



# Preparador Informática

[www.preparadorinformatica.com](http://www.preparadorinformatica.com)

PRÁCTICA 12

BASES DE DATOS

(SQL Avanzado 1ª parte)

### **EJERCICIO 33. CONSULTAS EN AGENDA**

33. Utilizando el ejercicio 30 de la práctica 11 denominado AGENDA vamos a realizar las siguientes consultas:

(Recuerda la estructura de la tabla era la siguiente:)

```
create table personas
(
    cod_hospital int(3),
    dni int(8) primary key,
    apellidos varchar(50),
    funcion varchar(30),
    salario float,
    localidad varchar(20)
);
```

- a) Mostrar los apellidos de las personas que vivan en MURCIA o LORCA.
- b) Mostrar los datos de las personas que vivan en MURCIA, tengan un salario superior a los 1500 euros y sean DIRECTORES.
- c) Seleccionar aquellas personas cuyo apellido comience por M.
- d) Mostrar aquellas personas que tengan un salario entre 1500 y 2000 euros.
- e) Obtén los apellidos en mayúsculas de las personas que trabajen en el hospital número 1.
- f) Obtener los apellidos y localidad en minúscula de todas aquellas personas que no trabajen en el hospital número 1.
- g) Visualizar los datos de aquellas personas que no trabajen en el hospital número 2 y que sean de MURCIA.
- h) Mostrar los datos de todas las localidades que hay en la tabla personas sin repeticiones
- i) Mostrar los datos de las personas cuya función sea MÉDICO ordenados por apellidos descendientemente.

### **EJERCICIO 33. CONSULTAS EN AGENDA. SOLUCIÓN PROPUESTA**

- a) **Mostrar los apellidos de las personas que vivan en MURCIA o LORCA.**

```
SELECT apellidos
FROM personas
WHERE localidad = 'lorca' OR localidad = 'murcia';
```

```
SELECT apellidos
FROM personas
WHERE localidad IN ('lorca', 'murcia');
```

- b) **Mostrar los datos de las personas que vivan en MURCIA, tengan un salario superior a los 1500 euros y sean DIRECTORES.**

```
SELECT *
FROM personas
WHERE localidad = 'murcia'
AND salario > 1500
AND funcion = 'director';
```

- c) **Seleccionar aquellas personas cuyo apellido comience por M.**

```
SELECT apellidos
FROM personas
WHERE apellidos LIKE 'M%';
```

- d) **Mostrar aquellas personas que tengan un salario entre 1500 y 2000 euros.**

```
SELECT *
FROM personas
WHERE salario BETWEEN 1500 AND 2000;
```

```
SELECT apellidos, salario
FROM personas
WHERE salario >= 1500 AND salario <= 2000;
```

- e) **Obtén los apellidos en mayúsculas de las personas que trabajen en el hospital número 1.**

```
SELECT upper(apellidos)
FROM personas
WHERE cod_hospital = 1;
```

- f) **Obtener los apellidos y localidad en minúscula de todas aquellas personas que no trabajen en el hospital número 1.**

```
SELECT apellidos, lower(localidad)
FROM personas
WHERE cod_hospital <> 1;
```

- g) **Visualizar los datos de aquellas personas que no trabajen en el hospital número 2 y que sean de MURCIA.**

```
SELECT *
FROM personas
WHERE cod_hospital <> 2 AND localidad = 'murcia';
```

- h) **Mostrar los datos de todas las localidades que hay en la tabla personas sin repeticiones**

```
SELECT DISTINCT localidad
FROM personas;
```

- i) **Mostrar los datos de las personas cuya función sea MÉDICO ordenados por apellidos descendientemente.**

```
SELECT *
FROM personas
WHERE funcion = 'medico'
ORDER BY apellidos DESC;
```

**EJERCICIO 34. EMPLEADOS**

34. Dadas las siguientes tablas piensa sin crear la tabla en SQL cómo realizarías las siguientes consultas:

Tabla **emple**

emp_no	apellido	oficio	dir	fecha_alt	salario	comision	dept_no
7369	SÁNCHEZ	EMPLEADO	7902	1990-12-17	1040	null	20
7499	ARROYO	VENDEDOR	7698	1990-02-20	1500	390	30
7521	SALA	VENDEDOR	7698	1991-02-22	1625	650	30
7566	JIMÉNEZ	DIRECTOR	7839	1991-04-02	2900	null	20
7654	MARTÍN	VENDEDOR	7698	1991-09-29	1600	1020	30
7698	NEGRO	DIRECTOR	7839	1991-05-01	3005	null	30
7782	CEREZO	DIRECTOR	7839	1991-06-09	2885	null	10
7788	GIL	ANALISTA	7566	1991-11-09	3000	null	20
7839	REY	PRESIDENTE	null	1991-11-17	4100	null	10
7844	TOVAR	VENDEDOR	7698	1991-09-08	1350	0	30
7876	ALONSO	EMPLEADO	7788	1991-09-23	1430	null	20
7900	JIMENO	EMPLEADO	7698	1991-12-03	1335	null	30
7902	FERNÁNDEZ	ANALISTA	7566	1991-12-03	3000	null	20
7934	MUÑOZ	EMPLEADO	7782	1992-01-23	1690	null	10

Tabla **depart**

dept_no	dnombre	loc
10	CONTABILIDAD	SEVILLA
20	INVESTIGACIÓN	MADRID
30	VENTAS	BARCELONA
40	PRODUCCIÓN	BILBAO

- Mostrar el apellido, oficio y número de departamento de cada empleado.
- Mostrar el número, nombre y localización de cada departamento.
- Datos de los empleados ordenados por número de departamento descendientemente y dentro de cada departamento ordenados por apellido ascendientemente.
- Mostrar los empleados que tengan un salario mayor que 2000 o que pertenezcan al departamento número 20.
- Seleccionar de la tabla EMPLE los empleados cuyo apellido termine por 'Z'.
- Número y apellidos de los empleados cuyo apellido termine por 'Z' y tengan un salario superior a 2000.
- Datos de los departamentos cuya localización empiece por 'B'.
- Seleccionar el apellido, salario y número de departamento de los empleados cuyo salario sea mayor que 2000 en los departamentos 10 ó 30.
- En una consulta concatena el apellido de cada empleado con su oficio.
- Mostrar el apellido y la longitud del apellido (función LENGTH) de todos los empleados, ordenados por la longitud de los apellidos de los empleados descendientemente.
- Obtener el año de contratación de todos los empleados (función YEAR).

### **EJERCICIO 34. EMPLEADOS. SOLUCIÓN PROPUESTA**

- a) **Mostrar el apellido, oficio y número de departamento de cada empleado.**

```
SELECT apellido, oficio, dept_no  
FROM emple;
```

- b) **Mostrar el número, nombre y localización de cada departamento.**

```
SELECT dept_no, dnombre, loc  
FROM depart;
```

- c) **Datos de los empleados ordenados por número de departamento descendente y dentro de cada departamento ordenados por apellido ascendentemente.**

```
SELECT *  
FROM emple  
ORDER BY dept_no DESC, apellido;
```

- d) **Mostrar los empleados que tengan un salario mayor que 2000 o que pertenezcan al departamento número 20.**

```
SELECT *  
FROM emple  
WHERE salario > 2000 OR dept_no = 20;
```

- e) **Seleccionar de la tabla EMPLE los empleados cuyo apellido termine por 'Z'.**

```
SELECT *  
FROM emple  
WHERE apellido LIKE '%Z';
```

- f) **Número y apellidos de los empleados cuyo apellido termine por 'Z' y tengan un salario superior a 2000.**

```
SELECT emp_no, apellido  
FROM emple  
WHERE apellido LIKE '%Z' AND salario > 2000;
```



- g) **Datos de los departamentos cuya localización empiece por 'B'.**

```
SELECT *  
FROM depart  
WHERE loc LIKE 'B%';
```

- h) **Seleccionar el apellido, salario y número de departamento de los empleados cuyo salario sea mayor que 2000 en los departamentos 10 ó 30.**

```
SELECT apellido, salario, dept_no  
FROM emple  
WHERE salario > 2000 AND (dept_no = 10 OR dept_no = 30);
```

```
SELECT apellido, salario, dept_no  
FROM emple  
WHERE salario > 2000 AND dept_no IN (10, 30);
```

- i) **En una consulta concatena el apellido de cada empleado con su oficio.**

```
SELECT concat(apellido, ' ', oficio) empleado_oficio  
FROM emple  
ORDER BY 1;
```

Preparador Informática

- j) **Mostrar el apellido y la longitud del apellido (función LENGTH) de todos los empleados, ordenados por la longitud de los apellidos de los empleados descendientemente.**

```
SELECT apellido, length(apellido)  
FROM emple  
ORDER BY length(apellido) DESC;  
SELECT apellido, length(apellido) largo  
FROM emple  
ORDER BY 2 DESC;
```

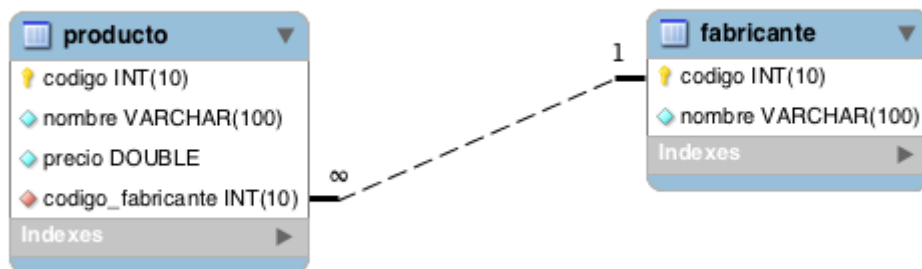
- k) **Obtener el año de contratación de todos los empleados (función YEAR).**

```
SELECT DISTINCT year(fecha_alt) año  
FROM emple;
```



**EJERCICIO 35. TIENDA DE INFORMÁTICA**

35. Dadas las siguientes tablas creadas en el ejercicio 21 de la práctica 9.



```
CREATE TABLE fabricante
(
    codigo INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL
);

CREATE TABLE producto
(
    codigo INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    precio DOUBLE NOT NULL,
    codigo_fabricante INT UNSIGNED NOT NULL,
    FOREIGN KEY (codigo_fabricante) REFERENCES
    fabricante(codigo)
);
```

Escribe cómo realizarías las siguientes consultas:

- Obtener los nombres y los precios de los productos de la tienda.
- Obtener el nombre de los productos cuyo precio sea menor o igual a 200€.
- Obtener todos los datos de los artículos cuyo precio esté entre los 60€ y los 120€ (ambas cantidades incluidas).
- Obtener el precio medio de los artículos cuyo código de fabricante sea 2.
- Obtener un listado completo de artículos, incluyendo por cada artículo los datos del artículo y de su fabricante.
- Obtener el precio medio de los productos de cada fabricante, mostrando sólo los códigos de fabricante.
- Obtener el nombre y precio del artículo más barato.
- Obtener una lista con el nombre y precio de los artículos más caros de cada proveedor (incluyendo el nombre del proveedor).



### **EJERCICIO 35. TIENDA DE INFORMÁTICA. SOLUCIÓN PROPUESTA**

- a) Obtener los nombres y los precios de los productos de la tienda.

```
SELECT nombre, precio FROM producto;
```

- b) Obtener el nombre de los productos cuyo precio sea menor o igual a 200€.

```
SELECT Nombre FROM producto WHERE Precio <= 200;
```

- c) Obtener todos los datos de los artículos cuyo precio esté entre los 60€ y los 120€ (ambas cantidades incluidas).

```
/* Con AND */
```

```
SELECT * FROM producto WHERE precio >= 60 AND precio <= 120;
```

```
/* Con BETWEEN */
```

```
SELECT * FROM producto WHERE precio BETWEEN 60 AND 120;
```

- d) Obtener el precio medio de los artículos cuyo código de fabricante sea 2.

```
SELECT AVG(precio) FROM producto WHERE codigo_fabricante =2;
```

- e) Obtener un listado completo de artículos, incluyendo por cada artículo los datos del artículo y de su fabricante.

```
/* Sin INNER JOIN */
```

```
SELECT * FROM producto, fabricante  
WHERE producto.codigo_fabricante = fabricante.codigo;
```

```
/* Con INNER JOIN */
```

```
SELECT * FROM producto INNER JOIN fabricante  
ON producto.codigo_fabricante = fabricante.codigo;
```

- f) Obtener el precio medio de los productos de cada fabricante, mostrando sólo los códigos de fabricante.

```
SELECT AVG(precio), codigo_fabricante  
FROM producto GROUP BY codigo_fabricante;
```

**g) Obtener el nombre y precio del artículo más barato.**

```
SELECT nombre, precio FROM producto
WHERE precio = (SELECT MIN(precio) FROM producto);
```

**h) Obtener una lista con el nombre y precio de los artículos más caros de cada proveedor (incluyendo el nombre del proveedor).**

```
/* Sin INNER JOIN */

SELECT P.nombre, P.precio, F.nombre
FROM producto P, fabricante F
WHERE P.codigo_fabricante = F.Codigo AND
P.precio = (SELECT MAX(P.precio) FROM producto P
           WHERE P.codigo_fabricante= F.codigo)

/* Con INNER JOIN */

SELECT P.nombre, P.precio, F.nombre
FROM producto P INNER JOIN fabricante F
ON P.codigo_fabricante= F.codigo
AND P.precio =(SELECT MAX(P.precio) FROM producto P
               WHERE P.codigo_fabricante= F.codigo)
```

Preparador Informática

### EJERCICIO 36. PELÍCULAS.

36. Dado el siguiente esquema:

PELÍCULAS (Codigo, Nombre, CalificacionEdad)

SALAS (Codigo, Nombre, Pelicula)



Debes realizar las siguientes consultas:

- a) Mostrar el nombre de todas las películas
- b) Mostrar las distintas calificaciones de edad que existen
- c) Mostrar todas las salas que no proyectan ninguna película
- d) Mostrar la información de todas las salas y, si se proyecta alguna película en la sala, mostrar también la información de la película
- e) Añadir una nueva película 'Uno, Dos, Tres', para mayores de 7 años
- f) Hacer constar que todas las películas no calificadas han sido calificadas 'no recomendables para menores de 13 años'

Preparador Informática

### **EJERCICIO 36. PELÍCULAS. SOLUCIÓN PROPUESTA**

- a) **Mostrar el nombre de todas las películas**

```
SELECT Nombre FROM PELICULAS;
```

- b) **Mostrar las distintas calificaciones de edad que existen**

```
SELECT DISTINCT CalificacionEdad FROM PELICULAS;
```

- c) **Mostrar todas las salas que no proyectan ninguna película**

```
SELECT * FROM SALAS WHERE Pelicula IS NULL;
```

- d) **Mostrar la información de todas las salas y, si se proyecta alguna película en la sala, mostrar también la información de la película**

```
SELECT * FROM SALAS LEFT JOIN PELICULAS  
ON SALAS.Pelicula = PELICULAS.Codigo;
```

- e) **Añadir una nueva película 'Uno, Dos, Tres', para mayores de 7 años**

```
INSERT INTO PELICULAS (Nombre, CalificacionEdad)  
VALUES ('Uno, Dos, Tres', 7)
```

- f) **Hacer constar que todas las películas no calificadas han sido calificadas 'no recomendables para menores de 13 años'**

```
UPDATE PELICULAS SET CalificacionEdad=13  
WHERE CalificacionEdad IS NULL
```

**EJERCICIO 37. HOSPITAL COMPLETO**

37. Dadas las siguientes tablas y datos de cada tabla.

**TABLA HOSPITAL**

COLUMNA	NULOS	TIPO
HOSPITAL_COD	NOT NULL	NUMBER(2)
NOMBRE		VARCHAR2(10)
DIRECCION		VARCHAR2(20)
TELEFONO		VARCHAR2(8)
NUM_CAMA		NUMBER(3)

HOSPITAL-COD	NOMBRE	DIRECCION	TELEFONO	NUM-CAMA
13	Provincial	0 Donell 50,	964-4264	502
18	General	Atocha s/n	595-3111	987
22	La Paz	Castellana 1000	923-5411	412
45	San Carlos	Ciudad Universitaria	597-1500	845

**TABLA SALA**

COLUMNA	NULOS	TIPO
HOSPITAL_COD	NOT NULL	NUMBER(2)
SALA_COD	NOT NULL	NUMBER(2)
NOMBRE		VARCHAR2(20)
NUM_CAMA		NUMBER(3)

HOSPITAL-COD	SALA-COD	NOMBRE	NUM-CAMA
13	3	Cuidados Intensivos	21
13	6	Psiquiátrico	67
18	3	Cuidados Intensivos	10
18	4	Cardiología	53
22	1	Recuperación	10
22	6	Psiquiátrico	118
22	2	Maternidad	34
45	4	cardiología	55
45	1	Recuperación	13
45	2	Maternidad	24

**TABLA PLANTILLA**

COLUMNA	NULOS	TIPO
HOSPITAL_COD	NOT NULL	NUMBER(2)
SALA_COD	NOT NULL	NUMBER(2)
EMPLEADO_NO	NOT NULL	NUMBER(4)
APELLIDO		VARCHAR2(15)
FUNCION		VARCHAR2(10)
TURNOS		VARCHAR2(1)
SALARIO		NUMBER(10)



HOSPITAL-COD	SALA-COD	EMPLEADO-NO	APELLIDO	FUNCION	TURNO	SALARIO
13	6	3754	Diaz B.	Enfermera	T	2262000
13	6	3106	Hernandez J.	Enfermero	T	2755000
18	4	6357	Karplus W.	Interno	T	3379000
22	6	1009	Higuera D.	Enfermera	T	2005000
22	6	8422	Bocina G.	Enfermero	M	1638000
22	2	9901	Nuñez C.	Interno	M	2210000
22	1	6065	Rivera G.	Enfermera	N	1626000
22	1	7379	Carlos R.	Enfermera	T	2119000
45	4	1280	Amigo R.	Interno	N	2210000
45	1	8526	Frank H.	Enfermera	T	2522000
22	2	1234	Garcia J.	Enfermo	M	3000000
2	22	2222	Garcia J			

TABLA OCUPACIÓN

COLUMNA	NULOS	TIPO
INSCRIPCION	NOT NULL	NUMBER(5)
HOSPITAL_COD	NOT NULL	NUMBER(2)
SALA_COD	NOT NULL	NUMBER(2)
CAMA		NUMBER(4)

INSCRIPCION	HOSPITAL_COD	SALA_COD	CAMA
10995	13	3	1
18004	13	3	2
14024	13	3	3
36658	18	4	1
38702	18	4	2
39217	22	6	1
59076	22	6	2
63827	22	6	3
64823	22	2	1

TABLA DOCTOR

COLUMNA	NULOS	TIPO
HOSPITAL_COD	NOT NULL	NUMBER(2)
DOCTOR_NO	NOT NULL	NUMBER(3)
APELLIDO		VARCHAR2(13)
ESPECIALIDAD		VARCHAR2(16)

HOSPITAL_COD	DOCTOR_NO	APELLIDO	ESPECIALIDAD
13	435	Lopez A.	Cardiología
18	585	Miller G.	Ginecología
18	982	Cajal R.	Cardiología
22	453	Galo D.	Pediatría
22	398	Best K.	Urología
22	386	Cabeza D.	Psiquiatría
45	607	Niqo P.	Pediatría

45

522

Adams C.

Neurología

## TABLA ENFERMO

COLUMNA	NULOS	TIPO
-----	-----	-----
INSCRIPCION	NOT NULL	NUMBER(5)
APELLIDO		VARCHAR2(15)
DIRECCION		VARCHAR2(20)
FECHA-NAC		DATE
S		VARCHAR2(1)
NSS		NUMBER(9)

INSCRIPCION	APELLIDO	DIRECCION	FECHA-NAC	S	NSS
-----	-----	-----	-----	-----	-----
10995	Laguia M.	Recoletos 50	23-JUN-67	M	280862482
18004	Serrano V.	Alcala 12	21-MAY-60	F	284991452
14024	Fernandez M	Recoletos 50	23-JUN-67	F	321790059
36658	Domin S.	Mayor 71	01-JAN-42	M	160657471
38702	Neal R.	Orense 11	18-JUN-40	F	380010217
39217	Cervantes M.	Peron 38	29-FEB-52	M	440294390
59076	Miller G.	Lopez de Hoyos 2	16-SEP-45	F	311969044
63827	Ruiz P.	Esquerdo 103	26-DEC-80	M	100973253
64823	Fraser A.	Soto 3	10-JUL-80	F	285201776
74835	Benitez E.	Argentina 5	05-OCT-57	M	154811767

## TABLA EMP

COLUMNA	NULOS	TIPO
-----	-----	-----
EMP_NO	NO	NUMBER(4)
APELLIDO		VARCHAR2(10)
OFICIO		VARCHAR2(10)
DIR		NUMBER(4)
FECHA_ALTA		DATE
SALARIO		NUMBER(10)
COMISIÓN		NUMBER(10)
DEPT_NO	NO	NUMBER(2)

EMP_NO	APELLIDO	OFICIO	DIR	FECHA_ALTA	SALARIO	COMISIÓN	DEPT_NO
-----	-----	-----	-----	-----	-----	-----	-----
7902	17-DEC-80	104000			20	7369	SANCHEZ EMPLEADO
7499	ARROYO	VENDEDOR	7698	20-FEB-81	208000	39000	30
7521	SALA	VENDEDOR	7698	22-FEB-81	162500	65000	30
7566	JIMENEZ	DIRECTOR	7839	02-APR-81	386750		20
7654	ARENAS	VENDEDOR	7698	28-SEP-81	162500	182000	30
7698	NEGRO	DIRECTOR	7839	01-MAY-81	370500		30
7782	CEREZO	DIRECTOR	7839	09-JUN-81	318500		10
7788	GIL	ANALISTA	7566	09-DEC-82	390000		20
7839	REY	PRESIDENTE		17-NOV-81	650000		10
7844	TOVAR	VENDEDOR	7698	08-SEP-81	195000	0	30
7876	ALONSO	EMPLEADO	7788	12-JAN-83	143000		20
7900	JIMENO	EMPLEADO	7698	03-DEC-81	123500		30
7902	FERNANDEZ	ANALISTA	7566	03-DEC-81	390000		20
	MUÑOZ	EMPLEADO	7782	23-JAN-82	169000		10



TABLA DEPT2

COLUMNA	NULOS	TIPO
DEPT_NO	NO	NUMBER(2)
DNOMBRE		VARCHAR2(14)
LOC	VARCHAR2(14)	

DEPT_NO	NOMBRE	LOC
10	CONTABILIDAD	MADRID
20	INVESTIGACIÓN	BILBAO
30	VENTAS	SEVILLA
40	OPERACIONES	MALAGA

Escribe cómo realizarías las siguientes consultas:

- Encuentre a todos los miembros del personal cuyo apellido empiece por 'H'.
- Quiénes son las enfermeras y enfermeros que trabajan en turnos de Tarde o Mañana
- Mostrar, para todos los hospitales, el código de hospital, el nombre completo del hospital y su nombre abreviado de tres letras (a esto podemos llamarlo ABR) Ordenar la recuperación por esta abreviatura.
- Encontrar el salario medio de los Analistas.
- Calcular el número de personas que realizan cada oficio en cada departamento.
- Buscar que departamentos tienen más de cuatro personas trabajando.
- Listar, a partir de las tablas EMP y DEPT2, el nombre de cada empleado, su oficio, su número de departamento y el nombre del departamento donde trabajan.
- Obtener el apellido, departamento y oficio de aquellos empleados que tengan un oficio que este en el departamento 20 y que no sea ninguno de los oficios que está en el departamento de VENTAS.
- Queremos averiguar el apellido del individuo mas antiguo de la empresa.
- Se desea saber el nombre, oficio y departamento del empleado que más gana del departamento con la media salarial más alta.
- Cambiar al paciente (tabla ENFERMO) número 74835 la dirección a Alcalá 411.
- Crear una vista para los departamentos 10 y 20.
- Crear una vista para los departamentos 10 y 30.
- Hacer una JOIN de las dos vistas anteriores.
- Se va a realizar un programa de consulta de la información sobre enfermos. Los datos a mostrar serán sus apellidos, dirección, fecha de nacimiento y hospital en el que se encuentran. ¿Qué vista se definirá?



### **EJERCICIO 37. HOSPITAL COMPLETO. SOLUCIÓN PROPUESTA**

- a) Encuentre a todos los miembros del personal cuyo apellido empiece por 'H'.

```
select apellido
from plantilla
where upper(apellido) like 'H%';
```

- b) Quienes son las enfermeras y enfermeros que trabajan en turnos de Tarde o Mañana

```
select apellido
from plantilla
where upper(funcion) in ('ENFERMERO' , 'ENFERMERA')
and upper(turno) in ('T', 'M');
```

- c) Mostrar, para todos los hospitales, el código de hospital, el nombre completo del hospital y su nombre abreviado de tres letras (a esto podemos llamarlo ABR) Ordenar la recuperación por esta abreviatura.

```
select substr(nombre,1,3) abr, hospital_cod, nombre
from hospital
order by 1;
```

- d) Encontrar el salario medio de los Analistas.

```
select avg(salario) "SALARIO MEDIO"
from emp
where upper(oficio) = 'ANALISTA';
```

- e) Calcular el número de personas que realizan cada oficio en cada departamento.

```
select dept_no, oficio, count(*)
from emp group by dept_no, oficio;
```

- f) Buscar que departamentos tienen más de cuatro personas trabajando.

```
select dept_no, count(*)
from emp
group by dept_no
having count(*) > 4;
```



- g) Listar, a partir de las tablas EMP y DEPT2, el nombre de cada empleado, su oficio, su número de departamento y el nombre del departamento donde trabajan.

```
select apellido, oficio, e.dept_no, dnombre
from emp e, dept2 d
where e.dept_no = d.dept_no;
```

- h) Obtener el apellido, departamento y oficio de aquellos empleados que tengan un oficio que este en el departamento 20 y que no sea ninguno de los oficios que está en el departamento de VENTAS.

```
select apellido, dept_no, oficio
from emp
where oficio in (select oficio
from emp
where dept_no = 20)
and oficio not in (select oficio
from emp e, dept2 d
where e.dept_no = d.dept_no
and upper(dnombre) = 'VENTAS');
```

- i) Queremos averiguar el apellido del individuo mas antiguo de la empresa.

```
select apellido, fecha_alt Fecha
from emp
where fecha_alt = (select min(fecha_alt)
from emp);
```

- j) Se desea saber el nombre, oficio y departamento del empleado que más gana del departamento con la media salarial más alta.

```
select apellido, oficio, dept_no
from emp e
where salario in (select max(salario)
from emp e2
where e.dept_no = e2.dept_no
group by dept_no)
```



```
having avg(salario) in  
(select max(avg(salario))  
from emp  
group by dept_no));
```

**Solución alternativa al apartado j)**

La parte que he marcado en negrita no es correcta en MySQL, no se puede utilizar una función de agregado sobre otra en MySQL. Para solucionarlo lo haríamos de la siguiente manera donde cambiamos esa línea errónea y lo que hacemos es una tabla intermedia donde metemos las medias de los salarios de empleados agrupados por departamentos y sobre esa cogemos el máximo salario:

```
mysql> select nomEmp, salEmp from Empleado e where salEmp in
```

```
(select max(salEmp) from Empleado e2  
where e.nroDpto=e2.nroDpto  
group by nroDpto  
having avg(salEmp) in  
(select max(avg_salary)  
from  
(select avg(salEmp) AS avg_salary  
from Empleado  
group by nroDpto) AS MAXSALARY));
```

**k) Cambiar al paciente (tabla ENFERMO) número 74835 la dirección a Alcala 411.**

```
update enfermo  
set direccion = 'Alcala 411'  
where inscripcion = 74835
```

- l) Crear una vista para los departamentos 10 y 20.
- m) Crear una vista para los departamentos 10 y 30.
- n) Hacer una JOIN de las dos vistas anteriores.

```
create view emp10 as
select *
from emp
where dept_no in (10,20)
create view emp30 as
select *
from emp
where dept_no in (10,30)
select e1.dept_no, e2.dept_no, e1.apellido, e2.oficio,
e1.salario, e1.comision
from emp10 e1, emp30 e2
where e1.dept_no = e2.dept_no
```

- o) Se va a realizar un programa de consulta de la información sobre enfermos. Los datos a mostrar serán sus apellidos, dirección, fecha de nacimiento y hospital en el que se encuentran. ¿Qué vista se definirá?

```
create view enferm
as select e.apellido, e.direccion, e.fecha_nac, h.nombre
from enfermo e, hospital h, ocupación o
where e.inscripcion = o.inscripcion and
o.hospital_cod = h.hospital-cod
```