



BASE DE DATOS

## Unidad 4

### Realización de consultas

Juan Carlos Bohórquez Plato  
I.E.S. TRASSIERRA

## TAREA 4

## Tabla de contenido

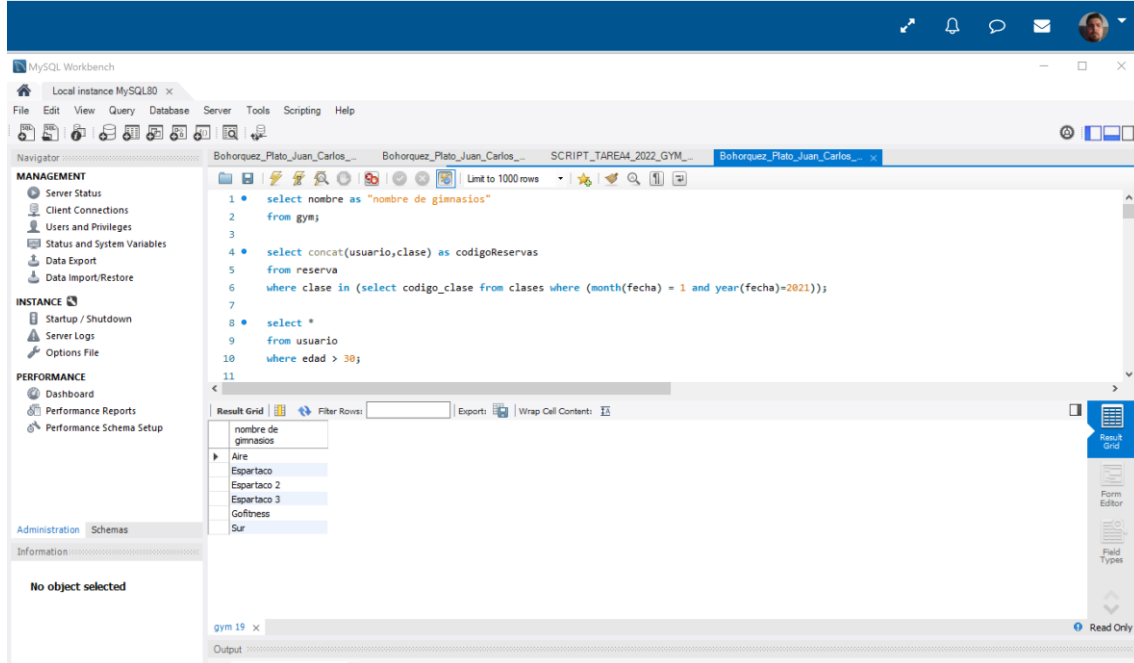
APARTADO A:	2
1.- Obtener un listado de todos los gimnasios.....	2
2.- Obtener el código de reservas que se realizaron el mes de enero de 2021.....	2
3.- Obtener todos los datos de los usuarios abonados mayores de 30 años. ....	2
4.- Obtener el nombre de los monitores y el sueldo que hayan ganado este año (2022). ....	3
APARTADO B:	3
5.- Obtener un listado con el nombre y apellidos de todos los usuarios que hayan participado en la clase de zumba. ....	3
6.- Obtener el número de usuarios que se han apuntado hasta el momento, por cada tipo de clase (a cardio, a box, etc.).....	4
7.- Obtener por cada gimnasio el número total de clases ofertadas.....	4
8.- Obtener por cada gimnasio los monitores que han realizado al menos una clase en ese gimnasio, ordenando el número de clases de mayor a menor.....	5
9.- Obtener un listado con el total de clases que se han impartido en gimnasio de la forma Nombre gimnasio "-" + Sala "-" + TipoClase "-" + Monitor. ....	5
10.- Obtener el nombre, descripción y sueldo del monitor que ha impartido en clases que superen los 4 usuarios.....	6
11.- Obtener un listado con el nombre y DNI de los abonados que no han realizado nunca clase.....	7
12.- Obtener un listado con el nombre de todas las clases en las que no se ha apuntado nadie.....	7
13.- Obtener por abonado el pago que se ha hecho a cada uno de los gimnasios en los últimos 90 días. ....	7
14.- Obtener por cada tipo_clase, los monitores que la han realizado y cuánto han cobrado.	8
15.- Obtener un listado ordenado por importe que ha generado la clase, indicando a qué gimnasio corresponde.....	9
16.- Obtener una lista con el número de usuarios inscritos en cada clase, el código y la fecha de dicha clase, el nombre del tipo de clase, el nombre y apellidos de algún monitor de dicha clase, encontrándose la fecha entre el 1 y el 15 de febrero de 2022, ordenado por número de usuarios inscritos de menor a mayor. ....	9
APARTADO C:	10
17.- Mostrar un listado con el número de alumnos/as por clase, el código de la clase, el nombre de tipo de clase, donde el número de alumnos/as por clase sea mayor que dos. ...	10
18.- Mostrar un listado de los usuarios en el que se desglose lo que han pagado por tipo de clases en cada uno de los gimnasios. ....	10

## TAREA 4

## APARTADO A:

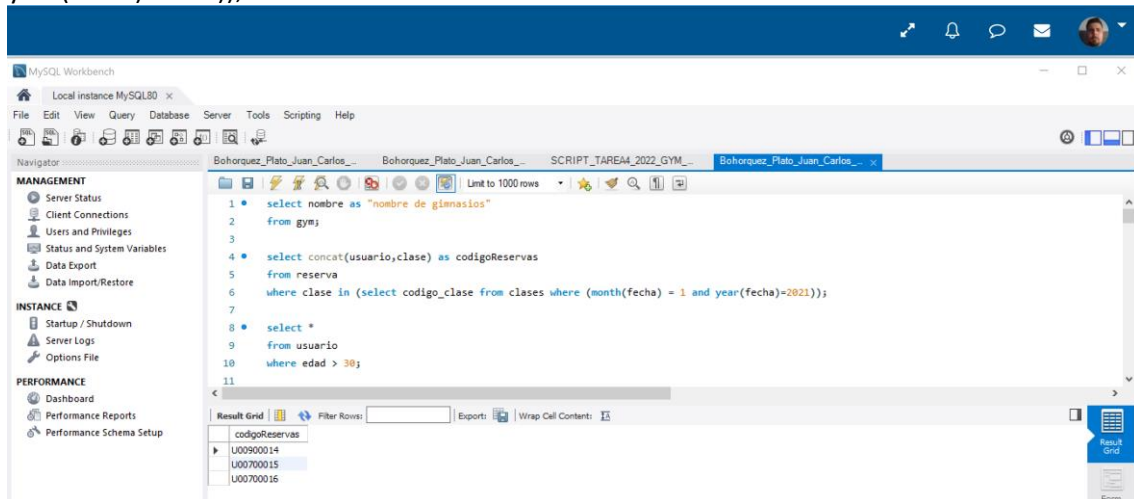
## 1.- Obtener un listado de todos los gimnasios.

```
select nombre as "nombre de gimnasios"  
from gym;
```



## 2.- Obtener el código de reservas que se realizaron el mes de enero de 2021.

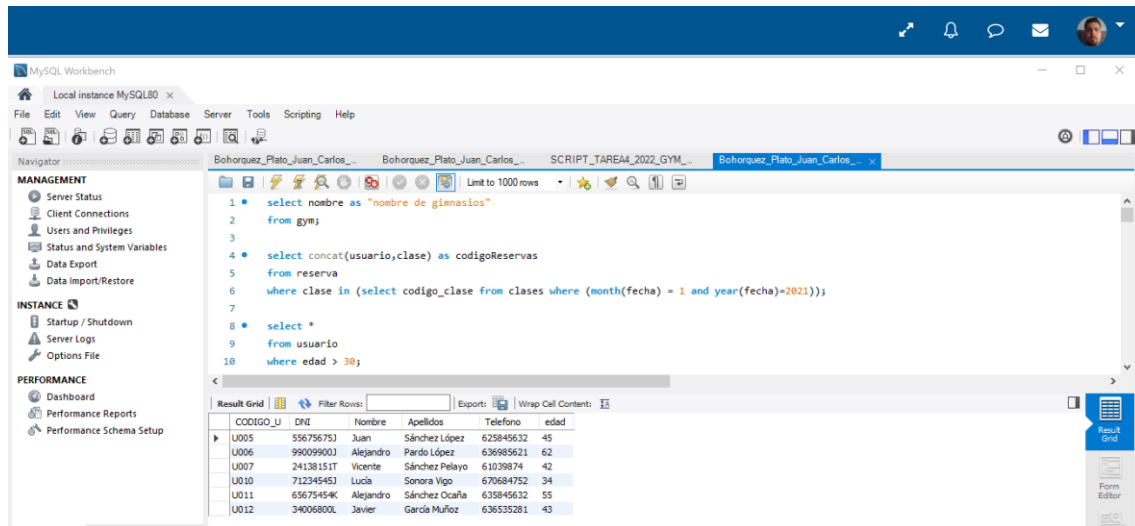
```
select concat(usuario,clase) as codigoReservas  
from reserva  
where clase in (select codigo_clase from clases where (month(fecha) = 1 and  
year(fecha)=2021));
```



## 3.- Obtener todos los datos de los usuarios abonados mayores de 30 años.

```
select *  
from usuario  
where edad > 30;
```

## TAREA 4

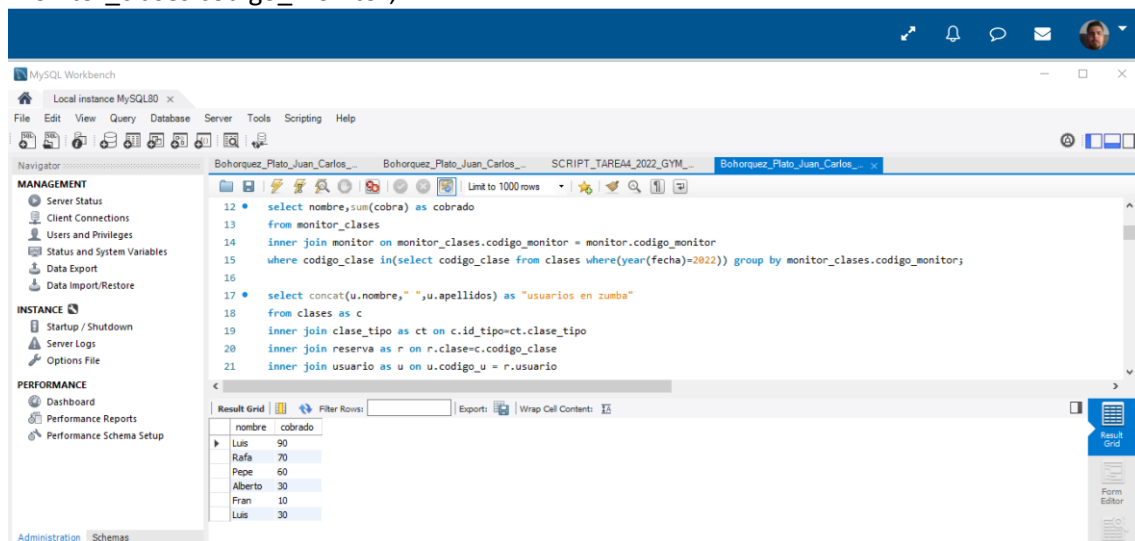


#### 4.- Obtener el nombre de los monitores y el sueldo que hayan ganado este año (2022).

```

select nombre,sum(cobra) as cobrado
from monitor_clases
inner join monitor on monitor_clases.codigo_monitor = monitor.codigo_monitor
where codigo_clase in(select codigo_clase from clases where(year(fecha)=2022)) group by
monitor_clases.codigo_monitor;

```



#### APARTADO B:

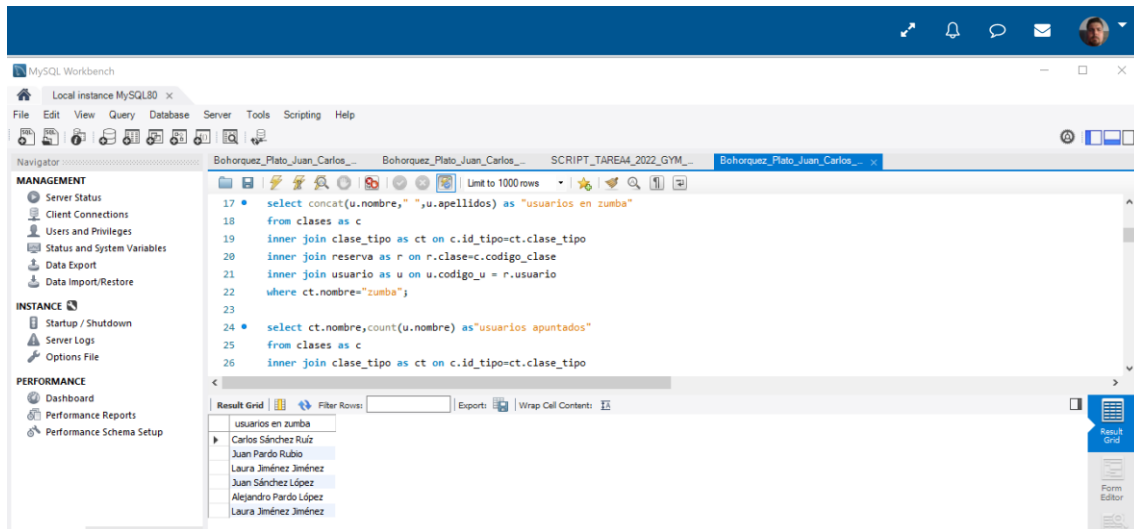
#### 5.- Obtener un listado con el nombre y apellidos de todos los usuarios que hayan participado en la clase de zumba.

```

select concat(u.nombre," ",u.apellidos) as "usuarios en zumba"
from clases as c
inner join clase_tipo as ct on c.id_tipo=ct.clase_tipo
inner join reserva as r on r.clase=c.codigo_clase
inner join usuario as u on u.codigo_u = r.usuario
where ct.nombre="zumba";

```

## TAREA 4

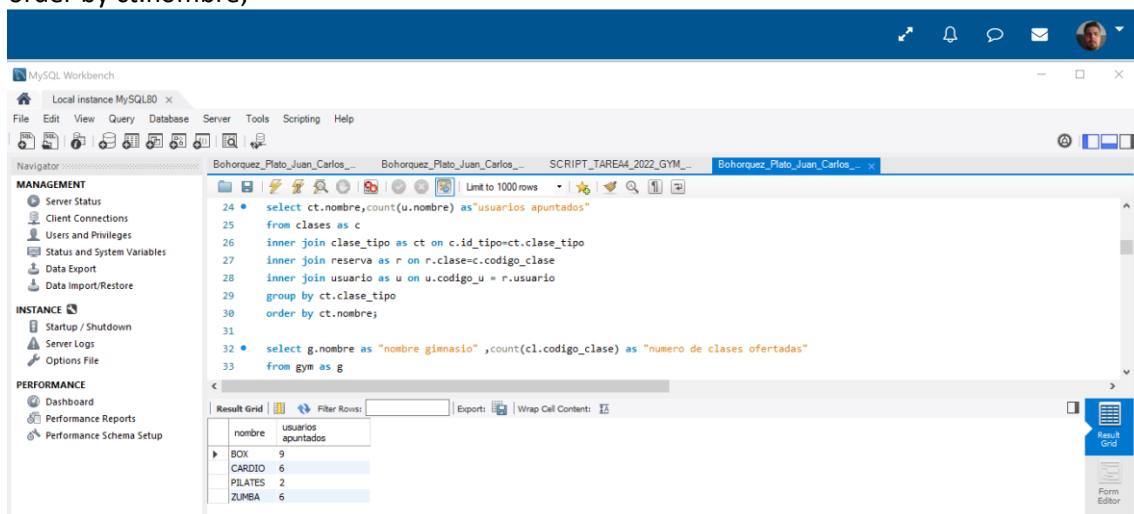


6.- Obtener el número de usuarios que se han apuntado hasta el momento, por cada tipo de clase (a cardio, a box, etc.).

```

select ct.nombre, count(u.nombre) as "usuarios apuntados"
from clases as c
inner join clase_tipo as ct on c.id_tipo=ct.clase_tipo
inner join reserva as r on r.clase=c.codigo_clase
inner join usuario as u on u.codigo_u = r.usuario
group by ct.clase_tipo
order by ct.nombre;

```



7.- Obtener por cada gimnasio el número total de clases ofertadas.

```

select g.nombre as "nombre gimnasio", count(cl.codigo_clase) as "numero de clases
ofertadas"
from gym as g
inner join salas as s on s.id_gym = g.codigo_gym
inner join clases as cl on cl.id_sala = s.codigo_salas
group by g.nombre;

```

## TAREA 4

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

```

32 select g.nombre as "nombre gimnasio", count(cl.codigo_clase) as "numero de clases ofertadas"
33 from gym as g
34 inner join salas as s on s.id_gym = g.codigo_gym
35 inner join clases as cl on cl.id_sala = s.codigo_salas
36 group by g.nombre;
37
38 select g.nombre as "gimnasio", concat(m.nombre, " ", m.apellidos) as "monitor", count(mc.codigo_monitor) as "numero de clases"
39 from gym as g
40 inner join salas as s on s.id_gym = g.codigo_gym
41 inner join clases as c on c.id_sala = s.codigo_salas

```

The result grid shows the following data:

nombre gimnasio	numero de clases ofertadas
Esportaco	4
Goffitness	3
Sur	5
Aire	4
Esportaco 2	2

8.- Obtener por cada gimnasio los monitores que han realizado al menos una clase en ese gimnasio, ordenando el número de clases de mayor a menor.

```

select g.nombre as "gimnasio", concat(m.nombre, " ", m.apellidos) as "monitor",
count(mc.codigo_monitor) as "numero de clases"
from gym as g
inner join salas as s on s.id_gym = g.codigo_gym
inner join clases as c on c.id_sala = s.codigo_salas
inner join monitor_clases mc on mc.codigo_clase = c.codigo_clase
inner join monitor as m on m.codigo_monitor = mc.codigo_monitor
group by concat(m.codigo_monitor, g.nombre)
order by count(mc.codigo_monitor) desc;

```

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

```

38 select g.nombre as "gimnasio", concat(m.nombre, " ", m.apellidos) as "monitor", count(mc.codigo_monitor) as "numero de clases"
39 from gym as g
40 inner join salas as s on s.id_gym = g.codigo_gym
41 inner join clases as c on c.id_sala = s.codigo_salas
42 inner join monitor_clases mc on mc.codigo_clase = c.codigo_clase
43 inner join monitor as m on m.codigo_monitor = mc.codigo_monitor
44 group by concat(m.codigo_monitor, g.nombre)
45 order by count(mc.codigo_monitor) desc;

```

The result grid shows the following data:

gimnasio	monitor	numero de clases
Sur	Rafa Ramirez Pardo	3
Aire	Rafa Ramirez Pardo	3
Esportaco	Pepe Ramirez Pardo	2
Goffitness	Alberto Ramirez Pardo	2
Esportaco	Luis Ramirez Pardo	1
Goffitness	Luis Ramirez Pardo	1
Aire	Luis Ramirez Pardo	1
Esportaco	Rafa Ramirez Pardo	1
Esportaco 2	Rafa Ramirez Pardo	1
Sur	Alberto Ramirez Pardo	1
Sur	Fran Ramirez Pardo	1
Aire	Luis Ramirez Pardo	1
Esportaco 2	Luis Ramirez Pardo	1
Esportaco	Luis Ramirez Pardo	1

9.- Obtener un listado con el total de clases que se han impartido en gimnasio de la forma Nombre gimnasio "-" + Sala "-" + TipoClase "-" + Monitor.

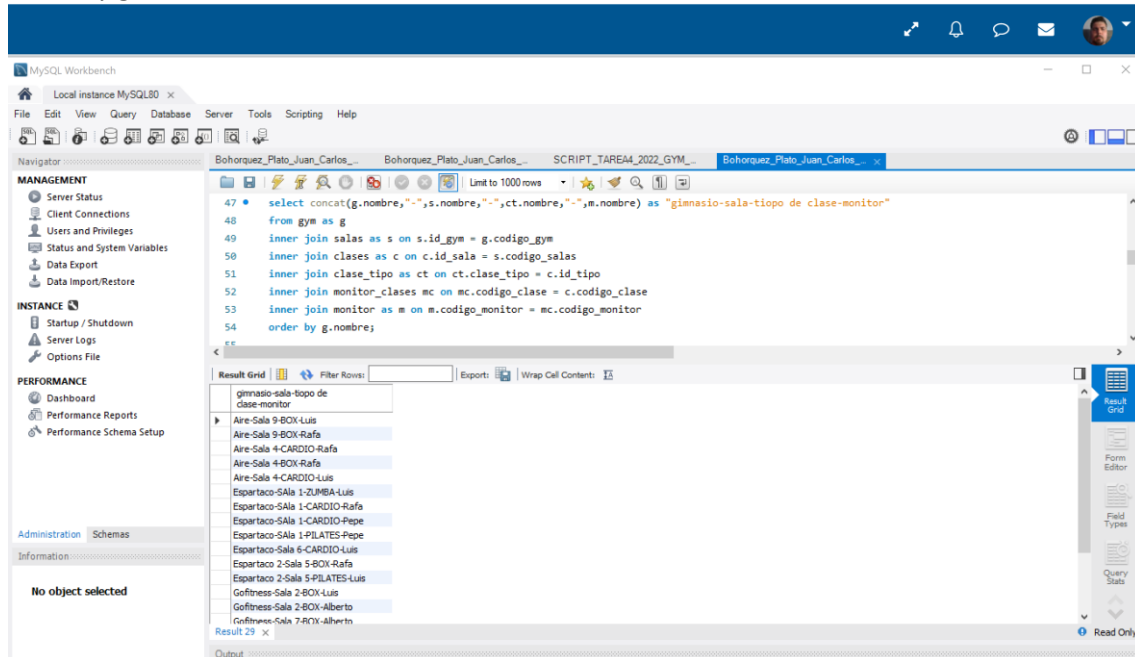
```

select concat(g.nombre, "-", s.nombre, "-", ct.nombre, "-", m.nombre) as "gimnasio-sala-tipo de clase-monitor"
from gym as g
inner join salas as s on s.id_gym = g.codigo_gym

```

## TAREA 4

inner join clases as c on c.id\_sala = s.codigo\_salas  
 inner join clase\_tipo as ct on ct.clase\_tipo = c.id\_tipo  
 inner join monitor\_clases mc on mc.codigo\_clase = c.codigo\_clase  
 inner join monitor as m on m.codigo\_monitor = mc.codigo\_monitor  
 order by g.nombre;

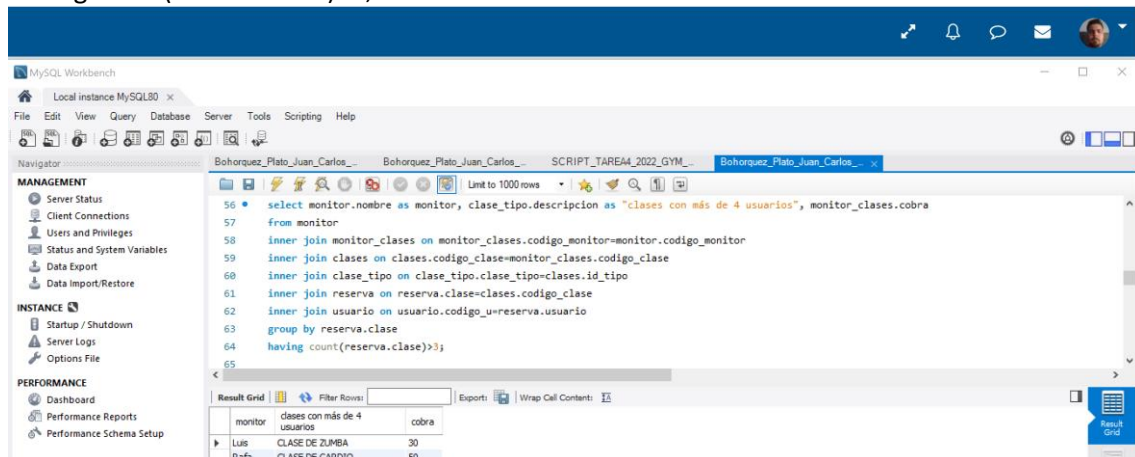


## 10.- Obtener el nombre, descripción y sueldo del monitor que ha impartido en clases que superen los 4 usuarios.

```

select monitor.nombre as monitor, clase_tipo.descripcion as "clases con más de 4 usuarios",
monitor_clases.cobra
from monitor
inner join monitor_clases on monitor_clases.codigo_monitor=monitor.codigo_monitor
inner join clases on clases.codigo_clase=monitor_clases.codigo_clase
inner join clase_tipo on clase_tipo.clase_tipo=clases.id_tipo
inner join reserva on reserva.clase=clases.codigo_clase
inner join usuario on usuario.codigo_u=reserva.usuario
group by reserva.clase
having count(reserva.clase)>3;

```

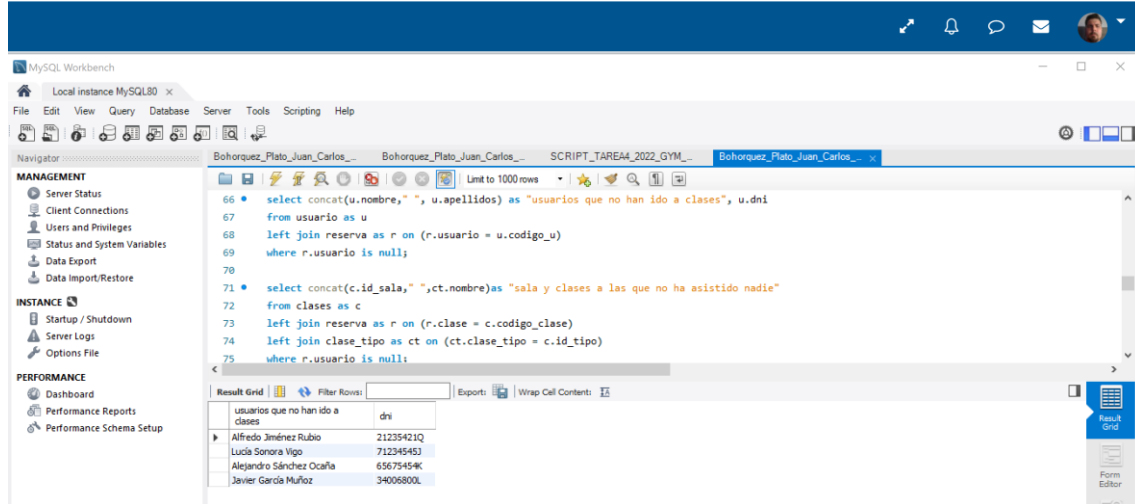




## TAREA 4

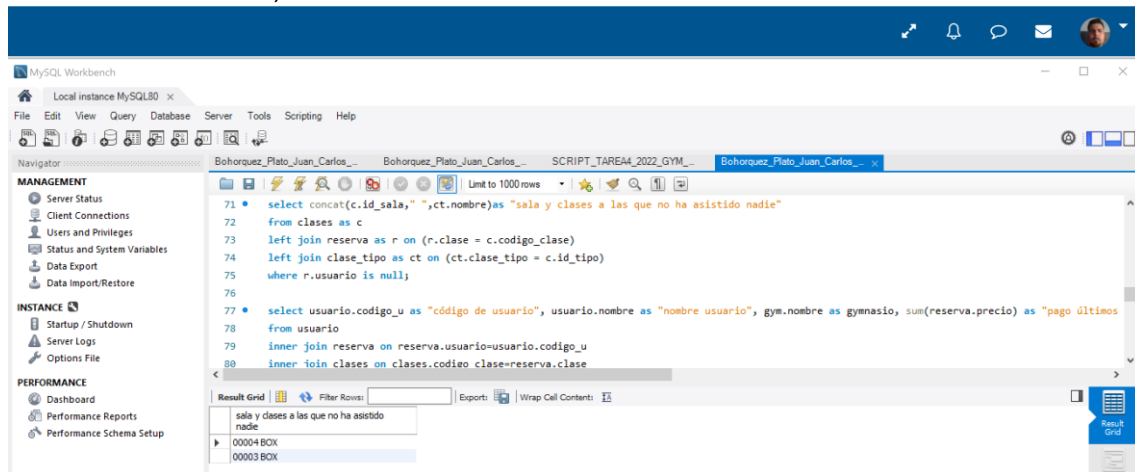
## 11.- Obtener un listado con el nombre y DNI de los abonados que no han realizado nunca clase.

```
select concat(u.nombre," ", u.apellidos) as "usuarios que no han ido a clases", u.dni
from usuario as u
left join reserva as r on (r.usuario = u.codigo_u)
where r.usuario is null;
```



## 12.- Obtener un listado con el nombre de todas las clases en las que no se ha apuntado nadie.

```
select concat(c.id_sala," ",ct.nombre)as "sala y clases a las que no ha asistido nadie"
from clases as c
left join reserva as r on (r.clase = c.codigo_clase)
left join clase_tipo as ct on (ct.clase_tipo = c.id_tipo)
where r.usuario is null;
```



## 13.- Obtener por abonado el pago que se ha hecho a cada uno de los gimnasios en los últimos 90 días.

```
select usuario.codigo_u as "código de usuario", usuario.nombre as "nombre usuario",
gym.nombre as gymnasio, sum(reserva.precio) as "pago últimos 90 días"
from usuario
inner join reserva on reserva.usuario=usuario.codigo_u
inner join clases on clases.codigo_clase=reserva.clase
inner join salas on salas.codigo_salas= clases.id_sala
```



## TAREA 4

inner join gym on gym.codigo\_gym=salas.id\_gym  
 where fecha between date\_add(now(), interval -90 day) and now()  
 group by usuario.dni, gym.nombre;

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

```

77 select usuario.codigo_u as "código de usuario", usuario.nombre as "nombre usuario", gym.nombre as gymnasio, sum(reserva.precio) as "pago últimos
78 from usuario
79 inner join reserva on reserva.usuario=usuario.codigo_u
80 inner join clases on clases.codigo_clase=reserva.clase
81 inner join salas on salas.codigo_salas=clases.id_sala
82 inner join gym on gym.codigo_gym=salas.id_gym
83 where fecha between date_add(now(), interval -90 day) and now()
84 group by usuario.dni, gym.nombre;

```

The Result Grid shows the following data:

código de usuario	nombre usuario	gymnasio	pago últimos 90 días
U001	Carlos	Espartaco	10.00
U002	Juan	Espartaco	20.00
U003	Laura	Espartaco	10.00
U005	Juan	Espartaco	10.00
U006	Alejandro	Espartaco	5.00
U002	Juan	Gofitness	10.00
U003	Laura	Gofitness	10.00
U005	Juan	Gofitness	15.00
U004	Carmen	Espartaco	15.00
U003	Laura	Sur	10.00
U009	Maria	Espartaco	15.00
U007	Vicente	Gofitness	30.00
U007	Vicente	Sur	30.00
U007	Vicente	Aire	30.00
U007	Vicente	Espartaco 2	15.00

14.- Obtener por cada tipo\_clase, los monitores que la han realizado y cuánto han cobrado.

select clase\_tipo.nombre as "nombre de clase", monitor.nombre as monitor,  
 monitor\_clases.cobra  
 from clase\_tipo  
 inner join clases on clases.id\_tipo=clase\_tipo.clase\_tipo  
 inner join monitor\_clases on monitor\_clases.codigo\_clase=clases.codigo\_clase  
 inner join monitor on monitor.codigo\_monitor=monitor\_clases.codigo\_monitor  
 order by clase\_tipo.nombre;

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

```

86 select clase_tipo.nombre as "nombre de clase", monitor.nombre as monitor, monitor_clases.cobra
87 from clase_tipo
88 inner join clases on clases.id_tipo=clase_tipo.clase_tipo
89 inner join monitor_clases on monitor_clases.codigo_clase=clases.codigo_clase
90 inner join monitor on monitor.codigo_monitor=monitor_clases.codigo_monitor
91 order by clase_tipo.nombre;

```

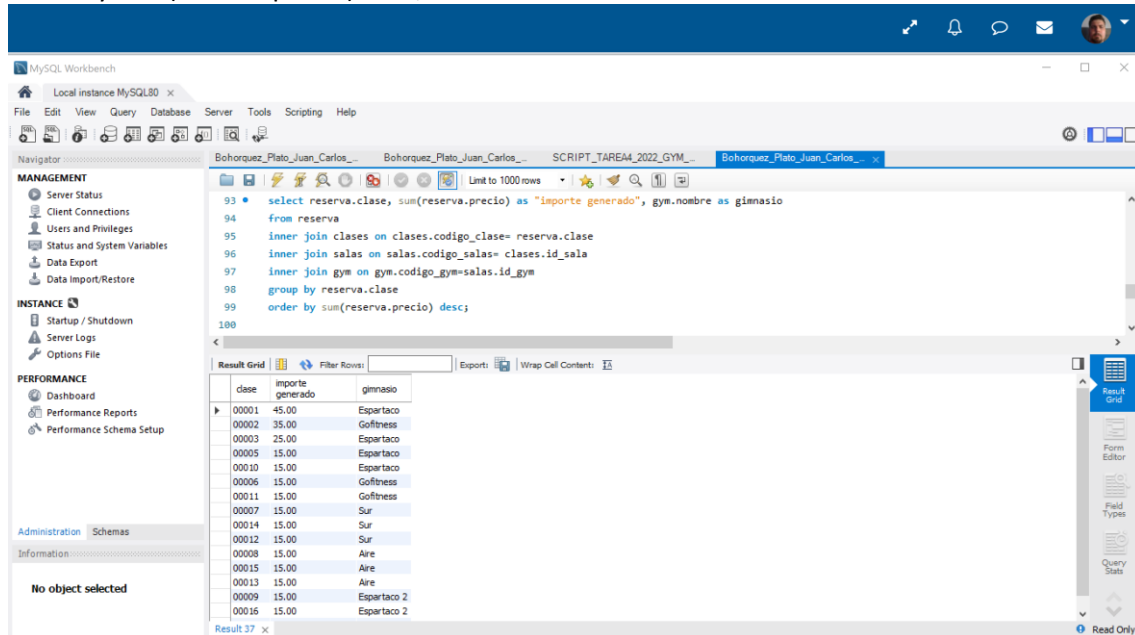
The Result Grid shows the following data:

nombre de clase	monitor	cobra
BOX	Luis	50
BOX	Luis	10
BOX	Rafa	10
BOX	Rafa	10
BOX	Rafa	10
BOX	Rafa	10
BOX	Alberto	10
BOX	Alberto	10
BOX	Fran	10
CARDIO	Rafa	50
CARDIO	Rafa	10
CARDIO	Pepe	50
CARDIO	Alberto	10
CARDIO	Luis	10
CARDIO	Luis	10
PILATES	Pepe	10
PILATES	Pepe	10

## TAREA 4

15.- Obtener un listado ordenado por importe que ha generado la clase, indicando a qué gimnasio corresponde.

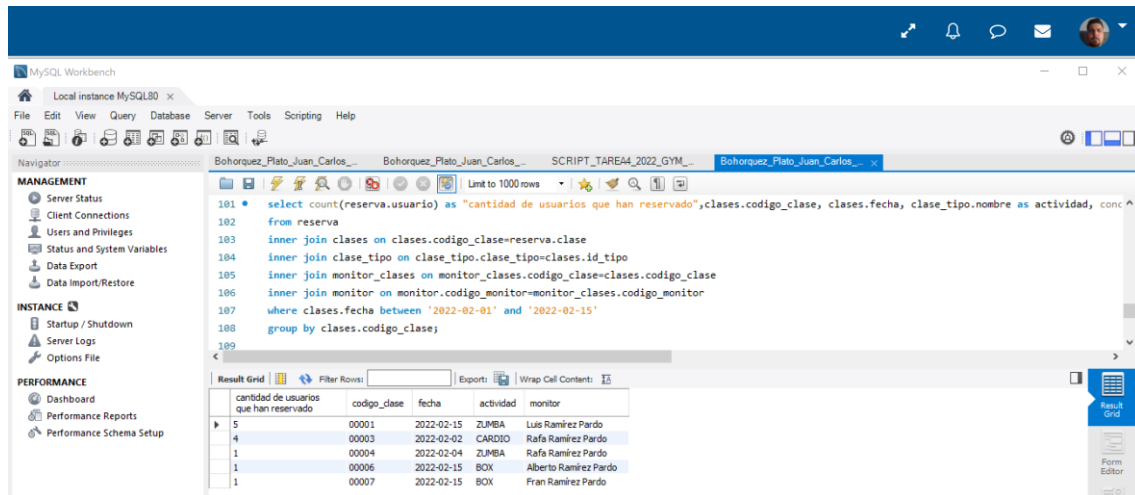
```
select reserva.clase, sum(reserva.precio) as "importe generado", gym.nombre as gimnasio
from reserva
inner join clases on clases.codigo_clase= reserva.clase
inner join salas on salas.codigo_salas= clases.id_sala
inner join gym on gym.codigo_gym=salas.id_gym
group by reserva.clase
order by sum(reserva.precio) desc;
```



16.- Obtener una lista con el número de usuarios inscritos en cada clase, el código y la fecha de dicha clase, el nombre del tipo de clase, el nombre y apellidos de algún monitor de dicha clase, encontrándose la fecha entre el 1 y el 15 de febrero de 2022, ordenado por número de usuarios inscritos de menor a mayor.

```
select count(reserva.usuario) as "cantidad de usuarios que han reservado",clases.codigo_clase,
clases.fecha, clase_tipo.nombre as actividad, concat(monitor.nombre," ", monitor.apellidos) as
monitor
from reserva
inner join clases on clases.codigo_clase=reserva.clase
inner join clase_tipo on clase_tipo.clase_tipo=clases.id_tipo
inner join monitor_clases on monitor_clases.codigo_clase=clases.codigo_clase
inner join monitor on monitor.codigo_monitor=monitor_clases.codigo_monitor
where clases.fecha between '2022-02-01' and '2022-02-15'
group by clases.codigo_clase;
```

## TAREA 4



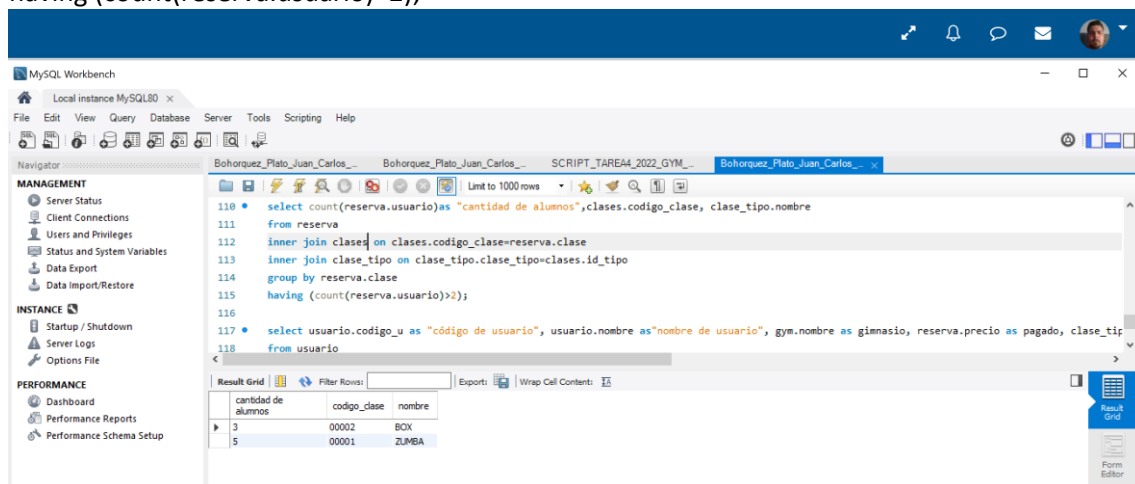
## APARTADO C:

17.- Mostrar un listado con el número de alumnos/as por clase, el código de la clase, el nombre de tipo de clase, donde el número de alumnos/as por clase sea mayor que dos.

```

select count(reserva.usuario)as "cantidad de alumnos",clases.codigo_clase, clase_tipo.nombre
from reserva
inner join clases on clases.codigo_clase=reserva.clase
inner join clase_tipo on clase_tipo.clase_tipo=clases.id_tipo
group by reserva.clase
having (count(reserva.usuario)>2);

```



18.- Mostrar un listado de los usuarios en el que se desglose lo que han pagado por tipo de clases en cada uno de los gimnasios.

```

select usuario.codigo_u as "código de usuario", usuario.nombre as "nombre de usuario",
gym.nombre as gimnasio, reserva.precio as pagado, clase_tipo.nombre as actividad
from usuario
inner join reserva on reserva.usuario=usuario.codigo_u
inner join clases on clases.codigo_clase=reserva.clase
inner join clase_tipo on clase_tipo.clase_tipo=clases.id_tipo
inner join salas on salas.codigo_salas=clases.id_sala
inner join gym on gym.codigo_gym=salas.id_gym

```

## TAREA 4

group by concat(reserva.usuario, reserva.clase);

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

```
117 select usuario.codigo_u as "código de usuario", usuario.nombre as "nombre de usuario", gym.nombre as gimnasio, reserva.precio as pagado, clase_tip
118 from usuario
119 inner join reserva on reserva.usuario=usuario.codigo_u
120 inner join clases on clases.codigo_clase=reserva.clase
121 inner join clase_tipo on clase_tipo.clase_tipo=clases.id_tipo
122 inner join salas on salas.codigo_salas=clases.id_sala
123 inner join gym on gym.codigo_gym=salas.id_gym
124 group by concat(reserva.usuario, reserva.clase);
125
```

The Results window displays the following data:

código de usuario	nombre de usuario	gimnasio	pagado	actividad
U001	Carlos	Espartaco	10.00	ZUMBA
U002	Juan	Espartaco	10.00	ZUMBA
U002	Juan	Goffitness	10.00	BOX
U002	Juan	Espartaco	10.00	CARDIO
U003	Laura	Espartaco	10.00	ZUMBA
U003	Laura	Goffitness	10.00	BOX
U003	Laura	Sur	10.00	ZUMBA
U004	Carmen	Espartaco	15.00	CARDIO
U005	Juan	Espartaco	10.00	ZUMBA
U005	Juan	Goffitness	15.00	BOX
U006	Allegandro	Espartaco	5.00	ZUMBA
U007	Vicente	Goffitness	15.00	BOX
U007	Vicente	Sur	15.00	BOX
U007	Vicente	Aire	15.00	CARDIO