

# **Database Management Project**

## **2nd Semester 2020-21**



## **AIRLINES MANAGEMENT SMART SYSTEM**

**GitHub Link- <https://github.com/aditya7777/AIRLINES-MANAGEMENT-SMART-SYSTEM>**

**Aditya Saxena-2019A7PS0089U**

**Giri Prasad.S-2019A7PS0097U**

# ACKNOWLEDGMENT

We sincerely thank our Instructors,  
Ms. Sapna Sadhwani, and Dr.  
Tamizharasan Periyswamy for their  
invaluable guidance without which  
this project could not have been  
completed.



# TABLE OF CONTENT

- 1. Introduction**
- 2. ER Diagram**
- 3. Relational Schema**
- 4. Normalization**
- 5. SQL Tables**
- 6. SQL Queries**
- 7. Connect to Database**
- 6. GUI Images**

# INTRODUCTION

The Airline industry is booming with over 144 million customers in the year 2019. One of the most crucial elements which the Airline industry depends upon is the Database Management System for every Airline, which is used to store, organize and retrieve the details of all the Staff, Passenger bookings, Airline and Aircraft details, Arrivals, Departures, Finances, and many more segments.

This requires a robust and efficient design of the Database to manage the data so that it can be accessed, modified, and stored quickly, and in the least space. A DBMS may also be called a “Transaction Processing System”. A transaction is the basic atomic unit of a DBMS and hundreds of transactions take place every minute concurrently. A transaction may be a Search, Delete, Update or Insert operation into the database. The Database system and SQL commands described in this report makes up the basic operations of the airline such as reservation, cancellation, and updating of a flight trip data.

# FEASIBILITY AND RISK ANALYSIS

These are the following risks as per our analysis:

**Resource availability:** • Are the necessary skilled staff and computation available to develop this system?

**Development Risk:** • Can a system be developed so that it satisfies the pre-decided operational benchmarks and limits?

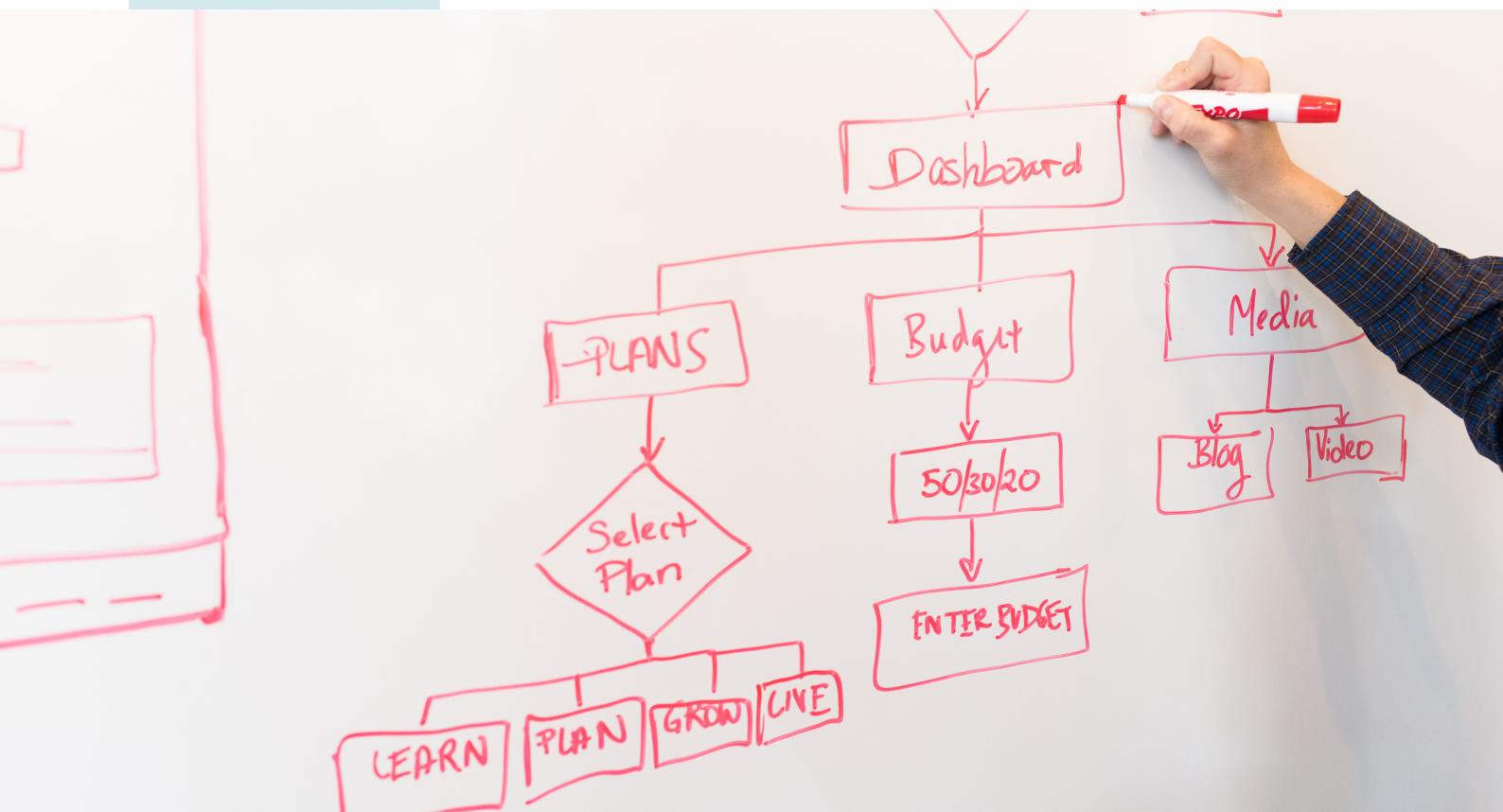
**Technology:** • Has the technology progressed to a state that will support the system?

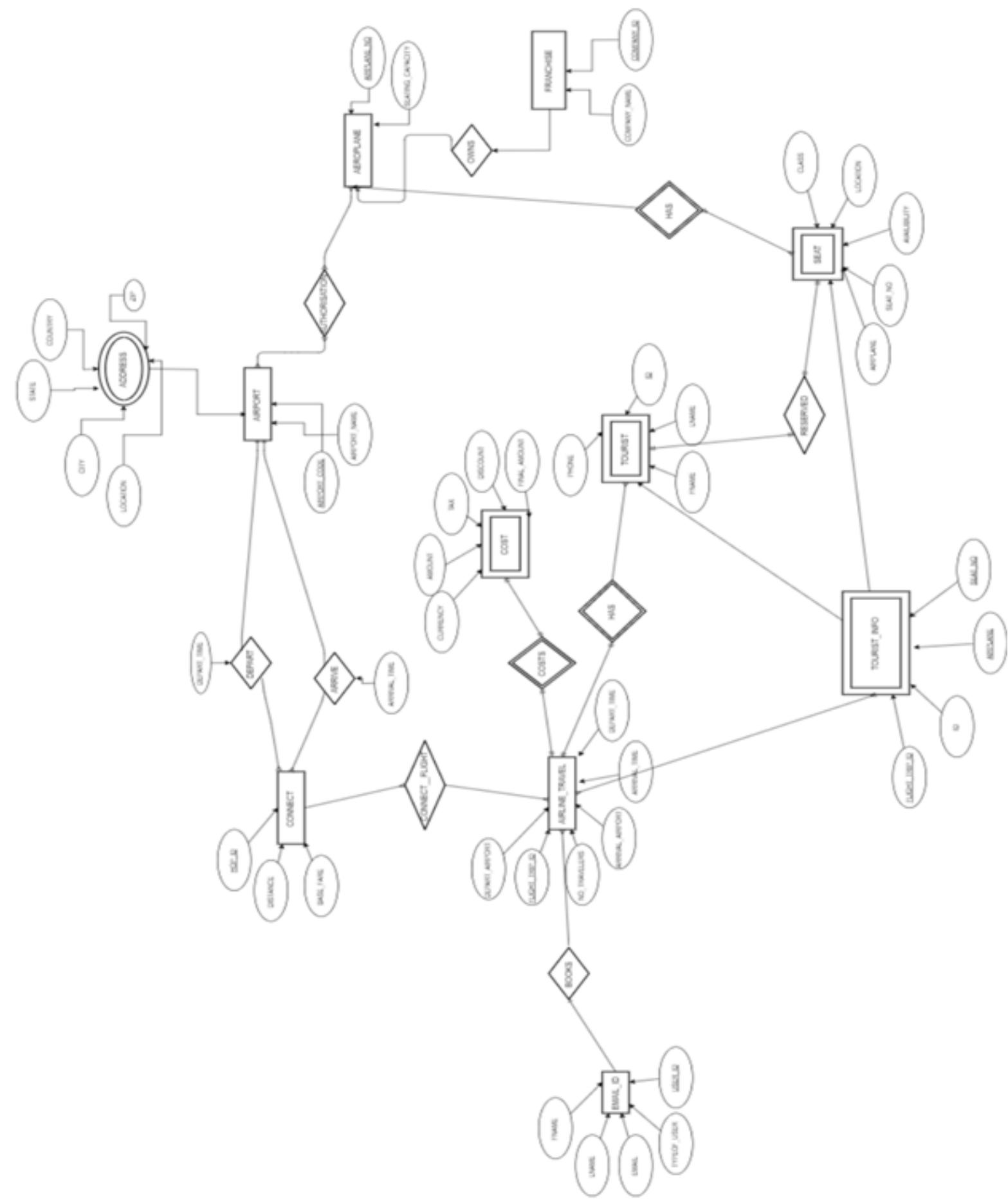
Our answer is that Yes, we currently are able to account for these risks in a proper manner. The resources that are needed to develop this system are present with us, we are the only staff who are working on this project, and we are satisfied that we will manage to execute this project within the given constraints with our current programming ability.

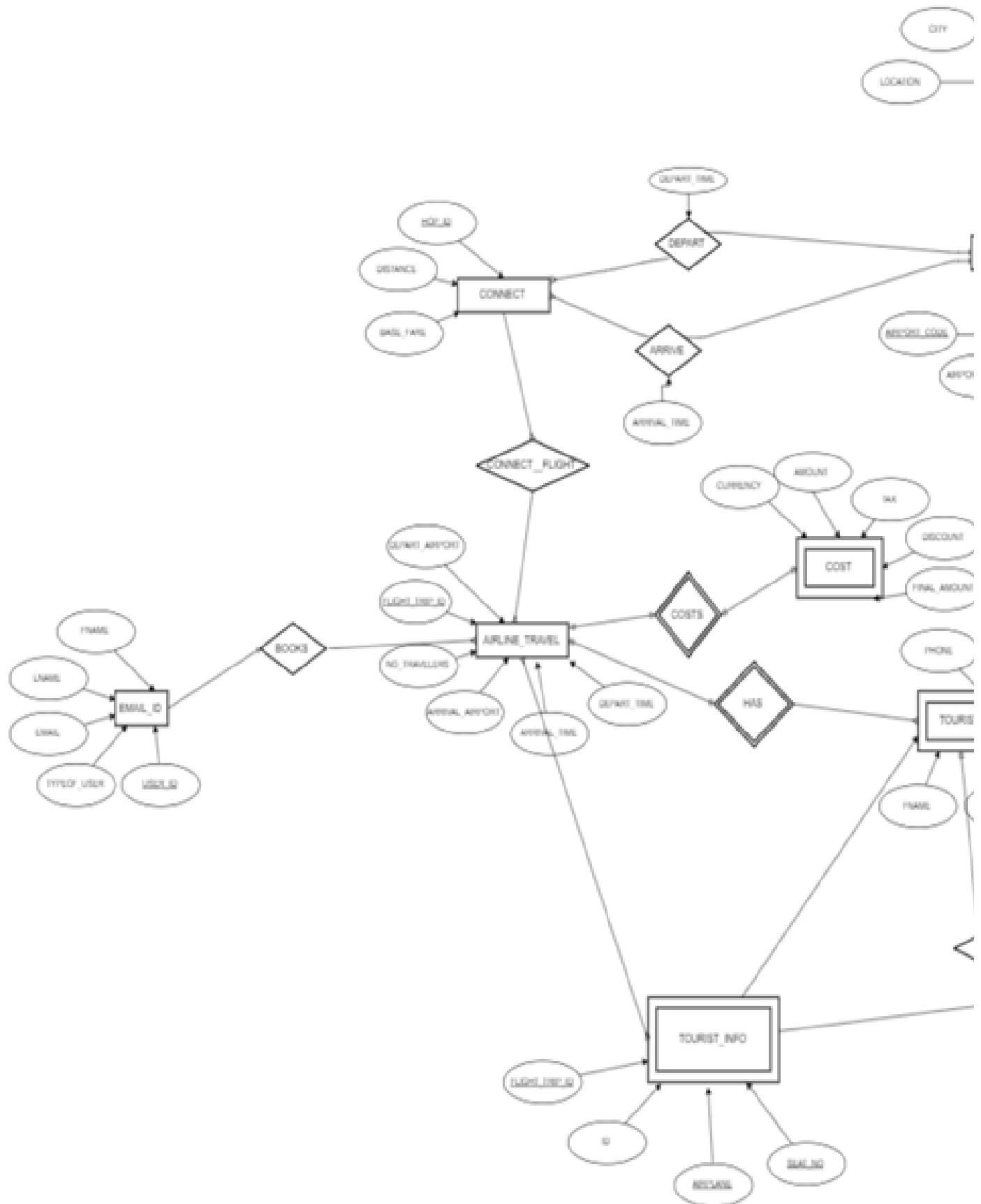
# ER DIAGRAM

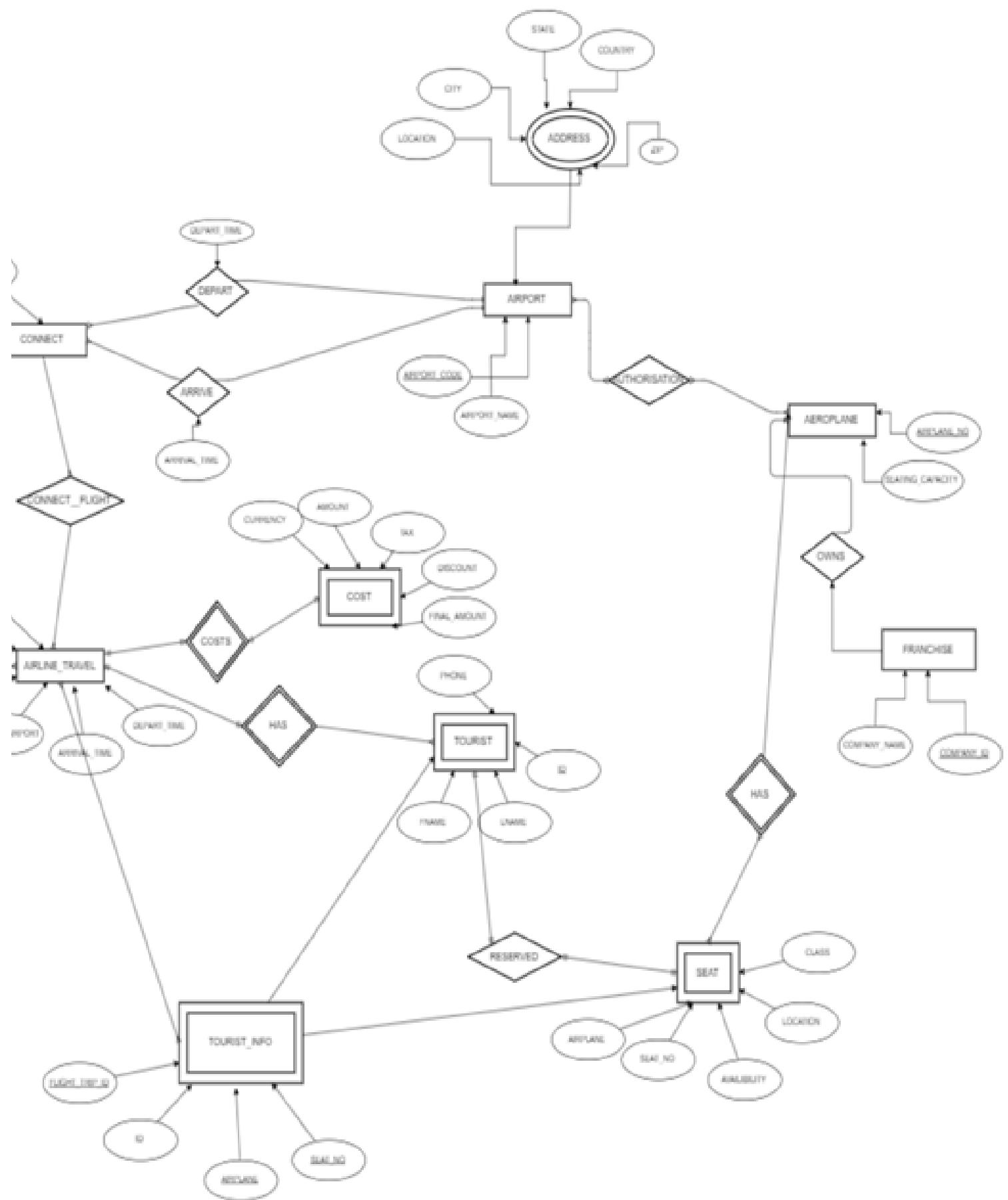
This section describes the logic behind the relations between all the different databases in our designed DBMS for Airline Reservation System.

We start with the Airport table. The Airport is an independent entity, identified uniquely by a code with an address as an important attribute. The address is a composite attribute as it contains different attributes under it, and similarly, we define other attributes for different entities in the report. In the Airline relation, and each plane of the airline has access to the Airport.









# Our assumptions are as follows:

Now, let us understand what a “hop ” is; a trip can be from A to C via B, i.e, A->B->C; where A->B and B->C are hops. Each hop has a unique ID, and it is not necessary that a single hop belongs to only one trip, since one passenger may be going to one place, while another may be going somewhere else, for example, Dubai connects all parts of the world together, so one flight coming into Dubai from one city has passengers traveling to all different places in the world.

5. So our assumption regarding hops is that a hop can belong to multiple trips and a trip may have multiple hops. Every hop has the attributes of Airport, arrival airport, and plane.

There exists a fare for every flight trip, and both cannot be defined without each other. We can define the total fare of a Traveller as the sum of all the fares for the hops minus the discount he/she gets. Also, there cannot be any traveler without any seat or trip.

6. So, our next assumption is that traveler and fare cannot exist without a trip, a trip has only one fare but may have multiple travelers. A User is a person who books the trip, and every user has a unique UserID. He/ She can be of three types: Agent, Traveller, or Airline Agent. We can assume that a user books tickets for multiple travelers and is an independent entity apart from the traveler.

# Our assumptions are as follows:

1. Each plane has access to multiple airports and each airport can give access to multiple planes. The airline company does not have the capability to access the airport because in case of an emergency if the company of a particular plane is not allowed to land at an airport then it will lead to disaster. In the Airline company, every company can own multiple planes and every plane must belong to a company. We assume that even private planes and Defence aircraft also belong to some company such as the Ministry of Defence, or some private plane business.
2. Every plane must belong to an airline company. A plane must belong to one company, but a company can have multiple planes. The basic element of the reservation is a seat in a plane.
3. We assume that a seat belongs to one plane but one plane can have many seats. Also, a seat can have only one passenger, but one passenger can have many seats. A seat is a dependent attribute because, without a plane, a seat cannot exist.
4. A traveler is a person who occupies the seat. Each traveler must have a seat, but it is not necessary that a seat must have a traveler, i.e, it may be empty.

# RELATIONAL SCHEMA

A relational database schema is the tables, columns, and relationships that make up a relational database.

Purpose: A relational database schema helps us to organize and understand the structure of a database. This is particularly useful when designing a new database, modifying an existing database to support more functionality or building integration between databases.

Creating the Schema: There are two steps to creating a relational database schema: creating the logical schema and creating the physical schema. The logical schema depicts the structure of the database, showing the tables, columns, and relationships with other tables in the database, and is a direct mapping of the Entity-Relationship diagram. The physical schema is created by actually generating the tables, columns, and relationships in the relational database management software (RDBMS) i.e SQL queries to create the database tables and relationships defined.

# RELATIONAL SCHEMA

AIRPORT

AIRPORT_CODE	AIRPORT_NAME	LOCATION	CITY	STATE	COUNTRY	ZIP
--------------	--------------	----------	------	-------	---------	-----

FRANCHISE

COMPANY_ID	COMPANY_NAME
------------	--------------

AEROPLANE

AIRPLANE_NO	SEATING_CAPACITY	COMPANY
-------------	------------------	---------

AUTHORIZATION

AIRPORT	AIRPLANE
---------	----------

SEAT

AIRPLANE	SEAT_NO	AVAILABILITY	LOCATION	CLASS
----------	---------	--------------	----------	-------

TOURIST\_INFO

FLIGHT	ID	AIRPLANE	SEAT_NO
--------	----	----------	---------

TOURIST

ID	FNAME	PHONE	LNAME
----	-------	-------	-------

AIRLINE\_TRAVEL

FLIGHT_TRIP_ID	NO_OF_TRAVELLERS	ARRIVAL_AIRPORT	ARRIVAL_TIME	DEPART_AIRPORT	DEPART_TIME	USER_EMAIL
----------------	------------------	-----------------	--------------	----------------	-------------	------------

COST

FLIGHT_TRIP_ID	FINAL_AMOUNT	AMOUNT	CURRENCY	DISCOUNT	TAX
----------------	--------------	--------	----------	----------	-----

EMAIL\_ID

EMAIL	FNAME	LNAME	PHONE	TYPE_OF_USER
-------	-------	-------	-------	--------------

CONNECT

HOP_ID	DISTANCE	BASE_FARE	AIRPLANE	ARRIVAL_AIRPORT	ARRIVAL_TIME	DEPART_AIRPORT	DEPART_TIME
--------	----------	-----------	----------	-----------------	--------------	----------------	-------------

CONNECT\_FLIGHT

HOP	FLIGHT_TRIP
-----	-------------

# NORMALIZATION (3NF)

AIRPORT

AIRPORT_CODE	AIRPORT_NAME	LOCATION	CITY	STATE	COUNTRY	ZIP
--------------	--------------	----------	------	-------	---------	-----

FRANCHISE

COMPANY_ID	COMPANY_NAME
------------	--------------

AEROPLANE

AIRPLANE_NO	SEATING_CAPACITY	COMPANY
-------------	------------------	---------

AUTHORIZATION

AIRPORT	AIRPLANE
---------	----------

SEAT

AIRPLANE	SEAT_NO	AVAILABILITY	LOCATION	CLASS
----------	---------	--------------	----------	-------

TOURIST\_INFO

FLIGHT	ID	AIRPLANE	SEAT_NO
--------	----	----------	---------

TOURIST

ID	FNAME	PHONE	LNAME
----	-------	-------	-------

AIRLINE\_TRAVEL

FLIGHT_TRIP_ID	NO_OF_TRAVELLERS	ARRIVAL_AIRPORT	ARRIVAL_TIME	DEPART_AIRPORT	DEPART_TIME	USER_EMAIL
----------------	------------------	-----------------	--------------	----------------	-------------	------------

COST

FLIGHT_TRIP_ID	FINAL_AMOUNT	AMOUNT	CURRENCY	DISCOUNT	TAX
----------------	--------------	--------	----------	----------	-----

EMAIL\_ID

EMAIL	FNAME	LNAME	PHONE	TYPE_OF_USER
-------	-------	-------	-------	--------------

CONNECT

HOP_ID	DISTANCE	BASE_FARE	AIRPLANE	ARRIVAL_AIRPORT	ARRIVAL_TIME	DEPART_AIRPORT	DEPART_TIME
--------	----------	-----------	----------	-----------------	--------------	----------------	-------------

CONNECT\_FLIGHT

HOP	FLIGHT_TRIP
-----	-------------

# MySQL Tables

- Aeroplane Table

```
CREATE TABLE AEROPLANE(  
AIRPLANE_NUMBER    VARCHAR(25) ,  
SEATING_CAPACITY NUMBER    NOT NULL,  
COMPANY_NAME VARCHAR(50) NOT NULL,  
CONSTRAINT AIRPLANE_PK PRIMARY KEY (AIRPLANE_NO),  
CONSTRAINT PLANE_COMPANY_FK FOREIGN KEY (COMPANY)  
REFERENCES AIRLINE_COMPANY(COMPANY_ID)  
);
```

```
INSERT ALL  
INTO AEROPLANE VALUES( 'AB751' , 140, 'AB' )  
INTO AEROPLANE VALUES( 'CD851' , 260, 'CD' )  
INTO AEROPLANE VALUES( 'EF100' , 170, 'EF' )  
INTO AEROPLANE VALUES( 'GH785' , 200, 'GH' )  
INTO AEROPLANE VALUES( 'IJ747' , 180, 'IJ' )  
INTO AEROPLANE VALUES( 'KL747' , 210, 'KL' )  
INTO AEROPLANE VALUES( 'MN747' , 190, 'MN' )
```

# • Airline Travel Table

```
CREATE TABLE AIRLINE_TRAVEL(
FLIGHT_TRIP_ID VARCHAR(15) ,
NO_OF_TRAVELLERS NUMBER NOT NULL,
ARRIVAL_AIRPORT VARCHAR(25) NOT NULL,
ARRIVAL_TIME DATE NOT NULL,
DEPART_AIRPORT VARCHAR(25) NOT NULL,
DEPART_TIME DATE NOT NULL,
USER_EMAIL VARCHAR(25) NOT NULL,
CONSTRAINT FLIGHT_TRIP_PK PRIMARY KEY
(FLIGHT_TRIP_ID),
CONSTRAINT FLGHT_TRIP_USER_FK FOREIGN KEY (USER_EMAIL)
REFERENCES END_USER(EMAIL) ON DELETE SET NULL
);
```

```
INSERT ALL
INTO AIRLINE_TRAVEL
VALUES('giri23apr',1,'NVM',to_date('2021/12/27:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'MUM',to_date('2021/12/26:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'abc@gmail.com' )
```

```
INTO AIRLINE_TRAVEL
VALUES('aditya11nvm',1,'CHG',to_date('2021/12/27:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'BNG',to_date('2021/12/26:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'def@gmail.com' )
```

```
INTO AIRLINE_TRAVEL
VALUES('yash31dec',1,'BNG',to_date('2021/11/27:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam') , 'KLK',to_date('2021/11/26:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'ghi@gmail.com' )
```

INTO AIRLINE\_TRAVEL

VALUES('yash31dec',1,'BNG',to\_date('2021/11/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'KLK',to\_date('2021/11/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'ghi@gmail.com' )

INTO AIRLINE\_TRAVEL

VALUES('shoaga37dec',1,'BNG',to\_date('2021/11/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'CHG',to\_date('2021/11/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'jkl@gmail.com' )

INTO AIRLINE\_TRAVEL

VALUES('bala17jan',1,'MUM',to\_date('2021/11/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'PTN',to\_date('2021/11/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'mno@gmail.com' )

INTO AIRLINE\_TRAVEL

VALUES('akshit12may',1,'MUM',to\_date('2021/11/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'NVM',to\_date('2021/11/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'pqr@gmail.com' )

INTO AIRLINE\_TRAVEL

VALUES('parth09mrh',1,'BNG',to\_date('2021/11/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'CHG',to\_date('2021/11/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'stu@gmail.com' )

INTO AIRLINE\_TRAVEL

VALUES('yash31dec',1,'KLK',to\_date('2021/11/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'CHN',to\_date('2021/11/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'vwx@gmail.com' )

INTO AIRLINE\_TRAVEL

VALUES('avi20nov',1,'CHG',to\_date('2021/11/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'BNG',to\_date('2021/11/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'yza@gmail.com' )

INTO AIRLINE\_TRAVEL

VALUES('sham21feb',1,'PTN',to\_date('2021/11/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'MUM',to\_date('2021/11/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'bcd@gmail.com' )

# ● Airport Table

```
CREATE TABLE AIRPORT(
```

```
    AIRPORT_ID VARCHAR(25),
```

```
    AIRPORT_NAME VARCHAR(100) NOT NULL,
```

```
    AIRPORT_LOCATION VARCHAR(100) NOT NULL,
```

```
    AIRPORT_CITY VARCHAR(100) NOT NULL,
```

```
    AIRPORT_STATE VARCHAR(100) NOT NULL,
```

```
    AIRPORT_COUNTRY VARCHAR(100) NOT NULL,
```

```
);
```

```
INSERT ALL
```

```
INTO AIRPORT VALUES( 'MUM' , 'Chatrapati Shivaji International  
Airport', '30#N','Mumbai','Maharashtra','India')
```

```
INTO AIRPORT VALUES( 'BNG' , 'Benguluru International Airport',  
'87#S','Banglore','Andhra Pradesh','India')
```

```
INTO AIRPORT VALUES( 'CHN' , 'Chennai International Airport',  
'90#W','Chennai','Tamil Nadu','India')
```

```
INTO AIRPORT VALUES( 'KLK' , 'Kolkata International Airport',  
'11#E','Kolkata','West Bengal','India')
```

```
INTO AIRPORT VALUES( 'CHG' , 'Chandigarh International Airport',  
'55#N','Chandigarh','Punjab','India')
```

```
INTO AIRPORT VALUES( 'NVM' , 'Navi Mumbai International Airport',  
'04#E','Navi Mumbai','Maharashtra','India')
```

```
INTO AIRPORT VALUES( 'PTN' , 'Patna International Airport',  
'77#W','Patna','Bihar','India')
```

## • Authorization Table

```
CREATE TABLE AUTHORISATION(
    AIRPORT VARCHAR(25) NOT NULL,
    AIRPLANE VARCHAR(25) NOT NULL,
    CONSTRAINT ACCESS_PK PRIMARY KEY (AIRPORT, AIRPLANE),
    CONSTRAINT ACCESS_PORT_FK FOREIGN KEY (AIRPORT)
        REFERENCES AIRPORT(AIRPORT_CODE) ON DELETE CASCADE,
    CONSTRAINT ACCESS_PLANE_FK FOREIGN KEY (AIRPLANE)
        REFERENCES AIRPLANE(AIRPLANE_NO) ON DELETE CASCADE
);
```

```
INSERT ALL
INTO AUTHORISATION VALUES( 'MUM','AB751' )
INTO AUTHORISATION VALUES( 'CHN','MN747' )
INTO AUTHORISATION VALUES( 'BNG','IJ747' )
INTO AUTHORISATION VALUES( 'NVM','GH785' )
INTO AUTHORISATION VALUES( 'PTN','EF100' )
INTO AUTHORISATION VALUES( 'MUM','MN747' )
INTO AUTHORISATION VALUES( 'PTN','AB751' )
INTO AUTHORISATION VALUES( 'KLK','IJ747' )
INTO AUTHORISATION VALUES( 'CHG','KL747' )
```

- Connect Flight Table

```
CREATE TABLE CONNECT_FLIGHT(  
HOP VARCHAR(25) NOT NULL,  
FLIGHT_TRIP VARCHAR(15) NOT NULL,  
CONSTRAINT HP_HAS_FT_PK PRIMARY KEY  
(HOP,FLIGHT_TRIP),  
CONSTRAINT HOP_HAS_FK FOREIGN KEY (HOP) REFERENCES  
HOP(HOP_ID),  
CONSTRAINT FT_HAS_FK FOREIGN KEY(FLIGHT_TRIP)  
REFERENCES FLIGHT_TRIP(FLIGHT_TRIP_ID)  
ON DELETE CASCADE  
);
```

```
INSERT ALL  
INTO CONNECT_FLIGHT VALUES( 'AX7458','giri23apr' )  
INTO CONNECT_FLIGHT VALUES( 'AX7459','aditya11nvm' )  
INTO CONNECT_FLIGHT VALUES( 'AX7458','yash31dec' )  
INTO CONNECT_FLIGHT VALUES( 'AX7458','shoaga17dec' )  
INTO CONNECT_FLIGHT VALUES( 'AX7460','bala17dec' )  
INTO CONNECT_FLIGHT VALUES( 'AX7458','akshit12may' )  
INTO CONNECT_FLIGHT VALUES( 'AX7459','aditya11nvm' )  
INTO CONNECT_FLIGHT VALUES( 'AX7458','sham21feb' )  
INTO CONNECT_FLIGHT VALUES( 'AX7458','avi20nov' )  
INTO CONNECT_FLIGHT VALUES( 'AX7460','parth09mrh' )
```

- Connect Table

```
CREATE TABLE CONNECT(
    HOP_ID VARCHAR(15) ,
    DISTANCE DECIMAL(7,2) NOT NULL,
    BASE_FARE DECIMAL(9,2) NOT NULL,
    AIRPLANE VARCHAR(25) NOT NULL,
    ARRIVAL_AIRPORT VARCHAR(25) NOT NULL,
    ARRIVAL_TIME DATE NOT NULL,
    DEPART_AIRPORT VARCHAR(25) NOT NULL,
    DEPART_TIME DATE NOT NULL,
    CONSTRAINT HOP_PK PRIMARY KEY (HOP_ID),
    CONSTRAINT HOP_APLNE_FK FOREIGN KEY (AIRPLANE)
        REFERENCES AIRPLANE(AIRPLANE_NO),
    CONSTRAINT HOP_ARVL_ARPRT_FK
        FOREIGN KEY(ARRIVAL_AIRPORT)REFERENCES
        AIRPORT(AIRPORT_CODE),
    CONSTRAINT HOP_DRPT_ARPRT_FK FOREIGN
        KEY(DEPART_AIRPORT) REFERENCES AIRPORT(AIRPORT_CODE)
);
```

```
INSERT ALL
INTO CONNECT VALUES
('re428',600,200,'AB751','MUM',to_date('2021/12/27:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'PTN',to_date('2021/12/26:12:00:00AM'
,'yyyy/mm/dd:hh:mi:ssam'))
```

INTO CONNECT VALUES  
( 'bt459',600,200,'CD851','BNG',to\_date('2021/12/27:12:00:00AM','yyyy/mm/dd:hh:mi:ssam'),'CHG',to\_date('2021/12/26:12:00:00AM','yy/mm/dd:hh:mi:ssam') )

INTO CONNECT VALUES  
( 'vf650',600,200,'EF100','CHG',to\_date('2021/12/27:12:00:00AM','yyyy/mm/dd:hh:mi:ssam'),'MUM',to\_date('2021/12/26:12:00:00AM','yyy/mm/dd:hh:mi:ssam') )

INTO CONNECT VALUES  
( 'xs221',600,200,'GH785','NVM',to\_date('2021/12/27:12:00:00AM','yyyy/mm/dd:hh:mi:ssam'),'CHN',to\_date('2021/12/26:12:00:00AM','yy/mm/dd:hh:mi:ssam') )

INTO CONNECT VALUES  
( 'nj864',600,200,'IJ747','KLK',to\_date('2021/12/27:12:00:00AM','yyyy/mm/dd:hh:mi:ssam'),'MUM',to\_date('2021/12/26:12:00:00AM','yyy/mm/dd:hh:mi:ssam') )

INTO CONNECT VALUES  
( 'mj645',600,200,'AB751','MUM',to\_date('2021/12/27:12:00:00AM','yyy/mm/dd:hh:mi:ssam'),'KLK',to\_date('2021/12/26:12:00:00AM','yyyy/mm/dd:hh:mi:ssam') )

INTO CONNECT VALUES  
( 'bg423',600,200,'CD851','BNG',to\_date('2021/12/27:12:00:00AM','yyyy/mm/dd:hh:mi:ssam'),'PTN',to\_date('2021/12/26:12:00:00AM','yyy/mm/dd:hh:mi:ssam') )

INTO CONNECT VALUES  
( 'mk653',600,200,'EF100','BNG',to\_date('2021/12/27:12:00:00AM','yyyy/mm/dd:hh:mi:ssam'),'MUM',to\_date('2021/12/26:12:00:00AM','yyyy/mm/dd:hh:mi:ssam') )

INTO CONNECT VALUES

( 'pl888',600,200,'GH785','NVM',to\_date('2021/12/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'KLK',to\_date('2021/12/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam') )

INTO CONNECT VALUES

( 'qw322',600,200,'IJ747','MUM',to\_date('2021/12/27:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam'),'NVM',to\_date('2021/12/26:12:00:00AM',  
'yyyy/mm/dd:hh:mi:ssam') )

## • Cost Table

```
CREATE TABLE COST(  
FLIGHT_TRIP_ID VARCHAR(15) ,  
FINAL_AMOUNT DECIMAL(9,2) NOT NULL,  
AMOUNT DECIMAL(9,2) NOT NULL,  
CURRENCY VARCHAR(5) NOT NULL,  
DISCOUNT DECIMAL(4,2) NOT NULL,  
TAX DECIMAL(5,2) NOT NULL,  
CONSTRAINT FARE_PK PRIMARY KEY (FLIGHT_TRIP_ID),  
CONSTRAINT FARE_FLT_TRIP_FK FOREIGN KEY (FLIGHT_TRIP_ID)  
REFERENCES FLIGHT_TRIP(FLIGHT_TRIP_ID)  
ON DELETE CASCADE  
);
```

```
INSERT ALL  
INTO COST VALUES( 'giri23apr',176,200,'AED',20,10 )  
INTO COST VALUES( 'aditya11nvm',176,200,'AED',20,10 )  
INTO COST VALUES( 'yash31dec',176,200,'AED',20,10 )  
INTO COST VALUES( 'shoaga17dec',176,200,'AED',20,10 )  
INTO COST VALUES( 'bala17jan',176,200,'AED',20,10 )  
INTO COST VALUES( 'akshit12may',176,200,'AED',20,10 )  
INTO COST VALUES( 'sham21feb',176,200,'AED',20,10 )  
INTO COST VALUES( 'yash31dec',176,200,'AED',20,10 )  
INTO COST VALUES( 'avi20nov',176,200,'AED',20,10 )  
INTO COST VALUES( 'parth09mrh',176,200,'AED',20,10 )
```

## ● Email ID Table

```
CREATE TABLE EMAIL_ID (
EMAIL VARCHAR(25),
FNAME VARCHAR(25) NOT NULL,
LNAME VARCHAR(25) NOT NULL,
PHONE NUMBER,
TYPE_OF_USER VARCHAR(25) NOT NULL,
CONSTRAINT USER_PK PRIMARY KEY (EMAIL)
);
INSERT ALL
INTO EMAIL_ID VALUES
('abc@gmail.com','Giri','Prasad',32453267988,'AGENT' )
INTO EMAIL_ID VALUES
('def@gmail.com','Aditya','Saxena',9867544332,'NORMAL' )
INTO EMAIL_ID VALUES
('ghi@gmail.com','Yash','Agrawal',5463433254,'NORMAL' )
INTO EMAIL_ID VALUES ('jkl@gmail.com'
,'Indervir','Singh',676576890967,'NORMAL' )
INTO EMAIL_ID VALUES ('mno@gmail.com'
,'Bala','Yadav',456576212134,'AGENT' )
INTO EMAIL_ID VALUES
('pqr@gmail.com','Giri','Prasad',32453267988,'AGENT' )
INTO EMAIL_ID VALUES
('stu@gmail.com','Aditya','Saxena',9867544332,'NORMAL' )
INTO EMAIL_ID VALUES
('vwx@gmail.com','Yash','Agrawal',5463433254,'NORMAL' )
INTO EMAIL_ID VALUES ('yza@gmail.com'
,'Indervir','Singh',676576890967,'NORMAL' )
INTO EMAIL_ID VALUES ('bcd@gmail.com'
,'Bala','Yadav',456576212134,'AGENT' )
```

## ● Franchise Table

```
CREATE TABLE FRANCHISE(  
COMPANY_ID VARCHAR(100) ,  
COMPANY_NAME VARCHAR(100) NOT NULL,  
COMPANY_CITY VARCHAR(100) NOT NULL,  
CONSTRAINT COMPANY_PKEY PRIMARY KEY (COMPANY_ID)  
);
```

```
INSERT ALL  
INTO FRANCHISE VALUES( 'AB' , 'Spice Jet','Mumbai')  
INTO FRANCHISE VALUES( 'CD' , 'Indigo Airlines','Banglore')  
INTO FRANCHISE VALUES( 'EF' , 'AirIndia Airlines','Kolkata')  
INTO FRANCHISE VALUES( 'GH' , 'Kingfisher Airlines','Patna')  
INTO FRANCHISE VALUES( 'IJ' , 'Vistara Airlines','Mumbai')  
INTO FRANCHISE VALUES( 'KL' , 'Thar Airlines','Delhi')  
INTO FRANCHISE VALUES( 'MN' , 'AirAsia Airlines','Chandigarh')  
INTO FRANCHISE VALUES( 'OP' , 'Jet Airways','Chennai')
```

## • Seat Table

```
CREATE TABLE SEAT(  
AIRPLANE VARCHAR(25) NOT NULL,  
SEAT_NO VARCHAR(5) NOT NULL,  
AVAILABILITY VARCHAR(10) DEFAULT 'TRUE',  
LOCATION VARCHAR(25) NOT NULL,  
SEAT_CLASS VARCHAR(25) NOT NULL,  
CONSTRAINT SEAT_PK PRIMARY KEY (AIRPLANE,SEAT_NO),  
CONSTRAINT SEAT_PLANE_FK FOREIGN KEY (AIRPLANE) REFERENCES  
AIRPLANE(AIRPLANE_NO) ON DELETE CASCADE  
);
```

```
INSERT ALL  
INTO SEAT VALUES( 'EF100','A23','TRUE','WINDOW','ECONOMIC')  
INTO SEAT VALUES( 'AB751' , 'A24','TRUE','AILE','FIRST CLASS')  
INTO SEAT VALUES( 'KL747' , 'A25','TRUE','WINDOW','ECONOMIC')  
INTO SEAT VALUES( 'MN747' , 'A26','TRUE','AILE','ECONOMIC')  
INTO SEAT VALUES( 'IJ747' , 'A28','TRUE','AILE','FIRST CLASS')  
INTO SEAT VALUES( 'GH785','A29','TRUE','WINDOW','FIRST CLASS')  
INTO SEAT VALUES( 'CD851' , 'A34','TRUE','WINDOW','ECONOMIC')  
INTO SEAT VALUES( 'IJ747' , 'A55','TRUE','AILE','ECONOMIC')  
INTO SEAT VALUES( 'MN747' , 'A33','TRUE','WINDOW','ECONOMIC')  
INTO SEAT VALUES( 'GH785' , 'A44','TRUE','WINDOW','FIRST CLASS')
```

## • Tourist Table

```
CREATE TABLE TOURIST(  
ID NUMBER,  
PHONE INTEGER,  
FNAME VARCHAR(25) NOT NULL,  
LNAME VARCHAR(25) NOT NULL,  
CONSTRAINT TRAVELLER_PK PRIMARY KEY (ID)  
);
```

```
INSERT INTO TOURIST  
VALUES(traveller_id_increment.nextVal,4567891231,'Giri','Prasad' );  
INSERT INTO TOURIST  
VALUES( traveller_id_increment.nextval,4567891241,'Aditya','Saxena' );  
INSERT INTO TOURIST  
VALUES( traveller_id_increment.nextVal,4567891274,'Bala','Yadav' );  
INSERT INTO TOURIST  
VALUES(traveller_id_increment.nextVal,4567891278,'Shobhi','Agrawal);  
INSERT INTO TOURIST  
VALUES( traveller_id_increment.nextVal,4567891214,'Nitish','Salwan' );  
INSERT INTO TOURIST  
VALUES(traveller_id_increment.nextVal,4567891231,'Giri','Prasad' );  
INSERT INTO TOURIST  
VALUES( traveller_id_increment.nextval,4567891241,'Aditya','Saxena' );  
INSERT INTO TOURIST  
VALUES( traveller_id_increment.nextVal,4567891274,'Bala','Yadav' );  
INSERT INTO TOURIST  
VALUES(traveller_id_increment.nextVal,4567891278,'Shobhi','Agrawal);  
INSERT INTO TOURIST  
VALUES( traveller_id_increment.nextVal,4567891214,'Nitish','Salwan' );
```

## • Tourist Information Table

```
CREATE TABLE TOURIST_INFO (
    FLIGHT VARCHAR(15) NOT NULL,
    ID VARCHAR(15),
    SEAT_NO VARCHAR(5),
    AIRPLANE VARCHAR(25),
    CONSTRAINT TRAVELLER_PKK PRIMARY KEY
    (FLIGHT,SEAT_NO,AIRPLANE),
    CONSTRAINT TRVLR_FLT_TRIP_FK FOREIGN KEY (FLIGHT)
    REFERENCES FLIGHT_TRIP(FLIGHT_TRIP_ID)
    ON DELETE CASCADE,
    CONSTRAINT TRVLR_SEAT_FK FOREIGN
    KEY(SEAT_NO, AIRPLANE)
    REFERENCES SEAT(SEAT_NO, AIRPLANE)
);
```

INSERT ALL

```
INTO TOURIST_INFO VALUES ('giri23apr',2,'A23','AB751' )
INTO TOURIST_INFO VALUES ('kuniaa741',6,'A24','CD851' )
INTO TOURIST_INFO VALUES( 'inder17dec',3,'A25','EF100' )
INTO TOURIST_INFO VALUES( 'shoaga17dec',4,'A26','GH785' )
INTO TOURIST_INFO VALUES( 'aditya11nvm',5,'A27','IJ747' )
INTO TOURIST_INFO VALUES( 'bala17jan',2,'A33','KL747' )
INTO TOURIST_INFO VALUES( 'kuniaa741',6,'A34','MN747' )
INTO TOURIST_INFO VALUES( 'inder17dec',3,'A35','GH785' )
INTO TOURIST_INFO VALUES( 'avi20nov',4,'A43','CD851' )
INTO TOURIST_INFO VALUES( 'sham21feb',5,'A44','MN747' )
```

# Queries

```
SELECT FNAME, LNAME  
FROM TOURIST  
WHERE ID IN (SELECT ID  
FROM TOURIST_INFO  
WHERE FLIGHT='giri12dec');
```

```
//GROUP BY & aggregate function  
SELECT USER_EMAIL, COUNT (NO_OF_TRAVELLERS)  
FROM AIRLINE_TRAVELLER  
WHERE USER_EMAIL IN (SELECT EMAIL  
FROM EMAIL_ID  
WHERE TYPE_OF_USER ='NORMAL')  
GROUP BY USER_EMAIL;
```

```
//Join  
SELECT *  
FROM AIRLINE_TRAVELLER  
INNER JOIN FARE ON  
AIRLINE_TRAVELLER.FLIGHT_TRIP_ID = COST.FLIGHT_TRIP_ID;
```

```
//Alter  
ALTER TABLE COST ADD (CHECKING VARCHAR(25), CONSTRAINT FK  
FOREIGN KEY(CHECKING) REFERENCES AIRPORT(AIRPORT_CODE));
```

```
//Dropping constraints and columns  
ALTER TABLE COST DROP CONSTRAINT FK;  
ALTER TABLE COST DROP COLUMN CHECKING;
```

```
//Union
SELECT FNAME, LNAME
FROM TOURIST
WHERE ID IN(SELECT ID
FROM TOURIST_INFO
WHERE FLIGHT='giri17dec')
UNION
SELECT FNAME, LNAME
FROM TOURIST
WHERE ID IN(SELECT ID
FROM TOURIST_INFO
WHERE FLIGHT='sham22jan');
```

```
//DROP
DROP table COST;
```

```
//Trigger
delimiter \\
create trigger update_cost
after update on COST
for each row
begin
    if 'FINAL_AMOUNT' >= 175 then
        update FINAL_AMOUNT
        set FINAL_AMOUNT=FINAL_AMOUNT*DISCOUNT
        where TAX='10' AND CURRENCY='AED';
    end if;
end;
//
```

```
update IGNORE capacity,Flight set `economy seats`= 3 where
capacity.`aircraft_Reg. No.`=140 AND Flight.date="2021/05/18";
Select * from capacity where `aircraft_Reg. No.`=101
select * from fares where Flight_Number=402 AND Flight_date="2020-
05-18" and class_name="economy";
drop procedure update_seats
delimiter \\
```

```
create procedure update_seats(IN p1 varchar(30),IN p2 INT,in p3
DATE)
```

```
begin
```

```
    update IGNORE capacity INNER JOIN flight_has_class ON
capacity.`aircraft_Reg. No.`=flight_has_class.`Flight_aircraft_Reg. No.`
```

```
        set `economy seats`=`economy seats`-1
```

```
        where flight_has_class.class_name=p1 AND
Flight_has_class.Flight_Number=p2 AND flight_has_class.Flight_date
=p3;
```

```
end;
```

```
//
```

```
call update_seats('economy',402,"2020-05-18");
```

```
select * from capacity where `aircraft_Reg. No.`=101
```

```
drop trigger delete_flight
```

```
delimiter \\
```

```
create trigger delete_flight
after update on capacity
for each row
begin
    if 'economy seats'= 0 AND 'business seats'=0 AND 'premium
economy seats'=0 AND 'first class seats'=0 then
        delete from flight where number=2471 AND date = "2020-05-18";
    end if;
end;
//
update IGNORE capacity set `economy seats`= 0,`business seats`=0 ,
`premium economy seats`=0 , `first class seats`=0 where
`aircraft_Reg. No.`=301
select * from capacity where `aircraft_Reg. No.`=301
select * from flight where `aircraft_Reg. No.`=301 and date="2020-05-
18";
delimiter \\
create PROCEDURE show_higher_class(IN p1 VARCHAR(30),OUT p2
VARCHAR(30))
begin
    if p1='economy' then
        set p2='business class';
    elseif p1='business class' then
        set p2='first class';
    else
        set p2= 'first class';
    end if;
END
\\
Call show_higher_class ( 'economy',@p2);
select * from class where name=@p2;
```

# Connect to Database

```
import java.sql.CallableStatement;
import java.sql.Connection;

import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Date;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

//import com.mysql.jdbc.Statement;
import com.toedter.calendar.JDateChooser;

import net.sourceforge.jdatepicker.impl.JDatePanelImpl;
import net.sourceforge.jdatepicker.impl.JDatePickerImpl;
import net.sourceforge.jdatepicker.impl.UtilDateModel;
```

```
import java.io.File;
import java.io.IOException;
public class ConnectDB extends JPanel implements ActionListener {
//DB Connection variables
    static JLabel label0=new JLabel("");
    static Connection connection= null;
    static String databaseName=""; //leave this as it is
    static String url= "jdbc:mysql://localhost:3306/" +
databaseName; //leave this as it is
    static String username="root"; //leave this as it is if you haven't
changed your user name from root to something else
    static String password="ansh2814"; //password for your
connection, e.g 'a' or whatever it is
    static JComboBox dd1;
    static JComboBox dd2;
    static JComboBox dd3;
    static JTextField tf1;
    static JTextField tf2;
    static JFrame frame1=new JFrame();
    static JButton button=new JButton();
    static JFrame frame = new JFrame("K.A.R.D.S. Flights");
    static String[] columnNames = {"Name", "Id", "Grade"};
    static JTable table;
    static JTable table2;
    static JFrame frame2;
    static JFrame frame3;
    static JLabel jl=new JLabel("Enter the option number to check
details :");
    static JTextField jb=new JTextField("",10);
    static JLabel jl2=new JLabel("Enter the option number to check
details :");
    static JTextField jb2=new JTextField("Enter 0 if one-way trip",10);
    static JButton okbtn;
    static JDatePickerImpl datePicker1;
```

```
static JDatePickerImpl datePicker2;
    static JTextField start;
    static JTextField end;
    static JButton cal=new JButton();
    static JButton confi=new JButton("Book");

public ConnectDB()
{
    Image img=new
ImageIcon(this.getClass().getResource("/GUI_DBS.jpg")).getImage();
    Image x=img.getScaledInstance(label0.getWidth(),
label0.getHeight(), Image.SCALE_SMOOTH);
label0.setIcon(new ImageIcon(x));

    Image img2=new ImageIcon(this.getClass().getResource("/search-
icon.png")).getImage();
button.setIcon(new ImageIcon(img2));
button.setFont(new Font("Tamoha",Font.PLAIN,12));
button.addActionListener(this);

    Image img3=new ImageIcon(this.getClass().getResource("/Calendar-
icon.png")).getImage();
cal.setIcon(new ImageIcon(img3));

}

@SuppressWarnings("deprecation")
    public static void main(String[] args) throws
InstantiationException, IllegalAccessException,
ClassNotFoundException, SQLException, IOException ,
NullPointerException {
```

```
JLabel label81=new JLabel("Total Passengers:");
label81.setFont(new Font("Tamaha",Font.PLAIN,15));

JPanel panel0 = new JPanel(new CardLayout());
JPanel panel1 = new JPanel();
panel1.setLayout(new CardLayout());
JPanel panel2 = new JPanel(new FlowLayout());
JPanel panel3 = new JPanel(new FlowLayout());
JPanel panel4 = new JPanel(new FlowLayout());
JPanel panel5 = new JPanel(new FlowLayout());
JLabel label1 = new JLabel("Flights");

String dds1[]={ "One way", "Round Trip"};
dd1=new JComboBox(dds1); //drop down for type of trip, don't include multi-trip
String dds2[]={ "Adult", "Children"};
dd2=new JComboBox(dds2); //drop down for type of passenger
String dds3[]={ "Economy", "Premium Economy", "Business", "First Class"};
dd3=new JComboBox(dds3); //drop down for type of ticket he/she books
tf1=new JTextField("Adults(12+)",7);
tf1.setBounds(50,50, 10,15);
tf2=new JTextField("Children(0-11)",8);
tf2.setBounds(50,50, 10,15);
JLabel label2=new JLabel("Options");
Dimension size = label1.getPreferredSize();
start=new JTextField("Delhi",15); //default place of departure
end=new JTextField("Try Florida",15); //default place of arrival
UtilDateModel model = new UtilDateModel();
model.setDate(2020, 8, 24);
UtilDateModel model2 = new UtilDateModel();
model2.setDate(2020, 8, 24);
JDatePanellImpl datePanel = new JDatePanellImpl(model);
JDatePanellImpl datePanel2 = new JDatePanellImpl(model2);
datePicker1 = new JDatePickerImpl(datePanel);
datePicker2 = new JDatePickerImpl(datePanel2);
JLabel label9=new JLabel("From:");
JLabel label10=new JLabel("To:");
```

```
label1.setBounds(80, 800, size.width, size.height);
panel0.setPreferredSize(new Dimension(600, 100));
panel1.setPreferredSize(new Dimension(600, 80));
panel2.setPreferredSize(new Dimension(600, 50));
panel3.setPreferredSize(new Dimension(600, 50));
panel4.setPreferredSize(new Dimension(600, 50));
panel5.setPreferredSize(new Dimension(600, 70));
frame.setBackground(Color.WHITE);
label1.setFont(label1.getFont().deriveFont(54.0f));
    label1.setVerticalTextPosition(JLabel.TOP);
label1.setHorizontalAlignment( JLabel.CENTER );
panel0.add(label0);
button.setText("Search");
button.setBounds(300, 340, 15, 15);
panel1.add(label1);
panel5.add(button);
panel2.add(dd1);
//panel2.add(dd2);
panel2.add(label81);
panel2.add(tf1);
panel2.add(tf2);
panel2.add(dd3);
panel3.add(label9);
panel3.add(start);
panel3.add(label10);
panel3.add(end);
panel4.add(cal);
panel4.add(datePicker1);
    panel4.add(datePicker2);
frame.add(panel0);
frame.setLayout(new FlowLayout());
frame.add(panel1);
frame.add(panel2);
frame.add(panel3);
frame.add(panel4);
frame.add(panel5);
frame.setSize(600, 500);
    frame.setLocationRelativeTo(null);
```

```
frame.setLocationRelativeTo(null);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
ConnectDB obj=new ConnectDB();
}
public void actionPerformed(ActionEvent ae)
{
try {
    if(ae.getSource()==button)
    {
        showTableData();
        okbtn.addActionListener(this);
    }
    else if(ae.getSource()==okbtn)
    {showdetails();
     confi.addActionListener(this);
    }
    else if(ae.getSource()==confi)
    confirmBook();
    else
    {
    }
} catch (InstantiationException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IllegalAccessException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
```

```
public void confirmBook()
{
    frame3=new JFrame("Confirm Booking");
    frame3.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame3.setLayout(new FlowLayout());
    JPanel plast=new JPanel(new FlowLayout());
    JLabel llast=new JLabel("Confirmed Booking");
    plast.add(llast);
    frame3.add(plast);
    frame3.setVisible(true);
    frame3.setSize(100,100);
}
```

```
public void showdetails() throws InstantiationException,
IllegalAccessException, ClassNotFoundException, SQLException
{
```

```
    frame2=new JFrame("Details of Airlines");
    frame2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame2.setLayout(new GridLayout(5,1));
    String text=jb.getText();
    String text2=jb2.getText();
    int itext=Integer.parseInt(text);//which option number he chose for going
    int itext2=Integer.parseInt(text2);//which option number he chose for
return
```

```
//For Going
```

```
Class.forName("com.mysql.jdbc.Driver").newInstance();
connection=DriverManager.getConnection( url,username,password);
PreparedStatement ps= connection.prepareStatement("select*from
world.st");//Enter query written in showTableData() in brackets
ResultSet rs = ps.executeQuery();
```

```
Statement stmt = (Statement)connection.createStatement();
int i =0;
String name= "";//variables for storing details,can ignore
String id= "";//variables for storing details,can ignore
String grade = "";//variables for storing details,can ignore
JPanel p0=new JPanel(new FlowLayout());
JPanel p4=new JPanel(new FlowLayout());
JPanel p=new JPanel(new FlowLayout());
JLabel res=new JLabel("");
String query1="";
String query2="";
while(rs.next())
{
    i++;
}
ps= connection.prepareStatement("select*from
world.st");//Enter the same query written above
rs = ps.executeQuery();
int j=1,flag=0;
if((itext<=i)&&(itext>=1)) //for valid entries

{
    while(rs.next())

    {
        if(j==itext)

        { flag=1;
            res.setText("Valid Entry for Departure Ticket");
            JLabel res1=new
JLabel(rs.getString("st_name"));//to print details if there is a valid entry,just
change the attribute name in brackets
            JLabel res2=new JLabel(rs.getString("st_id"));//to
print details if there is a valid entry,just change the attribute name in
brackets
```

```

JLabel res3=new JLabel(rs.getString("grade"));//to print details if there is a valid entry,just
change the attribute name in brackets
    p0.add(res);
    p.add(res1);
    p.add(res2);
    p.add(res3);
}
j++;} }

else//for invalid entries
{
    res.setText("Invalid Entry for Departure Ticket");
    p0.add(res);
}

if(flag==1)
{//query1= "";//Enter the updation query
    // stmt.executeUpdate(query1);//to execute update query
}

PreparedStatement ps2= connection.prepareStatement("select*from
world.st");//Enter query written in showTableData() in brackets

ResultSet rs2 = ps.executeQuery();
Statement stmt2 = (Statement)connection.createStatement();
int i2 =0;
String name2= "";//variables for storing details,can ignore
String id2= "";//variables for storing details,can ignore
String grade2 = "";//variables for storing details,can ignore
JPanel p2=new JPanel(new FlowLayout());
JPanel p3=new JPanel(new FlowLayout());
JLabel res4=new JLabel("");
while(rs2.next())
{
    i2++;
}
ps2= connection.prepareStatement("select*from world.st");//Enter the
same query written above

rs2 = ps2.executeQuery();
int j2=1,flag2=0;
if((itext2<=i2)&&(itext2>=1)) //for valid entries
{
    while(rs2.next())
    {
        if(j2==itext2)

```

```
{ flag2=1;
        res4.setText("Valid Entry for Return Ticket");
        JLabel res5=new JLabel(rs2.getString("st_name"));//to print
details if there is a valid entry,just change the attribute name in brackets
        JLabel res6=new JLabel(rs2.getString("st_id"));//to print
details if there is a valid entry,just change the attribute name in brackets
        JLabel res7=new JLabel(rs2.getString("grade"));//to print
details if there is a valid entry,just change the attribute name in brackets
        p2.add(res4);
        p3.add(res5);
        p3.add(res6);
        p3.add(res7);
        p4.add(confi);
    }
    j2++;
}
}
else if(itext==0)
{
}
else//for invalid entries
{
    res4.setText("Invalid Entry for Return Ticket");
    p2.add(res4);
}
if(flag2==1)
{//query2= "";//Enter the updation query
    //stmt2.executeUpdate(query2);//to execute update query
}
frame2.add(p0);
frame2.add(p);
frame2.add(p2);
frame2.add(p3);
frame2.add(p4);
frame2.setVisible(true);
frame2.setSize(600,250);
}
```

```

public void showTableData() throws SQLException, InstantiationException,
IllegalAccessException, ClassNotFoundException
{
    frame1 = new JFrame("Database Search Result");
    frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame1.setLayout(new GridLayout(4,1));
    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(columnNames);
    table = new JTable();
    table.setModel(model);
    table.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);

    table.setFillsViewportHeight(true);
    JScrollPane scroll = new JScrollPane(table);

    DefaultTableModel model2 = new DefaultTableModel();
    model2.setColumnIdentifiers(columnNames);
    table2 = new JTable();
    table2.setModel(model2);
    table2.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
    table2.setFillsViewportHeight(true);
    JScrollPane scroll2 = new JScrollPane(table2);

    JPanel p1=new JPanel(new BorderLayout());
    JPanel p2=new JPanel(new FlowLayout());
    JPanel p3=new JPanel(new BorderLayout());
    JPanel p4=new JPanel(new FlowLayout());
    Date d1=new Date();
    Date d2=new Date();
    d1=(Date)datePicker1.getModel().getValue(); //day of travel
    d2=(Date)datePicker2.getModel().getValue(); //day of return , ignore if not a round trip
    String sdes=start.getText();
    String edes=end.getText();
    String trip_type=(String) dd1.getSelectedItem(); //One way or round trip
    //String pass_type=(String) dd2.getSelectedItem(); //Adult or child
    String seat_type=(String) dd3.getSelectedItem(); //Economy or any other
}

```

```

int textvalue = Integer.parseInt(tf1.getText());//number of adult passengers
int textvalue2 = Integer.parseInt(tf2.getText());//number of infant passengers
String name= "";
String id= "";
String grade = "";

Class.forName("com.mysql.jdbc.Driver").newInstance();
connection=DriverManager.getConnection( url,username,password);
PreparedStatement ps= connection.prepareStatement("select*from world.st");//Enter
display query in brackets for going
    ResultSet rs = ps.executeQuery();
    int i =0;
    while(rs.next())//for going
    {
        name = rs.getString("st_name");
        id = rs.getString("st_id");
        grade = rs.getString("grade");
        model.addRow(new Object[]{name,id,grade});
        i++;
    }
    //String queryf="{}"; // if no return value
    //String queryf="?={}"; // if return value
    //CallableStatement stmt=connection.prepareCall(queryf);//for functions
    //rs=stmt.executeQuery();//for executing functions, pass parameters in
executeQuery()
    PreparedStatement ps2= connection.prepareStatement("select*from
world.st");//Enter display query in brackets for return
    ResultSet rs2 = ps2.executeQuery();
    i =0;
    while(rs2.next())//for return
    {
        name = rs2.getString("st_name");
        id = rs2.getString("st_id");
        grade = rs2.getString("grade");
        model2.addRow(new Object[]{name,id,grade});
        i++;
    }

```

```

int textvalue = Integer.parseInt(tf1.getText());//number of adult passengers
int textvalue2 = Integer.parseInt(tf2.getText());//number of infant passengers
String name= "";
String id= "";
String grade = "";

Class.forName("com.mysql.jdbc.Driver").newInstance();
connection=DriverManager.getConnection( url,username,password);
PreparedStatement ps= connection.prepareStatement("select*from world.st");//Enter
display query in brackets for going
    ResultSet rs = ps.executeQuery();
    int i =0;
    while(rs.next())//for going
    {
        name = rs.getString("st_name");
        id = rs.getString("st_id");
        grade = rs.getString("grade");
        model.addRow(new Object[]{name,id,grade});
        i++;
    }
    //String queryf="{}"; // if no return value
    //String queryf="?={}"; // if return value
    //CallableStatement stmt=connection.prepareCall(queryf);//for functions
    //rs=stmt.executeQuery();//for executing functions, pass parameters in
executeQuery()
    PreparedStatement ps2= connection.prepareStatement("select*from
world.st");//Enter display query in brackets for return
    ResultSet rs2 = ps2.executeQuery();
    i =0;
    while(rs2.next())//for return
    {
        name = rs2.getString("st_name");
        id = rs2.getString("st_id");
        grade = rs2.getString("grade");
        model2.addRow(new Object[]{name,id,grade});
        i++;
    }

```

```
p1.add(scroll);  
  
okbtn=new JButton("Confirm");  
p2.add(jl);  
p2.add(jb);  
p3.add(scroll2);  
p4.add(jl2);  
p4.add(jb2);  
p4.add(okbtn);
```

```
frame1.add(p1);  
frame1.add(p2);  
frame1.add(p3);  
frame1.add(p4);  
frame1.setVisible(true);  
frame1.setSize(400,700);  
}
```

```
}
```

