Database Management Project

Second Semester 2021-22

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Submitted by:

Agrim Jain - 2020A7PS0143U

Arshdeep Singh - 2020A7PS0144U

Jay Parida - 2020A7PS0087U

Harsh Vegad - 2020A7PS0152U

Nidhi Widhani - 2020A7PS0239U

TABLE OF CONTENTS

Acknowledgement

I would like to express my sincere gratitude to our Database Management Instructor, Dr. Pramod Gaur, for his vital support, guidance and encouragement without which this project would not come forth from our side. He helped us complete the project by giving ideas, thoughts and making this project easy and accurate.

I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go our own way, without which we would be unable to complete our project.

INTRODUCTION

Airport Management System

Airports are one of the busiest places, where people travel to and from countries thus there must be an airport management system. Everyone including the airplane employees, passengers, agents, and staff benefit from this management system. Since thousands of people travel in a day, it is important for airports to meet the passenger demands and make a constant, up to data management system that displays waiting times, tickets, and other details to the passengers. The managers who are in charge of the airport activity make sure that the system runs in a productive and smooth manner.

The Airline industry is booming with over 150 million customers. One of the most crucial elements which the Airline industry depends upon is the Database Management System for Airports. 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. The Database system and SQL commands described in this report makes up the basic operations and functions of an airport

We have made an airport database management system which includes the name and description of the airlines that depart from and arrive at an airport. There is sufficient information such as the name, state, city, and code of the airport. Every airplane can only be accessed through the purchase of a ticket. A ticket can be identified through the ID and it is booked by an agent. The ticket is given by the agent to the customer only when the agent receives payment from the customer. The agent can be identified by their ID and the payment by the transaction ID
After that, we have written 5 queries which include nested, correlated, functions, and triggers. Furthermore, we have constructed the ER Diagram and then converted it into a relational database. Following that, we normalized the relational model by creating functional dependencies.

ER DIAGRAM

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

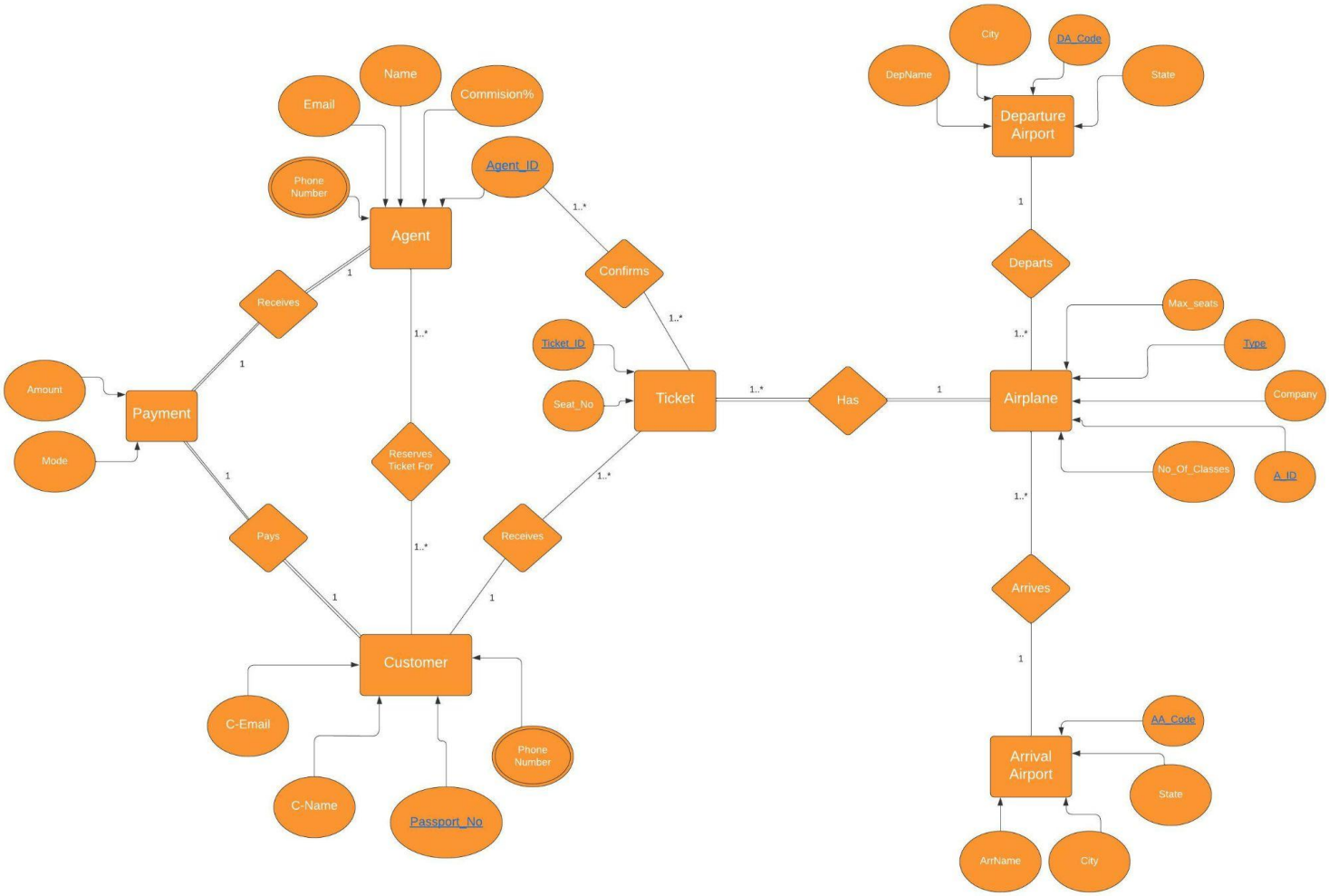
ER Diagram, also known as Entity-Relationship Model is a flowchart which makes a system look organized. In the system there are entities and the relationship between them.

There are different shapes that are used for different purposes. Rectangle is used for an entity, double rectangle for weak entity, diamond for relationship, double diamond for a weak relationship, an oval is used for attribute, double oval is used for multi-valued attribute.

1. Let's start with the entity airplane because it is accessed by all entities. Each airplane has a unique type and airplane ID(A_ID). An airplane is of a company and includes maximum number of seats, and the number of classes(Economy, Business, First Class).
2. The departure airport has a name and has a unique departure airport code (DA_Code). An airplane departs from a specific city and state to another city and state. One departure airport departs many airplanes. In other words, many airplanes depart from one departure airport.
3. Similarly, the arrival airport has a name and a unique arrival airport code (AA_Code). An airplane arrives to a specific city and state from another city and state. Many airplanes arrive at an arrival airport.
4. An airplane has tickets, which include a unique Ticket_ID and a Seat_No.
5. There is an agent who has an unique Agent_ID. The agent has a name, email, phone-number(multi-valued attribute) and a commission percentage. Commission percentages are fixed payments that the agent receives through the buying of a ticket from a customer. One agent can confirm many tickets.
6. A customer has a unique passport number, a phone-number (multi-valued attribute), name, and an email address. One customer can receive many tickets. One agent can reserve tickets for many agents. An agent receives payment from one customer. There is a mode of payment, it can either be through a debit card, credit card, cash, or through paytm. One customer pays the payment to one agent.

Airline Management System - Agrim, Harsh, Jay, Arshdeep, Nidhi

Agrim Jain - 2/5/2022



Schemas:
In a relational database schema, we need the columns, tables, and the relationships between the entities. The relational database schema is divided into two, the logical and physical schema. Logical schema is used for the structure wherein it depicts the columns, tables, and relationships of tables in the database. Whereas, the physical schema is when “actually” the columns and tables have been created, relationships have been established between tables. The physical schema is done in the relational database management software where the sql queries are performed in order to make tables and relationships.

Airplane(A_ID, Type, Max_Seats, Company, No_Of_Classes)
Departure_Airport(DA_Code, DepName, City, State)
Arrival_Airport(AA_Code, ArrName, City, State)
Ticket(Ticket_ID, Seat_No)
Agent(Agent_ID, Phone-Number, Email, Name, Commission%)
Customer(Pass_No, Phone-Number, C-Email, C-Name)

Cardinal Relationships:
Cardinalities are numerical attributes between two entity sets. There are 4 types of cardinal relationships: one-to-one(1..1), one-to-many(1..*), and many-to-many(M..N).

One-to-many:
Agent - reserves ticket for - Customer
Customer - receives - ticket
Agent - confirms - ticket
Airplane - has - ticket
Departure Airport - departs - airplanes
Arrival Airport - arrives - airplanes

One-to-One:
Customer - Pays - Payment
Agent - Receives - Payment

TABLES

1. Departure_Airport - DA_Code

```
CREATE TABLE DEPARTURE_AIRPORT
(
        DA_CODE VARCHAR(5),
        DNAME VARCHAR(50),
        STATE VARCHAR(20),
        CITY VARCHAR(20),
        PRIMARY KEY(DA_CODE)
);
```

Populating Depature_Airport:

```
INSERT INTO DEPARTURE_AIRPORT VALUES('HYD', 'RAJIV GANDHI
INTERNATIONAL AIRPORT', 'TELANGANA', 'HYDERABAD');
INSERT INTO DEPARTURE_AIRPORT VALUES('DEL', 'INDIRA GANDHI
INTERNATIONAL AIRPORT', 'DELHI', 'DELHI');
INSERT INTO DEPARTURE_AIRPORT VALUES('BOM', 'CHHATRAPATI SHIVAJI
INTERNATIONAL AIRPORT', 'MAHARASHTRA', 'MUMBAI');
INSERT INTO DEPARTURE_AIRPORT VALUES('IXC', 'CHANDIGARH
INTERNATIONAL AIRPORT', 'PUNJAB', 'CHANDIGARH');
INSERT INTO DEPARTURE_AIRPORT VALUES('IDR', 'DEVI AHILYABHAI HOLKAR
AIRPORT', 'MADHYA PRADESH', 'INDORE');
```

DA_CODE	DNAME	STATE	CITY
BOM	CHHATRAPATI SHIVAJI INTERNATIONAL AIRP...	MAHARASHTRA	MUMBAI
DEL	INDIRA GANDHI INTERNATIONAL AIRPORT	DELHI	DELHI
HYD	RAJIV GANDHI INTERNATIONAL AIRPORT	TELANGANA	HYDERABAD
IDR	DEVI AHILYABHAI HOLKAR AIRPORT	MADHYA PRADESH	INDORE
IXC	CHANDIGARH INTERNATIONAL AIRPORT	PUNJAB	CHANDIGARH

2. Arrival_Airport(AA) - AA_Code

```
CREATE TABLE ARRIVAL_AIRPORT
(
   AA_CODE VARCHAR(5),
   ANAME VARCHAR(50),
   STATE VARCHAR(20),
   CITY VARCHAR(20),
   PRIMARY KEY(AA_CODE)
);
```

Populating Arrival_Airport

```
INSERT INTO ARRIVAL_AIRPORT VALUES('AIA', 'ATHENS INTERNATIONAL AIRPORT',
'GREECE', 'ATHENS');
INSERT INTO ARRIVAL_AIRPORT VALUES('DXB', 'DUBAI INTERNATIONAL AIRPORT',
'UAE', 'DUBAI');
INSERT INTO ARRIVAL_AIRPORT VALUES('MRU', 'SIR SEEWOOSAGAR RAMGOOLAM
INTERNATIONAL AIRPORT', 'MAURITIUS', 'PLAINE MANGIEN');
INSERT INTO ARRIVAL_AIRPORT VALUES('BKK', 'SURVANABHUMI AIRPORT', 'THAILAND',
'BANGKOK');
INSERT INTO ARRIVAL_AIRPORT VALUES('LHR', 'HEATHROW INTERNATIONAL AIRPORT',
'UNITED KINGDOM', 'LONDON');
```

AA_CODE	ANAME	STATE	CITY
AIA	ATHENS INTERNATIONAL AIRPORT	GREECE	ATHENS
BKK	SURVANABHUMI AIRPORT	THAILAND	BANGKOK
DXB	DUBAI INTERNATIONAL AIRPORT	UAE	DUBAI
LHR	HEATHROW INTERNATIONAL AIRPORT	UNITED KINGDOM	LONDON
MRU	SIR SEEWOOSAGAR RAMGOOLAM INTERNATIONAL AIRPORT	MAURITIUS	PLAINE MANGIEN

3. Agent - <u>Agent_ID</u>

```
CREATE TABLE AGENT
(
        AGENT_ID INT,
   AGENT_NAME VARCHAR(20),
   AGENT_MAIL VARCHAR(40),
   AGENT_PHONE VARCHAR(15),
   COMMISSION VARCHAR(5),
   PRIMARY KEY(AGENT_ID)
);
```

Populating Agent:

```
INSERT INTO AGENT VALUES(200, 'CHAMPAGNE PAPI', 'DRAKE@OVO.COM', '+1
8321456842', '13%');
INSERT INTO AGENT VALUES(201, 'JACK HARLOW', 'JACKH@FIRSTCLASS.COM',
'+1 9862317261', '18%');
INSERT INTO AGENT VALUES(202, 'DUA LIPA', 'LEVITATING@GMAIL.COM', '+1
9876543210', '14%');
INSERT INTO AGENT VALUES(203, 'LEBRON JAMES', 'KINGJAMES@GMAIL.COM',
'+1 7468215260', '20%');
INSERT INTO AGENT VALUES(204, 'VIRAT KOHLI', 'VK18I@GMAIL.COM', '+91
8146800561', '10%');
```

AGENT_ID	AGENT_NAME	AGENT_MAIL	AGENT_PHONE	COMMISSION
200	CHAMPAGNE PAPI	DRAKE@OVO.COM	+1 8321456842	13%
201	JACK HARLOW	JACKH@FIRSTCLASS.COM	+1 9862317261	18%
202	DUA LIPA	LEVITATING@GMAIL.COM	+1 9876543210	14%
203	LEBRON JAMES	KINGJAMES@GMAIL.COM	+1 7468215260	20%
204	VIRAT KOHLI	VK18I@GMAIL.COM	+91 8146800561	10%

NORMALIZATION

A_ID	A_Type	Max_Seats	Class_No	Company	DA_Code	AA_Code

In the entity Airplane, A_ID and A_Type are primary keys. To bring the table into 2NF we need to split the table into Airplane and Airplane(2NF).

4. **Airplane - A_ID and A_Type** can give all the attributes.

   Airplane(A_ID, A_Type, Max_Seats, Class_No, Company, DA_Code, AA_Code)

   The following is the Functional Dependency for Airplane:

   {A_ID, A_Type} → {Max_Seats, Class_No, Company, DA_Code, AA_Code)

So we can conclude that the candidate key that is present in Airplane is {A_ID, A_Type) as it gives closure to all the attributes of the Airplane entity.

a) It is in 1NF(1st normal form) because there are no multi-valued attributes and only atomic values are present in the entity.

b) There are no partial dependencies as {A_ID, A_Type} are the candidate key, but there is transitive dependency, so Airplane is not in 2NF.

To convert to 2NF, we need to split the table into two tables, Airplane and Airplane 2NF(2nd Normal Form).

In the Airplane_2NF table we notice transitivity in the functional dependencies and thus we convert the table into 3NF(3rd Normal Form)

To conclude, we will need to decompose the Airplane table into 3NF and to do so we will need to split the tables into Airplane and Airplane_2NF.

```
CREATE TABLE AIRPLANE
(
   A_ID INT,
   A_TYPE VARCHAR(20),
   COMPANY VARCHAR(20),
   DA_CODE VARCHAR(5),
   AA_CODE VARCHAR(5),
   PRIMARY KEY(A_ID, A_TYPE)
);
```

Populating Airplane:

```
INSERT INTO AIRPLANE VALUES(1, 'BOEING 787-8', 'AIR INDIA', 'IDR', 'AIA');
INSERT INTO AIRPLANE VALUES(2, 'BOEING 777', 'EMIRATES', 'BOM', 'DXB');
INSERT INTO AIRPLANE VALUES(3, 'AIRBUS A350', 'BRITISH AIRWAYS', 'DEL',
'LHR');
INSERT INTO AIRPLANE VALUES(1, 'AIRBUS A350-900', 'AIR MAURITIUS', 'HYD',
'MRU');
INSERT INTO AIRPLANE VALUES(2, 'AIRBUS A320', 'INDIGO', 'IXC', 'BKK');
```

A_ID	A_TYPE	COMPANY	DA_CODE	AA_CODE
1	AIRBUS A350-900	AIR MAURITIUS	HYD	MRU
1	BOEING 787-8	AIR INDIA	IDR	AIA
2	AIRBUS A320	INDIGO	IXC	BKK
2	BOEING 777	EMIRATES	BOM	DXB
3	AIRBUS A350	BRITISH AIRWAYS	DEL	LHR

Airplane(2NF):

```
CREATE TABLE AIRPLANE_2NF
(
   MAX_SEATS_2NF INT,
   A_TYPE_2NF VARCHAR(20),
   CLASS_NO INT,
   PRIMARY KEY(A_TYPE_2NF)
);
```

Populating Airplane(2NF):

```
INSERT INTO AIRPLANE_2NF VALUES(256, 'BOEING 787-8',3);
INSERT INTO AIRPLANE_2NF VALUES(364, 'BOEING 777',1);
INSERT INTO AIRPLANE_2NF VALUES(150, 'AIRBUS A320',2);
```

MAX_SEATS_2NF	A_TYPE_2NF	CLASS_NO
150	AIRBUS A320	2
364	BOEING 777	1
256	BOEING 787-8	3

Since there's transitive dependency in Airplane(2NF), we split it further into Airplane(2NF) and Airplane(3NF).

Airplane 2NF:

MAX_SEATS_2NF	A_TYPE_2NF
150	AIRBUS A320
364	BOEING 777
256	BOEING 787-8

Airplane 3NF:

CLASS_NO	MAX_SEATS
2	150
1	364
3	400

5. **Customer - <u>Pass_No</u>** can give all the attributes.

   Customer(<u>Pass_No,</u> C-number, C-Email, C-Name, Agent_ID)

   The following is the Functional Dependency for Customer:

   Pass_No → {C-number, C-Email, C-Name, Agent_ID)
   So we can conclude that the candidate key that is present in Customer is Pass_No as it gives closure to all the attributes of the Customer entity.

   a) It is not in 1NF because there are multi-valued attributes such as phone-number

   b) Functional dependency of Customer is complete dependency as Pass_No is the candidate key. Passport_Number and C-num are not a part of the candidate key, so they are in 2NF.

c) Since the functional dependency of customer is transitive dependency as Passport_Number and C_num are non-prime to non-prime attributes. These conditions don't meet the conditions of 3NF, thus they are not in 3NF.

So, concluding we will need to decompose the Customer table into 3NF and to do so we will need to split the tables into Customer and Customer_Number.

Customer(<u>Pass_No,</u> C-number, C-Email, C_Name, Agent_ID)
Customer_Number(<u>Passport_Number,</u> C-num)

Customer:

```
CREATE TABLE CUSTOMER
(
   C_NUMBER VARCHAR(20),
   PASS_NO VARCHAR(10),
   C_EMAIL VARCHAR(30),
   C_NAME VARCHAR(20),
   AGENT_ID INT,
   PRIMARY KEY(C_NUMBER)
);
```

Populating Customer:

```
INSERT INTO CUSTOMER VALUES('971544432342','M1345267', 'AJ@GMAIL.COM', 'AGRIM
JAIN',200);
INSERT INTO CUSTOMER VALUES('971434543345','A50123321', 'PAJJI@HOTMAIL.COM',
'ARSHDEEP SINGH',201);
INSERT INTO CUSTOMER VALUES('971523423234','J6124361', 'SHIJU@SORTMAIL.COM',
'HARSH VEGAD',202);
INSERT INTO CUSTOMER VALUES('971545834545','P2368145', 'JP@GMAIL.COM', 'JAY
PARIDA',203);
INSERT INTO CUSTOMER VALUES('971234232345','N11366775', 'NIDHIBOT@GMAIL.COM',
'NIDHI WIDHANI',204);
```

C_NUMBER	PASS_NO	C_EMAIL	C_NAME	AGENT_ID
971544432342	M1345267	AJ@GMAIL.COM	AGRIM JAIN	200
971434543345	A50123321	PAJJI@GMAIL.COM	ARSHDEEP SINGH	201
971523423234	J6124361	SHIJU@GMAIL.COM	HARSH VEGAD	202
971545834545	P2368145	JP@GMAIL.COM	JAY PARIDA	203
971234232345	N11366775	NEEDY@GMAIL.COM	NIDHI WIDHANI	204

Customer-Number: Pass_No

```
CREATE TABLE CUSTOMER_NUMBER
(
   PASS_NO VARCHAR(20),
   C_NUMBER VARCHAR(20),
   PRIMARY KEY(PASS_NO)
);
```

Populating Customer_Number:

```
INSERT INTO CUSTOMER_NUMBER VALUES('U24567867', '+971 534220842');
INSERT INTO CUSTOMER_NUMBER VALUES('Z34563359', '+91 8290323400');
INSERT INTO CUSTOMER_NUMBER VALUES('J23489203', '+91 8304580330');
INSERT INTO CUSTOMER_NUMBER VALUES('K19239202', '+971 523507900');
INSERT INTO CUSTOMER_NUMBER VALUES('A12910323', '+91 9811232449');
INSERT INTO CUSTOMER_NUMBER VALUES('Z12013092', '+971 546685849');
INSERT INTO CUSTOMER_NUMBER VALUES('W52392307', '+971 530842393');
```

PASS_NO	C_NUMBER
A12910323	+91 9811232449
J23489203	+91 8304580330
K19239202	+971 523507900
U24567867	+971 534220842
W52392307	+971 530842393
Z12013092	+971 546685849
Z34563359	+91 8290323400

6. Ticket - <u>TICKET_ID</u>

```
CREATE TABLE TICKET
(
   TICKET_ID INT,
   SEAT_NO VARCHAR(5),
   AGENT_ID INT,
   PASS_NO VARCHAR(10),
   A_ID INT,
   A_TYPE VARCHAR(20),
   PRIMARY KEY(TICKET_ID)
);
```

Populating Ticket:

```
INSERT INTO TICKET VALUES(100, '2A', 100, 'Z12013092', 2, 'AIRBUS A320');
INSERT INTO TICKET VALUES(105, '15C', 101, 'A12910323', 1, 'AIRBUS A350');
INSERT INTO TICKET VALUES(128, '5E', 102, 'K19239202', 3, 'BOEING 787-8');
INSERT INTO TICKET VALUES(156, '40B', 103, 'U24567867', 1, 'BOEING 787-8');
INSERT INTO TICKET VALUES(191, '37D', 104, 'J23489203', 2, 'BOEING 777');
```

TICKET_ID	SEAT_NO	AGENT_ID	PASS_NO	A_ID	A_TYPE
100	2A	100	Z12013092	2	AIRBUS A320
105	15C	101	A12910323	1	AIRBUS A350
128	5E	102	K19239202	3	BOEING 787-8
156	40B	103	U24567867	1	BOEING 787-8
191	37D	104	J23489203	2	BOEING 777

7. **Payment - <u>Transaction_ID</u>**

```
CREATE TABLE PAYMENT
(
   TRANSACTION_ID INT,
   FARE INT,
   MODE VARCHAR(20),
   PASS_NO VARCHAR(10),
   AGENT_ID INT,
   PRIMARY KEY(TRANSACTION_ID)
);
```

Populating Payment:

```
INSERT INTO PAYMENT VALUES(1000, 10000, 'CASH', 'M1345267', 100);
INSERT INTO PAYMENT VALUES(1001, 30000, 'DEBIT CARD', 'A50123321', 101);
INSERT INTO PAYMENT VALUES(1002, 4500, 'CREDIT CARD', 'J6124361', 102);
INSERT INTO PAYMENT VALUES(1003, 25300, 'NET-BANKING', 'P2368145', 103);
INSERT INTO PAYMENT VALUES(1004, 17000, 'DEBIT CARD', 'N11366775', 104);
```

TRANSACTION_ID	FARE	MODE	PASS_NO	AGENT_ID
1000	10000	CASH	M1345267	100
1001	30000	DEBIT CARD	A50123321	101
1002	4500	CREDIT CARD	J6124361	102
1003	25300	NET-BANKING	P2368145	103
1004	17000	DEBIT CARD	N11366775	104

8. **Old_Ticket_Information -**

```
CREATE TABLE OLD_TICKET_INFORMATION
(
        TICKET_ID INT,
   SEAT_NO VARCHAR(5),
   PASSPORT_NO VARCHAR(10)
);
```

TICKET_ID	SEAT_NO	PASSPORT_NO

QUERIES

Query 1 - SELECT ALL AIRLINES FLYING BOEING 787-8

SELECT COMPANY FROM AIRPLANE
WHERE STRCMP(A_TYPE,'BOEING 787-8') = 0;

```
160     #QUERY 1 - SELECT ALL AIRLINES FLYING BOEING 787-8
161 •   SELECT COMPANY FROM AIRPLANE
162     WHERE STRCMP(A_TYPE,'BOEING 787-8') = 0;
163
164     #Query 2 - SELECT THE PASSENGER WHOSE PASSPORT NUMBER IS J6124361
165 •   SELECT * FROM CUSTOMER C
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

COMPANY
AIR INDIA

Query 2 - SELECT THE PASSENGER WHOSE PASSPORT NUMBER IS M1345267

SELECT * FROM CUSTOMER C
WHERE PASS_NO = 'M1345267';

```
164     #Query 2 - SELECT THE PASSENGER WHOSE PASSPORT NUMBER IS M1345267
165 •   SELECT * FROM CUSTOMER C
166     WHERE PASS_NO = 'M1345267';
```

Result Grid | Filter Rows: | Edit: | Export/Import:

C_NUMBER	PASS_NO	C_EMAIL	C_NAME	AGENT_ID
971544432342	M1345267	AJ@GMAIL.COM	AGRIM JAIN	200

Query 3 - SELECT ALL PASSENGERS WHOSE AGENT IS 'Lebron James'

SELECT * FROM CUSTOMER
WHERE AGENT_ID = 203;

```
168     #QUERY 3 - SELECT ALL PASSENGERS WHO'S AGENT IS 'Lebron James'
169 •   SELECT * FROM CUSTOMER
170     WHERE AGENT_ID = 203;
171
172     #Query 4 - SELECT ALL PASSENGERS WHO PAID LESS THAN 20000 FOR
173 •   SELECT * FROM CUSTOMER C, PAYMENT P
174     WHERE C.PASS_NO = P.PASS_NO
175     AND (FARE < 20000);
```

Result Grid | Filter Rows: | Edit: | Export/Import:

C_NUMBER	PASS_NO	C_EMAIL	C_NAME	AGENT_ID
971545834545	P2368145	JP@GMAIL.COM	JAY PARIDA	203

Query 4 - SELECT ALL PASSENGERS WHO PAID LESS THAN 20000 FOR THEIR FLIGHT TICKET

```
SELECT * FROM CUSTOMER C, PAYMENT P
WHERE C.PASS_NO = P.PASS_NO
AND (FARE < 20000);
```

```
172  #Query 4 - SELECT ALL PASSENGERS WHO PAID LESS THAN 20000 FOR THEIR FLIGHT TICKET
173  SELECT * FROM CUSTOMER C, PAYMENT P
174  WHERE C.PASS_NO = P.PASS_NO
175  AND (FARE < 20000);
176
177  #Query 5 - SELECT ALL FLIGHT DETAILS OF ARRIVAL AT DXB
178  SELECT * FROM AIRPLANE A
179  WHERE A.AA_CODE = 'DXB';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

C_NUMBER	PASS_NO	C_EMAIL	C_NAME	AGENT_ID	TRANSACTION_ID	FARE	MODE	PASS_NO	AGENT_ID
971234232345	N11366775	NEEDY@GMAIL.COM	NIDHI WIDHANI	204	1004	17000	DEBIT CARD	N11366775	104
971523423234	J6124361	SHIJU@GMAIL.COM	HARSH VEGAD	202	1002	4500	CREDIT CARD	J6124361	102
971544432342	M1345267	AJ@GMAIL.COM	AGRIM JAIN	200	1000	10000	CASH	M1345267	100

Query 5 - SELECT ALL FLIGHT DETAILS OF ARRIVAL AT DXB

```
SELECT * FROM AIRPLANE A
WHERE A.AA_CODE = 'DXB';
```

```
177  #Query 5 - SELECT ALL FLIGHT DETAILS OF ARRIVAL AT DXB
178  SELECT * FROM AIRPLANE A
179  WHERE A.AA_CODE = 'DXB';
180
181  #Query 6 - PERFORM LEFT JOIN ON AGENT AND CUSTOMER TABLE
182  SELECT * FROM AGENT LEFT JOIN CUSTOMER
```

Result Grid | Filter Rows: | Edit: | Export/Import:

A_ID	A_TYPE	COMPANY	DA_CODE	AA_CODE
2	BOEING 777	EMIRATES	BOM	DXB

Query 6 - PERFORM LEFT JOIN ON AGENT AND CUSTOMER TABLE

```
SELECT * FROM AGENT LEFT JOIN CUSTOMER
ON AGENT.AGENT_ID = CUSTOMER.AGENT_ID
ORDER BY AGENT.AGENT_ID;
```

```
181  #Query 6 - PERFORM LEFT JOIN ON AGENT AND CUSTOMER TABLE
182  SELECT * FROM AGENT LEFT JOIN CUSTOMER
183  ON AGENT.AGENT_ID = CUSTOMER.AGENT_ID
184  ORDER BY AGENT.AGENT_ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AGENT_ID	AGENT_NAME	AGENT_MAIL	AGENT_PHONE	COMMISSION	C_NUMBER	PASS_NO	C_EMAIL	C_NAME	AGENT
200	CHAMPAGNE PAPI	DRAKE@OVO.COM	+1 8321456842	13%	971544432342	M1345267	AJ@GMAIL.COM	AGRIM JAIN	200
201	JACK HARLOW	JACKH@FIRSTCLASS.COM	+1 9862317261	18%	971434543345	A50123321	PAJJI@GMAIL.COM	ARSHDEEP SINGH	201
202	DUA LIPA	LEVITATING@GMAIL.COM	+1 9876543210	14%	971523423234	J6124361	SHIJU@GMAIL.COM	HARSH VEGAD	202
203	LEBRON JAMES	KINGJAMES@GMAIL.COM	+1 7468215260	20%	971545834545	P2368145	JP@GMAIL.COM	JAY PARIDA	203
204	VIRAT KOHLI	VK18I@GMAIL.COM	+91 8146800561	10%	971234232345	N11366775	NEEDY@GMAIL.COM	NIDHI WIDHANI	204

Query 7 - SELECT ALL PASSENGERS WHO USED CASH AS MODE OF PAYMENT

```
SELECT C_NAME, C.PASS_NO FROM CUSTOMER C, PAYMENT P
WHERE C.PASS_NO = P.PASS_NO
AND STRCMP(P.MODE, 'CASH') = 0;
```

```
186     #QUERY 7 - SELECT ALL PASSENGERS WHO USED CASH AS MODE OF PAYMENT
187 •   SELECT C_NAME, C.PASS_NO FROM CUSTOMER C, PAYMENT P
188     WHERE C.PASS_NO = P.PASS_NO
189     AND STRCMP(P.MODE, 'CASH') = 0;
190
191     #QUERY 8 - SELECT ALL PASSENGERS WHO ARE TRAVELLING IN AIR INDIA OR EM
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

C_NAME	PASS_NO
AGRIM JAIN	M1345267

PROCEDURES

***/**
PROCEDURE 1
CREATE A PROCEDURE WHICH DISPLAYS THE NAMES OF THE COMPANIES WHOSE AIRPLANES HAVE MORE THAN 100 MAXIMUM SEATS
/*

```
DELIMITER //
CREATE PROCEDURE GetAirplaneCompanies()
BEGIN
        SELECT A1.COMPANY AS 'COMPANY NAME'
   FROM AIRPLANE A1, AIRPLANE_2NF A2
   WHERE A2.MAX_SEATS_2NF > 100
   AND A2.A_TYPE_2NF = A1.A_TYPE;
END //
```

/*
PROCEDURE 2
CREATE A PROCEDURE WHICH DISPLAYS THE NAMES OF ALL THE AIRPORTS WHICH ARE IN INDIA
***/**

```
DELIMITER //
CREATE PROCEDURE GetAirportsInIndia()
BEGIN
        SELECT DNAME
   FROM DEPARTURE_AIRPORT
   WHERE CITY <> 'DUBAI';
END //

call GetAirplaneCompanies();
call GetAirportsInIndia();
```

Procedure 1

```
199  PROCEDURE 1
200  CREATE A PROCEDURE WHICH DISPLAYS THE NAMES OF THE COMPANIES WHOSE AIRPLANES HAVE MORE THAN 100 MAXIMUM SEATS
201  */
202  DELIMITER //
203  CREATE PROCEDURE GetAirplaneCompanies()
204  BEGIN
205      SELECT A1.COMPANY AS 'COMPANY NAME'
206      FROM AIRPLANE A1, AIRPLANE_2NF A2
207      WHERE A2.MAX_SEATS_2NF > 100
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

COMPANY NAME
AIR INDIA
INDIGO
EMIRATES

Procedure 2

```
217  BEGIN
218      SELECT DNAME
219      FROM DEPARTURE_AIRPORT
220      WHERE CITY <> 'DUBAI';
221  END //
222
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

DNAME
CHHATRAPATI SHIVAJI INTERNATIONAL AIRP...
INDIRA GANDHI INTERNATIONAL AIRPORT
RAJIV GANDHI INTERNATIONAL AIRPORT
DEVI AHILYABHAI HOLKAR AIRPORT
CHANDIGARH INTERNATIONAL AIRPORT

TRIGGERS

Trigger 1: CREATE A TRIGGER FOR STORING OLD TICKET INFORMATION IF THE TICKET TABLE IS UPDATED

```
CREATE TRIGGER `ticket_BEFORE_UPDATE` BEFORE UPDATE ON TICKET
FOR EACH ROW
INSERT INTO OLD_TICKET_INFORMATION values(OLD.TICKET_ID, OLD.SEAT_NO,
OLD.PASS_NO);
```

Trigger 2: CREATE A TRIGGER FOR THE PAYMENT TABLE WHICH, IF FARE INSERTED IS LESS THAN 0, ASSIGNS VALUE 0 TO FARE

```
DELIMITER //
CREATE TRIGGER `payment_BEFORE_INSERT` BEFORE INSERT ON PAYMENT
FOR EACH ROW BEGIN
        IF NEW.FARE < 0 THEN SET NEW.FARE = 0;
END IF;
END //
```

Trigger	Event	Table	Statement	Timing	Created	sql_mode
payment_BEFORE_INSERT	INSERT	payment	BEGIN IF NEW.FARE < 0 THEN SET NEW.FARE ...	BEFORE	2022-05-06 15:18...	STRICT_TRANS_TABLES,NO_ENGINE_SUBSTIT...
ticket_BEFORE_UPDATE	UPDATE	ticket	INSERT INTO OLD_TICKET_INFORMATION valu...	BEFORE	2022-05-06 15:18...	STRICT_TRANS_TABLES,NO_ENGINE_SUBSTIT...

REFERENCES

1. Class Notes
2. Silberschatz A, H F Korth, S Sudarsan, Database System Concepts, Sixth Ed, McGraw-Hill, 2010
3. (Ramakrishna R. & Gehrke J, Database Management Systems, 3 rd Edition, Mc-Graw Hill, 2002.
4. Hector G Molina, Jeffrey D.Ullman and Jennifer Widom, Database Systems – The Complete Book, Pearson Education, 2nd Ed 2009.
5. Elmarsi R, & Navathe S B, Fundamental of Database System, 5e, Pearson Education, 2008.
6. Jason S. Couchman, Oracle Certified Professional, DBA Certification Exam Guide,Tata Mc-Graw Hill
7. www.geeksforgeeks.org/dbms/?ref=ghm