DBMS Project Report

PES University

Database Management Systems

UE18CS252

AIRPORT MANAGEMENT SYSTEM

Submitted By

PES2201800098 SAURAV PRASAD

Travel is really important in the current world of economic growth. Passengers travel around the world for holidays, business meetings or just to visit their near and dear ones. This database keeps a track of all passengers that travel via different airlines to airports in different cities. To facilitate the travel a number of employees work at each airport. The miniworld is hence identified. An entity-relation diagram is made and the relational schema is concluded from it. Different queries to retrieve passenger, employee, flight and airport details in different situations is written. For example, to know the details of flights which are delayed. We can also find average price of a ticket or average age of passengers flying. This can help in analysing the market of travel, its growth rate. Triggers have also been written to capture results of certain operations(Insert/Update) on tables.The system is limited to a sample data set and certain assumptions discussed in Conclusion section. The limitations can be seen as a scope for future enhancement.

Introduction

The system is based on airport management. Airport management system primarily deals with management of airport, airlines and passengers. The system provides broad overview of underlying operational factors that influence the airport management.

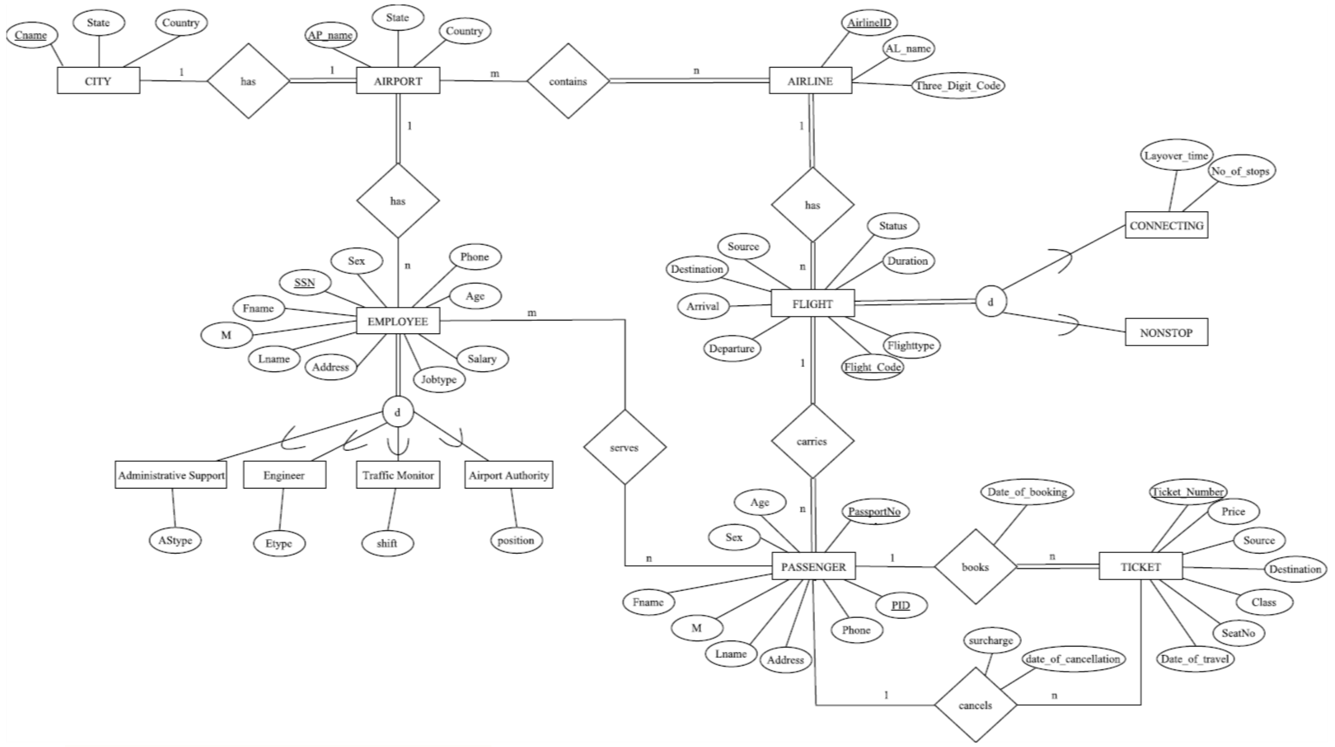
- The database system has the data of all commercial service airports.
- An airport is located in a city.
- All International airlines operating through various countries across the world have their offices located in all major cities and airports they cover. Hence, an airport may have many airline offices.
- Every airline is identified uniquely by an airline code. Airline code is a two-letter airline designator. Airline also has three-digit code which is printed on an air ticket.
- Airline companies serve flights.
- Every flight is uniquely identified by a flight code. Flight code is a combination of an airline code and four-digit number.
    - Flight takes off from one airport and lands on another airport. Therefore, most important aspect of a flight is, its source and destination. Source and destination airports are identified using an airport's IATA code.
- International Air Transport Airport code is simply a location identifier. IATA code is a three-letter code designating many airports across the world. These codes are prominently displayed on baggage tags and printed on an air ticket.
    - Flight has an arrival time, departure time, duration. Flight has three types of classes: business, economy and first class.
    - Flight can be of two types such non-stop flight and a connecting flight.
    - Connecting flight is a flight which takes intermediate stop and changes a flight possibly change of an airline. But we are assuming that connecting flight does not change a flight that is at each stop, after layover time gets over, passengers aboard the same flight.
- Flight serves passengers. Flight carries passengers from source to destination.
- A passenger is uniquely identified by a passenger id and a passport number.
    - Every passenger has details such as name, address, age, sex, phone.
- For a passenger to travel by a flight, he needs a ticket. A ticket or air ticket is used to confirm that an individual has reserved a seat on a flight. With the ticket, a passenger is allowed to board the flight.
    - An air ticket has information such as the passenger's name, the issuing airline, ticket number, source, destination, journey date, seat no, class, fare.
    - Ticket number is the combination of airline's 3-digit code, 4-digit form number, 6-digit serial number.

- Hence, depending on airline, source, destination, journey date and most importantly class, which a passenger chooses fare or price of an air ticket is determined.

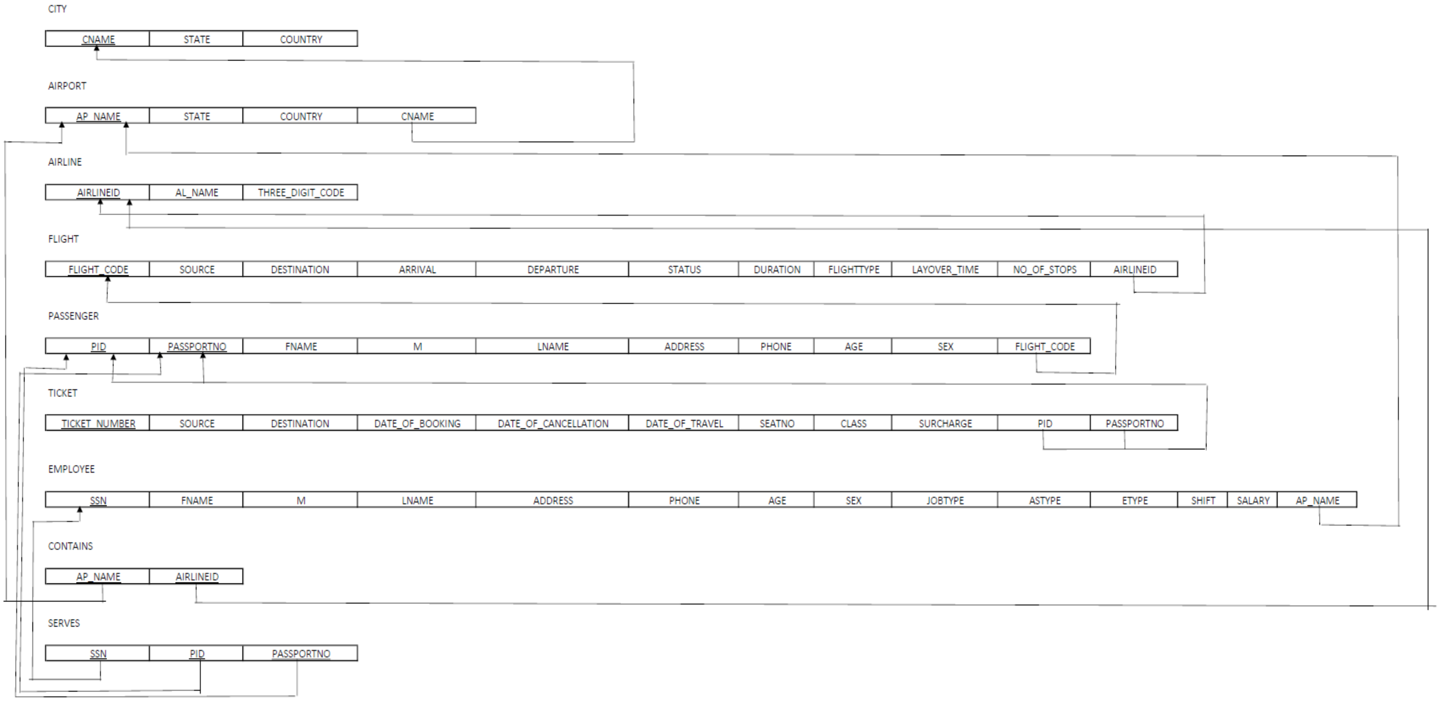
- A passenger can book one or multiple tickets. The day on which he books an air ticket is a booking date. Similarly, a passenger can cancel one or multiple tickets. The day on which he cancels an air ticket is cancellation date and there will be a surcharge that a passenger has to pay after cancelling a ticket.
- Every airport has employees working for it.
  - Every employee is identified by SSN. Every employee has an information such name, address, phone, age, sex, salary.
  - Employees in the role of administrative support, engineer, traffic controller and airport authority work at the airport.
  - Every airline needs administrative support staff to keep the office running smoothly. The different positions include secretaries, data entry workers, receptionists, communications and PR specialists and human resources department.
  - There are different types of engineers who work specifically with information technologies, electronics, flight structure, environmental regulations, etc.
  - Traffic Monitor works in different shifts such as day or night.
  - There are different positions that airport authorities might work at such as manager, attendee, assistant, pilot, etc.
- Employees working in the role of administrative support may help passengers with various tasks such as booking a flight ticket, solving passenger's questions, etc.

Data Model

ER DIAGRAM



CONCEPTUAL SCHEMA



CITY

CNAME	STATE	COUNTRY

AIRPORT

AP_NAME	STATE	COUNTRY	CNAME

AIRLINE

AIRLINEID	AL_NAME	THREE_DIGIT_CODE

FLIGHT

FLIGHT_CODE	SOURCE	DESTINATION	ARRIVAL	DEPARTURE	STATUS	DURATION	FLIGHTTYPE	LAYOVER_TIME	NO_OF_STOPS	AIRLINEID

PASSENGER

PID	PASSPORTNO	FNAME	M	LNAME	ADDRESS	PHONE	AGE	SEX	FLIGHT_CODE

TICKET

TICKET_NUMBER	SOURCE	DESTINATION	DATE_OF_BOOKING	DATE_OF_CANCELLATION	DATE_OF_TRAVEL	SEATNO	CLASS	SURCHARGE	PID	PASSPORTNO

EMPLOYEE

SSN	FNAME	M	LNAME	ADDRESS	PHONE	AGE	SEX	JOBTYPE	ASTYPE	ETYPE	SHIFT	SALARY	AP_NAME

CONTAINS

AP_NAME	AIRLINEID

SERVES

SSN	PID	PASSPORTNO

Functional Dependencies and Normalization

VIOLATION OF FIRST NORMAL FORM

- The 1st normal form says that the values at the intersection of each row and column should be atomic. The EMPLOYEE1 and PASSENGER2 table contains a column for address which is a multi-valued attribute. It contains street number, city name etc. So, it violates 1st normal form which can be solved by dividing address into separate columns for street no., city name etc.

VIOLATION OF SECOND NORMAL FORM

- The PASSPORTNO and PID is the candidate key in the table for entity PASSENGER. But the PASSPORTNO, the part of candidate key shows partial dependency.
  PASSPORTNO -> FNAME, M, LNAME, ADDRESS, PHONE, AGE, SEX
  Similarly, PID the part of candidate key also shows partial dependency.
  PID -> FLIGHT_CODE
  So, the 2nd normal form is violated as both PASSPORTNO and PID as a part of candidate key shows partial dependency.
  To normalize the table for PASSENGERS it is divided into 3 tables as follows
  PASSENGER1 (PID, PASSPORTNO)
  PASSENGER2(PASSPORTNO, FNAME, M, LNAME, ADDRESS, PHONE, AGE, SEX)
  PASSENGER3 (PID, FLIGHT_CODE)

VIOLATION OF THIRD NORMAL FORM

- The TICKET_NUMBER in the table for entity TICKET is the candidate key which determines the DATE_OF_BOOKING and DATE_OF_CANCELLATION which further determines some other attributes.
  TICKE_TNUMBER -> DATE_OF_BOOKING, DATE_OF_CANCELLATION.
  DATE_OF_BOOKING, SOURCE, DESTINATION, CLASS -> PRICE
  DATE_OF_CANCELLATION -> SURCHARGE

Thus DATE_OF_BOOKING and DATE_OF_CANCELLATION shows transitive dependency which violates 3rd normal form. So, the table is not in 3rd normal form and to normalize it; it is divided into 3 tables as follows

TICKET1 (TICKET_NUMBER, SOURCE, DESTINATION, DATE_OF_BOOKING, DATE_OF_TRAVEL, SEATNO, CLASS, DATE_OF_CANCELLATION, PID, PASSPORTNO)

TICKET2 (DATE_OF_BOOKING, SOURCE, DESTINATION, CLASS, PRICE)

TICKET3 (DATE_OF_CANCELLATION, SURCHARGE)

- Similarly, in table EMPLOYEE, SSN is the candidate key which determines JOBTYPE.

  SSN-> JOBTYPE

  JOBTYPE -> SALARY

  So, it shows transitive dependency and violates 3rd normal form. To normalize it is dived into 2 tables as follows.

  EMPLOYEE1 (SSN, FNAME, M, LNAME, ADDRESS, PHONE, AGE, SEX, JOBTYPE, ASTYPE, ETYPE, SHIFT, POSITION, AP_NAME)

  EMPLOYEE2(JOBTYPE, SALARY)

DDL

Creating Table: CITY

```
CREATE TABLE CITY
(CNAME VARCHAR (15) NOT NULL,
STATE VARCHAR (15),
COUNTRY VARCHAR (30),
PRIMARY KEY(CNAME));
```

Creating Table: AIRPORT

```
CREATE TABLE AIRPORT
(AP_NAME VARCHAR2(100) NOT NULL,
STATE VARCHAR (15),
COUNTRY VARCHAR (30),
CNAME VARCHAR (15),
PRIMARY KEY(AP_NAME),
FOREIGN KEY(CNAME) REFERENCES CITY(CNAME) ON DELETE CASCADE);
```

Creating Table: AIRLINE

```
CREATE TABLE AIRLINE
(AIRLINEID VARCHAR (3) NOT NULL,
AL_NAME VARCHAR (50),
THREE_DIGIT_CODE VARCHAR (3),
PRIMARY KEY(AIRLINEID));
```

Creating Table: CONTAINS

```
CREATE TABLE CONTAINS
(AIRLINEID VARCHAR (3) NOT NULL,
AP_NAME VARCHAR (100) NOT NULL,
PRIMARY KEY (AIRLINEID, AP_NAME),
FOREIGN KEY (AIRLINEID) REFERENCES AIRLINE(AIRLINEID) ON DELETE CASCADE,
FOREIGN KEY (AP_NAME) REFERENCES AIRPORT(AP_NAME) ON DELETE
CASCADE);
```

Creating Table: FLIGHT

```
CREATE TABLE FLIGHT
(FLIGHT_CODE VARCHAR (10) NOT NULL,
SOURCE VARCHAR (3),
DESTINATION VARCHAR (3),
ARRIVAL VARCHAR (10),
DEPARTURE VARCHAR2(10),
```

Creating Table: PASSENGER1

```
CREATE TABLE PASSENGER1
(PID INT NOT NULL,
PASSPORTNO VARCHAR (10) NOT NULL,
PRIMARY KEY (PID, PASSPORTNO));
```

Creating Table: PASSENGER2

```
CREATE TABLE PASSENGER2
(PASSPORTNO VARCHAR(10) NOT NULL,
FNAME VARCHAR(20),
M VARCHAR(1),
LNAME VARCHAR(20),
ADDRESS VARCHAR(100),
PHONE BIGINT,
AGE INT,
SEX VARCHAR(1),
PRIMARY KEY(PASSPORTNO));
```

Creating Table: PASSENGER3

```
CREATE TABLE PASSENGER3
(PID INT NOT NULL,
FLIGHT_CODE VARCHAR(10),
PRIMARY KEY(PID),
FOREIGN KEY(FLIGHT_CODE) REFERENCES FLIGHT(FLIGHT_CODE) ON DELETE CASCADE);
```

Creating Table: EMPLOYEE

```
CREATE TABLE EMPLOYEE1
(SSN INT NOT NULL,
FNAME VARCHAR(20),
M VARCHAR(1),
LNAME VARCHAR(20),
ADDRESS VARCHAR(100),
PHONE BIGINT,
AGE INT,
SEX VARCHAR(1),
JOBTYPE VARCHAR(30),
ASTYPE VARCHAR(30),
ETYPE VARCHAR(30),
SHIFT VARCHAR(20),
POSITION VARCHAR(30),
AP_NAME VARCHAR(100),
PRIMARY KEY(SSN),
FOREIGN KEY(AP_NAME) REFERENCES AIRPORT(AP_NAME) ON DELETE CASCADE);
```

Adding CHECK constraint on an attribute AGE

AGE OF AN EMPLOYEE WORKING FOR AN AIRPORT SHOULD NOT BE GREATER THAN 65--

```
ALTER TABLE EMPLOYEE1
ADD CONSTRAINT AGE_LIMIT CHECK(AGE < 65);
```

Creating Table: EMPLOYEE2

```
CREATE TABLE EMPLOYEE2
(JOBTYPE VARCHAR(30) NOT NULL,
SALARY INT,
PRIMARY KEY(JOBTYPE));
```

Creating Table: SERVES

```
CREATE TABLE SERVES
(SSN INT NOT NULL,
PID INT NOT NULL,
PASSPORTNO VARCHAR(10) NOT NULL,
PRIMARY KEY(SSN, PID, PASSPORTNO),
FOREIGN KEY(SSN) REFERENCES EMPLOYEE1(SSN) ON DELETE CASCADE,
FOREIGN KEY(PID, PASSPORTNO) REFERENCES PASSENGER1(PID, PASSPORTNO)
ON DELETE CASCADE);
```

Creating Table: TICKET1

```
CREATE TABLE TICKET1
(TICKET_NUMBER VARCHAR(13) NOT NULL,
SOURCE VARCHAR(3),
DESTINATION VARCHAR(3),
DATE_OF_BOOKING DATE,
DATE_OF_TRAVEL DATE,
SEATNO VARCHAR(5),
CLASS VARCHAR(15),
DATE_OF_CANCELLATION DATE,
PID INT,
PASSPORTNO VARCHAR(10),
FOREIGN KEY(PID, PASSPORTNO) REFERENCES PASSENGER1(PID, PASSPORTNO)
ON DELETE CASCADE);
```

Adding CHECK constraint on an attribute TICKET_NUMBER

```
ALTER TABLE TICKET1
ADD CONSTRAINT TICKET_NO_LENGTH CHECK(LENGTH(TICKET_NUMBER)=13);
```

Creating Table: TICKET2

```
CREATE TABLE TICKET2
(DATE_OF_BOOKING DATE NOT NULL,
SOURCE VARCHAR(3) NOT NULL,
DESTINATION VARCHAR(3) NOT NULL,
CLASS VARCHAR(15) NOT NULL,
PRICE INT,
PRIMARY KEY(DATE_OF_BOOKING, SOURCE, DESTINATION, CLASS));
```

Creating Table: TICKET3

```
CREATE TABLE TICKET3
(DATE_OF_CANCELLATION DATE NOT NULL,
SURCHARGE INT,
PRIMARY KEY(DATE_OF_CANCELLATION));
```

Triggers

- **Trigger to log an entry in a different table i.e. DELAYED_FLIGHTS when a flight is delayed**

```
CREATE TABLE DELAYED_FLIGHTS
(FLIGHT_CODE VARCHAR(10),
DESTINATION VARCHAR(3),
SOURCE VARCHAR(3),
AIRLINEID VARCHAR(3));

CREATE TRIGGER DELAYEDFLIGHTS ON FLIGHT AFTER INSERT
AS
DECLARE @STATUS VARCHAR(10);
SELECT @STATUS= i.STATUS FROM inserted i;
BEGIN
IF (@STATUS='DELAYED')
BEGIN
INSERT INTO DELAYED_FLIGHTS (FLIGHT_CODE, SOURCE, DESTINATION, AIRLINEID )
SELECT i.FLIGHT_CODE,i.SOURCE,i.DESTINATION,i.AIRLINEID
from inserted i;
END
END
GO
```

- **Trigger to update TICKET_PRICE_HISTORY table when the price of the air ticket is updated in TICKET2 table.**

```
CREATE TABLE TICKET_PRICE_HISTORY
(DATE_OF_BOOKING DATE NOT NULL,
SOURCE VARCHAR(3) NOT NULL,
DESTINATION VARCHAR(3) NOT NULL,
CLASS VARCHAR(15) NOT NULL,
PRICE INT,
PRIMARY KEY(DATE_OF_BOOKING, SOURCE, DESTINATION, CLASS));
```

```
CREATE TRIGGER TRIGG_TICKET_PRICE_HISTORY ON TICKET2 FOR UPDATE
AS
DECLARE @PRICE INT;
SELECT @PRICE= i.PRICE from inserted i;
BEGIN
IF UPDATE(PRICE)
BEGIN
INSERT INTO TICKET_PRICE_HISTORY(DATE_OF_BOOKING, SOURCE,
DESTINATION, CLASS,PRICE)
SELECT i.DATE_OF_BOOKING,i.SOURCE,i.DESTINATION,i.CLASS, @PRICE
FROM inserted i;
END
END
GO
```

SQL Queries

- SQL Query to find average price of ticket from BOM To DFW on ’11th May’.

```
SELECT AVG(PRICE) FROM TICKET2 WHERE DATE_OF_BOOKING=’11-MAY-16’
AND SOURCE= ‘BOM’
AND DESTINATION= ‘DFW’;
```

- SQL Query to display the details of Flights which are delayed.

```
SELECT DISTINCT F.FLIGHT_CODE ,AL.AL_NAME ,F.ARRIVAL,F.DEPARTURE,
F.SOURCE, F.DESTINATION, F.STATUS, F.FLIGHTTYPE
FROM AIRLINE AL, AIRPORT AP, FLIGHT F
WHERE AL. AIRLINEID=F.AIRLINEID
AND F.STATUS = ‘Delayed ’;
```

- SQL Query to display the details of passengers flying to Bombay in economy class.

```
SELECT
AL.AL_NAME, FL.FLIGHT_CODE, P2.FNAME, P2.LNAME,P2.PASSPORTNO,
T.CLASS, T.DATE_OF_TRAVEL, T.DESTINATION, T.SOURCE, T.SEATNO,
T.TICKET_NUMBER FROM AIRLINE AL, FLIGHT FL, PASSENGER1 P1,
PASSENGER2 P2, PASSENGER3 P3, TICKET1 T
WHERE AL.AIRLINEID = FL.AIRLINEID
AND P1.PID= P3.PID
AND P1.PASSPORTNO =P2.PASSPORTNO
AND FL.FLIGHT_CODE = P3.FLIGHT_CODE
AND T.PASSPORTNO = P2.PASSPORTNO
AND T.CLASS='ECONOMY' AND T.DESTINATION='BOM';
```

- SQL Query to find the average age of passengers traveling from BOM to DFW.

```
SELECT AVG(AGE) FROM PASSENGER2 WHERE PASSPORTNO IN (SELECT
P.PASSPORTNO FROM PASSENGER1 P, TICKET1 T WHERE P.
PASSPORTNO=T.PASSPORTNO AND P.PID=T.PID AND T.SOURCE= ‘BOM’ AND
T.DESTINATION=‘DFW’)
```

- SQL Query to find the number of employees who work in India’.

```
SELECT COUNT(*) FROM EMPLOYEE1 WHERE AP_NAME IN
(SELECT AP_NAME FROM AIRPORT WHERE COUNTRY=‘INDIA’);
```

- SQL Query to display passport number and ticket number of passengers flying to Delhi along with their flight name and flight code

  ```
  SELECT AL_NAME,FLIGHT.FLIGHT_CODE, TICKET1.TICKET_NUMBER,
  TICKET1.PASSPORTNO
  FROM TICKET1 FULL OUTER JOIN FLIGHT
  ON TICKET1.SOURCE = FLIGHT.SOURCE
  AND TICKET1.DESTINATION=FLIGHT.DESTINATION
  FULL OUTER JOIN AIRLINE ON FLIGHT.AIRLINEID = AIRLINE.AIRLINEID
  WHERE TICKET1.DESTINATION='DEL'
  ```

- SQL Query to figure out which employees are serving which passengers.

  ```
  SELECT EMPLOYEE1.SSN, FNAME,LNAME, PID, PASSPORTNO, JOBTYPE
  FROM EMPLOYEE1 FULL OUTER JOIN SERVES
  ON EMPLOYEE1.SSN = SERVES.SSN;
  ```

- SQL Query to find out the ssn and job type of employees who work at the airport but do not serve any passenger

  ```
  SELECT E.SSN,E.FNAME,E.JOBTYPE
  FROM EMPLOYEE1 E
  WHERE NOT EXISTS (SELECT * FROM SERVES S WHERE E.SSN =S.SSN );
  ```

Conclusion

The system is capable of keeping a track of airports in different cities , employees working in those airports, different airlines that travel to and from the airport, different passengers that travel in those airlines.

Limitation:

- We are not considering privately managed airports. We are only considering publicly owned airports.

  Several Categories of airports:

  (1) Commercial Service airport - These are publicly owned airports that serve aircrafts which provide scheduled passenger service.

  (2) Cargo Service airports- These airports serve aircrafts carrying cargo only
  A city has at most one international airport.

- For Connecting flights, flight and airline remains same at layover stops
- There are different types of jobs available at the airport. For simplicity our system considers a few jobs only.

Future Enhancement:

All the above limitations can be implemented and incorporated in the database in future. The data stored in the database is currently a sample set. Real time data will help in further enhancements.