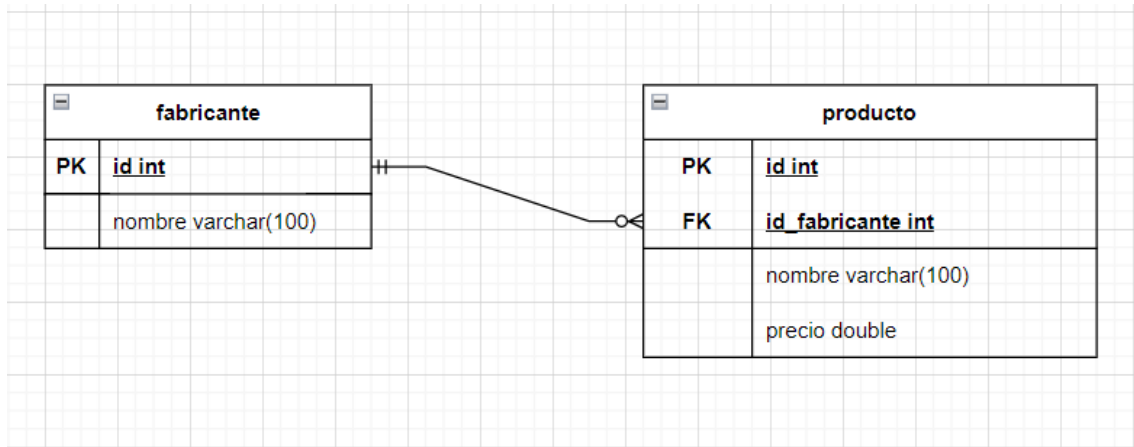


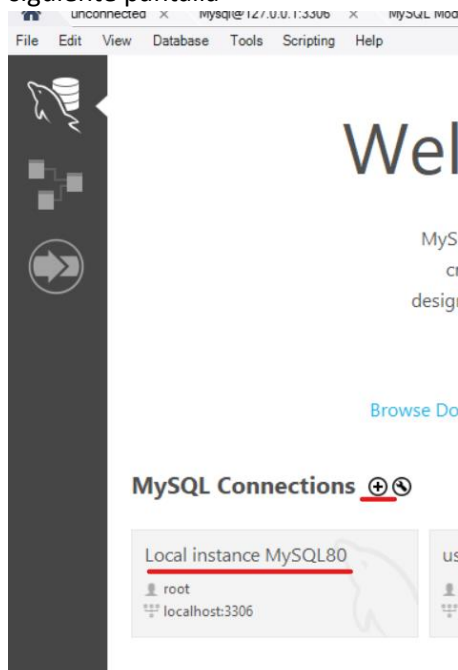
A4-1 Creación base de datos



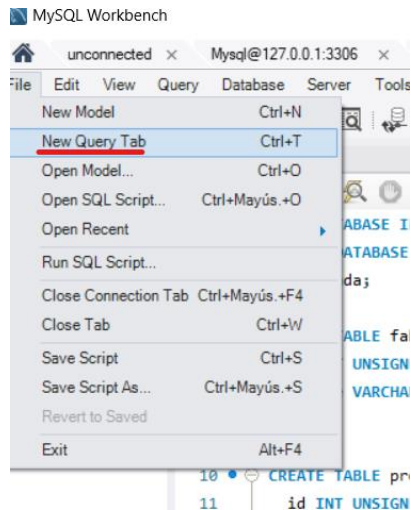
Realizar los siguientes apartados que se ponen a continuación y sacar una captura de pantalla de cada uno de ellos. Pegar la captura debajo de cada apartado. El fichero se grabará con el formato ***nombre_apellido1_apellido2.pdf***. Sacar capturas de pantalla a partir del apartado 2 que es cuando comienza la creación de la base de datos. Si os sale algún error podéis sacar captura de pantalla del error e indicar cómo lo solucionasteis.

Pasos a seguir para realizar el ejercicio.

- 1- Es necesario haber instalado MySQL Workbench. Abrimos el programa y nos aparece la siguiente pantalla



Abrimos la conexión existente (*local instance MySQL80*) o creamos una nueva con el botón +. Nos solicitará la contraseña que elegimos al instalar *Workbench*. Una vez dentro abrimos una nueva pestaña de consultas.



- 1- Crear una **base de datos** llamada **tienda** con las especificaciones de las tablas de arriba. Todos los atributos de ambas tablas son obligatorios (not null). Para crear la base de datos y la tabla se debe usar el comando **CREATE TABLE**, no es necesario indicar el juego de caracteres que va a usar, podemos dejar el que viene por defecto. Una vez creada la base de datos guardar el script

```
DROP DATABASE IF EXISTS tiendab;
CREATE DATABASE tiendab CHARACTER SET utf8mb4;
```

- 2- Una vez creada una base de datos, para empezar a poder trabajar con ella, deberemos poner el comando **USE**.

```
USE tiendab;
```

- 3- Una vez creada la base de datos con ambas tablas, mostrarlas con el comando **SHOW**

```
5 • CREATE TABLE fabricante (
6   id INT PRIMARY KEY,
7   nombre VARCHAR(100) NOT NULL
8 );
9
10 • CREATE TABLE producto (
11   id INT PRIMARY KEY,
12   nombre VARCHAR(100),
13   precio DOUBLE,
14   id_fabricante INT,
15   FOREIGN KEY (id_fabricante) REFERENCES fabricante (id)
16 );
17
18 • SHOW TABLES;
```

	Tables_in_tiadab
►	fabricante
	producto

- 4- Visualizamos la estructura de las dos tablas con el comando **DESCRIBE** para consultar los atributos de cada una de las tablas.

```
DESCRIBE fabricante;
```

	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	NULL	
	nombre	varchar(100)	NO		NULL	

```
DESCRIBE producto;
```

	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	NULL	
	nombre	varchar(100)	YES		NULL	
	precio	double	YES		NULL	
	id_fabricante	int	YES	MUL	NULL	

- 5- Cambiamos el nombre a la tabla **producto** para que pase a llamarse **productos_informaticos**

```
ALTER TABLE producto RENAME TO productos_informaticos;
```

```
SHOW TABLES;
```

- 6- Mostrar las tablas con el comando **SHOW** para comprobar que se ha realizado el cambio de nombre.

	Tables_in_tiadab
►	fabricante
	productos_informaticos

- 7- Añadir una nueva columna a la tabla **fabricante** que se llame **provincia** y que sea de tipo **VARCHAR(20)**. Debe estar situada justo después del campo **id**. Se debe usar el comando **ALTER**

```
ALTER TABLE fabricante
ADD provincia VARCHAR(20) AFTER id;
DESCRIBE fabricante;
```

	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	NULL	
	provincia	varchar(20)	YES		NULL	
	nombre	varchar(100)	NO		NULL	

- 8- Añadir otra columna a la tabla **fabricante** que se llame **garantia**. Debe estar situada justo después del campo **nombre**. Se debe usar el comando **ALTER... ADD...**

```
ALTER TABLE fabricante
ADD garantia VARCHAR(50) AFTER nombre;
DESCRIBE fabricante;
```

- 9- Borrar la columna garantía recién creada garantía. Usar el comando **ALTER... DROP...** Sacar captura de pantalla del antes y el después.

	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	NULL	
	provincia	varchar(20)	YES		NULL	
	nombre	varchar(100)	NO		NULL	
	garantia	varchar(50)	YES		NULL	

```
ALTER TABLE fabricante
DROP garantia;
DESCRIBE fabricante;
```

	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	NULL	
	provincia	varchar(20)	YES		NULL	
	nombre	varchar(100)	NO		NULL	

- 10- Crear una tabla que se llame **alumnos**. Debe tener un atributo **idAlumnos de tipo INT** que sea la **clave primaria**, y otro atributo llamado **nombre** que sea **VARCHAR(50)**.

```
CREATE TABLE alumno (  
    idAlumno INT PRIMARY KEY,  
    nombre VARCHAR(50)  
);
```

Tables_in_tiadab
alumno
fabricante
productos_informaticos

- 11- Insertar 6 alumnos en ella. Utilizando 3 métodos distintos (insertar 2 alumnos por cada método), comando **INSERT INTO**

```
INSERT INTO alumno VALUES  
(1, 'Antonio'),  
(2, 'Pepa');  
INSERT INTO alumno VALUES (3, 'Juanma');  
INSERT INTO alumno VALUES (4, 'Marta');  
INSERT INTO alumno (idAlumno, nombre) VALUES  
(5, 'Maricarmen'),  
(6, 'Chechu');
```

- 12- Mostrar todos los alumnos de la tabla ALUMNO con el comando **SELECT**.

```
SELECT * FROM alumno;
```

	idAlumno	nombre
▶	1	Antonio
	2	Pepa
	3	Juanma
	4	Marta
	5	Maricarmen
	6	Chechu
*	NULL	NULL

- 13- Añadir un atributo llamado **notas de tipo decimal** a la tabla **alumnos** ue permita introducir **6 dígitos en total, dos de ellos decimales**.

```
ALTER TABLE alumno  
ADD notas DECIMAL(6,2) AFTER nombre;  
DESCRIBE alumno;
```

	Field	Type	Null	Key	Default	Extra
►	idAlumno	int	NO	PRI	NULL	
	nombre	varchar(50)	YES		NULL	
	notas	decimal(6,2)	YES		NULL	

14- Insertar una nota a cada uno de los **alumnos**.

```
UPDATE alumno SET notas = 9.3 WHERE idAlumno = 1;
UPDATE alumno SET notas = 3.5 WHERE idAlumno = 2;
UPDATE alumno SET notas = 8.8 WHERE idAlumno = 3;
UPDATE alumno SET notas = 6.5 WHERE idAlumno = 4;
UPDATE alumno SET notas = 10.0 WHERE idAlumno = 5;
UPDATE alumno SET notas = 9.1 WHERE idAlumno = 6;
```

	idAlumno	nombre	notas
►	1	Antonio	9.30
	2	Pepa	3.50
	3	Juanma	8.80
	4	Marta	6.50
	5	Maricarmen	10.00
	6	Chechu	9.10
*	NULL	NULL	NULL

15- Borrar el campo **notas** de la base de datos **alumnos**.

```
ALTER TABLE alumno
DROP notas;
DESCRIBE alumno;
```

	Field	Type	Null	Key	Default	Extra
►	idAlumno	int	NO	PRI	NULL	
	nombre	varchar(50)	YES		NULL	

16- Borrar la tabla **alumnos** completamente de la base de datos

```
DROP TABLE alumno;
```

	Tables_in_tiadab
►	fabricante
	productos_informaticos

