AIRPLANE RESERVATION SYSTEM DBMS DESIGN USING SQL

December 23, 2021

1 Introduction

1.1 Overview

Airline reservation system is one of the most used database system in the world. It is an example of Transaction processing systems. Transaction processing systems are systems with large databases and hundreds of concurrent users executing database transactions. These systems require high availability and fast response time for hundreds of concurrent users.We define the concept of a transaction, which is used to represent a logical unit of database processing, that must be completed in its entirety to ensure correct
ness. A transaction is typically implemented by a computer program, which includes database commands such as retrievals, insertions, deletions, and updates. In this project, we deal with the database part of the whole system with insertions, deletions and updates as the primary task of our system along with maintenance of the integrity of the system at all stages of a transaction.The system that is being described in this report handles everything that the most practical systems would do. The complexity of the database system has been handled by trying to make most basic entities resembling the real world objects. This DBMS handles the most basic functions of an airline reservation system, including reservation, cancellation and updating of a flight trip transaction.

1.2 Requirement Analysis

1.2.1 Feasibility Study Risk Analysis:

It is the most difficult area to assess because objectives, functions, and performance are somewhat hazy; anything seems possible if right assumptions are made. A clinical attitude should prevail during an evaluation of technical feasibility. The considerations normally attached with the technical feasibility:
Development Risk:
Can the system element be designed so that necessary function and performance are achieved within the constraints uncovered during analysis
Resource availability:
Are skilled staffs available to develop the system element in Question? Are any other necessary resources (hardware and software) available to build the system?

Technology:

i Has the relevant technology progressed to a state that will support the system?

ii All of the above consideration also applies to the work we have done. As far as developments, risks are concerned, yes necessary functions and the constraints under which they have to perform have been identified and divided into modules so that each module performs its own assigned task.

iii As for skilled staff for development is concerned, we are the only person performing this task and we have fully understood the problem. We are sufficiently equipped with the use of programming and can perform this tasks in the given time constraints.

iv The use of programming language enables the programmer to develop software that can help end-user to operate the system more easily. We can use GUI tool VB.NET or Java to build a GUI for the system but the report deals with only the database design part of the system and for that we use Oracle SQL developer. The tool comprises of all the components required to solve the problem system.

1.2.2 Economical Feasibility:

An evaluation of development cost weighed against the ultimate income or benefit derived from the development system or product. It includes a broad range of concerns such as:

i Cost benefit Analysis

ii Long term corporate income strategies

iii Impact on other profits/products

iv Cost of resources needed for development

v Potential market growth

The work being done is economically feasible since the work is not being done at very large scale, although it might be a bit complex. The cost of resources needed to do the work would not be big. The whole task could be completed by a single resource in given time.

1.2.3 Operational Feasibility:

This study helps us in finding whether the work to be done will be operational with the available staff and with in the given time. The staff is fully capable of handling the database system. The IT literacy is of good order and the software has been made in such a way that it becomes easier for the user to answer queries being asked. This will facilitate easy use and adopttability of the system. Based on this, it was felt that the proposed system would be operationally feasible. With the use of menus, simple command buttons and proper validation required it become fully understandable to the common user and operational with the user. Moreover, we can pack and associate different canned queries with the GUI components to make it more friendly for users and also apply principles from cognitive science to deal with the user interaction with our system.

1.2.4 Targeted Audience

i Agent: Usually books for multiple travellers regularly

ii Traveller: Usually books for himself or at most family/friends

iii Airline agent: Can be found at company counter usually at the airport

1.2.5 Targeted Organisation

i Hotels

ii Travel Agency Company

iii International Business Company

iv Airline Company

1.2.6 Accessibility

Only Admin and Developer are able to access the information to manage and update.

1.2.7 Need To Standardized

Yes,the data need to be standardized due to disparity. As the data contains different ages of people in different classes of airplane.

1.2.8 Volume of Data

All the information about customers,airplane flights etc are required to store and manage.Therefore,huge amount of storage is required to sustain the Airplane Reservation System database.

2 ER DIAGRAM

We start with the airport. Airport is an independent entity, which is uniquely identified by a code with address as an important attribute. Address contains different attributes under it; we make this as a composite attribute. Similarly we will define the primary attributes and other attributes for different entities in the report subsequently. We will lay more stress on the relationships. Now we move to airline. We uniquely identify the plane and assume that a plane has an access to the airport or not. First we thought of giving the access to Airline Company but practically, it is a very strict criterion. Sometimes planes might be landed on an airport because of emergency landing or other issues and because the company is not allowed, it might lead to a disaster. So we limit it to a plane. If we give access to a company, we are giving the access to all the planes of that company.

Our Assumption:

i The access to an airport is dependent on the airplane and not the airline company. An airport can give access to multiple planes and a plane can have access to multiple airports.

ii Every plane must belong to an airline company. So a full participation of plane is justified. A company can have multiple planes but a plane must belong to one company.

iii Each seat belongs to a plane and plane has multiple seats.

iv Admin can scheduled and modified flight trip.

v Fare cannot exist without a trip and a trip has one far.

3 RELATIONSHIP ASSUMPTION FOR ER DIAGRAM

i Airline company can own many airplanes, but no airplanes can be without an airline company.

ii Address is a multi-value attribute

iii Multiple airplanes can land on an airport and also multiple airports can handle multiple airplanes.

iv An airplane-seat cannot exists without an airplane.

v Fare exists only if there is a flight trip.

vi Customer can book multiple flight trip.

vii Admin can modified and scheduled multiple number of flight trip

viii Customer can pay for multiple flight.

3.1 ER-Diagram for Airplane Reservation System

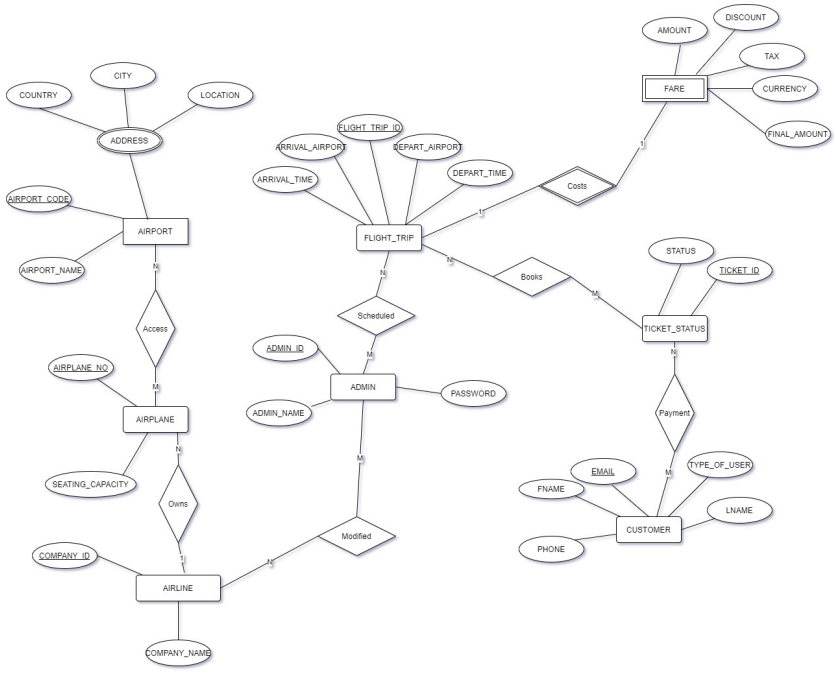

Figure 1: ERD

4 RELATIONAL DIAGRAM

A relational diagram refers to a visual representation of the relational database's entities, the relationships between those entities, and the attributes within those entities.

Rules For Translating ER diagram to Relational Model

Step 1-For Strong Entity

i Create a table that includes all the simple attributes

ii Include only the simple components of composite attributes

iii Choose a primary key

Step 2-For Weak Entity

i Create a table that includes all the simple attributes

ii Include the primary key from the owner entity, this will become a foreign key

Step 3- 1:1 Relations

i Copy the primary key from the first table into the second table – it becomes a foreign key in the second table

ii Any attributes tied to the relationship goes to the table with the foreign key

Step 4- 1:n Relations

i Choose the entity on the n-side and include a copy of the primary key of the entity on the 1-side. This field becomes a foreign key in the table on the nside

ii Any attributes tied to the relationship goes to the table with the foreign key

Step 5- N:M Relations

i Create a new table and include the primary keys from the two participating entities

ii Both of these fields become foreign keys in the new table

iii Any attributes tied to the relationship goes to the new table

Step 6- For Multivalued Attributes

i For each multivalued attribute (A) create a new table and move the attribute (A) to this table

ii Include the primary key from the entity that originally had (A) as an attribute

iii The combination of these attributes will be the primary key of the new table

Following all the above rules, the Relational diagram as follows:

4.1 Relational Diagram For Airplane Reservation System

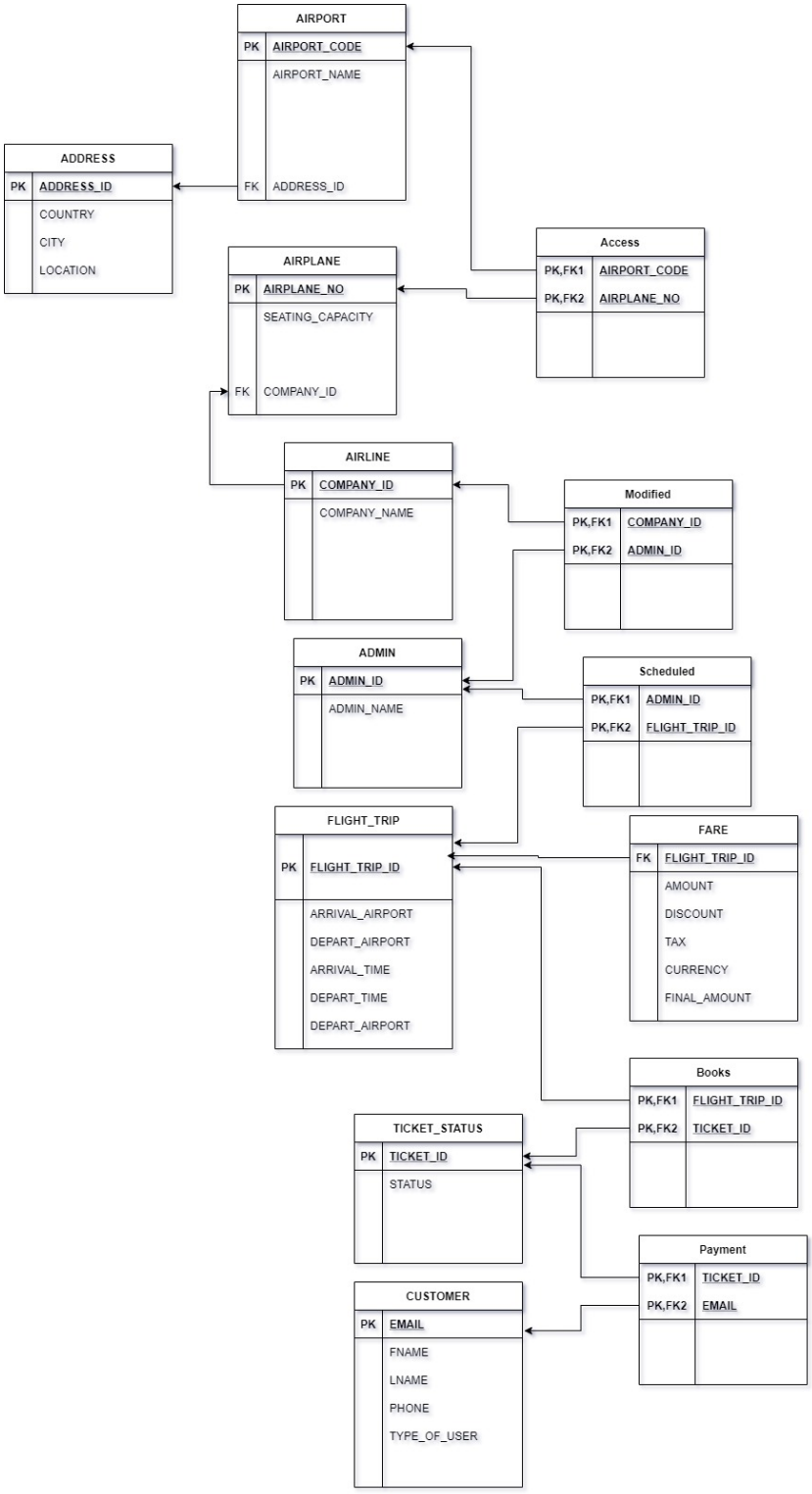

Figure 2: Relation Diagram

5 TABLES AND PRIMARY KEYS

i AIRPORT - Airport_Code

ii AIRLINE_COMPANY - Company_ID

iii AIRPLANE - Airplane_no

iv ACCESS - Airplane, Airport

v Scheduled - COMPANY_ID,ADMIN_ID

vi Modified - COMPANY_ID,ADMIN_ID

vii ADMIN - ADMIN_ID

viii FLIGHT_TRIP - Flight_trip_id

ix FARE - Flight_trip_id

x PAYMENT - TICKET_ID,EMAIL

xi CUSTOMER - Email

xii TICKET_STATUS - TICKET_ID

6 QUERIES

6.1 Insert and Create

AIRPORT:

```
 CREATE TABLE AIRPORT(
   AIRPORT_CODE   VARCHAR(25),
   AIRPORT_NAME   VARCHAR(100)    NOT NULL,
   LOCATION        VARCHAR(50)    NOT NULL,
   CITY          VARCHAR(50)   NOT NULL,

   COUNTRY       VARCHAR(50)   NOT NULL,

CONSTRAINT AIRPORT_CODE PRIMARY KEY (AIRPORT_CODE),
CONSTRAINT AIRPORT_CODE FOREIGN KEY (ADDRESS_ID) REFERENCE
   );
INSERT ALL
INTO AIRPORT VALUES( 'KTM' , 'TIA','Gaushala','Kathmandu','Nepal')
INTO AIRPORT VALUES( 'PKR' , 'Pokhara Airport','Siddhartha
    Highway','Pokhara','Nepal');
```

AIRPLANE:

```
CREATE TABLE AIRPLANE(
      AIRPLANE_NO     VARCHAR(25) ,
      SEATING_CAPACITY   NUMBER     NOT NULL,
      COMPANY       VARCHAR(50)   NOT NULL,
      CONSTRAINT AIRPLANE_NO PRIMARY KEY (AIRPLANE_NO),
CONSTRAINT FK_COMPANY_ID FOREIGN KEY (COMPANY_ID) REFERENCES
    AIRLINE(COMPANY_ID)
);
INSERT ALL
INTO AIRPLANE VALUES( 'AA751' , 40, 'AA' )
INTO AIRPLANE VALUES( 'AA851' , 60, 'AA' )
```

AIRLINE:

```
 CREATE TABLE AIRLINE(
      COMPANY_ID          VARCHAR(50) ,
      COMPANY_NAME     VARCHAR(100)  NOT NULL,
      CONSTRAINT COMPANY_PK PRIMARY KEY (COMPANY_ID)
);


INSERT ALL
INTO AIRLINE VALUES( 'AA' , 'Buddha Airline')
INTO AIRLINE VALUES( 'BA' , 'Turkish Airline')
```

FLIGHT_TRIP:

```
 CREATE TABLE FLIGHT_TRIP(
      FLIGHT_TRIP_ID     VARCHAR(15) ,

      ARRIVAL_AIRPORT  VARCHAR(25)  NOT NULL,
      ARRIVAL_TIME     DATE       NOT NULL,
      DEPART_AIRPORT   VARCHAR(25)   NOT NULL,
      DEPART_TIME      DATE       NOT NULL,
      CUSTOMER      VARCHAR(25)   NOT NULL,
CONSTRAINT FLIGHT_TRIP_PK PRIMARY KEY
(FLIGHT_TRIP_ID),
CONSTRAINT FK_TICKET_ID FOREIGN KEY
(TICKET_ID) REFERENCES TICKET_STATUS(TICKET_ID),
CONSTRAINT FK_ADMIN_ID FOREIGN KEY
(ADMIN_ID) REFERENCES ADMIN(ADMIN_ID)

);

INSERT ALL
INTO FLIGHT_TRIP
VALUES('kunish17dec','KTM',to_date('2014/12/27:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'PKR',to_date('2014/12/26:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'kunal.jagdish@gmail.com' )
INTO FLIGHT_TRIP
VALUES('kuniaa741','PKR',to_date('2014/12/27:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'KTM',to_date('2014/12/26:12:00:00AM',
'yyyy/mm/dd:hh:mi:ssam'),'kunal.jagdish@gmail.com' )
```

CUSTOMER:

```
CREATE TABLE CUSTOMER (
      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),
     CONSTRAINT FK_TICKET_ID FOREIGN KEY
(TICKET_ID) REFERENCES TICKET_STATUS(TICKET_ID)

);

INSERT ALL
INTO CUSTOMER VALUES
('kunaljagdish@gmail.com','Kunal','Arora',4694529484,'NORMAL' )
INTO CUSTOMER VALUES
('kxa142230@utdallas.edu,'Kunal','Arora',4694529484,'NORMAL' )
```

FARE:

```
CREATE TABLE FARE(
   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 FARE VALUES( 'kunish17dec',1760,200,'Rs',10,10 )
  INTO FARE VALUES( 'kuniaa741',176,200,'Rs',10,10 );
```

TICKET_STATUS:

```
 CREATE TABLE_STATUS(
   TICKET_ID     VARCHAR(25),
        STATUS      VARCHAR(100)  NOT NULL,


CONSTRAINT TICKET_ID PRIMARY KEY (TICKET_ID)
   );
INSERT ALL
INTO TICKET_ID('2009','Available' )
INTO TICKET_ID VALUES('2999','Cancelled' );
```

ADMIN:

```
 CREATE ADMIN(
   ADMIN_ID          VARCHAR(25),
   ADMIN_NAME       VARCHAR(25)   NOT NULL,


CONSTRAINT AIRPORT_PK PRIMARY KEY (ADMIN_ID)
   );
INSERT ALL
INTO ADMIN VALUES ('9001','Arora')
INTO ADMIN VALUES ('9002','Kunal');
```

7 Conclusion

Thus,using concept of DBMS Airplane Reservation System is designed.