

Lección 1: Introducción a SQL



Tiempo de ejecución: 4 horas

Planteamiento de la sesión



Materiales

- [GuiaInstalacion_MySql_Community_8.0.19.pdf](#)
- [¿Qué es una base de datos? - Explicación de las bases de datos en la nube - AWS](#)
- [Conceptos básicos sobre bases de datos - Soporte técnico de Microsoft](#)
- [Qué es MySQL: Características y ventajas | OpenWebinars](#)
- [¿Qué Es MySQL? Explicación Detallada Para Principiantes](#)
- [Qué es una base de datos relacional | Oracle Colombia](#)
- [SQL | DDL, DQL, DML, DCL and TCL Commands - GeeksforGeeks](#)
- [SQL Police Department](#)
- [Practice SQL](#)
- [GitHub - MisaoDev/SQL-Ejercicios: Ejercicios SQL para usar con Sakila](#)
- [Los 5 tipos de datos en MySQL y cómo utilizarlos](#)
- [Sakila Sample Database](#)
- [MySQL :: Sakila Sample Database :: 4 Installation](#)
- [MySQL :: Sakila Sample Database :: 5 Structure](#)
- [SALIKA EER Diagram.png](#)
- [w3schools.sql](#)

Dentro de lo visto de Pandas, se habló de que los dataset pueden provenir de diferentes fuentes, entre ellas, bases de datos, así entonces, es hora de comenzar con la revisión de conceptos, instalación y configuración de las primeras bases de datos.

Inicialmente la explicación se centrará en los conceptos propios sobre bases de datos relaciones, se trabajará con un servidor local llamado community server y un cliente de bases de datos, que será la interfaz de conexión entre nosotros como usuarios y las bases de datos almacenadas en el servidor, llamado workbench.

Posterior a conocer los conceptos necesarios y la lógica detrás de las consultas, es hora de integrar las bases de datos con un proyecto de Python, creando cadenas de conexión y operaciones como consultas o inserciones desde el código mismo.

Conceptos base

Para iniciar con el proceso de instalación de las herramientas y la utilización de las mismas, es necesario tener claridad algunos conceptos como base de datos o mysql que se definen a continuación:

Qué es una base de datos

Una base de datos es una recopilación de datos sistemática (datos estructurados) y almacenada electrónicamente. Puede contener cualquier tipo de datos, incluidos palabras, números, imágenes, vídeos y archivos, las bases de datos pueden almacenar información sobre personas, productos, pedidos u otras cosas. Puede usar un software denominado sistema de administración de bases de datos (DBMS) para almacenar, recuperar y editar datos. En los sistemas informáticos, la palabra base de datos también puede referirse a cualquier DBMS, al sistema de base de datos o a una aplicación asociada con la base de datos.

Una base de datos está compuesta por una o más tablas que pueden estar relacionadas entre sí, esto quiere decir que si mi aplicación requiere guardar información de un usuario y de sus pedidos, no es necesario crear diferentes bases de datos, sino crear tablas con los datos necesarios para diferenciar cada uno de los objetos o elementos o actores que están involucrados en la aplicación.

¿Por qué son importantes las bases de datos?

Una base de datos de alto rendimiento es crucial para cualquier organización. Las bases de datos sustentan las operaciones internas de las empresas y almacenan las interacciones con clientes y proveedores. Además, albergan información administrativa y datos más especializados, como modelos de ingeniería o económicos. Algunos ejemplos son los sistemas de bibliotecas digitales, los sistemas de reserva de viajes y los sistemas de inventario. A continuación se exponen algunas razones por las que las bases de datos son esenciales.

Escalado eficiente

Las aplicaciones de bases de datos pueden administrar grandes cantidades de datos, que pueden escalar a millones, miles de millones y más. Es imposible almacenar esta cantidad de datos digitales sin una base de datos.

Integridad de los datos

Las bases de datos con frecuencia tienen reglas y condiciones incorporadas para mantener la coherencia de datos.

Seguridad de los datos

Las bases de datos son compatibles con los requisitos de privacidad y conformidad asociados a cualquier dato. Por ejemplo, para acceder a la base de datos, los usuarios deben iniciar sesión. Los diferentes usuarios también pueden tener diferentes niveles de acceso, como el de solo lectura.



Análisis de datos

Los sistemas de software modernos utilizan bases de datos para analizar los datos. Estos sistemas son capaces de identificar tendencias y patrones, así como de realizar predicciones. Los análisis de datos ayudan a las organizaciones a tomar decisiones empresariales con confianza.

Tablas

Una tabla de base de datos es similar en apariencia a una hoja de cálculo en cuanto a que los datos se almacenan en filas y columnas. Por ende, es bastante fácil importar una hoja de cálculo en una tabla de base de datos. La principal diferencia entre almacenar los datos en una hoja de cálculo y almacenarlos en una base de datos es la forma en la que están organizados los datos.

Para aprovechar al máximo la flexibilidad de una base de datos, los datos deben organizarse en tablas para que no se produzcan redundancias. Por ejemplo, si quiere almacenar información sobre los empleados, cada empleado debe especificarse solo una vez en la tabla que está configurada para los datos de los empleados. Los datos sobre los productos se almacenarán en su propia tabla y los datos sobre las sucursales se almacenarán en otra tabla.

Cada fila de una tabla se denomina registro. En los registros se almacena información. Cada registro está formado por uno o varios campos. Los campos equivalen a las columnas de la tabla. Por ejemplo, puede tener una tabla llamada "Empleados" donde cada registro (fila) contiene información sobre un empleado distinto y cada campo (columna) contiene otro tipo de información como nombre, apellido, dirección, etc. Los campos deben designarse como un determinado tipo de datos, ya sea texto, fecha u hora, número o algún otro tipo.

MySQL

MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor.

RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional.

Ofrecido por Oracle. Los desarrolladores pueden descargar y usar MySQL sin pagar tarifas de licencia. Pueden instalar MySQL en diferentes sistemas operativos o servidores en la nube. MySQL es un sistema de base de datos popular para aplicaciones web.

Bases de datos relacional

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila en una tabla es un registro con una ID única, llamada clave. Las columnas de la tabla contienen los atributos de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.

Ejemplo de una base de datos relacional

Este es un ejemplo simple de dos tablas que una pequeña empresa puede usar para procesar pedidos de sus productos. La primera tabla es una tabla de información del cliente, por lo que cada registro incluye el nombre, la dirección, información de envío y facturación, el número de teléfono y otra información de contacto del cliente. Cada fragmento de información (cada atributo) está en su propia columna y la base de datos asigna una ID única (una clave) a cada fila. En la segunda tabla, una tabla de pedidos del cliente, cada registro incluye el ID del cliente que realizó el pedido, el producto solicitado, la cantidad, el tamaño y el color seleccionados, etc., pero no el nombre del cliente ni su información de contacto.

Estas dos tablas tienen una sola cosa en común: la columna de ID (la clave). Gracias a esa columna en común, la base de datos relacional puede establecer una relación entre las dos tablas. Entonces, cuando la aplicación de procesamiento de pedidos de la empresa envíe un pedido a la base de datos, la base de datos podrá examinar la tabla de pedidos del cliente, extraer la información correcta sobre el pedido de productos y usar el ID del cliente de esa tabla para buscar la información de facturación y envío del cliente en la tabla de información del cliente. A continuación, el almacén puede extraer el producto correcto, el cliente puede recibir la entrega del pedido a tiempo y la empresa puede obtener el pago.



Propiedades de ACID y sistema de gestión de bases de datos relacionales

Son cuatro las propiedades cruciales que definen las transacciones de las bases de datos relacionales: atomicidad, uniformidad, aislamiento y durabilidad. Estas suelen conocerse como ACID, su acrónimo en inglés.

- La **atomicidad** define todos los elementos que conforman una transacción completa de base de datos.
- La **uniformidad** define las reglas para mantener los puntos de datos en un estado correcto después de una transacción.
- El **aislamiento** impide que el efecto de una transacción sea visible a otros hasta que se establezca el compromiso, a fin de evitar confusiones.
- La **durabilidad** garantiza que los cambios en los datos se vuelvan permanentes cuando la transacción se haya fijado y hayamos llegado a un compromiso.

Instalación de las herramientas

Para el proceso de instalación, se han consultado dos guías adicional a la documentación que puede entregar la página oficial. [GuiaInstalacion_MySql_Community_8.0.19.pdf](#) y [MANUAL_MYSQL_FY.pdf](#)

La primera instalación que se explica es la del community server, que es la única obligatoria, lo que allí se instala es un servidor soportado por una comunidad de desarrolladores y de código abierto, en donde se almacenan los datos de la aplicación.

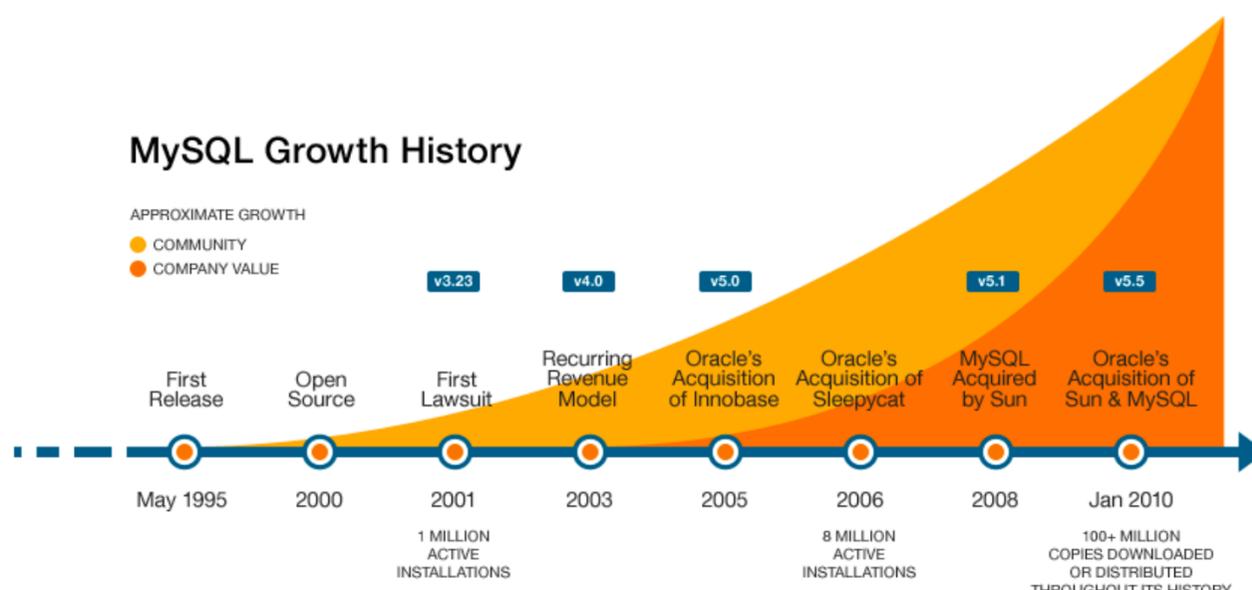
Workbench, es un cliente que suministra una interfaz gráfica para conectarse al servidor, local o remoto, de bases de datos relacional. Workbench es una herramienta que entrega más funcionalidades como la posibilidad de realizar ingeniería inversa y generar un diagrama entidad-relación, desde una base de datos y viceversa.

Características de MySQL

MySQL presenta algunas ventajas que lo hacen muy interesante para los desarrolladores. La más evidente es que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.

1. **Arquitectura Cliente y Servidor:** MySQL basa su funcionamiento en un modelo cliente y servidor. Es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros, por ejemplo.
2. **Compatibilidad con SQL:** SQL es un lenguaje generalizado dentro de la industria. Al ser un estándar MySQL ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a MySQL.
3. **Vistas:** Desde la versión 5.0 de MySQL se ofrece compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos SQL. En bases de datos de gran tamaño las vistas se hacen un recurso imprescindible.
4. **Procedimientos almacenados:** MySQL posee la característica de no procesar las tablas directamente sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.
5. **Desencadenantes:** MySQL permite además poder automatizar ciertas tareas dentro de nuestra base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros o optimizar su funcionalidad.
6. **Transacciones:** Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas. En caso por ejemplo de una falla de energía, cuando el monitor falla u ocurre algún otro inconveniente, el sistema opta por preservar la integridad de la base de datos resguardando la información.

Fuente: MySQL Growth History, consultado en febrero de 2024, disponible en [Qué es MySQL: Características y ventajas | OpenWebinars](#)



SQL

El lenguaje de consulta estructurado (SQL), como todos sabemos, es el lenguaje de base de datos mediante el cual podemos realizar ciertas operaciones en la base de datos existente, y también podemos usar este lenguaje para crear una base de datos. SQL utiliza ciertos comandos como CREATE, DROP, INSERT, etc. para realizar las tareas requeridas.

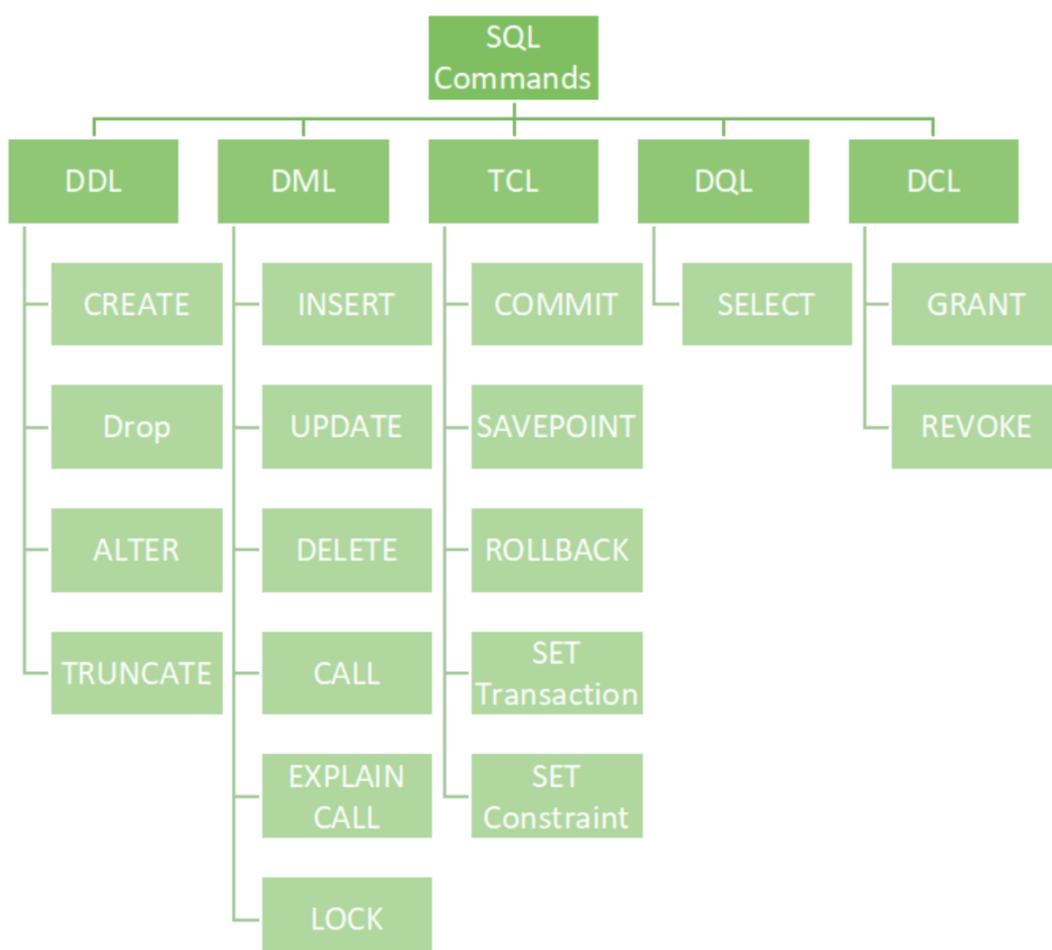
Los comandos SQL son como instrucciones para una tabla. Se utiliza para interactuar con la base de datos con algunas operaciones. También se utiliza para realizar tareas, funciones y consultas de datos específicas. SQL puede realizar varias tareas, como crear una tabla, agregar datos a las tablas, eliminar la tabla, modificarla y establecer permisos para los usuarios.

Estos comandos SQL se clasifican principalmente en cinco categorías:

- DDL: lenguaje de definición de datos
- DQL – Lenguaje de consulta de datos
- DML – Lenguaje de manipulación de datos
- DCL – Lenguaje de control de datos
- TCL – Lenguaje de control de transacciones

Fuente: SQL Commands, consultado en febrero de 2024, disponible en [SQL | DDL, DQL, DML, DCL and TCL Commands - GeeksforGeeks](#).

De los 5 grandes bloques se componen todo el lenguaje SQL, nos centraremos en 3 categorías



DDL (lenguaje de definición de datos)

DDL o lenguaje de definición de datos en realidad consta de comandos SQL que se pueden utilizar para definir el esquema de la base de datos. Simplemente se ocupa de las descripciones del esquema de la base de datos y se utiliza para crear y modificar la estructura de los objetos de la base de datos en la base de datos. DDL es un conjunto de comandos SQL que se utilizan para crear, modificar y eliminar estructuras de bases de datos, pero no datos. Estos comandos normalmente no los utiliza un usuario general, que debería acceder a la base de datos a través de una aplicación.

Lista de comandos DDL:

- **CREATE**: este comando se utiliza para crear la base de datos o sus objetos (como tabla, índice, función, vistas, procedimiento de almacenamiento y activadores).
- **DROP** : este comando se utiliza para eliminar objetos de la base de datos.
- **ALTER** : Se utiliza para alterar la estructura de la base de datos.
- **TRUNCATE** : Esto se utiliza para eliminar todos los registros de una tabla, incluidos todos los espacios asignados para los registros.
- **COMENTARIO** : Se utiliza para agregar comentarios al diccionario de datos.
- **RENAME** : Esto se utiliza para cambiar el nombre de un objeto existente en la base de datos.

DQL (lenguaje de consulta de datos)

Las declaraciones DQL se utilizan para realizar consultas sobre los datos dentro de los objetos del esquema. El propósito del comando DQL es obtener alguna relación de esquema basada en la consulta que se le pasa. Podemos definir DQL de la siguiente manera: es un componente de la declaración SQL que permite obtener datos de la base de datos e imponerles orden. Incluye la declaración SELECT. Este comando permite sacar los datos de la base de datos para realizar operaciones con ellos. Cuando se activa un SELECT contra una tabla o tablas, el resultado se compila en una tabla temporal adicional, que el programa muestra o tal vez recibe, es decir, una interfaz.

Lista de DQL:

SELECT : Se utiliza para recuperar datos de la base de datos.

DML (lenguaje de manipulación de datos)

Los comandos SQL que se ocupan de la manipulación de datos presentes en la base de datos pertenecen a DML o Lenguaje de manipulación de datos y esto incluye la mayoría de las declaraciones SQL. Es el componente de la declaración SQL que controla el acceso a los datos y a la base de datos. Básicamente, las declaraciones DCL se agrupan con las declaraciones DML.

Lista de comandos DML:

- INSERT : Se utiliza para insertar datos en una tabla.
- UPDATE : Se utiliza para actualizar datos existentes dentro de una tabla.
- DELETE : Se utiliza para eliminar registros de una tabla de base de datos.
- LOCK ; Concurrencia de control de tablas.

Diferencias entre SQL y MySQL

El lenguaje de consulta estructurada (SQL) es un lenguaje estándar para la creación y manipulación de bases de datos. MySQL es un programa de base de datos relacional que utiliza consultas SQL. Si bien los comandos SQL están definidos por estándares internacionales, el software MySQL se somete a actualizaciones y mejoras continuas.

Instrucciones SQL

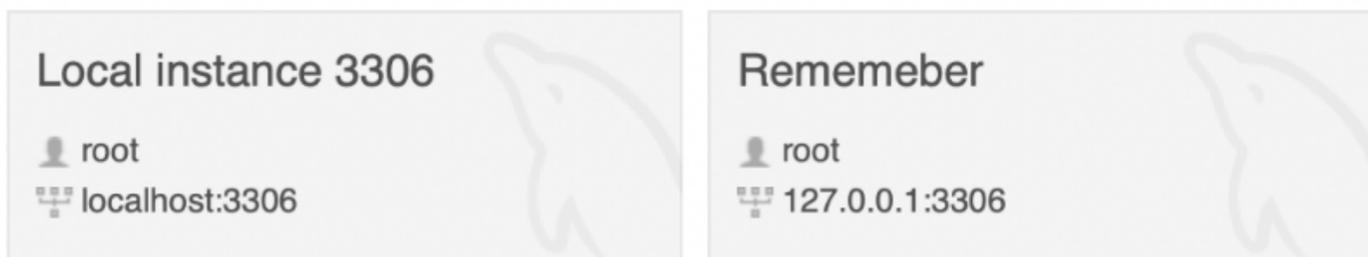
Se ha mencionado, dentro de las 3 categorías a trabajar, que cada una de ellas tiene asociado una serie de comandos o instrucciones, a continuación se revisarán cada una de las instrucciones populares con el fin de interactuar, desde una hoja de comandos de SQL en workbench, con una base de datos almacenada en el servidor.

Conectando con el servidor

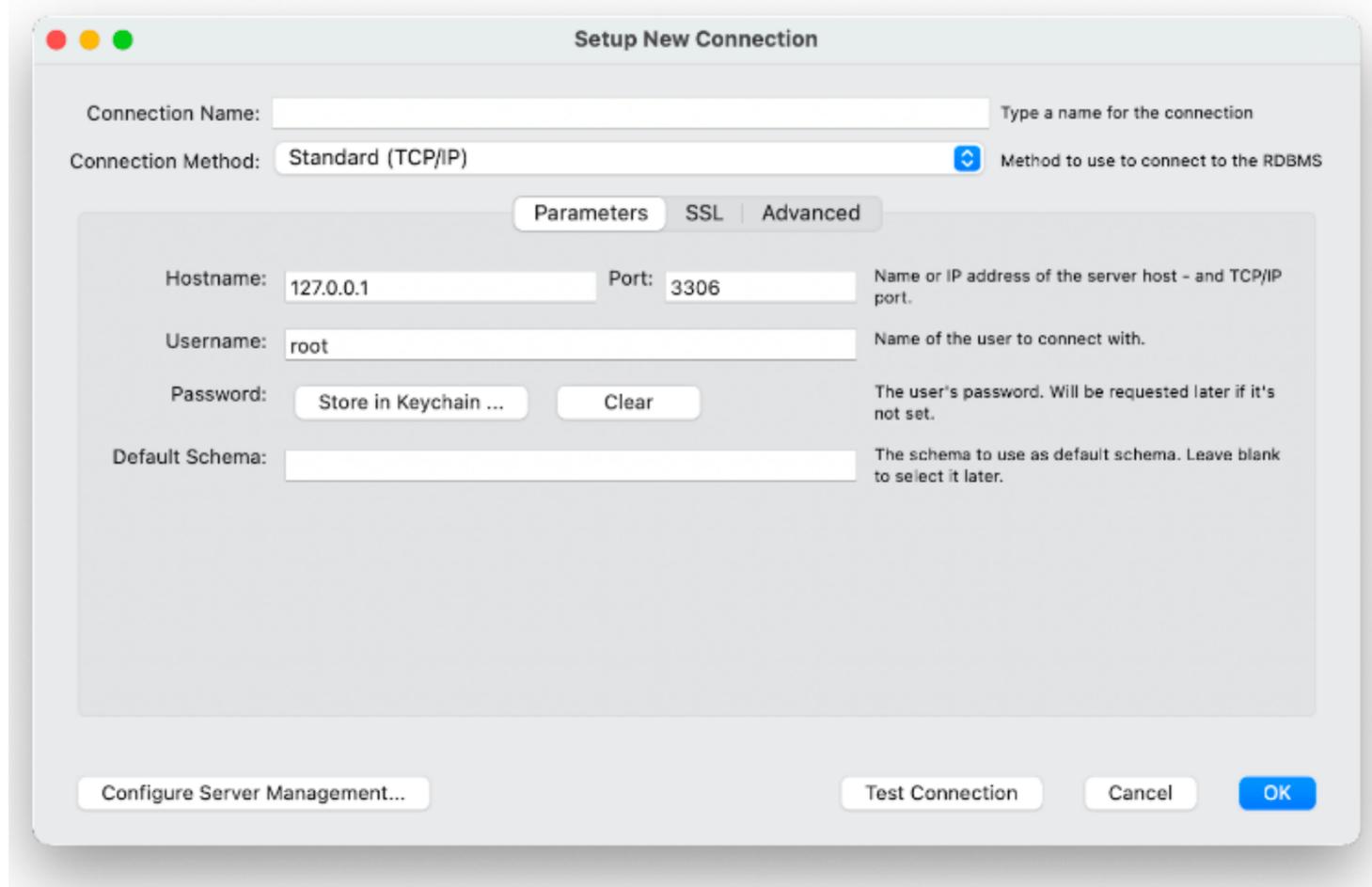
Si la instalación ha finalizado correctamente, debe contar con el community server y con workbench, es hora de conectarse por primera vez al servidor para verificar como trabajar desde la herramienta.

Para ello es necesario abrir workbench y en la sección, al inicio, que indica MySQL Connections debe crear una local. Para ello es necesario que recuerde el usuario y la contraseña que definió en el momento de la instalación del servidor, si no hizo modificaciones y dejó los valores por defecto, el usuario debe ser root y sin contraseña o root.

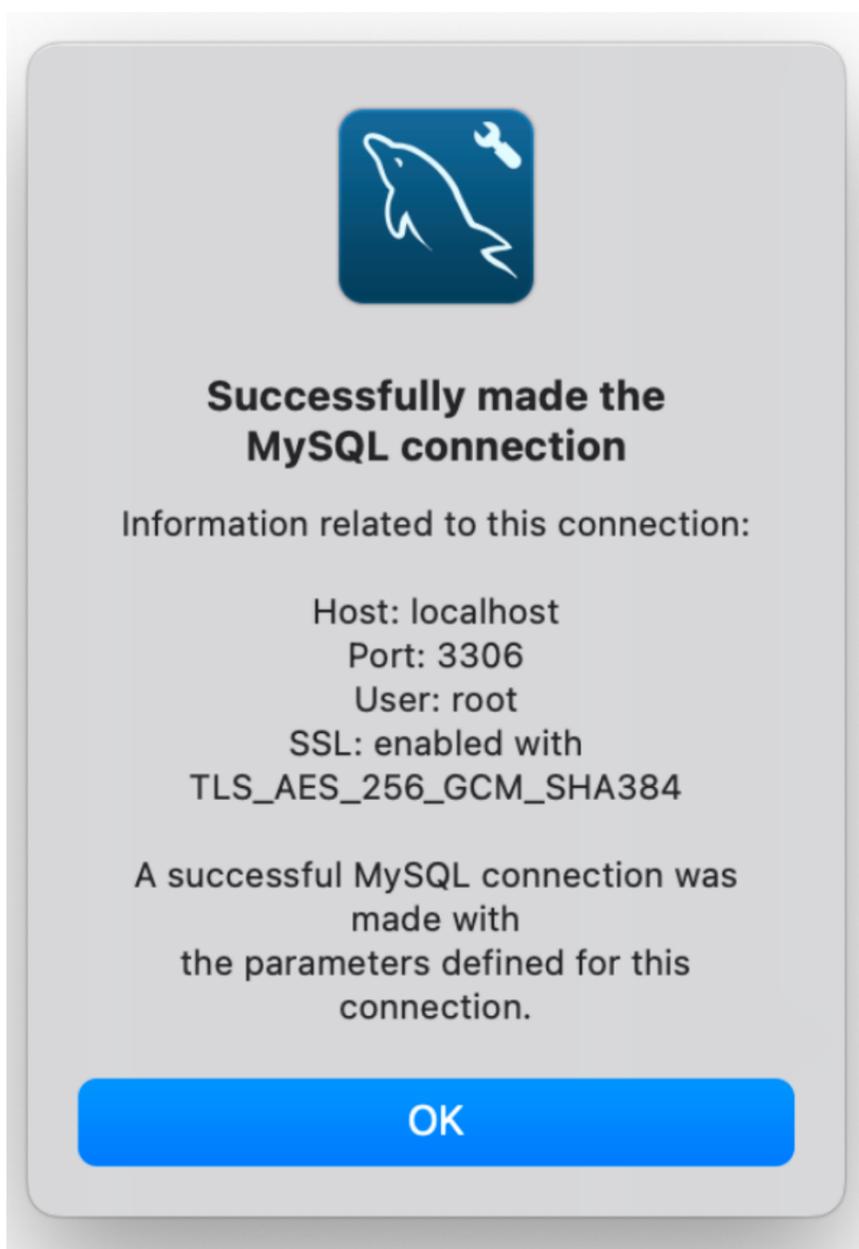
MySQL Connections + ↻



Clic en el botón de agregar (+) y a continuación diligencia los campos solicitados:

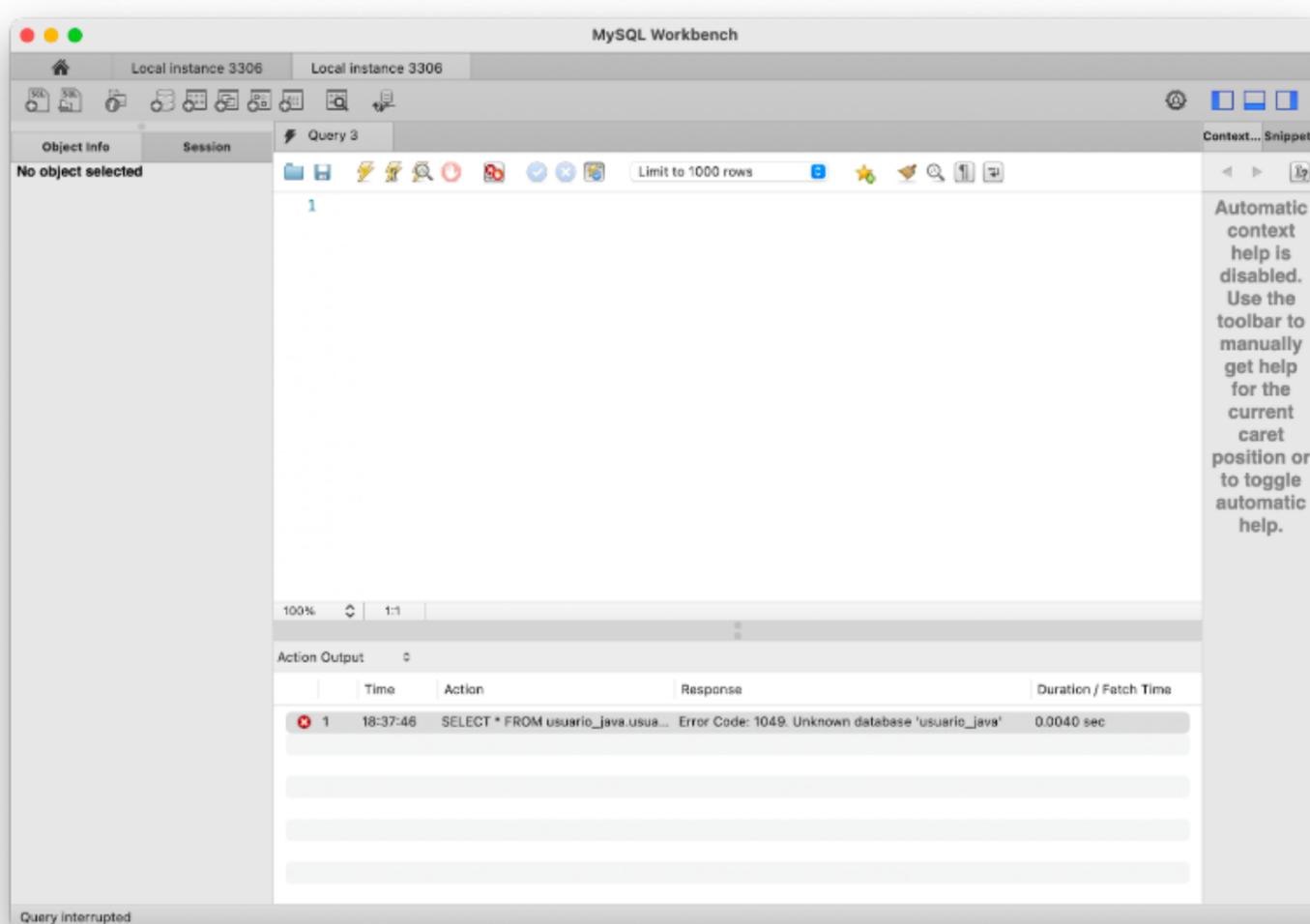


Lo primero es el nombre de la conexión, en este caso, local, y los demás valores los dejamos por defecto, a no ser que debe modificar su usuario y contraseña. Clic en test connection y debería salir, si todo es correcto, un mensaje como el siguiente:

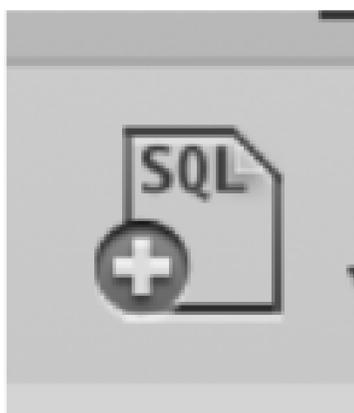


Esto indica que se ha conectado con éxito al servidor, finalmente clic en OK y ahora aparece la lista de conexiones guardadas. Para conectarse de manera rápida a una conexión guardada, solo es necesario hacer clic sobre el nombre de la misma.

Debe aparecer una ventana como la que se presenta a continuación, la ventana abre con un documento SQL que permite la ejecución de scripts:



Para abrir una nueva ventana de script, debe hacer clic en el ícono superior izquierdo que indica SQL +.



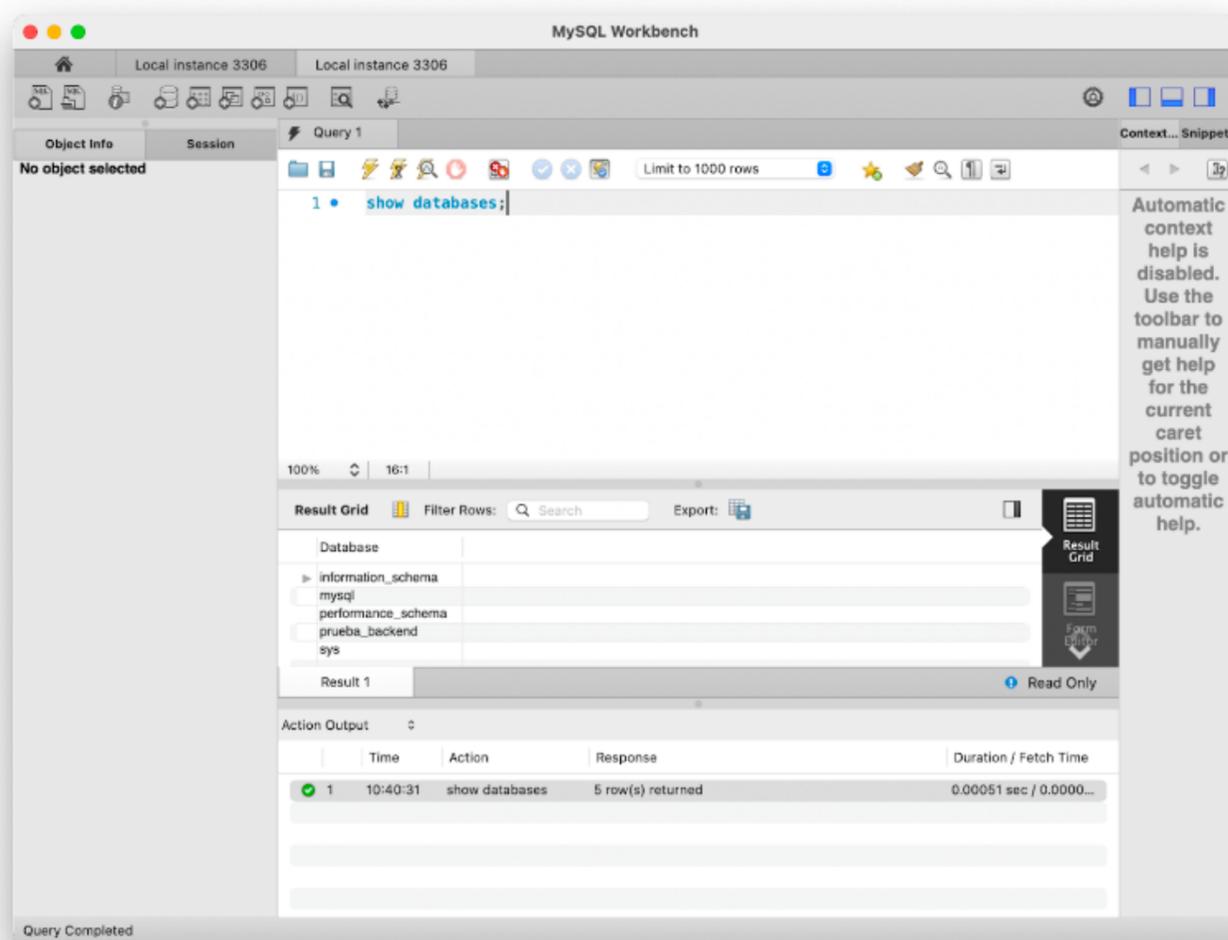
Ahora es momento de escribir las primeras líneas de SQL para conocer un poco de con qué contamos en el servidor instalado.

El primer comando permite conocer el listado de las bases de datos en el servidor, note que el final del comando se indica con un punto y coma (;), en caso de omitirse, el comando no se ejecuta. Para ejecutar el script debe hacer click en el ícono del rayo o presionando CTRL + ENTER.



show databases;

El resultado se muestra a continuación:

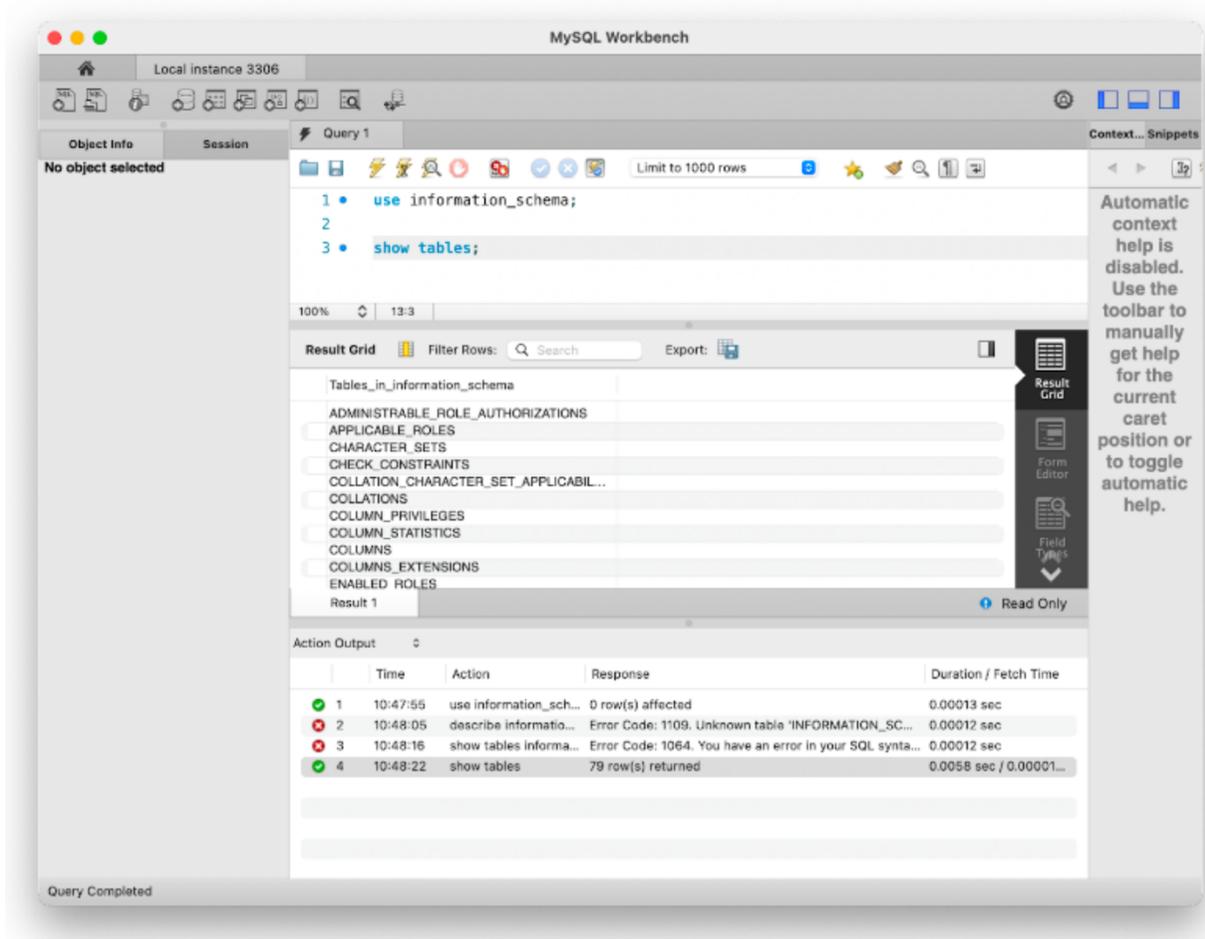


En la zona media, la grilla de resultado, se puede ver el listado de bases de datos con las que inicia el sistema.

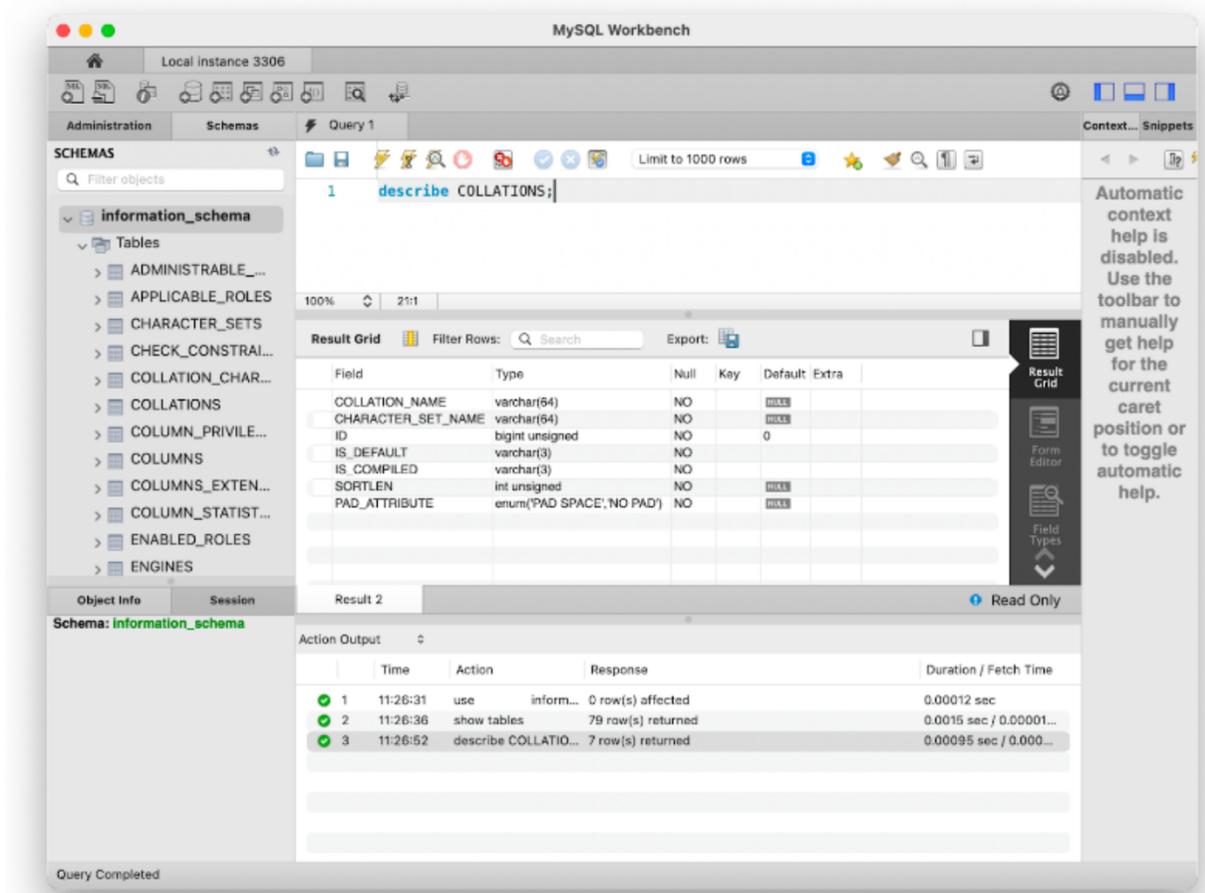
El siguiente paso sería seleccionar una base de datos, para ellos el comando es “use” junto con el nombre de la base de datos, por ejemplo:

use information_schema;

Ahora podemos conocer el listado de tablas con el que cuenta la base de datos, para ello el comando es “show tables”



Lo que presentará el listado de tablas completo de la base de datos, se puede revisar la estructura de cada tabla utilizando el comando “describe” seguido del nombre de la tabla.



Aquí se comienzan a presentar una serie de elementos que hacen parte de la estructura de las tablas, como sus atributos, entre ellos el tipo de dato, si el valor puede ser nulo o no, si tiene un valor por defecto o si es llave.

Antes de avanzar, es necesario conocer los tipos de datos principales con los que se pueden crear las columnas o campos de la tabla.

- **Alfanuméricos:** Contienen cifras y letras. Presentan a menudo una longitud limitada (255 caracteres) pero pueden ser también más cientos de caracteres.
- **Numéricos:** Todos los modelos de bases de datos permiten almacenar números de alguna manera, pero generalmente existen varios tipos de números, generalmente dedicados a diversas capacidades. Incluso puede haber tipos de campos que solamente son capaces de albergar enteros (sin decimales) y tipos de campos en los que podemos poner números reales (con decimales).
- **Booleanos:** Son los datos que pueden representar únicamente dos valores: Verdadero y falso (Sí o No, true o false, 0 o 1)
- **Fechas:** Almacenan fechas facilitando posteriormente su explotación y distintas operaciones. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra...
- **Memos:** Son campos alfanuméricos de longitud ilimitada, usados para textos largos.
- **Autoincrementables:** Son campos numéricos enteros que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta más que evidente: Servir de identificador ya que resultan exclusivos de un registro.

Y en la base de datos se encuentran como:

1. Numérico

TINYINT

Permite números desde -128 hasta 127.

Si se define como **UNSIGNED** (sin signo) permite números desde 0 hasta 255.

SMALLINT

Permite números desde -32768 hasta 32767.

Si se define como **UNSIGNED** (sin signo) permite números desde 0 hasta 65535.

MEDIUMINT

Permite números desde -8388608 hasta 8388607.

Si se define como **UNSIGNED** (sin signo) permite números desde 0 hasta 16777215.

INT

Permite números desde -2147483648 hasta 2147483647.

Si se define como **UNSIGNED** (sin signo) permite números desde 0 hasta 4294967295.

BIGINT

Permite números desde -9223372036854775808 hasta 9223372036854775807.

Si se define como **UNSIGNED** (sin signo) permite números desde 0 hasta 18446744073709551615.

Para los siguientes tipos se puede especificar el número máximo de dígitos y el número de decimales. Por ejemplo: (6,2) tendrá 4 dígitos enteros y 2 decimales dando un tamaño total de 6, por ejemplo, 1234.56.

DECIMAL

Permite almacenar números decimales de punto fijo grandes, por lo que sus cálculos son exactos.

FLOAT

Permite almacenar números decimales de punto flotante pequeños, por lo que sus cálculos son aproximados.

DOUBLE

Permite almacenar números decimales de punto flotante grandes, por lo que sus cálculos son aproximados.

2. Fecha y marca temporal

DATE

Permite almacenar una fecha con el formato YYYY-MM-DD (4 dígitos para el año, 2 dígitos para el mes y 2 dígitos para el día).

DATETIME

Permite almacenar fecha y hora con el formato YYYY-MM-DD HH:MM:SS (4 dígitos para el año, 2 dígitos para el mes, 2 dígitos para el día, 2 dígitos para las horas, 2 dígitos para los minutos y 2 dígitos para los segundos).

El rango soportado es de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'.

TIMESTAMP

Permite almacenar fecha y hora con el formato YYYY-MM-DD HH:MM:SS (4 dígitos para el año, 2 dígitos para el mes, 2 dígitos para el día, 2 dígitos para las horas, 2 dígitos para los minutos y 2 dígitos para los segundos).

El rango soportado es de '1970-01-01 00:00:01' UTC a '2038-01-19 03:14:07' UTC.

TIME

Permite almacenar la hora con el formato HH:MM:SS (2 dígitos para las horas, 2 dígitos para los minutos y 2 dígitos para los segundos).

YEAR

Permite almacenar el año con el formato YYYY (4 dígitos para los años).

3. Cadena

CHAR

Permite almacenar una cadena de datos con longitud fija. Siempre reservará espacio para la longitud definida aunque no se utilice. La longitud máxima es de 255.

VARCHAR

Permite almacenar una cadena de datos (caracteres, números y caracteres especiales) con longitud variable. No reserva el espacio de la longitud máxima definida, ya que ocupa espacio del tamaño real de los datos. La longitud máxima es de 255.

TINYTEXT

Permite almacenar una cadena de datos (solo caracteres; no admite números ni caracteres especiales) de una longitud máxima de 255 caracteres.

TEXT

Permite almacenar una cadena de caracteres de longitud máxima de 65,535.

MEDIUMTEXT

Permite almacenar una cadena con una longitud máxima de 16,777,215 caracteres.

LONGTEXT

Permite almacenar una cadena de longitud máxima de 4,294,967,295 caracteres.

BLOB

Permite almacenar datos de tipo BLOB (Binary Large Object). Admite una longitud máxima de 65,535 bytes de datos.

MEDIUMBLOB

Permite almacenar datos tipo BLOB con una longitud máxima de 16,777,215 bytes.

LOBLOB

Permite almacenar un BLOB de longitud máxima de 4.294,967,295 bytes.

ENUM

Permite definir una lista de posibles valores que pueden almacenarse. La longitud máxima es de 65,535. Si se intenta introducir un valor que no esté incluido en la lista, se insertará valor vacío.

SET

Permite definir una lista de posibles valores que pueden almacenarse. La longitud máxima es de 64. Los valores posibles se pueden combinar.

4. Espacial

MySQL soporta los siguientes tipos datos espaciales basados en la especificación OpenGIS:

GEOMETRY

Permite almacenar valores de cualquier tipo espacial.

POINT

Permite almacenar un punto.

LINestring

Permite almacenar una cadena de puntos.

POLYGON

Permite almacenar un polígono.

MULTIPOINT

Permite almacenar una colección de puntos.

MULTILINESTRING

Permite almacenar una colección de objetos LineString.

MULTIPOLYGON

Permite almacenar una colección de polígonos.

GEOMETRYCOLLECTION

Permite almacenar objetos geométricos de cualquier tipo.

5. JSON

Permite almacenar un dato con formato JSON (JavaScript Object Notation).

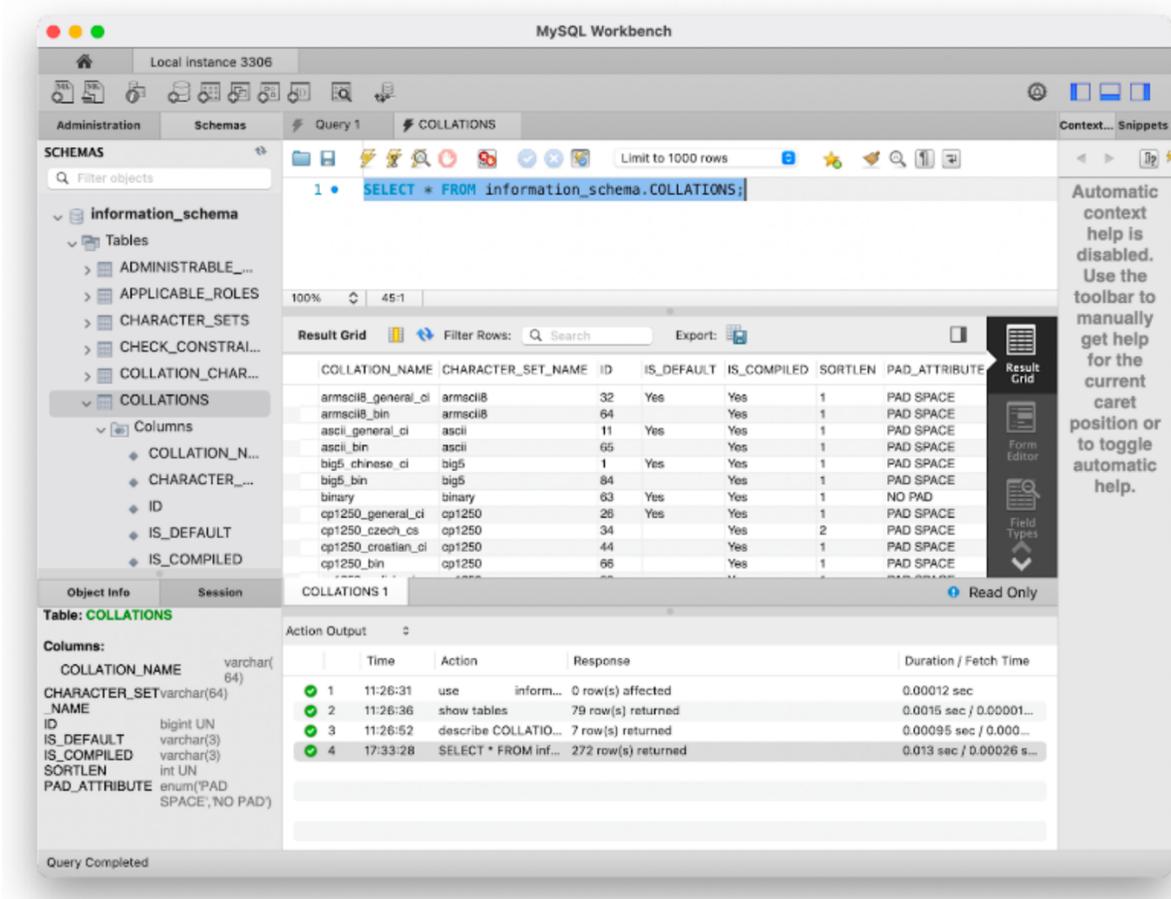
Primera consulta

Ya es posible conocer alguna información sobre las bases de datos y las tablas utilizando sentencias SQL, pero todavía no se ha visualizado ningún dato, ahora, aunque las bases de datos que se están consultando no parecen tener mucho sentido para nosotros, el siguiente paso es hacer un llamado de datos base.

Para realizar consultas o queries, el comando a utilizar el “select” seguido del nombre de los campos que queremos mostrar o asterisco “*” para traerlos todos, luego el comando “from” seguido del nombre de la tabla o base de datos punto (.) y nombre de la tabla.

```
SELECT * FROM information_schema.COLLATIONS;
```

```
SELECT * FROM COLLATIONS;
```



Para que se tenga un poco más entendimiento a la hora de consultar la base de datos, es hora de cargar una de ejemplo llamada SAKILA.

Trabajando con SAKILA

Para descargarla e instalarla se puede seguir el proceso indicado en la documentación oficial [MySQL :: Sakila Sample Database :: 4 Installation](#)

O descargarla desde <https://downloads.mysql.com/docs/sakila-db.zip> y cargarla utilizando la interfaz de workbench

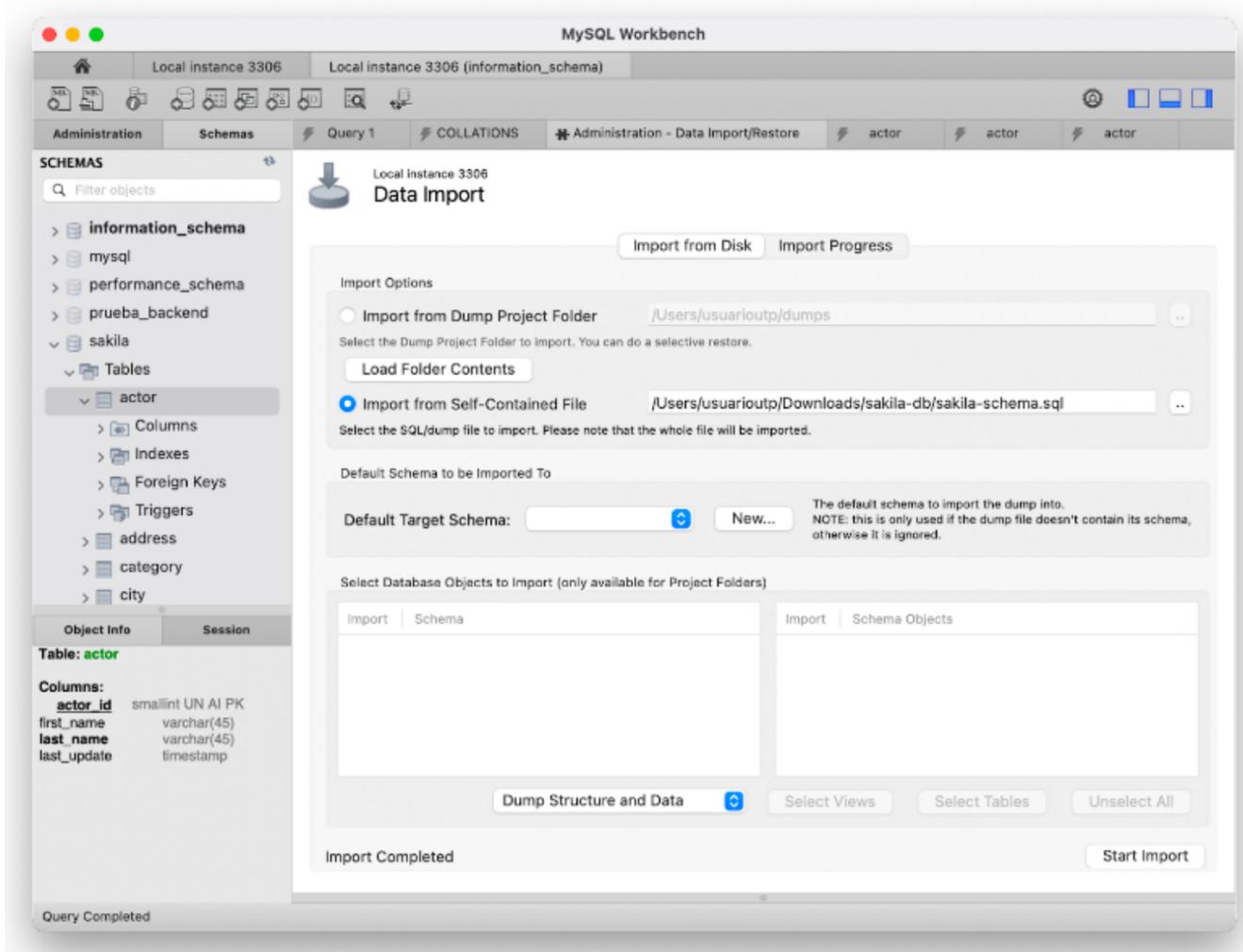
Para este caso, se trabajará con la segunda opción, utilizando workbench. Al descargar el archivo .zip, hay dos archivos que nos interesan, uno llamado sakila-schema.sql y otro sakila-data.sql.

El primero llamado SCHEMA, contiene la estructura de la base de datos, es decir el diseño de cada una de las tablas, sus nombres, campos, tipos de datos, atributos y relaciones. El segundo contiene los datos para poblar cada tabla, es decir, cada uno de los registros de la base de datos.

Para importar la base de datos SAKILA primero debemos cargar la estructura y posteriormente los datos, el proceso se puede hacer de dos maneras, utilizando la herramienta de importación o copiando el contenido del archivo .sql dentro de un documento de script.

En clase, se van a realizar los dos procesos, debe tener en cuenta que si el archivo es muy grande, no es recomendable copiar y pegar el contenido porque, por un lado la apertura del archivo puede ser demorada y por el otro, la ejecución del script puede fallar por falta de recursos en la máquina.

Para importar los archivos utilizando el asistente de workbench debe ir al menú server o servidor, luego la opción "data import" luego "import from disk", ahora en la opción "import from self-contained file" busca el archivo de SAKILA que indique "scheme", finalmente clic en "start import"



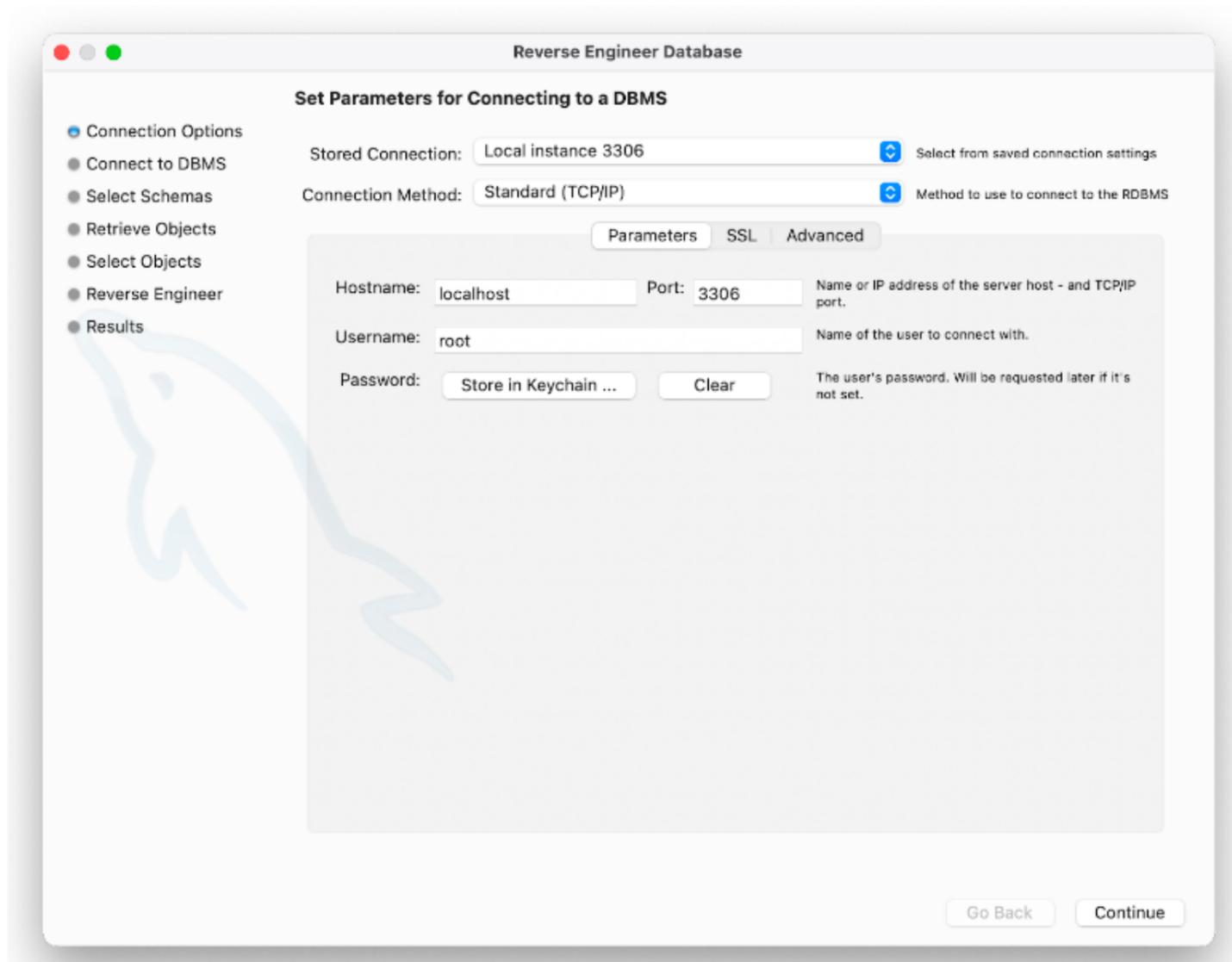
Si todo sale bien, al darle clic en el botón refrescar el panel izquierdo llamado "Schemas" o utilizar el comando "show databases", debe aparecer una nueva llamada "sakila". Si intentamos consultar algo de ella, nos daremos cuenta que no tiene datos todavía.

Para importar los datos, el proceso es igual, solo que ahora en el momento de seleccionar el archivo, debe seleccionar el que indica "data" y clic en "start import".

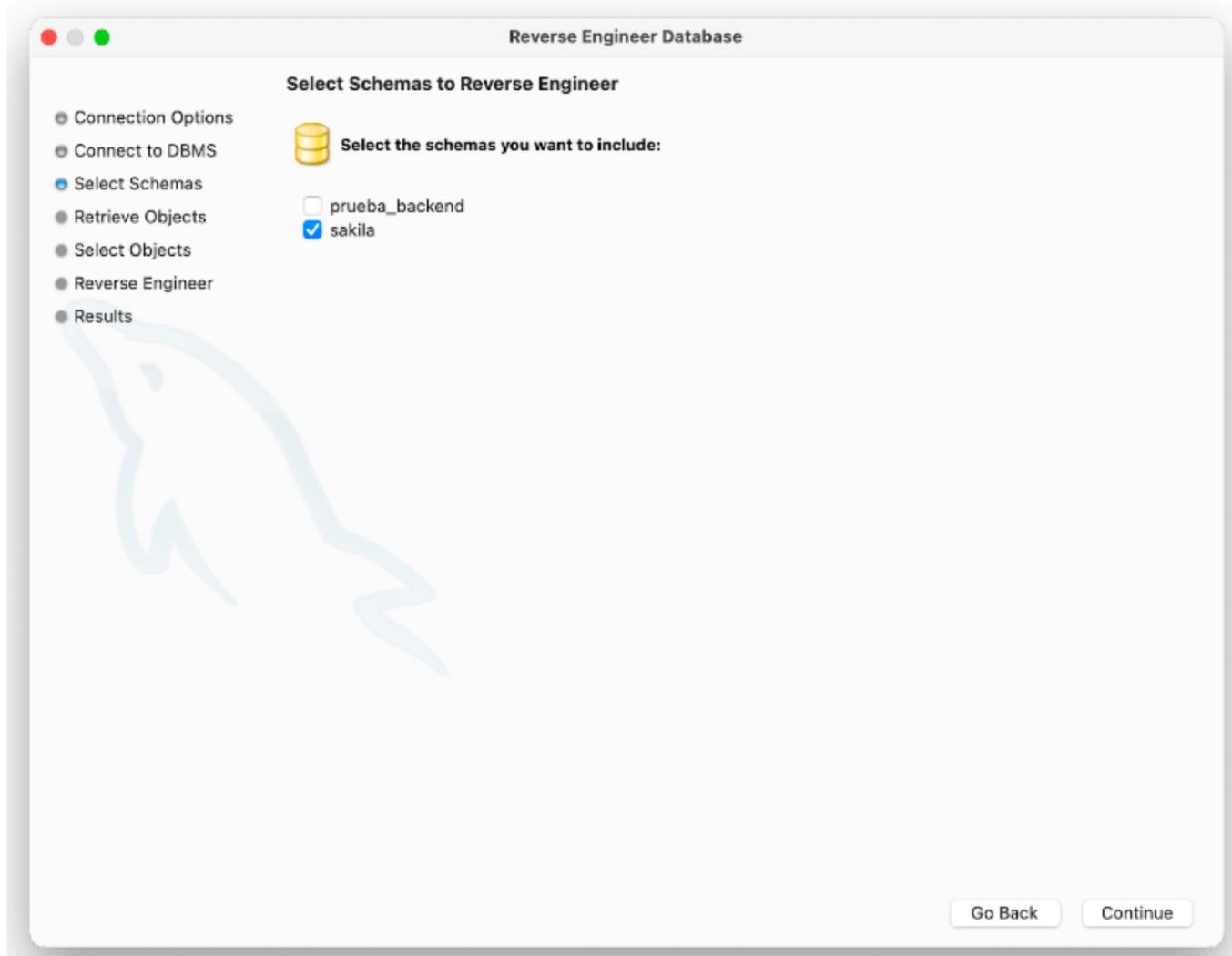
En este punto ya debe contar con la base de datos SAKILA con todos sus datos y lista para trabajar sobre ella.

Antes de iniciar los ejercicios, se puede generar un diagrama entidad relación de la base de datos utilizando la opción de ingeniería inversa que trae la herramienta, así se puede conocer como es la estructura de una base de datos y también entender de mejor manera la información que almacena esta base de datos.

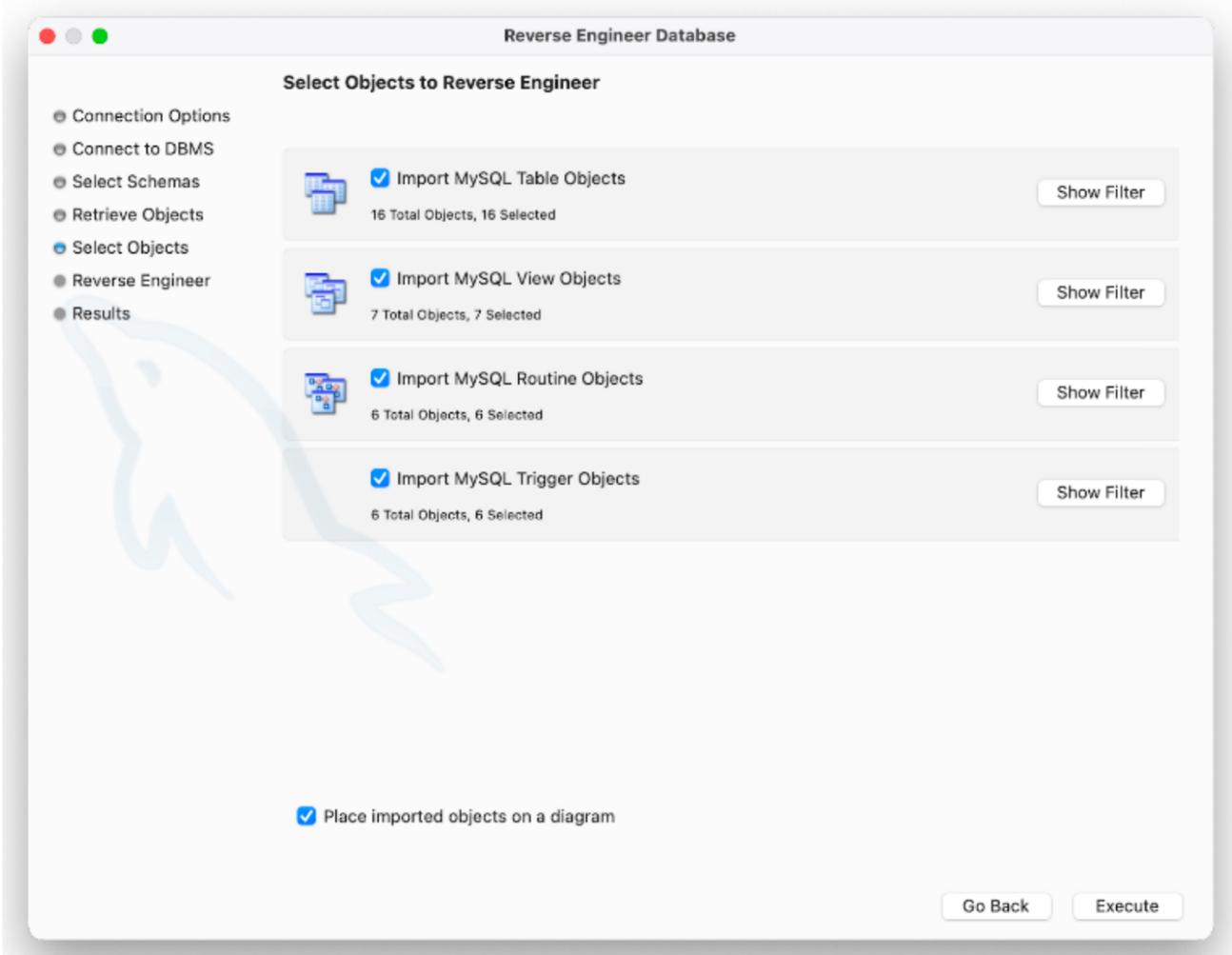
Para generar el diagrama, clic en la opción “database” luego en “reverse engineer”, el asistente inicia con los datos para la conexión al servidor.



Luego selecciona SAKILA:

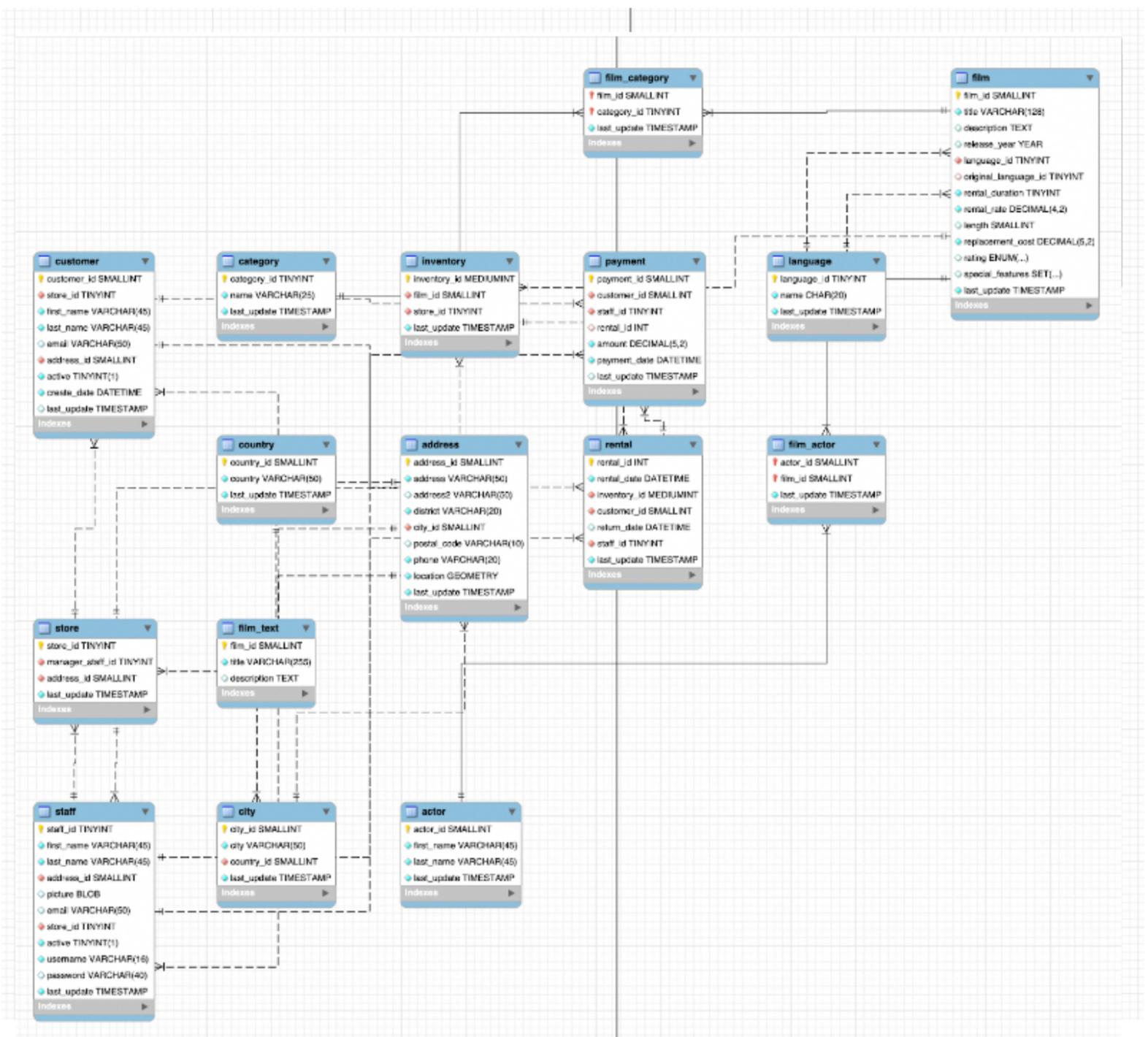


Ahora debe seleccionar lo que desea presentar en el diagrama, por ahora nos interesan solo las tablas.



Al finalizar el proceso, debe aparecer una ventana con un diagrama como el siguiente:

<https://drive.google.com/file/d/1zoEydoLMQgjbnp99VD8bm2FXwN3Nga4/view?usp=sharing>



Este es un diagrama que representa las relaciones entre las tablas y los campos de cada una de ellas. En este caso, la documentación oficial también puede entregar información, pero no siempre es el caso. [MySQL :: Sakila Sample Database :: 5 Structure](#)

Haciendo ejercicios con SAKILA [GitHub - MisaoDev/SQL-Ejercicios: Ejercicios SQL para usar con Sakila](#)

Vamos a realizar los ejercicios en clase y a medida que aumenta la dificultad se explican las sentencias que aparecen como el where, in, order by, group by, having y joins.

Comandos

Aunque SAKILA puede ser un punto de partida para interactuar con algunos comandos, seguir el tutorial que se entrega desde el sitio web de W3Schools para SQL permite interactuar de una mejor manera y aumentar las horas de práctica.

- SELECT - extrae la data de la base de datos
- UPDATE - Actualiza la data
- DELETE - Elimina la data
- INSERT INTO - inserta nueva data
- CREATE DATABASE - Crea una nueva base de datos
- ALTER DATABASE - Modifica una base de datos existente
- CREATE TABLE - crea una nueva tabla en una base de datos seleccionada
- ALTER TABLE - Modifica una tabla creada
- DROP TABLE - Elimina una tabla

El tutorial está disponible en [SQL Tutorial](#)

Base de datos: [w3schools.sql](#)

Ejercitaciones

En la web, existen diferentes recursos que ayudan en el proceso de aprendizaje de bases de datos y consultas en general, el reto para este momento, es completar los ejercicios disponibles en [Practice SQL](#)

La web de SQL Practice ofrece diferentes niveles de dificultad y ayudas como mostrar la solución o presentar el diagrama para comprender de mejor manera la estructura de la base de datos.

- O qué tal aprender de una manera más divertida resolviendo crímenes
[SQL Police Department](#)
- [SQLZoo](#)
- [SQLBolt - Learn SQL - SQL Lesson 1: SELECT queries 101](#)
- [DataLemur](#)
- [Tutorial SQL en Español interactivo con intérprete SQL](#)
- [Learn SQL: Interactive SQL Book](#)
- [The SQL Murder Mystery](#)