A
Project Report
on

Airline Reservation System

Developed by

KAUSHAL BHALAIYA(IT-006)
BHARGAV BHATIYA(IT-009)
KENIS DHAMELIYA(IT-025)

Guided By:
Prof. Mukesh Goswami
Department of Information Technology
Faculty of Technology
DD University

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad-387001
October-2021

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "**Airline Reservation System**" is bonadied report of the work carried out by

1) **Mr.KAUSHAL BHALAIYA,** Student ID No : **19ITUBS123**
2) **Mr.BHARGAV BHATIYA**, Student ID No : **19ITUOS060**
3) **Mr.KENIS DHAMELIYA**, Student ID No : **19ITUOS133**

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2021-2022.

Prof. Mukesh M. Goswami
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. Vipul Dabhi
Head, Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

We would like to give our sincere acknowledgement to everybody responsible for the successful completion of our project “AIRLINE RESERVATION SYSTEM”.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of this project.

We owe our deep gratitude to our project guide Prof. Mukesh M. Goswami, who took been interest on our project work and guided us all along till the completion of our project work by providing all the necessary help for developing a good Database System.

We would also like to thank all our lecturers.

Finally, we convoy our acknowledgement to all our friends and family members who directly or indirectly associated with us in the successful completion of the project. We thank one and all.

TABLE OF CONTENTS

1. SYSTEM OVERVIEW

1.1 CURRENT SYSTEM

Airline Management System is engineered to effectively administer every aspect of an election from nominations through managing the flight and passenger database along with their tickets.

With the help of this system, customers can view all different flight's availability with different timings for a particular date and it also allow them to reserve a seat, cancel reservation or modify it. The database is very flexible to insert new data or modify an existence data about a passenger, add or delete columns, retrieve data about specific passenger, remove table with all constraints that make it secure (not to allow anyone to make changes in it).

1.2 OBJECTIVES OF THE PROPOSED SYSTEM

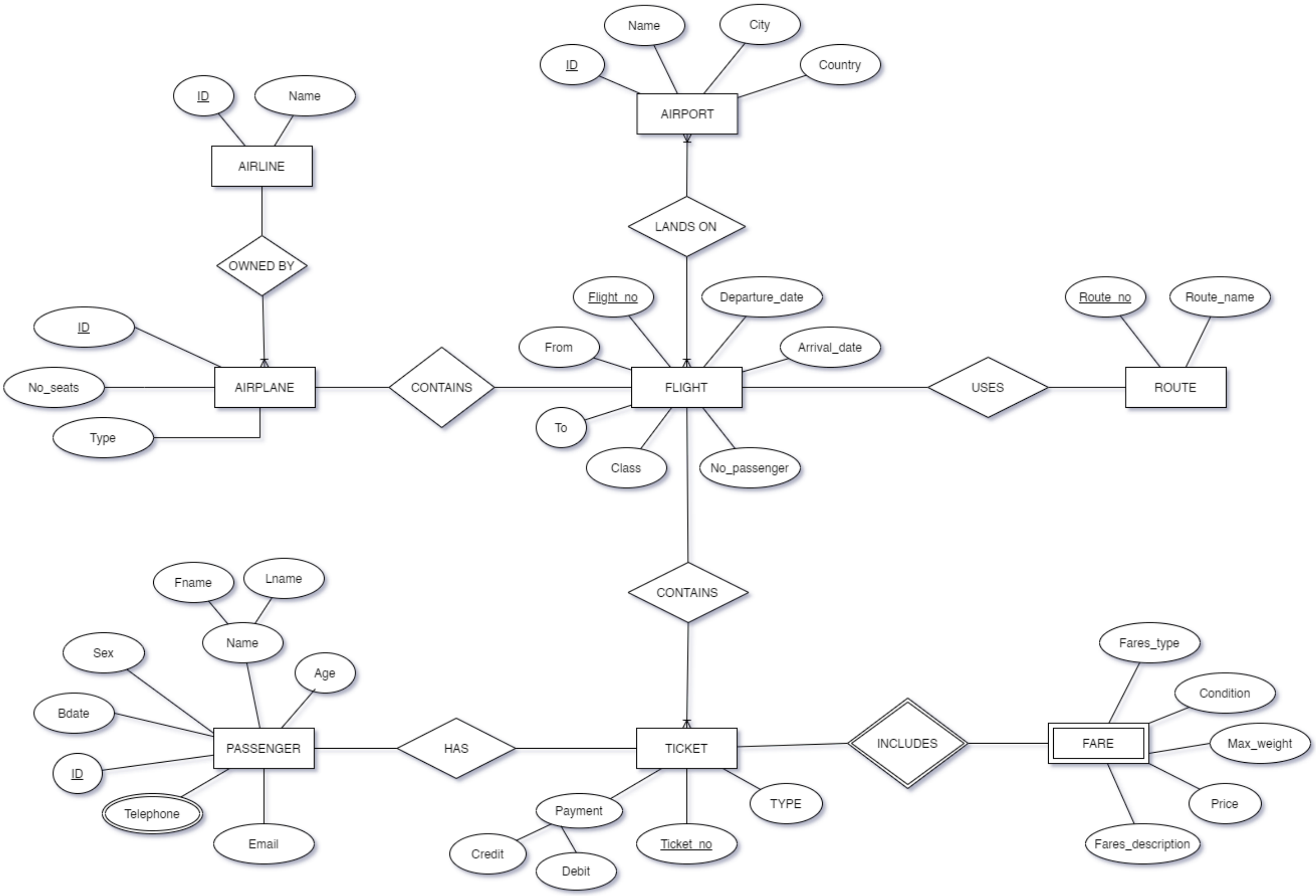
- Automate and facilitate process of managing the passenger and flights.
- Getting the whole data of flight, Airport, Flight routes, passengers as well as their ticket details.
- Get Information like number of passengers getting their ticket details, Routes of Flights, number of available seats or ticket in particular flight, types of ticket and details of discount according to their type

1.3 ADVANTAGES OF THE PROPOSED SYSTEM

Through this system we try to achieve the below objectives

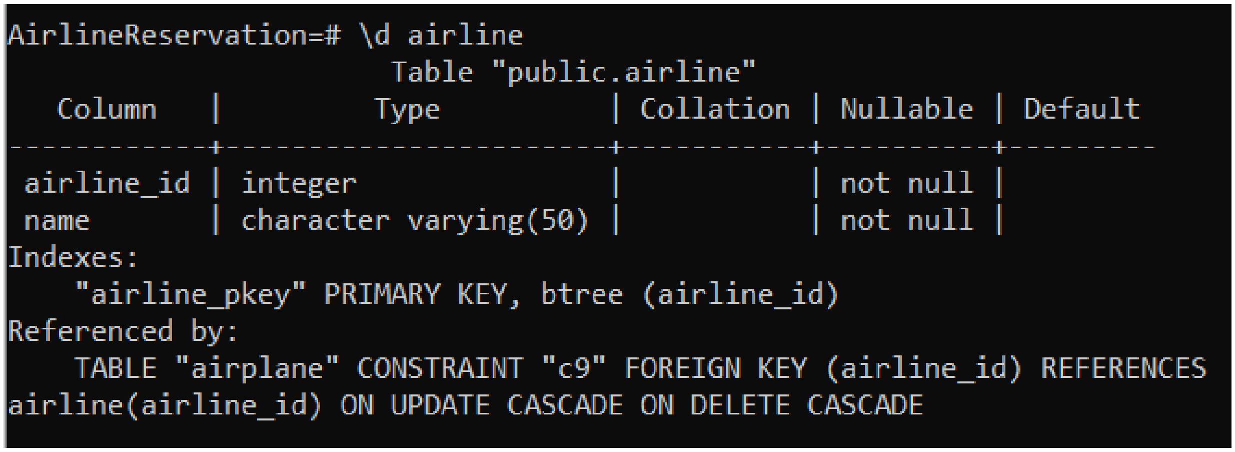
- To boost turnout of passengers in form of more no. of tickets.
- To inform users about their registration status.
- To allow users to change their ticket details.
- To counts number of passengers in particular flight.
- To know number of available seats in a plane.
- To prevent users from getting Invalid tickets.
- To Reduce cost, occur on management.

2.E-R DIAGRAM


Name
City
ID
Country
AIRPORT
ID
Name
AIRLINE
LANDS ON
OWNED BY
Flight_no
Departure_date
Route_no
Route_name
ID
From
Arrival_date
No_seats
AIRPLANE
CONTAINS
FLIGHT
USES
ROUTE
Type
To
Class
No_passenger
Fname
Lname
CONTAINS
Sex
Name
Fares_type
Age
Condition
Bdate
PASSENGER
HAS
TICKET
INCLUDES
FARE
Max_weight
ID
Telephone
Payment
TYPE
Price
Email
Credit
Ticket_no
Fares_description
Debit

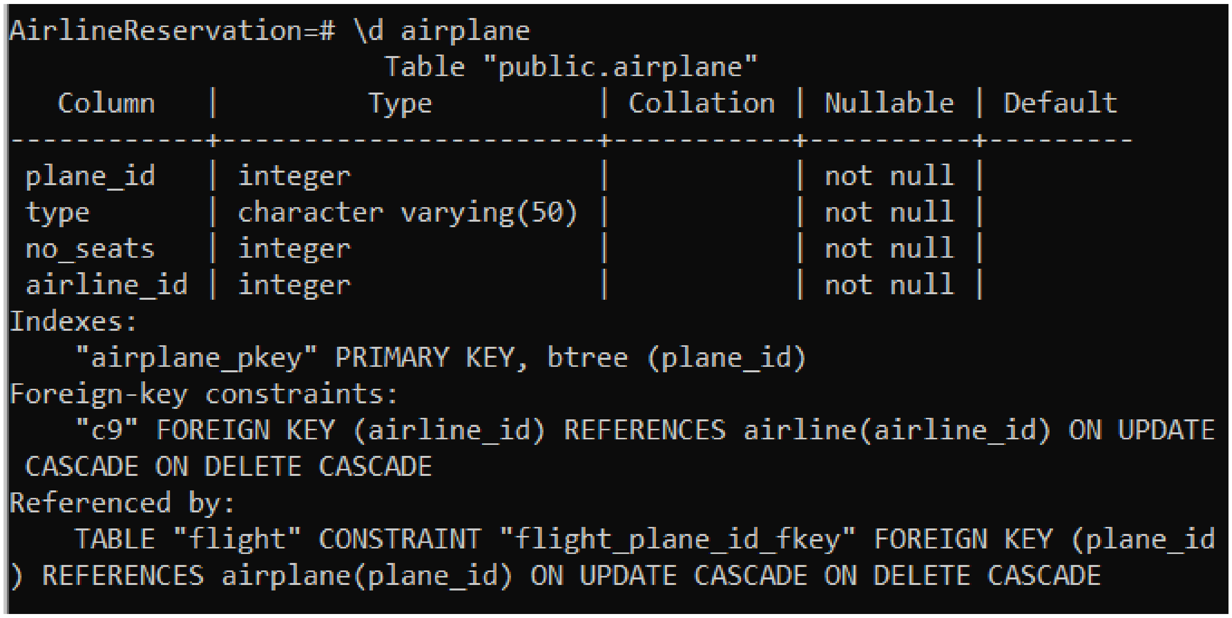
3. DATA DICTIONARY

3.1 airline



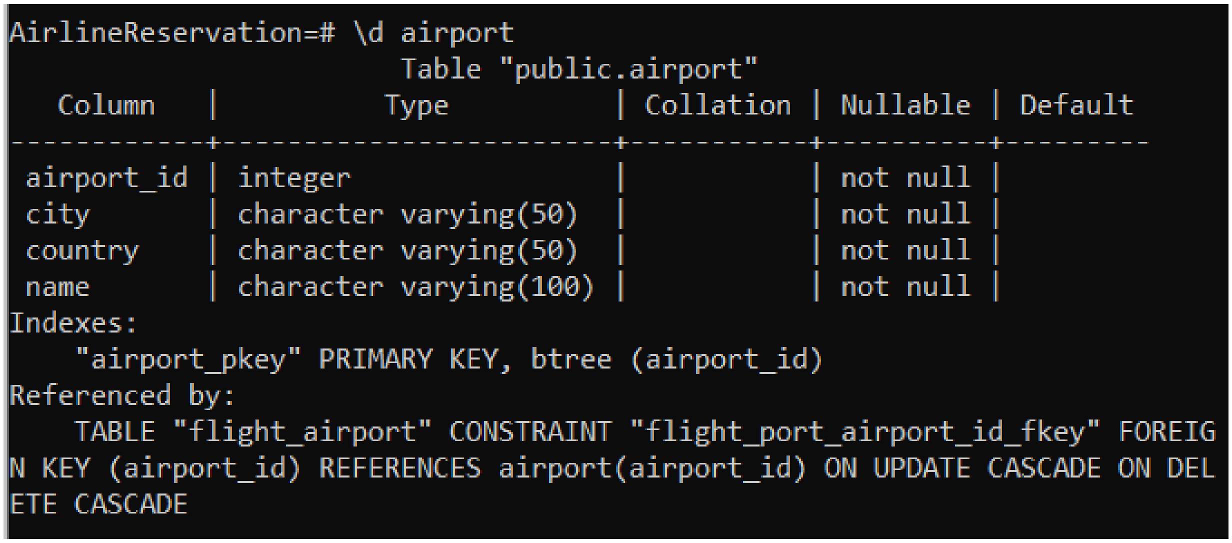
```
AirlineReservation=# \d airline
                      Table "public.airline"
   Column   |         Type          | Collation | Nullable | Default
------------+-----------------------+-----------+----------+---------
 airline_id | integer               |           | not null |
 name       | character varying(50) |           | not null |
Indexes:
    "airline_pkey" PRIMARY KEY, btree (airline_id)
Referenced by:
    TABLE "airplane" CONSTRAINT "c9" FOREIGN KEY (airline_id) REFERENCES
airline(airline_id) ON UPDATE CASCADE ON DELETE CASCADE
```

3.2 airplane



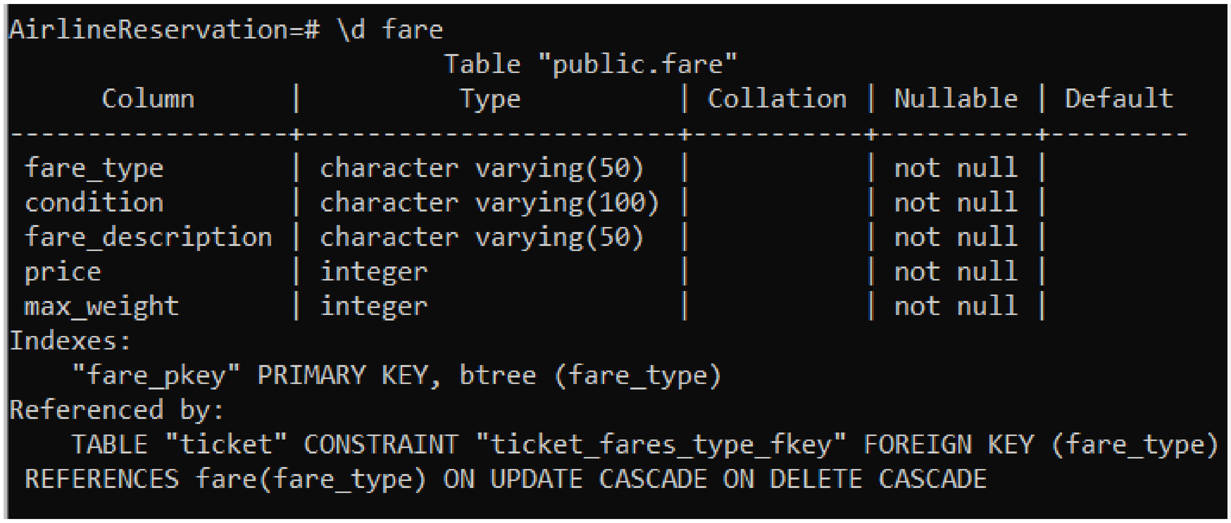
```
AirlineReservation=# \d airplane
                      Table "public.airplane"
   Column   |         Type          | Collation | Nullable | Default
------------+-----------------------+-----------+----------+---------
 plane_id   | integer               |           | not null |
 type       | character varying(50) |           | not null |
 no_seats   | integer               |           | not null |
 airline_id | integer               |           | not null |
Indexes:
    "airplane_pkey" PRIMARY KEY, btree (plane_id)
Foreign-key constraints:
    "c9" FOREIGN KEY (airline_id) REFERENCES airline(airline_id) ON UPDATE
 CASCADE ON DELETE CASCADE
Referenced by:
    TABLE "flight" CONSTRAINT "flight_plane_id_fkey" FOREIGN KEY (plane_id
) REFERENCES airplane(plane_id) ON UPDATE CASCADE ON DELETE CASCADE
```

3.3 airport



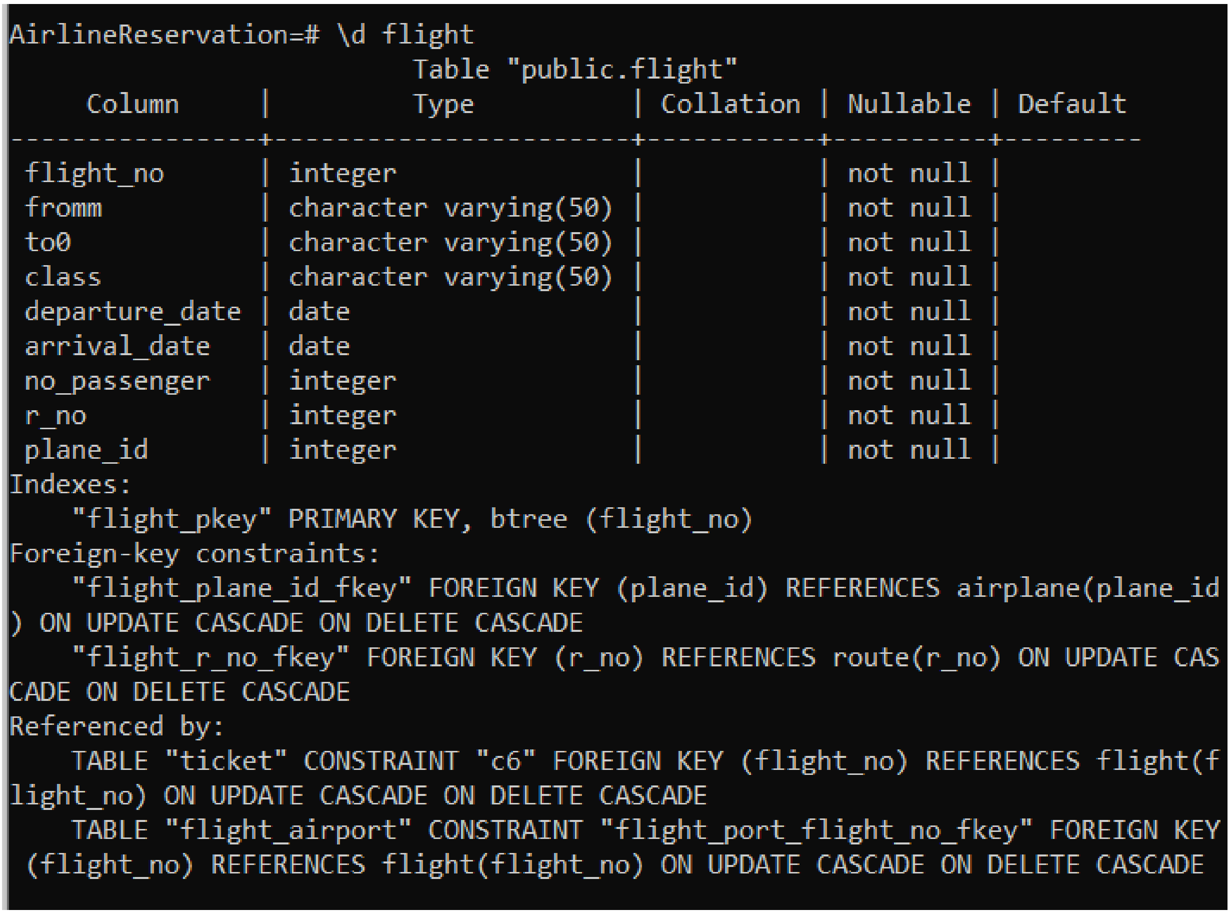
```
AirlineReservation=# \d airport
                        Table "public.airport"
   Column   |          Type          | Collation | Nullable | Default
------------+------------------------+-----------+----------+---------
 airport_id | integer                |           | not null |
 city       | character varying(50)  |           | not null |
 country    | character varying(50)  |           | not null |
 name       | character varying(100) |           | not null |
Indexes:
    "airport_pkey" PRIMARY KEY, btree (airport_id)
Referenced by:
    TABLE "flight_airport" CONSTRAINT "flight_port_airport_id_fkey" FOREIG
N KEY (airport_id) REFERENCES airport(airport_id) ON UPDATE CASCADE ON DEL
ETE CASCADE
```

3.4 fare



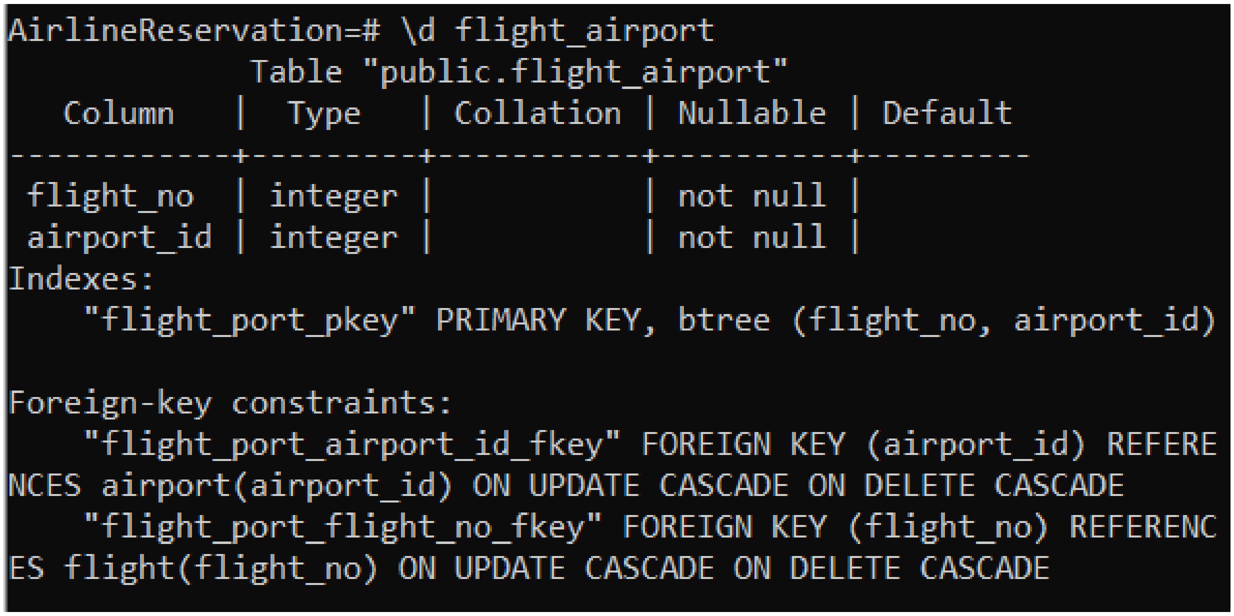
```
AirlineReservation=# \d fare
                             Table "public.fare"
      Column      |          Type          | Collation | Nullable | Default
------------------+------------------------+-----------+----------+---------
 fare_type        | character varying(50)  |           | not null |
 condition        | character varying(100) |           | not null |
 fare_description | character varying(50)  |           | not null |
 price            | integer                |           | not null |
 max_weight       | integer                |           | not null |
Indexes:
    "fare_pkey" PRIMARY KEY, btree (fare_type)
Referenced by:
    TABLE "ticket" CONSTRAINT "ticket_fares_type_fkey" FOREIGN KEY (fare_type)
 REFERENCES fare(fare_type) ON UPDATE CASCADE ON DELETE CASCADE
```

3.5 flight



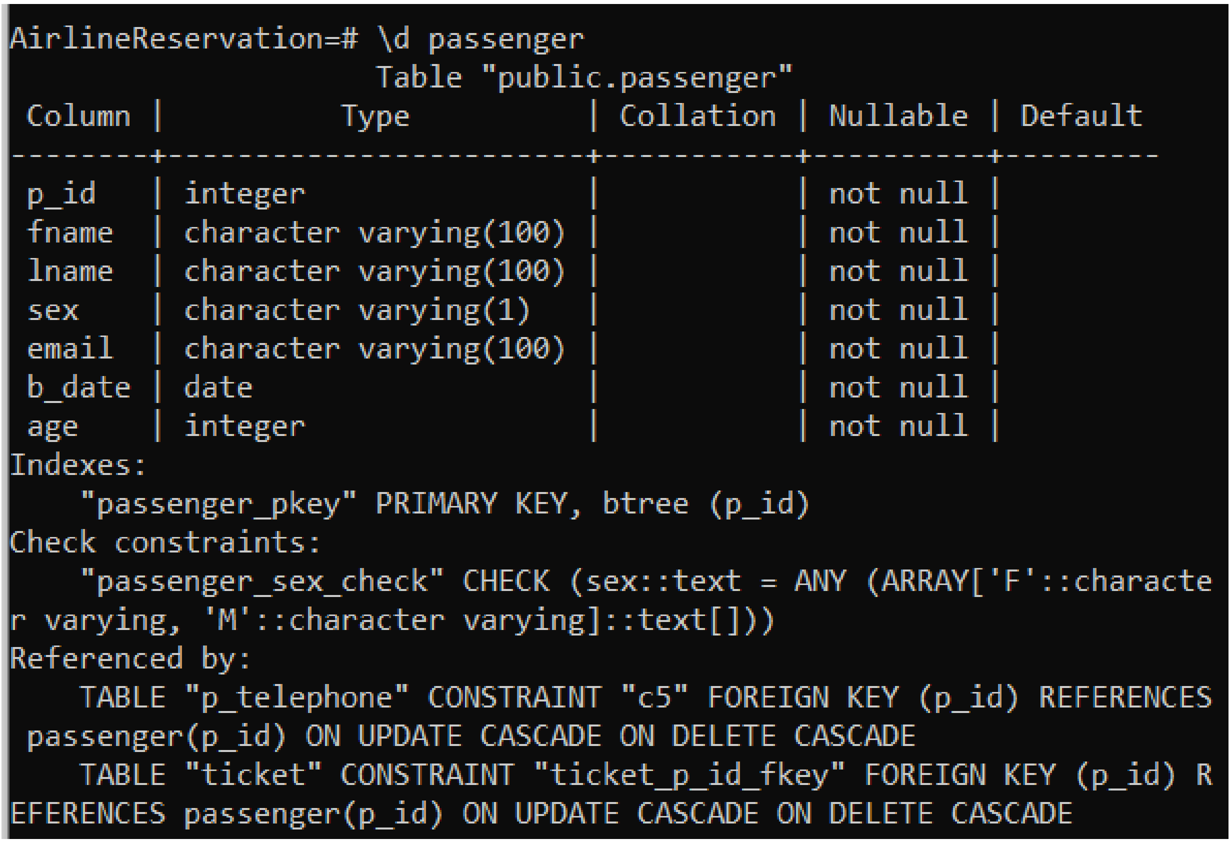
```
AirlineReservation=# \d flight
                          Table "public.flight"
     Column     |          Type          | Collation | Nullable | Default
----------------+------------------------+-----------+----------+---------
 flight_no      | integer                |           | not null |
 fromm          | character varying(50) |           | not null |
 to0            | character varying(50) |           | not null |
 class          | character varying(50) |           | not null |
 departure_date | date                   |           | not null |
 arrival_date   | date                   |           | not null |
 no_passenger   | integer                |           | not null |
 r_no           | integer                |           | not null |
 plane_id       | integer                |           | not null |
Indexes:
    "flight_pkey" PRIMARY KEY, btree (flight_no)
Foreign-key constraints:
    "flight_plane_id_fkey" FOREIGN KEY (plane_id) REFERENCES airplane(plane_id
) ON UPDATE CASCADE ON DELETE CASCADE
    "flight_r_no_fkey" FOREIGN KEY (r_no) REFERENCES route(r_no) ON UPDATE CAS
CADE ON DELETE CASCADE
Referenced by:
    TABLE "ticket" CONSTRAINT "c6" FOREIGN KEY (flight_no) REFERENCES flight(f
light_no) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "flight_airport" CONSTRAINT "flight_port_flight_no_fkey" FOREIGN KEY
 (flight_no) REFERENCES flight(flight_no) ON UPDATE CASCADE ON DELETE CASCADE
```

3.5 flight_airport



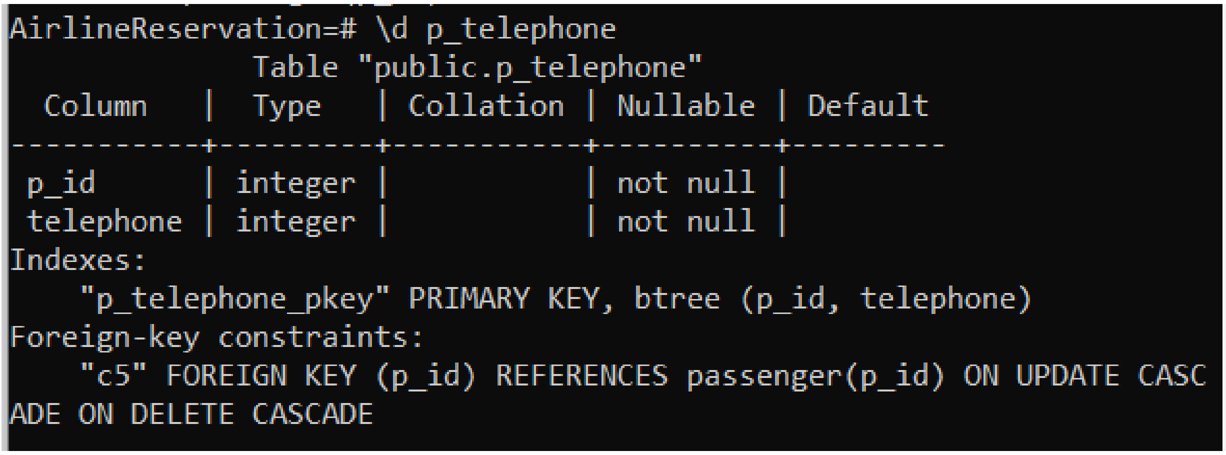
```
AirlineReservation=# \d flight_airport
                Table "public.flight_airport"
   Column   |  Type   | Collation | Nullable | Default
------------+---------+-----------+----------+---------
 flight_no  | integer |           | not null |
 airport_id | integer |           | not null |
Indexes:
    "flight_port_pkey" PRIMARY KEY, btree (flight_no, airport_id)

Foreign-key constraints:
    "flight_port_airport_id_fkey" FOREIGN KEY (airport_id) REFERE
NCES airport(airport_id) ON UPDATE CASCADE ON DELETE CASCADE
    "flight_port_flight_no_fkey" FOREIGN KEY (flight_no) REFERENC
ES flight(flight_no) ON UPDATE CASCADE ON DELETE CASCADE
```

3.6 passenger



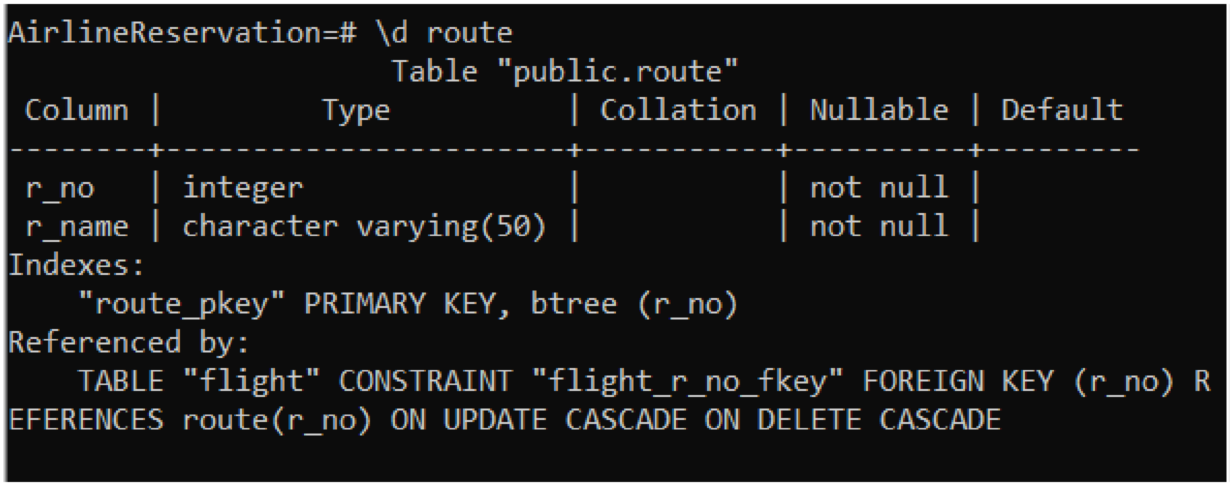
```
AirlineReservation=# \d passenger
                     Table "public.passenger"
 Column |          Type          | Collation | Nullable | Default
--------+------------------------+-----------+----------+---------
 p_id   | integer                |           | not null |
 fname  | character varying(100) |           | not null |
 lname  | character varying(100) |           | not null |
 sex    | character varying(1)   |           | not null |
 email  | character varying(100) |           | not null |
 b_date | date                   |           | not null |
 age    | integer                |           | not null |
Indexes:
    "passenger_pkey" PRIMARY KEY, btree (p_id)
Check constraints:
    "passenger_sex_check" CHECK (sex::text = ANY (ARRAY['F'::characte
r varying, 'M'::character varying]::text[]))
Referenced by:
    TABLE "p_telephone" CONSTRAINT "c5" FOREIGN KEY (p_id) REFERENCES
 passenger(p_id) ON UPDATE CASCADE ON DELETE CASCADE
    TABLE "ticket" CONSTRAINT "ticket_p_id_fkey" FOREIGN KEY (p_id) R
EFERENCES passenger(p_id) ON UPDATE CASCADE ON DELETE CASCADE
```

3.7 p_telephone



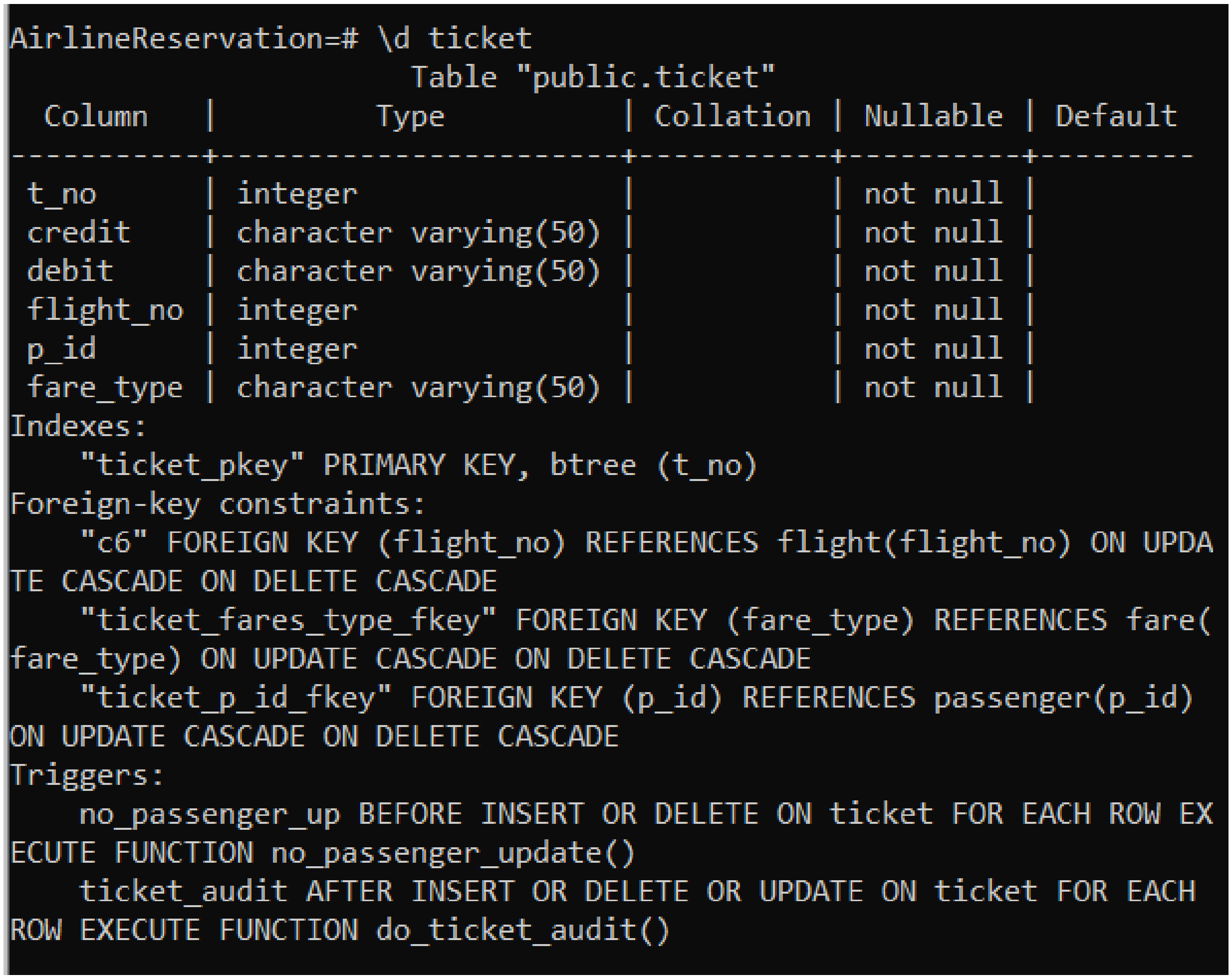
```
AirlineReservation=# \d p_telephone
              Table "public.p_telephone"
  Column   |  Type   | Collation | Nullable | Default
-----------+---------+-----------+----------+---------
 p_id      | integer |           | not null |
 telephone | integer |           | not null |
Indexes:
    "p_telephone_pkey" PRIMARY KEY, btree (p_id, telephone)
Foreign-key constraints:
    "c5" FOREIGN KEY (p_id) REFERENCES passenger(p_id) ON UPDATE CASC
ADE ON DELETE CASCADE
```

3.8 route



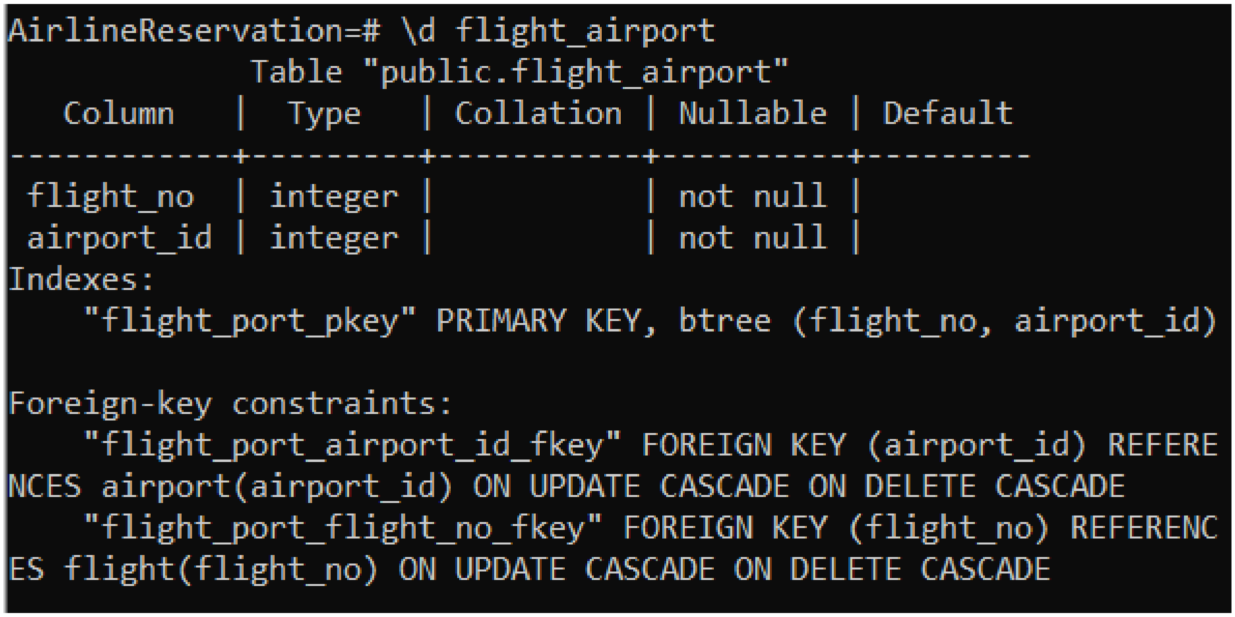
```
AirlineReservation=# \d route
                      Table "public.route"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 r_no   | integer               |           | not null |
 r_name | character varying(50) |           | not null |
Indexes:
    "route_pkey" PRIMARY KEY, btree (r_no)
Referenced by:
    TABLE "flight" CONSTRAINT "flight_r_no_fkey" FOREIGN KEY (r_no) R
EFERENCES route(r_no) ON UPDATE CASCADE ON DELETE CASCADE
```

3.9 Ticket



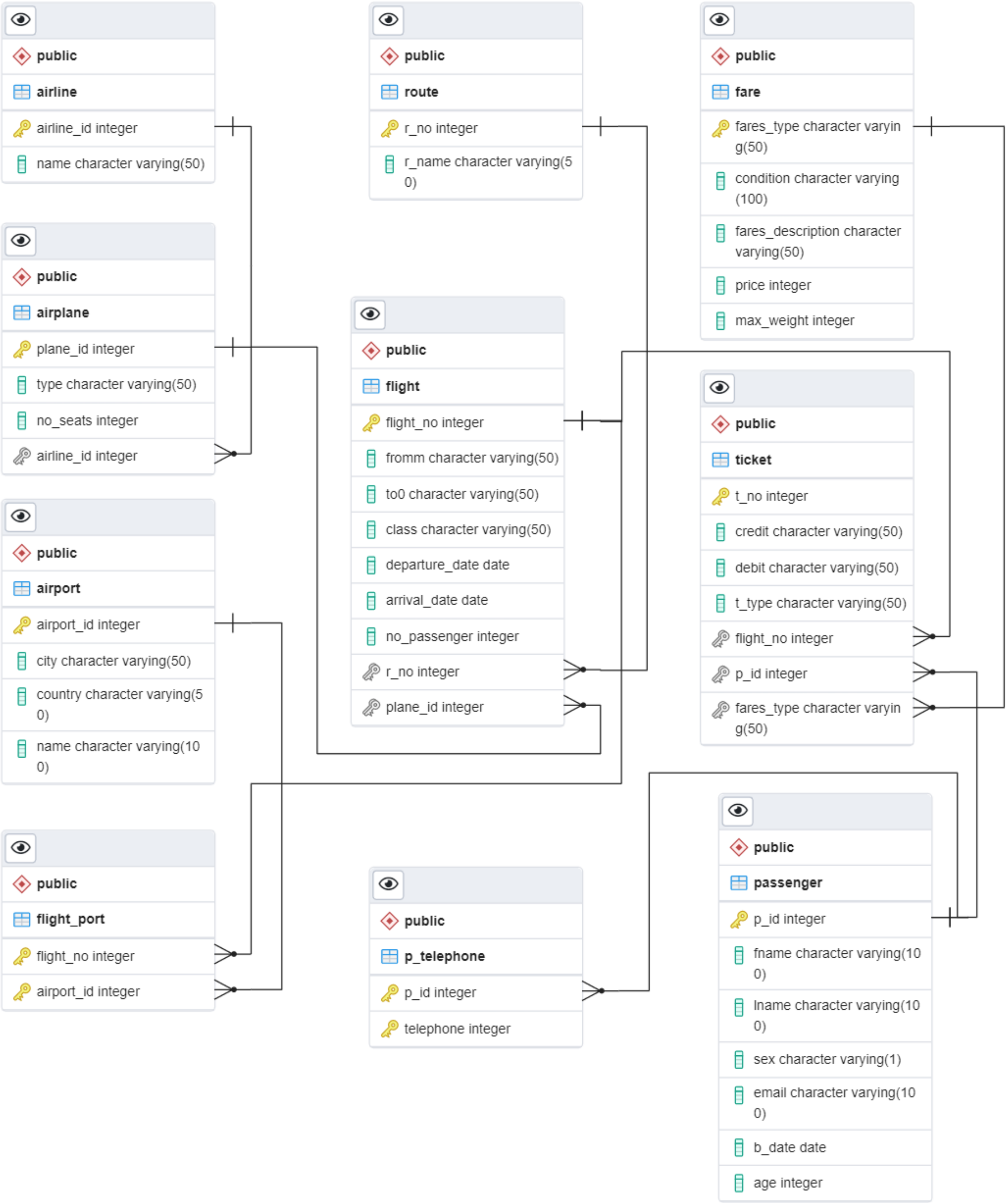
```
AirlineReservation=# \d ticket
                        Table "public.ticket"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 t_no      | integer               |           | not null |
 credit    | character varying(50) |           | not null |
 debit     | character varying(50) |           | not null |
 flight_no | integer               |           | not null |
 p_id      | integer               |           | not null |
 fare_type | character varying(50) |           | not null |
Indexes:
    "ticket_pkey" PRIMARY KEY, btree (t_no)
Foreign-key constraints:
    "c6" FOREIGN KEY (flight_no) REFERENCES flight(flight_no) ON UPDA
TE CASCADE ON DELETE CASCADE
    "ticket_fares_type_fkey" FOREIGN KEY (fare_type) REFERENCES fare(
fare_type) ON UPDATE CASCADE ON DELETE CASCADE
    "ticket_p_id_fkey" FOREIGN KEY (p_id) REFERENCES passenger(p_id)
ON UPDATE CASCADE ON DELETE CASCADE
Triggers:
    no_passenger_up BEFORE INSERT OR DELETE ON ticket FOR EACH ROW EX
ECUTE FUNCTION no_passenger_update()
    ticket_audit AFTER INSERT OR DELETE OR UPDATE ON ticket FOR EACH
ROW EXECUTE FUNCTION do_ticket_audit()
```

3.10flight_airport



```
AirlineReservation=# \d flight_airport
             Table "public.flight_airport"
   Column   |  Type   | Collation | Nullable | Default
------------+---------+-----------+----------+---------
 flight_no  | integer |           | not null |
 airport_id | integer |           | not null |
Indexes:
    "flight_port_pkey" PRIMARY KEY, btree (flight_no, airport_id)

Foreign-key constraints:
    "flight_port_airport_id_fkey" FOREIGN KEY (airport_id) REFERE
NCES airport(airport_id) ON UPDATE CASCADE ON DELETE CASCADE
    "flight_port_flight_no_fkey" FOREIGN KEY (flight_no) REFERENC
ES flight(flight_no) ON UPDATE CASCADE ON DELETE CASCADE
```

4. SCHEMA DIAGRAM


public
airline
airline_id integer
name character varying(50)
public
route
r_no integer
r_name character varying(50)
public
fare
fares_type character varying(50)
condition character varying(100)
fares_description character varying(50)
price integer
max_weight integer
public
airplane
plane_id integer
type character varying(50)
no_seats integer
airline_id integer
public
flight
flight_no integer
fromm character varying(50)
to0 character varying(50)
class character varying(50)
departure_date date
arrival_date date
no_passenger integer
r_no integer
plane_id integer
public
ticket
t_no integer
credit character varying(50)
debit character varying(50)
t_type character varying(50)
flight_no integer
p_id integer
fares_type character varying(50)
public
airport
airport_id integer
city character varying(50)
country character varying(50)
name character varying(100)
public
flight_port
flight_no integer
airport_id integer
public
p_telephone
p_id integer
telephone integer
public
passenger
p_id integer
fname character varying(100)
lname character varying(100)
sex character varying(1)
email character varying(100)
b_date date
age integer

5. DATABASE IMPLEMENTATION

5.1 CREATE SCHEMA

5.1.1 passenger

```
create table passenger
(
p_id int primary key  ,
fname varchar (100) NOT NULL,
lname varchar (100) NOT NULL,
sex varchar (1) CHECK(sex in(‘F’,’M’)) NOT NULL,
email varchar (100) NOT NULL,
b_date date NOT NULL,
age int NOT NULL,
);
```

5.1.2 ticket

```
create table ticket
(
t_no int primary key ,
credit varchar(50) NOT NULL,
debit varchar(50) NOT NULL,
flight_no int NOT NULL,
p_id int NOT NULL,
fare_type varchar(50) NOT NULL
)
```

5.1.3 fare

```
create table fare
(
fare_type varchar (50) NOT NULL,
price float NOT NULL,
condition varchar (100) NOT NULL,
max_weight int NOT NULL,
fare_description varchar (50) NOT NULL
)
```

5.1.4 flight

```
create table flight
(
flight_no int primary key ,
fromm varchar (50) NOT NULL,
to0 varchar (50) NOT NULL,
class varchar (50) NOT NULL,
arrival_date date NOT NULL,
departure_date date NOT NULL,
no_passenger int NOT NULL,
r_no int NOT NULL,
plane_id NOT NULL
)
```

5.1.5 airplane

```
create table airplane
(
plane_id int primary key,
type varchar (50) NOT NULL,
no_seats int NOT NULL,
airline_id int NOT NULL
)
```

5.1.6 airline

```
create table airline
(
airline_id int primary key,
name varchar (50) NOT NULL
)
```

5.1.7 airport

```
create table airport
(
airport_id int primary key,
city varchar (50) NOT NULL,
country varchar (50) NOT NULL,
name varchar (100) NOT NULL
)
```

5.1.8 route

```
create table route
(
r_no int primary key,
r_name varchar (50) NOT NULL
)
```

5.1.9 p_Telephone

```
create table p_Telephone
(
p_id int,
telephone int
constraint c1 primary key (p_id,telephone)
)
```

5.1.10 flight_airport

```
CREATE TABLE flight_airport
(
   flight_no integer NOT NULL,
   airport_id integer NOT NULL,
   PRIMARY KEY (flight_no, airport_id)
)
```

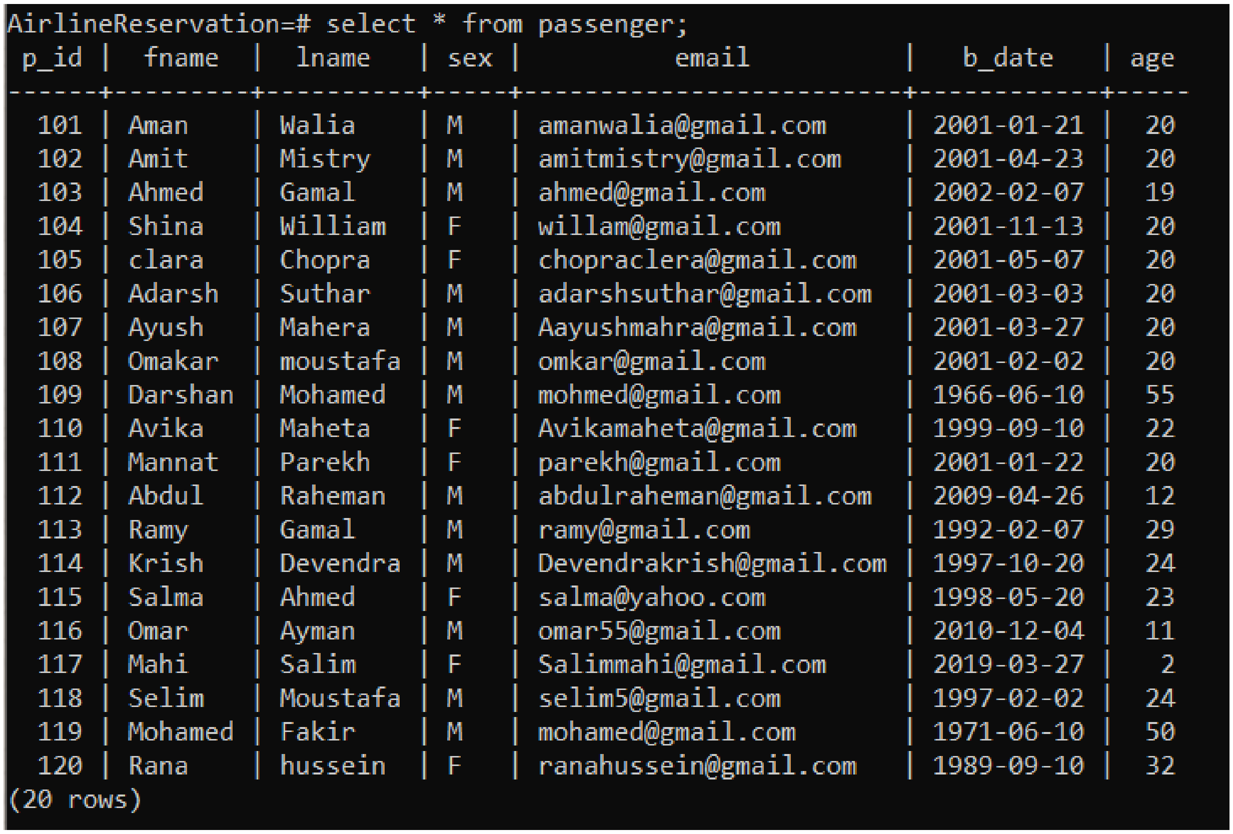
5.1.11 Constraints

```
alter table airplane add foreign key(airline_id) references airline(airline_id) on update
    cascade on delete cascade;
alter table p_telephone add foreign key(p_id) references passenger(p_id) on update
    cascade on delete cascade;
alter table ticket add foreign key(p_id) references passenger(p_id) on update cascade on
    delete cascade;
alter table ticket add foreign key(fare_type) references fare(fare_type) on update cascade
    on delete cascade;
alter table ticket add foreign key(flight_no) references flight(flight_no) on update
    cascade on delete cascade;
alter table flight add foreign key(r_no) references route(r_no) on update cascade on
    delete cascade;
alter table flight add foreign key(plane_id) references airplane(plane_id) on update
    cascade on delete cascade;

alter table flight_airport add foreign key(flight_no) references flight(flight_no) on update
    cascade on delete cascade;
    alter table flight_airport add foreign key(airport_id) references airport(airport_id) on
    update cascade on delete cascade;
```

5.2 INSERT DATA VALUE

5.2.1 passenger

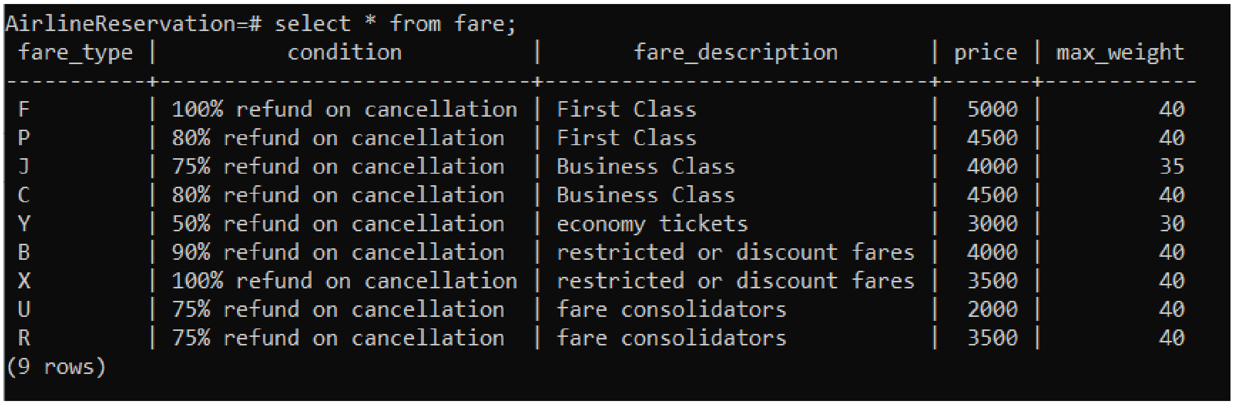
```
insert into passenger(fname,lname,p_id,sex,b_date,email,age,ticket_no) values
('Aman','Walia',101,'M','2001-1-21','amanwalia@gmail.com',35, 131101),
('Amit','Mistry',102,'M','2001-4-23','amitmistry@gmail.com',34,705001),
('Ahmed','Gamal',103,'M','2002-2-7','ahmed@gmail.com',22,120201),
('Shina','William',104,'F','2001-11-13','willam@gmail.com',56,702002),
('clara','Chopra',105,'F','2001-5-7','chopraclera@gmail.com',39,210101),
('Adarsh','Suthar',106,'M','2001-3-3','adarshsuthar@gmail.com',42,280312),
('Ayush','Mahera',107,'M','2001-3-27','Aayushmahra@gmail.com',52,100999),
('Omakar','moustafa',108,'M','2001-2-2','omkar@gmail.com',26,200594),
('Darshan','Mohamed',109,'M','1966-6-10','mohmed@gmail.com',35,270310),
('Avika','Maheta',110,'F','1999-9-10','Avikamaheta@gmail.com',69,202001),
('Mannat','Parekh',111,'F','2001-1-22','parekh@gmail.com',53,131102),
('Abdul','Raheman',112,'M','2009-4-26','abdulraheman@gmail.com',75,705002),
('Ramy','Gamal',113,'M','1992-2-7','ramy@gmail.com',25,120202),
('Krish','Devendra',114,'M','1997-10-20','Devendrakrish@gmail.com',36,702003),
('Salma','Ahmed',115,'F','1998-5-20','salma@yahoo.com',58,210102),
('Omar','Ayman',116,'M','2010-12-4','omar55@gmail.com',23,280313),
('Mahi','Salim',117,'F','2019-3-27','Salimmahi@gmail.com',22,100990),
('Selim','Moustafa',118,'M','1997-2-2','selim5@gmail.com',18,200595),
('Mohamed','Fakir',119,'M','1971-6-10','mohamed@gmail.com',72,270315),
('Rana','hussein',120,'F','1989-9-10','ranahussein@gmail.com',55,202005);
```



```
AirlineReservation=# select * from passenger;
 p_id |  fname  |  lname   | sex |          email          |   b_date   | age
------+---------+----------+-----+-------------------------+------------+-----
  101 | Aman    | Walia    | M   | amanwalia@gmail.com     | 2001-01-21 |  20
  102 | Amit    | Mistry   | M   | amitmistry@gmail.com    | 2001-04-23 |  20
  103 | Ahmed   | Gamal    | M   | ahmed@gmail.com         | 2002-02-07 |  19
  104 | Shina   | William  | F   | willam@gmail.com        | 2001-11-13 |  20
  105 | clara   | Chopra   | F   | chopraclera@gmail.com   | 2001-05-07 |  20
  106 | Adarsh  | Suthar   | M   | adarshsuthar@gmail.com  | 2001-03-03 |  20
  107 | Ayush   | Mahera   | M   | Aayushmahra@gmail.com   | 2001-03-27 |  20
  108 | Omakar  | moustafa | M   | omkar@gmail.com         | 2001-02-02 |  20
  109 | Darshan | Mohamed  | M   | mohmed@gmail.com        | 1966-06-10 |  55
  110 | Avika   | Maheta   | F   | Avikamaheta@gmail.com   | 1999-09-10 |  22
  111 | Mannat  | Parekh   | F   | parekh@gmail.com        | 2001-01-22 |  20
  112 | Abdul   | Raheman  | M   | abdulraheman@gmail.com  | 2009-04-26 |  12
  113 | Ramy    | Gamal    | M   | ramy@gmail.com          | 1992-02-07 |  29
  114 | Krish   | Devendra | M   | Devendrakrish@gmail.com | 1997-10-20 |  24
  115 | Salma   | Ahmed    | F   | salma@yahoo.com         | 1998-05-20 |  23
  116 | Omar    | Ayman    | M   | omar55@gmail.com        | 2010-12-04 |  11
  117 | Mahi    | Salim    | F   | Salimmahi@gmail.com     | 2019-03-27 |   2
  118 | Selim   | Moustafa | M   | selim5@gmail.com        | 1997-02-02 |  24
  119 | Mohamed | Fakir    | M   | mohamed@gmail.com       | 1971-06-10 |  50
  120 | Rana    | hussein  | F   | ranahussein@gmail.com   | 1989-09-10 |  32
(20 rows)
```

5.2.2 fare

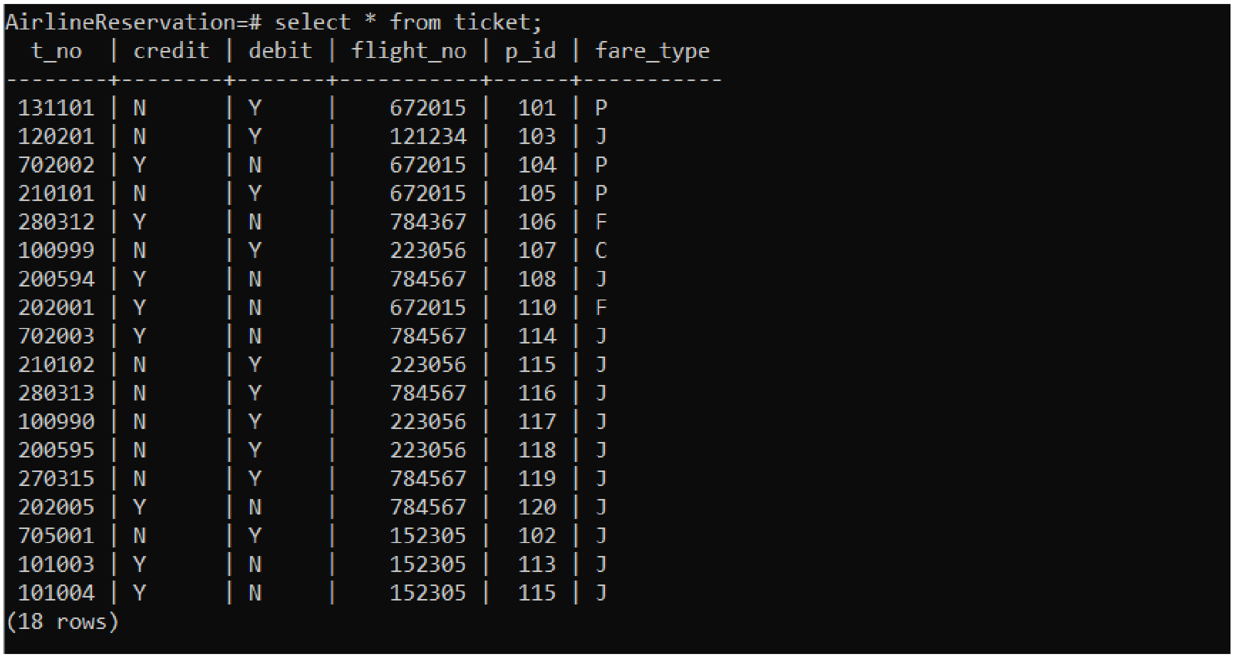
```
insert into fare(fare_type,fare_description,condition,price,max_weight) values
('F','First Class','100% refund on cancellation',5000,40),
('P','First Class','80% refund on cancellation',4500,40),
('J','Business Class','75% refund on cancellation',4000,35),
('C','Business Class','80% refund on cancellation',4500,40),
('Y','economy tickets','50% refund on cancellation',3000,30),
('B','restricted or discount fares','90% refund on cancellation',4000,40),
('X','restricted or discount fares','100% refund on cancellation',3500,40),
('U','fare consolidators','75% refund on cancellation',2000,40),
('R','fare consolidators','75% refund on cancellation',3500,40);
```



```
AirlineReservation=# select * from fare;
 fare_type |          condition          |       fare_description       | price | max_weight
-----------+-----------------------------+------------------------------+-------+------------
 F         | 100% refund on cancellation | First Class                  |  5000 |         40
 P         | 80% refund on cancellation  | First Class                  |  4500 |         40
 J         | 75% refund on cancellation  | Business Class               |  4000 |         35
 C         | 80% refund on cancellation  | Business Class               |  4500 |         40
 Y         | 50% refund on cancellation  | economy tickets              |  3000 |         30
 B         | 90% refund on cancellation  | restricted or discount fares |  4000 |         40
 X         | 100% refund on cancellation | restricted or discount fares |  3500 |         40
 U         | 75% refund on cancellation  | fare consolidators           |  2000 |         40
 R         | 75% refund on cancellation  | fare consolidators           |  3500 |         40
(9 rows)
```

5.2.3 ticket

insert into ticket(t_no,flight_no,debit,credit,p_id,fare_type) values
('131101',672015,'Y','N',101,'P'),
('705001',152305,'N','Y',102,'J'),
('120201',121234,'Y','N',103,'J'),
('702002',672015,'N','Y',104,'P'),
('210101',672015,'Y','N',105,'P'),
('280312',784367,'N','Y',106,'F'),
('100999',223056,'Y','N',107,'C'),
('200594',784567,'N','Y',108,'J'),
('202001',672015,'N','Y',110,'F'),
('702003',784567,'N','Y',114,'J'),
('210102',223056,'Y','N',115,'J'),
('280313',784567,'Y','N',116,'J'),
('100990',223056,'Y','N',117,'J'),
('200595',223056,'Y','N',118,'J'),
('270315',784567,'Y','N',119,'J'),
('202005',784567,'N','Y',120,'J');

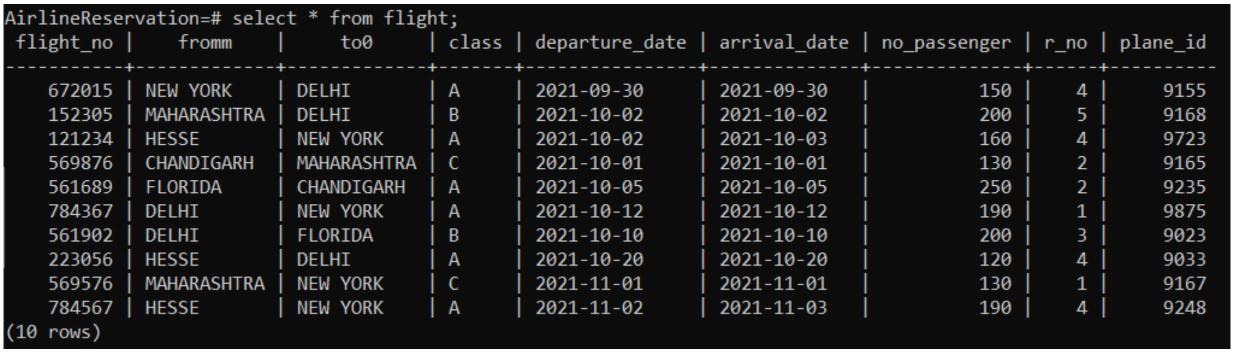


```
AirlineReservation=# select * from ticket;
  t_no  | credit | debit | flight_no | p_id | fare_type
--------+--------+-------+-----------+------+-----------
 131101 | N      | Y     |    672015 |  101 | P
 120201 | N      | Y     |    121234 |  103 | J
 702002 | Y      | N     |    672015 |  104 | P
 210101 | N      | Y     |    672015 |  105 | P
 280312 | Y      | N     |    784367 |  106 | F
 100999 | N      | Y     |    223056 |  107 | C
 200594 | Y      | N     |    784567 |  108 | J
 202001 | Y      | N     |    672015 |  110 | F
 702003 | Y      | N     |    784567 |  114 | J
 210102 | N      | Y     |    223056 |  115 | J
 280313 | N      | Y     |    784567 |  116 | J
 100990 | N      | Y     |    223056 |  117 | J
 200595 | N      | Y     |    223056 |  118 | J
 270315 | N      | Y     |    784567 |  119 | J
 202005 | Y      | N     |    784567 |  120 | J
 705001 | N      | Y     |    152305 |  102 | J
 101003 | Y      | N     |    152305 |  113 | J
 101004 | Y      | N     |    152305 |  115 | J
(18 rows)
```

5.2.4 flight

insert into flight(flight_no, fromm, to0, class, Arrival_date,Departure_date
,no_passenger,plane_id,r_no) values
('672015','NEW YORK','DELHI','A','2021-09-30','2021-09-30',150,9155,4),
('152305','MAHARASHTRA','DELHI','B','2021-10-2','2021-10-2',200,9168,5),
('121234','HESSE','NEW YORK','A','2021-10-3','2021-10-2',160,9723,4),
('569876','CHANDIGARH','MAHARASHTRA','C','2021-10-1','2021-10-1',130,9165,2),

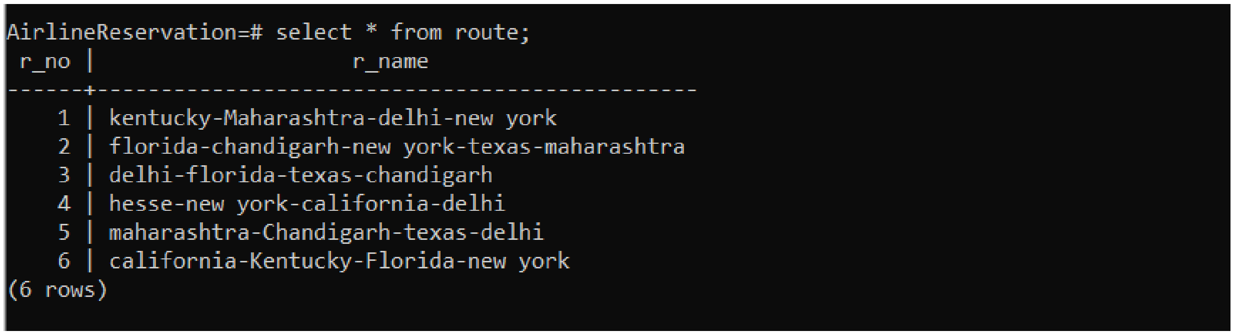
('561689','FLORIDA','CHANDIGARH','A','2021-10-5','2021-10-5',250,9235,2),
('784367','DELHI','NEW YORK','A','2021-10-12','2021-10-12',190,9875,1),
('561902','DELHI','FLORIDA','B','2021-10-10','2021-10-10',200,9023,3),
('223056','HESSE','DELHI','A','2021-10-20','2021-10-20',120,9033,4),
('569576','MAHARASHTRA','NEW YORK','C','2021-11-1','2021-11-1',130,9167,1),
('784567','HESSE','NEW YORK','A','2021-11-3','2021-11-2',190,9248,4);



```
AirlineReservation=# select * from flight;
 flight_no |    fromm    |     to0     | class | departure_date | arrival_date | no_passenger | r_no | plane_id
-----------+-------------+-------------+-------+----------------+--------------+--------------+------+----------
    672015 | NEW YORK    | DELHI       | A     | 2021-09-30     | 2021-09-30   |          150 |    4 |     9155
    152305 | MAHARASHTRA | DELHI       | B     | 2021-10-02     | 2021-10-02   |          200 |    5 |     9168
    121234 | HESSE       | NEW YORK    | A     | 2021-10-02     | 2021-10-03   |          160 |    4 |     9723
    569876 | CHANDIGARH  | MAHARASHTRA | C     | 2021-10-01     | 2021-10-01   |          130 |    2 |     9165
    561689 | FLORIDA     | CHANDIGARH  | A     | 2021-10-05     | 2021-10-05   |          250 |    2 |     9235
    784367 | DELHI       | NEW YORK    | A     | 2021-10-12     | 2021-10-12   |          190 |    1 |     9875
    561902 | DELHI       | FLORIDA     | B     | 2021-10-10     | 2021-10-10   |          200 |    3 |     9023
    223056 | HESSE       | DELHI       | A     | 2021-10-20     | 2021-10-20   |          120 |    4 |     9033
    569576 | MAHARASHTRA | NEW YORK    | C     | 2021-11-01     | 2021-11-01   |          130 |    1 |     9167
    784567 | HESSE       | NEW YORK    | A     | 2021-11-02     | 2021-11-03   |          190 |    4 |     9248
(10 rows)
```

5.2.5 route

insert into route(r_no,r_name) values
(1,'kentucky-Maharashtra-delhi-new york'),
(2,'florida-chandigarh-new york-texas-maharashtra'),
(3,'delhi-florida-texas-chandigarh'),
(4,'hesse-new york-california-delhi'),
(5,'maharashtra-Chandigarh-texas-delhi'),
(6,'california-Kentucky-Florida-new york');

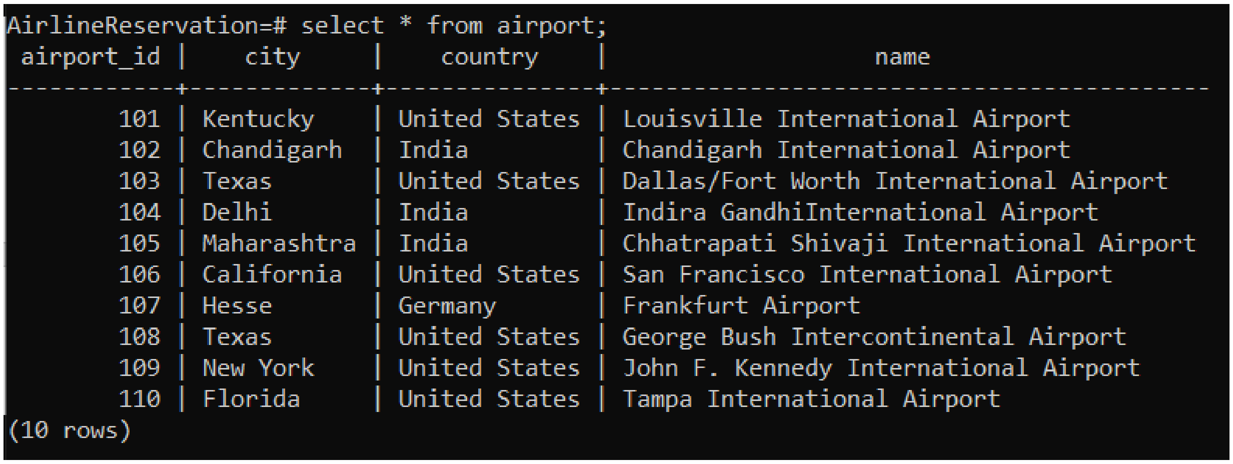


```
AirlineReservation=# select * from route;
 r_no |                    r_name
------+-----------------------------------------------
    1 | kentucky-Maharashtra-delhi-new york
    2 | florida-chandigarh-new york-texas-maharashtra
    3 | delhi-florida-texas-chandigarh
    4 | hesse-new york-california-delhi
    5 | maharashtra-Chandigarh-texas-delhi
    6 | california-Kentucky-Florida-new york
(6 rows)
```

5.2.6 airport

insert into airport(name,city,country,airport_id) VALUES
('Louisville International Airport','Kentucky','United States',101),
('Chandigarh International Airport','Chandigarh','India',102),
('Dallas/Fort Worth International Airport','Texas','United States',103),
('Indira GandhiInternational Airport','Delhi','India',104),
('Chhatrapati Shivaji International Airport','Maharashtra','India',105),
('San Francisco International Airport','California', 'United States',106),
('Frankfurt Airport','Hesse','Germany',107),

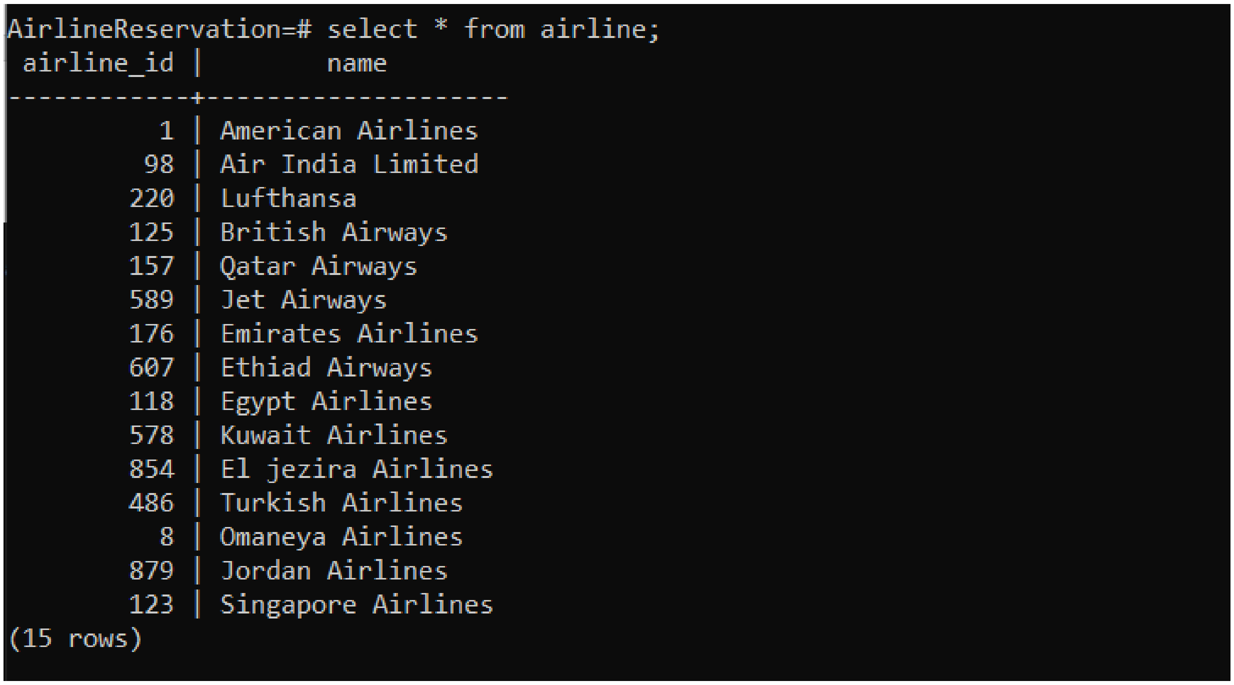
('George Bush Intercontinental Airport','Texas','United States',108),
('John F. Kennedy International Airport','New York','United States',109),
('Tampa International Airport','Florida', 'United States',110);



```
AirlineReservation=# select * from airport;
 airport_id |    city     |    country    |                  name
------------+-------------+---------------+-----------------------------------------
        101 | Kentucky    | United States | Louisville International Airport
        102 | Chandigarh  | India         | Chandigarh International Airport
        103 | Texas       | United States | Dallas/Fort Worth International Airport
        104 | Delhi       | India         | Indira GandhiInternational Airport
        105 | Maharashtra | India         | Chhatrapati Shivaji International Airport
        106 | California  | United States | San Francisco International Airport
        107 | Hesse       | Germany       | Frankfurt Airport
        108 | Texas       | United States | George Bush Intercontinental Airport
        109 | New York    | United States | John F. Kennedy International Airport
        110 | Florida     | United States | Tampa International Airport
(10 rows)
```

5.2.7 airline

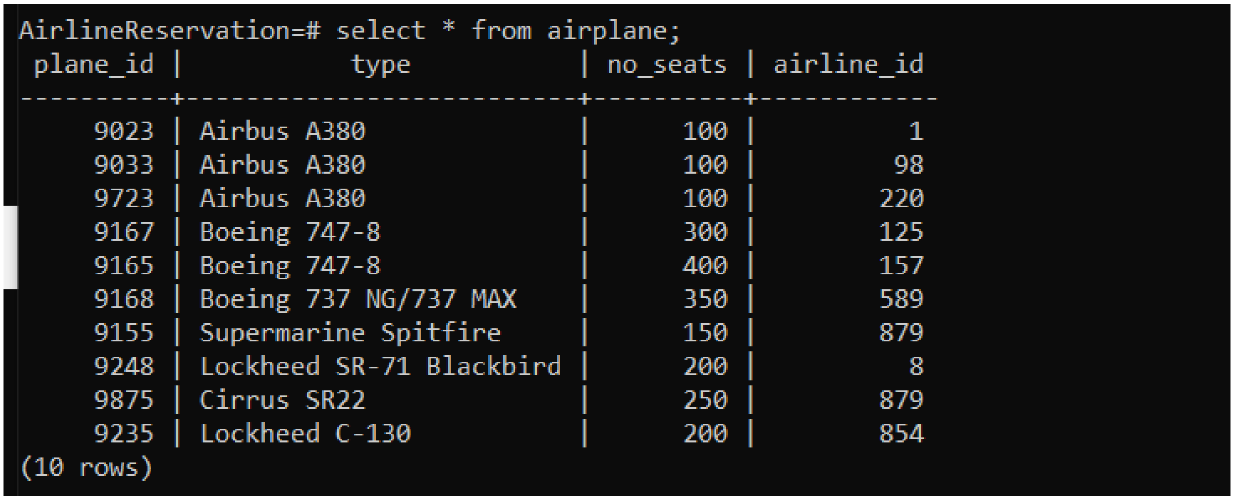
insert into airline (name,airline_id) values
('American Airlines',001),
('Air India Limited',098),
('Lufthansa',220),
('British Airways',125),
('Qatar Airways',157),
('Jet Airways',589),
('Emirates Airlines',176),
('Ethiad Airways',607),
('Egypt Airlines',118),
('Kuwait Airlines',578),
('El jezira Airlines',854),
('Turkish Airlines',486),
('Omaneya Airlines',008),
('Jordan Airlines',879),
('Singapore Airlines',123);



```
AirlineReservation=# select * from airline;
 airline_id |        name
------------+--------------------
          1 | American Airlines
         98 | Air India Limited
        220 | Lufthansa
        125 | British Airways
        157 | Qatar Airways
        589 | Jet Airways
        176 | Emirates Airlines
        607 | Ethiad Airways
        118 | Egypt Airlines
        578 | Kuwait Airlines
        854 | El jezira Airlines
        486 | Turkish Airlines
          8 | Omaneya Airlines
        879 | Jordan Airlines
        123 | Singapore Airlines
(15 rows)
```

5.2.8 airplane

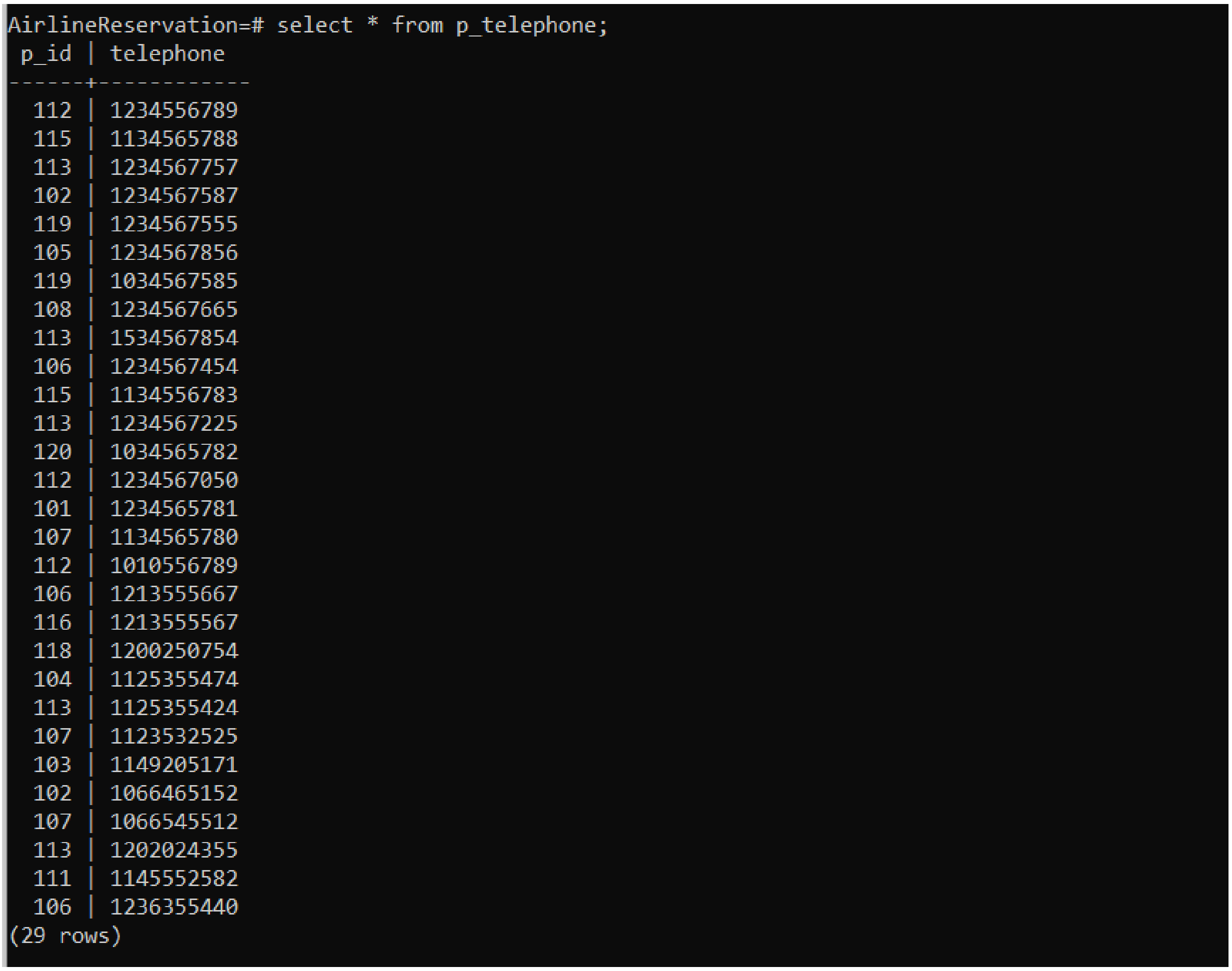
```
insert into airplane (plane_id,type,no_seats,airline_id)values
(9023,'Airbus A380',100,001),
(9033,'Airbus A380',100,098),
(9723,'Airbus A380',100,220),
(9167,'Boeing 747-8',300,125),
(9165,'Boeing 747-8',400,157),
(9168,'Boeing 737 NG/737 MAX',350,589),
(9155,'Supermarine Spitfire',150,879),
(9248,'Lockheed SR-71 Blackbird',200,008),
(9875,'Cirrus SR22',250,879),
(9235,'Lockheed C-130',200,854);
```



```
AirlineReservation=# select * from airplane;
 plane_id |           type            | no_seats | airline_id
----------+---------------------------+----------+------------
     9023 | Airbus A380               |      100 |          1
     9033 | Airbus A380               |      100 |         98
     9723 | Airbus A380               |      100 |        220
     9167 | Boeing 747-8              |      300 |        125
     9165 | Boeing 747-8              |      400 |        157
     9168 | Boeing 737 NG/737 MAX     |      350 |        589
     9155 | Supermarine Spitfire      |      150 |        879
     9248 | Lockheed SR-71 Blackbird  |      200 |          8
     9875 | Cirrus SR22               |      250 |        879
     9235 | Lockheed C-130            |      200 |        854
(10 rows)
```

5.2.9 p_Telephone

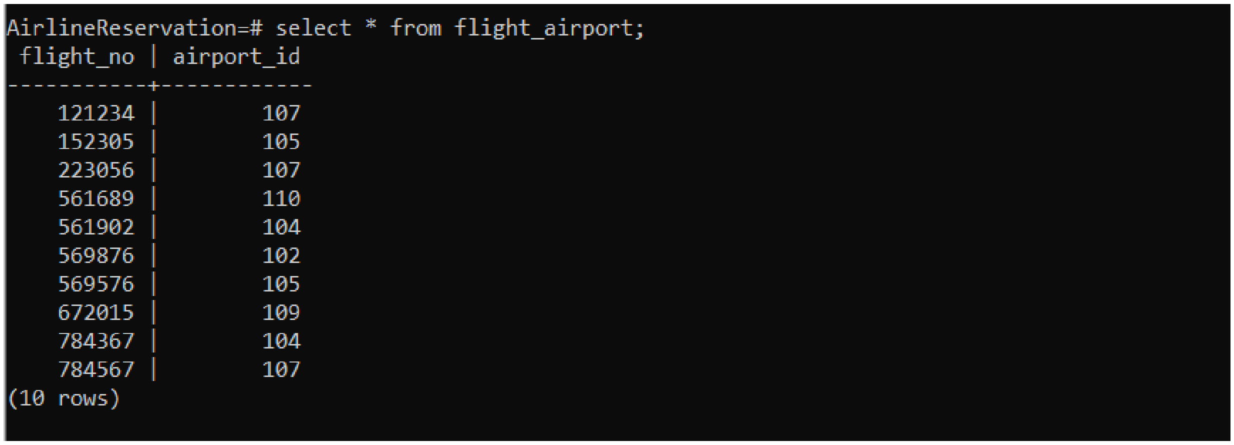
```
insert into p_Telephone(p_id,telephone) values
(112,1234556789), (115,1134565788),
(113,1234567757), (102,1234567587),
(119,1234567555), (105,1234567856),
(119,1034567585), (108,1234567665),
(113,1534567854), (106,1234567454),
(115,1134556783), (113,1234567225),
(120,1034565782), (112,1234567050),
(101,1234565781), (107,1134565780),
(112,1010556789), (106,1213555667),
(116,1213555567), (118,1200250754),
(104,1125355474), (113,1125355424),
(107,1123532525), (103,1149205171),
(102,1066465152), (107,1066545512),
(113,1202024355),
(111,1145552582),
(106,1236355440)
```



```
AirlineReservation=# select * from p_telephone;
 p_id | telephone
------+------------
  112 | 1234556789
  115 | 1134565788
  113 | 1234567757
  102 | 1234567587
  119 | 1234567555
  105 | 1234567856
  119 | 1034567585
  108 | 1234567665
  113 | 1534567854
  106 | 1234567454
  115 | 1134556783
  113 | 1234567225
  120 | 1034565782
  112 | 1234567050
  101 | 1234565781
  107 | 1134565780
  112 | 1010556789
  106 | 1213555667
  116 | 1213555567
  118 | 1200250754
  104 | 1125355474
  113 | 1125355424
  107 | 1123532525
  103 | 1149205171
  102 | 1066465152
  107 | 1066545512
  113 | 1202024355
  111 | 1145552582
  106 | 1236355440
(29 rows)
```

5.2.10 flight_airport

```
insert into flight_airport (flight_no,airport_id) values
(121234,107),
(152305,105),
(223056,107),
(561689,110),
(561902,104),
(569876,102),
(569576,105),
(672015,109),
(784367,104),
(784567,107);
```

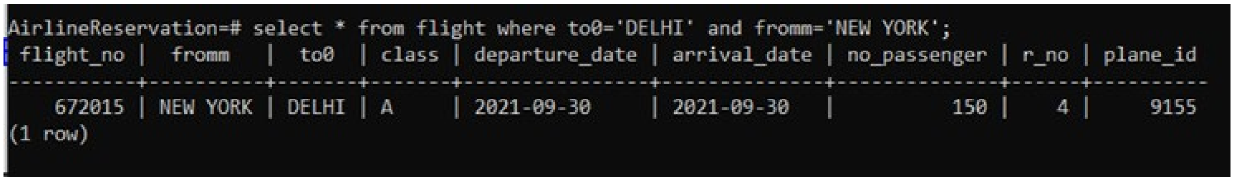


```
AirlineReservation=# select * from flight_airport;
 flight_no | airport_id
-----------+------------
    121234 |        107
    152305 |        105
    223056 |        107
    561689 |        110
    561902 |        104
    569876 |        102
    569576 |        105
    672015 |        109
    784367 |        104
    784567 |        107
(10 rows)
```

5.3 QUERIES

5.3.1 Display available flight from NEW YORK to DELHI.

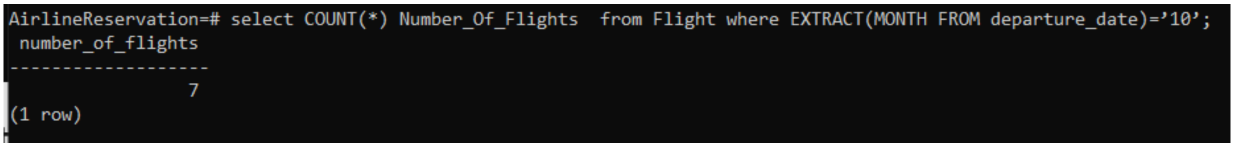
select * from flight where to0 = 'DELHI' and from = 'NEW YORK';



```
AirlineReservation=# select * from flight where to0='DELHI' and fromm='NEW YORK';
 flight_no |  fromm   |  to0  | class | departure_date | arrival_date | no_passenger | r_no | plane_id
-----------+----------+-------+-------+----------------+--------------+--------------+------+----------
    672015 | NEW YORK | DELHI | A     | 2021-09-30     | 2021-09-30   |          150 |    4 |     9155
(1 row)
```

5.3.2 Count How many flights is in month ‘10’.

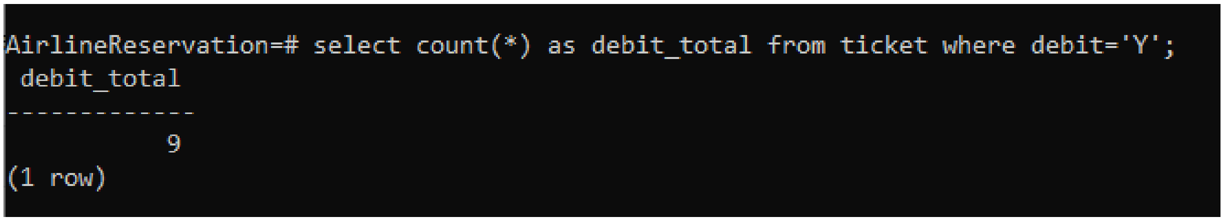
SELECT COUNT(*) Number_Of_Flights from Flight where EXTRACT(MONTH FROM departure_date) = ’10’;



```
AirlineReservation=# select COUNT(*) Number_Of_Flights  from Flight where EXTRACT(MONTH FROM departure_date)=’10’;
 number_of_flights
-------------------
                 7
(1 row)
```

5.3.3 Display total count of ticket where payment is done by debit card;

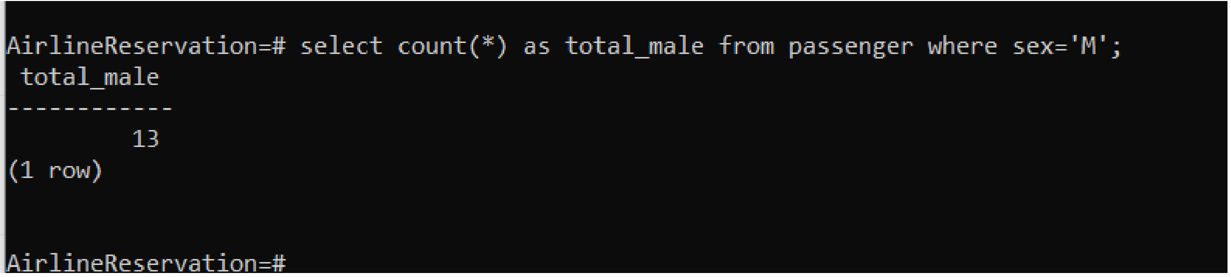
SELECT COUNT(*) as debit_total from ticket where debit='Y';



```
AirlineReservation=# select count(*) as debit_total from ticket where debit='Y';
 debit_total
-------------
           9
(1 row)
```

5.3.4 Count how many male passengers is there in passenger table.

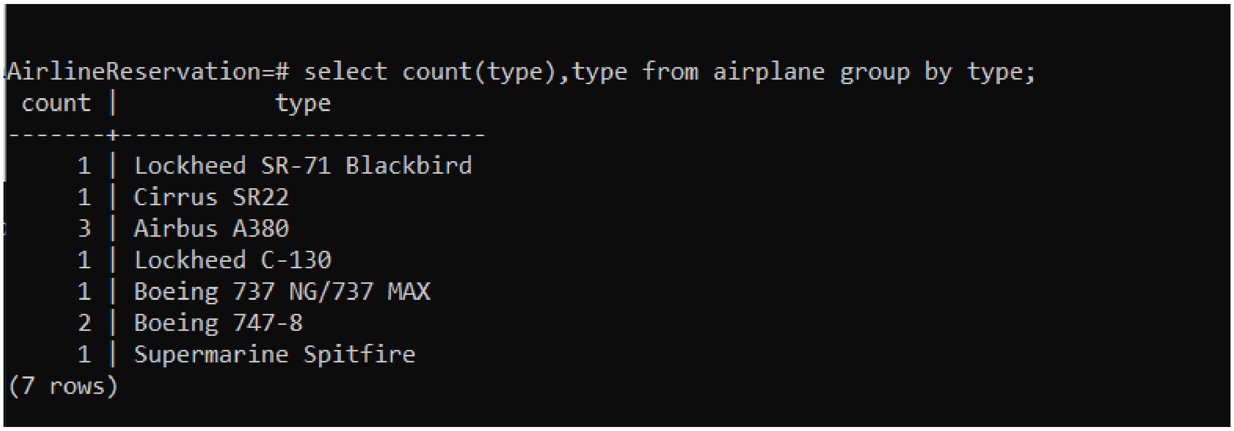
SELECT COUNT(*) as total_male from passenger where sex = 'M';



```
AirlineReservation=# select count(*) as total_male from passenger where sex='M';
 total_male
------------
         13
(1 row)


AirlineReservation=#
```

5.3.5 Count different type of airplane (using group by).

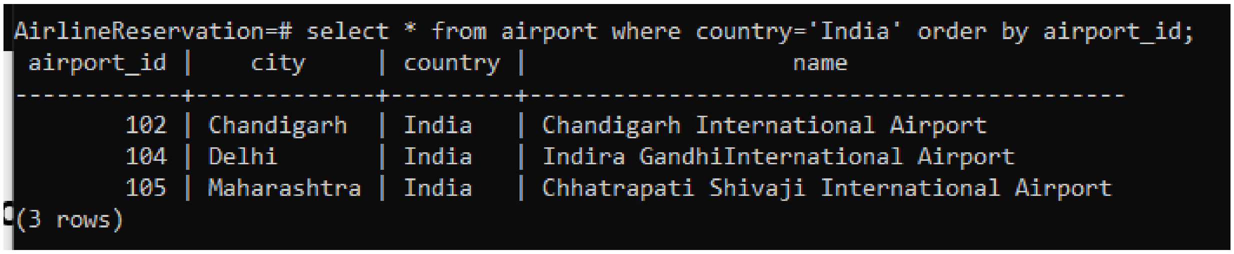
SELECT COUNT(type), type from airplane group by type;



```
AirlineReservation=# select count(type),type from airplane group by type;
 count |           type
-------+-------------------------
     1 | Lockheed SR-71 Blackbird
     1 | Cirrus SR22
     3 | Airbus A380
     1 | Lockheed C-130
     1 | Boeing 737 NG/737 MAX
     2 | Boeing 747-8
     1 | Supermarine Spitfire
(7 rows)
```

5.3.6 Display Airport details whose country is ‘India’.

select * from airport where country = 'India' order by airport_id;

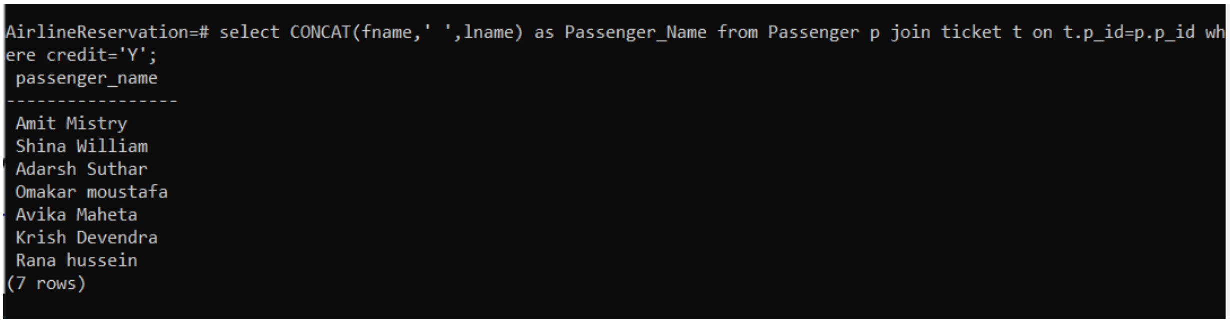


```
AirlineReservation=# select * from airport where country='India' order by airport_id;
 airport_id |    city     | country |                  name
------------+-------------+---------+------------------------------------------
        102 | Chandigarh  | India   | Chandigarh International Airport
        104 | Delhi       | India   | Indira GandhiInternational Airport
        105 | Maharashtra | India   | Chhatrapati Shivaji International Airport
(3 rows)
```

- **JOIN / SUBQUERIES**

5.3.7 Display passenger’s names whose flight is booked by credit card.

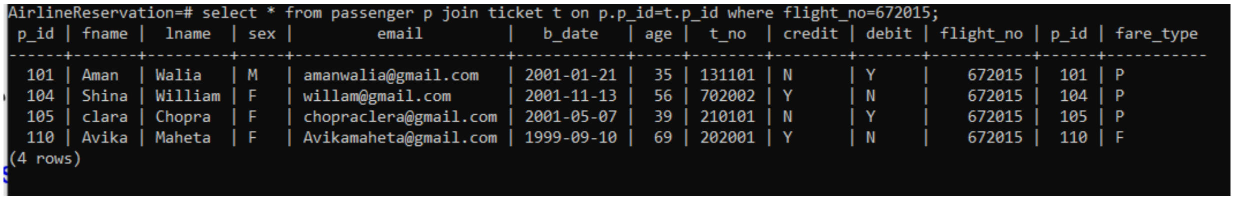
select CONCAT(fname, ' ' ,lname) as Passenger_Name from Passenger p join ticket t on t.p_id = p.p_id where credit = 'Y';



```
AirlineReservation=# select CONCAT(fname,' ',lname) as Passenger_Name from Passenger p join ticket t on t.p_id=p.p_id wh
ere credit='Y';
 passenger_name
----------------
 Amit Mistry
 Shina William
 Adarsh Suthar
 Omakar moustafa
 Avika Maheta
 Krish Devendra
 Rana hussein
(7 rows)
```

5.3.8 Display passenger information whose flight no is 672015.

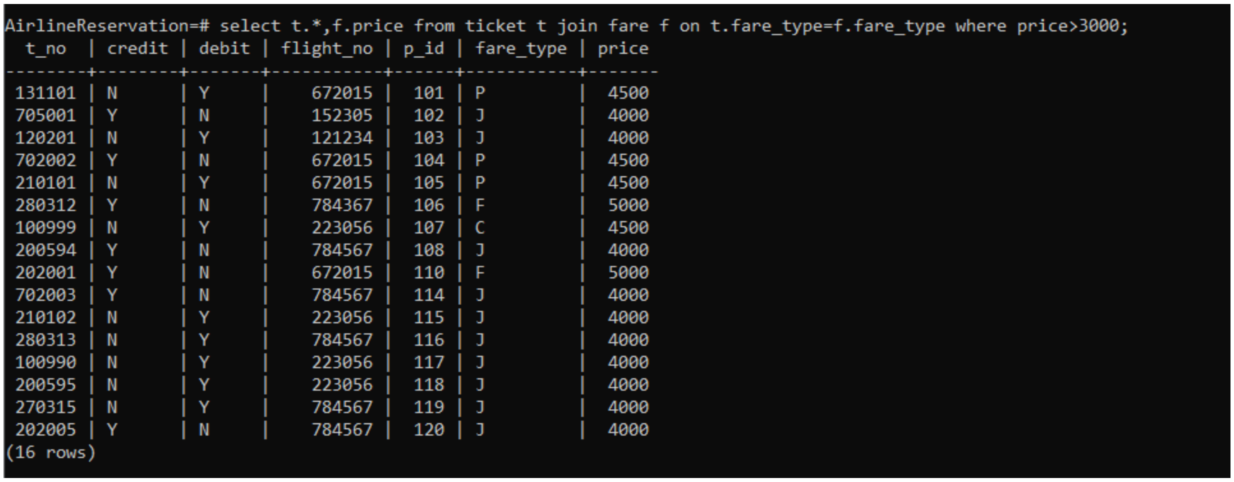
select * from passenger p join ticket t on p.p_id = t.p_id where flight_no = 672015;



```
AirlineReservation=# select * from passenger p join ticket t on p.p_id=t.p_id where flight_no=672015;
 p_id | fname |  lname  | sex |         email         |   b_date   | age |  t_no  | credit | debit | flight_no | p_id | fare_type
------+-------+---------+-----+-----------------------+------------+-----+--------+--------+-------+-----------+------+-----------
  101 | Aman  | Walia   | M   | amanwalia@gmail.com   | 2001-01-21 |  35 | 131101 | N      | Y     |    672015 |  101 | P
  104 | Shina | William | F   | willam@gmail.com      | 2001-11-13 |  56 | 702002 | Y      | N     |    672015 |  104 | P
  105 | clara | Chopra  | F   | chopraclera@gmail.com | 2001-05-07 |  39 | 210101 | N      | Y     |    672015 |  105 | P
  110 | Avika | Maheta  | F   | Avikamaheta@gmail.com | 1999-09-10 |  69 | 202001 | Y      | N     |    672015 |  110 | F
(4 rows)
```

5.3.9 Display ticket information and price where ticket price in greater than 3000.

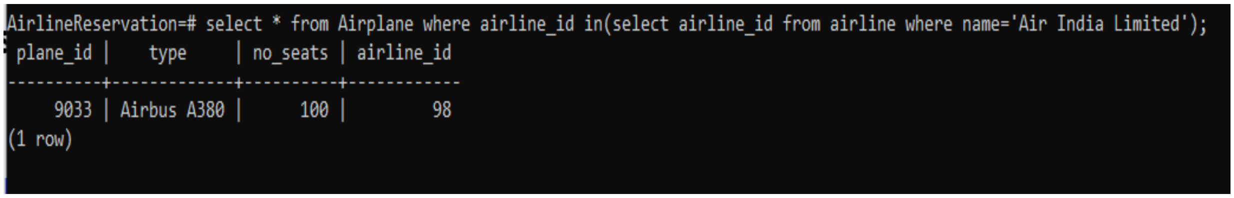
select t.*,f.price from ticket t join fare f on t.fare_type = f.fare_type where price>3000;



```
AirlineReservation=# select t.*,f.price from ticket t join fare f on t.fare_type=f.fare_type where price>3000;
  t_no  | credit | debit | flight_no | p_id | fare_type | price
--------+--------+-------+-----------+------+-----------+-------
 131101 | N      | Y     |    672015 |  101 | P         |  4500
 705001 | Y      | N     |    152305 |  102 | J         |  4000
 120201 | N      | Y     |    121234 |  103 | J         |  4000
 702002 | Y      | N     |    672015 |  104 | P         |  4500
 210101 | N      | Y     |    672015 |  105 | P         |  4500
 280312 | Y      | N     |    784367 |  106 | F         |  5000
 100999 | N      | Y     |    223056 |  107 | C         |  4500
 200594 | Y      | N     |    784567 |  108 | J         |  4000
 202001 | Y      | N     |    672015 |  110 | F         |  5000
 702003 | Y      | N     |    784567 |  114 | J         |  4000
 210102 | N      | Y     |    223056 |  115 | J         |  4000
 280313 | N      | Y     |    784567 |  116 | J         |  4000
 100990 | N      | Y     |    223056 |  117 | J         |  4000
 200595 | N      | Y     |    223056 |  118 | J         |  4000
 270315 | N      | Y     |    784567 |  119 | J         |  4000
 202005 | Y      | N     |    784567 |  120 | J         |  4000
(16 rows)
```

5.3.10 Display airplane details whose company is ‘Air India limited’.

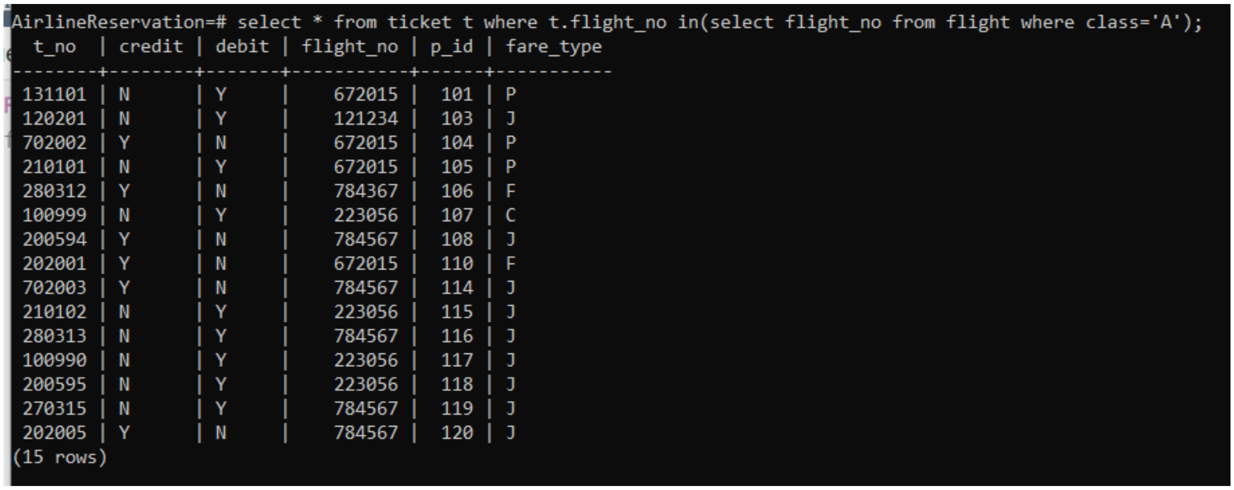
select * from Airplane where airline_id in(select airline_id from airline where name = 'Air India Limited');



```
AirlineReservation=# select * from Airplane where airline_id in(select airline_id from airline where name='Air India Limited');
 plane_id |    type     | no_seats | airline_id
----------+-------------+----------+------------
     9033 | Airbus A380 |      100 |         98
(1 row)
```

5.3.11 display ticket details of flight which has class ‘A’.

select * from ticket t where t.flight_no in(select flight_no from flight where class = 'A');



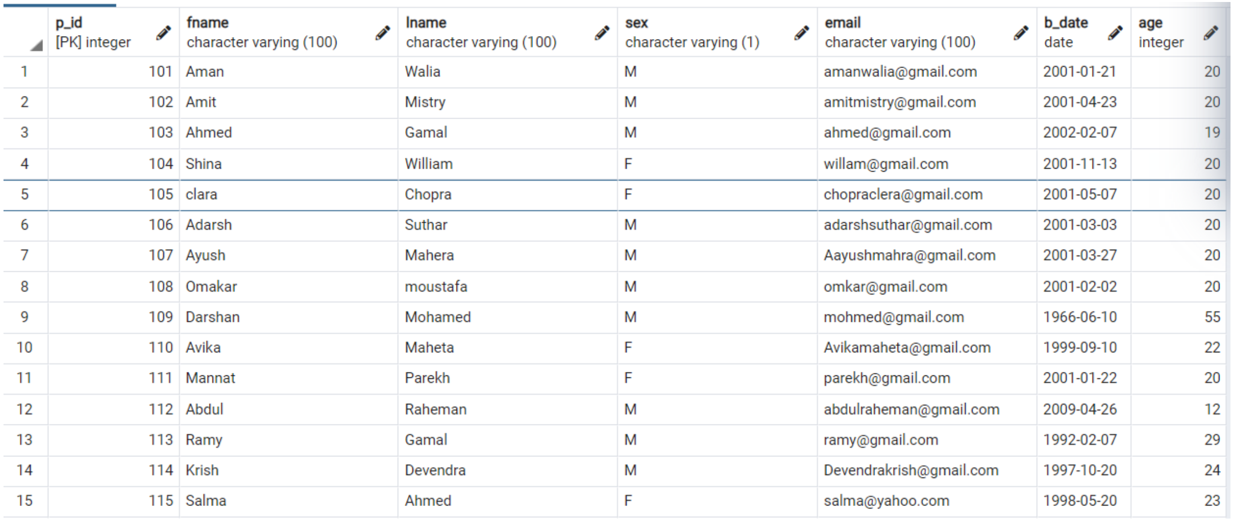
```
AirlineReservation=# select * from ticket t where t.flight_no in(select flight_no from flight where class='A');
  t_no  | credit | debit | flight_no | p_id | fare_type
--------+--------+-------+-----------+------+-----------
 131101 | N      | Y     |    672015 |  101 | P
 120201 | N      | Y     |    121234 |  103 | J
 702002 | Y      | N     |    672015 |  104 | P
 210101 | N      | Y     |    672015 |  105 | P
 280312 | Y      | N     |    784367 |  106 | F
 100999 | N      | Y     |    223056 |  107 | C
 200594 | Y      | N     |    784567 |  108 | J
 202001 | Y      | N     |    672015 |  110 | F
 702003 | Y      | N     |    784567 |  114 | J
 210102 | N      | Y     |    223056 |  115 | J
 280313 | N      | Y     |    784567 |  116 | J
 100990 | N      | Y     |    223056 |  117 | J
 200595 | N      | Y     |    223056 |  118 | J
 270315 | N      | Y     |    784567 |  119 | J
 202005 | Y      | N     |    784567 |  120 | J
(15 rows)
```

5.4 PL/SQL Queries

5.4.1 PL/SQL to find age of passenger using birthdate of passenger and to update the column age of passenger table.

```
CREATE OR REPLACE function find_age()
returns integer
language plpgsql
AS
$$
declare
    age1 integer;
    pass_id integer;
    cnt int;
    b_year int;
BEGIN
   cnt := (select count(*) from passenger);
   pass_id := 101;
     while cnt > 0 loop
            b_year := (select EXTRACT(year from b_date) from passenger where p_id =
   pass_id);
            age1 := (date_part('year', CURRENT_DATE) - b_year);
       raise notice 'age is %', age1;
       update passenger set age = age1 where p_id = pass_id;

   pass_id := pass_id + 1;
   cnt := cnt - 1;
   end loop;
   return 0;
END;
$$;

select find_age();
```

Output:



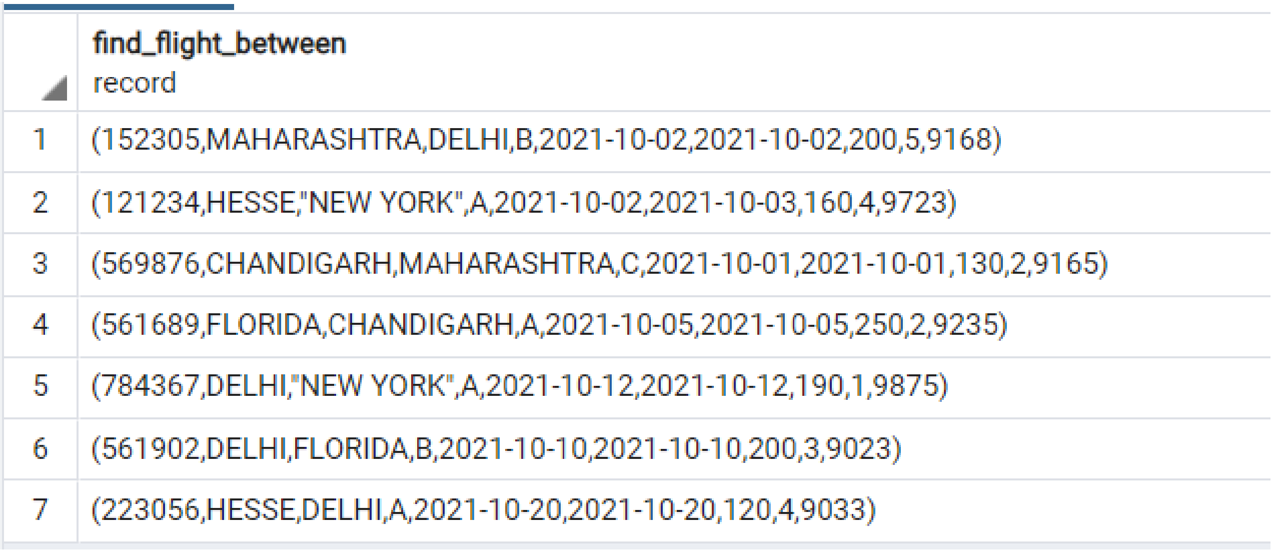
	p_id [PK] integer	fname character varying (100)	lname character varying (100)	sex character varying (1)	email character varying (100)	b_date date	age integer
1	101	Aman	Walia	M	amanwalia@gmail.com	2001-01-21	20
2	102	Amit	Mistry	M	amitmistry@gmail.com	2001-04-23	20
3	103	Ahmed	Gamal	M	ahmed@gmail.com	2002-02-07	19
4	104	Shina	William	F	willam@gmail.com	2001-11-13	20
5	105	clara	Chopra	F	chopraclera@gmail.com	2001-05-07	20
6	106	Adarsh	Suthar	M	adarshsuthar@gmail.com	2001-03-03	20
7	107	Ayush	Mahera	M	Aayushmahra@gmail.com	2001-03-27	20
8	108	Omakar	moustafa	M	omkar@gmail.com	2001-02-02	20
9	109	Darshan	Mohamed	M	mohmed@gmail.com	1966-06-10	55
10	110	Avika	Maheta	F	Avikamaheta@gmail.com	1999-09-10	22
11	111	Mannat	Parekh	F	parekh@gmail.com	2001-01-22	20
12	112	Abdul	Raheman	M	abdulraheman@gmail.com	2009-04-26	12
13	113	Ramy	Gamal	M	ramy@gmail.com	1992-02-07	29
14	114	Krish	Devendra	M	Devendrakrish@gmail.com	1997-10-20	24
15	115	Salma	Ahmed	F	salma@yahoo.com	1998-05-20	23

5.5 FUNCTIONS

5.5.1 Function to find flight details including flight no, from, to, departure date, arrival date, available seats, route no and plane id between two given dates.

```
CREATE or REPLACE FUNCTION find_flight_between(first_date date, last_date date)
returns table (flight_no integer, fromm varchar(50), to0 varchar(50), class varchar(50),
departure_date date, arrival_date date, no_passenger integer, r_no integer, plane_id
integer)
language plpgsql
as
$$
begin
  return query select * from flight f where f.departure_date >= first_date and
f.departure_date <= last_date;
end;
$$;

select find_flight_between('2021-10-1', '2021-10-30');
```

- Output



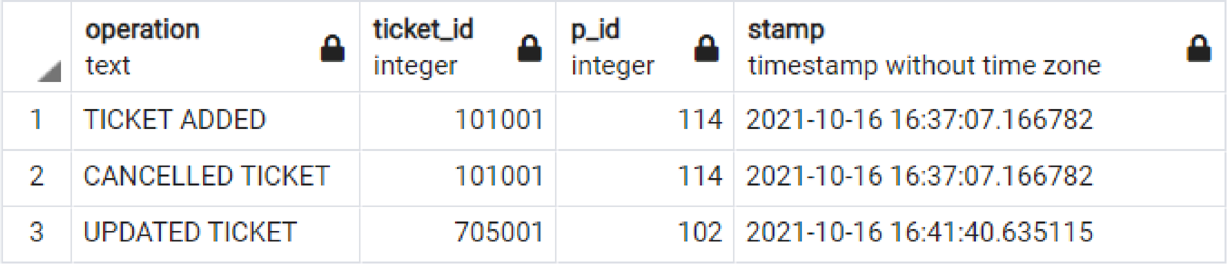
	find_flight_between record
1	(152305,MAHARASHTRA,DELHI,B,2021-10-02,2021-10-02,200,5,9168)
2	(121234,HESSE,"NEW YORK",A,2021-10-02,2021-10-03,160,4,9723)
3	(569876,CHANDIGARH,MAHARASHTRA,C,2021-10-01,2021-10-01,130,2,9165)
4	(561689,FLORIDA,CHANDIGARH,A,2021-10-05,2021-10-05,250,2,9235)
5	(784367,DELHI,"NEW YORK",A,2021-10-12,2021-10-12,190,1,9875)
6	(561902,DELHI,FLORIDA,B,2021-10-10,2021-10-10,200,3,9023)
7	(223056,HESSE,DELHI,A,2021-10-20,2021-10-20,120,4,9033)

5.6 TRIGGERS

5.6.1 Trigger to keep eye on ticket table & it makes new entry when changes occur in ticket table e.g., ticket added, cancelled ticket, updated ticket.

```
CREATE TABLE ticket_audit(operation text, ticket_id integer, p_id integer, stamp
timestamp);
drop trigger ticket_audit on ticket;

CREATE OR REPLACE FUNCTION do_ticket_audit()
RETURNS TRIGGER AS $ticket_audit$
begin
    if(TG_OP = 'DELETE') then
    INSERT INTO ticket_audit SELECT 'CANCELLED TICKET',OLD.t_no, OLD.p_id,
    now();
            RETURN OLD;

    elseif(TG_OP = 'UPDATE') then
            INSERT INTO ticket_audit SELECT 'UPDATED TICKET', NEW.t_no,
    NEW.p_id, now();
            RETURN NEW;

    elseif(TG_OP = 'INSERT') then
            INSERT INTO ticket_audit SELECT 'TICKET ADDED', NEW.t_no,
    NEW.p_id, now();
            RETURN NEW;
    END IF;
    RETURN 'NULL';
END;
$ticket_audit$ LANGUAGE PLPGSQL;

CREATE TRIGGER ticket_audit
AFTER INSERT or UPDATE or DELETE ON ticket
FOR EACH ROW EXECUTE PROCEDURE
do_ticket_audit();


insert into ticket values (101001,'Y','N',784567,114,'J');
delete from ticket where t_no = 101001;
UPDATE ticket set debit = 'Y', credit = 'N' where t_no = 705001;
```

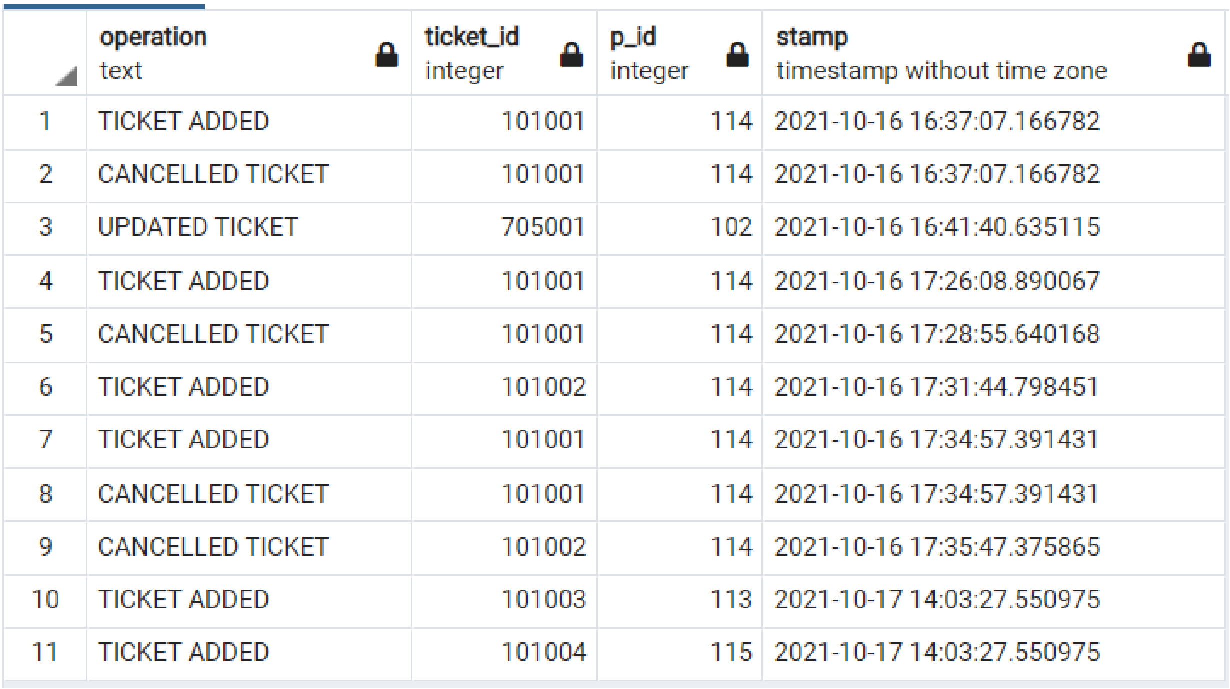
- Output:



	operation text	ticket_id integer	p_id integer	stamp timestamp without time zone
1	TICKET ADDED	101001	114	2021-10-16 16:37:07.166782
2	CANCELLED TICKET	101001	114	2021-10-16 16:37:07.166782
3	UPDATED TICKET	705001	102	2021-10-16 16:41:40.635115

insert into ticket values
(101003,'Y','N',152305,113,'J');
insert into ticket values
(101004,'Y','N',152305,115,'J');

- Output:

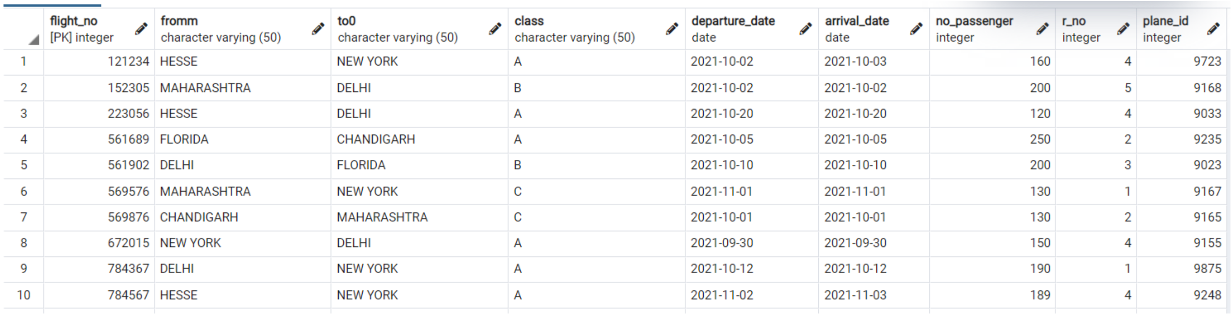


	operation text	ticket_id integer	p_id integer	stamp timestamp without time zone
1	TICKET ADDED	101001	114	2021-10-16 16:37:07.166782
2	CANCELLED TICKET	101001	114	2021-10-16 16:37:07.166782
3	UPDATED TICKET	705001	102	2021-10-16 16:41:40.635115
4	TICKET ADDED	101001	114	2021-10-16 17:26:08.890067
5	CANCELLED TICKET	101001	114	2021-10-16 17:28:55.640168
6	TICKET ADDED	101002	114	2021-10-16 17:31:44.798451
7	TICKET ADDED	101001	114	2021-10-16 17:34:57.391431
8	CANCELLED TICKET	101001	114	2021-10-16 17:34:57.391431
9	CANCELLED TICKET	101002	114	2021-10-16 17:35:47.375865
10	TICKET ADDED	101003	113	2021-10-17 14:03:27.550975
11	TICKET ADDED	101004	115	2021-10-17 14:03:27.550975

5.6.2 Trigger to keep eye on available seats in particular flight when new ticket is generated or ticket is cancelled. (e.g., available seats are 100 for flight x. after when new ticket is generated, then available seats will be 99 for flight x)

```
CREATE OR REPLACE FUNCTION no_passenger_update()
RETURNS TRIGGER as $$
declare
old_no_passenger integer;
begin
    if(TG_OP =  'DELETE') then
    old_no_passenger := (select no_passenger from flight where flight_no
     in(select flight_no from ticket where t_no = OLD.t_no));
    old_no_passenger := old_no_passenger + 1;
    update flight set no_passenger = old_no_passenger where flight_no
     in(select flight_no from ticket where t_no = OLD.t_no);
     return old;

    elseif(TG_OP = 'INSERT') then
    old_no_passenger := (select no_passenger from flight where flight_no
     in(select flight_no from ticket where t_no = NEW.t_no));
     old_no_passenger := old_no_passenger - 1;
    update flight set no_passenger = old_no_passenger where flight_no
     in(select flight_no from ticket where t_no = NEW.t_no);
     return new;
    end if;
END;
$$ LANGUAGE PLPGSQL;

create trigger no_passenger_up
before INSERT or DELETE ON ticket
FOR EACH ROW EXECUTE PROCEDURE
no_passenger_update();

insert into ticket values
(101001,'Y','N',784567,114,'J');
```

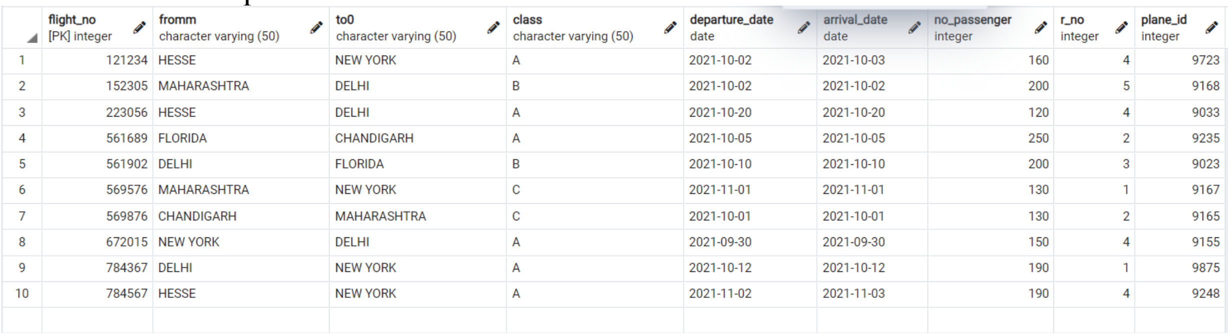
- Output:



	flight_no [PK] integer	fromm character varying (50)	to0 character varying (50)	class character varying (50)	departure_date date	arrival_date date	no_passenger integer	r_no integer	plane_id integer
1	121234	HESSE	NEW YORK	A	2021-10-02	2021-10-03	160	4	9723
2	152305	MAHARASHTRA	DELHI	B	2021-10-02	2021-10-02	200	5	9168
3	223056	HESSE	DELHI	A	2021-10-20	2021-10-20	120	4	9033
4	561689	FLORIDA	CHANDIGARH	A	2021-10-05	2021-10-05	250	2	9235
5	561902	DELHI	FLORIDA	B	2021-10-10	2021-10-10	200	3	9023
6	569576	MAHARASHTRA	NEW YORK	C	2021-11-01	2021-11-01	130	1	9167
7	569876	CHANDIGARH	MAHARASHTRA	C	2021-10-01	2021-10-01	130	2	9165
8	672015	NEW YORK	DELHI	A	2021-09-30	2021-09-30	150	4	9155
9	784367	DELHI	NEW YORK	A	2021-10-12	2021-10-12	190	1	9875
10	784567	HESSE	NEW YORK	A	2021-11-02	2021-11-03	189	4	9248

delete from ticket where t_no=101001;

- Output:



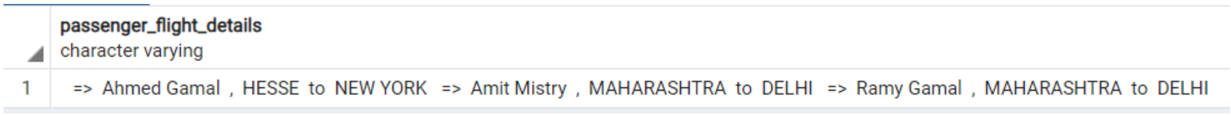
	flight_no [PK] integer	fromm character varying (50)	to0 character varying (50)	class character varying (50)	departure_date date	arrival_date date	no_passenger integer	r_no integer	plane_id integer
1	121234	HESSE	NEW YORK	A	2021-10-02	2021-10-03	160	4	9723
2	152305	MAHARASHTRA	DELHI	B	2021-10-02	2021-10-02	200	5	9168
3	223056	HESSE	DELHI	A	2021-10-20	2021-10-20	120	4	9033
4	561689	FLORIDA	CHANDIGARH	A	2021-10-05	2021-10-05	250	2	9235
5	561902	DELHI	FLORIDA	B	2021-10-10	2021-10-10	200	3	9023
6	569576	MAHARASHTRA	NEW YORK	C	2021-11-01	2021-11-01	130	1	9167
7	569876	CHANDIGARH	MAHARASHTRA	C	2021-10-01	2021-10-01	130	2	9165
8	672015	NEW YORK	DELHI	A	2021-09-30	2021-09-30	150	4	9155
9	784367	DELHI	NEW YORK	A	2021-10-12	2021-10-12	190	1	9875
10	784567	HESSE	NEW YORK	A	2021-11-02	2021-11-03	190	4	9248

5.7 CURSORS

5.7.1 Cursor to find details of all male passengers with boarding and destination details who have flight at particular given date.

```
create or replace function passenger_flight_details(departure_date1 date)
returns varchar
as $$

declare
    details text default '';
    passenger_record record;
    c1 CURSOR(departure_date1 date) FOR SELECT p.fname,p.lname,f.fromm,f.to0,p.sex
    FROM ticket t,passenger p,flight f where t.p_id = p.p_id and t.flight_no = f.flight_no
    and f.departure_date = departure_date1;

begin
    open c1(departure_date1 date);
    loop
    fetch c1 into passenger_record;
    exit when not found;
    if passenger_record.sex = 'M' then
    details := details || '   =>  ' || passenger_record.fname ||  ' ' || passenger_record.lname ||'  ,
    '|| passenger_record.fromm || '  to  ' || passenger_record.to0;
    end if;
    end loop;
    close c1;

    return details;
end;
$$
language plpgsql;

select passenger_flight_details('2021-10-2')
```

- Output



```
  passenger_flight_details
  character varying
1  => Ahmed Gamal , HESSE to NEW YORK  => Amit Mistry , MAHARASHTRA to DELHI  => Ramy Gamal , MAHARASHTRA to DELHI
```

6. FUTURE ENHANCEMENTS OF THE SYSTEM

- We will design Front-end Design in HTML, CSS, JavaScript and Develop Back-end in Python.
- For security purpose New Registration is done using OTP.
- We will make database more consistent and We are making this database efficient and easy to implement with huge data capacity.
- Methods and user data input will be lot easy after the implement of GUI.
- We will also add some extra features so that the users can get answer for their complaints as fast as possible.

7. BIBLIOGRAPHY

- For the successful implementation of this project, we referred to many websites and books.
- We created the ER Diagram and Schema Diagram on www.draw.io.
- Mostly we referred the online material for syntax of procedures, triggers, Exception and cursors.

Reference book:
Data Base System Concepts
-Henry F. Korth & A. Silberschatz 2nd Ed. McGraw-Hill 1991

Reference Websites:

- https://www.stackoverflow.com/
- https://www.postgresqltutorial.com/
- https://w3resource.com/PostgreSQL/tutorial.php
- http://www.mysqltutorial.org/