DATABASE MANAGEMENT SYSTEMS FINAL PROJECT REPORT

HALİME BEYZA YILMAZ-05170000050
ATIQURAHMAN MAYAR-05130001350
EMRE DURSUN-05160000303
KAAN GÖKTÜRK-05160000253

CONTENTS

PROCEDURES BEFORE IMPLEMENTING OUR DATABASE MODEL:

INSERTING VALUES TO THE TABLES:
Just to get it right, we have assigned values close to the real world as much as possible. We have written all our data according to the demands in part 5 of final project.

1)Tables

COMPANY:
Company table is consisting of two type of companies. Air and Ba are the companies for AIRPLANES while TK, PGS, EMR and BAW are companies for our AIRLINE.

Company id	Company Name
air	Airbus Se
ba	The beoeing company
tk	Turkish airlines a.s
pgs	Pegasus a.s
emr	The emirates group
baw	International airlines group

AIRPORT:
Our airport table is consisting of 7 airports defined by their codes, names, cities and states.

Airport code	Name	City	State
adb	Izmir adnan meneres	Izmir	ege
Esb	esenboga	ankara	Ic anadolu
ist	Istanbul airport	istanbul	marmara
Saw	Istanbul airport	istanbul	marmara
jfk	John f kennedy	newyork	newyork
lhr	Heathrow	london	North england
dxb	Dubai airport	Dubai	Middle east

FLIGHT:
We have 8 flights. Flights are defined as flight number along with its weekdays and the Airlines that belongs to it.

Flight number	Weekdays	Airline
Tk2313	monday	Turkish airlines
Tk3	tuesday	Turkish airlines
Ba0113	tuesday	British airways
Ek121	friday	emirates
Pc3168	wednesday	pegasus
Ba0104	thursday	British airways
Ek202	Wednesday	emirates
Tk2163	friday	Turkish airlines

FLIGHT_LEG:
Our 8 flights make a total of 12 legs. We defined 3 types of flight legs in our table:
1- flights with 0 leg numbers. These flights are direct flights.
as example: TK2313, TK3, EK121, PC3168 and BA0104 are direct flights.

2- flights with 2 leg numbers. These flights are connecting flights making more than one flight.
as example: TK2163 and BA0113
3- flights with 3 leg numbers. These flights are connecting flights making more than 2 flights.EK202 is the only flight in our table having 3 connecting flights.

Flight_number	Leg_num	Dp_airport_code	Sch_departure_t	Scheduled_arrival_time	Arr_airport_code
Tk2163	1	esb	10:05	10:55	adb
Tk2163	2	adb	11:45	12:35	ist
Tk2313	0	ist	06:45	07:40	esb
Tk3	0	ist	07:15	10:20	dxb
Ba0113	1	lhr	08:20	12:20	ist
Ba0113	2	ist	14:10	00:00	jfk
Ek121	0	jfk	15:00	23:55	dxb
Pc3168	0	saw	09:00	09:50	esb
Ek202	1	dxb	06:45	09:40	esb
Ek202	2	esb	11:45	16:30	lhr
Ek202	3	lhr	21:00	03:10	jfk
Ba0104	0	saw	16:20	22:15	lhr

LEG_INSTANCE:
In leg instance the number of flights we defined are the same as flight_leg table. The only additions are airplane ids and the Date, where it holds the information about the accurate time of flight numbers along with the number of available seats.
flights number's departed and arrival airport can be changed due to many circumstances. For instance, BA0104 scheduled arrival airport code was lhr of London, but it lands to ESB of Ankara.

Also, Flight number's actual departure time and arrival time might be very different from their scheduled one. for instance:
the second leg of flight number TK2163 departure time changes due to the late arrival in the first leg.
BA0113, TK3, PC3168 and BA0104 departure and arrival times are also different from the scheduled one.

Flight_number	Leg_number	Date	Num_avail_seats	Airplane_id	Dep_airport_code	Dep_time	Arr_airport_code	Arr_time
Tk2163	1	01.01.2021	10	Jx-1232	esb	10:05	adb	11:20
Tk2163	2	01.01.2021	10	Un-3102	adb	12:30	ist	13:25
Ba0113	1	03.01.2021	220	Fk-221	lhr	08:20	ist	12:20
Ba0113	2	03.01.2021	220	Bre-3112	ist	14:45	jfk	01:00
Ek202	1	03.01.2021	15	Tk-0007	dxb	06:45	esb	09:40
Ek202	2	03.01.2021	15	Hy-9841	esb	11:45	lhr	16:30
Ek202	3	03.01.2021	15	Kgx-8704	lhr	21:00	jfk	03:10
Tk2313	0	05.01.2021	50	Jx-1232	ist	06:45	esb	07:50
Tk3	0	06.01.2021	48	Pl-9630	ist	07:30	dxb	10:45
Ek121	0	02.01.2021	80	Ol-7410	jfk	15:00	dxb	23:55
Pc3168	0	07.01.2021	115	Yn-023	saw	09:45	esb	10:40
Ba0104	0	10.01.2021	150	Lbj-0023	saw	16:20	esb	17:20

FARE:

Every flight number has its own amount the restrictions and its fare code.
in fare code for example, letters such as "**L, M, N, Q, T, V**, and **X**" usually refer to discounted economy class tickets.
Y is a full fare economy ticket, J and C refer to full fare business class tickets, and F refers to full fare first class tickets.

Restriction: we have 3 type of restrictions based on the amount of ticket price.
economy class, business class and first class.

Flight_number	Fare_code	Amount	Restriction
Tk2313	L	450	economy class
Tk3	M	3.200	economy class
Ba0113	N	2.200	economy class
Ek121	Q	2.100	economy class
Pc3168	T	170	economy class
Ba0104	V	1.750	economy class
Ek202	J	3.200	business class
Tk2163	F	1.400	first class

AIRPLANE:

We have 11 airplanes either from BA type or AIR type. We also declared its model type and their total number of seats.

Airplane_id	Total_number_of_seats	AirPl_type	Company_id
Jx-1232	170	A320	air
Un-3102	170	A320	ba
Fk-221	350	B777	ba
Bre-3112	510	A380	air
Tk-0007	170	A320	air
Hy-9841	200	B737	ba
Kgx-8704	360	B777	ba
Pl-9630	350	B777	ba
Ol-7410	350	B747	ba
Yn-023	120	A220	air
Lbj-0023	350	B747	ba

AIRPLANE TYPE:
There is different type of airplanes with different capacities.

Airplane_type_name	Max_seats
A220	133
A320	186
A380	525
A330	277
B737	215
B747	366
B777	368
B787	300

CAN LAND:
Different airplane types can land to different airports. The data we define in can land table are accordingly to the flight leg and flight leg instance.
as example: the Izmir ADB airport is a possible destination for flights carried out by A220, A320 airplanes types. But however, A380 can not land into ADB airport. It can only land in IST airport of Istanbul, Dubai's DXB airport and New York's JFK airport.

Airplane_type_name	Airport_code
A220	ADB
A320	ADB
A220	ESB
B737	ESB
A320	ESB
A320	IST
A380	IST
B777	IST
B747	SAW
A330	SAW
A220	SAW
B737	DXB
A380	DXB
A320	DXB
B777	DXB
B747	DXB
B777	LHR
B747	LHR
B737	LHR
A380	JFK
B787	JFK
B747	JFK

SEAT RESERVATION:

For every flight leg, every passenger has its seat numbers. Passengers were defined by their passport number.
seat number A01 is reserved by the first-class passenger
seat number A05 is reserved by the business class passengers
the other seats are reserved by economy class passengers.

Flight_number	Leg_number	Date	Seat_number	Passport_no
Tk2163	1	01.01.2021	A01	S0001325
Tk2163	2	01.01.2021	A01	S0001325
Ba0113	1	03.01.2021	F23	FA000135
Ba0113	2	03.01.2021	C17	FA000135
Ek202	1	03.01.2021	A05	K8745513
Ek202	2	03.01.2021	F07	K8745513
Ek202	3	03.01.2021	A06	K8745513
Tk2313	N	05.01.2021	B15	R123456
Tk3	N	06.01.2021	F27	P031245
Ek121	N	02.01.2021	J22	S0147852
Pc3168	N	07.01.2021	D12	V025874
Ba0104	N	10.01.2021	D22	B020018

FFC:
every customer is a flyer, some fly frequently some fly for the first time. We described our frequent flyer customer by their total mileage rewarded for their flights, passport numbers and their FFC_id.

Total_milage	Passport#	Ffc_id
650	R123456	OI214500
2.580	P031245	QW48511
2.700	S0147852	LP784330
600	V025874	OI217800
850	B020018	BX147952
1.780	FA000135	UY320014
5.000	K8745513	OI4514500
1.100	S0001325	QW478811

CUSTOMER:

We have 8 customers from different nationalities with their personal information.

Costumer_name	Passport#	Country	Customer_phone	Address	Email
James angelinton	R123456	USA	01222333555444	59 street, No 10 ,D 5 ,Beverly hills , los Angeles	Jj13@gmail.com
Mehmet yildiz	P031245	turkey	00905553332222	4023 street ,NO 5, erzene , Bornova , Izmir	M35@gmail.com
Ahmad elmasri	S0147852	Egypt	00210006665551	894 street NO: 3 Cairo	Buffer43@yahoo.com
Wui chan	V025874	china	00310005556661	321 street no 17 Wuhan	Xindex32@hotmail.com
Miguel Sanchez	B020018	spain	00458884441110	0123 street no:3 Madrid	Yhet@gmail.com
Aleksie petrov	FA000135	russia	00721000033355	0155 street no:4 St. Petersburg	Geresho@mail.ru
Elif akarsu	K8745513	turkey	00905539540817	12 street no: 9 mithatpasa , istanbul	Elifakarsu@gmail.com
Jammy motinge	S0001325	somali	00876662225544	45 street no:9 mogadishu	motinge@hotmail.com

AIRLINE:

There are 4 airline companies in our database. Turkish airlines as TK, British Airways as BAW, Fly Emirates as EMR and Pegasus as PGS.

Airline_id	Company_id
Turkish airlines	Turkish airlines a.s
British airways	International airlines group
emirates	The emirates group
pegasus	Pegasus a.s

SINGLE_FFC:

this table hold the information of every passenger's flight records detailly.

in this table, for every flight leg of customer its mileage per leg, fare code and its frequent flyer program ID has been defined.

the mileage holds the reward mileage for every customer which than will be added to FFC.Total_mileage.

every information about flight legs are uniquely classified by single_FFC_ID. This will help us for reaching customers record about its flight and their rewards.

Mileage	Singleffc_no	Date	Leg_number	Passport_no	Ffc_id	Flight_no	Fare_code
600	1	01.01.2021	1	S0001325	QW478811	Tk2163	F
500	2	01.01.2021	2	S0001325	QW478811	Tk2163	F
850	3	03.01.2021	1	FA000135	UY320014	Ba0113	N
930	4	03.01.2021	2	FA000135	UY320014	Ba0113	N
1.700	5	03.01.2021	1	K8745513	OI4514500	Ek202	J
1.500	6	03.01.2021	2	K8745513	OI4514500	Ek202	J
1.800	7	03.01.2021	3	K8745513	OI4514500	Ek202	J
650	8	05.01.2021	N	R123456	OI214500	Tk2313	L
2.580	9	06.01.2021	N	P031245	QW48511	Tk3	M
2.700	10	02.01.2021	N	S0147852	LP784330	Ek121	Q
600	11	07.01.2021	N	V025874	OI217800	Pc3168	T
850	12	10.01.2021	N	B020018	BX147952	Ba0104	V

2)DATABASE IMPLEMENTATION PROCESS

PART1 - (DDL statements) for creating the database and its relational model.
We use MYSQL database service for implementing our database model.

firstly, we create our database;

```
create database AIRLINE;
use AIRLINE;
```

2.1)AFTER CREATING THE AIRLINE DATABASE, WE START CREATING TABLES.

```
CREATE TABLE COMPANY (
    Company_id CHAR(3) NOT NULL,
    Company_name CHAR(30),
    PRIMARY KEY (Company_id)
);

CREATE TABLE AIRPORT (
    Airport_code CHAR(3) NOT NULL,
    Name CHAR(20),
    City CHAR(10),
    State CHAR(15),
    PRIMARY KEY (Airport_code)
);

CREATE TABLE AIRLINE (
    Airline_name CHAR(20) NOT NULL,
    Company_id CHAR(3),
    PRIMARY KEY (Airline_id),
    FOREIGN KEY (Company_id)
        REFERENCES COMPANY (Company_id)
        ON DELETE SET NULL ON UPDATE CASCADE
);


CREATE TABLE FLIGHT(
Flight_number CHAR(10) NOT NULL,
Airline_name CHAR(20),
Weekdays CHAR(15),
PRIMARY KEY(Flight_number),
FOREIGN KEY (Airline_name) REFERENCES AIRLINE(Airline_name)    ON DELETE SET NULL
  ON UPDATE CASCADE;);



CREATE TABLE FLIGHT_LEG (
    Flight_number CHAR(10) NOT NULL,
    Leg_number TINYINT NOT NULL,
    Departure_airport_code CHAR(3),
    Scheduled_departure_time TIME,
    Arrival_airport_code CHAR(3),
    Scheduled_arrival_time TIME,
    PRIMARY KEY (Flight_number , Leg_number),
    FOREIGN KEY (Flight_number)
        REFERENCES FLIGHT (Flight_number)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Departure_airport_code)
        REFERENCES AIRPORT (Airport_code)
        ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (Arrival airport code)
        REFERENCES AIRPORT (Airport_code)
        ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE LEG_INSTANCE (
    Flight number CHAR(8) NOT NULL,
    Leg_number TINYINT NOT NULL,
    Dates DATE NOT NULL,
    Airplane_id CHAR(8),
    Number_of_available_seats SMALLINT,
    Departure_airport_code CHAR(3),
    Departure_time TIME,
```

```
    Arrival_airport_code CHAR(3),
    Arrival_time TIME,
    PRIMARY KEY (Flight_number , Leg_number , Date),
    FOREIGN KEY (Flight_number , Leg_number)
        REFERENCES FLIGHT_LEG (Flight_number , Leg_number)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Airplane_id)
        REFERENCES AIRPLANE (Airplane_id)
        ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (Departure_airport_code)
        REFERENCES AIRPORT (Airport_code)
        ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (Arrival_airport_code)
        REFERENCES AIRPORT (Airport_code)
        ON DELETE SET NULL ON UPDATE CASCADE
);


CREATE TABLE FARE (
    Flight_number CHAR(8) NOT NULL,
    Fare_code CHAR NOT NULL,
    Restrictions CHAR(15),
    Amount DOUBLE,
    PRIMARY KEY (Flight_number , Fare_code),
    FOREIGN KEY (Flight_number)
        REFERENCES FLIGHT (Flight number)
        ON DELETE CASCADE ON UPDATE CASCADE
);


CREATE TABLE AIRPLANE_TYPE (
    Airplane type name CHAR(5) NOT NULL,
    Max_seats SMALLINT,
    PRIMARY KEY (Airplane_type_name)
);

CREATE TABLE CAN_LAND (
    Airplane type name CHAR(5) NOT NULL,
    Airport_code CHAR(3) NOT NULL,
    PRIMARY KEY (Airplane_type_name , Airport_code),
    FOREIGN KEY (Airplane_type_name)
        REFERENCES AIRPLANE_TYPE (Airplane_type_name)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Airport_code)
        REFERENCES AIRPORT (Airport_code)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE AIRPLANE(
Airplane_id CHAR(8) NOT NULL,
Airplane_type CHAR(5) NOT NULL,
Company_id CAHR(3),
Total_number_of_seats SMALLINT,
PRIMARY KEY(Airplane_id),
FOREIGN KEY(Airplane_type) REFERENCES AIRPLANE_TYPE(Airplane_type_name)   ON DELETE CASCADE
  ON UPDATE CASCADE,
FOREIGN KEY(Company_id) REFERENCES COMPANY(Company_id)  ON DELETE SET NULL
  ON UPDATE CASCADE);

CREATE TABLE SEAT_RESERVATION (
    Flight_number CHAR(10) NOT NULL,
    Leg_number TINYINT NOT NULL,
    Date DATE NOT NULL,
    Seat_number CHAR(4) NOT NULL,
    Passport_no CHAR(10),
    PRIMARY KEY (Flight_number , Leg_number , Date , Seat_number),
    FOREIGN KEY (Flight_number , Leg_number , Date)
        REFERENCES LEG_INSTANCE (Flight_number , Leg_number , Date)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Passport_no)
        REFERENCES CUSTOMER (Passport_no)
        ON DELETE SET NULL ON UPDATE CASCADE
);



CREATE TABLE CUSTOMER (
```

```
    Passport_no CHAR(15) NOT NULL,
    Address CHAR(50),
    Country CHAR(25),
    Email CHAR(40),
    Customer_name CHAR(25),
    Customer_phone BIGINT,
    PRIMARY KEY (Passport_number)
);
```

```
CREATE TABLE FFC(
Passport_no CHAR(10),
Ffc_id CHAR(10),
Total_mileage SMALLINT DEFAULT 0,
PRIMARY KEY(Passport_no,Ffc_id)
FOREIGN KEY(Passport_no) REFERENCES CUSTOMER(Passport_no)   ON DELETE CASCADE
  ON UPDATE CASCADE);
```

```
CREATE TABLE SINGLE_FFC (
    Passport_no CHAR(10),
    Singleffc_no TINYINT NOT NULL,
    Mileage SMALLINT,
    Flight number CHAR(8),
    Leg_number TINYINT,
    Dates DATE,
    Fare_code CHAR(1),
    Ffc_id CHAR(10),
    Amount DOUBLE,

    PRIMARY KEY (Singleffc_no),
    FOREIGN KEY (Passport_no , Ffc_id)
        REFERENCES FFC (Passport_no , Ffc_id)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Flight_number , Leg_number , Date , Seat_number)
        REFERENCES SEAT RESERVATION (Flight number , Leg_number , Date , Seat_number)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Flight_number , Fare_code)
        REFERENCES FARE (Flight_number , Fare_code)
        ON DELETE SET NULL ON UPDATE CASCADE
);
```

2.2)INSERTING VALUES INTO THE TABLES

```
INSERT INTO COMPANY VALUES
 ('air', 'Airbus Se'),
 ('ba', 'The beoeing company'),
 ('tk', 'Turkish airlines a.s'),
 ('pgs', 'Pegasus a.s'),
 ('emr', 'The emirates group'),
 ('baw', 'International airlines group');

INSERT INTO AIRLINE VALUES
 ('Turkish airlines', 'tk'),
 ('Brtish airways', 'baw'),
 ('Emirates', 'emr'),
 ('Pegasus', 'pgs');


INSERT INTO AIRPLANE_TYPE VALUES
 ('A220', '133'),
 ('A320', '186'),
 ('A380', '525'),
('A330', '277'),
('B737', '215'),
('B747', '366'),
('B777', '368'),
('B787', '300');

INSERT INTO AIRPLANE VALUES
 ('Jx-1232', 'A320', 'air', '170'),
 ('Un-3102', 'A320', 'ba', '170'),
```

```
 ('Fk-221', 'B777', 'ba', '350'),
 ('Bre-3112', 'A380', 'air', '510'),
 ('Tk-00007', 'A320', 'air', '170'),
 ('Hy-9841', 'B737', 'ba', '200'),
 ('Kgx-8704', 'B777', 'ba', '360'),
 ('Pl-9630', 'B777', 'ba', '350'),
 ('Ol-7410', 'B747', 'ba', '350'),
 ('Yn-023', 'A220', 'air', '120'),
 ('Lbj-0023', 'B747', 'ba', '350');

INSERT INTO FLIGHT VALUES
 ('Tk2313', 'Turkish airlines', 'Monday'),
 ('Tk3', 'Turkish airlines', 'Tuesday'),
 ('Ba0113', 'British airways', 'Tuesday'),
 ('Ek121', 'Emirates', 'Friday'),
 ('Pc3168', 'Pegasus', 'Wednesday'),
 ('Ba0104', 'British airways', 'Thursday'),
 ('Ek202', 'Emirates', 'Wednesday'),
 ('Tk2163', 'Turkish airlines', 'Friday');



INSERT INTO FARE VALUES
 ('Tk2313', 'L', 'Economy class', '450');
 ('Tk3', 'M', 'Economy class', '3200');
 ('Ba0113', 'N', 'Economy class', '2200');
 ('Ek121', 'Q', 'Economy class', '2100');
 ('Pc3168', 'T', 'Economy class', '170');
 ('Ba0104', 'V', 'Economy class', '1750');
 ('Ek202', 'J', 'Business class', '3200');
 ('Tk2163', 'F', 'First Class', '1400');


INSERT INTO AIRPORT VALUES
 ('Adb', 'İzmir Adnan Menderes', 'İzmir', 'Ege'),
 ('Esb', 'Esenboğa', 'Ankara', 'İç anadolu'),
 ('İst', 'İstanbul Airport', 'İstanbul', 'Marmara'),
 ('Saw', 'İstanbul Airport', 'İstanbul', 'Marmara'),
 ('Jfk', 'John F Kennedy', 'Newyork', 'Newyork'),
 ('Ihr', 'Heathrow', 'North', 'North england'),
 ('Dxb', 'Dubai Airport', 'Dubai', 'Middle east');

INSERT INTO FLIGHT_LEG
 ('Tk2163', '1', 'Esb', '10:05', 'Adb', '10:45'),
 ('Tk2163', '2', 'Adb', '11:45', 'İst', '12:35'),
 ('Tk2313', '0', 'İst', '06:45', 'Esb', '07:40'),
 ('Tk3', '0', 'İst', '07:15', 'Dxb', '10:20'),
 ('Ba0113', '1', 'Ihr', '08:20', 'İst', '12:20'),
 ('Ba0113', '2', 'İst', '14:10', 'Jfk', '00:00'),
 ('Ek121', '0', 'Jfk', '15:00', 'Dxb', '23:55'),
 ('Pc3168', '0', 'Saw', '09:00', 'Esb', '09:50'),
 ('Ek202', '1', 'Dxb', '06:45', 'Esb', '09:40'),
 ('Ek202', '2', 'Esb', '11:45', 'Ihr', '16:30'),
 ('Ek202', '3', 'Ihr', '21:00', 'Jfk', '03:10'),
 ('Ba0104', '0', 'Saw', '16:20', 'Ihr', '22:15');

INSERT INTO LEG_INSTANCE VALUES
 ('Tk2163', '1', '2021.01.01', 'Jx-1232', 'Esb', '10:05', 'Adb', '11:20', '10'),
 ('Tk2163', '2', '2021.01.01', 'Un-3102', 'Adb', '12:30', 'İst', '13:25', '10'),
 ('Ba0113', '1', '2021.01.03', 'Fk-221', 'Ihr', '08:20', 'İst', '12:20', '220'),
 ('Ba0113', '2', '2021.01.03', 'Bre-3112', 'İst', '14:45', 'Jfk', '01:00', '220'),
 ('Ek202', '1', '2021.01.03', 'Tk-0007', 'Dxb', '06:45', 'Esb', '09:40', '15'),
 ('Ek202', '2', '2021.01.03', 'Hy-9841', 'Esb', '11:45', 'Ihr', '16:30', '15'),
 ('Ek202', '3', '2021.01.03', 'Kgx-8704', 'Ihr', '21:00', 'Jfk', '03:10', '15'),
 ('Tk2313', '0', '2021.01.05', 'Jx-1232', 'İst', '06:45', 'Esb', '07:50', '50'),
 ('Tk3', '0', '2021.01.06', 'Pl-9630', 'İst', '07:30', 'Dxb', '10:45', '48'),
 ('Ek121', '0', '2021.01.02', 'Ol-7410', 'Jfk', '15:00', 'Dxb', '23:55', '80'),
 ('Pc3168', '0', '2021.01.07', 'yn-023', 'Saw', '09:45', 'Esb', '10:40', '115'),
 ('Ba0104', '0', '2021.01.10', 'Lbj-0023', 'Saw', '16:20', 'Esb', '17:20', '150');

INSERT INTO CAND_LAND
 ('A220', 'Adb'),
 ('A320', 'Adb'),
 ('A220', 'Esb'),
 ('B737', 'Esb'),
 ('A320', 'Esb'),
 ('A380', 'İst'),
 ('B777', 'İst'),
```

```
 ('B747', 'Saw'),
 ('A330', 'Saw'),
 ('A220', 'Saw'),
 ('B737', 'Dxb'),
 ('A380', 'Dxb'),
 ('A320', 'Dxb'),
 ('B777', 'Dxb'),
 ('B747', 'Dxb'),
 ('B777', 'Lhr'),
 ('B747', 'Lhr'),
 ('B737', 'Lhr'),
 ('A380', 'Jfk'),
 ('B787', 'Jfk'),
 ('B747', 'Jfk'),
 ('A320', 'İst');

INSERT INTO CUSTOMER VALUES
('R123456', '59 street, No 10 ,D 5 ,Beverly hills , los Angeles ', ' USA ', ' Jj13@gmail.com
', ' James angelinton ', '01222333555444'),
('P031245', '4023 street ,NO 5, erzene , Bornova , Izmir ', ' Turkey ', ' M35@gmail.com ', '
Mehmet yildiz ', '00905553332222'),
('S0147852', '894 street NO: 3 Cairo ', ' Egypt ', ' Buffer43@yahoo.com ', ' Ahmad elmasri ',
'00210006665551'),
('V025874', '321 street no 17 Wuhan ', ' China ', ' Xindex32@hotmail.com ', ' Wui chan ',
'00310005556661'),
('B020018', '0123 street no:3 Madrid ', ' Spain ', ' Yhet@gmail.com ', ' Miguel Sanchez ',
'00458884441110'),
('FA000135', '0155 street no:4 St. Petersburg ', ' Russia ', ' Geresho@mail.ru ', ' Aleksie
petrov ', '00721000033355'),
('K8745513', '12 street no: 9 mithatpasa , istanbul ', ' Turkey ', ' Elifakarsu@gmail.com ', '
Elif akarsu ', '00905539540817'),
('S0001325', '45 street no:9 mogadishu ', ' Somali ', ' motinge@hotmail.com ', ' Jammy motinge
', '00876662225544');

INSERT INTO SEAT_RESERVATION VALUES
('Tk2163', '1', '2021.01.01', 'A01', 'S0001325'),
('Tk2163', '2', '2021.01.01', 'A01', 'S0001325'),
('Ba0113', '1', '2021.01.03', 'F23', 'FA000135'),
('Ba0113', '2', '2021.01.03', 'C17', 'FA000135'),
('Ek202', '1', '2021.01.03', 'A05', 'K8745513'),
('Ek202', '2', '2021.01.03', 'F07', 'K8745513'),
('Ek202', '3', '2021.01.03', 'A06', 'K8745513'),
('Tk2313', '0', '2021.01.05', 'B15', 'R123456'),
('Tk3', '0', '2021.01.06', 'F27', 'P031245'),
('Ek121', '0', '2021.01.02', 'J22', 'S0147852'),
('Pc3168', '0', '2021.01.07', 'D12', 'V025874'),
('Ba0104', '0', '2021.01.10', 'D22', 'B020018');

INSERT INTO FFC VALUES
('R123456', ' OI214500', '650'),
('P031245', ' QW48511', '2.580'),
('S0147852', ' LP784330', '2.700'),
('V025874', ' OI217800', '600'),
('B020018', ' BX147952', '850'),
('FA000135', ' UY320014', '1.780'),
('K8745513', ' OI4514500', '5.000'),
('S0001325', ' QW478811', '1.100');

INSERT INTO SINGLE_FFC VALUES
(' S0001325', '1', '600', ' Tk2163', '1', '2021.01.01', 'F', ' QW478811', ' 1260'),
(' S0001325', '2', 500'', ' Tk2163', '2', ''2021.01.01', 'F', ' QW478811', ' 1260'),
(' FA000135', '3', '850', ' Ba0113', '1', ''2021.01.03', 'N', ' UY320014', ' 2200'),
(' FA000135', '4', '930', ' Ba0113', '2', ''2021.01.03', 'N', ' UY320014', ' 1260'),
(' K8745513', '5', '1.700', ' Ek202', '1', ''2021.01.03', 'J', ' OI4514500', ' 2560'),
(' K8745513', '6', '1.500', ' Ek202', '2', ''2021.01.03', 'J', ' OI4514500', ' 2800'),
(' K8745513', '7', '1.800', ' Ek202', '3', ''2021.01.03', 'J', ' OI4514500', ' 2800'),
(' R123456', '8', '650', ' Tk2313', '0', ''2021.01.05', 'L', ' OI214500', ' 405'),
(' P031245', '9', '2.580', ' Tk3', '0', ''2021.01.06', 'M', ' QW48511', ' 3200'),
(' S0147852', '10', '2.700', ' Ek121', '0', ''2021.01.02', 'Q', ' LP784330', ' 2100'),
(' V025874', '11', '600', ' Pc3168', '0', ''2021.01.07', 'T', ' OI217800', ' 170'),
(' B020018', '12', '850', ' Ba0104', '0', ''2021.01.10', 'V', ' BX147952', ' 1750');
```

2.3)After defining DDL statements now we start populating our database:

```
select * from company;
```

Result Grid | Filter Rows:

company_id	company_name
air	Airbus Se
ba	The beoeing company
baw	International airlines group
emr	The emirates group
pgs	Pegasus a.s
tk	Turkish airlines a.s
NULL	NULL

```
SELECT * from airport;
```

Result Grid | Filter Rows: | Edit:

airport_code	airport_name	city	state
adb	Izmir adnan meneres	Izmir	ege
dxb	Dubai airport	Dubai	Middle east
Esb	esenboga	ankara	Ic anadolu
ist	Istanbul airport	istanbul	marmara
jfk	John f kennedy	newyork	newyork
lhr	Heathrow	london	North england
Saw	Istanbul airport	istanbul	marmara
NULL	NULL	NULL	NULL

```
select * from flight;
```

Result Grid | Filter Rows: | Edit:

flight_number	weekdays	airline_name
Ba0104	thursday	British airways
Ba0113	tuesday	British airways
Ek121	friday	emirates
Ek202	wednesday	emirates
Pc3168	wednesday	pegasus
Tk2163	friday	Turkish airlines
Tk2313	monday	Turkish airlines
Tk3	tuesday	Turkish airlines
NULL	NULL	NULL

```
select * from  flight_leg;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

flight_number	leg_num	dep_airport_code	sched_depart_time	sched_arrival_time	arr_airport_code	mileage
Ba0104	0	saw	16:20:00	22:15:00	lhr	600
Ba0113	1	lhr	08:20:00	12:20:00	ist	850
Ba0113	2	ist	14:10:00	00:00:00	jfk	930
Ek121	0	jfk	15:00:00	23:55:00	dxb	2700
Ek202	1	dxb	06:45:00	09:40:00	esb	1700
Ek202	2	esb	11:45:00	16:30:00	lhr	1500
Ek202	3	lhr	21:00:00	03:10:00	jfk	1800
Pc3168	0	saw	09:00:00	09:50:00	esb	660
Tk2163	1	esb	10:05:00	10:55:00	adb	600
Tk2163	2	adb	11:45:00	12:35:00	ist	500
Tk2313	0	ist	06:45:00	07:40:00	esb	650
Tk3	0	ist	07:15:00	10:20:00	dxb	2580
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
select * from leg_instance;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

flight_number	leg_num	flight_date	num_availible_seat	airplane_id	dep_airport_code	dep_time	arr_airport_code	arr_time
Ba0104	0	2021-01-07	150	Lbj-0023	saw	16:20:00	esb	17:20:00
Ba0113	1	2021-01-03	220	Fk-221	lhr	08:20:00	ist	12:20:00
Ba0113	2	2021-01-03	220	Bre-3112	ist	14:45:00	jfk	01:00:00
Ek121	0	2021-01-02	80	Ol-7410	jfk	15:00:00	dxb	23:55:00
Ek202	1	2021-01-03	15	Tk-0007	dxb	06:45:00	esb	09:40:00
Ek202	2	2021-01-03	15	Hy-9841	esb	11:45:00	lhr	16:30:00
Ek202	3	2021-01-03	15	Kgx-8704	lhr	21:00:00	jfk	03:10:00
Pc3168	0	2021-01-07	115	Yn-023	saw	09:45:00	esb	10:40:00
Tk2163	1	2021-01-01	10	Jx-1232	esb	10:05:00	adb	11:20:00
Tk2163	2	2021-01-01	10	Un-3102	adb	12:30:00	ist	13:25:00
Tk2313	0	2021-01-05	50	Jx-1232	ist	06:45:00	esb	07:50:00
Tk3	0	2021-01-06	48	Pl-9630	ist	07:30:00	dxb	10:45:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
select * from fare;
```

Result Grid | Filter Rows: | Edit:

flight_number	fare_code	amount	restriction
Ba0104	V	1750	economy class
Ba0113	N	2200	economy class
Ek121	Q	2100	economy class
Ek202	J	3200	business class
Pc3168	T	170	economy class
Tk2163	F	1400	first class
Tk2313	L	450	economy class
Tk3	M	3200	economy class
NULL	NULL	NULL	NULL

```
select * from airplane_type;
```

Result Grid | Filter Rows:

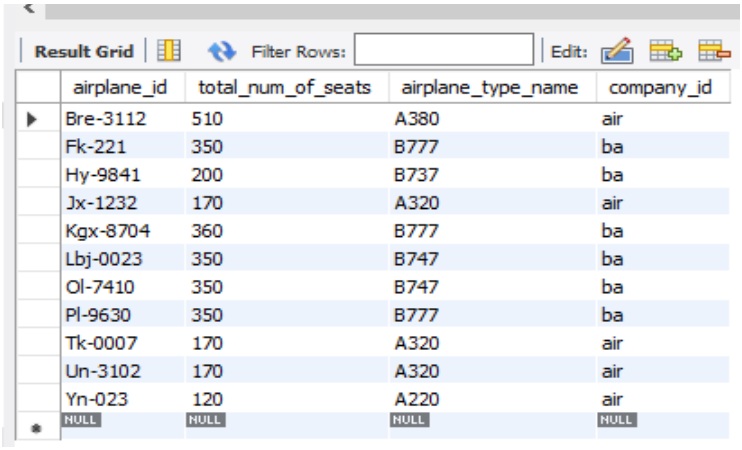
airplane_type_name	max_seats
A220	133
A320	186
A330	277
A380	525
B737	215
B747	366
B777	368
B787	300
NULL	NULL

```
select * from can_land;
```

Result Grid | Filter Rows:

airplane_type_name	airport_code
A320	ADB
A320	DXB
A380	DXB
B737	DXB
B747	DXB
B777	DXB
A220	ESB
A320	ESB
B737	ESB
A320	IST
A380	IST
B777	IST
A380	JFK
B747	JFK
B787	JFK
B737	LHR
B747	LHR
B777	LHR
A220	SAW
A330	SAW
B747	SAW
NULL	NULL

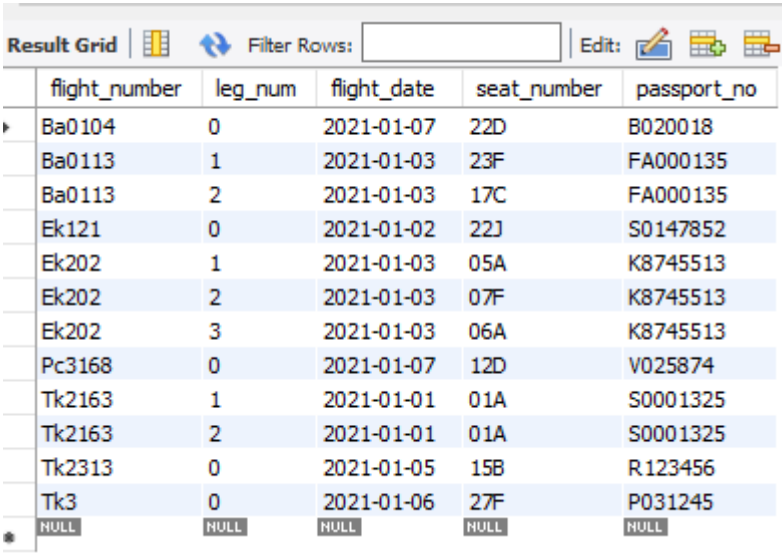
```
select * from airplane;
```



Result Grid | Filter Rows: | Edit:

airplane_id	total_num_of_seats	airplane_type_name	company_id
Bre-3112	510	A380	air
Fk-221	350	B777	ba
Hy-9841	200	B737	ba
Jx-1232	170	A320	air
Kgx-8704	360	B777	ba
Lbj-0023	350	B747	ba
Ol-7410	350	B747	ba
Pl-9630	350	B777	ba
Tk-0007	170	A320	air
Un-3102	170	A320	air
Yn-023	120	A220	air
NULL	NULL	NULL	NULL

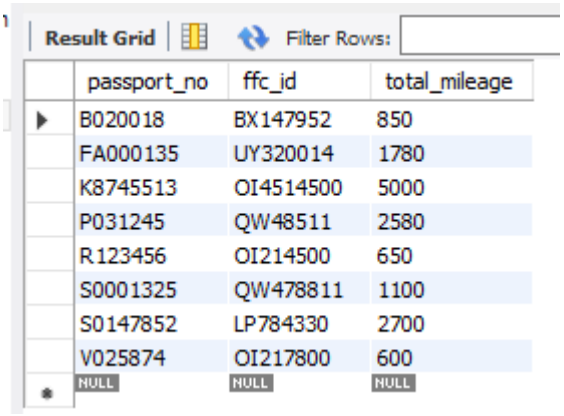
```
select * from seat_reservation;
```



Result Grid | Filter Rows: | Edit:

flight_number	leg_num	flight_date	seat_number	passport_no
Ba0104	0	2021-01-07	22D	B020018
Ba0113	1	2021-01-03	23F	FA000135
Ba0113	2	2021-01-03	17C	FA000135
Ek121	0	2021-01-02	22J	S0147852
Ek202	1	2021-01-03	05A	K8745513
Ek202	2	2021-01-03	07F	K8745513
Ek202	3	2021-01-03	06A	K8745513
Pc3168	0	2021-01-07	12D	V025874
Tk2163	1	2021-01-01	01A	S0001325
Tk2163	2	2021-01-01	01A	S0001325
Tk2313	0	2021-01-05	15B	R123456
Tk3	0	2021-01-06	27F	P031245
NULL	NULL	NULL	NULL	NULL

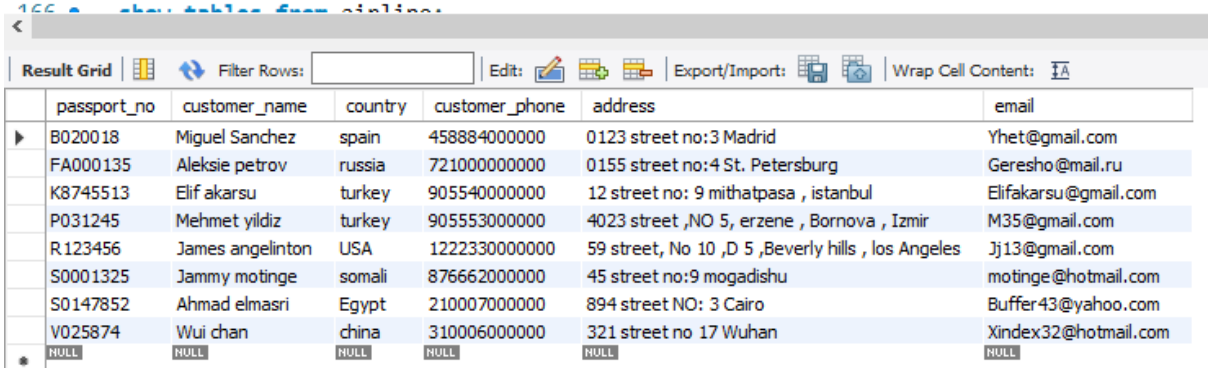
```
select * from ffc;
```



Result Grid | Filter Rows:

passport_no	ffc_id	total_mileage
B020018	BX147952	850
FA000135	UY320014	1780
K8745513	OI4514500	5000
P031245	QW48511	2580
R123456	OI214500	650
S0001325	QW478811	1100
S0147852	LP784330	2700
V025874	OI217800	600
NULL	NULL	NULL

```
select * from customer;
```

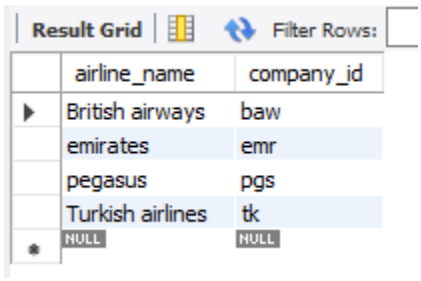


passport_no	customer_name	country	customer_phone	address	email
B020018	Miguel Sanchez	spain	458884000000	0123 street no:3 Madrid	Yhet@gmail.com
FA000135	Aleksie petrov	russia	721000000000	0155 street no:4 St. Petersburg	Geresho@mail.ru
K8745513	Elif akarsu	turkey	905540000000	12 street no: 9 mithatpasa , istanbul	Elifakarsu@gmail.com
P031245	Mehmet yildiz	turkey	905553000000	4023 street ,NO 5, erzene , Bornova , Izmir	M35@gmail.com
R123456	James angelinton	USA	1222330000000	59 street, No 10 ,D 5 ,Beverly hills , los Angeles	Jj13@gmail.com
S0001325	Jammy motinge	somali	876662000000	45 street no:9 mogadishu	motinge@hotmail.com
S0147852	Ahmad elmasri	Egypt	210007000000	894 street NO: 3 Cairo	Buffer43@yahoo.com
V025874	Wui chan	china	310006000000	321 street no 17 Wuhan	Xindex32@hotmail.com

```
select * from single_ffc;
```

Passport_no	Singleffc_no	Mileage	Flight_number	Leg_number	Dates	Fare_code	Ffc_id	Amount
S0001325	1	600	Tk2163	1	2021-01-01	F	QW478811	1260
S0001325	2	500	Tk2163	2	2021-01-01	F	QW478811	1260
FA000135	3	850	Ba0113	1	2021-01-03	N	UY320014	2200
FA000135	4	930	Ba0113	2	2021-01-03	N	UY320014	2200
K8745513	5	1700	Ek202	1	2021-01-03	J	OI4514500	2560
K8745513	6	1500	Ek202	2	2021-01-03	J	OI4514500	2800
K8745513	7	1800	Ek202	3	2021-01-03	J	OI4514500	2800
R123456	8	650	Tk2313	0	2021-01-05	L	OI214500	405
P031245	9	2580	Tk3	0	2021-01-06	M	QW48511	3200
S0147852	10	2700	Ek121	0	2021-01-02	Q	LP784330	2100
V025874	11	660	Pc3168	0	2021-01-07	T	OI217800	170
B020018	12	850	Ba0104	0	2021-01-10	V	BX147952	1750

```
select * from airline;
```



airline_name	company_id
British airways	baw
emirates	emr
pegasus	pgs
Turkish airlines	tk

3) 5 meaningful triggers with its descriptions:

```
/* yeni bir müşteri kayıt olurken sadece gmail Hotmail ya da
yahoo mailleri ile kayıt olabilirler */

CREATE DEFINER=`root`@`localhost` TRIGGER `customer_BEFORE_INSERT` BEFORE INSERT ON `customer`
FOR EACH ROW BEGIN
IF(NEW.Email LIKE '%@gmail%')
THEN SET NEW.Email=NEW.Email;
ELSEIF(NEW.Email LIKE '%@hotmail%')
THEN SET NEW.Email=NEW.Email;
ELSEIF(NEW.Email LIKE '%@yahoo%')THEN SET NEW.Email=NEW.Email;
ELSE
SET NEW.Email='-Wrong Mail-';
END IF;
END

/* Total Mileageleri hesaplayan trigger */
```

```
CREATE DEFINER=`root`@`localhost` TRIGGER `single_ffc_AFTER_INSERT` AFTER INSERT ON
`single_ffc` FOR EACH ROW BEGIN
UPDATE FFC SET Total mileage=Total mileage+NEW.Mileage
WHERE new.Passport_no=FFC.Passport_no ;
END
```

/*First classta yolculuk yapanların mileagelerine ekstra %10 mileage puanı, Business classta uçanlara ise ekstra %5 mileage puanı ekleyen trigger ve kalan koltuk sayısı 20 den az ise bilet fiyatına %10 indirim uygulama ve gece 12 ve sabah 7 arasındaki biletlet fiyatlarında %10 indirim olur */

```
CREATE DEFINER=`root`@`localhost` TRIGGER `single_ffc_BEFORE_INSERT` BEFORE INSERT ON
`single_ffc` FOR EACH ROW BEGIN
IF(
SELECT Restrictions
FROM FARE
WHERE NEW.Flight_number=FARE.Flight_number AND NEW.Fare_code=FARE.Fare_code)='First class'
THEN
SET new.Mileage=new.Mileage*1.1;
ELSEIF(
SELECT Restrictions
FROM FARE
WHERE NEW.Flight_number=FARE.Flight_number AND NEW.Fare_code=FARE.Fare_code)='Business class'
THEN
SET new.Mileage=new.Mileage*1.05;
END IF;
IF(
SELECT L.Number_of_available_seats
FROM LEG_INSTANCE AS L
WHERE L.Flight_number=NEW.Flight_number AND L.Leg_number=NEW.Leg_number AND
L.Dates=NEW.Dates)<=20 THEN SET NEW.Amount=NEW.Amount*0.9;
END IF;
IF(
SELECT L.Departure_airport_code
FROM LEG_INSTANCE AS L
WHERE L.Flight_number=NEW.Flight_number AND L.Leg_number=NEW.Leg_number AND
L.Dates=NEW.Dates)<=07.00 THEN SET NEW.Amount=NEW.Amount*0.9;
END IF;
END (Seat reservationa yeni bir bilgi girildiğinde kullanılabilecek koltuk sayısını 1 azaltır)
CREATE DEFINER=`root`@`localhost` TRIGGER `seat_reservation_AFTER_INSERT` AFTER INSERT ON
`seat_reservation` 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.Dates=New.Dates;
END
```

/* bilet iptal edildiğinde total mileageden iptal edilen biletin Mileage değeri düşer ve kullanılabilir koltuk sayısı bir artar */

```
CREATE DEFINER=`root`@`localhost` TRIGGER `single_ffc_AFTER_DELETE` AFTER DELETE ON
`single_ffc` FOR EACH ROW BEGIN
UPDATE FFC SET FFC.Total_mileage=FFC.Total_mileage-Old.Mileage WHERE
FFC.Passport_no=Old.Passport_no;
UPDATE LEG_INSTANCE SET number_of_available_seats=number_of_available_seats+1
WHERE LEG_INSTANCE.Flight_number=OLD.Flight_number AND LEG_INSTANCE.Leg_number=OLD.Leg_number
AND
LEG_INSTANCE.Dates=OLD.Dates;
END
```

4) CHECK constraints with its descriptions:

Checks if the total number of seats are more than 0.

```
alter table airplane
add check (total_num_of_seats > 0);
```

checks if the mileage is more than 0.

```
ALTER TABLE single_ffc
ADD CHECK (mileage>0);
```

Checks if the fare amount is more than 0.

```
ALTER TABLE fare
ADD CHECK (amount>0);
```

Checks if the total mileage of frequent flyer customer is more than 0.

```
alter table ffc
add check ( total_mileage > 0);
```

checks if the maximum seats of airplanes are more than 0.

```
alter table airplane_type
add check ( max_seats > 0) ;
```

5) SQL Statements

5.1) insert, delete, and update statements for 5 tables.

Inserts a new airplane type and its max seat into Airplane type table

```
INSERT into airplane_type
values ( 'A350', 180);
select * from airplane_type;
```

updates a new max seat into Airplane type A350

```
update airplane_type
set Max_seats=200
where airplane_type_name = 'A350';
```

deletes Airplane type A350 from Airplane type table

```
delete from airplane_type
where airplane_type_name ='A350';
```

Inserts a new customer and info related to it into customer.

```
INSERT into customer
values ( 'D13400034','yamato', 'Japan',78451239000,'13 st tokyo ','yamato@yahoo.com');
select * from customer;
```

updates customer name having passport no of 'D13400034' in customer table.

```
update customer
set customer_name='Yoshida'
where passport_no = 'D13400034';
```

deletes customer with passport no of 'D13400034'

```
delete from customer
where passport_no = 'D13400034';
```

deletes passport no 'V025874'' from FFC table

```
DELETE FROM FFC WHERE Passport_no='V025874';
```

Inserts new values into FFC table.

```
INSERT INTO FFC VALUES
('V025874', 'OI217800', '600');
```

Inserts new values into FFC table.

```
INSERT INTO SINGLE_FFC VALUES
('V025874', '11', '600', 'Pc3168', '0', '2021.01.07', 'T', 'OI217800');
```

updates total mileage 1260 with 600 in FFC.

```
UPDATE FFC SET Total_mileage=600 WHERE Total_mileage=1260;
```

Insert new values into airport table.

```
insert into airport(airport_code,airport_name,city,state) values("ada","Adana
Havalimanı","adana","Akdeniz");
```

Deletes passport S0001325 from Seat reservation Table.

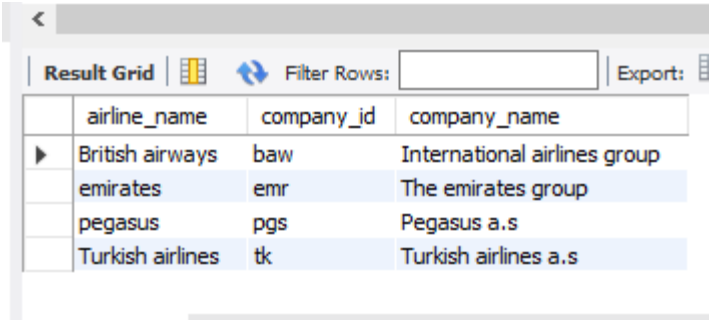
```
delete from seat_reservation where passport_no="S0001325";
update customer set customer_phone=905542600054 where passport_no="p031245";
```

5.2) 10 SELECT STATEMENTS

5.2.a) Using 2 Tables

Hava yolunun ait oldugu şirketin ismini ve bu havayolu ile ilgili isim ve şirket id sini döner.

```
select airline.airline_name,airline.company_id,company.company_name
from airline,company
where airline.company_id=company.company_id;
```

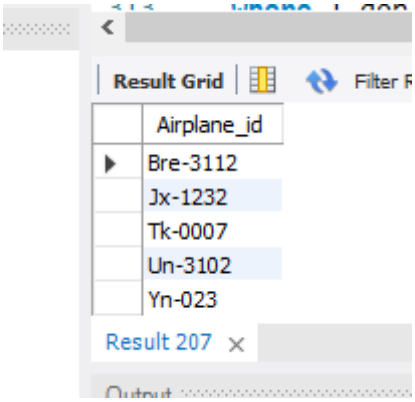


Result Grid | Filter Rows: | Export:

airline_name	company_id	company_name
British airways	baw	International airlines group
emirates	emr	The emirates group
pegasus	pgs	Pegasus a.s
Turkish airlines	tk	Turkish airlines a.s

Adı bilinen bir şirketin sahip olduğu uçakları listeler

```
SELECT A.Airplane_id
FROM AIRPLANE AS A, COMPANY AS C
WHERE C.Company_name='Airbus Se' AND C.Company_id=a.Company_id;
```



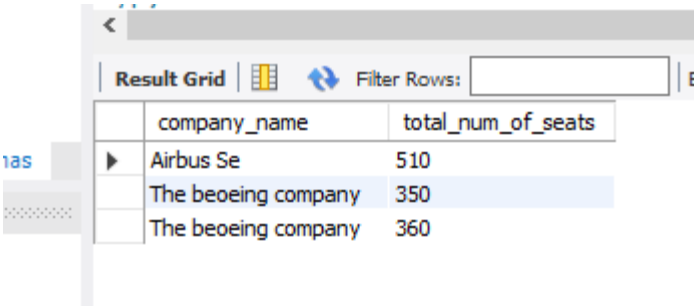
Result Grid | Filter R

Airplane_id
Bre-3112
Jx-1232
Tk-0007
Un-3102
Yn-023

Result 207

Retrieve the name of companies with its seat that have airplanes with total number of seats more than 200

```
select distinct c.company_name, a.total_num_of_seats
from company as c , airplane as a
where total_num_of_seats > 200
and c.company_id =a.company_id;
```



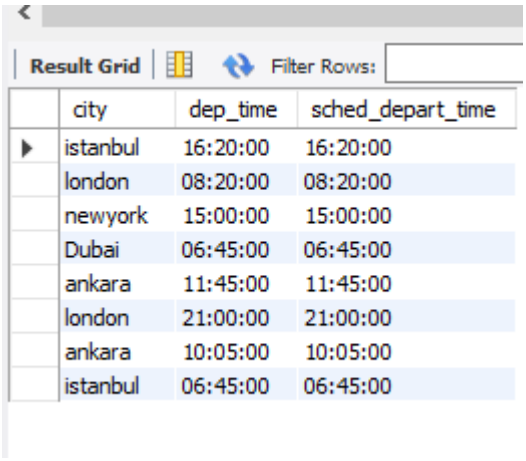
Result Grid | Filter Rows:

company_name	total_num_of_seats
Airbus Se	510
The beoeing company	350
The beoeing company	360

5.2.b) Using 3 Tables

Retrieve all the cities where flight departed accordingly with schedualed departure time

```
select distinct city,l.dep_time, fli.sched_depart_time
from airport, flight_leg as fli, leg_instance as l
where l.dep_time = fli.sched_depart_time
and l.dep_airport_code = airport.airport_code;
```

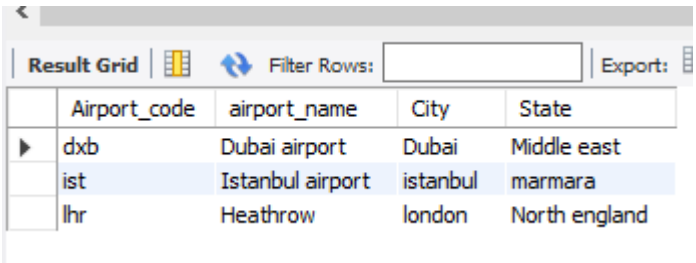


Result Grid | Filter Rows:

city	dep_time	sched_depart_time
istanbul	16:20:00	16:20:00
london	08:20:00	08:20:00
newyork	15:00:00	15:00:00
Dubai	06:45:00	06:45:00
ankara	11:45:00	11:45:00
london	21:00:00	21:00:00
ankara	10:05:00	10:05:00
istanbul	06:45:00	06:45:00

Bir uçağın hangi havalimanlarına gidebildiğini, bu havalimanlarının adları şehirleri ve bölgelerini alır

```
SELECT AI.Airport_code, AI.airport_name, AI.City, AI.State
FROM AIRPLANE AS A, CAN_LAND AS C, AIRPORT AS AI
WHERE A.Airplane_id='Fk-221' AND A.Airplane_type_name=C.Airplane_type_name AND
C.Airport_code=AI.Airport_code;
```

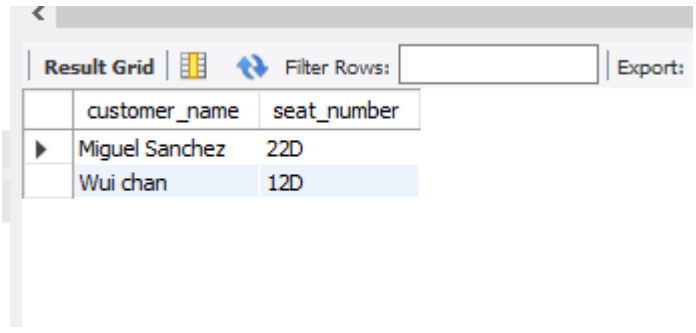


Result Grid | Filter Rows: | Export:

Airport_code	airport_name	City	State
dxb	Dubai airport	Dubai	Middle east
ist	Istanbul airport	istanbul	marmara
lhr	Heathrow	london	North england

customer names and seat numbers who are flying from SAW airport

```
select C.customer_name, S.seat_number
from customer as c , seat_reservation as S , leg_instance as l
where dep_airport_code = 'saw'
and l.flight_number = s.flight_number
and l.leg_num = s.leg_num
and l.flight_date = s.flight_date
and s.passport_no = c.passport_no;
```

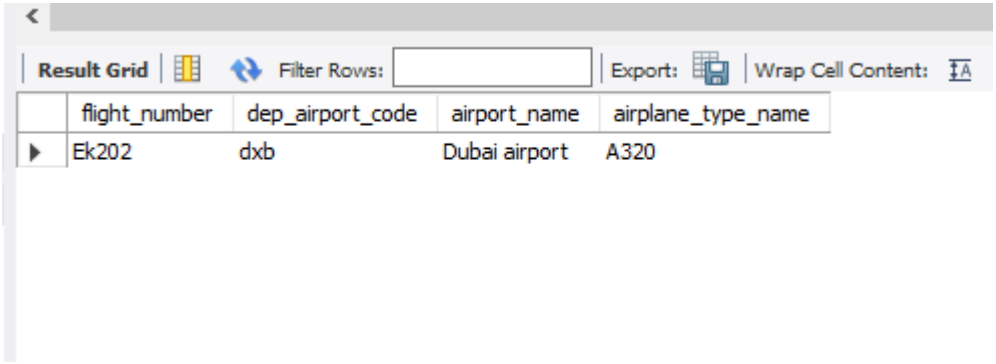


Result Grid | Filter Rows: | Export:

customer_name	seat_number
Miguel Sanchez	22D
Wui chan	12D

airport_code'u dxb olan uçuş numarasını,uçağın tipini, airport_codunu ve havaalanının ismini verir.

```
select
flight_leg.flight_number,flight_leg.dep_airport_code,airport.airport_name,can_land.airplane_ty
pe name
from flight_leg,airport,can_land
where airport.airport_code=flight_leg.dep_airport_code and
airport.airport code=can land.airport_code and flight_leg.dep_airport_code="dxb"
group by flight_leg.flight_number;
```

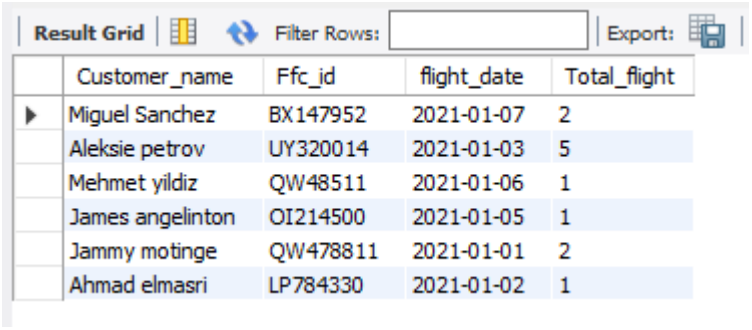


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

flight_number	dep_airport_code	airport_name	airplane_type_name
Ek202	dxb	Dubai airport	A320

Müşteri adları ve ffc_idlerine göre bir günde yaptıkları uçuş sayıları

```
SELECT C.Customer_name,F.Ffc_id, S.Dates, COUNT(*) AS Total_flight
FROM CUSTOMER AS C, SEAT_RESERVATION AS S, FFC AS F
WHERE C.Passport_no=F.Passport_no AND C.Passport_no=S.Passport_no
GROUP BY S.Dates
```



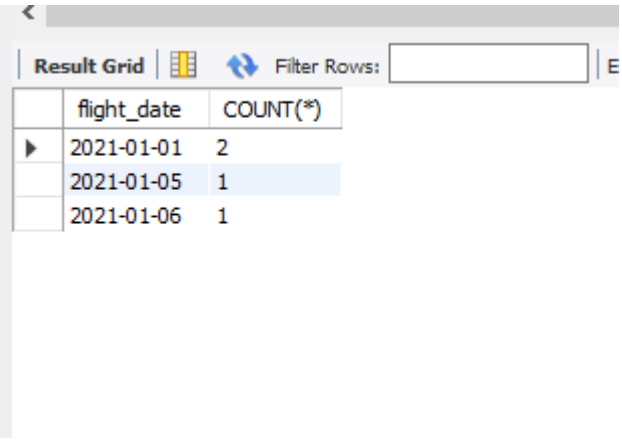
Result Grid | Filter Rows: | Export:

Customer_name	Ffc_id	flight_date	Total_flight
Miguel Sanchez	BX147952	2021-01-07	2
Aleksie petrov	UY320014	2021-01-03	5
Mehmet yildiz	QW48511	2021-01-06	1
James angelinton	OI214500	2021-01-05	1
Jammy motinge	QW478811	2021-01-01	2
Ahmad elmasri	LP784330	2021-01-02	1

5.2.c) Using 4 Tables

Bir şirketin uçaklarının, günlere göre kaç uçuş yaptığını listeler

```
SELECT flight_date, COUNT(*)
FROM COMPANY, AIRLINE, FLIGHT, LEG_INSTANCE
WHERE COMPANY.company_name='Turkish airlines a.s'
AND COMPANY.Company id=AIRLINE.Company id
AND AIRLINE.Airline_name=FLIGHT.Airline_name
AND FLIGHT.Flight_number=LEG_INSTANCE.Flight_number
GROUP BY flight_date;
```



Result Grid | Filter Rows: | E

flight_date	COUNT(*)
2021-01-01	2
2021-01-05	1
2021-01-06	1

Herhangi bir uçağın id sini, yolcu kapasitesini, ait oldugu şirketin ismini ve inebileceği havalimanının ismini listeler.

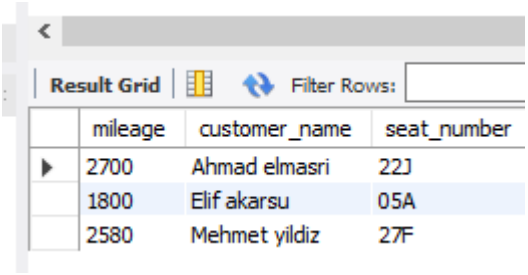
```
select airplane.airplane id,company.company name,airplane_type.max_seats,airport.airport_name
from company,airplane,airplane_type,can_land,airport
where company.company_id=airplane.company_id and
airplane.airplane_type_name=airplane_type.airplane_type_name and
airplane_type.airplane_type_name=can_land.airplane_type_name and
can_land.airport_code=airport.airport_code
group by airplane.airplane_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conter

airplane_id	company_name	max_seats	airport_name
Bre-3112	Airbus Se	525	Dubai airport
Fk-221	The beoeing company	368	Dubai airport
Hy-9841	The beoeing company	215	Dubai airport
Jx-1232	Airbus Se	186	Izmir adnan meneres
Kgx-8704	The beoeing company	368	Dubai airport
Lbj-0023	The beoeing company	366	Dubai airport
Ol-7410	The beoeing company	366	Dubai airport
Pl-9630	The beoeing company	368	Dubai airport
Tk-0007	Airbus Se	186	Izmir adnan meneres
Un-3102	Airbus Se	186	Izmir adnan meneres
Yn-023	Airbus Se	133	Izmir adnan meneres

Retreive cosutmer names and its seat number who have earn more than 1000 mileage

```
select distinct s.mileage , c.customer_name, se.seat_number
FROM single_ffc as s , customer as c , seat_reservation as se , ffc
where s.mileage > 1000
and ffc.passport_no= c.passport_no
and s.passport_no = ffc.passport_no
and se.passport_no = c.passport_no
group by c.customer_name
;
```



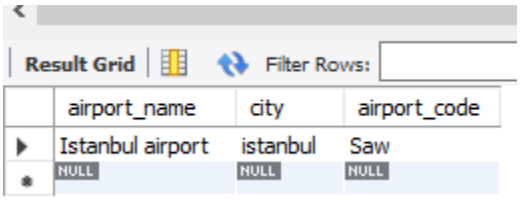
Result Grid | Filter Rows:

mileage	customer_name	seat_number
2700	Ahmad elmasri	22J
1800	Elif akarsu	05A
2580	Mehmet yildiz	27F

5.3) SELECT statements to exemplify nested and/or correlated nested queries

nested query
uçuş ayağı 0 ve kapasitesi 150 den az olan uçakların ismini, havaalanının şehrini ve kodunu listeler.

```
select airport.airport_name,airport.city,airport.airport_code
from airport
where airport.airport_code in
(select leg_instance.dep_airport_code
from leg_instance
where leg_instance.leg_num=0 and leg_instance.airplane_id in
(select airplane.airplane_id
from airplane
where airplane.total_num_of_seats<150));
```

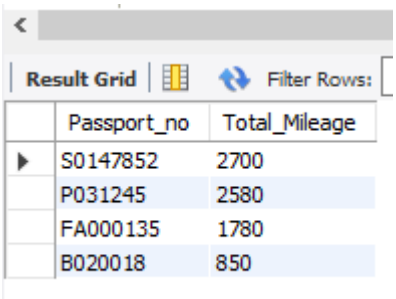


Result Grid | Filter Rows:

airport_name	city	airport_code
Istanbul airport	istanbul	Saw
NULL	NULL	NULL

nested or correlated
Fiyatı 1500 den yüksek ve Economy class olan uçuşların müşterilerin total mileagelerini yüksekten düşüğe sıralar

```
SELECT DISTINCT FFC.Passport_no, FFC.Total_Mileage
FROM SINGLE_FFC AS S, FFC
WHERE S.Passport_no=FFC.passport_no AND S.Flight_number IN (SELECT Flight_number
FROM FARE AS F
WHERE F.Restriction='Economy class' AND Amount>=1500)
ORDER BY Total_mileage desc;
```

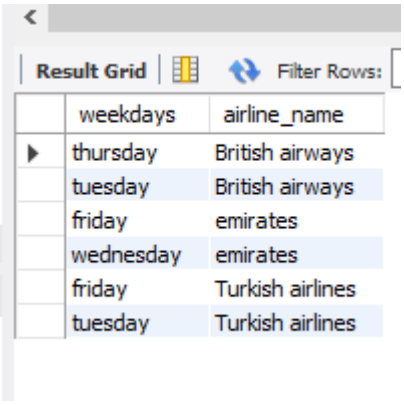


Result Grid | Filter Rows:

Passport_no	Total_Mileage
S0147852	2700
P031245	2580
FA000135	1780
B020018	850

nested queries
bilet fiyati 700 dan fazla olan uçuşların hangi havayoluyla ve hangi günler uçuş yaptığını yazan sorgu.

```
select flight.weekdays,flight.airline_name
From flight
where flight.flight_number in (
        select fare.flight_number
        from fare
        where amount > 700
);
```

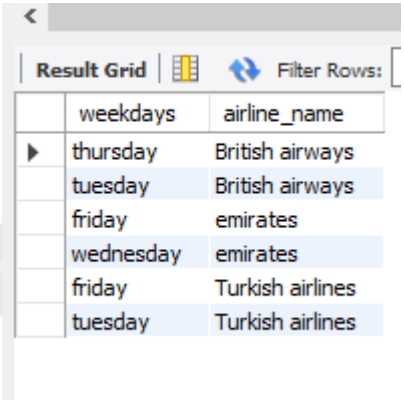


Result Grid | Filter Rows:

weekdays	airline_name
thursday	British airways
tuesday	British airways
friday	emirates
wednesday	emirates
friday	Turkish airlines
tuesday	Turkish airlines

nested queries
bilet fiyati 700 dan fazla olan uçuşların hangi havayoluyla ve hangi günler uçuş yaptığını yazan sorgu.

```
select flight.weekdays,flight.airline_name
From flight
where flight.flight_number in (
        select fare.flight_number
        from fare
        where amount > 700

);
```



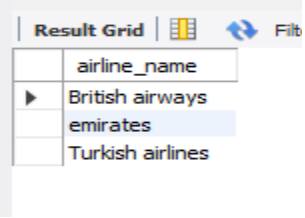
Result Grid | Filter Rows:

weekdays	airline_name
thursday	British airways
tuesday	British airways
friday	emirates
wednesday	emirates
friday	Turkish airlines
tuesday	Turkish airlines

5.4) SELECT statements to exemplify EXISTS and NOT EXISTS statements

hem exists hem correlated
birden fazla aktarma yapan uçuşların hangi firmaya ait olduğunu gösteren sorgu

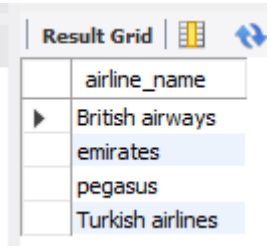
```
select flight.airline_name
from flight
where exists (
        select flight_number
        from flight_leg
        where leg_num > 1
    and flight.flight_number = flight_leg.flight_number


);
```



airline_name
British airways
emirates
Turkish airlines

/*leg numberı 2 ve 2 den düşük ve uçuş numarası Ek202 olmayan uçuş numaralarının hava yolu adı */

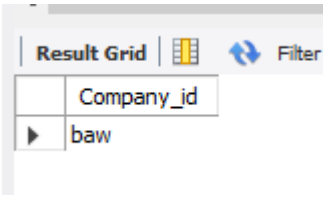
```
select distinct flight.airline_name
from flight
where not exists (
        select flight_number
        from flight_leg
        where leg_number > 2 and Flight_number='Ek202'
    and flight.flight_number = flight_leg.flight_number
);
```



airline_name
British airways
emirates
pegasus
Turkish airlines

Uçuş numarası bilinen bir uçuşun şirket idsine ulaşmamızı sağlar

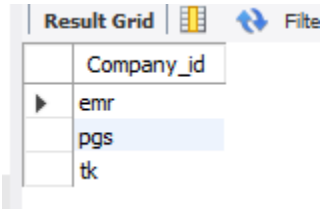
```
SELECT A.Company_id
FROM AIRLINE AS A
WHERE EXISTS(
        SELECT Airline_name
        FROM FLIGHT AS F
        WHERE A.Airline_name=F.Airline_name AND Flight_number='Ba0104');
```



Company_id
baw

Uçuş numarası bilinen bir uçuşun dışındaki şirket idsine ulaşmamızı sağlar

```
SELECT A.Company_id
FROM AIRLINE AS A
WHERE NOT EXISTS(
        SELECT Airline_name
        FROM FLIGHT AS F
        WHERE A.Airline_name=F.Airline_name AND Flight_number='Ba0104');
```



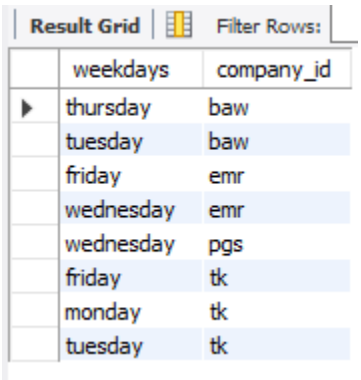
Result Grid | Filte

Company_id
emr
pgs
tk

5.5) SELECT statements to exemplify LEFT, RIGHT and FULL OUTER JOIN statements

Full outer join
uçağın uçuşun gününü ve uçağın ait olduğu şirketin id sini listeler.

```
(select flight.weekdays,airline.company_id
from flight left outer join airline on flight.airline_name=airline.airline_name)
union
(select flight.weekdays,airline.company_id
from flight right outer join airline on flight.airline_name=airline.airline_name);
```

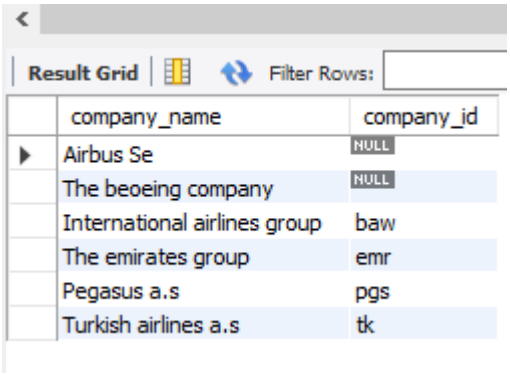


Result Grid | Filter Rows:

weekdays	company_id
thursday	baw
tuesday	baw
friday	emr
wednesday	emr
wednesday	pgs
friday	tk
monday	tk
tuesday	tk

Left outer join
herhangi bir şirketin ismi ve sahip oldugu havayolunun şirket id si

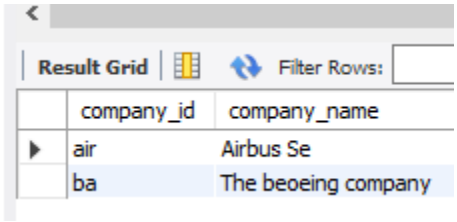
```
select company.company_name,airline.company_id
from company left outer join airline on company.company_id=airline.company_id;
```



company_name	company_id
Airbus Se	NULL
The beoeing company	NULL
International airlines group	baw
The emirates group	emr
Pegasus a.s	pgs
Turkish airlines a.s	tk

Right outer join
herhangi bir şirketin ismi ve sahip olduğu uçağın şirket ismi

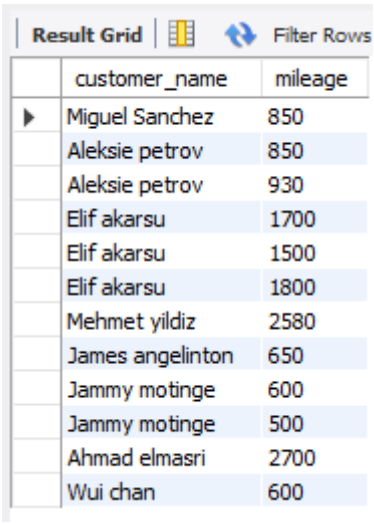
```
select distinct airplane.company_id,company.company_name
from company right outer join airplane on company.company_id=airplane.company_id;
```



company_id	company_name
air	Airbus Se
ba	The beoeing company

full outer join

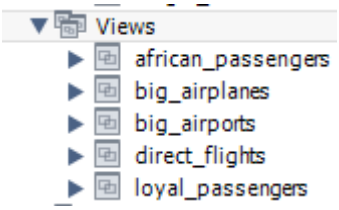
```
select customer.customer_name, single_ffc.mileage
from customer
left join single_ffc
on customer.passport_no = single_ffc.passport_no
union
select customer.customer_name, single_ffc.mileage
from customer
right join single_ffc
on customer.passport_no = single_ffc.passport_no;
```



customer_name	mileage
Miguel Sanchez	850
Aleksie petrov	850
Aleksie petrov	930
Elif akarsu	1700
Elif akarsu	1500
Elif akarsu	1800
Mehmet yildiz	2580
James angelinton	650
Jammy motinge	600
Jammy motinge	500
Ahmad elmasri	2700
Wui chan	600

5.6)REASONABLE Views

All View Tables:



view - airports that big airplanes can land
Max koltuk sayısı 500 den yüksek olan uçakların Airport codelarını tutan tablo

```
create view big_airports as
select c.airport_code
FROM  can_land as c, airplane_type as ait
where ait.max_seats  > 500
and ait.airplane_type_name = c.airplane_type_name;

select * from  big_airports;
```

Result Grid

airport_code
DXB
IST
JFK

view - big airplanes can land
Max koltuk sayısı 299 dan yüksek olan uçakların uçak tiplerini tutan tablo

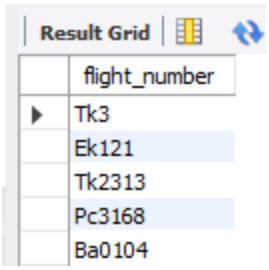
```
create view big_airplanes as
select airplane_type_name
from airplane_type
where max_seats > 299;

select * from big_airplanes;
```

airplane_type_name
A380
B747
B777
B787

view - big airplanes can land
Tek seferde ulaşım sağlayan uçakların nunmarasını tutan tablo

```
create view Direct_flights as
select flight_number
```

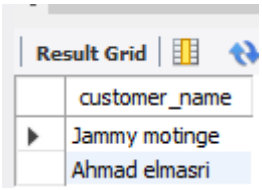
```
from flight_leg
where leg_num = 0;

select * from Direct_flights;
```



flight_number
Tk3
Ek121
Tk2313
Pc3168
Ba0104

view - passengers from africa continent
Afrikalı müşterilerin adını tutan tablo

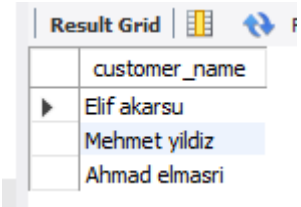
```
create view african_passengers as
select customer_name
from Customer
where country in ('egypt', 'somali');
select * from african_passengers;
```



customer_name
Jammy motinge
Ahmad elmasri

view - passengers who got loyal degree with having mileage more than 2500

```
create view loyal_passengers as
select c.customer_name
from customer as c, ffc as f
where c.passport_no = f.passport_no
and f.total_mileage > 2500;
select * from loyal_passengers;
```



customer_name
Elif akarsu
Mehmet yildiz
Ahmad elmasri

6) PART – III APPLICATION DEVELOPMENT
6.1) Propose a customer segmentation

we have used a segmentation model as below to classify our frequent flyer customers according to their Airlines and bonus Miles they received while flying.

Every Airline has their own policy and program regarding their customers. For every mileage points different companies have different policies.
For instance,
Turkish airlines use Miles & mile program for their customers.

- passenger with miles points more than 600 are classified as – Classic.
- passenger with miles points more than 2500 are classified as – Classic Plus.
- passenger with miles points more than 4000 are classified as – Elite.
- passenger with miles points more than 10000 are classified as – Elite Plus.

Pegasus has BOLBOL program.

- passenger with miles points more than 500 are classified as – economy.
- passenger with miles points more than 3000 are classified as – economy Plus.
- passenger with miles points more than 1500 are classified as – business.
- passenger with miles points more than 5000 are classified as – business Plus

EMIRATES has Skyward program.

- passenger with miles points more than 1000 are classified as – bronze.
- passenger with miles points more than 2500 are classified as – bronze Plus.
- passenger with miles points more than 3000 are classified as – silver.
- passenger with miles points more than 4500 are classified as – gold

British Airways has Executive club program.

- passenger with miles points more than 600 are classified as – economy.
- passenger with miles points more than 2500 are classified as – economy Plus.
- passenger with miles points more than 3000 are classified as – business.
- passenger with miles points more than 5000 are classified as – business plus.

CUSTOMER SEGMENTATION MODEL

Customer name	Airlines	program	Reward mileage	segment
James angelinton	Turkish airlines	Miles & miles	650	Classic
Mehmet yildiz	Turkish airlines	Miles & miles	2580	Classic Plus
Ahmad elmasri	emirates	Skyward	2700	Bronze Plus
Wui chan	pegasus	BOLBOL	600	economy
Miguel Sanchez	British airways	Executive club	850	Economy
Aleksie petrov	British airways	Executive club	1780	Economy
Elif akarsu	Emirates	Skyward	5000	Gold
Jammy motinge	Turkish airlines	Miles & miles	1100	Classic

6.2) Develop a customer segmentation

6.2.a) Source Code

App.py

```
from flask import Flask
app = Flask(__name__)
```

dp.py

```
from app import app
from flaskext.mysql import MySQL
mysql=MySQL()

app.config['MYSQL_DATABASE_USER'] = 'root'
app.config['MYSQL_DATABASE_PASSWORD'] = '12345'
app.config['MYSQL_DATABASE_DB'] = 'airline'
app.config['MYSQL_DATABASE_HOST'] = 'localhost'
mysql.init_app(app)
```

MySQL ile Python'I birleştirmek için flask kullanıldı.

Main.py

```
import pymysql
from app import app
from db import mysql
import pymysql
from flask import jsonify
from flask import flash, request

@app.route('/')
def page():
    return message

@app.route('/customer')
def customer():
    try:
        conn = mysql.connect()
        cursor = conn.cursor(pymysql.cursors.DictCursor)
        cursor.execute("SELECT * FROM CUSTOMER")
        rows = cursor.fetchall()
        resp = jsonify(rows)
        resp.status_code = 200
```

```
        return resp
    except Exception as e:
        print(e)
    finally:
        cursor.close()
        conn.close()
@app.errorhandler(404)
def not_found(error=None):
    message = {
        'status': 404,
        'message': 'Not Found: ' + request.url,
    }
    resp = jsonify(message)
    resp.status_code = 404
    return resp
conn = mysql.connect()
cursor = conn.cursor()
cursor.execute("SELECT * FROM CUSTOMER")
rows = cursor.fetchall()
p=[]
msg=""
for j in rows:
    msg+="<tr><td>"+str(j[0])+"</td><td>"+str(j[1])+"</td><td>"+str(j[2])+"</t
d><td>"+str(j[3])+"</td><td>"+str(j[4])+"</td><td>"+str(j[5])+"</td><td>"
    sql="SELECT Total_mileage FROM FFC WHERE Passport_no = %s"
    cursor.execute(sql,j[0])
    level=cursor.fetchone()
    msg+=str(level[0])+"</td><td>"
    if(int(level[0])>=5000):
        l="Gold"
    elif(int(level[0])>=2500):
        l="Silver"
    else:
        l="Bronze"
    msg+=l+"</td></tr>"
message = """<html>
<head></head>"""+"""<body><table border=1><tr><td><b><font size="5">Passport_n
o</font></b></td>
<td><b><font size="5">Address</font></b></td>
<td><b><font size="5">Country</font></b></td>
<td><b><font size="5">E_mail</font></b></td>
<td><b><font size="5">Customer_name</font></b></td>
<td><b><font size="5">Customer_phone</font></b></td>
<td><b><font size="5">Total_mileage</font></b></td>
<td><b><font size="5">Level</font></b></td></tr>"""+msg+"""</table></body>
</html>"""

if __name__ == "__main__":
    app.run(debug=True)
```

Main classının ilk başında .json ve html görünümünü sağlamak için fonksiyonlar oluşturuldu.

Customer tablosundaki veriler bir diziye atıldı. Sonrasında FFC içerisindeki gerekli bilgilerde alındı ve html görünümü oluşturmak için hazırlanan stringin gerekli yerlerine atandı.

Alınan FFC bilgilerine göre müşterinin total_mileage i 5.000 mil puanından yüksek ise 'Gold', 2.500 mil puanından yüksek ise 'Silver' ve daha düşük mil puanlarında ise 'Bronze' olarak sınıflanırıldı.

6.2.b)Screenshot

-html

Passport_no	Address	Country	E_mail	Customer_name	Customer_phone	Total_mileage	Level
B020018	0123 street no:3 Madrid	Spain	Yhet@gmail.com	Miguel Sanchez	458884441110	850	Bronze
FA000135	0155 street no:4 St. Petersburg	Russia	Geresho@mail.ru	Aleksie petrov	721000033355	1780	Bronze
K8745513	12 street no: 9 mithatpasa , istanbul	Turkey	Elifakarsu@gmail.com	Elif akarsu	905539540817	5000	Gold
P031245	4023 street ,NO 5, erzene , Bornova , Izmir	Turkey	M35@gmail.com	Mehmet yildiz	905553332222	2580	Silver
R123456	59 street, No 10 ,D 5 ,Beverly hills , los Angeles	USA	Jj13@gmail.com	James angelinton	1222333555444	650	Bronze
S0001325	45 street no:9 mogadishu	Somali	motinge@hotmail.com	Jammy motinge	876662225544	1100	Bronze
S0147852	894 street NO: 3 Cairo	Egypt	Buffer43@yahoo.com	Ahmad elmasri	210006665551	2700	Silver
V025874	321 street no 17 Wuhan	China	Xindex32@hotmail.com	Wui chan	310005556661	600	Bronze

-.json


```
[
  {
    "Address": "0123 street no:3 Madrid",
    "Country": "Spain",
    "Customer_name": "Miguel Sanchez",
    "Customer_phone": 458884441110,
    "Email": "Yhet@gmail.com",
    "Passport_no": "B020018"
  },
  {
    "Address": "0155 street no:4 St. Petersburg",
    "Country": "Russia",
    "Customer_name": "Aleksie petrov",
    "Customer_phone": 721000033355,
    "Email": "Geresho@mail.ru",
    "Passport_no": "FA000135"
  },
  {
    "Address": "12 street no: 9 mithatpasa , istanbul",
    "Country": "Turkey",
    "Customer_name": "Elif akarsu",
    "Customer_phone": 905539540817,
    "Email": "Elifakarsu@gmail.com",
    "Passport_no": "K8745513"
  },
  {
    "Address": "4023 street ,NO 5, erzene , Bornova , Izmir",
    "Country": "Turkey",
    "Customer_name": "Mehmet yildiz",
    "Customer_phone": 905553332222,
    "Email": "M35@gmail.com",
    "Passport_no": "P031245"
  },
  {
    "Address": "59 street, No 10 ,D 5 ,Beverly hills , los Angeles",
    "Country": "USA",
    "Customer_name": "James angelinton",
    "Customer_phone": 1222333555444,
    "Email": "Jj13@gmail.com",
    "Passport_no": "R123456"
  },
```

7)Min-Max Notation