**Ejercicio Individual 01 - módulo 03**

**NOMBRE ESTUDIANTE:** [Ricardo Cea Gallardo](mailto:ricardo.cea@ug.uchile.cl)

**CURSO: Aplicaciones Full Stack Java Trainee**

**DESARROLLO:**

**Parte 1. Exploración de MySQL**

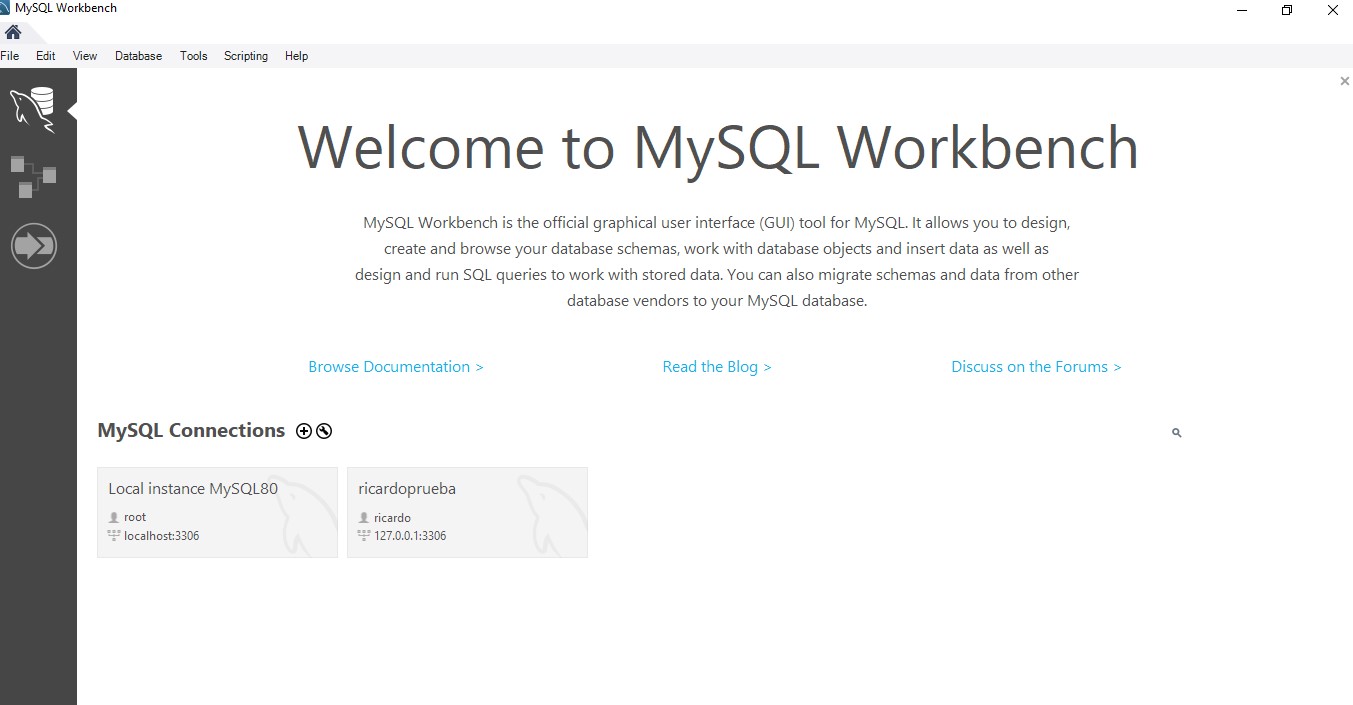
1. Defina los siguientes conceptos:

* **Base de datos:**  En términos generales, una base de datos es un conjunto de datos estructurados que pertenecen a un mismo contexto y, en cuanto a su función, se utiliza para administrar de forma electrónica grandes cantidades de información. Estos datos se organizan en tablas, que a su vez pueden relacionarse entre sí mediante claves primarias y foráneas, creando así una estructura relacional.
* **Tabla:** En las [bases de datos](https://es.wikipedia.org/wiki/Bases_de_datos), se refiere al tipo de modelado de datos donde se guardan los datos recogidos por un programa. Esta información se almacena en filas (registros) y columnas (campos). Las tablas son la principal forma de organizar y almacenar los datos en una base de datos relacional.
* **Registro:** representa un objeto único de datos implícitamente estructurados en una tabla. Cada registro (o fila) de una tabla representa un conjunto de datos relacionados, y todos los registros de la misma tabla tienen la misma estructura.
* **Columna:** Una columna o campo es cada uno de los valores únicos que proporcionan la estructura según la cual se descomponen las filas (registros o tuplas) de una tabla. Los datos de cada campo pueden ser de diferentes tipos, pero solo uno por columna: numéricos, alfanuméricos, lógicos (verdadero o falso), de texto, multimedia, binarios, etc.
* **Llave primaria:** También llamada Clave primaria, corresponde a un campo o a una combinación de campos que identifica de forma única a cada fila de una tabla. Una clave primaria comprende de esta manera una columna o conjunto de columnas. No puede haber dos filas en una tabla que tengan la misma clave primaria. La llave primaria se define al crear la tabla y se utiliza para garantizar la integridad de los datos, ya que impide que se ingresen registros duplicados o que se actualicen o eliminen registros por error. Además, la llave primaria se utiliza para establecer relaciones entre las tablas en la base de datos mediante el uso de llaves foráneas.
* **Llave foránea:** También llamada Clave foranea, es un campo o conjunto de campos en una tabla que se utilizan para establecer una relación con otra tabla en la base de datos. La llave foránea en una tabla hace referencia a la clave primaria de otra tabla, lo que significa que los valores de la llave foránea deben coincidir con los valores de la clave primaria en la tabla relacionada. Esta relación entre las tablas permite acceder a los datos relacionados de ambas tablas mediante consultas de la base de datos.

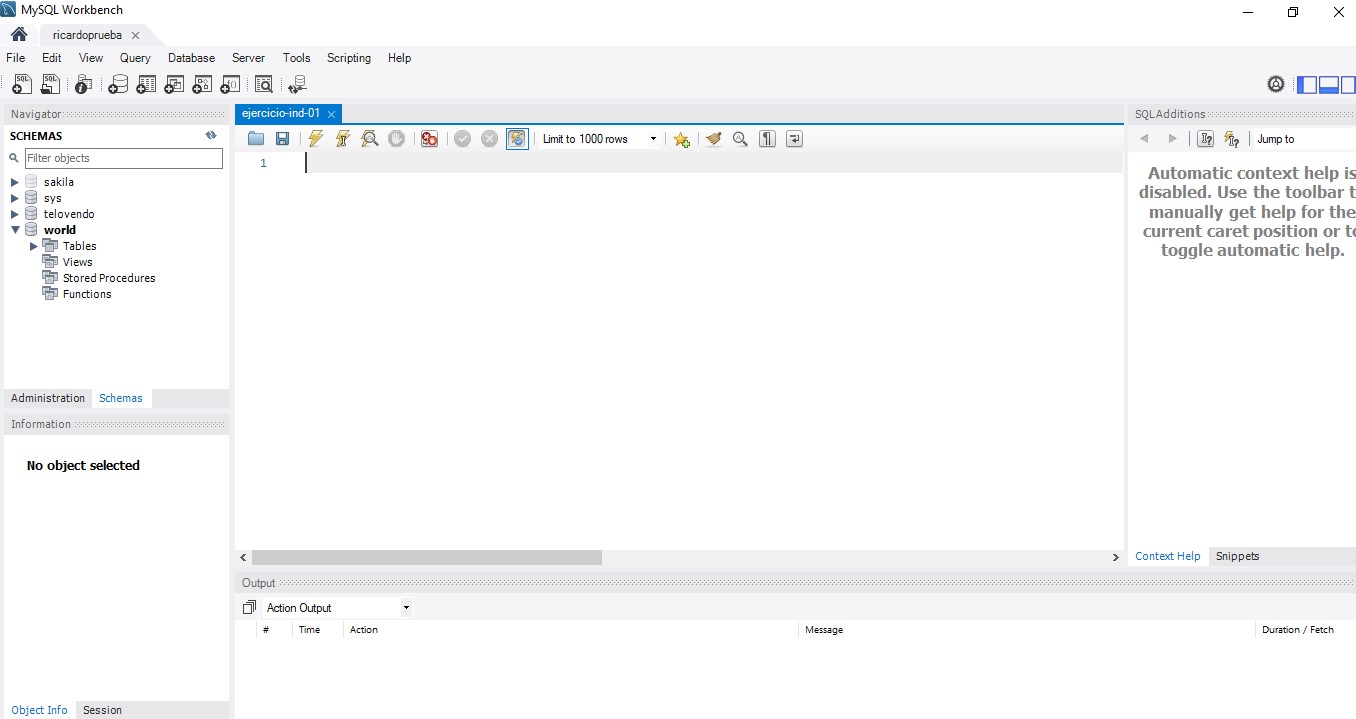
Cabe señalar que en una base de datos relacional, una relación entre dos tablas se establece mediante el uso de una llave primaria y una llave foránea. La llave primaria se utiliza para identificar de manera única cada registro en una tabla, mientras que la llave foránea se utiliza para establecer una relación entre dos tablas.

1. A continuación nos solicitan instalar el programa MySQL. Luego de esto nos crearemos un usuario y contraseña. A este usuario, se le entregarán los permisos necesarios de lectura y escritura en todas las tablas de la base de datos con el nombre “world” (base de datos de ejemplo).

Aparecerá la siguiente interfaz de usuario al ejecutar el programa My SQL Workbench:



Yo entraré a la conexión llamada “ricardoprueba”, que se puede visualizar en la imagen anterior. Nos aparecerá la siguiente interfaz de usuario:



Acá se ingresará a la base de datos world y verificaremos que se encuentren disponibles todas las tablas dentro. para esto usaremos el comando:

SELECT \* FROM nombre\_de\_la\_tabla;

Dentro de la base de datos verificamos que se encuentran las tablas llamadas “city”, “country” y “countrylanguage”. La cantidad de registros de cada una de estas tablas es la siguiente:

* city: 4079
* country: 239
* countrylanguage: 984

1. Tomado de referencia la base de datos “world”, contestaremos las siguientes preguntas:

* ¿Cuántas ciudades existen en la base de datos?

R. Para esto usamos el comando:

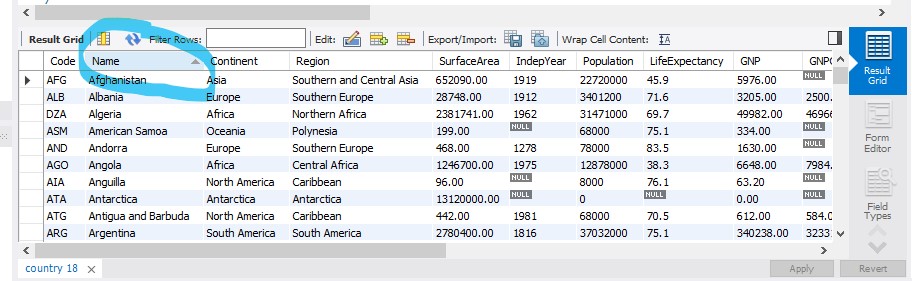
SELECT COUNT(Name)

FROM city;

lo que nos arroja un total de 4079 ciudades registradas en la tabla city de la base de datos.

* Si se ordenan los países de acuerdo con su nombre ¿cuál sería el primero y cuál el último?

R. SQL Workbench nos permite en su interfaz realizar este orden tal como se ve a continuación en la imagen:



Al hacer click sobre name, SQL Workbench nos ordena el campo de la tabla alfabéticamente. En este caso, Nos toma como el primer resultado a “Afghanistan” y como último a “Zimbabwe”

* Tomando en cuenta todos los países de Sudamérica ¿cuántas ciudades existen entre todos ellos?

R. Para determinar esto podemos usar el siguiente código en SQL Workbench:   
  
 SELECT SUM(Capital) AS TotalCiudades

FROM country

WHERE Continent = 'South America';

Esto nos entrega un total de 24633 ciudades existentes en sudamérica

* Considerando todos los idiomas declarados en la base, ¿cuántos idiomas corresponden a países de Sudamérica?

Para esto utilizamos el comando de consulta:

SELECT COUNT(DISTINCT Language) as num\_languages

FROM countrylanguage

WHERE CountryCode IN (SELECT Code FROM country WHERE Continent = 'South America');

Esto hará una subconsulta en la tabla “country” desde la tabla “countrylanguage”, para que la base de datos nos devuelva la cantidad de idiomas hablados en sudamérica. El resultado que nos entrega esta es 21 idiomas.

**Parte 2. Crear una base de datos.**

1. Para el proyecto, crearemos una base de datos local que llamaremos “myproyectogo”, esta tendrá 3 tablas con los siguientes datos:

* **OPERADORES**: RUN (\*), nombre, apellido, dirección, correo electrónico
* **USUARIOS**: id coder (\*), nombre, apellido, correo electrónico, teléfono
* **CAPACITACIÓN**: código curso (\*), nombre, horario

Aquí, los campos marcados (\*) serán de clave primaria en cada tabla. Además, se añadirán 10 registros a cada tabla.

Para hacer esto utilizaremos el comando:

CREATE DATABASE myproyectogo;

USE myproyectogo;

:

Luego, agregaremos 3 tablas:

CREATE TABLE OPERADORES (

RUN VARCHAR(20) PRIMARY KEY,

nombre VARCHAR(50),

apellido VARCHAR(50),

direccion VARCHAR(100),

correo\_electronico VARCHAR(100)

);

CREATE TABLE USUARIOS (

id\_coder INT PRIMARY KEY,

nombre VARCHAR(50),

apellido VARCHAR(50),

correo\_electronico VARCHAR(100),

telefono VARCHAR(20)

);

CREATE TABLE CAPACITACIÓN (

codigo\_curso INT PRIMARY KEY,

nombre VARCHAR(100),

horario VARCHAR(50)

);

E insertamos los datos de la siguiente forma:

-- Insertando datos a la base de datos

-- OPERADORES:

INSERT INTO OPERADORES (RUN, nombre, apellido, direccion, correo\_electronico) VALUES

('23476321-2', 'Juan', 'Marín', 'Av. Los Conquistadores 7000, Providencia', 'juan.marin@gmail.com'),

('14348907-3', 'Alberto', 'Ponce', 'Los Caracoles 345, Viña del Mar', 'a\_ponce54@gmail.com'),

('12556789-1', 'Pedro', 'Masterco', 'Av. Peñablanca 560, Algarrobo', 'pedromas556@gmail.com'),

('10756709-4', 'Ana', 'Rodríguez', 'Av. Vicuña Mackenna 5678, Macul', 'ana.rodriguez@gmail.com'),

('15674223-5', 'Jorge', 'López', 'Calle San Diego 901, Santiago', 'jorge.lopez@gmail.com'),

('12343457-9', 'Carla', 'Torres', 'Av. Matta 1234, Santiago', 'carla.torres@hotmail.com'),

('11343565-9', 'José', 'Fernández', 'Av. Libertador Bernardo O’Higgins 246, Santiago', 'jose.fernandez@gmail.com'),

('17677451-8', 'Lucía', 'Gómez', 'Av. Kennedy 123, Vitacura', 'lucia.gomez45@gmail.com'),

('12321499-8', 'Manuel', 'Martínez', 'Calle Los Nogales 789, Providencia', 'manuel.martinez@gmail.com'),

('13988789-4', 'Sofía', 'Sánchez', 'Av. Tobalaba 456, Providencia', 's\_sanchez34@yahoo.com');

-- USUARIOS:

INSERT INTO USUARIOS (id\_coder, nombre, apellido, correo\_electronico, telefono) VALUES

(1, 'Marcelo', 'González', 'marcelo.gonzalez@gmail.com', '+56987654321'),

(2, 'Fabiola', 'Muñoz', 'fabiola.munoz@gmail.com', '+56912345678'),

(3, 'Pablo', 'Martínez', 'pablo.martinez@gmail.com', '+56923456789'),

(4, 'Mónica', 'Pérez', 'monica.perez@gmail.com', '+56934567890'),

(5, 'Diego', 'Vargas', 'diego.vargas@hotmail.com', '+56945678901'),

(6, 'Valentina', 'Hernández', 'valentina.hernandez@gmail.com', '+56956789012'),

(7, 'Matías', 'Gómez', 'matias.gomez@gmail.com', '+56967890123'),

(8, 'Isabel', 'Fernández', 'isabel.fernandez@gmail.com', '+56978901234'),

(9, 'Andrés', 'Silva', 'andres.silva@hotmail.com', '+56989012345'),

(10, 'Camila', 'García', 'camila.garcia@gmail.com', '+56990123456');

-- CAPACITACIÓN

INSERT INTO CAPACITACIÓN (codigo\_curso, nombre, horario) VALUES

('01', 'Programación Básica', 'Lunes y Miércoles de 18:00 a 20:00 hrs.'),

('02', 'Diseño Gráfico', 'Martes y Jueves de 09:00 a 11:00 hrs.'),

('03', 'Inglés técnico para aplicaciones', 'Lunes y Miércoles de 10:00 a 12:00 hrs.'),

('04', 'Gestión de Proyectos', 'Martes y Jueves de 16:00 a 18:00 hrs.'),

('05', 'Marketing Digital', 'Miércoles y Viernes de 14:00 a 16:00 hrs.'),

('06', 'Análisis de Datos', 'Lunes y Miércoles de 14:00 a 16:00 hrs.'),

('07', 'Diseño Web', 'Martes y Jueves de 18:00 a 20:00 hrs.'),

('08', 'Redes Sociales', 'Miércoles y Viernes de 10:00 a 12:00 hrs.'),

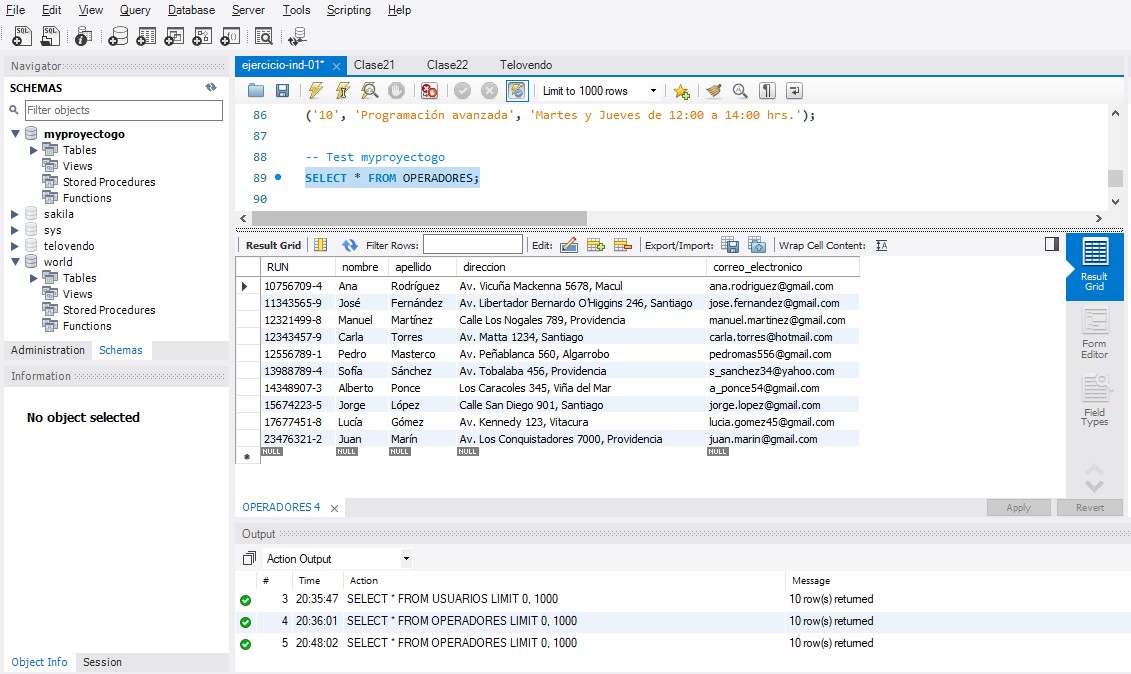
('09', 'Programación nivel medio', 'Lunes y Miércoles de 16:00 a 18:00 hrs.'),

('10', 'Programación avanzada', 'Martes y Jueves de 12:00 a 14:00 hrs.');

Al final, podemos revisar que las tablas se hayan completado correctamente insertando el comando:

SELECT \* FROM <nombre\_tabla>;

Podremos observar entonces, por ejemplo, para el caso de la tabla “OPERADORES” lo que se muestra en la siguiente imagen:



En la imagen podemos observar los 10 registros añadidos.

1. Dentro de My SQL Workbench, abriremos la base de datos “sakila”. Esta base, almacena datos de arriendo de películas, junto con datos de las películas y actores.

Usando estos datos como referencia, conteste las siguientes preguntas:

* ¿Cuántas películas están registradas?

R. Usamos el código:

SELECT COUNT(\*)

FROM film;

Esto nos da un resultado de 1000 películas registradas en la base de datos.

* ¿Cuántos clientes existen en la base de datos?

R. Usamos el código:

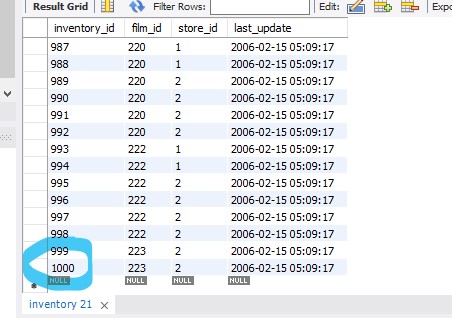
SELECT COUNT(\*)

FROM customer;

Esto nos da un resultado de 599 clientes registrados en la base de datos.

* ¿Cuántos títulos existen en inventario?

R. En la tabla “inventory” podemos visualizar que existen 1000 registros numerados por el campo “inventory\_id”, como se muestra en la siguiente imagen:



* ¿Cuál es la película más arrendada?

Para determinar esto hacemos la consulta:

SELECT f.title, COUNT(\*) as rentals

FROM rental r

JOIN inventory i ON r.inventory\_id = i.inventory\_id

JOIN film f ON i.film\_id = f.film\_id

GROUP BY f.title

ORDER BY rentals DESC;

Esto nos indica que la película más arrendada es “Bucked Brotherhood” con un número de 34 rentas.

1. Luego de utilizar MySQL Workbench, indaguen en el uso de MySQL Shell. ¿Qué diferencias existen entre utilizar una u otra plataforma?

R. Ambas permiten interactuar con las bases de datos MySQL, pero tienen propiedades, características diferentes.

MySQL Workbench es una herramienta de administración visual que proporciona una interfaz gráfica de usuario (GUI) para administrar bases de datos MySQL. Con ella, se pueden crear y modificar bases de datos, diseñar modelos de datos, crear y ejecutar consultas, hacer copias de seguridad y restaurar datos, entre otras tareas. También puede ser utilizada para conectar con servidores de bases de datos remotos.

Por otra parte, MySQL Shell es una herramienta de línea de comandos que proporciona una interfaz de programación de aplicaciones (API) para interactuar con bases de datos MySQL. Con ella, uno se puede conectar a un servidor de bases de datos MySQL y ejecutar comandos y scripts de JavaScript, Python y SQL. Además, MySQL Shell incluye herramientas para la administración de bases de datos, como la creación y modificación de usuarios y roles, y la realización de copias de seguridad y restauración de datos.

En resumidas cuentas, MySQL Workbench es una herramienta de administración visual para bases de datos MySQL y MySQL Shell es una herramienta de línea de comandos que proporciona una API para interactuar con bases de datos MySQL.