Sri Lanka Institute of Information Technology

Online Airline Reservation System

Group Assignment

IE2042- Database Management Systems for Security

Group Details:

	Student Registration Number	Student Name
1	IT21170416	Pemarathna I.R.C
2	IT21163340	Peiris H.R.T
3	IT21175602	Nilakshana H.A.N
4	IT21188572	Fernando S.D.K.

Table of Contents

Part 01

1. Assumptions

1) Flight_Leg entity is a weak entity of the Flight entity.

2) Leg_instance entity is a weak entity of the Flight Leg entity.

3) Seat entity is a weak entity of the Leg instance entity.

4) Flight_fare entity is a weak entity of the flight.

5) Flight_ No can determined the schedule_arr_time and schedule_dep_time .

   Flight_No → schedule_arr_time,schedule_dep_time

6) AirplaneID can determine the arr_time and dep_time

   AirplaneID → arr_time, dep_time

7) The AirplaneID in the Airplane table is the same as the AirplaneID in the Leg_instance table, the AirplaneID in the Leg_instance Table is the same as the AirplaneID in the Airplane_Schedule Table, so the AirplaneID in the Airplane Table is the same as the AirplaneID in the Airplane_Schedule table.Airplane A, Leg_instance L, Airplane_Schedule AS

   A.AirplaneID = ~~L. AirplaneID~~

   ~~L. AirplaneID~~ = AS.AirplaneID

   A. AirplaneID = AS.AirplaneID

8) The Flight_ No in the Flight table is the same as the Flight_ No in the Flight_Leg table, the Flight_ No in the Flight_Leg Table is the same as the Flight_ No in the Flight_Schedule Table, so the Flight_ No in the Flight Table is the same as the Flight_ No in the Flight_Schedule.

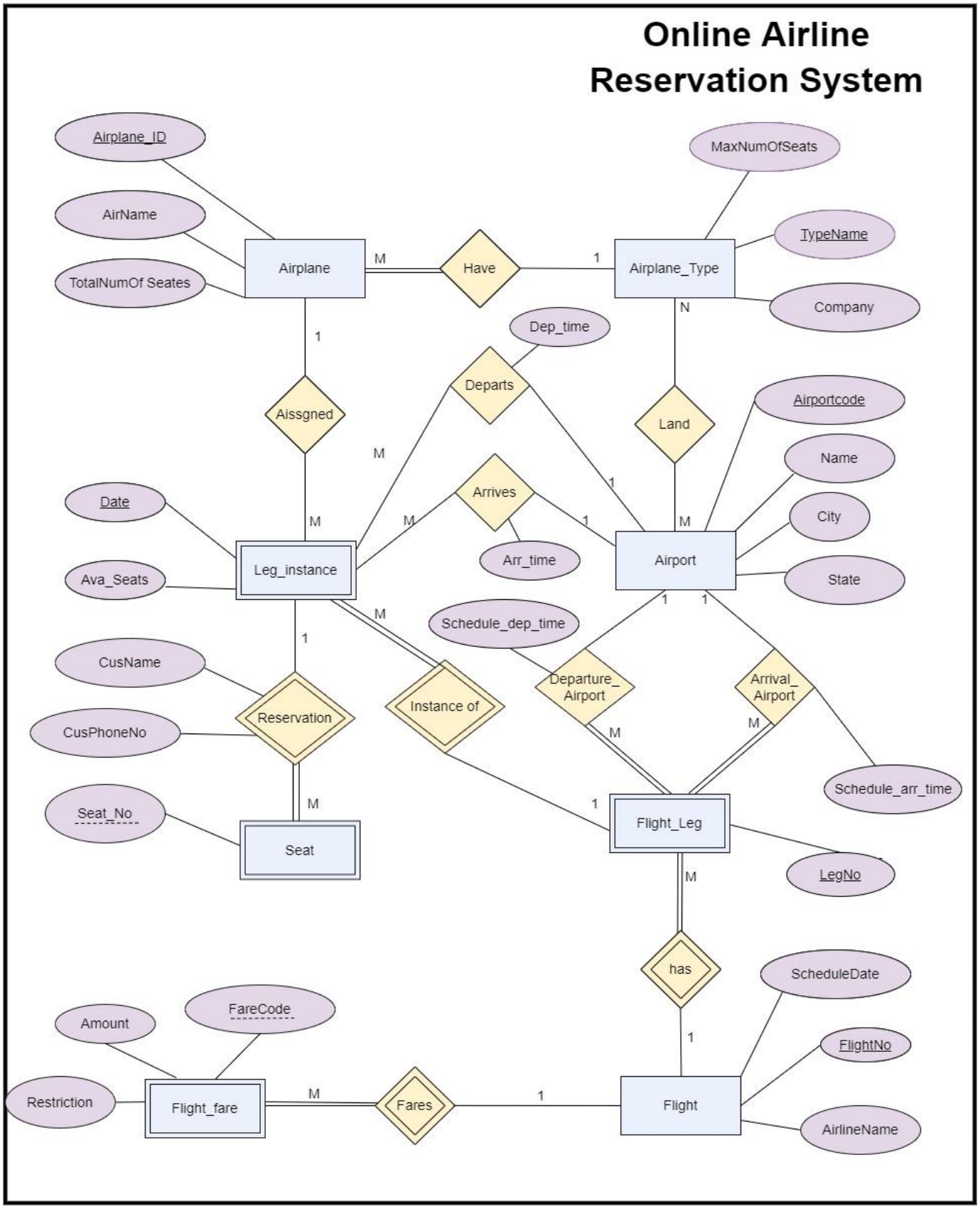
   Flight F, Flight_Leg FL, Flight_Schedule FS

   F. Flight_ No = ~~FL. Flight_ No~~

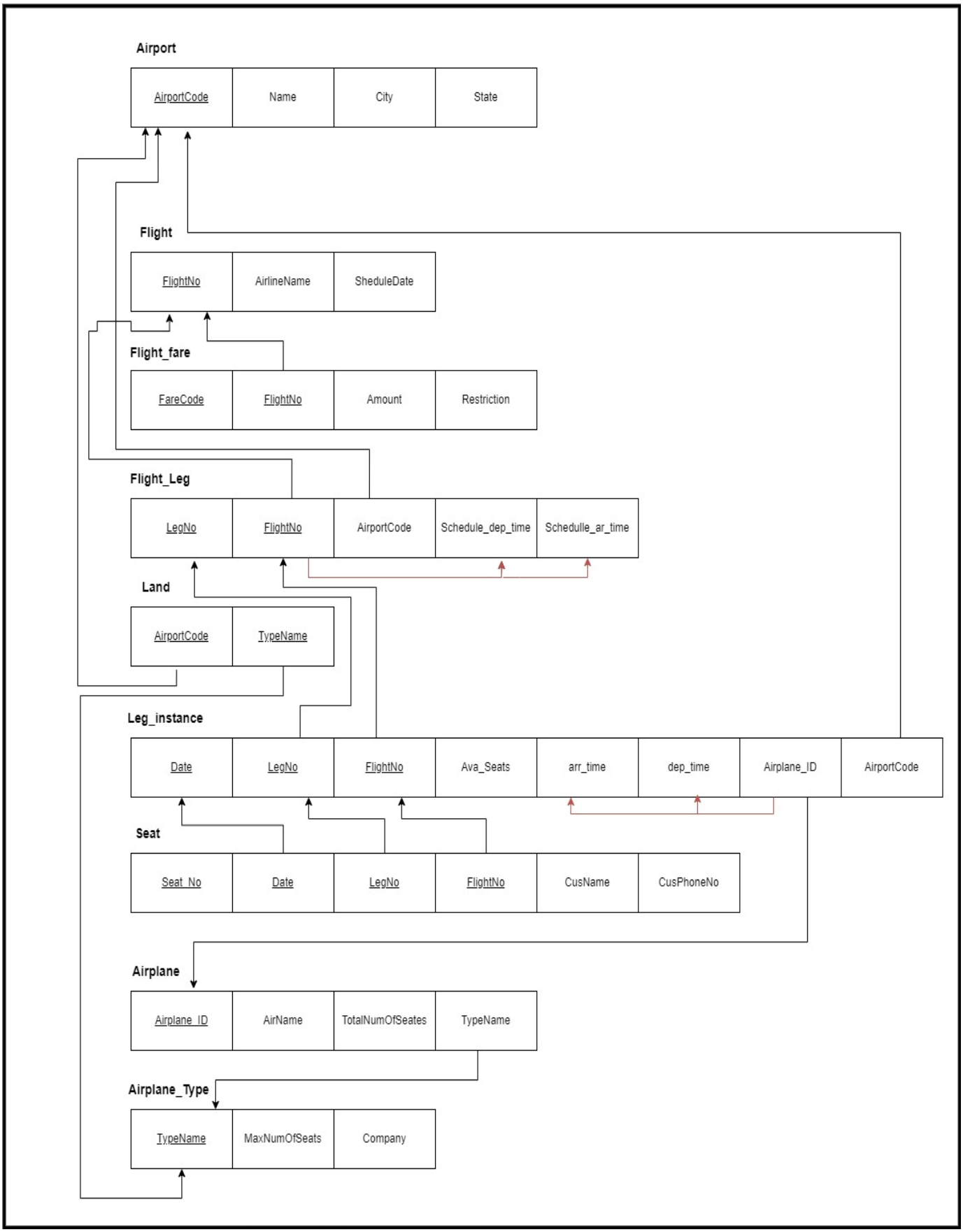
   ~~FL. Flight_ N~~o = FS. Flight_ No

   F. Flight_ No = FS. Flight_ No

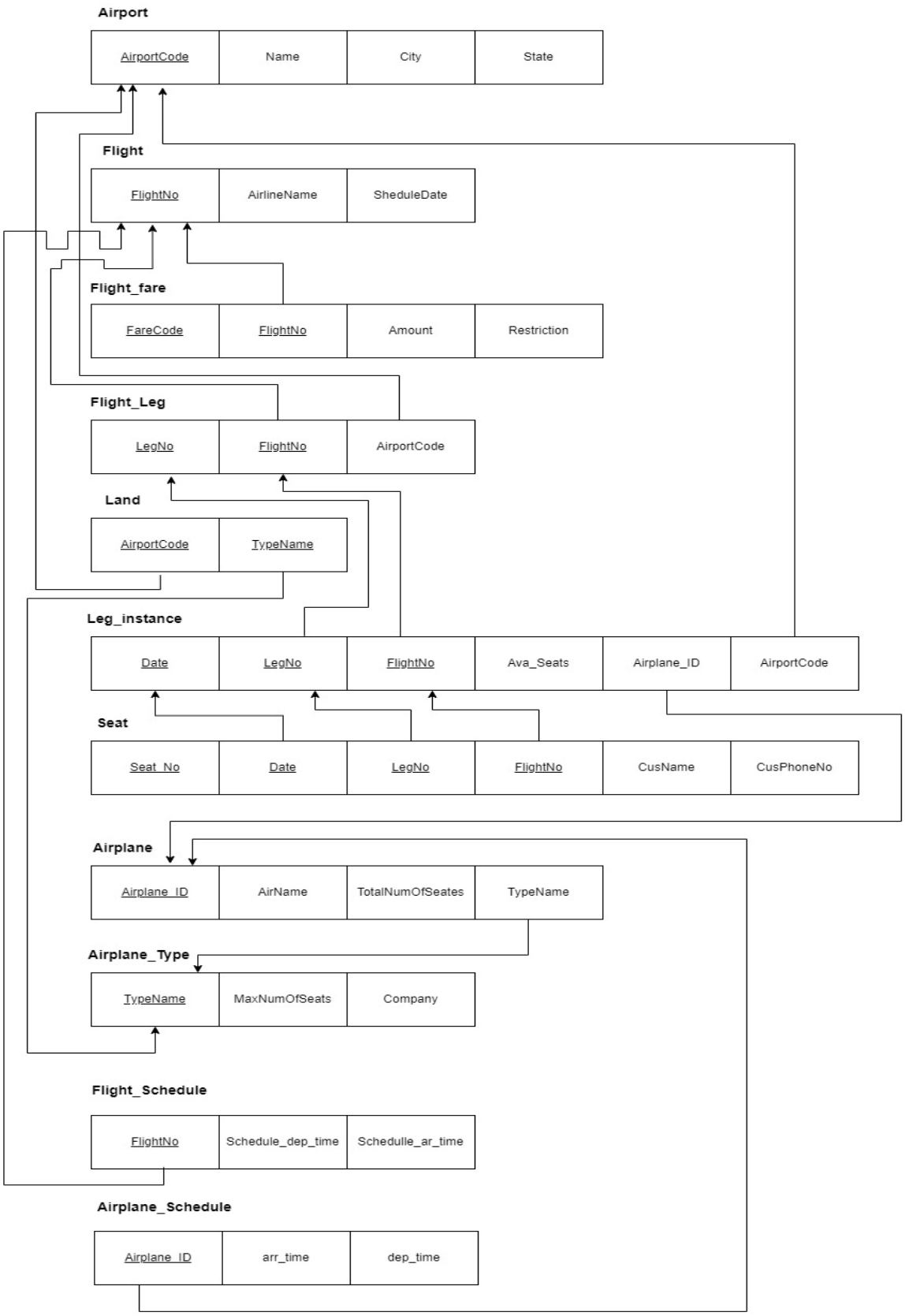
2. ERD Model


Online Airline Reservation System
Airplane_ID
AirName
TotalNumOf Seates
Airplane
M
Have
1
Airplane_Type
MaxNumOfSeats
TypeName
Company
N
1
Aissgned
Dep_time
Departs
Land
Airportcode
Name
City
State
Airport
M
1
Arrives
Arr_time
Date
Ava_Seats
Leg_instance
M
M
M
1
Schedule_dep_time
Departure_ Airport
Arrival_ Airport
Instance of
CusName
CusPhoneNo
Reservation
Seat_No
Seat
M
1
Flight_Leg
Schedule_arr_time
LegNo
M
has
ScheduleDate
FlightNo
AirlineName
Flight
1
Amount
FareCode
Restriction
Flight_fare
M
Fares
1

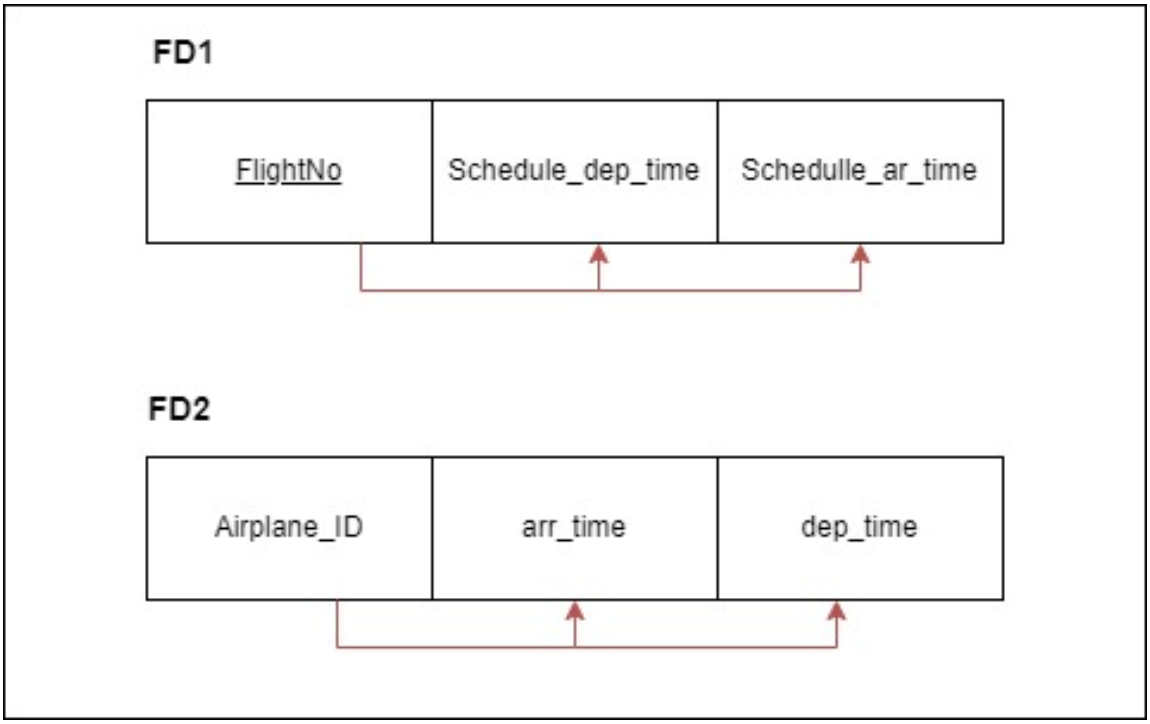
3. Logical Model



4. Normalized Logical Model



Functional Dependencies



5. MS SQL server and Suitable sample data

i. SQL Screenshots

```sql
--Creating Airport Table--

Create table Airport(
    Airportcode varchar(10) NOT NULL,
    Name varchar(50) NOT NULL,
    City varchar(50) NOT NULL,
    State varchar(50) NOT NULL,
    constraint Airport_pk primary key (Airportcode),
    CONSTRAINT checkAirport_Airportcode check (Airportcode LIKE '[A-Z][0-9][0-9][0-9][0-9][0-9]')
);

--Creating Airplane_Type Table--

Create table Airplane_Type(
    TypeName varchar(50) NOT NULL,
    MaxNumOfSeats varchar(50) NOT NULL,
    Company varchar(50) NOT NULL,
    constraint Airplane_Type_pk primary key (TypeName),
);

--Creating Land Table--

Create table Land(
    Airportcode varchar(10) NOT NULL,
    TypeName varchar(50) NOT NULL,
    constraint Land_pk primary key (Airportcode,TypeName),
    constraint Land_fk1 foreign key (Airportcode) REFERENCES Airport (Airportcode),
    constraint Land_fk2 foreign key (TypeName) REFERENCES Airplane_Type (TypeName),
);
```

```sql
--Creating Airplane Table--

Create table Airplane(
    AirplaneID varchar(10) NOT NULL,
    AirName varchar(50) NOT NULL,
    TotalNumOfSeats integer,
    TypeName varchar(50) NOT NULL,
    constraint airplane_pk primary key (AirplaneID),
    constraint airplane_fk foreign key (TypeName) REFERENCES Airplane_Type (TypeName),
    constraint checkAirplane_AirplaneID check (AirplaneID LIKE '[A-Z][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
);

--Creating Flight Table--

Create table Flight(
    FlightNo varchar(50) NOT NULL,
    AirlineName varchar(50) NOT NULL,
    ScheduleDate varchar(50) NOT NULL,
    constraint Flight_pk primary key (FlightNo),
    constraint checkFlight_FlightNo check (FlightNo LIKE '[A-Z][A-Z][0-9][0-9][0-9]')
);

--Creating Flight_Leg Table--

Create table Flight_Leg(
    LegNo varchar(10) NOT NULL,
    FlightNo varchar(50) NOT NULL,
    Airportcode varchar(10) NOT NULL,
    constraint Flight_Leg_pk primary key (LegNo,FlightNo),
    constraint Flight_Leg_fk1 foreign key (Airportcode) REFERENCES Airport (Airportcode),
    constraint Flight_Leg_fk2 foreign key (FlightNo) REFERENCES Flight (FlightNo)
);
```

```sql
--Creating Leg_instance Table--

Create table Leg_instance(
    Date date NOT NULL,
    LegNo varchar(10) NOT NULL,
    FlightNo varchar(50) NOT NULL,
    Ava_Seats varchar(50) NOT NULL,
    AirplaneID varchar(10) NOT NULL,
    Airportcode varchar(10) NOT NULL,
    constraint Leg_instance_pk primary key (Date,LegNo,FlightNo),
    constraint Leg_instance_fk1 foreign key (LegNo,FlightNo) REFERENCES Flight_Leg (LegNo,FlightNo),
    constraint Leg_instance_fk2 foreign key (AirplaneID) REFERENCES Airplane (AirplaneID),
    constraint Flight_Leg_fk3 foreign key (Airportcode) REFERENCES Airport (Airportcode),
);


--Creating Seat Table--

Create table Seat(
    Seat_No varchar(10) NOT NULL,
    Date date NOT NULL,
    LegNo varchar(10) NOT NULL,
    FlightNo varchar(50) NOT NULL,
    CusName varchar(50) NOT NULL,
    CusPhoneNo varchar(50) NOT NULL,
    constraint Seat_pk primary key (Seat_No,Date,LegNo,FlightNo),
    constraint check_Seat_CusPhoneNo check (CusPhoneNo LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    constraint Seat_fk foreign key (Date,LegNo,FlightNo) REFERENCES Leg_instance (Date,LegNo,FlightNo)
);

```

```sql
--Creating Flight_fare Table--

Create table Flight_fare(
    Farecode varchar(20) NOT NULL,
    FlightNo varchar(50) NOT NULL,
    Amount varchar(20) NOT NULL,
    Restriction varchar(50) NOT NULL,
    constraint Flight_fare_pk primary key (Farecode,FlightNo),
    constraint Flight_fare_fk foreign key (FlightNo) REFERENCES Flight (FlightNo)
);

--Creating Airplane_Schedule Table--

CREATE TABLE Airplane_Schedule (
    AirplaneID varchar (10),
    arr_time time NOT NULL,
    dep_time time NOT NULL,
    constraint Airplane_Schedule_PK primary key (AirplaneID),
    constraint Airplane_Schedule_fk foreign key (AirplaneID) REFERENCES Airplane (AirplaneID)
);

--Creating Flight_Schedule Table--

CREATE TABLE Flight_Schedule (
    FlightNo varchar (50) NOT NULL,
    schedule_arr_time time(0) NOT NULL,
    schedule_dep_time time(0) NOT NULL,
    constraint Flight_schedule_PK primary key (FlightNo),
    constraint Flight_Schedule_fk foreign key (FlightNo) REFERENCES Flight (FlightNo)
);
```

```sql
/*insert airport table*/
insert into Airport values('A020171','Sydney','San Mateo','California');
insert into Airport values('B020221','Frankfurt Airport','Frankfurt','Hesse');
insert into Airport values('C020271','Singapore','Salt Lake City','Utah');
insert into Airport values('D020251','Helsinki Airport','Ivalo','Kemi-Tornio');
insert into Airport values('E020230','Brisbane International Airport','Brisbane','Victoria');
insert into Airport values('F171223','Bandaranaike International Airport','katunayaka','Colombo');


/*insert Airplane_Type*/
insert into Airplane_Type values('Airlines 787','300',' Rolls Royce');
insert into Airplane_Type values('Airbus A330','600','Pratt & Whitney');
insert into Airplane_Type values('Boeing 747','400','Pratt & Whitney');
insert into Airplane_Type values('Boeing 757','500','Miami-based Eastern Airlines');
insert into Airplane_Type values('Douglas 80','200','Pratt & Whitney JT8-D');
insert into Airplane_Type values('Antonov AN28','350','LET Aircraft Industries');


/*insert Land*/
insert into Land values('A020171','Airlines 787');
insert into Land values('B020221','Airbus A330');
insert into Land values('C020271','Boeing 747');
insert into Land values('D020251','Boeing 757');
insert into Land values('E020230','Douglas 80');
insert into Land values('F171223','Antonov AN28');


/*insert Airplane*/
insert into Airplane values('R200171223','Douglas DC-3',295,'Boeing 757');
insert into Airplane values('N987654321','Airbus A321XLR ',467,'Boeing 747');
insert into Airplane values('Q224466889','COMAC C919',440,'Airbus A330');
insert into Airplane values('D779955331','Universal Hydrogen ATR 72',174,'Antonov AN28');
```

```sql
/*insert Flight*/
insert into Flight values('AL987','Air Berlin','2022-12-23');
insert into Flight values('TY456','Belair','2022-10-19');
insert into Flight values('KL203','Paramount','2022-11-04');
insert into Flight values('QE333','Oman Air','2022-10-26');
insert into Flight values('BH912','IndiGo','2023-01-02');
insert into Flight values('LK729','Jetstar Asia','2022-10-23');
insert into Flight values('PR914','SL Airways','2022-12-01');


/*insert Flight_leg*/
insert into Flight_leg values('2','AL987','A020171');
insert into Flight_leg values('3','PR914','B020221');
insert into Flight_leg values('4','AL987','C020271');
insert into Flight_leg values('1','QE333','D020251');
insert into Flight_leg values('2','LK729','E020230');
insert into Flight_leg values('3','BH912','F171223');


/*insert Leg_instance*/
insert into Leg_instance values('2022-11-14','2','AL987','74','R200171223','A020171');
insert into Leg_instance values('2022-12-23','3','PR914','52','N987654321','B020221');
insert into Leg_instance values('2023-01-01','2','LK729','158','D779955331','E020230');
insert into Leg_instance values('2022-12-14','1','QE333','74','Q224466889','D020251');


/*insert seat*/
insert into seat values('11F','2022-11-14','2','AL987','Kaun Perera','0859632147');
insert into seat values('55JH','2022-12-23','3','PR914','Niki watson','0856329741');
insert into seat values('14LA','2023-01-01','2','LK729','Mary Ann','1253698521');
insert into seat values('13HA','2022-12-14','1','QE333','Sadun perera','9476966518');
```

```sql
/*insert Flight_fare*/
insert into Flight_fare values('AF1015','AL987','650000','plastic');
insert into Flight_fare values('CDIJ17','TY456','275000','bring metal');
insert into Flight_fare values('GNQS03','KL203','74000','bring pets and plants');
insert into Flight_fare values('PW4589','QE333','65000','eat fish and meats');
insert into Flight_fare values('YBHKL1','PR914','54000','bring over 50KG travel bags');
insert into Flight_fare values('NQS144','BH912','41000','bring pets');
insert into Flight_fare values('NOQS51','LK729','41265','bring gold,spicies');


/*insert Airplane_Schedule*/
insert into Airplane_Schedule values('D779955331','09:28:48','15:45:47');
insert into Airplane_Schedule values('N987654321','01:11:18','08:15:27');
insert into Airplane_Schedule values('R200171223','11:31:48','19:45:47');
insert into Airplane_Schedule values('Q224466889','13:07:09','23:55:47');


/*insert Flight_Schedule*/
insert into Flight_Schedule values('AL987','08.07.09','14:58:14');
insert into Flight_Schedule values('TY456','00:14:15','07:13:17');
insert into Flight_Schedule values('KL203','10:27:26','18:08:16');
insert into Flight_Schedule values('QE333','12:48:17','00:05:05');
insert into Flight_Schedule values('BH912','07:22:26','14:08:16');
insert into Flight_Schedule values('PR914','17:48:17','04:05:05');
```

```sql

select *
from Airport;

select*
from Airplane_Type;

select*
from Land

select*
from Airplane

select*
from Flight

select*
from Flight_leg
```

100 %

Results | Messages

	Airportcode	Name	City	State
1	A020171	Sydney	San Mateo	California
2	B020221	Frank...	Frankfurt	Hesse
3	C020271	Sing...	Salt Lake...	Utah
4	D020251	Helsi...	Ivalo	Kemi-T...
5	E020230	Brisb...	Brisbane	Victoria
6	F171223	Band...	katunaya...	Colombo

	TypeName	MaxNumOfSeats	Company
1	Airbus A330	600	Pratt & Whitney
2	Airlines 787	300	Rolls Royce
3	Antonov A...	350	LET Aircraft I...
4	Boeing 747	400	Pratt & Whitney
5	Boeing 757	500	Miami-based...
6	Douglas 80	200	Pratt & Whitn...

	Airportcode	TypeName
1	A020171	Airlines 787

ii. SQL codes as a separate script

```
--Creating Airport Table--

Create table Airport(
	Airportcode varchar(10) NOT NULL,
	Name varchar(50) NOT NULL,
	City varchar(50) NOT NULL,
	State varchar(50) NOT NULL,
	constraint Airport_pk primary key (Airportcode),
	CONSTRAINT checkAirport_Airportcode check (Airportcode LIKE '[A-Z][0-9][0-9][0-
9][0-9][0-9][0-9]')
);

--Creating Airplane_Type Table--

Create table Airplane_Type(
	TypeName varchar(50) NOT NULL,
	MaxNumOfSeats varchar(50) NOT NULL,
	Company varchar(50) NOT NULL,
	constraint Airplane_Type_pk primary key (TypeName),
);

--Creating Airplane_Type Table--

Create table Land(
	Airportcode varchar(10) NOT NULL,
	TypeName varchar(50) NOT NULL,
	constraint Land_pk primary key (Airportcode,TypeName),
	constraint Land_fk1 foreign key (Airportcode) REFERENCES Airport (Airportcode),
	constraint Land_fk2 foreign key (TypeName) REFERENCES Airplane_Type (TypeName),
);


--Creating Airplane Table--

Create table Airplane(
	AirplaneID varchar(10) NOT NULL,
	AirName varchar(50) NOT NULL,
	TotalNumOfSeats integer,
	TypeName varchar(50) NOT NULL,
	constraint airplane_pk primary key (AirplaneID),
	constraint airplane_fk foreign key (TypeName) REFERENCES Airplane_Type
(TypeName),
	constraint checkAirplane_AirplaneID check (AirplaneID LIKE '[A-Z][0-9][0-9][0-
9][0-9][0-9][0-9][0-9][0-9][0-9]')
);

--Creating Flight Table--

Create table Flight(
	FlightNo varchar(50) NOT NULL,
	AirlineName varchar(50) NOT NULL,
	ScheduleDate varchar(50) NOT NULL,
	constraint Flight_pk primary key (FlightNo),
	constraint checkFlight_FlightNo check (FlightNo LIKE '[A-Z][A-Z][0-9][0-9][0-
9]')
);
```

```sql
--Creating Flight_Leg Table--

Create table Flight_Leg(
	LegNo varchar(10) NOT NULL,
	FlightNo varchar(50) NOT NULL,
	Airportcode varchar(10) NOT NULL,
	constraint Flight_Leg_pk primary key (LegNo,FlightNo),
	constraint Flight_Leg_fk1 foreign key (Airportcode) REFERENCES Airport
(Airportcode),
	constraint Flight_Leg_fk2 foreign key (FlightNo) REFERENCES Flight (FlightNo)
);


--Creating Leg_instance Table--

Create table Leg_instance(
	Date date NOT NULL,
	LegNo varchar(10) NOT NULL,
	FlightNo varchar(50) NOT NULL,
	Ava_Seats varchar(50) NOT NULL,
	AirplaneID varchar(10) NOT NULL,
	Airportcode varchar(10) NOT NULL,
	constraint Leg_instance_pk primary key (Date,LegNo,FlightNo),
	constraint Leg_instance_fk1 foreign key (LegNo,FlightNo) REFERENCES Flight_Leg
(LegNo,FlightNo),
	constraint Leg_instance_fk2 foreign key (AirplaneID) REFERENCES Airplane
(AirplaneID),
	constraint Flight_Leg_fk3 foreign key (Airportcode) REFERENCES Airport
(Airportcode),
);


--Creating Seat Table--

Create table Seat(
	Seat_No varchar(10) NOT NULL,
	Date date NOT NULL,
	LegNo varchar(10) NOT NULL,
	FlightNo varchar(50) NOT NULL,
	CusName varchar(50) NOT NULL,
	CusPhoneNo varchar(50) NOT NULL,
	constraint Seat_pk primary key (Seat_No,Date,LegNo,FlightNo),
	constraint check_Seat_CusPhoneNo check (CusPhoneNo LIKE '[0-9][0-9][0-9][0-
9][0-9][0-9][0-9][0-9][0-9][0-9]'),
	constraint Seat_fk foreign key (Date,LegNo,FlightNo) REFERENCES Leg_instance
(Date,LegNo,FlightNo)
);


--Creating Flight_fare Table--

Create table Flight_fare(
	Farecode varchar(20) NOT NULL,
	FlightNo varchar(50) NOT NULL,
	Amount varchar(20) NOT NULL,
	Restriction varchar(50) NOT NULL,
	constraint Flight_fare_pk primary key (Farecode,FlightNo),
	constraint Flight_fare_fk foreign key (FlightNo) REFERENCES Flight (FlightNo)
);
```

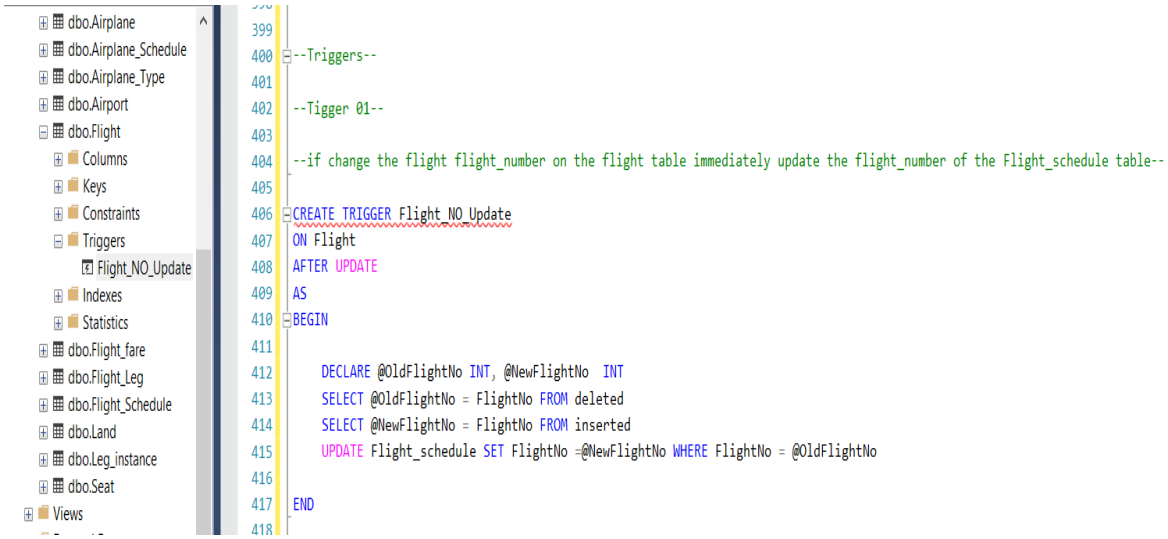
```sql
--Creating Airplane_Schedule Table--

CREATE TABLE Airplane_Schedule (
	AirplaneID varchar (10),
	arr_time time NOT NULL,
	dep_time time NOT NULL,
	constraint Airplane_Schedule_PK primary key (AirplaneID),
	constraint Airplane_Schedule_fk foreign key (AirplaneID) REFERENCES Airplane
(AirplaneID)
);

--Creating Flight_Schedule Table--

CREATE TABLE Flight_Schedule (
	FlightNo varchar (50) NOT NULL,
	schedule_arr_time time(0) NOT NULL,
	schedule_dep_time time(0) NOT NULL,
	constraint Flight_schedule_PK primary key (FlightNo),
	constraint Flight_Schedule_fk foreign key (FlightNo) REFERENCES Flight
(FlightNo)
);


/*insert airport table*/

insert into Airport values('A020171','Sydney','San Mateo','California');
insert into Airport values('B020221','Frankfurt Airport','Frankfurt','Hesse');
insert into Airport values('C020271','Singapore','Salt Lake City','Utah');
insert into Airport values('D020251','Helsinki Airport','Ivalo','Kemi-Tornio');
insert into Airport values('E020230','Brisbane International
Airport','Brisbane','Victoria');
insert into Airport values('F171223','Bandaranaike International
Airport','katunayaka','Colombo');


/*insert Airplane_Type*/
insert into Airplane_Type values('Airlines 787','300',' Rolls Royce');
insert into Airplane_Type values('Airbus A330','600','Pratt & Whitney');
insert into Airplane_Type values('Boeing 747','400','Pratt & Whitney');
insert into Airplane_Type values('Boeing 757','500','Miami-based Eastern Airlines');
insert into Airplane_Type values('Douglas 80','200','Pratt & Whitney JT8-D');
insert into Airplane_Type values('Antonov AN28','350','LET Aircraft Industries');


/*insert Land*/
insert into Land values('A020171','Airlines 787');
insert into Land values('B020221','Airbus A330');
insert into Land values('C020271','Boeing 747');
insert into Land values('D020251','Boeing 757');
insert into Land values('E020230','Douglas 80');
insert into Land values('F171223','Antonov AN28');


/*insert Airplane*/
insert into Airplane values('R200171223','Douglas DC-3',295,'Boeing 757');
insert into Airplane values('N987654321','Airbus A321XLR ',467,'Boeing 747');
insert into Airplane values('Q224466889','COMAC C919',440,'Airbus A330');
insert into Airplane values('D779955331','Universal Hydrogen ATR 72',174,'Antonov
AN28');
```

```sql
/*insert Flight*/
insert into Flight values('AL987','Air Berlin','2022-12-23');
insert into Flight values('TY456','Belair','2022-10-19');
insert into Flight values('KL203','Paramount','2022-11-04');
insert into Flight values('QE333','Oman Air','2022-10-26');
insert into Flight values('BH912','IndiGo','2023-01-02');
insert into Flight values('LK729','Jetstar Asia','2022-10-23');
insert into Flight values('PR914','SL Airways','2022-12-01');


/*insert Flight_leg*/
insert into Flight_leg values('2','AL987','A020171');
insert into Flight_leg values('3','PR914','B020221');
insert into Flight_leg values('4','AL987','C020271');
insert into Flight_leg values('1','QE333','D020251');
insert into Flight_leg values('2','LK729','E020230');
insert into Flight_leg values('3','BH912','F171223');


/*insert Leg_instance*/
insert into Leg_instance values('2022-11-14','2','AL987','74','R200171223','A020171');
insert into Leg_instance values('2022-12-23','3','PR914','52','N987654321','B020221');
insert into Leg_instance values('2023-01-
01','2','LK729','158','D779955331','E020230');
insert into Leg_instance values('2022-12-14','1','QE333','74','Q224466889','D020251');


/*insert seat*/
insert into seat values('11F','2022-11-14','2','AL987','Kaun Perera','0859632147');
insert into seat values('55JH','2022-12-23','3','PR914','Niki watson','0856329741');
insert into seat values('14LA','2023-01-01','2','LK729','Mary Ann','1253698521');
insert into seat values('13HA','2022-12-14','1','QE333','Sadun perera','9476966518');


/*insert Flight_fare*/
insert into Flight_fare values('AF1015','AL987','650000','plastic');
insert into Flight_fare values('CDIJ17','TY456','275000','bring metal');
insert into Flight_fare values('GNQS03','KL203','74000','bring pets and plants');
insert into Flight_fare values('PW4589','QE333','65000','eat fish and meats');
insert into Flight_fare values('YBHKL1','PR914','54000','bring over 50KG travel
bags');
insert into Flight_fare values('NQS144','BH912','41000','bring pets');
insert into Flight_fare values('NOQS51','LK729','41265','bring gold,spicies');


/*insert Airplane_Schedule*/
insert into Airplane_Schedule values('D779955331','09:28:48','15:45:47');
insert into Airplane_Schedule values('N987654321','01:11:18','08:15:27');
insert into Airplane_Schedule values('R200171223','11:31:48','19:45:47');
insert into Airplane_Schedule values('Q224466889','13:07:09','23:55:47');


/*insert Flight_Schedule*/
insert into Flight_Schedule values('AL987','08.07.09','14:58:14');
insert into Flight_Schedule values('TY456','00:14:15','07:13:17');
insert into Flight_Schedule values('KL203','10:27:26','18:08:16');
insert into Flight_Schedule values('QE333','12:48:17','00:05:05');
insert into Flight_Schedule values('BH912','07:22:26','14:08:16');
insert into Flight_Schedule values('PR914','17:48:17','04:05:05');
```

iii. Triggers

i. Trigger For Flight Schedule Table

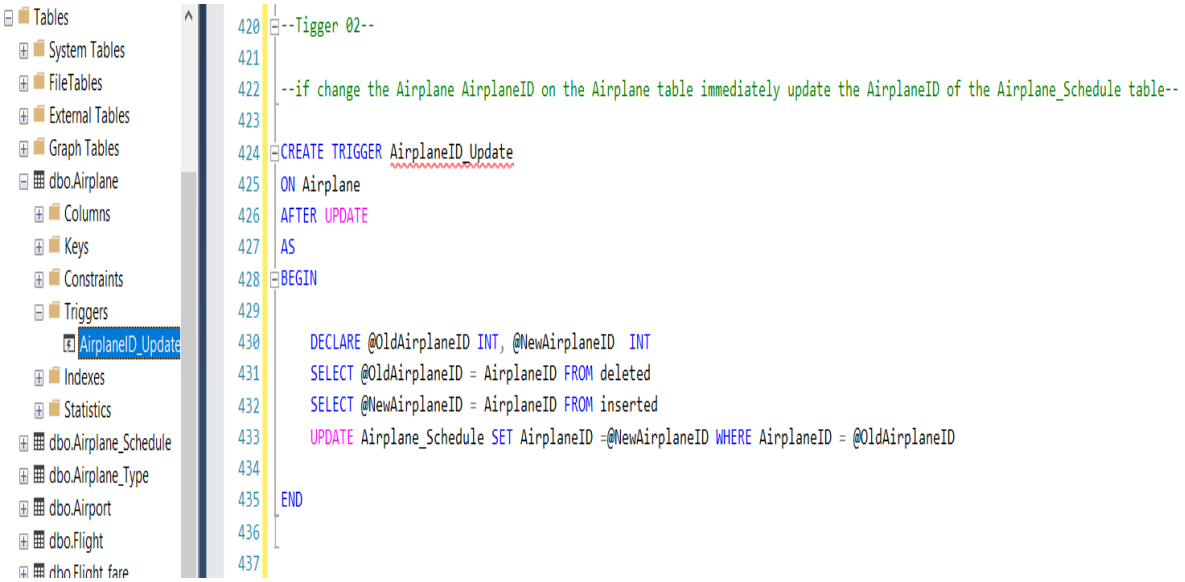
If change the flight number on the flight table immediately update the flight number on the Flight schedule table




```
--Tigger 01--

--if change the flight_number on the flight table immediately update the flight_number
of the Flight_schedule table--

CREATE TRIGGER Flight_NO_Update
ON Flight
AFTER UPDATE
AS
BEGIN

        DECLARE @OldFlightNo INT, @NewFlightNo  INT
        SELECT @OldFlightNo = FlightNo FROM deleted
        SELECT @NewFlightNo = FlightNo FROM inserted
        UPDATE Flight_schedule SET FlightNo =@NewFlightNo WHERE FlightNo = @OldFlightNo


END
```

ii. Trigger For Airplane Schedule table

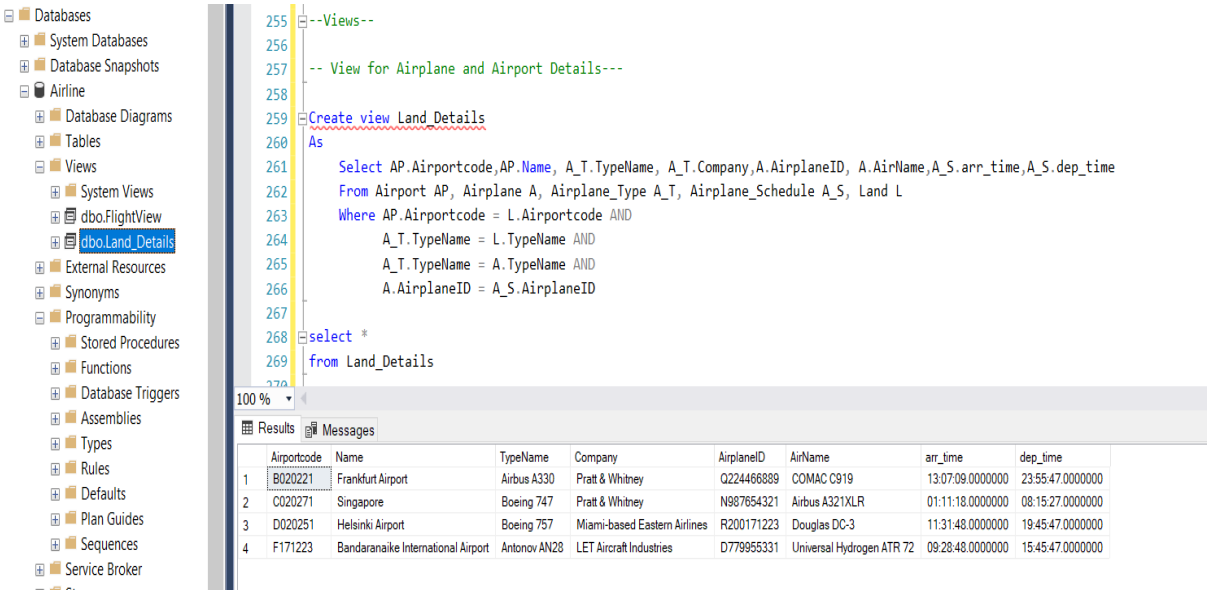
If change the Airplane ID on the Airplane table immediately update the Airplane ID of the Airplane Schedule table




```
--Tigger 02--

--if change the AirplaneID on the Airplane table immediately update the AirplaneID of
the Airplane_Schedule table--

CREATE TRIGGER AirplaneID_Update
ON Airplane
AFTER UPDATE
AS
BEGIN

        DECLARE @OldAirplaneID INT, @NewAirplaneID  INT
        SELECT @OldAirplaneID = AirplaneID FROM deleted
        SELECT @NewAirplaneID = AirplaneID FROM inserted
        UPDATE Airplane_Schedule SET AirplaneID =@NewAirplaneID WHERE AirplaneID =
        @OldAirplaneID

END
```

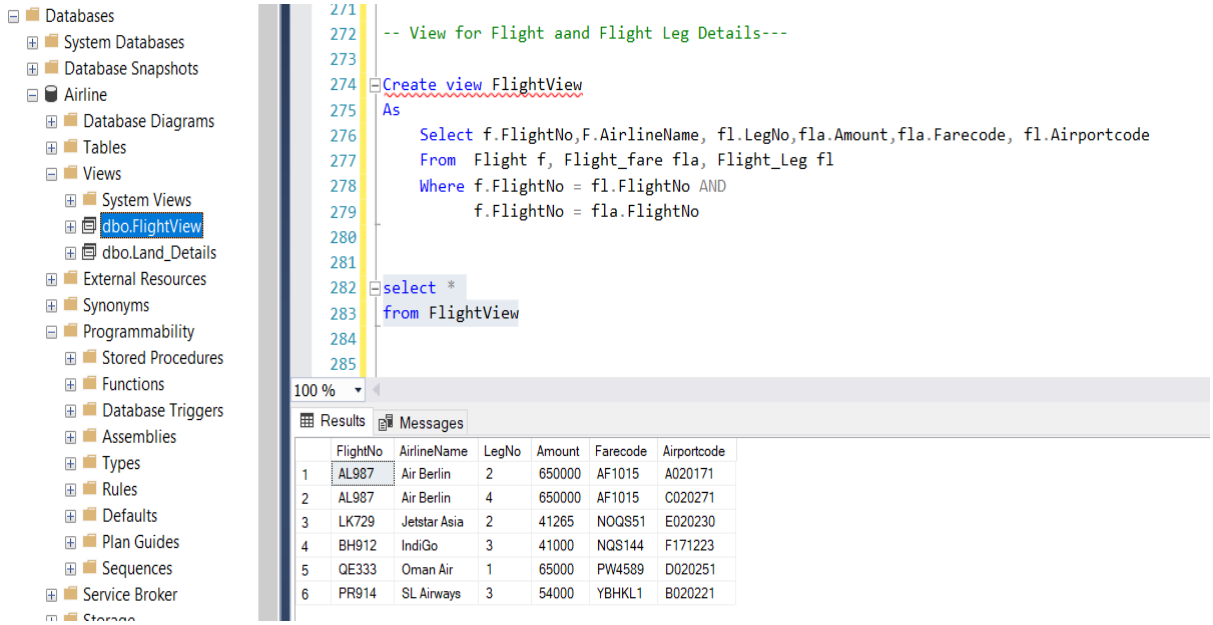
iv. Views

i. View for Airplane and Airport Details




```
-- View for Airplane and Airport Details---

Create view Land_Details
As
    Select AP.Airportcode,AP.Name, A_T.TypeName, A_T.Company,A.AirplaneID,
    A.AirName,A_S.arr_time,A_S.dep_time
    From Airport AP, Airplane A, Airplane_Type A_T, Airplane_Schedule A_S, Land L
    Where AP.Airportcode = L.Airportcode AND
            A_T.TypeName = L.TypeName AND
            A_T.TypeName = A.TypeName AND
            A.AirplaneID = A_S.AirplaneID

select *

from Land_Details
```

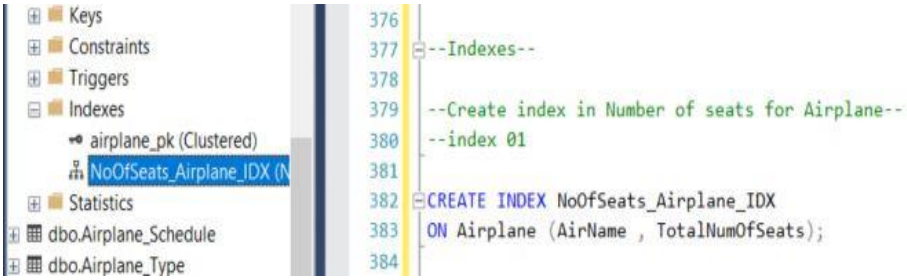
ii. View for Flight and Flight Leg Details




```
-- View for Flight and Flight Leg Details---

Create view FlightView
As
        Select f.FlightNo,F.AirlineName, fl.LegNo,fla.Amount,fla.Farecode,
fl.Airportcode
        From  Flight f, Flight_fare fla, Flight_Leg fl
        Where f.FlightNo = fl.FlightNo AND
                f.FlightNo = fla.FlightNo


select *
from FlightView
```

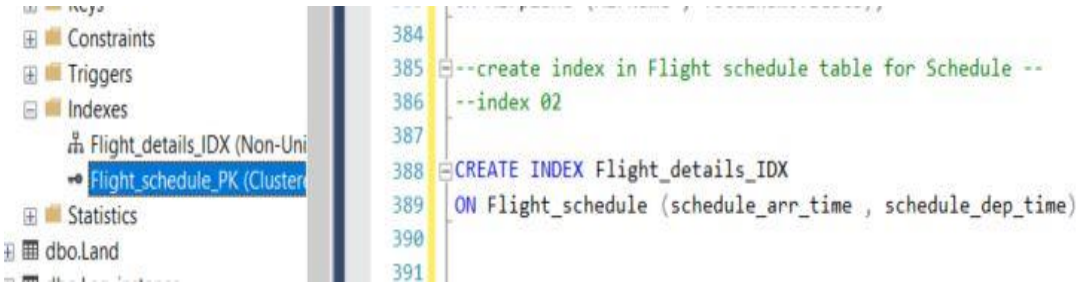
v. Indexes

i. Create an index in the Number of seats for an Airplane




```
--Create an index in the Number of seats for Airplane--
--index 01--

CREATE INDEX NoOfSeats_Airplane_IDX
ON Airplane (AirName , TotalNumOfSeats);
```

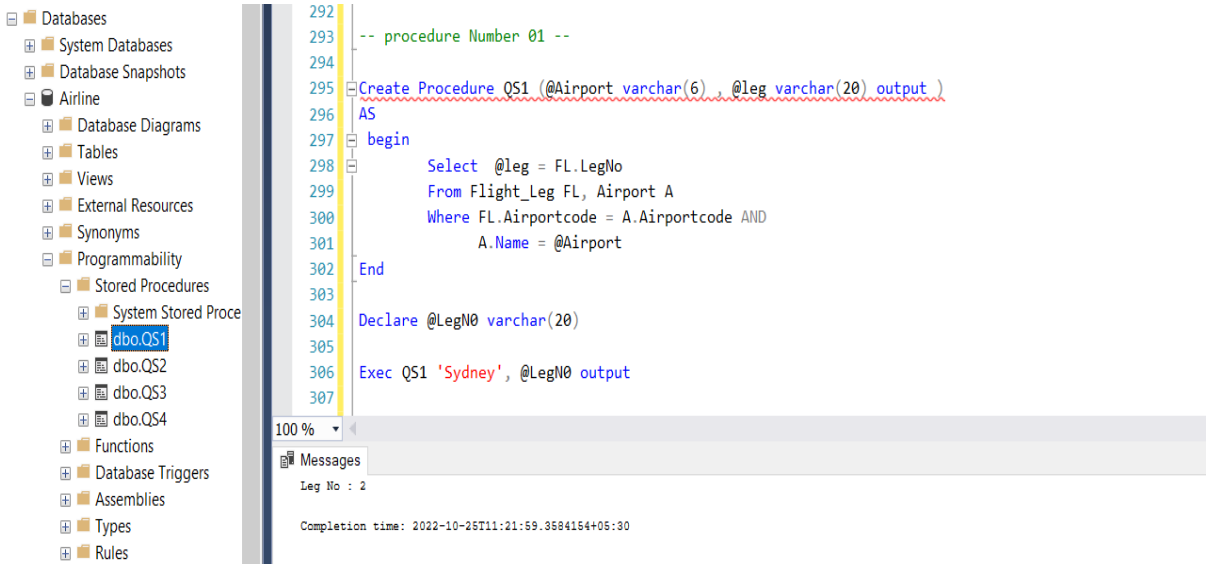
ii. Create index in Flight schedule table for the Schedule




```
--create index in Flight schedule table for Schedule --
--index 02--

CREATE INDEX Flight_details_IDX
ON Flight_schedule (schedule_arr_time , schedule_dep_time)
```

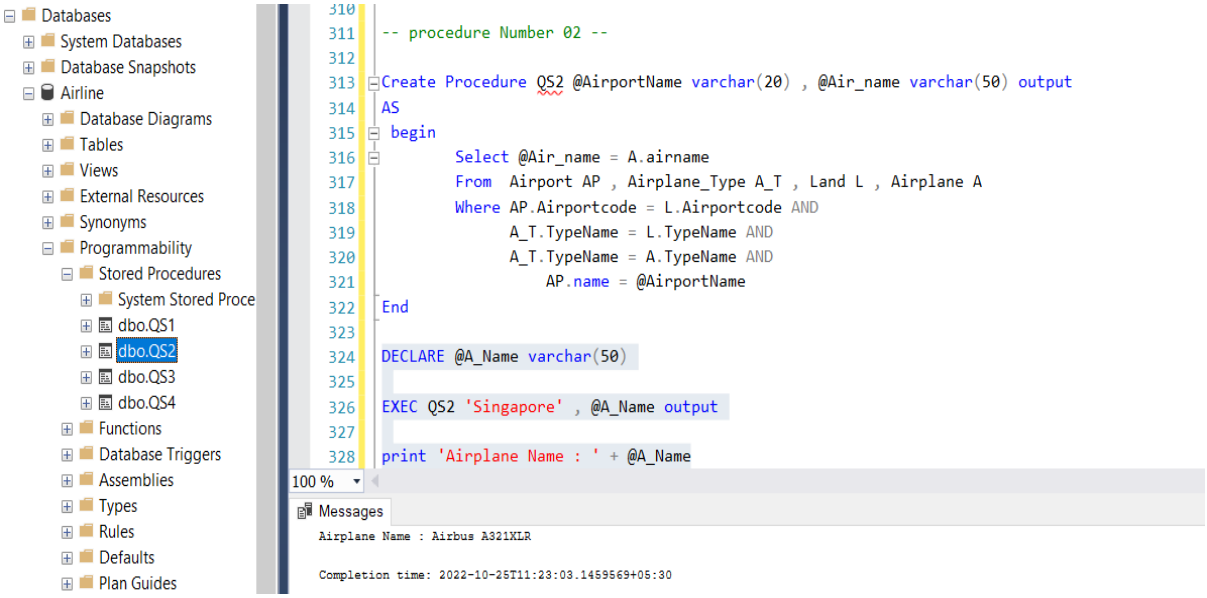
iii. Stored Procedures

1.



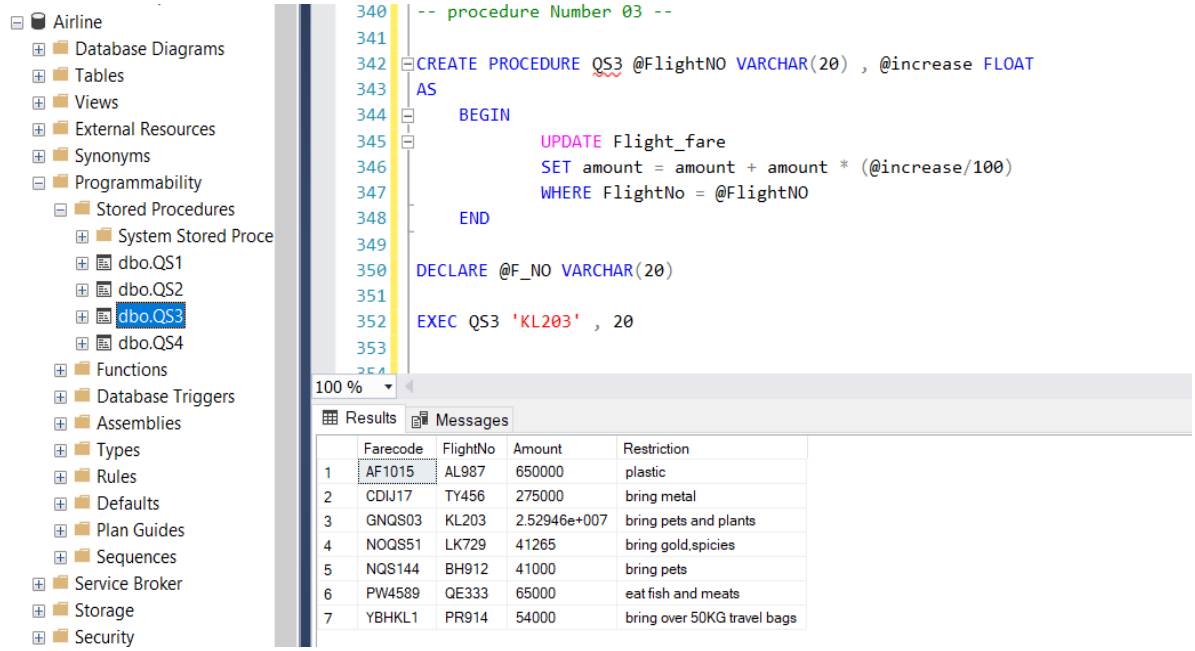

```
-- procedure Number 01 --

Create Procedure QS1 (@Airport varchar(6) , @leg varchar(20) output )
AS
 begin
              Select  @leg = FL.LegNo
              From Flight_Leg FL, Airport A
              Where FL.Airportcode = A.Airportcode AND
                       A.Name = @Airport
End

Declare @LegN0 varchar(20)

Exec QS1 'Sydney', @LegN0 output

Print 'Leg No : ' + @LegN0
```

2.



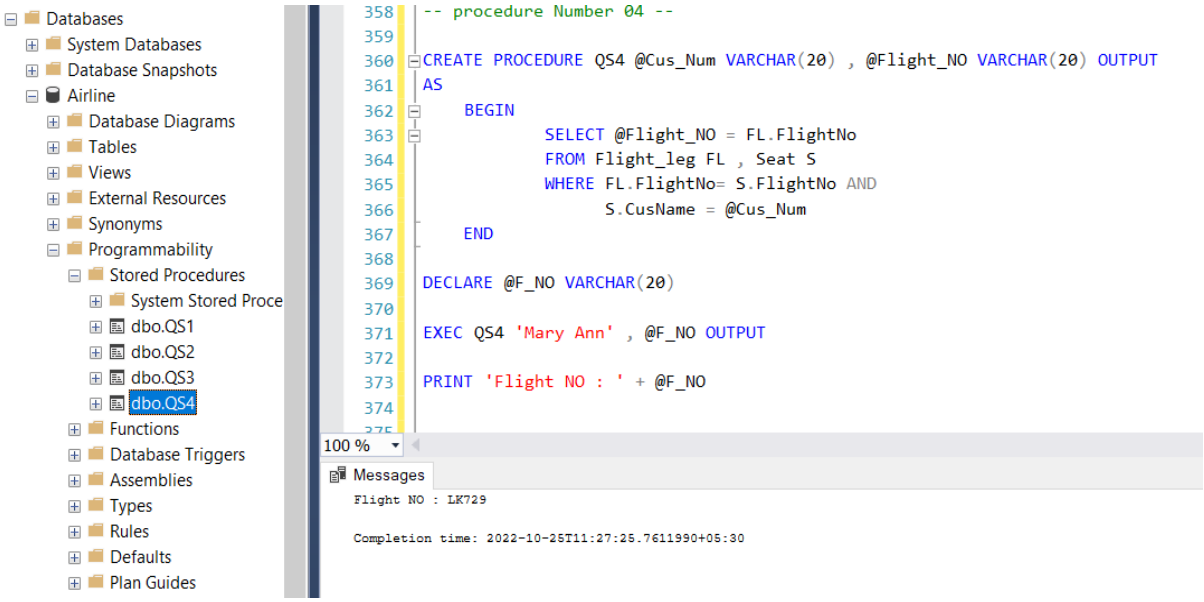

```
-- procedure Number 02 --

Create Procedure QS2 @AirportName varchar(20) , @Air_name varchar(50) output
AS
 begin
             Select @Air_name = A.airname
             From  Airport AP , Airplane_Type A_T , Land L , Airplane A
             Where AP.Airportcode = L.Airportcode AND
                   A_T.TypeName = L.TypeName AND
                   A_T.TypeName = A.TypeName AND
                    AP.name = @AirportName
End

DECLARE @A_Name varchar(50)

EXEC QS2 'Singapore' , @A_Name output


print 'Airplane Name : ' + @A_Name
```

3.




```
-- procedure Number 03 --

CREATE PROCEDURE QS3 @FlightNO VARCHAR(20) , @increase FLOAT
AS
       BEGIN
                    UPDATE Flight_fare
                    SET amount = amount + amount * (@increase/100)
                    WHERE FlightNo = @FlightNO
       END

DECLARE @F_NO VARCHAR(20)

EXEC QS3 'KL203' , 20


select *
from Flight_fare
```

4.




```
-- procedure Number 04 --

CREATE PROCEDURE Find_Flight_Details @Cus_Num VARCHAR(20) , @Flight_NO VARCHAR(20)
OUTPUT
AS
       BEGIN
                      SELECT @Flight_NO = FL.FlightNo
                      FROM Flight_leg FL , Seat S
                      WHERE FL.FlightNo= S.FlightNo AND
                               S.CusName = @Cus_Num
    END

DECLARE @F_NO VARCHAR(20)

EXEC Find_Flight_Details 'Mary Ann' , @F_NO OUTPUT

PRINT 'Flight NO : ' + @F_NO
```

Part 02

Database Vulnerabilities

i. SQL Injection (SQLi) Attack

A vulnerability known as SQL injection (SQLi) allows an attacker to change a database and acquire potentially important information by using a piece of SQL (structured query language) code. Given that it may be applied to any web application or website that makes use of a SQL-based database, it is one of the most common and dangerous types of attack (which is most of them). It has been explored since the late 1990s, but they are still relevant today. Fortnite (2019), Cisco (2018), and Tesla(2014) are some examples of SQLi.

SQL injections can be divided into three groups based on how they access the back-end data and the degree of potential harm they can do.

1) In-band SQLi Since attackers share the same communication route, they can easily execute.

    I. Error-based SQLi: The attacker's actions lead the database to output an error message. Based on the information produced by these error messages, the attacker compiles details about the database infrastructure.

    II. Union-based SQLi: The attacker combines several select queries into a single HTTP response and utilizes the UNION SQL operator to retrieve the needed data.

2) Inferential SQLi (Blind SQLi): The attacker uses the server's behavior patterns to learn more about the server's architecture. The attacker cannot see the result of an attack in-band.

    I. Time-based SQLi: Attackers send a SQL query to the database, making it wait a short while before returning a true or false answer.

    II. Boolean SQLi: Attackers send a SQL query to the database, allowing the application to return a result that is either true or false.

3) Out-of-band SQLi: When attackers cannot use the same channel to launch the attack and gather information, or when a server is too sluggish or unstable to do so, they may turn to SQLi out-of-band. DNS and HTTP rely on a server's computing capacity to convey data to an attacker. [1]

Tools associated with SQLI

1. SQLmap
2. SQLSus
3. Mole
4. Haviji
5. BSQL Hacker [2]

How works SQLi

An attacker must first locate weak user inputs within the web page or web application to launch a SQL Injection attack. Such user input is used directly in a SQL query on a web page or web application that contains a SQL Injection vulnerability. The assailant can input content. SQL Injections can be used by attackers to discover other users' login credentials in the database. You can choose which data to output from the database using SQL.

- Common Vulnerable Login Query

  SELECT * FROM users

  WHERE login = 'abc'

  AND password = '123'

  If it returns something, then log in.

- MS SQL Server Login Syntax

  VAR SQL = "SELECT * FROM users

  WHERE login = +formusr+

  AND password = +formpwd+.

  formusr = 'or 1=1

  formpws = anything

- Injecting through Strings

  SELECT * FROM users

  WHERE username = ' ' or 1=1

  AND password = 'anything'

✓ The argument for the string is closed.

✓ Subsequently, the SQL command is viewed as finished.

✓ In addition to string fields, there are also the date and numeric fields.

- ✓ The string contains data used to exploit systems. This series of characters make up SQL queries.
- ✓ A SQL query is sent from the program to the database.
- ✓ After a database query, including an attack, sends data back to the program
- ✓ The application sends data to the user.

Impacts of SQLi

- Disclose sensitive information.
- Data integrity is compromised.
- Compromise the privacy of users.
- Give an attacker system admin and general access.
- Loss of customer trust and reputational harm.

➢ Countermeasures and Mitigation

Any website or web application that makes use of a SQL database, such as MySQL, Oracle, SQL Server, or another one, may be vulnerable to a SQL Injection flaw. Injections are ranked as the top danger to web application security by the OWASP organization (Open Web Application Security Project) in their list of the top 10 threats for 2017. It is difficult to stop SQL Injection vulnerabilities. Input validation and parametrized queries with prepared statements are the only effective defenses against SQL Injection attacks. The application code shouldn't ever make direct use of the input. Not just web form inputs like login forms must be sanitized by the developer; all input must be done so. Single quotes and other potentially harmful code components must be removed. On your production website, it's a good idea to disable the display of database problems. SQL Injection can be used to learn more about your database by exploiting database faults. You might not be able to patch a SQL Injection vulnerability right away if you find it, say through an Acunetix scan. For instance, the flaw could be in open-source code. In these circumstances, you can use a web application firewall to sanitize your input momentarily.

1) Validate Input
2) Prepare a query
3) Create a prepared statement
4) Bind the parameters
5) Execute query
6) Fetch the result

ii. Denial Of Service Attack (DOS)

Attackers and online criminals utilize a sort of assault known as a denial-of-service (DoS) attack to take down a computer system or a network by overloading the target database with requests or traffic, rendering the system or network inaccessible to its intended users. DoS attacks can affect database management systems as well and refer to as network-based assaults. This malicious attempt aims to temporarily disable a networked system without permanently damaging it. The attacker employs specially crafted software to bombard the target machine with a torrent of data packets to tax the system's constrained resources. DoS attack costs victims' organizations the most money. [2]

1) Bandwidth Attacks

Each website's hosting is given a set amount of bandwidth; if more people visit the site than allowed, the hosting is compelled to restrict the site. A website will fall offline because of the attacker refreshing it frequently and opening a lot of pages. All the network's available bandwidth will be consumed by attacks.

2) Protocol Attacks

The system's resources, including the CPU and RAM, will be devoured when an assault is initiated. Protocols are required to move data across a network. The victim's system uses excessive amounts of resources because of these assaults due to a particular feature or implementation defect of a protocol installed there.

3) ICMP Flood Attacks

A "ping flood" is when an attacker repeatedly sends ICMP echo requests to a victim's computer. The attacker expects the victim to transmit an ICMP "echo reply" packet in response to each ICMP request. It is most effective when the attacker has more bandwidth than the target.

4) UDP Flood

Targets to overwhelm a targeted server with User Datagram Protocol (UDP) packets.

5) TCP SYN Flood Attacks

An attacker can take advantage of TCP-SYN flooding to cause a server to allocate extra RAM and wait for a pending connection as a result. The sender is unable to receive the last handshake message due to TCP- SYN flooding. Other systems are unable to communicate with the target system due to its clogged buffer.

Impacts of DOS

- Genuine users may not be able to find the information or take the necessary actions if they cannot access the resources.
- Businesses might be unable to complete time-sensitive tasks.
- They might have their reputation ruined.
- Customers can decide to utilize a rival.

➢ Countermeasures and Mitigation

Vulnerability is hard to understand. Removing network software, and preventive and investigative processes are required for high-profile databases. Making the right decisions reduces vulnerability to a DOS assault.

1) Build redundancy into the infrastructure

Split your servers across different data centers and employ a load-balancing system to distribute traffic among them.

2) Practice Basic Network Security

Use security measures like anti-phishing tools and firewalls that only allow a limited amount of outside traffic to reach your system.

3) Deploy anti-DDoS hardware and software modules

Network and web application firewalls are required for servers. The firewall or router can be set up to block some volumetric threats coming from the outside.

4) Check for security patches and keep updated.

Updates keep safe from attacks.

References

[1] "Imperva Learning Center," [Online]. Available: https://www.imperva.com/learn/.

[2] [Online]. Available: https://www.serverwatch.com/.