AIRLINE DATABASE PROJECT

PROJECT SCOPE: The Airline database stores details about airline's airports, flights, passengers, captains, airlines and seat bookings.

Some important notes:

- Airline Companies may have one or more planes.
- Airplanes have model, manufacturer and number specify that how many of this model actively used in the world.
- Passenger have name, surname, nationality, age, gender and passport number which is **unique** for all persons.
- There is two kind of airport one of that departure and another is arrival airport and airports have country, total passenger and elevation (feet) information.
- Airline companies have a name which is unique and total passenger and country information.
- Flight number is unique for all flights and it has departure and arrival date and airport information.
- There is a payment and payment id is unique, it has payment type, date and amount information. With one payment one or more ticket can be taken.
- In ticket you can find these information: Ticket Id which is not unique some different airlines may have same ticket id because of that it is not a candidate key, flight number, passenger nationality number, payment id, flight class, passenger seat and bag id.
- There is a bag information also it has id and weight information. Passenger may have separate bags but all bags should label with same id.
- Each flight is carried out by a single airplane.
- A passenger can book just one seat at airplane.

Relationships and some assumptions:

- One captain can only work in one airplane company but airplane companies may have one or more captains.
- One captain may have different flights but in one flight there is only one captain.
- Airplane belongs to one airline but an airline may have one or more airplane.
- A passenger may have different flight with different airplanes.
- In one flight and there is a one airplane and there is a one or more passenger.
- **Flight without passenger would not possible.**
- One passenger have one bag id and one bag id belongs to one person.
- A passenger may have one or more ticket but one ticket must belong to one passenger.
- In ticket there is only one bag id and one bag id belongs to one ticket.
- One passenger may have different payments, but one payment should belong one passenger.
- For ticket there is only one payment id, but with one payment passenger can take one or more tickets, for example: for their child's or parents.
- One flight must belong to just one airline, one airline may have one or more flights.
- Arrival and departure airports may have different flights but one flight have only one arrival airport and one departure airport.

Primary key = {Flight number, Passenger passport number}

Primary key selected as above because:

Flight number and passenger passport number is unique, in this database there is only one row with this combination.

So, flight without passenger is not possible because primary key column cannot be null.

For our project python is used to create tables for database and sending information as well.

And again python is used to inserting new row to database, deleting data and sending queries for selecting and filtering data.

In **airlinedatabase.py** you can find the codes for creating database and sending values to database.

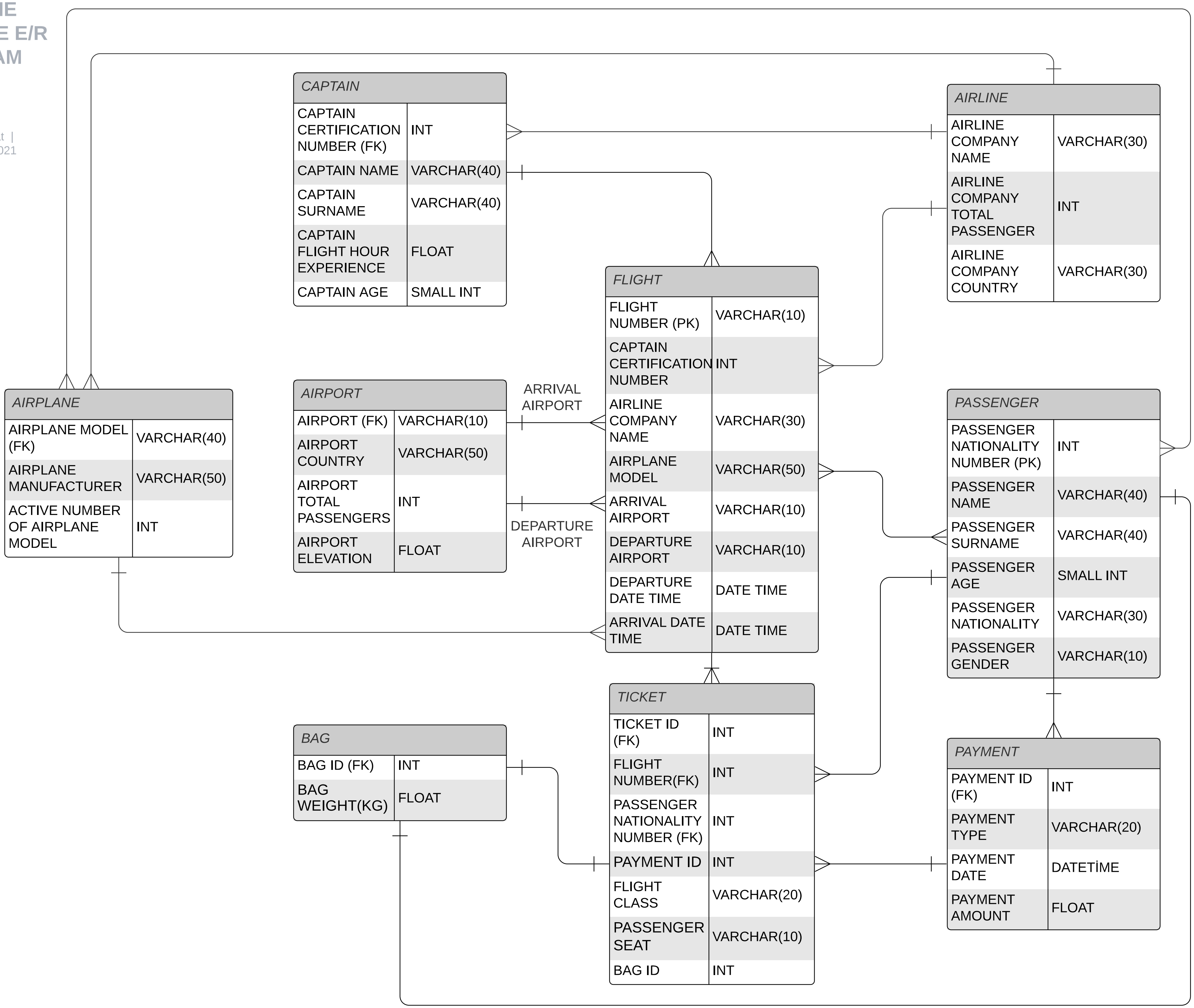
With **application.py** you can insert new row to database, delete data and send queries for selecting and filtering data.

In application.py for selecting and filtering data excel file is created automatically in your local folder to see the results.

You can create this database with **airlinedatabase SQL dump** as well.

AIRLINE DATABASE E/R DIAGRAM

Bulut Murat | January 1,2021



CAPTAIN	
CAPTAIN CERTIFICATION NUMBER (FK)	INT
CAPTAIN NAME	VARCHAR(40)
CAPTAIN SURNAME	VARCHAR(40)
CAPTAIN FLIGHT HOUR EXPERIENCE	FLOAT
CAPTAIN AGE	SMALL INT

AIRLINE	
AIRLINE COMPANY NAME	VARCHAR(30)
AIRLINE COMPANY TOTAL PASSENGER	INT
AIRLINE COMPANY COUNTRY	VARCHAR(30)

FLIGHT	
FLIGHT NUMBER (PK)	VARCHAR(10)
CAPTAIN CERTIFICATION NUMBER	INT
AIRLINE COMPANY NAME	VARCHAR(30)
AIRPLANE MODEL	VARCHAR(50)
ARRIVAL AIRPORT	VARCHAR(10)
DEPARTURE AIRPORT	VARCHAR(10)
DEPARTURE DATE TIME	DATE TIME
ARRIVAL DATE TIME	DATE TIME

AIRPLANE	
AIRPLANE MODEL (FK)	VARCHAR(40)
AIRPLANE MANUFACTURER	VARCHAR(50)
ACTIVE NUMBER OF AIRPLANE MODEL	INT

AIRPORT	
AIRPORT (FK)	VARCHAR(10)
AIRPORT COUNTRY	VARCHAR(50)
AIRPORT TOTAL PASSENGERS	INT
AIRPORT ELEVATION	FLOAT

PASSENGER	
PASSENGER NATIONALITY NUMBER (PK)	INT
PASSENGER NAME	VARCHAR(40)
PASSENGER SURNAME	VARCHAR(40)
PASSENGER AGE	SMALL INT
PASSENGER NATIONALITY	VARCHAR(30)
PASSENGER GENDER	VARCHAR(10)

TICKET	
TICKET ID (FK)	INT
FLIGHT NUMBER(FK)	INT
PASSENGER NATIONALITY NUMBER (FK)	INT
PAYMENT ID	INT
FLIGHT CLASS	VARCHAR(20)
PASSENGER SEAT	VARCHAR(10)
BAG ID	INT

BAG	
BAG ID (FK)	INT
BAG WEIGHT(KG)	FLOAT

PAYMENT	
PAYMENT ID (FK)	INT
PAYMENT TYPE	VARCHAR(20)
PAYMENT DATE	DATETİME
PAYMENT AMOUNT	FLOAT

Airlines Database Project

This Airline Database is aimed to obtain flight information, passenger information, captain information, airline, airport and plane information.

The information of the column that the database will have is as follows:

R = { Flight Number, Passenger Passport Number, Ticket ID, Arrival Airport, Departure Airport, Departure Date Time, Arrival Date Time, Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience, Passenger Name, Passenger Surname, Passenger Age, Passenger Gender, Passenger Nationality, Departure Airport Country, Departure Airport Total Passengers, Departure Airport Elevation (ft), Arrival Airport Country, Arrival Airport Total Passengers, Arrival Airport Elevation (ft), Flight Class, Bag ID, Bag Weight(kg), Passenger Seat, Payment ID, Payment Type, Payment Date, Payment Amount, Airplane Model, Airplane Manufacturer, Active number of Airplane Model, Airline Company Name, Airline Company Total Passenger, Airline Company Country }

- let B be the set of all attributes of the relation, and let A ⊆ B
- to be a candidate key, A has to be:
- unique: no two tuples have the same values for all attributes in A
- irreducible: no subset of A is unique
- every relation has at least one candidate key

Ticket ID is not considered as a candidate key because some airlines may have same ticket ID.

Candidate Key = {Passenger Passport Number, Flight Number} (Composite Key)

Primary Key = {Passenger Passport Number, Flight Number} (Composite Key)

- let Z be the set of all attributes of the relation R
- let A,B ⊆ Z
- **A functionally determines B**: A → B
  for every A value there can only be one B value
- every functional dependency is an integrity constraint

In my Project Functional Dependencies are:

Functional Dependencies Step 1.

A	**→**	**B**
Captain Certification Number	→	{ Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience }
Passenger Passport Number	→	{ Passenger Name, Passenger Surname, Passenger Age, Passenger Gender, Passenger Nationality }
Flight Number	→	{ Captain Certification Number, Arrival Airport, Departure Airport, Departure Date Time, Arrival Date Time, Airline Company Name, Airplane Model }
Departure Airport	→	{ Departure Airport Country, Departure Airport Total Passengers, Departure Airport Elevation (ft) }
Arrival Airport	→	{ Arrival Airport Country, Arrival Airport Total Passengers, Arrival Airport Elevation (ft) }
{ Flight Number, Passenger Passport Number }	→	Ticket ID
Ticket ID	→	{ Flight Class, Passenger Seat, Bag ID, Payment ID }
Airplane Model	→	{ Airplane Manufacturer, Active number of Airplane Model}
Airline Company Name	→	{ Airline Company Total Passenger, Airline Company Country }
Bag ID	→	Bag Weight(kg)
Payment ID	→	{ Payment Type, Payment Date, Payment Amount }

In my Project Functional Dependencies are:

Functional Dependencies (Irreducible Sets)

A	→	**B**
Captain Certification Number	→	Captain Name
Captain Certification Number	→	Captain Surname
Captain Certification Number	→	Captain Age
Captain Certification Number	→	Captain Flight Hour Experience
Passenger Passport Number	→	Passenger Name
Passenger Passport Number	→	Passenger Surname
Passenger Passport Number	→	Passenger Age
Passenger Passport Number	→	Passenger Gender
Passenger Passport Number	→	Passenger Nationality
Flight Number	→	Captain Certification Number
Flight Number	→	Arrival Airport
Flight Number	→	Departure Airport
Flight Number	→	Departure Date Time
Flight Number	→	Arrival Date Time
Flight Number	→	Airline Company Name

Flight Number	→	Airplane Model
Departure Airport	→	Departure Airport Country
Departure Airport	→	Departure Airport Total Passengers
Departure Airport	→	Departure Airport Elevation (ft)
Arrival Airport	→	Arrival Airport Country
Arrival Airport	→	Arrival Airport Total Passengers
Arrival Airport	→	Arrival Airport Elevation (ft)
{ Flight Number, Passenger Passport Number }	→	Ticket ID
Ticket ID	→	Flight Class
Ticket ID	→	Passenger Seat
Ticket ID	→	Bag ID
Ticket ID	→	Payment ID
Airplane Model	→	Airplane Manufacturer
Airplane Model	→	Active number of Airplane Model
Airline Company Name	→	Airline Company Total Passenger
Airline Company Name	→	Airline Company Country
Bag ID	→	Bag Weight(kg)
Payment ID	→	Payment Type
Payment ID	→	Payment Date
Payment ID	→	Payment Amount

NORMALIZATION

1NF: attribute values have to be atomic

Simplifying assumptions for definitions: A Table has only one candidate key, which is also primary key.

It is a relation and attributes are atomic and have a primary key so our table is now 1NF. It can be also seen by simplifying assumptions because we have just one candidate key and it is also primary key.

Primary Key = {Passenger Passport Number, Flight Number} (Composite Key)

1NF to 2NF

- 2NF: every non-key attribute depends on the primary key
- in a relation R that conforms to 1NF, if:
  - ✓ R(A,B, C,D), primary key: (A,B)
  - ✓ A → D
- to transform to 2NF:
  - ✓ R1(A,D), primary key: A
  - ✓ R2(A,B, C), primary key: (A,B)
  - ✓ A is a foreign key referencing R1

Because we have non-composite primary key so every non-key attribute depends on the primary key and we have already 2NF.

R = { Flight Number, Passenger Passport Number, Ticket ID, Arrival Airport, Departure Airport, Departure Date Time, Arrival Date Time, Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience, Passenger Name, Passenger Surname, Passenger Age, Passenger Gender, Passenger Nationality, Departure Airport Country, Departure Airport Total Passengers, Departure Airport Elevation (ft), Arrival Airport Country, Arrival Airport Total Passengers, Arrival Airport Elevation (ft), Flight Class, Bag ID, Bag Weight(kg), Passenger Seat, Payment ID, Payment Type, Payment Date, Payment Amount, Airplane Model, Airplane Manufacturer, Active number of Airplane Model, Airline Company Name, Airline Company Total Passenger, Airline Company Country }

1NF to 2NF

- 2NF: every non-key attribute depends on the primary key
- in a relation R that conforms to 1NF, if:
  - ✓ R(A,B, C,D), primary key: (A,B)
  - ✓ A → D
- to transform to 2NF:
  - ✓ R1(A,D), primary key: A
  - ✓ R2(A,B, C), primary key: (A,B)
  - ✓ A is a foreign key referencing R1

A: Flight Number

B: Passenger Passport Number

D: { Captain Certification Number, Arrival Airport, Departure Airport, Departure Date Time, Arrival Date Time, Airline Company Name, Airplane Model }

C: Other attributes

So:

> R1 = {A,D} = { Flight Number, Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience, Arrival Airport, Arrival Airport Country, Arrival Airport Total Passengers, Arrival Airport Elevation (ft), Departure Airport, Departure Airport Country, Departure Airport Total Passengers, Departure Airport Elevation (ft), Departure Date Time, Arrival Date Time, Airline Company Name, Airline Company Total Passenger, Airline Company Country, Airplane Model, Airplane Manufacturer, Active number of Airplane Model }
>
> R2 = {A,B,C} = { Flight Number, Passenger Passport Number, Ticket ID, Passenger Name, Passenger Surname, Passenger Age, Passenger Gender, Passenger Nationality, Flight Class, Bag ID, Bag Weight(kg), Passenger Seat, Payment ID, Payment Type, Payment Date, Payment Amount }
>
> Flight Number is a foreign key referencing R1

Now R1 is 2NF but R2 not because some non-key attribute does not depend on the primary key

For R2:

A: Passenger Passport Number

B: Flight Number

D: { Passenger Name, Passenger Surname, Passenger Age, Passenger Gender, Passenger Nationality }

C: Other attributes

So:

> **R3 = {A,D} = { Passenger Passport Number, Passenger Name, Passenger Surname, Passenger Age, Passenger Gender, Passenger Nationality }**
>
> R4 = {A,B,C} = { Flight Number, Passenger Passport Number, Ticket ID, Flight Class, Bag ID, Bag Weight(kg), Passenger Seat, Payment ID, Payment Type, Payment Date, Payment Amount }
>
> Passenger Passport Number is a foreign key referencing R3

R3 and R4 are 2NF.

2NF to 3NF

- 3NF: non-key attributes do not depend on any attributes other than the primary key
- in a relation R that conforms to 2NF, if:
  - ✓ R(A,B, C,D), primary key: A
  - ✓ C → D
- to transform to 3NF:
  - ✓ R1(C,D), primary key: C
  - ✓ R2(A,B, C), primary key: A
  - ✓ C is a foreign key referencing R1

R1 is not a 3NF:

For R1:

A: Flight Number

B: { Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience, Departure Airport, Departure Airport Country, Departure Airport Total Passengers, Departure Airport Elevation (ft), Departure Date Time, Arrival Date Time, Airline Company Name, Airline Company Total Passenger, Airline Company Country, Airplane Model, Airplane Manufacturer, Active number of Airplane Model }

C: Arrival Airport

D: { Arrival Airport Country, Arrival Airport Total Passengers, Arrival Airport Elevation (ft) }

So:

> **R5 = {C, D} = { Arrival Airport, Arrival Airport Country, Arrival Airport Total Passengers, Arrival Airport Elevation (ft) }**
>
> R6 = {A, B, C} = { Flight Number, Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience, Departure Airport, Departure Airport Country, Departure Airport Total Passengers, Departure Airport Elevation (ft), Departure Date Time, Arrival Date Time, Airline Company Name, Airline Company Total Passenger, Airline Company Country, Airplane Model, Airplane Manufacturer, Active number of Airplane Model, Arrival Airport }
>
> Arrival Airport is a foreign key referencing R5

R5 is 3NF but R6 not:

For R6:

A: Flight Number

B: { Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience, Departure Date Time, Arrival Date Time, Airline Company Name, Airline Company Total Passenger, Airline Company Country, Airplane Model, Airplane Manufacturer, Active number of Airplane Model, Arrival Airport }

C: Departure Airport

D: { Departure Airport Country, Departure Airport Total Passengers, Departure Airport Elevation (ft) }

R7 = {C, D} = { Departure Airport, Departure Airport Country, Departure Airport Total Passengers, Departure Airport Elevation (ft) }

R8 = {A, B, C} = {Flight Number, Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience, Departure Date Time, Arrival Date Time, Airline Company Name, Airline Company Total Passenger, Airline Company Country, Airplane Model, Airplane Manufacturer, Active number of Airplane Model, Arrival Airport, Departure Airport}

Departure Airport is a foreign key referencing R7

R7 is 3NF but R8 not:

For R8:

A: Flight Number

B: { Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience, Departure Date Time, Arrival Date Time, Airplane Model, Airplane Manufacturer, Active number of Airplane Model, Arrival Airport, Departure Airport }

C: Airline Company Name

D: { Airline Company Total Passenger, Airline Company Country }

R9 = {C, D} = { Airline Company Name, Airline Company Total Passenger, Airline Company Country }

R10 = {A, B, C} = {Flight Number, Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience, Departure Date Time, Arrival Date Time, Airplane Model, Airplane Manufacturer, Active number of Airplane Model, Arrival Airport, Departure Airport, Airline Company Name }

Airline Company Name is a foreign key referencing R9

R9 is 3NF but R10 not:

For R10:

A: Flight Number

B: { Departure Date Time, Arrival Date Time, Airplane Model, Airplane Manufacturer, Active number of Airplane Model, Arrival Airport, Departure Airport, Airline Company Name }

C: Captain Certification Number

D: { Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience }

R11 = {C, D} = { Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience }

R12 = {A, B, C} = {Flight Number, Departure Date Time, Arrival Date Time, Airplane Model, Airplane Manufacturer, Active number of Airplane Model, Arrival Airport, Departure Airport, Airline Company Name, Captain Certification Number}

Captain Certification Number is a foreign key referencing R11

R11 is 3NF but R12 not:

For R12:

A: Flight Number

B: { Departure Date Time, Arrival Date Time, Arrival Airport, Departure Airport, Airline Company Name, Captain Certification Number }

C: Airplane Model

D: { Airplane Manufacturer, Active number of Airplane Model }

R13 = {C, D} = { Airplane Model, Airplane Manufacturer, Active number of Airplane Model }

R14 = {A, B, C} = {Flight Number, Departure Date Time, Arrival Date Time, Arrival Airport, Departure Airport, Airline Company Name, Captain Certification Number, Airplane Model }

Airplane Model Number is a foreign key referencing R13

R13 and R14 are both 3NF.

For R3 and R4:

R3 = {A,D} = { Passenger Passport Number, Passenger Name, Passenger Surname, Passenger Age, Passenger Gender, Passenger Nationality }

R4 = {A,B,C} = { Flight Number, Passenger Passport Number, Ticket ID, Flight Class, Bag ID, Bag Weight(kg), Passenger Seat, Payment ID, Payment Type, Payment Date, Payment Amount }

R3 is 3NF but R4 not:

For R4:

A: {Flight Number, Passenger Passport Number}

B: -

C: Ticket ID

D: { Flight Class, Bag ID, Bag Weight(kg), Passenger Seat, Payment ID, Payment Type, Payment Date, Payment Amount }

R15 = {C, D} = { Ticket ID, Flight Class, Bag ID, Bag Weight(kg), Passenger Seat, Payment ID, Payment Type, Payment Date, Payment Amount }

R16 = {A, B, C} = {Flight Number, Passenger Passport Number, Ticket ID }

Ticket ID is a foreign key referencing R15

R16 is 3NF but R15 not:

For R15:

A: Ticket ID

B: { Flight Class, Bag ID, Bag Weight(kg), Passenger Seat }

C: Payment ID

D: { Payment Type, Payment Date, Payment Amount }

R17 = {C, D} = { Payment ID, Payment Type, Payment Date, Payment Amount }

R18= {A, B, C} = { Ticket ID, Flight Class, Bag ID, Bag Weight(kg), Passenger Seat, Payment ID}

Payment ID is a foreign key referencing R17

R17 is 3NF but R18 not:

For R18:

A: Ticket ID

B: { Flight Class, Passenger Seat, Payment ID}

C: Bag ID

D: Bag Weight(kg)

R19 = {C, D} = { Bag ID, Bag Weight(kg)}

R20= {A, B, C} = { Ticket ID, Flight Class, Passenger Seat, Payment ID, Bag ID }

Bag ID is a foreign key referencing R19

FINALLY DATABASE TABLES ARE:

R3 = PASSENGER = { Passenger Passport Number, Passenger Name, Passenger Surname, Passenger Age, Passenger Gender, Passenger Nationality }

R5 = ARRIVAL AIRPORT = { Arrival Airport, Arrival Airport Country, Arrival Airport Total Passengers, Arrival Airport Elevation (ft) }

R7 = DEPARTURE AIRPORT = { Departure Airport, Departure Airport Country, Departure Airport Total Passengers, Departure Airport Elevation (ft) }

R9 = AIRLINE COMPANY = { Airline Company Name, Airline Company Total Passenger, Airline Company Country }

R11 = CAPTAIN = { Captain Certification Number, Captain Name, Captain Surname, Captain Age, Captain Flight Hour Experience }

R13 = AIRPLANE = { Airplane Model, Airplane Manufacturer, Active number of Airplane Model }

R14 = FLIGHT = {Flight Number, Departure Date Time, Arrival Date Time, Arrival Airport, Departure Airport, Airline Company Name, Captain Certification Number, Airplane Model }

R16 = TICKET AND FLIGHT = {Flight Number, Passenger Passport Number, Ticket ID }

R17 = PAYMENT = { Payment ID, Payment Type, Payment Date, Payment Amount }

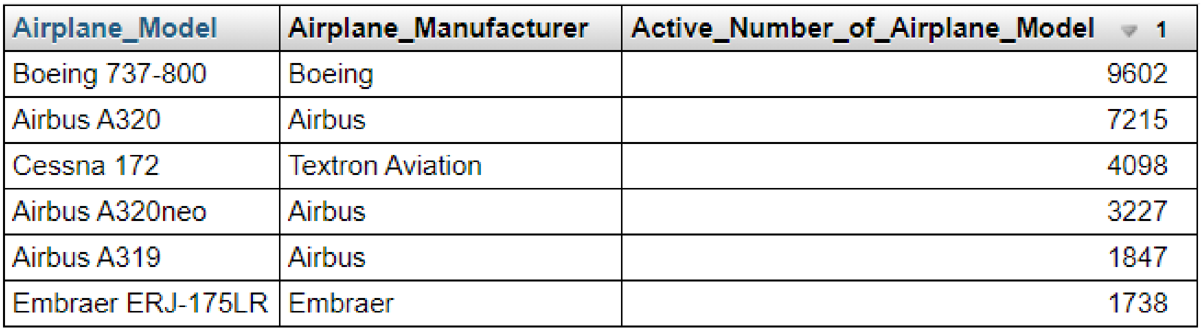
R19 = BAG = { Bag ID, Bag Weight(kg)}

R20= TICKET = { Ticket ID, Flight Class, Passenger Seat, Payment ID, Bag ID }

Some query examples to get information's from airline database:

1. Selecting active number of airplanes model in the world with decreasing number:

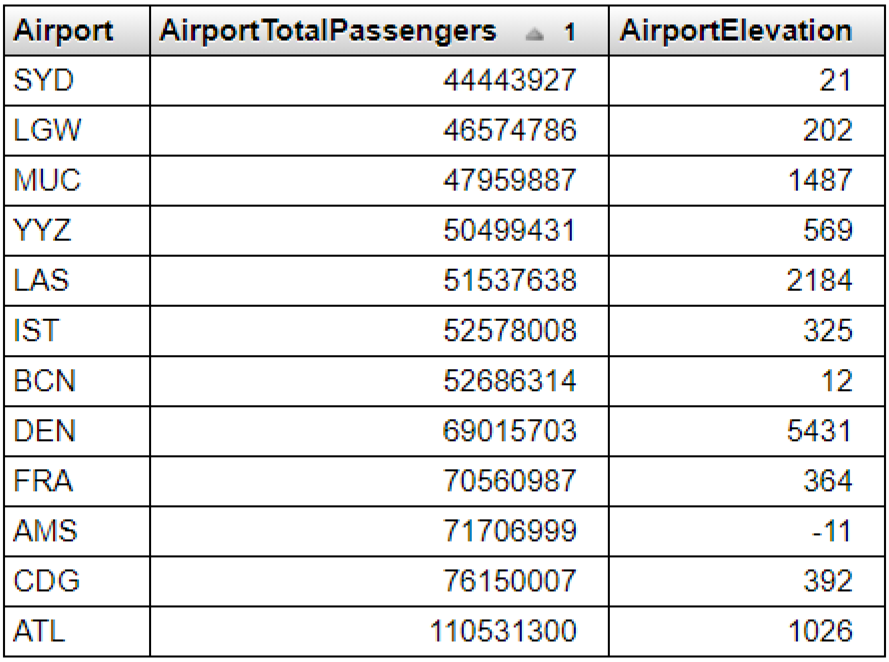
```
SELECT * from airplane ORDER BY airplane.Active_Number_of_Airplane_Model DESC
```



Airplane_Model	Airplane_Manufacturer	Active_Number_of_Airplane_Model ▼ 1
Boeing 737-800	Boeing	9602
Airbus A320	Airbus	7215
Cessna 172	Textron Aviation	4098
Airbus A320neo	Airbus	3227
Airbus A319	Airbus	1847
Embraer ERJ-175LR	Embraer	1738

2. Selecting all airports with increasing number in accordance to airport total passenger:

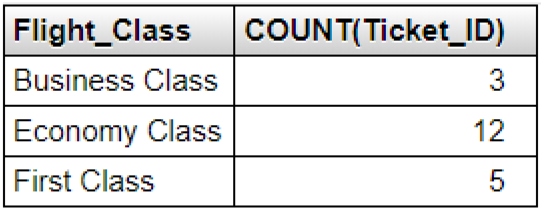
```
SELECT Departure_Airport as Airport,Departure_Airport_Total_Passengers AS
AirportTotalPassengers,Departure_Airport_Elevation_feet as AirportElevation FROM
departure_airport UNION SELECT arrival_airport as Airport,
Arrival_Airport_Total_Passengers AS AirportTotalPassengers,Arrival_Airport_Elevation_feet
as AirportElevation FROM arrival_airport ORDER by AirportTotalPassengers
```



Airport	AirportTotalPassengers ▲ 1	AirportElevation
SYD	44443927	21
LGW	46574786	202
MUC	47959887	1487
YYZ	50499431	569
LAS	51537638	2184
IST	52578008	325
BCN	52686314	12
DEN	69015703	5431
FRA	70560987	364
AMS	71706999	-11
CDG	76150007	392
ATL	110531300	1026

3. Selecting which flight class occurs in total:

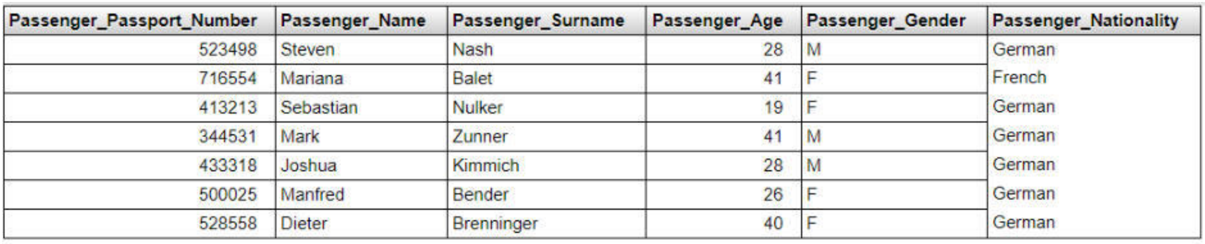
```
SELECT Flight_Class, COUNT(Ticket_ID) FROM ticket GROUP BY Flight_Class
```



Flight_Class	COUNT(Ticket_ID)
Business Class	3
Economy Class	12
First Class	5

4. Selecting Passengers whose nationality is German or French:

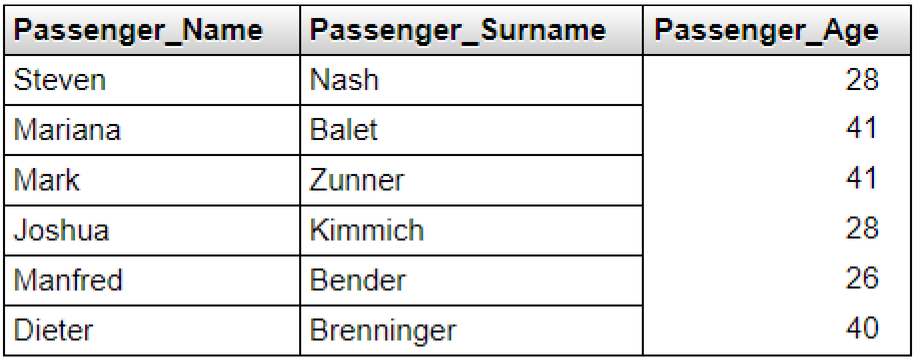
```
SELECT * FROM passenger where Passenger_Nationality in ( "German" ,"French")
```



Passenger_Passport_Number	Passenger_Name	Passenger_Surname	Passenger_Age	Passenger_Gender	Passenger_Nationality
523498	Steven	Nash	28	M	German
716554	Mariana	Balet	41	F	French
413213	Sebastian	Nulker	19	F	German
344531	Mark	Zunner	41	M	German
433318	Joshua	Kimmich	28	M	German
500025	Manfred	Bender	26	F	German
528558	Dieter	Brenninger	40	F	German

5. Selecting passengers whose nationality is French or German and older than 25:

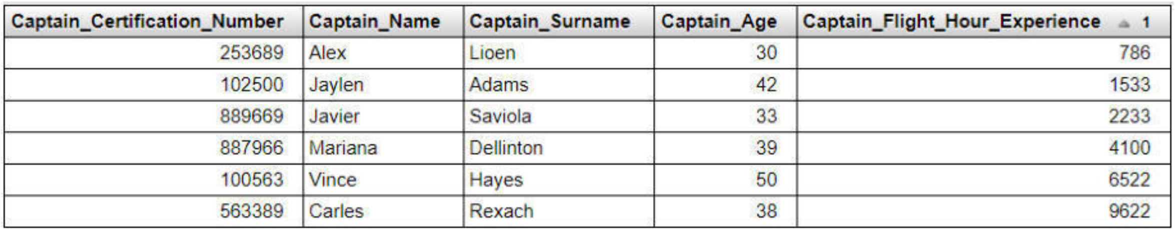
```
SELECT Passenger_Name,Passenger_Surname,Passenger_Age FROM passenger where
Passenger_Nationality = "German" and Passenger_Age>25 or Passenger_Nationality =
"French" and Passenger_Age>25
```



Passenger_Name	Passenger_Surname	Passenger_Age
Steven	Nash	28
Mariana	Balet	41
Mark	Zunner	41
Joshua	Kimmich	28
Manfred	Bender	26
Dieter	Brenninger	40

6. Ordering captains flight hour experience in increasing order:

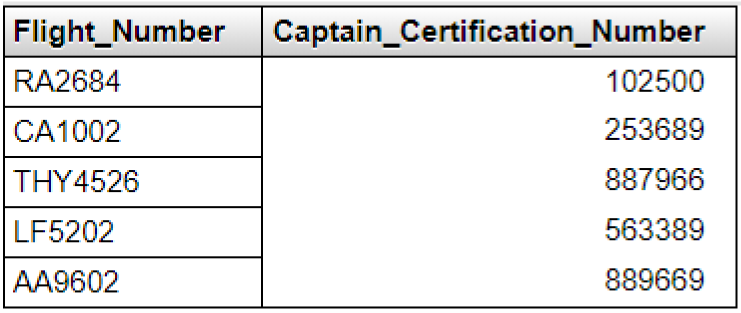
```
SELECT * from captain ORDER BY captain.Captain_Flight_Hour_Experience
```



Captain_Certification_Number	Captain_Name	Captain_Surname	Captain_Age	Captain_Flight_Hour_Experience ▲ 1
253689	Alex	Lioen	30	786
102500	Jaylen	Adams	42	1533
889669	Javier	Saviola	33	2233
887966	Mariana	Dellinton	39	4100
100563	Vince	Hayes	50	6522
563389	Carles	Rexach	38	9622

7. Selecting which captain take flight in which flight number:

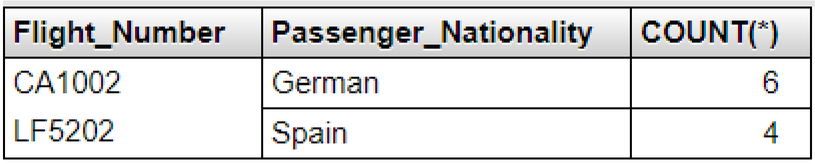
```
select Flight_Number, captain.Captain_Certification_Number from captain, flight where
captain.Captain_Certification_Number = flight.Captain_Certification_Number
```



Flight_Number	Captain_Certification_Number
RA2684	102500
CA1002	253689
THY4526	887966
LF5202	563389
AA9602	889669

8. Selecting nationalities which total passenger number in this nationality fly greater than two in total flight in the world.

```
SELECT flight.Flight_Number, passenger.Passenger_Nationality, COUNT(*) from ticket_flight,
flight, passenger where passenger.Passenger_Passport_Number =
ticket_flight.Passenger_Passport_Number and flight.Flight_Number =
ticket_flight.Flight_Number GROUP by passenger.Passenger_Nationality HAVING COUNT(*)
> 2
```



Flight_Number	Passenger_Nationality	COUNT(*)
CA1002	German	6
LF5202	Spain	4