Sri Lanka Institute of Information Technology

Online Airline Reservation System

Assignment Report

IE2042- Database Management Systems for Security

B.Sc. (Hons)in Information Technology Specializing in Cyber Security

Course Code: IE2042

Date: 11/02/2022

Batch: CS 2.2

Project Title: Online Airline Reservation System

Group Members :

Dissanayake W.P.D.B.	-	IT21372308
Zakey M.S.M.A.	-	IT21299902
Dilhara W. M. A.	-	IT21299452
Pemachandra T.H.R.T.	-	IT21301322

Terms Of Reference

The report was produced and submitted to fulfill the specifications for the IE2042 module at the Sri Lanka Institute of Information Technology.

Acknowledgement

We would want to take this chance to express our gratitude to our module lecturer, the SLIIT teaching staff, and the organizers for accommodating my last-minute inquiries and giving up their important time to mentor us through this assignment, which was a brand-new difficulty. We would also like to thank the lecturer for devoting long hours of his time to assist with the chosen topic. We also wish to say a big thank you to our parents for supporting us out.

Contents

A. Part I

1. Assumptions

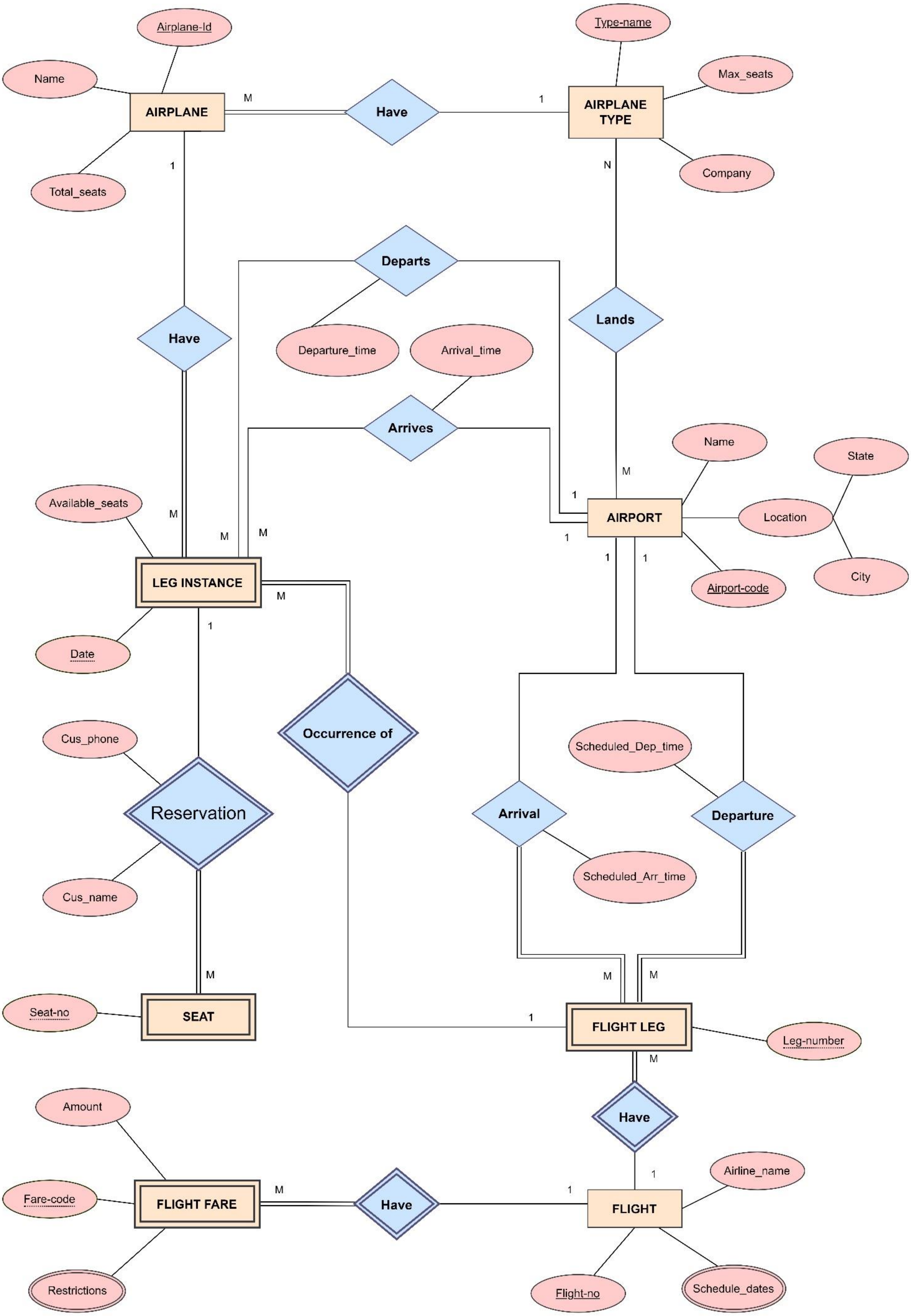
ER Diagram

- FLIGHT LEG is a weak entity of FLIGHT entity.
- FLIGHT FLARE is a weak entity of FLIGHT entity.
- LEG INSTANCE is a weak entity of FLIGHT LEG entity.
- SEAT is a weak entity of LEG INSTANCE entity.
- Schedule_dates attribute in FLIGHT entity is a multivalued attribute.
- Restrictions attribute in FLIGHT FLARE entity is a multivalued attribute.
- State and City are smaller attributes(subparts) of a composite attribute called Location.

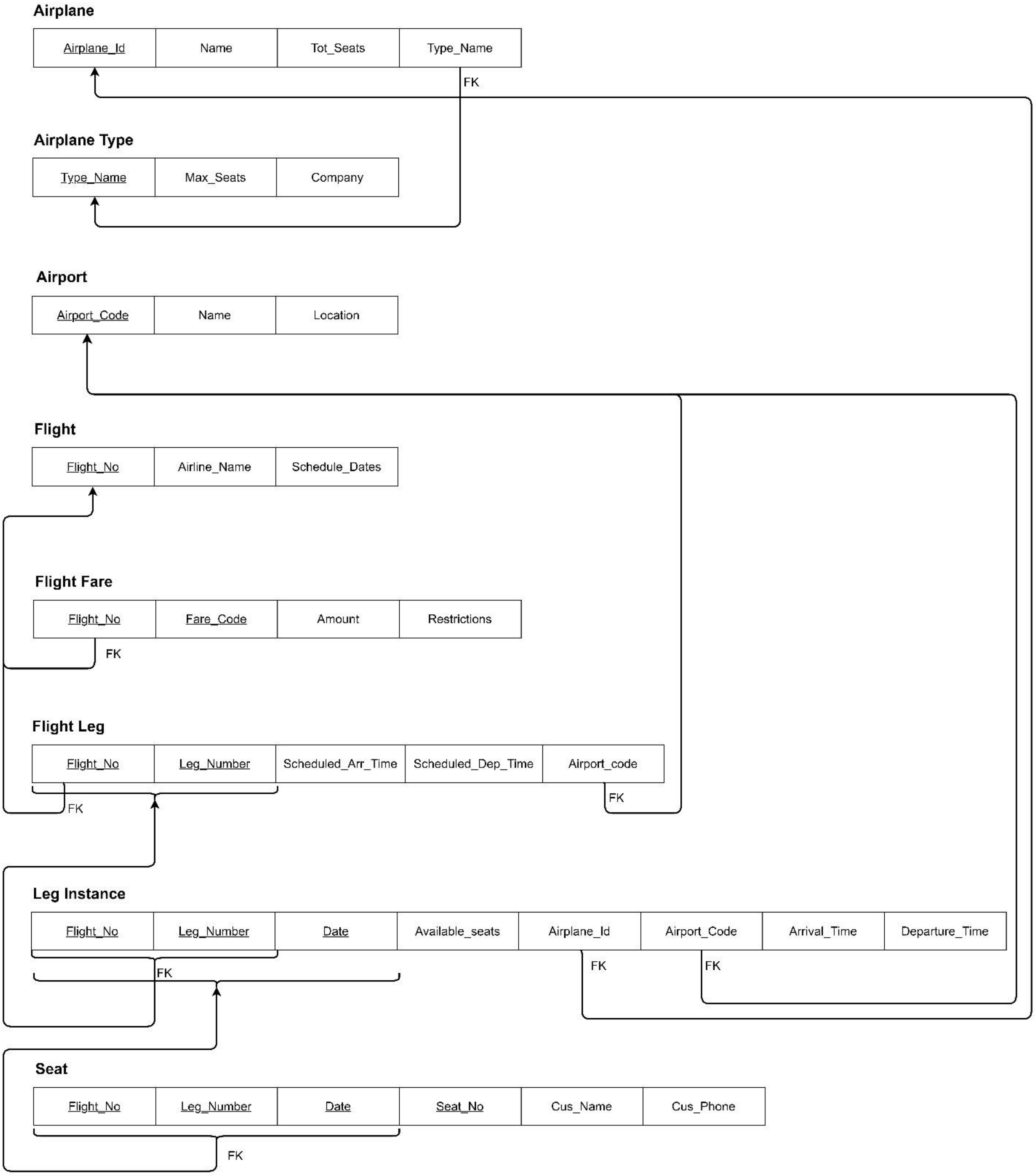
SQL codes

- There are 2 types of methods used to name airports in the world. One is IATA code which has a length of 3 characters. Second one is ICAO which has a length of 4 characters. For this assignment we used ICAO code for the airport code.

2. ER Diagram


Airplane-Id
Name
AIRPLANE
Total_seats
M
Have
1
Type-name
Max_seats
AIRPLANE TYPE
Company
N
1
Departs
Departure_time
Arrival_time
Lands
Have
Arrives
Name
State
M
Available_seats
M
M
M
1
AIRPORT
Location
1
LEG INSTANCE
1
1
Airport-code
City
M
Date
1
Occurrence of
Scheduled_Dep_time
Cus_phone
Reservation
Arrival
Departure
Scheduled_Arr_time
Cus_name
M
M
M
Seat-no
SEAT
1
FLIGHT LEG
Leg-number
M
Amount
Have
1
Airline_name
Fare-code
FLIGHT FARE
M
Have
1
FLIGHT
Restrictions
Flight-no
Schedule_dates

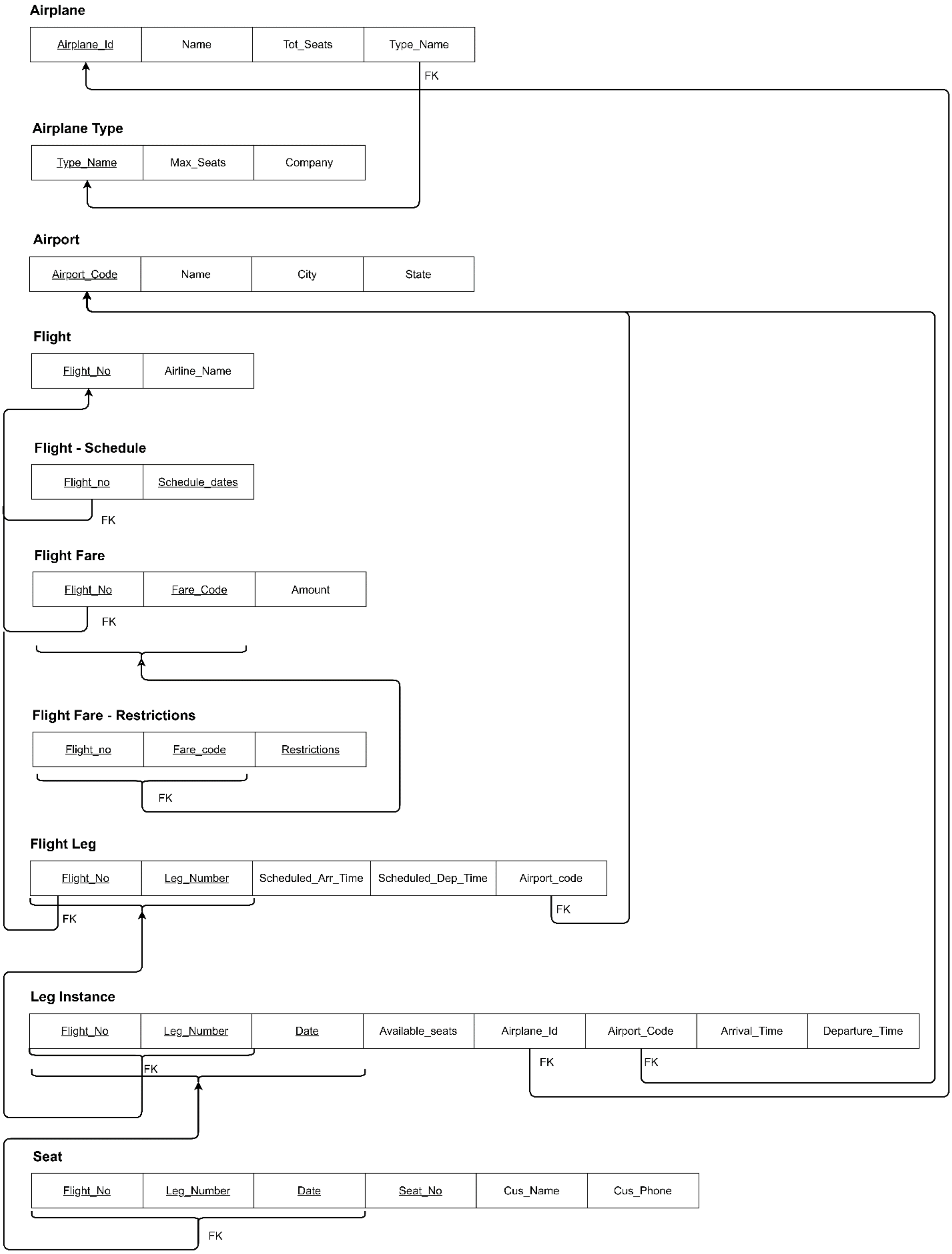
3. Logical Schema (UDF)


Airplane
Airplane_Id | Name | Tot_Seats | Type_Name
FK
Airplane Type
Type_Name | Max_Seats | Company
Airport
Airport_Code | Name | Location
Flight
Flight_No | Airline_Name | Schedule_Dates
Flight Fare
Flight_No | Fare_Code | Amount | Restrictions
FK
Flight Leg
Flight_No | Leg_Number | Scheduled_Arr_Time | Scheduled_Dep_Time | Airport_code
FK
FK
Leg Instance
Flight_No | Leg_Number | Date | Available_seats | Airplane_Id | Airport_Code | Arrival_Time | Departure_Time
FK
FK
FK
Seat
Flight_No | Leg_Number | Date | Seat_No | Cus_Name | Cus_Phone
FK

4. Normalization

i. 1NF

- Removed multi-valued attributes (Flight table and Flight Fare table)
- Removed composite attributes (Airport table)



Airplane

<u>Airplane_Id</u>	Name	Tot_Seats	Type_Name

Airplane Type

<u>Type_Name</u>	Max_Seats	Company

Airport

<u>Airport_Code</u>	Name	City	State

Flight

<u>Flight_No</u>	Airline_Name

Flight - Schedule

<u>Flight_no</u>	<u>Schedule_dates</u>

Flight Fare

<u>Flight_No</u>	<u>Fare_Code</u>	Amount

Flight Fare - Restrictions

<u>Flight_no</u>	<u>Fare_code</u>	<u>Restrictions</u>

Flight Leg

<u>Flight_No</u>	<u>Leg_Number</u>	Scheduled_Arr_Time	Scheduled_Dep_Time	Airport_code

Leg Instance

<u>Flight_No</u>	<u>Leg_Number</u>	<u>Date</u>	Available_seats	Airplane_Id	Airport_Code	Arrival_Time	Departure_Time

Seat

<u>Flight_No</u>	<u>Leg_Number</u>	<u>Date</u>	<u>Seat_No</u>	Cus_Name	Cus_Phone

ii. <u>2NF</u>

No partial dependencies found. Logical model is normalized to 2NF.

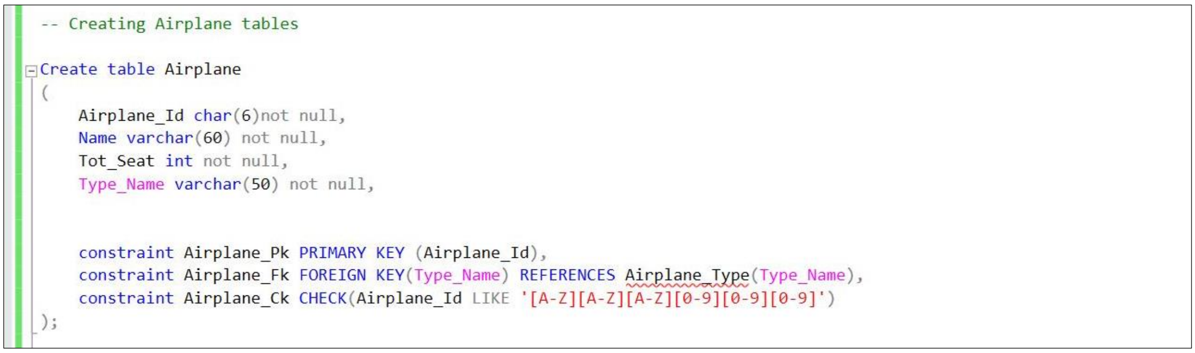
iii. <u>3NF</u>

No transitive dependencies found. Logical model is normalized to 3NF.

5. SQL Codes

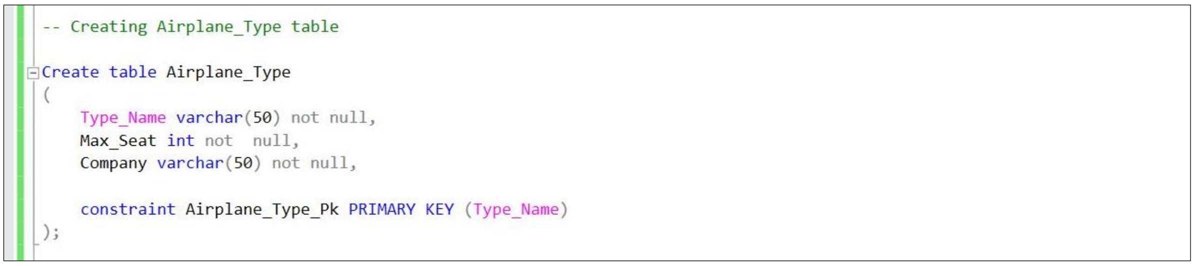
i. Create Tables (With constraints)

Airplane tables



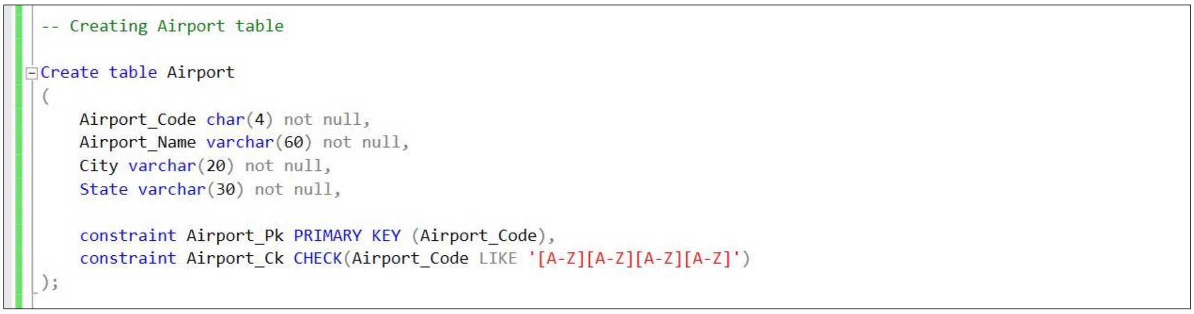
```
-- Creating Airplane tables

Create table Airplane
(
    Airplane_Id char(6)not null,
    Name varchar(60) not null,
    Tot_Seat int not null,
    Type_Name varchar(50) not null,


    constraint Airplane_Pk PRIMARY KEY (Airplane_Id),
    constraint Airplane_Fk FOREIGN KEY(Type_Name) REFERENCES Airplane_Type(Type_Name),
    constraint Airplane_Ck CHECK(Airplane_Id LIKE '[A-Z][A-Z][A-Z][0-9][0-9][0-9]')
);
```

Airplane_Type table



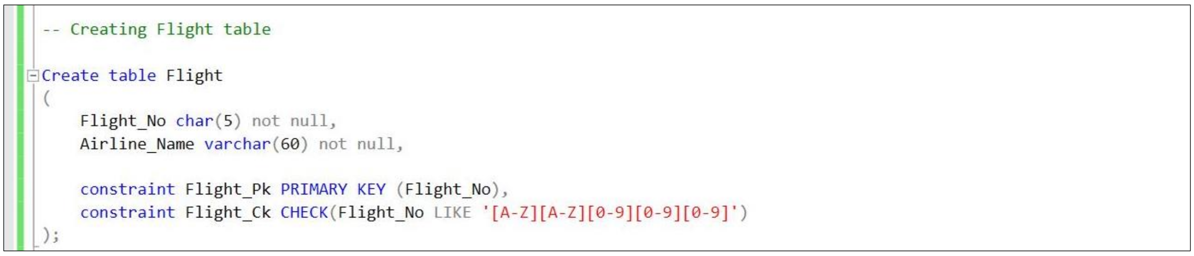
```
-- Creating Airplane_Type table

Create table Airplane_Type
(
    Type_Name varchar(50) not null,
    Max_Seat int not  null,
    Company varchar(50) not null,

    constraint Airplane_Type_Pk PRIMARY KEY (Type_Name)
);
```

Airport table



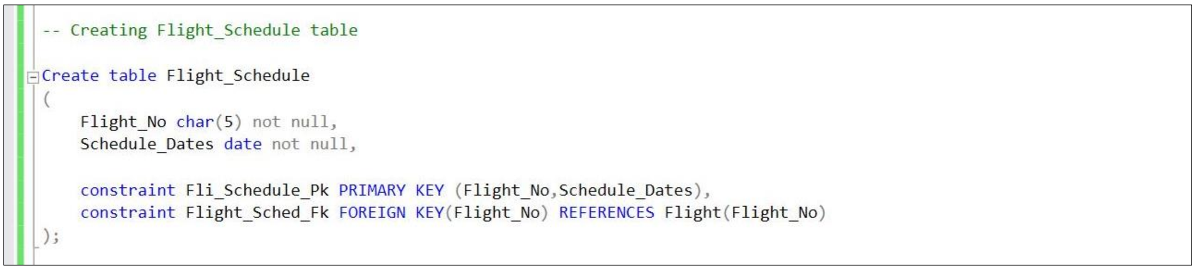
```
-- Creating Airport table

Create table Airport
(
    Airport_Code char(4) not null,
    Airport_Name varchar(60) not null,
    City varchar(20) not null,
    State varchar(30) not null,

    constraint Airport_Pk PRIMARY KEY (Airport_Code),
    constraint Airport_Ck CHECK(Airport_Code LIKE '[A-Z][A-Z][A-Z][A-Z]')
);
```

Flight table



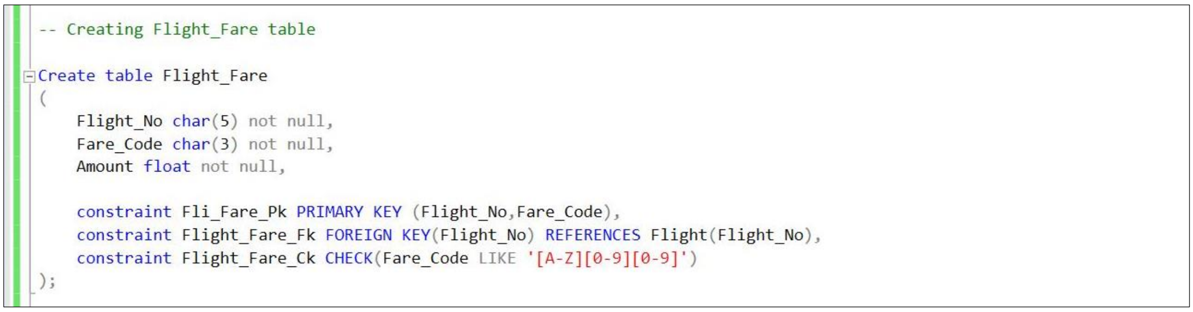
```sql
-- Creating Flight table

Create table Flight
(
    Flight_No char(5) not null,
    Airline_Name varchar(60) not null,

    constraint Flight_Pk PRIMARY KEY (Flight_No),
    constraint Flight_Ck CHECK(Flight_No LIKE '[A-Z][A-Z][0-9][0-9][0-9]')
);
```

Flight_Schedule table



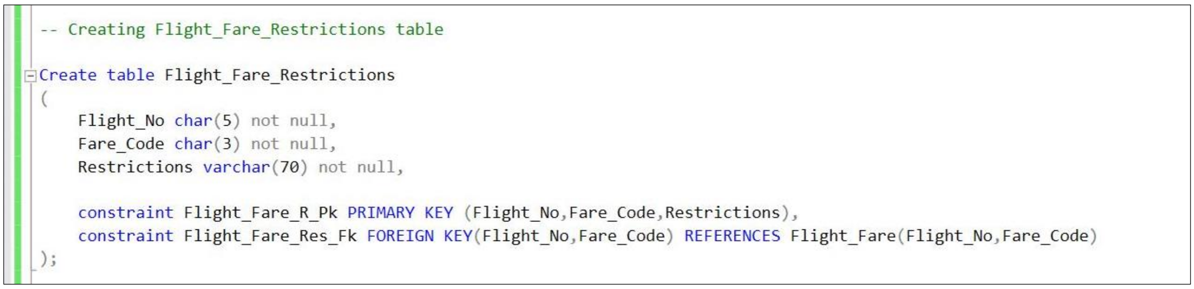
```sql
-- Creating Flight_Schedule table

Create table Flight_Schedule
(
    Flight_No char(5) not null,
    Schedule_Dates date not null,

    constraint Fli_Schedule_Pk PRIMARY KEY (Flight_No,Schedule_Dates),
    constraint Flight_Sched_Fk FOREIGN KEY(Flight_No) REFERENCES Flight(Flight_No)
);
```

Flight_Fare table



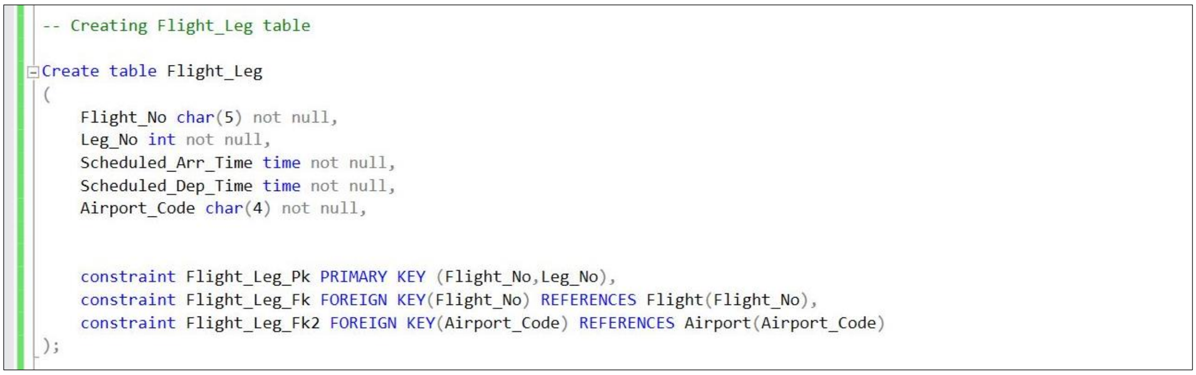
```sql
-- Creating Flight_Fare table

Create table Flight_Fare
(
    Flight_No char(5) not null,
    Fare_Code char(3) not null,
    Amount float not null,

    constraint Fli_Fare_Pk PRIMARY KEY (Flight_No,Fare_Code),
    constraint Flight_Fare_Fk FOREIGN KEY(Flight_No) REFERENCES Flight(Flight_No),
    constraint Flight_Fare_Ck CHECK(Fare_Code LIKE '[A-Z][0-9][0-9]')
);
```

Flight_Fare_Restrictions table



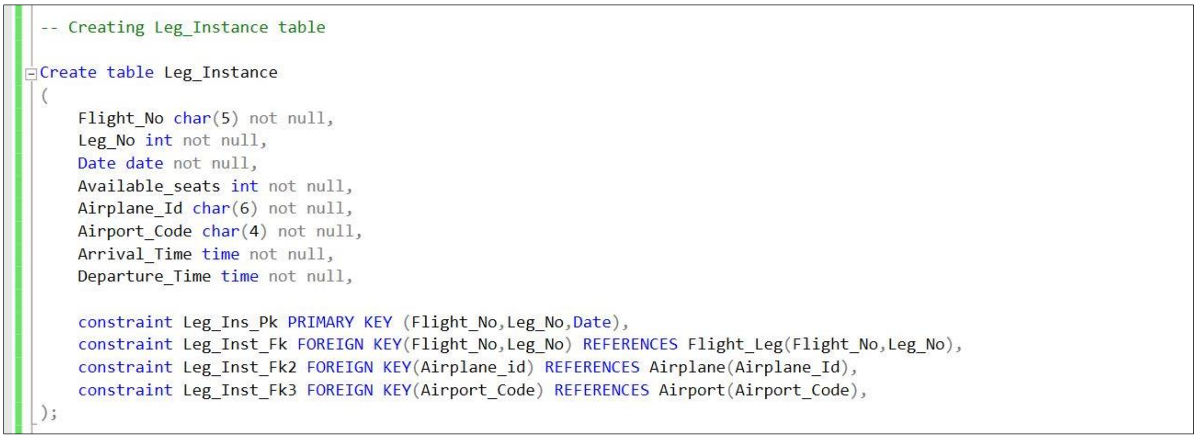
```sql
-- Creating Flight_Fare_Restrictions table

Create table Flight_Fare_Restrictions
(
    Flight_No char(5) not null,
    Fare_Code char(3) not null,
    Restrictions varchar(70) not null,

    constraint Flight_Fare_R_Pk PRIMARY KEY (Flight_No,Fare_Code,Restrictions),
    constraint Flight_Fare_Res_Fk FOREIGN KEY(Flight_No,Fare_Code) REFERENCES Flight_Fare(Flight_No,Fare_Code)
);
```

Flight_Leg table



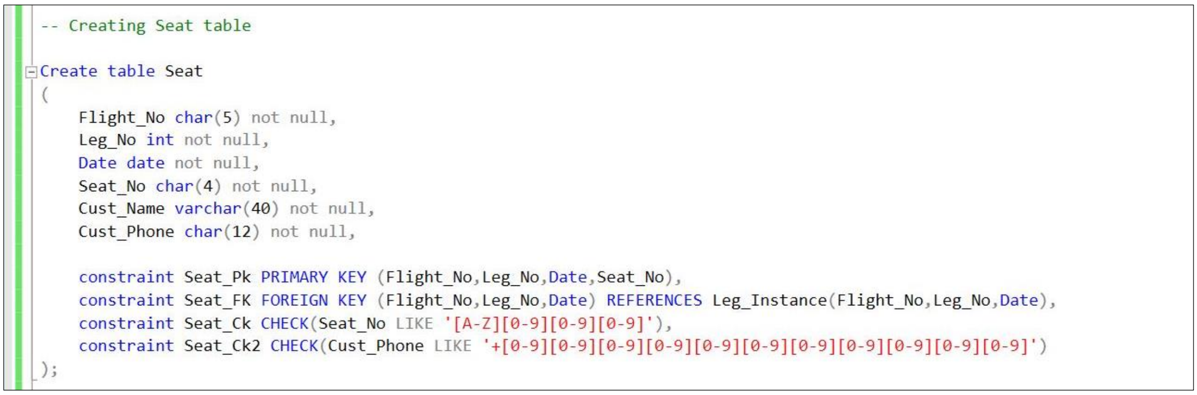
```
-- Creating Flight_Leg table

Create table Flight_Leg
(
    Flight_No char(5) not null,
    Leg_No int not null,
    Scheduled_Arr_Time time not null,
    Scheduled_Dep_Time time not null,
    Airport_Code char(4) not null,


    constraint Flight_Leg_Pk PRIMARY KEY (Flight_No,Leg_No),
    constraint Flight_Leg_Fk FOREIGN KEY(Flight_No) REFERENCES Flight(Flight_No),
    constraint Flight_Leg_Fk2 FOREIGN KEY(Airport_Code) REFERENCES Airport(Airport_Code)
);
```

Leg_Instance table



```
-- Creating Leg_Instance table

Create table Leg_Instance
(
    Flight_No char(5) not null,
    Leg_No int not null,
    Date date not null,
    Available_seats int not null,
    Airplane_Id char(6) not null,
    Airport_Code char(4) not null,
    Arrival_Time time not null,
    Departure_Time time not null,

    constraint Leg_Ins_Pk PRIMARY KEY (Flight_No,Leg_No,Date),
    constraint Leg_Inst_Fk FOREIGN KEY(Flight_No,Leg_No) REFERENCES Flight_Leg(Flight_No,Leg_No),
    constraint Leg_Inst_Fk2 FOREIGN KEY(Airplane_id) REFERENCES Airplane(Airplane_Id),
    constraint Leg_Inst_Fk3 FOREIGN KEY(Airport_Code) REFERENCES Airport(Airport_Code),
);
```

Seat table



```
-- Creating Seat table

Create table Seat
(
    Flight_No char(5) not null,
    Leg_No int not null,
    Date date not null,
    Seat_No char(4) not null,
    Cust_Name varchar(40) not null,
    Cust_Phone char(12) not null,

    constraint Seat_Pk PRIMARY KEY (Flight_No,Leg_No,Date,Seat_No),
    constraint Seat_FK FOREIGN KEY (Flight_No,Leg_No,Date) REFERENCES Leg_Instance(Flight_No,Leg_No,Date),
    constraint Seat_Ck CHECK(Seat_No LIKE '[A-Z][0-9][0-9][0-9]'),
    constraint Seat_Ck2 CHECK(Cust_Phone LIKE '+[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
);
```

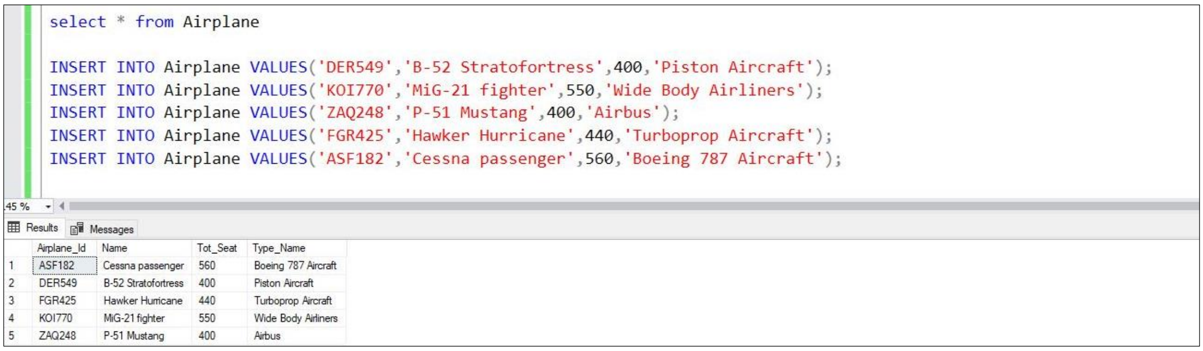
```
-- Creating Airplane tables

Create table Airplane
(
    Airplane_Id char(6)not null,
       Name varchar(60) not null,
       Tot_Seat int not null,
       Type_Name varchar(50) not null,


       constraint Airplane_Pk PRIMARY KEY (Airplane_Id),
       constraint Airplane_Fk FOREIGN KEY(Type_Name) REFERENCES Airplane_Type(Type_Name),
       constraint Airplane_Ck CHECK(Airplane_Id LIKE '[A-Z][A-Z][A-Z][0-9][0-9][0-9]')
);

-- Creating Airplane_Type table

Create table Airplane_Type
(
       Type_Name varchar(50) not null,
       Max_Seat int not  null,
       Company varchar(50) not null,

       constraint Airplane_Type_Pk PRIMARY KEY (Type_Name)
);

-- Creating Airport table

Create table Airport
(
       Airport_Code char(4) not null,
       Airport_Name varchar(60) not null,
       City varchar(20) not null,
       State varchar(30) not null,

       constraint Airport_Pk PRIMARY KEY (Airport_Code),
       constraint Airport_Ck CHECK(Airport_Code LIKE '[A-Z][A-Z][A-Z][A-Z]')
);

-- Creating Flight table

Create table Flight
(
       Flight_No char(5) not null,
       Airline_Name varchar(60) not null,

       constraint Flight_Pk PRIMARY KEY (Flight_No),
       constraint Flight_Ck CHECK(Flight_No LIKE '[A-Z][A-Z][0-9][0-9][0-9]')
);

-- Creating Flight_Schedule table

Create table Flight_Schedule
(
       Flight_No char(5) not null,
       Schedule_Dates date not null,

       constraint Fli_Schedule_Pk PRIMARY KEY (Flight_No,Schedule_Dates),
       constraint Flight_Sched_Fk FOREIGN KEY(Flight_No) REFERENCES Flight(Flight_No)
);
```

```sql
-- Creating Flight_Fare table

Create table Flight_Fare
(
	Flight_No char(5) not null,
	Fare_Code char(3) not null,
	Amount float not null,

	constraint Fli_Fare_Pk PRIMARY KEY (Flight_No,Fare_Code),
	constraint Flight_Fare_Fk FOREIGN KEY(Flight_No) REFERENCES Flight(Flight_No),
	constraint Flight_Fare_Ck CHECK(Fare_Code LIKE '[A-Z][0-9][0-9]')
);

-- Creating Flight_Fare_Restrictions table

Create table Flight_Fare_Restrictions
(
	Flight_No char(5) not null,
	Fare_Code char(3) not null,
	Restrictions varchar(70) not null,

	constraint Flight_Fare_R_Pk PRIMARY KEY (Flight_No,Fare_Code,Restrictions),
	constraint Flight_Fare_Res_Fk FOREIGN KEY(Flight_No,Fare_Code) REFERENCES 
Flight_Fare(Flight_No,Fare_Code)
);

-- Creating Flight_Leg table

Create table Flight_Leg
(
	Flight_No char(5) not null,
	Leg_No int not null,
	Scheduled_Arr_Time time not null,
	Scheduled_Dep_Time time not null,
	Airport_Code char(4) not null,


	constraint Flight_Leg_Pk PRIMARY KEY (Flight_No,Leg_No),
	constraint Flight_Leg_Fk FOREIGN KEY(Flight_No) REFERENCES Flight(Flight_No),
	constraint Flight_Leg_Fk2 FOREIGN KEY(Airport_Code) REFERENCES 
Airport(Airport_Code)
);

-- Creating Leg_Instance table

Create table Leg_Instance
(
	Flight_No char(5) not null,
	Leg_No int not null,
	Date date not null,
	Available_seats int not null,
	Airplane_Id char(6) not null,
	Airport_Code char(4) not null,
	Arrival_Time time not null,
	Departure_Time time not null,

	constraint Leg_Ins_Pk PRIMARY KEY (Flight_No,Leg_No,Date),
	constraint Leg_Inst_Fk FOREIGN KEY(Flight_No,Leg_No) REFERENCES 
Flight_Leg(Flight_No,Leg_No),
	constraint Leg_Inst_Fk2 FOREIGN KEY(Airplane_id) REFERENCES Airplane(Airplane_Id),
	constraint Leg_Inst_Fk3 FOREIGN KEY(Airport_Code) REFERENCES Airport(Airport_Code),
);
```

```
-- Creating Seat table

Create table Seat
(
        Flight_No char(5) not null,
        Leg_No int not null,
        Date date not null,
        Seat_No char(4) not null,
        Cust_Name varchar(40) not null,
        Cust_Phone char(12) not null,

    constraint Seat_Pk PRIMARY KEY (Flight_No,Leg_No,Date,Seat_No),
        constraint Seat_FK FOREIGN KEY (Flight_No,Leg_No,Date) REFERENCES Leg_Instance(Flight_No,Leg_No,Date),
        constraint Seat_Ck CHECK(Seat_No LIKE '[A-Z][0-9][0-9][0-9]'),
        constraint Seat_Ck2 CHECK(Cust_Phone LIKE '+[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
);
```

ii. Sample Data Entry

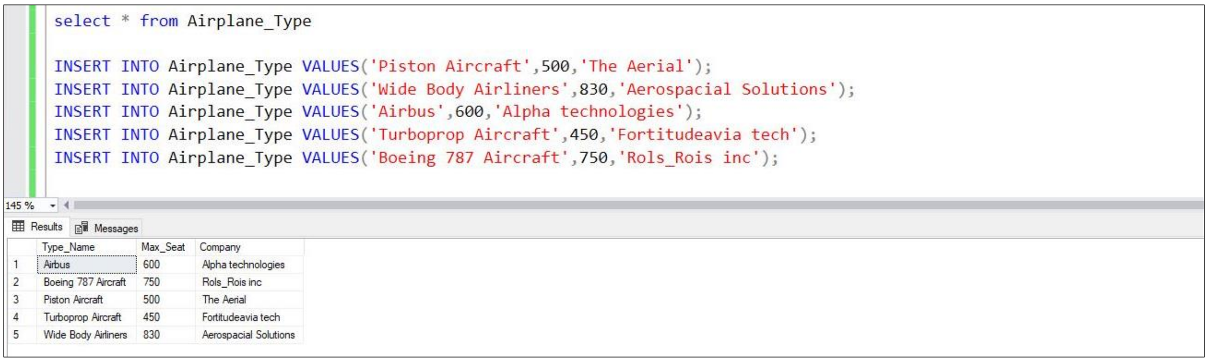
Airplane tables



```sql
select * from Airplane

INSERT INTO Airplane VALUES('DER549','B-52 Stratofortress',400,'Piston Aircraft');
INSERT INTO Airplane VALUES('KOI770','MiG-21 fighter',550,'Wide Body Airliners');
INSERT INTO Airplane VALUES('ZAQ248','P-51 Mustang',400,'Airbus');
INSERT INTO Airplane VALUES('FGR425','Hawker Hurricane',440,'Turboprop Aircraft');
INSERT INTO Airplane VALUES('ASF182','Cessna passenger',560,'Boeing 787 Aircraft');
```

	Airplane_Id	Name	Tot_Seat	Type_Name
1	ASF182	Cessna passenger	560	Boeing 787 Aircraft
2	DER549	B-52 Stratofortress	400	Piston Aircraft
3	FGR425	Hawker Hurricane	440	Turboprop Aircraft
4	KOI770	MiG-21 fighter	550	Wide Body Airliners
5	ZAQ248	P-51 Mustang	400	Airbus

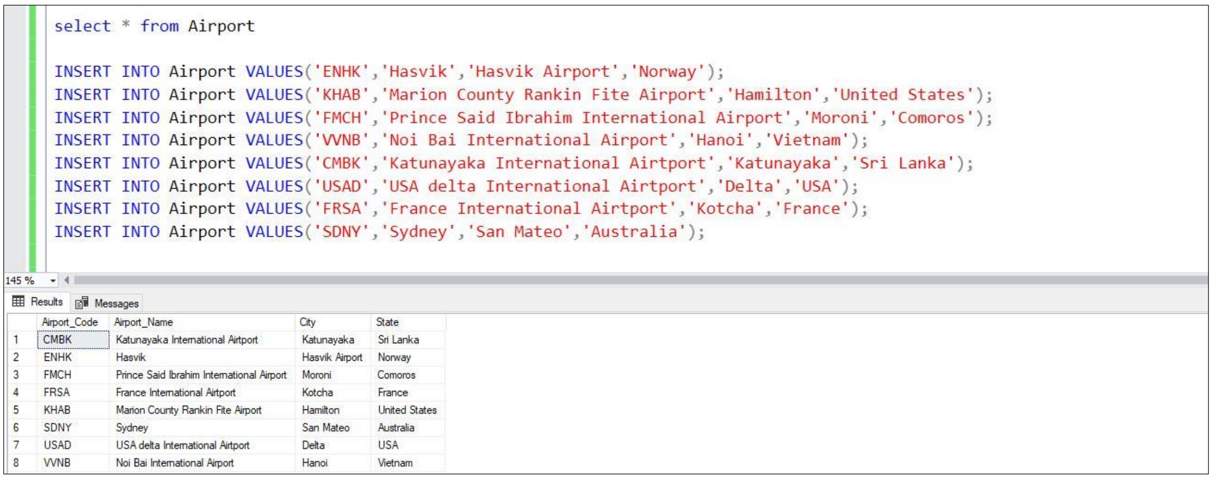
Airplane_Type table



```sql
select * from Airplane_Type

INSERT INTO Airplane_Type VALUES('Piston Aircraft',500,'The Aerial');
INSERT INTO Airplane_Type VALUES('Wide Body Airliners',830,'Aerospacial Solutions');
INSERT INTO Airplane_Type VALUES('Airbus',600,'Alpha technologies');
INSERT INTO Airplane_Type VALUES('Turboprop Aircraft',450,'Fortitudeavia tech');
INSERT INTO Airplane_Type VALUES('Boeing 787 Aircraft',750,'Rols_Rois inc');
```

	Type_Name	Max_Seat	Company
1	Airbus	600	Alpha technologies
2	Boeing 787 Aircraft	750	Rols_Rois inc
3	Piston Aircraft	500	The Aerial
4	Turboprop Aircraft	450	Fortitudeavia tech
5	Wide Body Airliners	830	Aerospacial Solutions

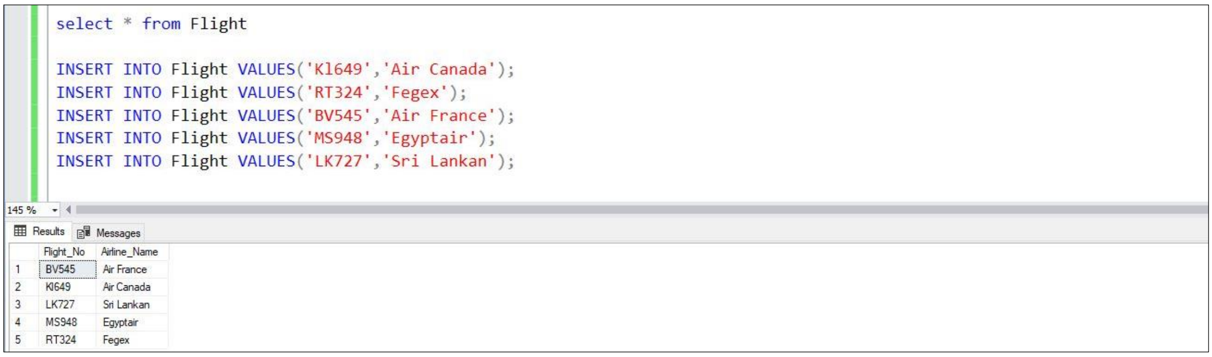
Airport table



```sql
select * from Airport

INSERT INTO Airport VALUES('ENHK','Hasvik','Hasvik Airport','Norway');
INSERT INTO Airport VALUES('KHAB','Marion County Rankin Fite Airport','Hamilton','United States');
INSERT INTO Airport VALUES('FMCH','Prince Said Ibrahim International Airport','Moroni','Comoros');
INSERT INTO Airport VALUES('VVNB','Noi Bai International Airport','Hanoi','Vietnam');
INSERT INTO Airport VALUES('CMBK','Katunayaka International Airtport','Katunayaka','Sri Lanka');
INSERT INTO Airport VALUES('USAD','USA delta International Airtport','Delta','USA');
INSERT INTO Airport VALUES('FRSA','France International Airtport','Kotcha','France');
INSERT INTO Airport VALUES('SDNY','Sydney','San Mateo','Australia');
```

	Airport_Code	Airport_Name	City	State
1	CMBK	Katunayaka International Airtport	Katunayaka	Sri Lanka
2	ENHK	Hasvik	Hasvik Airport	Norway
3	FMCH	Prince Said Ibrahim International Airport	Moroni	Comoros
4	FRSA	France International Airtport	Kotcha	France
5	KHAB	Marion County Rankin Fite Airport	Hamilton	United States
6	SDNY	Sydney	San Mateo	Australia
7	USAD	USA delta International Airtport	Delta	USA
8	VVNB	Noi Bai International Airport	Hanoi	Vietnam

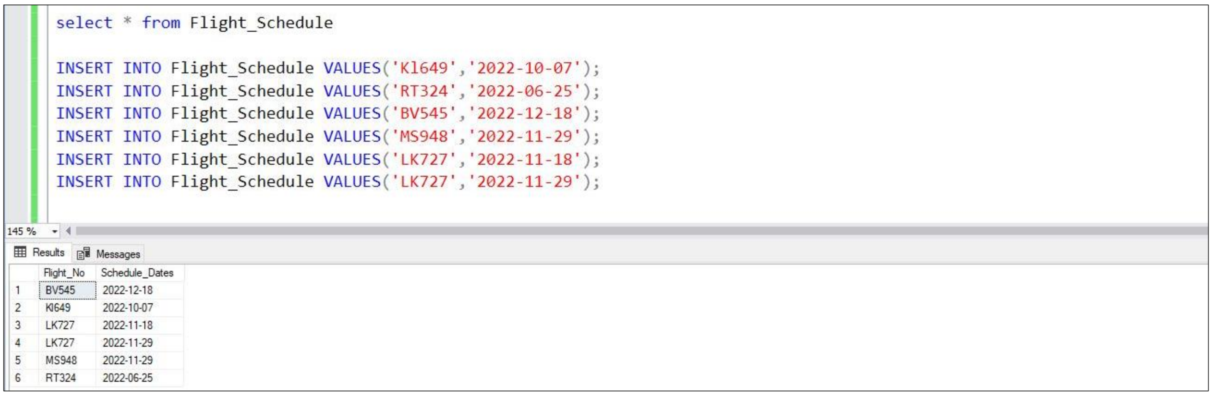
Flight table



```
select * from Flight

INSERT INTO Flight VALUES('Kl649','Air Canada');
INSERT INTO Flight VALUES('RT324','Fegex');
INSERT INTO Flight VALUES('BV545','Air France');
INSERT INTO Flight VALUES('MS948','Egyptair');
INSERT INTO Flight VALUES('LK727','Sri Lankan');
```

	Flight_No	Airline_Name
1	BV545	Air France
2	Kl649	Air Canada
3	LK727	Sri Lankan
4	MS948	Egyptair
5	RT324	Fegex

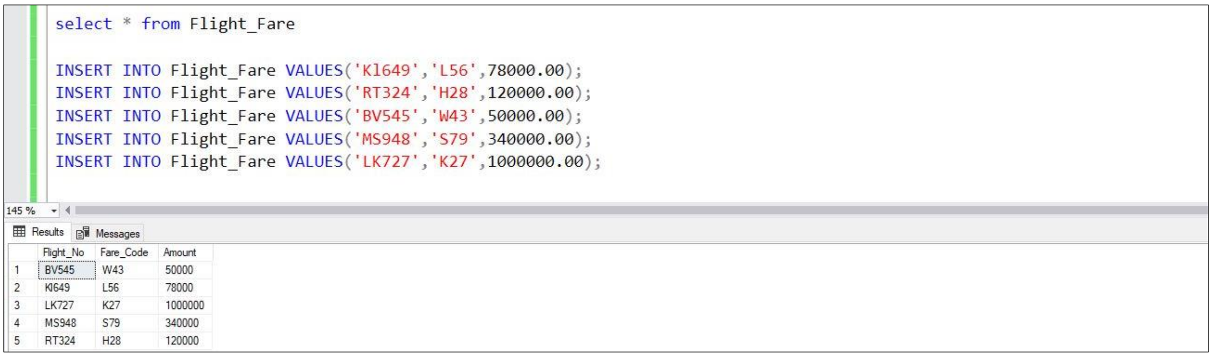
Flight_Schedule table



```
select * from Flight_Schedule

INSERT INTO Flight_Schedule VALUES('Kl649','2022-10-07');
INSERT INTO Flight_Schedule VALUES('RT324','2022-06-25');
INSERT INTO Flight_Schedule VALUES('BV545','2022-12-18');
INSERT INTO Flight_Schedule VALUES('MS948','2022-11-29');
INSERT INTO Flight_Schedule VALUES('LK727','2022-11-18');
INSERT INTO Flight_Schedule VALUES('LK727','2022-11-29');
```

	Flight_No	Schedule_Dates
1	BV545	2022-12-18
2	Kl649	2022-10-07
3	LK727	2022-11-18
4	LK727	2022-11-29
5	MS948	2022-11-29
6	RT324	2022-06-25

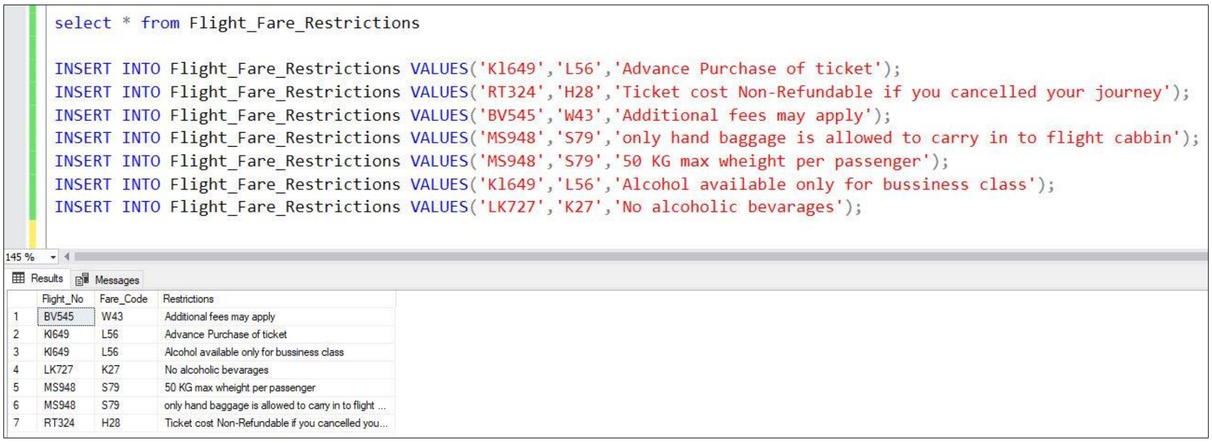
Flight_Fare table



```
select * from Flight_Fare

INSERT INTO Flight_Fare VALUES('Kl649','L56',78000.00);
INSERT INTO Flight_Fare VALUES('RT324','H28',120000.00);
INSERT INTO Flight_Fare VALUES('BV545','W43',50000.00);
INSERT INTO Flight_Fare VALUES('MS948','S79',340000.00);
INSERT INTO Flight_Fare VALUES('LK727','K27',1000000.00);
```

	Flight_No	Fare_Code	Amount
1	BV545	W43	50000
2	Kl649	L56	78000
3	LK727	K27	1000000
4	MS948	S79	340000
5	RT324	H28	120000

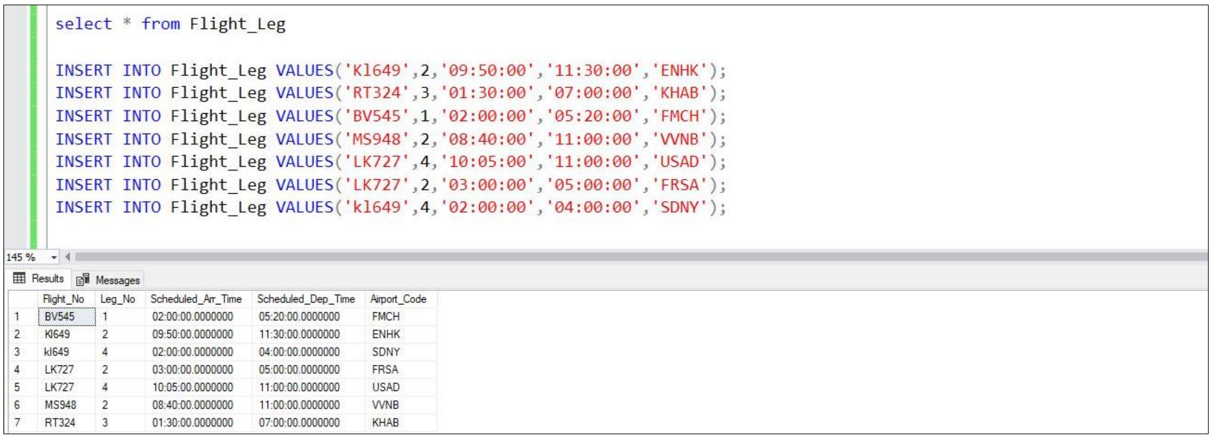
Flight_Fare_Restrictions table



```sql
select * from Flight_Fare_Restrictions

INSERT INTO Flight_Fare_Restrictions VALUES('Kl649','L56','Advance Purchase of ticket');
INSERT INTO Flight_Fare_Restrictions VALUES('RT324','H28','Ticket cost Non-Refundable if you cancelled your journey');
INSERT INTO Flight_Fare_Restrictions VALUES('BV545','W43','Additional fees may apply');
INSERT INTO Flight_Fare_Restrictions VALUES('MS948','S79','only hand baggage is allowed to carry in to flight cabbin');
INSERT INTO Flight_Fare_Restrictions VALUES('MS948','S79','50 KG max wheight per passenger');
INSERT INTO Flight_Fare_Restrictions VALUES('Kl649','L56','Alcohol available only for bussiness class');
INSERT INTO Flight_Fare_Restrictions VALUES('LK727','K27','No alcoholic bevarages');
```

	Flight_No	Fare_Code	Restrictions
1	BV545	W43	Additional fees may apply
2	KI649	L56	Advance Purchase of ticket
3	KI649	L56	Alcohol available only for bussiness class
4	LK727	K27	No alcoholic bevarages
5	MS948	S79	50 KG max wheight per passenger
6	MS948	S79	only hand baggage is allowed to carry in to flight ...
7	RT324	H28	Ticket cost Non-Refundable if you cancelled you...

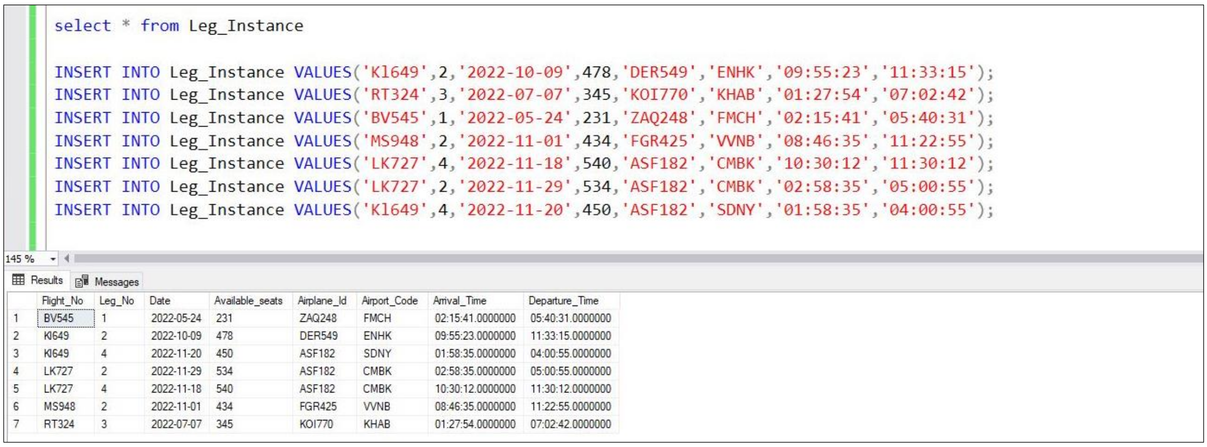
Flight_Leg table



```sql
select * from Flight_Leg

INSERT INTO Flight_Leg VALUES('Kl649',2,'09:50:00','11:30:00','ENHK');
INSERT INTO Flight_Leg VALUES('RT324',3,'01:30:00','07:00:00','KHAB');
INSERT INTO Flight_Leg VALUES('BV545',1,'02:00:00','05:20:00','FMCH');
INSERT INTO Flight_Leg VALUES('MS948',2,'08:40:00','11:00:00','VVNB');
INSERT INTO Flight_Leg VALUES('LK727',4,'10:05:00','11:00:00','USAD');
INSERT INTO Flight_Leg VALUES('LK727',2,'03:00:00','05:00:00','FRSA');
INSERT INTO Flight_Leg VALUES('kl649',4,'02:00:00','04:00:00','SDNY');
```

	Flight_No	Leg_No	Scheduled_Arr_Time	Scheduled_Dep_Time	Airport_Code
1	BV545	1	02:00:00.0000000	05:20:00.0000000	FMCH
2	KI649	2	09:50:00.0000000	11:30:00.0000000	ENHK
3	kI649	4	02:00:00.0000000	04:00:00.0000000	SDNY
4	LK727	2	03:00:00.0000000	05:00:00.0000000	FRSA
5	LK727	4	10:05:00.0000000	11:00:00.0000000	USAD
6	MS948	2	08:40:00.0000000	11:00:00.0000000	VVNB
7	RT324	3	01:30:00.0000000	07:00:00.0000000	KHAB

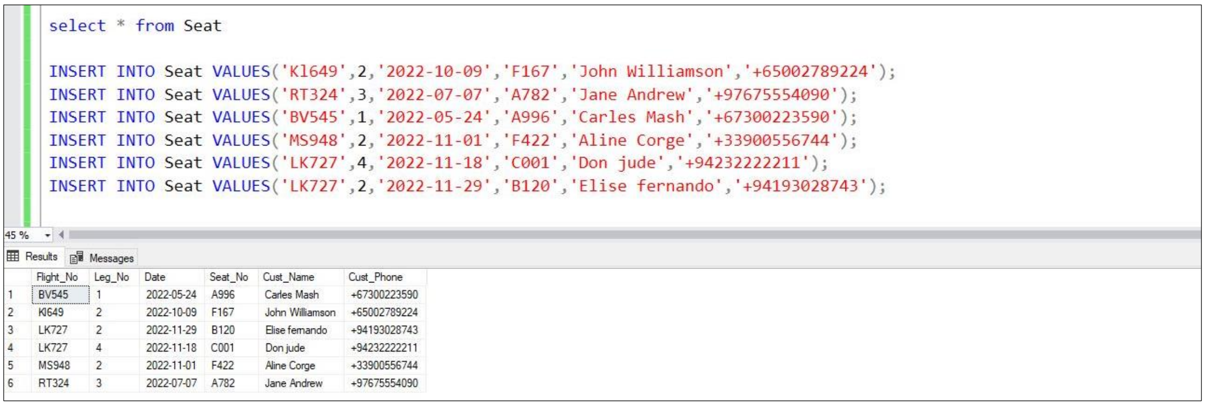
Leg_Instance table



```sql
select * from Leg_Instance

INSERT INTO Leg_Instance VALUES('Kl649',2,'2022-10-09',478,'DER549','ENHK','09:55:23','11:33:15');
INSERT INTO Leg_Instance VALUES('RT324',3,'2022-07-07',345,'KOI770','KHAB','01:27:54','07:02:42');
INSERT INTO Leg_Instance VALUES('BV545',1,'2022-05-24',231,'ZAQ248','FMCH','02:15:41','05:40:31');
INSERT INTO Leg_Instance VALUES('MS948',2,'2022-11-01',434,'FGR425','VVNB','08:46:35','11:22:55');
INSERT INTO Leg_Instance VALUES('LK727',4,'2022-11-18',540,'ASF182','CMBK','10:30:12','11:30:12');
INSERT INTO Leg_Instance VALUES('LK727',2,'2022-11-29',534,'ASF182','CMBK','02:58:35','05:00:55');
INSERT INTO Leg_Instance VALUES('Kl649',4,'2022-11-20',450,'ASF182','SDNY','01:58:35','04:00:55');
```

	Flight_No	Leg_No	Date	Available_seats	Airplane_Id	Airport_Code	Arrival_Time	Departure_Time
1	BV545	1	2022-05-24	231	ZAQ248	FMCH	02:15:41.0000000	05:40:31.0000000
2	KI649	2	2022-10-09	478	DER549	ENHK	09:55:23.0000000	11:33:15.0000000
3	KI649	4	2022-11-20	450	ASF182	SDNY	01:58:35.0000000	04:00:55.0000000
4	LK727	2	2022-11-29	534	ASF182	CMBK	02:58:35.0000000	05:00:55.0000000
5	LK727	4	2022-11-18	540	ASF182	CMBK	10:30:12.0000000	11:30:12.0000000
6	MS948	2	2022-11-01	434	FGR425	VVNB	08:46:35.0000000	11:22:55.0000000
7	RT324	3	2022-07-07	345	KOI770	KHAB	01:27:54.0000000	07:02:42.0000000

Seat table



```
select * from Seat

INSERT INTO Seat VALUES('Kl649',2,'2022-10-09','F167','John Williamson','+65002789224');
INSERT INTO Seat VALUES('RT324',3,'2022-07-07','A782','Jane Andrew','+97675554090');
INSERT INTO Seat VALUES('BV545',1,'2022-05-24','A996','Carles Mash','+67300223590');
INSERT INTO Seat VALUES('MS948',2,'2022-11-01','F422','Aline Corge','+33900556744');
INSERT INTO Seat VALUES('LK727',4,'2022-11-18','C001','Don jude','+94232222211');
INSERT INTO Seat VALUES('LK727',2,'2022-11-29','B120','Elise fernando','+94193028743');
```

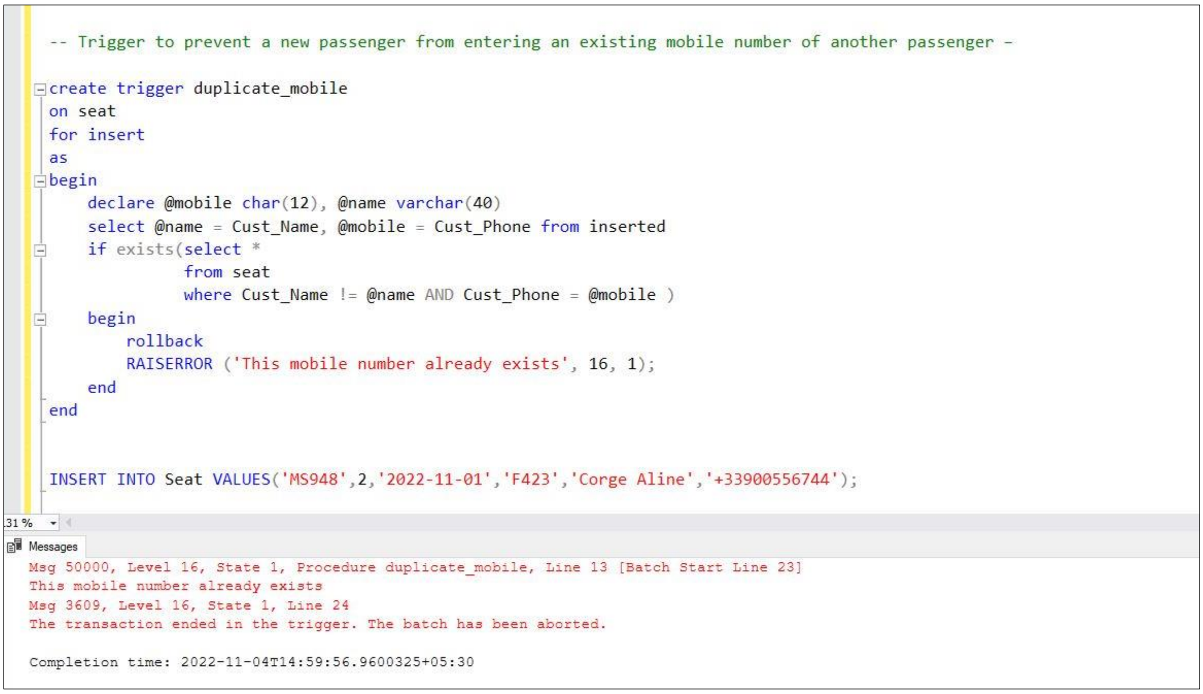
	Flight_No	Leg_No	Date	Seat_No	Cust_Name	Cust_Phone
1	BV545	1	2022-05-24	A996	Carles Mash	+67300223590
2	Kl649	2	2022-10-09	F167	John Williamson	+65002789224
3	LK727	2	2022-11-29	B120	Elise fernando	+94193028743
4	LK727	4	2022-11-18	C001	Don jude	+94232222211
5	MS948	2	2022-11-01	F422	Aline Corge	+33900556744
6	RT324	3	2022-07-07	A782	Jane Andrew	+97675554090

```
INSERT INTO Airplane VALUES('DER549','B-52 Stratofortress',400,'Piston Aircraft');
INSERT INTO Airplane VALUES('KOI770','MiG-21 fighter',550,'Wide Body Airliners');
INSERT INTO Airplane VALUES('ZAQ248','P-51 Mustang',400,'Airbus');
INSERT INTO Airplane VALUES('FGR425','Hawker Hurricane',440,'Turboprop Aircraft');
INSERT INTO Airplane VALUES('ASF182','Cessna passenger',560,'Boeing 787 Aircraft');


INSERT INTO Airplane_Type VALUES('Piston Aircraft',500,'The Aerial');
INSERT INTO Airplane_Type VALUES('Wide Body Airliners',830,'Aerospacial Solutions');
INSERT INTO Airplane_Type VALUES('Airbus',600,'Alpha technologies');
INSERT INTO Airplane_Type VALUES('Turboprop Aircraft',450,'Fortitudeavia tech');
INSERT INTO Airplane_Type VALUES('Boeing 787 Aircraft',750,'Rols_Rois inc');


INSERT INTO Airport VALUES('ENHK','Hasvik','Hasvik Airport','Norway');
INSERT INTO Airport VALUES('KHAB','Marion County Rankin Fite Airport','Hamilton','United
States');
INSERT INTO Airport VALUES('FMCH','Prince Said Ibrahim International
Airport','Moroni','Comoros');
INSERT INTO Airport VALUES('VVNB','Noi Bai International Airport','Hanoi','Vietnam');
INSERT INTO Airport VALUES('CMBK','Katunayaka International Airtport','Katunayaka','Sri
Lanka');
INSERT INTO Airport VALUES('USAD','USA delta International Airtport','Delta','USA');
INSERT INTO Airport VALUES('FRSA','France International Airtport','Kotcha','France');
INSERT INTO Airport VALUES('SDNY','Sydney','San Mateo','Australia');


INSERT INTO Flight VALUES('Kl649','Air Canada');
INSERT INTO Flight VALUES('RT324','Fegex');
INSERT INTO Flight VALUES('BV545','Air France');
INSERT INTO Flight VALUES('MS948','Egyptair');
INSERT INTO Flight VALUES('LK727','Sri Lankan');


INSERT INTO Flight_Schedule VALUES('Kl649','2022-10-07');
INSERT INTO Flight_Schedule VALUES('RT324','2022-06-25');
INSERT INTO Flight_Schedule VALUES('BV545','2022-12-18');
INSERT INTO Flight_Schedule VALUES('MS948','2022-11-29');
INSERT INTO Flight_Schedule VALUES('LK727','2022-11-18');
INSERT INTO Flight_Schedule VALUES('LK727','2022-11-29');
```

```sql
INSERT INTO Flight_Fare VALUES('Kl649','L56',78000.00);
INSERT INTO Flight_Fare VALUES('RT324','H28',120000.00);
INSERT INTO Flight_Fare VALUES('BV545','W43',50000.00);
INSERT INTO Flight_Fare VALUES('MS948','S79',340000.00);
INSERT INTO Flight_Fare VALUES('LK727','K27',1000000.00);


INSERT INTO Flight_Fare_Restrictions VALUES('Kl649','L56','Advance Purchase of ticket');
INSERT INTO Flight_Fare_Restrictions VALUES('RT324','H28','Ticket cost Non-Refundable if
you cancelled your journey');
INSERT INTO Flight_Fare_Restrictions VALUES('BV545','W43','Additional fees may apply');
INSERT INTO Flight_Fare_Restrictions VALUES('MS948','S79','only hand baggage is allowed to
carry in to flight cabbin');
INSERT INTO Flight_Fare_Restrictions VALUES('MS948','S79','50 KG max wheight per
passenger');
INSERT INTO Flight_Fare_Restrictions VALUES('Kl649','L56','Alcohol available only for
bussiness class');
INSERT INTO Flight_Fare_Restrictions VALUES('LK727','K27','No alcoholic bevarages');


INSERT INTO Flight_Leg VALUES('Kl649',2,'09:50:00','11:30:00','ENHK');
INSERT INTO Flight_Leg VALUES('RT324',3,'01:30:00','07:00:00','KHAB');
INSERT INTO Flight_Leg VALUES('BV545',1,'02:00:00','05:20:00','FMCH');
INSERT INTO Flight_Leg VALUES('MS948',2,'08:40:00','11:00:00','VVNB');
INSERT INTO Flight_Leg VALUES('LK727',4,'10:05:00','11:00:00','USAD');
INSERT INTO Flight_Leg VALUES('LK727',2,'03:00:00','05:00:00','FRSA');
INSERT INTO Flight_Leg VALUES('kl649',4,'02:00:00','04:00:00','SDNY');


INSERT INTO Leg_Instance VALUES('Kl649',2,'2022-10-
09',478,'DER549','ENHK','09:55:23','11:33:15');
INSERT INTO Leg_Instance VALUES('RT324',3,'2022-07-
07',345,'KOI770','KHAB','01:27:54','07:02:42');
INSERT INTO Leg_Instance VALUES('BV545',1,'2022-05-
24',231,'ZAQ248','FMCH','02:15:41','05:40:31');
INSERT INTO Leg_Instance VALUES('MS948',2,'2022-11-
01',434,'FGR425','VVNB','08:46:35','11:22:55');
INSERT INTO Leg_Instance VALUES('LK727',4,'2022-11-
18',540,'ASF182','CMBK','10:30:12','11:30:12');
INSERT INTO Leg_Instance VALUES('LK727',2,'2022-11-
29',534,'ASF182','CMBK','02:58:35','05:00:55');
INSERT INTO Leg_Instance VALUES('Kl649',4,'2022-11-
20',450,'ASF182','SDNY','01:58:35','04:00:55');



INSERT INTO Seat VALUES('Kl649',2,'2022-10-09','F167','John Williamson','+65002789224');
INSERT INTO Seat VALUES('RT324',3,'2022-07-07','A782','Jane Andrew','+97675554090');
INSERT INTO Seat VALUES('BV545',1,'2022-05-24','A996','Carles Mash','+67300223590');
INSERT INTO Seat VALUES('MS948',2,'2022-11-01','F422','Aline Corge','+33900556744');
INSERT INTO Seat VALUES('LK727',4,'2022-11-18','C001','Don jude','+94232222211');
INSERT INTO Seat VALUES('LK727',2,'2022-11-29','B120','Elise fernando','+94193028743');
```

iii. Triggers

- Trigger to prevent a new passenger from entering an existing mobile number of another passenger

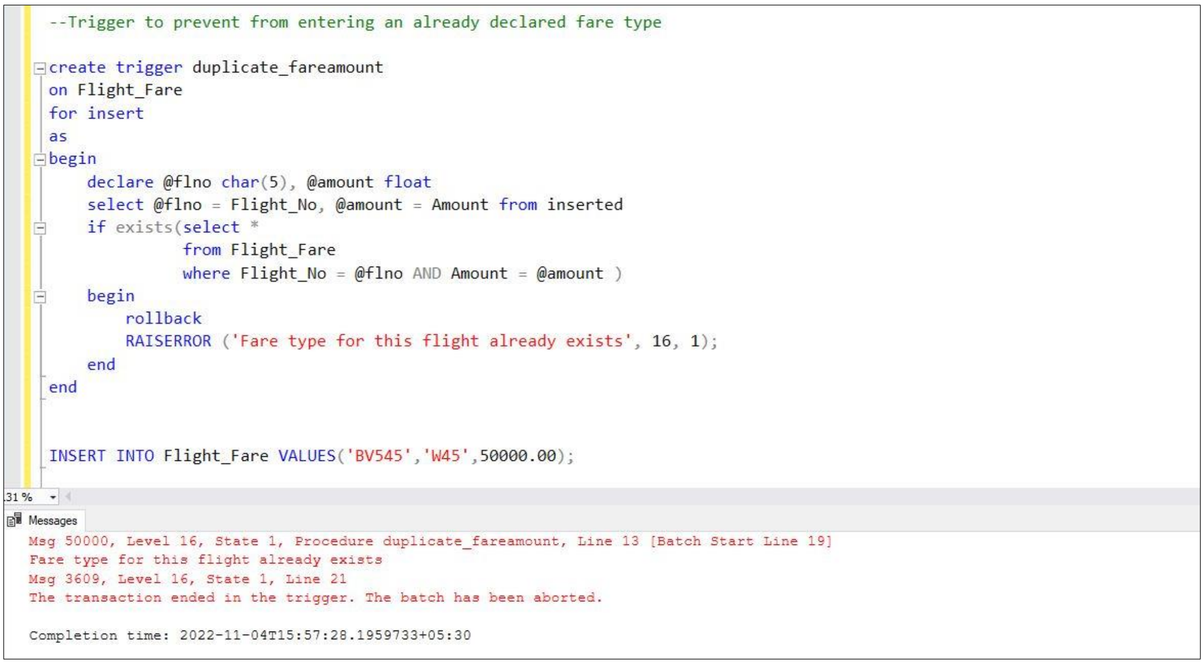


```
-- Trigger to prevent a new passenger from entering an existing mobile number of another passenger -

create trigger duplicate_mobile
on seat
for insert
as
begin
    declare @mobile char(12), @name varchar(40)
    select @name = Cust_Name, @mobile = Cust_Phone from inserted
    if exists(select *
              from seat
              where Cust_Name != @name AND Cust_Phone = @mobile )
    begin
        rollback
        RAISERROR ('This mobile number already exists', 16, 1);
    end
end


INSERT INTO Seat VALUES('MS948',2,'2022-11-01','F423','Corge Aline','+33900556744');
```

```
Msg 50000, Level 16, State 1, Procedure duplicate_mobile, Line 13 [Batch Start Line 23]
This mobile number already exists
Msg 3609, Level 16, State 1, Line 24
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2022-11-04T14:59:56.9600325+05:30
```

```
-- Trigger to prevent a new passenger from entering an existing mobile number
of another passenger --

create trigger duplicate_mobile
on seat
for insert
as
begin
        declare @mobile char(12), @name varchar(40)
        select @name = Cust_Name, @mobile = Cust_Phone from inserted
        if exists(select *
                        from seat
                        where Cust_Name != @name AND Cust_Phone = @mobile )
        begin
                rollback
                RAISERROR ('This mobile number already exists', 16, 1);
        end

end
```

- Trigger to prevent user from entering an already declared fare type



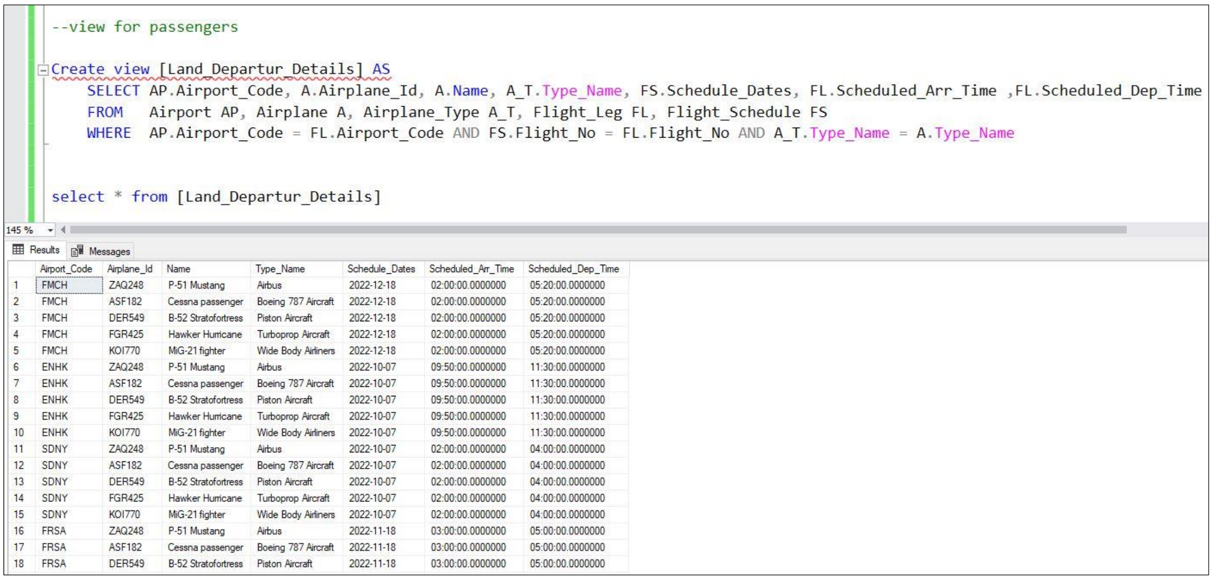
```
--Trigger to prevent from entering an already declared fare type

create trigger duplicate_fareamount
on Flight_Fare
for insert
as
begin
    declare @flno char(5), @amount float
    select @flno = Flight_No, @amount = Amount from inserted
    if exists(select *
              from Flight_Fare
              where Flight_No = @flno AND Amount = @amount )
    begin
        rollback
        RAISERROR ('Fare type for this flight already exists', 16, 1);
    end
end


INSERT INTO Flight_Fare VALUES('BV545','W45',50000.00);
```

```
Msg 50000, Level 16, State 1, Procedure duplicate_fareamount, Line 13 [Batch Start Line 19]
Fare type for this flight already exists
Msg 3609, Level 16, State 1, Line 21
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2022-11-04T15:57:28.1959733+05:30
```

```
--Trigger to prevent user from entering an already declared fare type

create trigger duplicate_fareamount
on Flight_Fare
for insert
as
begin
        declare @flno char(5), @amount float
        select @flno = Flight_No, @amount = Amount from inserted
        if exists(select *
                        from Flight_Fare
                        where Flight_No = @flno AND Amount = @amount )
        begin
                rollback
                RAISERROR ('Fare type for this flight already exists', 16, 1);
        end

end
```

iv. Views

- Passengers - to view airport, airplane details with scheduled date, arrival time and departure time.

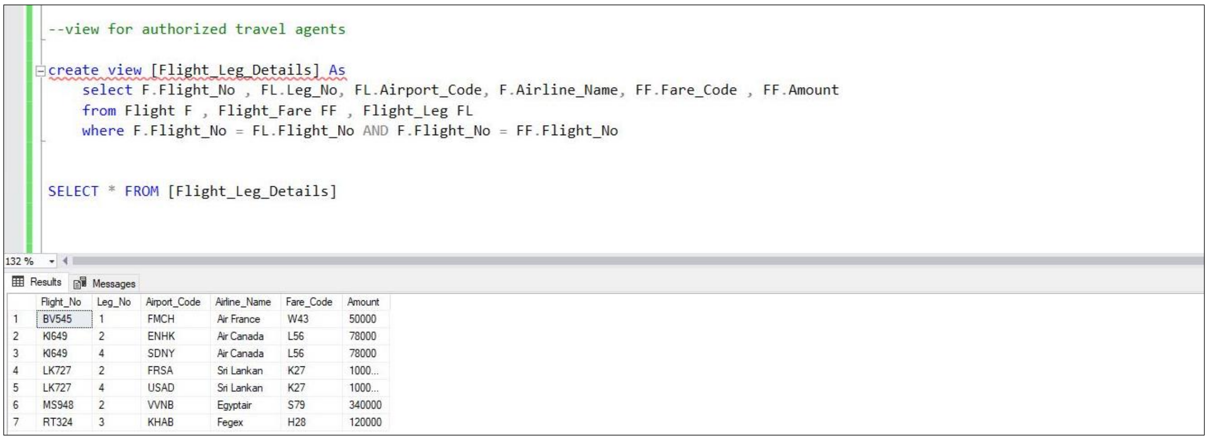


```
--view for passengers

Create view [Land_Departur_Details] AS
    SELECT AP.Airport_Code, A.Airplane_Id, A.Name, A_T.Type_Name, FS.Schedule_Dates, FL.Scheduled_Arr_Time ,FL.Scheduled_Dep_Time
    FROM   Airport AP, Airplane A, Airplane_Type A_T, Flight_Leg FL, Flight_Schedule FS
    WHERE  AP.Airport_Code = FL.Airport_Code AND FS.Flight_No = FL.Flight_No AND A_T.Type_Name = A.Type_Name


select * from [Land_Departur_Details]
```

	Airport_Code	Airplane_Id	Name	Type_Name	Schedule_Dates	Scheduled_Arr_Time	Scheduled_Dep_Time
1	FMCH	ZAQ248	P-51 Mustang	Airbus	2022-12-18	02:00:00.0000000	05:20:00.0000000
2	FMCH	ASF182	Cessna passenger	Boeing 787 Aircraft	2022-12-18	02:00:00.0000000	05:20:00.0000000
3	FMCH	DER549	B-52 Stratofortress	Piston Aircraft	2022-12-18	02:00:00.0000000	05:20:00.0000000
4	FMCH	FGR425	Hawker Hurricane	Turboprop Aircraft	2022-12-18	02:00:00.0000000	05:20:00.0000000
5	FMCH	KOI770	MiG-21 fighter	Wide Body Airliners	2022-12-18	02:00:00.0000000	05:20:00.0000000
6	ENHK	ZAQ248	P-51 Mustang	Airbus	2022-10-07	09:50:00.0000000	11:30:00.0000000
7	ENHK	ASF182	Cessna passenger	Boeing 787 Aircraft	2022-10-07	09:50:00.0000000	11:30:00.0000000
8	ENHK	DER549	B-52 Stratofortress	Piston Aircraft	2022-10-07	09:50:00.0000000	11:30:00.0000000
9	ENHK	FGR425	Hawker Hurricane	Turboprop Aircraft	2022-10-07	09:50:00.0000000	11:30:00.0000000
10	ENHK	KOI770	MiG-21 fighter	Wide Body Airliners	2022-10-07	09:50:00.0000000	11:30:00.0000000
11	SDNY	ZAQ248	P-51 Mustang	Airbus	2022-10-07	02:00:00.0000000	04:00:00.0000000
12	SDNY	ASF182	Cessna passenger	Boeing 787 Aircraft	2022-10-07	02:00:00.0000000	04:00:00.0000000
13	SDNY	DER549	B-52 Stratofortress	Piston Aircraft	2022-10-07	02:00:00.0000000	04:00:00.0000000
14	SDNY	FGR425	Hawker Hurricane	Turboprop Aircraft	2022-10-07	02:00:00.0000000	04:00:00.0000000
15	SDNY	KOI770	MiG-21 fighter	Wide Body Airliners	2022-10-07	02:00:00.0000000	04:00:00.0000000
16	FRSA	ZAQ248	P-51 Mustang	Airbus	2022-11-18	03:00:00.0000000	05:00:00.0000000
17	FRSA	ASF182	Cessna passenger	Boeing 787 Aircraft	2022-11-18	03:00:00.0000000	05:00:00.0000000
18	FRSA	DER549	B-52 Stratofortress	Piston Aircraft	2022-11-18	03:00:00.0000000	05:00:00.0000000

```
Create view [Land_Departur_Details] AS
SELECT AP.Airport_Code, A.Airplane_Id, A.Name, A_T.Type_Name, FS.Schedule_Dates,
FL.Scheduled_Arr_Time ,FL.Scheduled_Dep_Time
FROM   Airport AP, Airplane A, Airplane_Type A_T, Flight_Leg FL, Flight_Schedule
FS
WHERE  AP.Airport_Code = FL.Airport_Code AND FS.Flight_No = FL.Flight_No AND
A_T.Type_Name = A.Type_Name
```

- Authorized travel agents - to view flight details with relevant fare types and amounts.



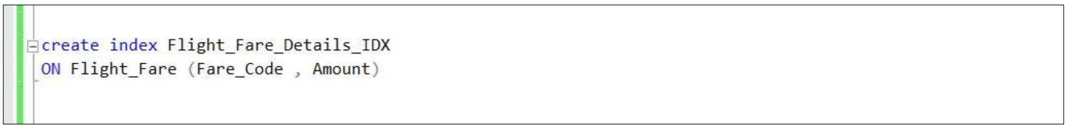
```
--view for authorized travel agents

create view [Flight_Leg_Details] As
    select F.Flight_No , FL.Leg_No, FL.Airport_Code, F.Airline_Name, FF.Fare_Code , FF.Amount
    from Flight F , Flight_Fare FF , Flight_Leg FL
    where F.Flight_No = FL.Flight_No AND F.Flight_No = FF.Flight_No


SELECT * FROM [Flight_Leg_Details]
```

	Flight_No	Leg_No	Airport_Code	Airline_Name	Fare_Code	Amount
1	BV545	1	FMCH	Air France	W43	50000
2	KI649	2	ENHK	Air Canada	L56	78000
3	KI649	4	SDNY	Air Canada	L56	78000
4	LK727	2	FRSA	Sri Lankan	K27	1000...
5	LK727	4	USAD	Sri Lankan	K27	1000...
6	MS948	2	VVNB	Egyptair	S79	340000
7	RT324	3	KHAB	Fegex	H28	120000

```
create view [Flight_Leg_Details] As
select F.Flight_No , FL.Leg_No, FL.Airport_Code, F.Airline_Name, FF.Fare_Code,
FF.Amount
from Flight F , Flight_Fare FF , Flight_Leg FL
where F.Flight_No = FL.Flight_No AND F.Flight_No = FF.Flight_No
```

v. Indexes

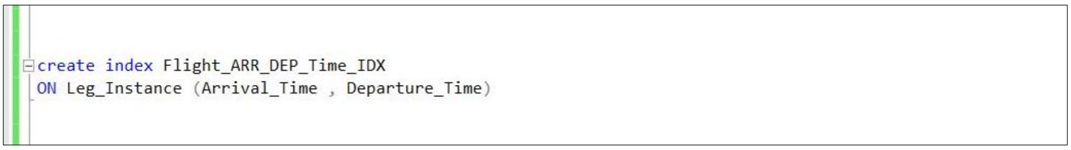
- Index 01 for flight fare details



```
create index Flight_Fare_Details_IDX
ON Flight_Fare (Fare_Code , Amount)
```

```
create index Flight_Fare_Details_IDX
ON Flight_Fare (Fare_Code , Amount)
```

- Index 02 for flight arrival and departure times.

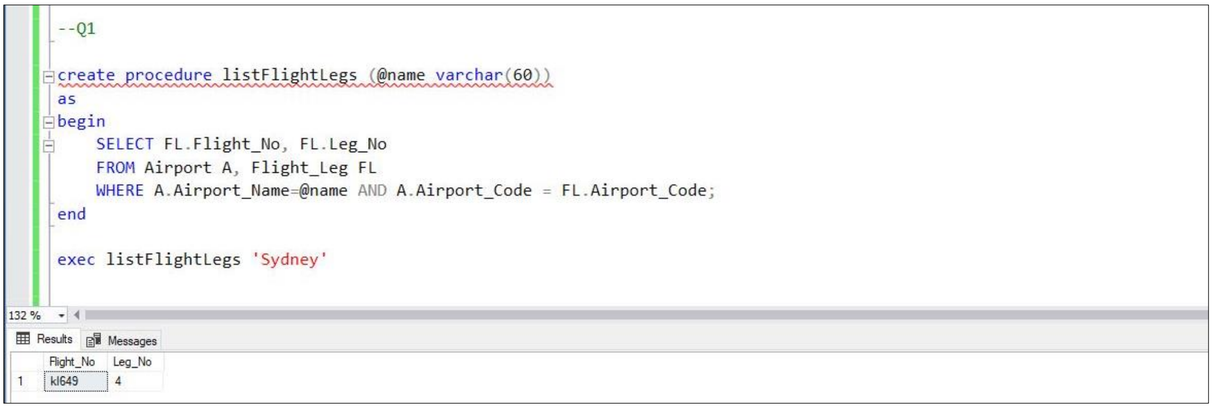


```
create index Flight_ARR_DEP_Time_IDX
ON Leg_Instance (Arrival_Time , Departure_Time)
```

```
create index Flight_ARR_DEP_Time_IDX
ON Leg_Instance (Arrival_Time , Departure_Time)
```

vi. Stored Procedures

- Q1

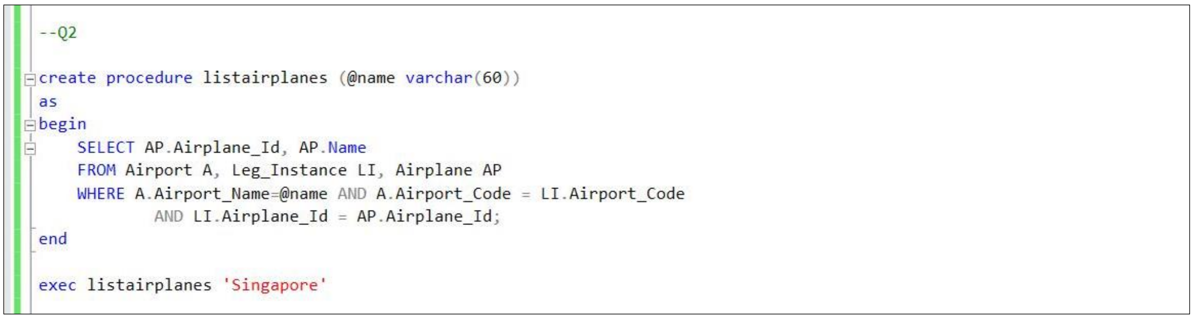


```
--Q1

create procedure listFlightLegs (@name varchar(60))
as
begin
    SELECT FL.Flight_No, FL.Leg_No
    FROM Airport A, Flight_Leg FL
    WHERE A.Airport_Name=@name AND A.Airport_Code = FL.Airport_Code;
end

exec listFlightLegs 'Sydney'
```

	Flight_No	Leg_No
1	kl649	4

```
create procedure listFlightLegs (@name varchar(60))
as
begin
        SELECT FL.Flight_No, FL.Leg_No
        FROM Airport A, Flight_Leg FL
        WHERE A.Airport_Name=@name AND A.Airport_Code = FL.Airport_Code;
end

exec listFlightLegs 'Sydney'
```

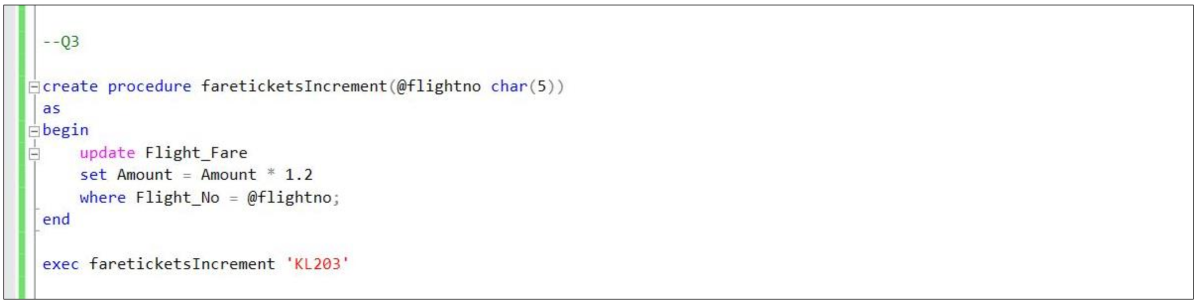
- Q2



```
--Q2

create procedure listairplanes (@name varchar(60))
as
begin
    SELECT AP.Airplane_Id, AP.Name
    FROM Airport A, Leg_Instance LI, Airplane AP
    WHERE A.Airport_Name=@name AND A.Airport_Code = LI.Airport_Code
            AND LI.Airplane_Id = AP.Airplane_Id;
end

exec listairplanes 'Singapore'
```

```
create procedure listairplanes (@name varchar(60))
as
begin
        SELECT AP.Airplane_Id, AP.Name
        FROM Airport A, Leg_Instance LI, Airplane AP
        WHERE A.Airport_Name=@name AND A.Airport_Code = LI.Airport_Code
                      AND LI.Airplane_Id = AP.Airplane_Id;
end

exec listairplanes 'Singapore'
```

- Q3



```
--Q3

create procedure fareticketsIncrement(@flightno char(5))
as
begin
    update Flight_Fare
    set Amount = Amount * 1.2
    where Flight_No = @flightno;
end

exec fareticketsIncrement 'KL203'
```

```
create procedure fareticketsIncrement(@flightno char(5))
as
begin
       update Flight_Fare
       set Amount = Amount * 1.2
       where Flight_No = @flightno;
end

exec fareticketsIncrement 'KL203'
```

- Q4

```
--Q4

create procedure listflights (@name varchar(40))
as
begin
    SELECT Flight_No, Leg_No, Date
    FROM Seat
    WHERE Cust_Name=@name
    ORDER BY date DESC;
end

exec listflights 'Mary Ann'
```

```
create procedure listflights (@name varchar(40))
as
begin
       SELECT Flight_No, Leg_No, Date
       FROM Seat
       WHERE Cust_Name=@name
       ORDER BY date DESC;
end

exec listflights 'Mary Ann'
```

B. Part II

1. Database Vulnerabilities

o Weak Authentication

Weak Authentication refers to any case in which the authentication mechanism's strength is relatively weak in regard to the value of the resources being defended. It also outlines cases where the authentication system is defective or susceptible.

Techniques

- **Brute Force -** The attacker inputs username/password combinations continuously until he obtains one that works. Simple guessing or systematic enumeration of all potential username/password combinations may be used in the brute force technique. An attacker will tend to utilize automated tools to speed up the brute force attack.
- **Social Engineering -** An attack strategy in which the attacker uses the inherent human instinct to trust to persuade others to disclose their login credentials. For example, an attacker may act as an IT manager over the phone and seek login credentials for "system maintenance."
- **Direct Credential Theft -** The attacker gains access to password files or discovers a piece of paper which the authorized user has mentioned the ID and password.

Impacts

Once your account has been compromised due to a failed authentication vulnerability, the attacker may do everything you have clearance to do, which can have major implications for your company's reputation.

Even gaining access to one administrator account allows attackers to compromise a whole online application. The consequences of session hijacking can vary from data breaches, identity theft, sensitive information leaks, to administrative access, depending on the goal of the compromised program.

- Denial of Service

Techniques

- **Data corruption -** Any unintended modification to a file that occurs during storage, transfer, or processing.
- **Network flooding -** When a router employs a non-adaptive routing method to transmit an incoming packet to all outbound links except the node from where the packet received.
- **Server resource overload**
- **Crash a server**

Impacts

- A DDoS assault may cost money, time, clients, and even company reputation. Depending on the intensity of an assault, resources may be unavailable for 24 hours, several days, or even a week.
- No workers may access network resources during an assault.
- Server and hosting issues can be occurred.

2. Mitigation Strategies & Countermeasures

○ Weak Authentication

How to mitigate

- Adopting a solid password policy and constantly enforcing it across all apps.
- When the risk level permits it, use two-factor or multi-factor authentication.
- Including a framework for industry-standard authentication.
- Implementing risk-based authentication and raising difficulties as needed.
- Assuring that authentication is required before accessing any application resources.
- Maintaining the authentication token safe and its lifetime restricted.

Strong Authentication - The most powerful practicable authentication techniques and regulations should be used. When feasible, two-factor authentication (tokens, certificates, biometrics, etc.) is preferred. Unfortunately, cost and usability difficulties frequently render two-factor authentication impracticable. Strong username/password regulations (minimum length, character variety, obscurity) should be applied in such instances.

Secure Sphere Authentication Protections - Unfortunately, despite the best efforts of effective authentication, malfunctions do occur on occasion. Password restrictions are disregarded; a fortunate attacker may be able to brute force even a strong password; a legacy authentication mechanism may be used for practical reasons; and so on. Secure Sphere's Dynamic Profiling, Failed Login Recognition, and Authentication Evaluation offer generally applicable authentication protection in these scenarios.

Countermeasures

- ○ **Dynamic Profiling -** Dynamic Profiling records a variety of user characteristics that detect exploited login credentials automatically. Client IP addresses, hostnames, operating system usernames, and client applications are examples of these properties. When attempting to utilize stolen credentials, the previously reported attacker who gained login credentials by acting as an IT administrator could generate many Secure Sphere notifications. The attacker's hostname, OS username, and potentially even IP address do not correspond to the profile of the true holder of the obtained login credentials.

- **Failed Login Detection -** To avoid brute force attacks, Secure Sphere's Failed Login Detection automatically imposes an unsuccessful database login limitation (count and period).
- **Password Policy Assessment -** Secure Sphere assesses password policy rules implemented by the database as part of its ongoing assessment capability. Secure Sphere, for example, can determine whether the database server enforces password length, character variety, and reset intervals.

○ Denial of Service

How to mitigate

- **Move to the cloud** - While shifting to the cloud will not prevent DDoS assaults, it will help to minimize them. The cloud, for example, has greater bandwidth than on-premises assets, and the cloud's structure implies that numerous servers are not situated in the same location.
- **Make use of anti-DDoS hardware and software** - DDoS assaults have existed for a long, and some types of attacks are quite prevalent. There are several tools available to repel or mitigate specific protocol and application threats, for example. Make use of their resources.
- **Make the network more resistant -** DDoS assaults should be as resistant to your infrastructure as feasible. This includes more than firewalls, as some DDoS assaults target firewalls. Consider placing data centers on various networks, ensuring that not all of data centers reside in the same geographical area, locating servers in multiple data centers, and ensuring that there are no traffic blockages in the network.

Countermeasures

Connection Controls minimizes server resource excess by restricting each database user's connection speeds, query levels, and other factors.

IPS and Protocol Validation prohibit attackers from creating DOS by using known software vulnerabilities. Buffer overflow is a frequent platform vulnerability that has the potential to crash database systems.

Dynamic Profiling enables query access control automatically to identify any illegal searches that may result in DOS. DOS assaults, for example, targeting platform vulnerabilities are likely to result in both IPS and Dynamic Profile breaches. Secure Sphere may reach unrivaled precision by correlating these infractions. For a more detailed discussion of Dynamic Profiling, see the section on Excessive Privilege Abuse in this article.

- **Response Timing -** Database DOS attacks that aim to overburden server resources cause database replies to be delayed. The Response Timing function in Secure Sphere identifies slowdowns in both single query answers and the whole system.
- **Reduce the time it takes to establish a connection**.
- **Use an Intrusion Detection System (IDS) on the network**.

Individual contribution

	Student Name	Works done
1	Dissanayake W.P.D.B (Leader)	• Identified entities , type of them and attributes from dB scenario. • Drew the finalized ERD. • Made Assumptions. • Helped to Normalize the Logical model. • Created tables in SQL with sample data with constraints. • Created stored procedures and triggers. • Helped to identify vulnerabilities of db. • Checked the finalized document
2	Zakey M.S.M. A	• Identified entities, type of them and attributes from dB scenario. • Helped to draw the ERD. • Made Assumptions. • Helped to Normalize the Logical model. • Created tables in SQL with sample data with constraints. • Created the finalized Report. • Created Index and views. • Helped to identify vulnerabilities of db. •

3	Dilhara W. M. A.	• Helped to identify entities and attributes from dB scenario. • Made Assumptions. • Helped to Normalize the Logical model. • Created tables in SQL with sample data with constraints. • Created stored procedures and triggers. • Identified vulnerabilities of dB and created the report to part 2.
4	Pemachandra T.H.R.T.	• Identified entities, type of them and attributes from dB scenario. • Helped to draw the ERD. • Made Assumptions. • Helped to Normalize the Logical model. • Drew the Normalized logical schema. • Created tables in SQL with sample data with constraints. • Created stored procedures and triggers.