

Script de la base de datos del proyecto

Danny Julián Perilla Mikán

Jesús David Garcés Díaz

Yesica Paola Garzón Plazas

Enero, 2025

1 Introducción

El código SQL (Structured Query Language) es un lenguaje estándar utilizado para interactuar con bases de datos relacionales. Permite realizar diversas operaciones como crear, modificar, eliminar y consultar tablas y registros dentro de una base de datos. A través de comandos como SELECT, INSERT, UPDATE y DELETE, los usuarios pueden gestionar la información de manera eficiente. SQL también soporta la definición de estructuras mediante CREATE y ALTER, así como el manejo de permisos con GRANT y REVOKE. Su amplio uso lo convierte en una herramienta fundamental en el desarrollo y administración de aplicaciones basadas en datos.

2 Script

```
1 CREATE DATABASE turismo_web;
2 USE turismo_web;
3
4 -- Tabla de Roles
5 CREATE TABLE roles (
6     idRol INT PRIMARY KEY AUTO_INCREMENT,
7     nombreRol VARCHAR(50) UNIQUE NOT NULL
8 );
9
10 -- Tabla de Países
11 CREATE TABLE paises (
12     idPais INT PRIMARY KEY AUTO_INCREMENT,
13     nombrePais VARCHAR(50) NOT NULL UNIQUE
14 );
15
16 -- Tabla de Destinos
17 CREATE TABLE destinos (
18     idDestino INT PRIMARY KEY AUTO_INCREMENT,
19     ciudadDestino VARCHAR(100) NOT NULL,
20     departamentoDestino VARCHAR(100) NOT NULL,
21     idPais INT,
22     descripcion TEXT,
23     FOREIGN KEY (idPais) REFERENCES paises(idPais)
24 );
25
```

Análisis y Desarrollo de Software

Ficha: 2791446

Competencia: construcción de software (220501096)

Evidencia: GA6-220501096-AA2-EV03.

Link: [github/dmikan](https://github.com/dmikan)

```
26 -- Tabla de Tipos de Documento
27 CREATE TABLE tiposDocumento (
28     idTipoDocumento INT PRIMARY KEY AUTO_INCREMENT,
29     nombreTipo VARCHAR(50) NOT NULL UNIQUE
30 );
31
32 -- Tabla de Usuarios
33 CREATE TABLE usuarios (
34     idUsuario INT PRIMARY KEY AUTO_INCREMENT,
35     nombre VARCHAR(100) NOT NULL,
36     apellido VARCHAR(100) NOT NULL,
37     email VARCHAR(100) UNIQUE NOT NULL,
38     passwordHash VARCHAR(255) NOT NULL,
39     telefono VARCHAR(15),
40     fechaRegistro DATE NOT NULL,
41     -- idRol INT,
42     idTipoDocumento INT,
43     numeroIdentificacion VARCHAR(20) NOT NULL,
44     -- FOREIGN KEY (idRol) REFERENCES roles(idRol),
45     FOREIGN KEY (idTipoDocumento) REFERENCES tiposDocumento(idTipoDocumento)
46 );
47
48 -- Tabla de Permisos
49 CREATE TABLE permisos (
50     idPermiso INT PRIMARY KEY AUTO_INCREMENT,
51     nombrePermiso VARCHAR(100) NOT NULL
52 );
53
54 -- Tabla de Roles_Permisos
55 CREATE TABLE rolesPermisos (
56     idRol INT,
57     idPermiso INT,
58     PRIMARY KEY (idRol, idPermiso),
59     FOREIGN KEY (idRol) REFERENCES roles(idRol),
60     FOREIGN KEY (idPermiso) REFERENCES permisos(idPermiso)
61 );
62
63 -- Tabla de Tours
64 CREATE TABLE tours (
65     idTour INT PRIMARY KEY AUTO_INCREMENT,
66     nombreTour VARCHAR(100) NOT NULL,
67     idDestino INT,
68     precio DECIMAL(10, 2) NOT NULL,
69     descripcion TEXT,
70     fechaInicio DATE NOT NULL,
71     fechaFin DATE NOT NULL,
72     FOREIGN KEY (idDestino) REFERENCES destinos(idDestino)
73 );
74
75 -- Tabla de Estados de Reservacion
76 CREATE TABLE estadosReservacion (
77     idEstadoReservacion INT PRIMARY KEY AUTO_INCREMENT,
78     nombreEstado VARCHAR(50) NOT NULL UNIQUE
79 );
80
81 -- Tabla de Reservaciones
82 CREATE TABLE reservaciones (
```

Análisis y Desarrollo de Software

Ficha: 2791446

Competencia: construcción de software (220501096)

Evidencia: GA6-220501096-AA2-EV03.

Link: [github/dmikan](https://github.com/dmikan)

```
83     idReservacion INT PRIMARY KEY AUTO_INCREMENT,
84     idUsuario INT,
85     idTour INT,
86     fechaReservacion DATE NOT NULL,
87     numeroPersonas INT NOT NULL,
88     idEstadoReservacion INT,
89     FOREIGN KEY (idUsuario) REFERENCES usuarios(idUsuario),
90     FOREIGN KEY (idTour) REFERENCES tours(idTour),
91     FOREIGN KEY (idEstadoReservacion) REFERENCES estadosReservacion(idEstadoReservacion)
92 );
93
94 -- Tabla de Estados de Pago
95 CREATE TABLE estadosPago (
96     idEstadoPago INT PRIMARY KEY AUTO_INCREMENT,
97     nombreEstado VARCHAR(50) NOT NULL UNIQUE
98 );
99
100 -- Tabla de Metodos de Pago
101 CREATE TABLE metodosPago (
102     idMetodoPago INT PRIMARY KEY AUTO_INCREMENT,
103     nombreMetodo VARCHAR(50) NOT NULL UNIQUE
104 );
105
106 -- Tabla de Pagos
107 CREATE TABLE pagos (
108     idPago INT PRIMARY KEY AUTO_INCREMENT,
109     idReservacion INT,
110     monto DECIMAL(10, 2) NOT NULL,
111     fechaPago DATE NOT NULL,
112     idMetodoPago INT,
113     idEstadoPago INT,
114     FOREIGN KEY (idReservacion) REFERENCES reservaciones(idReservacion),
115     FOREIGN KEY (idMetodoPago) REFERENCES metodosPago(idMetodoPago),
116     FOREIGN KEY (idEstadoPago) REFERENCES estadosPago(idEstadoPago)
117 );
118
119
120 -- Queries
121 SELECT * FROM tiposDocumento;
122 SELECT * FROM usuarios;
```

3 Conclusiones

Este script define una base de datos relacional para un sistema de turismo, creando tablas bien estructuradas con claves primarias y foráneas que aseguran la integridad referencial entre entidades como usuarios, roles, destinos, tours y pagos. La base de datos está diseñada para manejar distintos tipos de documentos, métodos de pago, permisos y roles, facilitando una gestión flexible de usuarios y sus accesos. Además, incluye tablas especializadas para reservaciones y pagos, lo que permite llevar un control detallado del proceso de compra de tours. La correcta normalización evita redundancias y garantiza un almacenamiento eficiente. Finalmente, el uso de restricciones UNIQUE y NOT NULL ayuda a mantener la consistencia y calidad de los datos.