

Construcción de software y toma de decisiones

TC2005B

Dr. Esteban Castillo Juarez

ITESM, Campus Santa Fe



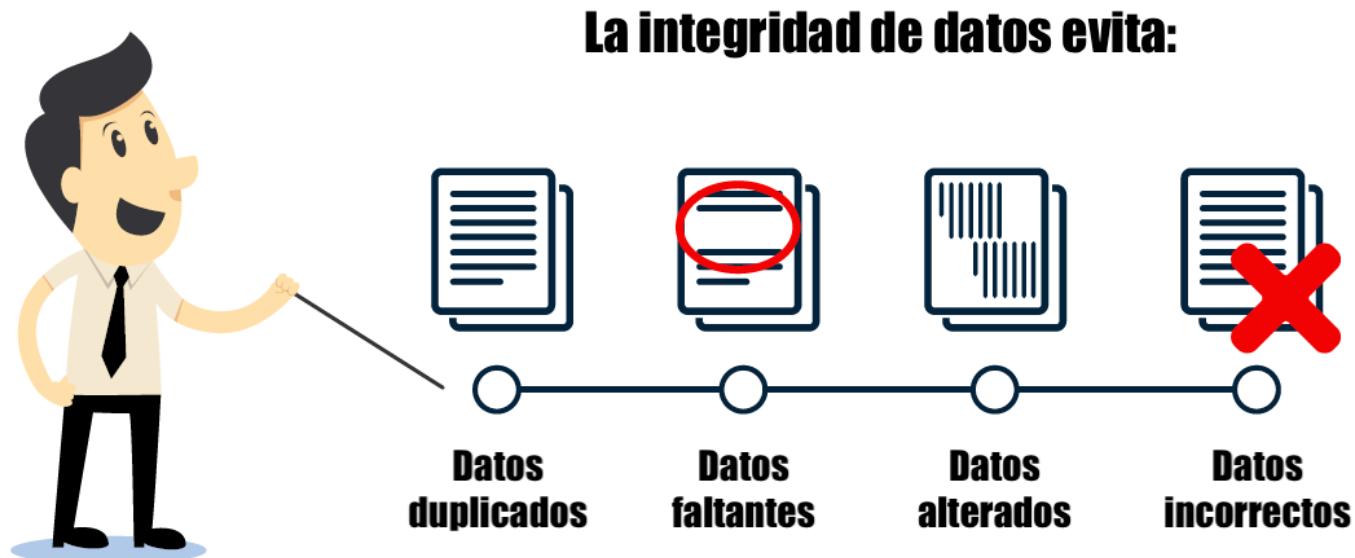
esteban.castillojz@tec.mx

Agenda

- Integridad en las bases de datos
- Uso de MySQL Workbench
- Creación de un esquema en MySQL
- Creación de una tabla en MySQL
- Referencias

Integridad de los datos

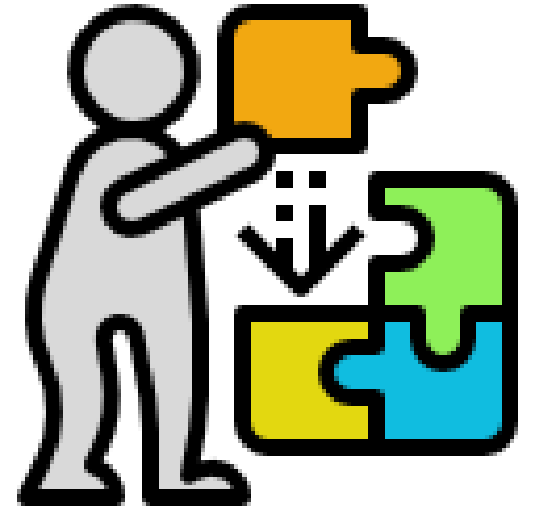
- El termino integridad de datos se refiere a la exactitud y completitud de la información en una base de datos.
- Cuando el contenido de una base de datos se modifica (agregar/eliminar tablas, filas columnas, etc.) la integridad de los datos almacenados puede perderse de distintas maneras.



Integridad de los datos

La integridad de los datos almacenados puede perderse de distintas maneras:

- Pueden añadirse datos no validos a la base de datos.
- Pueden modificarse datos existentes tomando un valor incorrecto.
- Los cambios pueden ser aplicados parcialmente.



Integridad de los datos

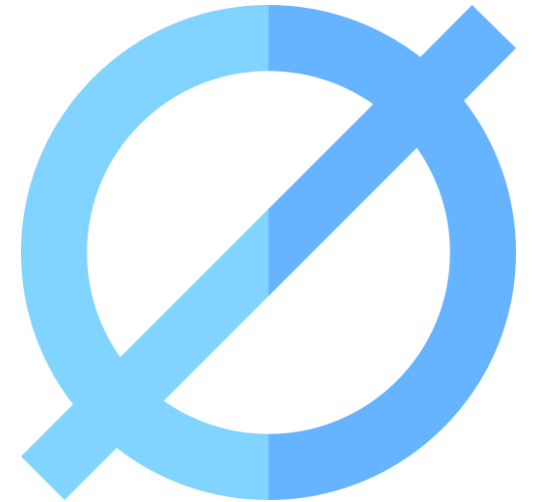
- Una función importante de SGBD relacional es el de preservar la integridad de los datos almacenados y de los que serán almacenados en el futuro.
- Para preservar la consistencia y corrección de la información, generalmente se imponen restricciones de integridad.
- Estas restricciones limitan el como se puede crear, cambiar o eliminar una tabla o un conjunto de tablas dentro de la base de datos.



Integridad de los datos

Varios tipos de restricciones de integridad suelen encontrarse en una base de datos relacional, incluyendo:

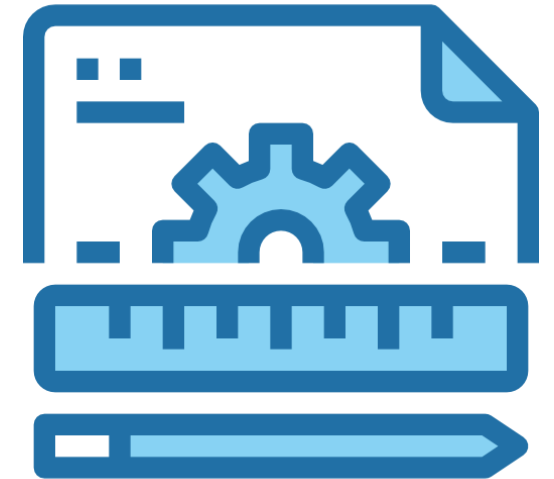
- **Datos requeridos:** Algunas columnas en una base de datos deben contener un valor de dato válido en cada fila, es decir, no se permite que tenga valores de tipo NULL o que falten.



Integridad de los datos

Varios tipos de restricciones de integridad suelen encontrarse en una base de datos relacional, incluyendo:

- **Chequeo de validez:** Cada columna de una base de datos tiene un **dominio** (ya lo habíamos dicho...), es decir, un conjunto de valores que son legales para esa columna. El SGBD puede estar preparado para impedir otros valores de datos en esas columnas.



Integridad de los datos

Varios tipos de restricciones de integridad suelen encontrarse en una base de datos relacional, incluyendo:

- **Integridad de entidad:** la **llave primaria** de una tabla debe contener un valor único en cada fila, diferente de los valores de todas las filas restantes.

Las restricciones de entidad aseguran que la llave primaria identifique inequívocamente a cada entidad representada en la base de datos.



Integridad de los datos

Varios tipos de restricciones de integridad suelen encontrarse en una base de datos relacional, incluyendo:

- **Integridad referencial:** Una **llave secundaria** (externa o foránea) en una base de datos relacional enlaza cada fila de una tabla “hijo” que contiene la llave foránea con la fila de la tabla “padre” que contiene el valor de la llave primaria correspondiente.

Las restricciones de integridad referencial aseguran que las relaciones entre tablas se preserven durante distintas actualizaciones de la base de datos (como manejar la supresión de filas que son referenciadas mediante otras filas).

Integridad de los datos

Varios tipos de restricciones de integridad suelen encontrarse en una base de datos relacional, incluyendo:

- **Consistencia:** muchas transacciones del mundo real producen múltiples actualizaciones a una base de datos.

La inserción de nuevas filas a una tabla debe ejecutarse de modo que la base de datos continúe en un estado correcto y consistente.

El SGBD puede ser preparada para forzar este tipo de regla de consistencia o para soportar aplicaciones que implementen tales reglas.

Integridad de los datos

**Me learning
MySQL**



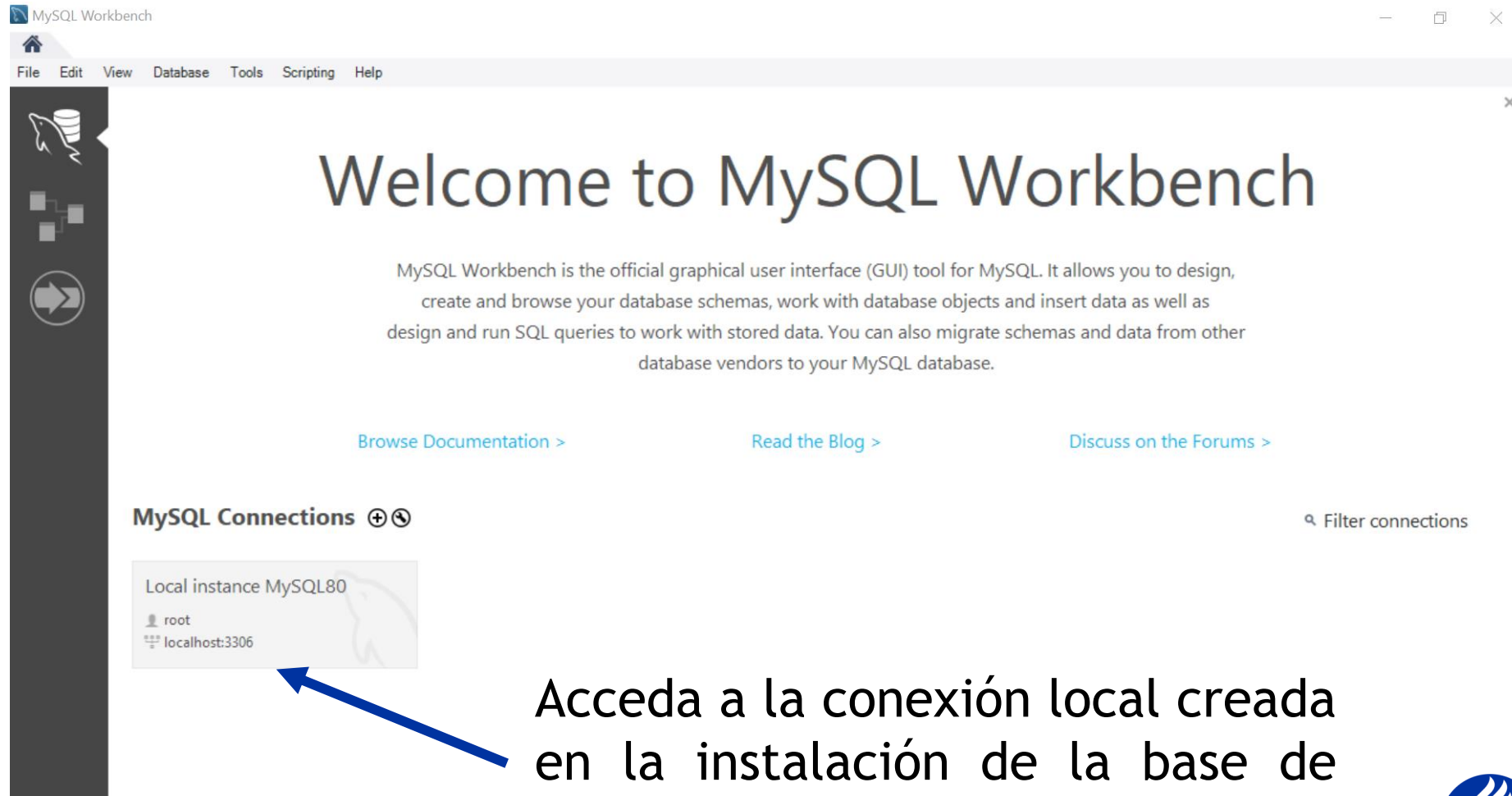
**Me learning
normalisation**



Uso de MySQL Workbench

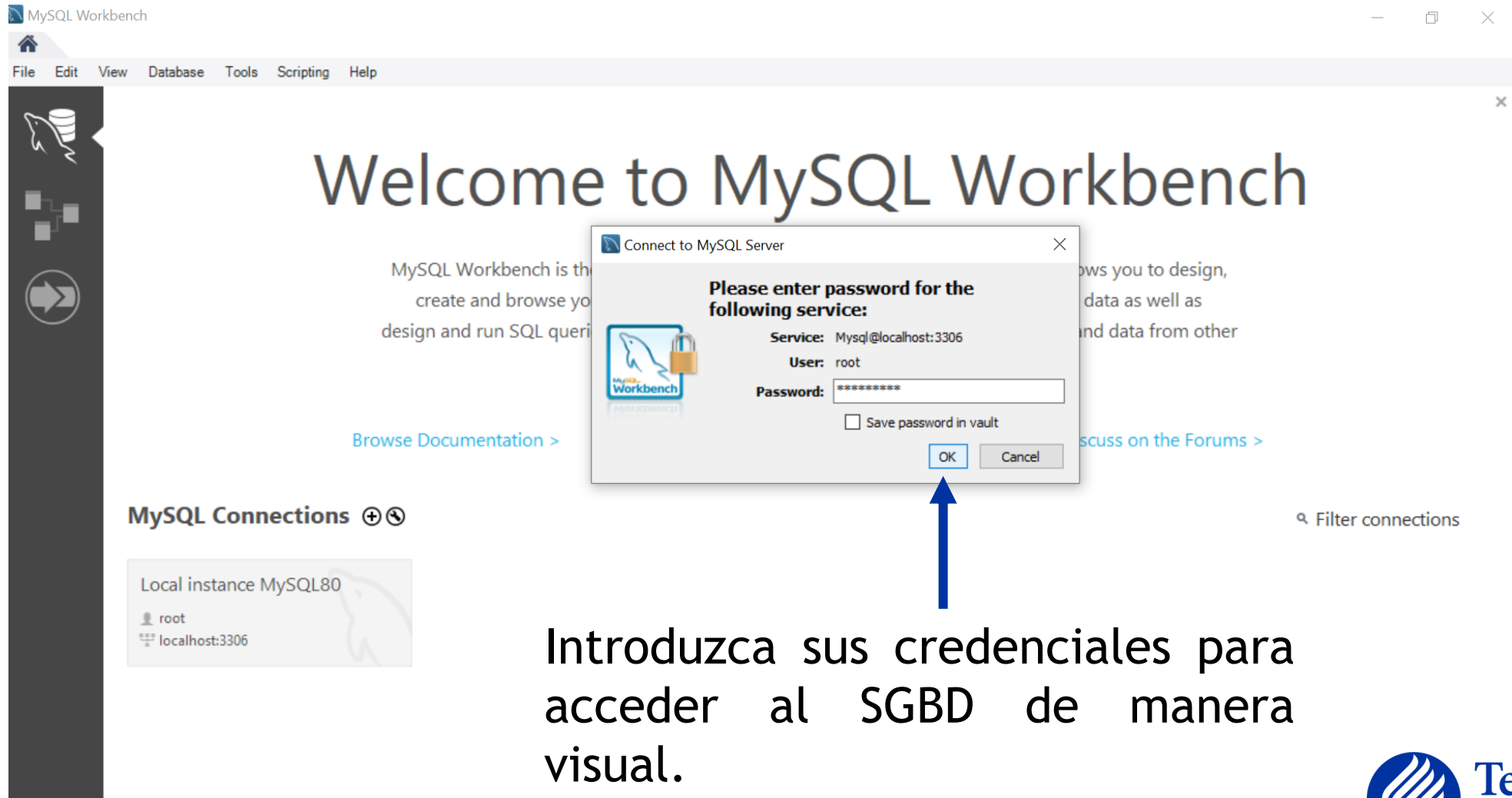


Uso de MySQL Workbench



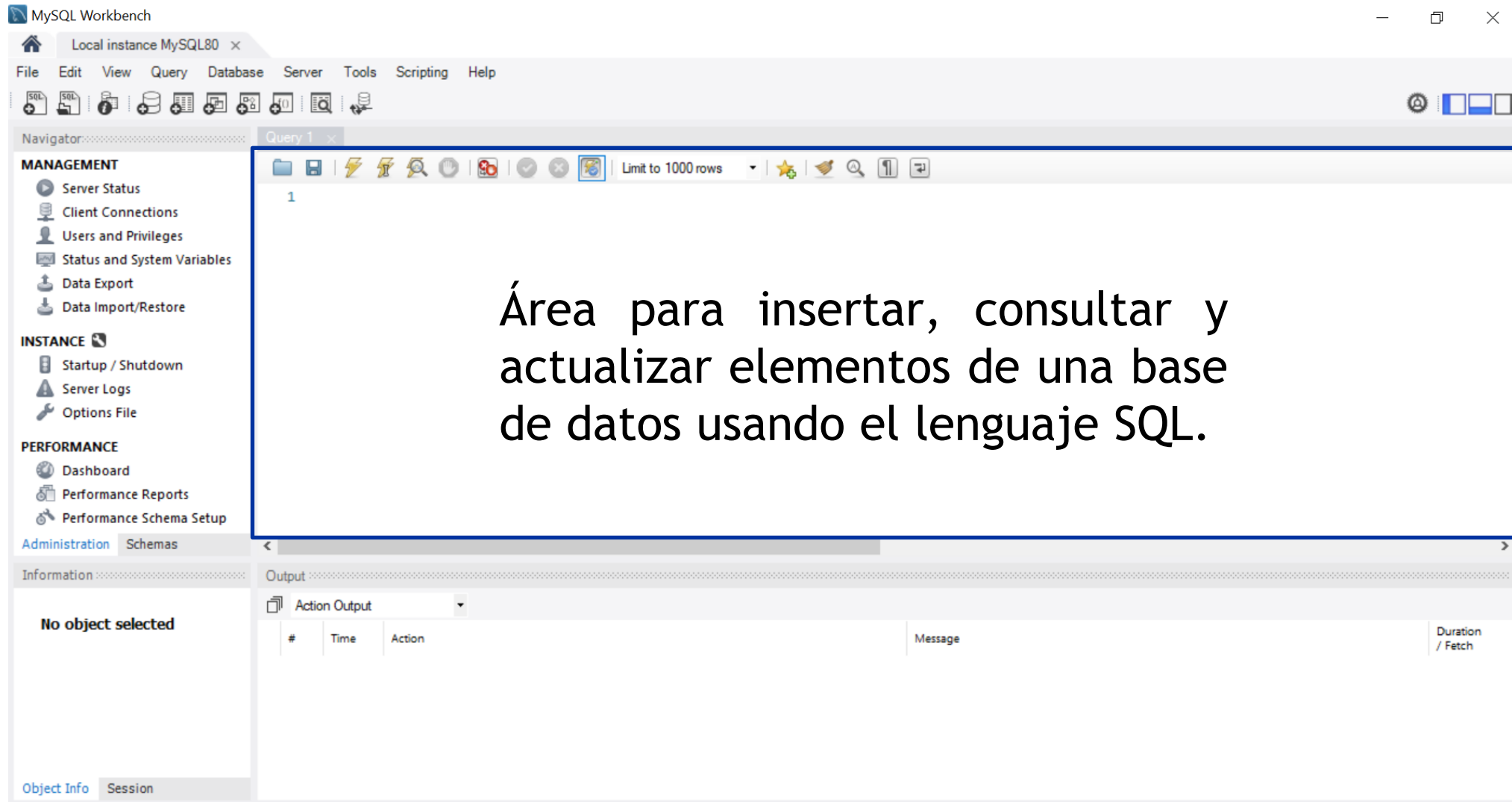
Acceda a la conexión local creada en la instalación de la base de datos.

Uso de MySQL Workbench

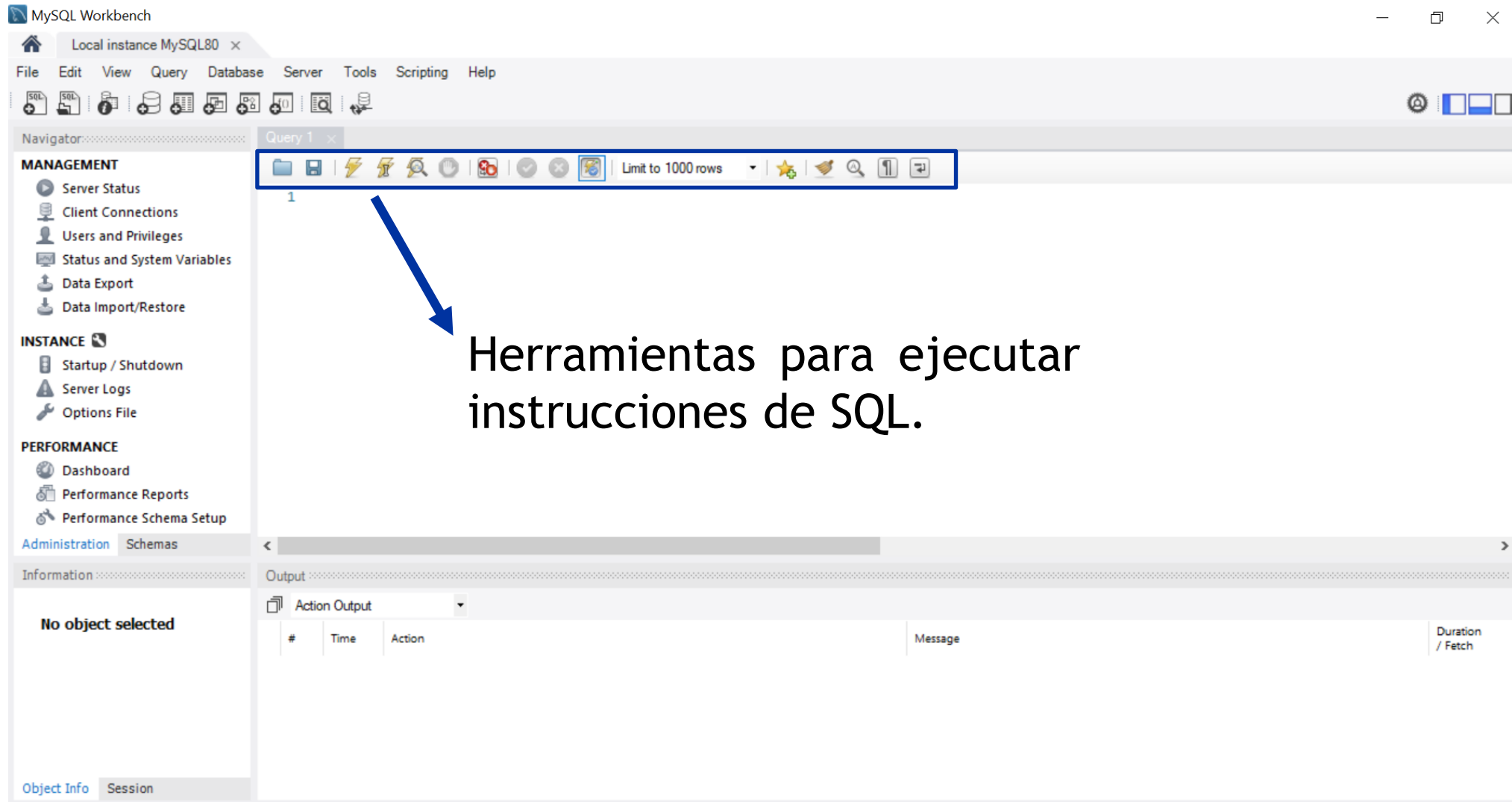


Introduzca sus credenciales para acceder al SGBD de manera visual.

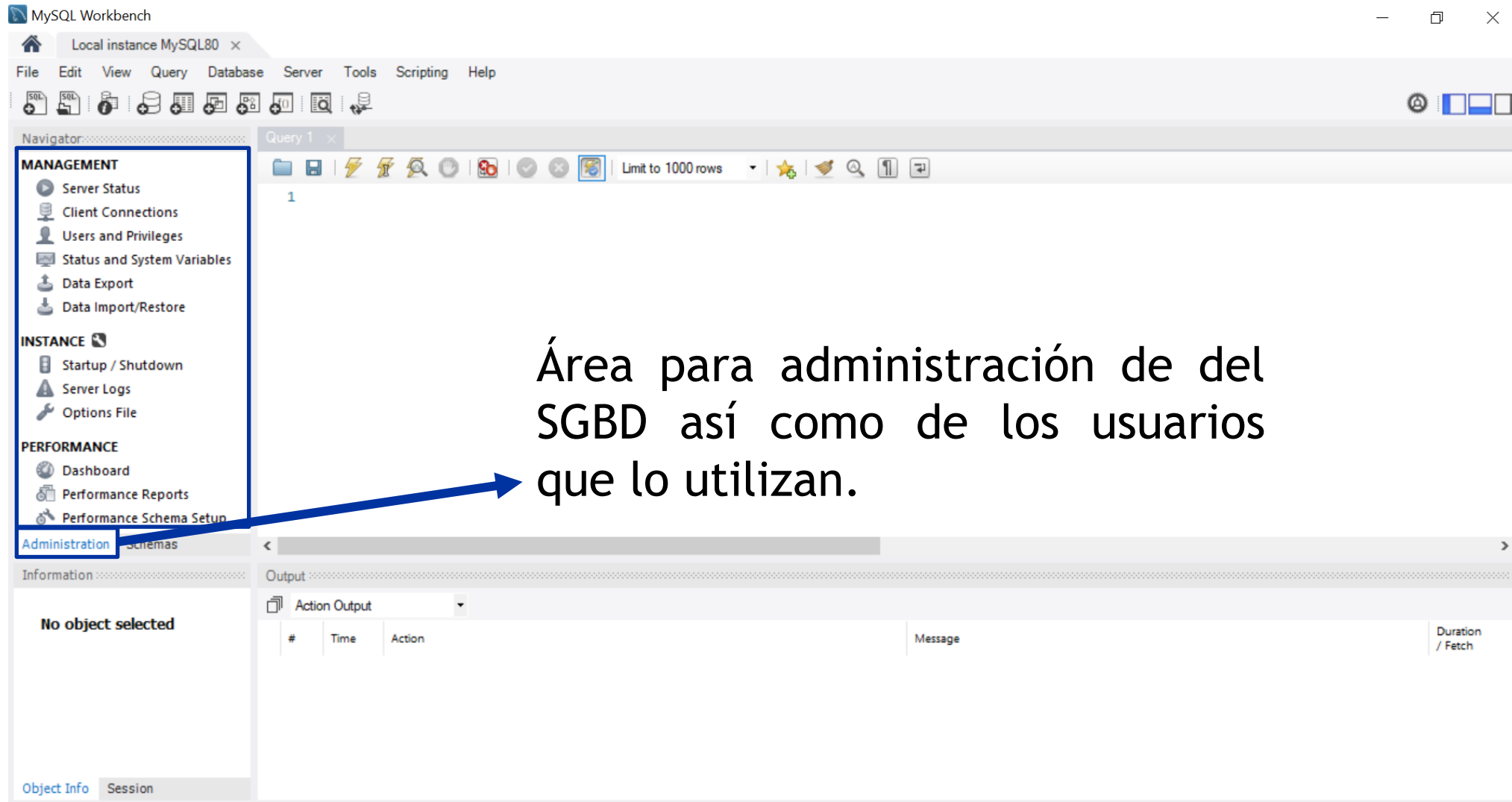
Uso de MySQL Workbench



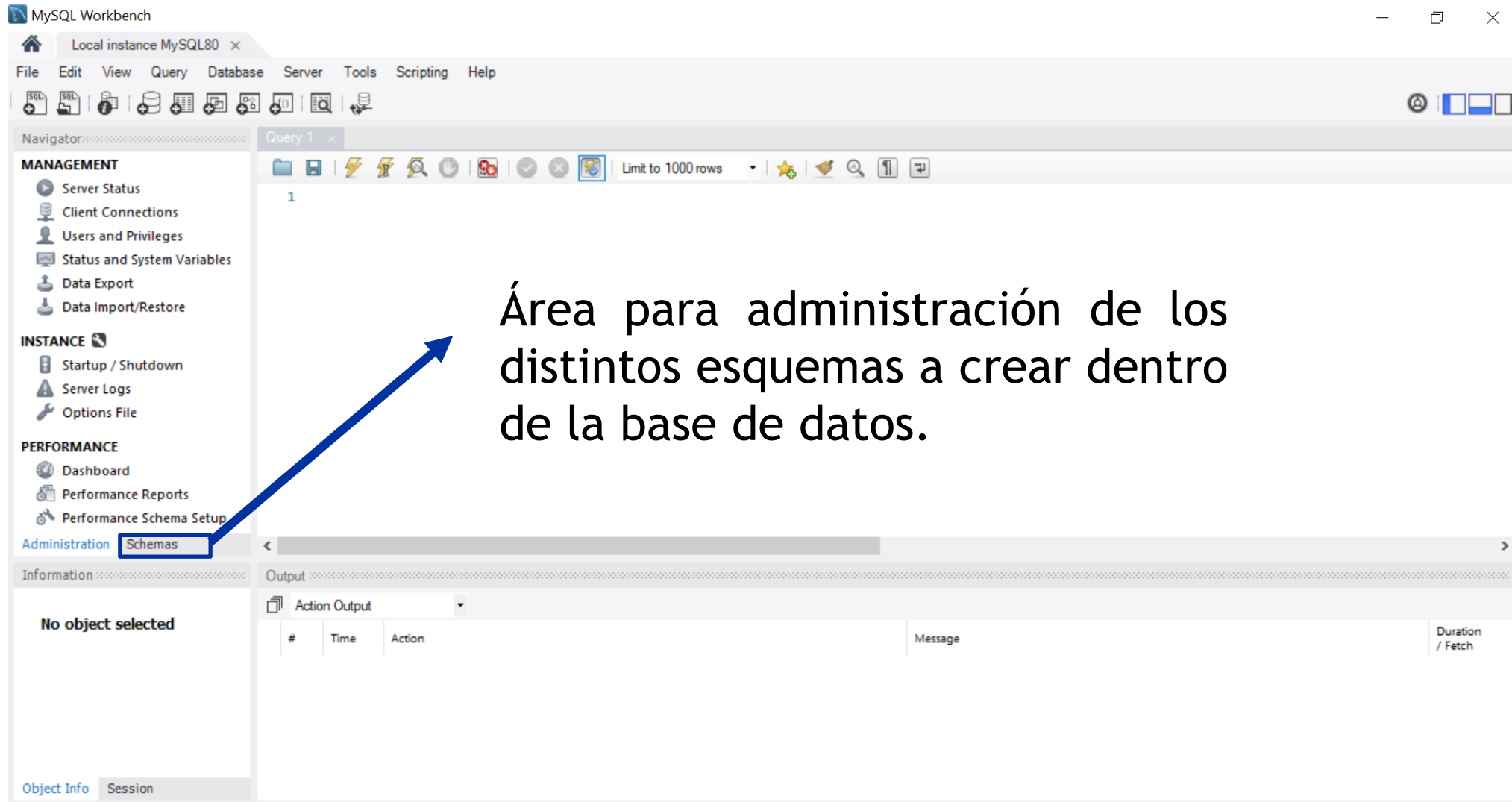
Uso de MySQL Workbench



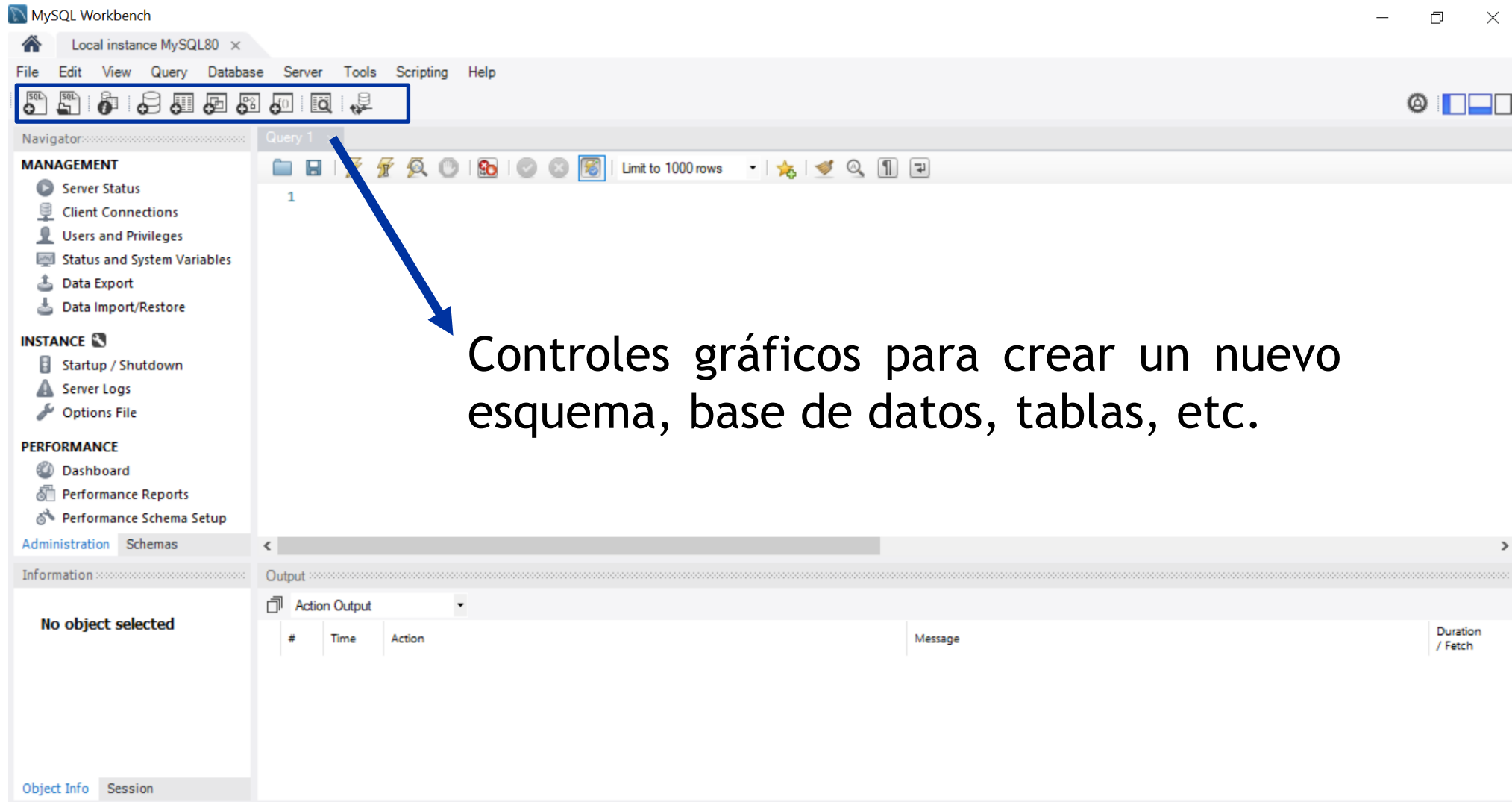
Uso de MySQL Workbench



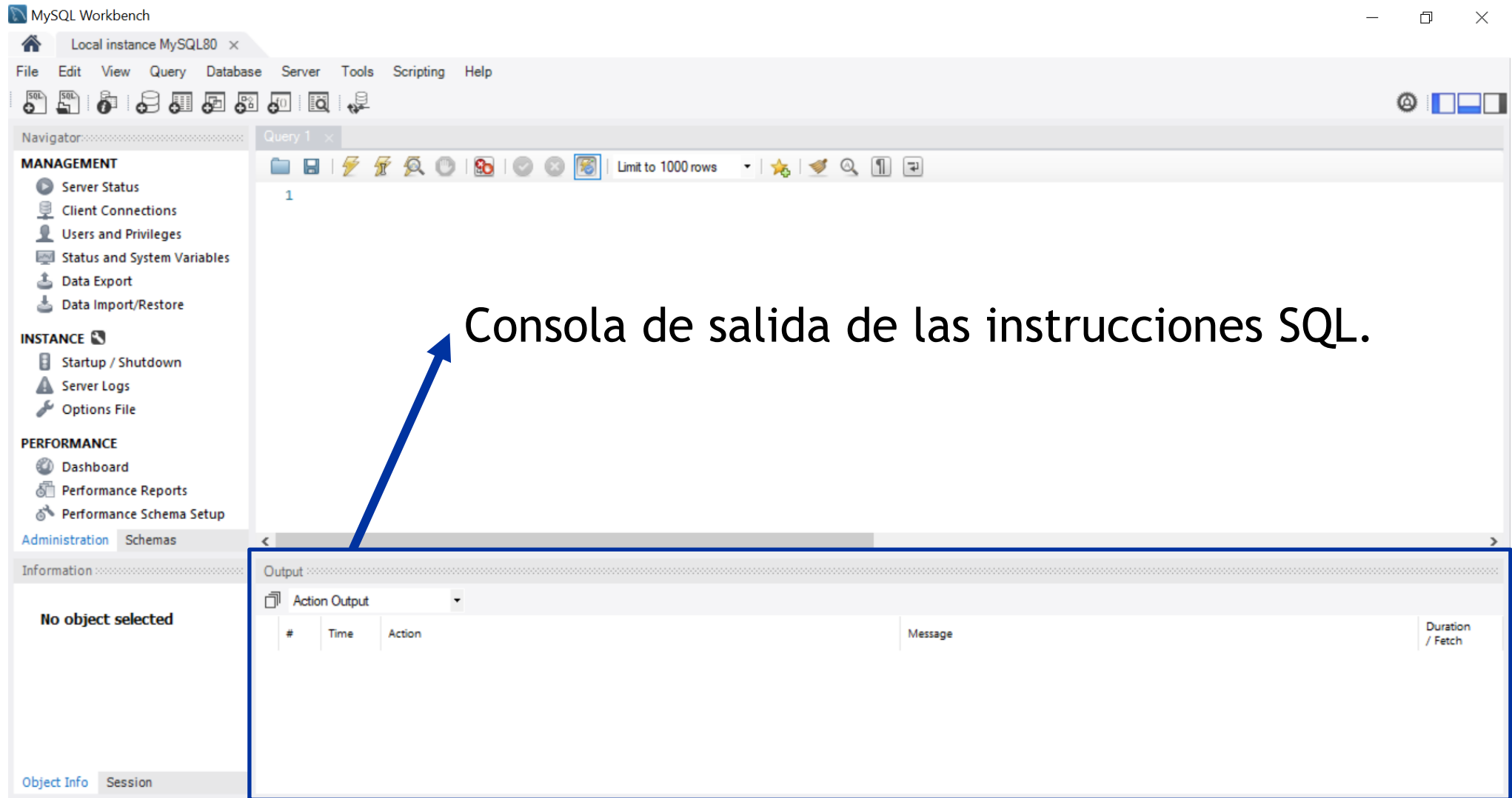
Uso de MySQL Workbench



Uso de MySQL Workbench



Uso de MySQL Workbench



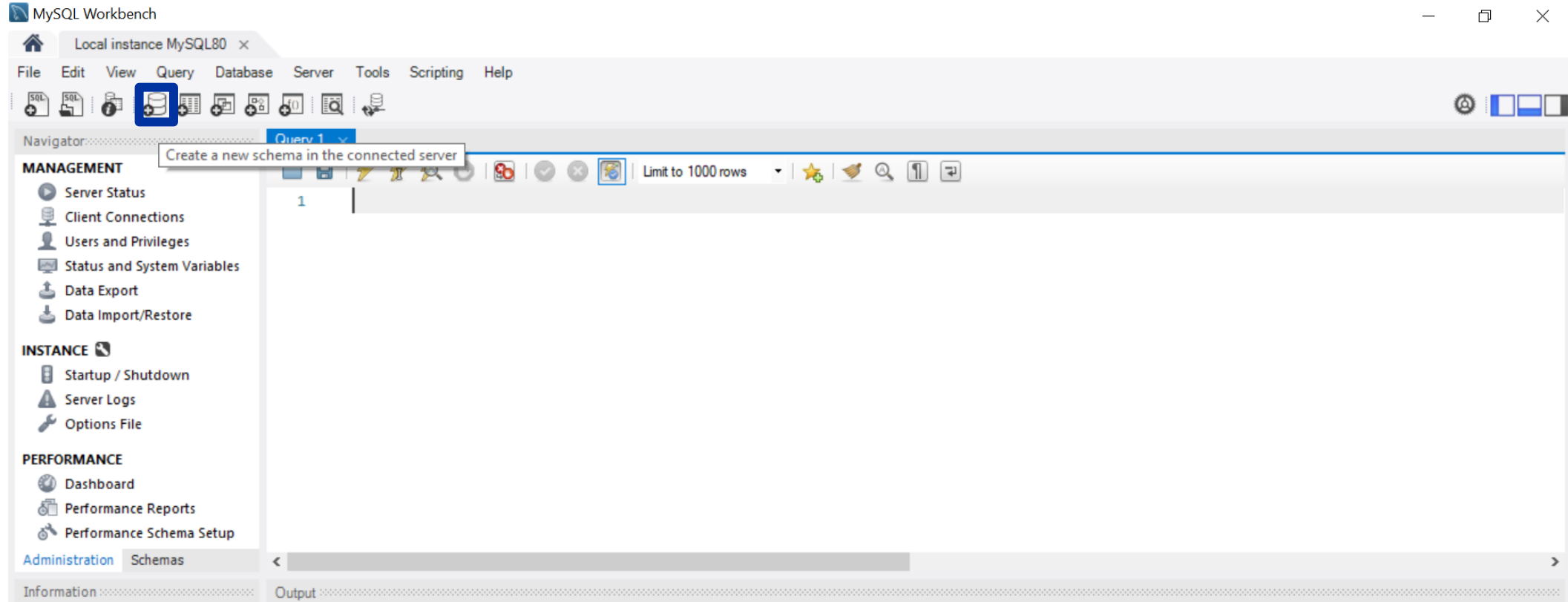
Creación de un esquema en MySQL

- Lo primero que tenemos que hacer antes de crear tablas es crear lo que se conoce como esquema. Esto nos permite tener varias bases de datos en un mismo SGBD (como si fueran carpetas, por ejemplo).
- Usando la terminología de una base de datos, un esquema es sinónimo de un base de datos específica.



Creación de un esquema en MySQL


Usando MySQL Workbench

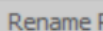


Creación de un esquema en MySQL

Usando MySQL Workbench

Query 1 | MiPrimerEsquema - Schema x

 Name: Specify the name of the schema here. You can use any combination of ANSI letters, numbers and the underscore character for names that don't

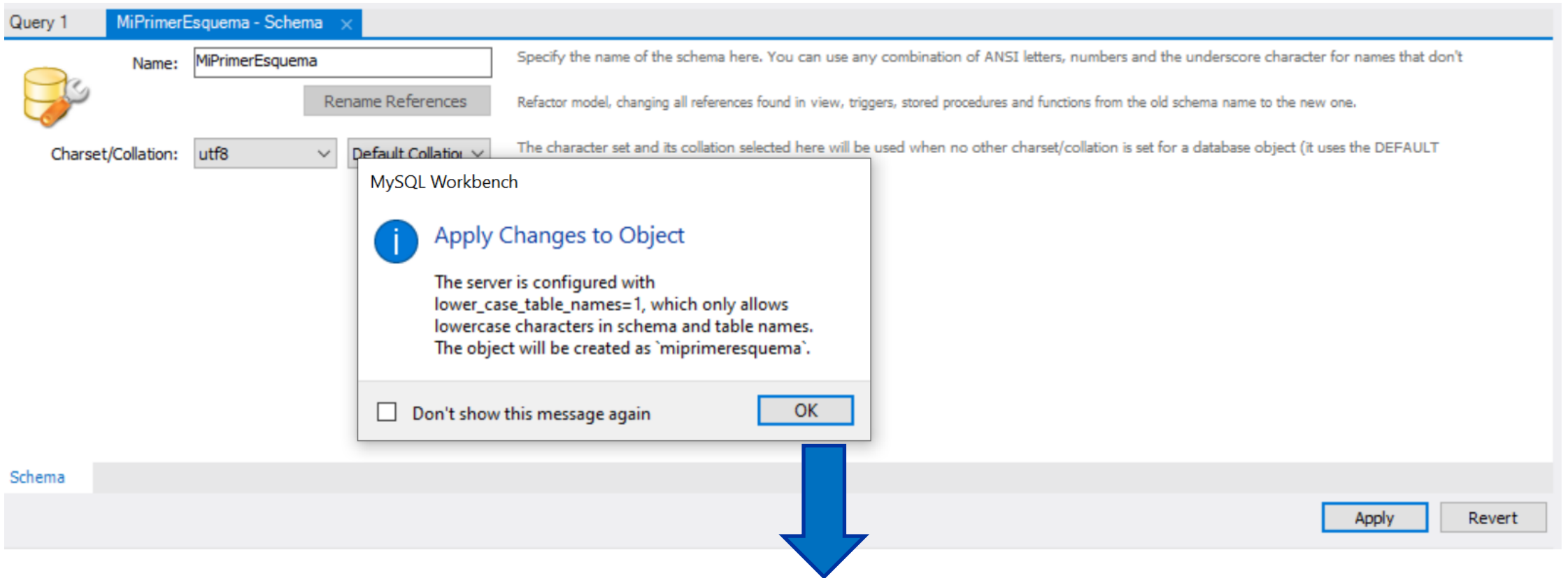
 Refactor model, changing all references found in view, triggers, stored procedures and functions from the old schema name to the new one.

Charset/Collation: The character set and its collation selected here will be used when no other charset/collation is set for a database object (it uses the DEFAULT)

Schema

Creación de un esquema en MySQL

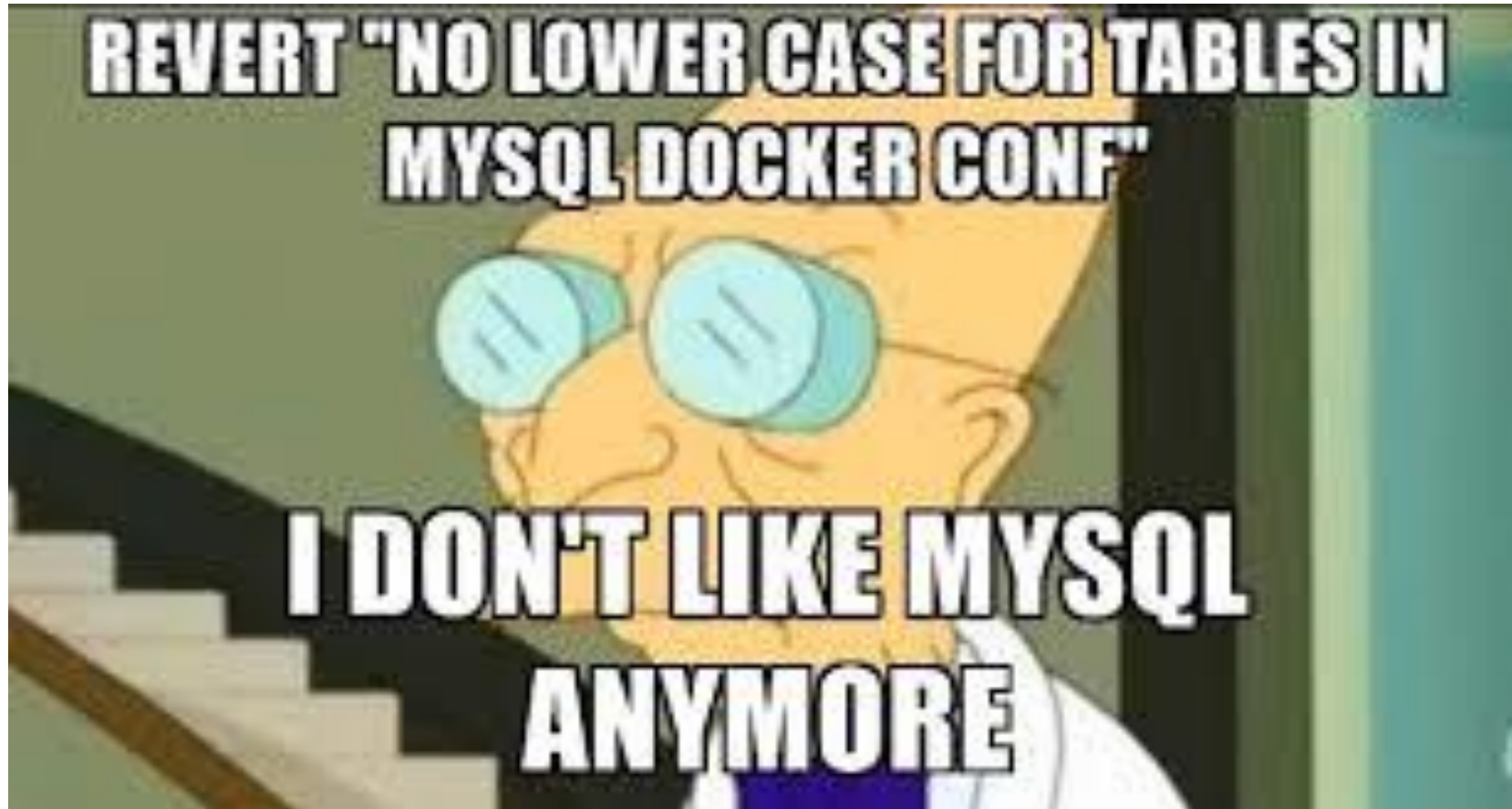
Usando MySQL Workbench



Integridad de los datos configurada desde la instalación del SGBD y MySQL Workbench!!

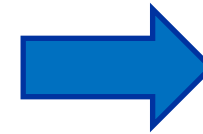
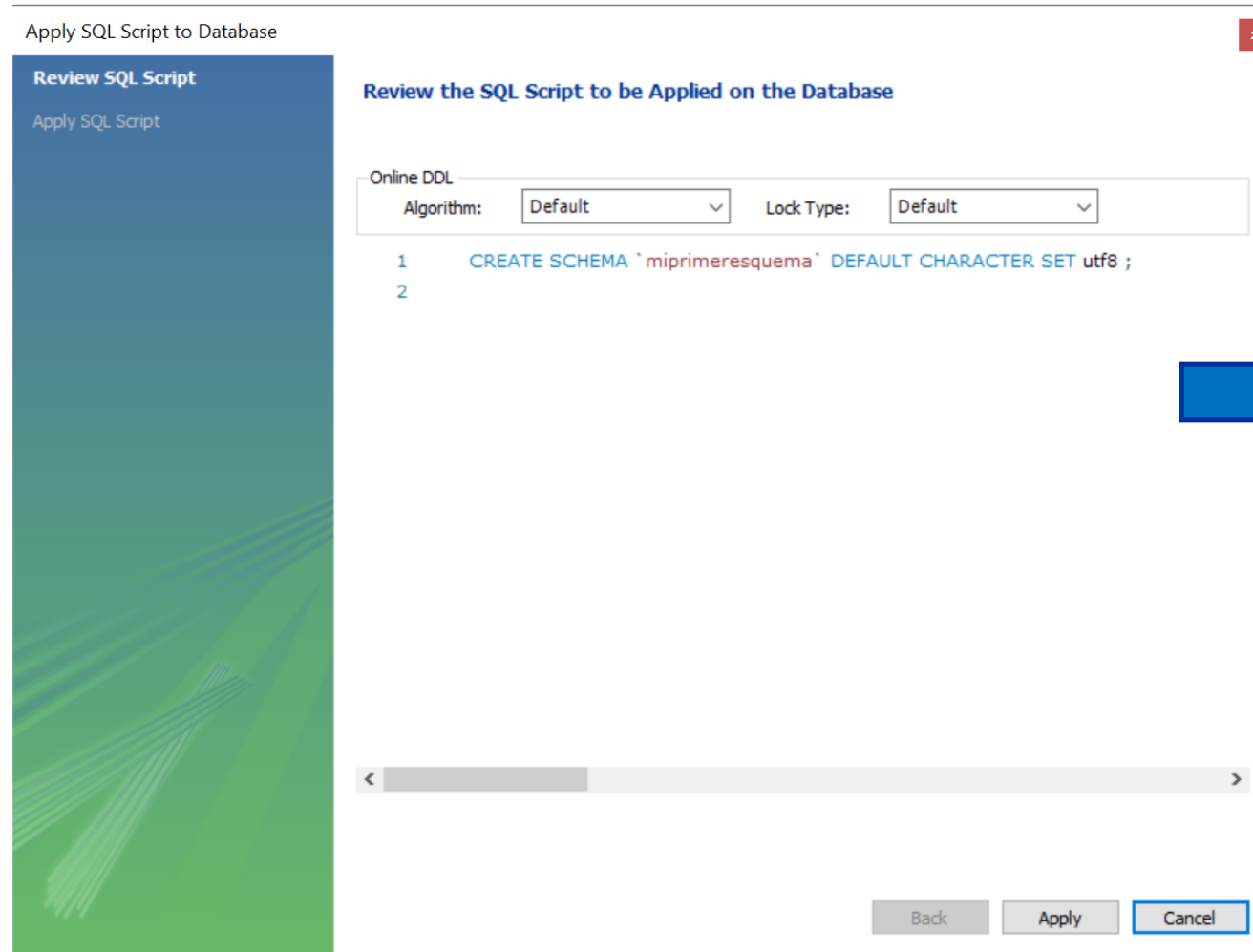
Creación de un esquema en MySQL

Usando MySQL Workbench



Creación de un esquema en MySQL

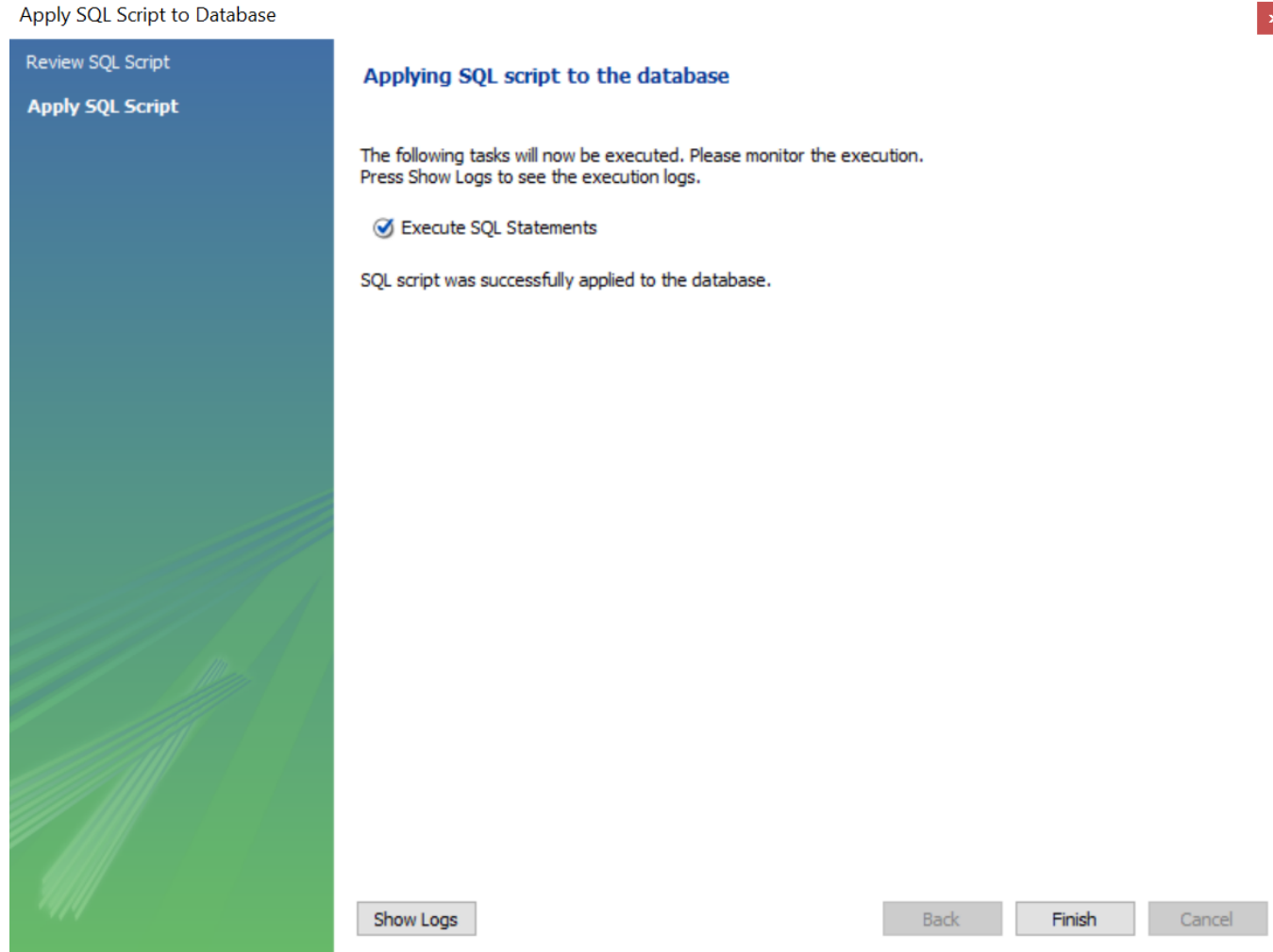
Usando MySQL Workbench



Se genera el código de SQL apropiado.

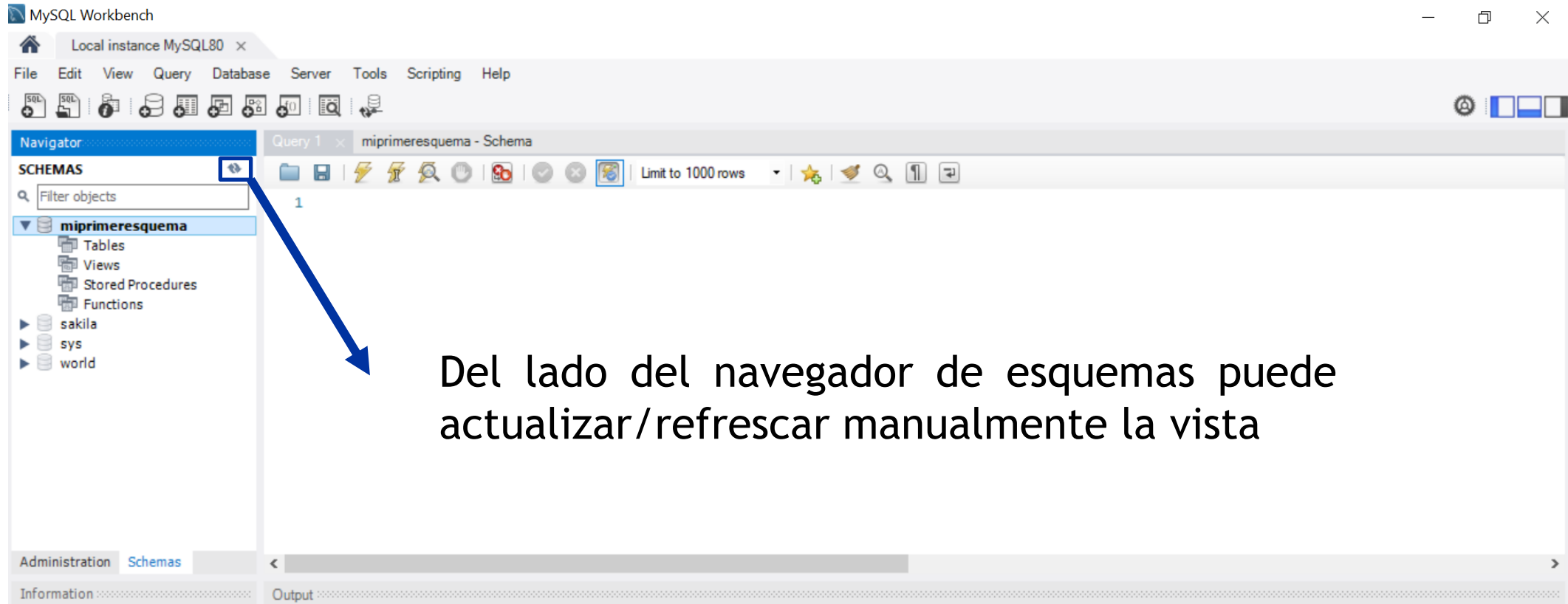
Creación de un esquema en MySQL

Usando MySQL Workbench



Creación de un esquema en MySQL

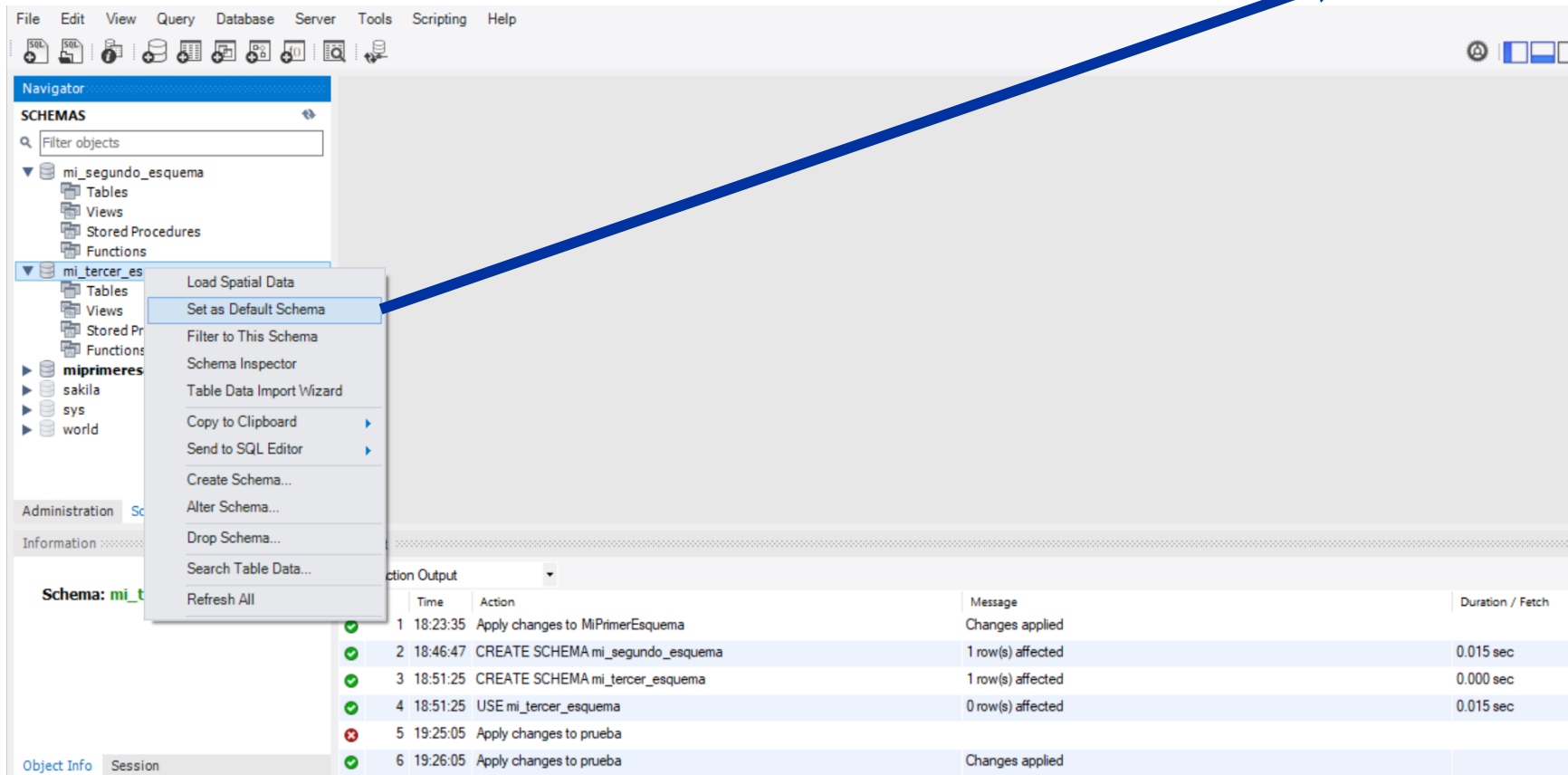
Usando MySQL Workbench



Creación de un esquema en MySQL

Usando MySQL Workbench

Cuando existe mas de un esquema puede seleccionar cual será el default para el SGBD



Creación de un esquema en MySQL

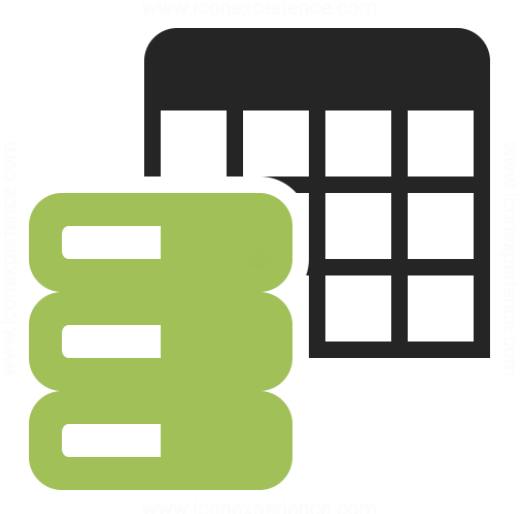
Usando código de SQL

La forma de crear un esquema en SQL dentro del motor de datos de MySQL es:

```
CREATE SCHEMA mi_segundo_esquema;
```

Palabras reservadas
de MySQL

Nombre de la
base de datos

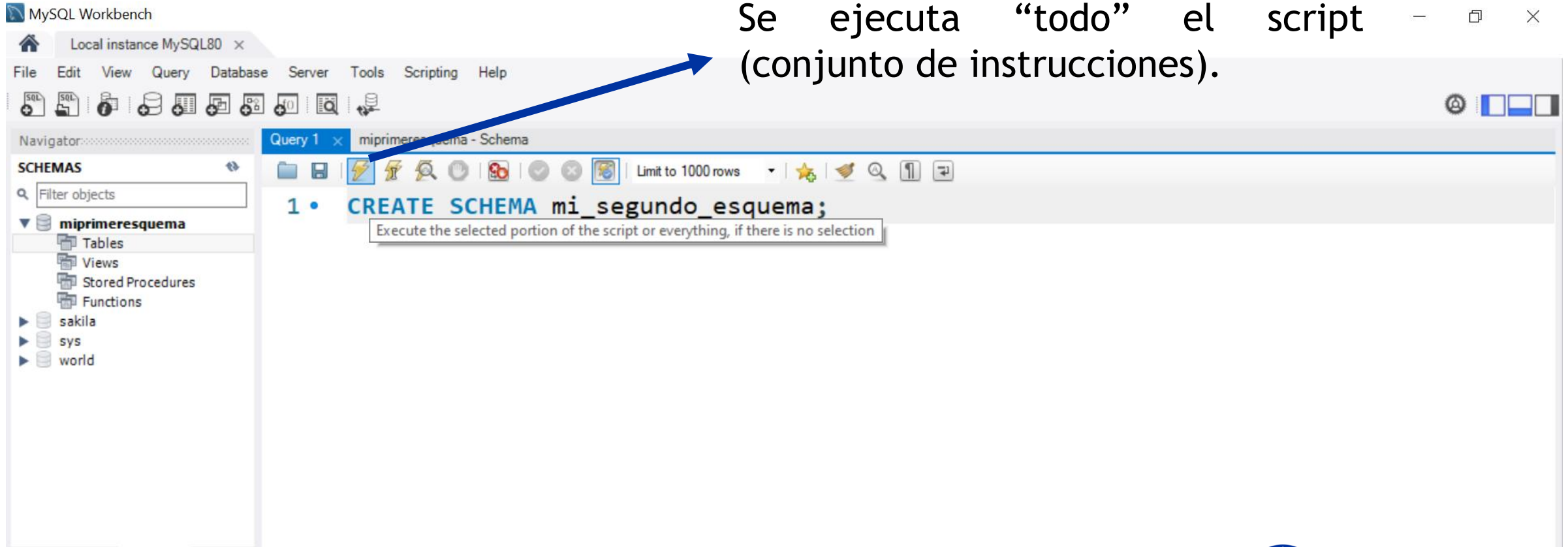


Nota: todas las instrucciones de SQL en la mayoría de motores de SGBD terminan las instrucciones con un punto y coma.

Creación de un esquema en MySQL

Usando código de SQL

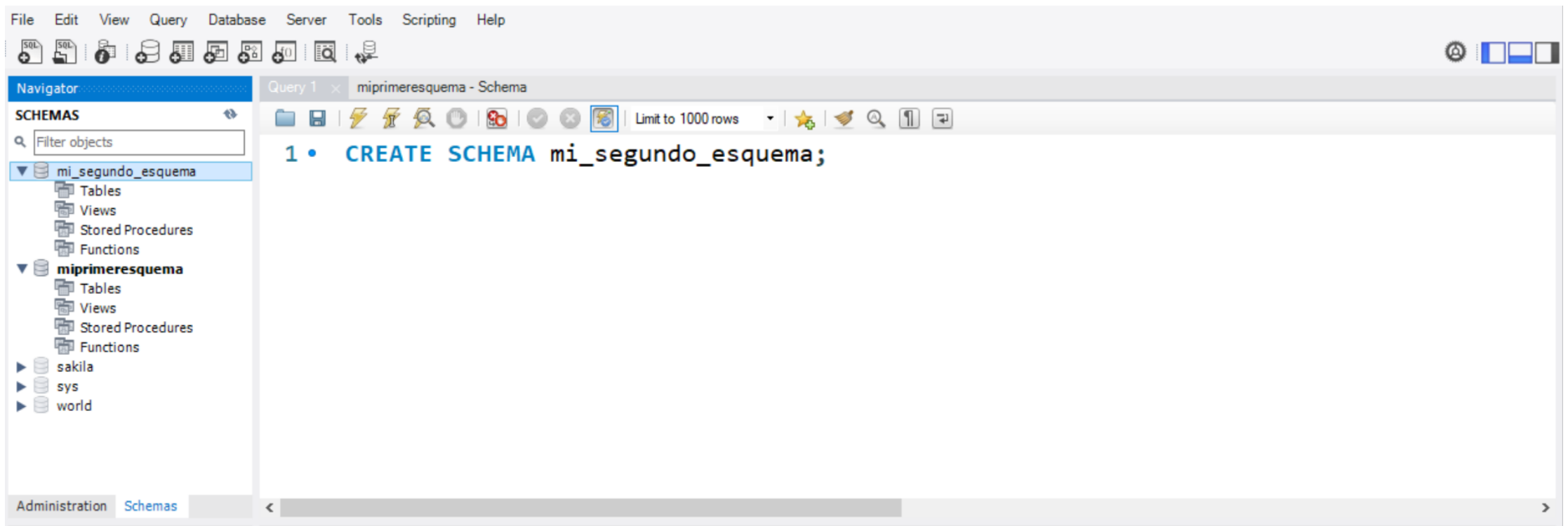
Se ejecuta “todo” el script (conjunto de instrucciones).



The screenshot displays the MySQL Workbench interface. On the left, the 'Navigator' pane shows a tree of schemas: 'miprimeresquema' (expanded), 'sakila', 'sys', and 'world'. Under 'miprimeresquema', there are sub-items for 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main window, titled 'Query 1 x miprimeresquema - Schema', contains a single SQL statement: `1 • CREATE SCHEMA mi_segundo_esquema;`. A blue arrow points from the text 'Se ejecuta “todo” el script (conjunto de instrucciones).' to the lightning bolt icon in the toolbar, which is used to execute the script. A tooltip for this icon reads: 'Execute the selected portion of the script or everything, if there is no selection'.

Creación de un esquema en MySQL

Usando código de SQL



Creación de un esquema en MySQL

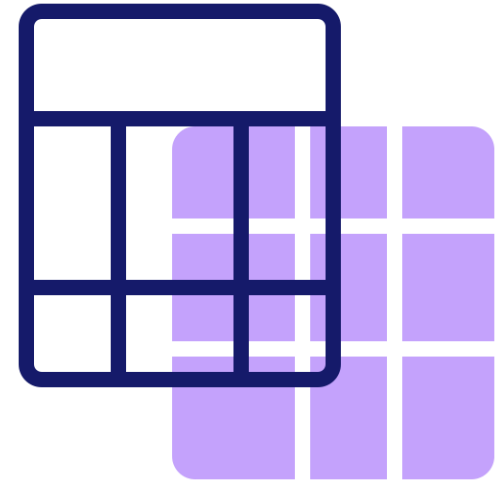
Usando código de SQL

Para usar el esquema o colocarnos dentro de él, tenemos que usar la siguiente instrucción;

```
USE mi_segundo_esquema;
```

Palabra
reservada
de MySQL

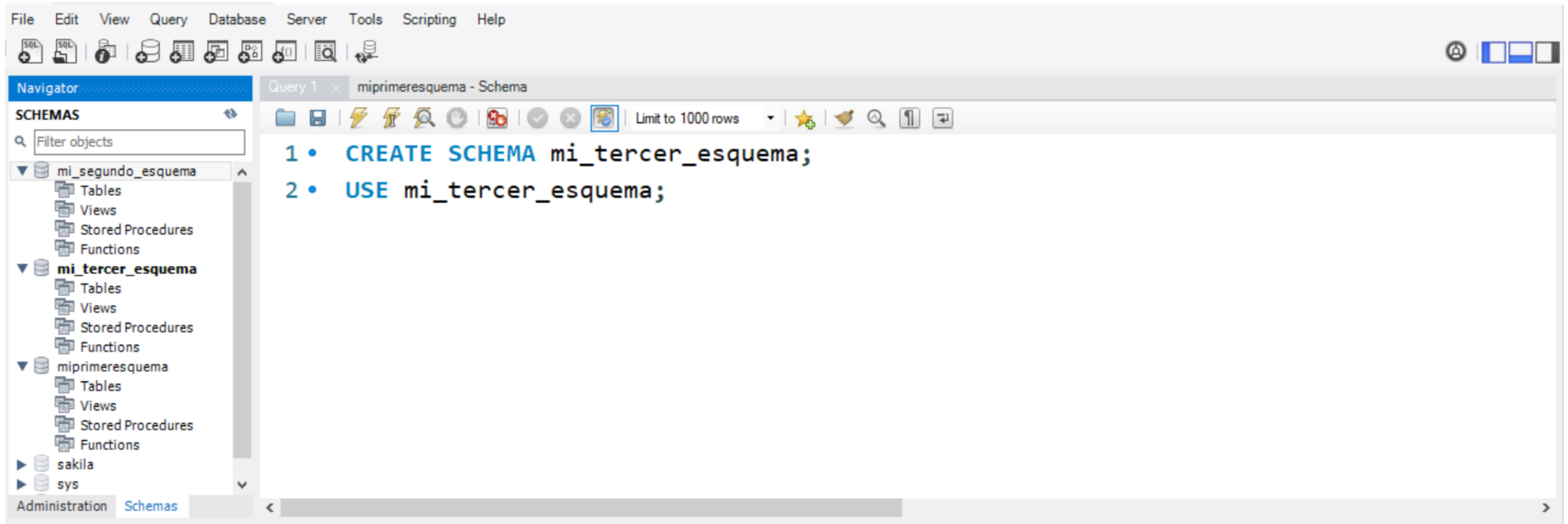
Nombre de la
base de datos



Nota: todas las instrucciones de SQL en la mayoría de motores de SGBD terminan las instrucciones con un punto y coma.

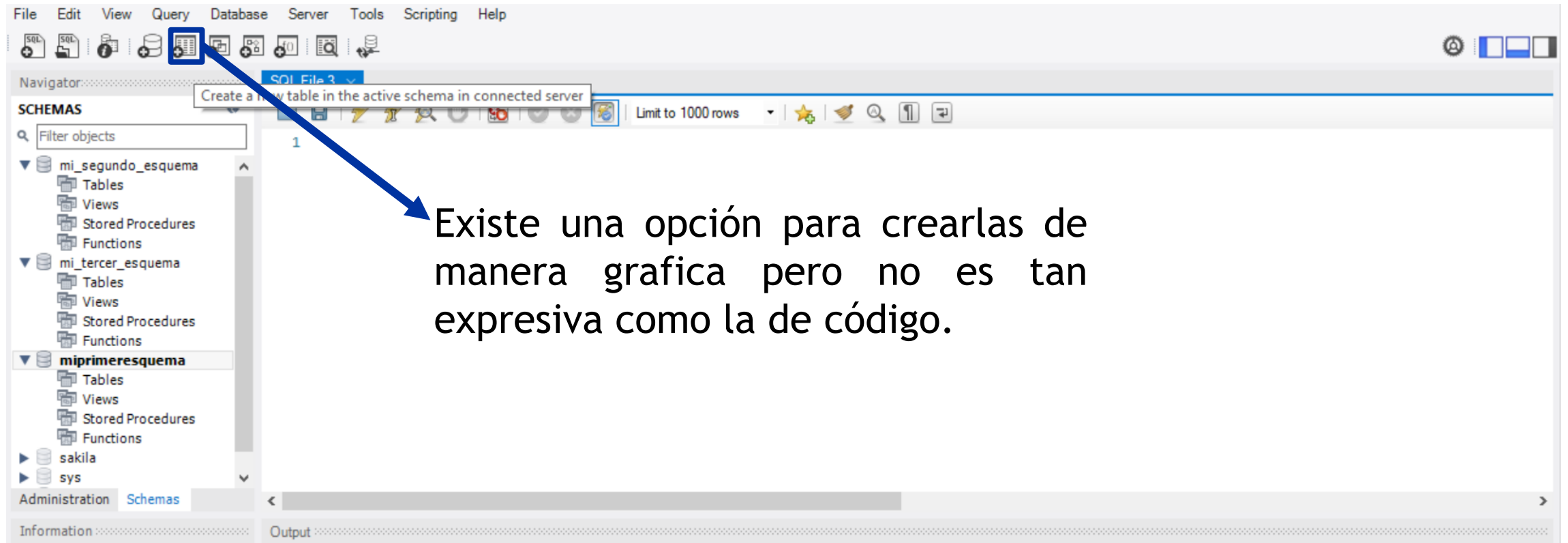
Creación de un esquema en MySQL

Usando código de SQL



Creación de una tabla en MySQL

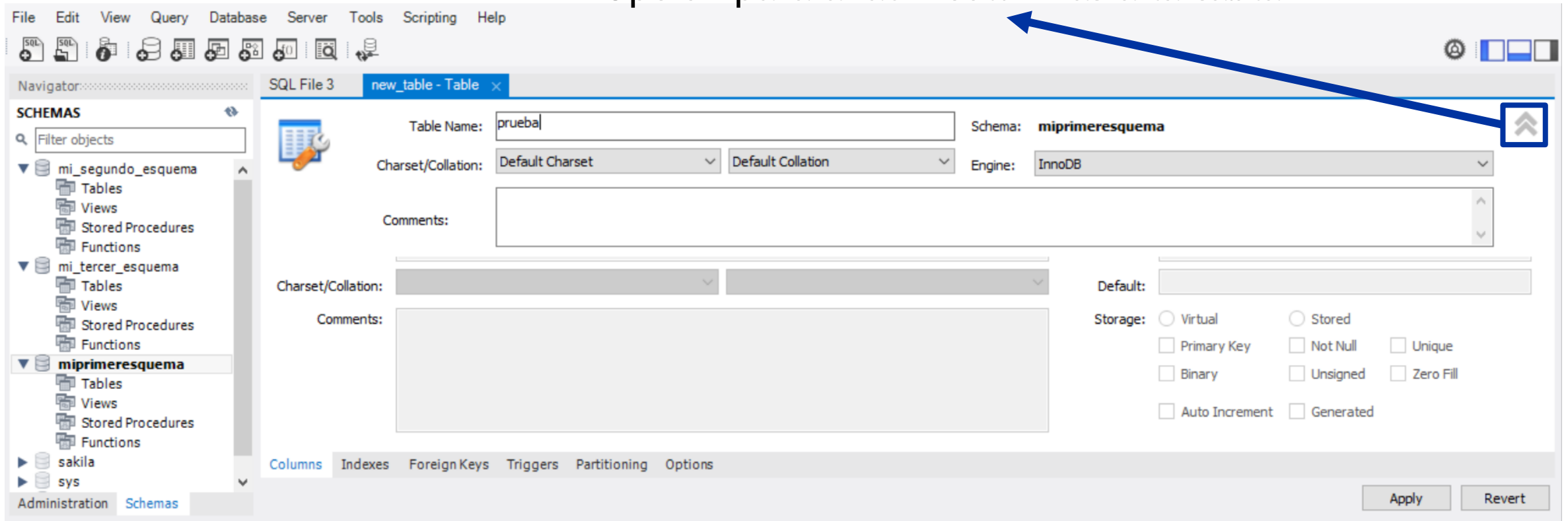
Usando MySQL Workbench



Creación de una tabla en MySQL

Usando MySQL Workbench

Opción para añadir columnas a la tabla



Creación de una tabla en MySQL

Usando MySQL Workbench

The screenshot displays the MySQL Workbench interface for creating a new table. The main window is titled "prueba - Table" and shows the following details:

- Table Name:** prueba
- Schema:** miprimeresquema

The table structure is defined as follows:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idprueba	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nombre	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

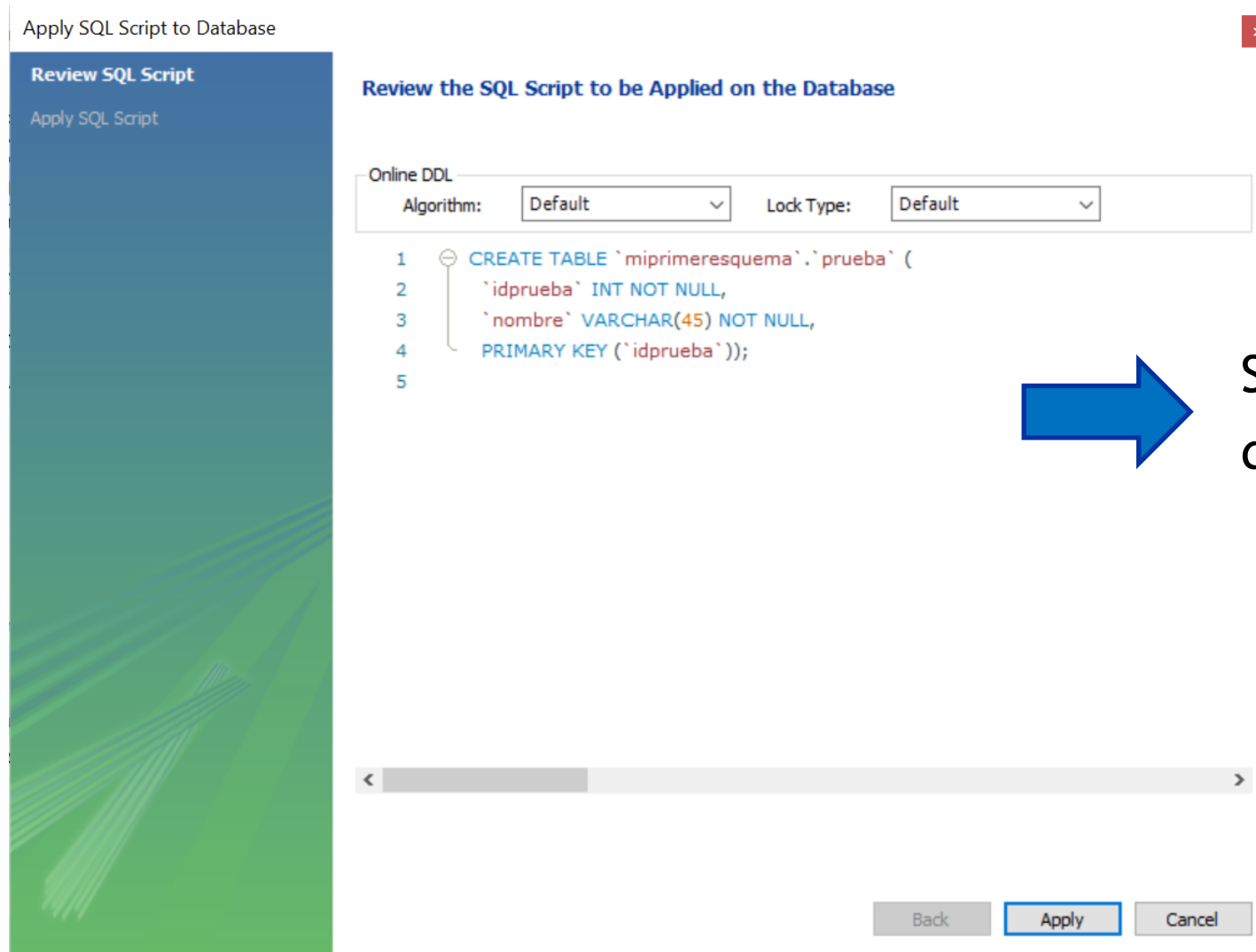
The right-hand configuration pane for the selected column "idprueba" shows the following settings:

- Column Name:** idprueba
- Data Type:** INT
- Charset/Collation:** Default Charset / Default Collation
- Comments:** (Empty text area)
- Storage:** ☐ Virtual ☐ Stored
- Primary Key:** ☒ (Other options: ☐ Binary, ☐ Auto Increment)
- Not Null:** ☒ (Other options: ☐ Unsigned, ☐ Generated)
- Unique:** ☐ (Other options: ☐ Zero Fill)

At the bottom of the interface, there are tabs for "Columns", "Indexes", "Foreign Keys", "Triggers", "Partitioning", and "Options". The "Columns" tab is currently selected. Buttons for "Apply" and "Revert" are located at the bottom right.

Creación de una tabla en MySQL

Usando MySQL Workbench



Se genera el código de SQL apropiado.

Creación de una tabla en MySQL

Usando MySQL Workbench

Apply SQL Script to Database



Applying SQL script to the database

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

☒ Execute SQL Statements

SQL script was successfully applied to the database.

Show Logs

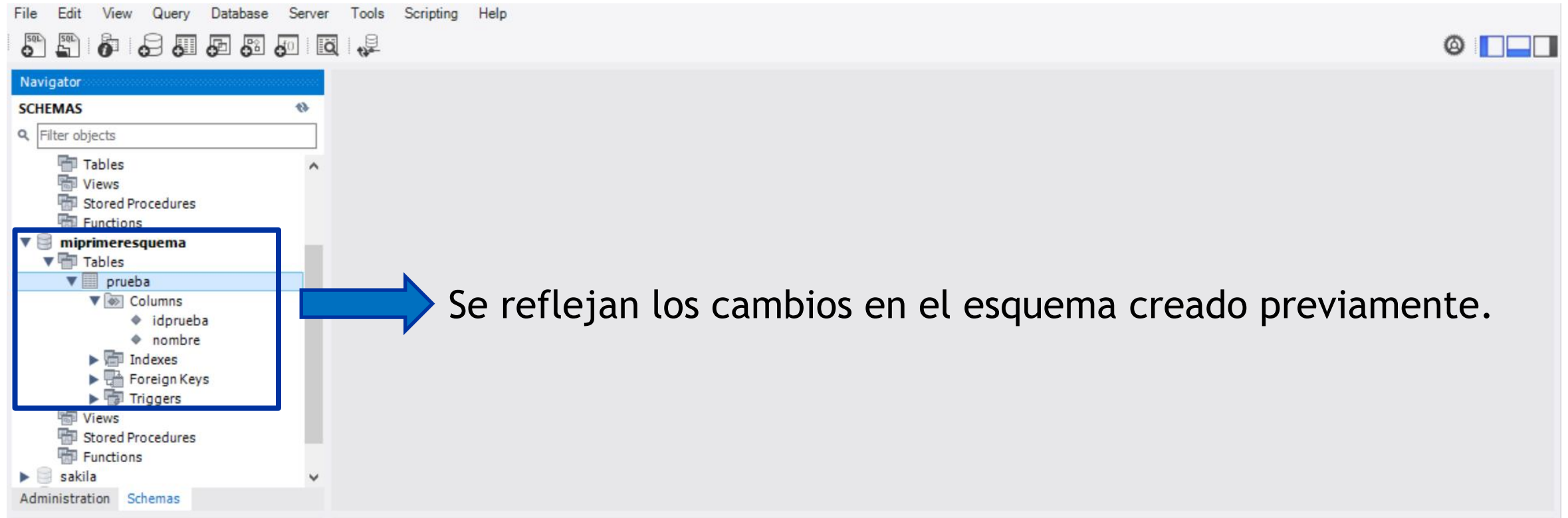
Back

Finish

Cancel

Creación de una tabla en MySQL

Usando MySQL Workbench

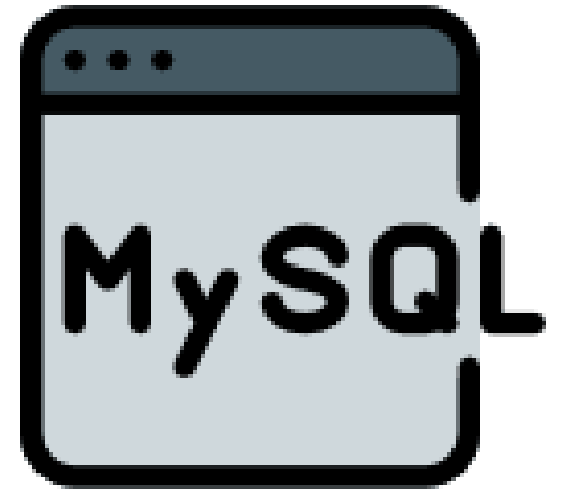


Creación de una tabla en MySQL

Usando código de SQL

Para crear tablas en MySQL, necesitamos tener la siguientes definiciones:

```
CREATE TABLE IF NOT EXISTS nombre_tabla  
(definición de la tabla,  
definición de columnas,  
tipos de columnas  
);
```



Creación de una tabla en MySQL

Usando código de SQL

Para crear tablas en MySQL, necesitamos tener la siguientes definiciones de tabla:

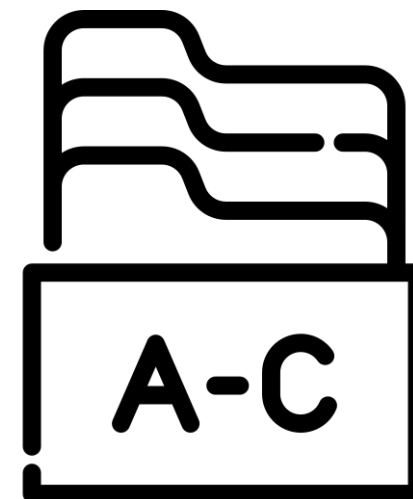
```
| [CONSTRAINT [nombre]] PRIMARY KEY [index_tipo] (index_col_nombre,...)
    [index_option] ...
| {INDEX|KEY} [index_name] [index_tipo] (index_col_nombre,...)
    [index_option] ...
| [CONSTRAINT [nombre]] UNIQUE [INDEX|KEY]
    [index_name] [index_tipo] (index_col_nombre,...)
    [index_option] ...
| {FULLTEXT|SPATIAL} [INDEX|KEY] [index_name] (index_col_nombre,...)
    [index_opción] ...
| [CONSTRAINT [nombre]] FOREIGN KEY
    [index_nombre] (index_col_nombre,...) reference_definition
| CHECK (expr)
```



Creación de una tabla en MySQL

Detalles de los elementos **anteriores**:

- **PRIMARY KEY**: Solo se puede crear una llave primaria por tabla, es la clave primaria que identifica de manera única cada registro/fila de la tabla. Por ejemplo el documento de identidad de una persona.
- **INDEX|KEY**: Ambas son sinónimos, puede haber una o varias. Establecen los índices de la tabla con los cuales se pueden agilizar las búsquedas en la base de datos. De esta manera se evita la búsqueda de un parámetro por cada columna de la tabla. Es como un índice de un libro con el que nos evitamos recorrer cada página.



Creación de una tabla en MySQL

Detalles de los elementos **anteriores**:

- **UNIQUE:** Es una restricción por la cual el valor de dicha columna debe ser único y diferente al del valor de dicha columna de resto de registros. Por ejemplo se suele usar con las columnas declaradas como llave primaria.
- **FULLTEXT:** Es un índice que solo funciona con las columnas con formato char, varchar, text y con almacenamiento [MyISAM](#). Este índice facilita las grandes búsquedas sobre texto y realiza automáticamente comparaciones de texto sobre una cadena dada.

Creación de una tabla en MySQL

Detalles de los elementos **anteriores**:

- **FOREIGN KEY**: llave secundaria, es un índice por el cual podemos relacionar dos tablas. Este valor debe existir en ambas tablas, por ejemplo el código postal de la tabla 'usuarios' y la tabla 'población'.



Creación de una tabla en MySQL

Usando código de SQL

Para crear tablas en MySQL, necesitamos tener la siguientes definiciones de columnas:

```
tipo_dato [NOT NULL | NULL] [DEFAULT valor_defecto]  
[AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]  
[COMMENT 'string']  
[COLUMN_FORMAT {FIXED|DYNAMIC|DEFAULT}]  
[STORAGE {DISK|MEMORY|DEFAULT}]  
[reference_definition]
```



MySQL

Creación de una tabla en MySQL

Usando código de SQL

Detalles de los elementos **anteriores**:

- **NOT NULL | NULL**: Establecemos si el valor de la columna debe rellenarse obligatoriamente o no.
- **AUTO_INCREMENT**: Establece un valor inicial para un incremento posterior con la inserción de cada nueva fila en la tabla.
- **COMMENT**: Comentario a modo informativo de la columna.
- **COLUMN_FORMAT**: Establece la ocupación de la columna, FIXED para un valor fijo, DYNAMIC para un valor variable y DEFAULT para una ocupación por defecto.
- **STORAGE**: Posibilidad de almacenamiento en memoria o disco.

Creación de una tabla en MySQL

Usando código de SQL

Para crear tablas en MySQL, necesitamos tener la siguientes definiciones de tipos de columnas (1):

```
BIT[(length)]  
| TINYINT[(length)] [UNSIGNED] [ZEROFILL]  
| SMALLINT[(length)] [UNSIGNED] [ZEROFILL]  
| MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]  
| INT[(length)] [UNSIGNED] [ZEROFILL]  
| INTEGER[(length)] [UNSIGNED] [ZEROFILL]  
| BIGINT[(length)] [UNSIGNED] [ZEROFILL]
```



Creación de una tabla en MySQL

Usando código de SQL

Para crear tablas en MySQL, necesitamos tener la siguientes definiciones de tipos de columnas (2):

- | REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
- | DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
- | FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
- | DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
- | NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
- | DATE
- | TIME



Creación de una tabla en MySQL

Usando código de SQL

Para crear tablas en MySQL, necesitamos tener la siguientes definiciones de tipos de columnas (2):

| **TIMESTAMP**

| **DATETIME**

| **YEAR**

| **CHAR(length) [BINARY]**

[CHARACTER SET charset_name] [COLLATE collation_name]

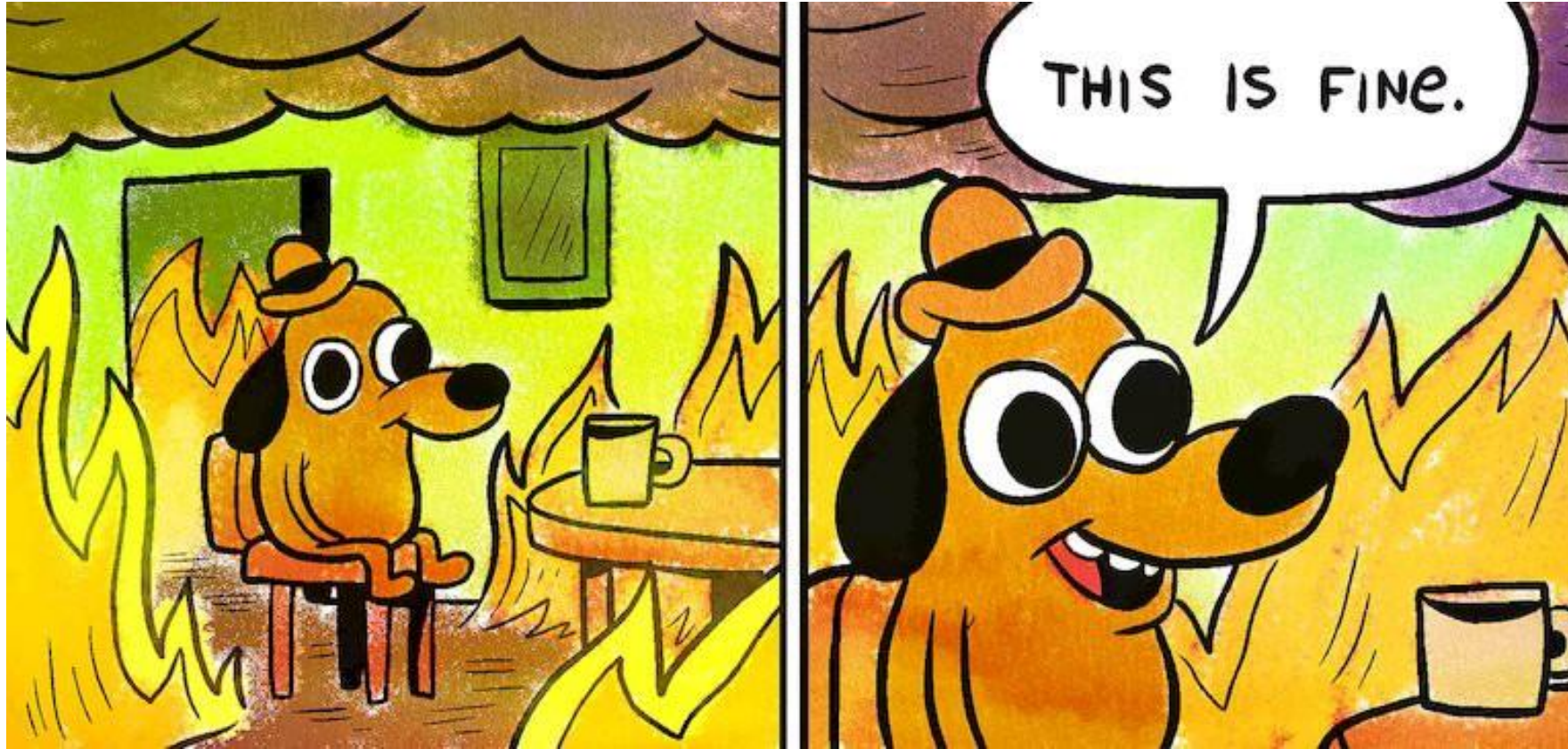
| **VARCHAR(length) [BINARY]**

[CHARACTER SET charset_name] [COLLATE collation_name]

...



Creación de una tabla en MySQL

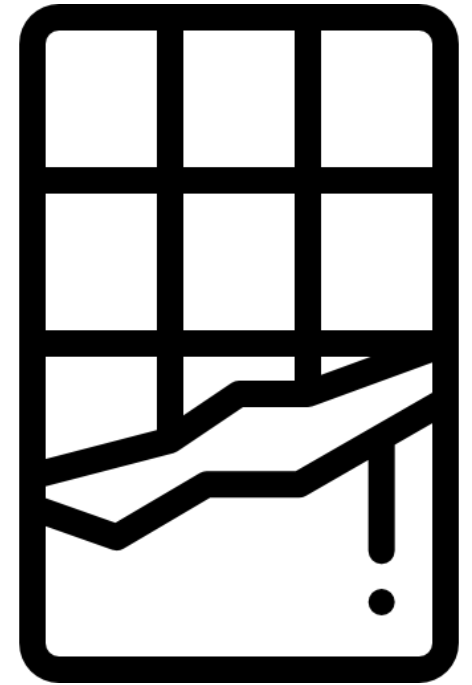


Creación de una tabla en MySQL

Usando código de SQL

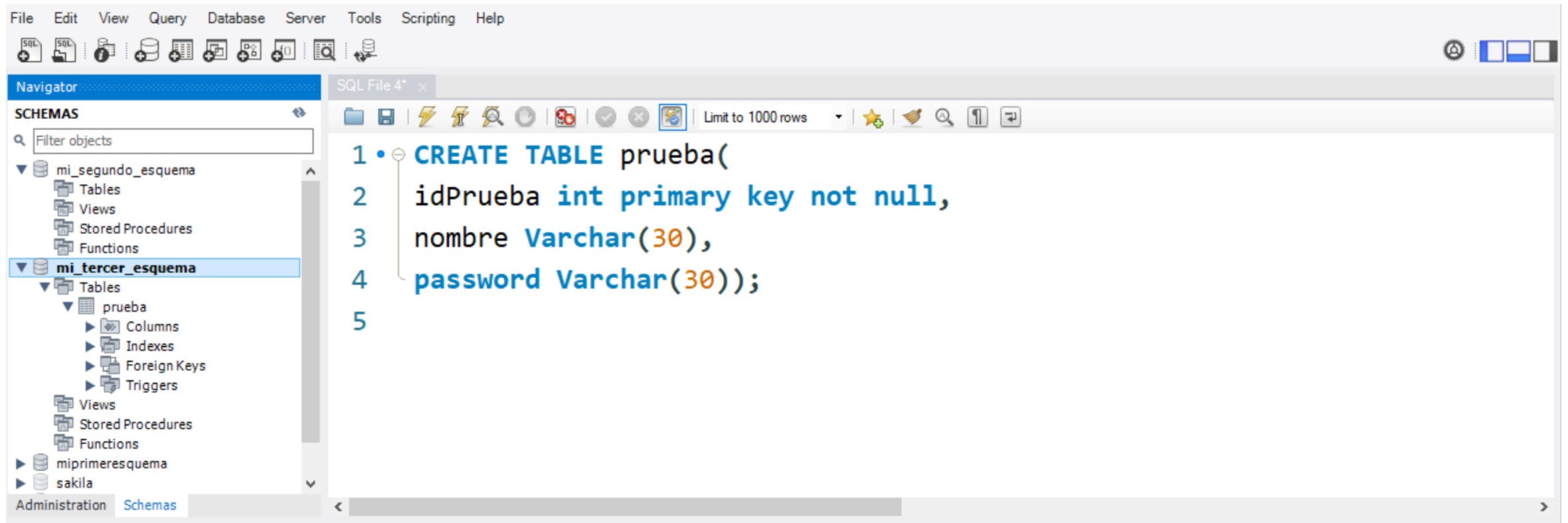
Tomando en cuenta lo anterior, para crear una tabla en MySQL, podemos hacer lo siguiente:

```
CREATE TABLE prueba(  
  idPrueba int primary key not null,  
  nombre Varchar(30),  
  password Varchar(30));
```



Creación de una tabla en MySQL

Usando código de SQL



Me: *clears off table after dinner*

Database team:



Que sigue.. Seguir practicando y ver inserción y eliminación de información.

Referencias

- Sommerville, I., Software Engineering, 10th Edition, Pearson, 2016, IN, 1292096144, 9781292096148.
- Connolly Thomas M, Database systems : a practical approach to design, implementation and management, 5thed., London : Addison-Wesley, 2010, 9780321523068.
- Perez, C., MySQL para windows y Linux, España, Alfaomega, 2004.
- <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

Gracias!
Preguntas...



Dr. Esteban Castillo Juarez

Google academics:

<https://scholar.google.com/citations?user=JfZpVO8AAAAJ&hl=en>

<https://dblp.uni-trier.de/pers/hd/c/Castillo:Esteban>