

**AVANCE 03**

**GRUPO 6**

Integrantes:

JIMENEZ VELOZ FRANCIS RAPHAEL

LOPEZ GONZABAY GERSON ANDRES

MEZA SOLIS SAMUEL DIEGO

SALAZAR GORDILLO ELKIN MATIAS

Contenido

[**Triggers en la Base de Transportes** 3](#_Toc188221553)

[**Reportes (Vistas) en la Base Transporte** 10](#_Toc188221554)

[**Stored Procedures en la Base Transportes** 12](#_Toc188221555)

[**Tabla de permisos de Usuario de la Base Transporte** 13](#_Toc188221556)

[**Diagrama del schema de Base Transporte** 14](#_Toc188221557)

[**Manual de Usuario - Sistema CRUD para Gestión de Clientes** 15](#_Toc188221558)

[**Requisitos del sistema** 15](#_Toc188221559)

[**Instalación del conector MySQL** 15](#_Toc188221560)

[**Configuración inicial** 15](#_Toc188221561)

[**1. Crear la base de datos y la tabla** 15](#_Toc188221562)

[**2. Configurar la conexión en el código Python** 16](#_Toc188221563)

[**Uso del sistema** 16](#_Toc188221564)

[**1. Ejecutar el programa** 16](#_Toc188221565)

[**2. Menú principal** 16](#_Toc188221566)

[**Opciones del menú** 16](#_Toc188221567)

[**Notas adicionales** 17](#_Toc188221568)

[**Contacto** 18](#_Toc188221569)

### **Triggers en la Base de Transportes**

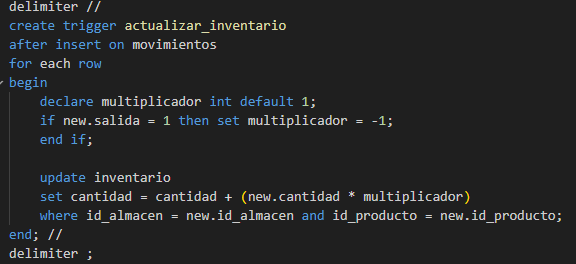
Los triggers son procedimientos almacenados que se ejecutan automáticamente al producirse eventos específicos en las tablas de la base de datos. A continuación, se describen los triggers implementados:

Trigger: actualizar\_inventario

Tabla asociada: movimientos

Evento: AFTER INSERT

Propósito: Actualizar automáticamente el inventario al registrar un movimiento, ya sea de entrada o salida.

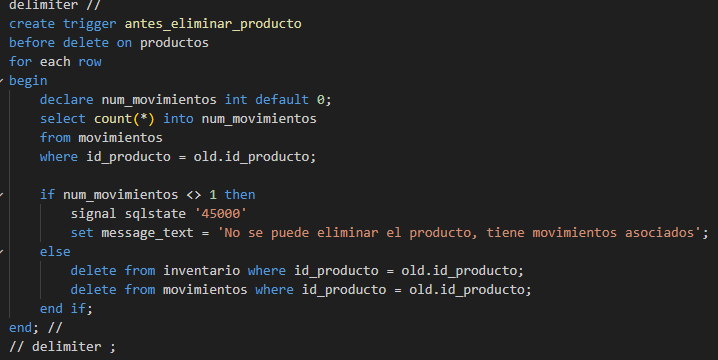
Código: 

Trigger: antes\_eliminar\_producto

Tabla asociada: productos

Evento: BEFORE DELETE

Propósito: Impedir la eliminación de productos con movimientos asociados y garantizar la integridad del inventario.

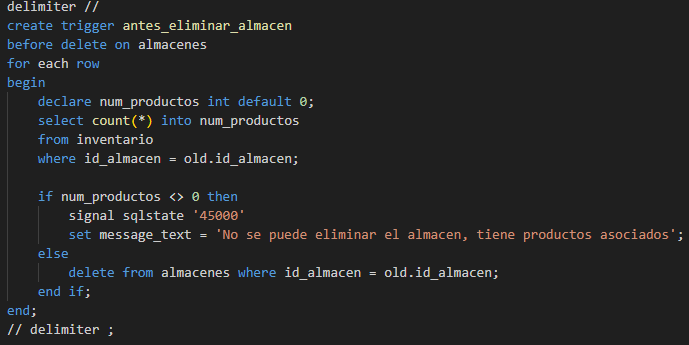
Código: 

Trigger: antes\_eliminar\_almacen

Tabla asociada: almacenes

Evento: BEFORE DELETE

Propósito: Evitar la eliminación de almacenes que contengan productos asociados en el inventario.

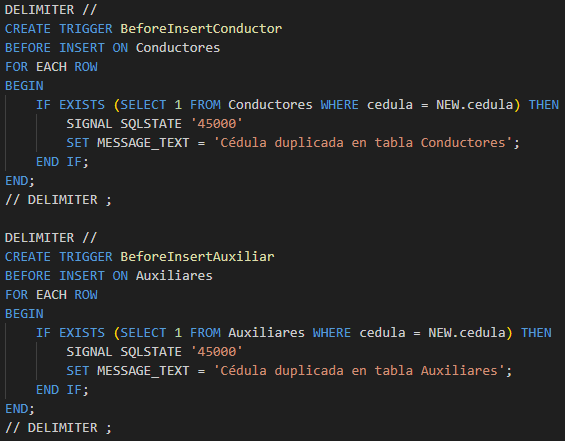
Código: 

Trigger: BeforeInsertConductor

Tabla asociada: Conductores

Evento: BEFORE INSERT

Propósito: Prevenir la inserción de conductores con una cédula duplicada en la tabla.

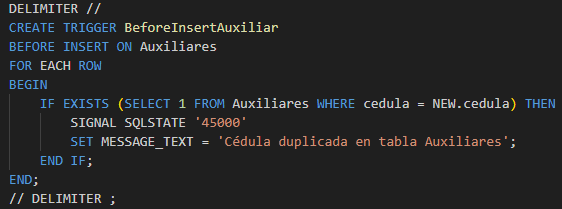
Código: 

Trigger: BeforeInsertAuxiliar

Tabla asociada: Auxiliares

Evento: BEFORE INSERT

Propósito: Prevenir la inserción de auxiliares con una cédula duplicada.

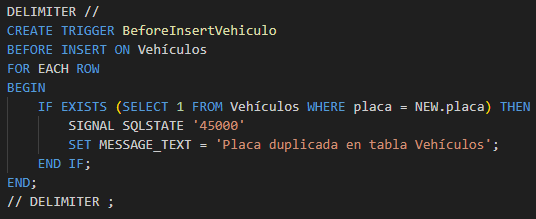
Código: 

Trigger: BeforeInsertVehiculo

Tabla asociada: Vehículos

Evento: BEFORE INSERT

Propósito: Prevenir la inserción de vehículos con una placa duplicada.

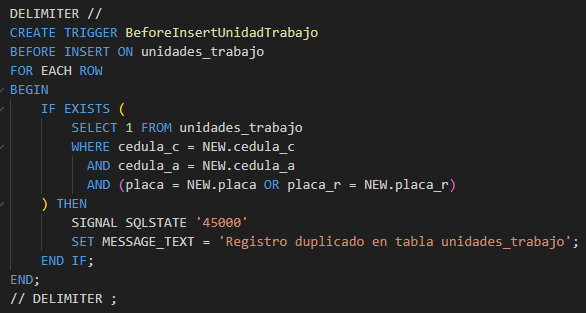
Código: 

Trigger: BeforeInsertUnidadTrabajo

Tabla asociada: unidades\_trabajo

Evento: BEFORE INSERT

Propósito: Prevenir la creación de registros duplicados en la tabla de unidades de trabajo, verificando conductores, auxiliares y placas.

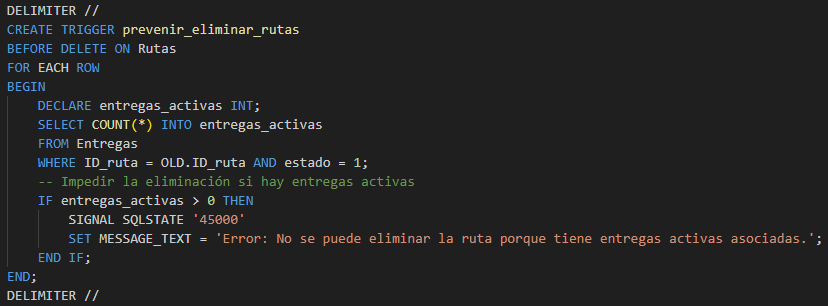
Código: 

Trigger: prevenir\_eliminar\_rutas

Tabla asociada: Rutas

Evento: BEFORE DELETE

Propósito: Impedir la eliminación de rutas que tengan entregas activas asociadas.

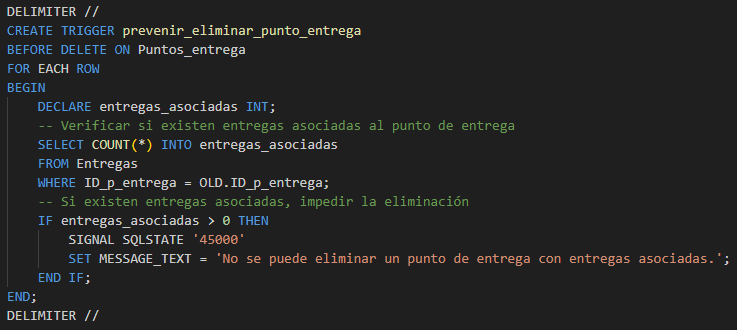
Código: 

Trigger: prevenir\_eliminar\_punto\_entrega

Tabla asociada: Puntos\_entrega

Evento: BEFORE DELETE

Propósito: Evitar la eliminación de puntos de entrega con entregas asociadas.

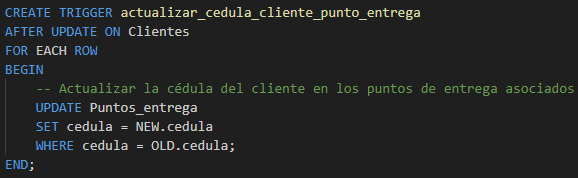
Código: 

Trigger: actualizar\_cedula\_cliente\_punto\_entrega

Tabla asociada: Clientes

Evento: AFTER UPDATE

Propósito: Actualizar automáticamente la cédula de los clientes en los puntos de entrega asociados cuando se modifique.

Código: 

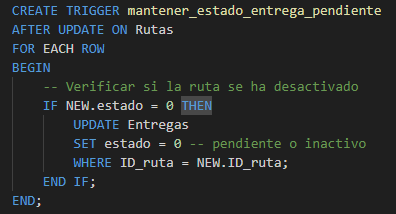
Trigger: mantener\_estado\_entrega\_pendiente

Tabla asociada: Rutas

Evento: AFTER UPDATE

Propósito: Mantener el estado de las entregas como pendiente o inactivo si la ruta se desactiva.

Código:

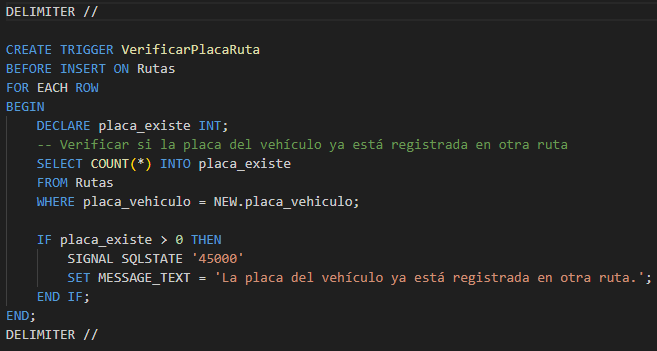


Trigger: VerificarPlacaRuta

Tabla asociada: Rutas

Evento: BEFORE INSERT

Propósito: Prevenir la inserción de una ruta si la placa del vehículo ya está registrada en otra ruta.

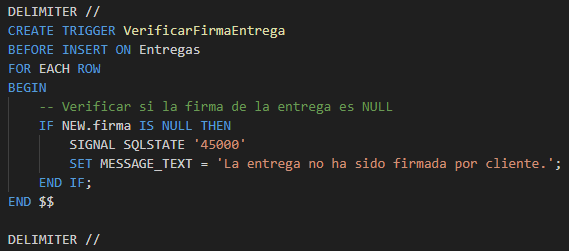
Código: 

Trigger: VerificarFirmaEntrega

Tabla asociada: Entregas

Evento: BEFORE INSERT

Propósito: Evitar la creación de entregas sin una firma de cliente.

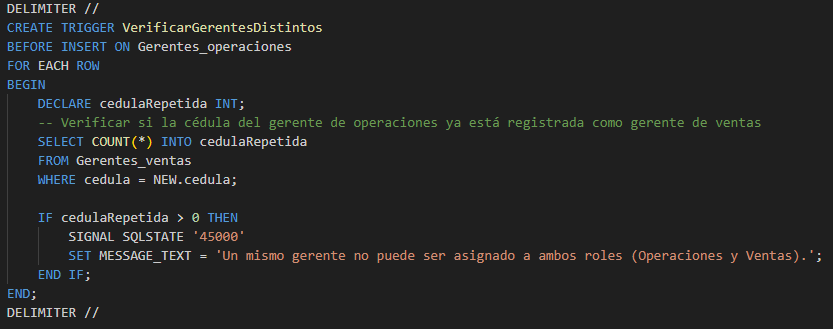
Código: 

Trigger: VerificarGerentesDistintos

Tabla asociada: Gerentes\_operaciones

Evento: BEFORE INSERT

Propósito: Impedir que un gerente sea asignado a ambos roles, operaciones y ventas.

Código: 

### **Reportes (Vistas) en la Base Transporte**

**1.ReporteProductosAlmacen**

Esta vista proporciona información detallada sobre los productos en los almacenes, incluyendo su nombre, descripción, tipo, valor, cantidad, ubicación (almacén) y los datos del gerente responsable (cédula, primer nombre y apellido).

Uso: Utilizada para obtener un panorama general de los productos almacenados, sus cantidades y las responsabilidades de los gerentes de cada almacén. Es útil visualizar inventarios y gestión de recursos.

**2. ReporteMovimientosProductos**

Esta vista rastrea los movimientos de los productos (entrada o salida) en los almacenes, mostrando la cantidad de productos movidos, su estado (entrada o salida), y la cantidad actual en inventario. También incluye la información del almacén donde ocurrió el movimiento.

Uso: Se utiliza para el control de inventarios y gestión de productos, permitiendo saber cuándo y cuánto producto ha sido movido y el estado actual de cada artículo.

**3. ReporteTotalProductos**

Muestra el total de cada producto en inventario, sumando las cantidades de los productos por cada proveedor. También incluye el nombre del proveedor y sus datos de contacto (razón social, dirección, teléfono, y correo).

Uso: Es útil para conocer el stock total de los productos y las relaciones con los proveedores, ayudando a gestionar las compras y ventas.

**4. ReporteProductosEntrega**

Esta vista proporciona información sobre las entregas de productos a clientes, incluyendo el ID de la entrega, el nombre del producto, el valor y la cantidad entregada, así como los datos del cliente (razón social).

Uso: Se utiliza para rastrear las entregas realizadas a clientes, útil para la gestión de ventas y la planificación de futuras entregas.

**5. ReporteRutasActivas**

Muestra las rutas activas, aquellas que están dentro del horario actual, con la distancia y los horarios de inicio y finalización. Filtra solo aquellas rutas cuyo estado es "activo".

Uso: Es útil para la planificación y monitoreo de las rutas activas en tiempo real, permitiendo una gestión eficiente de los recursos de transporte.

**6. ReporteEntregasConFirma**

Proporciona un listado de entregas realizadas que han sido firmadas por el cliente, incluyendo la fecha de entrega, la firma, el valor total y los datos del cliente (razón social).

Uso: Ayuda a verificar las entregas completadas y validarlas con la firma del cliente.

**7. ReporteGerentesRutasAsignadas**

Muestra los gerentes de operaciones y las rutas asignadas a cada uno, incluyendo la cédula del gerente y la distancia de la ruta.

Uso: Utilizada para asignar y verificar las rutas a los gerentes de operaciones

**8. ReporteEntregasPorClienteYEstado**

Presenta las entregas de productos ordenadas por cliente y fecha de entrega, incluyendo el estado y el valor total de la entrega.

Uso: Es útil para organizar el historial de entregas por cliente y evaluar el estado de las mismas, facilitando el análisis y la toma de decisiones de seguimiento.

**9. ReporteRutasConEntregasPendientes**

Lista las rutas activas que tienen entregas pendientes, es decir, aquellas entregas que aún no se han firmado por el cliente.

Uso: Permite gestionar las rutas con entregas pendientes y priorizar las acciones de entrega, asegurando la eficiencia en la operación.

**10. ReporteGerentesVentasPorCliente**

Muestra los gerentes de ventas asignados a cada cliente, junto con los datos del cliente (ID y razón social).

Uso: Esta vista es útil para realizar un seguimiento de las relaciones comerciales, analizando qué gerentes están trabajando con qué clientes.

**11. ReporteResumenEntregasPorFecha**

Proporciona un resumen de entregas agrupadas por fecha, mostrando la cantidad total de entregas y el valor total por fecha.

Uso: Es útil para generar reportes diarios de entregas, analizando el volumen de entregas y su valor total para gestionar el rendimiento de las operaciones.

**12. DetallesPersonal**

Ofrece detalles completos de los conductores y auxiliares, incluyendo cédula, nombres, teléfonos, correos, licencias y experiencia de los conductores, así como los datos del auxiliar (cedula, nombre, teléfono y correo).

Uso: Facilita la consulta de información de los empleados, útil para gestionar el personal de transporte y asignación de recursos.

**13. AsignacionesConductoresAuxiliares**

Muestra las asignaciones de conductores y auxiliares a vehículos específicos, junto con los detalles de los vehículos (placa, capacidad y datalogger).

Uso: Utilizada para asignar y controlar el equipo de transporte, asegurando que los vehículos sean operados por los equipos correctos y verificando las capacidades disponibles.

**14. ConductoresSinVehiculo**

Muestra una lista de los conductores que actualmente no están asignados a ningún vehículo.

Uso: Ayuda a identificar conductores sin asignación de vehículos, lo que facilita la asignación de vehículos disponibles de manera más eficiente.

**15. AuxiliaresSinConductor**

Presenta una lista de auxiliares sin conductor asignado.

Uso: Permite identificar auxiliares disponibles sin asignación, optimizando la asignación de recursos y personal en las rutas.

### **Stored Procedures en la Base Transportes**

**1. agregar\_nuevo\_producto**

Este procedimiento almacena un nuevo producto en la base de datos. Primero, inserta los detalles del producto en la tabla productos (como nombre, descripción, tipo, valor, y el proveedor asociado). Luego, inserta un registro en la tabla inventario para asociar el nuevo producto con un almacén, estableciendo una cantidad inicial de 0. Finalmente, registra un movimiento inicial de 0 cantidad para ese producto en la tabla movimientos, lo que ayuda a mantener un historial de movimientos desde el primer momento en que el producto es ingresado.

Uso: Se utiliza para agregar nuevos productos al inventario, vinculándolos con almacenes específicos y registrando su primer movimiento de entrada.

**2. mostrar\_productos**

Este procedimiento muestra todos los productos disponibles en el sistema, mostrando el nombre, descripción, tipo, valor, y el proveedor asociado. Si el parámetro in\_almacen es diferente de 0, solo se mostrarán los productos que pertenecen a un almacén específico. Si in\_almacen es 0, se mostrarán todos los productos, independientemente del almacén. Se agrupan los resultados por id\_producto para obtener un resumen de la cantidad disponible por producto.

Uso: Se utiliza para consultar los productos en el inventario, con la capacidad de filtrar por almacén. Es útil para obtener informes del inventario total o por almacén específico.

**3. actualizar\_producto**

Este procedimiento actualiza la información de un producto existente, como su nombre, descripción, tipo, valor y proveedor asociado. Utiliza el id\_producto como clave primaria para identificar el producto que se actualizará.

Uso: Se utiliza cuando se necesitan realizar cambios en las características de un producto ya existente. Esto puede ser útil en casos de corrección de datos o actualización de precios y proveedores.

**4. eliminar\_producto**

Descripción: Este procedimiento elimina un producto de la tabla productos según el id\_producto proporcionado.

Uso: Se utiliza para eliminar productos del sistema, lo cual puede ser necesario cuando un producto ya no está disponible

Los procedimientos mostrados pertenecen a la tabla producto, para otras tablas, su estructura y lógica de funcionamiento serán bastante semejantes, variando principalmente en los nombres de las tablas y las columnas de las mismas.

Agregar: Inserción de nuevos registros, similar al agregar\_nuevo\_producto, pero adaptado a las columnas y relaciones de la otra tabla.

Mostrar: Consultas para obtener los registros, como en el mostrar\_productos, con filtros específicos según el parámetro de entrada.

Actualizar: Modificaciones a los registros existentes, como en el actualizar\_producto, ajustado para actualizar la tabla correspondiente.

Eliminar: Eliminación de registros de las tablas, similar a eliminar\_producto.

### **Índices en la Base Transportes**

Índice en la columna telefono de la tabla Conductores: Este índice mejora las búsquedas rápidas por el número de teléfono de los conductores, un campo comúnmente utilizado para localizarlos, optimizando así las consultas basadas en el número telefónico.

Índice en la columna correo de la tabla Conductores: Similar al índice en telefono, este índice facilita las consultas de búsqueda por correo electrónico, un identificador único que puede ser utilizado para encontrar un conductor rápidamente.

Índice en la columna telefono de la tabla Auxiliares: Al igual que para los conductores, tener un índice en el número de teléfono de los auxiliares mejora la eficiencia de las consultas que los localizan por teléfono.

Índice en la columna correo de la tabla Auxiliares: Este índice mejora las búsquedas basadas en el correo electrónico de los auxiliares, lo que es útil si se requiere encontrar o filtrar información a partir de su dirección de correo electrónico.

Índice en la columna nombre de la tabla Productos: Dado que los productos a menudo se buscan por su nombre, este índice mejora el rendimiento de las consultas que filtran por nombre del producto, facilitando búsquedas rápidas y eficientes.

Índice en la columna tipo de la tabla Productos: Las consultas que filtran por el tipo de producto se benefician de este índice, ya que optimiza la recuperación de productos según su tipo, lo que es frecuente en sistemas de gestión de inventarios o ventas.

Índice en la columna razon\_social de la tabla Clientes: El índice en la razón social de los clientes es útil para optimizar las consultas que buscan clientes por su nombre comercial, un campo que frecuentemente se utiliza en las búsquedas.

Índice en la columna telefono de la tabla Clientes: Este índice mejora las búsquedas rápidas de clientes mediante su número telefónico, un dato que suele ser utilizado para contactarlos o para hacer filtrados rápidos en la base de datos.

Índice en la columna correo de la tabla Clientes: Un índice sobre el correo electrónico de los clientes optimiza las consultas que los buscan a través de este identificador único, facilitando la localización rápida de un cliente por su correo electrónico.

Índice en la columna estado de la tabla Entregas: Las consultas que filtran las entregas por su estado (por ejemplo, entregas pendientes o completadas) se benefician de este índice, lo que mejora el rendimiento en la recuperación de registros de entregas basados en su estado.

Índice en la columna fecha\_entrega de la tabla Entregas: Las consultas que requieren filtrar entregas por fecha se benefician de este índice, ya que optimiza las búsquedas relacionadas con fechas específicas de entrega, un proceso común en sistemas logísticos.

Índice en la columna valor\_total de la tabla Entregas: Este índice mejora las consultas que buscan entregas con un valor total específico, optimizando las búsquedas relacionadas con el monto de las entregas y generando reportes más eficientes.

Índice en la columna descripcion de la tabla Reclamos: El índice en la descripción de los reclamos mejora las búsquedas por palabras clave dentro de las descripciones, facilitando la localización de reclamos específicos en sistemas de atención al cliente o gestión de incidencias.



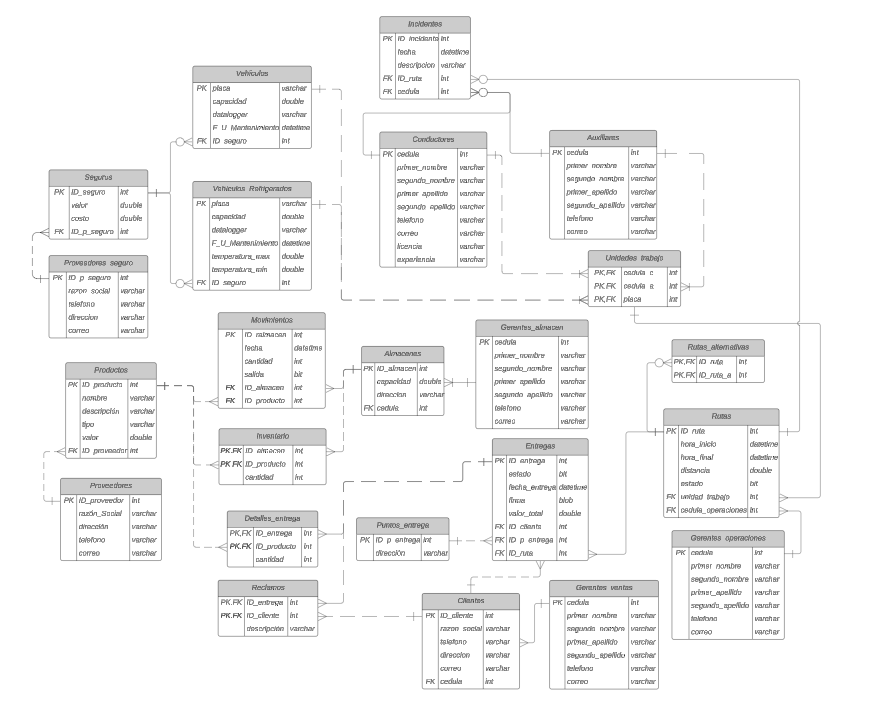




### **Tabla de permisos de Usuario de la Base Transporte**

|  |  |  |
| --- | --- | --- |
| **Usuario** | **Permiso** | **Tabla/Procedimiento** |
| **GersonL** | SELECT | ReporteProductosAlmacen |
| SELECT | ReporteMovimientosProductos |
| EXECUTE | agregar\_nuevo\_producto |
| **ElkinS** | SELECT | ReporteTotalProductos |
| SELECT | ReporteRutasActivas |
| EXECUTE | AddConductor |
| **SamuelM** | SELECT | ReporteProductosEntrega |
| SELECT | ReporteEntregasConFirma |
| EXECUTE | DeleteConductor |
| **FrancisJ** | SELECT | ReporteGerentesRutasAsignadas |
| SELECT | ReporteEntregasPorClienteYEstado |
| EXECUTE | InsertarRuta |
| **IreneCheung** | SELECT | ReporteRutasConEntregasPendientes |
| SELECT | ReporteGerentesVentasPorCliente |
| EXECUTE | registrar\_entrega |

### **Diagrama del schema de Base Transporte**



En el link a continuación se encuentra el pdf con mejor resolución del diagrama:

[Modelo\_logico.pdf](https://espolec-my.sharepoint.com/:b:/g/personal/geranlop_espol_edu_ec/EbHVwzJdaf1HgKtZ-CgdeMoBGGsmovqk6A25DZClNIgBSg?e=64cvA3)

**Manual de Usuario - Sistema CRUD para Gestión de Clientes**

Este manual describe cómo utilizar el sistema CRUD desarrollado en Python con conexión a MySQL para gestionar los registros de clientes. El sistema permite añadir, consultar, editar y eliminar clientes desde una interfaz de consola interactiva.

**Requisitos del sistema**

1. **Python 3.x** instalado.
2. **MySQL Server** instalado y en ejecución.
3. **MySQL Workbench** para gestionar visualmente la base de datos.
4. **Conector MySQL para Python** (mysql-connector-python).

**Instalación del conector MySQL**

Ejecuta el siguiente comando en la terminal para instalar el conector:

pip install mysql-connector-python

**Configuración inicial**

**1. Crear la base de datos y la tabla**

Utiliza el siguiente script SQL: schema.sql y datos.sql para crear la base de datos y la tabla:

CREATE DATABASE transporte;

USE transporte;

CREATE TABLE Clientes (

ID\_cliente INT PRIMARY KEY,

razon\_social VARCHAR(255),

telefono VARCHAR(20),

direccion VARCHAR(255),

correo VARCHAR(100),

cedula INT,

FOREIGN KEY (cedula) REFERENCES Gerentes\_ventas(cedula)

);

INSERT INTO Clientes (ID\_cliente, razon\_social, telefono, direccion, correo, cedula) VALUES

(1, 'Hospital XYZ', '0981234567', 'Calle 123, Quito', 'cliente1@gmail.com', 1723456789),

(2, 'Industria ABC', '0971234568', 'Avenida Siempre Viva', 'cliente2@outlook.com', 1729876543),

(3, 'Empresa Delta', '0969876543', 'Av. Amazonas, Cuenca', 'cliente3@gmail.com', 1712345678);

**2. Configurar la conexión en el código Python**

Edita los parámetros de conexión en el archivo conexion.py para que coincidan con tu configuración de MySQL:

self.host = 'localhost'

self.database = 'transporte'

self.user = 'root'

self.password = 'contraseña'

**Uso del sistema**

**1. Ejecutar el programa**

Para ejecutar el programa, abre una terminal o consola y ejecuta el archivo principal:

python avance.py

**2. Menú principal**

Al iniciar el programa, se mostrará el siguiente menú interactivo:

--- Menú CRUD ---

1. Crear registros

2. Consultar registros

3. Editar registros

4. Eliminar registros

6. Salir

**Opciones del menú**

**Opcion 1: Añadir cliente**

El sistema solicitará los siguientes datos:

* **ID**: Identificador único del cliente.
* **Razón social**: Nombre de la empresa.
* **Teléfono**: Número de contacto.
* **Dirección**: Ubicación del cliente.
* **Correo**: Correo electrónico.
* **Cédula**: Número de cédula del cliente o gerente asociado.

**Ejemplo de entrada:**

ID del cliente: 4

Nombre del cliente: Hospital XYZ

Teléfono: 0981234567

Dirección: Calle Real 123

Correo: cliente4@gmail.com

Cédula: 1729876543

**Opción 2: Consultar clientes**

El sistema mostrará una lista de todos los clientes registrados en la base de datos.

**Ejemplo de salida:**

1. Hospital XYZ, 0981234567, Calle 123, Quito, cliente1@gmail.com, cédula: 1723456789

2. Industria ABC, 0971234568, Avenida Siempre Viva, cliente2@outlook.com, cédula: 1729876543

3. Empresa Delta, 0969876543, Av. Amazonas, Cuenca, cliente3@gmail.com, cédula: 1712345678

**Opción 3: Editar cliente**

El sistema solicitará el ID del cliente a editar, seguido de los nuevos datos:

**Ejemplo de entrada:**

ID del cliente a editar: 2

Nueva razón social: Industria Renovada

Nuevo teléfono: 0945678934

Nueva dirección: Calle Principal 456

Nuevo correo: cliente2actualizado@gmail.com

Nueva cédula: 1729876543

**Opción 4: Eliminar cliente**

El sistema solicitará el ID del cliente a eliminar.

**Ejemplo de entrada:**

ID del cliente a eliminar: 3

**Opción 5: Salir**

Selecciona esta opción para salir del programa.

**Notas adicionales**

* Asegúrate de que el servidor MySQL esté en ejecución antes de iniciar el programa.
* Si hay errores de conexión, verifica que los parámetros de conexión en el archivo conexion.py sean correctos.
* Es recomendable hacer una copia de seguridad de la base de datos antes de realizar cambios importantes.
* Para las demás opciones el proceso es similar cambiando únicamente el tipo de dato que se desea ingresar

**Contacto**

Si tienes alguna pregunta o necesitas soporte, puedes contactar al desarrollador en el correo:

**geranlop@espol.edu.ec**