**ABPRO 1, MODULO 5, DESARROLLO**

**Integrantes: ClaudioI – Ignacio – Ricardo Ponce – Patricia (25.03.2022)**

Como parte de este ejercicio se necesita crear una base de datos MySQL.

● La base de datos debe ser local, y debe tener por nombre “telovendo”.

● Debes crear un usuario de nombre “admintienda”, con una clase determinada por ti. Este usuario

debe tener permisos totales sobre la base de datos anteriormente creada. Tres Tablas:

A. Cliente: todo cliente se identifica con un código único, sus nombres, sus apellidos, un teléfono,

su dirección, su comuna, el correo electrónico y fecha de registro. Recuerden incluir el tipo de dato,

número máximo de caracteres (en caso de ser necesario).

B. Producto: por cada producto se debe conocer su identificador de producto (SKU), su nombre, la

categoría, quien lo produce y la cantidad existente en stock.

C. Vendedor: se compone de RUN, nombre, apellidos, fecha de nacimiento y la sección a la que

pertenece. Utilicen su imaginación para crear las secciones. Inserten en la tabla al menos 10 registros distintos.

Respondan las siguientes preguntas en grupo:

1. ¿Qué tipo de datos les permite guardar fechas? ¿Y horas?

DATE

DATETIME: YYYY-MM-DD HH:MI:SS

TIMESTAMP: YYYY-MM-DD HH:MI:SS

YEAR: YYYY or YY (Para el manejo de años)

b. ¿Qué utilidad tiene especificar el número de caracteres en SQL?

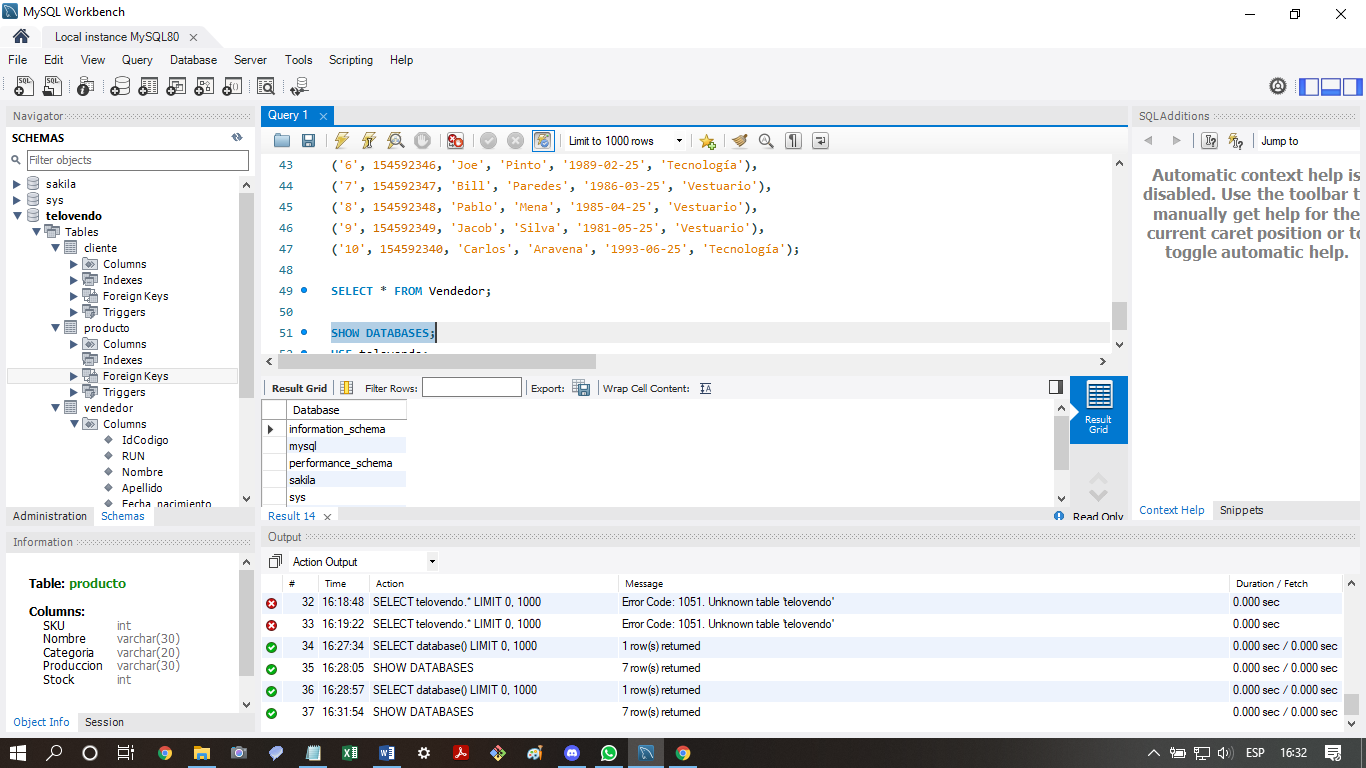
Colocamos aquí entonces la longitud máxima del valor más largo que creemos que podemos llegar a tener. No se refiere al valor máximo, sino a lo largo que va a ser el datos más largo en ese campo. Por ejemplo, el número 10000 es 5 cifras de largo, el 3 es de una cifra de largo, la palabra "capicúa" es de 7 carácteres de largo, no? Pues ese es el dato a poner en esta tabla, el largo máximo que podría llegar a tener un valor.

c. Por último, definan el acrónimo CRUD.

CRUD (Create, Read, Update, Delete) es un acrónimo para las maneras en las que se puede operar sobre información almacenada. Es un nemónico para las cuatro funciones del almacenamiento persistente.

d. Qué palabra clave se utiliza para:

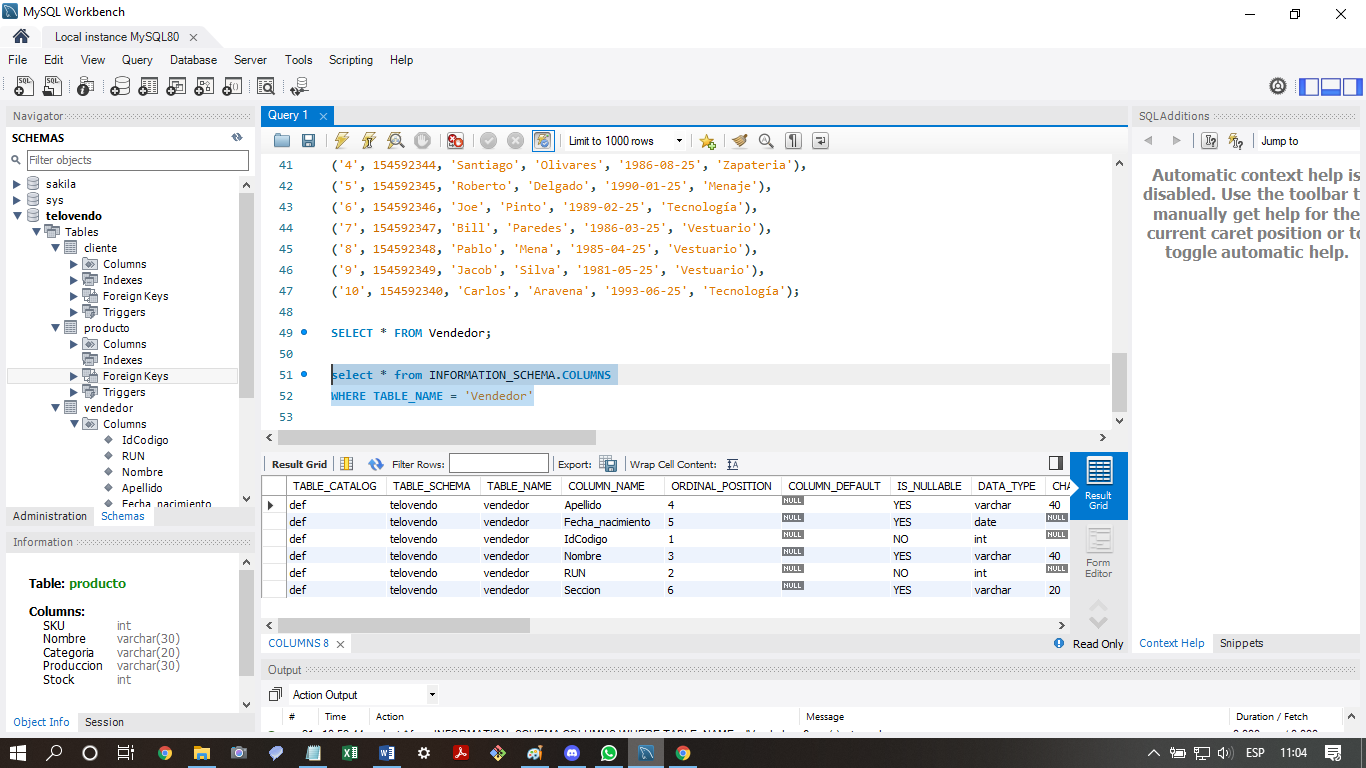
1. Mostrar bases de datos en SQL. SHOW DATABASES;



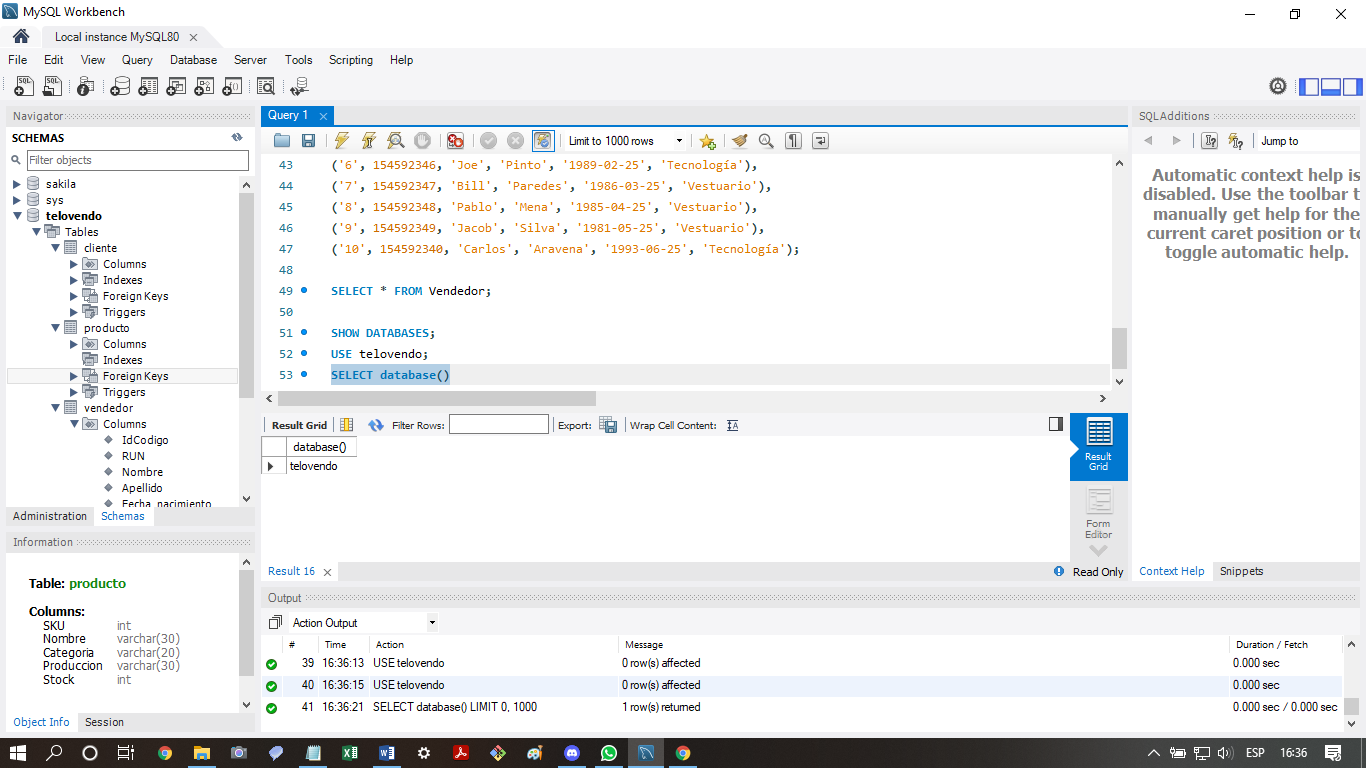
1. Describir una base de datos en SQL.

select \* from INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'Vendedor'



1. Seleccionar una base de datos. SELECT DATABASE()



1. Crear una tabla.

CREATE TABLE Cliente (

IdCodigo int primary key not null,

Nombre varchar (30),

Apellido varchar (30),

Telefono int not null,

Direccion varchar (60),

Comuna varchar (30),

Fecha\_Registro date not null

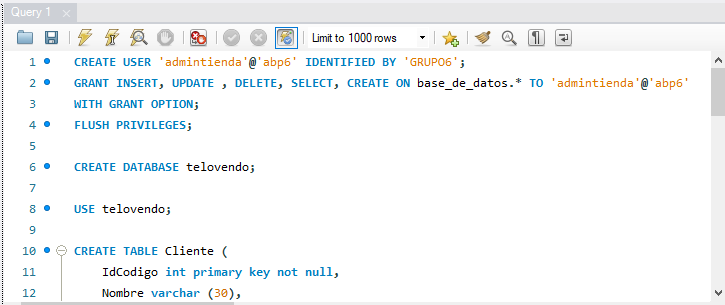
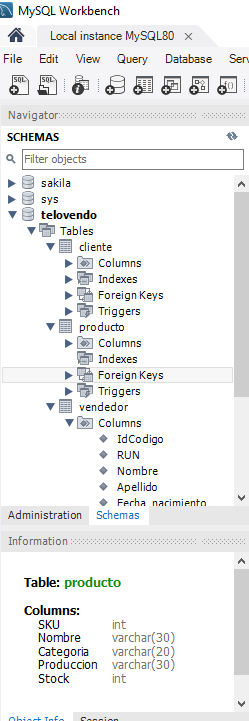
);

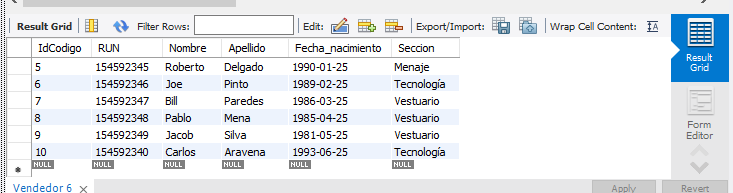
1. ¿Qué es una primary key? y una foreign key?

|  |  |
| --- | --- |
| Primary Key: | Foreign Key |
| se utiliza una clave principal para garantizar que los datos de la columna específica sean únicos. Es una columna que no puede tener valores NULL. Es una columna de tabla existente o una columna generada específicamente por la base de datos de acuerdo con una secuencia definida. | una clave externa es una columna o grupo de columnas en una tabla de base de datos relacional que proporciona un vínculo entre los datos de dos tablas. Es una columna (o columnas) que hace referencia a una columna (la mayoría de las veces, la clave principal) de otra tabla. |
| Se utiliza una clave principal para garantizar que los datos de la columna específica sean únicos. | Una clave externa es una columna o grupo de columnas en una tabla de base de datos relacional que proporciona un vínculo entre los datos de dos tablas. |
| Identifica de forma única un registro en la tabla de la base de datos relacional. | Hace referencia al campo de una tabla que es la clave principal de otra tabla. |
| Solo se permite una clave principal en una tabla. | Mientras que más de una clave externa está permitida en una tabla. |
| Es una combinación de restricciones ÚNICAS y No nulas. | Puede contener valores duplicados y una tabla en una base de datos relacional. |
| No permite valores NULL. | También puede contener valores NULL. |
| Su valor no se puede eliminar de la tabla principal. | Su valor se puede eliminar de la tabla secundaria. |
| Su restricción se puede definir implícitamente en las tablas temporales. | Su restricción no se puede definir en las tablas temporales locales o globales. |

1. ¿Qué diferencias hay entre utilizar MySQL shell y MySQL Workbench?

* **MySQL Workbench** es el entorno integrado oficial de **MySQL**. Fue desarrollado por **MySQL** AB, y permite a los usuarios administrar gráficamente las bases de datos **MySQL** y diseñar visualmente las estructuras de las bases de datos. **MySQL Workbench** reemplaza el anterior paquete de software, **MySQL** GUI Tools.
* ***MySQL Shell*** es un cliente avanzado y un editor de código para MySQL. Este documento describe las características principales de MySQL Shell. Además de la funcionalidad SQL provista, similar a **mysql** , MySQL Shell brinda capacidades de secuencias de comandos para JavaScript y Python e incluye API para trabajar con MySQL. X DevAPI le permite trabajar con datos tanto relacionales como de documentos, consulte [Uso de MySQL como almacén de documentos](https://dev.mysql.com/doc/refman/8.0/en/document-store.html) . AdminAPI le permite trabajar con InnoDB Cluster, InnoDB ClusterSet e InnoDB ReplicaSet.





CREATE USER 'admintienda'@'abp6' IDENTIFIED BY 'GRUPO6';

GRANT INSERT, UPDATE , DELETE, SELECT, CREATE ON base\_de\_datos.\* TO 'admintienda'@'abp6'

WITH GRANT OPTION;

FLUSH PRIVILEGES;

CREATE DATABASE telovendo;

USE telovendo;

CREATE TABLE Cliente (

IdCodigo int primary key not null,

Nombre varchar (30),

Apellido varchar (30),

Telefono int not null,

Direccion varchar (60),

Comuna varchar (30),

Fecha\_Registro date not null

);

CREATE TABLE Producto (

SKU int,

Nombre varchar (30),

Categoria varchar (20),

Produccion varchar (30),

Stock int

);

CREATE TABLE Vendedor (

IdCodigo int primary key not null,

RUN int not null,

Nombre varchar (40),

Apellido varchar (40),

Fecha\_nacimiento date,

Seccion varchar (20)

);

INSERT INTO Vendedor (IdCodigo, RUN, Nombre, Apellido, Fecha\_nacimiento, Seccion)

VALUES ('1', 154592341, 'Mario', 'Sepulveda', '1979-09-25', 'Zapateria'),

('2', 154592342, 'Pedro', 'Abarca', '1989-10-25', 'Jugueteria'),

('3', 154592343, 'Juan', 'Bravo', '1981-11-25', 'Zapateria'),

('4', 154592344, 'Santiago', 'Olivares', '1986-08-25', 'Zapateria'),

('5', 154592345, 'Roberto', 'Delgado', '1990-01-25', 'Menaje'),

('6', 154592346, 'Joe', 'Pinto', '1989-02-25', 'Tecnología'),

('7', 154592347, 'Bill', 'Paredes', '1986-03-25', 'Vestuario'),

('8', 154592348, 'Pablo', 'Mena', '1985-04-25', 'Vestuario'),

('9', 154592349, 'Jacob', 'Silva', '1981-05-25', 'Vestuario'),

('10', 154592340, 'Carlos', 'Aravena', '1993-06-25', 'Tecnología');

SELECT \* FROM Vendedor;

SHOW DATABASES;

USE telovendo;

SELECT DATABASE();

SELECT \* from INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'Vendedor'