienen más de dos personas trabajando en la misma profesión

```
select dept_no,count(*)
from emple group by dept_no,oficio
having count(*)>2;
```

REM 9.- Dada la tabla LIBRERIA, visualizar por cada estante la suma de los ejemplares.

```
select estante, sum(ejemplares)
from libreria group by estante;
```

REM 10.- Visualizar el estante con más ejemplares de la tabla libreria.

```
select estante,sum(ejemplares)
from libreria group by estante
having sum(ejemplares)=
(select max(sum(ejemplares))
from libreria group by estante);
```

CUARTOS EJEMPLOS DE CONSULTAS

=====

REM 1.- mostrar el oficio y media de salarios de aquellos empleados cuya media de salario sea mayor que 200000

```
select avg(salario),oficio
from emple group by oficio
having avg(salario)>200000;
```


REM 2.- mostrar el nombre y la comisión de aquellos empleados que tengan una comisión mayor que la de Sánchez

```
select apellido,comision
from emple where comision >
(select comision from emple
where apellido like 'SÁNCHEZ');
```

REM 3.- mostrar el nombre salario y n° de departamento de aquellos empleados que ganan el salario máximo de su departamento

```
select apellido,salario, dept_no
from emple where salario in
(select max(salario) from emple
group by dept_no);
```

REM 4.- mostrar el nombre del departamento que tanga mas empleados cuyo oficio sea presidente

```
select dnombre from depart
group by dnombre having count(*) =
(select max(count(*)) from emple
where oficio
like 'PRESIDENTE' group by dept_no);
```

REM 5.- mostrar el numero de directores de la tabla emple que sean dep departamento producción

```
select count(oficio) from emple
where oficio = 'DIRECTOR' and dept_no =
(select dept no from depart where
dnombre = 'PRODUCCIÓN');
```

REM 6.- A partir de la tabla emple, visualizar cuantos apellidos de los empleados empiezan por la letra 'A'

```
SELECT COUNT (APELLIDO)
from EMPLE WHERE APELLIDO LIKE 'A%';
```

REM 7.- Dada la tabla emple, obtener el sueldo medio, el número de comisiones no nulas, el máximo sueldo y el mínimo sueldo de los empleados del departamento 30. Emplear el formato adecuado para la salida y las cantidades adecuadas.

```
SELECT AVG(SALARIO), COUNT(COMISION),
MAX(SALARIO), MIN(SALARIO)
FROM EMPLE WHERE DEPT_NO= 30;
```

REM 8.- Contar las filas de librería cuyo tema tenga por lo menos una 'A'.

```
SELECT COUNT (TEMA) FROM LIBRERIA
WHERE TEMA LIKE '%A%' ;
```

REM 9.- Visualizar los temas con mayor número de ejemplares de la tabla librería y que tengan al menos una 'E' (pueden ser un tema o varios).

```
SELECT TEMA FROM LIBRERIA
WHERE EJEMPLARES =(SELECT MAX(EJEMPLARES)
```

```
FROM LIBRERIA )AND TEMA LIKE ('%E%');
```

REM 10.- Visualizar el número de estantes diferentes que hay en la tabla librería.

```
SELECT COUNT (DISTINCT ESTANTE )  
FROM LIBRERIA;
```

REM 11.- Visualizar el número de estantes distintos que hay en la tabla librería de aquellos temas que contienen al menos una 'E'.

```
SELECT COUNT (DISTINCT ESTANTE )  
FROM LIBRERIA WHERE TEMA LIKE ('%E%');
```

REM 12.- Dada la tabla librería, hacer una sentencia select que visualice el tema, el ultimo carácter del tema que no sea blanco y el número de caracteres de tema (sin contar los blancos de la derecha) ordenados por tema.

```
Select tema ,substr(tema,instr(tema,' ')-1),  
instr(tema,' ')-1 from libreria order by tema;
```

REM1 3.- Convertir la cadena '01051998' a fecha y visualizar su nombre de mes en mayúsculas.

```
SELECT UPPER(TO_CHAR(TO_DATE('01051998'),'MONTH'))  
) FROM DUAL;
```

REM 14.- A partir de la tabla emple, obtener el apellido de los empleados que lleven mas de 30 años trabajando.

```
SELECT APELLIDO FROM EMPLE WHERE 2022 -  
TO_NUMBER(TO_CHAR(FECHA_ALT,'YYYY'))>30;
```

REM 15.- Seleccionar el apellido de los empleados de la tabla emple que lleven mas de 18 años trabajando en el en el departamento 'ventas'.

```
SELECT APELLIDO FROM EMPLE WHERE 2004 - TO_NUMBER  
(TO_CHAR (FECHA_ALT,'YYYY'))>19 AND  
DEPT NO = (SELECT DEPT NO FROM DEPART  
WHERE DNOMBRE = 'VENTAS');
```

REM 16.- Visualizar el apellido, el salario y el número de departamento de aquellos empleados de la tabla emple cuyo salario sea el mayor de su departamento.

```
SELECT APELLIDO, SALARIO , DEPT NO  
FROM EMPLE E WHERE SALARIO IN  
(SELECT MAX(SALARIO) FROM EMPLE  
where DEPT_NO=E.DEPT_NO);
```

REM 17.- Visualizar el apellido, el salario y el número de departamento de aquellos empleados de la tabla emple cuyo salario supere a la media en su departamento.

```
SELECT APELLIDO,SALARIO,DEPT_NO  
FROM EMPLE E WHERE SALARIO>  
(SELECT AVG(SALARIO) FROM EMPLE
```

```
WHERE DEPT_NO=E.DEPT_NO);
```

REM 18.- visualizar los departamentos con mayor numero de empleados

```
select dnombre from depart
where dept no = (select dept_no from emple
where emp no =
(select max(emp_no) from emple));
```

y que tengan un salario mayor que 200000.

```
select dnombre from depart where dept no =
(select dept no from emple where emp_no =
(select max(emp no) from emple
where salario in (select salario
from emple where salario > 200000 )));
```

REM 19.- seleccionar los empleados que hayan trabajado mas de 5 años y sumarles 10000 euros a su salario.

```
select apellido, salario + 10000
from emple where months_between
(sysdate, fecha_alt)/12>5;
```

REM 20.- Dada la tabla emple, sumar dos meses a la fecha de alta.

```
select add_months(fecha_alt,2) from emple;
```

REM 21.- Dada la tabla emple, restar dos meses a la fecha de alta.

```
select months_between(fecha_alt,fecha_alt-2) from emple;
```

REM 22.- Obtener de la tabla emple el ultimo dia del mes para cada uno de las fechas de alta.

```
select last_day(fecha_alt) from emple;
```

REM 23.- A partir de la tabla emple obtener la fecha de alta de manera que aparezca el nombre del mes con todas sus letras el numero del dia del mes y el año.

```
select to_char(fecha_alt,'month,ddyyyy') from emple;
```

REM 24.- A partir de la tabla emple, obtener la fecha de alta de forma que aparezca el nombre del mes con tres

letras (month), el numero del día del año (ddd), el ultimo digito del año(y) y los tres digitos ultimos del año(yyy).

```
select to_char(fecha_alt,'mon ddd y yyy')from emple;
```

REM 25.- Obtener la fecha de hoy con el siguiente formato: Hoy es NOMBRE_DIA, DIA_MES de NOMBRE_MES de AÑO.

```
select to_char(sysdate,'"Hoy es " day "," dd "de" month "de" yyyy')
from dual;
```

REM 26.- Convertir la cadena 01012001 a tipo date.

```
select to_date('01012001')from dual;
```

```
REM 27.- Obtener el nombre del día, el nombre del mes, el día y el año en  
ingles a partir de la fecha '12121997'  
select to_char(to_date('12121997')), 'day, month dd yyyy' from dual;
```