**EJERCICIO GRUPAL 5 MODULO 5**

**Link GitHub**

https://github.com/jeremy-andres/grupal5.5.git

**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.

create database grupal555;

use grupal555;

create user 'admin5555'@'localhost'

identified by '1234567';

grant all privileges on grupal555.\* to 'admin5555'@'localhost';

**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)).**

*create table usuario*(

id\_usuario int not null auto\_increment,

nombre varchar(20),

apellido varchar(20),

contraseña varchar (10),

zona\_horaria varchar(10) default 'UTC-3',

genero varchar(10),

telefono\_contacto varchar(10),

primary key (id\_usuario)

);

create table ingreso\_usuario(

id\_ingreso int not null auto\_increment,

id\_usuario int,

fecha\_hora\_ingreso datetime default now(),

foreign key(id\_usuario) references usuario(id\_usuario),

primary key (id\_ingreso)

);

**Parte 3: Modificación de la tabla**

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

alter table grupal555.usuario

alter column zona\_horaria set default 'UTC-2';

**Parte 4: Creación de registros.**

**- Para cada tabla crea 8 registros**.

insert into usuario(nombre, apellido, contraseña, zona\_horaria, genero, telefono\_contacto)

values

("Juan", "Pérez", "contra123", "GMT-5", "M", "1234567890"),

("María", "García", "seguro456", "GMT+1", "F", "9876543210"),

("Miguel", "Rodríguez", "p@55w0rd", "GMT-8", "M", "4567890123"),

("Sofía", "Martínez", "secreto789", "GMT-3", "F", "3210987654"),

("David", "Fernández", "123abc", "GMT+2", "M", "5678901234"),

("Olivia", "Torres", "qwerty321", "GMT-6", "F", "8901234567"),

("Daniel", "Gómez", "contra123", "GMT-4", "M", "2345678901"),

("Ava", "Thomas", "seguro123", "GMT+3", "F", "6789012345");

insert into ingreso\_usuario( id\_usuario, fecha\_hora\_ingreso)

values

(1, "2023-01-01"),

(2, "2023-02-15"),

(3, "2023-03-10"),

(4, "2023-04-05"),

(5, "2023-05-20"),

(6, "2023-06-12"),

(7, "2023-07-08"),

(8, "2023-08-25");

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

**Tabla Usuarios:**

***id\_usuario int not null auto\_increment:*** Se utiliza el tipo de dato int para almacenar un

identificador único para cada usuario. Al ser una clave primaria, se utiliza

auto\_increment para generar automáticamente valores crecientes para cada usuario.

***nombre varchar(20) - apellido varchar(20):*** para ambos datos se utiliza el tipo

varchar(20) para almacenar el nombre y apellido de los usuarios. El tamaño máximo se

establece en 20 caracteres pero se puede editar de ser necesario.

contraseña: varchar (10): Se utiliza el tipo de dato varchar(20) para almacenar la

contraseña de los usuarios. Al igual que los campos de nombre y apellido, se establece

un tamaño máximo de 20 caracteres.

***zona\_horaria:*** varchar(10) default 'UTC-3': Se utiliza el tipo de dato varchar(10) para

almacenar la zona horaria de los usuarios. Se establece como not null para garantizar

que siempre se proporcione un valor y se establece el valor predeterminado como

'UTC-3'.

**género varchar(5):** Se utiliza el tipo de dato varchar(5) para almacenar el género de los

usuarios. Se establece un tamaño máximo de 5 caracteres, correspondiendo a inicial de

**la palabra.**

**teléfono contacto:** varchar(10): Se utiliza el tipo de dato varchar(10) para almacenar el

número de teléfono de los usuarios. Se establece un tamaño máximo de 10 caracteres,

pudiendo incluir además de números, algún otro carácter.

**Tabla Ingreso Usuarios:**

***id\_ingreso int not null auto\_increment***: Se utiliza el tipo de dato int para almacenar un

identificador único para cada ingreso de usuario. Al ser una clave primaria, se utiliza

auto\_increment para generar automáticamente valores crecientes para cada usuario.

id\_usuario int: Se utiliza el tipo de dato int para almacenar el identificador del usuario

que realizó el ingreso. Esta columna establece una relación con la tabla usuario a través

de la clave foránea (foreign key).

***fecha\_hora\_ingreso datetime default now()***: Se utiliza el tipo de dato datetime para

almacenar la fecha y hora del ingreso del usuario. Se establece el valor predeterminado

como la fecha y hora actual (NOW()).

**Tabla Contactos:**

***id\_contacto int not null auto\_increment***: Se utiliza el tipo de dato int para almacenar un

identificador único para cada contacto. Al ser una clave primaria, se utiliza

auto\_increment para generar automáticamente valores crecientes para cada contacto.

id\_usuario int: Se utiliza el tipo de dato int para almacenar el identificador del usuario

que realizó el ingreso y que se añade a la tabla de contactos. Esta columna establece

una relación con la tabla usuario a través de la clave foránea (foreign key).

***número\_telefono varchar(10):*** Se utiliza el tipo de dato varchar(10) para almacenar el

número de teléfono de los usuarios. Se establece un tamaño máximo de 10 caracteres.

correo\_electronico varchar(25): Se utiliza el tipo de dato varchar(25) para almacenar el

correo electrónico de los contactos. Se establece un tamaño máximo de 25 caracteres.

**Parte 6: Creen una nueva tabla llamada Contactos (id\_contacto, id\_usuario, número de teléfono,**

**correo electrónico).**

create table contactos(

id\_contacto int not null auto\_increment,

id\_usuario int,

numero\_telefono varchar(10),

correo\_electronico varchar(25),

foreign key(id\_usuario) references usuario(id\_usuario),

primary key (id\_contacto)

);

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

**tabla Contactos.**

UPDATE contactos

SET numero\_telefono = (

SELECT telefono\_contacto

FROM usuario

WHERE usuario.id\_usuario = contactos.id\_usuario

);

/\* En este caso se importaro el id de usuario de la tabla usuarios, y el numero de telefono de la misma tabla. La columna “correo\_electronico”, quedo vacía, ya que no se pedia completar. Pero de todas formas se pueden ingresar a futuro, ya sea modificando cada registro o mediante alguna carga de datos externa\*/