Parte 1: Crear entorno de trabajo

- Crear una base de datos

- Crear un usuario con todos los privilegios para trabajar con la base de datos recién creada.

Parte 2: Crear dos tablas.

- La primera almacena a los usuarios de la aplicación (id\_usuario, nombre, apellido,

contraseña, zona horaria (por defecto UTC-3), género y teléfono de contacto).

- La segunda tabla almacena información relacionada a la fecha-hora de ingreso de los

usuarios a la plataforma (id\_ingreso, id\_usuario y la fecha-hora de ingreso (por defecto la

fecha-hora actual)).

Parte 3: Modificación de la tabla

- Modifique el UTC por defecto.Desde UTC-3 a UTC-2.

Parte 4: Creación de registros.

- Para cada tabla crea 8 registros.

Parte 5: Justifique cada tipo de dato utilizado. ¿Es el óptimo en cada caso?

Parte 6: Creen una nueva tabla llamada Contactos (id\_contacto, id\_usuario, numero de telefono,

correo electronico).

Parte 7: Modifique la columna teléfono de contacto, para crear un vínculo entre la tabla Usuarios y la

tabla Contactos.

El ejercicio debe ser subido a github y al Nodo Virtual.

—------------------------------------------------------------------

Parte 1: Crear entorno de trabajo

CREATE DATABASE telovendobbdd;

CREATE USER 'nombre\_de\_usuario'@'localhost' IDENTIFIED BY 'contraseña';

GRANT ALL PRIVILEGES ON telovendobbdd TO 'nombre\_de\_usuario'@'localhost';

FLUSH PRIVILEGES;

Parte 2: Crear dos tablas.

CREATE TABLE usuarios (

id\_usuario INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

nombre VARCHAR(50) NOT NULL,

apellido VARCHAR(50) NOT NULL,

contraseña VARCHAR(100) NOT NULL,

zona\_horaria VARCHAR(50) DEFAULT 'UTC-3',

género VARCHAR(20),

telefono\_contacto INT

);

CREATE TABLE ingresos (

id\_ingreso INT PRIMARY KEY NOT NULL,

id\_usuario INT NOT NULL,

fecha\_hora\_ingreso DATETIME DEFAULT CURRENT\_TIMESTAMP NOT NULL,

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario)

);

Parte 3: Modificación de la tabla

ALTER TABLE usuarios

ALTER COLUMN zona\_horaria SET DEFAULT 'UTC-2';

Parte 4: Creación de registros.

INSERT INTO usuarios (id\_usuario, nombre, apellido, contraseña, zona\_horaria, género, telefono\_contacto)

VALUES

(1, 'Juan', 'Pérez', 'contra01', 'UTC-2', 'Masculino', '955683302'),

(2, 'María', 'Gómez', 'contra02', 'UTC-2', 'Femenino', '955683301'),

(3, 'Carlos', 'López', 'contra03', 'UTC-2', 'Masculino', '955683303'),

(4, 'Ana', 'Rodríguez', 'contra04', 'UTC-2', 'Femenino', '955683304'),

(5, 'Pedro', 'Martínez', 'contra05', 'UTC-2', 'Masculino', '955683305'),

(6, 'Laura', 'Sánchez', 'contra06', 'UTC-2', 'Femenino', '955683306'),

(7, 'Miguel', 'Torres', 'contra07', 'UTC-2', 'Masculino', '955683307'),

(8, 'Sofía', 'Ramírez', 'contra08', 'UTC-2', 'Femenino', '955683308');

INSERT INTO ingresos (id\_ingreso, id\_usuario, fecha\_hora\_ingreso)

VALUES

(1, 1, '2023-05-02 10:00:00'),

(2, 2, '2023-05-02 11:30:00'),

(3, 3, '2023-05-02 12:15:00'),

(4, 4, '2023-05-02 14:45:00'),

(5, 5, '2023-05-02 16:20:00'),

(6, 6, '2023-05-02 17:10:00'),

(7, 7, '2023-05-02 19:30:00'),

(8, 8, '2023-05-02 20:45:00');

Parte 5: Justifique cada tipo de dato utilizado. ¿Es el óptimo en cada caso?

Para la tabla de usuarios:

* id\_usuario (INT): Se utiliza el tipo de dato INT para representar un identificador numérico único para cada usuario.
* Nombre (VARCHAR): Se utiliza el tipo de dato VARCHAR para almacenar el nombre del usuario, que es una cadena de caracteres de longitud variable.
* Apellido (VARCHAR): Al igual que el campo nombre, se utiliza el tipo de dato VARCHAR para almacenar el apellido del usuario.
* Contraseña (VARCHAR): Se utiliza el tipo de dato VARCHAR para almacenar la contraseña del usuario.
* zona\_horaria (VARCHAR): Se utiliza el tipo de dato VARCHAR para almacenar la zona horaria del usuario. El uso de VARCHAR es adecuado para almacenar cadenas de texto de longitud variable, como los nombres de las zonas horarias.
* Género (VARCHAR): Se utiliza el tipo de dato VARCHAR para almacenar el genero del usuario.
* Telefono\_contacto (INT): se utiliza el tipo de dato INT para representar un identificador numérico del telefono de contacto.

Parte 6: Creen una nueva tabla llamada Contactos (id\_contacto, id\_usuario, numero de telefono, correo electronico)

CREATE TABLE Contactos (

id\_contacto INT PRIMARY KEY NOT NULL,

id\_usuario INT NOT NULL,

numero\_telefono INT NOT NULL,

correo\_electronico VARCHAR(100) ,

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario)

);

Parte 7: Modifique la columna teléfono de contacto, para crear un vínculo entre la tabla Usuarios y la tabla Contactos.

Para agregar la columna "id\_contacto" a la tabla "usuarios", se utiliza el comando SQL "ALTER TABLE". La sintaxis para agregar una columna es la siguiente.

En este caso, la sentencia SQL completa para agregar la columna "id\_contacto" a la tabla "usuarios" es la siguiente.

ALTER TABLE usuarios

ADD COLUMN id\_contacto INT,

ADD FOREIGN KEY (id\_contacto) REFERENCES Contactos(id\_contacto);

Una vez agregada la columna "id\_contacto", se utiliza el comando SQL "ALTER TABLE" nuevamente para crear una relación de clave foránea entre la columna "id\_contacto" de la tabla "usuarios" y la columna "id\_contacto" de la tabla "contactos". La sintaxis para crear una clave foránea es la siguiente.

Al crear esta clave foránea, se asegura que cada usuario tenga un único registro de contacto en la tabla "contactos" y que el registro de contacto esté vinculado correctamente con el usuario correspondiente. Esto permite un acceso más eficiente a la información de contacto de cada usuario y evita la duplicación innecesaria de datos en la base de datos.