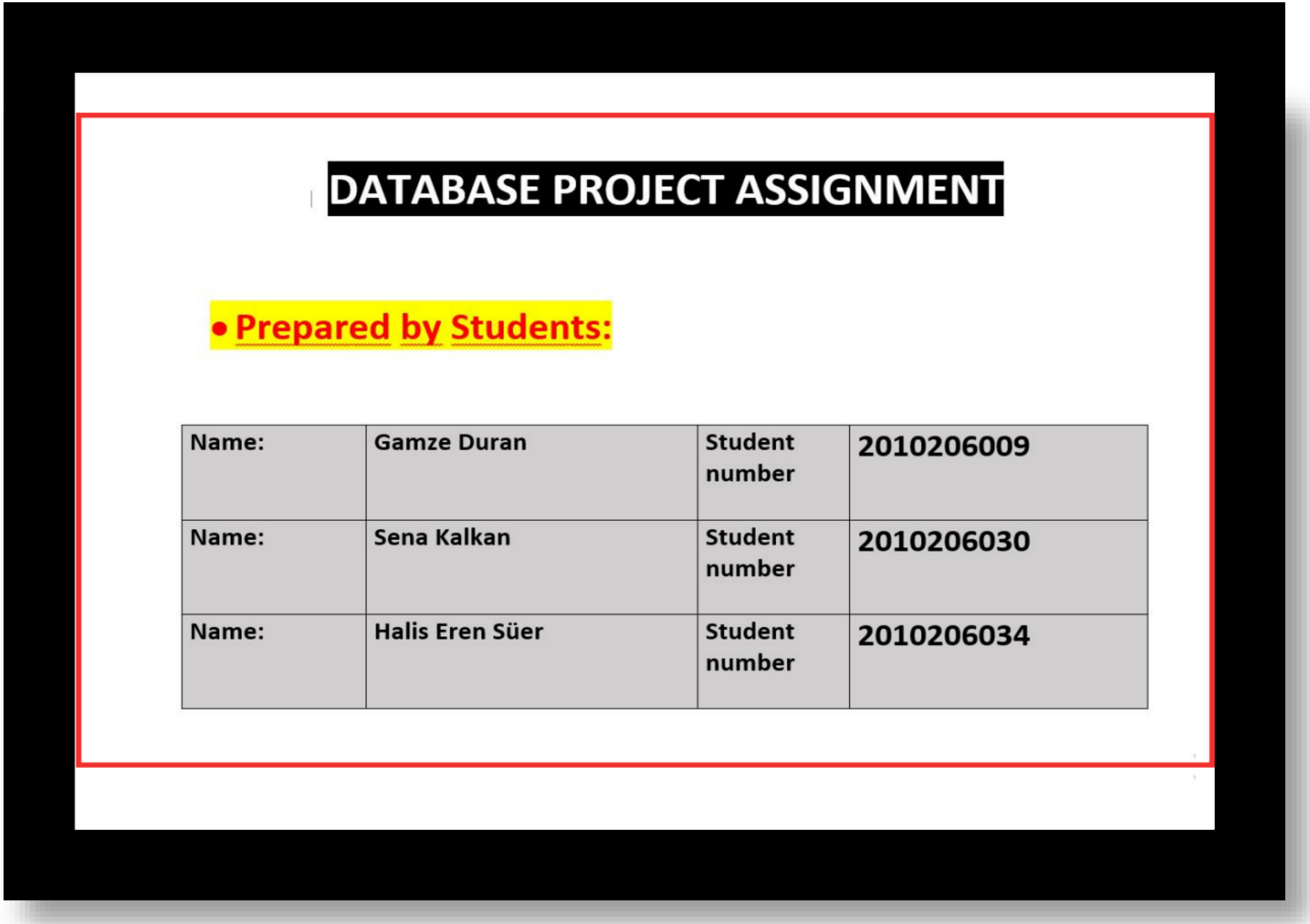


DATABASE PROJECT ASSIGNMENT

- **Prepared by Students:**

Name:	**Gamze Duran**	**Student number**	**2010206009**
Name:	**Sena Kalkan**	**Student number**	**2010206030**
Name:	**Halis Eren Süer**	**Student number**	**2010206034**

Scenario

Your database will be designed for a bus ticket company and will contain entities such as customers, seats, reservations, and offers

Scenario Details:

1. Customer Registration: Customers need to be recorded in the database. It is important to store the following information for each customer: first name, last name, email, phone number, date of birth, gender, and address details.
2. Seats: The seats on the buses should be defined in the database. Each seat should have a number, information about its location (e.g., window-side, aisle-side), and the bus number it belongs to.
3. Reservations: Customers' reservations should be stored in the database. Each reservation should be linked to a customer and a seat. It should also include the date and time of the reservation. The system should prevent multiple reservations for the same seat.
4. Campaigns: The campaigns offered by the company should be defined in the database. Each campaign should have a name, description, start date, and end date. These are the main elements to consider for designing a database for an bus ticket company. They represent the initial requirements and can be further elaborated and developed during the project.

Developing an ERD:

1. Needs Analysis: Firstly, we conducted a detailed needs analysis regarding customers, seats, reservations, and campaigns. We determined the information we wanted to store for each entity.
2. Entity Identification: Based on the needs analysis, we identified the entities that should be present in the database design. In this case, customers, seats, reservations, and campaigns are the main entities.
3. Entity Attribute Identification: We defined the attributes for each entity.
4. Determining Relationships: We established relationships between entities. For example, the customer-reservation relationship reflects a "one-to-many" relationship, while the reservation-seat relationship can also reflect a "one-to-many" relationship.
5. Creating an ERD: We created an ERD to clearly represent the entities and relationships.
6. Normalization: We normalized your database. Normalization aims to better organize the database to avoid data redundancy and ensure consistency.
7. Table Creation: We created a table for each entity. For example, a "Customer" table for the "Customer" entity.

Define At least 7 Entities:

1. Customers 2. Buses 3. Expeditions 4. Seats 5. Reservations 6. Tickets 7. Campaigns

Description of Attributes:

1. Customers:

- **Customer ID (unique ID)**
- **Name**
- **Surname**
- **Phone number**
- **E-mail address**
- **Address**

2. Buses:

- **Bus ID (unique ID)**
- **Bus number**
- **Brand/model**
- **Capacity**

3. Expeditions:

- **Expedition ID (unique ID)**
- **Departure point**
- **Destination**
- **Departure time**
- **Arrival time**

4. Seats:

- **Seat ID (unique ID)**
- **Bus ID (ID of the bus to which the seat is attached)**
- **Seat number**
- **Seating type (windowside, aisle)**
- **Reservation status (booked or empty)**

5. Reservations:

- **Reservation ID (unique ID)**

- Customer ID (ID of the customer who made the reservation)
- Voyage ID (ID of the voyage to which the reservation is bound)
- Seat ID (ID of the seat to which the reservation is connected)
- Reservation date

6. Tickets:

- Ticket ID (unique ID)
- Customer ID (ID of the customer who purchased the ticket)
- Expedition ID (ID of the voyage to which the ticket is bound)
- Seat ID (ID of the seat to which the ticket is bound)
- Ticket price
- Ticket sales date

7. Campaigns:

- Campaign ID (unique ID)
- Campaign name
- Discount rate
- Validity start date
- Expiry date

Identification of all necessary relationships (define with Matrix diagram):

Entities	Customers	Buses	Expeditions	Seats	Reservations	Tickets	Campaigns
Customers							x
Buses			x	x			
Expeditions		x					
Seats		x					
Reservations	X		x	x		x	
Tickets	X				x		
Campaigns	X						

Supertype-subtype, spring and hierarchical structures, nontransferable relationship :

There is no explicit supertype-subtype relationship mentioned in the provided entities.

The arc relationships include:

Reservations are associated with Seats and Expeditions.

Tickets are generated from Reservations.

The hierarchical structure exists between Tickets, Reservations, Seats, Buses, and Expeditions.

A non-transferable relationship can be established between Tickets and Customers or between Tickets and Reservations, indicating that a ticket is specifically tied to a particular customer or reservation and cannot be transferred to another customer or reservation.

Prepare "table example graph" to map ERD to DB structure:

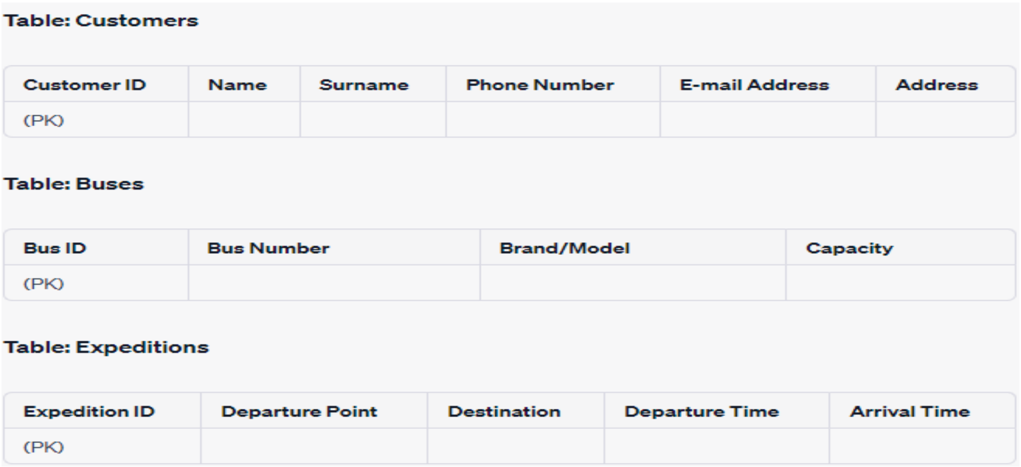


Table: Customers

Customer ID	Name	Surname	Phone Number	E-mail Address	Address
(PK)					

Table: Buses

Bus ID	Bus Number	Brand/Model	Capacity
(PK)			

Table: Expeditions

Expedition ID	Departure Point	Destination	Departure Time	Arrival Time
(PK)				

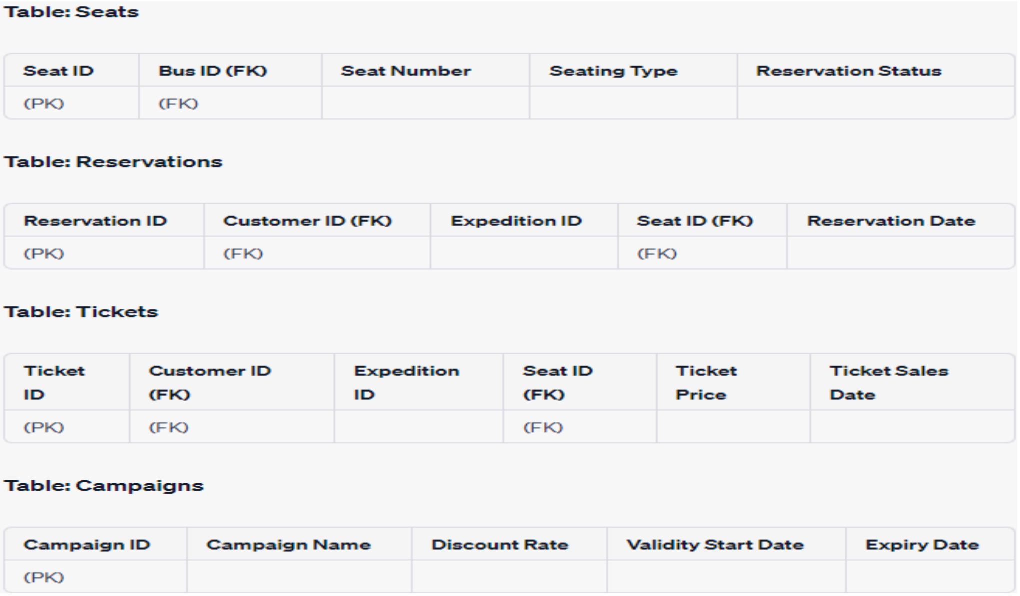


Table: Seats

Seat ID	Bus ID (FK)	Seat Number	Seating Type	Reservation Status
(PK)	(FK)			

Table: Reservations

Reservation ID	Customer ID (FK)	Expedition ID	Seat ID (FK)	Reservation Date
(PK)	(FK)		(FK)	

Table: Tickets

Ticket ID	Customer ID (FK)	Expedition ID	Seat ID (FK)	Ticket Price	Ticket Sales Date
(PK)	(FK)		(FK)		

Table: Campaigns

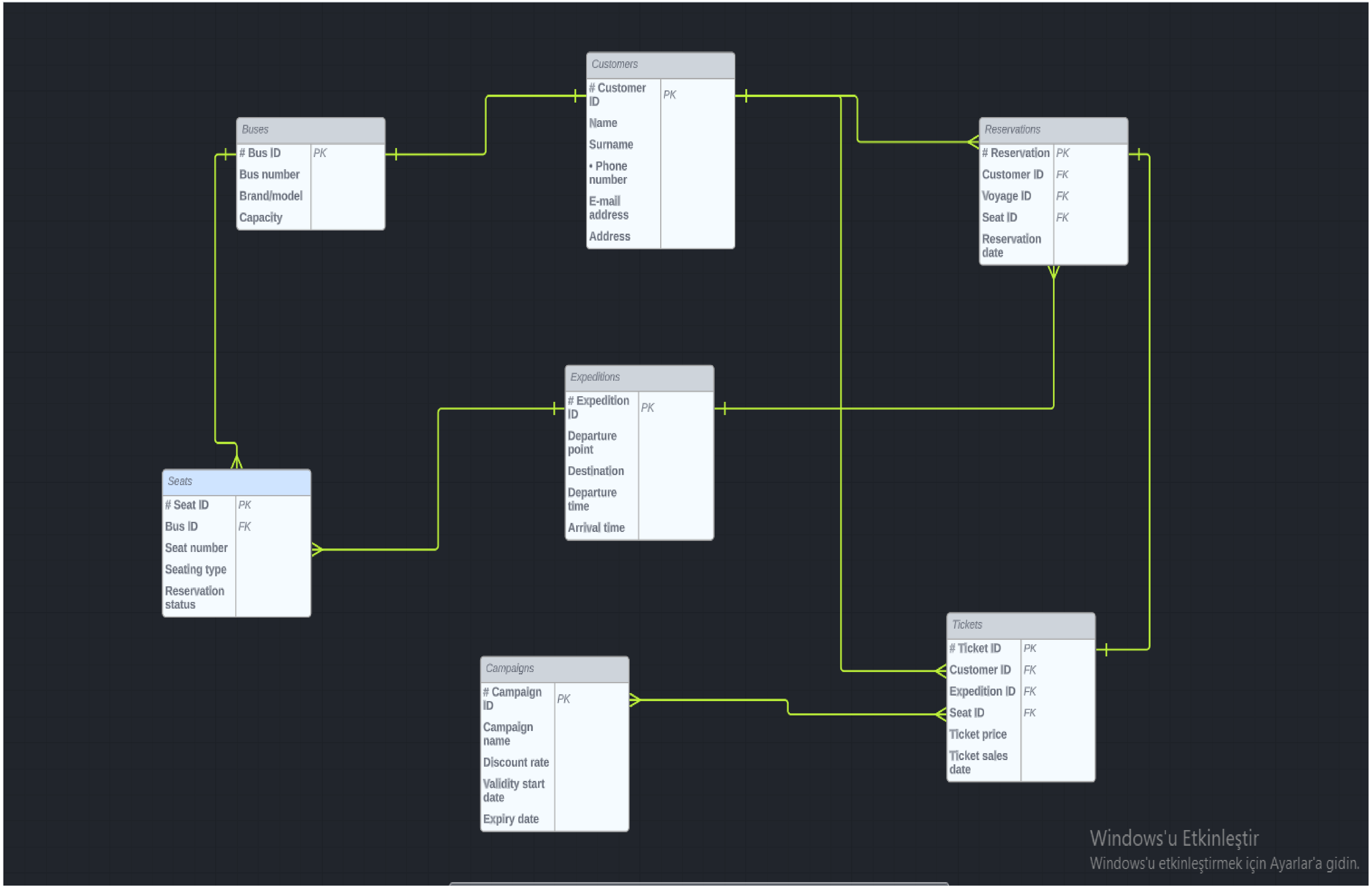
Campaign ID	Campaign Name	Discount Rate	Validity Start Date	Expiry Date
(PK)				

Writing SQL DDL statements for implementing ERD (create table, constraints, defining keys: pks and fks)

```
-- Create Customers table
CREATE TABLE Customers (
  CustomerID INT PRIMARY KEY,
  Name VARCHAR(50),
  Surname VARCHAR(50),
  PhoneNumber VARCHAR(15),
  EmailAddress VARCHAR(100),
  Address VARCHAR(100)
);

-- Create Buses table
CREATE TABLE Buses (
  BusID INT PRIMARY KEY,
  BusNumber VARCHAR(20),
  BrandModel VARCHAR(50),
  Capacity INT
);

-- Create Expeditions table
CREATE TABLE Expeditions (
  ExpeditionID INT PRIMARY KEY,
  DeparturePoint VARCHAR(100),
  Destination VARCHAR(100),
  DepartureTime DATETIME,
  ArrivalTime DATETIME
);

-- Create Seats table
CREATE TABLE Seats (
  SeatID INT PRIMARY KEY,
  BusID INT,
  SeatNumber INT,
  SeatingType VARCHAR(20),
  ReservationStatus VARCHAR(10),
  FOREIGN KEY (BusID) REFERENCES Buses(BusID)
);
```

```
-- Create Reservations table
CREATE TABLE Reservations (
  ReservationID INT PRIMARY KEY,
  CustomerID INT,
  ExpeditionID INT,
  SeatID INT,
  ReservationDate DATE,
  FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
  FOREIGN KEY (ExpeditionID) REFERENCES Expeditions(ExpeditionID),
  FOREIGN KEY (SeatID) REFERENCES Seats(SeatID)
);

-- Create Tickets table
CREATE TABLE Tickets (
  TicketID INT PRIMARY KEY,
  CustomerID INT,
  ExpeditionID INT,
  SeatID INT,
  TicketPrice DECIMAL(10,2),
  TicketSalesDate DATE,
  FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
  FOREIGN KEY (ExpeditionID) REFERENCES Expeditions(ExpeditionID),
  FOREIGN KEY (SeatID) REFERENCES Seats(SeatID)
);

-- Create Campaigns table
CREATE TABLE Campaigns (
  CampaignID INT PRIMARY KEY,
  CampaignName VARCHAR(50),
  DiscountRate DECIMAL(5,2),
  ValidityStartDate DATE,
  ExpiryDate DATE
);
```

BUS COMPANY SCENARIO ER DIAGRAM



Customers	
# Customer ID	PK
Name	
Surname	
• Phone number	
E-mail address	
Address	

Buses	
# Bus ID	PK
Bus number	
Brand/model	
Capacity	

Reservations	
# Reservation	PK
Customer ID	FK
Voyage ID	FK
Seat ID	FK
Reservation date	

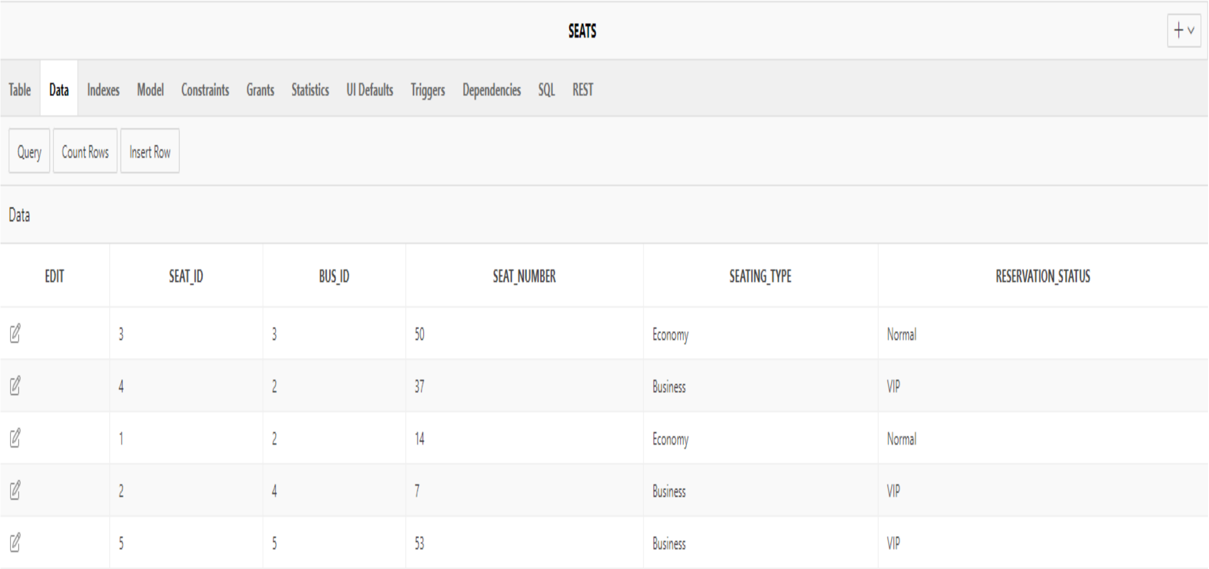
Expeditions	
# Expedition ID	PK
Departure point	
Destination	
Departure time	
Arrival time	

Seats	
# Seat ID	PK
Bus ID	FK
Seat number	
Seating type	
Reservation status	

Campaigns	
# Campaign ID	PK
Campaign name	
Discount rate	
Validity start date	
Expiry date	

Tickets	
# Ticket ID	PK
Customer ID	FK
Expedition ID	FK
Seat ID	FK
Ticket price	
Ticket sales date	

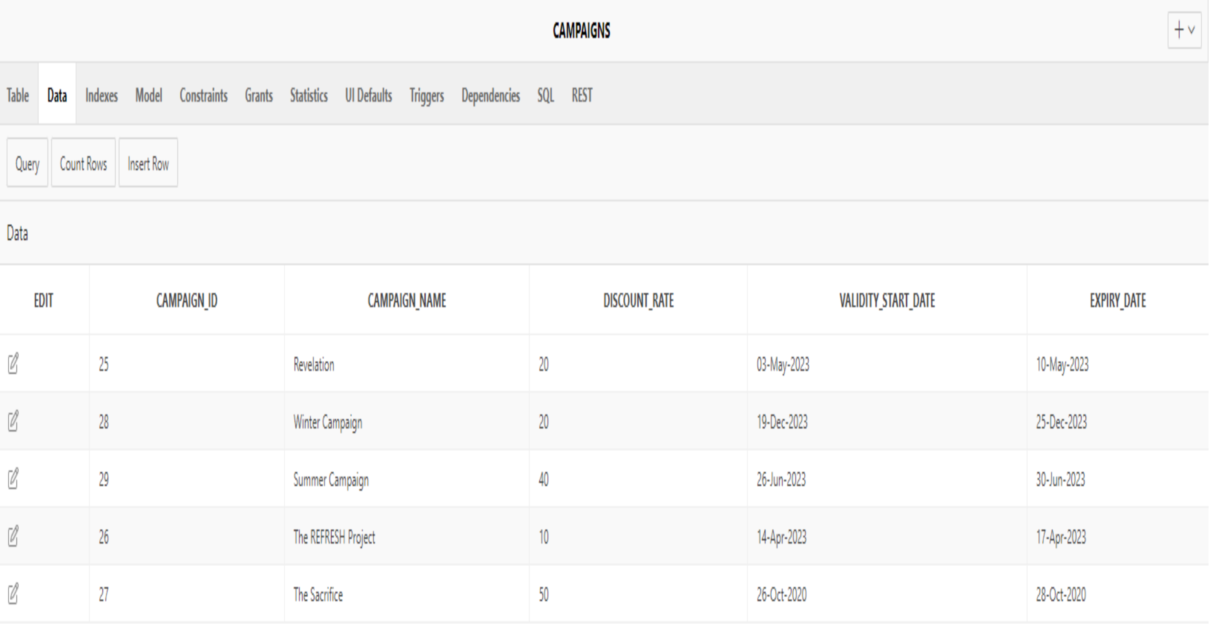
TABLES



SEATS

Data

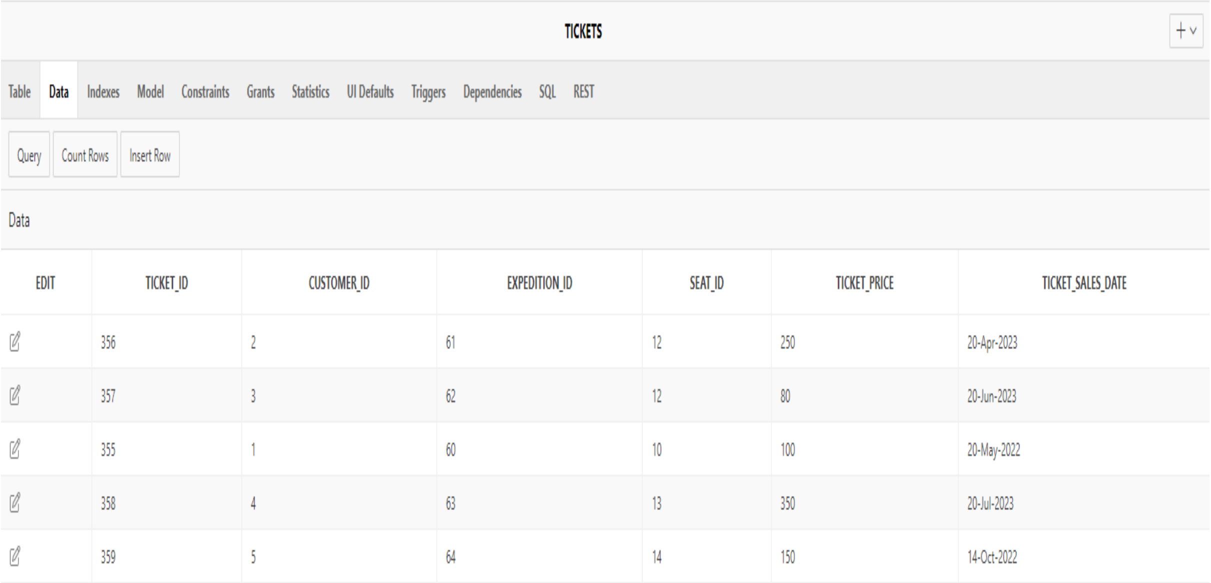
EDIT	SEAT_ID	BUS_ID	SEAT_NUMBER	SEATING_TYPE	RESERVATION_STATUS
✎	3	3	50	Economy	Normal
✎	4	2	37	Business	VIP
✎	1	2	14	Economy	Normal
✎	2	4	7	Business	VIP
✎	5	5	53	Business	VIP



CAMPAIGNS

Data

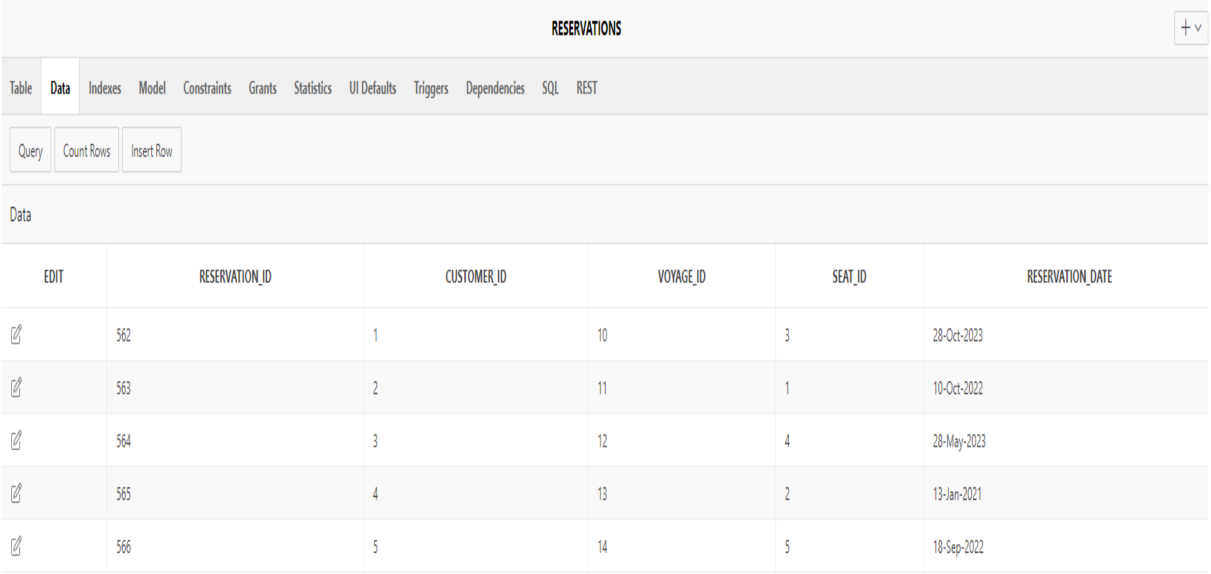
EDIT	CAMPAIGN_ID	CAMPAIGN_NAME	DISCOUNT_RATE	VALIDITY_START_DATE	EXPIRY_DATE
✎	25	Revelation	20	03-May-2023	10-May-2023
✎	28	Winter Campaign	20	19-Dec-2023	25-Dec-2023
✎	29	Summer Campaign	40	26-Jun-2023	30-Jun-2023
✎	26	The REFRESH Project	10	14-Apr-2023	17-Apr-2023
✎	27	The Sacrifice	50	26-Oct-2020	28-Oct-2020



TICKETS

Data

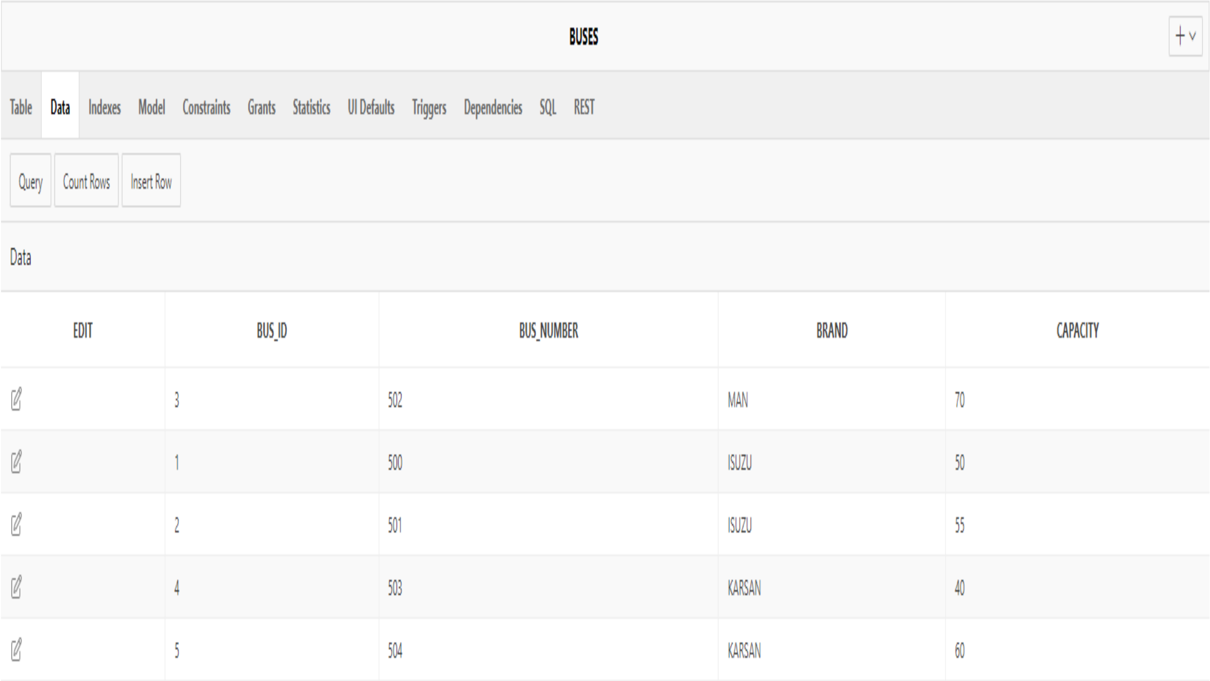
EDIT	TICKET_ID	CUSTOMER_ID	EXPEDITION_ID	SEAT_ID	TICKET_PRICE	TICKET_SALES_DATE
✎	356	2	61	12	250	20-Apr-2023
✎	357	3	62	12	80	20-Jun-2023
✎	355	1	60	10	100	20-May-2022
✎	358	4	63	13	350	20-Jul-2023
✎	359	5	64	14	150	14-Oct-2022



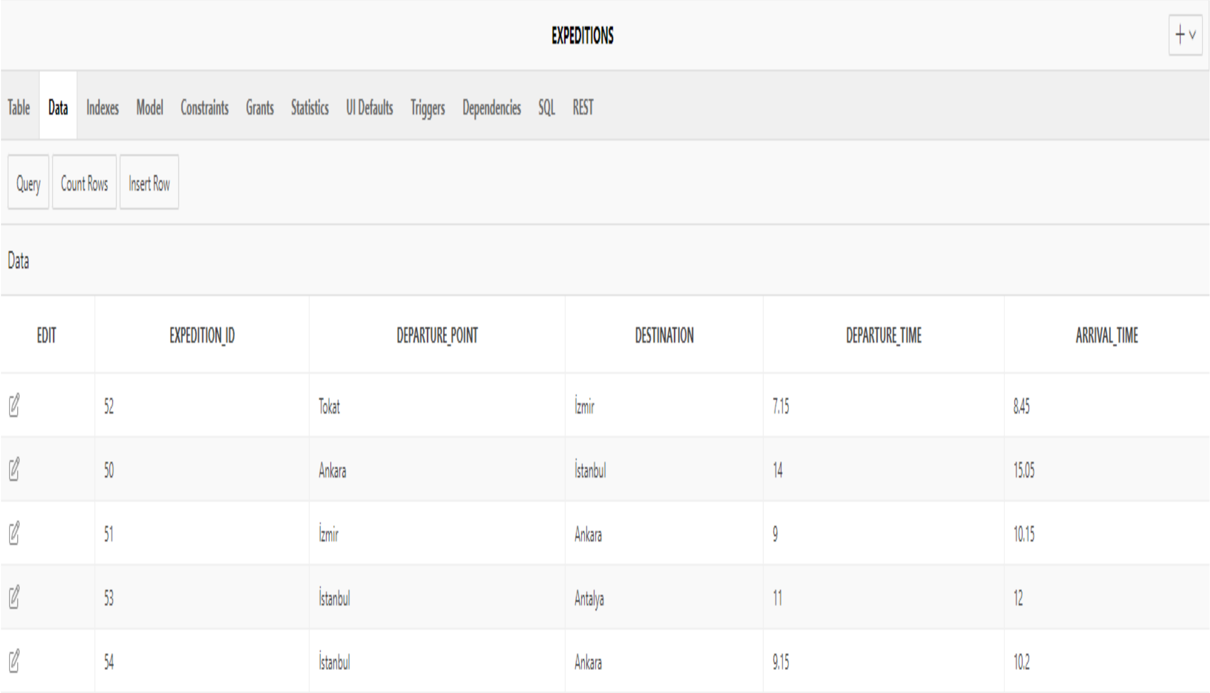
RESERVATIONS

Data

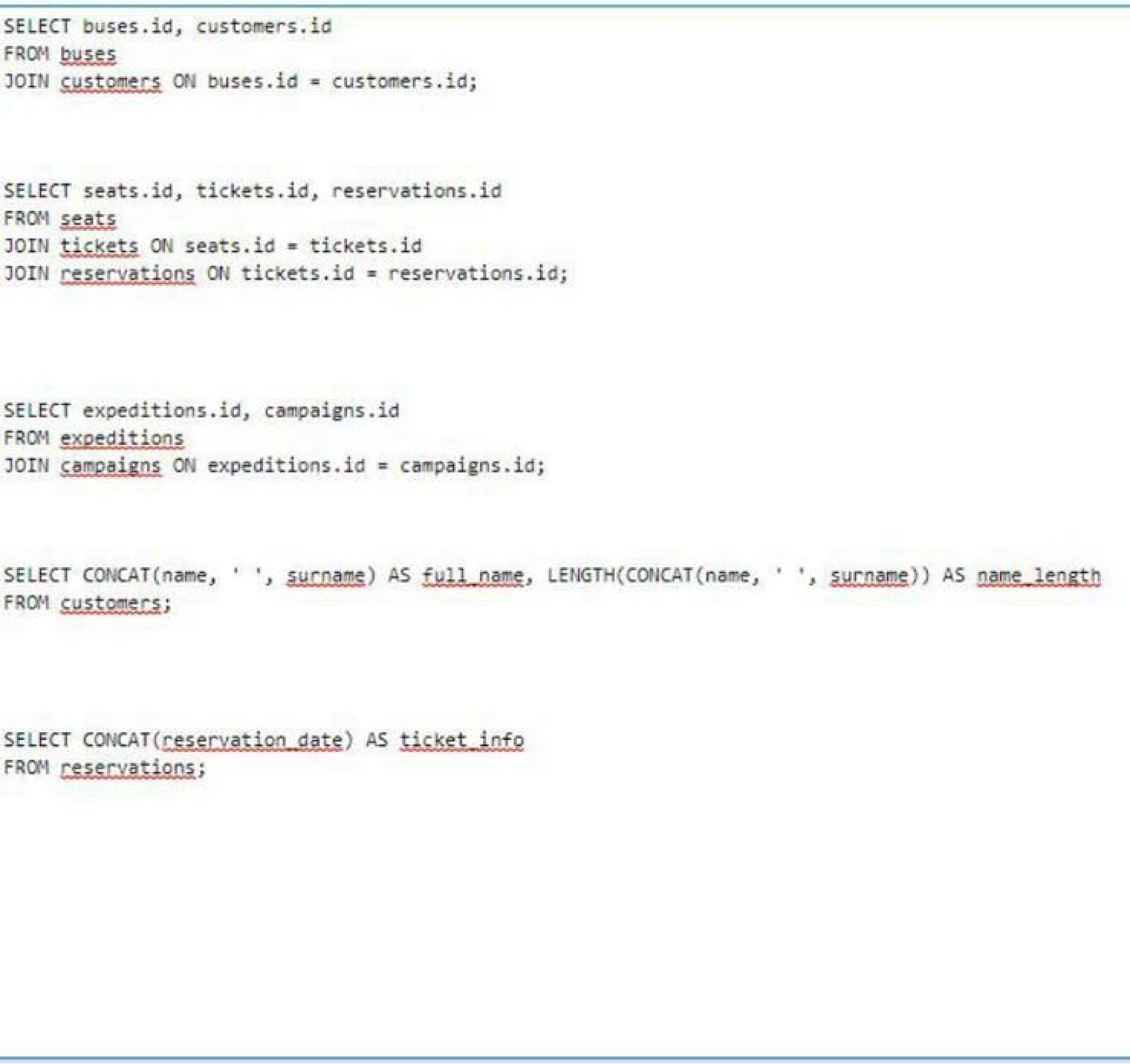
EDIT	RESERVATION_ID	CUSTOMER_ID	VOYAGE_ID	SEAT_ID	RESERVATION_DATE
✎	562	1	10	3	28-Oct-2023
✎	563	2	11	1	10-Oct-2022
✎	564	3	12	4	28-May-2023
✎	565	4	13	2	13-Jan-2021
✎	566	5	14	5	18-Sep-2022



EDIT	BUS_ID	BUS_NUMBER	BRAND	CAPACITY
✎	3	502	MAN	70
✎	1	500	ISUZU	50
✎	2	501	ISUZU	55
✎	4	503	KARSAN	40
✎	5	504	KARSAN	60



EDIT	EXPEDITION_ID	DEPARTURE_POINT	DESTINATION	DEPARTURE_TIME	ARRIVAL_TIME
✎	52	Tokat	İzmir	7.15	8.45
✎	50	Ankara	İstanbul	14	15.05
✎	51	İzmir	Ankara	9	10.15
✎	53	İstanbul	Antalya	11	12
✎	54	İstanbul	Ankara	9.15	10.2



```
SELECT buses.id, customers.id
FROM buses
JOIN customers ON buses.id = customers.id;



SELECT seats.id, tickets.id, reservations.id
FROM seats
JOIN tickets ON seats.id = tickets.id
JOIN reservations ON tickets.id = reservations.id;



SELECT expeditions.id, campaigns.id
FROM expeditions
JOIN campaigns ON expeditions.id = campaigns.id;



SELECT CONCAT(name, ' ', surname) AS full_name, LENGTH(CONCAT(name, ' ', surname)) AS name_length
FROM customers;



SELECT CONCAT(reservation_date) AS ticket_info
FROM reservations;
```