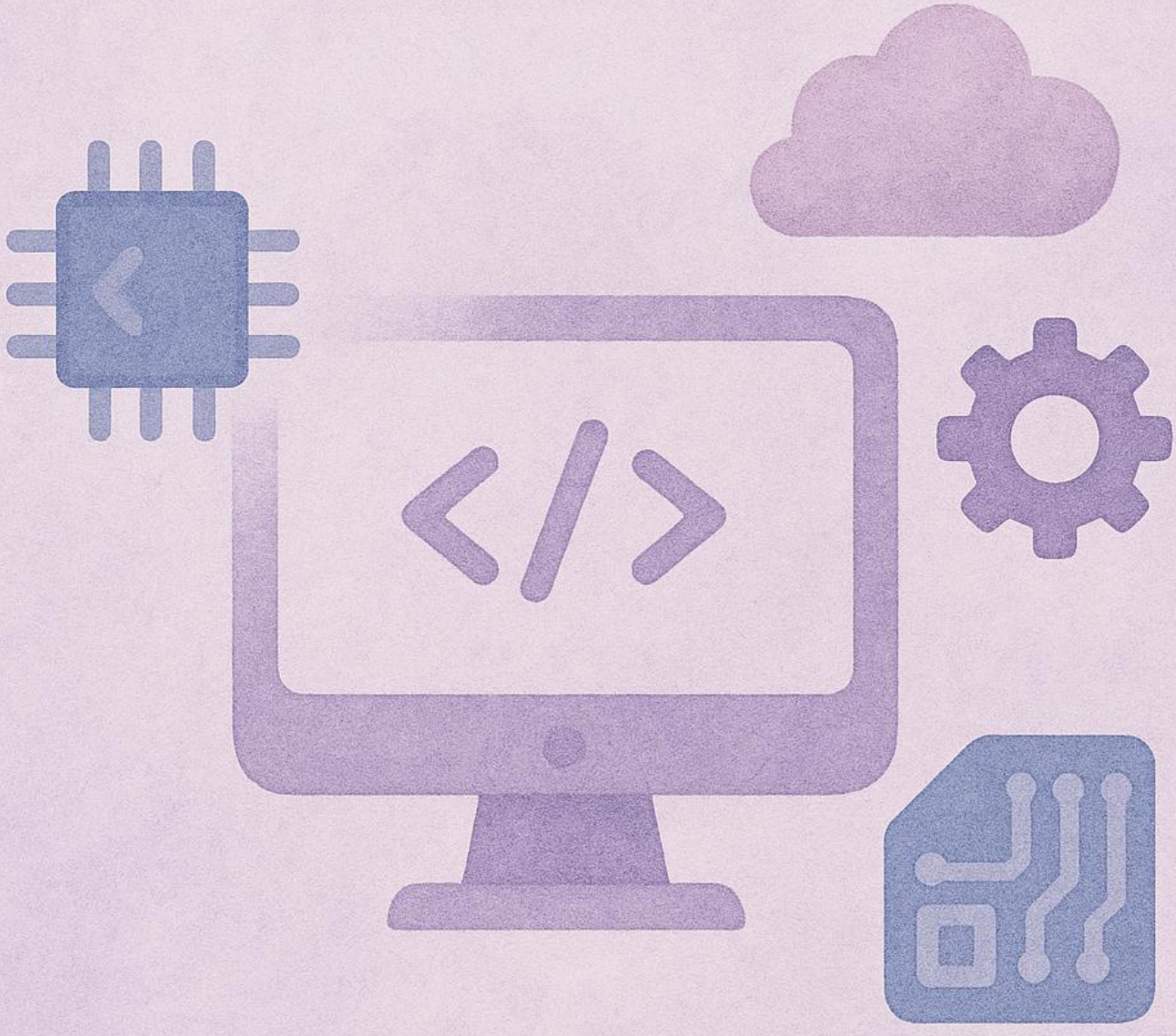
DATABASE PROJECT



BUS COMPANY

The database system to be designed for a bus company will track which tickets are sold by employees. It will record multiple departures occurring on the same line at different times throughout the day. Additionally, the system will monitor which vehicle and driver are assigned to each trip. When customers purchase tickets online, they are required to pay by credit card, whereas branch ticket purchases can be made using either cash or credit card.

ENTITIES

Entity -1	Entity-2	Entity-3	Entity-4	Entity-5	Entity-6	Entity-7	Entity-8	Entity-9
Trip	Vehicle	Branch	Payment	Line	Driver	Ticket	Employee	Customer

Entitiy Name	Attribute Name
TICKET	Ticket ID
	Seat Number
	Price
	Date
	Trip ID
	Customer ID
	Payment ID

Entitiy Name	Attribute Name
EMPLOYEE	Employee ID
	Employee ID
	First Name
	Last Name

Entitiy Name	Attribute Name
VEHICLE	Vehicle ID
	Plate Number
	Model
	Brand

Entitiy Name	Attribute Name
TRP	Trip ID
	Departure Time
	Arrival Time
	Departure location
	Arrival location
	Vehicle ID
	Driver ID

Entitiy Name	Attribute Name
CUSTOMER	Customer ID
	First Name
	Last Name
	Phone Number
	Email

Entitiy Name	Attribute Name
PAYMENT	Payment ID
	Amount
	Customer ID

Entitiy Name	Attribute Name
BRANCH	Branch ID
	Branch Name
	Branch Location

Entitiy Name	Attribute Name
DRIVER	DRIVER ID
	First Name
	Last Name
	Vehicle ID
	Trip ID

Entitiy Name	Attribute Name
LINE	Line ID
	Branch ID
	Number

	TICKET	EMPLOYEE	VEHICLE	TRIP	PAYMENT	CUSTOMER	BRANCH	DRIVER	LINE
TICKET	—	Sold by (1:M)		Issued for(M:1)	Has (1:1)	Purchased by(1:M)			
EMPLOYEE	Sales (1:M)	—				Serve (1:M)			
VEHICLE			—	Make (1:M)				Driven by(M:1)	
TRIP	Has (1:M)		Uses (1:M)	—					Belongs to(M:1)
PAYMENT	Belong to(1:1)				—	Done by (M:1)	Done in (1:M)		
CUSTOMER	Buy (1:M)	Served by (1:M)			Have (1:M)	—			
BRANCH					Get (1:M)		—	Employs (1:M)	
DRIVER			Drive (1:M)				Works at (M:1)	—	Assigned To(M:1)
LINE				Has (1:M)				Have (1:M)	—

ERD_MAP

CUSTOMER		
PK	*	Customer ID
	*	First Name
	*	Last Name
	*	Phone Number
	*	E Mail

PAYMENT		
PK	*	Payment ID
	*	Amount
FK	*	Customer ID
	*	Card Number
	*	Payment Type
FK	*	Other ID

BRANCH		
PK	*	Branch ID
	*	Branch Name
	*	Location

EMPLOYEE		
PK	*	Employee ID
	*	First Name
	*	Last Name

ERD_MAP

TICKET		
PK	*	Ticket ID
	*	Seat Number
	*	Price
	*	Date
FK	*	Trip ID
FK	*	Customer ID
FK	*	Payment ID
FK	*	Employee ID

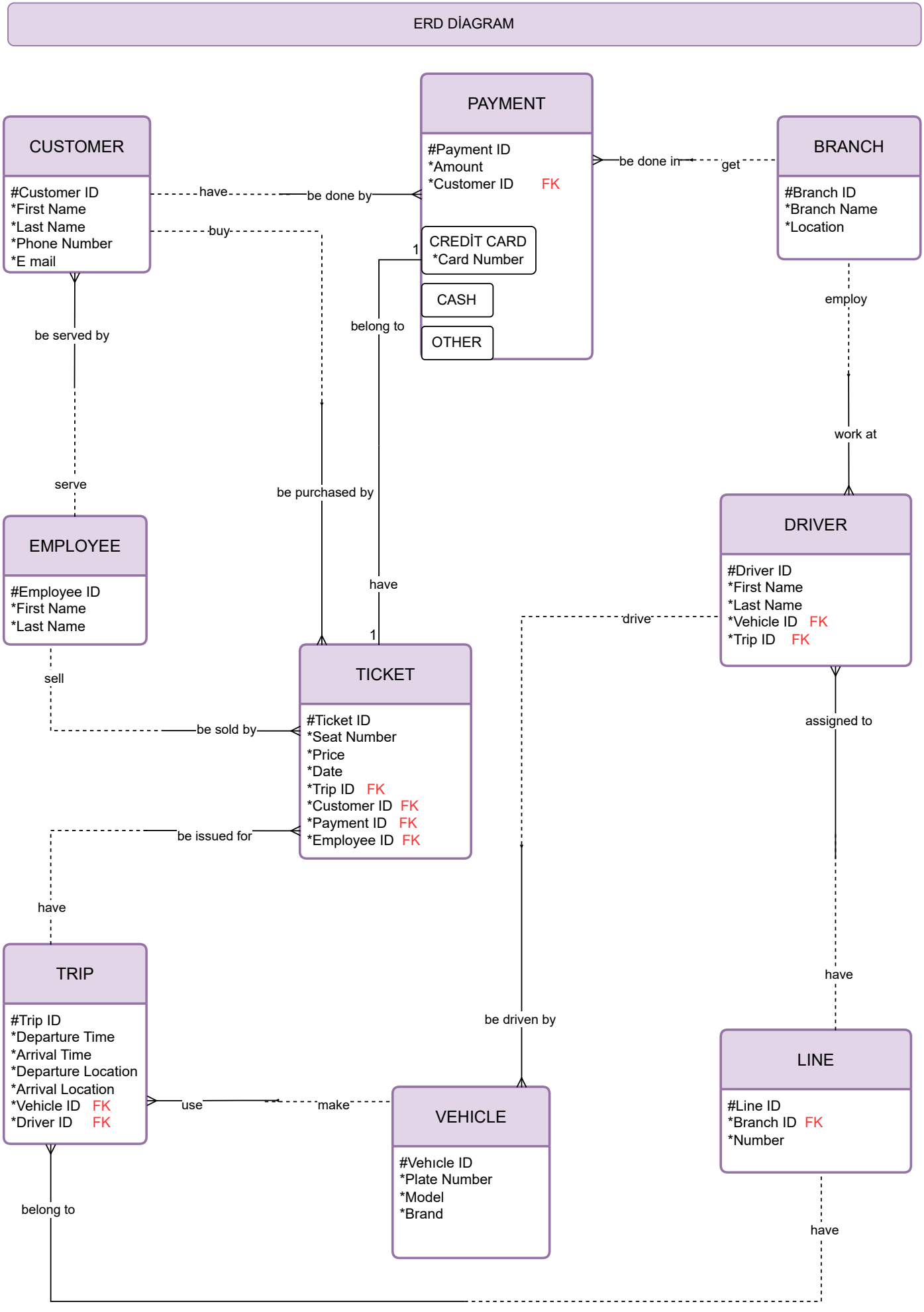
VEHICLE		
PK	*	Vehicle ID
	*	Plate Number
	*	Model
	*	Brand

LINE		
PK	*	Line ID
FK	*	Branch ID
	*	Number

ERD_MAP

TRIP		
PK	*	Trip ID
	*	Departure Time
	*	Arrival Time
	*	Departure Location
	*	Arrival Location
FK	*	Vehicle ID
FK	*	Driver ID

DRIVER		
PK	*	Driver ID
	*	First Name
	*	Last Name
FK	*	Vehicle ID
FK	*	Trip ID


ERD DİAGRAM
PAYMENT
#Payment ID
*Amount
*Customer ID FK
CUSTOMER
#Customer ID
*First Name
*Last Name
*Phone Number
*E mail
BRANCH
#Branch ID
*Branch Name
*Location
CREDİT CARD
*Card Number
CASH
OTHER
have
be done by
be done in
get
buy
1
belong to
be served by
employ
work at
serve
be purchased by
EMPLOYEE
#Employee ID
*First Name
*Last Name
DRIVER
#Driver ID
*First Name
*Last Name
*Vehicle ID FK
*Trip ID FK
have
1
drive
sell
TICKET
#Ticket ID
*Seat Number
*Price
*Date
*Trip ID FK
*Customer ID FK
*Payment ID FK
*Employee ID FK
be sold by
assigned to
be issued for
have
have
TRIP
#Trip ID
*Departure Time
*Arrival Time
*Departure Location
*Arrival Location
*Vehicle ID FK
*Driver ID FK
be driven by
LINE
#Line ID
*Branch ID FK
*Number
use
make
VEHICLE
#Vehicle ID
*Plate Number
*Model
*Brand
belong to
have

TICKET → EMPLOYEE

- Each ticket **must be sold by one and only one** employee
- Each employee **may sell one or more** tickets

TICKET → TRIP

- Each ticket **must be issued for one and only one** trip
- Each trip **may have one or more** tickets

TICKET → PAYMENT

- Each ticket **must have one and only one** payment
- Each payment **must belong to one and only one** ticket

TICKET → CUSTOMER

- Each ticket **must be purchased by one and only one** customer
- Each customer **may buy one or more** tickets

VEHICLE → TRIP

- Each vehicle **may make one or more** trips
- Each trip **must use one and only one** vehicle

VEHICLE → DRIVER

- Each vehicle **must be driven by one and only one** driver
- Each driver **may drive one or more** vehicles

TRIP → LINE

- Each trip **must belong to one and only one** line
- Each line **may have one or more** trips

PAYMENT → CUSTOMER

- Each payment **must be done by one and only one** customer
- Each customer **may have one or more** payments

PAYMENT → BRANCH

- Each payment **must be done in one and only one** branch
- Each branch **may get one or more** payments

BRANCH → DRIVER

- Each branch **may employ one or more** drivers
- Each driver **must work at one and only one** branch

DRIVER → LINE

- Each **driver** must be **assigned to one and only one** line
- Each **line** may have **one or more** drivers

CUSTOMER → EMPLOYEE

- **Each employee may serve one or more customers.**
- **Each customer must be served by one and only one employee.**

SQL STATEMENTS

CREATİNG TABLES

```
CREATE TABLE TICKET(
Ticket_ID INT PRIMARY KEY,
SeatNumber INT NOT NULL,
Trip_ID VARCHAR(50) NOT NULL,
Price DECIMAL(10,2),
TicketDate DATE NOT NULL,
Customer_ID VARCHAR(50) NOT NULL,
Payment_ID VARCHAR(50) NOT NULL,
Employee_ID VARCHAR(50) NOT NULL,
FOREIGN KEY (Trip_ID) REFERENCES TRIP(Trip_ID),
FOREIGN KEY (Customer_ID) REFERENCES Customers(Customer_ID),
FOREIGN KEY (Payment_ID) REFERENCES PAYMENT(Payment_ID),
FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(Employee_ID)
);
```

```
CREATE TABLE PAYMENT(
Payment_ID INT PRIMARY KEY,
Amount DECIMAL(10,2),
Customer_ID INT NOT NULL,
Card_Number VARCHAR(50) NOT NULL,
Payment_type VARCHAR(50) NOT NULL,
FOREIGN KEY (Customer_ID) REFERENCES Customers(Customer_ID)
);
CREATE TABLE EMPLOYEE(
Employee_ID INT PRIMARY KEY,
First_Name VARCHAR(255) NOT NULL,
Last_Name VARCHAR(255) NOT NULL
);
CREATE TABLE Customers (
Customer_ID INT PRIMARY KEY,
First_Name VARCHAR(255) NOT NULL,
Last_Name VARCHAR(255) NOT NULL,
Email VARCHAR(255),
PhoneNumber VARCHAR(20)
);
```

```
CREATE TABLE DRIVER (
Driver_ID INT PRIMARY KEY,
First_Name VARCHAR(255) NOT NULL,
Last_Name VARCHAR(255) NOT NULL,
Vehicle_ID VARCHAR(50) NOT NULL,
Trip_ID INT NOT NULL,
FOREIGN KEY (Vehicle_ID) REFERENCES VEHICLE(Vehicle_ID),
FOREIGN KEY (Trip_ID) REFERENCES TRIP(Trip_ID)
);
CREATE TABLE BRANCH (
Branch_ID INT PRIMARY KEY,
Branch_Name VARCHAR(100) NOT NULL,
Location_1 VARCHAR(100) NOT NULL
);
CREATE TABLE VEHICLE (
Vehicle_ID INT PRIMARY KEY,
Plate_Number VARCHAR(20) NOT NULL,
Model VARCHAR(50),
Brand VARCHAR(50)
);
```

```
CREATE TABLE TRIP (
Trip_ID INT PRIMARY KEY,
Driver_ID INT NOT NULL,
Departure_Time DATETIME,
Arrival_Time DATETIME,
Departure_Location VARCHAR(100) NOT NULL,
Arrival_Location VARCHAR(100) NOT NULL,
Vehicle_ID INT NOT NULL,
FOREIGN KEY (Driver_ID) REFERENCES DRIVER(Driver_ID),
FOREIGN KEY (Vehicle_ID) REFERENCES VEHICLE(Vehicle_ID)
);
CREATE TABLE LINE_num (
Line_ID INT PRIMARY KEY,
Branch_ID INT NOT NULL,
Line_Number VARCHAR(10),
FOREIGN KEY (Branch_ID) REFERENCES BRANCH(Branch_ID)
);
```

CUSTOMERS

```
INSERT INTO Customers (Customer_ID, First_Name, Last_Name, PhoneNumber, Email) VALUES
(1,
'Alice'
,
'musk'
,
'123-456-7890'
,
'alice@example.com');
INSERT INTO Customers (Customer_ID, First_Name, Last_Name, PhoneNumber, Email) VALUES
(2,
'Bob'
,
'brown'
,
'234-567-8901'
,
'bob@example.com');
INSERT INTO Customers (Customer_ID, First_Name, Last_Name, PhoneNumber, Email) VALUES
(3,
'Carol'
,
'john'
,
'345-678-9012'
,
'carol@example.com');
```

TICKET

```
INSERT INTO TICKET (Ticket_ID, SeatNumber, Trip_ID, Price,TicketDate,Customer_ID,Payment_ID,Employee_ID)
VALUES
(1, 1,
'123456789'
,120.00,TO_DATE('2024-05-01'
,
'YYYY-MM-DD'),
'Cust 1'
,
'Pay 1'
,
'Emp 1');
INSERT INTO TICKET (Ticket_ID, SeatNumber, Trip_ID, Price,TicketDate,Customer_ID,Payment_ID,Employee_ID)
VALUES
(2, 2,
'987654321'
,170.00,TO_DATE('2024-05-01'
,
'YYYY-MM-DD'),
'Cust 2'
,
'Pay 2'
,
'Emp 2');
INSERT INTO TICKET (Ticket_ID, SeatNumber, Trip_ID, Price,TicketDate,Customer_ID,Payment_ID,Employee_ID)
VALUES
(3, 3,
'456789123'
,250.00,TO_DATE('2024-05-01'
,
'YYYY-MM-DD'),
'Cust 3'
,
'Pay 3'
,
'Emp 3');
```

PAYMENT

```
INSERT INTO PAYMENT (Payment_ID, Amount, Customer_ID,Card_Number,Payment_type) VALUES
(1, 120.00, 1,
'2346234897'
,
'Cash');
INSERT INTO PAYMENT (Payment_ID, Amount, Customer_ID,Card_Number,Payment_type) VALUES
(2, 120.00, 2,
'7452398456'
,
'Cash');
INSERT INTO PAYMENT (Payment_ID, Amount, Customer_ID,Card_Number,Payment_type) VALUES
(3, 120.00, 3,
'8342865111'
,Credit Card');
```

EMPLOYEE

```
INSERT INTO EMPLOYEE (Employee_ID, First_Name,Last_Name) VALUES
(1,
'Emily'
,
'Johnson');
INSERT INTO EMPLOYEE (Employee_ID, First_Name,Last_Name) VALUES
(2,
'Liam'
,
'ahmad');
INSERT INTO EMPLOYEE (Employee_ID, First_Name,Last_Name) VALUES
(3,
'Olivia'
,
'salah');
```

DRIVER

```
INSERT INTO DRIVER (Driver_ID, First_Name,Last_Name, Vehicle_ID,Trip_ID) VALUES
(1,
'james'
,
'mcCghill'
,
'2234'
,1);
INSERT INTO DRIVER (Driver_ID, First_Name,Last_Name, Vehicle_ID,Trip_ID) VALUES
(2,
'kai'
,
'kevin'
,
'6243'
,2);
INSERT INTO DRIVER (Driver_ID, First_Name,Last_Name, Vehicle_ID,Trip_ID) VALUES
(3,
'jarad'
,
'wikins'
,
'7789'
,3);
```

BRANCH

```
INSERT INTO BRANCH (Branch_ID, Branch_Name, Location_1) VALUES
(1,
'Spring yard'
,
'Austin');
INSERT INTO BRANCH (Branch_ID, Branch_Name, Location_1) VALUES
(2,
'Holiday holy'
,
'Denver');
INSERT INTO BRANCH (Branch_ID, Branch_Name, Location_1) VALUES
(3,
'Black Friday'
,
'Seattle');
```

TRIP

```
INSERT INTO TRIP (Trip_ID, Driver_ID, Departure_Time, Arrival_Time, Departure_Location, Arrival_Location, Vehicle_ID)
VALUES
(1, 1,
'HH:MM:SS'
,
'HH:MM:SS'
,
'123 Apple St'
,
'769 green St' 1);
INSERT INTO TRIP (Trip_ID, Driver_ID, Departure_Time, Arrival_Time, Departure_Location, Arrival_Location, Vehicle_ID)
VALUES
(2, 2,
'HH:MM:SS'
,
'HH:MM:SS'
,
'456 Berry Blvd'
,
'123 new star '
, 2);
INSERT INTO TRIP (Trip_ID, Driver_ID, Departure_Time, Arrival_Time, Departure_Location, Arrival_Location, Vehicle_ID)
VALUES
(3, 3,
'HH:MM:SS'
,
'HH:MM:SS'
,
'789 Cherry Cir'
,
'877 blue Cir'
, 3);
```

VEHICLE

```
INSERT INTO VEHICLE (Vehicle_ID, Plate_Number, Model, Brand) VALUES
(1,
'2431'
,
'Camry'
,
'Toyota');
INSERT INTO VEHICLE (Vehicle_ID, Plate_Number, Model, Brand) VALUES
(2,
'7287'
,
'SUV'
,
'Ford Explorer');
INSERT INTO VEHICLE (Vehicle_ID, Plate_Number, Model, Brand) VALUES
(3,
'9932'
,
'Coupe'
,
'audi');
```

LINE

```
INSERT INTO LINE_num (Line_ID, Branch_ID, Line_Number) VALUES
(1, 1,
'1');
INSERT INTO LINE_num (Line_ID, Branch_ID, Line_Number) VALUES
(2, 2,
'2');
INSERT INTO LINE_num (Line_ID, Branch_ID, Line_Number) VALUES
(3,3,
'3');
INSERT INTO LINE_num (Line_ID, Branch_ID, Line_Number) VALUES
(4, 4,
'4');
INSERT INTO LINE_num (Line_ID, Branch_ID, Line_Number) VALUES
(5, 5,
'5');
INSERT INTO LINE_num (Line_ID, Branch_ID, Line_Number) VALUES
(6, 6,
'6');
```

DML STATEMENTS

SUBQUERY

```
SELECT First_Name, Email
FROM CUSTOMER
WHERE Customer_ID IN (
SELECT Customer_ID
FROM PAYMENT
WHERE Amount > 100
);
```

JOIN

```
SELECT C.First_Name, T.Ticket_ID, V.Model, T.Seat_Number
FROM CUSTOMER C
JOIN TICKET T ON C.Customer_ID = T.Customer_ID
JOIN TRIP TR ON T.Trip_ID = TR.Trip_ID
JOIN VEHICLE V ON TR.Vehicle_ID = V.Vehicle_ID
WHERE T.Date > '2025-01-01';
```

GROUP BY

```
SELECT TR.Vehicle_ID, COUNT(*) AS TotalTrips
FROM TRIP TR
GROUP BY TR.Vehicle_ID;
```

DATE FUNCTİON

```
SELECT Trip_ID, DATE_FORMAT(Departure_Time,
'%Y-%m-%d') AS DepartureDate
FROM TRIP;
```

CHARACTER FUNCTİON

```
SELECT UPPER(First_Name) AS UpperName, Email
FROM CUSTOMER;
```

UPDATE

```
UPDATE CUSTOMER
SET Phone_Number = '123-456-7890'
WHERE Customer_ID = 1;
```

ALTER TABLE

```
ALTER TABLE CUSTOMER
ADD COLUMN Address VARCHAR(255);
```

Ebru İlhan 2210206061

Samin Fatihzade 2210206519

OTHMAN MOHAMMAD URAS 2210206546