

CREATE BASEDATOS aerolinea:

Tabla de paises =

id INT AUTO-INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL

Tabla de ciudades =

id INT AUTO-INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
pais_id INT,
FOREIGN KEY (pais_id) REFERENCES pais(id)

Tabla de sucursales =

id INT AUTO-INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
direccion VARCHAR(255),
ciudad_id INT,
FOREIGN KEY (ciudad_id) REFERENCES ciudad(id)

Tabla de clientes =

id INT AUTO-INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
correo VARCHAR(100),
direccion VARCHAR(255),
telefono VARCHAR(15)

Tabla de vehiculos =

id INT AUTO-INCREMENT PRIMARY KEY,
placa VARCHAR(50) NOT NULL UNIQUE,
modelo VARCHAR(100),
capacidad DECIMAL(10,2)

Tabla de conductores

id INT AUTO-INCREMENT PRIMARY KEY,
nombre VARCHAR (100) NOT NULL,
Telefono VARCHAR (15)

Tabla de envios

id INT AUTO-INCREMENT PRIMARY KEY,
cliente_id INT,
paquete_id INT,
vehiculo_id INT,
auxiliar_id INT,
Sucursal_id INT,
fecha_envio DATETIME DEFAULT CURRENT_TIMESTAMP,
estado ENUM ('recibido', 'en_transito', 'entregado', 'retenido-aduana') NOT NULL,

CREATE BASE DATOS campusbike;

```
CREATE TABLE clientes (  
  cliente_id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100),  
  direccion VARCHAR(255),  
  ciudad VARCHAR(100),  
  pais VARCHAR(100),  
  email VARCHAR(100),  
  telefono VARCHAR(20)  
);
```

```
CREATE TABLE proveedores (  
  proveedor_id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100),  
  direccion VARCHAR(255),  
  ciudad VARCHAR(100),  
  pais VARCHAR(100),  
  email VARCHAR(100),  
  telefono VARCHAR(20)  
);
```

```
CREATE TABLE bicicletas (  
  bicicleta_id INT AUTO_INCREMENT PRIMARY KEY,  
  modelo VARCHAR(100),  
  marca VARCHAR(100),  
  precio DECIMAL(10,2),  
  stock INT  
);
```

```
CREATE TABLE detalles_ventas (  
  detalle_id INT AUTO_INCREMENT PRIMARY KEY,  
  venta_id INT,  
  bicicleta_id INT,  
  cantidad INT,  
  precio_unitario DECIMAL(10,2),  
  FOREIGN KEY (venta_id) REFERENCES ventas (venta_id),  
  FOREIGN KEY (bicicleta_id) REFERENCES bicicletas (bicicleta_id)  
);
```



```

CREATE TABLE Ventas (
    Venta_id INT AUTO-INCREMENT PRIMARY KEY,
    Cliente_id INT,
    fecha_venta DATE,
    total DECIMAL (70,2),
    FOREIGN KEY (Cliente_id) REFERENCES Clientes (Cliente_id)
);

```

```

CREATE TABLE compras (
    Compra_id INT AUTO-INCREMENT PRIMARY KEY,
    proveedor_id INT,
    fecha_compra DATE,
    total DECIMAL (70,2),
    FOREIGN KEY (proveedor_id) REFERENCES proveedores (proveedor_id)
);

```

```

CREATE TABLE respuestas (
    respuesta_id INT AUTO-INCREMENT PRIMARY KEY,
    nombre VARCHAR (100),
    precio DECIMAL (70,2),
    stock INT,
    proveedor_id INT,
    FOREIGN KEY (proveedor_id) REFERENCES proveedores (proveedor_id)
);

```

```

SELECT C.nombre, sum (V.total) AS
total_compras
FROM Clientes C
JOIN Ventas V ON C.cliente_id = V.cliente_id
GROUP BY C.cliente_id;

```

```

SELECT B.marca, B.modelo, sum (DV.cantidad) AS
total_vendido
FROM Bicicleta B
JOIN detalles_venta DV ON B.bicicleta_id =
DV.bicicleta_id
GROUP BY B.marca, B.modelo
ORDER BY total_vendido DESC;

```

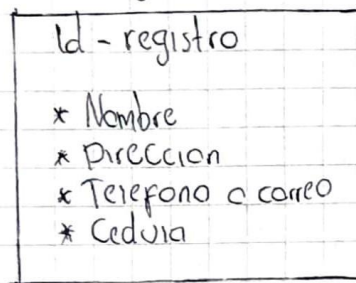


```
SELECT C.nombre, Sum (V.total) AS total-gastado  
FROM clientes C  
JOIN ventas V ON C.clientes_id = V.cliente_id  
WHERE YEAR (V.fecha-venta) = 2023  
GROUP BY C.cliente_id  
ORDER BY total-gastado DESC  
limit 10;
```

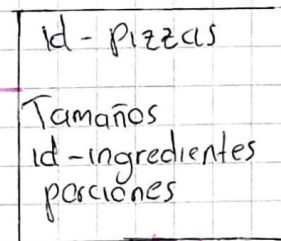
```
SELECT modelo, marca, stock  
FROM bicicletas  
WHERE stock > 0;
```

Pizzeria.

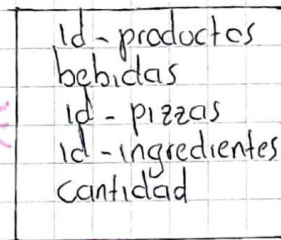
Registros



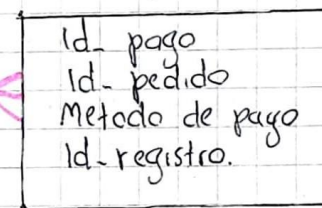
Pizzas



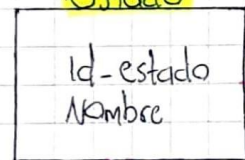
Productos



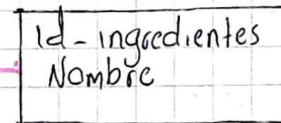
Pago



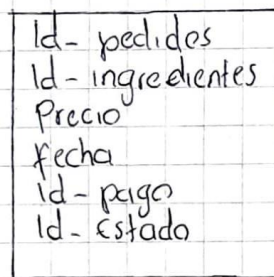
Estado



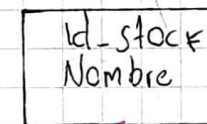
Ingredientes



Pedidos



Stock



--CODIGO AEROLINEA--

```
CREATE TABLE Pais (  
  id VARCHAR(5) PRIMARY KEY,  
  nombre VARCHAR(30)  
);  
CREATE TABLE Ciudad (  
  id VARCHAR(5) PRIMARY KEY,  
  nombre VARCHAR(30),  
  idpais VARCHAR(5),  
  FOREIGN KEY (idpais) REFERENCES Pais(id)  
);
```

```
CREATE TABLE Aeropuerto (  
  id VARCHAR(5) PRIMARY KEY,  
  nombre VARCHAR(50),  
  idciudad VARCHAR(5),  
  FOREIGN KEY (idciudad) REFERENCES Ciudad(id)  
);
```

```
CREATE TABLE Gates (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nroPuerta VARCHAR(5),  
  idAeropuerto VARCHAR(5),  
  FOREIGN KEY (idAeropuerto) REFERENCES Aeropuerto(id)  
);
```

```
CREATE TABLE Fabricante (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(40)  
);
```

```
CREATE TABLE Modelo (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(30),  
  idfabricante INT,  
  FOREIGN KEY (idfabricante) REFERENCES Fabricante(id)  
);
```

```
CREATE TABLE Estado (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(30)  
);
```

```
CREATE TABLE Avion (  
    nromatricula VARCHAR(30) UNIQUE,  
    capacidad INT(3),  
    fechaFabricacion DATE,  
    idestado INT,  
    idmodelo INT,  
    idAerolinea INT,  
    PRIMARY KEY (nromatricula),  
    FOREIGN KEY (idestado) REFERENCES Estado(id),  
    FOREIGN KEY (idmodelo) REFERENCES Modelo(id),  
    FOREIGN KEY (idAerolinea) REFERENCES Aerolinea(id)  
);
```

```
CREATE TABLE Aerolinea (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(30)  
);
```

```
CREATE TABLE Trayecto (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    fechaTrayecto DATE,  
    valor DOUBLE,  
    ciudadOrigen VARCHAR(5),  
    ciudadDestino VARCHAR(5),  
    FOREIGN KEY (ciudadOrigen) REFERENCES Ciudad(id),  
    FOREIGN KEY (ciudadDestino) REFERENCES Ciudad(id)  
);
```

```
CREATE TABLE Escala (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    idtrayecto INT,  
    idavion VARCHAR(30),
```



```

    nrovuelo INT,
    idAeropuerto VARCHAR(5),
    FOREIGN KEY (idtrayecto) REFERENCES Trayecto(id),
    FOREIGN KEY (idavion) REFERENCES Avion(nromatricula),
    FOREIGN KEY (idAeropuerto) REFERENCES Aeropuerto(id)
);
CREATE TABLE Roltripulacion (
    id INT PRIMARY KEY,
    nombre VARCHAR(30)
);

CREATE TABLE Empleado (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50),
    idrol INT,
    fechaIngreso DATE,
    idAerolinea INT,
    idAeropuerto VARCHAR(5),
    FOREIGN KEY (idrol) REFERENCES Roltripulacion(id),
    FOREIGN KEY (idAerolinea) REFERENCES Aerolinea(id),
    FOREIGN KEY (idAeropuerto) REFERENCES Aeropuerto(id)
);
CREATE TABLE TrayectoTripulacion (
    idempleado INT,
    idescala INT,
    FOREIGN KEY (idempleado) REFERENCES Empleado(id),
    FOREIGN KEY (idescala) REFERENCES Escala(id)
);

CREATE TABLE Revision (
    id INT AUTO_INCREMENT PRIMARY KEY,
    fecha DATE,
    idavion VARCHAR(30),
    FOREIGN KEY (idavion) REFERENCES Avion(nromatricula)
);

CREATE TABLE Revempleado (
    idempleado INT,

```

```
    idrevision INT,  
    FOREIGN KEY (idempleado) REFERENCES Empleado(id),  
    FOREIGN KEY (idrevision) REFERENCES Revision(id)  
);
```

```
CREATE TABLE DetalleRevision (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    idrevision INT,  
    descripcion TEXT,  
    fechaRev DATE,  
    idempleado INT,  
    FOREIGN KEY (idrevision) REFERENCES Revision(id),  
    FOREIGN KEY (idempleado) REFERENCES Empleado(id)  
);
```

```
CREATE TABLE Clientes (  
    id VARCHAR(20) PRIMARY KEY,  
    edad INT,  
    idtipodoc INT,  
    FOREIGN KEY (idtipodoc) REFERENCES Tipodocumento(id)  
);
```

```
CREATE TABLE Tipodocumento (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(30)  
);
```

```
CREATE TABLE Reservaviaje(  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    fecha DATE,  
    idtrayecto INT,  
    FOREIGN KEY (idtrayecto) REFERENCES Trayecto(id)  
);
```

```
CREATE TABLE DetalleReserva (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    idcliente VARCHAR(20),  
    idtarifa INT,
```



```
    valorTarifa DOUBLE,  
    FOREIGN KEY (idcliente) REFERENCES Clientes(id),  
    FOREIGN KEY (idtarifa) REFERENCES TarifasVuelo(id)  
);
```

```
CREATE TABLE TarifasVuelo (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    descripcion VARCHAR(100),  
    valor DOUBLE  
);
```

--CODIGO DE CAMPUSBIKE--

```
CREATE TABLE Clientes (  
    cliente_id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    direccion VARCHAR(255),  
    ciudad VARCHAR(100),  
    pais VARCHAR(100),  
    email VARCHAR(100),  
    telefono VARCHAR(20)  
);  
CREATE TABLE Proveedores (  
    proveedor_id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    direccion VARCHAR(255),  
    ciudad VARCHAR(100),  
    pais VARCHAR(100),  
    email VARCHAR(100),  
    telefono VARCHAR(20)  
);  
CREATE TABLE Bicicletas (  
    bicicleta_id INT AUTO_INCREMENT PRIMARY KEY,  
    modelo VARCHAR(100),  
    marca VARCHAR(100),  
    precio DECIMAL(10, 2),  
    stock INT
```

```

);
CREATE TABLE Detalles_Venta (
    detalle_id INT AUTO_INCREMENT PRIMARY KEY,
    venta_id INT,
    bicicleta_id INT,
    cantidad INT,
    precio_unitario DECIMAL(10, 2),
    FOREIGN KEY (venta_id) REFERENCES Ventas(venta_id),
    FOREIGN KEY (bicicleta_id) REFERENCES Bicicletas(bicicleta_id)
);
CREATE TABLE Ventas (
    venta_id INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT,
    fecha_venta DATE,
    total DECIMAL(10, 2),
    FOREIGN KEY (cliente_id) REFERENCES Clientes(cliente_id)
);
CREATE TABLE Compras (
    compra_id INT AUTO_INCREMENT PRIMARY KEY,
    proveedor_id INT,
    fecha_compra DATE,
    total DECIMAL(10, 2),
    FOREIGN KEY (proveedor_id) REFERENCES Proveedores(proveedor_id)
);
CREATE TABLE Repuestos (
    repuesto_id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    precio DECIMAL(10, 2),
    stock INT,
    proveedor_id INT,
    FOREIGN KEY (proveedor_id) REFERENCES Proveedores(proveedor_id)
);
SELECT C.nombre, SUM(V.total) AS total_compras
FROM Clientes C
JOIN Ventas V ON C.cliente_id = V.cliente_id
GROUP BY C.cliente_id;
[14:55, 21/10/2024] * Adrian-Pérez: SELECT B.marca, B.modelo, SUM(DV.cantidad) AS total_vendido
FROM Bicicletas B
JOIN Detalles_Venta DV ON B.bicicleta_id = DV.bicicleta_id
GROUP BY B.marca, B.modelo
ORDER BY total_vendido DESC;
SELECT C.nombre, SUM(V.total) AS total_gastado

```



```

FROM Clientes C
JOIN Ventas V ON C.cliente_id = V.cliente_id
WHERE YEAR(V.fecha_venta) = 2023
GROUP BY C.cliente_id
ORDER BY total_gastado DESC
LIMIT 10;
SELECT modelo, marca, stock
FROM Bicicletas
WHERE stock > 0;
SELECT P.nombre, COUNT(C.compra_id) AS total_compras
FROM Proveedores P
JOIN Compras C ON P.proveedor_id = C.proveedor_id
WHERE C.fecha_compra BETWEEN '2024-09-01' AND '2024-09-30'
GROUP BY P.proveedor_id
ORDER BY total_compras DESC;
DELIMITER $$

```

```

CREATE PROCEDURE RegistrarVenta (
    IN p_cliente_id INT,
    IN p_fecha_venta DATE,
    IN p_total DECIMAL(10, 2)
)
BEGIN
    INSERT INTO Ventas (cliente_id, fecha_venta, total)
    VALUES (p_cliente_id, p_fecha_venta, p_total);
END $$

```

```

DELIMITER ;
[14:55, 21/10/2024] * Adrian-Pérez: DELIMITER $$

```

```

CREATE PROCEDURE ActualizarInventario (
    IN p_bicicleta_id INT,
    IN p_cantidad INT
)
BEGIN
    UPDATE Bicicletas
    SET stock = stock - p_cantidad
    WHERE bicicleta_id = p_bicicleta_id;
END $$

```

```

DELIMITER ;
[14:55, 21/10/2024] * Adrian-Pérez: CREATE FUNCTION TotalVentasMensuales(mes INT, anio INT)

```

```

RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
    DECLARE total DECIMAL(10, 2);
    SELECT SUM(total) INTO total
    FROM Ventas
    WHERE MONTH(fecha_venta) = mes AND YEAR(fecha_venta) = anio;
    RETURN total;
END;
[14:55, 21/10/2024] * Adrian-Pérez: CREATE INDEX idx_fecha_venta ON Ventas(fecha_venta);
CREATE INDEX idx_marca ON Bicicletas(marca);
[14:55, 21/10/2024] * Adrian-Pérez: CREATE DATABASE TiendaBicicletas;
USE TiendaBicicletas;

--CODIGO LOGICA--
CREATE TABLE Pais (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL
);

CREATE TABLE Ciudad (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    id_pais INT,
    FOREIGN KEY (id_pais) REFERENCES Pais(id)
);

CREATE TABLE Sucursal (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    direccion VARCHAR(100),
    id_ciudad INT,
    FOREIGN KEY (id_ciudad) REFERENCES Ciudad(id)
);

CREATE TABLE Cliente (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    correo VARCHAR(100),
    direccion VARCHAR(100),
    telefono VARCHAR(15)
);

```



```

CREATE TABLE Paquete (
    id INT AUTO_INCREMENT PRIMARY KEY,
    numero_seguimiento VARCHAR(50) NOT NULL UNIQUE,
    peso DECIMAL(5,2),
    dimensiones VARCHAR(50),
    contenido VARCHAR(100),
    valor_declarado DECIMAL(10,2),
    tipo_servicio ENUM('nacional', 'internacional', 'expres', 'estándar')
);

CREATE TABLE Envio (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_cliente INT,
    id_paquete INT,
    fecha_envio DATE,
    estado ENUM('recibido', 'en tránsito', 'entregado', 'retenido en aduana'),
    id_sucursal INT,
    FOREIGN KEY (id_cliente) REFERENCES Cliente(id),
    FOREIGN KEY (id_paquete) REFERENCES Paquete(id),
    FOREIGN KEY (id_sucursal) REFERENCES Sucursal(id)
);

CREATE TABLE SeguimientoPaquete (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_paquete INT,
    ubicacion VARCHAR(50),
    fecha DATETIME,
    estado ENUM('en tránsito', 'entregado', 'retenido en aduana'),
    FOREIGN KEY (id_paquete) REFERENCES Paquete(id)
);

CREATE TABLE Vehiculo (
    id INT AUTO_INCREMENT PRIMARY KEY,
    placa VARCHAR(15) NOT NULL UNIQUE,
    modelo VARCHAR(50),
    capacidad DECIMAL(5,2)
);

CREATE TABLE Conductor (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    telefono VARCHAR(15)
);

```

```
CREATE TABLE Ruta (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  descripcion VARCHAR(100),  
  id_sucursal INT,  
  FOREIGN KEY (id_sucursal) REFERENCES Sucursal(id)  
);
```

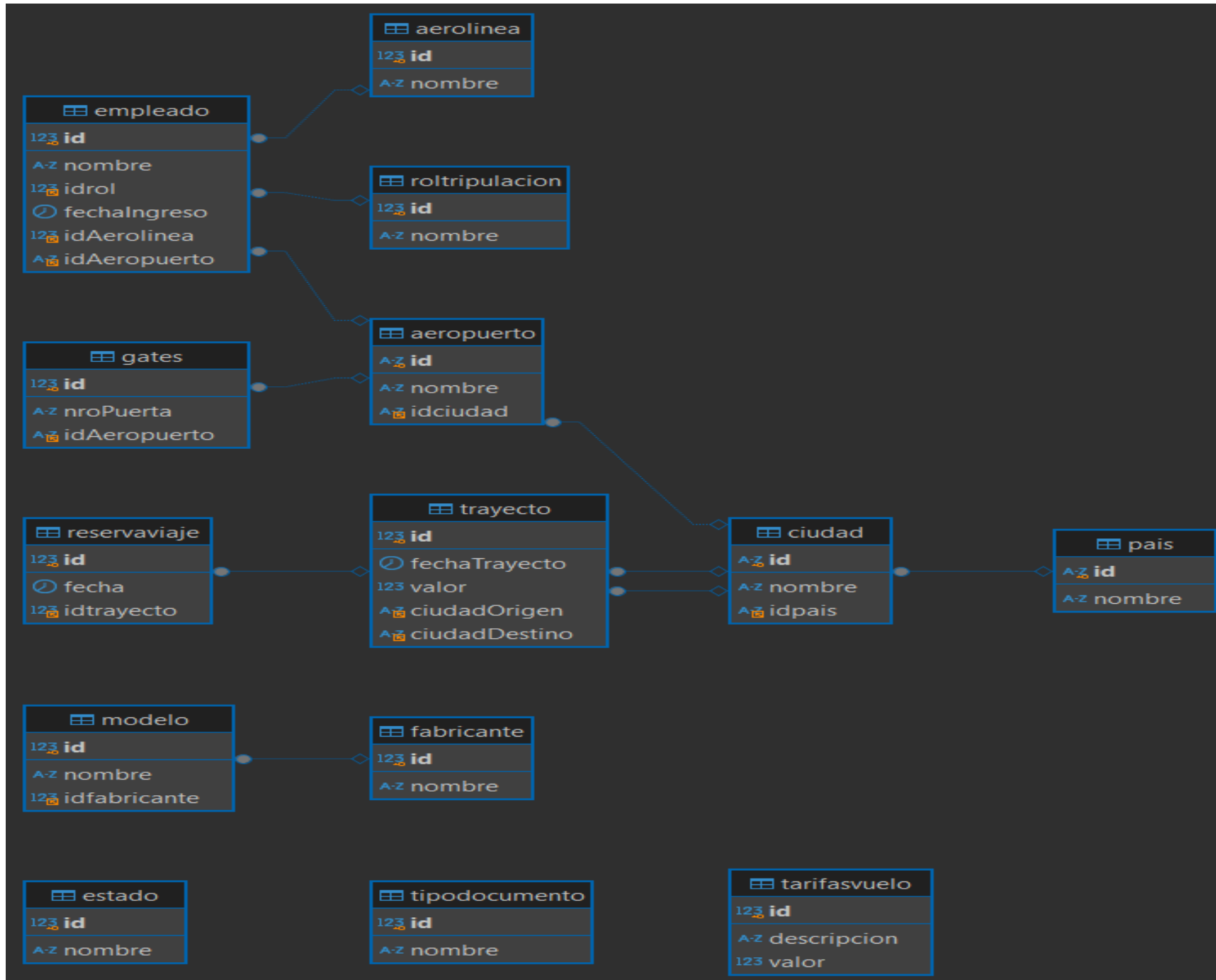
```
CREATE TABLE AsignacionRuta (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  id_ruta INT,  
  id_conductor INT,  
  id_vehiculo INT,  
  fecha_asignacion DATE,  
  FOREIGN KEY (id_ruta) REFERENCES Ruta(id),  
  FOREIGN KEY (id_conductor) REFERENCES Conductor(id),  
  FOREIGN KEY (id_vehiculo) REFERENCES Vehiculo(id)  
);
```

```
CREATE TABLE AuxiliarReparto (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100) NOT NULL,  
  telefono VARCHAR(15)  
);
```

```
CREATE TABLE AsignacionAuxiliar (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  id_ruta INT,  
  id_auxiliar INT,  
  fecha_asignacion DATE,  
  FOREIGN KEY (id_ruta) REFERENCES Ruta(id),  
  FOREIGN KEY (id_auxiliar) REFERENCES AuxiliarReparto(id)  
);
```

```
SELECT E.id, E.fecha_envio, P.numero_seguimiento, E.estado  
FROM Envio E  
JOIN Cliente C ON E.id_cliente = C.id  
JOIN Paquete P ON E.id_paquete = P.id  
WHERE C.id = :id_cliente;
```

```
SELECT ubicacion, fecha, estado  
FROM SeguimientoPaquete  
WHERE id_paquete = :id_paquete  
ORDER BY fecha DESC  
LIMIT 1;
```



compras
123 compra_id
123 proveedor_id
🕒 fecha_compra
123 total

proveedores
123 proveedor_id
A-Z nombre
A-Z direccion
A-Z ciudad
A-Z pais
A-Z email
A-Z telefono

repuestos
123 repuesto_id
A-Z nombre
123 precio
123 stock
123 proveedor_id

ventas
123 venta_id
123 cliente_id
🕒 fecha_venta
123 total

clientes
123 cliente_id
A-Z nombre
A-Z direccion
A-Z ciudad
A-Z pais
A-Z email
A-Z telefono

bicicletas
123 bicicleta_id
A-Z modelo
A-Z marca
123 precio
123 stock

