

### Ejercicio 3

a) CREATE DATABASE

EXAMEN1; USE EXAMEN1;

CREATE TABLE EQUIPOS (

CodEquipo            VARCHAR(4) PRIMARY

KEY, Nombre            VARCHAR(30),

Localidad            VARCHAR(15),

Prima            int

) Engine=InnoDB;

CREATE TABLE JUGADORES (

CodJugador            VARCHAR (4) PRIMARY KEY,

DNI            VARCHAR (9),

Nombre            VARCHAR (30),

FechaNacimiento    DATE,

Demarcacion            VARCHAR (10),

Ficha            INT,

CodEquipo            VARCHAR(4),

FOREIGN KEY (CodEquipo) REFERENCES EQUIPOS(CodEquipo) ON DELETE CASCADE

) Engine=InnoDB;

CREATE TABLE PARTIDOS (

CodPartido            VARCHAR (4) PRIMARY KEY,

CodEquipoLocal            VARCHAR(4),

CodEquipoVisitante    VARCHAR(4),

Fecha            DATE,

Demarcacion            VARCHAR (10),

Competicion            VARCHAR(4),

FOREIGN KEY (CodEquipoLocal) REFERENCES EQUIPOS(CodEquipo) ON DELETE CASCADE,

FOREIGN KEY (CodEquipoVisitante) REFERENCES EQUIPOS(CodEquipo) ON DELETE CASCADE

) Engine=InnoDB;

**b)Añadir las siguientes restricciones a las tablas**

**1. La competicion puede ser Liga o Copa**

```
ALTER TABLE PARTIDOS ADD CONSTRAINT chk_competicion CHECK (Competicion IN ('Liga', 'Copa'));
```

**2. El nombre del equipo debe escribirse con mayúsculas**

**3. Crea un atributo en la tabla Equipos que indique el número de títulos con las restricción de que debe ser mayor de 2 (Dos instrucciones)**

```
ALTER TABLE EQUIPOS ADD Títulos INT;
```

```
ALTER TABLE EQUIPOS ADD CONSTRAINT chk_titulos CHECK (Títulos > 2);
```

**4. Borra el atributo jornada de la tabla Partidos**

```
ALTER TABLE PARTIDOS DROP COLUMN jornada;
```

**5. Borra la tabla equipo teniendo en cuenta la integridad referencial (4 instrucciones) Eliminar la tabla JUGADORES**

```
DROP TABLE IF EXISTS JUGADORES;
```

**Eliminar la tabla PARTIDOS**

```
DROP TABLE IF EXISTS PARTIDOS;
```

**Finalmente, eliminar la tabla EQUIPOS**

```
DROP TABLE IF EXISTS EQUIPOS;
```

#### Ejercicio 4

```
CREATE TABLE PERSONAL(  
  COD_CENTRO INT(4) NOT NULL,  
  DNI INT(10) NOT NULL,  
  APELLIDOS VARCHAR(30),  
  FUNCION VARCHAR(15),  
  SALARIO INT(10)  
)ENGINE=InnoDB;
```

```
INSERT INTO PERSONAL VALUES (10,1112345,'Martinez Salas, Fernando',  
'PROFESOR', 220000);  
INSERT INTO PERSONAL VALUES (10,4123005,'Bueno Zarco, Elisa',  
'PROFESOR', 220000);  
INSERT INTO PERSONAL VALUES (10,4122025,'Montes García, M.Pilar',  
'PROFESOR', 220000);  
INSERT INTO PERSONAL VALUES (15,1112346,'Rivera Silvestre, Ana',  
'PROFESOR', 205000);  
INSERT INTO PERSONAL VALUES (15,9800990, 'Ramos Ruiz, Luis',  
'PROFESOR', 205000);  
INSERT INTO PERSONAL VALUES (15,8660990, 'De Lucas Fdez, M.Angel',  
'PROFESOR', 205000);  
INSERT INTO PERSONAL VALUES (22,7650000, 'Ruiz Lafuente, Manuel',  
'PROFESOR', 220000);  
INSERT INTO PERSONAL VALUES (45,43526789, 'Serrano Laguia, Maria',  
'PROFESOR', 205000);  
INSERT INTO PERSONAL VALUES (10,4480099,'Ruano Cerezo, Manuel',  
'ADMINISTRATIVO', 180000);  
INSERT INTO PERSONAL VALUES (15,1002345,'Albarran Serrano, Alicia',  
'ADMINISTRATIVO', 180000);  
INSERT INTO PERSONAL VALUES (15,7002660,'Munoz Rey, Felicia',  
'ADMINISTRATIVO', 180000);  
INSERT INTO PERSONAL VALUES (22,5502678,'Marin Marin, Pedro',  
'ADMINISTRATIVO', 180000);  
INSERT INTO PERSONAL VALUES (22,6600980, 'Peinado Gil, Elena',  
'CONSERJE', 175000);  
INSERT INTO PERSONAL VALUES (45,4163222, 'Sarro Molina, Carmen',  
'CONSERJE', 175000);
```

```
DROP TABLE IF EXISTS PROFESORES;
```

```
CREATE TABLE PROFESORES (  
  COD_CENTRO INT(4) NOT NULL,  
  DNI INT(10) NOT NULL,  
  APELLIDOS VARCHAR(30),  
  ESPECIALIDAD VARCHAR(16)  
)ENGINE=InnoDB;
```

```
INSERT INTO PROFESORES VALUES (10,1112345,'Martinez Salas, Fernando',  
'INFORMATICA');
```

```
INSERT INTO PROFESORES VALUES (10,4123005,'Bueno Zarco, Elisa',  
'MATEMATICAS');
```

```
INSERT INTO PROFESORES VALUES (10,4122025,'Montes Garcia, M.Pilar',  
'MATEMATICAS');
```

```
INSERT INTO PROFESORES VALUES (15,9800990, 'Ramos Ruiz, Luis',  
    'LENGUA');
```

```
INSERT INTO PROFESORES VALUES (15,1112346,'Rivera Silvestre, Ana',  
    'DIBUJO');
```

```
INSERT INTO PROFESORES VALUES (15,8660990, 'De Lucas Fdez, M.Angel',  
'LENGUA');
```

```
INSERT INTO PROFESORES VALUES (22,7650000, 'Ruiz Lafuente, Manuel',  
'MATEMATICAS');
```

```
INSERT INTO PROFESORES VALUES (45,43526789, 'Serrano Laguia, Maria',  
'INFORMATICA');
```

```
DROP TABLE IF EXISTS CENTROS;
```

```
CREATE TABLE CENTROS (  
    COD_CENTRO  INT(4) NOT NULL,  
    TIPO_CENTRO CHAR(1),  
    NOMBRE     VARCHAR(30),  
    DIRECCION  VARCHAR(26),  
    TELEFONO   VARCHAR(10),  
    NUM_PLAZAS INT(4)  
)ENGINE=InnoDB ;
```

```
INSERT INTO CENTROS VALUES (10,'S','IES El Quijote',  
'Avda. Los Molinos 25', '965-887654',538);
```

```
INSERT INTO CENTROS VALUES (15,'P','CP Los Danzantes', 'C/Las Musas s/n',  
'985-112322',250);
```

```
INSERT INTO CENTROS VALUES (22,'S', 'IES Planeta Tierra', 'C/Mina 45',  
'925-443400',300);
```

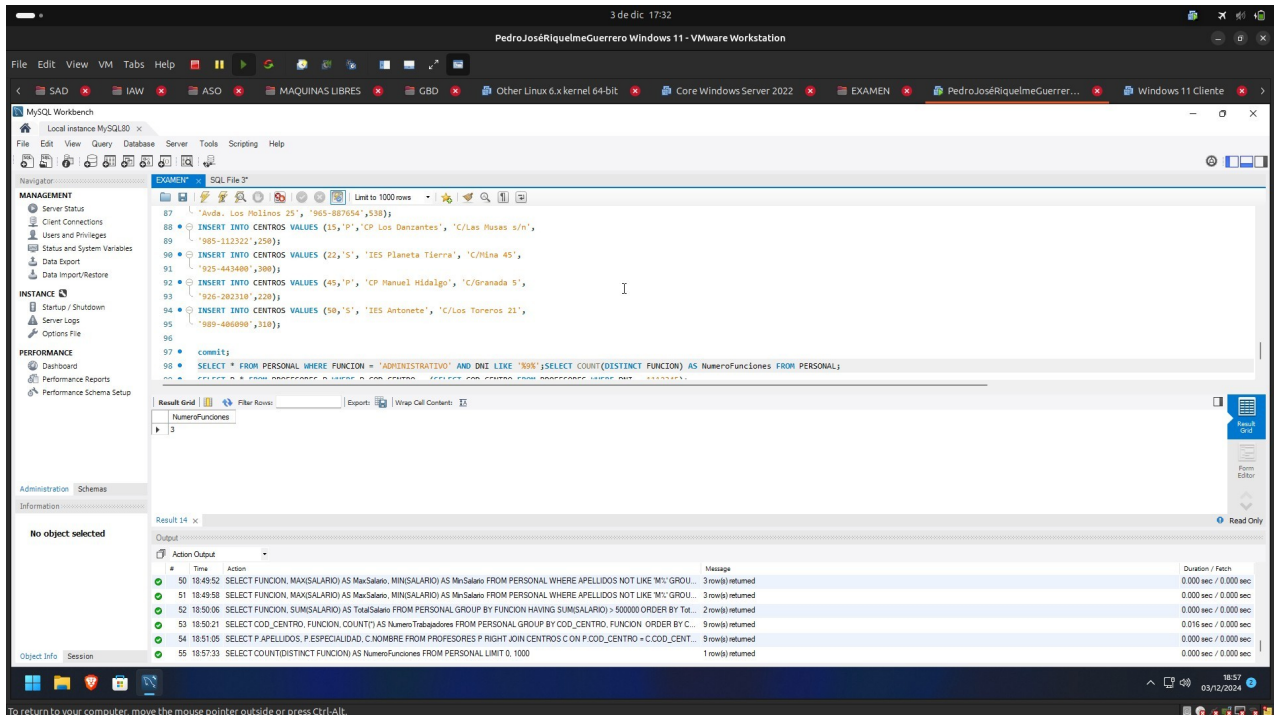
```
INSERT INTO CENTROS VALUES (45,'P', 'CP Manuel Hidalgo', 'C/Granada 5',  
'926-202310',220);
```

```
INSERT INTO CENTROS VALUES (50,'S', 'IES Antonete', 'C/Los Toreros 21',  
'989-406090',310);
```

```
Commit;
```

1 Muestra todos los datos de los administrativos (tabla PERSONAL) que en su DNI tenga un 9

SELECT \* FROM PERSONAL WHERE FUNCION = 'ADMINISTRATIVO' AND DNI LIKE '%9%';SELECT COUNT(DISTINCT FUNCION) AS NumeroFunciones FROM PERSONAL;



## 2. Muestra el numero de FUNCION distintas de la tabla personal

SELECT P.\* FROM PROFESORES P WHERE P.COD\_CENTRO = (SELECT COD\_CENTRO FROM PROFESORES WHERE DNI = 1112345);

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```
88 INSERT INTO CENTROS VALUES (15,'P','CP Los Dencantes', 'C/Las Russas s/n',  
89 '985-112322',250);  
90 INSERT INTO CENTROS VALUES (22,'S','IES Planeta Tierra', 'C/Mina 45',  
91 '925-443488',300);  
92 INSERT INTO CENTROS VALUES (45,'P','CP Manuel Hidalgo', 'C/Granada 5',  
93 '926-282319',220);  
94 INSERT INTO CENTROS VALUES (50,'S','IES Antonete', 'C/Los Toreros 21',  
95 '989-486090',310);  
96  
97 commit;  
98 SELECT * FROM PERSONAL WHERE FUNCION = 'ADMINISTRATIVO' AND DNI LIKE '80%';SELECT COUNT(DISTINCT FUNCION) AS NumeroFunciones FROM PERSONAL;  
99 SELECT P.* FROM PROFESORES P WHERE P.COD_CENTRO = (SELECT COD_CENTRO FROM PROFESORES WHERE DNI = 1112345);  
100
```

The Results Grid shows the following data:

Row	COD_CENTRO	DNI	APELLIDOS	ESPECIALIDAD
10	111234	111234	Marquez Siles, Fernando	INFORMATICA
10	412305	412305	Bueno Zarco, Elisa	MATEMATICAS
10	4122025	4122025	Montes Garcia, M.Pilar	MATEMATICAS

The Output tab shows the following messages:

#	Time	Action	Message	Duration / Fetch
51	18:49:58	SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'M%'; GROUP BY FUNCION	3 row(s) returned	0.000 sec / 0.000 sec
52	18:50:06	SELECT FUNCION, SUM(SALARIO) AS TotalSalario FROM PERSONAL GROUP BY FUNCION HAVING SUM(SALARIO) > 500000 ORDER BY TotalSalario	2 row(s) returned	0.000 sec / 0.000 sec
53	18:50:21	SELECT COD_CENTRO, FUNCION, COUNT(*) AS NumeroTrabajadores FROM PERSONAL GROUP BY COD_CENTRO, FUNCION ORDER BY COD_CENTRO	9 row(s) returned	0.016 sec / 0.000 sec
54	18:51:05	SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO	9 row(s) returned	0.000 sec / 0.000 sec
55	18:51:33	SELECT COUNT(DISTINCT FUNCION) AS NumeroFunciones FROM PERSONAL LIMIT 0.1000	1 row(s) returned	0.000 sec / 0.000 sec
56	18:58:05	SELECT P.* FROM PROFESORES P WHERE P.COD_CENTRO = (SELECT COD_CENTRO FROM PROFESORES WHERE DNI = 1112345) LIMIT 1	3 row(s) returned	0.000 sec / 0.000 sec

### 3. Muestra los profesores que trabajan en el mismo centro que el PROFESOR cuyo DNI es '1112345'

SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'M%' GROUP BY FUNCION; SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'M%' GROUP BY FUNCION;

The screenshot shows the MySQL Workbench interface. The SQL Editor contains a query that filters for the professor with DNI '1112345' and finds other professors in the same center. The query is as follows:

```
SELECT P1.FUNCION, MAX(P1.SALARIO) AS MaxSalario, MIN(P1.SALARIO) AS MinSalario FROM PERSONAL P1 WHERE P1.APELLIDOS NOT LIKE 'M%' GROUP BY FUNCION; SELECT P2.FUNCION, MAX(P2.SALARIO) AS MaxSalario, MIN(P2.SALARIO) AS MinSalario FROM PERSONAL P2 WHERE P2.APELLIDOS NOT LIKE 'M%' GROUP BY FUNCION;
```

The Results window shows the following data:

FUNCION	MaxSalario	MinSalario
PROFESOR	220000	200000
ADMINISTRATIVO	180000	180000
CONSERJE	175000	175000

The Output window shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
52	18:50:06	SELECT FUNCION, SUM(SALARIO) AS TotalSalario FROM PERSONAL GROUP BY FUNCION HAVING SUM(SALARIO) > 500000 ORDER BY TotalSalario	2 row(s) returned	0.000 sec / 0.000 sec
53	18:50:21	SELECT COD_CENTRO, FUNCION, COUNT(*) AS NumeroTrabajadores FROM PERSONAL GROUP BY COD_CENTRO, FUNCION ORDER BY COD_CENTRO	9 row(s) returned	0.016 sec / 0.000 sec
54	18:51:05	SELECT P1.APELLIDOS, P1.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P1 RIGHT JOIN CENTROS C ON P1.COD_CENTRO = C.COD_CENTRO	9 row(s) returned	0.000 sec / 0.000 sec
55	18:51:30	SELECT COUNT(DISTINCT FUNCION) AS NumeroFunciones FROM PERSONAL LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
56	18:58:09	SELECT P1 FROM PROFESORES P1 WHERE P1.COD_CENTRO = (SELECT COD_CENTRO FROM PROFESORES WHERE DNI = '1112345') LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
57	18:58:26	SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'M%' GROUP BY FUNCION	3 row(s) returned	0.000 sec / 0.000 sec

**4. A partir de la tabla PERSONAL muestra el maximo salario y el minimo salario para cada FUNCION, pero sin tener en cuenta a aquellos cuyo primer apellido comience por 'M'. Utiliza MAX() y MIN()**

SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'M%' GROUP BY FUNCION; SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'M%' GROUP BY FUNCION;

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```

90 INSERT INTO CENTROS VALUES (21,'S', 'IES Planeta Tierra', 'C/Hima 45',
91 '925-443480',300);
92 INSERT INTO CENTROS VALUES (45,'P', 'CP Manuel Hidalgo', 'C/Granada 5',
93 '926-282310',220);
94 INSERT INTO CENTROS VALUES (59,'S', 'IES Antoneta', 'C/Los Toreros 21',
95 '989-486890',310);
96
97 commit;
98 SELECT * FROM PERSONAL WHERE FUNCION = 'ADMINISTRATIVO' AND DNI LIKE '11%';SELECT COUNT(DISTINCT FUNCION) AS NumeroFunciones FROM PERSONAL;
99 SELECT P.* FROM PROFESORES P WHERE P.COD_CENTRO = (SELECT COD_CENTRO FROM PROFESORES WHERE DNI = 1112345);
100 SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'M%' GROUP BY FUNCION;SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELL:
101 SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO;

```

The Results Grid shows the following data:

APELLIDOS	ESPECIALIDAD	NOMBRE
Montes García, M.Pilar	MATEMATICAS	IES El Quijote
Buato Zazo, Elia	MATEMATICAS	IES El Quijote
Martínez Salas, Fernando	INFORMATICA	IES El Quijote
De Lucas Fdez, M.Ángel	LENGUA	CP Los Cantantes
Rivera Silvestre, Ana	DEBIDO	CP Los Cantantes
Ramos Ruiz, Luis	LENGUA	CP Los Cantantes
Ruiz Lafuente, Manuel	MATEMATICAS	IES Planeta Tierra
Serrano Lagula, Maria	INFORMATICA	CP Manuel Hidalgo

The Output pane shows the execution of the queries with the following messages:

```

53 18:50:21 SELECT COD_CENTRO, FUNCION, COUNT(*) AS NumeroTrabajadores FROM PERSONAL GROUP BY COD_CENTRO, FUNCION ORDER BY C... 9 rows(s) returned 0.016 sec / 0.000 sec
54 18:51:05 SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO... 9 rows(s) returned 0.000 sec / 0.000 sec
55 18:57:33 SELECT COUNT(DISTINCT FUNCION) AS NumeroFunciones FROM PERSONAL LIMIT 0. 1000 1 row(s) returned 0.000 sec / 0.000 sec
56 18:58:05 SELECT P.* FROM PROFESORES P WHERE P.COD_CENTRO = (SELECT COD_CENTRO FROM PROFESORES WHERE DNI = 1112345) LIMIT... 3 rows(s) returned 0.000 sec / 0.000 sec
57 18:58:26 SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'M%' GROU... 3 rows(s) returned 0.000 sec / 0.000 sec
58 18:58:37 SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO... 9 rows(s) returned 0.000 sec / 0.000 sec

```



5. Muestra la funcion y la suma de los salarios de la tabla PERSONAL para cada FUNCION, pero como condición mostrando solo la funcion cuya suma de los salarios sea superior a 500000 y ordenados descendientemente por el total del salario. Utiliza SUM()

```
SELECT FUNCION, SUM(SALARIO) AS TotalSalario FROM PERSONAL GROUP BY
FUNCION HAVING SUM(SALARIO) > 500000 ORDER BY TotalSalario DESC;SELECT
P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN
CENTROS C ON P.COD_CENTRO = C.COD_CENTRO;
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains a query that inserts data into the 'CENTROS' table and then selects functions and salaries from the 'PERSONAL' table, filtered by a total salary greater than 500,000. The Results window shows the output of the query, which includes the function names and their corresponding total salaries.

**SQL Query:**

```
91 INSERT INTO CENTROS VALUES (45,'P', 'CP Manuel Hidalgo', 'C/Granada 5',
92 '926-282318',228));
93
94 INSERT INTO CENTROS VALUES (58,'S', 'IES Antonete', 'C/Los Toreros 21',
95 '988-486898',318));
96
97 commit;
98 SELECT * FROM PERSONAL WHERE FUNCION = 'ADMINISTRATIVO' AND DNI LIKE '10%';SELECT COUNT(DISTINCT FUNCION) AS NumeroFunciones FROM PERSONAL;
99 SELECT P.* FROM PROFESORES P WHERE P.COD_CENTRO = (SELECT COD_CENTRO FROM PROFESORES WHERE DNI = 1112345);
100 SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE '10%' GROUP BY FUNCION;SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELL
101 SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO;
102 SELECT FUNCION, SUM(SALARIO) AS TotalSalario FROM PERSONAL GROUP BY FUNCION HAVING SUM(SALARIO) > 500000 ORDER BY TotalSalario DESC;SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CEN
```

**Result Grid:**

FUNCION	TotalSalario
PROFESOR	170000
ADMINISTRATIVO	720000

**Action Output:**

Time	Action	Message	Duration / Fetch
57 18:58:26	SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE '10%';	3 row(s) returned	0.000 sec / 0.000 sec
58 18:58:37	SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO;	9 row(s) returned	0.000 sec / 0.000 sec
59 18:58:50	SELECT FUNCION, SUM(SALARIO) AS TotalSalario FROM PERSONAL GROUP BY FUNCION HAVING SUM(SALARIO) > 500000 ORDER BY TotalSalario DESC;	2 row(s) returned	0.000 sec / 0.000 sec
60 18:59:00	SELECT COD_CENTRO, FUNCION, COUNT(*) AS NumeroTrabajadores FROM PERSONAL GROUP BY COD_CENTRO, FUNCION ORDER BY COD_CENTRO;	9 row(s) returned	0.000 sec / 0.000 sec
61 18:59:25	SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO;	9 row(s) returned	0.000 sec / 0.000 sec
62 18:59:31	SELECT FUNCION, SUM(SALARIO) AS TotalSalario FROM PERSONAL GROUP BY FUNCION HAVING SUM(SALARIO) > 500000 ORDER BY TotalSalario DESC;	2 row(s) returned	0.000 sec / 0.000 sec

**6. Muestra los apellidos y la especialidad de los PROFESORES junto con el nombre de su centro (tabla CENTROS). Deben aparecer tambien aquellos centros que no tienen profesor asignado.**

SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD\_CENTRO = C.COD\_CENTRO;

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```

91 --
92 -- INSERT INTO CENTROS VALUES (45,'P', 'CP Manuel Hidalgo', 'C/Granada 5',
93 -- '926-282310',220));
94 -- INSERT INTO CENTROS VALUES (50,'S', 'IES Antoneta', 'C/Los Toreros 21',
95 -- '989-486998',310));
96 --
97 -- commit;
98 -- SELECT * FROM PERSONAL WHERE FUNCION = 'ADMINISTRATIVO' AND DNI LIKE '50%';SELECT COUNT(DISTINCT FUNCION) AS NumeroFunciones FROM PERSONAL;
99 -- SELECT P.* FROM PROFESORES P WHERE P.COD_CENTRO = (SELECT COD_CENTRO FROM PROFESORES WHERE DNI = 1112345);
100 -- SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'ME' GROUP BY FUNCION;SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELL
101 -- SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO;
102 -- SELECT FUNCION, SUM(SALARIO) AS TotalSalario FROM PERSONAL GROUP BY FUNCION HAVING SUM(SALARIO) > 500000 ORDER BY TotalSalario DESC;SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CEN

```

The Results window shows the following data:

FUNCION	TotalSalario
PROFESOR	170000
ADMINISTRATIVO	720000

The Output window shows the execution of the query, with the following messages:

```

54 18:51:05 SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO; 9 rows(s) returned 0.000 sec / 0.000 sec
55 18:57:33 SELECT COUNT(DISTINCT FUNCION) AS NumeroFunciones FROM PERSONAL LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
56 18:58:05 SELECT P.* FROM PROFESORES P WHERE P.COD_CENTRO = (SELECT COD_CENTRO FROM PROFESORES WHERE DNI = 1112345) LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec
57 18:58:26 SELECT FUNCION, MAX(SALARIO) AS MaxSalario, MIN(SALARIO) AS MinSalario FROM PERSONAL WHERE APELLIDOS NOT LIKE 'ME' GROUP BY FUNCION; 3 row(s) returned 0.000 sec / 0.000 sec
58 18:58:37 SELECT P.APELLIDOS, P.ESPECIALIDAD, C.NOMBRE FROM PROFESORES P RIGHT JOIN CENTROS C ON P.COD_CENTRO = C.COD_CENTRO; 9 rows(s) returned 0.000 sec / 0.000 sec
59 18:58:50 SELECT FUNCION, SUM(SALARIO) AS TotalSalario FROM PERSONAL GROUP BY FUNCION HAVING SUM(SALARIO) > 500000 ORDER BY TotalSalario DESC; 2 row(s) returned 0.000 sec / 0.000 sec

```

7. A partir de la tabla PERSONAL, muestra para cada centro y para cada función, el número de trabajadores (que realizan cada FUNCION), ordenado por codigo de centro y por función.

SELECT COD\_CENTRO, FUNCION, COUNT(\*) AS NumeroTrabajadores FROM PERSONAL  
GROUP BY COD\_CENTRO, FUNCION ORDER BY COD\_CENTRO, FUNCION;

The screenshot shows the MySQL Workbench interface with a SQL script named 'EXAMEN' containing several queries. The 'Result Grid' displays the output of the final query, which is a grouped list of centers and functions with their respective number of workers.

COD_CENTRO	FUNCION	NumeroTrabajadores
10	ADMINISTRATIVO	1
10	PROFESOR	3
15	ADMINISTRATIVO	2
15	PROFESOR	3
22	ADMINISTRATIVO	1
22	CONSERJE	1
22	PROFESOR	1
45	CONSERJE	1

The 'Output' pane shows the execution of the queries, including the final query that produces the result grid.

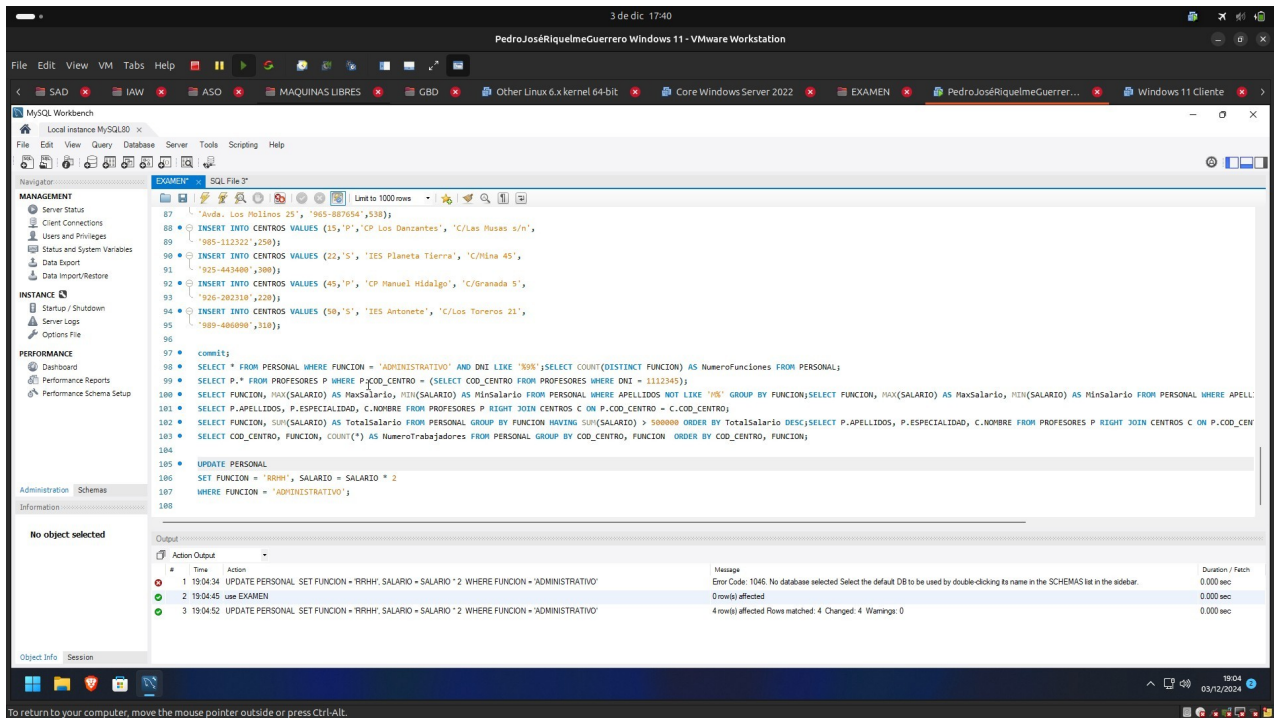
## Ejercicio 5

**1. Modifica la FUNCION de las filas de la tabla PERSONAL, cambiandola por RRHH, y modifica el campo SALARIO al doble del salario, para el personal cuya funcion sea 'ADMINISTRATIVO' (4 Filas)**

UPDATE PERSONAL

SET FUNCION = 'RRHH', SALARIO = SALARIO \* 2

WHERE FUNCION = 'ADMINISTRATIVO';



**2. Añade un nuevo profesor en la tabla PROFESORES, con el código de centro igual al del profesor con especialidad de DIBUJO, especialidad de 'IDIOMA', con DNI 2590055 y de nombre 'Antonio Salas' (1 fila insertada)**

**Aquí no me ha funcionado pero por si te funciona a ti te la paso.**

```
INSERT INTO PROFESORES (COD_CENTRO, DNI, APELLIDOS, ESPECIALIDAD)
VALUES (
    (SELECT COD_CENTRO FROM PROFESORES WHERE ESPECIALIDAD = 'DIBUJO'),
    2590055,
    'Salas, Antonio',
    'IDIOMA'
);
```

**3. Modifica el número de plazas con un valor igual a la mitad en aquellos centros con menos de tres profesores (es decir, en los centros donde la cuenta de profesores para cada centro sea menor que 3).(2 filas modificadas)**

UPDATE CENTROS

SET NUM\_PLAZAS = NUM\_PLAZAS / 2

WHERE COD\_CENTRO IN (

SELECT COD\_CENTRO

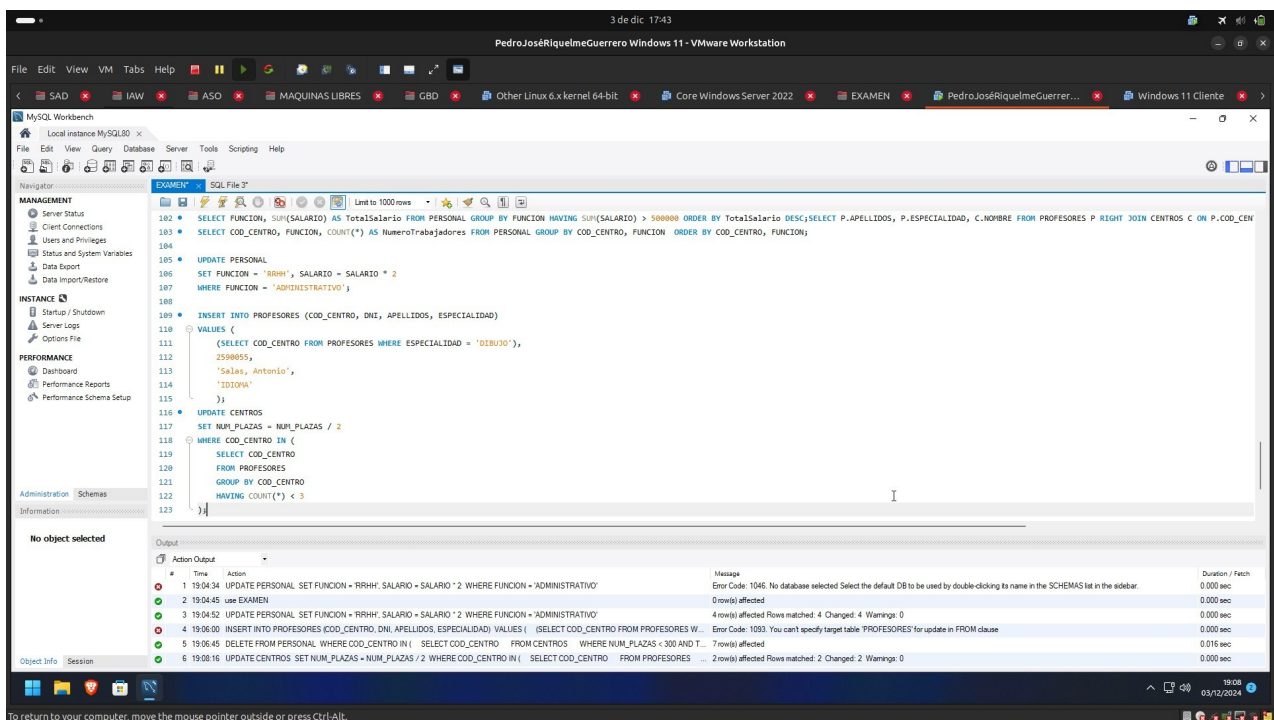
FROM PROFESORES

GROUP BY

COD\_CENTRO HAVING

COUNT(\*) < 3

);



4. Borra al personal que esté en centros de menos de 300 plazas y que cumplan que el tipo de centro sea del tipo 'P'. (7 filas borradas)DELETE FROM PERSONAL

WHERE COD\_CENTRO IN (

SELECT COD\_CENTRO

FROM CENTROS

WHERE NUM\_PLAZAS < 300 AND TIPO\_CENTRO = 'P');

