**CASE STUDY**

**OF**

**AIRPORT**

**MANAGEMENT**

**SYSTEM**

**USING**

**RELATIONAL**

**DATABASE DESIGN**

**Case Study in Relational Database Design**

* **Title:** Airport Management System
* **Student Names:** Pranav Arora(2010990537)

Vishal Kumar(2010990786)

Saksham Kaushal (2010990625)

Shivam Soni (2010990665)

Paranav Mahajan(2010990514)

* **Guide:** Jyoti Snehi

**Contents**

* **Abstract………………………………………………………………………….**
* **Acknowledgments……………………………………………………………….**
* **Chapter 1: Introduction………………………………………………………...**
* **Chapter 2: Name of the Case Study………………………………………...….**

1. Case Study Informal Description………………………………………….
2. Terminologies and Symbols of ER diagram……………………………….
3. Case Study Logical Model (ER Diagram) ………………………………...
4. Case Study Physical Schema………………………………………………
5. Case Study Interactive Queries…………………………………………….

* **Conclusion……………………………………………………………………….**
* **Bibliography…………………………………………………………………….**

### Abstract

The objective of this thesis is to get some hands-on experience for creating the relational schemas and implementing the data extraction queries related to them. Our case study “Airport Management System” is presented. Input for this case study is taken from its informal specification to a relational schema using entity-relationship modeling and its translation to the relational model, to database schema, to implementation of the database, to interactive SQL querying of the installed database (Oracle).

### Acknowledgments

##### I would like to express my gratitude to all of those who made it possible to complete this thesis, in particular to my supervisor Jyoti Snehi. I would also like to thank my family for their understanding and continuous support.

# Chapter 1: Introduction

**Database Management Systems**

**Database** is a collection of data, in which we can retrieve, insert, delete data. It can also be organized into tables, schema, views, etc.

**Database Management Systems** is a software that is used to manage databases.

It provides us an interface that can be used to perform operations like database creation, creating tables, updating data, etc.

It helps us control data redundancy, and because it is centralized we can easily maintain the database. We can easily backup the data and restore it in case of a software or hardware failure.

**Relational Database Management System:**

**Relational Database Management System** is a collection of programs that help us to create, delete, update, administer and interact with the database. The data in a RDBMS is stored in the form of tables. And it is accessed using ***Structured Query Language.***

It is the most popular database system among companies. It also provides us with data dictionaries and metadata which is useful in data handling.They provide a systematic view of data which helps in better understanding and decision making process.

**ER Model**

**ER Model** stands for Entity Relationship Model, It displays the relationship of entity sets stored in a database.

It uses different symbols like rectangles to represent entities, ovals to define attributes and diamonds to represent relationships, etc.

It helps in creating the blueprint of the database. Also it gives a preview of how the tables should connect.

**Main components and symbols of ER Model**

** Rectangles:** This is used to represent the entity types.

**Ellipses:** It is used to represent the attributes.

**Diamond:** It is used to represent relationship types.

**Lines:** It links attributes to entity types and entity types with other relationship types.

**Double Ellipses:** It is used to represent multi-valued attributes.

**Brief introduction of case study**

We have chosen to produce an Airport management System as our case study. In our project we have depicted how airport management works.This provides us with details like airports, the Airline companies, details about passengers and employees. Passengers are eligible for a discount of 10% if they are frequent flyers or are employees in the Airport. Detailed description is discussed further.

**Objective of the case study**

We made this project to make the work easier for the managing team of the airport. With this they can easily access the details of flights, details of passengers travelling, status of flights, Also it can help them Monitor the flight schedule.

**Chapter 2: Airport Management System**

**An informal specification of the project**

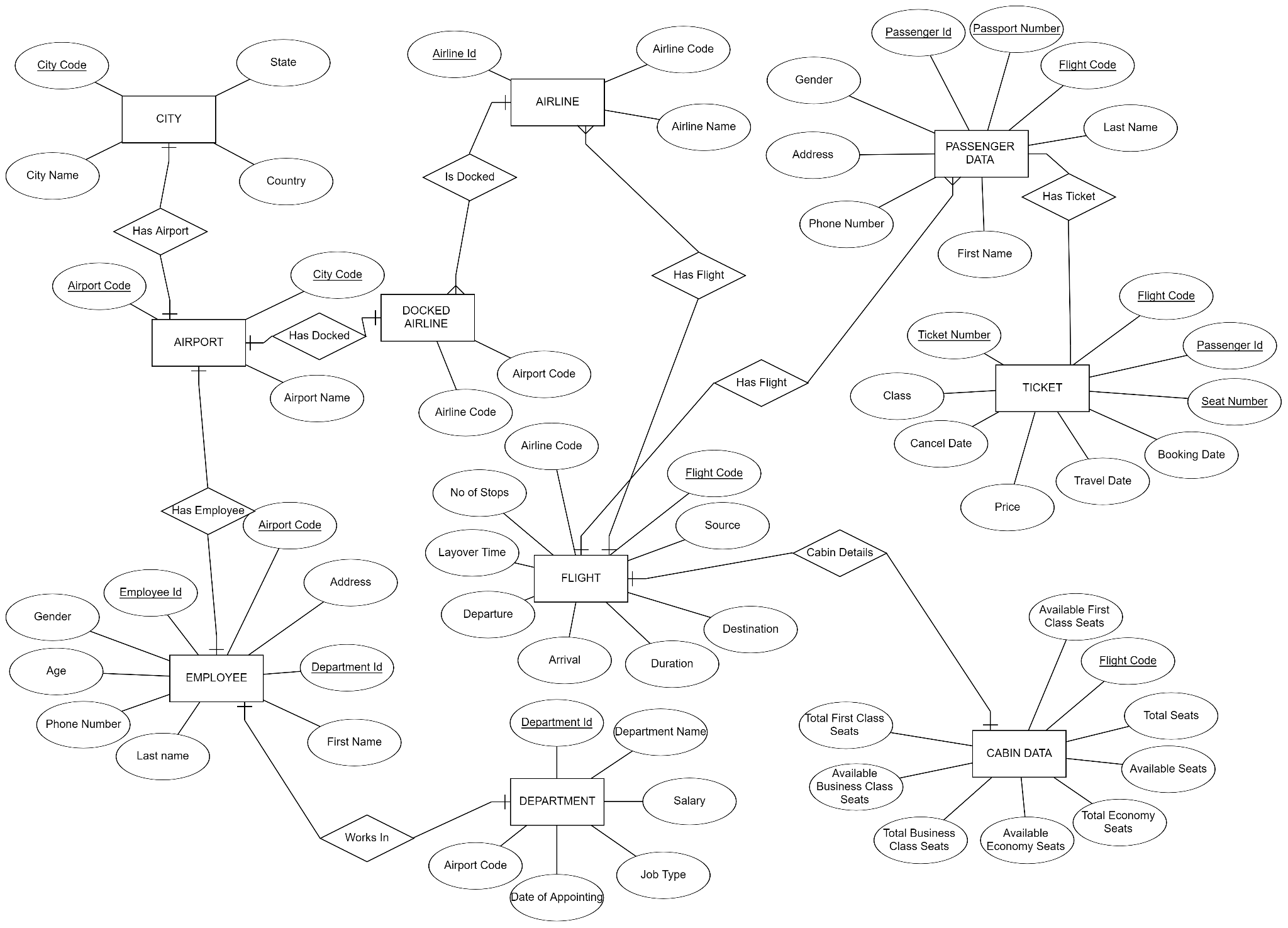
In this system a passenger can see how many seats are available in his particular flight, the flight status, and check if he is eligible for discounts.

If an employee travels, he/she is also eligible for employee discounts.

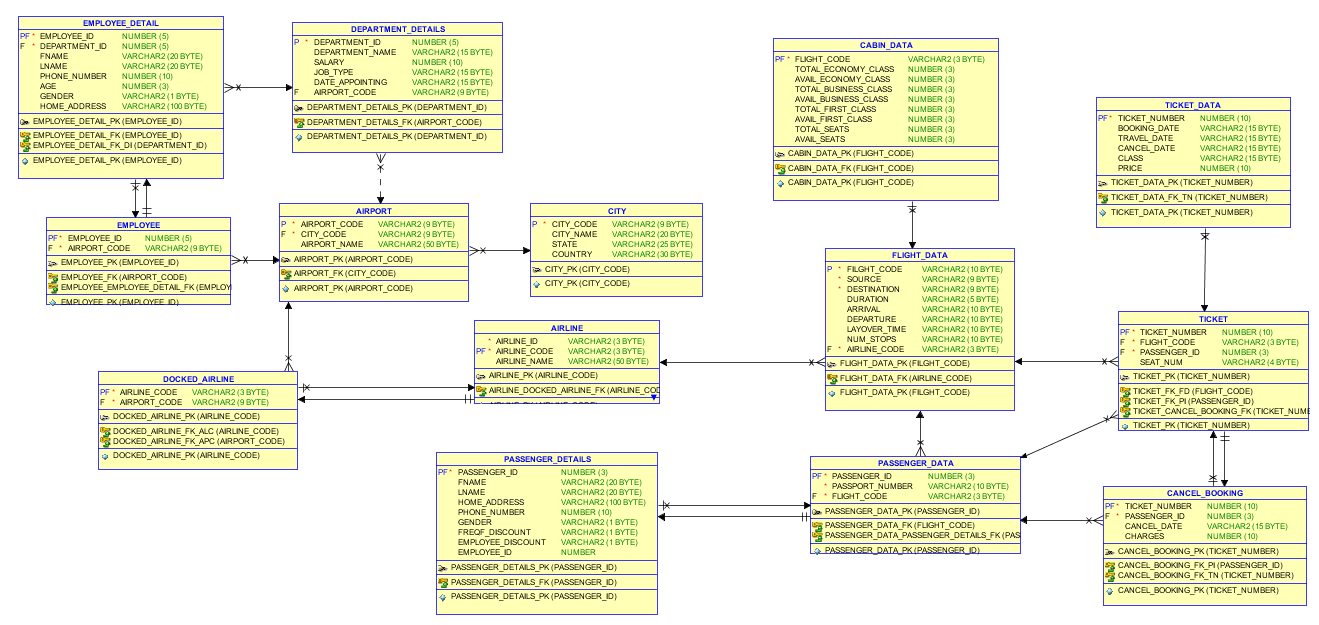
Passengers can also get discounts if they have flight points they can redeem those points.

# 

# 

**An Entity-Relationship Model** (Made using erdplus.com)

**Case study Logical Model (made using Oracle SQL Developer)**

****

**Database Schema**

CREATE table "CITY" (

"CITY\_CODE" *VARCHAR2*(9) NOT NULL,

"CITY\_NAME" *VARCHAR2*(20),

"STATE" *VARCHAR2*(25),

"COUNTRY" *VARCHAR2*(30),

constraint "CITY\_PK" primary key ("CITY\_CODE")

);

CREATE table "AIRPORT" (

"AIRPORT\_CODE" *VARCHAR2*(9) NOT NULL,

"CITY\_CODE" *VARCHAR2*(9) NOT NULL,

"AIRPORT\_NAME" *VARCHAR2*(50),

constraint "AIRPORT\_PK" primary key ("AIRPORT\_CODE")

);

ALTER TABLE "AIRPORT" ADD CONSTRAINT "AIRPORT\_FK"

FOREIGN KEY ("CITY\_CODE")

REFERENCES "CITY" ("CITY\_CODE")

ON DELETE CASCADE;

CREATE table "AIRLINE" (

"AIRLINE\_ID" *VARCHAR2*(3) NOT NULL,

"AIRLINE\_CODE" *VARCHAR2*(3) NOT NULL,

"AIRLINE\_NAME" *VARCHAR2*(50),

constraint "AIRLINE\_PK" primary key ("AIRLINE\_CODE")

);

CREATE table "DOCKED\_AIRLINE" (

"AIRLINE\_CODE" *VARCHAR2*(3) NOT NULL,

"AIRPORT\_CODE" *VARCHAR2*(9) NOT NULL,

constraint "DOCKED\_AIRLINE\_PK" primary key ("AIRLINE\_CODE")

);

ALTER TABLE "DOCKED\_AIRLINE" ADD CONSTRAINT "DOCKED\_AIRLINE\_FK\_APC"

FOREIGN KEY ("AIRPORT\_CODE")

REFERENCES "AIRPORT" ("AIRPORT\_CODE")

ON DELETE CASCADE;

ALTER TABLE "DOCKED\_AIRLINE" ADD CONSTRAINT "DOCKED\_AIRLINE\_FK\_ALC"

FOREIGN KEY ("AIRLINE\_CODE")

REFERENCES "AIRLINE" ("AIRLINE\_CODE")

ON DELETE CASCADE;

CREATE table "FLIGHT\_DATA" (

"FILGHT\_CODE" *VARCHAR2*(10) NOT NULL,

"SOURCE" *VARCHAR2*(9) NOT NULL,

"DESTINATION" *VARCHAR2*(9) NOT NULL,

"DURATION" *VARCHAR2*(5),

"ARRIVAL" *VARCHAR2*(10),

"DEPARTURE" *VARCHAR2*(10),

"LAYOVER\_TIME" *VARCHAR2*(10),

"NUM\_STOPS" *VARCHAR2*(10),

"AIRLINE\_CODE" *VARCHAR2*(3) NOT NULL,

constraint "FLIGHT\_DATA\_PK" primary key ("FILGHT\_CODE")

);

ALTER TABLE "FLIGHT\_DATA" ADD CONSTRAINT "FLIGHT\_DATA\_FK"

FOREIGN KEY ("AIRLINE\_CODE")

REFERENCES "AIRLINE" ("AIRLINE\_CODE")

ON DELETE CASCADE;

CREATE table "CABIN\_DATA" (

"FLIGHT\_CODE" *VARCHAR2*(10) NOT NULL,

"TOTAL\_ECONOMY\_CLASS" *NUMBER*(3),

"AVAIL\_ECONOMY\_CLASS" *NUMBER*(3),

"TOTAL\_BUSINESS\_CLASS" *NUMBER*(3),

"AVAIL\_BUSINESS\_CLASS" *NUMBER*(3),

"TOTAL\_FIRST\_CLASS" *NUMBER*(3),

"AVAIL\_FIRST\_CLASS" *NUMBER*(3),

"TOTAL\_SEATS" *NUMBER*(3),

"AVAIL\_SEATS" *NUMBER*(3),

constraint "CABIN\_DATA\_PK" primary key ("FLIGHT\_CODE")

);

ALTER TABLE "CABIN\_DATA" ADD CONSTRAINT "CABIN\_DATA\_FK"

FOREIGN KEY ("FLIGHT\_CODE")

REFERENCES "FLIGHT\_DATA" ("FILGHT\_CODE")

ON DELETE CASCADE;

CREATE table "PASSENGER\_DATA" (

"PASSENGER\_ID" *NUMBER*(3) NOT NULL,

"PASSPORT\_NUMBER" *VARCHAR2*(10) NOT NULL,

"FLIGHT\_CODE" *VARCHAR2*(10) NOT NULL,

constraint "PASSENGER\_DATA\_PK" primary key ("PASSENGER\_ID")

);

ALTER TABLE "PASSENGER\_DATA" ADD CONSTRAINT "PASSENGER\_DATA\_FK"

FOREIGN KEY ("FLIGHT\_CODE")

REFERENCES "FLIGHT\_DATA" ("FILGHT\_CODE")

ON DELETE CASCADE;

CREATE table "PASSENGER\_DETAILS" (

"PASSENGER\_ID" *NUMBER*(3) NOT NULL,

"FNAME" *VARCHAR2*(20),

"LNAME" *VARCHAR2*(20),

"HOME\_ADDRESS" *VARCHAR2*(100),

"PHONE\_NUMBER" *NUMBER*(10),

"GENDER" *VARCHAR2*(1),

"FREQF\_DISCOUNT" *VARCHAR2*(1),

"EMPLOYEE\_DISCOUNT" *VARCHAR2*(1),

"EMPLOYEE\_ID" *NUMBER*(10),

constraint "PASSENGER\_DETAILS\_PK" primary key ("PASSENGER\_ID")

);

ALTER TABLE "PASSENGER\_DETAILS" ADD CONSTRAINT "PASSENGER\_DETAILS\_FK"

FOREIGN KEY ("PASSENGER\_ID")

REFERENCES "PASSENGER\_DATA" ("PASSENGER\_ID")

ON DELETE CASCADE;

CREATE table "TICKET" (

"TICKET\_NUMBER" *NUMBER*(10) NOT NULL,

"FLIGHT\_CODE" *VARCHAR2*(10) NOT NULL,

"PASSENGER\_ID" *NUMBER*(3) NOT NULL,

"SEAT\_NUM" *VARCHAR2*(4),

constraint "TICKET\_PK" primary key ("TICKET\_NUMBER")

);

ALTER TABLE "TICKET" ADD CONSTRAINT "TICKET\_FK\_FD"

FOREIGN KEY ("FLIGHT\_CODE")

REFERENCES "FLIGHT\_DATA" ("FILGHT\_CODE")

ON DELETE CASCADE;

ALTER TABLE "TICKET" ADD CONSTRAINT "TICKET\_FK\_PI"

FOREIGN KEY ("PASSENGER\_ID")

REFERENCES "PASSENGER\_DATA" ("PASSENGER\_ID")

ON DELETE CASCADE;

CREATE table "TICKET\_DATA" (

"TICKET\_NUMBER" *NUMBER*(10) NOT NULL,

"BOOKING\_DATE" *VARCHAR2*(15),

"TRAVEL\_DATE" *VARCHAR2*(15),

"CANCEL\_DATE" *VARCHAR2*(15),

"CLASS" *VARCHAR2*(15),

"PRICE" *NUMBER*(10),

constraint "TICKET\_DATA\_PK" primary key ("TICKET\_NUMBER")

);

ALTER TABLE "TICKET\_DATA" ADD CONSTRAINT "TICKET\_DATA\_FK\_TN"

FOREIGN KEY ("TICKET\_NUMBER")

REFERENCES "TICKET" ("TICKET\_NUMBER")

ON DELETE CASCADE;

CREATE table "CANCEL\_BOOKING" (

"TICKET\_NUMBER" *NUMBER*(10) NOT NULL,

"PASSENGER\_ID" *NUMBER*(3) NOT NULL,

"CANCEL\_DATE" *VARCHAR2*(15),

"CHARGES" *NUMBER*(10),

constraint "CANCEL\_BOOKING\_PK" primary key ("TICKET\_NUMBER")

);

ALTER TABLE "CANCEL\_BOOKING" ADD CONSTRAINT "CANCEL\_BOOKING\_FK\_TN"

FOREIGN KEY ("TICKET\_NUMBER")

REFERENCES "TICKET" ("TICKET\_NUMBER")

ON DELETE CASCADE;

ALTER TABLE "CANCEL\_BOOKING" ADD CONSTRAINT "CANCEL\_BOOKING\_FK\_PI"

FOREIGN KEY ("PASSENGER\_ID")

REFERENCES "PASSENGER\_DATA" ("PASSENGER\_ID")

ON DELETE CASCADE;

CREATE table "EMPLOYEE" (

"EMPLOYEE\_ID" *NUMBER*(5) NOT NULL,

"AIRPORT\_CODE" *VARCHAR2*(9) NOT NULL,

constraint "EMPLOYEE\_PK" primary key ("EMPLOYEE\_ID")

);

ALTER TABLE "EMPLOYEE" ADD CONSTRAINT "EMPLOYEE\_FK"

FOREIGN KEY ("AIRPORT\_CODE")

REFERENCES "AIRPORT" ("AIRPORT\_CODE")

ON DELETE CASCADE;

CREATE table "EMPLOYEE\_DETAIL" (

"EMPLOYEE\_ID" *NUMBER*(5) NOT NULL,

"DEPARTMENT\_ID" *NUMBER*(5) NOT NULL,

"FNAME" *VARCHAR2*(20),

"LNAME" *VARCHAR2*(20),

"PHONE\_NUMBER" *NUMBER*(10),

"AGE" *NUMBER*(3),

"GENDER" *VARCHAR2*(1),

"HOME\_ADDRESS" *VARCHAR2*(100),

constraint "EMPLOYEE\_DETAIL\_PK" primary key ("EMPLOYEE\_ID")

);

ALTER TABLE "EMPLOYEE\_DETAIL" ADD CONSTRAINT "EMPLOYEE\_DETAIL\_FK"

FOREIGN KEY ("EMPLOYEE\_ID")

REFERENCES "EMPLOYEE" ("EMPLOYEE\_ID")

ON DELETE CASCADE;

ALTER TABLE "EMPLOYEE\_DETAIL" ADD CONSTRAINT "EMPLOYEE\_DETAIL\_FK\_DI"

FOREIGN KEY ("DEPARTMENT\_ID")

REFERENCES "DEPARTMENT\_DETAILS" ("DEPARTMENT\_ID")

ON DELETE CASCADE;

CREATE table "DEPARTMENT\_DETAILS" (

"DEPARTMENT\_ID" *NUMBER*(5) NOT NULL,

"DEPARTMENT\_NAME" *VARCHAR2*(15),

"SALARY" *NUMBER*(10),

"JOB\_TYPE" *VARCHAR2*(15),

"DATE\_APPOINTING" *VARCHAR2*(15),

"AIRPORT\_CODE" *VARCHAR2*(9),

constraint "DEPARTMENT\_DETAILS\_PK" primary key ("DEPARTMENT\_ID")

);

ALTER TABLE "DEPARTMENT\_DETAILS" ADD CONSTRAINT "DEPARTMENT\_DETAILS\_FK"

FOREIGN KEY ("AIRPORT\_CODE")

REFERENCES "AIRPORT" ("AIRPORT\_CODE")

ON DELETE CASCADE;

**Sequences**

**EMPLOYEE ID SEQUENCE**

create sequence "EMPLOYEE\_ID\_SEQ"

start with 0

increment by 1

maxvalue 99999

minvalue 0

nocache

nocycle

noorder;

**TICKET NUMBER SEQUENCE**

create sequence "TICKET\_NUMBER\_SEQ"

start with 123

increment by 1

maxvalue 99999

minvalue 0

nocache

nocycle

noorder;

**PASSENGER ID SEQUENCE**

create sequence "PASSENGER\_ID\_SEQ"

start with 1

increment by 1

maxvalue 999

minvalue 0

nocache

nocycle

order;

**Procedures**

**NEW\_PASSENGER\_PRO**

create or replace procedure "NEW\_PASSENGER\_PRO"

(passport\_number IN *VARCHAR2*, flight\_code IN *VARCHAR2*,

firstname IN *VARCHAR2*, lastname IN *VARCHAR2*,

homeaddress IN *VARCHAR2*, phonenumber IN *NUMBER*,

p\_gender IN *VARCHAR2*, p\_freqfdiscount IN *VARCHAR2*,

p\_employeediscount IN *VARCHAR2*, p\_employeeid IN *VARCHAR2*,

p\_class IN *VARCHAR2*, p\_seat IN *VARCHAR2*,

p\_price IN *NUMBER*)

is

BEGIN

INSERT INTO PASSENGER\_DATA ("PASSENGER\_ID","PASSPORT\_NUMBER","FLIGHT\_CODE") VALUES

(PASSENGER\_ID\_SEQ.nextval, passport\_number,flight\_code);

COMMIT;

INSERT INTO PASSENGER\_DETAILS ("PASSENGER\_ID","FNAME","LNAME","HOME\_ADDRESS",

"PHONE\_NUMBER","GENDER","FREQF\_DISCOUNT","EMPLOYEE\_DISCOUNT","EMPLOYEE\_ID") VALUES

(PASSENGER\_ID\_SEQ.currval, firstname, lastname, homeaddress, phonenumber, P\_gender, p\_freqfdiscount, p\_employeediscount, p\_employeeid);

COMMIT;

INSERT INTO TICKET ("TICKET\_NUMBER", "FLIGHT\_CODE", "PASSENGER\_ID", "SEAT\_NUM") VALUES

(TICKET\_NUMBER\_SEQ.nextval, flight\_code, PASSENGER\_ID\_SEQ.currval, p\_seat);

COMMIT;

INSERT INTO TICKET\_DATA ("TICKET\_NUMBER","BOOKING\_DATE","TRAVEL\_DATE","CANCEL\_DATE","CLASS","PRICE") VALUES

(TICKET\_NUMBER\_SEQ.currval, *sysdate*, *sysdate*+7, '', p\_class, p\_price);

COMMIT;

END;

/

**NEW\_EMPLOYEE\_PRO**

create or replace procedure "NEW\_EMPLOYEE\_PRO"

(airport\_code IN *VARCHAR2*,department\_id IN *NUMBER*,

p\_fname IN *VARCHAR2*,p\_lname IN *VARCHAR2*,

p\_phone\_num IN *NUMBER*,p\_age IN *NUMBER*,

p\_gender IN *VARCHAR2*,p\_home\_address IN *VARCHAR2*)

is

BEGIN

INSERT INTO EMPLOYEE ("EMPLOYEE\_ID","AIRPORT\_CODE") VALUES

(EMPLOYEE\_ID\_SEQ.nextval, airport\_code);

COMMIT;

INSERT INTO EMPLOYEE\_DETAILS ("EMPLOYEE\_ID","DEPARTMENT\_ID","FNAME",

"LNAME","PHONE\_NUMBER","AGE","GENDER","HOME\_ADDRESS")

VALUES

(EMPLOYEE\_ID\_SEQ.currval, department\_id,p\_fname,p\_lname,p\_phone\_num,p\_age,p\_gender,p\_home\_address);

COMMIT;

END;

/;

create or replace procedure "UPDATE\_SEATS\_PRO"

(p\_flight\_code IN *VARCHAR2*,

p\_class IN *VARCHAR2*,p\_seat\_num IN *VARCHAR2*)

is

begin

DECLARE

T\_AVAIL\_SEATS *number*; AVAIL\_ECO\_SEATS *number*; AVAIL\_BUS\_SEATS *number*;

AVAIL\_FIR\_SEATS *number*;

FL\_CLASS VARCHAR(30);

BEGIN

SELECT AVAIL\_SEATS INTO T\_AVAIL\_SEATS FROM CABIN\_DATA WHERE FLIGHT\_CODE = p\_flight\_code;

SELECT AVAIL\_ECONOMY\_CLASS INTO AVAIL\_ECO\_SEATS FROM CABIN\_DATA WHERE FLIGHT\_CODE = p\_flight\_code;

SELECT AVAIL\_BUSINESS\_CLASS INTO AVAIL\_BUS\_SEATS FROM CABIN\_DATA WHERE FLIGHT\_CODE = p\_flight\_code;

SELECT AVAIL\_FIRST\_CLASS INTO AVAIL\_FIR\_SEATS FROM CABIN\_DATA WHERE FLIGHT\_CODE = p\_flight\_code;

IF T\_AVAIL\_SEATS > 0 THEN

dbms\_output.put\_line('ALL SEATS ARE FULL IN FLIGHT');

ELSE

IF p\_class = 'economy' OR p\_class = 'ECONOMY' OR p\_class = 'ECO' THEN

FL\_CLASS:='AVAIL\_ECONOMY\_CLASS';

IF AVAIL\_ECO\_SEATS <= 0 THEN

dbms\_output.put\_line('ALL SEATS ARE FULL IN ECONOMY CLASS');

ELSE

UPDATE CABIN\_DATA SET AVAIL\_SEATS=AVAIL\_SEATS-1,

AVAIL\_ECONOMY\_CLASS=AVAIL\_ECONOMY\_CLASS-1

WHERE FLIGHT\_CODE=p\_flight\_code;

COMMIT;

END IF;

ELSIF p\_class = 'BUSINESS' OR p\_class = 'business' OR p\_class = 'busi' THEN

FL\_CLASS:='AVAIL\_BUSINESS\_CLASS';

FL\_CLASS:='AVAIL\_ECONOMY\_CLASS';

IF AVAIL\_ECO\_SEATS <= 0 THEN

dbms\_output.put\_line('ALL SEATS ARE FULL IN BUSINESS CLASS');

ELSE

UPDATE CABIN\_DATA SET AVAIL\_SEATS=AVAIL\_SEATS-1,

AVAIL\_BUSINESS\_CLASS=AVAIL\_BUSINESS\_CLASS-1

WHERE FLIGHT\_CODE=p\_flight\_code;

COMMIT;

END IF;

ELSE

FL\_CLASS:='AVAIL\_FIRST\_CLASS';

IF AVAIL\_ECO\_SEATS <= 0 THEN

dbms\_output.put\_line('ALL SEATS ARE FULL IN BUSINESS CLASS');

ELSE

UPDATE CABIN\_DATA SET AVAIL\_SEATS=AVAIL\_SEATS-1,

AVAIL\_FIRST\_CLASS=AVAIL\_FIRST\_CLASS-1

WHERE FLIGHT\_CODE=p\_flight\_code;

COMMIT;

END IF;

END IF;

END IF;

END;

END;

**Case Study Interactive Queries**

INSERT INTO CITY ("CITY\_CODE","CITY\_NAME","STATE","COUNTRY") VALUES('LKUS','Louisville','Kentucky','United States');

INSERT INTO CITY ("CITY\_CODE","CITY\_NAME","STATE","COUNTRY") VALUES('CCI','Chandigarh','Chandigarh','India');

INSERT INTO AIRPORT ("AIRPORT\_NAME","AIRPORT\_CODE","CITY\_CODE") VALUES('Louisville International Airport','SDF','LKUS');

INSERT INTO AIRPORT ("AIRPORT\_NAME","AIRPORT\_CODE","CITY\_CODE") VALUES('Chandigarh International Airport','IXC','CCI');

INSERT INTO AIRLINE ("AIRLINE\_ID","AIRLINE\_NAME","AIRLINE\_CODE") VALUES('AI','Air India Limited','098');

INSERT INTO AIRLINE ("AIRLINE\_ID","AIRLINE\_NAME","AIRLINE\_CODE") VALUES('LH','Lufthansa', '220');

INSERT INTO DOCKED\_AIRLINE ("AIRLINE\_CODE","AIRPORT\_CODE") VALUES('001','SDF');

INSERT INTO DOCKED\_AIRLINE ("AIRLINE\_CODE","AIRPORT\_CODE") VALUES('001','JFK');

INSERT INTO FLIGHT\_DATA("FLIGHT\_CODE","SOURCE","DESTINATION","DURATION","ARRIVAL","DEPARTURE","LAYOVER\_TIME","NUM\_STOPS","AIRLINE\_CODE") VALUES('AI2014','BOM','DFW','24hr','02:10','03:15' ,'3', '1','098');

INSERT INTO FLIGHT\_DATA("FLIGHT\_CODE","SOURCE","DESTINATION","DURATION","ARRIVAL","DEPARTURE","LAYOVER\_TIME","NUM\_STOPS","AIRLINE\_CODE") VALUES('EY1234','JFK','TPA','3hrs','19:20','20:05' ,'5', '2','607');

**Insertion of Passenger Related Data using procedure**

BEGIN

NEW\_PASSENGER\_PRO('A1234568','AI2014','Pranav','Arora', '123-sector-28-panchkula',9877012312,'M','N','N','','FIRST-CLASS','B021',12345);

END;

**Insertion of Employee Related Data using procedure**

BEGIN

NEW\_EMPLOYEE\_PRO('SDF','125','carl','lewis',0123456795,26,'M','7820 MCCALLUM COURTS, APT 234, AKRON, OH');

END;

**Conclusion**

While working on this project, we learnt a lot about creating a database and implementing all the queries related to creation of the database, modifying it and fetching the data from it. We have a good hold on the concepts related to ER diagrams and normalization now.

We also learned about procedures, sequences by this project and triggers.

**Bibliography**

* <https://www.javatpoint.com/dbms-functional-dependency>
* <https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_6009.htm>
* <https://docs.oracle.com/cd/B12037_01/server.101/b10759/statements_6014.htm>
* <https://www.techonthenet.com/oracle/sequences.php>
* <https://www.techonthenet.com/oracle/procedures.php>
* <https://www.javatpoint.com/dbms-er-model-concept>
* <https://www.guru99.com/er-diagram-tutorial-dbms.html>
* <https://www.techonthenet.com/oracle/tables/alter_table.php>