DATABASE MANAGEMENT FINAL PROJECT

Subject: Database Implementation and Application Development

Group Number: 23

Student Name / Numbers:

- Selen Çağla Kurtay / 05180000022
- Egemen Aydın / 05170000078
- Hasan Oral / 05170000073
- Mehmet Akif Konur / 05170000077

DBMS: MySQL in XAMPP and MySQL WorkBench

1. CREATE DATABASE AND MAPPINGS

A) DATABASE

```
CREATE TABLE Airport (
    Airport_code char(3) NOT NULL,
    Name varchar(255),
    City varchar(255),
    State varchar(255),
    PRIMARY KEY (Airport_code)
);

CREATE TABLE Flight (
    Flight_number char(5) NOT NULL,
    Airline_id int NOT NULL,
    Weekdays varchar(255) CHECK (Weekdays IN ('Pazartesi','Salı','Çarşamba','Perşembe','Cuma',
    'Cumartesi','Pazar')),
    PRIMARY KEY (Flight_number)
);

CREATE TABLE Flight_leg (
        Flight_Number char(5) NOT NULL,
        Leg_Number int NOT NULL,
        Mileage int,
        Departure_airport_code char(3) NOT NULL,
        Scheduled_departure_time TIME,
        Arrival_airport_code char(3) NOT NULL,
        Scheduled_arrival_time TIME,
        PRIMARY KEY (Leg_number,Flight_number)

);

CREATE TABLE Leg_instance (
        Flight_Number char(5) NOT NULL,
        Leg_Number int NOT NULL,
        Date DATE NOT NULL,
        Number_of_available_seats int,
        Airplane_id int NOT NULL,
        Departure_airport_code char(3) NOT NULL,
        Departure_time TIME,
        Arrival_airport_code char(3) NOT NULL,
```

```
        Arrival_time TIME CHECK (Arrival_time<>Departure_time), -- en uzun uçuş süresi 17 saat 25 dakika olduğu için bu şekilde yaptık.
    PRIMARY KEY (Date,Leg_number,Flight_number)

);

CREATE TABLE Fare (
    Flight_Number char(5) NOT NULL,
    Fare_code char(1) NOT NULL,
    Amount INT,
    Restrictions varchar(255),
    PRIMARY KEY (Flight_Number,Fare_code)
);


CREATE TABLE Airplane_type (
    Airplane_type_name varchar(255) NOT NULL,
        Max_seats int,
        Manufacturer_company varchar(255),
    PRIMARY KEY (Airplane_type_name)
);

CREATE TABLE Can_land (
    Airplane_type_name varchar(255) NOT NULL,
        Airport_code char(3) NOT NULL,
    PRIMARY KEY (Airplane_type_name,Airport_code)
);

CREATE TABLE Airplane (
    Property_id int NOT NULL,
        Total_number_of_seats int CHECK (Total_number_of_seats > 0),
        Airplane_type varchar(255) NOT NULL,
    PRIMARY KEY (Property_id)
);


CREATE TABLE Seat(
    Seat_Number int NOT NULL,
    Max_luggage int CHECK (Max_luggage < 35),
    Passaport_no int NOT NULL,
```

```
    Flight_Number char(5) NOT NULL,
            Leg_Number int NOT NULL,
            Date DATE,
    PRIMARY KEY(Seat_Number,Flight_number,Leg_number,Date)

);

CREATE TABLE Customer(
    Passaport_no int NOT NULL CHECK (Passaport_no>99999 AND Passaport_no < 999999),
            Address varchar(255),
            Country varchar(255),
            E_mail varchar(255),
            Phone int,
    PRIMARY KEY (Passaport_no)
);

CREATE TABLE Airline(
    Property_id int NOT NULL,
    PRIMARY KEY (Property_id)
);

CREATE TABLE Property(
    Property_id int NOT NULL,
    Owner_id int NOT NULL,
    PRIMARY KEY (Property_id)
);

CREATE TABLE Company(
    Company_id int NOT NULL AUTO_INCREMENT,
    PRIMARY KEY (Company_id)
);

CREATE TABLE FFC(
    Passaport_no int NOT NULL,
    Flight_number char(5) NOT NULL,
    Leg_number int NOT NULL,
    Seat_Number int NOT NULL,
    Date DATE NOT NULL,
    PRIMARY KEY(Flight_Number,Seat_Number,Passaport_no,Leg_Number,Date)
);
```

```
ALTER TABLE Airplane
ADD FOREIGN KEY (Airplane_type) REFERENCES Airplane_type(Airplane_type_name);

ALTER TABLE Flight
ADD FOREIGN KEY (Airline_id) REFERENCES Airline(Property_id);

ALTER TABLE Fare
ADD FOREIGN KEY (Flight_number) REFERENCES Flight(Flight_number);

ALTER TABLE Flight_leg
ADD FOREIGN KEY (Flight_number) REFERENCES Flight(Flight_number),
ADD FOREIGN KEY (Departure_airport_code) REFERENCES Airport(Airport_code),
ADD FOREIGN KEY (Arrival_airport_code) REFERENCES Airport(Airport_code);

ALTER TABLE Leg_instance
ADD FOREIGN KEY (Flight_number) REFERENCES Flight(Flight_number),
ADD FOREIGN KEY (Leg_number) REFERENCES Flight_leg(Leg_number),
ADD FOREIGN KEY (Airplane_id) REFERENCES Airplane(Property_id),
ADD FOREIGN KEY (Departure_airport_code) REFERENCES Airport(Airport_code),
ADD FOREIGN KEY (Arrival_airport_code) REFERENCES Airport(Airport_code);

ALTER TABLE Seat
ADD FOREIGN KEY (Flight_number) REFERENCES Flight(Flight_number),
ADD FOREIGN KEY (Passaport_no) REFERENCES Customer(Passaport_no),
ADD FOREIGN KEY (Leg_number) REFERENCES Flight_leg(Leg_number),
ADD FOREIGN KEY (Date) REFERENCES Leg_instance(Date);

ALTER TABLE Can_land
ADD FOREIGN KEY (Airplane_type_name) REFERENCES Airplane_type(Airplane_type_name),
ADD FOREIGN KEY (Airport_code) REFERENCES Airport(Airport_code);

ALTER TABLE Property
ADD FOREIGN KEY (Owner_id) REFERENCES Company(Company_id);

ALTER TABLE FFC
ADD FOREIGN KEY (Flight_Number) REFERENCES Flight_leg(Flight_Number),
ADD FOREIGN KEY (Leg_Number) REFERENCES Flight_leg(Leg_Number),
ADD FOREIGN KEY (Passaport_no) REFERENCES Customer(Passaport_no),
ADD FOREIGN KEY (Date) REFERENCES Seat(Date),
```

ADD FOREIGN KEY (Seat_Number) REFERENCES Seat(Seat_Number);

B) MAPPING

-FARE is weak entity according to FLIGHT. So FARE gets FLIGHT's primary key as foreign key which is Flight_number.

-FLIGHT_LEG is weak entity according to FLIGHT. So FLIGHT_LEG gets FLIGHT's primary key as foreign key which is Flight_number.

-The relationship between LEG_INSTANCE and FLIGHT_LEG; LEG_INSTANCE gets primary key of FLIGHT_LEG as foreign key, which are Flight_number and Leg_number, because LEG_INSTANCE is weak entity.

-The relationship between LEG_INSTANCE and FLIGHT via FLIGHT LEG; LEG_INSTANCE gets primary key of FLIGHT as foreign key, which are Flight_number, because LEG_INSTANCE is weak entity.

-There is a 1:M relationship between FFC and FLIGHT_LEG, so FFC gets primary key of FLIGHT_LEG as foreign key, which are Flight_number and Leg_number.

- There is a 1:1 relationship between FFC and SEAT, so FFC gets primary key of SEAT as foreign key, which are Date and Seat_number.

- There is a 1:M relationship between FFC and CUSTOMER, so FFC gets primary key of CUSTOMER as foreign key, which is Passaport_no.

- There are two 1:M relationships between FLIGHT_LEG and AIRPORT, so FLIGHT_LEG gets primary key of AIRPORT as foreign key, which are Departure_airport_code,(via DEPARTURE_AIRPORT relation), Scheduled_departure_time,(via DEPARTURE_AIRPORT

relation), Arrival_airport_code,(via ARRIVAL_AIRPORT relation). Also FLIGHT_LEG gets attributes of relations between AIRPORT and FLIGHT_LEG, which are Scheduled_departure_time,(via DEPARTURE_AIRPORT relation) and Scheduled_arrival_time. (via ARRIVAL_AIRPORT relation).

- There are two 1:M relationships between LEG_INSTANCE and AIRPORT, so LEG_INSTANCE gets primary key of AIRPORT as foreign key, which are Departure_airport_code(via DEPARTS relation) , Arrival_airport_code(via ARRIVES relation). Also LEG_INSTANCE gets attributes of relations between AIRPORT and LEG_INSTANCE , which are Arrival_time(via ARRIVES relation) and Departure_time(via DEPARTS relation).

-The relationship between SEAT and LEG_INSTANCE, so SEAT gets primary key of LEG_INSTANCE as foreign key, which is Date because SEAT entity is weak.

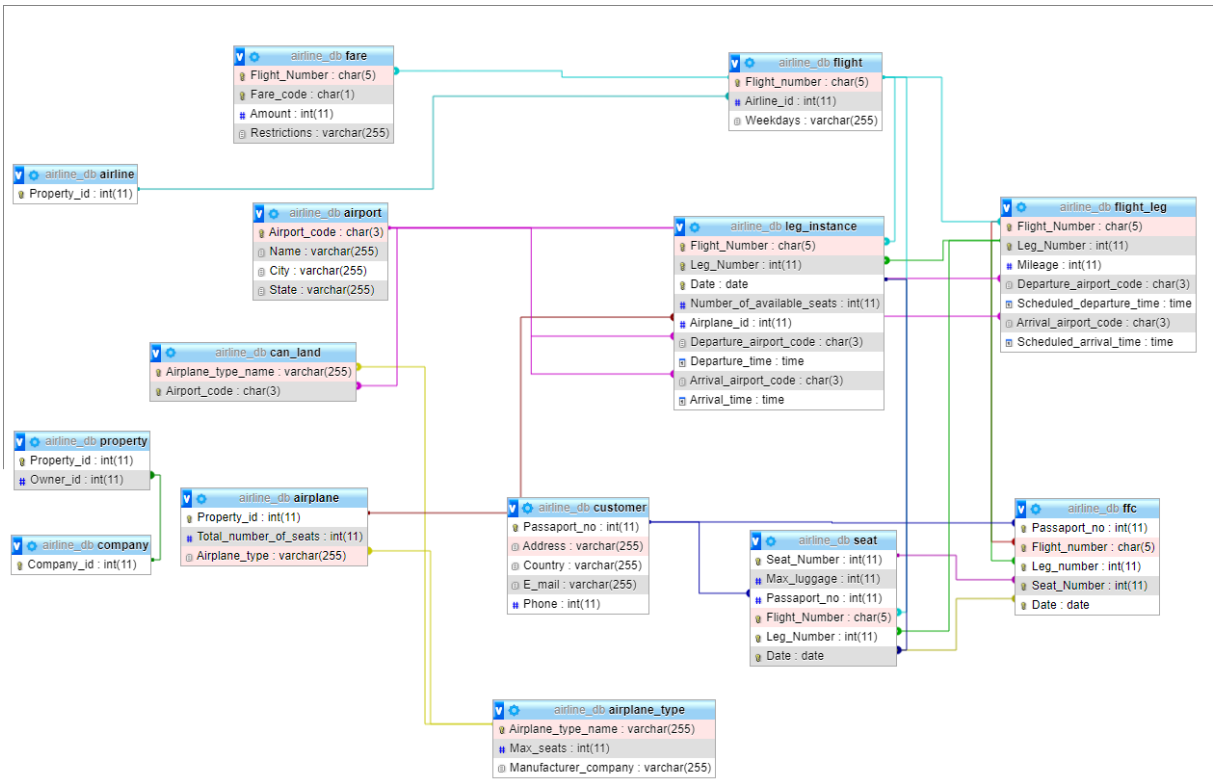
-The relationship between SEAT and FLIGHT_LEG via LEG_INSTANCE, so SEAT gets primary key of FLIGHT_LEG as foreign key, which is Leg_number because SEAT entity is weak.

-The relationship between SEAT and FLIGHT via FLIGHT_LEG, so SEAT gets primary key of FLIGHT as foreign key, which is Flight_number because SEAT entity is weak.

-There is 1:M relationship between AIRPLANE and AIRPLANE_TYPE, so AIRPLANE gets primary key of AIRPLANE_TYPE as foreign key, which is Airplane_type_name.

-There is 1:1 “assigned” relationship between LEG_INSTANCE and AIRPLANE. We represent this relation with an attribute called “Airplane_id” on LEG_INSTANCE.

-AIRLINE and AIRPLANE has generalization-specialization with PROPERTY. So both AIRPLANE and AIRLINE get PROPERTY’s primary key as foreign key, which is Property_id.



2. Inserted Tuple Examples (2 for each table)

❖ **AIRPORT**

INSERT INTO Airport VALUES ('SBA','Sabiha Gökçen Airport','İstanbul','Marmara');

INSERT INTO Airport VALUES ('ADB','Adnan Menderes Airport','İzmir','Ege');

❖ **COMPANY**

INSERT INTO Company VALUES (); (Auto incremented)

❖ **PROPERTY**

INSERT INTO Property VALUES ('0001','001');

INSERT INTO Property VALUES ('0002','001');

❖ **AIRPLANE_TYPE**

INSERT INTO Airplane_type VALUES ('Airbus A220','130','Airbus Comp.');

INSERT INTO Airplane_type VALUES ('Boeing 767','181','Boeing Comp.');

❖ **AIRPLANE**

INSERT INTO Airplane VALUES ('001','130','Airbus A220');

INSERT INTO Airplane VALUES ('002','132','Airbus A318');

- **AIRLINE**

INSERT INTO Airline VALUES ('0011');

INSERT INTO Airline VALUES ('0012');

- **FLIGHT**

INSERT INTO Flight VALUES ('D753S','0011','Pazartesi');

INSERT INTO Flight VALUES ('BR65L','0012','Pazartesi');

- **FARE**

INSERT INTO Fare VALUES ('D753S','J','999','required to wear masks');

INSERT INTO Fare VALUES ('BR65L','F','1299','must complete the passenger information form');

- **CAN_LAND**

INSERT INTO Can_land VALUES ('Airbus A220','SBA');

INSERT INTO Can_land VALUES ('Airbus A220','ADB');

- **CUSTOMER**

INSERT INTO Customer VALUES ('878954','Urla/İzmir','Turkey','hasan7635.ho@gmail.com','541563488');

INSERT INTO Customer VALUES ('789456','Bornova/İzmir','Turkey','egemenbursali16@gmail.com','541866779');

- **FLIGHT_LEG**

INSERT INTO flight_leg VALUES('D753S','1','451','SAT','14:30','BLA','16:30');

INSERT INTO flight_leg VALUES('NR73A','1','773','ERA','15:30','AYT','19:30');

- **LEG_INSTANCE**

INSERT INTO Leg_instance VALUES ('D753S','1','2021-1-25','130','005','SAT','14:30','BLA','16:30');

INSERT INTO Leg_instance VALUES ('I34B7','1','2020-10-10','100','002','SBA','22:30','BLA','23:00');

❖ **SEAT**

INSERT INTO Seat VALUES ('001','20','878954','D753S','1','2021-1-25');

INSERT INTO Seat VALUES ('002','20','789456','D753S','1','2021-1-25');

Seat_Number	Max_luggage	Passaport_no	Flight_Number	Leg_Number	Date
1	20	741258	BR65L	2	2020-11-15
1	20	878954	D753S	1	2021-01-25
2	20	963258	BR65L	2	2020-11-15
2	20	789456	D753S	1	2021-01-25
2	20	878954	ER88Z	1	2020-08-02
10	20	789456	ER88Z	1	2020-08-02

'Seat' Table Example.

❖ **FFC**

INSERT INTO ffc VALUES ('878954','D753S','1','1','2021-1-25');

INSERT INTO ffc VALUES ('878954','ER88Z','1','2','2020-8-2');

3. Triggers

1-

```
DELIMITER //
CREATE TRIGGER mileage_control
BEFORE INSERT
ON flight_leg
FOR EACH ROW
IF NEW.mileage < 0 THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Mileage has exceeded the allowed amount of lower than 0.';
END IF//
DELIMITER ;
```

This trigger prevents mileage amount to be lower than 0.

```
INSERT INTO flight_leg VALUES('NR73A','3','-400','AYT','20:00','BLA','22:30');
```

-Here mileage amount is -400.

Error Code: 1644. Mileage has exceeded the allowed amount of lower than 0.

-So the program gives warning.

2-

```
DELIMITER $$
CREATE TRIGGER TRG_decrease_available_seats
AFTER INSERT ON seat

FOR EACH ROW
BEGIN
   UPDATE  LEG_INSTANCE
   SET     Number_of_available_seats = Number_of_available_seats - 1
   WHERE LEG_INSTANCE.Flight_Number= NEW.Flight_Number
   AND LEG_INSTANCE.Leg_Number=NEW.Leg_Number
   AND LEG_INSTANCE.Date = NEW.Date;
END $$
DELIMITER ;
```

This trigger deletes available seat number whenever new tuple inserted on seat table.

```
INSERT INTO Seat VALUES ('083','33','963258','D753S','1','2021-1-25');
```

Flight_Number	Leg_Number	Date	Number_of_available_seats
D753S	1	2021-01-25	127

Number of available seat decrease 1 as you can see the table.

3-

```
DELIMITER //
CREATE TRIGGER max_seat_control
BEFORE INSERT
ON airplane_type
FOR EACH ROW
IF NEW.Max_seats<0 THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Max_seat has exceeded the allowed amount of lower than 0.';
END IF//
```

DELIMITER ;

This trigger controls if ‘max_seats’ value is lower than 0

```
INSERT INTO Airplane_type VALUES ('Airbus B220','-65','Airbus Comp.');
```

-Here max_seat amount is -65.

Error Code: 1644. Max_seat has exceeded the allowed amount of lower than 0.

-So the program gives warning.

4-

```
DELIMITER //
CREATE TRIGGER amount_control
BEFORE INSERT
ON fare
FOR EACH ROW
IF NEW.amount<0 THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Sale has exceeded the allowed amount of lower than 0.';
END IF//
DELIMITER ;
```

This trigger checks if sale amount is lower than 0.

```
INSERT INTO Fare VALUES ('W39S5','J','-6','required to wear masks');
```

-Here fare amount is -6

Error Code: 1644. Sale has exceeded the allowed amount of lower than 0.

-So the program gives warning.

5-

```
DELIMITER //
CREATE TRIGGER leg_instance_time
BEFORE INSERT
ON leg_instance
FOR EACH ROW
IF NEW.Arrival_time < NEW.Departure_time THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Arrival time cannot be before departure time!';
END IF//
DELIMITER ;
```

This trigger checks if arrival time is before than departure time.

```
INSERT INTO Leg_instance VALUES ('BL10B','2','2020-11-15','5','007','BLA','23:00','AYT','22:30')
```

-Here arrival time is before than departure's.

```
Error Code: 1644. Arrival time cannot be before departure time!
```

-So the program gives warning.

4- CHECK CONSTRAINTS

CHECK constraints are used to limit the value range that can be placed in a column. We used 5 different check constraints to limit some columns.

- We used a check constraint for Weekdays attribute in FLIGHT table. This constraint checks if the value given for Weekdays attribute is in list given English characters.

- We used a check constraint for Arrival_time attribute in LEG_INSTANCE table. This constraint checks whether the value given for Arrival_time is not equal to Departure_time

value. We used hour as TIME variable, after searching we learned that longest flight lasted 17 hours and 25 minutes. That's why we didn't check if the trip lasts exactly 1 day.

- We used a check constraint for Total_number_of_seats attribute in AIRPLANE table. This constraint checks if the value given for Total_number_of_attribute value is greater than zero.

- We used a check constraint for Passaport_no attribute in CUSTOMER table. This constraint checks if the value given for Passaport_no attribute is 6 digits long.

- We used a check constraint for Max_luggage attribute in SEAT table. This constraint checks if the value given for Max_luggage attribute is greater than 35.

5- SQL Statements

A) Insert, Update, Delete Statements

I. INSERT INTO customer VALUES ('378520','Uncubozköy/Manisa','Uganda','willetteclem@gmail.com','550992348');

II. INSERT INTO fare VALUES ('I34B7','Y','350','required to wear masks');

III. INSERT INTO flight VALUES ('YT652','20','Pazartesi');

IV. INSERT INTO flight_leg VALUES ('NR73A','3','447','BDM','15:45','SAT','17:15');

V. INSERT INTO airplane_type VALUES ('MİG 21','200','MİG Comp.');

Passaport_no	Address	Country	E_mail	Phone
378520	Uncubozköy/Manisa	Uganda	willetteclem@gmail.com	550992348

I. UPDATE CUSTOMER

SET Phone=5332789218

WHERE customer.E_mai l= "cafeege35@gmail.com";

II. UPDATE FARE

SET Amount=Amount*1.20

WHERE Flight_Number='AB60K';

DELETE FROM fare WHERE fare.Flight_Number='W39S5';

III. UPDATE FLIGHT

SET Weekdays="Çarşamba"

WHERE Flight_number='YT652';

IV. UPDATE FLIGHT_LEG

SET Scheduled_departure_time=DATE_ADD(Scheduled_departure_time,INTERVAL 60 MINUTE)

, Scheduled_arrival_time=DATE_ADD(Scheduled_arrival_time,INTERVAL 60 MINUTE)

WHERE Flight_Number= 'NR73A' AND Leg_Number=3;

V. UPDATE AIRPLANE_TYPE

SET Max_seats=70

WHERE Airplane_type_name="Douglas DC-2";

Airplane_type_name	Max_seats	Manufacturer_company
Douglas DC-2	70	Douglas Aircraft Comp.

I. DELETE FROM customer WHERE customer.Country='Uganda' AND customer.Passaport_no='378520';

II. DELETE FROM fare WHERE fare.Flight_Number='W39S5';

III. DELETE FROM flight WHERE flight.Flight_number='YT652';

IV. DELETE FROM flight_leg WHERE flight_leg.Flight_Number='NR73A' AND flight_leg.Leg_Number=3 ;

V. DELETE FROM Airplane_type WHERE airplane_type.Manufacturer_company='MİG Comp.';

```
SELECT * FROM customer WHERE country = "Uganda";
```

Result Grid | Filter Rows: | Edit:

Passaport_no	Address	Country	E_mail	Phone
NULL	NULL	NULL	NULL	NULL

B) Select Statements

1)Using minimum 2 Tables

1-

SELECT flight.Flight_number AS Uçuş, AVG(fare.Amount) AS Ortalamaücret

FROM flight, fare

WHERE

flight.Flight_number = fare.Flight_Number

GROUP BY flight.Flight_number

This SELECT shows average amount of flights.

Uçuş	Ortalamaücret
AB60K	359.0000
BL10B	789.0000
BR65L	1049.5000
D753S	794.0000
ER88Z	665.0000

2-

SELECT flight.Flight_number, COUNT(*) as Leg_sayisi

FROM flight, flight_leg

WHERE flight.Flight_number = flight_leg.Flight_Number

GROUP BY Flight_number

ORDER BY COUNT(*) DESC;

This SELECT shows number of legs each flight has.

3-

```
SELECT can_land.Airport_code, airplane_type.Max_seats
FROM can_land, airplane_type
WHERE can_land.Airplane_type_name = airplane_type.Airplane_type_name
ORDER BY Max_seats DESC
```

This SELECT shows airports and the plane with the maximum number of seats that can land at that airport.

2)Using minimum 3 Tables

1-

```
SELECT customer.E_mail
FROM customer, seat, leg_instance
WHERE
seat.Passaport_no = customer.Passaport_no
AND seat.Date = leg_instance.Date
AND seat.Leg_Number = leg_instance.Leg_Number
AND seat.Flight_Number = leg_instance.Flight_Number
AND leg_instance.Flight_Number = 'I34B7'
AND leg_instance.Leg_Number = '1'
AND leg_instance.Date = '2020-10-10';
```

This SELECT shows e-mails of customers from a specific flight.

2-

```
SELECT flight.Flight_number AS uçuş, flight_leg.Mileage AS Mesafe, airport.Name AS Kalkış
FROM flight, flight_leg, airport
WHERE
flight.Flight_number = flight_leg.Flight_Number
AND flight_leg.Departure_airport_code = airport.Airport_code
AND flight_leg.Mileage > '428';
```

This SELECT shows departure airports which have flights where the distance is more than 428 miles.

3-

SELECT airplane_type.airplane_type_name AS Uçak_Tipi, COUNT(*) AS İniş_yapılabilen_havalimanı

FROM airport, can_land, airplane_type

WHERE

airplane_type.airplane_type_name=can_land.Airplane_type_name

AND can_land.Airport_code=airport.Airport_code

GROUP BY airplane_type.Airplane_type_name

ORDER BY COUNT(*) DESC;

This SELECT shows how many airports each airplane type can land.

Uçak_Tipi	İniş_yapılabilen_havalimanı
Airbus A220	8
Airbus A318	6
Boeing 717	6
Boeing 767	5
Douglas DC-2	1

4-

SELECT company.Company_id AS Şirket, airplane.Airplane_type AS Uçaklar

FROM company, property, airplane

WHERE

company.Company_id = property.Owner_id

AND property.Property_id = airplane.Property_id;

This SELECT shows all the planes that companies have.

3)Using minimum 4 Tables

1-

SELECT flight_leg.Flight_Number AS Uçuş, fare.Amount AS Ücretler, airport.Name AS İniş_noktası

FROM airport, flight_leg, flight, fare

WHERE

flight_leg.Flight_Number = flight.Flight_number

AND flight.Flight_number = fare.Flight_Number

AND fare.Amount > 300

```
AND flight_leg.Arrival_airport_code = airport.Airport_code
AND airport.Name <> 'Bursa Airport';
```

This SELECT shows flights with fare amount more than 200 and will not land in Bursa city.

Uçuş	Ücretler	İniş_noktası
HX3K5	1300	Adnan Menderes Airport
HX3K5	899	Adnan Menderes Airport
MS57T	455	Antalya Airport
NR73A	350	Antalya Airport
BR65L	1299	Antalya Airport
BR65L	800	Antalya Airport
BR65L	1299	Balıkesir Airport
BR65L	800	Balıkesir Airport
D753S	999	Balıkesir Airport
D753S	589	Balıkesir Airport
I34B7	1400	Balıkesir Airport
NR73A	350	Balıkesir Airport
ER88Z	665	Sabiha Gökçen Airport

2-

```
SELECT company.Company_id AS Şirket, flight.Flight_number AS Uçuş
FROM flight, property, airline,company
WHERE
flight.Airline_id = airline.Property_id
AND airline.Property_id = property.Property_id
AND property.Property_id = property.Owner_id
AND property.Owner_id = company.Company_id
AND flight.Weekdays = 'Pazartesi';
```

This SELECT shows flights’ companies which have flight in Mondays.

3-

```
SELECT customer.Passaport_no AS Müşteri, fare.Restrictions AS Uyulması_gereken_kısıtlama
FROM customer, seat, leg_instance, flight_leg, flight, fare
WHERE
customer.Passaport_no = seat.Passaport_no
AND seat.Date = leg_instance.Date
AND seat.Leg_Number = leg_instance.Leg_Number
AND seat.Flight_Number = leg_instance.Flight_Number
AND flight_leg.Leg_Number = leg_instance.Leg_Number
AND flight_leg.Flight_Number = leg_instance.Flight_Number
```

```
AND flight_leg.Flight_Number = flight.Flight_number
AND fare.Flight_Number = flight.Flight_number
 ORDER BY customer.Passaport_no;
```

This SELECT shows the rules that customers should pay attention (Ordered by passport number).

C) NESTED SELECT QUERIES

1)

```
SELECT Flight_Number, City, Date
FROM leg_instance, airport
WHERE
leg_instance.Arrival_airport_code = airport.Airport_code
AND airport.City='Bursa'
AND leg_instance.Flight_Number IN (
   SELECT leg_instance.Flight_Number
   FROM airplane, leg_instance
   WHERE
   leg_instance.Airplane_id = airplane.Property_id
   AND airplane.Airplane_type = 'Boeing 717'
   );
```

This statement above shows flights made with Boeing 717 type and landed in one of the airports in Bursa.

```
-- NESTED Bursa'da bulunan havalimanlarından birine 'Boeing 717' ile uçan uçuşlar
SELECT Flight_Number, City, Date FROM leg_instance, airport
WHERE
leg_instance.Arrival_airport_code = airport.Airport_code
AND airport.City='Bursa'
AND leg_instance.Flight_Number IN (
    SELECT leg_instance.Flight_Number
    FROM airplane, leg_instance
    WHERE
    leg_instance.Airplane_id = airplane.Property_id
    AND airplane.Airplane_type = 'Boeing 717'
    )
```

Temizle | Biçim | Otomatik kaydedilmiş sorguyu al

☐ Parametreleri bağla

Bu SQL sorgusunu işaretle:

[Sınırlayıcı ;] ☐ Bu sorguyu burada tekrar göster ☐ Sorgu kutusunu tut ☐ Tamamlandığında geri döndür ☑ Dış anahtar denetlemelerini etkinleştir Git

Sorgu kutusunu gizle

Gösterilen satır 0 - 1 (toplam 2, Sorgu 0,0052 saniye sürdü.)

```
-- NESTED Bursa'da bulunan havalimanlarından birine 'Boeing 717' ile uçan uçuşlar SELECT Flight_Number, City, Date FROM leg_instance, airport WHERE
leg_instance.Arrival_airport_code = airport.Airport_code AND airport.City='Bursa' AND leg_instance.Flight_Number IN ( SELECT leg_instance.Flight_Number FROM airplane,
leg_instance WHERE leg_instance.Airplane_id = airplane.Property_id AND airplane.Airplane_type = 'Boeing 717' )
```

Flight_Number	City	Date
BL10B	Bursa	2020-04-23
BL10B	Bursa	2020-05-07

2)

SELECT Country, COUNT(*) as Customers

FROM customer

WHERE Passaport_no IN

(SELECT Passaport_no FROM seat WHERE Date>='2020-08-01' AND Date<='2020-08-31')

GROUP BY Country;

This statement above shows customers who flew during in August 2020.

3)

SELECT MIN(Amount) FROM fare WHERE Flight_Number IN

(SELECT Flight_number FROM flight WHERE Weekdays = "Salı");

This statement above shows the minimum fare amount from flights which happened in Tuesdays.

4)

SELECT City FROM airport WHERE Airport_code IN

```
(SELECT Departure_airport_code
 FROM flight_leg
 WHERE Scheduled_departure_time <= ("10:00"));
```

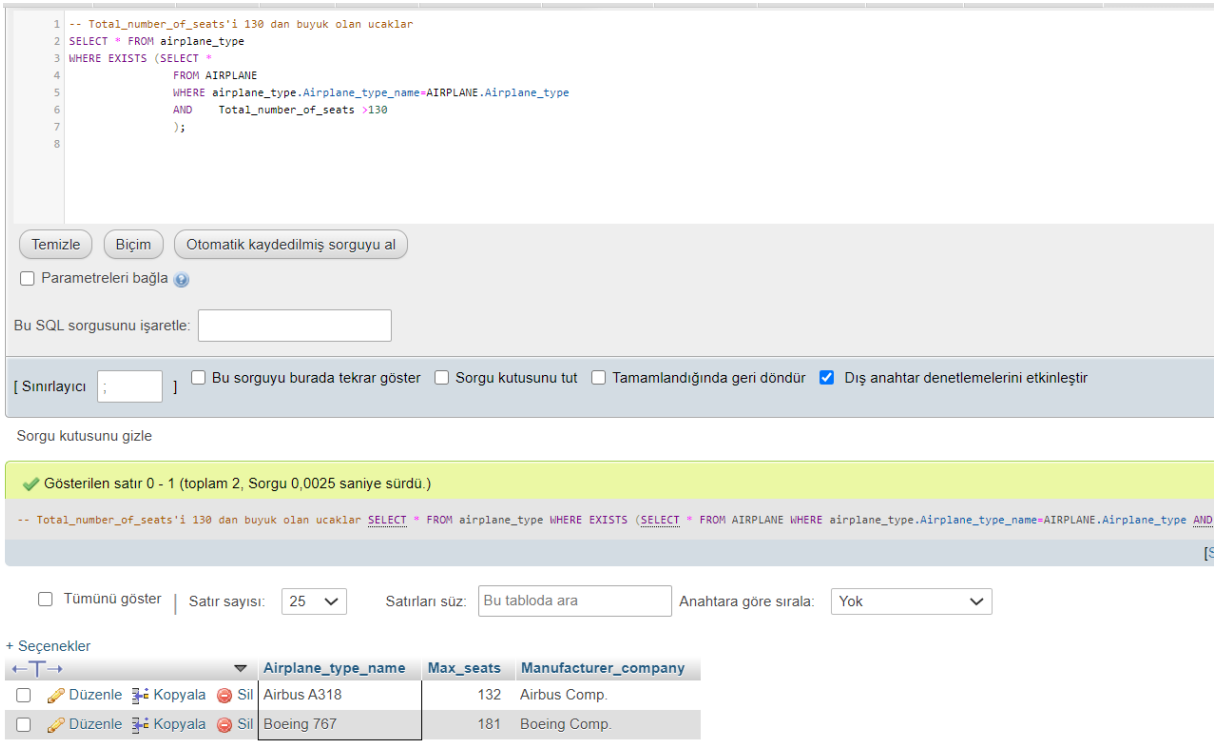
This statement above shows which cities have flights before 10:00 a.m

D) EXISTS and NOT EXISTS Statements

1-

```
SELECT *
FROM airplane_type
WHERE EXISTS (SELECT *
        FROM AIRPLANE
        WHERE airplane_type.Airplane_type_name=AIRPLANE.Airplane_type
        AND    Total_number_of_seats >130
        );
```

The statement above shows airplane types which have more than 130 seats.



2-

```
SELECT *
FROM flight_leg
WHERE NOT EXISTS (SELECT *
        FROM leg_instance
        WHERE leg_instance.Arrival_time > '00:00' AND leg_instance.Arrival_time < '07:00'
        AND flight_leg.Scheduled_arrival_time = leg_instance.Arrival_time);
```

The statement above shows flights' flight leg information where arrival time is not during night time (Between 00:00 and 07:00).

3-

```
SELECT *
FROM flight_leg
WHERE NOT EXISTS (SELECT *
        FROM leg_instance
```

```
WHERE flight_leg.Leg_Number=flight_leg.Leg_Number
AND Mileage >= 200
);
```

The statement above shows flights' flight leg information where mileage is less than 200.

E) JOIN Statements

1)Left Join

```
SELECT DISTINCT * FROM customer
LEFT JOIN seat ON customer.Passaport_no= seat.Passaport_no
WHERE Phone LIKE ('541%');
```

Customers whose phone numbers start with 541.

2)Right Join

```
SELECT  * from can_land
RIGHT JOIN airplane_type ON airplane_type.Airplane_type_name =
can_land.Airplane_type_name;
```

A join that shows airplane types which produced but do not have permission to land any airport.

3)Full-outer Join

```
-- We use UNION with Left and Right Joins because of MySql.
SELECT leg_instance.Flight_Number, airplane.Total_number_of_seats,
leg_instance.Number_of_available_seats FROM leg_instance
LEFT JOIN airplane ON leg_instance.Airplane_id = airplane.Property_id
UNION
SELECT leg_instance.Flight_Number, airplane.Total_number_of_seats,
leg_instance.Number_of_available_seats FROM leg_instance
RIGHT JOIN airplane ON leg_instance.Airplane_id = airplane.Property_id
WHERE (Arrival_time) >= ('15:00')
```

Number of total seats and available seats from flights which their arrival time is before 3 p.m.

```
-- telefon numarası 541 ile başlayan müşterilerin seat tablosu
SELECT DISTINCT * FROM customer
LEFT JOIN seat ON customer.Passaport_no= seat.Passaport_no
WHERE Phone LIKE ('541%');
```

Temizle | Biçim | Otomatik kaydedilmiş sorguyu al

☐ Parametreleri bağla

Bu SQL sorgusunu işaretle:

[Sınırlayıcı ;] ☐ Bu sorguyu burada tekrar göster ☐ Sorgu kutusunu tut ☐ Tamamlandığında geri döndür ☑ Dış anahtar denetlemelerini etkinleştir

Sorgu kutusunu gizle

⚠ Şu anki seçim benzersiz bir sütun içermiyor. Kılavuz düzenleme, onay kutusu, Düzenle, Kopyala ve Sil özellikleri kullanılabilir değil.

✔ Gösterilen satır 0 - 17 (toplam 18, Sorgu 0,0054 saniye sürdü.)

```
-- telefon numarası 541 ile başlayan müşterilerin seat tablosu SELECT DISTINCT * FROM customer LEFT JOIN seat ON customer.Passaport_no= seat.Passaport_no WHERE Phone LIKE ('541%')
```

☐ Tümünü göster | Satır sayısı: 25 Satırları süz: Bu tabloda ara Anahtara göre sırala: Yok

+ Seçenekler

Passaport_no	Address	Country	E_mail	Phone	Seat_Number	Max_luggage	Passaport_no	Flight_Number	Leg_Number	Date
741258	Muratpaşa/Antalya	Turkey	selen07@gmail.com	541866539	1	20	741258	BR65L	2	2020-11-15
878954	Urla/İzmir	Turkey	hasan7635.ho@gmail.com	541563488	1	20	878954	D753S	1	2021-01-25
963258	London	United Kingdom	crispin704@gmail.com	541866449	2	20	963258	BR65L	2	2020-11-15
789456	Bornova/İzmir	Turkey	egemenbursali16@gmail.com	541866779	2	20	789456	D753S	1	2021-01-25
878954	Urla/İzmir	Turkey	hasan7635.ho@gmail.com	541563488	2	20	878954	ER88Z	1	2020-08-02
789456	Bornova/İzmir	Turkey	egemenbursali16@gmail.com	541866779	10	20	789456	ER88Z	1	2020-08-02
456852	Bandırma/Balıkesir	Turkey	makifkonur@gmail.com	541786489	12	20	456852	I34B7	1	2020-10-10
456852	Bandırma/Balıkesir	Turkey	makifkonur@gmail.com	541786489	14	20	456852	ER88Z	1	2020-08-02
456852	Bandırma/Balıkesir	Turkey	makifkonur@gmail.com	541786489	15	20	456852	BR65L	1	2020-07-15
741258	Muratpaşa/Antalya	Turkey	selen07@gmail.com	541866539	20	20	741258	ER88Z	1	2020-08-02
789456	Bornova/İzmir	Turkey	egemenbursali16@gmail.com	541866779	22	20	789456	BR65L	1	2020-07-15
878954	Urla/İzmir	Turkey	hasan7635.ho@gmail.com	541563488	23	20	878954	BR65L	1	2020-07-15
963258	London	United Kingdom	crispin704@gmail.com	541866449	34	20	963258	BR65L	1	2020-07-15
741258	Muratpaşa/Antalya	Turkey	selen07@gmail.com	541866539	35	20	741258	BR65L	1	2020-07-15
741258	Muratpaşa/Antalya	Turkey	selen07@gmail.com	541866539	45	20	741258	I34B7	1	2020-10-10
456852	Bandırma/Balıkesir	Turkey	makifkonur@gmail.com	541786489	65	20	456852	D753S	1	2021-01-25
963258	London	United Kingdom	crispin704@gmail.com	541866449	73	20	963258	D753S	1	2021-01-25
963258	London	United Kingdom	crispin704@gmail.com	541866449	84	20	963258	D753S	1	2021-02-18

6) VIEWS

1-

```
CREATE VIEW flight_leg_inis_bilgileri AS

SELECT flight_leg.Leg_Number, flight_leg.Flight_Number, airport.Name AS Havalimanı,
flight_leg.Scheduled_arrival_time AS Tarih

FROM flight_leg, airport

WHERE flight_leg.Arrival_airport_code = airport.Airport_code;
```

A view that shows landing information of flights.

2-

```
CREATE VIEW amount_Size AS

    SELECT Flight_Number, Fare_code, Amount

    FROM fare

    WHERE Amount > (
```

```
        SELECT  AVG(Amount)
        FROM fare)
    ORDER BY Amount DESC;
```

A view that shows ticket amounts which are more expensive than average amount.

3-

```
CREATE VIEW VW_50percent_empty
AS
SELECT Flight_Number, Leg_Number, Date, Number_of_available_seats, Total_number_of_seats,
CAST(Number_of_available_seats * 100 as float) / CAST(Total_number_of_seats as float) as Rate
FROM LEG_INSTANCE, AIRPLANE
WHERE leg_instance.Arrival_time IS NOT NULL
AND leg_instance.Airplane_id = airplane.Property_id
AND (CAST(Number_of_available_seats * 100 as float) / CAST(Total_number_of_seats as float) ) >
50;
```

A view that shows completed flights which their seats are less than 50% full.

4-

```
CREATE VIEW AvailableSeats
AS SELECT leg_instance.Flight_Number, leg_instance.Number_of_available_seats,
leg_instance.Departure_airport_code, airport.City
FROM LEG_INSTANCE JOIN AIRPORT ON
leg_instance.Departure_airport_code=airport.Airport_code;
```

A view that shows available seats from current flights.

5-

```
CREATE VIEW marmaradaki_havalimanları AS
    SELECT Airport_code, Name
    FROM airport
    WHERE  (airport.State = 'Marmara');
```

A view that shows airports which are in Marmara region.

PART-III

CUSTOMER SEGMENTATION

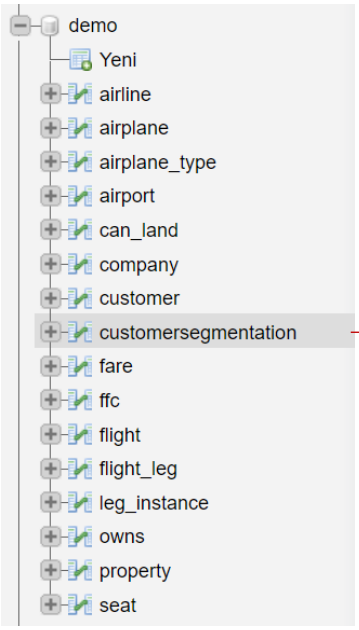
Customer segmentation is the process of dividing customers into groups based on common characteristics so companies can market to each group effectively and appropriately. So we created a new python project and we connected it with our database by “mysql.connector “ library. By the python program, We calculated the mileage that related to the passaports where on the ffc table which is contains all the check-in that has made until today. For rewarding system, we decided two segmentation which are “gold class” and “silver class”. The passengers entering the 25% segment according to the mile ranking are considered 'Gold class' and the passengers in the 25% -50% range are accepted as 'Silver class'. For this reason, we created a table named customerSegmentation on the python application to keep the class information of the passengers and inserted it to this table if there is a customer included in any class. While using the database for a real system, we thought that during ticket purchasing, by using the customerSegmentation table, we could reward our customer with discounts or rewards such as gift miles can be made according to the class of the customer.

Python Code

```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="",
  database="demo"
)

gold_class_percentage=0.25
silver_class_percentage=0.5


# tüm customer tablosunu okuyup passaportları listeye attım
mycursor = mydb.cursor()
mycursor.execute("SELECT FFC.Passaport_no, SUM(flight_leg.Mileage) FROM
ffc,flight_leg WHERE ffc.Flight_number = flight_leg.Flight_Number AND
flight_leg.Leg_Number = ffc.Leg_number GROUP BY ffc.Passaport_no;")
myresult = mycursor.fetchall()

print(myresult)
print(len(myresult))

# liste sırala

mycursor.execute("CREATE TABLE IF NOT EXISTS
customerSegmentation(passaport_no int NOT NULL,class varchar(255),FOREIGN
KEY (passaport_no) REFERENCES customer(Passaport_no),PRIMARY KEY
(passaport_no));")
mycursor.execute("TRUNCATE customerSegmentation")

n = len(myresult)

# Traverse through all array elements
```

```
for i in range(n - 1):
  # range(n) also work but outer loop will repeat one time more than
needed.

  # Last i elements are already in place
  for j in range(0, n - i - 1):

    # traverse the array from 0 to n-i-1
    # Swap if the element found is greater
    # than the next element
    if myresult[j][1] > myresult[j + 1][1]:
      myresult[j], myresult[j + 1] = myresult[j + 1], myresult[j]

print(myresult)


sql = "INSERT INTO customerSegmentation VALUES ('{0}','{1}');"
maxCustomer = len(myresult)
for i in range(round(n*gold_class_percentage)):
  mycursor.execute("INSERT INTO customerSegmentation VALUES
('{0}','{1}');".format(myresult[maxCustomer-1][0],'GOLD'))
  maxCustomer-=1
  mydb.commit()

for i in range(round((n*silver_class_percentage)-
(n*gold_class_percentage))):
  mycursor.execute("INSERT INTO customerSegmentation VALUES
('{0}','{1}');".format(myresult[maxCustomer-1][0],'SILVER'))
  maxCustomer-=1
  mydb.commit()
```

<u>After Running Our Code:</u>



As you can see, after the running our code; our python project created a new table to our database.

Gösterilen satır 0 - 3 (toplam 4, Sorgu 0,0005 saniye sürdü.)

```
SELECT * FROM `customersegmentation`
```

Tümünü göster | Satır sayısı: 25 Satırları s

+ Seçenekler

				passaport_no	class
Düzenle	Kopyala	Sil		159753	SILVER
Düzenle	Kopyala	Sil		268459	GOLD
Düzenle	Kopyala	Sil		878954	GOLD
Düzenle	Kopyala	Sil		988655	SILVER

After our customerSegmentation table created, Python inserted to gold and silver customers to our database.

We can see, reserved seats and the connected flight on the seat table. It is assumed that each airplane has the number of seats available_seats, starting with number 1. If the desired seat from reserved airplane is not included in this table, it is available to be reserved. When the airplane takes off, the reserve seats connected in this table are deleted. Therefore, the seat table changes dynamically.

Customers' flights records must be kept in an order to access the flights made by the customers until now. For this purpose, the FFC table was created. It stores the data of each customers' check-in informations. If a reservation is made but flying process did not happen which means check-in not made, the information of this flight is not stored. This table does not hold Frequent-flyer customers. It is a table created to calculate the total mileage customers have traveled so far when you want to find frequent flyers. Frequent flyers are kept in the customerSegmentation table. If a customer is in this table, it is considered to be a Frequent-flyer customer. These customers are also divided into 2 within themselves, aiming to differ in rewarding. The customerSegmentation table is created by using ffc records with the python program.

POPULATING DATABASE

We used these INSERT queries for reasonable SELECT queries. Other parts of the Project's SQL codes all included in report.

```
 -- airport ****************************
INSERT INTO Airport VALUES ('SBA','Sabiha Gökçen Airport','İstanbul','Marmara');
INSERT INTO Airport VALUES ('ADB','Adnan Menderes Airport','İzmir','Ege');
INSERT INTO Airport VALUES ('AYT','Antalya Airport','Antalya','Akdeniz');
INSERT INTO Airport VALUES ('BLA','Balıkesir Airport','Balıkesir','Marmara');
INSERT INTO Airport VALUES ('APT','Atatürk Airport','Ankara','İç Anadolu');
INSERT INTO Airport VALUES ('SAT','Samsun Airport','Samsun','Karadeniz');
INSERT INTO Airport VALUES ('ERA','Erzurum Airport','Erzurum','Doğu Anadolu');
INSERT INTO Airport VALUES ('KPT','Kars Airport','Kars','Doğu Anadolu');
INSERT INTO Airport VALUES ('BDM','Bandırma Airport','Bandırma','Marmara');
INSERT INTO Airport VALUES ('BRS','Bursa Airport','Bursa','Marmara');
-- company ***************************
INSERT INTO Company VALUES ('');
INSERT INTO Company VALUES ('');
INSERT INTO Company VALUES ('');
INSERT INTO Company VALUES ('');
INSERT INTO Company VALUES ('');
-- property  ********************
INSERT INTO Property VALUES ('0001','001');
INSERT INTO Property VALUES ('0002','001');
INSERT INTO Property VALUES ('0003','001');
INSERT INTO Property VALUES ('0004','001');
INSERT INTO Property VALUES ('0005','001');
INSERT INTO Property VALUES ('0006','002');
INSERT INTO Property VALUES ('0007','002');
INSERT INTO Property VALUES ('0008','002');
INSERT INTO Property VALUES ('0009','003');
INSERT INTO Property VALUES ('0010','003');
INSERT INTO Property VALUES ('0011','003');
INSERT INTO Property VALUES ('0012','003');
INSERT INTO Property VALUES ('0013','003');
INSERT INTO Property VALUES ('0014','004');
INSERT INTO Property VALUES ('0015','004');
INSERT INTO Property VALUES ('0016','004');
INSERT INTO Property VALUES ('0017','004');
INSERT INTO Property VALUES ('0018','005');
```

```
INSERT INTO Property VALUES ('0019','005');
INSERT INTO Property VALUES ('0020','005');

-- airplane_type ********************************
INSERT INTO Airplane_type VALUES ('Airbus A220','130','Airbus Comp.');
INSERT INTO Airplane_type VALUES ('Airbus A318','132','Airbus Comp.');
INSERT INTO Airplane_type VALUES ('Boeing 767','181','Boeing Comp.');
INSERT INTO Airplane_type VALUES ('Boeing 717','106','Boeing Comp.');
INSERT INTO Airplane_type VALUES ('Douglas DC-2','14','Douglas Aircraft Comp.');
-- airplane ********************************
INSERT INTO Airplane VALUES ('001','130','Airbus A220');
INSERT INTO Airplane VALUES ('002','132','Airbus A318');
INSERT INTO Airplane VALUES ('003','132','Airbus A318');
INSERT INTO Airplane VALUES ('004','181','Boeing 767');
INSERT INTO Airplane VALUES ('005','181','Boeing 767');
INSERT INTO Airplane VALUES ('006','106','Boeing 717');
INSERT INTO Airplane VALUES ('007','106','Boeing 717');
INSERT INTO Airplane VALUES ('008','106','Boeing 717');
INSERT INTO Airplane VALUES ('009','106','Boeing 717');
INSERT INTO Airplane VALUES ('010','14','Douglas DC-2');
-- airline ***********************************
INSERT INTO Airline VALUES ('0011');
INSERT INTO Airline VALUES ('0012');
INSERT INTO Airline VALUES ('0013');
INSERT INTO Airline VALUES ('0014');
INSERT INTO Airline VALUES ('0015');
INSERT INTO Airline VALUES ('0016');
INSERT INTO Airline VALUES ('0017');
INSERT INTO Airline VALUES ('0018');
INSERT INTO Airline VALUES ('0019');
INSERT INTO Airline VALUES ('0020');
-- flight  ***********************************
INSERT INTO Flight VALUES ('D753S','0011','Pazartesi');
INSERT INTO Flight VALUES ('BR65L','0012','Pazartesi');
INSERT INTO Flight VALUES ('HX3K5','0013','Salı');
INSERT INTO Flight VALUES ('W39S5','0014','Pazar');
INSERT INTO Flight VALUES ('I34B7','0015','Perşembe');
INSERT INTO Flight VALUES ('BL10B','0016','Çarşamba');
INSERT INTO Flight VALUES ('ER88Z','0017','Cumartesi');
```

```
INSERT INTO Flight VALUES ('AB60K','0018','Cumartesi');
INSERT INTO Flight VALUES ('NR73A','0019','Salı');
INSERT INTO Flight VALUES ('MS57T','0020','Pazar');
-- fare ******************************
INSERT INTO Fare VALUES ('D753S','J','999','required to wear masks');
INSERT INTO Fare VALUES ('BR65L','F','1299','must complete the passenger information form');
INSERT INTO Fare VALUES ('HX3K5','J','899','must complete the passenger information form');
INSERT INTO Fare VALUES ('W39S5','Y','205','required to wear masks');
INSERT INTO Fare VALUES ('I34B7','F','1400','must complete the passenger information form');
INSERT INTO Fare VALUES ('BL10B','J','789','must complete the passenger information form');
INSERT INTO Fare VALUES ('ER88Z','J','665','required to wear masks');
INSERT INTO Fare VALUES ('AB60K','Y','299','must complete the passenger information form');
INSERT INTO Fare VALUES ('NR73A','Y','350','required to wear masks');
INSERT INTO Fare VALUES ('MS57T','W','455','must complete the passenger information form');
INSERT INTO Fare VALUES ('D753S','W','589','required to wear masks');
INSERT INTO Fare VALUES ('BR65L','J','800','must complete the passenger information form');
INSERT INTO Fare VALUES ('HX3K5','F','1300','must complete the passenger information form');

-- can_land ****************************
INSERT INTO Can_land VALUES ('Airbus A220','SBA');
INSERT INTO Can_land VALUES ('Airbus A220','ADB');
INSERT INTO Can_land VALUES ('Airbus A220','BLA');
INSERT INTO Can_land VALUES ('Airbus A220','SAT');
INSERT INTO Can_land VALUES ('Airbus A220','ERA');
INSERT INTO Can_land VALUES ('Airbus A220','KPT');
INSERT INTO Can_land VALUES ('Airbus A220','BDM');
INSERT INTO Can_land VALUES ('Airbus A220','BRS');
INSERT INTO Can_land VALUES ('Airbus A318','ADB');
INSERT INTO Can_land VALUES ('Airbus A318','AYT');
INSERT INTO Can_land VALUES ('Airbus A318','BLA');
INSERT INTO Can_land VALUES ('Airbus A318','APT');
INSERT INTO Can_land VALUES ('Airbus A318','ERA');
INSERT INTO Can_land VALUES ('Airbus A318','BDM');
INSERT INTO Can_land VALUES ('Boeing 767','SBA');
INSERT INTO Can_land VALUES ('Boeing 767','AYT');
INSERT INTO Can_land VALUES ('Boeing 767','APT');
INSERT INTO Can_land VALUES ('Boeing 767','ERA');
INSERT INTO Can_land VALUES ('Boeing 767','BDM');
INSERT INTO Can_land VALUES ('Boeing 717','SBA');
```

```
INSERT INTO Can_land VALUES ('Boeing 717','ADB');
INSERT INTO Can_land VALUES ('Boeing 717','AYT');
INSERT INTO Can_land VALUES ('Boeing 717','SAT');
INSERT INTO Can_land VALUES ('Boeing 717','ERA');
INSERT INTO Can_land VALUES ('Boeing 717','KPT');
INSERT INTO Can_land VALUES ('Douglas DC-2','BRS');

-- customer ****************************************
INSERT INTO Customer VALUES ('878954','Urla/İzmir','Turkey','hasan7635.ho@gmail.com','541563488');
INSERT INTO Customer VALUES ('789456','Bornova/İzmir','Turkey','egemenbursali16@gmail.com','541866779');
INSERT INTO Customer VALUES ('159753','Mudanya/Bursa','Turkey','databaseproject@gmail.com','543866489');
INSERT INTO Customer VALUES ('456852','Bandırma/Balıkesir','Turkey','makifkonur@gmail.com','541786489');
INSERT INTO Customer VALUES ('963258','London','United Kingdom','crispin704@gmail.com','541866449');
INSERT INTO Customer VALUES ('741258','Muratpaşa/Antalya','Turkey','selen07@gmail.com','541866539');
INSERT INTO Customer VALUES ('268459','Keçiören/Ankara','Turkey','keciorengucu06@gmail.com','544894889');
INSERT INTO Customer VALUES ('486213','South California','United States','wess798645@gmail.com','545664889');
INSERT INTO Customer VALUES ('988655','Kadıköy/İstanbul','Turkey','projectxxx@gmail.com','546912488');
INSERT INTO Customer VALUES ('145632','Çeşme/İzmir','Turkey','cafeege35@gmail.com','535664889');

-- flight_leg  ****************************************************
INSERT INTO flight_leg VALUES('D753S','1','451','SAT','14:30','BLA','16:30');
INSERT INTO flight_leg VALUES('NR73A','1','773','ERA','15:30','AYT','19:30');
INSERT INTO flight_leg VALUES('AB60K','1','201','BRS','08:30','APT','09:30');
INSERT INTO flight_leg VALUES('HX3K5','1','863','KPT','09:30','ADB','13:30');
INSERT INTO flight_leg VALUES('I34B7','1','120','SBA','22:30','BLA','23:00');
INSERT INTO flight_leg VALUES('BL10B','1','72','BLA','08:30','BRS','10:30');
INSERT INTO flight_leg VALUES('BR65L','1','72','BRS','12:00','BLA','13:00');
INSERT INTO flight_leg VALUES('W39S5','1','427','AYT','14:10','SAT','15:45');
INSERT INTO flight_leg VALUES('MS57T','1','427','SAT','23:55','AYT','01:30');
INSERT INTO flight_leg VALUES('ER88Z','1','651','ERA','07:00','SBA','13:00');
INSERT INTO flight_leg VALUES('NR73A','2','450','AYT','20:00','BLA','22:30');
INSERT INTO flight_leg VALUES('AB60K','2','200','APT','10:00','KPT','11:15');
INSERT INTO flight_leg VALUES('W39S5','2','120','SAT','20:00','BRS','22:30');
INSERT INTO flight_leg VALUES('BR65L','2','90','BLA','20:00','AYT','22:30');
-- leg_instance ******************************
INSERT INTO Leg_instance VALUES ('D753S','1','2021-1-25','130','005','SAT','14:30','BLA','16:30');
INSERT INTO Leg_instance VALUES ('I34B7','1','2020-10-10','100','002','SBA','22:30','BLA','23:00');
INSERT INTO Leg_instance VALUES ('ER88Z','1','2020-8-2','25','001','ERA','07:00','SBA','13:00');
INSERT INTO Leg_instance VALUES ('D753S','1','2021-2-18','10','003','SAT','14:30','BLA','16:30');
```

```
INSERT INTO Leg_instance VALUES ('BR65L','1','2020-7-15','5','007','BLA','12:00','AYT','13:00');
INSERT INTO Leg_instance VALUES ('NR73A','1','2020-6-20','90','008','ERA','15:30','AYT','19:30');
INSERT INTO Leg_instance VALUES ('MS57T','1','2020-3-26','8','0010','SAT','23:55','AYT','01:30');
INSERT INTO Leg_instance VALUES ('BL10B','1','2020-5-7','30','009','BLA','08:30','BRS','10:30');
INSERT INTO Leg_instance VALUES ('BL10B','1','2020-4-23','50','008','BLA','08:30','BRS','10:30');
INSERT INTO Leg_instance VALUES ('W39S5','1','2020-12-8','2','0010','AYT','14:10','SAT','15:45');
INSERT INTO Leg_instance VALUES ('BR65L','2','2020-11-15','5','007','BLA','20:00','AYT','22:30');
-- seat ****************
INSERT INTO Seat VALUES ('001','20','878954','D753S','1','2021-1-25');
INSERT INTO Seat VALUES ('002','20','789456','D753S','1','2021-1-25');
INSERT INTO Seat VALUES ('058','20','159753','D753S','1','2021-1-25');
INSERT INTO Seat VALUES ('065','20','456852','D753S','1','2021-1-25');
INSERT INTO Seat VALUES ('073','20','963258','D753S','1','2021-1-25');

INSERT INTO Seat VALUES ('011','20','159753','I34B7','1','2020-10-10');
INSERT INTO Seat VALUES ('012','20','456852','I34B7','1','2020-10-10');
INSERT INTO Seat VALUES ('045','20','741258','I34B7','1','2020-10-10');
INSERT INTO Seat VALUES ('046','20','268459','I34B7','1','2020-10-10');
INSERT INTO Seat VALUES ('047','20','486213','I34B7','1','2020-10-10');
INSERT INTO Seat VALUES ('048','20','988655','I34B7','1','2020-10-10');
INSERT INTO Seat VALUES ('050','20','145632','I34B7','1','2020-10-10');

INSERT INTO Seat VALUES ('002','20','878954','ER88Z','1','2020-8-2');
INSERT INTO Seat VALUES ('010','20','789456','ER88Z','1','2020-8-2');
INSERT INTO Seat VALUES ('014','20','456852','ER88Z','1','2020-8-2');
INSERT INTO Seat VALUES ('020','20','741258','ER88Z','1','2020-8-2');
INSERT INTO Seat VALUES ('025','20','268459','ER88Z','1','2020-8-2');
INSERT INTO Seat VALUES ('029','20','988655','ER88Z','1','2020-8-2');

INSERT INTO Seat VALUES ('045','20','988655','D753S','1','2021-2-18');
INSERT INTO Seat VALUES ('080','20','268459','D753S','1','2021-2-18');
INSERT INTO Seat VALUES ('084','20','963258','D753S','1','2021-2-18');
INSERT INTO Seat VALUES ('090','20','159753','D753S','1','2021-2-18');

INSERT INTO Seat VALUES ('015','20','456852','BR65L','1','2020-7-15');
INSERT INTO Seat VALUES ('016','20','159753','BR65L','1','2020-7-15');
INSERT INTO Seat VALUES ('022','20','789456','BR65L','1','2020-7-15');
INSERT INTO Seat VALUES ('023','20','878954','BR65L','1','2020-7-15');
INSERT INTO Seat VALUES ('034','20','963258','BR65L','1','2020-7-15');
```

```
INSERT INTO Seat VALUES ('035','20','741258','BR65L','1','2020-7-15');


INSERT INTO Seat VALUES ('001','20','741258','BR65L','2','2020-11-15');
INSERT INTO Seat VALUES ('002','20','963258','BR65L','2','2020-11-15');


-- yeni ffc için insertlerrr
-- ***********************************
INSERT INTO ffc VALUES ('878954','D753S','1','1','2021-1-25');
INSERT INTO ffc VALUES ('878954','ER88Z','1','2','2020-8-2');
INSERT INTO ffc VALUES ('159753','I34B7','1','11','2020-10-10');
INSERT INTO ffc VALUES ('789456','BR65L','1','22','2020-7-15');
INSERT INTO ffc VALUES ('268459','I34B7','1','46','2020-10-10');
INSERT INTO ffc VALUES ('988655','ER88Z','1','29','2020-08-02');
INSERT INTO ffc VALUES ('268459','ER88Z','1','25','2020-08-02');
INSERT INTO ffc VALUES ('159753','BR65L','1','16','2020-07-15');
INSERT INTO ffc VALUES ('741258','BR65L','1','35','2020-07-15');
INSERT INTO ffc VALUES ('486213','I34B7','1','47','2020-10-10');
```