



**PES UNIVERSITY**

100 feet Ring Road, BSK 3<sup>rd</sup> Stage, Bengaluru 560085

UE17CS252

Database Management Systems

# Project Report

## Airline Database Management System

PES1201700160 - Shrutiya M

PES1201701576 - Manoj Pissay A

PES1201701597 - Sachetan G S

# Table of Contents

- Introduction
- Problem Statement
- Requirements Specification
- Design
  - ER Diagram
  - Relational Schema Diagram
  - SQL
    - DDL Statements
    - INSERT statements for bulk loading of initial data
- Implementation
  - Web User Interfaces
    - CRUD Operations
  - SQL Complex Queries with results
- Conclusion with proposed enhancements
- References
- Appendices
  - Abbreviations & Acronyms
  - Databases, Tools & Technologies Used
  - Metadata

# Introduction

Airline reservation System is a computerized system used to store and retrieve information and conduct transactions related to air travel. The project is aimed at exposing the relevance and importance of Airline Reservation Systems. It is projected towards enhancing the relationship between customers and airline agencies and thereby making it convenient for the customers to book the flights as when they require such that they can utilize this software to make reservation.

## Problem Statement

The project aims to build a distributed database(online) for system administration of an Airline Company and automated booking of flights. The airline allows advanced and on spot reservation of tickets. This system allows the passengers to know about the flight details like type of aircraft, class, baggage facilities etc.

Each passenger is allowed up to a particular cargo free with each ticket. Thus this system reduces manual work to a great extent allows smooth flow of airline activities by removing chances of errors in the details.

## Requirements Specification

### **Queries the Passenger can get the answer to:**

1. Select Source and Destination cities whose fare is greater than Rs.8000
2. Employees who do not work for any particular flight.
3. Select the flight that has business class and travels from Bangalore to Delhi on 22-03-19
4. Select employees who work for the same flight and get the same salary as dispatcher.
5. Flights which depart on 22-03-2019 except for the ones in the morning(using EXCEPT)
6. Select employees who is travelling as a passenger
7. Flights which do not have business class
8. Flights with total passengers less than 10
9. Create a trigger if the passenger tries to book a flight two days or one day back and increase the fare.
10. Query to find the maximum salary paid to employees in particular roles
11. To select routes which have more than one trip a day with the departure time after 4am

## Features

- Free Account
- Selection among Large no of Airways
- Full detailed Flights
- Easy to Get Flights
- Easy to edit and view your Personal Information
- User Friendly
- Secure

## Conclusion with proposed enhancements

We are trying to give a live reporting which is updated by Airline Companies so that the customer gets a live Flight check, available seats, pricing and also planning to provide seats as per their choice so that they can travel comfortably.

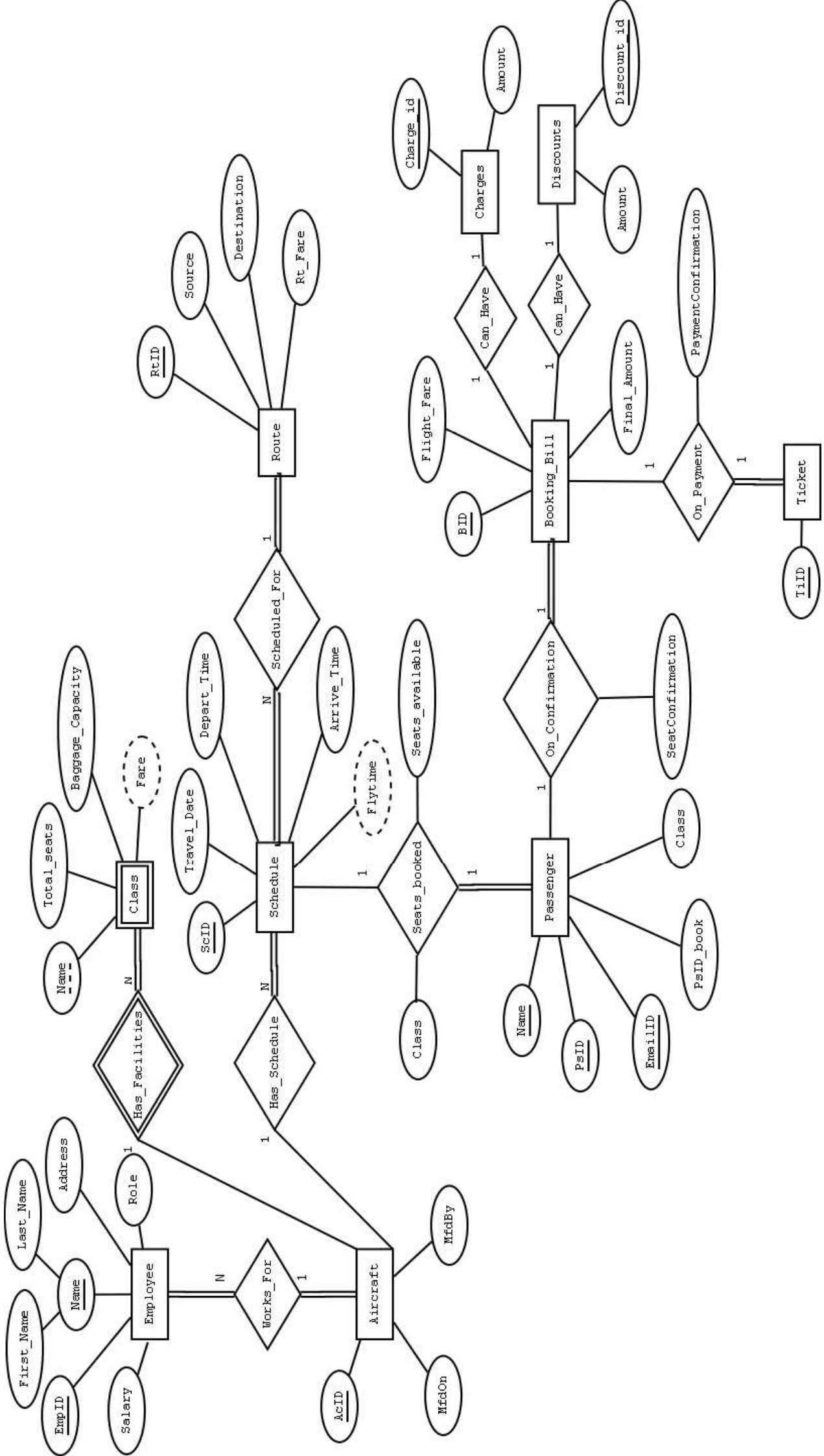
We will try to provide food facility choices to customers to make their experience in our airline even more memorable. Our future planning is to take this project towards an Android App so that the Customers can contact the Airline and get quick services from airline.

## References

- Fundamentals of Database Systems, Ramez Elamsri, Shamkant B Navathe, PEARSON

## Databases, Tools & Technologies Used

- Postgres
- HTML
- CSS
- JavaScript
- PHP



AIRCRAFT:

<u>AcID</u>	MfdBy	MfdOn
-------------	-------	-------

EMPLOYEE:

<u>EmpID</u>	AcID	FirstName	LastName	Address	Salary	Role	Email
--------------	------	-----------	----------	---------	--------	------	-------

CLASS:

<u>AcID</u>	<u>Name</u>	TotalSeats	BaggageCapacity
-------------	-------------	------------	-----------------

ROUTE:

<u>RtID</u>	Source	Destination	Fare
-------------	--------	-------------	------

SCHEDULE:

<u>ScID</u>	TravelDate	DepartTime	ArriveTime	AcID	RtID
-------------	------------	------------	------------	------	------

PASSENGER:

<u>PsID</u>	Name	Email	Psid_book	ScID	AcClass	AcSeatsAvailable
-------------	------	-------	-----------	------	---------	------------------

BOOKING\_BILL:

<u>BkID</u>	Flight_Fare	Baggage_Fare	Charge_id	Discnt_id	PsID	SeatConfirmation	TotalAmt
-------------	-------------	--------------	-----------	-----------	------	------------------	----------

TICKET:

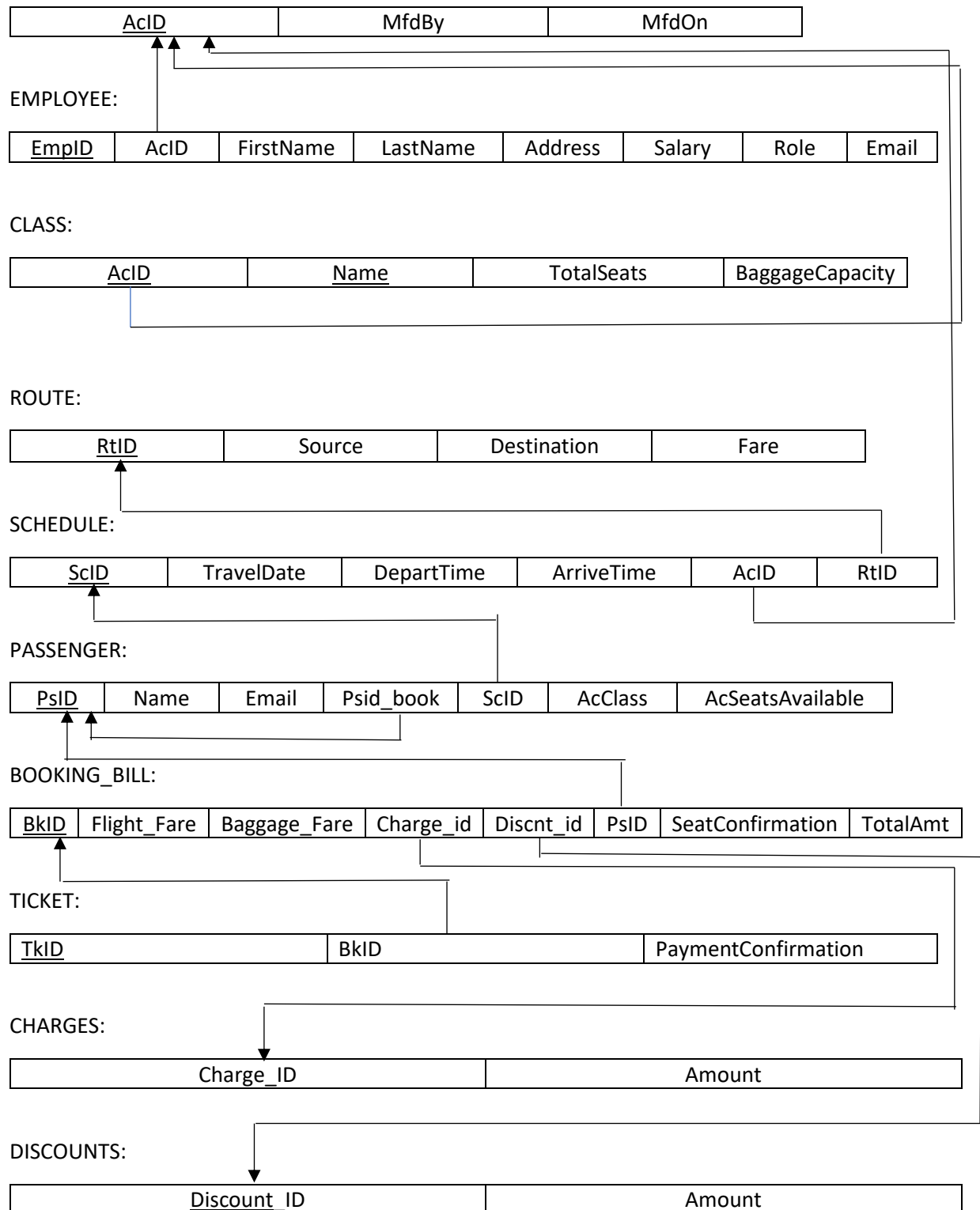
<u>TkID</u>	BkID	PaymentConfirmation
-------------	------	---------------------

CHARGES:

<u>Charge_ID</u>	Amount
------------------	--------

DISCOUNTS:

<u>Discount_ID</u>	Amount
--------------------	--------



### **SQL DDL Statements:**

```
CREATE TABLE Aircraft (  
    AcID VARCHAR(32) NOT NULL,  
    MfdBy VARCHAR(32) NOT NULL,  
    MfdOn DATE NOT NULL,  
    PRIMARY KEY (AcID)  
);
```

```
CREATE TABLE Employee (  
    EmpID VARCHAR(32) NOT NULL,  
    FirstName VARCHAR(32) NOT NULL,  
    LastName VARCHAR(32) NOT NULL,  
    Address VARCHAR(252) NOT NULL,  
    Salary DECIMAL NOT NULL,  
    Role VARCHAR(32) NOT NULL,  
    Email VARCHAR(32) NOT NULL,  
    AcID VARCHAR(32) NOT NULL,  
    PRIMARY KEY (EmpID),  
    FOREIGN KEY (AcID) REFERENCES Aircraft(AcID)  
);
```

```
CREATE TABLE Class (  
    Name VARCHAR(32) NOT NULL UNIQUE,  
    TotalSeats INT NOT NULL,  
    BaggageCapacity INT NOT NULL,  
    AcID VARCHAR(32) NOT NULL,  
    FOREIGN KEY (AcID) REFERENCES Aircraft(AcID)  
);
```

```
CREATE TABLE Route (  
    RtID VARCHAR(32) NOT NULL,  
    Source VARCHAR(32) NOT NULL,  
    Destination VARCHAR(32) NOT NULL,  
    Fare DECIMAL NOT NULL,  
    PRIMARY KEY (RtID)  
);
```

```
CREATE TABLE Schedule (  
    ScID VARCHAR(32) NOT NULL,  
    TravelDate DATE NOT NULL,  
    DepartTime TIME NOT NULL,  
    ArriveTime TIME NOT NULL,  
    AcID VARCHAR(32) NOT NULL,  
    RtID VARCHAR(32) NOT NULL,  
    PRIMARY KEY (ScID),  
    FOREIGN KEY (AcID) REFERENCES Aircraft(AcID),  
    FOREIGN KEY (RtID) REFERENCES Route(RtID)  
);
```

```
CREATE TABLE Passenger (  
    PsID VARCHAR(32) NOT NULL,  
    Name VARCHAR(32) NOT NULL,  
    Email VARCHAR(32) NOT NULL,  
    NumSeats INT NOT NULL,  
    AcClass VARCHAR(32) NOT NULL,  
    AcSeatsAvailable INT NOT NULL,  
    ScID VARCHAR(32) NOT NULL,  
    PRIMARY KEY (PsID),  
    FOREIGN KEY (ScID) REFERENCES Schedule(ScID)  
);
```

```
CREATE TABLE Booking_Bill (  
    BkID VARCHAR(32) NOT NULL,  
    FlightFare DECIMAL NOT NULL,  
    BaggageFare DECIMAL NOT NULL,  
    ChargeID VARCHAR(32) NOT NULL,  
    DiscountID VARCHAR(32) NOT NULL,  
    PsId VARCHAR(32) NOT NULL,  
    SeatConfirmation CHAR(1) NOT NULL,  
    TotalAmt DECIMAL NOT NULL,  
    PRIMARY KEY (BkID),  
    FOREIGN KEY (ChargeID) REFERENCES Charges(ChargeID),  
    FOREIGN KEY (DiscountID) REFERENCES Discounts(DiscountID),  
    FOREIGN KEY (PsID) REFERENCES Passenger(PsID)  
);
```

```
CREATE TABLE Ticket (  
    TID VARCHAR(32) NOT NULL,  
    BkID VARCHAR(32) NOT NULL,  
    PaymentConfirmation CHAR(1) NOT NULL,  
    PRIMARY KEY (TID),  
    FOREIGN KEY (BkID) REFERENCES Booking_Bill(BkID)  
);
```

```
CREATE TABLE Charges (  
    ChargeID VARCHAR(32) NOT NULL,  
    Amount DECIMAL NOT NULL,  
    PRIMARY KEY (ChargeID)  
);
```

```
CREATE TABLE Discounts (  
    DiscountID VARCHAR(32) NOT NULL,  
    Amount DECIMAL NOT NULL,  
    PRIMARY KEY (DiscountID)  
);
```

```
ALTER TABLE Class  
ADD CONSTRAINT weakPrimary PRIMARY KEY (AcID,Name);
```



```
ALTER TABLE Class
DROP CONSTRAINT class_name_key;
```

```
ALTER TABLE booking_bill
DROP CONSTRAINT booking_bill_psid_fkey,
ADD CONSTRAINT booking_bill_psid_fkey FOREIGN KEY (psid)
REFERENCES passenger (psid) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE PASSENGER
DROP COLUMN numseats
```

```
ALTER TABLE PASSENGER
ADD COLUMN psid_book varchar(32),
ADD CONSTRAINT P_fkey foreign key (psid_book) references passenger(psid) ON delete
cascade on update cascade;
```

```
ALTER TABLE Employee ALTER COLUMN Acid DROP NOT NULL;
```

```
UPDATE Employee
SET Acid=NULL
WHERE role='Sales Representative';
```

```
--updating the flight fare
UPDATE Booking_Bill AS B
SET FLightFare=
(Select R.Fare
From (Passenger as P natural join Schedule AS S) AS C, Route AS R
WHERE B.PSiD=C.PSiD AND C.RtID=R.RtID);
```

```
--updating the total amount with flightfare and baggageFare
UPDATE Booking_Bill AS B
SET ToatalAmt=FlightFare+BaggageFare;
```

```
--updating the total amt with charge and discount
UPDATE Booking_Bill AS B
SET TotalAmt=
Case when ChargeID=NULL and DiscountID=NULL Then B.TotalAmt+0
  when chargeID is not NULL and DiscountID=null Then B.TotalAmt+(Select amount from
charges as C where B.chargeID=C.chargeid)
  when chargeID=NULL and DiscountID is not null Then B.TotalAmt-(Select amount from
discounts as D where B.discountID=D.discountID)
else
  B.TotalAmt+(Select amount from charges as C where B.chargeID=C.chargeid)-(Select
amount from discounts as D where B.discountID=D.discountID)
end;
```

### **INSERT Statements:**

INSERT INTO Aircraft VALUES ('K2948','Hindustan Aeronautics Limited(HAL)','2018-01-01');  
INSERT INTO Aircraft VALUES ('K3556','Hindustan Aeronautics Limited(HAL)','2018-01-01');  
INSERT INTO Aircraft VALUES ('K9869','Hindustan Aeronautics Limited(HAL)','2018-01-01');  
INSERT INTO Aircraft VALUES ('K5348','Hindustan Aeronautics Limited(HAL)','2018-01-01');  
INSERT INTO Aircraft VALUES ('K2132','Hindustan Aeronautics Limited(HAL)','2018-01-01');  
INSERT INTO Aircraft VALUES ('K9983','Hindustan Aeronautics Limited(HAL)','2018-01-01');  
INSERT INTO Aircraft VALUES ('K8732','Hindustan Aeronautics Limited(HAL)','2018-01-01');  
INSERT INTO Aircraft VALUES ('K6943','Hindustan Aeronautics Limited(HAL)','2018-01-01');  
INSERT INTO Aircraft VALUES ('K8733','Hindustan Aeronautics Limited(HAL)','2018-01-01');  
INSERT INTO Aircraft VALUES ('K3278','Hindustan Aeronautics Limited(HAL)','2018-01-01');

INSERT INTO Employee VALUES ('E20182138','James','Smith','1607 23rd Street NW, Washington, DC 20008','45000.00','Flight Dispatcher','jamessmith@gmail.com','K2948');  
INSERT INTO Employee VALUES ('E20185345','Nancy','Ferguson','664 Dickens Road, Lilburn, Georgia 30047','45000.00','Flight Dispatcher','nancyferguson@gmail.com','K3556');  
INSERT INTO Employee VALUES ('E20185732','Maria','Rodriguez','2312 Dorrington Dr., Dallas, TX 75228','45000.00','Flight Dispatcher','mariarodriguez@gmail.com','K9869');  
INSERT INTO Employee VALUES ('E20186546','David','Johnson','2026 Scott Street, Hollywood, FL 33020','50000.00','Operations Agent','davidjohnson@gmail.com','K5348');  
INSERT INTO Employee VALUES ('E20187532','Henry','Thomas','318 Canino Road, Houston, TX 77076','50000.00','Operations Agent','henrythomas@gmail.com','K2132');  
INSERT INTO Employee VALUES ('E20187567','Sarah','Catherine','1907 Spruce Street, Philadelphia, PA 19103-5732','35000.00','Sales Representative','sarahcatherine@gmail.com','K9983');  
INSERT INTO Employee VALUES ('E20181437','Samuel','Williams','1250 East Moore Lake Drive, Suite 242, Minneapolis (Fridley), MN 55432','35000.00','Sales Representative','samuelwilliams@gmail.com','K8732');  
INSERT INTO Employee VALUES ('E20188673','Charles','Joseph','1350 Red Rock Street, Las Vegas, NV 89146','40000.00','Airline Flight Instructor','charlesjoseph@gmail.com','K6943');  
INSERT INTO Employee VALUES ('E20187562','Mary','George','5306 Walnut Ave., Building A, Sacramento, CA 95841','40000.00','Airline Flight Instructor','marygeorge@gmail.com','K8733');  
INSERT INTO Employee VALUES ('E20186275','Daniel','Peter','1151 SW Vermont, Portland, OR 97219','30000.00','Avionics Technician','danielpeter@gmail.com','K3278');  
--new  
INSERT INTO Employee VALUES ('E20182145','Olenna','Reddington','1607 24rd Street NW, Washington, DC 20008','45000.00','Air Hostess','olennared@gmail.com','K2948');

```

INSERT INTO Route VALUES ('BLRBOM','Bengaluru','Mumbai','6179.00');
INSERT INTO Route VALUES ('BLRHYD','Bengaluru','Hyderabad','14894.00');
INSERT INTO Route VALUES ('BLRKLK','Bengaluru','Kolkata','7832.00');
INSERT INTO Route VALUES ('BLRDEL','Bengaluru','Delhi','14640.00');
INSERT INTO Route VALUES ('BOMHYD','Mumbai','Hyderabad','4567.00');
INSERT INTO Route VALUES ('KLKDEL','Kolkata','Delhi','3245.00');
INSERT INTO Route VALUES ('BOMKLK','Mumbai','Kolkata','4589.00');
INSERT INTO Route VALUES ('HYDDEL','Hyderabad','Delhi','6179.00');
INSERT INTO Route VALUES ('DELHYD','Delhi','Hyderabad','5439.00');
INSERT INTO Route VALUES ('KLKBOM','Kolkata','Mumbai','3585.00');
INSERT INTO Route VALUES ('HYDKLK','Hyderabad','Kolkata','14894.00');
INSERT INTO Route VALUES ('DELKLK','Delhi','Kolkata','5439.00');
INSERT INTO Route VALUES ('DELBOM','Delhi','Mumbai','7543.00');
INSERT INTO Route VALUES ('KLKBBLR','Kolkata','Bengaluru','6179.00');
INSERT INTO Route VALUES ('BOMBLR','Mumbai','Bengaluru','9164.00');
INSERT INTO Route VALUES ('HYDBOM','Hyderabad','Mumbai','14894.00');
INSERT INTO Route VALUES ('BOMDEL','Mumbai','Delhi','8714.00');
INSERT INTO Route VALUES ('KLKHYP','Kolkata','Hyderabad','4871.00');
INSERT INTO Route VALUES ('HYDBLR','Hyderabad','Bengaluru','14894.00');
INSERT INTO Route VALUES ('DELBLR','Delhi','Bengaluru','9879.00');

```

```

INSERT INTO Schedule VALUES ('S201903221','2019-03-22','07:00:00','08:40:00','K2948','BLRBOM');
INSERT INTO Schedule VALUES ('S201903222','2019-03-22','18:54:00','20:50:00','K3556','BLRHYD');
INSERT INTO Schedule VALUES ('S201903223','2019-03-22','11:35:00','08:40:00','K9869','BLRKLK');
INSERT INTO Schedule VALUES ('S201903224','2019-03-22','06:40:00','05:10:00','K5348','BLRDEL');
INSERT INTO Schedule VALUES ('S201903225','2019-03-22','23:00:00','02:40:00','K2132','BOMHYD');
INSERT INTO Schedule VALUES ('S201903231','2019-03-23','16:45:00','20:40:00','K9983','KLKDEL');
INSERT INTO Schedule VALUES ('S201903232','2019-03-23','03:00:00','08:40:00','K8732','BOMKLK');
INSERT INTO Schedule VALUES ('S201903233','2019-03-23','14:00:00','20:40:00','K6943','HYDDEL');
INSERT INTO Schedule VALUES ('S201903234','2019-03-23','19:00:00','21:40:00','K8733','DELHYD');
INSERT INTO Schedule VALUES ('S201903235','2019-03-23','10:00:00','16:40:00','K3278','KLKBOM');

```

--new

```

INSERT INTO Schedule VALUES ('S201903226','2019-03-22','10:00:00','11:40:00','K8732','BLRBOM');
INSERT INTO Schedule VALUES ('S201903227','2019-03-22','05:00:00','10:40:00','K2948','KLKBOM');
INSERT INTO Schedule VALUES ('S201903237','2019-03-23','05:00:00','11:40:00','K6943','DELHYD');;

```

INSERT INTO Passenger VALUES ('P87421','Bary Allen','baryallen@gmail.com','4','First Class','10','S201903221');

INSERT INTO Passenger VALUES ('P38171','Harvy Spectre','harvyspec@gmail.com','1','Business Class','12','S201903222');

INSERT INTO Passenger VALUES ('P68910','Tom Cruise','tomcruise@gmail.com','2','Business Class','14','S201903223');

INSERT INTO Passenger VALUES ('P86731','John Smith','johnsmith@gmail.com','5','Economy Class','8','S201903224');

--new

INSERT INTO Passenger VALUES ('P85432','James Smith','jamessmith@gmail.com','5','Economy Class','8','S201903222');

INSERT INTO Passenger VALUES ('P87843','Sarah Catherine','sarahcatherine@gmail.com','3','First Class','12','S201903223');

INSERT INTO Class VALUES ('Economy Class','50','25','K2948');

INSERT INTO Class VALUES ('Business Class','10','35','K2948');

INSERT INTO Class VALUES ('First Class','50','15','K2948');

INSERT INTO Class VALUES ('Economy Class','50','25','K3556');

INSERT INTO Class VALUES ('Business Class','10','35','K3556');

INSERT INTO Class VALUES ('First Class','50','15','K3556');

INSERT INTO Class VALUES ('First Class','50','15','K9869');

INSERT INTO Class VALUES ('Economy Class','50','25','K9869');

INSERT INTO Class VALUES ('First Class','50','15','K5348');

INSERT INTO Class VALUES ('Economy Class','50','25','K5348');

INSERT INTO Class VALUES ('First Class','50','15','K2132');

INSERT INTO Class VALUES ('Economy Class','50','25','K2132');

INSERT INTO Class VALUES ('First Class','50','15','K9983');

INSERT INTO Class VALUES ('Economy Class','50','25','K9983');

INSERT INTO Class VALUES ('First Class','50','15','K8732');

INSERT INTO Class VALUES ('Economy Class','50','25','K8732');

INSERT INTO Class VALUES ('First Class','50','15','K6943');

INSERT INTO Class VALUES ('Economy Class','50','25','K6943');

INSERT INTO Class VALUES ('Business Class','50','15','K8733');

INSERT INTO Class VALUES ('Economy Class','50','25','K8733');

INSERT INTO Class VALUES ('Business Class','50','15','K3278');

INSERT INTO Class VALUES ('Economy Class','50','25','K3278');

INSERT INTO Charges VALUES ('Tax','2000.00');

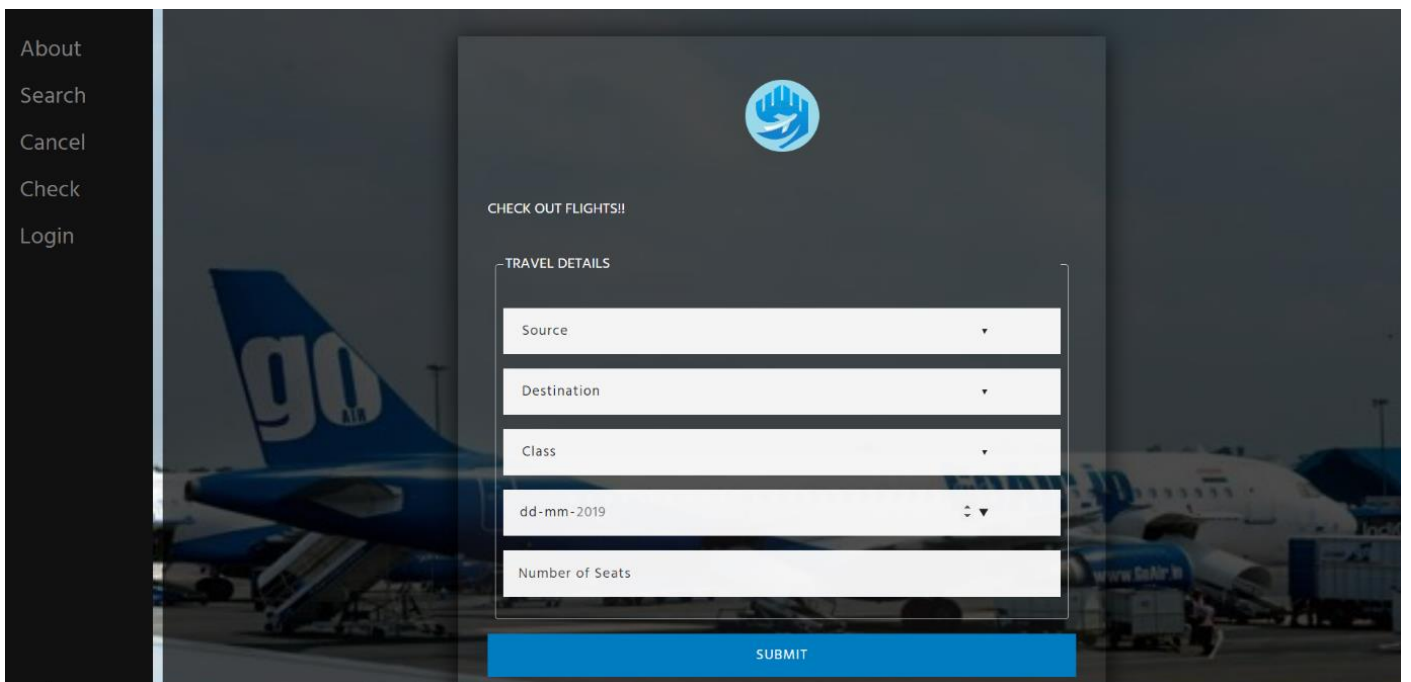
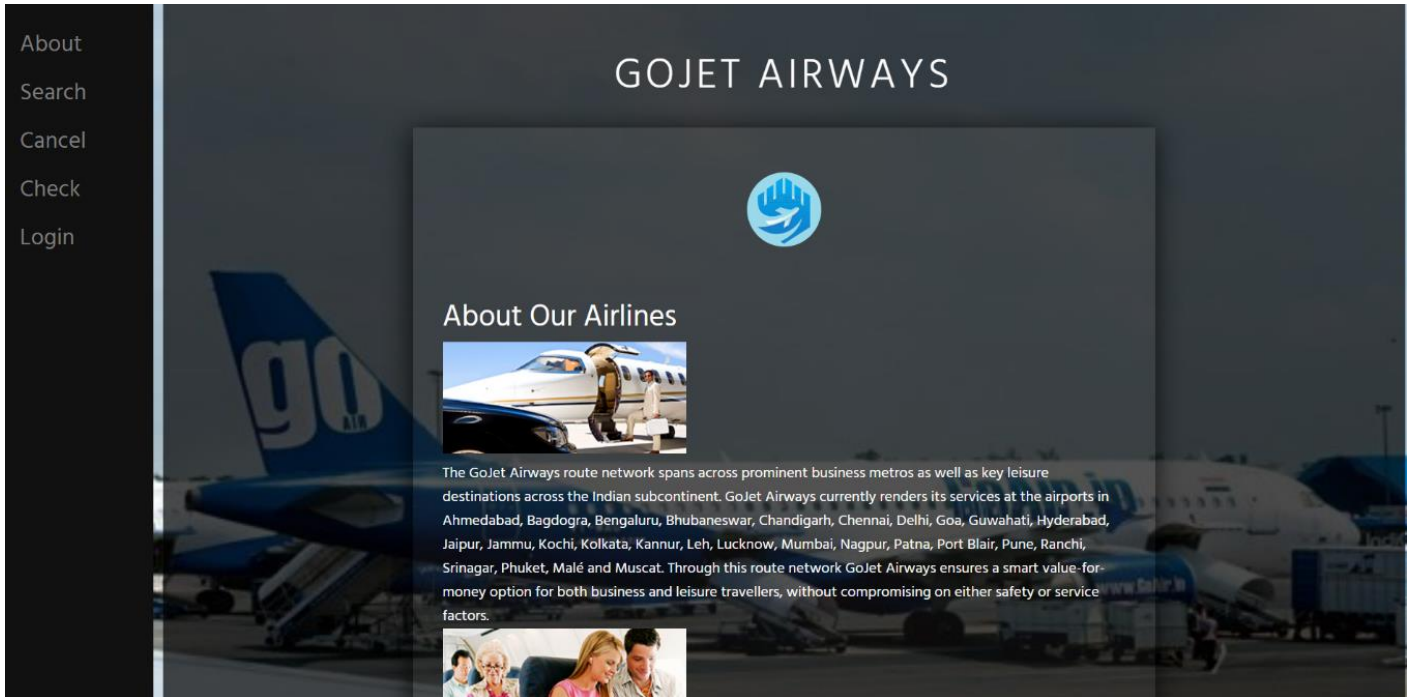
INSERT INTO Charges VALUES ('Extra Baggage (5kg)','1900.00');

INSERT INTO Charges VALUES ('Carriage of Weapons','5000.00');

INSERT INTO Discounts VALUES ('go2000','2000.00');

INSERT INTO Discounts VALUES ('go3000','3000.00');

## Website UI:



About  
Search  
Cancel  
Check  
Login

## GOJET AIRWAYS TICKET CANCELLATION



HEY ARE U SURE U WANNA CANCEL THE TICKET?!?!

[GO BACK](#)

PassengerID

[NEXT](#)

NOT SURE ABOUT UR PASSENGER ID ?? NO PROBLEM !


[CLICK HERE](#)


© 2019 GoJet Airways. All rights reserved |

About  
Search  
Cancel  
Check  
Login

## GOJET AIRWAYS LOGIN FORM



 Admin Name

 Password

[LOG IN](#)

© 2019 GoJet Airways. All rights reserved |

pgAdmin 4 File Object Tools Help

Browser

- Languages
- Schemas (1)
  - public
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Procedures
    - Sequences
    - Tables (10)
      - aircraft
      - booking\_bill
      - charges
      - class
      - discounts
      - employee
      - passenger
      - route
      - schedule
      - ticket
    - Trigger Functions
    - Types
    - Views
- postgres
- Login/Group Roles

Dependencies Dependents Edit Data - goJetAirways on postgres@PostgreSQL 11 Query - goJetAir... Edit Data - goJet... Edit Data - goJet... Edit Data - goJet... < > X

goJetAirways on postgres@PostgreSQL 11

Query Editor Query History

```
1 SELECT * FROM public.passenger
2
```

Data Output Explain Messages Notifications

	psid [PK] character varying (32)	name character varying (32)	email character varying (32)	acclass character varying (32)	acseatsavailable integer	scid character varying (32)	psid_book character varying (32)
15	P56139	shobika	shobimohan19@gmail.com	Business Class	-16	S201903226	P56139
16	P61771	shrutiya	prathik.sag3@gmail.com	Business Class	-16	S201903226	P61771
17	P63008	shrutiya	prathik.sag3@gmail.com	Business Class	-16	S201903226	P63008
18	P37856	Mohan	prathik.sag3@gmail.com	First Class	34	S201903226	P37856
19	P98867	Shobika	shobi@gmail.com	First Class	34	S201903221	[null]
20	P51417	Manoj	manoj@gmail.com	First Class	34	S201903221	[null]
21	P85391	rita	rita@gmail.com	First Class	34	S201903221	P85391
22	P91910	shilpa	shilpa@gmail.com	First Class	34	S201903221	P85391

pgAdmin 4 File Object Tools Help

Browser

- Languages
- Schemas (1)
  - public
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Procedures
    - Sequences
    - Tables (10)
      - aircraft
      - booking\_bill
      - charges
      - class
      - discounts
      - employee
      - passenger
      - route
      - schedule
      - ticket
    - Trigger Functions
    - Types
    - Views
- postgres
- Login/Group Roles

Dependencies Dependents Edit Data - goJetAirways on postgres@PostgreSQL 11 Query - goJetAir... Edit Data - goJet... Edit Data - goJet... Edit Data - goJet... < > X

goJetAirways on postgres@PostgreSQL 11

Query Editor Query History

```
1 SELECT * FROM public.passenger
2
```

Data Output Explain Messages Notifications

	psid [PK] character varying (32)	name character varying (32)	email character varying (32)	acclass character varying (32)	acseatsavailable integer	scid character varying (32)	psid_book character varying (32)
15	P56139	shobika	shobimohan19@gmail.com	Business Class	-16	S201903226	P56139
16	P61771	shrutiya	prathik.sag3@gmail.com	Business Class	-16	S201903226	P61771
17	P63008	shrutiya	prathik.sag3@gmail.com	Business Class	-16	S201903226	P63008
18	P37856	Mohan	prathik.sag3@gmail.com	First Class	34	S201903226	P37856
19	P98867	Shobika	shobi@gmail.com	First Class	34	S201903221	[null]
20	P51417	Manoj	manoj@gmail.com	First Class	34	S201903221	[null]
21	P85391	rita	rita@gmail.com	First Class	34	S201903221	P85391
22	P91910	shilpa	shilpa@gmail.com	First Class	34	S201903221	P85391

### Queries:

--simple query

--1.Select Source and Destination cities whose fare is greater than Rs.8000

```
SELECT Source, Destination
FROM Route
Where Fare > 8000.00
```

/\*

OUTPUT:

source | destination

-----+-----

Bengaluru | Hyderabad

Bengaluru | Delhi

Hyderabad | Kolkata

Mumbai | Bengaluru

Hyderabad | Mumbai

Mumbai | Delhi

Hyderabad | Bengaluru

Delhi | Bengaluru

(8 rows)

\*/

--IS operator

--2. Employees who do not work for any particular flight.

```
SELECT FirstName, LastName
FROM Employee
WHERE AcID IS NULL;
```

/\*

OUTPUT:

firstname | lastname

-----+-----

Sarah | Catherine

Samuel | Williams

(2 rows)

\*/

--Nested query

--3. Select the flight that has business class and travels from Bangalore to Delhi on 22-03-19

```
SELECT DISTINCT A.AcID
FROM Aircraft AS A
WHERE AcID IN (SELECT C.AcID
FROM Class AS C
WHERE Name = 'Business Class')
AND
```



```
AcID IN (SELECT S.AcID
FROM Route AS R,Schedule AS S
WHERE R.RtID='BLRDEL' AND S.TravelDate='2019-03-22');
```

```
/*
OUTPUT:
acid
-----
K2948
K3556
(2 rows)
*/
```

```
--IN operator
--4.Select employees who work for the same flight and get the same salary as dispatcher.
```

```
SELECT FirstName,LastName
FROM Employee
WHERE (AcID,Salary) IN (SELECT AcID,Salary
FROM Employee
WHERE Role='Flight Dispatcher');
```

```
/*
OUTPUT
firstname | lastname
-----+-----
James    | Smith
Nancy    | Ferguson
Maria    | Rodriguez
Olenna  | Reddington
(4 rows)
*/
```

```
--EXCEPT operator
--5.Flights which depart on 22-03-2019 except for the ones in the morning(using EXCEPT)
```

```
SELECT *
FROM Schedule
WHERE TravelDate='2019-03-22'
EXCEPT (SELECT *
FROM Schedule
WHERE DepartTIme<'12:00:00');
```

```
/*
OUTPUT:
scid | traveldate | departtime | arrivetime | acid | rtid
-----+-----+-----+-----+-----+-----
S201903225 | 2019-03-22 | 23:00:00 | 02:40:00 | K2132 | BOMHYD
S201903222 | 2019-03-22 | 18:54:00 | 20:50:00 | K3556 | BLRHYD
(2 rows)
*/
```

--JOIN, CONCATENATION,IS NULL OPERATOR  
--6.Select employees who is travelling as a passenger

```
SELECT P.name
FROM Employee As E JOIN Passenger AS P
  ON E.firstname||' '||E.lastname LIKE P.name
WHERE E.AcID IS NULL;
```

/\*

OUTPUT:

name

-----  
Sarah Catherine  
(1 row)  
\*/

--NOT EXISTS  
--7.Flights which donot have business class

```
SELECT A.AcID
FROM Aircraft AS A
WHERE NOT EXISTS (SELECT *
  FROM Class AS C
  WHERE Name='Business Class' AND A.AcID=C.AcID);
```

/\*

OUTPUT:

acid

-----  
K9869  
K5348  
K2132  
K9983  
K8732  
K6943  
(6 rows)  
\*/

--COUNT  
--8.Flights with total passengers less than 10

```
SELECT *
FROM Schedule
WHERE (SELECT Count(*)
  FROM (Ticket NATURAL JOIN Booking_bill) NATURAL JOIN Passenger)<=10;
```

/\*

OUTPUT:

scid | traveldate | departtime | arrivetime | acid | rtid  
-----+-----+-----+-----+-----+-----

S201903221	2019-03-22	07:00:00	08:40:00	K2948	BLRBOM
S201903222	2019-03-22	18:54:00	20:50:00	K3556	BLRHYD
S201903223	2019-03-22	11:35:00	08:40:00	K9869	BLRKLK
S201903224	2019-03-22	06:40:00	05:10:00	K5348	BLRDEL
S201903225	2019-03-22	23:00:00	02:40:00	K2132	BOMHYD
S201903231	2019-03-23	16:45:00	20:40:00	K9983	KLKDEL
S201903232	2019-03-23	03:00:00	08:40:00	K8732	BOMKLK
S201903233	2019-03-23	14:00:00	20:40:00	K6943	HYDDEL
S201903234	2019-03-23	19:00:00	21:40:00	K8733	DELHYD
S201903235	2019-03-23	10:00:00	16:40:00	K3278	KLKBOM

(10 rows)

\*/

--TRIGGER

--9.Create a trigger if the passenger tries to book a flight two days or one day back and increase the fare.

CREATE OR REPLACE FUNCTION update\_fare()

RETURNS trigger AS

\$\$

BEGIN

    UPDATE Booking\_bill AS B

    SET FlightFare=(CASE

        WHEN S.TravelDate-CURRENT\_DATE=2 THEN B.FlightFare+3000.00

        WHEN S.TravelDate-CURRENT\_DATE=1 Then B.Flightfare+5000.00

        else B.Flightfare+0.00

    END)

    From Schedule AS S;

RETURN NEW;

END;

\$\$

LANGUAGE 'plpgsql';

CREATE TRIGGER fare\_increase

AFTER INSERT

ON Passenger

FOR EACH ROW

EXECUTE PROCEDURE update\_fare();

/\*

OUTPUT:

CREATE TRIGGER

Query returned successfully in 62 msec.

\*/

--GROUP BY

--10.Query to find the maximum salary paid to employees in particular roles

SELECT role,COUNT(\*),MAX(Salary)

FROM Employee

GROUP BY role

/\*

Output

role | count | max

```
-----+-----+-----
Flight Dispatcher | 3 | 45000.00
Airline Flight Instructor | 2 | 40000.00
Avionics Technician | 1 | 30000.00
Sales Representative | 2 | 35000.00
Operations Agent | 2 | 50000.00
(5 rows)
```

\*/

--GROUP BY with HAVING

--11.To select routes which have more than one trip a day with the departure time after 4am

```
SELECT TravelDate,RtID,Count(*)
FROM Schedule
WHERE DepartTime>'4:00:00' AND RtID IN
(SELECT RtID
FROM Schedule
GROUP BY RtID,TravelDate
HAVING COUNT(*)>1)
GROUP BY RtID,TravelDate;
```

/\*

OUTPUT:

traveldate | rtid | count

```
-----+-----+-----
2019-03-23 | DELHYD | 2
2019-03-22 | BLRBOM | 2
(2 rows)
```

\*/

--12. To select the passengers who have booked tickets

select p.psid,p.name from passenger as p

group by psid

having psid\_book='P85391'

/\*

OUTPUT:

psid | name

```
-----+-----
P85391 | rita
P91910 | Shilpa
```

\*/

```
select *
from (passenger as p left outer join booking_bill as b on p.psid=b.psid) left outer join ticket
as t on b.bkid=t.bkid;
```

```
/*
```

## OUTPUT

psid	name	email	acclass	acseatsavailable	scid	psid_book	bkid	flightfare	baggagefare	chargeid	discountid	psid	seatconfirmation	totalamt	tid	bkid	paymentconfirmation
P4563	Ganesh	ganesh@gmail.com	First Class	10	S201903226												
P4563	B20190322605	6179.00	4500	;Tax	go3000	P4563	Y										
16858	T20190322605	B20190322605	Y														
P89818	Shobika	susi@gmail.com	First Class	34	S201903226												
P89818	B20190322637	6179.00	4500	;Tax	go2000	P89818	Y										
16858	B20190322637	B20190322637	Y														
P50354	Sachetan	sach@gmail.com	First Class	34	S201903221												
P50354	B20190322122	6179.00	4500	;Tax		P50354	Y										
16858	B20190322122	B20190322122	Y														
P75573	Shobika	rita@gmail.com	Economy Class	34	S201903226												
P75573	B20190322660	6179.00	4500	;Tax		P75573	Y										
16858																	
P63803	Suseela	susi@gmail.com	First Class	34	S201903221												
P63803	B20190322131	6179.00	4500	;Tax	go3000	P63803	Y										
16858																	
P56139	shobika	shobimohan19@gmail.com	Business Class	-16													
S201903226	P56139	B20190322617	6179.00	4500	;Tax	go3000	P56139										
N	16858																
P32740	Tanu	tanu@gmail.com	First Class	34	S201903221												
P32740	B20190322118	6179.00	4500	;Tax	go3000	P32740	N										
16858																	
P98592	shrutya	prathik.sag3@gmail.com	Economy Class	34													
S201903222	P98592	B20190322217	6179.00	4500	;Tax	go3000	P98592										
N	16858																
P94393	Suseela	susi@gmail.com	First Class	34	S201903221												
P94393	B20190322117	6179.00	4500	;Tax	go3000	P94393	N										
16858																	
P78490	shilpa	shilpa@gmail.com	First Class	34	S201903221												
P63803																	

```
*/
```

## --Using Views

```
create view passenger_booked as
```

```
select * from passenger natural inner join schedule natural inner join route;
```