



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Materia: Bases de Datos

Profesora: M.I Martha López Pelcastre

Grupo: 2

Equipo: 9

PROYECTO FINAL - DOCUMENTO FINAL

Alumno	No. Cuenta
Ayala Hernández María Fernanda	320068234
Jiménez Ayala Yordi Josué	320257599
Portilla Hermenegildo Elizabeth	320278107
Valenzuela Ascencio Gustavo	117002029

Entrega: 25/mayo/2025



Índice

Diseño conceptual	2
Diseño lógico	4
Normalización	16
Consideraciones	19
Diseño físico	19
Anexo	114

AppSafe

AppSafe es una empresa que ofrece servicios de transporte terrestre a través de la creación de una red de conductores que ofrecen servicios de transporte de personas empleando sus propios autos. La empresa desea construir una base de datos para sistematizar sus operaciones (sitio web y a su aplicación móvil). Como proyecto final tuvimos la labor de implementar la base de datos para AppSafe, comenzando desde la definición de la misma con un modelo conceptual hasta su implementación física.

Diseño Conceptual

El diseño conceptual consta únicamente del modelo entidad relación, es importante mencionar que se pasó por una serie de borradores antes de tener una versión final, la siguiente imagen muestra el primer borrador del modelo en cuestión. **(Para mejor observación véase el anexo)**

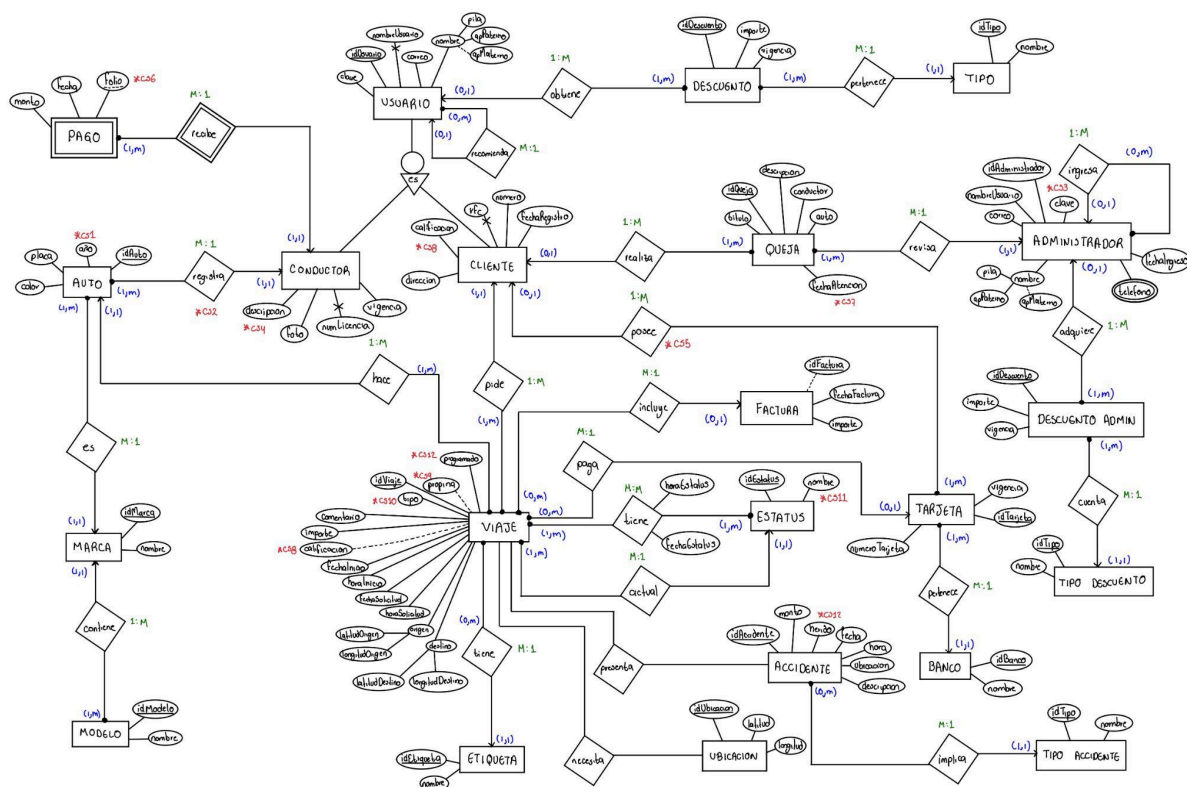


Figura 1. Borrador 1 MER

Posteriormente, luego de una revisión, se realizaron los cambios necesarios para así poder llegar a un borrador final el cual fue aprobado para su elaboración en una herramienta case (DIA en nuestro caso), las siguientes imágenes muestran el borrador final así como el modelo final implementado en DIA. **(Para mejor observación véase el anexo)**

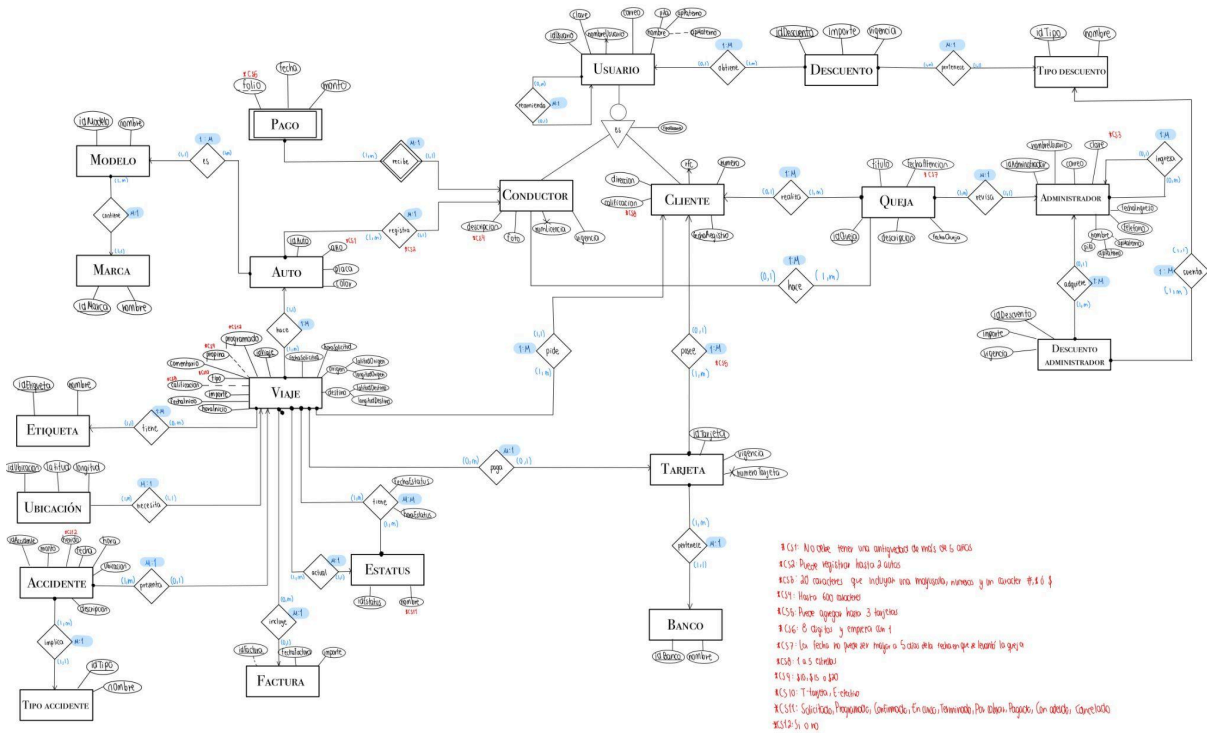


Figura 2. Borrador final MER

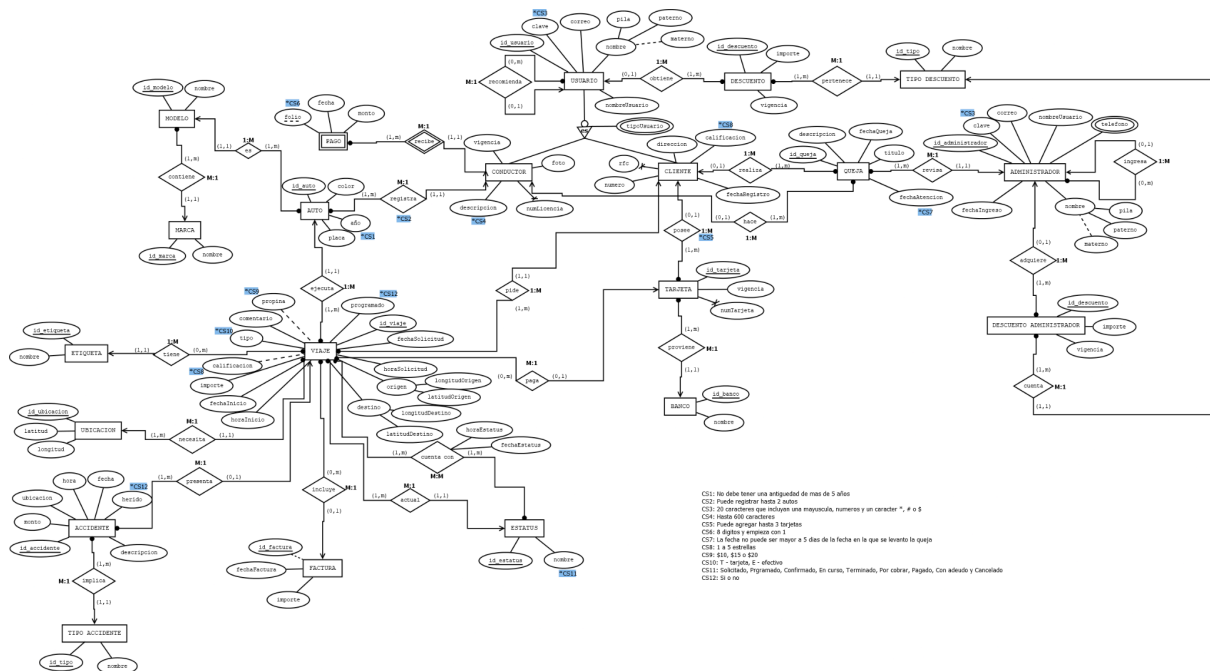


Figura 3. MER en herramienta case

Diseño Lógico

Siguiendo con el desarrollo del proyecto, pasamos a la parte del diseño lógico, esta etapa cuenta con 2 partes, la del modelo relacional y la del diccionario de datos.

a) Modelo relacional

Para esta parte en específico, se realizó una transformación a partir del modelo conceptual previamente realizado, hacer esta transformación implica tener un borrador del modelo relacional para evitar tener errores al momento de estar realizando la misma. La siguiente imagen muestra el borrador realizado. **(Para mejor observación véase el anexo)**

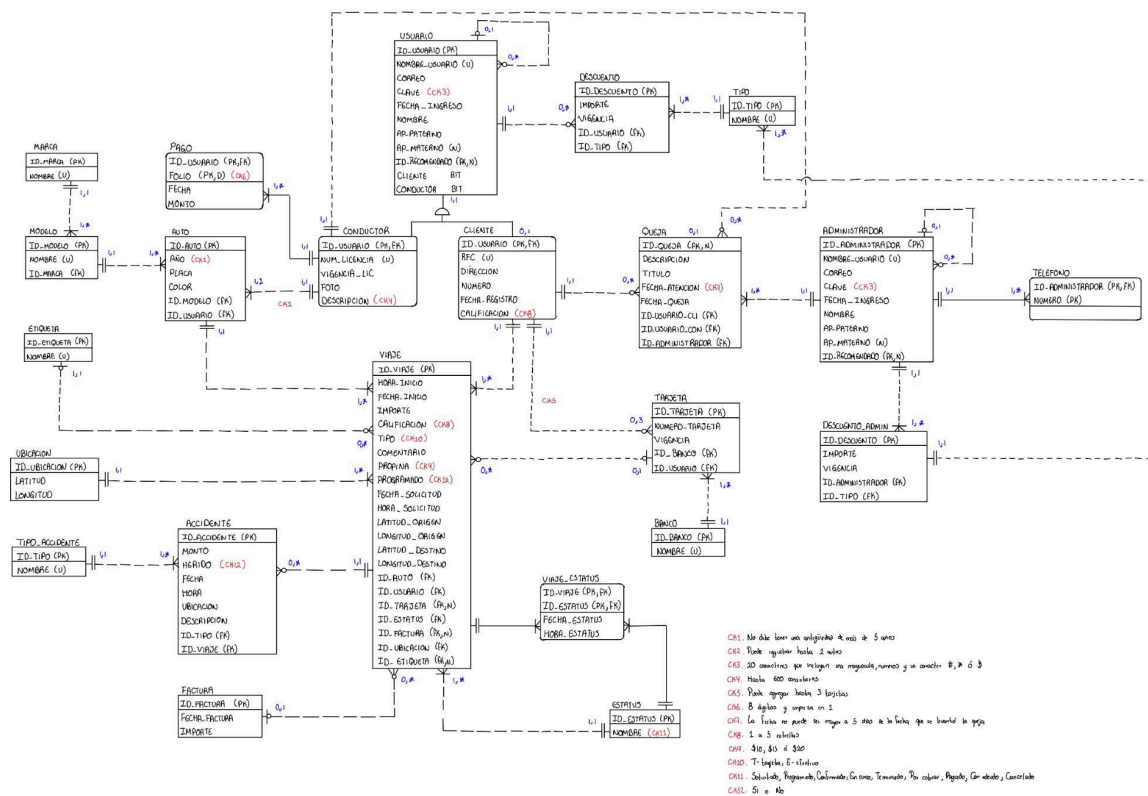


Figura 4. Borrador modelo relacional

Ya con este borrador aprobado, se realizó el mismo en una herramienta seleccionada (ER Studio para nuestro caso). La siguiente imagen muestra el modelo relacional final. **(Para mejor observación véase el anexo)**

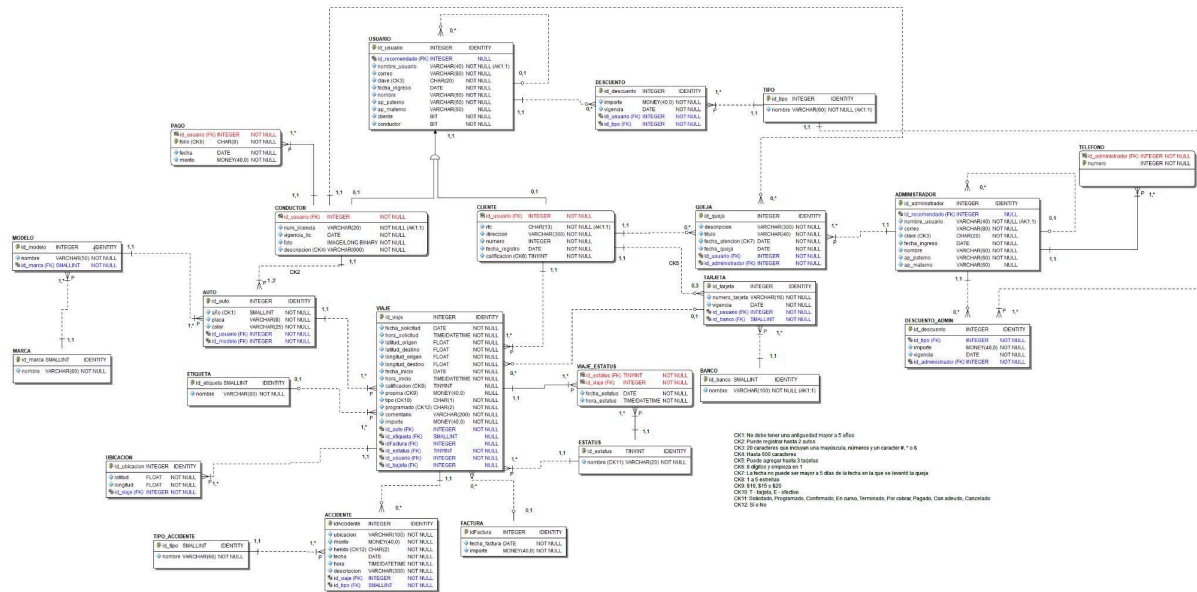


Figura 5. Modelo relacional final

b) Diccionario de datos

A continuación, se anexa el diccionario de datos correspondiente a la base de datos del proyecto, tiene como propósito documentar de manera clara y estructurada la definición de cada una de las tablas que conforman la base de datos, así como sus campos, tipos de datos, claves primarias, foráneas y restricciones. (Nota. en el nombre de las filas ‘T’ significa tipo y ‘L’ significa longitud)

Tabla	Nombre	T	Dato	L	Descripción
-	USUARIO	T	-	-	Tabla que almacena la información del usuario
USUARIO	id_usuario	A	int	10	Almacena una pk, el id de cada usuario
USUARIO	id_recomendado	A	int	10	Almacena una fk, el id del usuario que le recomendó la aplicación
USUARIO	nombre_usuario	A	varchar	40	Almacena el nombre que usará el usuario
USUARIO	correo	A	varchar	80	Almacena el correo del usuario
USUARIO	clave	A	char	20	clave de acceso del usuario
USUARIO	fecha_ingreso	A	date	10	Almacena la fecha en la que se registro el usuario

USUARIO	nombre	A	varchar	60	Almacena el nombre pila del usuario
USUARIO	ap_paterno	A	varchar	60	Almacena el primer apellido del usuario
USUARIO	ap_materno	A	varchar	60	Almacena el segundo apellido del usuario
USUARIO	cliente	A	bit	1	Almacena con un bit si el usuario en cuestión es cliente
USUARIO	conductor	A	bit	1	Almacena con un bit si el usuario en cuestión es conductor
USUARIO	CK3	CK	-	-	Valida 10 caracteres que incluyan una mayúscula, números y un carácter especial *, # ó \$.
-	DESCUENTO	T	-	-	Tabla que almacena la información del descuento
DESCUENTO	id_descuento	A	int	10	Almacena una pk, que identifica cada descuento
DESCUENTO	importe	A	money	20	Almacena el importe monetario descontado al usuario
DESCUENTO	vigencia	A	date	10	Almacena la fecha límite hasta la que se permite utilizar este descuento
DESCUENTO	id_usuario	A	int	10	Almacena una fk, el id del usuario al que pertenece el descuento
DESCUENTO	id_tipo	A	int	10	Almacena una fk, del id que tiene el tipo de descuento en el catálogo de tipo
-	TIPO	T	-	-	Tabla que almacena la información del tipo
TIPO	id_tipo	A	int	10	Almacena una pk, diferencia el tipo de descuento
TIPO	nombre	A	varchar	60	Almacena el nombre del tipo de descuento
-	CONDUCTOR	T	-	-	Tabla que almacena la información del conductor

CONDUCTOR	id_usuario	A	int	10	Almacena una fk, el identificador del usuario (de tipo conductor)
CONDUCTOR	num_licencia	A	int	10	Almacena el número de licencia del conductor
CONDUCTOR	vigencia_lic	A	date	10	Almacena la fecha en la que la licencia se vencerá
CONDUCTOR	foto	A	image	10	Almacena una imagen (long binary) del conductor
CONDUCTOR	descripcion	A	varchar	600	Almacena una descripción del conductor
CONDUCTOR	CK4	CK	-	-	Hasta 600 caracteres
-	PAGO	T	-	-	Tabla que almacena la información del pago
PAGO	id_usuario	A	int	10	Almacena una fk y pk, id del conductor que recibe el pago
PAGO	folio	A	char	8	Almacena una pk, folio de cada pago
PAGO	fecha	A	date	10	Almacena la fecha en la que se realizó el pago
PAGO	monto	A	money	20	Almacena el monto monetario del pago
PAGO	CK6	CK	-	-	8 dígitos y empieza con 1
-	AUTO	T	-	-	Tabla que almacena la información del auto
AUTO	id_auto	A	int	10	Almacena una pk, id que diferencia a cada auto
AUTO	año	A	smallint	5	Almacena el año del auto
AUTO	placa	A	varchar	8	Almacena la placa del automóvil
AUTO	color	A	varchar	25	Almacena el color del auto
AUTO	id_usuario	A	int	10	Almacena una fk, id del conductor del auto
AUTO	id_modelo	A	int	10	Almacena una fk, id de modelo del auto
AUTO	CK1	CK	-	-	La antigüedad no debe ser

					mayor a 5 años
-	MODELO	T	-	-	Tabla que almacena la información del modelo
MODELO	id_modelo	A	int	10	Almacena una pk, con el id del modelo que lo hace único
MODELO	nombre	A	varchar	50	Almacena el nombre del modelo de auto
MODELO	id_marca	A	smallint	5	Almacena una fk, id de marca
-	MARCA	T	-	-	Tabla que almacena la información del marca
MARCA	id_marca	A	smallint	5	Almacena una pk, con el id de marca
MARCA	nombre	A	varchar	80	Almacena el nombre de la marca
-	CLIENTE	T	-	-	Tabla que almacena la información del cliente
CLIENTE	id_usuario	A	int	10	Almacena una fk, almacena el identificador del usuario de tipo cliente
CLIENTE	rfc	A	varchar	13	Almacena el registro federal de contribuyentes
CLIENTE	direccion	A	varchar	300	Almacena la dirección del cliente que se usará para la facturación
CLIENTE	numero	A	bigint	19	Almacena el numero de celular del cliente
CLIENTE	fecha_registro	A	date	10	Almacena la fecha de registro del cliente
CLIENTE	calificacion	A	tinyint	3	Almacena la calificacion que el conductor le dio al cliente en el viaje
CLIENTE	CK8	CK	-	-	1 a 5 estrellas
-	QUEJA	T	-	-	Tabla que almacena la información de queja
QUEJA	id_queja	A	int	10	Almacena una pk, que identifica la queja generada
QUEJA	descripcion	A	varchar	300	Almacena una descripción de

					la queja
QUEJA	titulo	A	varchar	40	Almacena el título de la queja
QUEJA	fecha_atencion	A	date	10	Almacena la fecha en la que se le proporciona atención a la queja (por parte de un administrador)
QUEJA	fecha_queja	A	date	10	Almacena la fecha en la que se generó la queja
QUEJA	id_usuario	A	int	10	Almacena una fk, almacena el id del cliente que metió la queja
QUEJA	id_administrador	A	int	10	Almacena una fk, almacena el identificador del administrador
QUEJA	id_conductor	A	int	10	Almacena una fk, el id del conductor del que el usuario se queja
QUEJA	id_auto	A	int	10	Almacena una fk, el id del auto del conductor
QUEJA	CK7	CK	-	-	La fecha de atención no puede ser mayor a 5 días de la fecha en la que se levantó a queja
-	TARJETA	T	-	-	Tabla que almacena la información de tarjeta
TARJETA	id_tarjeta	A	int	10	Almacena una pk, almacena el identificador de la tarjeta
TARJETA	numero_tarjeta	A	varchar	24	Almacena el número de la tarjeta
TARJETA	vigencia	A	date	10	Almacena la fecha en la que la tarjeta dejará de ser vigente
TARJETA	id_usuario	A	int	10	Almacena una fk, que almacena el identificador del usuario
TARJETA	id_banco	A	smallint	5	Almacena una fk, que almacena el identificador del banco
-	BANCO	T	-	-	Tabla que almacena la información de banco

BANCO	id_banco	A	smallint	5	Almacena una pk, con el id que corresponde a un banco
BANCO	nombre	A	varchar	100	Almacena el nombre del banco
-	ADMINISTRADOR	T	-	-	Tabla que almacena la información de administrador
ADMINISTRADOR	id_administrador	A	int	10	Almacena una pk, id del administrador
ADMINISTRADOR	id_recomendado	A	int	10	Almacena una fk, el id del usuario que le recomendó la aplicación
ADMINISTRADOR	nombre_usuario	A	varchar	40	Almacena el nombre de usuario que tiene el administrador
ADMINISTRADOR	correo	A	varchar	80	Almacena el correo del administrador
ADMINISTRADOR	clave	A	char	20	Almacena la clave de acceso del administrador
ADMINISTRADOR	fecha_ingreso	A	date	10	Almacena la fecha en la que se registro el administrador a la aplicación
ADMINISTRADOR	nombre	A	varchar	60	Almacena el nombre pila del administrador
ADMINISTRADOR	ap_paterno	A	varchar	60	Almacena el primer apellido del administrador
ADMINISTRADOR	ap_materno	A	varchar	60	Almacena el segundo apellido del administrador
ADMINISTRADOR	CK3	CK	-	-	Se compone de 10 caracteres que incluyan una mayúscula, números y un carácter especial *, # ó \$
-	TELEFONO	T	-	-	Tabla que almacena la información del teléfono
TELEFONO	id_administrador	A	int	10	Almacena una fk y pk, id del administrador al que pertenece el número
TELEFONO	numero	A	bigint	19	Almacena el numero de celular del administrador

-	DESCUENTO_ADMIN	T	-	-	Tabla que almacena la información del descuento del administrador
DESCUENTO_ADMIN	id_descuento	A	int	10	Almacena una pk, el id del descuento del administrador
DESCUENTO_ADMIN	id_tipo	A	int	10	Almacena una fk, del id que tiene el tipo de descuento
DESCUENTO_ADMIN	importe	A	money	20	Almacena el importe monetario descontado al usuario
DESCUENTO_ADMIN	vigencia	A	date	10	Almacena la fecha límite hasta la que se permite utilizar este descuento
DESCUENTO_ADMIN	id_administrador	A	int	10	Almacena una fk, el id del administrador al que pertenece el descuento
-	VIAJE	T	-	-	Tabla que almacena la información del viaje
VIAJE	id_viaje	A	int	10	Almacena una pk, el id del viaje
VIAJE	fecha_solicitud	A	date	10	Almacena la fecha en la que se realizó la petición del viaje
VIAJE	hora_solicitud	A	time	15	Almacena la hora en la que se realizó la petición del viaje
VIAJE	latitud_origen	A	float	25	Almacena la latitud de donde inicia el viaje
VIAJE	latitud_destino	A	float	25	Almacena la latitud de el destino al que se desea llegar
VIAJE	longitud_origen	A	float	25	Almacena la longitud de donde inicia el viaje
VIAJE	longitud_destino	A	float	25	Almacena la longitud de el destino al que se desea llegar
VIAJE	fecha_inicio	A	date	10	Almacena la fecha en la que se inicia el viaje
VIAJE	hora_inicio	A	time	15	Almacena la hora en la que se inicia el viaje
VIAJE	calificacion	A	tinyint	3	Almacena la calificación asignada al viaje

VIAJE	propina	A	money	20	Almacena la cantidad monetaria que se da de propina al conductor por el viaje
VIAJE	tipo	A	char	1	Almacena un carácter que describe el tipo de pago que se realizará en el viaje
VIAJE	programado	A	char	2	Almacena información de si el viaje fue programado previamente
VIAJE	comentario	A	varchar	200	Almacena un comentario sobre el viaje
VIAJE	importe	A	money	20	Almacena el importe monetario del viaje, este sin incluir la propina, solamente es el costo del viaje en sí
VIAJE	id_auto	A	int	10	Almacena una fk, id del auto que realiza el viaje
VIAJE	id_etiqueta	A	smallint	5	Almacena una fk, id de la etiqueta que el cliente le da al conductor en el viaje
VIAJE	id_factura	A	int	10	Almacena una fk, id de la factura en la que se incluye el viaje (si es el caso)
VIAJE	id_estatus	A	tinyint	3	Almacena una fk, id del estatus que tiene el viaje
VIAJE	id_usuario	A	int	10	Almacena una fk, id del cliente que va a tomar el viaje
VIAJE	id_tarjeta	A	int	10	Almacena una fk, id de la tarjeta con la que se realiza el pago (si fuera el caso)
VIAJE	CK8	CK	-	-	La calificación va de 1 a 5 estrellas
VIAJE	CK9	CK	-	-	La propina puede ser de \$10, \$15 o \$20
VIAJE	CK10	CK	-	-	‘T’ es si fue con tarjeta y ‘E’ si fue con efectivo
VIAJE	CK12	CK	-	-	Se coloca ‘Si’ o ‘No’
-	ETIQUETA	T	-	-	Tabla que almacena la información de etiqueta

ETIQUETA	id_etiqueta	A	smallint	5	Almacena una pk, id del tipo de etiqueta
ETIQUETA	nombre	A	varchar	80	Almacena el nombre de la etiqueta asignada
-	UBICACION	T	-	-	Tabla que almacena la información de ubicacion
UBICACION	id_ubicacion	A	int	10	Almacena una pk, el id de la ubicación
UBICACION	latitud	A	float	25	Almacena la latitud de la ubicación en la que se encuentra el viaje
UBICACION	longitud	A	float	25	Almacena la longitud de la ubicación en la que se encuentra el viaje
UBICACION	id_viaje	A	int	10	Almacena una fk del viaje a la que pertenece la ubicacion
-	ACCIDENTE	T	-	-	Tabla que almacena la información del accidente
ACCIDENTE	id_accidente	A	int	10	Almacena una pk, el id del accidente
ACCIDENTE	ubicacion	A	varchar	100	Almacena la ubicación de donde ocurrió el accidente
ACCIDENTE	monto	A	money	20	Almacena el monto/costo de los daños del accidente
ACCIDENTE	herido	A	char	2	Almacena si hubo heridos en el accidente
ACCIDENTE	fecha	A	date	10	Almacena la fecha en la que ocurrió el accidente
ACCIDENTE	hora	A	time	15	Almacena la hora en la que ocurrió el accidente
ACCIDENTE	descripcion	A	varchar	300	Almacena una descripción de lo sucedido en el accidente
ACCIDENTE	culpable	A	char	2	Almacena si el conductor tuvo la culpa
ACCIDENTE	id_viaje	A	int	10	Almacena una fk, id del viaje en el que ocurrió el accidente
ACCIDENTE	id_tipo	A	smallint	5	Almacena una fk, id del tipo de accidente

ACCIDENTE	CK12	CK	-	-	Se coloca 'Si' o 'No'
-	TIPO_ACCIDENTE	T	-	-	Tabla que almacena la información del tipo_accidente
TIPO_ACCIDENTE	id_tipo	A	smallint	5	Almacena una pk, del id de tipo accidente
TIPO_ACCIDENTE	nombre	A	varchar	60	Almacena el nombre del tipo accidente
-	FACTURA	T	-	-	Tabla que almacena la información de factura
FACTURA	id_factura	A	int	10	Almacena la pk, del id de la factura
FACTURA	fecha_factura	A	date	10	Almacena la fecha en la que se genera la factura
FACTURA	importe	A	money	20	Almacena el importe monetario que se realiza en la factura
-	VIAJE_ESTATUS	T	-	-	Tabla que almacena la información de viaje_estatus
VIAJE_ESTATUS	id_estatus	A	tinyint	3	Almacena una fk y pk, id del estatus actual
VIAJE_ESTATUS	id_viaje	A	int	10	Almacena una fk y pk, id del viaje que tiene el estatus
VIAJE_ESTATUS	fecha_estatus	A	date	10	Almacena la fecha en la que se encuentra en un estado el viaje
VIAJE_ESTATUS	hora_esatus	A	time	15	Almacena la hora de un estatus en el viaje
-	ESTATUS	T	-	-	Tabla que almacena la información del estatus
ESTATUS	id_estatus	A	tinyint	3	Almacena una pk, id del estatus
ESTATUS	nombre	A	varchar	20	Almacena nombre del estatus
ESTATUS	CK11	CK	-	-	Solicitado, Programado, Confirmado, En curso, Terminado, Por cobrar, Pagado, Con adeudo o Cancelado

T - Tabla
A - Atributo
CK - Check

Nombre	Origen	Destino	Tipo
USUARIO_DESCUENTO	USUARIO	DESCUENTO	1,*
USUARIO_USUARIO	USUARIO	USUARIO	*,1
TIPO_DESCUENTO	TIPO	DESCUENTO	1,*
CONDUCTOR_PAGO	CONDUCTOR	PAGO	1,*
CONDUCTOR_AUTO	CONDUCTOR	AUTO	1,2
MODELO_AUTO	MODELO	AUTO	1,*
MARCA_MODELO	MARCA	MODELO	1,*
CLIENTE_QUEJA	CLIENTE	QUEJA	1,*
CLIENTE_TARJETA	CLIENTE	TARJETA	1,*
BANCO_TARJETA	BANCO	TARJETA	1,*
ADMINISTRADOR_QUEJA	ADMINISTRADOR	QUEJA	1,*
ADMINISTRADOR_TELEFONO	ADMINISTRADOR	TELEFONO	1,*
ADMINISTRADOR_ADMINISTRADOR	ADMINISTRADOR	ADMINISTRADOR	1,*
ADMINISTRADOR_DESCUENTO_ADMIN	ADMINISTRADOR	DESCUENTO_ADMIN	1,*
TIPO_DESCUENTO_ADMIN_	TIPO	DESCUENTO_ADMIN	1,*
AUTO_VIAJE	AUTO	VIAJE	1,*
ETIQUETA_VIAJE	ETIQUETA	VIAJE	1,*
VIAJE_UBICACION	VIAJE	UBICACION	1,*
VIAJE_ACCIDENTE	VIAJE	ACCIDENTE	1,*
FACTURA_VIAJE_	FACTURA	VIAJE	1,*
ESTATUS_VIAJE	ESTATUS	VIAJE	1,*
VIAJE_VIAJE_ESTATUS	VIAJE	VIAJE_ESTATUS	1,*
TARJETA_VIAJE	TARJETA	VIAJE	1,3
CLIENTE_VIAJE	CLIENTE	VIAJE	1,*
TIPO_ACCIDENTE_ACCIDENTE	TIPO_ACCIDENTE	ACCIDENTE	1,*

ESTATUS_VIAJE_ESTATUS	ESTATUS	VIAJE_ESTATUS	1,*
USUARIO_CONDUCTOR	USUARIO	CONDUCTOR	1,1
USUARIO_CLIENTE	USUARIO	CLIENTE	1,1

Normalización

Una vez implementada la parte relacional de la base de datos, se verificó que no existieran inconsistencias en los datos. Para garantizar la integridad y coherencia de la información, es necesario aplicar un proceso de normalización.



En nuestro caso, la normalización fue considerada e implementada desde la elaboración del Modelo Entidad-Relación (MER). A continuación, se presenta la justificación que demuestra que nuestro modelo cumple con las formas normales necesarias para asegurar la consistencia de los datos. Se abordan los siguientes puntos:

1. Eliminar redundancia con la creación de los siguientes catálogos

Se han identificado atributos que presentan un conjunto limitado y reutilizable de valores, los cuales fueron extraídos y almacenados en tablas independientes a modo de catálogos. Esto evita la duplicación innecesaria de datos en las tablas principales y mejora la consistencia. Los catálogos creados son:



a. Tipo de descuento

TIPO

 id_tipo	INTEGER	IDENTITY
 nombre	VARCHAR(60)	NOT NULL (AK1:1)



b. Banco

BANCO

 id_banco	SMALLINT	IDENTITY
 nombre	VARCHAR(100)	NOT NULL (AK1:1)



c. Estatus

ESTATUS

 id_estatus	TINYINT	IDENTITY
 nombre (CK11)	VARCHAR(20)	NOT NULL




d. Tipo de accidente

TIPO_ACCIDENTE

 id_tipo	SMALLINT	IDENTITY
 nombre	VARCHAR(60)	NOT NULL



e. Modelo

MODELO

 id_modelo	INTEGER	IDENTITY
 nombre	VARCHAR(50)	NOT NULL
 id_marca (FK)	SMALLINT	NOT NULL



f. Marca

MARCA

 id_marca	SMALLINT	IDENTITY
 nombre	VARCHAR(80)	NOT NULL

g. Etiqueta de viaje

ETIQUETA

 id_etiqueta	SMALLINT	IDENTITY
 nombre	VARCHAR(80)	NOT NULL

Con esta separación, se optimiza el almacenamiento y se facilita la actualización de valores sin necesidad de modificar múltiples registros, asegurando así que el modelo cumple al menos con la segunda forma normal (2FN).

2. Uso de llaves primarias artificiales (surrogates)

Se han implementado llaves primarias artificiales para todas las relaciones, excepto en el caso de la entidad Pago, donde se decidió mantener como clave primaria el campo “folio” por tratarse de una clave candidata natural que identifica de forma única a cada pago dentro del sistema.

Este enfoque evita complicaciones derivadas de claves compuestas o claves externas dependientes de cambios en atributos naturales.

3. Eliminación de dependencias transitivas y parciales

El modelo ha sido diseñado cuidadosamente para eliminar dependencias parciales y transitivas, cumpliendo así con la tercera forma normal (3FN).

a. No dependencias parciales

Todas las tablas utilizan llaves primarias simples (como id_usuario, id_viaje, id_queja, etc.), en la base de datos programada en SQL se emplean llaves artificiales autogeneradas (IDENTITY), disminuyendo el uso de claves compuestas.

Los atributos dependen únicamente de la clave primaria correspondiente de su entidad. Por ejemplo:

- En la tabla AUTO, atributos como placa, color o foto dependen de id_auto, que es clave única.
- En CLIENTE, dirección y teléfono dependen de id_usuario.

b. No dependencias transitivas

Una dependencia transitiva ocurre cuando un atributo no clave depende de otro atributo no clave, que a su vez depende de la clave primaria.

- Por ejemplo, en la tabla VIAJE, no hay atributos que dependen de otros atributos no clave como longitud_destino, hora_estatus, etc. Todos dependen de id_viaje.
- Las entidades como TIPO, BANCO, TIPO_ACCIDENTE, ESTATUS, etc., son catálogos, para evitar que atributos como “tipo de descuento” o “estatus de viaje” se repitan en varias tablas.

Consideraciones

Antes de empezar a realizar la implementación física, se acordó usar la siguiente nomenclatura para la creación de tablas:

- **Tablas:** Se nombran con mayúscula ‘NOMBRE TABLA’
- **Esquemas:** Se nombran con mayúscula ‘NOMBRE ESQUEMA’
- **Indices:** Se nombran siguiendo camel case ‘nombreConstraint’
- **Atributos:** Se nombran siguiendo snake case ‘nombre_atributo’
- **Constraints:** Se nombran siguiendo camel case ‘nombreConstraint’
- **Triggers:** Se nombran siguiendo camel case ‘nombreTrigger’

Diseño Físico

Por último, tenemos la implementación física de la base, este es el apartado más extenso ya que contiene desde la definición de las tablas hasta generación de reportes ya con información extraída de las mismas.

NOMBRE	ARCHIVO	OBSERVACIONES
DCL	seguridad.sql	<p>Incluye</p> <ul style="list-style-type: none"> • Creación de 4 roles llamados <ul style="list-style-type: none"> ○ ADMINISTRADOR ○ CONDUCTOR ○ CLIENTE ○ ANALISTA • Asignación de permisos para los roles de <ul style="list-style-type: none"> ○ SELECT ○ DELETE ○ UPDATE • Asignación de permisos adicionales para el rol de ADMINISTRADOR

		<ul style="list-style-type: none"> Creación de 5 usuarios <ul style="list-style-type: none"> 4 ADMINISTRADORES 1 ANALISTA
DDL	creaBase.sql	<p>Incluye</p> <ul style="list-style-type: none"> Creación de 5 esquemas llamados <ul style="list-style-type: none"> USUARIO CONDUCTOR CLIENTE VIAJE DESCUENTO Creación de 23 tablas llamadas <ul style="list-style-type: none"> BANCO QUEJA TARJETA AUTO MARCA MODELO PAGO TELEFONO DESCUENTO DESCUENTO_ADMIN TIPO ADMINISTRADOR CLIENTE CONDUCTOR USUARIO ACCIDENTE ESTATUS ETIQUETA FACTURA TIPO_ACCIDENTE UBICACION VIAJE VIAJE_ESTATUS Creación de 1 indice para el rfc llamado <ul style="list-style-type: none"> idxRfc
DML	dml.sql	<p>Incluye</p> <ul style="list-style-type: none"> Creación de 13 triggers <ul style="list-style-type: none"> Cuatro para la tabla VIAJE:

		<ul style="list-style-type: none"> ■ trgValidaFechaProgramado ■ trgValidaFechaNoProgramado ■ trgValidaPagoEfectivo ■ trgMaxTarjetas <ul style="list-style-type: none"> ○ Uno para la tabla TARJETA: <ul style="list-style-type: none"> ■ trgMaxTarjetas ○ Uno para la tabla AUTO: <ul style="list-style-type: none"> ■ trgMaxAutos ○ Uno para la tabla CONDUCTOR: <ul style="list-style-type: none"> ■ trgValidaConductor ○ Dos para la tabla CLIENTE: <ul style="list-style-type: none"> ■ trgValidaCliente ■ trgCalNull ○ Uno para la tabla CONDUCTOR: <ul style="list-style-type: none"> ■ trgClienteConductor ○ Tres para la tabla QUEJA: <ul style="list-style-type: none"> ■ trgFechaAtencion ■ trgFechaNull ■ trgAutoQueja
Informes	informes.sql	<p>Incluye:</p> <ul style="list-style-type: none"> ● Creación de consultas considerando los siguientes casos: <ul style="list-style-type: none"> ○ Mostrar viajes diarios por conductor; datos del conductor, número de viajes, monto total, por un periodo de tiempo. ○ Mostrar consolidado mensual; día, monto total, monto mensual ○ Mostrar top 5 de conductores por un periodo de tiempo ○ Mostrar top 5 de clientes, es decir, los clientes con mayor número de viajes. ○ Mostrar listado de conductores con más quejas y motivo

		<ul style="list-style-type: none"> ○ Mostrar listado de accidentes: fecha, ubicación, tipo, descripción, heridos si o no, monto gastado, nombre del conductor y auto, si el conductor fue el responsable o no. ○ Mostrar listado de los clientes con menos estrellas ○ Mostrar listado de los conductores con el total que les han dado por cada estrella ○ Mostrar listado de autos, placa, número de serie, marca, modelo, año y color y su dueño ○ Mostrar listado de quejas incluyendo el conductor y auto, con filtro para obtenerse por un periodo de tiempo o por conductor ● Creación de consultas empleando álgebra relacional: <ul style="list-style-type: none"> ○ Mostrar clientes que tengan una calificación de 5 y al menos un descuento que vence antes de julio del 2025 ● Empleo de subconsultas, joins y funciones de agregación: <ul style="list-style-type: none"> ○ Mostrar nombre completo y número de licencia de los conductores cuyos pagos totales están arriba del promedio ○ Mostrar porcentaje de pagos con tarjeta vs pagos en efectivo ○ Mostrar diferencia entre el total pagado y el total facturado de los conductores ○ Mostrar cada usuario y quién lo recomendó ○ Mostrar usuarios que nunca han recomendado a alguien ○ Mostrar ¿Cuántos usuarios ha recomendado cada usuario?
Carga Inicial	cargaInicial.sql	<p>Incluye:</p> <ul style="list-style-type: none"> ● Inserts para poblar tabla USUARIO ● Inserts para poblar tabla CONDUCTOR ● Inserts para poblar tabla CLIENTE ● Inserts para poblar tabla PAGO ● Inserts para poblar tabla TIPO ● Inserts para poblar tabla DESCUENTO ● Inserts para poblar tabla MARCA ● Inserts para poblar tabla MODELO ● Inserts para poblar tabla AUTO ● Inserts para poblar tabla ADMINISTRADOR ● Inserts para poblar tabla QUEJA ● Inserts para poblar tabla BANCO ● Inserts para poblar tabla TARJETA ● Inserts para poblar tabla DESCUENTO_ADMIN

		<ul style="list-style-type: none"> ● Inserts para poblar tabla TELEFONO ● Inserts para poblar tabla ETIQUETA ● Inserts para poblar tabla FACTURA ● Inserts para poblar tabla ESTATUS ● Inserts para poblar tabla VIAJE ● Inserts para poblar tabla VIAJE_ESTATUS ● Inserts para poblar tabla UBICACION ● Inserts para poblar tabla TIPO_ACCIDENTE ● Inserts para poblar tabla ACCIDENTE
Valida Triggers	validaTriggers.sql	<p>Incluye:</p> <ul style="list-style-type: none"> ● Creación de validación para los 13 trigger creados: <ul style="list-style-type: none"> ○ Validación para trgValidaFechaProgramado (si viaje es programado el inicio no pase de 2 días) ○ Validación para trgValidaFechaNoProgramado (si no es programado la fecha y hora inicio sea la misma que la de solicitud) ○ Validación de trgValidaPagoEfectivo (no se ponga un id de tarjeta si no se va a pagar con tarjeta) ○ Validación de trgMaxTarjetas (cliente tenga maximo 3 tarjetas) ○ Validación de trgMaxAutos (conductor tenga maximo 2 autos) ○ Validación de trgValidaConductor (conductores se inserten en conductores y no en cliente) ○ Validación de trgValidaCliente (valida que los clientes se inserten en clientes y no en conductores) ○ Validación de trgClienteConductor (si un valor bit es 0 o 1 en usuario el otro debe ser el contrario) ○ Validación de trgCalNull (si se pone null en calificación del cliente no se inserte el dato) ○ Validación de trgFechaAtencion (fecha de atención no sea mayor a 5 días) ○ Validación trgFechaNull (no inserta si se pone null en la fecha queja) ○ Validación de trgPropietarioTarjeta (tarjeta pertenezca al usuario en el viaje) ○ Validación de trgAutoQueja (auto y el conductor en la queja coincidan) ● Creación de validación para todas las cks de las tablas creadas: <ul style="list-style-type: none"> ○ Validación cks para tabla USUARIO:

		<ul style="list-style-type: none"> ■ Claves candidatas ■ Check de longitud ■ Fks ■ Valores por default ○ Validación cks para tabla CONDUCTOR: <ul style="list-style-type: none"> ■ Pk ■ Fks ■ Campo unique ■ Check de longitud ○ Validación cks para tabla CLIENTE: <ul style="list-style-type: none"> ■ Pk ■ Fks ■ Check 'IN' ■ Check de longitud ■ Campo unique ○ Validación cks para tabla PAGO: <ul style="list-style-type: none"> ■ Pk compuesta ■ Fk ■ Check 'LIKE' ○ Validación cks para tabla TIPO: <ul style="list-style-type: none"> ■ Campo unique ○ Validación cks para tabla DESCUENTO: <ul style="list-style-type: none"> ■ Fks ○ Validación cks para tabla MARCA: <ul style="list-style-type: none"> ■ Campo unique ○ Validación cks para tabla MODELO: <ul style="list-style-type: none"> ■ Fk ■ Campo unique ○ Validación cks para tabla AUTO: <ul style="list-style-type: none"> ■ Check ■ Fks ○ Validación cks para tabla ADMINISTRADOR: <ul style="list-style-type: none"> ■ Fk ■ Campo unique ■ Check de longitud ○ Validación cks para tabla QUEJA: <ul style="list-style-type: none"> ■ Fks ○ Validación cks para tabla BANCO: <ul style="list-style-type: none"> ■ Campo unique ○ Validación cks para tabla TARJETA: <ul style="list-style-type: none"> ■ Check de longitud ■ Fks ■ Campo unique ○ Validación cks para tabla DESCUENTO_ADMIN: <ul style="list-style-type: none"> ■ Fks ○ Validación cks para tabla TELEFONO:
--	--	---

		<ul style="list-style-type: none"> ■ Pk ■ Fk ■ Check de longitud ○ Validación cks para tabla ESTATUS: <ul style="list-style-type: none"> ■ Check 'IN' ■ Campo unique ○ Validación cks para tabla VIAJE_ESTATUS: <ul style="list-style-type: none"> ■ Pk ○ Validación cks para tabla VIAJE: <ul style="list-style-type: none"> ■ Fks ■ Check 'IN' ○ Validación cks para tabla UBICACION: <ul style="list-style-type: none"> ■ Fk ○ Validación cks para tabla TIPO_ACCIDENTE: <ul style="list-style-type: none"> ■ Campo unique ○ Validación cks para tabla ACCIDENTE: <ul style="list-style-type: none"> ■ Fks ■ Check 'IN'
--	--	--

A continuación, se presentan los seis scripts implementados en su versión final:

Seguridad.sql

```

/*

AUTORES:
    AYALA HERNÁNDEZ MARÍA FERNANDA
    PORTILLA HERMENEGILDO ELIZABETH
    VALENZUELA ASCENCIO GUSTAVO
    JIMENEZ AYALA YORDI JOSUE

FECHA DE CREACION: 18 DE MAYO DE 2025

ULTIMA ACTUALIZACION: 18 DE MAYO DE 2025

DESCRIPCION: DCL QUE CONTIENE LA CREACION DE USUARIOS PARA LA
GESTION DE LA BASE

*/

-- SELECCIONAMOS NUESTRA BASE PARA TRABAJAR SOBRE ELLA

USE Equipo09AppSafe
GO

-- CREAMOS LOS ROLES QUE VAMOS A USAR EN LA BASE

CREATE ROLE ADMINISTRADOR
CREATE ROLE CONDUCTOR

```



```
CREATE ROLE CLIENTE
CREATE ROLE ANALISTA

-- OTORGAMOS PERMISOS A LOS ROLES

-- 1) OTORGANDO PERMISOS DE LECTURA

GRANT SELECT ON ACCIDENTE TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON ADMINISTRADOR TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON [AUTO] TO ANALISTA, ADMINISTRADOR, CONDUCTOR
GO

GRANT SELECT ON BANCO TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON CLIENTE TO ANALISTA, ADMINISTRADOR, CLIENTE
GO

GRANT SELECT ON CONDUCTOR TO ANALISTA, ADMINISTRADOR, CONDUCTOR
GO

GRANT SELECT ON DESCUENTO TO ANALISTA, ADMINISTRADOR, CLIENTE
GO

GRANT SELECT ON DESCUENTO_ADMIN TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON ESTATUS TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON ETIQUETA TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON FACTURA TO ANALISTA, ADMINISTRADOR, CLIENTE
GO

GRANT SELECT ON MARCA TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON MODELO TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON PAGO TO ANALISTA, ADMINISTRADOR, CONDUCTOR
GO

GRANT SELECT ON QUEJA TO ANALISTA, ADMINISTRADOR, CLIENTE,
CONDUCTOR
GO

GRANT SELECT ON TARJETA TO ANALISTA, ADMINISTRADOR
```



```
GO

GRANT SELECT ON TELEFONO TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON TIPO TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON TIPO_ACCIDENTE TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON UBICACION TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON USUARIO TO ANALISTA, ADMINISTRADOR
GO

GRANT SELECT ON VIAJE TO ANALISTA, ADMINISTRADOR, CLIENTE
GO

GRANT SELECT ON VIAJE_ESTATUS TO ANALISTA, ADMINISTRADOR
GO

-- 2) OTORGANDO PERMISOS DE INSERTAR

GRANT INSERT ON ACCIDENTE TO ADMINISTRADOR
GO

GRANT INSERT ON ADMINISTRADOR TO ADMINISTRADOR
GO

GRANT INSERT ON [AUTO] TO ADMINISTRADOR, CONDUCTOR
GO

GRANT INSERT ON BANCO TO ADMINISTRADOR
GO

GRANT INSERT ON CLIENTE TO ADMINISTRADOR
GO

GRANT INSERT ON CONDUCTOR TO ADMINISTRADOR
GO

GRANT INSERT ON DESCUENTO TO ADMINISTRADOR
GO

GRANT INSERT ON DESCUENTO_ADMIN TO ADMINISTRADOR
GO

GRANT INSERT ON ESTATUS TO ADMINISTRADOR
GO

GRANT INSERT ON ETIQUETA TO ADMINISTRADOR
GO
```



```
GRANT INSERT ON FACTURA TO ADMINISTRADOR, CLIENTE
GO

GRANT INSERT ON MARCA TO ADMINISTRADOR
GO

GRANT INSERT ON MODELO TO ADMINISTRADOR
GO

GRANT INSERT ON PAGO TO ADMINISTRADOR
GO

GRANT INSERT ON QUEJA TO ADMINISTRADOR, CLIENTE
GO

GRANT INSERT ON TARJETA TO ADMINISTRADOR, CLIENTE
GO

GRANT INSERT ON TELEFONO TO ADMINISTRADOR
GO

GRANT INSERT ON TIPO TO ADMINISTRADOR
GO

GRANT INSERT ON TIPO_ACCIDENTE TO ADMINISTRADOR
GO

GRANT INSERT ON UBICACION TO ADMINISTRADOR
GO

GRANT INSERT ON USUARIO TO ADMINISTRADOR
GO

GRANT INSERT ON VIAJE TO ADMINISTRADOR, CLIENTE
GO

GRANT INSERT ON VIAJE_ESTATUS TO ADMINISTRADOR
GO

-- 3) OTORGANDO PERMISOS DE ACTUALIZAR

GRANT UPDATE ON ACCIDENTE TO ADMINISTRADOR
GO

GRANT UPDATE ON ADMINISTRADOR TO ADMINISTRADOR
GO

GRANT UPDATE ON [AUTO] TO ADMINISTRADOR
GO

GRANT UPDATE ON BANCO TO ADMINISTRADOR
GO

GRANT UPDATE ON CLIENTE TO ADMINISTRADOR
GO
```

```
GRANT UPDATE ON CONDUCTOR TO ADMINISTRADOR
GO

GRANT UPDATE ON DESCUENTO TO ADMINISTRADOR
GO

GRANT UPDATE ON DESCUENTO_ADMIN TO ADMINISTRADOR
GO

GRANT UPDATE ON ESTATUS TO ADMINISTRADOR
GO

GRANT UPDATE ON ETIQUETA TO ADMINISTRADOR
GO

GRANT UPDATE ON FACTURA TO ADMINISTRADOR
GO

GRANT UPDATE ON MARCA TO ADMINISTRADOR
GO

GRANT UPDATE ON MODELO TO ADMINISTRADOR
GO

GRANT UPDATE ON PAGO TO ADMINISTRADOR
GO

GRANT UPDATE ON QUEJA TO ADMINISTRADOR
GO

GRANT UPDATE ON TARJETA TO ADMINISTRADOR
GO

GRANT UPDATE ON TELEFONO TO ADMINISTRADOR
GO

GRANT UPDATE ON TIPO TO ADMINISTRADOR
GO

GRANT UPDATE ON TIPO_ACCIDENTE TO ADMINISTRADOR
GO

GRANT UPDATE ON UBICACION TO ADMINISTRADOR
GO

GRANT UPDATE ON USUARIO TO ADMINISTRADOR
GO

GRANT UPDATE ON VIAJE TO ADMINISTRADOR
GO

GRANT UPDATE ON VIAJE_ESTATUS TO ADMINISTRADOR
GO
```

```
-- 4) OTORGANDO PERMISOS DE ACTUALIZAR

GRANT DELETE ON ACCIDENTE TO ADMINISTRADOR
GO

GRANT DELETE ON ADMINISTRADOR TO ADMINISTRADOR
GO

GRANT DELETE ON [AUTO] TO ADMINISTRADOR
GO

GRANT DELETE ON BANCO TO ADMINISTRADOR
GO

GRANT DELETE ON CLIENTE TO ADMINISTRADOR
GO

GRANT DELETE ON CONDUCTOR TO ADMINISTRADOR
GO

GRANT DELETE ON DESCUENTO TO ADMINISTRADOR
GO

GRANT DELETE ON DESCUENTO_ADMIN TO ADMINISTRADOR
GO

GRANT DELETE ON ESTATUS TO ADMINISTRADOR
GO

GRANT DELETE ON ETIQUETA TO ADMINISTRADOR
GO

GRANT DELETE ON FACTURA TO ADMINISTRADOR
GO

GRANT DELETE ON MARCA TO ADMINISTRADOR
GO

GRANT DELETE ON MODELO TO ADMINISTRADOR
GO

GRANT DELETE ON PAGO TO ADMINISTRADOR
GO

GRANT DELETE ON QUEJA TO ADMINISTRADOR
GO

GRANT DELETE ON TARJETA TO ADMINISTRADOR
GO

GRANT DELETE ON TELEFONO TO ADMINISTRADOR
GO

GRANT DELETE ON TIPO TO ADMINISTRADOR
GO
```

```
GRANT DELETE ON TIPO_ACCIDENTE TO ADMINISTRADOR
GO

GRANT DELETE ON UBICACION TO ADMINISTRADOR
GO

GRANT DELETE ON USUARIO TO ADMINISTRADOR
GO

GRANT DELETE ON VIAJE TO ADMINISTRADOR
GO

GRANT DELETE ON VIAJE_ESTATUS TO ADMINISTRADOR
GO

-- DANDOLE EL RESTO DE PERMISOS AL ADMIN QUE ES LA MAXIMA
AUTORIDAD EN LA BASE

GRANT ALTER ANY USER TO ADMINISTRADOR; -- CREACION,
MODIFICACION, ETC DE USUARIOS
GRANT ALTER ANY ROLE TO ADMINISTRADOR; -- AGREGAR MIEMBROS A
ROLES
GRANT CREATE PROCEDURE TO ADMINISTRADOR; -- CREAR PROCEDIMIENTOS

-- CREAMOS ALGUNOS USUARIOS

CREATE LOGIN marthaAnalista WITH PASSWORD = N'qwe321r',
DEFAULT_DATABASE = [Equipo09AppSafe], CHECK_EXPIRATION = OFF
GO
CREATE USER marthaAnalista FOR LOGIN marthaAnalista
GO
ALTER ROLE ANALISTA ADD MEMBER marthaAnalista
GO

CREATE LOGIN maferAdmin WITH PASSWORD = N'789qwez',
DEFAULT_DATABASE = [Equipo09AppSafe], CHECK_EXPIRATION = OFF
GO
CREATE USER maferAdmin FOR LOGIN maferAdmin
GO
ALTER ROLE ADMINISTRADOR ADD MEMBER maferAdmin
GO

CREATE LOGIN eliAdmin WITH PASSWORD = N'1a2b3cd',
DEFAULT_DATABASE = [Equipo09AppSafe], CHECK_EXPIRATION = OFF
GO
CREATE USER eliAdmin FOR LOGIN eliAdmin
GO
ALTER ROLE ADMINISTRADOR ADD MEMBER eliAdmin
GO

CREATE LOGIN gusAdmin WITH PASSWORD = N'pass456',
DEFAULT_DATABASE = [Equipo09AppSafe], CHECK_EXPIRATION = OFF
```



```
GO
CREATE USER gusAdmin FOR LOGIN gusAdmin
GO
ALTER ROLE ADMINISTRADOR ADD MEMBER gusAdmin
GO

CREATE LOGIN yordiAdmin WITH PASSWORD = N'zxc987m',
DEFAULT_DATABASE = [Equipo09AppSafe], CHECK_EXPIRATION = OFF
GO
CREATE USER yordiAdmin FOR LOGIN yordiAdmin
GO
ALTER ROLE ADMINISTRADOR ADD MEMBER yordiAdmin
GO
```

CreaBase.sql

```
/*

AUTORES:
    AYALA HERNÁNDEZ MARÍA FERNANDA
    PORTILLA HERMENEGILDO ELIZABETH
    VALENZUELA ASCENCIO GUSTAVO
    JIMENEZ AYALA YORDI JOSUE

FECHA DE CREACION: 28 DE ABRIL DE 2025

ULTIMA ACTUALIZACION: 25 DE MAYO DE 2025

DESCRIPCION: DDL QUE CONTIENE LA CREACION DE TODAS LAS TABLAS DE LA
BASE, INCLUYENDO LOS CONSTRAINTS QUE REQUIEREN

*/

-- CREAMOS LA BASE

CREATE DATABASE Equipo09AppSafe

-- SELECCIONAMOS LA BASE PARA EMPEZAR CON EL DDL

USE Equipo09AppSafe
GO

-- CREAMOS LOS ESQUEMAS

CREATE SCHEMA USUARIO
GO

CREATE SCHEMA CONDUCTOR
GO

CREATE SCHEMA CLIENTE
GO
```

```
CREATE SCHEMA VIAJE
GO

CREATE SCHEMA DESCUENTO
GO

-- TABLA USUARIO

CREATE TABLE USUARIO.USUARIO(
    id_usuario INTEGER NOT NULL IDENTITY(1,1),
    nombre_usuario VARCHAR(40) NOT NULL,
    correo VARCHAR(80) NULL CONSTRAINT dfCorreoUsuario DEFAULT
'SIN CORREO',
    clave CHAR(20) NOT NULL,
    fecha_ingreso DATE NOT NULL,
    nombre VARCHAR(60) NOT NULL,
    ap_paterno VARCHAR(60) NOT NULL,
    ap_materno VARCHAR(60) NULL,
    cliente BIT NOT NULL,
    conductor BIT NOT NULL,
    id_recomendado INTEGER NULL,

    CONSTRAINT pkUsuario PRIMARY KEY (id_usuario),
    CONSTRAINT akNombreUsuario UNIQUE (nombre_usuario),
    CONSTRAINT ckClave CHECK (
        LEN(clave) = 20
        AND clave LIKE '%[A-Z]%'
        AND clave LIKE '%[0-9]%'
        AND (clave LIKE '%#%' OR clave LIKE '%*%' OR clave LIKE
'%$%')),
    CONSTRAINT fkUsuarioRecomendado FOREIGN KEY (id_recomendado)
REFERENCES USUARIO.USUARIO(id_usuario),

);

-- TABLA CONDUCTOR

CREATE TABLE USUARIO.CONDUCTOR(
    id_usuario INTEGER NOT NULL,
    num_licencia INTEGER NOT NULL,
    vigencia_lic DATE NOT NULL,
    foto IMAGE NOT NULL,
    descripcion VARCHAR(600) NOT NULL,

    CONSTRAINT pkConductor PRIMARY KEY (id_usuario),
    CONSTRAINT fkUsuarioConductor FOREIGN KEY (id_usuario)
REFERENCES USUARIO.USUARIO(id_usuario)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT akNumLicencia UNIQUE(num_licencia),
    CONSTRAINT ckDescripcion CHECK(LEN (descripcion) <= 600)

);

-- TABLA CLIENTE
```

```
CREATE TABLE USUARIO.CLIENTE (
    id_usuario INTEGER NOT NULL,
    rfc varchar(13) NOT NULL,
    direccion VARCHAR(300) NOT NULL,
    numero BIGINT NOT NULL,
    fecha_registro DATE NOT NULL,
    calificacion TINYINT NULL CONSTRAINT dfCalificacionCliente
    DEFAULT (5),

    CONSTRAINT pkCliente PRIMARY KEY (id_usuario),
    CONSTRAINT fkUsuarioCliente FOREIGN KEY (id_usuario)
    REFERENCES USUARIO.USUARIO(id_usuario)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT ckCalificacion CHECK(calificacion IN (1,2,3,4,5)),
    CONSTRAINT ckRfc CHECK (LEN(rfc) = 13)
);
```

-- TABLA PAGO

```
CREATE TABLE CONDUCTOR.PAGO(
    id_usuario INT NOT NULL,
    folio CHAR(8) NOT NULL,
    fecha DATE NOT NULL,
    monto MONEY NOT NULL,
    CONSTRAINT pkPago PRIMARY KEY (id_usuario ASC, folio ASC),
    CONSTRAINT fkIdUsuarioPago FOREIGN KEY (id_usuario) REFERENCES
    USUARIO.CONDUCTOR (id_usuario)
    on DELETE CASCADE,
    CONSTRAINT ckFolio CHECK (folio LIKE '1%')
);
```

-- TABLA TIPO

```
CREATE TABLE DESCUENTO.TIPO (
    id_tipo INT NOT NULL IDENTITY(1,1),
    nombre VARCHAR(60) NOT NULL,
    CONSTRAINT pkTipo PRIMARY KEY (id_tipo),
    CONSTRAINT akNombre UNIQUE (nombre)
);
```

-- TABLA DESCUENTO

```
CREATE TABLE DESCUENTO.DESCUENTO(
    id_descuento INT NOT NULL IDENTITY(1,1),
    importe MONEY NOT NULL,
    vigencia DATE NOT NULL,
    id_usuario INT NOT NULL,
    id_tipo INT NOT NULL,
    CONSTRAINT pkDescuento PRIMARY KEY (id_descuento),
    CONSTRAINT fkIdUsuario FOREIGN KEY (id_usuario) REFERENCES
```

```
USUARIO.USUARIO(id_usuario)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT fkIdTipo FOREIGN KEY (id_tipo) REFERENCES
DESCUENTO.TIPO(id_tipo)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

-- TABLA MARCA

CREATE TABLE CONDUCTOR.MARCA(
    id_marca SMALLINT IDENTITY(1,1),
    nombre VARCHAR(80) NOT NULL,
    CONSTRAINT pkMarca PRIMARY KEY (id_marca),
    CONSTRAINT akNombreMarca UNIQUE (nombre)
);

-- TABLA MODELO

CREATE TABLE CONDUCTOR.MODELO(
    id_modelo INT NOT NULL IDENTITY(1,1),
    nombre VARCHAR(50) NOT NULL,
    id_marca SMALLINT NOT NULL,
    CONSTRAINT pkModelo PRIMARY KEY (id_modelo),
    CONSTRAINT fkMarcaModelo FOREIGN KEY (id_marca) REFERENCES
CONDUCTOR.MARCA(id_marca)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT akNombreModelo UNIQUE (nombre)
);

-- TABLA AUTO

CREATE TABLE CONDUCTOR.[AUTO](
    id_auto INT NOT NULL IDENTITY(1,1),
    año SMALLINT NOT NULL,
    placa VARCHAR(8) NOT NULL,
    color VARCHAR(25) NOT NULL,
    id_modelo INT NOT NULL,
    id_usuario INT NOT NULL,
    CONSTRAINT pkAuto PRIMARY KEY (id_auto),
    CONSTRAINT ckAño CHECK (YEAR(GETDATE())-año <=5),
    CONSTRAINT fkIdModelo FOREIGN KEY (id_modelo) REFERENCES
CONDUCTOR.MODELO(id_modelo)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT fkIdUsuarioAuto FOREIGN KEY (id_usuario) REFERENCES
USUARIO.CONDUCTOR(id_usuario)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

```
--TABLA ADMINISTRADOR
```

```
CREATE TABLE USUARIO.ADMINISTRADOR(  
    id_administrador INTEGER NOT NULL IDENTITY(1,1),  
    nombre_usuario VARCHAR(40) NOT NULL,  
    correo VARCHAR(80) NULL CONSTRAINT dfCorreoAdmin DEFAULT 'SIN  
CORREO' ,  
    clave CHAR(20) NOT NULL,  
    fecha_ingreso DATE NOT NULL,  
    nombre VARCHAR(60) NOT NULL,  
    ap_paterno VARCHAR(60) NOT NULL,  
    ap_materno VARCHAR(60) NULL,  
    id_recomendado INTEGER NULL,  
  
    CONSTRAINT pkAdministrador PRIMARY KEY (id_administrador),  
    CONSTRAINT fkAdministradorRecomendado FOREIGN KEY  
(id_recomendado) REFERENCES  
USUARIO.ADMINISTRADOR(id_administrador),  
    CONSTRAINT akNombreUsuarioAdmin UNIQUE (nombre_usuario),  
    CONSTRAINT ckClaveAdmin CHECK (  
        LEN(clave) = 20  
        AND clave LIKE '%[A-Z]%'  
        AND clave LIKE '%[0-9]%'  
        AND (clave LIKE '%#%' OR clave LIKE '%*%' OR clave LIKE  
'%$%'))  
);
```

```
-- TABLA QUEJA
```

```
CREATE TABLE CLIENTE.QUEJA(  
    id_queja INTEGER NOT NULL IDENTITY (1,1),  
    descripcion VARCHAR(300) NOT NULL,  
    titulo VARCHAR(40) NOT NULL,  
    fecha_atencion DATE NOT NULL,  
    fecha_queja DATE NULL CONSTRAINT dfFechaQueja DEFAULT  
GETDATE(),  
    dias_respuesta AS (DATEDIFF(DAY, fecha_queja, fecha_atencion)),  
    id_usuario INT NOT NULL,  
    id_administrador INT NOT NULL,  
    id_conductor INT NOT NULL,  
    id_auto INT NOT NULL,  
    CONSTRAINT pkQueja PRIMARY KEY (id_queja),  
    CONSTRAINT fkIdUsuarioQueja FOREIGN KEY (id_usuario) REFERENCES  
USUARIO.CLIENTE(id_usuario)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
    CONSTRAINT fkIdConductorQueja FOREIGN KEY (id_conductor)  
REFERENCES USUARIO.CONDUCTOR(id_usuario),  
    CONSTRAINT fkIdAutoQueja FOREIGN KEY (id_auto) REFERENCES  
CONDUCTOR.[AUTO] (id_auto),  
    CONSTRAINT fkIdAdministradorQueja FOREIGN KEY  
(id_administrador) REFERENCES  
USUARIO.ADMINISTRADOR(id_administrador)
```

```
        ON DELETE CASCADE
        ON UPDATE CASCADE,

);

-- TABLA BANCO

CREATE TABLE CLIENTE.BANCO(
    id_banco SMALLINT NOT NULL IDENTITY(1,1),
    nombre VARCHAR(100) NOT NULL,
    CONSTRAINT pkBanco PRIMARY KEY (id_banco),
    CONSTRAINT akNombreBanco UNIQUE (nombre)
);

-- TABLA TARJETA

CREATE TABLE CLIENTE.TARJETA(
    id_tarjeta INTEGER NOT NULL IDENTITY(1,1),
    numero_tarjeta VARCHAR(24) NOT NULL,
    vigencia DATE NOT NULL,
    id_usuario INTEGER NOT NULL,
    id_banco SMALLINT NOT NULL,

    CONSTRAINT pkTarjeta PRIMARY KEY (id_tarjeta),
    CONSTRAINT ckNumeroTarjeta CHECK (LEN(numero_Tarjeta) = 16),
    CONSTRAINT akNumeroTarjeta UNIQUE (numero_Tarjeta),
    CONSTRAINT fkBancoTarjeta FOREIGN KEY (id_banco) REFERENCES
CLIENTE.BANCO(id_banco)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT fkClienteTarjeta FOREIGN KEY (id_usuario)
REFERENCES USUARIO.CLIENTE(id_usuario)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

-- TABLA DESCUENTO_ADMIN

CREATE TABLE DESCUENTO.DESCUENTO_ADMIN(
    id_descuento INTEGER NOT NULL IDENTITY(1,1),
    importe MONEY NOT NULL,
    vigencia DATE NOT NULL,
    id_tipo INTEGER NOT NULL,
    id_administrador INTEGER NOT NULL,

    CONSTRAINT pkDescuentoAdmin PRIMARY KEY (id_descuento),
    CONSTRAINT fkTipoDescuentoAdmin FOREIGN KEY (id_tipo)
REFERENCES DESCUENTO.TIPO(id_tipo)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT fkAdministradorDescuentoAdmin FOREIGN KEY
(id_administrador) REFERENCES USUARIO.ADMINISTRADOR
```

```
(id_administrador)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

-- TABLA TELEFONO

CREATE TABLE TELEFONO (
    id_administrador INT NOT NULL,
    numero BIGINT NOT NULL,
    CONSTRAINT pkTelefono PRIMARY KEY CLUSTERED (id_administrador
ASC, numero ASC),
    CONSTRAINT fkAdministradorTelefono FOREIGN KEY
(id_administrador) REFERENCES
USUARIO.ADMINISTRADOR(id_administrador)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT lonNumero CHECK (LEN(numero) = 10)
);

-- TABLA ETIQUETA

CREATE TABLE VIAJE.ETIQUETA(
    id_etiqueta SMALLINT NOT NULL IDENTITY(1,1),
    nombre VARCHAR(80) NOT NULL,
    CONSTRAINT pkEtiqueta PRIMARY KEY (id_etiqueta),
    CONSTRAINT akNombreEtiqueta UNIQUE (nombre)
);

-- TABLA FACTURA

CREATE TABLE VIAJE.FACTURA(
    id_factura INTEGER NOT NULL IDENTITY(1,1),
    fecha_factura DATE NOT NULL,
    importe MONEY NOT NULL,
    CONSTRAINT pkFactura PRIMARY KEY (id_factura)
);

-- TABLA ESTATUS

CREATE TABLE VIAJE.ESTATUS(
    id_estatus TINYINT NOT NULL IDENTITY(1,1),
    nombre VARCHAR(20) NOT NULL,
    CONSTRAINT pkEstatus PRIMARY KEY (id_estatus),
    CONSTRAINT ckNombreEstatus CHECK (nombre in
('Solicitado','Programado', 'Confirmado',
'En curso',
'Terminado', 'Por cobrar',
'Pagado', 'Con
adeudo','Cancelado')),
    CONSTRAINT akNombreEstatus UNIQUE (nombre)
```

```
);

-- TABLA VIAJE ESTATUS

CREATE TABLE VIAJE.VIAJE_ESTATUS(
    id_estatus TINYINT NOT NULL,
    id_viaje INT NOT NULL,
    fecha_estatus DATE NOT NULL,
    hora_estatus TIME NOT NULL
    CONSTRAINT pkViajeEstatus PRIMARY KEY CLUSTERED (id_estatus
ASC, id_viaje ASC)
);

-- TABLA VIAJE

CREATE TABLE VIAJE.VIAJE (
    id_viaje INTEGER NOT NULL IDENTITY(1,1),
    fecha_solicitud DATE NOT NULL,
    hora_solicitud TIME NOT NULL,
    latitud_origen FLOAT NOT NULL,
    latitud_destino FLOAT NOT NULL,
    longitud_origen FLOAT NOT NULL,
    longitud_destino FLOAT NOT NULL,
    fecha_inicio DATE NOT NULL,
    hora_inicio TIME NOT NULL,
    calificacion TINYINT NULL CONSTRAINT dfCalificacionViaje
DEFAULT 5,
    propina MONEY NULL,
    tipo CHAR(1) NOT NULL,
    programado CHAR(2) NOT NULL,
    comentario VARCHAR(200) NOT NULL,
    importe MONEY NOT NULL,
    id_auto INT NOT NULL,
    id_etiqueta SMALLINT NULL,
    id_factura INT NULL,
    id_estatus TINYINT NOT NULL,
    id_usuario INT NOT NULL,
    id_tarjeta INT NULL,

    CONSTRAINT pkViaje PRIMARY KEY (id_viaje),
    CONSTRAINT fkAutoViaje FOREIGN KEY (id_auto) REFERENCES
CONDUCTOR.[AUTO] (id_auto)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT fkEtiquetaViaje FOREIGN KEY (id_etiqueta) REFERENCES
VIAJE.ETIQUETA (id_etiqueta)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT fkFacturaViaje FOREIGN KEY (id_factura) REFERENCES
VIAJE.FACTURA (id_factura)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT fkEstatusViaje FOREIGN KEY (id_estatus) REFERENCES
```



```
VIAJE.ESTATUS (id_estatus)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT fkUsuarioViaje FOREIGN KEY (id_usuario) REFERENCES
USUARIO.CLIENTE (id_usuario),
    CONSTRAINT fkTarjetaViaje FOREIGN KEY (id_tarjeta) REFERENCES
CLIENTE.TARJETA (id_tarjeta),
    CONSTRAINT ckCalificacionViaje CHECK(calificacion IN
(1,2,3,4,5)),
    CONSTRAINT ckPropina CHECK(propina IN (10, 15, 20)),
    CONSTRAINT ckTipo CHECK(tipo IN ('T', 'E')),
    CONSTRAINT ckProgramado CHECK (programado IN ('SI', 'NO')),
);

-- TABLA UBICACION

CREATE TABLE VIAJE.UBICACION(
    id_ubicacion INTEGER NOT NULL IDENTITY(1,1),
    latitud FLOAT NOT NULL,
    longitud FLOAT NOT NULL,
    id_viaje INTEGER NOT NULL,
    CONSTRAINT pkUbicacion PRIMARY KEY (id_ubicacion),
    CONSTRAINT fkViajeUbicacion FOREIGN KEY (id_viaje) REFERENCES
VIAJE.VIAJE(id_viaje)
    ON DELETE CASCADE
);

-- TABLA TIPO_ACCIDENTE

CREATE TABLE VIAJE.TIPO_ACCIDENTE(
    id_tipo SMALLINT NOT NULL IDENTITY(1,1),
    nombre VARCHAR(60) NOT NULL,
    CONSTRAINT pkTipoAccidente PRIMARY KEY (id_tipo),
    CONSTRAINT akNombreTipoAccidente UNIQUE (nombre)
);

-- TABLA ACCIDENTE

CREATE TABLE VIAJE.ACCIDENTE(
    id_accidente INTEGER NOT NULL IDENTITY(1,1),
    ubicacion VARCHAR(100) NOT NULL,
    monto MONEY NOT NULL,
    herido CHAR(2) NOT NULL,
    fecha DATE NOT NULL,
    hora TIME NOT NULL,
    culpable CHAR(1) NOT NULL,
    descripcion VARCHAR(300) NOT NULL,
    id_viaje INTEGER NOT NULL,
    id_tipo SMALLINT NOT NULL,
    CONSTRAINT pkAccidente PRIMARY KEY (id_accidente),
    CONSTRAINT fkTipoAccidente FOREIGN KEY (id_tipo) REFERENCES
```

```
VIAJE.TIPO_ACCIDENTE(id_tipo)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT fkViajeAccidente FOREIGN KEY (id_viaje) REFERENCES
VIAJE.VIAJE(id_viaje)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT chkHerido CHECK (herido IN ('SI', 'NO')),
  CONSTRAINT ckCulpable CHECK (culpable IN ('SI', 'NO'))

);

-- CREACION DE INIDICES

CREATE UNIQUE NONCLUSTERED INDEX idxRFC ON USUARIO.CLIENTE (rfc)
GO
```

Dml.sql

```
/*

AUTORES:
    AYALA HERNÁNDEZ MARÍA FERNANDA
    PORTILLA HERMENEGILDO ELIZABETH
    VALENZUELA ASCENCIO GUSTAVO
    JIMENEZ AYALA YORDI JOSUE

FECHA DE CREACION: 24 DE MAYO DE 2025

ULTIMA ACTUALIZACION: 25 DE MAYO DE 2025

DESCRIPCION: DML QUE CONTIENE LA CREACION DE TRIGGERS

*/

-- SELECCIONAMOS NUESTRA BASE PARA TRABAJAR SOBRE ELLA

USE Equipo09AppSafe
GO

-- Trigger que valida en la tabla VIAJE que si un viaje es
programado el inicio no pase de 2 dias

CREATE TRIGGER trgValidaFechaProgramado ON VIAJE.VIAJE
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
```

```
SELECT 1
FROM inserted
WHERE programado = 'SI'
      AND DATEADD(DAY, 2, CAST(fecha_solicitud AS DATETIME) +
CAST(hora_solicitud AS DATETIME))
      < (CAST(fecha_inicio AS DATETIME) + CAST(hora_inicio
AS DATETIME))
)
BEGIN
    RAISERROR('CUIDADO, LA FECHA DE INICIO EXCEDE LA FECHA
LÍMITE PERMITIDA', 16, 1);
    ROLLBACK TRANSACTION;
END
END;
GO

-- Trigger que valida que si no es programado la fecha y hora
inicio sea la misma que la de solicitud

CREATE TRIGGER trgValidaFechaNoProgramado ON VIAJE.VIAJE
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE programado = 'NO'
              AND (CAST(fecha_inicio AS DATETIME) + CAST(hora_inicio AS
DATETIME))
              <> (CAST(fecha_solicitud AS DATETIME) +
CAST(hora_solicitud AS DATETIME))
        )
    BEGIN
        RAISERROR('EL VIAJE NO ES PROGRAMADO, LA FECHA Y HORA DE
INICIO NO PUEDEN SER DIFERENTES A LAS DE SOLICITUD', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO

-- Trigger que valida que no se ponga un id de tarjeta si no se va
a pagar con tarjeta (tipo = 'E')

CREATE TRIGGER trgValidaPagoEfectivo ON VIAJE.VIAJE
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE tipo = 'E' AND id_tarjeta IS NOT NULL
```

```
)
BEGIN
    RAISERROR('EL VIAJE SE VA A PAGAR CON EFECTIVO, NO SE PUEDE
INSERTAR UNA TARJETA', 16, 1);
    ROLLBACK TRANSACTION;
END
END;
GO

-- Tigger que validar que un cliente tenga maximo 3 tarjetas

CREATE TRIGGER trgMaxTarjetas ON CLIENTE.TARJETA
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        GROUP BY i.id_usuario
        HAVING (
            SELECT COUNT(*)
            FROM CLIENTE.TARJETA t
            WHERE t.id_usuario = i.id_usuario
        ) > 3
    )
    BEGIN
        RAISERROR('NO SE PUEDEN INSERTAR MAS TARJETAS, LIMITE
ALCANZADO', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO

-- Trigger que valida que un conductor tenga maximo 2 autos

CREATE TRIGGER trgMaxAutos ON CONDUCTOR.[AUTO]
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        GROUP BY i.id_usuario
        HAVING (
            SELECT COUNT(*)
            FROM CONDUCTOR.[AUTO] a
            WHERE a.id_usuario = i.id_usuario
        ) > 2
    )
    BEGIN
        RAISERROR('NO SE PUEDEN REGISTRAR MAS AUTOS, LIMITE
```

```
ALCANZADO', 16, 1);
    ROLLBACK TRANSACTION;
END
END;
GO

-- Trigger que valida que los conductores se inserten en
conductores y no en cliente

CREATE TRIGGER trgValidaConductor ON USUARIO.CONDUCTOR
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN USUARIO.USUARIO u ON i.id_usuario = u.id_usuario
        WHERE u.cliente = 1 AND u.conductor = 0
    )
    BEGIN
        RAISERROR('ERROR: EL USUARIO CORRESPONDE A LA CATEGORIA
CLIENTE', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO

-- Trigger que valida que los clientes se inserten en clientes y no
en conductores

CREATE TRIGGER trgValidaCliente ON USUARIO.CLIENTE
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN USUARIO.USUARIO u ON i.id_usuario = u.id_usuario
        WHERE u.cliente = 0 AND u.conductor = 1
    )
    BEGIN
        RAISERROR('ERROR: EL USUARIO CORRESPONDE A LA CATEGORIA
CONDUCTOR', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO

-- Trigger que valida que si un valor bit es 0 o 1 en usuario el
otro debe ser el contrario
```

```
CREATE TRIGGER trgClienteConductor ON USUARIO.USUARIO
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE (cliente = 1 AND conductor = 1)
            OR (cliente = 0 AND conductor = 0)
    )
    BEGIN
        RAISERROR('ERROR: EL ROL INGRESADO NO ES VALIDO', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO

-- Trigger que valida que si se pone null en calificacion del
cliente no se inserte el dato

CREATE TRIGGER trgCalNull ON USUARIO.CLIENTE
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE calificacion IS NULL
    )
    BEGIN
        RAISERROR('ERROR: LA CALIFICACION NO PUEDE SER NULL, DEJA
EL VALOR DEFAULT', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO

-- Trigger que valida que la fecha de atencion no sea mayor a 5
días

CREATE TRIGGER trgFechaAtencion ON CLIENTE.QUEJA
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE DATEDIFF(DAY, fecha_queja, fecha_atencion) > 5
    )
```

```
BEGIN
    RAISERROR('LA FECHA DE ATENCION NO PUEDE SER MAYOR A 5
DIAS', 16, 1);
    ROLLBACK TRANSACTION;
END
END;
GO

-- Trigger que no inserta si se pone null en la fecha queja

CREATE TRIGGER trgFechaNull ON CLIENTE.QUEJA
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE fecha_queja IS NULL
    )
    BEGIN
        RAISERROR('ERROR: LA FECHA DE LA QUEJA NO DEBE SER NULA,
DEJA EL DEFAULT', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO

-- Trigger que valida que la tarjeta pertenezca al usuario en el
viaje

CREATE TRIGGER trgPropietarioTarjeta ON VIAJE.VIAJE
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        LEFT JOIN CLIENTE.TARJETA t ON i.id_tarjeta = t.id_tarjeta
        WHERE i.id_tarjeta IS NOT NULL
            AND t.id_usuario <> i.id_usuario
    )
    BEGIN
        RAISERROR('EL PROPIETARIO DE LA TARJETA NO COINCIDE CON EL
SELECCIONADO', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO

-- Trigger que valida que el auto y el conductor en la queja
coincidan
```

```
CREATE TRIGGER trgAutoQueja ON CLIENTE.QUEJA
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        LEFT JOIN CONDUCTOR.AUTO a ON i.id_auto = a.id_auto
        WHERE a.id_usuario <> i.id_conductor
    )
    BEGIN
        RAISERROR('EL PROPIETARIO DEL AUTO NO COINCIDE CON EL
CONDUCTOR SELECCIONADO', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO
```

Informes.sql

```
/*
AUTORES:
    AYALA HERNÁNDEZ MARÍA FERNANDA
    PORTILLA HERMENEGILDO ELIZABETH
    VALENZUELA ASCENCIO GUSTAVO
    JIMENEZ AYALA YORDI JOSUE

FECHA DE CREACIÓN: 25 DE MAYO DE 2025

ULTIMA ACTUALIZACIÓN: 25 DE MAYO DE 2025

DESCRIPCIÓN: SCRIPT DQL QUE CONTIENE LAS CONSULTAS REQUERIDAS PARA
OBTENER INFORMES DE LA BASE

*/

USE [Equipo09AppSafe] -- Usamos la base de datos

-- A) Viajes diarios por conductor; datos del conductor, número de
viajes, monto total. Este informe debe poder obtener por un periodo
de tiempo

SELECT * FROM USUARIO.USUARIO
SELECT * FROM USUARIO.CONDUCTOR
SELECT * FROM VIAJE.VIAJE

SELECT
    U.nombre_usuario, -- Nombre de usuario
    U.ap_paterno,     -- Nombre completo
```



```
U.ap_materno,
C.num_licencia,    -- Número de licencia
CAST(V.fecha_inicio AS DATE) AS fecha_viaje,  -- Fecha diaria
COUNT(*) AS num_viajes, -- Número de viajes
SUM(V.importe + COALESCE(V.propina, 0)) AS monto_total  -- Suma
con propinas
FROM
    USUARIO.USUARIO U  -- Consultamos la tabla USUARIO
JOIN
    USUARIO.CONDUCTOR C ON U.id_usuario = C.id_usuario  -- Unimos
con su subclase
JOIN
    CONDUCTOR.[AUTO] A ON C.id_usuario = A.id_usuario  -- Unimos
con auto para obtener el ID del auto
JOIN
    VIAJE.VIAJE V ON A.id_auto = V.id_auto  -- Obtenemos el auto del
conductor asociado al viaje
WHERE
    V.fecha_inicio BETWEEN '2025-05-01' AND '2025-05-31'  -- Periodo
de consulta
GROUP BY
    U.nombre_usuario,  -- Agrupamos por nombre y apellidos
    U.ap_paterno,
    U.ap_materno,
    C.num_licencia,
    CAST(V.fecha_inicio AS DATE)  -- Agrupamos por día
ORDER BY
    fecha_viaje, U.nombre_usuario;  -- Ordenamos resultados

-- B) Consolidado mensual; día, monto total diario, monto mensual
acumulado

SELECT * FROM VIAJE.VIAJE

SELECT
    CAST(fecha_inicio AS DATE) AS dia,  -- Fecha del día
    SUM(importe) AS monto_total_diario, -- Monto diario
    SUM(SUM(importe)) OVER (PARTITION BY YEAR(fecha_inicio),
MONTH(fecha_inicio)) AS monto_mensual -- Usamos partition by para
segmentar en meses
FROM
    VIAJE.VIAJE
WHERE
    fecha_inicio BETWEEN '2025-05-01' AND '2025-05-31'  -- Filtro
por un mes específico
GROUP BY
    CAST(fecha_inicio AS DATE),  -- Agrupamos por día
    YEAR(fecha_inicio),  -- Necesario para la función de ventana
    MONTH(fecha_inicio)
ORDER BY
    dia;  -- Ordenamos con base en el día

-- C) Top 5 de conductores por un periodo de tiempo
```

```
SELECT TOP 5
    U.id_usuario,
    U.nombre_usuario,
    U.nombre,
    U.ap_paterno,
    U.ap_materno,
    C.num_licencia,
    FORMAT(AVG(CAST(V.calificacion AS DECIMAL(3,1))), 'N2') AS
calificacion_promedio -- Cast a DECIMAL y redondeo a 2 decimales
FROM
    USUARIO.USUARIO U
JOIN
    USUARIO.CONDUCTOR C ON U.id_usuario = C.id_usuario
JOIN
    CONDUCTOR.[AUTO] A ON C.id_usuario = A.id_usuario
JOIN
    VIAJE.VIAJE V ON A.id_auto = V.id_auto -- Unimos directamente
con VIAJE
WHERE
    V.fecha_solicitud BETWEEN '2025-05-01' AND '2025-05-31' --
Filtro por periodo
GROUP BY
    U.id_usuario,
    U.nombre_usuario,
    U.nombre,
    U.ap_paterno,
    U.ap_materno,
    C.num_licencia
ORDER BY
    calificacion_promedio DESC;

-- D) Top 5 de clientes, es decir, los clientes con mayor número de
viajes (nombre completo y correo)

SELECT TOP 5
    U.nombre,
    U.ap_paterno,
    U.ap_materno,
    U.correo,
    COUNT(*) AS num_viajes
FROM USUARIO.USUARIO U
JOIN
    VIAJE.VIAJE V ON U.id_usuario = V.id_usuario
GROUP BY U.nombre,U.ap_paterno,U.ap_materno,U.correo
ORDER BY num_viajes DESC

-- E) Listado de conductores con más quejas y motivo (se maneja un
catálogo, ejemplo, irrespetuoso, maneja muy rápido, no respeta las
reglas de vialidad, etc.)

SELECT * FROM CLIENTE.QUEJA
SELECT * FROM VIAJE.ETIQUETA

WITH QuejasConductor AS (
    SELECT
```

```
        C.id_usuario,
        E.nombre AS motivo,
        COUNT(*) AS num_quejas,
        ROW_NUMBER() OVER (PARTITION BY C.id_usuario ORDER BY
COUNT(*) DESC) AS ranking_motivo
    FROM
        CLIENTE.QUEJA Q
    JOIN
        VIAJE.VIAJE V ON Q.id_auto = V.id_auto
    JOIN
        CONDUCTOR.[AUTO] A ON V.id_auto = A.id_auto
    JOIN
        USUARIO.CONDUCTOR C ON A.id_usuario = C.id_usuario
    JOIN
        VIAJE.ETIQUETA E ON V.id_etiqueta = E.id_etiqueta -- Motivo
desde la etiqueta del viaje
    GROUP BY
        C.id_usuario, E.nombre
)

SELECT
    U.id_usuario,
    U.nombre_usuario,
    U.nombre,
    U.ap_paterno,
    U.ap_materno,
    C.num_licencia,
    QC.num_quejas,
    QC.motivo AS motivo_principal
FROM
    QuejasConductor QC
JOIN
    USUARIO.USUARIO U ON QC.id_usuario = U.id_usuario
JOIN
    USUARIO.CONDUCTOR C ON QC.id_usuario = C.id_usuario
WHERE
    QC.ranking_motivo = 1 -- Filtra solo el motivo más frecuente
por conductor
ORDER BY
    QC.num_quejas DESC;

-- F) Listado de accidentes; fecha, ubicación, tipo, descripción,
--heridos sí o no, monto gastado, nombre del conductor y auto, si el
conductor fue el
--responsable o no. Con filtros para poder obtener el listado desde
un día
--o un periodo de tiempo

SELECT * FROM VIAJE.ACCIDENTE
SELECT * FROM VIAJE.TIPO_ACCIDENTE

SELECT
    A.fecha,
    A.ubicacion,
    T.nombre AS tipo_accidente,
```

```
A.descripcion,
A.herido AS heridos,
A.monto AS monto_gastado,
U.nombre AS nombre_conductor,
U.ap_paterno AS apellido_paterno,
U.ap_materno AS apellido_materno,
AU.placa AS placa_auto,
A.culpable AS conductor_responsable
FROM
  VIAJE.ACCIDENTE A
JOIN
  VIAJE.TIPO_ACCIDENTE T ON A.id_tipo = T.id_tipo
JOIN
  VIAJE.VIAJE V ON A.id_viaje = V.id_viaje
JOIN
  CONDUCTOR.[AUTO] AU ON V.id_auto = AU.id_auto
JOIN
  USUARIO.CONDUCTOR C ON AU.id_usuario = C.id_usuario
JOIN
  USUARIO.USUARIO U ON C.id_usuario = U.id_usuario
WHERE
  A.fecha BETWEEN '2025-05-01' AND '2025-05-31' -- Filtro por
periodo
ORDER BY
  A.fecha;

-- G) Listado de los clientes con menos estrellas

SELECT
  U.id_usuario,
  U.nombre_usuario,
  U.nombre,
  U.ap_paterno,
  U.ap_materno,
  U.correo,
  C.calificacion
FROM USUARIO.USUARIO U
JOIN
  USUARIO.CLIENTE C ON U.id_usuario = C.id_usuario
WHERE C.calificacion <= 3
ORDER BY C.calificacion

-- H) Listado de conductores con el total que les han dado por cada
estrella

SELECT
  U.id_usuario,
  U.nombre_usuario,
  U.nombre,
  U.ap_paterno,
  U.ap_materno,
  C.num_licencia,
  COUNT(*) AS total_viajes,
  SUM(CASE WHEN V.calificacion = 1 THEN 1 ELSE 0 END) AS
una_estrella,
```

```
SUM(CASE WHEN V.calificacion = 2 THEN 1 ELSE 0 END) AS
dos_estrellas,
SUM(CASE WHEN V.calificacion = 3 THEN 1 ELSE 0 END) AS
tres_estrellas,
SUM(CASE WHEN V.calificacion = 4 THEN 1 ELSE 0 END) AS
cuatro_estrellas,
SUM(CASE WHEN V.calificacion = 5 THEN 1 ELSE 0 END) AS
cinco_estrellas
FROM
USUARIO.USUARIO U
JOIN
USUARIO.CONDUCTOR C ON U.id_usuario = C.id_usuario
JOIN
CONDUCTOR.[AUTO] A ON C.id_usuario = A.id_usuario
JOIN
VIAJE.VIAJE V ON A.id_auto = V.id_auto
GROUP BY
U.id_usuario, U.nombre_usuario, U.nombre, U.ap_paterno,
U.ap_materno, C.num_licencia
ORDER BY
U.id_usuario;

-- I) Listado de autos, placa, numero de serie, marca, modelo, año,
color y su dueño

SELECT * FROM CONDUCTOR.[AUTO]

SELECT
AU.id_auto,
AU.placa,
MA.nombre,
MO.nombre,
AU.año,
AU.color,
U.nombre_usuario as usuario_dueño,
U.nombre as nombre_dueño,
U.ap_paterno,
U.ap_materno,
C.num_licencia
FROM CONDUCTOR.AUTO AU
JOIN
CONDUCTOR.MODELO MO ON AU.id_modelo = MO.id_modelo
JOIN
CONDUCTOR.MARCA MA ON MO.id_marca = MA.id_marca
JOIN
USUARIO.CONDUCTOR C ON C.id_usuario = AU.id_usuario
JOIN
USUARIO.USUARIO U ON C.id_usuario = U.id_usuario

-- J) Listado de quejas incluyendo el conductor y auto, con filtro
para obtenerse
-- por un periodo de tiempo o por conductor

SELECT * FROM CLIENTE.QUEJA
```

```
-- Por periodo de tiempo

SELECT
    Q.id_queja,
    Q.titulo,
    Q.descripcion,
    Q.fecha_queja,
    Q.fecha_atencion,
    Q.dias_respuesta,
    U.nombre_usuario as usuario_conductor,
    U.nombre as nombre_conductor,
    U.ap_paterno,
    U.ap_materno,
    C.num_licencia,
    AU.id_auto,
    AU.placa,
    AU.color
FROM CLIENTE.QUEJA Q
JOIN
    USUARIO.USUARIO U ON Q.id_conductor = U.id_usuario
JOIN
    USUARIO.CONDUCTOR C ON C.id_usuario = U.id_usuario
JOIN
    CONDUCTOR.[AUTO] AU ON AU.id_usuario = C.id_usuario
WHERE
    Q.fecha_queja BETWEEN '2025-05-01' AND '2025-05-31'

-- Por conductor

SELECT
    Q.id_queja,
    Q.titulo,
    Q.descripcion,
    Q.fecha_queja,
    Q.fecha_atencion,
    Q.dias_respuesta,
    U.nombre_usuario as usuario_conductor,
    U.nombre as nombre_conductor,
    U.ap_paterno,
    U.ap_materno,
    C.num_licencia,
    AU.id_auto,
    AU.placa,
    AU.color
FROM CLIENTE.QUEJA Q
JOIN
    USUARIO.USUARIO U ON Q.id_conductor = U.id_usuario
JOIN
    USUARIO.CONDUCTOR C ON C.id_usuario = U.id_usuario
JOIN
    CONDUCTOR.[AUTO] AU ON AU.id_usuario = C.id_usuario
WHERE
    U.nombre_usuario = 'cperez_97'
```

```
/*

----- CONSULTAS -----

*/

-- Álgebra relacional: Obtener toda la información de los
-- clientes que tengan una calificación de 5 y al menos un descuento
-- que vence antes de julio del 2025

SELECT * FROM DESCUENTO.DESCUENTO

SELECT * FROM USUARIO.USUARIO U
JOIN DESCUENTO.DESCUENTO D
ON U.id_usuario = D.id_usuario
JOIN USUARIO.CLIENTE C
ON C.id_usuario = U.id_usuario
WHERE D.vigencia < '2025-07-01'

INTERSECT

SELECT * FROM USUARIO.USUARIO U
JOIN DESCUENTO.DESCUENTO D
ON U.id_usuario = D.id_usuario
JOIN USUARIO.CLIENTE C
ON C.id_usuario = U.id_usuario
WHERE C.calificacion = 5

/*

     $\theta = id\_usuario$ 

     $R1 \leftarrow (\sigma(D.vigencia < '2025-07-01')) (USUARIO \bowtie_{\theta} CLIENTE \bowtie_{\theta} DESCUENTO)$ 

     $\theta = id\_usuario$ 

     $R2 \leftarrow (\sigma(C.calificacion = 5)) (USUARIO \bowtie_{\theta} CLIENTE \bowtie_{\theta} DESCUENTO)$ 

     $R3 = R1 \cap R2$  (Resultado)

*/

-- Obtener el nombre completo y número de licencia de los
-- conductores cuyos pagos totales están arriba del promedio
-- Subconsulta + Funciones de agregación

SELECT AVG(total_pagos) as promedio_pagos -- Promedio
FROM (
    SELECT SUM(monto) AS total_pagos
    FROM CONDUCTOR.PAGO
    GROUP BY id_usuario
) AS promedios
```

```
SELECT
    C.id_usuario,
    U.nombre,
    U.ap_paterno,
    U.ap_materno,
    C.num_licencia,
    SUM(P.monto) AS total_pagos
FROM
    CONDUCTOR.PAGO P
JOIN
    USUARIO.CONDUCTOR C ON P.id_usuario = C.id_usuario
JOIN
    USUARIO.USUARIO U ON C.id_usuario = U.id_usuario
GROUP BY
    C.id_usuario, U.nombre, U.ap_paterno, U.ap_materno,
    C.num_licencia
HAVING
    SUM(P.monto) > (
        SELECT AVG(total_pagos)
        FROM (
            SELECT SUM(monto) AS total_pagos
            FROM CONDUCTOR.PAGO
            GROUP BY id_usuario
        ) AS promedios
    );

-- Porcentaje de pagos con tarjeta vs pagos en efectivo

SELECT
    FORMAT(
        (COUNT(CASE WHEN V.tipo = 'T' THEN 1 END) * 100.0 /
COUNT(*)),
        'N2'
    ) AS porcentaje_tarjeta,

    FORMAT(
        (COUNT(CASE WHEN V.tipo = 'E' THEN 1 END) * 100.0 /
COUNT(*)),
        'N2'
    ) AS porcentaje_efectivo
FROM
    VIAJE.VIAJE V;

-- Diferencia entre el total pagado y el total facturado de los
conductores (INNER JOIN)

SELECT * FROM CONDUCTOR.AUTO
SELECT * FROM VIAJE.FACTURA
SELECT * FROM CONDUCTOR.PAGO
SELECT * FROM USUARIO.CONDUCTOR

SELECT
    C.id_usuario,
    U.nombre,
    U.ap_paterno,
```



```
        U.ap_materno,
        C.num_licencia,
        YEAR(F.fecha_factura) AS año_facturacion,
        MONTH(F.fecha_factura) AS mes_facturacion,
        SUM(F.importe) AS total_facturado,
        SUM(P.monto) AS total_pagado,
        SUM(F.importe) - SUM(P.monto) AS diferencia
FROM
    USUARIO.CONDUCTOR C
INNER JOIN
    CONDUCTOR.AUTO A ON C.id_usuario = A.id_usuario
INNER JOIN
    VIAJE.VIAJE V ON A.id_auto = V.id_auto
INNER JOIN
    VIAJE.FACTURA F ON V.id_factura = F.id_factura
INNER JOIN
    CONDUCTOR.PAGO P ON C.id_usuario = P.id_usuario
    AND YEAR(F.fecha_factura) = YEAR(P.fecha)
    AND MONTH(F.fecha_factura) = MONTH(P.fecha)

INNER JOIN
    USUARIO.USUARIO U ON C.id_usuario = U.id_usuario
GROUP BY
    C.id_usuario,
    U.nombre,
    U.ap_paterno,
    U.ap_materno,
    C.num_licencia,
    YEAR(F.fecha_factura), -- Agrupamos por año y mes
    MONTH(F.fecha_factura)
ORDER BY
    año_facturacion DESC,
    mes_facturacion DESC;

-- Muestra cada usuario y quién lo recomendó (SELF LEFT JOIN)

SELECT
    U1.id_usuario AS id_usuario,
    U1.nombre AS nombre_usuario,
    U1.ap_paterno AS apellido_usuario,
    U1.correo AS correo_usuario,
    U2.id_usuario AS id_recomendador,
    U2.nombre AS nombre_recomendador,
    U2.ap_paterno AS apellido_recomendador,
    U2.correo AS correo_recomendador
FROM
    USUARIO.USUARIO U1
LEFT JOIN
    USUARIO.USUARIO U2 ON U1.id_recomendado = U2.id_usuario -- Self
Join
ORDER BY
    U1.id_usuario;
```

```
-- Usuarios que nunca han recomendado a alguien (SELF LEFT JOIN)

SELECT
    U.id_usuario,
    U.nombre,
    U.ap_paterno
FROM
    USUARIO.USUARIO U
LEFT JOIN
    USUARIO.USUARIO Recomendados ON U.id_usuario =
Recomendados.id_recomendado
WHERE
    Recomendados.id_recomendado IS NULL;

-- ¿Cuántos usuarios ha recomendado cada usuario? (Si es que ha
recomendado)

SELECT
    U2.id_usuario,
    U2.nombre_usuario,
    U2.nombre,
    U2.ap_paterno,
    U2.correo,
    COUNT(U1.id_usuario) AS total_recomendados
FROM
    USUARIO.USUARIO U1
JOIN
    USUARIO.USUARIO U2 ON U1.id_recomendado = U2.id_usuario -- Self
Join
GROUP BY
    U2.id_usuario, U2.nombre_usuario, U2.nombre, U2.ap_paterno,
U2.correo
ORDER BY
    total_recomendados DESC;
```

CargaInicial.sql

```
/*

AUTORES:
    AYALA HERNÁNDEZ MARÍA FERNANDA
    PORTILLA HERMENEGILDO ELIZABETH
    VALENZUELA ASCENCIO GUSTAVO
    JIMENEZ AYALA YORDI JOSUE

FECHA DE CREACION: 24 DE MAYO DE 2025

ULTIMA ACTUALIZACION: 25 DE MAYO DE 2025

DESCRIPCION: DML QUE CONTIENE LA CARGA INICIAL DE DATOS PARA POBLAR
LA BASE

*/
```

```
-- SELECCIONAMOS NUESTRA BASE PARA TRABAJAR SOBRE ELLA
```

```
USE Equipo09AppSafe
```

```
GO
```

```
----- POBLANDO LA TABLA USUARIO -----
```

```
SELECT * FROM USUARIO.USUARIO
```

```
INSERT INTO USUARIO.USUARIO VALUES
```

```
('cperez_97', 'cperez97@gmail.com', 'ClaveSegura#A1B2C3D4',  
'2024-01-01', 'Carlos', 'Pérez', 'López', 0, 1, NULL),  
( 'ana_gz23', 'ana.gz23@yahoo.com', 'SeguraClave$1A2B3C4D',  
'2024-01-02', 'Ana', 'Gómez', NULL, 1, 0, NULL),  
( 'luism_88', 'luism88@outlook.com', 'FuerteClave*123ABC45',  
'2024-01-03', 'Luis', 'Martínez', 'Ramírez', 0, 1, NULL),  
( 'torres.laura', 'torreslaura@hotmail.com', 'ABCclave*2024X1Y2Z3W',  
'2024-01-04', 'Laura', 'Torres', 'Díaz', 1, 0, NULL),  
( 'psan_21', 'pedros21@protonmail.com', 'Clave2024$ZXCvbnm123',  
'2024-01-05', 'Pedro', 'Sánchez', NULL, 0, 1, NULL);
```

```
INSERT INTO USUARIO.USUARIO VALUES
```

```
('reyes_maria91', 'maria.reyes91@icloud.com',  
'ClaveFuerte*0987ABCD', '2024-01-06', 'María', 'Reyes', 'Flores', 1,  
0, 2),  
( 'jose.morales', 'jmoral34@gmail.com', 'StrongClave#12X34Y56',  
'2024-01-07', 'José', 'Morales', 'Luna', 0, 1, NULL),  
( 'lucortega77', 'lucortega@yahoo.es', 'ClaveSeguro*Q1W2E3R4',  
'2024-01-08', 'Lucía', 'Ortega', NULL, 1, 0, 3),  
( 'diego.silva_9', 'dsilva@outlook.com', 'X1Y2Z3#2024ClaveABCb',  
'2024-01-09', 'Diego', 'Silva', 'Cruz', 1, 0, NULL),  
( 'isah_10', 'isabelherrera@hotmail.com', 'Clave$Seguro2024ZXCa',  
'2024-01-10', 'Isabel', 'Herrera', 'Jiménez', 1, 0, 5);
```

```
INSERT INTO USUARIO.USUARIO VALUES
```

```
('emiparedes33', 'eparedes@live.com', '2024Clave*ZXCvbn12AB',  
'2024-01-11', 'Emilia', 'Paredes', NULL, 0, 1, NULL),  
( 'manvarg_12', 'manvargas@gmail.com', 'Q1W2E3R4$Clave2024Z3',  
'2024-01-12', 'Manuel', 'Vargas', 'Salinas', 1, 0, NULL),  
( 'dnavarro22', 'dnavarro@aol.com', 'Clave2024*LMNOPQ12Aa',  
'2024-01-13', 'Daniela', 'Navarro', 'Velasco', 0, 1, 7),  
( 'rcarrillo_x', 'robcar@gmail.com', 'ClaveFuerte2024*LKJ1',  
'2024-01-14', 'Roberto', 'Carrillo', NULL, 1, 0, NULL),  
( 'fersuarez87', 'fersuarez@hotmail.es', 'Clave2024#1234ASDFGb',  
'2024-01-15', 'Fernanda', 'Suárez', 'Guerra', 1, 0, 9);
```

```
INSERT INTO USUARIO.USUARIO VALUES
```

```
('jorgedom88', 'jorge.dz@live.com.mx', '1234Clave$Seguro2025',  
'2024-01-16', 'Jorge', 'Dominguez', 'Pineda', 1, 0, 5),  
( 'rios.andrea', 'arios@icloud.com', 'ClaveFuerte*ZXCvbn11',  
'2024-01-17', 'Andrea', 'Ríos', 'Cervantes', 0, 1, 10),  
( 'ibcristian', 'cibarra@zoho.com', 'Fuerte$2024ClaveQW77',
```

```
'2024-01-18', 'Cristian', 'Ibarra', NULL, 1, 0, 11),
('campos.paty', 'pcampos@hotmail.com', '02Clave2024*ABCD1234',
'2024-01-19', 'Patricia', 'Campos', 'Zúñiga', 0, 1, 13),
('raulagui', 'raguilar@outlook.es', 'Clave1234$ASDFQWERAz',
'2024-01-20', 'Raúl', 'Aguilar', NULL, 1, 0, 14);

INSERT INTO USUARIO.USUARIO VALUES
('valdel99', 'vdelgado@protonmail.com', 'ClaveSegura#A1B2C3D4',
'2025-02-12', 'Valeria', 'Delgado', 'Bravo', 0, 1, 15),
('escesteban', 'esc.esteban@outlook.com', 'SeguraClave$1A2B3C4D',
'2025-05-01', 'Esteban', 'Escobar', NULL, 1, 0, NULL),
('gabi_leon', 'gleon@icloud.com', 'FuerteClave*123ABC45',
'2024-07-23', 'Gabriela', 'León', 'Serrano', 0, 1, 1),
('ivanmen24', 'imendoza@gmail.com', 'ABCclave*2024X1Y2Z3s',
'2025-11-22', 'Iván', 'Mendoza', 'Solís', 1, 0, NULL),
('natagar92', 'n.garcia@yahoo.com', 'aClave2024$ZXCvbnm12',
'2024-10-25', 'Natalia', 'García', NULL, 0, 1, 8);

INSERT INTO USUARIO.USUARIO VALUES
('loztomas', 'tlopez@outlook.com', 'ClaveFuerte*0987ABCD',
'2025-08-26', 'Tomás', 'López', 'Ramos', 1, 0, 7),
('cami.nuz', 'cnunez@hotmail.com', 'StrongClave#12X34Y5g',
'2025-05-13', 'Camila', 'Núñez', NULL, 0, 1, 3),
('franpena', 'fp.acosta@live.com', 'ClaveSeguro*Q1W2E3R4',
'2025-01-12', 'Francisco', 'Peña', 'Acosta', 1, 0, 12),
('elenarv29', 'eromero@yahoo.com', 'X1Y2Z3#2024ClaveAB12',
'2025-02-10', 'Elena', 'Romero', 'Valle', 0, 1, NULL),
('migcas_30', 'miguelcas@outlook.com', 'Clave$Seguro2024ZXwl',
'2024-06-14', 'Miguel', 'Castañeda', 'Silva', 1, 0, 4);

INSERT INTO USUARIO.USUARIO VALUES
('mariac_2024', 'mariacruz2024@gmail.com', 'ClaveSegura#123ABCDX',
'2024-03-15', 'María', 'Cruz', 'Gómez', 1, 0, 2),
('juanp_1988', 'juanp88@yahoo.com', 'SeguraClave$45MNOPQZ',
'2025-07-21', 'Juan', 'Pérez', 'Ramírez', 0, 1, 1),
('isabelr_77', 'isabelr77@outlook.com', 'FuerteClave*QW12ER34',
'2024-11-02', 'Isabel', 'Rodríguez', NULL, 1, 0, NULL),
('carlosm_99', 'carlosm99@hotmail.com', 'ABCclave*ZXC1VBN2345',
'2025-01-30', 'Carlos', 'Martínez', 'López', 0, 1, 4),
('paulas_55', 'paulas55@protonmail.com', 'Clave2024$ZXCVBNM123',
'2024-09-09', 'Paula', 'Santos', NULL, 1, 0, NULL);

SELECT * FROM USUARIO.USUARIO

----- POBLANDO LA TABLA CONDUCTOR
-----

-- ESTO SOLO FUNCIONA CON AZURE, LA SINTAXIS QUE SE NOS HABIA DADO
ERA PARA SQL SERVER

-- CREAMO EL MASTER KEY

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'FeidNosBautizo';

-- HACEMOS LA CREDENCIAL
```

```
CREATE DATABASE SCOPED CREDENTIAL ConductoresCred
WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
SECRET =
'sp=r&st=2025-05-25T01:05:01Z&se=2026-01-13T09:05:01Z&sv=2024-11-04&
sr=c&sig=5E%2BVYhuNS%2Bd2p%2BslvGOOZ7Xfn1c02AryWs3es1pTUNA%3D';

-- APUNTAMOS AL CONTENEDOR CON NUESTRAS IMAGENES CARGADAS

CREATE EXTERNAL DATA SOURCE ConductoresSrc
WITH (
    TYPE = BLOB_STORAGE,
    LOCATION =
'https://imagenesb.blob.core.windows.net/conductores',
    CREDENTIAL = ConductoresCred
);

-- INSERTAMOS LOS DATOS AHORA SI

SELECT * FROM USUARIO.USUARIO -- OBTENEMOS LOS USUARIOS CONDUCTORES
WHERE CONDUCTOR = 1

SELECT * FROM USUARIO.CONDUCTOR

INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    1,
    10000001,
    '2028-12-31',
    BulkColumn,
    'Soy Carlos, tengo 35 años y cuento con más de 10 años de
experiencia al volante en rutas urbanas y de larga distancia. Me
caracterizo por un manejo defensivo, puntualidad y excelente trato
al cliente. Poseo licencias vigentes y un historial impecable de
seguridad vial. Siempre estoy comprometido con la eficiencia, el
cuidado de la carga y el cumplimiento de horarios.'
FROM
OPENROWSET(
    BULK '1.jpg',
    DATA_SOURCE = 'ConductoresSrc',
    SINGLE_BLOB
) AS img;

-----

INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    3,
    10000003,
    '2027-02-15',
    BulkColumn,
    'Me llamo Luis, tengo 40 años y he trabajado como conductor
durante más de una década, recorriendo tanto rutas urbanas como
```

carreteras largas. Soy cuidadoso al volante, siempre priorizo la seguridad y el respeto hacia los demás conductores y pasajeros. Me caracterizo por la puntualidad y la responsabilidad en cada viaje. Cuento con licencias vigentes y un historial limpio, comprometido con entregar un servicio confiable y eficiente.'

```
FROM OPENROWSET(  
    BULK '3.jpg',  
    DATA_SOURCE = 'ConductoresSrc',  
    SINGLE_BLOB  
) AS img;  
  
-----  
  
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,  
vigencia_lic, foto, descripcion)  
SELECT  
    5,  
    10000005,  
    '2027-06-20',  
    BulkColumn,  
    'Soy Pedro, conductor con 38 años y más de 11 años de  
experiencia en el manejo de vehículos para transporte urbano y de  
larga distancia. Me destaco por mantener una conducción segura y  
responsable, siempre respetando las normas de tránsito. Valoro la  
puntualidad y la calidad del servicio que ofrezco a mis clientes.  
Poseo licencias vigentes y un historial sin incidentes, comprometido  
en garantizar viajes seguros y eficientes.'  
FROM OPENROWSET(  
    BULK '5.jpg',  
    DATA_SOURCE = 'ConductoresSrc',  
    SINGLE_BLOB  
) AS img;  
  
-----  
  
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,  
vigencia_lic, foto, descripcion)  
SELECT  
    7,  
    10000007,  
    '2028-03-10',  
    BulkColumn,  
    'José, 42 años, con amplia trayectoria en el manejo profesional  
de vehículos para transporte urbano e interurbano. Me esfuerzo por  
mantener siempre un entorno seguro y cómodo para los pasajeros,  
cumpliendo estrictamente con las normas de tránsito. Mi compromiso  
se centra en la eficiencia, la puntualidad y la excelencia en el  
servicio, respaldado por licencias vigentes y un historial  
intachable.'  
FROM OPENROWSET(  
    BULK '7.jpg',  
    DATA_SOURCE = 'ConductoresSrc',  
    SINGLE_BLOB  
) AS img;
```

```
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    11,
    10000011,
    '2028-08-25',
    BulkColumn,
    'Emilia, 37 años, conductora profesional con amplia experiencia
en transporte urbano e interurbano. Me dedico a brindar un servicio
seguro y cómodo, siempre respetando las normas de tránsito y
priorizando la satisfacción de los pasajeros. Soy puntual,
responsable y comprometida con la excelencia en cada viaje. Cuento
con licencias vigentes y un historial limpio que respalda mi
trayectoria.'
```

```
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    13,
    10000013,
    '2029-01-05',
    BulkColumn,
    'Daniela, con 36 años, llevo más de una década manejando en
diferentes rutas tanto urbanas como largas distancias. Mi prioridad
es garantizar la seguridad y comodidad de quienes viajan conmigo,
siempre respetando las leyes de tránsito. Me considero una
conductora responsable, puntual y dedicada a ofrecer un servicio de
calidad. Poseo licencias vigentes y un historial sin accidentes que
respalda mi profesionalismo.'
```

```
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    17,
    10000017,
    '2029-07-19',
    BulkColumn,
    'Andrea, 34 años, conductora experimentada en rutas urbanas y de
```

larga distancia. Me enfoco en brindar viajes seguros y confiables, manteniendo siempre una actitud profesional y respetuosa en la carretera. La puntualidad y el cuidado hacia los pasajeros son mis prioridades. Dispongo de licencias actualizadas y un historial limpio que refleja mi compromiso con la seguridad y el buen servicio.'

```
FROM OPENROWSET(  
    BULK '17.jpg',  
    DATA_SOURCE = 'ConductoresSrc',  
    SINGLE_BLOB  
) AS img;
```

```
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,  
vigencia_lic, foto, descripcion)  
SELECT  
    19,  
    10000019,  
    '2027-11-11',  
    BulkColumn,  
    'Patricia, 39 años, con amplia experiencia al volante en  
transporte urbano y regional. Me esfuerzo por garantizar trayectos  
seguros y cómodos, manteniendo siempre el respeto por las normas de  
tránsito y un trato amable hacia los pasajeros. Soy puntual,  
responsable y comprometida con la calidad del servicio. Cuento con  
licencias vigentes y un historial impecable que respalda mi  
dedicación profesional.'
```

```
FROM OPENROWSET(  
    BULK '19.jpg',  
    DATA_SOURCE = 'ConductoresSrc',  
    SINGLE_BLOB  
) AS img;
```

```
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,  
vigencia_lic, foto, descripcion)  
SELECT  
    21,  
    10000021,  
    '2028-04-30',  
    BulkColumn,  
    'Me llamo Valeria, tengo 37 años y he trabajado como conductora  
durante más de 12 años en distintas rutas del país. Me especializo  
en ofrecer un servicio eficiente y seguro, priorizando siempre la  
tranquilidad de los pasajeros. Soy organizada, cuidadosa con los  
vehículos que manejo y comprometida con cada recorrido. Poseo  
licencias vigentes y un historial de conducción sin incidentes.'
```

```
FROM OPENROWSET(  
    BULK '21.jpg',  
    DATA_SOURCE = 'ConductoresSrc',  
    SINGLE_BLOB  
) AS img;
```

```
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    23,
    10000023,
    '2029-09-22',
    BulkColumn,
    'Soy Gabriela, tengo 36 años y me dedico con pasión a la
conducción profesional. Durante mi trayectoria he aprendido a
adaptarme a distintos tipos de rutas y condiciones, manteniendo
siempre un enfoque en la seguridad y la responsabilidad. Disfruto
brindar un trato cordial a los pasajeros y cumplir con los horarios
establecidos. Cuento con licencias actualizadas y un expediente
limpio que respalda mi compromiso en el trabajo.'
```

```
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    25,
    10000025,
    '2027-05-07',
    BulkColumn,
    'Mi nombre es Natalia, tengo 40 años y más de 15 años de
experiencia como conductora profesional. Me destaco por ser precisa,
paciente y muy enfocada en la seguridad vial. He trabajado tanto en
zonas urbanas como en trayectos largos, siempre cuidando cada
detalle del servicio. Mi prioridad es ofrecer un viaje seguro,
eficiente y con excelente atención. Poseo licencias vigentes y un
historial sin sanciones.'
```

```
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    27,
    10000027,
    '2028-10-16',
    BulkColumn,
    'Soy Camila, tengo 33 años y llevo más de una década dedicada a
la conducción profesional. Me caracterizo por ser responsable,
```

```
organizada y tener un alto sentido del compromiso. He trabajado en
diferentes tipos de rutas y horarios, adaptándome con facilidad a
cada situación. Mi enfoque está en garantizar un viaje seguro,
cómodo y puntual. Cuento con licencias en regla y un historial de
conducción impecable.'
```

```
FROM OPENROWSET(
    BULK '27.jpg',
    DATA_SOURCE = 'ConductoresSrc',
    SINGLE_BLOB
) AS img;

-----

INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    29,
    10000029,
    '2029-03-13',
    BulkColumn,
    'Mi nombre es Elena, tengo 26 años y soy una conductora
comprometida y en formación constante. Aunque llevo apenas 2 años en
el área del transporte, he demostrado responsabilidad, atención a
las normas viales y un trato amable con los pasajeros. Me esfuerzo
cada día por mejorar mis habilidades al volante y ofrecer un
servicio seguro y confiable. Cuento con licencia vigente y gran
disposición para seguir creciendo profesionalmente.'
```

```
FROM OPENROWSET(
    BULK '29.jpg',
    DATA_SOURCE = 'ConductoresSrc',
    SINGLE_BLOB
) AS img;

-----

INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    32,
    10000032,
    '2027-07-28',
    BulkColumn,
    'Soy Juan, tengo 30 años y cuento con 5 años de experiencia como
conductor profesional. Durante este tiempo he trabajado
principalmente en rutas urbanas y de reparto, siempre priorizando la
seguridad y la puntualidad. Me considero una persona confiable,
paciente y con buen trato hacia los clientes. Estoy comprometido con
ofrecer un servicio eficiente y cumplo con todas las normativas de
tránsito vigentes.'
```

```
FROM OPENROWSET(
    BULK '32.jpg',
    DATA_SOURCE = 'ConductoresSrc',
    SINGLE_BLOB
) AS img;
```

```
INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    34,
    10000034,
    '2028-12-05',
    BulkColumn,
    'Mi nombre es Carlos, tengo 24 años y recientemente comencé mi
carrera como conductor profesional. En este primer año he adquirido
experiencia en rutas cortas y servicios locales, destacándome por mi
responsabilidad, respeto a las normas de tránsito y actitud
positiva. Aunque estoy empezando, me esfuerzo constantemente por
mejorar y aprender de cada recorrido. Estoy motivado y listo para
asumir nuevos retos en el camino.'
```

FROM OPENROWSET(

```
    BULK '34.jpg',
    DATA_SOURCE = 'ConductoresSrc',
    SINGLE_BLOB
) AS img;
```

SELECT * FROM USUARIO.CONDUCTOR

----- POBLANDO LA TABLA CLIENTE -----

```
SELECT * FROM USUARIO.USUARIO -- OBTENEMOS LOS USUARIOS CLIENTES
WHERE CLIENTE = 1
```

```
SELECT * FROM USUARIO.CLIENTE
```

```
INSERT INTO USUARIO.CLIENTE (id_usuario, rfc, direccion, numero,
fecha_registro, calificacion) VALUES
(2, 'JUAC850101AB3', 'Av. Insurgentes Sur 1200, Del Valle, CDMX',
5534127890, '2024-01-02', 5),
(4, 'MORL900202CD1', 'Calle Reforma 45, Centro Histórico, CDMX',
5612345678, '2024-01-04', 4),
(8, 'GARC990708GH9', 'Blvd. Puerto Aéreo 550, Venustiano Carranza,
CDMX', 5611987654, '2024-01-08', 4),
(9, 'RODJ010911IJ5', 'Calle Juárez 80, Cuauhtémoc, CDMX',
5567894321, '2024-01-09', 3),
(10, 'HERN030102KL8', 'Paseo de la Reforma 320, Juárez, CDMX',
5543218765, '2024-01-10', 5),
(14, 'VALG950623OP4', 'Calle Colima 62, Roma Norte, CDMX',
5566012233, '2024-01-14', 4),
(15, 'SANC961213QR1', 'Av. Juárez 15, Centro, CDMX', 5599223344,
'2024-01-15', 5),
(16, 'ZARJ991229ST0', 'Periférico Norte 88, Azcapotzalco, CDMX',
5610789078, '2024-01-16', 4),
(20, 'BAEZ890101WX3', 'Boulevard Puerto Aéreo 234, Venustiano
Carranza, CDMX', 5599098123, '2024-01-20', 5),
(22, 'NIEA900730YZ5', 'Prol. Paseo de la Reforma 125, Miguel
Hidalgo, CDMX', 5612233445, '2025-05-01', 5),
(24, 'REYG860416AC7', 'Av. Lázaro Cárdenas 3000, Coyoacán, CDMX',
5566567878, '2025-11-22', 4),
```

```
(26, 'PEPM880519BD2', 'Callejón 5 de Mayo 12, Tlalpan, CDMX',  
5613322110, '2025-08-26', 4),  
(28, 'AGUI970710CE4', 'Calle Yucatán 45, Condesa, CDMX', 5599912345,  
'2025-01-12', 5),  
(31, 'JIMY820305EG1', 'Camino Real 88, Tlalpan, CDMX', 5544887766,  
'2024-03-15', 3),  
(33, 'RAMS990112FG6', 'Calzada de la Viga 10, Iztapalapa, CDMX',  
5567012030, '2024-11-02', 5),  
(35, 'SALD940221GH3', 'Calz. Ignacio Zaragoza 78, Iztapalapa, CDMX',  
5611298734, '2024-09-09', 4);
```

```
INSERT INTO USUARIO.CLIENTE (id_usuario, rfc, direccion, numero,  
fecha_registro) VALUES -- ESTOS VALORES VAN A RECIBIR EL DEFAULT  
(6, 'LOPR920315EF7', 'Av. Universidad 300, Coyoacán, CDMX',  
5598761234, '2024-01-06'),  
(12, 'TORR870421MN2', 'Calz. de Tlalpan 15, Xochimilco, CDMX',  
5610554433, '2024-01-12'),  
(18, 'CRUR850318UV6', 'Av. Benito Juárez 400, Iztapalapa, CDMX',  
5561230987, '2024-01-18'),  
(30, 'DELL750824DF9', 'Av. del Taller 100, Iztacalco, CDMX',  
5614433221, '2024-06-14');
```

```
SELECT * FROM USUARIO.CLIENTE
```

```
----- POBLANDO LA TABLA PAGO -----
```

```
SELECT id_usuario, fecha_ingreso from USUARIO.USUARIO -- OBTENEMOS  
LA FECHA DE INRESO PARA LOS PAGOS  
WHERE CONDUCTOR = 1
```

```
SELECT * FROM CONDUCTOR.PAGO
```

```
INSERT INTO CONDUCTOR.PAGO (id_usuario, folio, fecha, monto) VALUES  
(1, '19328475', '2024-01-01', 1250.00),  
(1, '17283461', '2024-01-08', 1380.00),  
(1, '18592147', '2024-01-15', 1450.00),  
(1, '16273859', '2024-01-22', 1590.00),  
(1, '19928374', '2024-01-29', 1700.00),  
(1, '17485923', '2024-02-05', 1220.00),  
(1, '18374629', '2024-02-12', 1300.00),  
(1, '17649283', '2024-02-19', 1400.00),  
(1, '19827345', '2024-02-26', 1550.00),  
(1, '16473829', '2024-03-04', 1600.00),  
  
(3, '18273946', '2024-01-03', 950.00),  
(3, '19837452', '2024-01-10', 1050.00),  
(3, '16382947', '2024-01-17', 990.00),  
(3, '17928374', '2024-01-24', 1080.00),  
(3, '19384756', '2024-01-31', 1125.00),  
(3, '17563829', '2024-02-07', 1190.00),  
(3, '18472938', '2024-02-14', 980.00),  
(3, '19638274', '2024-02-21', 1050.00),  
(3, '16573829', '2024-02-28', 1100.00),  
(3, '17829463', '2024-03-06', 970.00),
```

```
(5, '18274596', '2024-01-05', 870.00),
(5, '17638294', '2024-01-12', 1020.00),
(5, '19473829', '2024-01-19', 950.00),
(5, '16384759', '2024-01-26', 1070.00),
(5, '19827345', '2024-02-02', 1150.00),
(5, '17294583', '2024-02-09', 980.00),
(5, '18649283', '2024-02-16', 1040.00),
(5, '17938465', '2024-02-23', 1110.00),
(5, '19384726', '2024-03-01', 1180.00),
(5, '16473821', '2024-03-08', 1250.00),

(7, '17384920', '2024-01-07', 1000.00),
(7, '19837465', '2024-01-14', 1075.00),
(7, '16293847', '2024-01-21', 1100.00),
(7, '18475639', '2024-01-28', 1030.00),
(7, '19384722', '2024-02-04', 990.00),
(7, '17583947', '2024-02-11', 1080.00),
(7, '19638472', '2024-02-18', 1120.00),
(7, '18274635', '2024-02-25', 1150.00),
(7, '17738462', '2024-03-03', 1170.00),
(7, '16473825', '2024-03-10', 1200.00),

(11, '19384759', '2024-01-11', 1250.00),
(11, '18573629', '2024-01-18', 1190.00),
(11, '17384961', '2024-01-25', 1110.00),
(11, '19837462', '2024-02-01', 1160.00),
(11, '18473926', '2024-02-08', 1225.00),
(11, '17638429', '2024-02-15', 1280.00),
(11, '16294837', '2024-02-22', 1330.00),
(11, '19384720', '2024-02-29', 1375.00),
(11, '17738426', '2024-03-07', 1420.00),
(11, '16573842', '2024-03-14', 1470.00),

(13, '18274639', '2024-01-13', 1180.00),
(13, '19483726', '2024-01-20', 1240.00),
(13, '17294835', '2024-01-27', 1300.00),
(13, '16384752', '2024-02-03', 1195.00),
(13, '19384762', '2024-02-10', 1235.00),
(13, '17583945', '2024-02-17', 1285.00),
(13, '18649275', '2024-02-24', 1340.00),
(13, '19837456', '2024-03-02', 1395.00),
(13, '16473829', '2024-03-09', 1450.00),
(13, '18274638', '2024-03-16', 1500.00),

(17, '19384762', '2024-01-17', 980.00),
(17, '17294831', '2024-01-24', 1020.00),
(17, '19837461', '2024-01-31', 1065.00),
(17, '16384722', '2024-02-07', 1120.00),
(17, '18473928', '2024-02-14', 1175.00),
(17, '17638492', '2024-02-21', 1230.00),
(17, '16294873', '2024-02-28', 1280.00),
(17, '19384731', '2024-03-06', 1335.00),
(17, '16573849', '2024-03-13', 1390.00),
(17, '17938476', '2024-03-20', 1450.00),
```

(19, '18473928', '2024-01-19', 1100.00),
 (19, '17384962', '2024-01-26', 1150.00),
 (19, '19837463', '2024-02-02', 1200.00),
 (19, '16384753', '2024-02-09', 1250.00),
 (19, '19384721', '2024-02-16', 1300.00),
 (19, '17638428', '2024-02-23', 1350.00),
 (19, '16294872', '2024-03-01', 1400.00),
 (19, '18274632', '2024-03-08', 1450.00),
 (19, '17738463', '2024-03-15', 1500.00),
 (19, '16473820', '2024-03-22', 1550.00),

(21, '19384725', '2025-02-12', 1400.00),
 (21, '18573629', '2025-02-19', 1450.00),
 (21, '17384962', '2025-02-26', 1500.00),
 (21, '19837425', '2025-03-05', 1550.00),
 (21, '18473927', '2025-03-12', 1600.00),
 (21, '17638427', '2025-03-19', 1650.00),
 (21, '16294839', '2025-03-26', 1700.00),
 (21, '19384764', '2025-04-02', 1750.00),
 (21, '17738428', '2025-04-09', 1800.00),
 (21, '16573840', '2025-04-16', 1850.00),

(23, '17384921', '2024-07-23', 1150.00),
 (23, '19837466', '2024-07-30', 1200.00),
 (23, '16293848', '2024-08-06', 1250.00),
 (23, '18475630', '2024-08-13', 1300.00),
 (23, '19384723', '2024-08-20', 1350.00),
 (23, '17583948', '2024-08-27', 1400.00),
 (23, '19638473', '2024-09-03', 1450.00),
 (23, '18274636', '2024-09-10', 1500.00),
 (23, '17738463', '2024-09-17', 1550.00),
 (23, '16473826', '2024-09-24', 1600.00),

(25, '19384760', '2024-10-25', 1120.00),
 (25, '18573628', '2024-11-01', 1170.00),
 (25, '17384960', '2024-11-08', 1220.00),
 (25, '19837460', '2024-11-15', 1270.00),
 (25, '18473925', '2024-11-22', 1320.00),
 (25, '17638425', '2024-11-29', 1370.00),
 (25, '16294835', '2024-12-06', 1420.00),
 (25, '19384761', '2024-12-13', 1470.00),
 (25, '17738425', '2024-12-20', 1520.00),
 (25, '16573845', '2024-12-27', 1570.00),

(27, '18274630', '2025-05-13', 1300.00),
 (27, '19483725', '2025-05-20', 1340.00),
 (27, '17294836', '2025-05-27', 1380.00),
 (27, '16384751', '2025-06-03', 1420.00),
 (27, '19384726', '2025-06-10', 1460.00),
 (27, '17583949', '2025-06-17', 1500.00),
 (27, '18649276', '2025-06-24', 1540.00),
 (27, '19837457', '2025-07-01', 1580.00),
 (27, '16473824', '2025-07-08', 1620.00),
 (27, '18274639', '2025-07-15', 1660.00),

```
(29, '19384763', '2025-02-10', 990.00),
(29, '17294832', '2025-02-17', 1040.00),
(29, '19837467', '2025-02-24', 1090.00),
(29, '16384727', '2025-03-03', 1140.00),
(29, '18473922', '2025-03-10', 1190.00),
(29, '17638422', '2025-03-17', 1240.00),
(29, '16294874', '2025-03-24', 1290.00),
(29, '19384728', '2025-03-31', 1340.00),
(29, '16573843', '2025-04-07', 1390.00),
(29, '17938470', '2025-04-14', 1440.00),
```

```
(32, '18473929', '2025-07-21', 1500.00),
(32, '17384963', '2025-07-28', 1550.00),
(32, '19837468', '2025-08-04', 1600.00),
(32, '16384754', '2025-08-11', 1650.00),
(32, '19384727', '2025-08-18', 1700.00),
(32, '17638421', '2025-08-25', 1750.00),
(32, '16294875', '2025-09-01', 1800.00),
(32, '18274633', '2025-09-08', 1850.00),
(32, '17738461', '2025-09-15', 1900.00),
(32, '16473823', '2025-09-22', 1950.00),
```

```
(34, '19384729', '2025-01-30', 2000.00),
(34, '18573625', '2025-02-06', 2050.00),
(34, '17384964', '2025-02-13', 2100.00),
(34, '19837428', '2025-02-20', 2150.00),
(34, '18473923', '2025-02-27', 2200.00),
(34, '17638426', '2025-03-06', 2250.00),
(34, '16294876', '2025-03-13', 2300.00),
(34, '19384765', '2025-03-20', 2350.00),
(34, '17738427', '2025-03-27', 2400.00),
(34, '16573841', '2025-04-03', 2450.00);
```

```
SELECT * FROM CONDUCTOR.PAGO
```

```
----- POBLANDO LA TABLA TIPO -----
```

```
SELECT * FROM DESCUENTO.TIPO
```

```
INSERT INTO DESCUENTO.TIPO (nombre) VALUES
```

```
('Primer viaje'),
('Recomendación de amigo'),
('Viaje frecuente'),
('Hora feliz'),
('Descuento por ruta larga'),
('Promoción fin de semana'),
('Descuento por tarjeta bancaria'),
('Viaje compartido'),
('Descuento por cancelación anterior'),
('Descuento por cumpleaños'),
('Promoción estudiantes'),
('Promoción empleados corporativos'),
('Descuento por baja demanda'),
('Promoción por evento especial'),
```

```
('Descuento por pago anticipado'),
('Promoción de temporada'),
('Descuento por calificación alta'),
('Descuento por suscripción mensual'),
('Promoción zona específica'),
('Descuento por combinación con delivery');

SELECT * FROM DESCUENTO.TIPO
ORDER BY id_tipo

----- POBLANDO LA TABLA DESCUENTO
-----

SELECT id_usuario FROM USUARIO.USUARIO
ORDER BY id_usuario

select id_tipo FROM DESCUENTO.TIPO
ORDER BY id_tipo

SELECT * FROM DESCUENTO.DESCUENTO

INSERT INTO DESCUENTO.DESCUENTO (importe, vigencia, id_usuario,
id_tipo) VALUES
(50.00, '2025-06-01', 1, 1),
(30.00, '2025-06-08', 1, 3),
(20.00, '2025-06-15', 2, 2),
(25.00, '2025-06-20', 3, 5),
(40.00, '2025-06-25', 4, 4),
(35.00, '2025-06-30', 5, 6),
(60.00, '2025-07-03', 5, 1),
(45.00, '2025-07-06', 6, 7),
(30.00, '2025-07-10', 8, 8),
(50.00, '2025-07-13', 9, 9),
(55.00, '2025-07-17', 10, 10),
(20.00, '2025-07-20', 12, 11),
(25.00, '2025-07-22', 12, 3),
(15.00, '2025-07-25', 14, 13),
(30.00, '2025-07-28', 15, 4),
(35.00, '2025-07-30', 17, 6),
(45.00, '2025-08-01', 18, 14),
(50.00, '2025-08-03', 18, 2),
(40.00, '2025-08-06', 19, 15),
(60.00, '2025-08-09', 20, 16),
(20.00, '2025-08-11', 21, 17),
(30.00, '2025-08-14', 23, 7),
(25.00, '2025-08-16', 24, 18),
(35.00, '2025-08-18', 25, 9),
(20.00, '2025-08-21', 27, 19),
(30.00, '2025-08-23', 29, 10),
(40.00, '2025-08-26', 30, 20),
(45.00, '2025-08-28', 31, 5),
(50.00, '2025-08-30', 33, 12),
(30.00, '2025-09-02', 34, 3),
(25.00, '2025-09-05', 35, 8),
(20.00, '2025-06-03', 2, 4),
```



```
(35.00, '2025-06-06', 3, 6),
(45.00, '2025-06-09', 4, 8),
(25.00, '2025-06-11', 6, 10),
(30.00, '2025-06-13', 7, 11),
(50.00, '2025-06-15', 8, 12),
(40.00, '2025-06-18', 9, 13),
(55.00, '2025-06-21', 10, 14),
(60.00, '2025-06-24', 11, 15),
(20.00, '2025-06-26', 13, 16),
(35.00, '2025-06-28', 14, 17),
(45.00, '2025-07-01', 16, 18),
(25.00, '2025-07-04', 17, 19),
(30.00, '2025-07-07', 19, 20),
(50.00, '2025-07-09', 21, 1),
(40.00, '2025-07-11', 22, 2),
(55.00, '2025-07-14', 23, 3),
(60.00, '2025-07-16', 24, 5),
(20.00, '2025-07-18', 25, 7),
(35.00, '2025-07-21', 26, 9),
(45.00, '2025-07-24', 28, 11),
(25.00, '2025-07-26', 29, 13),
(30.00, '2025-07-29', 30, 15),
(50.00, '2025-08-01', 32, 17),
(40.00, '2025-08-04', 33, 19),
(55.00, '2025-08-07', 34, 20),
(60.00, '2025-08-10', 35, 4),
(20.00, '2025-08-12', 1, 6),
(35.00, '2025-08-15', 2, 8),
(45.00, '2025-08-18', 3, 10);
```

```
SELECT * FROM DESCUENTO.DESCUENTO
```

```
----- POBLANDO LA TABLA MARCA -----
```

```
SELECT * FROM CONDUCTOR.MARCA
ORDER BY id_marca
```

```
INSERT INTO CONDUCTOR.MARCA (nombre) VALUES
```

```
('Toyota'),
('Honda'),
('Ford'),
('Chevrolet'),
('Nissan'),
('Volkswagen'),
('BMW'),
('Mercedes-Benz'),
('Audi'),
('Hyundai'),
('Kia'),
('Mazda'),
('Subaru'),
('Jeep'),
('Dodge'),
('Lexus'),
('Acura'),
```

```
('Mitsubishi'),
('Volvo'),
('Jaguar'),
('Land Rover'),
('Porsche'),
('Ferrari'),
('Lamborghini'),
('Bentley'),
('Rolls-Royce'),
('Alfa Romeo'),
('Infiniti'),
('Citroën'),
('Peugeot'),
('Renault'),
('Fiat'),
('Suzuki'),
('Mini'),
('GMC'),
('Chrysler'),
('Ram'),
('Cadillac'),
('Buick'),
('Lincoln'),
('Tesla'),
('McLaren'),
('Bugatti'),
('Genesis'),
('Tata'),
('Mahindra'),
('Skoda'),
('Seat'),
('Opel'),
('Isuzu');
```

```
SELECT * FROM CONDUCTOR.MARCA
ORDER BY id_marca
```

----- POBLANDO LA TABLA MODELO -----

```
SELECT * FROM CONDUCTOR.MODELO
ORDER BY id_modelo
```

```
INSERT INTO CONDUCTOR.MODELO (nombre, id_marca) VALUES
('Corolla', 1),
('Camry', 1),
('RAV4', 1),
('Prius', 1),
('Highlander', 1),
('Civic', 2),
('Accord', 2),
('CR-V', 2),
('Pilot', 2),
('Fit', 2),
('F-150', 3),
('Mustang', 3),
```

```
('Explorer', 3),
('Escape', 3),
('Fusion', 3),
('Silverado', 4),
('Malibu', 4),
('Equinox', 4),
('Tahoe', 4),
('Camaro', 4),
('Altima', 5),
('Sentra', 5),
('Rogue', 5),
('Versa', 5),
('Pathfinder', 5),
('Golf', 6),
('Passat', 6),
('Tiguan', 6),
('Jetta', 6),
('Atlas', 6),
('3 Series', 7),
('5 Series', 7),
('X3', 7),
('X5', 7),
('Z4', 7),
('C-Class', 8),
('E-Class', 8),
('GLC', 8),
('GLE', 8),
('S-Class', 8),
('A3', 9),
('A4', 9),
('Q5', 9),
('Q7', 9),
('A6', 9),
('Elantra', 10),
('Sonata', 10),
('Tucson', 10),
('Santa Fe', 10),
('Accent', 10),
('Soul', 11),
('Sorento', 11),
('Sportage', 11),
('Optima', 11),
('Forte', 11),
('Mazda3', 12),
('Mazda6', 12),
('CX-5', 12),
('CX-9', 12),
('MX-5 Miata', 12),
('Impreza', 13),
('Outback', 13),
('Forester', 13),
('Crosstrek', 13),
('Legacy', 13),
('Wrangler', 14),
('Grand Cherokee', 14),
```

```
('Compass', 14),
('Renegade', 14),
('Cherokee', 14),
('Charger', 15),
('Challenger', 15),
('Durango', 15),
('Journey', 15),
('Ram 1500', 15),
('RX', 16),
('ES', 16),
('NX', 16),
('IS', 16),
('GX', 16),
('TLX', 17),
('RDX', 17),
('MDX', 17),
('ILX', 17),
('NSX', 17),
('Outlander', 18),
('Eclipse Cross', 18),
('Mirage', 18),
('Lancer', 18),
('ASX', 18),
('XC40', 19),
('XC60', 19),
('XC90', 19),
('S60', 19),
('V60', 19),
('XE', 20),
('XF', 20),
('F-Pace', 20),
('E-Pace', 20),
('F-Type', 20),
('Discovery', 21),
('Range Rover', 21),
('Defender', 21),
('Evoque', 21),
('Velar', 21),
('911', 22),
('Cayenne', 22),
('Macan', 22),
('Panamera', 22),
('Taycan', 22),
('488 GTB', 23),
('F8 Tributo', 23),
('Portofino', 23),
('Roma', 23),
('SF90 Stradale', 23),
('Huracán', 24),
('Aventador', 24),
('Urus', 24),
('Continental GT', 25),
('Bentayga', 25),
('Flying Spur', 25),
('Phantom', 26),
```

```
('Ghost', 26),
('Cullinan', 26),
('Giulia', 27),
('Stelvio', 27),
('Tonale', 27),
('Q50', 28),
('Q60', 28),
('QX50', 28),
('QX60', 28),
('C3', 29),
('C4', 29),
('C5 Aircross', 29),
('208', 30),
('3008', 30),
('5008', 30),
('Clio', 31),
('Captur', 31),
('Megane', 31),
('500', 32),
('Panda', 32),
('Tipo', 32),
('Swift', 33),
('Vitara', 33),
('Jimny', 33),
('Cooper', 34),
('Countryman', 34),
('Clubman', 34),
('Sierra', 35),
('Yukon', 35),
('Terrain', 35),
('300', 36),
('Pacifica', 36),
('1500', 37),
('2500', 37),
('Escalade', 38),
('CT5', 38),
('Enclave', 39),
('Encore', 39),
('Navigator', 40),
('Aviator', 40),
('Model S', 41),
('Model 3', 41),
('Model X', 41),
('Model Y', 41),
('720S', 42),
('P1', 42),
('Artura', 42),
('Chiron', 43),
('Veyron', 43),
('G70', 44),
('G80', 44),
('GV80', 44),
('Nexon', 45),
('Harrier', 45),
('Thar', 46),
```

```
('XUV700', 46),
('Octavia', 47),
('Kodiaq', 47),
('Ibiza', 48),
('Leon', 48),
('Astra', 49),
('Corsa', 49),
('D-Max', 50),
('MU-X', 50);

SELECT * FROM CONDUCTOR.MODELO
ORDER BY id_modelo

----- POBLANDO LA TABLA AUTO -----

SELECT id_usuario FROM USUARIO.USUARIO
WHERE CONDUCTOR = 1

SELECT * FROM CONDUCTOR.[AUTO]

INSERT INTO CONDUCTOR.[AUTO] (año, placa, color, id_modelo,
id_usuario) VALUES
(2023, 'ABC12345', 'Rojo', 1, 1),
(2021, 'DEF56789', 'Azul', 15, 1),
(2024, 'GHI90123', 'Negro', 33, 3),
(2020, 'JKL34567', 'Blanco', 47, 5),
(2025, 'MNO78901', 'Gris', 89, 5),
(2022, 'PQR23456', 'Plata', 102, 7),
(2023, 'STU67890', 'Azul', 110, 11),
(2021, 'VWX01234', 'Rojo', 115, 11),
(2024, 'YZA45678', 'Negro', 134, 13),
(2020, 'BCD89012', 'Blanco', 141, 17),
(2025, 'EFG23456', 'Gris', 150, 19),
(2022, 'HIJ67890', 'Rojo', 160, 21),
(2023, 'KLM01234', 'Azul', 170, 23),
(2021, 'NOP45678', 'Negro', 175, 25),
(2024, 'QRS89012', 'Blanco', 180, 27),
(2020, 'TUV23456', 'Plata', 185, 29),
(2025, 'WXY67890', 'Gris', 186, 32),
(2022, 'ZAB01234', 'Azul', 120, 32),
(2023, 'CDE45678', 'Rojo', 95, 34),
(2021, 'FGH89012', 'Negro', 50, 34);

SELECT * FROM CONDUCTOR.[AUTO]

----- POBLANDO LA TABLA ADMINISTRADOR -----

SELECT * FROM USUARIO.ADMINISTRADOR

INSERT INTO USUARIO.ADMINISTRADOR (nombre_usuario, correo, clave,
fecha_ingreso, nombre, ap_paterno, ap_materno) VALUES
('admin01', 'm.lopez@gmail.com', 'Clave#A1B2C3D4E5F6G7',
'2022-03-12', 'Mario', 'Lopez', 'Gomez'),
('admin02', 'lucia.mtz@yahoo.com', 'Admin$ClaveF6G7H8I9J',
```

```
'2023-01-21', 'Lucía', 'Martinez', NULL),
('admin03', 'p.ramirez@outlook.com', 'Fuerte*Clavel23ABCDX',
'2022-07-08', 'Pablo', 'Ramirez', 'Santos'),
('admin04', 'vhernandez@protonmail.com', 'Admin$ClaveB3C4D5E6F',
'2023-11-05', 'Valeria', 'Hernandez', 'Flores'),
('admin05', 'rafa.torres@hotmail.com', 'Clave#123LMNOPQRSTUV',
'2022-06-01', 'Rafael', 'Torres', NULL);

INSERT INTO USUARIO.ADMINISTRADOR (nombre_usuario, correo, clave,
fecha_ingreso, nombre, ap_paterno, ap_materno, id_recomendado)
VALUES
('admin06', 'elena.gtz@gmail.com', 'Clave*UVWXYZ123456AB',
'2024-02-13', 'Elena', 'Gutierrez', 'Ramos', 1),
('admin07', 'carlos.sdz@outlook.com', 'Admin$ZXCVBNM1234567',
'2023-05-30', 'Carlos', 'Sanchez', 'Diaz', 2),
('admin08', 'l.vargas@protonmail.com', 'Clave#ABCDLMNOPQRST1',
'2022-09-15', 'Laura', 'Vargas', NULL, 3),
('admin09', 'hugo.ortega@yahoo.com', 'Seguro$123ABCXYZWERT',
'2024-03-04', 'Hugo', 'Ortega', 'Fernandez', 4),
('admin10', 'marta.cruz@gmail.com', 'Clave*2024WXYZ123ABC',
'2022-10-10', 'Marta', 'Cruz', NULL, 5),
('admin11', 'alejandro.m@hotmail.com', 'Clave*ASDFGHJKL12345',
'2023-08-18', 'Alejandro', 'Morales', 'Delgado', 1),
('admin12', 's.navarro@live.com', 'Admin#2024QWERTYUIOP',
'2022-12-25', 'Sandra', 'Navarro', NULL, 2),
('admin13', 'fernando.ibz@gmail.com', 'Fuerte$123456ABCXYZa',
'2024-01-09', 'Fernando', 'Ibarra', 'Mendoza', 3),
('admin14', 'cris.reyes@outlook.com', 'Clave*ZXC123456MNOPQ',
'2023-06-06', 'Cristina', 'Reyes', 'Vega', 4),
('admin15', 'esteban.silva@icloud.com', 'Clave#MNB098ASDFGHJK',
'2024-04-22', 'Esteban', 'Silva', NULL, 5);

INSERT INTO USUARIO.ADMINISTRADOR (nombre_usuario, clave,
fecha_ingreso, nombre, ap_paterno, ap_materno, id_recomendado)
VALUES
('admin16', 'Clave#MAYO2024ZXYWVU', '2023-01-11', 'Patricia',
'Campos', 'Lozano', 6),
('admin17', 'Fuerte*2024JIKLOPQR1', '2022-02-17', 'David',
'Gallardo', NULL, 7),
('admin18', 'Seguro$WERTY2024ZXCV', '2024-05-01', 'Isabel',
'Dominguez', 'Nieto', 8),
('admin19', 'Clave#XYZ123QWER123a', '2022-04-27', 'Jorge',
'Salinas', NULL, 9),
('admin20', 'Segura*TaYU789POI123', '2023-10-09', 'Monica',
'Pineda', 'Carrillo', 10);

SELECT * FROM USUARIO.ADMINISTRADOR

----- POBLANDO LA TABLA QUEJA -----

SELECT * FROM CLIENTE.QUEJA

INSERT INTO CLIENTE.QUEJA (
    descripcion, titulo, fecha_atencion, fecha_queja,
    id_usuario, id_administrador, id_conductor, id_auto
```

```
) VALUES
('Durante el viaje el auto se quedó varado por falla en el motor.
', 'Falla mecánica en ruta ', '2025-05-23', '2025-05-21', 2, 1, 11,
8),
('El GPS del auto falló en plena carretera, complicando el trayecto.
', 'GPS inservible ', '2025-05-22', '2025-05-20', 4, 2, 3, 3),
('Nos entregaron el auto tarde y perdimos la excursión del primer
día. ', 'Entrega tardía del vehículo ', '2025-05-21',
'2025-05-18', 6, 3, 29, 16),
('El vehículo no tenía llanta de refacción y sufrimos un
contratiempo. ', 'Falta de refacción ', '2025-05-24',
'2025-05-21', 8, 4, 5, 4),
('Nos dejaron esperando por más de 2 horas al llegar al aeropuerto.
', 'Retraso en entrega en aeropuerto ', '2025-05-23', '2025-05-19',
10, 5, 27, 15),
('El aire acondicionado dejó de funcionar a mitad del viaje. ',
'Falla en aire acondicionado ', '2025-05-24', '2025-05-20', 12, 6,
1, 1),
('El auto comenzó a emitir ruidos extraños en carretera. ',
'Problemas de motor en ruta ', '2025-05-22', '2025-05-19', 14, 7,
23, 13),
('Tuvimos problemas con el sistema de navegación al cruzar la
frontera. ', 'Falla en navegación ', '2025-05-21', '2025-05-18',
15, 8, 13, 9),
('El auto consumía más combustible de lo indicado. ', 'Consumo
excesivo de gasolina ', '2025-05-23', '2025-05-20', 16, 9, 32, 18),
('La atención al cliente durante el viaje fue nula ante una
emergencia. ', 'Sin asistencia en viaje ', '2025-05-24',
'2025-05-23', 18, 10, 7, 6);
```

```
INSERT INTO CLIENTE.QUEJA (
    descripcion, titulo, fecha_atencion, fecha_queja,
    id_usuario, id_administrador, id_conductor, id_auto
) VALUES
('Durante un viaje en abril, el vehículo se apagó repentinamente en
carretera. ', 'Apagón en ruta ', '2025-04-16', '2025-04-15', 30,
16, 25, 14),
('En marzo tuvimos que cambiar de auto a mitad del trayecto por una
falla crítica. ', 'Cambio de vehículo forzado ', '2025-03-29',
'2025-03-28', 31, 17, 1, 1),
('El sistema eléctrico del auto falló en un paso montañoso en
febrero. ', 'Falla eléctrica ', '2025-02-22', '2025-02-20', 33,
18, 17, 10),
('En enero, el auto no pasó una inspección policial por
documentación incompleta. ', 'Documentación incompleta ',
'2025-01-30', '2025-01-30', 35, 19, 7, 6),
('En un viaje de marzo, las llantas estaban en muy mal estado. ',
'Llantas desgastadas ', '2025-03-21', '2025-03-18', 2, 20, 32, 17);
```

```
INSERT INTO CLIENTE.QUEJA (
    descripcion, titulo, fecha_atencion, id_usuario,
    id_administrador, id_conductor, id_auto
) VALUES
```



```
('En medio del viaje se soltó una parte del parabrisas, poniendo en
riesgo la seguridad. ', 'Parabrisas suelto ', '2025-05-27', 14, 6,
1, 1),
('No funcionaba el cinturón del asiento trasero durante un viaje
familiar. ', 'Cinturón inservible ', '2025-05-28', 15, 7, 11, 8),
('El vehículo se entregó con el tanque casi vacío, complicando la
salida del viaje. ', 'Tanque vacío ', '2025-05-27', 16, 8, 19,
11),
('Durante el viaje por carretera, el sistema de frenos comenzó a
fallar. ', 'Frenos defectuosos ', '2025-05-27', 18, 9, 3, 3),
('Se activó el testigo del motor en plena autopista. ', 'Alerta de
motor ', '2025-05-27', 20, 10, 25, 14),
('El auto olía a combustible en el interior durante el trayecto. ',
'Olor a gasolina ', '2025-05-27', 22, 11, 7, 6),
('El sistema de audio no funcionaba, lo cual fue molesto en viajes
largos. ', 'Fallo en audio ', '2025-05-27', 24, 12, 27, 15),
('El auto presentaba vibraciones fuertes al superar los 80 km/h. ',
'Vibración peligrosa ', '2025-05-28', 26, 13, 5, 5),
('El velocímetro dejó de funcionar durante un viaje por autopista.
', 'Velocímetro dañado ', '2025-05-28', 28, 14, 17, 10),
('El control remoto del vehículo no abría las puertas a distancia.
', 'Control remoto fallido ', '2025-05-27', 30, 15, 13, 9);
```

```
SELECT * FROM CLIENTE.QUEJA
```

```
----- POBLANDO LA TABLA BANCO -----
```

```
SELECT * FROM CLIENTE.BANCO
```

```
INSERT INTO CLIENTE.BANCO (nombre) VALUES
('BBVA'),
('Santander'),
('Citibanamex'),
('HSBC'),
('Banorte'),
('Scotiabank'),
('Inbursa');
```

```
SELECT * FROM CLIENTE.BANCO
ORDER BY id_banco
```

```
----- POBLANDO LA TABLA TARJETA -----
```

```
SELECT * FROM CLIENTE.TARJETA
```

```
INSERT INTO CLIENTE.TARJETA (numero_tarjeta, vigencia, id_usuario,
id_banco) VALUES
('4532123412345678', '2026-08-31', 2, 1),
('5244334455667788', '2027-03-15', 4, 2),
('6011123456789012', '2028-10-20', 4, 3),
('3782822463100055', '2027-05-10', 6, 4),
('4000123412341234', '2026-12-31', 8, 1),
('4012888888881881', '2028-01-01', 8, 5),
('4222222222222222', '2029-09-09', 8, 6),
```

```
('4111111111111111', '2027-07-07', 10, 2),
('5500000000000000', '2028-04-14', 10, 7),
('6011000990139424', '2026-11-11', 12, 3),
('30569309025904AA', '2029-02-28', 14, 4),
('3530111333300000', '2027-06-30', 14, 5),
('6304000000000000', '2026-06-06', 15, 6),
('6767700000000000', '2029-03-20', 15, 2),
('6759649826438453', '2028-09-01', 15, 1),
('1234567812345678', '2028-12-12', 16, 7),
('5412345678901234', '2027-01-01', 18, 1),
('4539964888888888', '2029-05-05', 18, 6),
('4567890123456789', '2027-10-10', 20, 4),
('4987654321098765', '2026-04-04', 22, 5),
('4111111133334444', '2028-08-08', 22, 3),
('5555555555554444', '2027-07-17', 22, 6),
('6011444444444444', '2026-03-03', 24, 7),
('4111111199999999', '2029-01-01', 26, 1),
('4000000000000002', '2027-09-09', 26, 2),
('4111111177777777', '2028-05-15', 28, 3),
('4111111166666666', '2029-06-06', 30, 4),
('5105105105105100', '2028-11-11', 31, 5),
('4916111111111111', '2026-10-10', 31, 6),
('4532015112830366', '2027-04-04', 33, 7),
('4024007189110000', '2026-02-02', 35, 1),
('4111111112345678', '2028-07-07', 35, 3);
```

```
SELECT * FROM CLIENTE.TARJETA
```

```
----- POBLANDO LA TABLA DESCUENTO_ADMIN
-----
```

```
SELECT * FROM DESCUENTO.DESCUENTO_ADMIN
```

```
INSERT INTO DESCUENTO.DESCUENTO_ADMIN (importe, vigencia, id_tipo,
id_administrador) VALUES
```

```
(50.00, '2025-06-01', 1, 1),
(30.00, '2025-06-08', 5, 3),
(20.00, '2025-06-15', 8, 4),
(25.00, '2025-06-20', 2, 6),
(40.00, '2025-06-25', 3, 7),
(35.00, '2025-06-30', 6, 8),
(60.00, '2025-07-03', 10, 10),
(45.00, '2025-07-06', 12, 12),
(30.00, '2025-07-10', 4, 13),
(50.00, '2025-07-13', 15, 14),
(55.00, '2025-07-17', 11, 15),
(20.00, '2025-07-20', 7, 16),
(25.00, '2025-07-22', 13, 17),
(15.00, '2025-07-25', 9, 18),
(30.00, '2025-07-28', 14, 19),
(35.00, '2025-07-30', 16, 20),
(45.00, '2025-08-01', 18, 2),
(50.00, '2025-08-03', 20, 4),
(40.00, '2025-08-06', 17, 6),
(60.00, '2025-08-09', 19, 8),
```

```
(45.00, '2025-08-10', 2, 2),
(30.00, '2025-08-12', 3, 4),
(20.00, '2025-08-14', 5, 4),
(55.00, '2025-08-16', 6, 5),
(25.00, '2025-08-18', 8, 6),
(35.00, '2025-08-20', 10, 7),
(40.00, '2025-08-22', 1, 7),
(60.00, '2025-08-24', 12, 9),
(50.00, '2025-08-26', 13, 10),
(45.00, '2025-08-28', 14, 11),
(20.00, '2025-08-30', 15, 12),
(30.00, '2025-09-01', 16, 13),
(25.00, '2025-09-03', 17, 13),
(55.00, '2025-09-05', 18, 15),
(40.00, '2025-09-07', 19, 16),
(35.00, '2025-09-09', 20, 17),
(60.00, '2025-09-11', 4, 18),
(50.00, '2025-09-13', 6, 19),
(45.00, '2025-09-15', 7, 20),
(20.00, '2025-09-17', 9, 2),
(30.00, '2025-09-19', 11, 3),
(25.00, '2025-09-21', 13, 5),
(55.00, '2025-09-23', 1, 6),
(40.00, '2025-09-25', 2, 8),
(35.00, '2025-09-27', 3, 9),
(50.00, '2025-09-29', 5, 12),
(45.00, '2025-10-01', 10, 13),
(20.00, '2025-10-03', 16, 15),
(30.00, '2025-10-05', 17, 16),
(25.00, '2025-10-07', 20, 17);
```

```
SELECT * FROM DESCUENTO.DESCUENTO_ADMIN
ORDER BY id_descuento
```

----- POBLANDO LA TABLA TELEFONO -----

```
SELECT * FROM TELEFONO
```

```
INSERT INTO TELEFONO (id_administrador, numero) VALUES
```

```
(1, 5512345678),
(2, 5612345678),
(2, 5612345679),
(3, 5598765432),
(4, 5611112222),
(5, 5511122233),
(5, 5511122244),
(6, 5612349876),
(7, 5599988877),
(8, 5612340000),
(8, 5612340001),
(8, 5612340002),
(9, 5512349999),
(10, 5599998888),
(11, 5611223344),
(12, 5599887766),
```

```
(13, 5611778899),  
(14, 5511001100),  
(15, 5612334455),  
(15, 5612334456),  
(16, 5599554433),  
(17, 5612999888),  
(18, 5511223344),  
(19, 5611445566),  
(20, 5599223344),  
(20, 5599223345);
```

```
SELECT * FROM TELEFONO
```

```
----- POBLANDO LA TABLA ETIQUETA -----
```

```
SELECT * FROM VIAJE.ETIQUETA
```

```
INSERT INTO VIAJE.ETIQUETA (nombre) VALUES  
( 'Irrespetuoso' ),  
( 'Maneja muy rápido' ),  
( 'No respeta las reglas de vialidad' ),  
( 'Conducción agresiva' ),  
( 'Conductor amable' ),  
( 'Cumple con los límites de velocidad' ),  
( 'Conducción segura' ),  
( 'Usa el celular mientras maneja' ),  
( 'Muy cortés y educado' ),  
( 'No usa el cinturón de seguridad' ),  
( 'Hace paradas frecuentes' ),  
( 'Conduce con precaución en zonas escolares' ),  
( 'Ignora señales de tránsito' ),  
( 'Conduce cansado' ),  
( 'Conduce bajo influencia' ),  
( 'Muy paciente en el tráfico' ),  
( 'Habla mucho mientras conduce' ),  
( 'No respeta los semáforos' ),  
( 'Conduce de forma defensiva' );
```

```
SELECT * FROM VIAJE.ETIQUETA  
ORDER BY id_etiqueta
```

```
----- POBLANDO LA TABLA FACTURA -----
```

```
SELECT * FROM VIAJE.FACTURA
```

```
INSERT INTO VIAJE.FACTURA (fecha_factura, importe) VALUES  
( '2025-01-15', 378.80), -- (85.50, 44.30, 79.00, 55.00, 60.00,  
55.00)  
( '2025-02-03', 289.75), -- (100.00, 89.75, 50.00, 50.00)  
( '2025-03-12', 455.40), -- (150.00, 70.00, 65.40, 90.00, 80.00)  
( '2025-03-27', 132.00), -- (32.00, 25.00, 45.00, 30.00)  
( '2025-04-05', 525.60), -- (100.00, 125.60, 80.00, 70.00, 150.00)  
( '2025-04-29', 394.90), -- (120.00, 90.00, 89.90, 95.00)  
( '2025-05-01', 287.00), -- (50.00, 77.00, 80.00, 80.00)  
( '2025-05-10', 158.25), -- (45.00, 30.00, 28.25, 55.00)
```

```
('2025-05-22', 470.00), -- (90.00, 60.00, 100.00, 110.00, 110.00)
('2025-05-25', 333.33); -- (111.11, 45.00, 77.22, 100.00)
```

```
SELECT * FROM VIAJE.FACTURA
ORDER BY id_factura
```

```
----- POBLANDO LA TABLA ESTATUS -----
```

```
SELECT * FROM VIAJE.ESTATUS
```

```
INSERT INTO VIAJE.ESTATUS (nombre) VALUES
('Solicitado'),
('Programado'),
('Confirmado'),
('En curso'),
('Terminado'),
('Por cobrar'),
('Pagado'),
('Con adeudo'),
('Cancelado');
```

```
SELECT * FROM VIAJE.ESTATUS
ORDER BY id_estatus
```

```
----- POBLANDO LA TABLA VIAJE -----
```

```
SELECT * FROM VIAJE.VIAJE
```

```
INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta
)
```

```
VALUES
```

```
('2025-05-01', '10:00', 19.4326, 19.3900, -99.1332, -99.1500,
'2025-05-01', '10:00', 15,
'T', 'NO', 'Viaje puntual y eficiente', 85.50, 7, 5, 1, 7, 2, 1),
```

```
('2025-05-02', '12:00', 19.4400, 19.4100, -99.1400, -99.1600,
'2025-05-02', '12:20', 10,
'E', 'SI', 'Buena conducción', 44.30, 3, 11, 1, 3, 2, NULL),
```

```
('2025-05-03', '14:00', 19.4500, 19.3950, -99.1200, -99.1550,
'2025-05-03', '14:00', 20,
'T', 'NO', 'Excelente servicio', 79.00, 15, 3, 1, 5, 2, 1),
```

```
('2025-05-04', '08:30', 19.4200, 19.3800, -99.1100, -99.1450,
'2025-05-04', '08:30', 10,
'E', 'NO', 'Viaje económico', 55.00, 9, 8, 1, 2, 2, NULL),
```

```
('2025-05-05', '09:00', 19.4300, 19.4100, -99.1350, -99.1500,
```

```
'2025-05-05', '09:00', 15,
'T', 'SI', 'Conductor amable', 60.00, 18, 14, 1, 4, 2, 1),

('2025-05-06', '11:00', 19.4350, 19.4000, -99.1250, -99.1490,
'2025-05-06', '11:00', 20,
'E', 'NO', 'Llegamos tarde', 55.00, 1, 2, 1, 9, 2, NULL);

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta
)
VALUES
('2025-05-10', '09:00', 19.4300, 19.4200, -99.1300, -99.1400,
'2025-05-10', '09:00', NULL,
'T', 'NO', 'Viaje cómodo y rápido', 100.00, 4, 7, 2, 5, 10, 9),

('2025-05-11', '10:30', 19.4250, 19.4100, -99.1350, -99.1380,
'2025-05-12', '10:45', NULL,
'E', 'SI', 'Conductor atento', 89.75, 15, 12, 2, 5, 10, NULL),

('2025-05-12', '08:00', 19.4400, 19.4350, -99.1200, -99.1250,
'2025-05-12', '08:00', NULL,
'T', 'NO', 'Viaje sin contratiempos', 50.00, 7, 3, 2, 5, 10, 8),

('2025-05-13', '07:30', 19.4450, 19.4500, -99.1150, -99.1100,
'2025-05-13', '09:45', NULL,
'E', 'SI', 'Servicio excelente', 50.00, 20, 16, 2, 5, 10, NULL);

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta
)
VALUES
('2025-05-20', '08:00', 19.4305, 19.4203, -99.1315, -99.1403,
'2025-05-20', '08:00', NULL,
'T', 'NO', 'Viaje sin inconvenientes', 150.00, 3, 5, 3, 5, 15, 13),

('2025-05-21', '11:30', 19.4251, 19.4101, -99.1351, -99.1381,
'2025-05-22', '11:45', NULL,
'E', 'SI', 'Servicio puntual', 70.00, 8, 11, 3, 6, 15, NULL),

('2025-05-22', '09:00', 19.4402, 19.4352, -99.1202, -99.1252,
'2025-05-22', '09:00', NULL,
```

```
'T', 'NO', 'Viaje rápido', 65.40, 14, 9, 3, 5, 15, 15),
('2025-05-23', '07:30', 19.4453, 19.4503, -99.1153, -99.1103,
'2025-05-23', '07:30', NULL,
'E', 'NO', 'Conductor amable', 90.00, 17, 14, 3, 6, 15, NULL),
('2025-05-24', '10:15', 19.4354, 19.4454, -99.1304, -99.1404,
'2025-05-24', '10:15', NULL,
'T', 'NO', 'Viaje tranquilo', 80.00, 6, 7, 3, 5, 15, 15);

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta
)
VALUES
('2025-05-10', '08:10', 19.4300, 19.4310, -99.1300, -99.1310,
'2025-05-10', '08:10', NULL,
'T', 'NO', 'Viaje puntual', 32.00, 2, 3, 4, 5, 6, 4),
('2025-05-11', '09:20', 19.4250, 19.4265, -99.1350, -99.1365,
'2025-05-12', '09:35', 15.00,
'E', 'SI', 'Servicio amable', 25.00, 5, 7, 4, 6, 6, NULL),
('2025-05-12', '10:30', 19.4400, 19.4420, -99.1200, -99.1220,
'2025-05-12', '10:30', NULL,
'T', 'NO', 'Viaje rápido', 45.00, 12, 9, 4, 5, 6, 4),
('2025-05-13', '11:40', 19.4450, 19.4465, -99.1150, -99.1165,
'2025-05-13', '11:40', 10.00,
'E', 'NO', 'Conductor atento', 30.00, 15, 6, 4, 6, 6, NULL);

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta
)
VALUES
('2025-05-10', '07:30', 19.4200, 19.4210, -99.1400, -99.1410,
'2025-05-10', '07:30', 15,
'T', 'NO', 'Viaje sin inconvenientes', 100.00, 3, 2, 5, 5, 35, 31),
('2025-05-11', '09:00', 19.4300, 19.4315, -99.1300, -99.1315,
```

```
'2025-05-12', '09:10', 20,
'E', 'SI', 'Viaje programado', 125.60, 7, 4, 5, 6, 35, NULL),

('2025-05-12', '10:15', 19.4400, 19.4420, -99.1200, -99.1220,
'2025-05-12', '10:15', NULL,
'T', 'NO', 'Conductor amable', 80.00, 10, 10, 5, 5, 35, 32),

('2025-05-13', '11:45', 19.4450, 19.4465, -99.1150, -99.1165,
'2025-05-13', '11:45', 10,
'E', 'NO', 'Servicio correcto', 70.00, 18, 7, 5, 6, 35, NULL),

('2025-05-14', '13:00', 19.4500, 19.4515, -99.1100, -99.1115,
'2025-05-14', '13:00', NULL,
'T', 'NO', 'Viaje rápido', 150.00, 20, 9, 5, 5, 35, 31);

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta, calificacion
)
VALUES
('2025-05-20', '08:00', 19.4100, 19.4120, -99.1500, -99.1520,
'2025-05-20', '08:00', 15,
'T', 'NO', 'Buen viaje', 120.00, 5, 12, 6, 5, 30, 27, 4),

('2025-05-21', '09:30', 19.4205, 19.4230, -99.1405, -99.1430,
'2025-05-22', '09:30', 10,
'E', 'SI', 'Viaje programado', 90.00, 8, 8, 6, 6, 30, NULL, 3),

('2025-05-22', '10:45', 19.4305, 19.4340, -99.1305, -99.1340,
'2025-05-22', '10:45', NULL,
'T', 'NO', 'Servicio puntual', 89.90, 12, 15, 6, 5, 30, 27, 5),

('2025-05-23', '12:15', 19.4400, 19.4425, -99.1200, -99.1225,
'2025-05-23', '12:15', NULL,
'E', 'NO', 'Sin comentarios', 95.00, 16, 6, 6, 6, 30, NULL, 2);

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta, calificacion
)
VALUES
('2025-05-22', '07:45', 19.4000, 19.4050, -99.1800, -99.1850,
```



```
'2025-05-22', '07:45', 10,
'T', 'NO', 'Buen servicio', 50.00, 7, 9, 7, 5, 28, 26, 4),

('2025-05-22', '09:00', 19.4100, 19.4160, -99.1700, -99.1760,
'2025-05-23', '09:20', 15,
'E', 'SI', 'Un poco tarde', 77.00, 10, 3, 7, 6, 28, NULL, 3),

('2025-05-23', '10:30', 19.4200, 19.4280, -99.1600, -99.1680,
'2025-05-23', '10:30', NULL,
'T', 'NO', 'Viaje normal', 80.00, 12, 16, 7, 5, 28, 26, 5),

('2025-05-23', '11:15', 19.4300, 19.4380, -99.1500, -99.1580,
'2025-05-23', '11:15', 20,
'E', 'NO', 'Sin problemas', 80.00, 14, 2, 7, 6, 28, NULL, 2);

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta, calificacion
)
VALUES
('2025-05-24', '08:10', 19.4200, 19.4285, -99.1600, -99.1685,
'2025-05-24', '08:10', 10,
'T', 'NO', 'Buen servicio', 45.00, 8, 5, 8, 5, 14, 11, 4),

('2025-05-24', '09:20', 19.4100, 19.4170, -99.1700, -99.1770,
'2025-05-25', '09:40', 15,
'E', 'SI', 'Viaje regular', 30.00, 13, 14, 8, 6, 14, NULL, 3),

('2025-05-25', '10:00', 19.4300, 19.4385, -99.1500, -99.1585,
'2025-05-25', '10:00', NULL,
'T', 'NO', 'Manejo brusco y malas actitudes', 28.25, 16, 11, 8, 6,
14, 12, 1),

('2025-05-25', '11:45', 19.4400, 19.4490, -99.1400, -99.1490,
'2025-05-25', '11:45', 20,
'E', 'NO', 'Excelente conductor y puntual', 55.00, 6, 1, 8, 5, 14,
NULL, 5);

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta, calificacion
)
)
```

```
VALUES
('2025-05-24', '07:10', 19.4000, 19.4050, -99.1400, -99.1450,
'2025-05-24', '07:10', 15,
'T', 'NO', 'Excelente servicio, muy amable', 90.00, 3, 2, 9, 5, 16,
16, 5),

('2025-05-24', '08:15', 19.4100, 19.4150, -99.1500, -99.1550,
'2025-05-25', '08:35', NULL,
'E', 'SI', 'Conductor algo distraído pero cumplió', 60.00, 7, 9, 9,
6, 16, NULL, 3),

('2025-05-25', '09:05', 19.4200, 19.4280, -99.1600, -99.1680,
'2025-05-25', '09:05', 20,
'T', 'NO', 'Viaje sin contratiempos, todo excelente', 100.00, 11,
5, 9, 5, 16, 16, 5),

('2025-05-25', '10:10', 19.4300, 19.4380, -99.1700, -99.1780,
'2025-05-25', '10:10', 15,
'E', 'NO', 'Conductor un poco acelerado', 110.00, 15, 13, 9, 6, 16,
NULL, 2),

('2025-05-25', '11:20', 19.4400, 19.4480, -99.1800, -99.1880,
'2025-05-25', '11:20', 10,
'T', 'NO', 'Muy buen trato, música agradable', 110.00, 19, 17, 9,
5, 16, 16, 4);

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta, calificacion
)
VALUES
('2025-05-25', '07:45', 19.4500, 19.4580, -99.1900, -99.1980,
'2025-05-25', '07:45', 20,
'T', 'NO', 'Excelente atención y conducción impecable', 111.11, 4,
3, 10, 5, 20, 19, 5),

('2025-05-25', '09:10', 19.4600, 19.4650, -99.2000, -99.2050,
'2025-05-25', '09:10', NULL,
'E', 'NO', 'Conductor algo callado pero cumplido', 45.00, 8, 14,
10, 5, 20, NULL, 4),

('2025-05-25', '10:15', 19.4700, 19.4750, -99.2100, -99.2150,
'2025-05-26', '10:40', 15,
'T', 'SI', 'Buen viaje, llegó puntual', 77.22, 12, 6, 10, 6, 20,
19, 4),

('2025-05-25', '11:20', 19.4800, 19.4880, -99.2200, -99.2280,
'2025-05-25', '11:20', 10,
```

```
'E', 'NO', 'Muy buen trato y buena ruta', 100.00, 16, 11, 10, 6,
20, NULL, 5);

-- VIAJES QUE FUERON CANCELADOS

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta, calificacion
)
VALUES
('2022-12-24', '09:10', 19.4202, 19.4685, -99.1630, -99.1085,
'2022-12-24', '09:10', NULL,
'T', 'NO', 'Viaje cancelado 1', 05.00, 8, NULL, NULL, 9, 14, 11,
1),

('2021-05-24', '08:20', 19.4100, 19.4170, -99.1700, -99.1770,
'2021-05-25', '09:10', NULL,
'E', 'SI', 'Viaje cancelado 2', 05.00, 13, NULL, NULL, 9, 14, NULL,
1),

('2020-05-22', '09:00', 19.4300, 19.4385, -99.1500, -99.1585,
'2020-05-22', '09:00', NULL,
'T', 'NO', 'Viaje cancelado 3', 05.00, 16, NULL, NULL, 9, 14, 12,
1),

('2012-02-25', '11:00', 19.4400, 19.4490, -99.1400, -99.1490,
'2012-02-25', '11:00', NULL,
'E', 'NO', 'Viaje cancelado 4', 05.00, 6, NULL, NULL, 9, 14, NULL,
1);

-- VIAJES QUE QUEDARON CON ADEUDO

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta, calificacion
)
VALUES
('2021-05-24', '08:20', 19.4100, 19.4170, -99.1700, -99.1770,
'2021-05-25', '09:40', 20,
'E', 'SI', 'Sin comentario', 234.00, 13, NULL, NULL, 8, 9, NULL,
4),

('2020-05-22', '09:00', 19.4300, 19.4385, -99.1500, -99.1585,
'2020-05-22', '09:00', 15,
'E', 'NO', 'Lento', 100.00, 11, NULL, NULL, 8, 10, NULL, 3),
```

```
('2012-02-25', '11:00', 19.4400, 19.4490, -99.1400, -99.1490,
'2012-02-25', '11:00', 15,
'E', 'NO', 'Le quede a deber :(', 340.00, 3, NULL, NULL, 8, 12,
NULL, 5);

-- VIAJES PROGRAMADOS

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta, calificacion
)
VALUES
('2023-01-15', '07:45', 19.4202, 19.4685, -99.1630, -99.1085,
'2023-01-16', '08:00', 10,
'T', 'SI', 'Buen viaje', 590.00, 8, NULL, NULL, 2, 14, 11, 4),

('2024-04-10', '15:20', 19.4100, 19.4170, -99.1700, -99.1770,
'2024-04-11', '07:30', 20,
'E', 'SI', 'Increible conductor', 234.00, 13, NULL, NULL, 2, 18,
NULL, 5),

('2022-08-05', '10:00', 19.4300, 19.4385, -99.1500, -99.1585,
'2022-08-06', '10:30', 15,
'E', 'SI', 'Traslado al aeropuerto', 100.00, 11, NULL, NULL, 2, 2,
NULL, 3),

('2022-09-18', '16:15', 19.4300, 19.4385, -99.1500, -99.1585,
'2022-09-19', '17:00', 15,
'E', 'SI', 'Me gusto el conductor', 120.00, 11, NULL, NULL, 2, 33,
NULL, 4),

('2023-12-05', '12:30', 19.4400, 19.4490, -99.1400, -99.1490,
'2023-12-06', '13:00', 15,
'T', 'SI', 'Viaje rapido', 340.00, 3, NULL, NULL, 2, 14, 12, 5);

-- VIAJES SOLICITADOS

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
    latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
    propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
    id_factura,
    id_estatus, id_usuario, id_tarjeta, calificacion
)
VALUES
('2025-05-01', '06:15', 19.4250, 19.4600, -99.1650, -99.1050,
'2025-05-01', '06:45', 10,
```

```
'T', 'SI', 'Buen servicio', 600.00, 8, NULL, NULL, 1, 14, 11, 5),
('2025-04-20', '18:40', 19.4095, 19.4190, -99.1720, -99.1785,
'2025-04-20', '18:40', 15,
'E', 'NO', 'Amo la app', 240.00, 10, NULL, NULL, 1, 10, NULL, 4),
('2025-03-15', '08:00', 19.4310, 19.4390, -99.1510, -99.1590,
'2025-03-15', '08:30', 10,
'E', 'SI', 'Amo al conductor', 105.00, 11, NULL, NULL, 1, 4, NULL,
3),
('2025-02-28', '13:30', 19.4320, 19.4400, -99.1490, -99.1570,
'2025-02-28', '13:30', 20,
'E', 'NO', 'Increible servicio', 130.00, 11, NULL, NULL, 1, 6,
NULL, 5),
('2025-02-28', '13:30', 19.4320, 19.4400, -99.1490, -99.1570,
'2025-02-28', '13:30', 20,
'E', 'NO', 'Bonito carro', 130.00, 12, NULL, NULL, 1, 33, NULL, 5),
('2025-01-10', '10:45', 19.4430, 19.4500, -99.1410, -99.1480,
'2025-01-10', '10:45', 15,
'T', 'NO', 'Solicitado desde app web', 355.00, 9, NULL, NULL, 1,
14, 12, 4);

SELECT * FROM VIAJE.VIAJE

----- POBLANDO LA TABLA VIAJE_ESTATUS
-----

SELECT * FROM VIAJE.VIAJE_ESTATUS

INSERT INTO VIAJE.VIAJE_ESTATUS (id_estatus, id_viaje,
fecha_estatus, hora_estatus)
VALUES
(1, 1, '2025-05-01', '09:45'),
(3, 1, '2025-05-01', '09:50'),
(4, 1, '2025-05-01', '10:15'),
(5, 1, '2025-05-01', '10:45'),
(6, 1, '2025-05-01', '11:00'),
(7, 1, '2025-05-01', '12:00'),
(1, 2, '2025-05-02', '11:00'),
(2, 2, '2025-05-02', '11:15'),
(3, 2, '2025-05-02', '11:30'),
(1, 3, '2025-05-03', '13:20'),
(3, 3, '2025-05-03', '13:30'),
(4, 3, '2025-05-03', '14:30'),
(5, 3, '2025-05-03', '15:00'),
(1, 4, '2025-05-04', '08:00'),
(2, 4, '2025-05-04', '08:10'),
(1, 5, '2025-05-05', '08:00'),
(2, 5, '2025-05-05', '08:10'),
(3, 5, '2025-05-05', '08:30'),
(4, 5, '2025-05-05', '09:20'),
(1, 6, '2025-05-06', '10:00'),
```

```
(3, 6, '2025-05-06', '10:30')
```

```
INSERT INTO VIAJE.VIAJE_ESTATUS (id_estatus, id_viaje,  
fecha_estatus, hora_estatus)
```

```
VALUES
```

```
(1, 7, '2025-05-10', '08:30'),  
(3, 7, '2025-05-10', '08:45'),  
(4, 7, '2025-05-10', '09:15'),  
(5, 7, '2025-05-10', '09:45'),  
(1, 8, '2025-05-11', '09:45'),  
(2, 8, '2025-05-11', '10:00'),  
(3, 8, '2025-05-11', '10:15'),  
(4, 8, '2025-05-12', '10:45'),  
(5, 8, '2025-05-12', '11:15'),  
(1, 9, '2025-05-12', '07:20'),  
(3, 9, '2025-05-12', '07:30'),  
(4, 9, '2025-05-12', '08:10'),  
(5, 9, '2025-05-12', '08:40'),  
(1, 10, '2025-05-13', '06:45'),  
(2, 10, '2025-05-13', '07:00'),  
(3, 10, '2025-05-13', '07:15'),  
(4, 10, '2025-05-13', '09:45'),  
(5, 10, '2025-05-13', '10:15');
```

```
INSERT INTO VIAJE.VIAJE_ESTATUS (id_estatus, id_viaje,  
fecha_estatus, hora_estatus)
```

```
VALUES
```

```
(1, 11, '2025-05-20', '07:30'),  
(3, 11, '2025-05-20', '07:45'),  
(4, 11, '2025-05-20', '08:15'),  
(5, 11, '2025-05-20', '08:45'),  
(1, 12, '2025-05-21', '11:00'),  
(2, 12, '2025-05-21', '11:15'),  
(3, 12, '2025-05-22', '11:30'),  
(4, 12, '2025-05-22', '11:45'),  
(5, 12, '2025-05-22', '12:15'),  
(6, 12, '2025-05-22', '12:30'),  
(1, 13, '2025-05-22', '08:30'),  
(3, 13, '2025-05-22', '08:45'),  
(4, 13, '2025-05-22', '09:10'),  
(5, 13, '2025-05-22', '09:40'),  
(1, 14, '2025-05-23', '07:00'),  
(3, 14, '2025-05-23', '07:15'),  
(4, 14, '2025-05-23', '07:45'),  
(5, 14, '2025-05-23', '08:15'),  
(6, 14, '2025-05-23', '08:30'),  
(1, 15, '2025-05-24', '09:45'),  
(3, 15, '2025-05-24', '10:00'),  
(4, 15, '2025-05-24', '10:30'),  
(5, 15, '2025-05-24', '11:00');
```

```
-- Viajes cancelados
```

```
INSERT INTO VIAJE.VIAJE_ESTATUS (id_estatus, id_viaje,
fecha_estatus, hora_estatus)
VALUES
(1, 46, '2022-12-24', '08:25'),
(3, 46, '2022-12-24', '09:10'),
(9, 46, '2022-12-24', '09:25'),
(1, 47, '2021-05-25', '08:40'),
(3, 47, '2021-05-25', '09:20'),
(9, 47, '2021-05-25', '09:40'),
(1, 48, '2020-05-22', '09:20'),
(3, 48, '2020-05-22', '10:00'),
(9, 48, '2020-05-22', '10:20'),
(1, 49, '2012-02-25', '10:00'),
(3, 49, '2012-02-25', '11:30'),
(9, 49, '2012-02-25', '12:00');
```

-- Viajes con adeudos

```
INSERT INTO VIAJE.VIAJE_ESTATUS (id_estatus, id_viaje,
fecha_estatus, hora_estatus)
VALUES
(1, 50, '2021-05-25', '07:10'),
(2, 50, '2021-05-25', '07:40'),
(3, 50, '2021-05-25', '08:10'),
(4, 50, '2021-05-25', '09:40'),
(5, 50, '2021-05-25', '10:10'),
(6, 50, '2021-05-25', '10:40'),
(8, 50, '2021-05-25', '11:10'),
(1, 51, '2020-05-22', '06:30'),
(2, 51, '2020-05-22', '07:00'),
(3, 51, '2020-05-22', '08:00'),
(4, 51, '2020-05-22', '09:00'),
(5, 51, '2020-05-22', '09:30'),
(6, 51, '2020-05-22', '10:00'),
(8, 51, '2020-05-22', '10:30'),
(1, 52, '2012-02-25', '07:00'),
(2, 52, '2012-02-25', '08:00'),
(3, 52, '2012-02-25', '10:00'),
(4, 52, '2012-02-25', '11:00'),
(5, 52, '2012-02-25', '11:30'),
(6, 52, '2012-02-25', '12:00'),
(8, 52, '2012-02-25', '12:30');
```

-- Viajes programados

```
INSERT INTO VIAJE.VIAJE_ESTATUS (id_estatus, id_viaje,
fecha_estatus, hora_estatus)
VALUES
(1, 53, '2023-01-16', '07:30'),
(2, 53, '2023-01-16', '08:00'),
(1, 54, '2024-04-11', '07:00'),
(2, 54, '2024-04-11', '07:30'),
(1, 55, '2022-08-06', '10:00'),
(2, 55, '2022-08-06', '10:30'),
(1, 56, '2022-09-19', '16:30');
```

```
(2, 56, '2022-09-19', '17:00'),
(1, 57, '2023-12-06', '12:30'),
(2, 57, '2023-12-06', '13:00');

-- Viajes solicitados

INSERT INTO VIAJE.VIAJE_ESTATUS (id_estatus, id_viaje,
fecha_estatus, hora_estatus)
VALUES
(1, 58, '2025-05-01', '06:45'),
(1, 59, '2025-04-20', '18:40'),
(1, 60, '2025-03-15', '08:30'),
(1, 61, '2025-02-28', '13:30'),
(1, 62, '2025-02-28', '13:30'),
(1, 63, '2025-01-10', '10:45');

SELECT * FROM VIAJE.VIAJE_ESTATUS

----- POBLANDO LA TABLA UBICACION
-----

SELECT * FROM VIAJE.UBICACION

INSERT INTO VIAJE.UBICACION (latitud, longitud, id_viaje) VALUES
(19.4326, -99.1332, 1),
(19.4330, -99.1335, 1),
(19.4335, -99.1340, 1),
(19.4340, -99.1345, 1),
(19.4345, -99.1350, 1),
(19.4350, -99.1355, 1),
(19.4355, -99.1360, 1),
(19.4360, -99.1365, 1),
(19.4365, -99.1370, 1),
(19.4370, -99.1375, 1),
(19.4400, -99.1400, 2),
(19.4405, -99.1405, 2),
(19.4410, -99.1410, 2),
(19.4415, -99.1415, 2),
(19.4420, -99.1420, 2),
(19.4425, -99.1425, 2),
(19.4430, -99.1430, 2),
(19.4435, -99.1435, 2),
(19.4440, -99.1440, 2),
(19.4445, -99.1445, 2),
(19.4500, -99.1200, 3),
(19.4505, -99.1205, 3),
(19.4510, -99.1210, 3),
(19.4515, -99.1215, 3),
(19.4520, -99.1220, 3),
(19.4525, -99.1225, 3),
(19.4530, -99.1230, 3),
(19.4535, -99.1235, 3),
(19.4540, -99.1240, 3),
(19.4545, -99.1245, 3),
```



```
(19.4200, -99.1100, 4),  
(19.4205, -99.1105, 4),  
(19.4210, -99.1110, 4),  
(19.4215, -99.1115, 4),  
(19.4220, -99.1120, 4),  
(19.4225, -99.1125, 4),  
(19.4230, -99.1130, 4),  
(19.4235, -99.1135, 4),  
(19.4240, -99.1140, 4),  
(19.4245, -99.1145, 4),  
(19.4300, -99.1350, 5),  
(19.4305, -99.1355, 5),  
(19.4310, -99.1360, 5),  
(19.4315, -99.1365, 5),  
(19.4320, -99.1370, 5),  
(19.4325, -99.1375, 5),  
(19.4330, -99.1380, 5),  
(19.4335, -99.1385, 5),  
(19.4340, -99.1390, 5),  
(19.4345, -99.1395, 5),  
(19.4350, -99.1250, 6),  
(19.4355, -99.1255, 6),  
(19.4360, -99.1260, 6),  
(19.4365, -99.1265, 6),  
(19.4370, -99.1270, 6),  
(19.4375, -99.1275, 6),  
(19.4380, -99.1280, 6),  
(19.4385, -99.1285, 6),  
(19.4390, -99.1290, 6),  
(19.4395, -99.1295, 6);
```

```
INSERT INTO VIAJE.UBICACION (latitud, longitud, id_viaje) VALUES
```

```
(19.4300, -99.1300, 7),  
(19.4305, -99.1305, 7),  
(19.4310, -99.1310, 7),  
(19.4315, -99.1315, 7),  
(19.4320, -99.1320, 7),  
(19.4325, -99.1325, 7),  
(19.4330, -99.1330, 7),  
(19.4335, -99.1335, 7),  
(19.4340, -99.1340, 7),  
(19.4345, -99.1345, 7),  
(19.4250, -99.1350, 8),  
(19.4255, -99.1355, 8),  
(19.4260, -99.1360, 8),  
(19.4265, -99.1365, 8),  
(19.4270, -99.1370, 8),  
(19.4275, -99.1375, 8),  
(19.4280, -99.1380, 8),  
(19.4285, -99.1385, 8),  
(19.4290, -99.1390, 8),  
(19.4295, -99.1395, 8),  
(19.4400, -99.1200, 9),  
(19.4405, -99.1205, 9),
```

```
(19.4410, -99.1210, 9),  
(19.4415, -99.1215, 9),  
(19.4420, -99.1220, 9),  
(19.4425, -99.1225, 9),  
(19.4430, -99.1230, 9),  
(19.4435, -99.1235, 9),  
(19.4440, -99.1240, 9),  
(19.4445, -99.1245, 9),  
(19.4450, -99.1150, 10),  
(19.4455, -99.1155, 10),  
(19.4460, -99.1160, 10),  
(19.4465, -99.1165, 10),  
(19.4470, -99.1170, 10),  
(19.4475, -99.1175, 10),  
(19.4480, -99.1180, 10),  
(19.4485, -99.1185, 10),  
(19.4490, -99.1190, 10),  
(19.4495, -99.1195, 10);
```

```
INSERT INTO VIAJE.UBICACION (latitud, longitud, id_viaje) VALUES
```

```
(19.4305, -99.1315, 11),  
(19.4310, -99.1320, 11),  
(19.4315, -99.1325, 11),  
(19.4320, -99.1330, 11),  
(19.4325, -99.1335, 11),  
(19.4330, -99.1340, 11),  
(19.4335, -99.1345, 11),  
(19.4340, -99.1350, 11),  
(19.4345, -99.1355, 11),  
(19.4350, -99.1360, 11),  
(19.4251, -99.1351, 12),  
(19.4256, -99.1356, 12),  
(19.4261, -99.1361, 12),  
(19.4266, -99.1366, 12),  
(19.4271, -99.1371, 12),  
(19.4276, -99.1376, 12),  
(19.4281, -99.1381, 12),  
(19.4286, -99.1386, 12),  
(19.4291, -99.1391, 12),  
(19.4296, -99.1396, 12),  
(19.4402, -99.1202, 13),  
(19.4407, -99.1207, 13),  
(19.4412, -99.1212, 13),  
(19.4417, -99.1217, 13),  
(19.4422, -99.1222, 13),  
(19.4427, -99.1227, 13),  
(19.4432, -99.1232, 13),  
(19.4437, -99.1237, 13),  
(19.4442, -99.1242, 13),  
(19.4447, -99.1247, 13),  
(19.4453, -99.1153, 14),  
(19.4458, -99.1158, 14),  
(19.4463, -99.1163, 14),  
(19.4468, -99.1168, 14),  
(19.4473, -99.1173, 14),
```

```
(19.4478, -99.1178, 14),  
(19.4483, -99.1183, 14),  
(19.4488, -99.1188, 14),  
(19.4493, -99.1193, 14),  
(19.4498, -99.1198, 14),  
(19.4354, -99.1304, 15),  
(19.4359, -99.1309, 15),  
(19.4364, -99.1314, 15),  
(19.4369, -99.1319, 15),  
(19.4374, -99.1324, 15),  
(19.4379, -99.1329, 15),  
(19.4384, -99.1334, 15),  
(19.4389, -99.1339, 15),  
(19.4394, -99.1344, 15),  
(19.4399, -99.1349, 15);
```

```
INSERT INTO VIAJE.UBICACION (latitud, longitud, id_viaje) VALUES
```

```
(19.4300, -99.1300, 16),  
(19.4303, -99.1303, 16),  
(19.4306, -99.1306, 16),  
(19.4309, -99.1309, 16),  
(19.4312, -99.1312, 16),  
(19.4315, -99.1315, 16),  
(19.4318, -99.1318, 16),  
(19.4321, -99.1321, 16),  
(19.4324, -99.1324, 16),  
(19.4327, -99.1327, 16),  
(19.4250, -99.1350, 17),  
(19.4253, -99.1353, 17),  
(19.4256, -99.1356, 17),  
(19.4259, -99.1359, 17),  
(19.4262, -99.1362, 17),  
(19.4265, -99.1365, 17),  
(19.4268, -99.1368, 17),  
(19.4271, -99.1371, 17),  
(19.4274, -99.1374, 17),  
(19.4277, -99.1377, 17),  
(19.4400, -99.1200, 18),  
(19.4403, -99.1203, 18),  
(19.4406, -99.1206, 18),  
(19.4409, -99.1209, 18),  
(19.4412, -99.1212, 18),  
(19.4415, -99.1215, 18),  
(19.4418, -99.1218, 18),  
(19.4421, -99.1221, 18),  
(19.4424, -99.1224, 18),  
(19.4427, -99.1227, 18),  
(19.4450, -99.1150, 19),  
(19.4453, -99.1153, 19),  
(19.4456, -99.1156, 19),  
(19.4459, -99.1159, 19),  
(19.4462, -99.1162, 19),  
(19.4465, -99.1165, 19),  
(19.4468, -99.1168, 19),  
(19.4471, -99.1171, 19),
```

```
(19.4474, -99.1174, 19),  
(19.4477, -99.1177, 19);
```

```
INSERT INTO VIAJE.UBICACION (latitud, longitud, id_viaje) VALUES
```

```
(19.4200, -99.1400, 20),  
(19.4203, -99.1403, 20),  
(19.4206, -99.1406, 20),  
(19.4209, -99.1409, 20),  
(19.4212, -99.1412, 20),  
(19.4215, -99.1415, 20),  
(19.4218, -99.1418, 20),  
(19.4221, -99.1421, 20),  
(19.4224, -99.1424, 20),  
(19.4227, -99.1427, 20),
```

```
(19.4300, -99.1300, 21),  
(19.4304, -99.1304, 21),  
(19.4308, -99.1308, 21),  
(19.4312, -99.1312, 21),  
(19.4316, -99.1316, 21),  
(19.4320, -99.1320, 21),  
(19.4324, -99.1324, 21),  
(19.4328, -99.1328, 21),  
(19.4332, -99.1332, 21),  
(19.4336, -99.1336, 21),
```

```
(19.4400, -99.1200, 22),  
(19.4404, -99.1204, 22),  
(19.4408, -99.1208, 22),  
(19.4412, -99.1212, 22),  
(19.4416, -99.1216, 22),  
(19.4420, -99.1220, 22),  
(19.4424, -99.1224, 22),  
(19.4428, -99.1228, 22),  
(19.4432, -99.1232, 22),  
(19.4436, -99.1236, 22),
```

```
(19.4450, -99.1150, 23),  
(19.4454, -99.1154, 23),  
(19.4458, -99.1158, 23),  
(19.4462, -99.1162, 23),  
(19.4466, -99.1166, 23),  
(19.4470, -99.1170, 23),  
(19.4474, -99.1174, 23),  
(19.4478, -99.1178, 23),  
(19.4482, -99.1182, 23),  
(19.4486, -99.1186, 23),
```

```
(19.4500, -99.1100, 24),  
(19.4504, -99.1104, 24),  
(19.4508, -99.1108, 24),  
(19.4512, -99.1112, 24),  
(19.4516, -99.1116, 24),  
(19.4520, -99.1120, 24),
```

```
(19.4524, -99.1124, 24),
(19.4528, -99.1128, 24),
(19.4532, -99.1132, 24),
(19.4536, -99.1136, 24);

INSERT INTO VIAJE.UBICACION (latitud, longitud, id_viaje) VALUES
(19.4100, -99.1500, 25),
(19.4103, -99.1503, 25),
(19.4106, -99.1506, 25),
(19.4109, -99.1509, 25),
(19.4112, -99.1512, 25),
(19.4115, -99.1515, 25),
(19.4118, -99.1518, 25),
(19.4121, -99.1521, 25),
(19.4124, -99.1524, 25),
(19.4127, -99.1527, 25),

(19.4205, -99.1405, 26),
(19.4208, -99.1408, 26),
(19.4211, -99.1411, 26),
(19.4214, -99.1414, 26),
(19.4217, -99.1417, 26),
(19.4220, -99.1420, 26),
(19.4223, -99.1423, 26),
(19.4226, -99.1426, 26),
(19.4229, -99.1429, 26),
(19.4232, -99.1432, 26),

(19.4305, -99.1305, 27),
(19.4308, -99.1308, 27),
(19.4311, -99.1311, 27),
(19.4314, -99.1314, 27),
(19.4317, -99.1317, 27),
(19.4320, -99.1320, 27),
(19.4323, -99.1323, 27),
(19.4326, -99.1326, 27),
(19.4329, -99.1329, 27),
(19.4332, -99.1332, 27),

(19.4400, -99.1200, 28),
(19.4403, -99.1203, 28),
(19.4406, -99.1206, 28),
(19.4409, -99.1209, 28),
(19.4412, -99.1212, 28),
(19.4415, -99.1215, 28),
(19.4418, -99.1218, 28),
(19.4421, -99.1221, 28),
(19.4424, -99.1224, 28),
(19.4427, -99.1227, 28);

INSERT INTO VIAJE.UBICACION (latitud, longitud, id_viaje) VALUES
(19.4000, -99.1800, 29),
(19.4005, -99.1805, 29),
(19.4010, -99.1810, 29),
(19.4015, -99.1815, 29),
```

```
(19.4020, -99.1820, 29),  
(19.4025, -99.1825, 29),  
(19.4030, -99.1830, 29),  
(19.4035, -99.1835, 29),  
(19.4040, -99.1840, 29),  
(19.4045, -99.1845, 29),
```

```
(19.4100, -99.1700, 30),  
(19.4107, -99.1707, 30),  
(19.4114, -99.1714, 30),  
(19.4121, -99.1721, 30),  
(19.4128, -99.1728, 30),  
(19.4135, -99.1735, 30),  
(19.4142, -99.1742, 30),  
(19.4149, -99.1749, 30),  
(19.4156, -99.1756, 30),  
(19.4163, -99.1763, 30),
```

```
(19.4200, -99.1600, 31),  
(19.4210, -99.1610, 31),  
(19.4220, -99.1620, 31),  
(19.4230, -99.1630, 31),  
(19.4240, -99.1640, 31),  
(19.4250, -99.1650, 31),  
(19.4260, -99.1660, 31),  
(19.4270, -99.1670, 31),  
(19.4280, -99.1680, 31),  
(19.4290, -99.1690, 31),
```

```
(19.4300, -99.1500, 32),  
(19.4310, -99.1510, 32),  
(19.4320, -99.1520, 32),  
(19.4330, -99.1530, 32),  
(19.4340, -99.1540, 32),  
(19.4350, -99.1550, 32),  
(19.4360, -99.1560, 32),  
(19.4370, -99.1570, 32),  
(19.4380, -99.1580, 32),  
(19.4390, -99.1590, 32);
```

```
INSERT INTO VIAJE.UBICACION (latitud, longitud, id_viaje) VALUES
```

```
(19.4100, -99.1700, 50),  
(19.4110, -99.1711, 50),  
(19.4120, -99.1722, 50),  
(19.4130, -99.1733, 50),  
(19.4140, -99.1744, 50),  
(19.4150, -99.1755, 50),  
(19.4160, -99.1766, 50),  
(19.4170, -99.1770, 50),  
(19.4175, -99.1772, 50),  
(19.4180, -99.1775, 50),
```

```
(19.4300, -99.1500, 51),  
(19.4310, -99.1511, 51),  
(19.4320, -99.1522, 51),
```

```
(19.4330, -99.1533, 51),
(19.4340, -99.1544, 51),
(19.4350, -99.1555, 51),
(19.4360, -99.1566, 51),
(19.4370, -99.1577, 51),
(19.4380, -99.1580, 51),
(19.4385, -99.1585, 51),

(19.4400, -99.1400, 52),
(19.4410, -99.1410, 52),
(19.4420, -99.1420, 52),
(19.4430, -99.1430, 52),
(19.4440, -99.1440, 52),
(19.4450, -99.1450, 52),
(19.4460, -99.1460, 52),
(19.4470, -99.1470, 52),
(19.4480, -99.1480, 52),
(19.4490, -99.1490, 52);

SELECT * FROM VIAJE.UBICACION

----- POBLANDO LA TABLA TIPO ACCIDENTE
-----

SELECT * FROM VIAJE.TIPO_ACCIDENTE

INSERT INTO VIAJE.TIPO_ACCIDENTE (nombre) VALUES
('Choque frontal'),
('Choque trasero'),
('Volcadura'),
('Atropellamiento'),
('Deslizamiento'),
('Colisión lateral'),
('Caída de objetos'),
('Accidente por incendio'),
('Accidente mecánico');

SELECT * FROM VIAJE.TIPO_ACCIDENTE
ORDER BY ID_TIPO

----- POBLANDO LA TABLA ACCIDENTE
-----

SELECT * FROM VIAJE.ACCIDENTE

INSERT INTO VIAJE.ACCIDENTE (ubicacion, monto, herido, fecha, hora,
descripcion, id_viaje, id_tipo, culpable) VALUES
('Carretera Federal 45', 15000.00, 'SI', '2025-05-01', '10:20',
'Choque frontal entre dos vehículos en zona de curvas.', 1, 1,
'SI'),
('Autopista Norte', 8000.00, 'NO', '2025-05-03', '14:35', 'Choque
trasero leve por distracción del conductor.', 3, 2, 'NO'),
('Km 12, Ruta 5', 20000.00, 'SI', '2025-05-05', '09:30', 'Volcadura
de camión de carga debido a exceso de velocidad.', 5, 3, 'SI'),
```

```
('Zona urbana Centro', 5000.00, 'NO', '2025-05-10', '09:18',  
'Atropellamiento leve de peatón en cruce peatonal.', 10, 4, 'NO'),  
('Camino de montaña', 12000.00, 'SI', '2025-05-22', '11:50',  
'Deslizamiento de tierra bloquea la carretera.', 15, 5, 'SI'),  
('Intersección Avenida Principal', 7000.00, 'NO', '2025-05-24',  
'10:40', 'Colisión lateral entre dos autos en semáforo.', 17, 6,  
'NO'),  
('Zona industrial', 11000.00, 'SI', '2025-05-25', '09:35', 'Caída de  
objetos de un camión en movimiento.', 20, 7, 'SI'),  
('Parque industrial', 9000.00, 'NO', '2025-05-25', '10:40',  
'Accidente por incendio en motor de vehículo.', 21, 8, 'NO'),  
('Carretera estatal', 16000.00, 'SI', '2025-05-26', '10:45',  
'Accidente mecánico con pérdida de control del vehículo.', 23, 9,  
'SI');
```

ValidaTriggers.sql

```
/*  
  
AUTORES:  
    AYALA HERNÁNDEZ MARÍA FERNANDA  
    PORTILLA HERMENEGILDO ELIZABETH  
    VALENZUELA ASCENCIO GUSTAVO  
    JIMENEZ AYALA YORDI JOSUE  
  
FECHA DE CREACION: 24 DE MAYO DE 2025  
  
ULTIMA ACTUALIZACION: 25 DE MAYO DE 2025  
  
DESCRIPCION: DML QUE VALIDA LOS TRIGGERS Y LOS CONSTRAINTS  
  
*/  
  
-- SELECCIONAMOS NUESTRA BASE PARA TRABAJAR SOBRE ELLA  
  
USE Equipo09AppSafe  
GO  
  
-----  
TRIGGERS  
-----  
  
-- VALIDACION (Trigger que valida en la tabla VIAJE que si un viaje  
es programado el inicio no pase de 2 días)  
  
BEGIN TRAN  
  
INSERT INTO VIAJE.VIAJE (  
    fecha_solicitud, hora_solicitud, latitud_origen,  
    latitud_destino,
```



```
        longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
propina,
        tipo, programado, comentario, importe, id_auto, id_etiqueta,
id_factura,
        id_estatus, id_usuario, id_tarjeta
    )
VALUES
('2025-05-27', '19:00', 19.4326, 19.3900, -99.1332, -99.1500,
'2025-05-29', '20:15', 15,
'E', 'SI', 'Viaje Programado', 100.50, 12, 1, NULL, 2, 2, NULL)

ROLLBACK TRAN

-- VALIDACION (Trigger que valida que si no es programado la fecha y
hora inicio sea la misma que la de solicitud)

BEGIN TRAN

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
id_factura,
    id_estatus, id_usuario, id_tarjeta
)
VALUES
('2025-05-27', '19:00', 19.4326, 19.3900, -99.1332, -99.1500,
'2025-05-29', '20:15', 15,
'E', 'NO', 'Viaje por iniciar', 100.50, 12, 1, NULL, 2, 2, NULL)

ROLLBACK TRAN

-- VALIDACION (Trigger que valida que no se ponga un id de tarjeta
si no se va a pagar con tarjeta (tipo = 'E'))

BEGIN TRAN

INSERT INTO VIAJE.VIAJE (
    fecha_solicitud, hora_solicitud, latitud_origen,
latitud_destino,
    longitud_origen, longitud_destino, fecha_inicio, hora_inicio,
propina,
    tipo, programado, comentario, importe, id_auto, id_etiqueta,
id_factura,
    id_estatus, id_usuario, id_tarjeta
)
VALUES
('2025-05-27', '19:00', 19.4326, 19.3900, -99.1332, -99.1500,
'2025-05-27', '20:15', 15,
'E', 'SI', 'Viaje Programado', 100.50, 12, 1, NULL, 2, 2, 1)

ROLLBACK TRAN
```

```
-- VALIDACION (Tigger que valida que un cliente tenga maximo 3
tarjetas)

SELECT * FROM CLIENTE.TARJETA -- El cliente 8 ya tiene las 3
tarjetas

BEGIN TRAN

INSERT INTO CLIENTE.TARJETA (numero_tarjeta, vigencia, id_usuario,
id_banco) VALUES
('4532123412345690', '2027-12-31', 8 ,1)

ROLLBACK TRAN

-- VALIDACION (Trigger que valida que un conductor tenga maximo 2
autos)

SELECT * FROM CONDUCTOR.[AUTO] -- El conductor 11 ya tiene los 2
autos

BEGIN TRAN

INSERT INTO CONDUCTOR.[AUTO] (año, placa, color, id_modelo,
id_usuario) VALUES
(2023, 'ABC12X4Z', 'Verde', 5, 11)

ROLLBACK TRAN

-- VALIDACION (Trigger que valida que los conductores se inserten en
conductores y no en cliente)

SELECT * FROM USUARIO.USUARIO

INSERT INTO USUARIO.USUARIO VALUES
('yordibblue', 'yordi.josueja@gmail.com', 'C14v3SegUr4#A1B2C3D8',
'2025-05-25', 'Yordi', 'Jimenez', 'Ayala', 1, 0, NULL) -- Insert de
prueba, id 36 (VERIFICAR EL ID POR EL IDENTITY)

BEGIN TRAN

INSERT INTO USUARIO.CONDUCTOR (id_usuario, num_licencia,
vigencia_lic, foto, descripcion)
SELECT
    36,
    10000036,
    '2028-10-31',
    BulkColumn,
    'Soy Yordi, no soy conductor pero quiero violar el sistema y
registrarme como conductor'
FROM
OPENROWSET(
    BULK '1.jpg',
    DATA_SOURCE = 'ConductoresSrc',
    SINGLE_BLOB
```

```
) AS img;

ROLLBACK TRAN

DELETE FROM USUARIO.USUARIO -- Borramos el insert de prueba
WHERE id_usuario = 36; -- CAMBIAR POR EL ID GENERADO

-- VALIDACION (Trigger que valida que los clientes se inserten en
clientes y no en conductores)

SELECT * FROM USUARIO.USUARIO

INSERT INTO USUARIO.USUARIO VALUES
('yordibblue', 'yordi.josueja@gmail.com', 'Cl4v3SegUr4#A1B2C3D8',
'2025-05-25', 'Yordi', 'Jimenez', 'Ayala', 0, 1, NULL) -- Insert de
prueba, id 37 (VERIFICAR EL QUE GENERA POR EL IDENTITY)

BEGIN TRAN

INSERT INTO USUARIO.CLIENTE (id_usuario, rfc, direccion, numero,
fecha_registro, calificacion) VALUES
(37, 'JIAY040101AB3', 'Copal 155C, Pedregal de Santo DOMingo, CDMX',
5611655306, '2025-05-25', 5)

ROLLBACK TRAN

DELETE FROM USUARIO.USUARIO -- Borramos el insert de prueba
WHERE id_usuario = 37; -- CAMBIAR POR EL ID GENERADO

-- VALIDACION (Trigger que valida que si un valor bit es 0 o 1 en
usuario el otro debe ser el contrario)

BEGIN TRAN

INSERT INTO USUARIO.USUARIO VALUES
('yordibblue', 'yordi.josueja@gmail.com', 'Cl4v3SegUr4#A1B2C3D8',
'2025-05-25', 'Yordi', 'Jimenez', 'Ayala', 0, 0, NULL)

ROLLBACK TRAN

BEGIN TRAN

INSERT INTO USUARIO.USUARIO VALUES
('yordibblue', 'yordi.josueja@gmail.com', 'Cl4v3SegUr4#A1B2C3D8',
'2025-05-25', 'Yordi', 'Jimenez', 'Ayala', 1, 1, NULL)

ROLLBACK TRAN

-- VALIDACION (Trigger que valida que si se pone null en
calificacion del cliente no se inserte el dato)

SELECT * FROM USUARIO.USUARIO

INSERT INTO USUARIO.USUARIO VALUES
('yordibblue', 'yordi.josueja@gmail.com', 'Cl4v3SegUr4#A1B2C3D8',
```

```
'2025-05-25', 'Yordi', 'Jimenez', 'Ayala', 1, 0, NULL) -- Insert de prueba, id 54 (VERIFICAR EL QUE GENERA POR EL IDENTITY)
```

```
BEGIN TRAN
```

```
INSERT INTO USUARIO.CLIENTE (id_usuario, rfc, direccion, numero, fecha_registro, calificacion) VALUES (54, 'JIAY040101AB3', 'Copal 155C, Pedregal de Santo DOMingo, CDMX', 5611655306, '2025-05-25', NULL)
```

```
ROLLBACK TRAN
```

```
DELETE FROM USUARIO.USUARIO -- Borramos el insert de prueba WHERE id_usuario = 54; -- CAMBIAR POR EL ID GENERADO
```

```
-- VALIDACION (Trigger que valida que la fecha de atencion no sea mayor a 5 días)
```

```
BEGIN TRAN
```

```
INSERT INTO CLIENTE.QUEJA (descripcion, titulo, fecha_atencion, fecha_queja, id_usuario, id_administrador, id_conductor, id_auto) VALUES ('Queja de prueba', 'Queja', '2025-05-27', '2025-05-21', 2, 1, 5, 4)
```

```
ROLLBACK TRAN
```

```
-- VALIDACION (Trigger que no inserta si se pone null en la fecha queja)
```

```
BEGIN TRAN
```

```
INSERT INTO CLIENTE.QUEJA (descripcion, titulo, fecha_atencion, fecha_queja, id_usuario, id_administrador) VALUES ('Queja de prueba', 'Queja', '2025-05-27', NULL, 2, 1)
```

```
ROLLBACK TRAN
```

```
-- VALIDACION (Trigger que valida que la tarjeta pertenezca al usuario en el viaje)
```

```
BEGIN TRAN
```

```
INSERT INTO VIAJE.VIAJE ( fecha_solicitud, hora_solicitud, latitud_origen, latitud_destino, longitud_origen, longitud_destino, fecha_inicio, hora_inicio, propina, tipo, programado, comentario, importe, id_auto, id_etiqueta, id_factura, id_estatus, id_usuario, id_tarjeta )
```

```
VALUES
```

```
('2025-05-27', '19:00', 19.4326, 19.3900, -99.1332, -99.1500, '2025-05-28', '20:15', 15,
```

```
'T', 'SI', 'Viaje Programado', 100.50, 12, 1, NULL, 2, 2, 15)

ROLLBACK TRAN

-- VALIDACION (Trigger que valida que el auto y el conductor en la
queja coincidan)

BEGIN TRAN

INSERT INTO CLIENTE.QUEJA (descripcion, titulo, fecha_atencion,
fecha_queja, id_usuario, id_administrador, id_conductor, id_auto)
VALUES
('Queja de prueba', 'Queja', '2025-05-22', '2025-05-21', 2, 1, 5,
10)

ROLLBACK TRAN

-----
CONSTRAINTS
-----

-- USUARIO
SELECT * FROM USUARIO.USUARIO
-- 1. Validar clave candidata como Unique (nombre_usuario)

BEGIN TRAN
-- Insertamos nombre_usuario 'psan_21', no se inserta ya que
este nombre de usuario ya existe
INSERT INTO USUARIO.USUARIO (nombre_usuario, correo,
clave, fecha_ingreso, nombre, ap_paterno, ap_materno, cliente,
conductor, id_recomendado)
VALUES ('psan_21', 'juan@example.com',
'P4sswordSeguro*ABC1', '2021/12/18', 'Juan', 'Pérez', 'López', 1, 0,
NULL);
ROLLBACK TRAN

-- 2. Validar que clave tenga una longitud de 20 caracteres,
que incluya por lo menos una mayúscula, un número y un carácter (#,*
o $) en cualquier posición

-- No cumple con longitud
BEGIN TRAN
INSERT INTO USUARIO.USUARIO (nombre_usuario, correo,
clave, fecha_ingreso, nombre, ap_paterno, ap_materno, cliente,
conductor, id_recomendado)
VALUES ('juan123', 'juan@example.com',
'ClaveSegura2024#', '2023/12/18', 'Juan', 'Pérez', 'López', 1, 0,
NULL);
ROLLBACK TRAN

-- No contiene un caracter
BEGIN TRAN
INSERT INTO USUARIO.USUARIO (nombre_usuario, correo,
clave, fecha_ingreso, nombre, ap_paterno, ap_materno, cliente,
conductor, id_recomendado)
```

```
VALUES ('juan123', 'juan@example.com',
'P4sswordelpguroABAC1', '2021/12/18', 'Juan','Pérez', 'López', 1, 0,
NULL);
ROLLBACK TRAN

-- No contiene un numero
BEGIN TRAN
INSERT INTO USUARIO.USUARIO (nombre_usuario, correo,
clave, fecha_ingreso, nombre, ap_paterno, ap_materno, cliente,
conductor, id_recomendado)
VALUES ('juan123', 'juan@example.com',
'Pesswor$elpguroABACu', '2021/12/18', 'Juan','Pérez', 'López', 1, 0,
NULL);
ROLLBACK TRAN

-- 3. Validar que id_recomendado sea una fk
BEGIN TRAN
INSERT INTO USUARIO.USUARIO (nombre_usuario, correo,
clave, fecha_ingreso, nombre, ap_paterno, ap_materno, cliente,
conductor, id_recomendado)
VALUES ('pancho123', 'pancho@example.com',
'ClaveSegura2024#$$10', '2021/12/18', 'Pancho','Pineda', 'Portilla',
0, 1, 40);
ROLLBACK TRAN

-- 4. Validar que correo tenga como default 'SIN CORREO'
BEGIN TRAN
INSERT INTO USUARIO.USUARIO (nombre_usuario, correo,
clave, fecha_ingreso, nombre, ap_paterno, ap_materno, cliente,
conductor, id_recomendado)
VALUES ('el123', default,
'P4sswordelpguro*ABC1', '2021/12/18', 'Elsa','Pérez', 'López', 1, 0,
NULL);
SELECT * FROM USUARIO.USUARIO
ROLLBACK TRAN

--CONDUCTOR
SELECT * FROM USUARIO.CONDUCTOR
--1. Validar que id_usuario si sea una pk
BEGIN TRAN
INSERT INTO USUARIO.CONDUCTOR (id_usuario,
num_licencia, vigencia_lic, foto, descripcion)
SELECT 1,10000001,'2028-12-31',BulkColumn,'Soy
Carlos, tengo 35 años y cuento con más de 10 años de experiencia al
volante en rutas urbanas y de larga distancia. Me caracterizo por un
manejo defensivo, puntualidad y excelente trato al cliente. Poseo
licencias vigentes y un historial impecable de seguridad vial.
Siempre estoy comprometido con la eficiencia, el cuidado de la carga
y el cumplimiento de horarios.'
FROM OPENROWSET(BULK '3.jpg',DATA_SOURCE =
'ConductoresSrc', SINGLE_BLOB) AS img;
SELECT * FROM USUARIO.CONDUCTOR
```

```
ROLLBACK TRAN

--2. Validar que id_usuario sea una fk (fkUsuarioConductor),
BEGIN TRAN
    INSERT INTO USUARIO.CONDUCTOR (id_usuario,
num_licencia, vigencia_lic, foto, descripcion)
    SELECT
261,10000010,'2028-12-31',BulkColumn,'Soy Carlos, tengo 35 años y
cuento con más de 10 años de experiencia al volante en rutas urbanas
y de larga distancia. Me caracterizo por un manejo defensivo,
puntualidad y excelente trato al cliente. Poseo licencias vigentes y
un historial impecable de seguridad vial. Siempre estoy comprometido
con la eficiencia, el cuidado de la carga y el cumplimiento de
horarios.'
    FROM OPENROWSET(BULK '3.jpg',DATA_SOURCE =
'ConductoresSrc', SINGLE_BLOB) AS img;
    SELECT * FROM USUARIO.CONDUCTOR
ROLLBACK TRAN

--3. Validar que clave candidata num_licencia sea UNIQUE
BEGIN TRAN
    UPDATE USUARIO.CONDUCTOR
    SET num_licencia=10000001
    WHERE id_usuario=3;
    SELECT * FROM USUARIO.CONDUCTOR
ROLLBACK TRAN

--4. Validar que descripción tenga una longitud de max 600
caracteres
BEGIN TRAN
    UPDATE USUARIO.CONDUCTOR
    SET descripcion='Vivo en un surrealismo puro de
algun comic total,
Sin argumento y sin final

Donde a la orilla de la luna he encontrado mi hogar,
Y ya no puedo regresar.
A los 15 la vida es un sueño fugaz,
Siempre en una corniza sin poder parar.

No soy Alicia y nunca fui Peter Pan,
No estoy perdida y no lo paso tan mal.
MI corazon no es Wonderland.
No soy Alicia y nunca fui Peter Pan,
Este es mi mundo y no lo quiero cambiar.
MI corazon no es Wonderland.

En este juego tan intenso el es amor fugaz,
Y un beso siempre es opcional.
Por eso voy sin equipaje haci el nunca jamas,
Sin importarme el que diran.
A los 15 la vida es un sueño fugaz,
Siempre en una corniza sin poder parar.

Aunque de vez en cuando caigo,
```

```
Siempre me puedo reinventar.

MI Corazon no es Wonderland.

MI Corazon no es Wonderland.

No soy Alicia y nunca fui Peter Pan,
No estoy perdida y no paso tan mal.
MI Corazon no es Wonderland.'
    WHERE id_usuario=3;
    SELECT * FROM USUARIO.CONDUCTOR
ROLLBACK TRAN

--CLIENTE
    SELECT * FROM USUARIO.CLIENTE
--1. Validar que id_usuario sea una pk
    BEGIN TRAN
        INSERT INTO USUARIO.CLIENTE (id_usuario, direccion,
numero, fecha_registro, calificacion, rfc)
        VALUES (2,'Xola 714,Del Valle,
CDMX',5576983411, '2024-08-08', 2,'MLOA111519BD2')
    SELECT * FROM USUARIO.CLIENTE
    ROLLBACK TRAN

--2. Validar que id_usuario sea una fk (fkUsuarioCliente),
    BEGIN TRAN
        INSERT INTO USUARIO.CLIENTE (id_usuario, direccion,
numero, fecha_registro, calificacion, rfc)
        VALUES (37,'Xola 714,Del Valle,
CDMX',5576983411, '2024-08-08', 2,'MLOA111519BD2')
    SELECT * FROM USUARIO.CLIENTE
    ROLLBACK TRAN

--3. Validar que calificación sea un valor en (1, 2, 3, 4, 5)
    BEGIN TRAN
        UPDATE USUARIO.CLIENTE
        SET calificacion=8
        WHERE id_usuario=35;
    SELECT * FROM USUARIO.CLIENTE
    ROLLBACK TRAN

--4. Validar que la longitud de rfc sea de 13 caracteres
    BEGIN TRAN
        UPDATE USUARIO.CLIENTE
        SET rfc= 'MLOA111519BD222'
        WHERE id_usuario=35;
    SELECT * FROM USUARIO.CLIENTE
    ROLLBACK TRAN

-- 5. Validar ak (indice) del rfc
    BEGIN TRAN
        UPDATE USUARIO.CLIENTE
```



```
SET rfc = 'GARC990708GH9'
WHERE rfc = 'JUAC850101AB3';
SELECT * FROM USUARIO.CLIENTE
ROLLBACK TRAN

--6. Validar que calificacion tenga como default (5)
(VVALIDADO EN INSERT)

--PAGO
SELECT * FROM CONDUCTOR.PAGO
--1. Validar que id_usuario y folio sean la pk compuesta, y
que vayan en orden ascendente ambas
BEGIN TRAN
INSERT INTO CONDUCTOR.PAGO (id_usuario, folio,
fecha, monto)
VALUES (1,16273859,'2024-01-22',1590.00)
SELECT * FROM CONDUCTOR.PAGO
ROLLBACK TRAN

--2. Validar que id_usuario es una fk (fkIdUsuarioPago2),
agregando ON DELETE CASCADE
BEGIN TRAN
UPDATE CONDUCTOR.PAGO
SET id_usuario=261
WHERE folio=16273859;
SELECT * FROM CONDUCTOR.PAGO
ROLLBACK TRAN

--3. Validar que folio inicie con 1
BEGIN TRAN
UPDATE CONDUCTOR.PAGO
SET folio=96273859
WHERE id_usuario=1;
SELECT * FROM CONDUCTOR.PAGO
ROLLBACK TRAN

--TIPO
SELECT * FROM DESCUENTO.TIPO
--1.Validar que al insertar un nombre que ya está marque el
error
BEGIN TRAN
UPDATE DESCUENTO.TIPO
SET nombre='Descuento por cancelación anterior'
WHERE id_tipo=13;
SELECT * FROM DESCUENTO.TIPO
ROLLBACK TRAN

--DESCUENTO
SELECT * FROM DESCUENTO.DESCUENTO
```

```
--1. Validar que id_usuario sea fk (fkIdUsuario)
BEGIN TRAN
UPDATE DESCUENTO.DESCUENTO
SET id_usuario = 54
WHERE id_descuento = 61;
SELECT * FROM DESCUENTO.DESCUENTO
ROLLBACK TRAN

--2. Validar que id_tipo sea fk (fkIdTipo)
BEGIN TRAN
UPDATE DESCUENTO.DESCUENTO
SET id_tipo = 78
WHERE id_descuento = 61;
SELECT * FROM DESCUENTO.DESCUENTO
ROLLBACK TRAN

--MARCA
SELECT * FROM CONDUCTOR.MARCA
--1. Validar que en el catalogo no se repita nombre
BEGIN TRAN
UPDATE CONDUCTOR.MARCA
SET nombre = 'Acura'
WHERE id_marca = 27;
SELECT * FROM CONDUCTOR.MARCA
ROLLBACK TRAN

--MODELO
SELECT * FROM CONDUCTOR.MODELO
--1. Validar que id_marca sea fk (fkMarcaModelo)
BEGIN TRAN
UPDATE CONDUCTOR.MODELO
SET id_marca= 261
WHERE id_modelo = 1;
SELECT * FROM CONDUCTOR.MODELO
ROLLBACK TRAN

--2. Validar que clave candidata nombre sea UNIQUE
BEGIN TRAN
UPDATE CONDUCTOR.MODELO
SET nombre = 'Camry'
WHERE id_modelo = 1;
SELECT * FROM CONDUCTOR.MODELO
ROLLBACK TRAN

--AUTO
SELECT * FROM CONDUCTOR.[AUTO]
--1. Validar que la diferencia entre el año actual y el valor
de la columna año sea de 5 años o menos
BEGIN TRAN
UPDATE CONDUCTOR.[AUTO]
SET año = 2019
WHERE id_auto = 20;
SELECT * FROM CONDUCTOR.[AUTO]
ROLLBACK TRAN
```

```
--2. Validar que id_modelo es fk (fkIdModelo)
BEGIN TRAN
    UPDATE CONDUCTOR.[AUTO]
    SET id_modelo = 261
    WHERE id_auto = 2;
    SELECT * FROM CONDUCTOR.[AUTO]
ROLLBACK TRAN

--3. Validar que id_usuario es fk (fkIdUsuarioAuto)
BEGIN TRAN
    UPDATE CONDUCTOR.[AUTO]
    SET id_usuario = 261
    WHERE id_auto = 2;
    SELECT * FROM CONDUCTOR.[AUTO]
ROLLBACK TRAN

--ADMINISTRADOR
SELECT * FROM USUARIO.ADMINISTRADOR

--1. Validar que id_recomendado es fk
(fkAdministradorRecomendado)
BEGIN TRAN
    UPDATE USUARIO.ADMINISTRADOR
    SET id_recomendado = 261
    WHERE id_administrador = 2;
    SELECT * FROM USUARIO.ADMINISTRADOR
ROLLBACK TRAN

--2. Validar que clave candidata nombre_usuario es UNIQUE
BEGIN TRAN
    UPDATE USUARIO.ADMINISTRADOR
    SET nombre_usuario = 'admin01'
    WHERE id_administrador = 2;
    SELECT * FROM USUARIO.ADMINISTRADOR
ROLLBACK TRAN

--3. Validar que clave tenga longitud de 20 caracteres, y
contenga al menos una letra mayúscula, un número, y un carácter (#,
* o $)

BEGIN TRAN
    UPDATE USUARIO.ADMINISTRADOR
    SET clave = 'Clave#A1B2C3D4E5F6G77'
    WHERE id_administrador = 2;
    SELECT * FROM USUARIO.ADMINISTRADOR
ROLLBACK TRAN

BEGIN TRAN
    UPDATE USUARIO.ADMINISTRADOR
    SET clave = 'ClaveAA1B2C3D4E5F6G7'
    WHERE id_administrador = 2;
    SELECT * FROM USUARIO.ADMINISTRADOR
ROLLBACK TRAN

--QUEJA
SELECT * FROM CLIENTE.QUEJA
```

```
--1. Validar que id_usuario es fk (fkIdUsuarioQueja),
BEGIN TRAN
    UPDATE CLIENTE.QUEJA
    SET id_usuario = 261
    WHERE id_queja = 2;
    SELECT * FROM CLIENTE.QUEJA
ROLLBACK TRAN

--2. Validar que id_administrador es fk
(fkIdAdministradorQueja),
BEGIN TRAN
    UPDATE CLIENTE.QUEJA
    SET id_administrador = 261
    WHERE id_queja = 2;
    SELECT * FROM CLIENTE.QUEJA
ROLLBACK TRAN

--3. Validar que fecha_queja tenga como default la fecha
actual en el servidor (VALIDADO EN INSERT)

--BANCO
SELECT * FROM CLIENTE.BANCO
--1. Validar que en el catalogo no se repita nombre
BEGIN TRAN
    UPDATE CLIENTE.BANCO
    SET nombre = 'Inbursa'
    WHERE id_banco = 2;
    SELECT * FROM CLIENTE.BANCO
ROLLBACK TRAN

--TARJETA
SELECT * FROM CLIENTE.TARJETA
--1. Validar que numero_Tarjeta tiene longitud de 16
caracteres
BEGIN TRAN
    UPDATE CLIENTE.TARJETA
    SET numero_Tarjeta = 453212341234567811
    WHERE id_tarjeta = 1;
    SELECT * FROM CLIENTE.TARJETA
ROLLBACK TRAN

--2. Validar que id_banco es fk (fkBancoTarjeta),
BEGIN TRAN
    UPDATE CLIENTE.TARJETA
    SET id_banco = 261
    WHERE id_tarjeta = 1;
    SELECT * FROM CLIENTE.TARJETA
ROLLBACK TRAN

--3. Validar que id_usuario es fk (fkClienteTarjeta),
BEGIN TRAN
    UPDATE CLIENTE.TARJETA
    SET id_usuario = 261
    WHERE id_tarjeta = 1;
    SELECT * FROM CLIENTE.TARJETA
```

```
ROLLBACK TRAN

-- 4. Validar ak del número de tarjeta
BEGIN TRAN
    UPDATE CLIENTE.TARJETA
    SET numero_tarjeta = '5244334455667788'
    WHERE numero_tarjeta = '4532015112830366';
    SELECT * FROM CLIENTE.TARJETA
ROLLBACK TRAN

--DESCUENTO_ADMIN
SELECT * FROM DESCUENTO.DESCUENTO_ADMIN
--1. Validar que id_tipo es fk (fkTipoDescuentoAdmin),
BEGIN TRAN
    UPDATE DESCUENTO.DESCUENTO_ADMIN
    SET id_tipo = 261
    WHERE id_descuento = 1;
    SELECT * FROM DESCUENTO.DESCUENTO_ADMIN
ROLLBACK TRAN

--2. Validar que id_administrador es fk
(fkAdministradorDescuentoAdmin),
BEGIN TRAN
    UPDATE DESCUENTO.DESCUENTO_ADMIN
    SET id_administrador = 261
    WHERE id_descuento = 1;
    SELECT * FROM DESCUENTO.DESCUENTO_ADMIN
ROLLBACK TRAN

--TELEFONO
SELECT * FROM DBO.TELEFONO
--1. Validar que id_administrador y numero son pk compuesta, y
que van en orden ascendente
BEGIN TRAN
    INSERT INTO DBO.TELEFONO (id_administrador, numero)
    VALUES (1,5512345678)
    SELECT * FROM DBO.TELEFONO
ROLLBACK TRAN

--2. Validar que id_administrador es fk
(fkAdministradorTelefono),
BEGIN TRAN
    UPDATE DBO.TELEFONO
    SET id_administrador = 261
    WHERE numero = 5512345678;
    SELECT * FROM DBO.TELEFONO
ROLLBACK TRAN

--3. Validar que numero tiene longitud de 10
BEGIN TRAN
    UPDATE DBO.TELEFONO
    SET numero= 551234567812
    WHERE id_administrador = 1;
    SELECT * FROM DBO.TELEFONO
ROLLBACK TRAN
```

```
--ETIQUETA
SELECT * FROM VIAJE.ETIQUETA
--1. Validar que en el catalogo no se repita nombre
BEGIN TRAN
    UPDATE VIAJE.ETIQUETA
    SET nombre= 'Conducción segura'
    WHERE id_etiqueta = 4;
    SELECT * FROM VIAJE.ETIQUETA
ROLLBACK TRAN

--FACTURA
-- No hay constraint que validar

--ESTATUS
SELECT * FROM VIAJE.ESTATUS
--1. Validar que solo se pueda insertar en nombre:
'Solicitado', 'Programado', 'Confirmado', 'En curso', 'Terminado',
'Por cobrar', 'Pagado', 'Con adeudo' o 'Cancelado'
BEGIN TRAN
    UPDATE VIAJE.ESTATUS
    SET nombre= 'Palabra'
    WHERE id_estatus = 5;
    SELECT * FROM VIAJE.ESTATUS
ROLLBACK TRAN

--2. Validar que no se repitan los nombres con el UNIQUE
BEGIN TRAN
    UPDATE VIAJE.ESTATUS
    SET nombre= 'Confirmado'
    WHERE id_estatus = 5;
    SELECT * FROM VIAJE.ESTATUS
ROLLBACK TRAN

--VIAJE ESTATUS
SELECT * FROM VIAJE.VIAJE_ESTATUS
--1. Validar que id_estatus y id_viaje sean pk compuesta y que
vayan en orden ascendente
BEGIN TRAN
    INSERT INTO VIAJE.VIAJE_ESTATUS
(id_estatus,id_viaje, fecha_estatus, hora_estatus)
VALUES (1,1,'2025-05-01','09:45:00.0000000')
    SELECT * FROM VIAJE.VIAJE_ESTATUS
ROLLBACK TRAN

--VIAJE
SELECT * FROM VIAJE.VIAJE
--1. Validar que id_auto sea fk (fkAutoViaje), agregando ON
DELETE CASCADE y ON UPDATE CASCADE
BEGIN TRAN
    UPDATE VIAJE.VIAJE
    SET id_auto = 261
```

```
WHERE id_viaje = 1;
SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN

--2. Validar que id_etiqueta sea fk (fkEtiquetaViaje),
agregando ON DELETE CASCADE y ON UPDATE CASCADE
BEGIN TRAN
UPDATE VIAJE.VIAJE
SET id_etiqueta = 261
WHERE id_viaje = 1;
SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN

--3. Validar que id_factura sea fk (fkFacturaViaje),
agregando ON DELETE CASCADE y ON UPDATE CASCADE
BEGIN TRAN
UPDATE VIAJE.VIAJE
SET id_factura = 261
WHERE id_viaje = 1;
SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN

--4. Validar que id_estatus sea fk (fkEstatusViaje),
agregando ON DELETE CASCADE y ON UPDATE CASCADE
BEGIN TRAN
UPDATE VIAJE.VIAJE
SET id_estatus = 222
WHERE id_viaje = 1;
SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN

--5. Validar que id_usuario sea fk
BEGIN TRAN
UPDATE VIAJE.VIAJE
SET id_usuario = 261
WHERE id_viaje = 1;
SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN

--6. Validar que id_tarjeta sea fk
BEGIN TRAN
UPDATE VIAJE.VIAJE
SET id_tarjeta = 261
WHERE id_viaje = 1;
SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN

--7. Validar que calificacion sea un valor de 1, 2, 3, 4 o 5
BEGIN TRAN
UPDATE VIAJE.VIAJE
SET calificacion = 6
WHERE id_viaje = 1;
SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN
```

```
--8. Validar que propina sea un valor de 10, 15 o 20
BEGIN TRAN
    UPDATE VIAJE.VIAJE
    SET propina = 11
    WHERE id_viaje = 1;
    SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN

--9. Validar que tipo sea un caracter 'T' o 'E'
BEGIN TRAN
    UPDATE VIAJE.VIAJE
    SET tipo = 'A'
    WHERE id_viaje = 1;
    SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN

--10. Validar que programado sea 'SI' o 'NO'
BEGIN TRAN
    UPDATE VIAJE.VIAJE
    SET programado = 'AA'
    WHERE id_viaje = 1;
    SELECT * FROM VIAJE.VIAJE
ROLLBACK TRAN

--11. Validar que calificacion tenga como default el valor de
5 (SE HIZO EN INSERT)

--UBICACION
SELECT * FROM VIAJE.UBICACION
--1. Validar que id_viaje sea fk (fkViajeUbicacion),
agregando ON DELETE CASCADE
BEGIN TRAN
    UPDATE VIAJE.UBICACION
    SET id_viaje = 261
    WHERE id_ubicacion = 1;
    SELECT * FROM VIAJE.UBICACION
ROLLBACK TRAN

--TIPO ACCIDENTE
SELECT * FROM VIAJE.TIPO_ACCIDENTE
--1. Validar que clave candidata nombre sea UNIQUE
BEGIN TRAN
    UPDATE VIAJE.TIPO_ACCIDENTE
    SET nombre = 'Accidente mecánico'
    WHERE id_tipo = 1;
    SELECT * FROM VIAJE.TIPO_ACCIDENTE
ROLLBACK TRAN

--ACCIDENTE
SELECT * FROM VIAJE.ACCIDENTE
--1. Validar que id_tipo sea fk (fkTipoAccidente)
BEGIN TRAN
```



```
UPDATE VIAJE.ACCIDENTE
SET id_tipo = 261
WHERE id_accidente = 1;
SELECT * FROM VIAJE.ACCIDENTE
ROLLBACK TRAN

--2. Validar que id_viaje sea fk (fkViajeAccidente),
agregando ON DELETE CASCADE y ON UPDATE CASCADE
BEGIN TRAN
UPDATE VIAJE.ACCIDENTE
SET id_viaje = 261
WHERE id_accidente = 1;
SELECT * FROM VIAJE.ACCIDENTE
ROLLBACK TRAN

BEGIN TRAN -- Validando el delete on cascade
DELETE FROM VIAJE.VIAJE
WHERE id_viaje = 1
SELECT * FROM VIAJE.ACCIDENTE
ROLLBACK TRAN

--3. Validar que herido sea 'SI' o 'NO'
BEGIN TRAN
UPDATE VIAJE.ACCIDENTE
SET herido= 'AA'
WHERE id_accidente = 1;
SELECT * FROM VIAJE.ACCIDENTE
ROLLBACK TRAN

--3. Validar que culpable sea 'SI' o 'NO'
BEGIN TRAN
UPDATE VIAJE.ACCIDENTE
SET culpable = 'AA'
WHERE id_accidente = 1;
SELECT * FROM VIAJE.ACCIDENTE
ROLLBACK TRAN
```

Anexo

Véase la figura 1 en el siguiente enlace:

<https://drive.google.com/file/d/1UeU-R1EguPRKjtB2O01MsFfSiSw0KbvQ/view?usp=sharing>

Véase la figura 2 en el siguiente enlace:

<https://drive.google.com/file/d/1CI9BWmdbTXfrdYkHbnQcJ4I82xY6n0Up/view?usp=sharing>

Véase la figura 3 en el siguiente enlace:

<https://drive.google.com/file/d/1CWpcJPRS64btpCyWvkg2I3hvsNIptZGY/view?usp=sharing>

Véase la figura 4 en el siguiente enlace:



<https://drive.google.com/file/d/1-6Y1wbH6WnZTaxdD9VIyI5DC-l7GU6Z0/view?usp=sharing>

Véase la figura 5 en el siguiente enlace:

https://drive.google.com/file/d/1FINaL_MzK2ZkKcodQVmmmaxhaC8Y-HD5g/view?usp=sharing