

# **Xestión de Bases de datos**

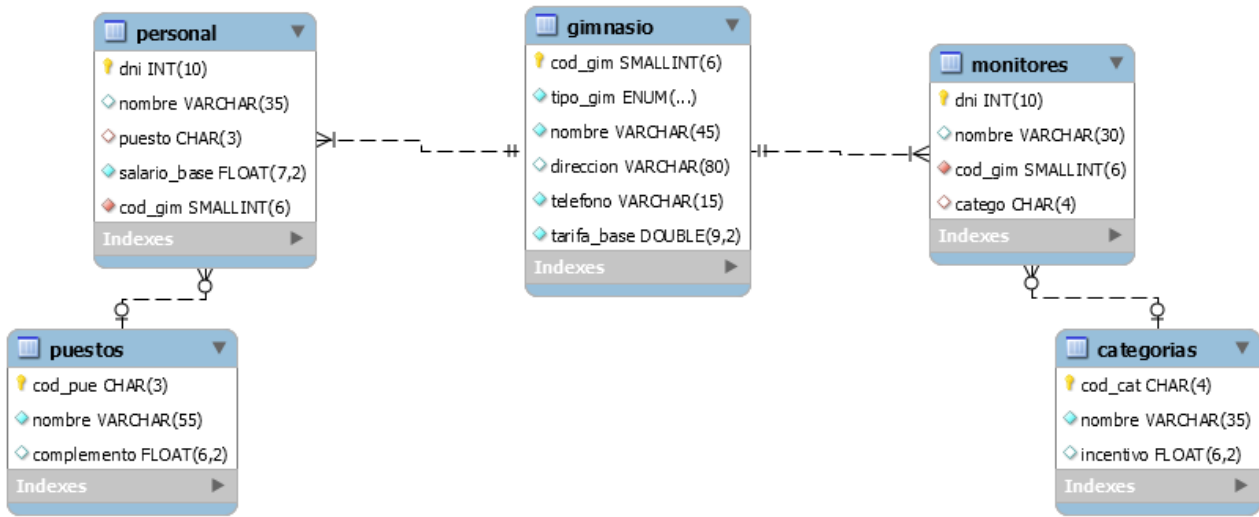
## **Tarefa 5**

## Sumario

ESCENARIO.....	3
APARTADO 1.....	4
1.1).....	4
1.2).....	6
1.3).....	9
1.4).....	10
APARTADO 2.....	12
2.1).....	12
2.2).....	13
2.3).....	14
2.4).....	15
2.5).....	16
2.6).....	17
2.7).....	18
2.8).....	19
2.9).....	20
a).....	20
b).....	20
APARTADO 3.....	21
3.1).....	22
3.2).....	23
3.3).....	24
APARTADO 4.....	25
4.1).....	25
4.2).....	26
4.3).....	27
4.4).....	27
4.5).....	28
4.6).....	28

## ESCENARIO

Tenemos una base de datos, simplificada, que guarda información sobre una cadena de gimnasios. En ella se almacenan datos sobre diferentes gimnasios, los monitores y otro personal laboral. El modelo o esquema relacional de la base de datos es el de la siguiente imagen.



En un gimnasio trabajan los monitores de actividades y otro personal que no es monitor.

Cada monitor trabaja en un único gimnasio.

Cada empleado que no es monitor trabaja en un único gimnasio.

Cada monitor tiene una categoría. Puede haber monitores sin categoría.

Cada empleado tiene un puesto. Puede haber empleados sin puesto.

Carga la base de datos en tu servidor MySQL y realiza los siguientes ejercicios sobre ella.

# APARTADO 1.

Se trata de realizar las siguientes operaciones SIN utilizar sentencias SQL, debe ser de forma gráfica o visual, Para ello utiliza la herramienta gráfica Workbench.

IMPORTANTE: Realiza una a una las operaciones indicadas, capturando las correspondientes pantallas en cada caso, esto es, haz cada inserción, modificación y eliminación de datos de forma individual, no varias a la vez.

## 1.1)

Inserta tres nuevos gimnasios con los datos indicados a continuación. Si no se indican datos para alguna columna, es por desconocer su valor en ese momento. Si alguna de las inserciones te da error, captura pantalla con el error y razona/explica a que es debido el error.

### Datos a insertar

Datos gimnasio 1	Datos gimnasio 2	Datos gimnasio 3
cod_gim: 600 tipo_gim: SUP nombre: SuperGym direccion: Calle Serezade 10 telefono: 666554411 tarifa_base: 30	cod_gim: 700 tipo_gim: MED nombre: EgoGym direccion: Calle Arenal telefono: 666442288 tarifa_base: 25	cod_gim: 800 tipo_gim: INF nombre: VivaGym direccion: telefono: 666998833 tarifa_base:

Datos gimnasio 1:

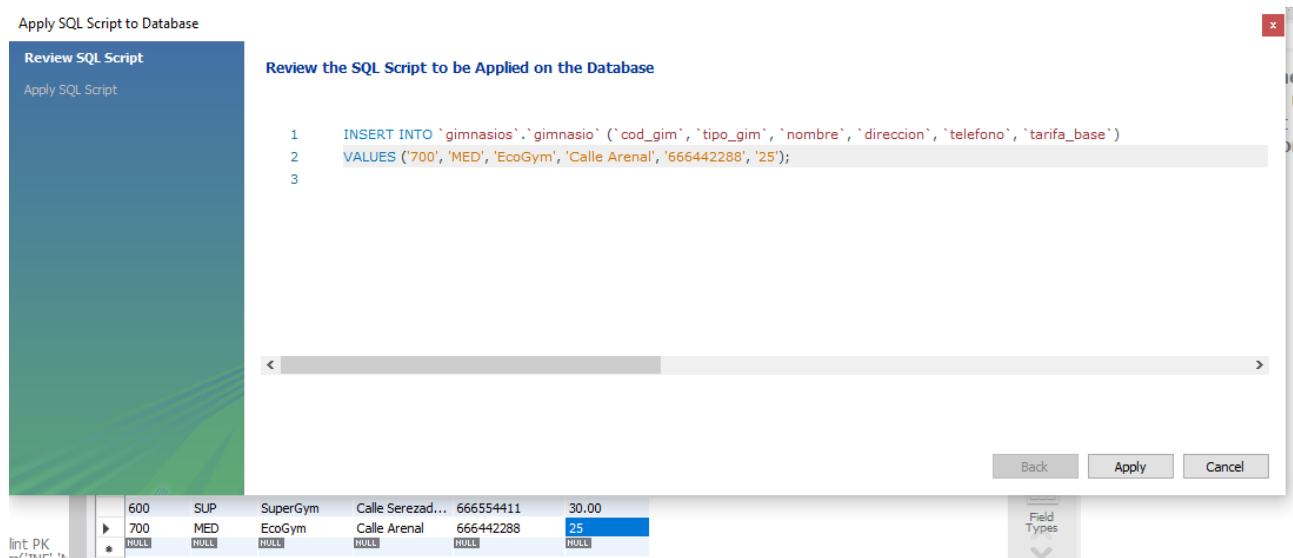
The screenshot displays the Oracle SQL Developer interface. A 'Review SQL Script' dialog box is open, showing the following SQL script:

```
1 INSERT INTO 'gimnasios'. 'gimnasio' ('cod_gim', 'tipo_gim', 'nombre', 'direccion', 'telefono', 'tarifa_base')
2 VALUES ('600', 'SUP', 'SuperGym', 'Calle Serezade, 10', '666554411', '30');
```

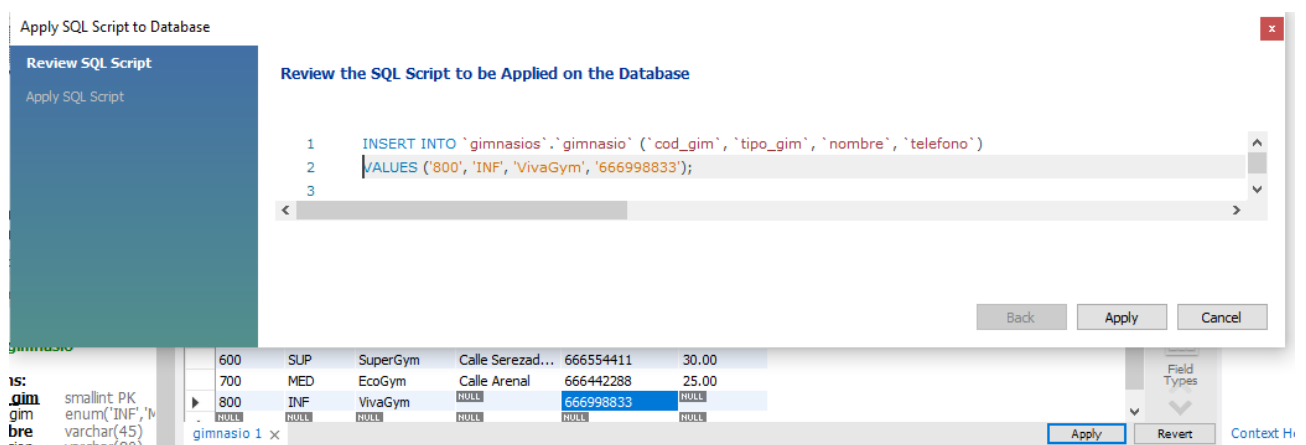
The background shows the 'gimnasios' table structure and data:

cod_gim	tipo_gim	nombre	direccion	telefono	tarifa_base
40	MED	Kankunete	Perdidos, 18	0044661835910	70.00
50	SUP	EgoGym	Maravillas, 6	655845713	81.00
60	SUP	SportGym	Plazuelas, 2	625342743	95.00
77	SUP	TopGym	Gran via, 122	627121212	90.00
600	SUP	SuperGym	Calle Serezad...	666554411	30

## Datos gimnasio 2:



## Datos gimnasio 3:



Los tres se han podido insertar. En el gimnasio 3, se puso el valor por default 11.00

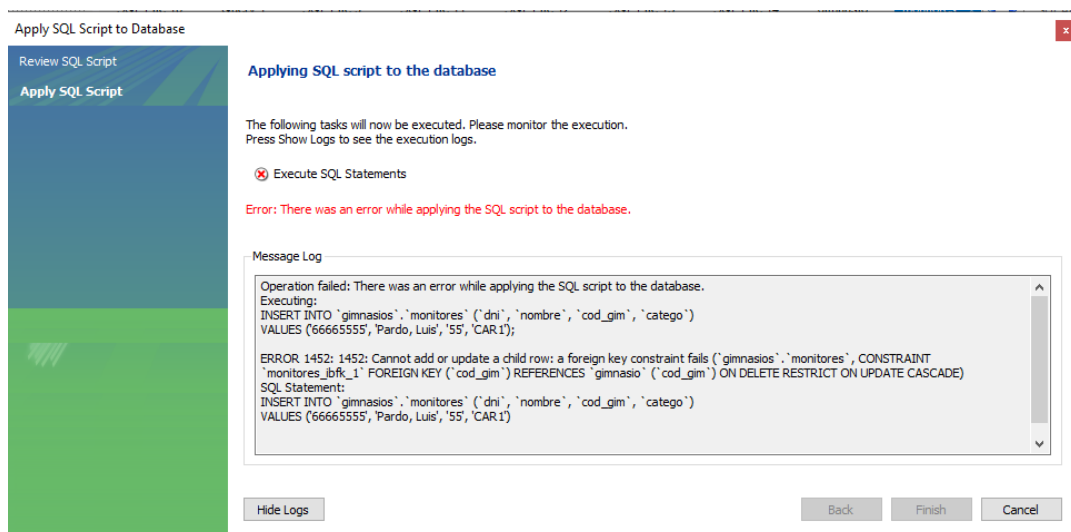
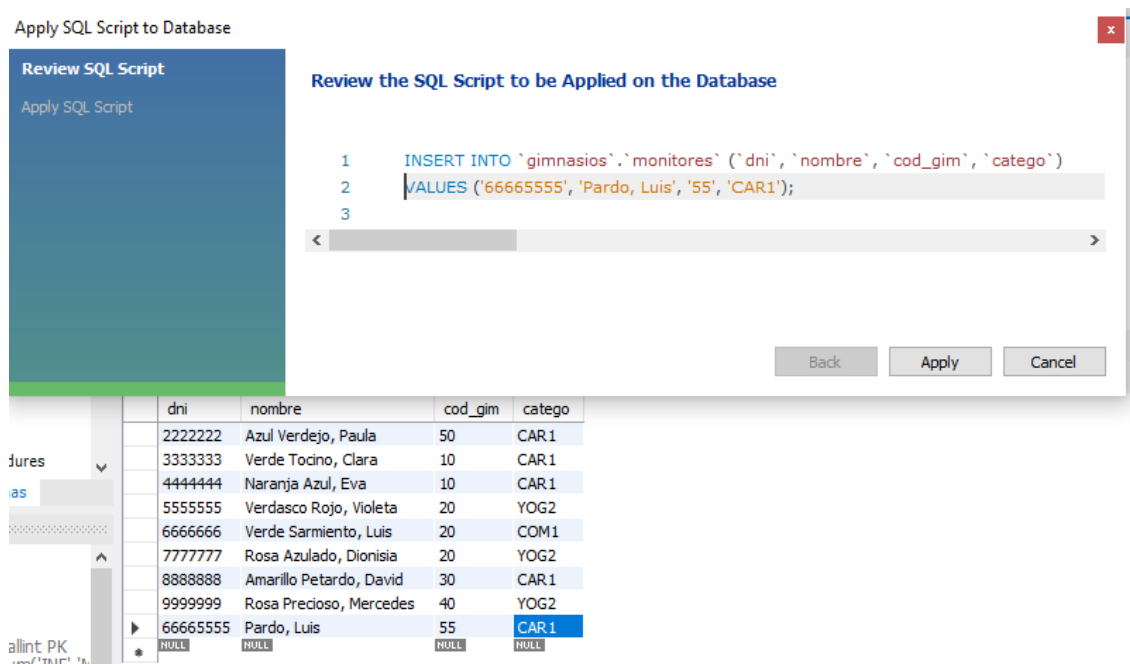
## 1.2)

Inserta los siguientes monitores. Si no se indican datos para alguna columna, es por desconocer su valor en ese momento. Si se produce algún error en alguna inserción, captura la pantalla, explica y razona el motivo del error.

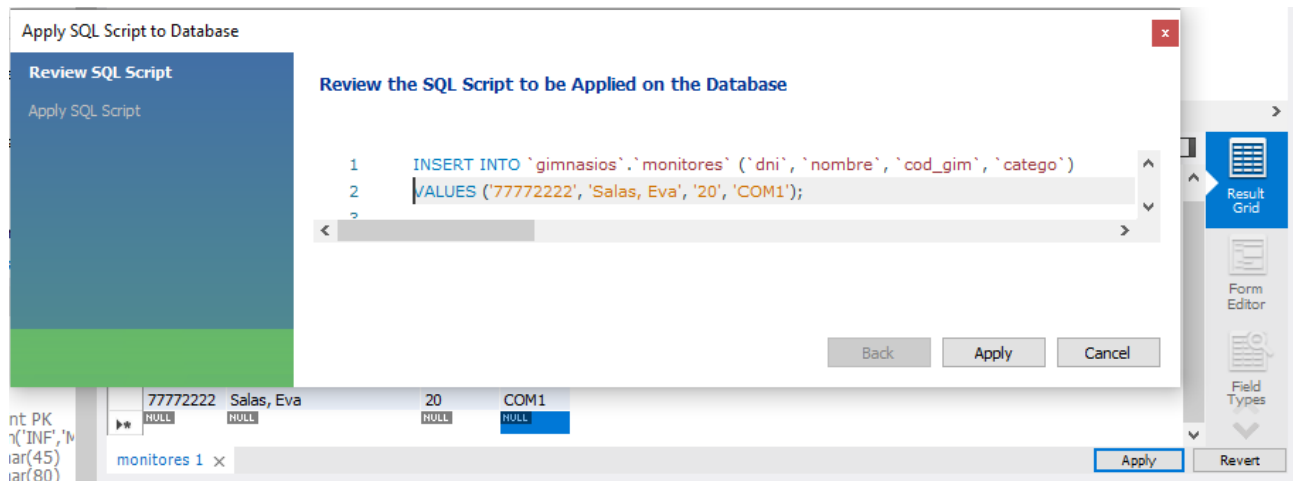
### Datos a insertar

monitor 1	monitor 2	monitor 3	monitor 4
dni: 66665555 nombre: Pardo, Luis cod_gim: 55 catego: 'CAR1'	dni: 77772222 nombre: Salas, Eva cod_gim: 20 catego: 'COM1'	dni: 33338888 nombre: Cruz, Marta cod_gim: 77 catego: 'YOG1'	dni: 44441111 nombre: Gil, Jorge cod_gim: 77 catego: 'YOG2'

### Monitor 1:

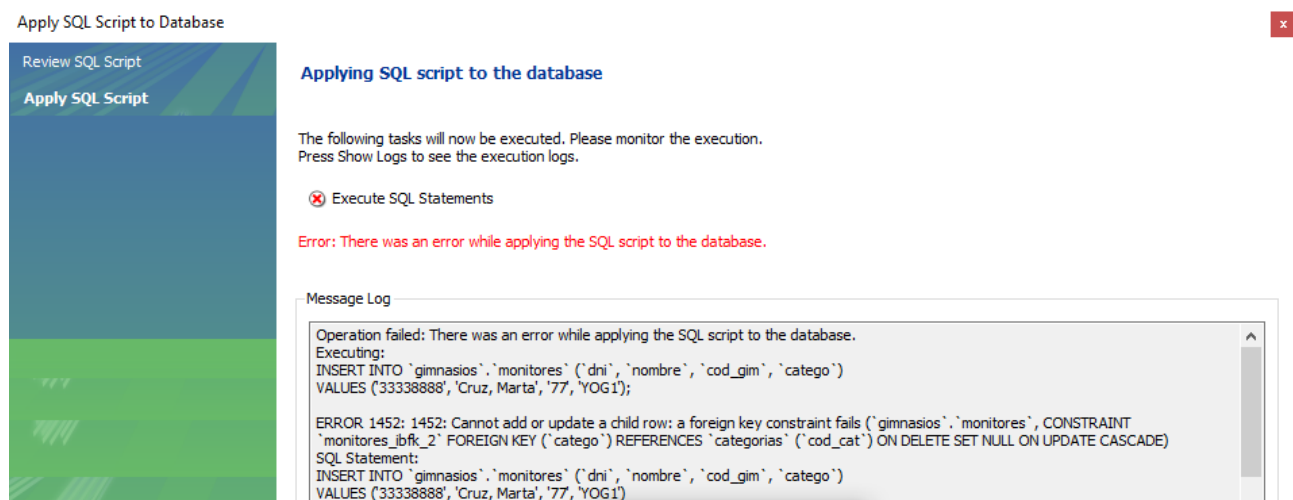
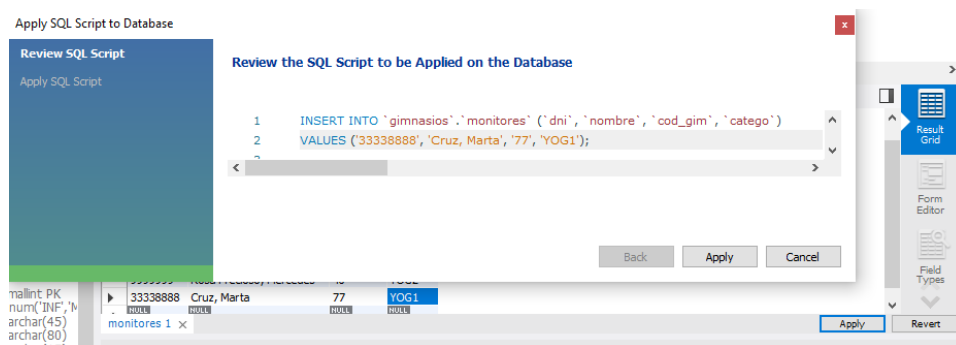


## Monitor 2:

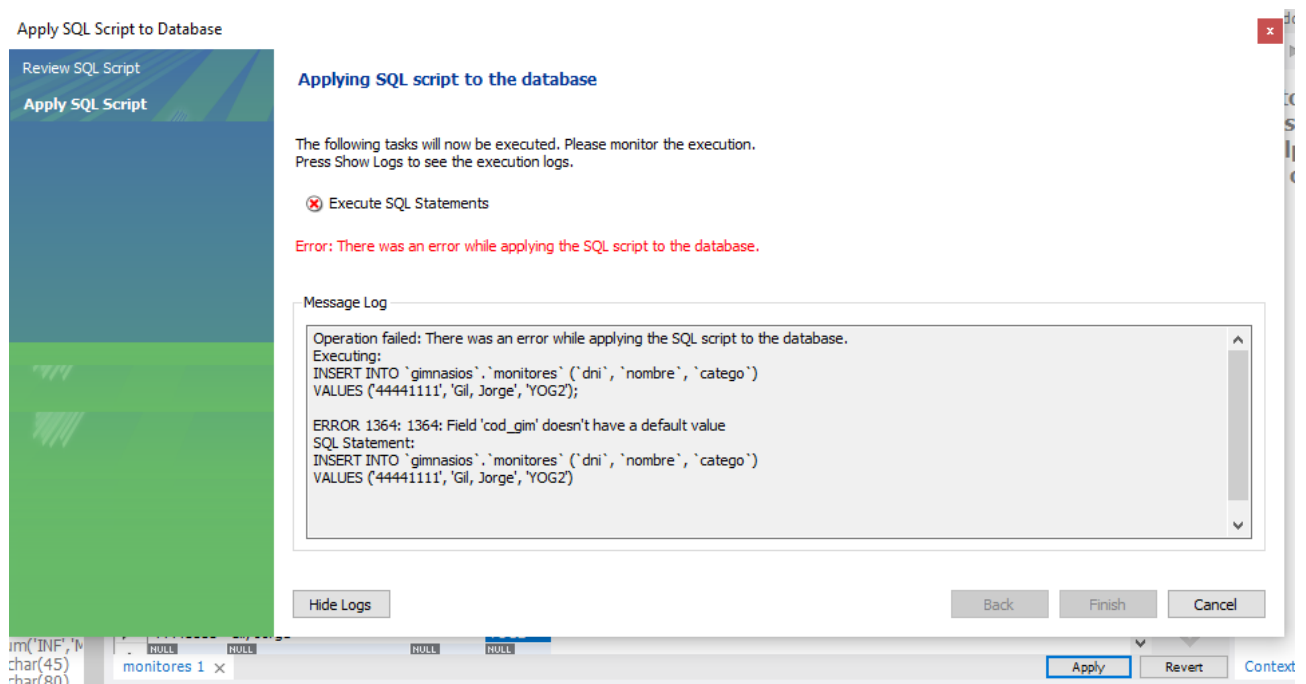
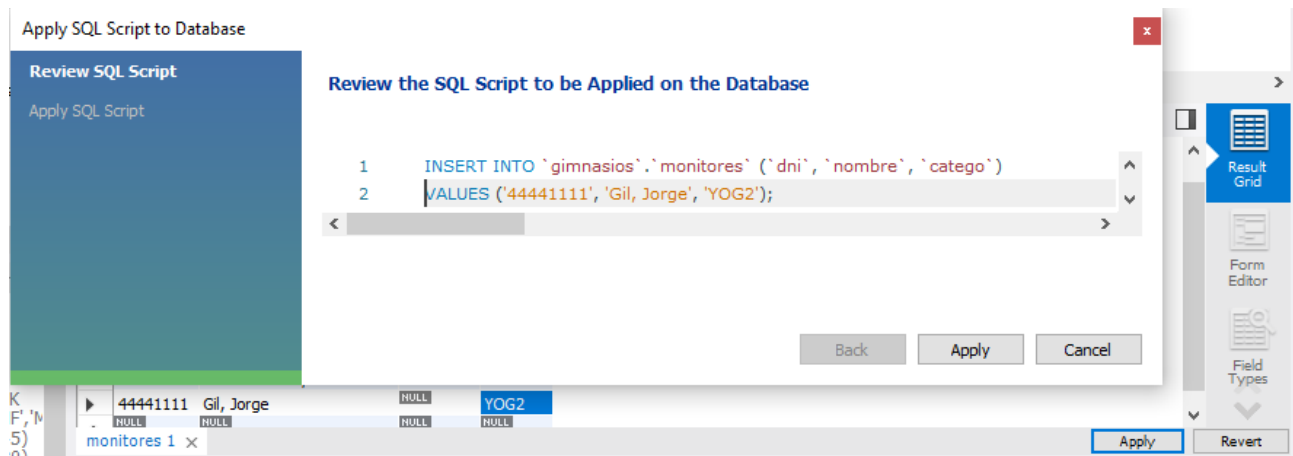


Ok

## Monitor 3:



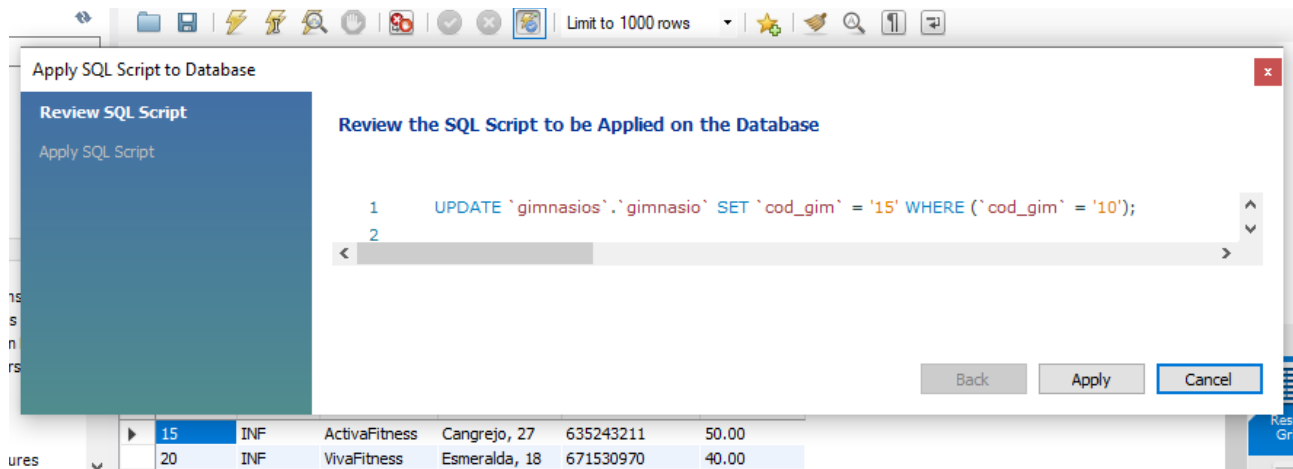
## Monitor 4:





### 1.3)

Modifica el código del gimnasio 10, por el código 15. ¿Se ha podido realizar esa modificación?



Sí, se puede realizar la modificación.

Indica si ha tenido alguna repercusión en el personal y monitores de ese gimnasio, y a que puede ser debido.

Sí, se actualiza la información debido al on update cascade:

```
-- TABLA PERSONAL
CREATE TABLE personal (
  dni          CHAR(10) NOT NULL,
  nombre       VARCHAR(35),
  puesto       CHAR(3),
  salario_base FLOAT(7,2) UNSIGNED NOT NULL,
  cod_gim      SMALLINT NOT NULL,
  PRIMARY KEY (dni),
  FOREIGN KEY (cod_gim)
  REFERENCES gimnasio (cod_gim)
  ON DELETE RESTRICT
  ON UPDATE CASCADE,
  FOREIGN KEY (puesto)
  REFERENCES puestos (cod_pue)
  ON DELETE SET NULL
  ON UPDATE CASCADE
);
```

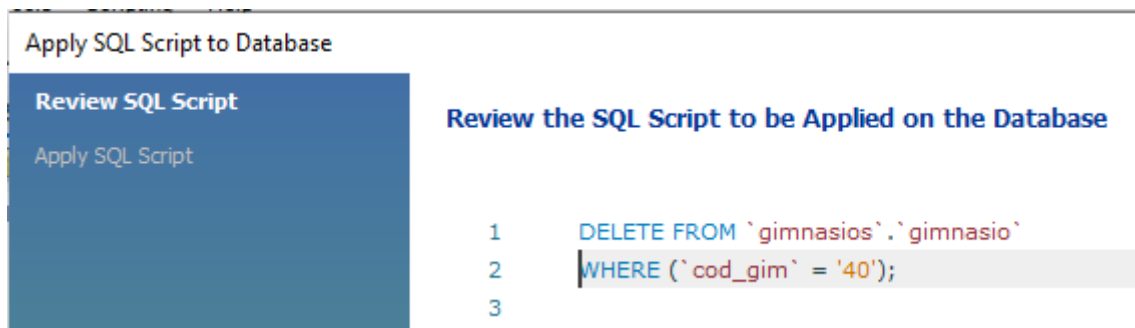
Captura pantallas en la que se pueda apreciar lo ocurrido.

Personal que antes trabajaba en el cod\_gim 10 ahora trabajará en el 15:

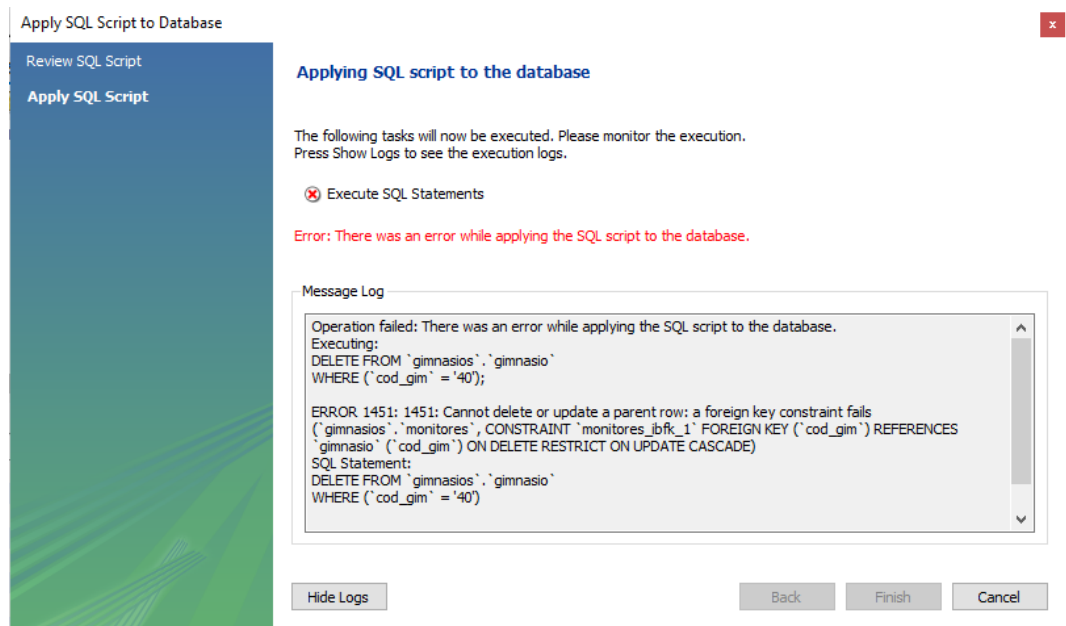
	dni	nombre	puesto	salario_base	cod_gim
▶	1010100	Gómez Reina, Marcela	01	100.00	20
	1212120	Peinado Guindilla, Eva	01	100.00	15
	1313130	Maza Carpio, Susana	01	100.00	15

## 1.4)

Elimina el gimnasio de código 40 y el de código 77. ¿En algún caso da error? Explica y razona en cada caso.



Código 40 no se puede eliminar:



Código 77:

**Review the SQL Script to be Applied on the Database**

```
1      DELETE FROM `gimnasios`.`gimnasio` WHERE (`cod_gim` = '77');  
2
```

Sí, se puede eliminar, ya que no hay trabajadores en el gimnasio de código 77.

## APARTADO 2.

Escribe las sentencias SQL que permiten realizar las siguientes operaciones.

Para cada sentencia debes hacer lo siguiente:

Comprueba lo ocurrido una vez ejecutada. (capturando pantalla)

En caso de error, además de la captura razona el motivo de ese error.

IMPORTANTE: en el caso de las inserciones de datos, hay que redactar la sentencia SQL que inserta únicamente los datos indicados, no hay que poner un null en aquellos datos no proporcionados.

### 2.1)

Inserta los siguientes monitores, cada uno con un INSERT.

1. Inserta en la tabla MONITORES a la monitora de nombre: 'Frias Reche, María', gimnasio 20 y de DNI : '10101010'.

```
3 • INSERT INTO `gimnasios`.`monitores`  
4 VALUES ('10101010','Frias Reche, María',20)
```

2. Inserta en la tabla MONITORES al monitor de nombre 'Verde Botella, Esperanza', gimnasio 40, DNI '20202020' de la categoría 'CAR2'.

```
6 • INSERT INTO `gimnasios`.`monitores`  
7 VALUES ('20202020','Verde Botella, Esperanza',40,'CAR2')
```

Fallo de integridad referencial porque no hay la categoría CAR2

```
21:51:37 INSERT INTO `gimnasios`.`monitores` VALUES ('20202020','Verde Botella,  
Esperanza',40,'CAR2') Error Code: 1452. Cannot add or update a child row: a foreign key  
constraint fails (`gimnasios`.`monitores`, CONSTRAINT `monitores_ibfk_2` FOREIGN KEY  
(`catego`) REFERENCES `categorias` (`cod_cat`) ON DELETE SET NULL ON UPDATE  
CASCADE) 0.031 sec
```

## 2.2)

Inserta un monitor en la tabla MONITORES de nombre 'Amarillo Rojo, Luis', DNI: '80808080', gimnasio 30 y de categoría 'YOGA'.

En la inserción, debes basarte en que solo conoces el nombre de la categoría, NO su código.

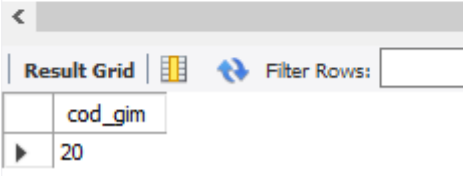
```
1 • INSERT INTO monitores (dni, nombre, cod_gim, catego)
2 VALUES ('80808080', 'Amarillo Rojo, Luis', 30, (SELECT cod_cat FROM categorias WHERE nombre = 'YOGA'));
3 |
```

## 2.3)

Inserta en la tabla MONITORES a uno nuevo de nombre 'Sanz Bueno, Mariano', con DNI 67676767, de la categoría 'COM1' y del gimnasio, uno de los gimnasios que tengan 1 o 2 personas del puesto '03'. Ten en cuenta que puede haber más de un gimnasio en esa situación.

Primero buscamos un gimnasio que cumpla con el criterio:

```
1 • SELECT cod_gim
2 FROM personal
3 WHERE puesto = '03'
4 GROUP BY cod_gim
5 HAVING COUNT(*) <= 2
6 LIMIT 1;
```



The screenshot shows a SQL query editor with a query that selects the gym code (cod\_gim) from the personal table where the position (puesto) is '03', grouped by gym code, with a maximum count of 2 per gym, and limited to 1 result. Below the query, a 'Result Grid' is displayed with a single row showing the value '20' for the column 'cod\_gim'.

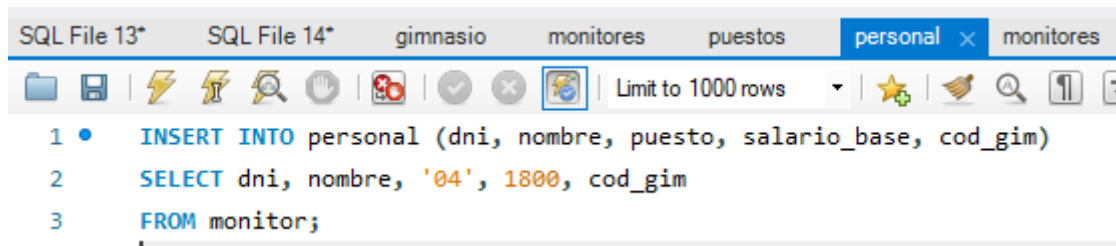
cod_gim
20

Y después (podríamos sustituir por cod\_gim=20):

```
1 • INSERT INTO monitores (dni, nombre, cod_gim, catego)
2 VALUES ('67676767', 'Sanz Bueno, Mariano',
3         (SELECT cod_gim FROM personal WHERE puesto = '03' GROUP BY cod_gim HAVING COUNT(*) <= 2 LIMIT 1), 'COM1');
4
```

## 2.4)

Inserta en la tabla PERSONAL todos los datos de la tabla MONITORES, con su código de gimnasio, su DNI y nombre, y asignándole un puesto '04' y el salario\_base 1800.



The screenshot shows a SQL IDE interface with a toolbar and a query editor. The toolbar includes icons for file operations, execution, and search. The query editor contains the following SQL code:

```
1 • INSERT INTO personal (dni, nombre, puesto, salario_base, cod_gim)
2   SELECT dni, nombre, '04', 1800, cod_gim
3   FROM monitor;
```

## 2.5)

Modifica en la tabla GIMNASIO el código del gimnasio de nombre 'EgoGym', por el código 60. ¿Algún problema o error? En caso afirmativo explica el por qué.

```
13 • UPDATE gimnasio
14     SET cod_gim = 60
15     WHERE nombre = 'EgoGym';
```

Si el nombre 'EgoGym' es único y solo existe un gimnasio con ese nombre en la tabla "gimnasio", entonces la sentencia funcionará correctamente y el código del gimnasio se cambiará a 60.

Sin embargo, si hay más de un gimnasio con el nombre 'EgoGym', la consulta modificará el código de todos los gimnasios con ese nombre, lo cual no es deseado. En este caso, es mejor usar la columna "cod\_gim" en la cláusula WHERE en lugar del nombre del gimnasio, ya que la columna "cod\_gim" es única y garantiza la modificación correcta del gimnasio.



## 2.6)

En la tabla GIMNASIO, cambia el teléfono del gimnasio 60 a 777334455 y la tarifa base la incrementas en 10 euros, solo si la tarifa actual es menor de 100 euros. (Con una sola sentencia UPDATE).

```
3 • UPDATE gimnasio
4 SET telefono = '777334455',
5     tarifa_base = tarifa_base + 10
6 WHERE cod_gim = 60 AND tarifa_base < 100;
7 |
```

## 2.7)

Incrementa en un 5% la columna `tarifa_base` (redondeando ese valor con 2 decimales) de la tabla GIMNASIO, sólo a los gimnasios que tengan algún monitor categoría 'CAR1'.

```
5 • UPDATE gimnasio
6   SET tarifa_base = ROUND(tarifa_base * 1.05, 2)
7   WHERE cod_gim IN (
8     SELECT DISTINCT cod_gim FROM monitor
9     WHERE categoria = 'CAR1'
10  );
```

## 2.8)

Elimina de la tabla CATEGORIAS la de nombre 'Yoga'. ¿Se puede? Razona y explica lo ocurrido en la tabla monitores..


No siempre se puede eliminar una categoría de la tabla CATEGORIAS. En este caso, al intentar eliminar la categoría 'Yoga', se producirá un error al existir registros en la tabla MONITORES que tengan ese valor en la columna 'catego', debido a que la restricción FOREIGN KEY que se ha establecido impide la eliminación de una categoría referenciada en otra tabla.

Para eliminar la categoría 'Yoga' de forma segura, primero se deberían eliminar todos los registros de la tabla MONITORES que tengan esa categoría como valor en la columna 'catego', o bien cambiar su categoría a otra existente en la tabla CATEGORIAS.

Si se intenta eliminar la categoría 'Yoga' sin realizar previamente estas acciones, se producirá un error de integridad referencial y no se realizará la eliminación. En este caso, los registros de la tabla MONITORES que tengan 'Yoga' como valor en la columna 'catego' quedarán huérfanos, es decir, sin una categoría válida referenciada en la tabla CATEGORIAS.

En nuestro caso se ha podido lanzar sin error:

```
1 • SELECT * FROM gimnasios.categorias;
2
3 -- DELETE FROM categorias
4 -- WHERE nombre = 'Yoga';
5
```



	cod_cat	nombre	incentivo
▶	CAR1	CARDIO	0.00
	COM1	COMBO	0.00
	HIT1	HIT	0.00
*	NULL	NULL	NULL

## 2.9)

### a)

Inserta en la tabla GIMNASIO uno nuevo de nombre 'TodoGym', código 900, dirección 'Maravillas, 110', teléfono 666112233, tipo 'SUP'. Razona y explica si se ha podido o no insertar. En tu razonamiento observa que no se ha dado valor para la tarifa\_base y es obligatoria.

```
3 • INSERT INTO `gimnasios`.`gimnasio`  
4   VALUES ('900', 'SUP', 'ActivaFitness', 'Maravillas, 110', '666112233')  
-
```

Sí, porque hay una tarifa base por defecto (11.00).

### b)

Una vez insertado, asígnale al gimnasio 900 la tarifa máxima de todos los gimnasios.

```
8 • UPDATE gimnasio  
9   SET tarifa_base = (  
10     SELECT MAX(tarifa_base) FROM gimnasio  
11   )  
12 WHERE cod_gim = 900;  
|
```

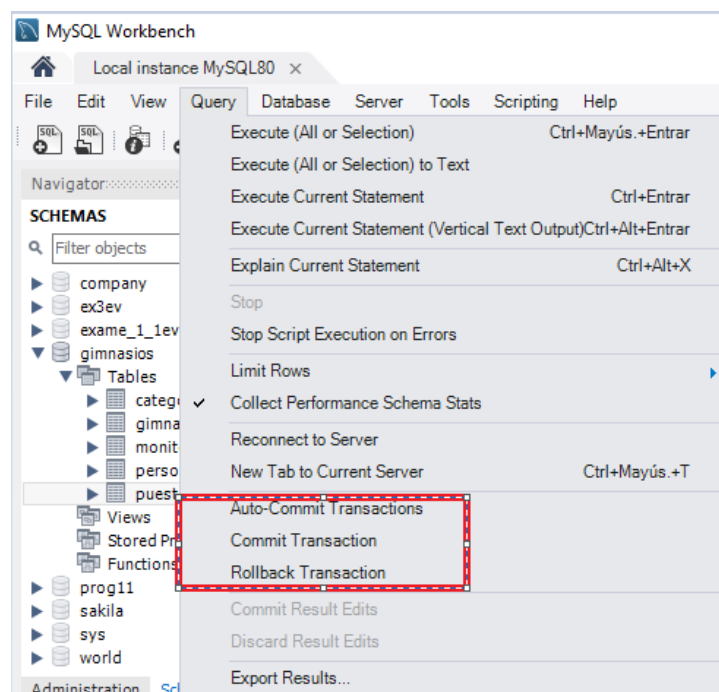
## APARTADO 3.

Ahora vas a practicar con transacciones. Redacta las sentencias SQL que permiten realizar y probar las siguientes transacciones.

Por defecto MySQL trabaja con el modo AUTOCOMMIT activado. Si utilizas tablas InnoDB y quieres trabajar con transacciones de forma permanente puedes desactivar este modo poniendo:

```
SET AUTOCOMMIT=0;  
COMMIT;
```

También a través de interfaz gráfica:



Ante errores, seguí estas indicaciones:

“You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column. To disable safe mode, toggle the option in Preferences -> SQL Editor and reconnect.”

```
SET SQL_SAFE_UPDATES = 0;  
-- Mi código SQL  
SET SQL_SAFE_UPDATES = 1;
```

### 3.1)

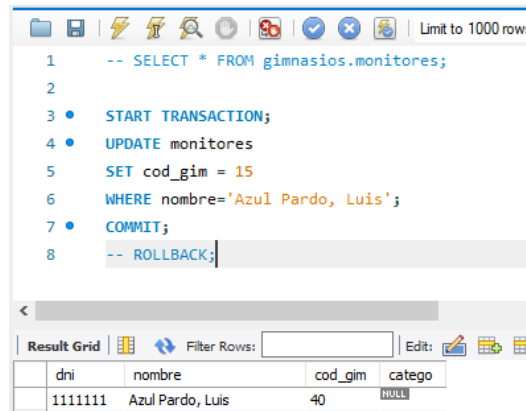
Inicia una transacción

Asigna al monitor 'Azul Pardo, Luis' al gimnasio '20'

Confirma o valida los cambios finalizando la transacción.

Comprueba que se ha cambiado.

Comprueba que ya no se pueden deshacer los cambios realizados.



```
1  -- SELECT * FROM gimnasios.monitores;
2
3  •  START TRANSACTION;
4  •  UPDATE monitores
5     SET cod_gim = 15
6     WHERE nombre='Azul Pardo, Luis';
7  •  COMMIT;
8  -- ROLLBACK;
```

Result Grid

dni	nombre	cod_gim	catego
1111111	Azul Pardo, Luis	40	NULL

Una vez ejecutado:

	dni	nombre	cod_gim	catego	email
►	1111111	Azul Pardo, Luis	15	NULL	NULL

Y hacer un rollback no tiene ningún efecto:

#	Time	Action	Message
77	21:17:12	ROLLBACK	0 row(s) affected

### 3.2)

Inicia otra transacción

Elimina los gimnasios de código 60 y 600.

Modifica en la tabla PUESTOS el nombre del puesto '04' por 'Otros'.

Comprueba que se ha cambiado el puesto y eliminado los gimnasios.

Deshaz la operación finalizando la transacción.

Comprueba que se vuelve a tener el gimnasio 60 y 600, y el nombre original del puesto '04'

Los cambios no se llegan a producir porque con rollback se deshacen los de la transacción.

```
1      -- SELECT * FROM gimnasios.gimnasio;
2
3  •   SET SQL_SAFE_UPDATES = 0;
4  •   START TRANSACTION;
5  •   DELETE FROM gimnasio
6      WHERE cod_gim=60 or cod_gim=600;
7
8  •   UPDATE puestos
9      SET nombre = 'Otros'
10     WHERE cod_pue='04';
11
12     -- COMMIT;
13  •   ROLLBACK;
14  •   SET SQL_SAFE_UPDATES = 1;
```

### 3.3)

Inicia una transacción que suponga la modificación de la tarifa\_base de dos gimnasios.  
Incluye un punto de salvaguarda entre las dos modificaciones.  
Debes deshacer la transacción hasta el punto de salvaguarda  
Comprueba qué modificaciones, de las anteriores, se ha realizado.

Se recupera hasta el punto de salvaguarda, perdiéndose el resto de cambios.

```
-- SELECT * FROM gimnasios.gimnasio;

SET SQL_SAFE_UPDATES = 0;
START TRANSACTION;
UPDATE gimnasios
SET tarifa_base = 12
WHERE cod_gim='20';
SAVEPOINT sp1;
SET tarifa_base = 13
WHERE cod_gim='30';
ROLLBACK TO SAVEPOINT sp1;
COMMIT;
SET SQL_SAFE_UPDATES = 1;
```



## APARTADO 4.

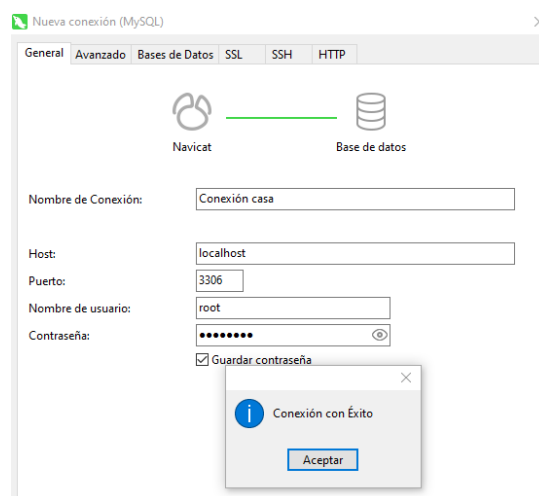
Descarga el software [Cliente Navicat para MySQL](#) e instálalo en tu equipo.

### 4.1.

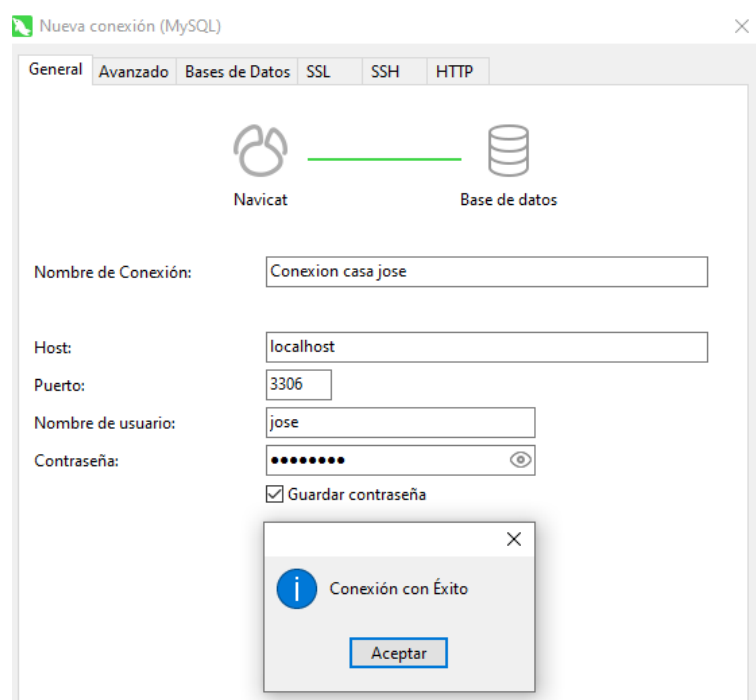
Crea una conexión a tu servidor MySQL como usuario root y otra conexión para otro usuario.

Conéctate como usuario root a Navicat y realiza en modo gráfico, no mediante sentencias SQL, los siguientes apartados sobre la base de datos GIMNASIOS

Root:

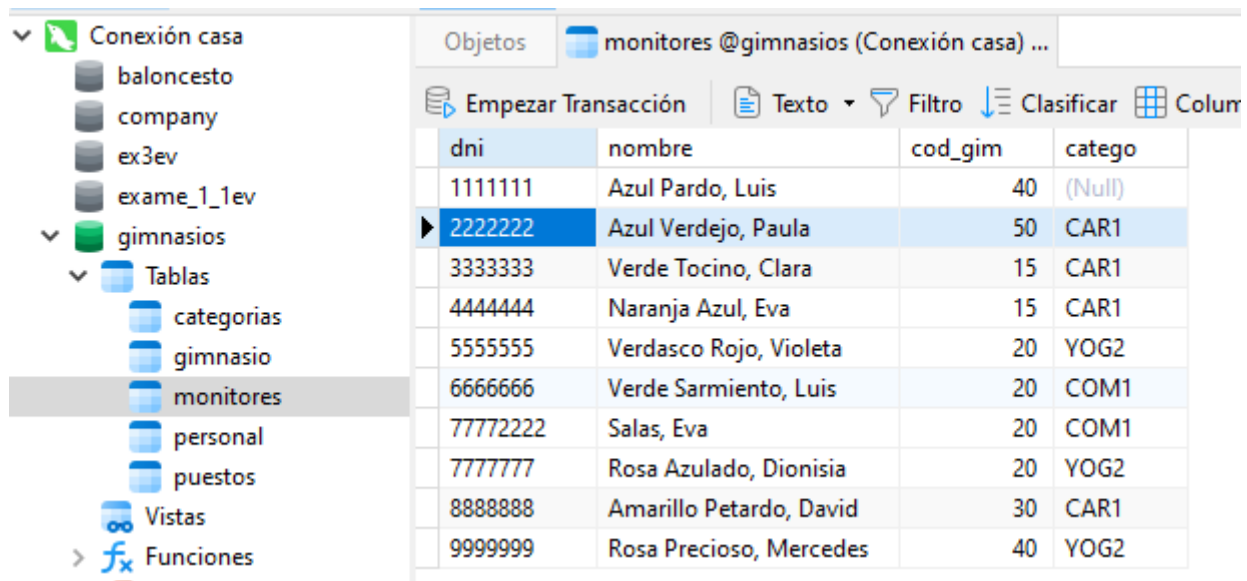


Usuario jose:



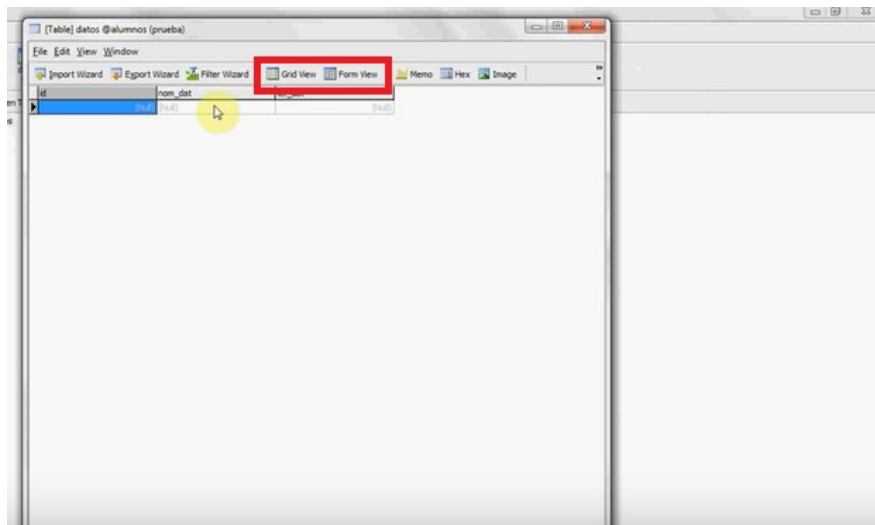
## 4.2

Visualiza la tabla MONITORES de la base de datos como Formulario y como Cuadrícula.

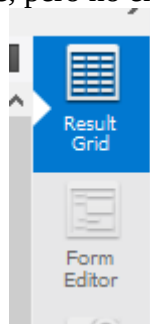


dni	nombre	cod_gim	catego
1111111	Azul Pardo, Luis	40	(Null)
2222222	Azul Verdejo, Paula	50	CAR1
3333333	Verde Tocino, Clara	15	CAR1
4444444	Naranja Azul, Eva	15	CAR1
5555555	Verdasco Rojo, Violeta	20	YOG2
6666666	Verde Sarmiento, Luis	20	COM1
7777222	Salas, Eva	20	COM1
7777777	Rosa Azulado, Dionisia	20	YOG2
8888888	Amarillo Petardo, David	30	CAR1
9999999	Rosa Precioso, Mercedes	40	YOG2

Creo que las opciones Formulario y Cuadrícula son de una versión anterior de Navicat. No las encontré en la versión actual, pero si referencias a ellas en un videotutorial de hace 9 años:  
<https://www.youtube.com/watch?v=Y84g6knDPcM&t=69s>



También en MySql Workbench las encontré, pero no en la versión actual de navicat



## 4.3

Añade un campo 'observaciones' tipo TEXT a la tabla GIMNASIO para guardar un comentario u observación sobre el gimnasio.

Guardar

Agregar Campo

Insertar Campo

Eliminar Campo

Clave primaria

Mover Arriba

Mover Abajo

Campos

Índices

Clave Foraneas

Comprobaciones

Triggers

Opciones


Comentar


Vista previa de SQL


Nombre	Tipo	Longitud	Decimale	No es nul	Virtual	Clave	Comentar
cod_gim	smallint			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
tipo_gim	enum			<input checked="" type="checkbox"/>	<input type="checkbox"/>		
nombre	varchar	45		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
direccion	varchar	80		<input type="checkbox"/>	<input type="checkbox"/>		
telefono	varchar	15		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
tarifa_base	double	9	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
* observaciones	text	255		<input type="checkbox"/>	<input type="checkbox"/>		


## 4.4


Añade un nuevo campo o columna tipo VARCHAR(45) a la tabla MONITORES para guardar su e-mail.


 Guardar


 Agregar Campo

 Insertar Campo

 Eliminar Campo

 Clave primaria

 Mover Arriba

 Mover Abajo

Campos

Índices

Clave Foraneas


Comprobaciones

Triggers

Opciones

Comentar

Vista previa de SQL

Nombre	Tipo	Longitud	Decimale	No es nul	Virtual	Clave	Comentar
dni	char	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	
nombre	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>		
cod_gim	smallint			<input checked="" type="checkbox"/>	<input type="checkbox"/>		
catego	char	4		<input type="checkbox"/>	<input type="checkbox"/>		
* email	varchar	45		<input type="checkbox"/>	<input type="checkbox"/>		

## 4.5

Modifica los datos de dos monitores y dos gimnasios para incluir esta nueva información en los columnas añadidas.

Monitores:

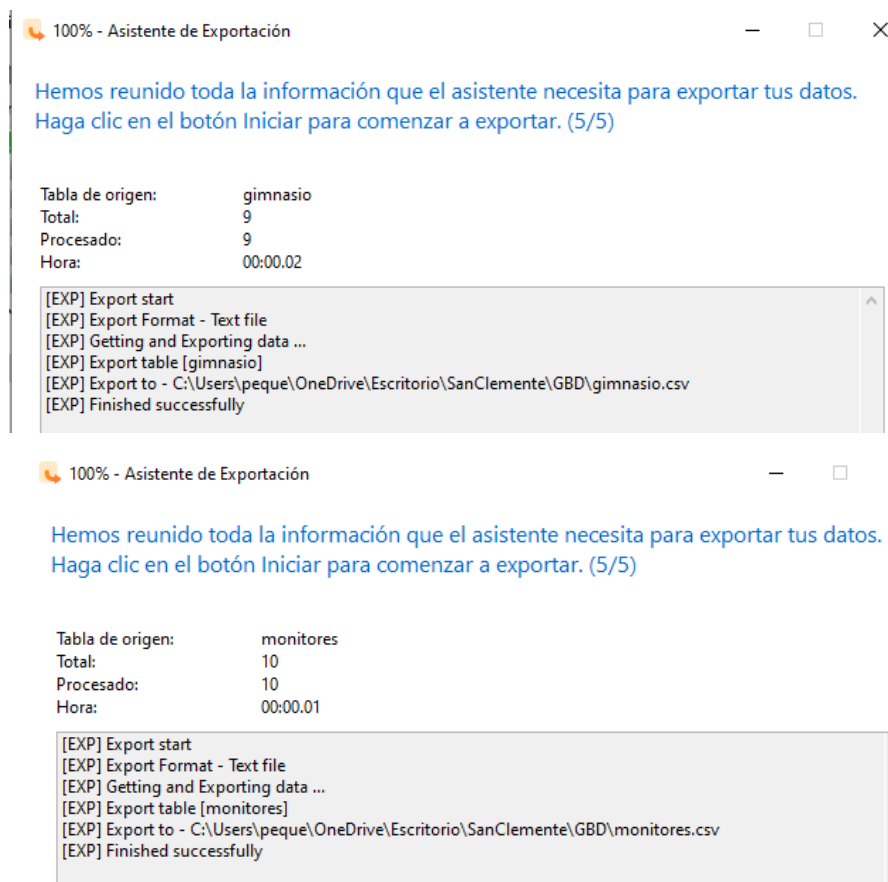
8888888	Amarillo Petardo, David	30	CAR1	david@sanclemente.com
9999999	Rosa Precioso, Mercedes	40	YOG2	rosa@sanclemente.com

Gimnasios:

cod_gim	tipo_gim	nombre	direccion	telefono	tarifa_base	observaciones
15	INF	ActivaFitness	Cangrejo, 27	635243211	50,00	Probando...
20	INF	VivaFitness	Esmeralda, 18	671530970	40,00	Otra prueba...

## 4.6

Exporta los datos de las tablas monitores y gimnasios a formato CSV. (los ficheros exportados los debes adjuntar a la tarea)



Adjuntados los ficheros exportados.