JAVA SWINGS BASED- AIRLINES MANAGEMENT SYSTEM

- SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfillment of the Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

Medchalam Ashwini <1602-19-737-068>

Under the guidance of Ms B. Leelavathy

Department of Information Technology
Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31

BONAFIDE CERTIFICATE :

This is to certify that this project report titled '**AIRLINES MANAGEMENT SYSTEM**' is a project work of Ms. Medchalam Ashwini bearing roll no.1602-19-737-068 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature	Signature
Internal Examiner	External Examiner

ABSTRACT:

Airline reservation system is one of the most used database system in the world. These systems require high availability and fast response time for hundreds of concurrent users. This system provides options for viewing different flights available with different timings for a particular date and provides date and provides customers with the facility to book a ticket, modify or cancel a particular reservation but it does not provide the customers with details of cost of the ticket. It can also manages all the information about customer, booking enquiry, reservation. Users can view the status and schedule of a flight directly, from the online. It also provides time to time current information related to airlines schedules. It tracks all the details about the airlines booking, ticket booking. It can keep to record airlines employee detail, daily attendance and salary calculation of the employee.

Requirement Analysis:

List of Tables:

- AIRLINES_COMPANY
- AIRPLANE
- AIRPORT
- PASSENGER
- PAYMENT
- EMPLOYEES
- HAS
- OWNS
- ACCESS
- RESERVES
- DETAILS
- COSTS

List of attributes with their domain types:-

Airlines Company:

Company’s id : company_id-varchar2(10)

Company’s name : company_name-varchar2(50)

Airplane:

Airplane’s number : airplane_no-varchar2(10)

Seating Capacity of the Airplane : seating_capacity-number(3)

Airport:

Airport’s id : a_id-varchar2(10)

Airport’s name : a_name-varchar2(50)

Airport’s address : a_address (country-varchar2(20), state-varchar2(20), city-varchar2(20), zip-number(10))

Seat:

Seat Number : seat_no-number(3)

Class : class-varchar2(2)

Is seat available or not? : availability-varchar2(3)

Passenger:

Passenger’s id : p_id-varchar2(10)

Passenger’s name : p_name(f_name-varchar2(30), l_name-varchar2(30))

Passenger’s phone number : p_phone-number(10)

Passenger’s country : p_country-varchar2(20)

Payment:

Payment’s id : pay_id-varchar2(10)

Initial Amount : amount-number(5)

Discount : discount-number(2)

Tax Amount : tax-number(3)

Total Amount : total_amount-number(5)

Flight Hours:

Airplane number : airplane_no-varchar2(10)

Flight arrival airport : arrival_airport-varchar2(30)

Flight arrival time : arrival_time-varchar2(10)

Flight departure airport : departure_airport-varchar2(30)

Flight departure time : departure_time-varchar2(10)

Employees:

Employee’s id : e_id-varchar2(10)

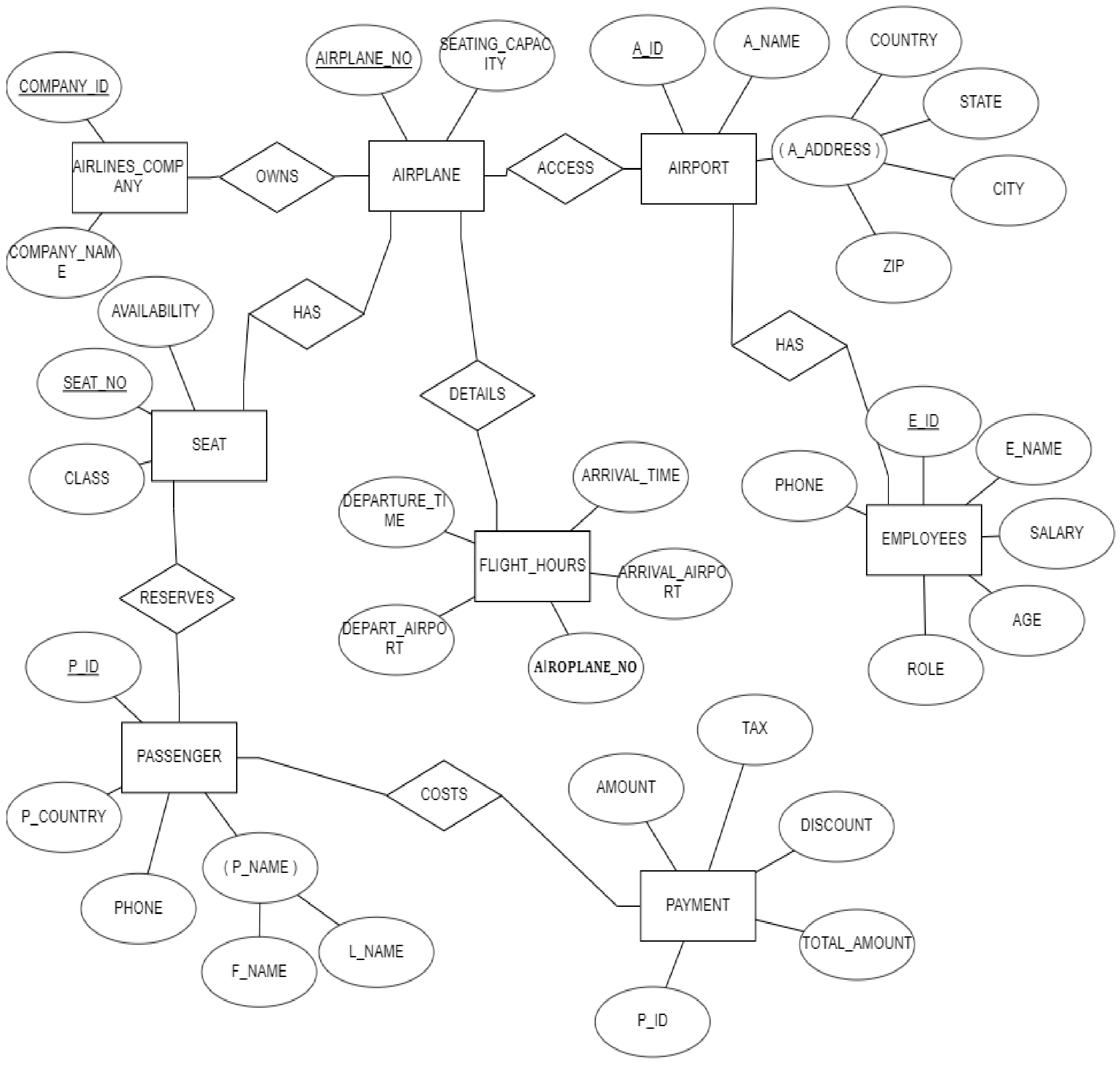
Employee’s name : e_name-varchar2(50)

Employee’s age : e_age-number(2)

Employee’s role : e_role-varchar2(30)

Employee’s phone number : e_phone-varchar2(10)

ER-Diagram:



Mapping Cardinality and Participation Constraints:

The Airlines Company is connected to Airplane. A single Airplane Company can have many Airplanes, so it is one many relationship.

Airplane is connected to Airport. One Airplane can belong to two or more airports, because it departs and arrive at different Airports. So it is many to many relationship.

Airport is connected to Employees. A single has many employees but a employee cannot belong to two or more Airports, which is one to many relationship.

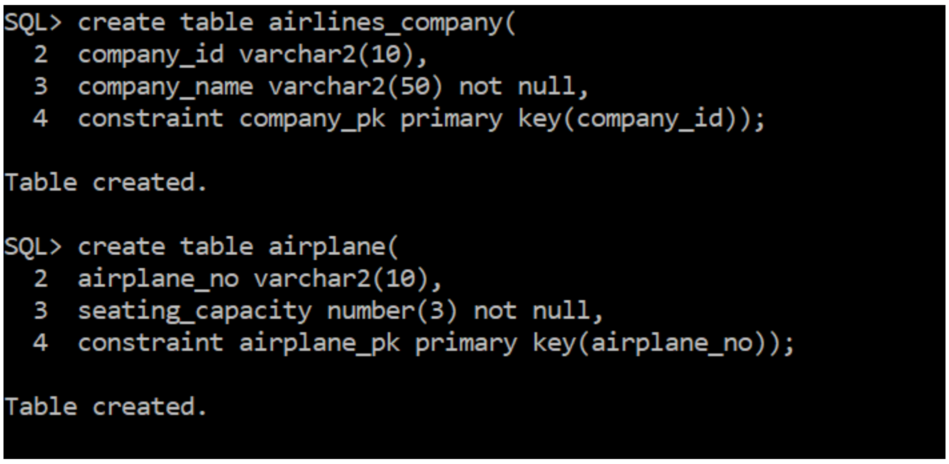
DDL COMMANDS:

```
create table airlines_company(
   company_id varchar2(10),
   company_name varchar2(50) not null,
   constraint company_pk primary key(company_id));
create table airplane(
   airplane_no varchar2(10),
   seating_capacity number(3) not null,
   constraint airplane_pk primary key(airplane_no));
create table owns(
   company_id varchar2(10),
   airplane_no varchar2(10),
   foreign key(company_id) references airlines_company(company_id),
   foreign key(airplane_no) references airplane(airplane_no));
create table airport(
   a_id varchar2(10) primary key,
   a_name varchar2(50) not null,
   a_country varchar2(20) not null,
   a_state varchar2(20) not null,
   a_city varchar2(20) not null,
   a_zip number(10) not null);
create table acess(
   airplane_no varchar2(10),
   a_id varchar2(10),

   foreign key(airplane_no) references airplane(airplane_no),

   foreign key(a_id) references airport(a_id));

create table employees(

   e_id varchar2(10) primary key,

   e_name varchar2(50) not null,

   e_age number not null,

   e_phone number(10) not null,

   e_role varchar2(30) not null,

   e_salary number not null);

create table airport_has_employees(

   a_id varchar2(10),

   e_id varchar2(10),

   foreign key(a_id) references airport(a_id),

   foreign key(e_id) references employees(e_id));

create table seats(

   seat_no number(3) primary key,

   class varchar2(2) not null,

   availability varchar2(3) default'YES');

create table airplane_has_seats(

   airplane_no varchar2(10),

   seat_no number(3),

   foreign key(airplane_no) references airplane(airplane_no),

   foreign key(seat_no) references seats(seat_no));

create table flight_hours(

   airplane_no varchar2(10),
   departure_airport varchar2(30) not null,
   departure_time varchar2(10) not null,
   arrival_airport varchar2(30) not null,
   arrival_time varchar2(10) not null,
   foreign key(airplane_no) references airplane(airplane_no));
create table passengers(
   p_id varchar2(10) primary key,
   p_f_name varchar2(30) not null,
   p_l_name varchar2(30) not null,
   p_country varchar2(20) not null,
   p_phone number(10) not null);
create table reserves(
   p_id varchar2(10),
   seat_no number(3),
   foreign key(p_id) references passengers(p_id),
   foreign key(seat_no) references seats(seat_no));
create table payments(
   pay_id varchar2(10) primary key,
   amount number(5) not null,
   discount number(2) not null,
   tax number(5) not null,
   total_amount number(5));
create table costs(
   p_id varchar2(10),
   pay_id varchar2(10),
   foreign key(p_id) references passengers(p_id),
```

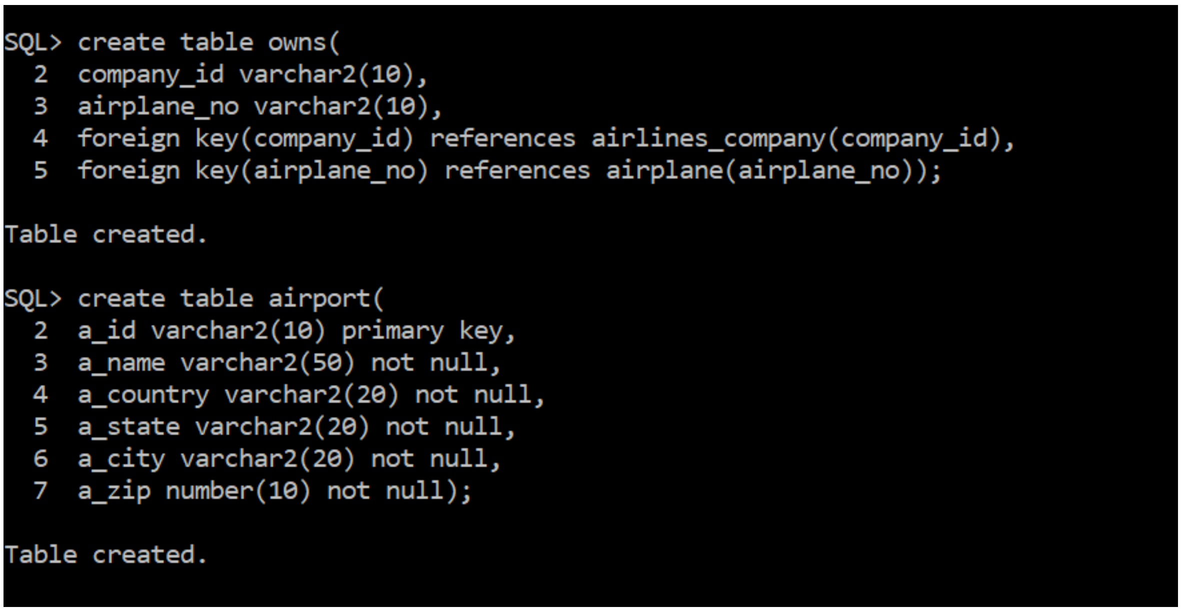
foreign key(pay_id) references payments(pay_id));

DML COMMANDS:

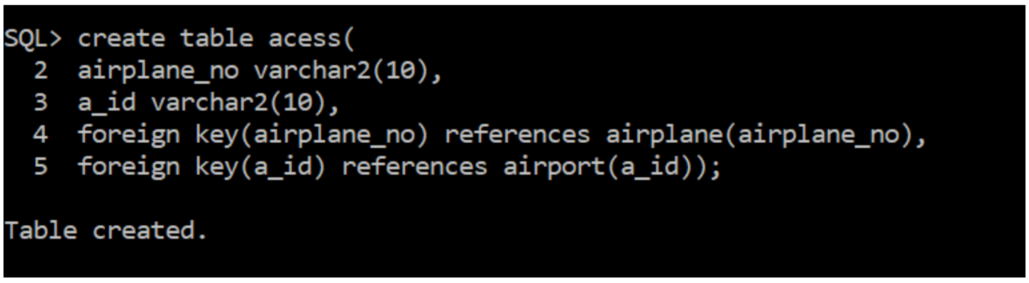
1. insert into airlines_company values('&company_id','&company_name');
2. insert into airplane values('&airplane_no',&seating_capacity);
3. insert into owns values('&company_id','&airplane_no');
4. insert into airport values('&a_id','&a_name','&a_country','&a_state','&a_city','&a_zip');
5. insert into acess values('&airplane_no','&a_id');
6. insert into employees values('&e_id','&e_name','&e_age','&e_phone','&e_role',&e_salary);
7. insert into airport_has_employees values('&a_id','&e_id');
8. insert into seats values(&seat_no,'&class','&availability');
9. insert into airplane_has_seats values('&airplane_no',&seat_no);
10. insert into passengers values('&p_id','&p_f_name','&p_l_name','&p_country',&p_phone);
11. insert into reserves values('&varchar2',&number);
12. insert into payments values('&pay_id',&amount,&discount,&tax,&total_amount);
13. insert into costs values('&p_id','&pay_id');
14. insert into flight_hours values('&airplane_no','&departure_airport','&departure_time','&arrival_airport','&arrival_time');



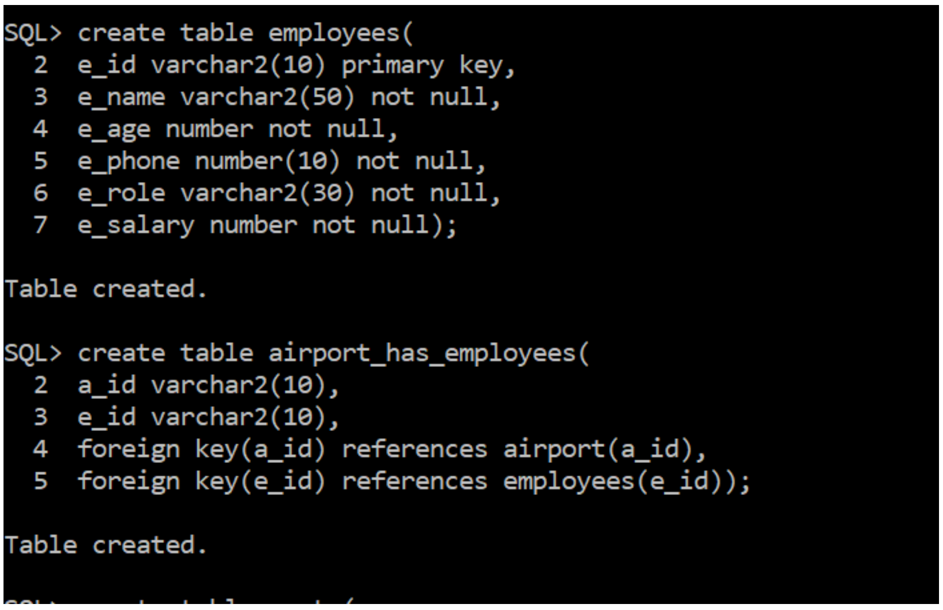
```
SQL> create table airlines_company(
  2  company_id varchar2(10),
  3  company_name varchar2(50) not null,
  4  constraint company_pk primary key(company_id));

Table created.

SQL> create table airplane(
  2  airplane_no varchar2(10),
  3  seating_capacity number(3) not null,
  4  constraint airplane_pk primary key(airplane_no));

Table created.
```



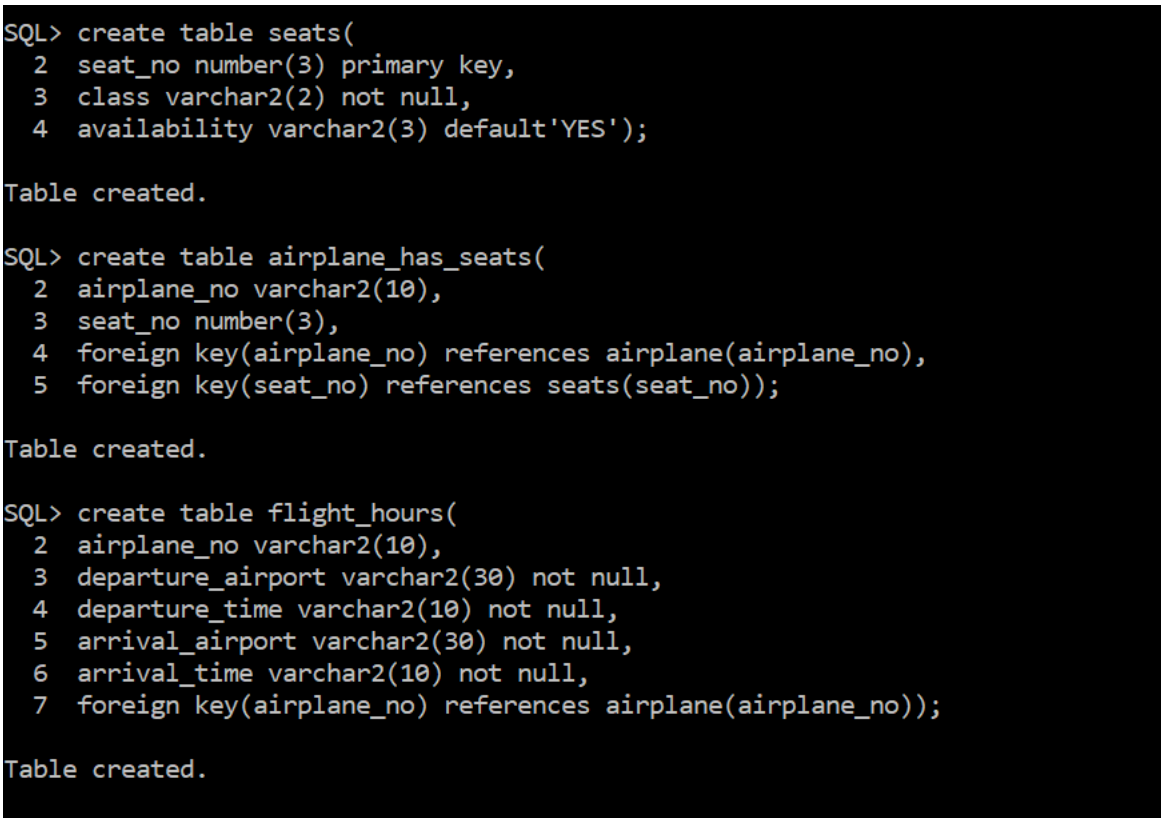
```
SQL> create table owns(
  2  company_id varchar2(10),
  3  airplane_no varchar2(10),
  4  foreign key(company_id) references airlines_company(company_id),
  5  foreign key(airplane_no) references airplane(airplane_no));

Table created.

SQL> create table airport(
  2  a_id varchar2(10) primary key,
  3  a_name varchar2(50) not null,
  4  a_country varchar2(20) not null,
  5  a_state varchar2(20) not null,
  6  a_city varchar2(20) not null,
  7  a_zip number(10) not null);

Table created.
```



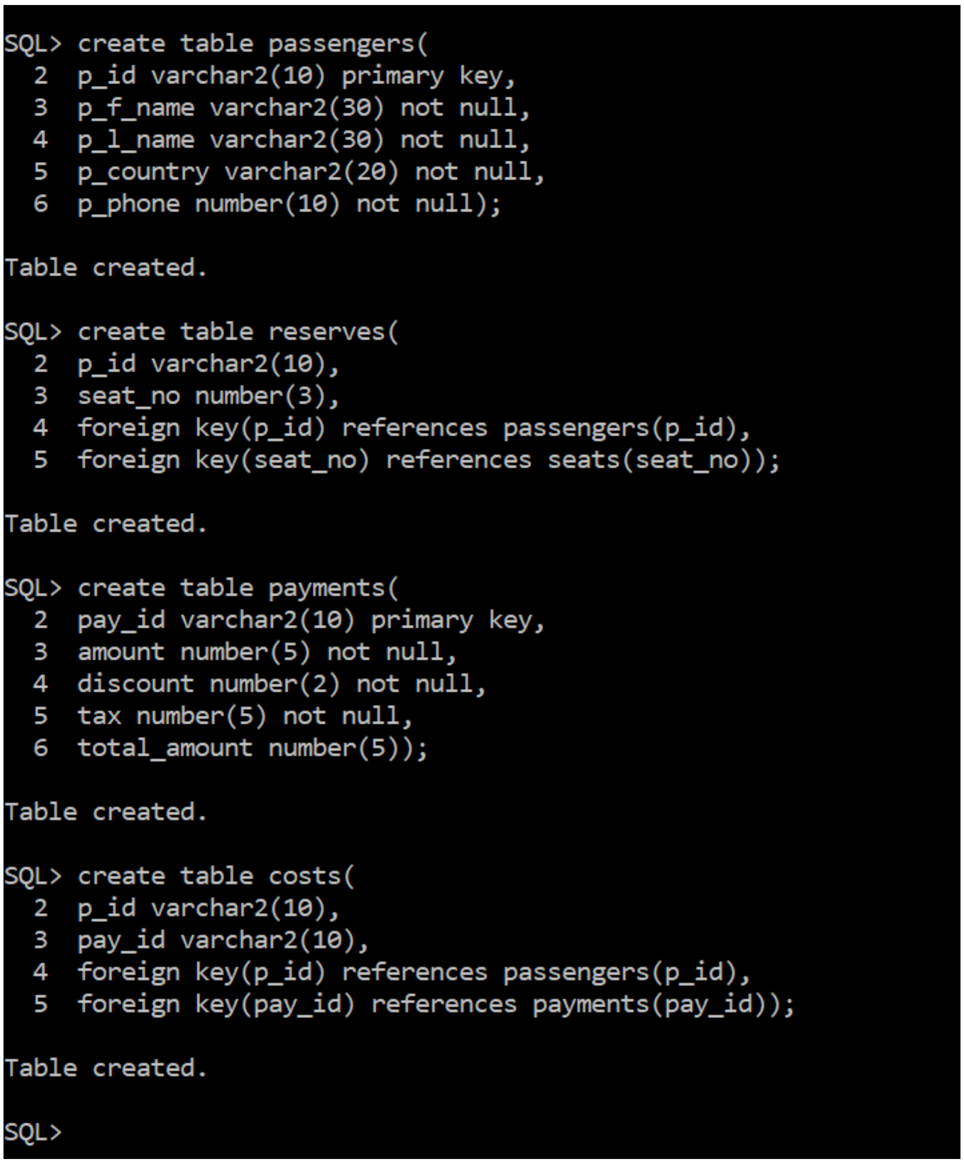
```
SQL> create table acess(
  2  airplane_no varchar2(10),
  3  a_id varchar2(10),
  4  foreign key(airplane_no) references airplane(airplane_no),
  5  foreign key(a_id) references airport(a_id));

Table created.
```



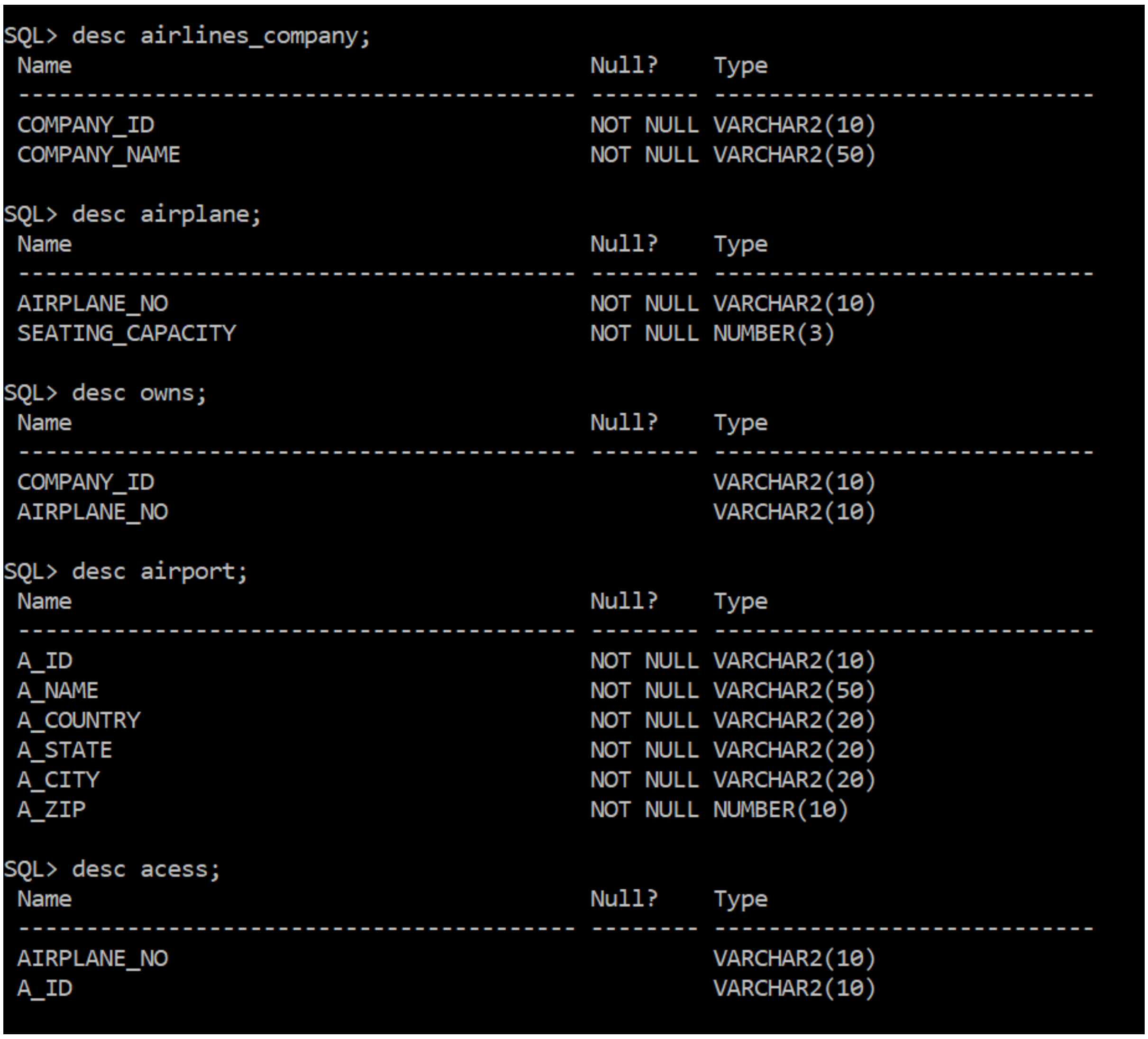
```
SQL> create table employees(
  2  e_id varchar2(10) primary key,
  3  e_name varchar2(50) not null,
  4  e_age number not null,
  5  e_phone number(10) not null,
  6  e_role varchar2(30) not null,
  7  e_salary number not null);

Table created.

SQL> create table airport_has_employees(
  2  a_id varchar2(10),
  3  e_id varchar2(10),
  4  foreign key(a_id) references airport(a_id),
  5  foreign key(e_id) references employees(e_id));

Table created.
```



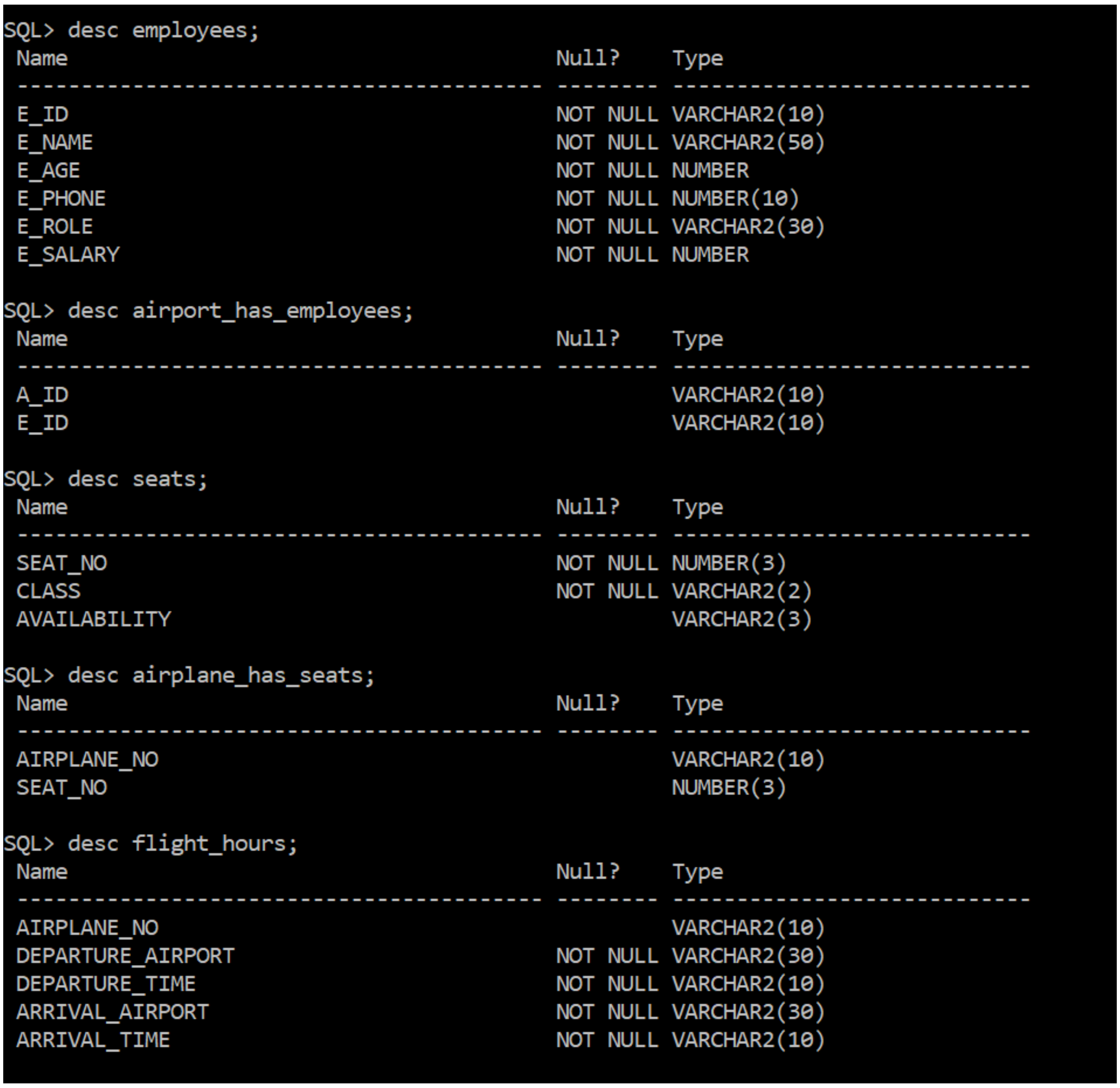
```
SQL> create table seats(
seat_no number(3) primary key,
class varchar2(2) not null,
availability varchar2(3) default'YES');

Table created.

SQL> create table airplane_has_seats(
airplane_no varchar2(10),
seat_no number(3),
foreign key(airplane_no) references airplane(airplane_no),
foreign key(seat_no) references seats(seat_no));

Table created.

SQL> create table flight_hours(
airplane_no varchar2(10),
departure_airport varchar2(30) not null,
departure_time varchar2(10) not null,
arrival_airport varchar2(30) not null,
arrival_time varchar2(10) not null,
foreign key(airplane_no) references airplane(airplane_no));

Table created.
```



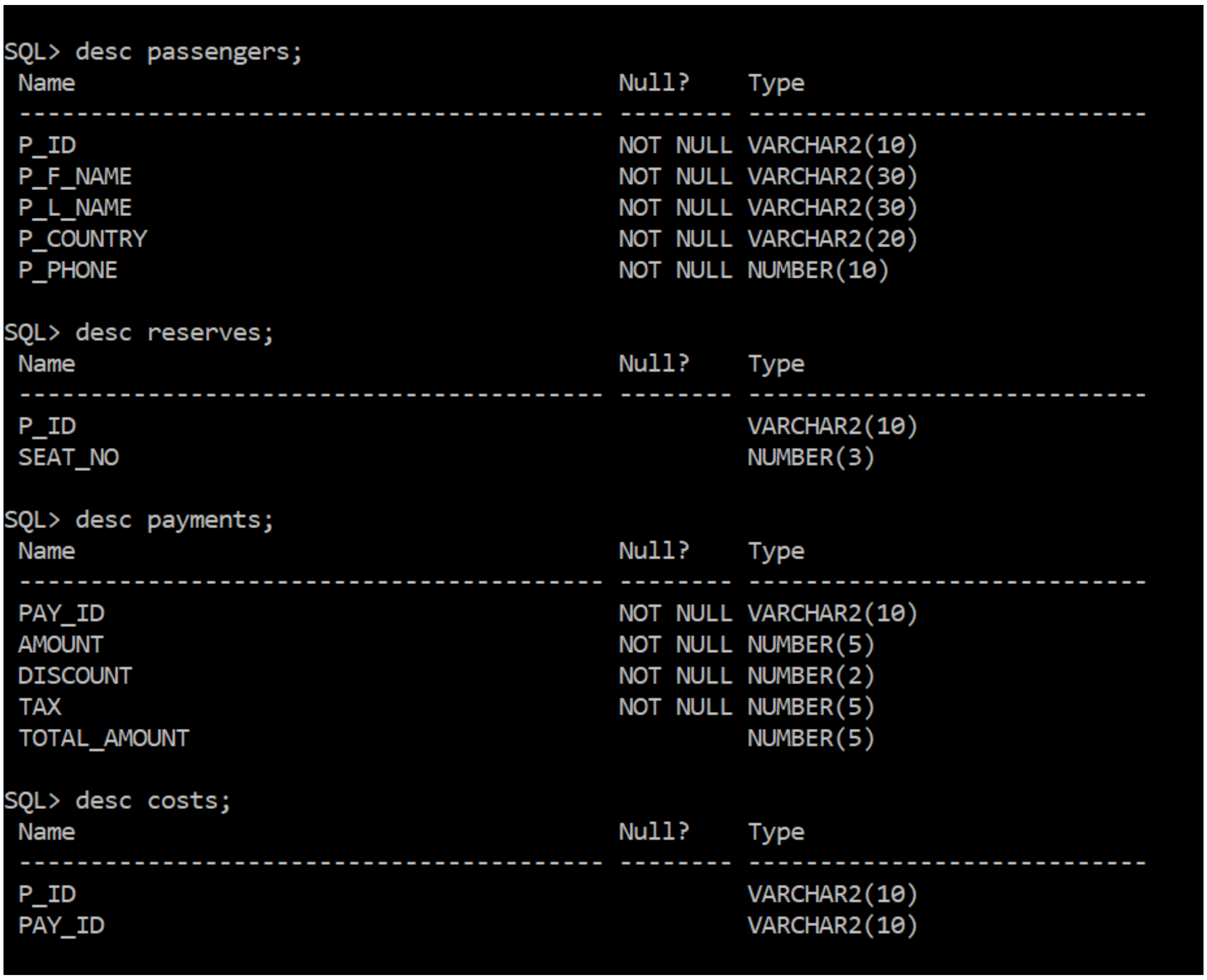
```
SQL> create table passengers(
  2  p_id varchar2(10) primary key,
  3  p_f_name varchar2(30) not null,
  4  p_l_name varchar2(30) not null,
  5  p_country varchar2(20) not null,
  6  p_phone number(10) not null);

Table created.

SQL> create table reserves(
  2  p_id varchar2(10),
  3  seat_no number(3),
  4  foreign key(p_id) references passengers(p_id),
  5  foreign key(seat_no) references seats(seat_no));

Table created.

SQL> create table payments(
  2  pay_id varchar2(10) primary key,
  3  amount number(5) not null,
  4  discount number(2) not null,
  5  tax number(5) not null,
  6  total_amount number(5));

Table created.

SQL> create table costs(
  2  p_id varchar2(10),
  3  pay_id varchar2(10),
  4  foreign key(p_id) references passengers(p_id),
  5  foreign key(pay_id) references payments(pay_id));

Table created.

SQL>
```



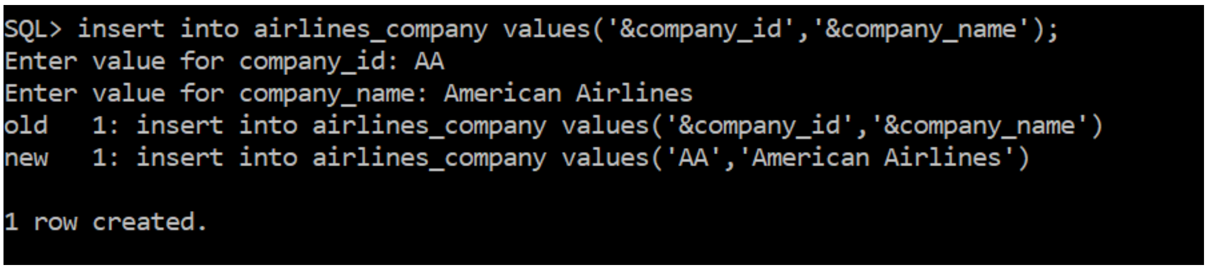
```
SQL> desc airlines_company;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COMPANY_ID                                NOT NULL VARCHAR2(10)
 COMPANY_NAME                              NOT NULL VARCHAR2(50)

SQL> desc airplane;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 AIRPLANE_NO                               NOT NULL VARCHAR2(10)
 SEATING_CAPACITY                          NOT NULL NUMBER(3)

SQL> desc owns;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COMPANY_ID                                         VARCHAR2(10)
 AIRPLANE_NO                                        VARCHAR2(10)

SQL> desc airport;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 A_ID                                      NOT NULL VARCHAR2(10)
 A_NAME                                    NOT NULL VARCHAR2(50)
 A_COUNTRY                                 NOT NULL VARCHAR2(20)
 A_STATE                                   NOT NULL VARCHAR2(20)
 A_CITY                                    NOT NULL VARCHAR2(20)
 A_ZIP                                     NOT NULL NUMBER(10)

SQL> desc acess;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 AIRPLANE_NO                                        VARCHAR2(10)
 A_ID                                               VARCHAR2(10)
```



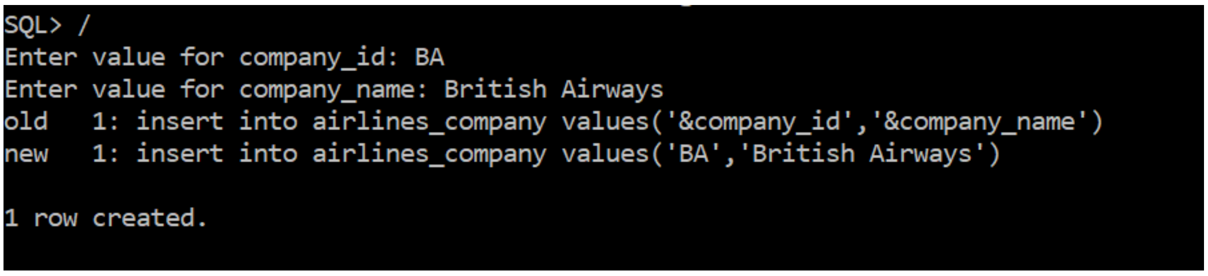
```
SQL> desc employees;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 E_ID                                      NOT NULL VARCHAR2(10)
 E_NAME                                    NOT NULL VARCHAR2(50)
 E_AGE                                     NOT NULL NUMBER
 E_PHONE                                   NOT NULL NUMBER(10)
 E_ROLE                                    NOT NULL VARCHAR2(30)
 E_SALARY                                  NOT NULL NUMBER

SQL> desc airport_has_employees;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 A_ID                                               VARCHAR2(10)
 E_ID                                               VARCHAR2(10)

SQL> desc seats;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SEAT_NO                                   NOT NULL NUMBER(3)
 CLASS                                     NOT NULL VARCHAR2(2)
 AVAILABILITY                                       VARCHAR2(3)

SQL> desc airplane_has_seats;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 AIRPLANE_NO                                        VARCHAR2(10)
 SEAT_NO                                            NUMBER(3)

SQL> desc flight_hours;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 AIRPLANE_NO                                        VARCHAR2(10)
 DEPARTURE_AIRPORT                         NOT NULL VARCHAR2(30)
 DEPARTURE_TIME                            NOT NULL VARCHAR2(10)
 ARRIVAL_AIRPORT                           NOT NULL VARCHAR2(30)
 ARRIVAL_TIME                              NOT NULL VARCHAR2(10)
```



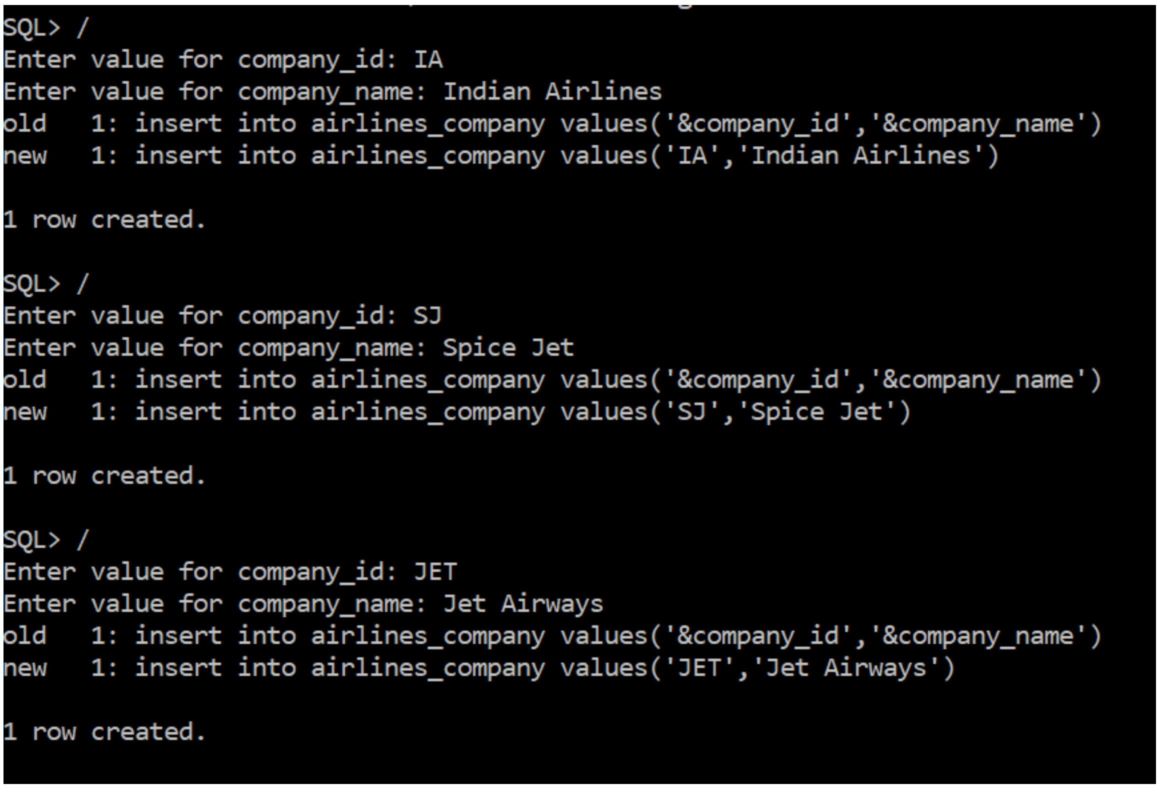
```
SQL> desc passengers;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 P_ID                                      NOT NULL VARCHAR2(10)
 P_F_NAME                                  NOT NULL VARCHAR2(30)
 P_L_NAME                                  NOT NULL VARCHAR2(30)
 P_COUNTRY                                 NOT NULL VARCHAR2(20)
 P_PHONE                                   NOT NULL NUMBER(10)

SQL> desc reserves;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 P_ID                                               VARCHAR2(10)
 SEAT_NO                                            NUMBER(3)

SQL> desc payments;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PAY_ID                                    NOT NULL VARCHAR2(10)
 AMOUNT                                    NOT NULL NUMBER(5)
 DISCOUNT                                  NOT NULL NUMBER(2)
 TAX                                       NOT NULL NUMBER(5)
 TOTAL_AMOUNT                                       NUMBER(5)

SQL> desc costs;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 P_ID                                               VARCHAR2(10)
 PAY_ID                                             VARCHAR2(10)
```

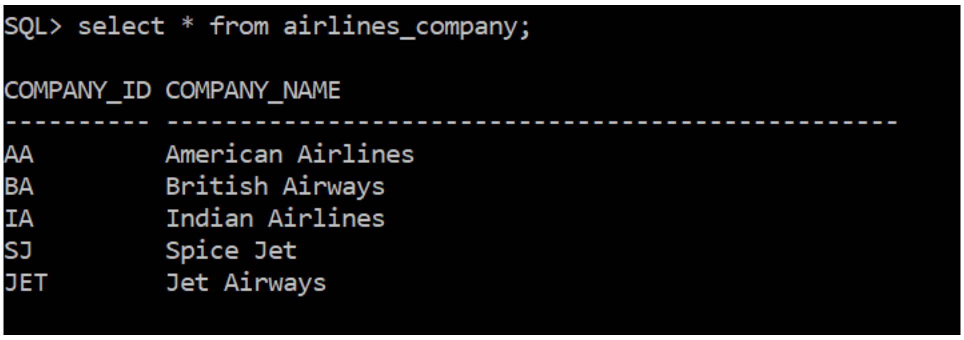


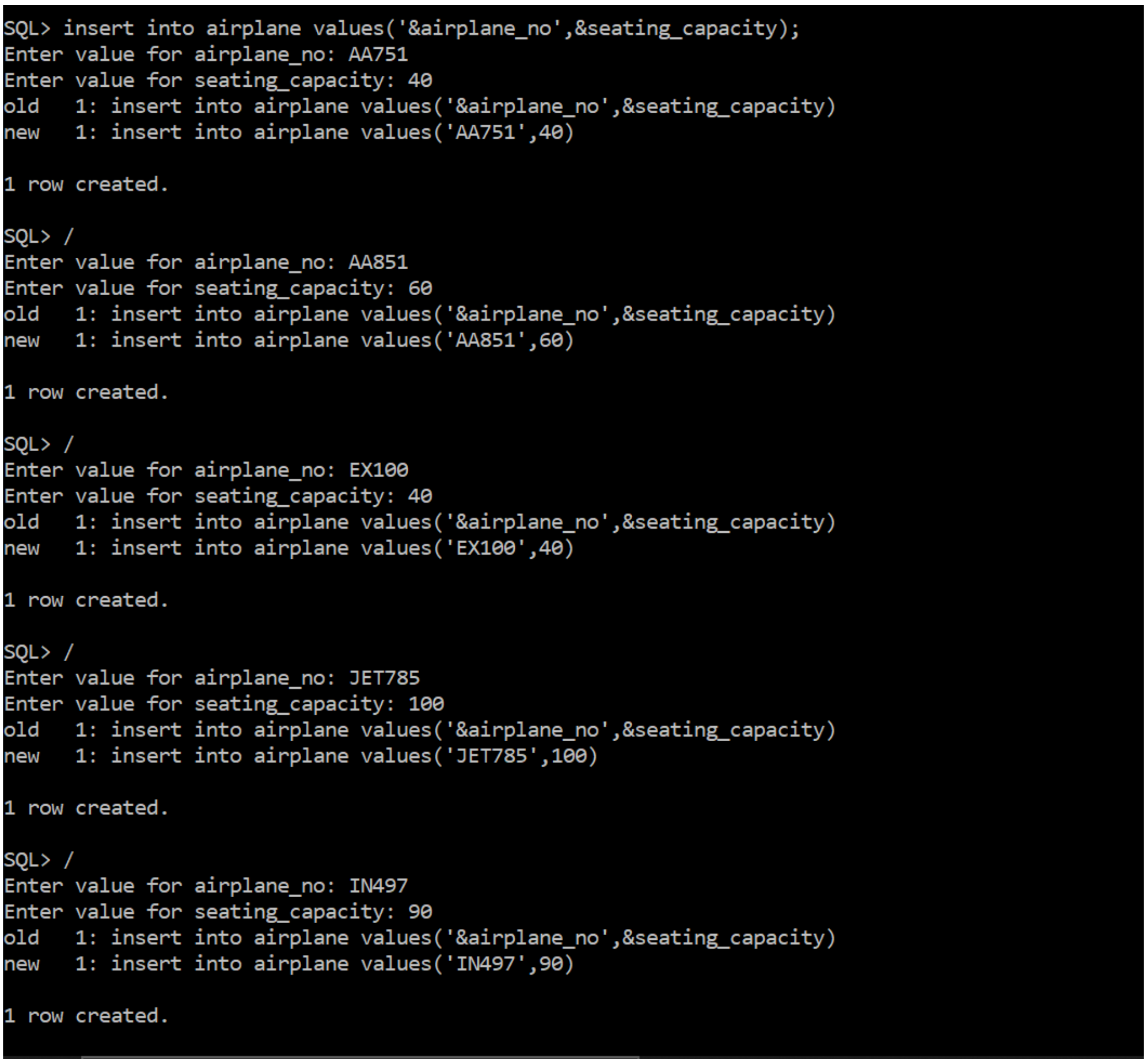
```
SQL> insert into airlines_company values('&company_id','&company_name');
Enter value for company_id: AA
Enter value for company_name: American Airlines
old   1: insert into airlines_company values('&company_id','&company_name')
new   1: insert into airlines_company values('AA','American Airlines')

1 row created.
```



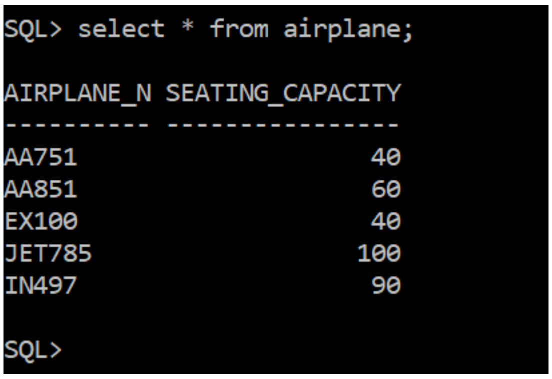
```
SQL> /
Enter value for company_id: BA
Enter value for company_name: British Airways
old   1: insert into airlines_company values('&company_id','&company_name')
new   1: insert into airlines_company values('BA','British Airways')

1 row created.
```


```
SQL> /
Enter value for company_id: IA
Enter value for company_name: Indian Airlines
old   1: insert into airlines_company values('&company_id','&company_name')
new   1: insert into airlines_company values('IA','Indian Airlines')

1 row created.

SQL> /
Enter value for company_id: SJ
Enter value for company_name: Spice Jet
old   1: insert into airlines_company values('&company_id','&company_name')
new   1: insert into airlines_company values('SJ','Spice Jet')

1 row created.

SQL> /
Enter value for company_id: JET
Enter value for company_name: Jet Airways
old   1: insert into airlines_company values('&company_id','&company_name')
new   1: insert into airlines_company values('JET','Jet Airways')

1 row created.
```

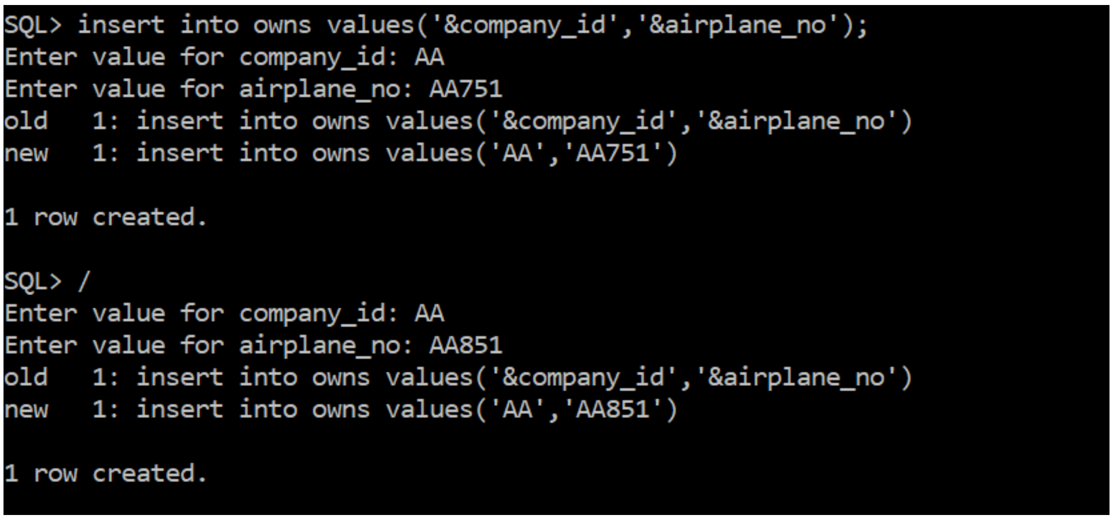

```
SQL> select * from airlines_company;

COMPANY_ID COMPANY_NAME
---------- --------------------------------------------------
AA         American Airlines
BA         British Airways
IA         Indian Airlines
SJ         Spice Jet
JET        Jet Airways
```



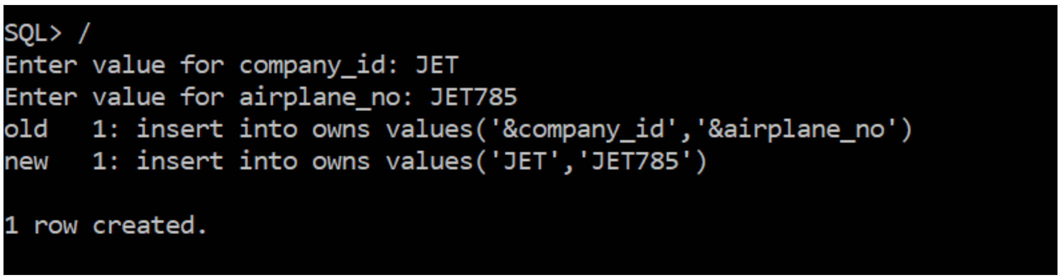
```
SQL> insert into airplane values('&airplane_no',&seating_capacity);
Enter value for airplane_no: AA751
Enter value for seating_capacity: 40
old   1: insert into airplane values('&airplane_no',&seating_capacity)
new   1: insert into airplane values('AA751',40)

1 row created.

SQL> /
Enter value for airplane_no: AA851
Enter value for seating_capacity: 60
old   1: insert into airplane values('&airplane_no',&seating_capacity)
new   1: insert into airplane values('AA851',60)

1 row created.

SQL> /
Enter value for airplane_no: EX100
Enter value for seating_capacity: 40
old   1: insert into airplane values('&airplane_no',&seating_capacity)
new   1: insert into airplane values('EX100',40)

1 row created.

SQL> /
Enter value for airplane_no: JET785
Enter value for seating_capacity: 100
old   1: insert into airplane values('&airplane_no',&seating_capacity)
new   1: insert into airplane values('JET785',100)

1 row created.

SQL> /
Enter value for airplane_no: IN497
Enter value for seating_capacity: 90
old   1: insert into airplane values('&airplane_no',&seating_capacity)
new   1: insert into airplane values('IN497',90)

1 row created.
```



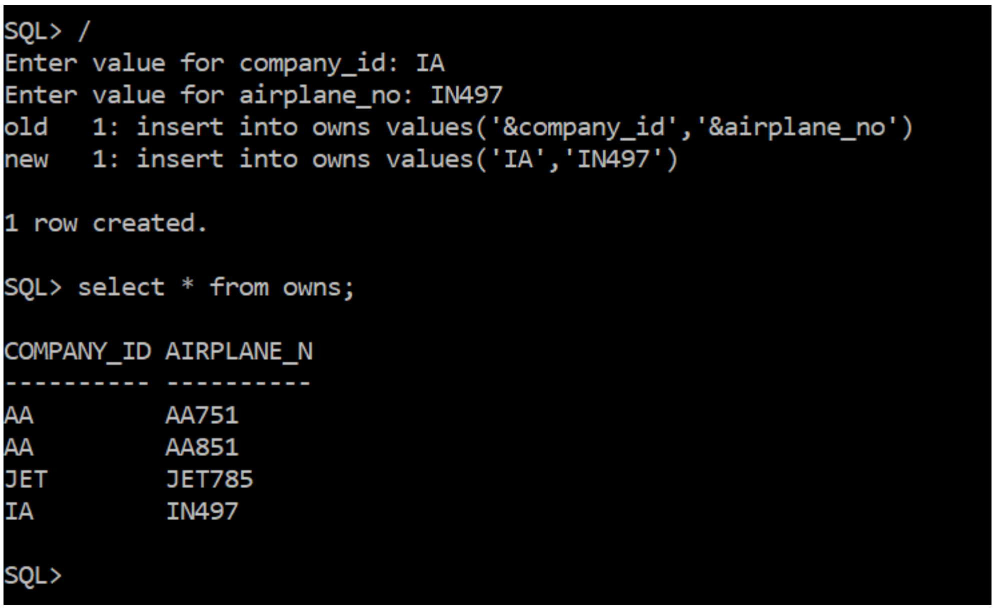
```
SQL> select * from airplane;

AIRPLANE_N SEATING_CAPACITY
---------- ----------------
AA751                    40
AA851                    60
EX100                    40
JET785                  100
IN497                    90

SQL>
```



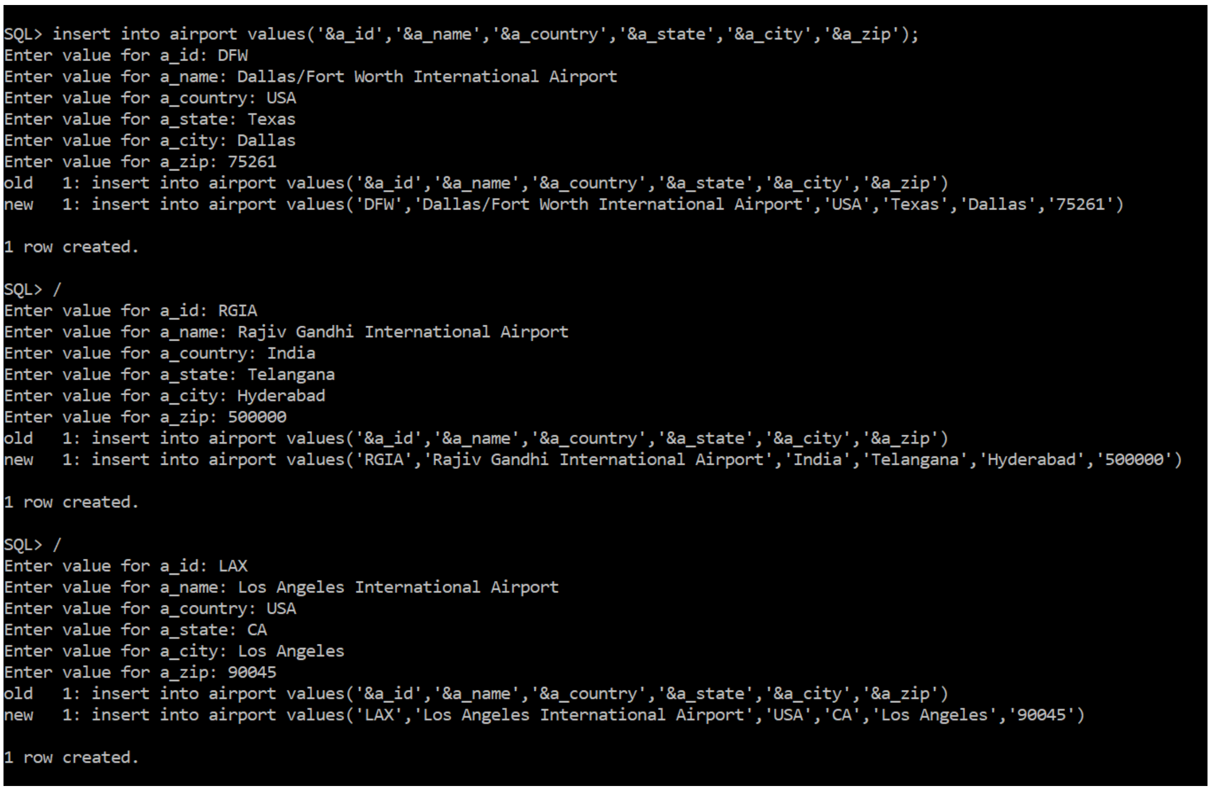
```
SQL> insert into owns values('&company_id','&airplane_no');
Enter value for company_id: AA
Enter value for airplane_no: AA751
old   1: insert into owns values('&company_id','&airplane_no')
new   1: insert into owns values('AA','AA751')

1 row created.

SQL> /
Enter value for company_id: AA
Enter value for airplane_no: AA851
old   1: insert into owns values('&company_id','&airplane_no')
new   1: insert into owns values('AA','AA851')

1 row created.
```



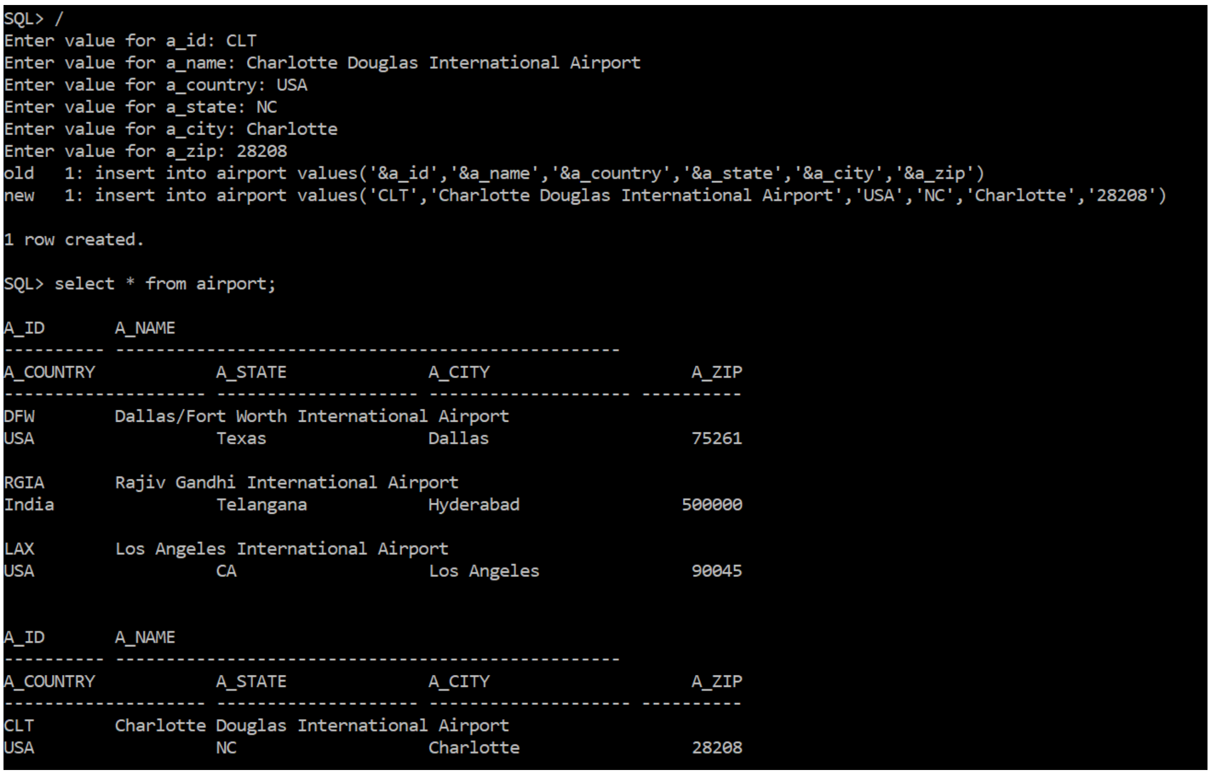
```
SQL> /
Enter value for company_id: JET
Enter value for airplane_no: JET785
old   1: insert into owns values('&company_id','&airplane_no')
new   1: insert into owns values('JET','JET785')

1 row created.
```



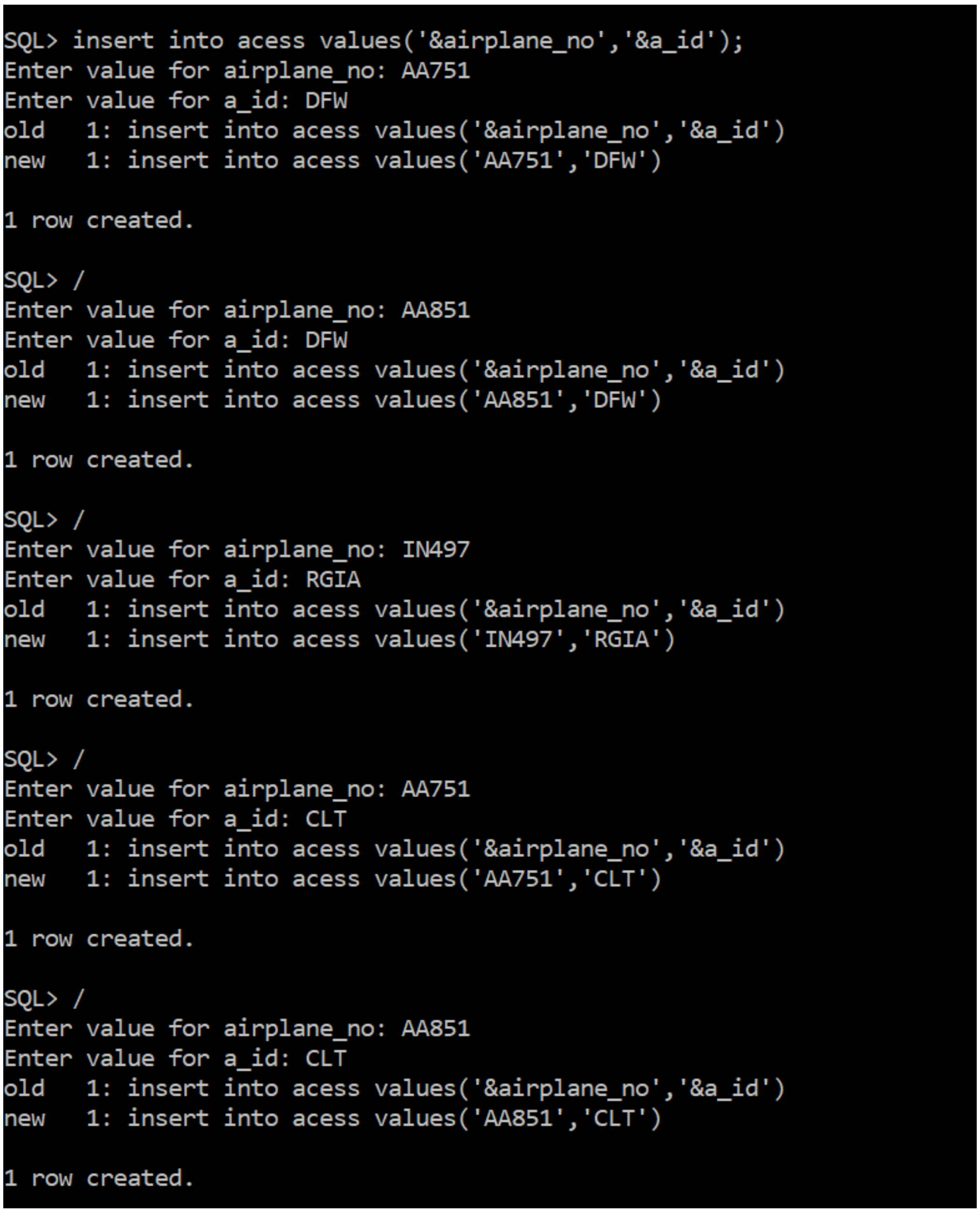
```
SQL> /
Enter value for company_id: IA
Enter value for airplane_no: IN497
old   1: insert into owns values('&company_id','&airplane_no')
new   1: insert into owns values('IA','IN497')

1 row created.

SQL> select * from owns;

COMPANY_ID AIRPLANE_N
---------- ----------
AA         AA751
AA         AA851
JET        JET785
IA         IN497

SQL>
```



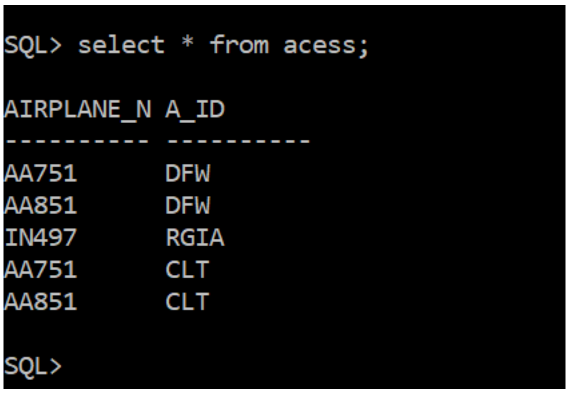
```
SQL> insert into airport values('&a_id','&a_name','&a_country','&a_state','&a_city','&a_zip');
Enter value for a_id: DFW
Enter value for a_name: Dallas/Fort Worth International Airport
Enter value for a_country: USA
Enter value for a_state: Texas
Enter value for a_city: Dallas
Enter value for a_zip: 75261
old   1: insert into airport values('&a_id','&a_name','&a_country','&a_state','&a_city','&a_zip')
new   1: insert into airport values('DFW','Dallas/Fort Worth International Airport','USA','Texas','Dallas','75261')

1 row created.

SQL> /
Enter value for a_id: RGIA
Enter value for a_name: Rajiv Gandhi International Airport
Enter value for a_country: India
Enter value for a_state: Telangana
Enter value for a_city: Hyderabad
Enter value for a_zip: 500000
old   1: insert into airport values('&a_id','&a_name','&a_country','&a_state','&a_city','&a_zip')
new   1: insert into airport values('RGIA','Rajiv Gandhi International Airport','India','Telangana','Hyderabad','500000')

1 row created.

SQL> /
Enter value for a_id: LAX
Enter value for a_name: Los Angeles International Airport
Enter value for a_country: USA
Enter value for a_state: CA
Enter value for a_city: Los Angeles
Enter value for a_zip: 90045
old   1: insert into airport values('&a_id','&a_name','&a_country','&a_state','&a_city','&a_zip')
new   1: insert into airport values('LAX','Los Angeles International Airport','USA','CA','Los Angeles','90045')

1 row created.
```



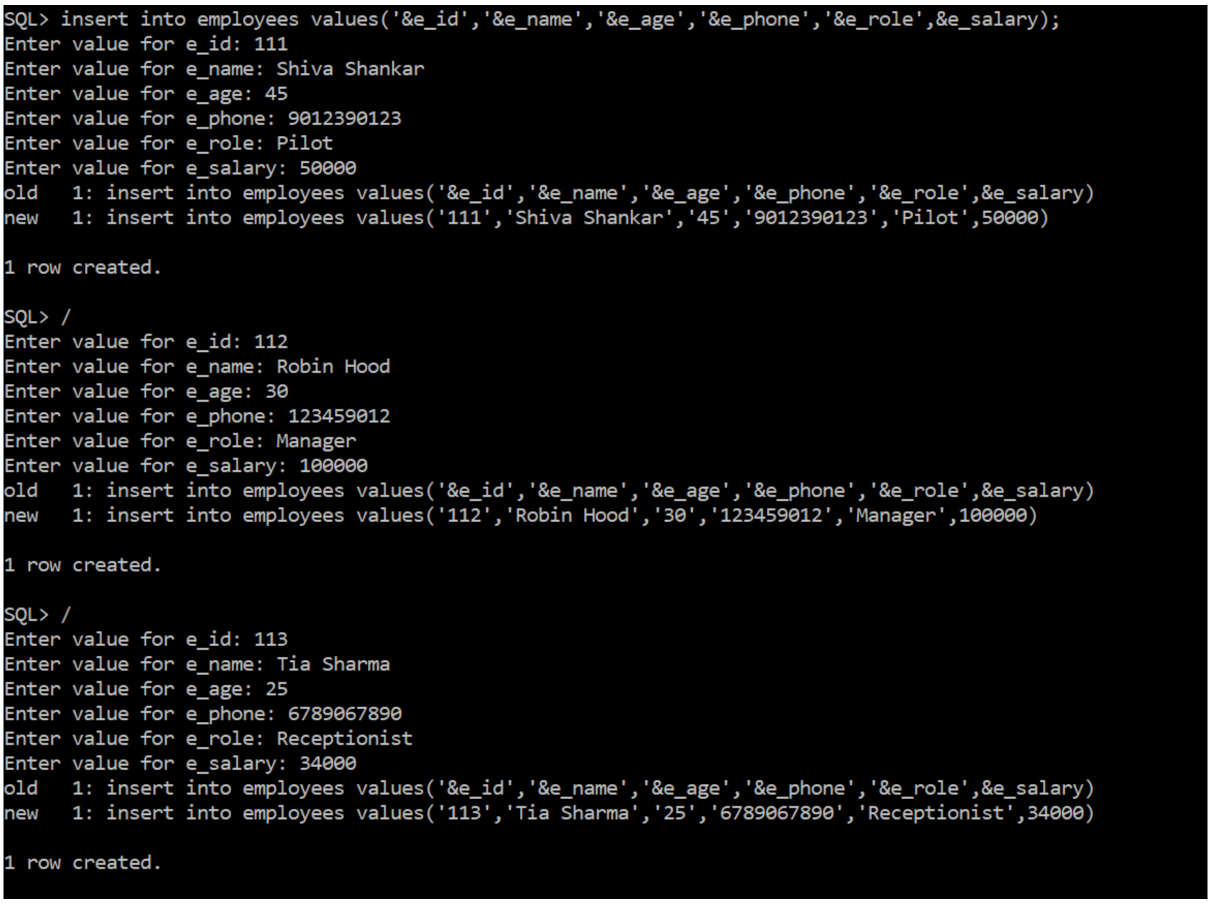
```
SQL> /
Enter value for a_id: CLT
Enter value for a_name: Charlotte Douglas International Airport
Enter value for a_country: USA
Enter value for a_state: NC
Enter value for a_city: Charlotte
Enter value for a_zip: 28208
old   1: insert into airport values('&a_id','&a_name','&a_country','&a_state','&a_city','&a_zip')
new   1: insert into airport values('CLT','Charlotte Douglas International Airport','USA','NC','Charlotte','28208')

1 row created.

SQL> select * from airport;

A_ID       A_NAME
---------- --------------------------------------------------
A_COUNTRY            A_STATE              A_CITY               A_ZIP
-------------------- -------------------- -------------------- ----------
DFW        Dallas/Fort Worth International Airport
USA                  Texas                Dallas               75261

RGIA       Rajiv Gandhi International Airport
India                Telangana            Hyderabad            500000

LAX        Los Angeles International Airport
USA                  CA                   Los Angeles          90045


A_ID       A_NAME
---------- --------------------------------------------------
A_COUNTRY            A_STATE              A_CITY               A_ZIP
-------------------- -------------------- -------------------- ----------
CLT        Charlotte Douglas International Airport
USA                  NC                   Charlotte            28208
```



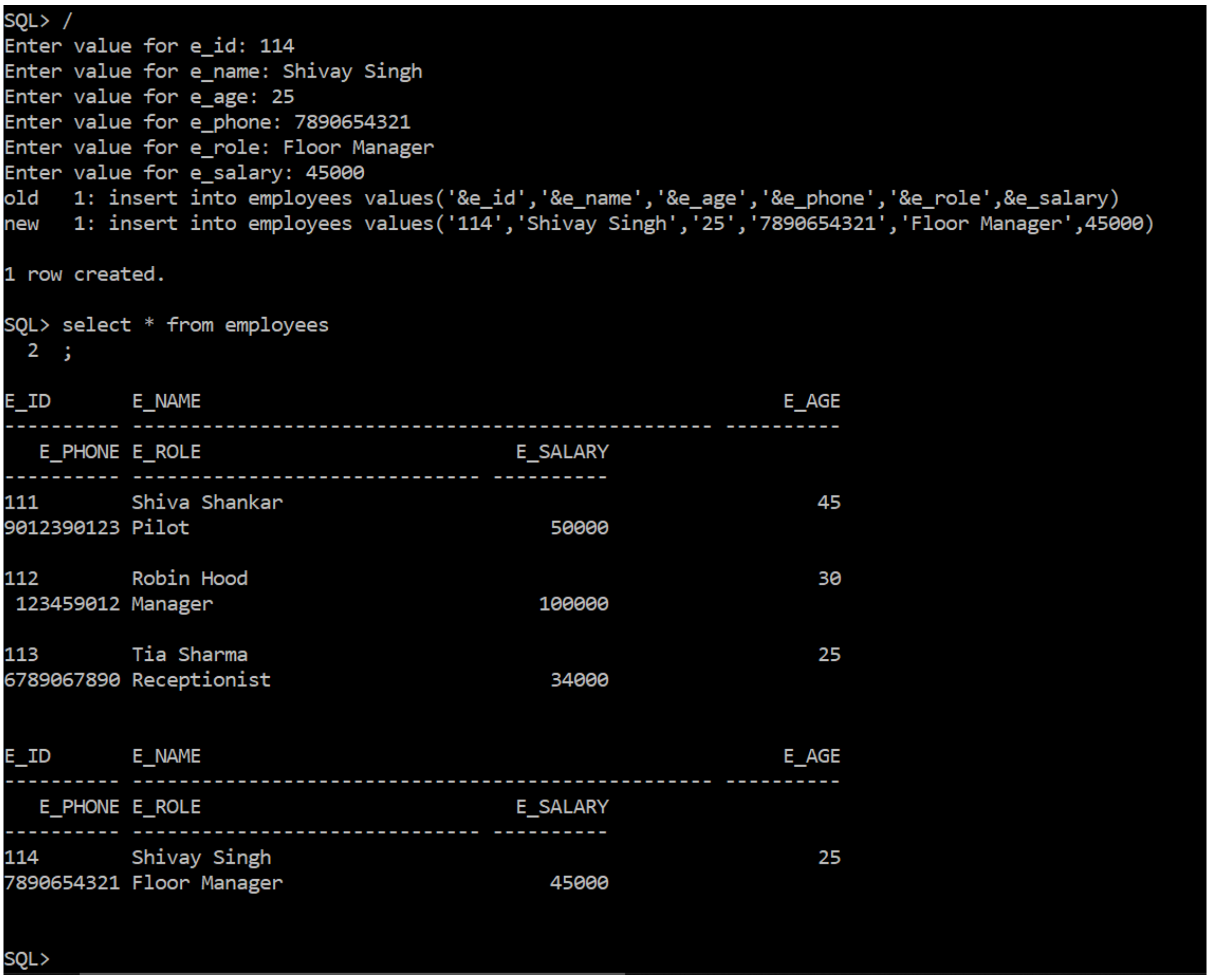
```
SQL> insert into acess values('&airplane_no','&a_id');
Enter value for airplane_no: AA751
Enter value for a_id: DFW
old   1: insert into acess values('&airplane_no','&a_id')
new   1: insert into acess values('AA751','DFW')

1 row created.

SQL> /
Enter value for airplane_no: AA851
Enter value for a_id: DFW
old   1: insert into acess values('&airplane_no','&a_id')
new   1: insert into acess values('AA851','DFW')

1 row created.

SQL> /
Enter value for airplane_no: IN497
Enter value for a_id: RGIA
old   1: insert into acess values('&airplane_no','&a_id')
new   1: insert into acess values('IN497','RGIA')

1 row created.

SQL> /
Enter value for airplane_no: AA751
Enter value for a_id: CLT
old   1: insert into acess values('&airplane_no','&a_id')
new   1: insert into acess values('AA751','CLT')

1 row created.

SQL> /
Enter value for airplane_no: AA851
Enter value for a_id: CLT
old   1: insert into acess values('&airplane_no','&a_id')
new   1: insert into acess values('AA851','CLT')

1 row created.
```



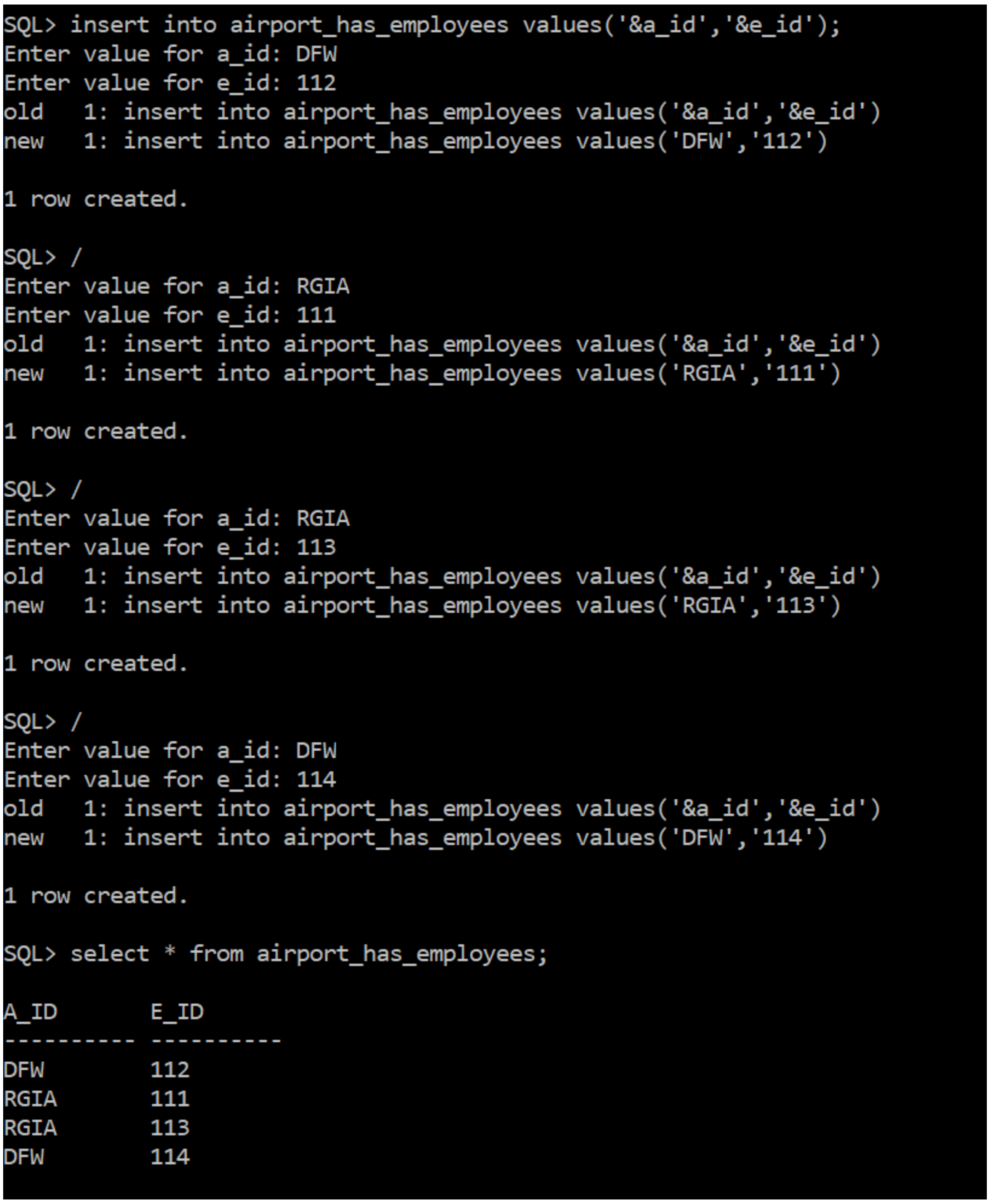
```
SQL> select * from acess;

AIRPLANE_N A_ID
---------- ----------
AA751      DFW
AA851      DFW
IN497      RGIA
AA751      CLT
AA851      CLT

SQL>
```



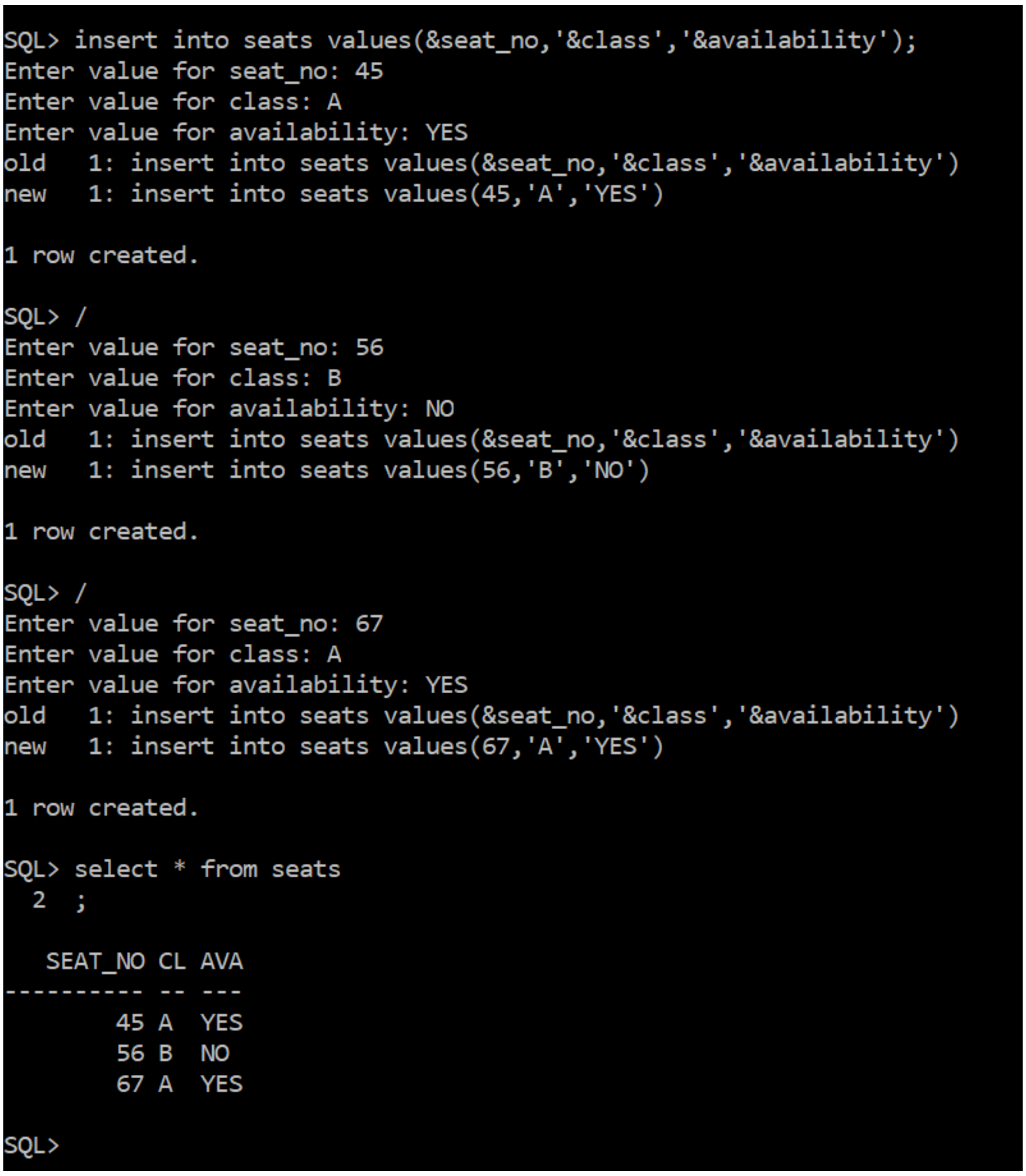
```
SQL> insert into employees values('&e_id','&e_name','&e_age','&e_phone','&e_role',&e_salary);
Enter value for e_id: 111
Enter value for e_name: Shiva Shankar
Enter value for e_age: 45
Enter value for e_phone: 9012390123
Enter value for e_role: Pilot
Enter value for e_salary: 50000
old   1: insert into employees values('&e_id','&e_name','&e_age','&e_phone','&e_role',&e_salary)
new   1: insert into employees values('111','Shiva Shankar','45','9012390123','Pilot',50000)

1 row created.

SQL> /
Enter value for e_id: 112
Enter value for e_name: Robin Hood
Enter value for e_age: 30
Enter value for e_phone: 123459012
Enter value for e_role: Manager
Enter value for e_salary: 100000
old   1: insert into employees values('&e_id','&e_name','&e_age','&e_phone','&e_role',&e_salary)
new   1: insert into employees values('112','Robin Hood','30','123459012','Manager',100000)

1 row created.

SQL> /
Enter value for e_id: 113
Enter value for e_name: Tia Sharma
Enter value for e_age: 25
Enter value for e_phone: 6789067890
Enter value for e_role: Receptionist
Enter value for e_salary: 34000
old   1: insert into employees values('&e_id','&e_name','&e_age','&e_phone','&e_role',&e_salary)
new   1: insert into employees values('113','Tia Sharma','25','6789067890','Receptionist',34000)

1 row created.
```



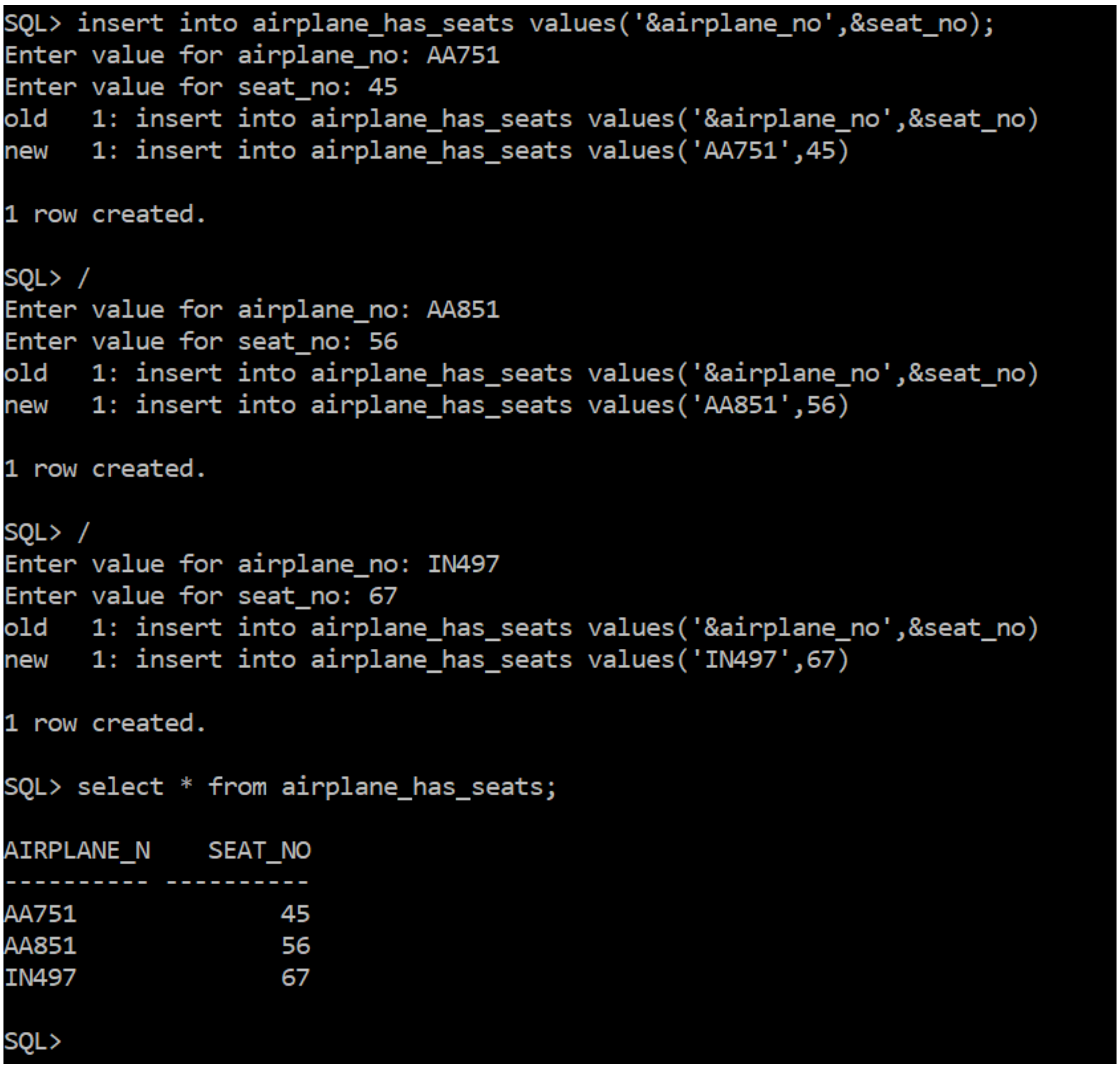
```
SQL> /
Enter value for e_id: 114
Enter value for e_name: Shivay Singh
Enter value for e_age: 25
Enter value for e_phone: 7890654321
Enter value for e_role: Floor Manager
Enter value for e_salary: 45000
old   1: insert into employees values('&e_id','&e_name','&e_age','&e_phone','&e_role',&e_salary)
new   1: insert into employees values('114','Shivay Singh','25','7890654321','Floor Manager',45000)

1 row created.

SQL> select * from employees
  2  ;

E_ID       E_NAME                                             E_AGE
---------- -------------------------------------------------- ----------
   E_PHONE E_ROLE                           E_SALARY
---------- ------------------------------ ----------
111        Shiva Shankar                                              45
9012390123 Pilot                               50000

112        Robin Hood                                                 30
 123459012 Manager                            100000

113        Tia Sharma                                                 25
6789067890 Receptionist                        34000


E_ID       E_NAME                                             E_AGE
---------- -------------------------------------------------- ----------
   E_PHONE E_ROLE                           E_SALARY
---------- ------------------------------ ----------
114        Shivay Singh                                               25
7890654321 Floor Manager                       45000


SQL>
```



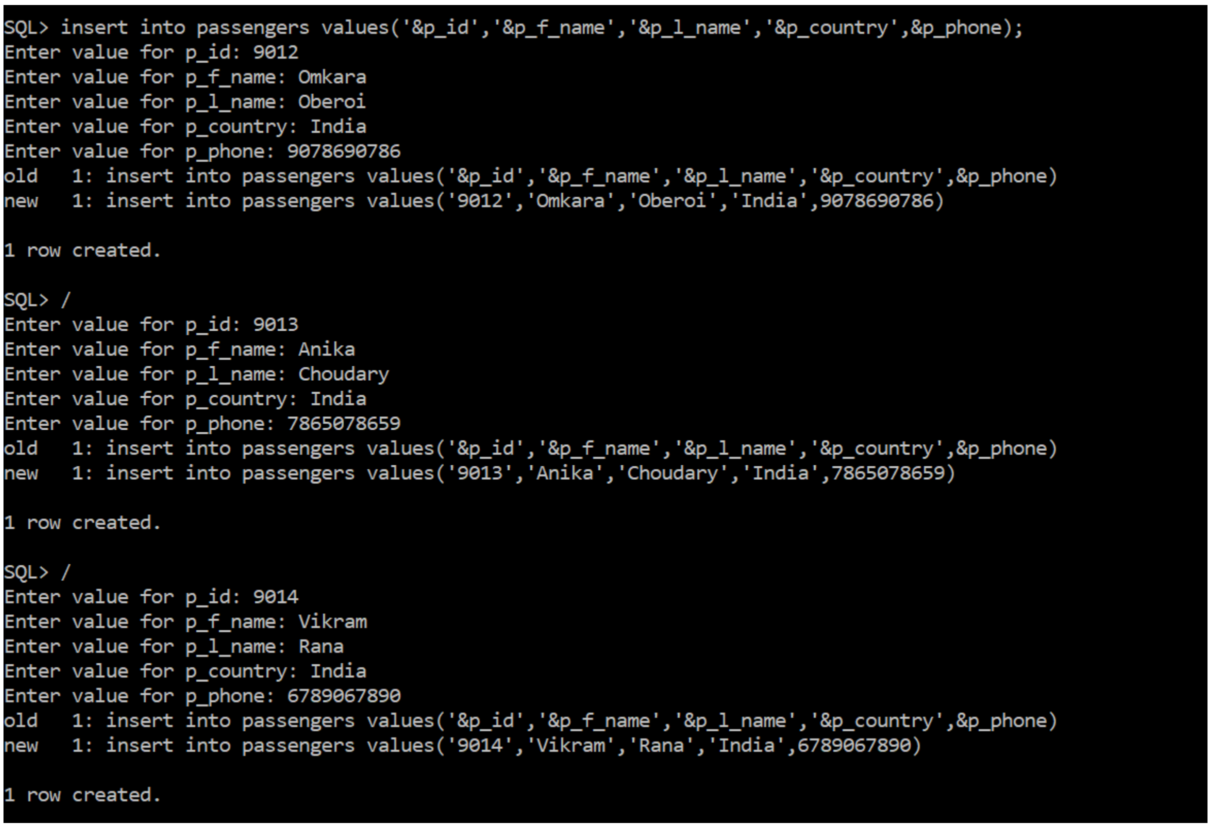
```
SQL> insert into airport_has_employees values('&a_id','&e_id');
Enter value for a_id: DFW
Enter value for e_id: 112
old   1: insert into airport_has_employees values('&a_id','&e_id')
new   1: insert into airport_has_employees values('DFW','112')

1 row created.

SQL> /
Enter value for a_id: RGIA
Enter value for e_id: 111
old   1: insert into airport_has_employees values('&a_id','&e_id')
new   1: insert into airport_has_employees values('RGIA','111')

1 row created.

SQL> /
Enter value for a_id: RGIA
Enter value for e_id: 113
old   1: insert into airport_has_employees values('&a_id','&e_id')
new   1: insert into airport_has_employees values('RGIA','113')

1 row created.

SQL> /
Enter value for a_id: DFW
Enter value for e_id: 114
old   1: insert into airport_has_employees values('&a_id','&e_id')
new   1: insert into airport_has_employees values('DFW','114')

1 row created.

SQL> select * from airport_has_employees;

A_ID       E_ID
---------- ----------
DFW        112
RGIA       111
RGIA       113
DFW        114
```



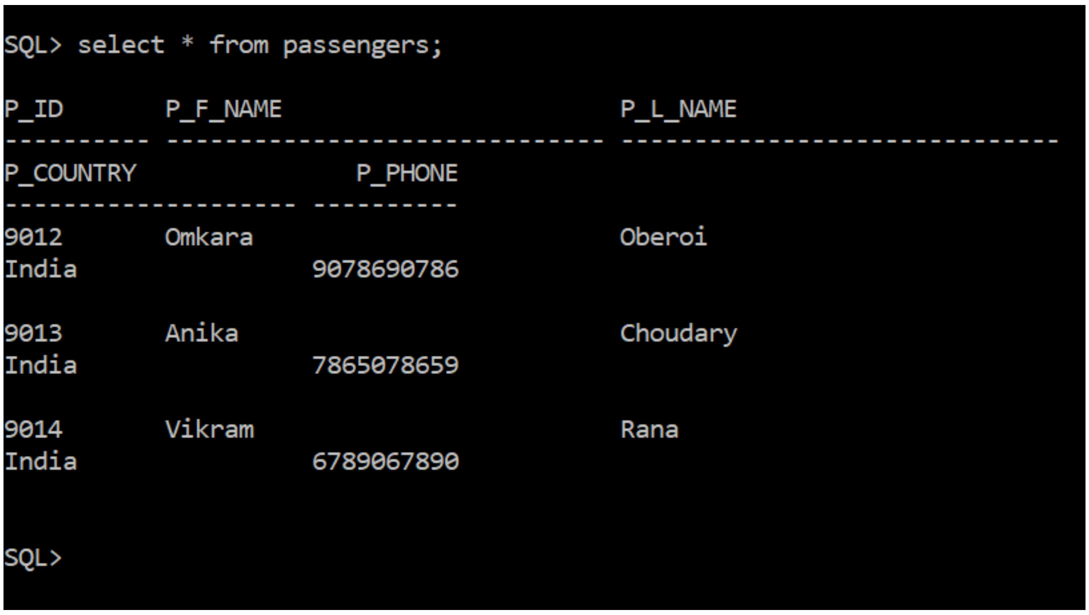
```
SQL> insert into seats values(&seat_no,'&class','&availability');
Enter value for seat_no: 45
Enter value for class: A
Enter value for availability: YES
old   1: insert into seats values(&seat_no,'&class','&availability')
new   1: insert into seats values(45,'A','YES')

1 row created.

SQL> /
Enter value for seat_no: 56
Enter value for class: B
Enter value for availability: NO
old   1: insert into seats values(&seat_no,'&class','&availability')
new   1: insert into seats values(56,'B','NO')

1 row created.

SQL> /
Enter value for seat_no: 67
Enter value for class: A
Enter value for availability: YES
old   1: insert into seats values(&seat_no,'&class','&availability')
new   1: insert into seats values(67,'A','YES')

1 row created.

SQL> select * from seats
  2  ;

   SEAT_NO CL AVA
---------- -- ---
        45 A  YES
        56 B  NO
        67 A  YES

SQL>
```



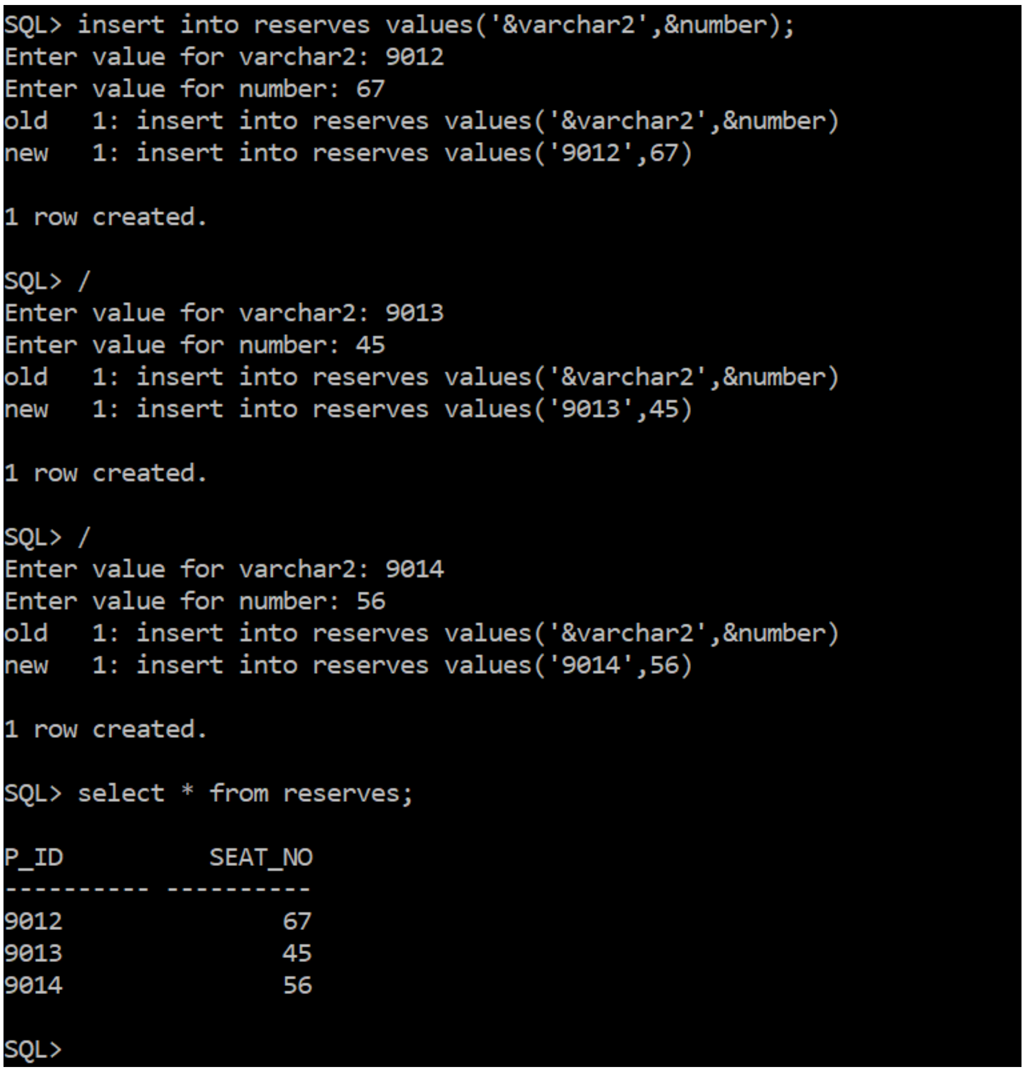
```
SQL> insert into airplane_has_seats values('&airplane_no',&seat_no);
Enter value for airplane_no: AA751
Enter value for seat_no: 45
old   1: insert into airplane_has_seats values('&airplane_no',&seat_no)
new   1: insert into airplane_has_seats values('AA751',45)

1 row created.

SQL> /
Enter value for airplane_no: AA851
Enter value for seat_no: 56
old   1: insert into airplane_has_seats values('&airplane_no',&seat_no)
new   1: insert into airplane_has_seats values('AA851',56)

1 row created.

SQL> /
Enter value for airplane_no: IN497
Enter value for seat_no: 67
old   1: insert into airplane_has_seats values('&airplane_no',&seat_no)
new   1: insert into airplane_has_seats values('IN497',67)

1 row created.

SQL> select * from airplane_has_seats;

AIRPLANE_N    SEAT_NO
---------- ----------
AA751              45
AA851              56
IN497              67

SQL>
```

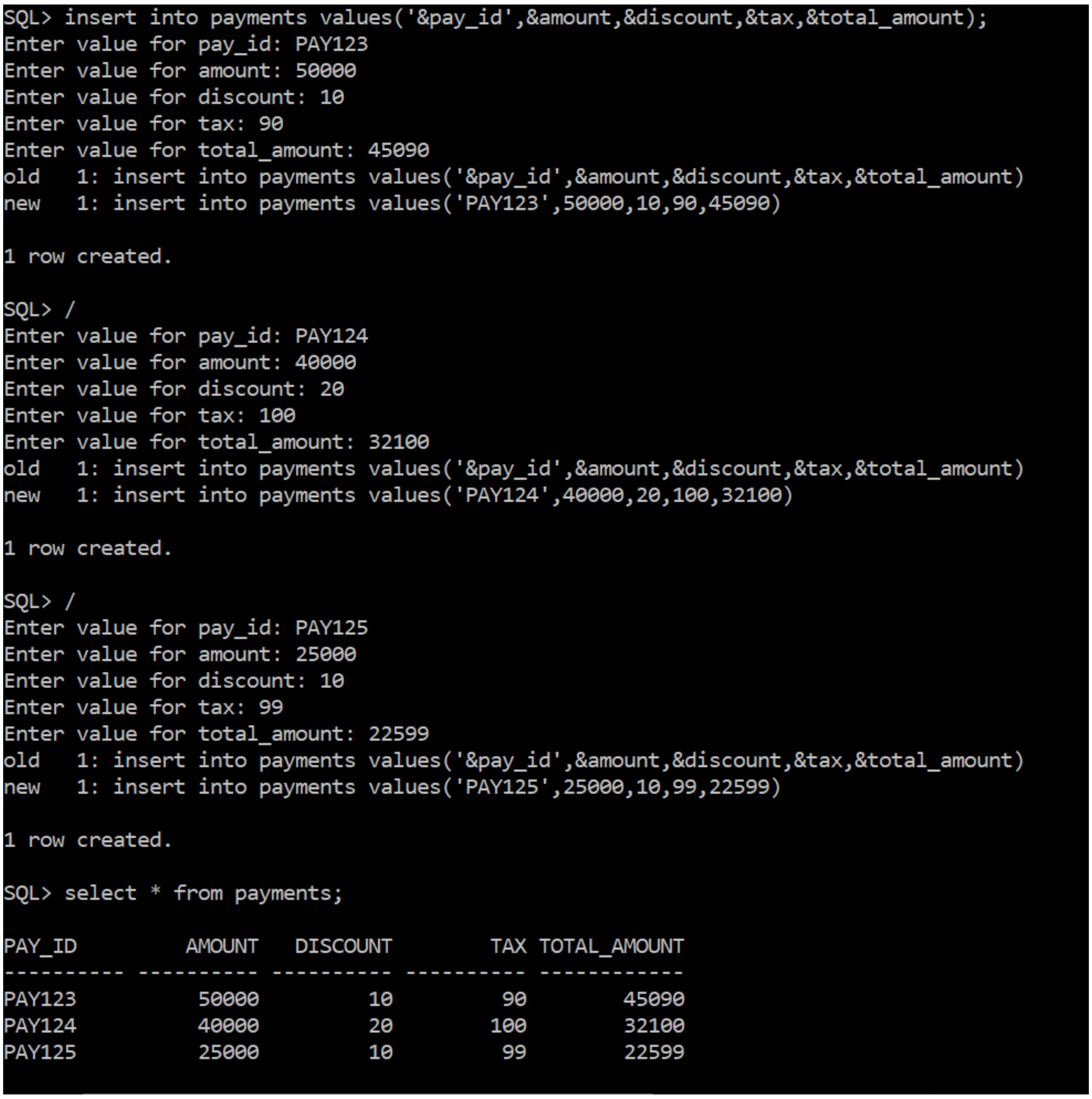


```
SQL> insert into passengers values('&p_id','&p_f_name','&p_l_name','&p_country',&p_phone);
Enter value for p_id: 9012
Enter value for p_f_name: Omkara
Enter value for p_l_name: Oberoi
Enter value for p_country: India
Enter value for p_phone: 9078690786
old   1: insert into passengers values('&p_id','&p_f_name','&p_l_name','&p_country',&p_phone)
new   1: insert into passengers values('9012','Omkara','Oberoi','India',9078690786)

1 row created.

SQL> /
Enter value for p_id: 9013
Enter value for p_f_name: Anika
Enter value for p_l_name: Choudary
Enter value for p_country: India
Enter value for p_phone: 7865078659
old   1: insert into passengers values('&p_id','&p_f_name','&p_l_name','&p_country',&p_phone)
new   1: insert into passengers values('9013','Anika','Choudary','India',7865078659)

1 row created.

SQL> /
Enter value for p_id: 9014
Enter value for p_f_name: Vikram
Enter value for p_l_name: Rana
Enter value for p_country: India
Enter value for p_phone: 6789067890
old   1: insert into passengers values('&p_id','&p_f_name','&p_l_name','&p_country',&p_phone)
new   1: insert into passengers values('9014','Vikram','Rana','India',6789067890)

1 row created.
```

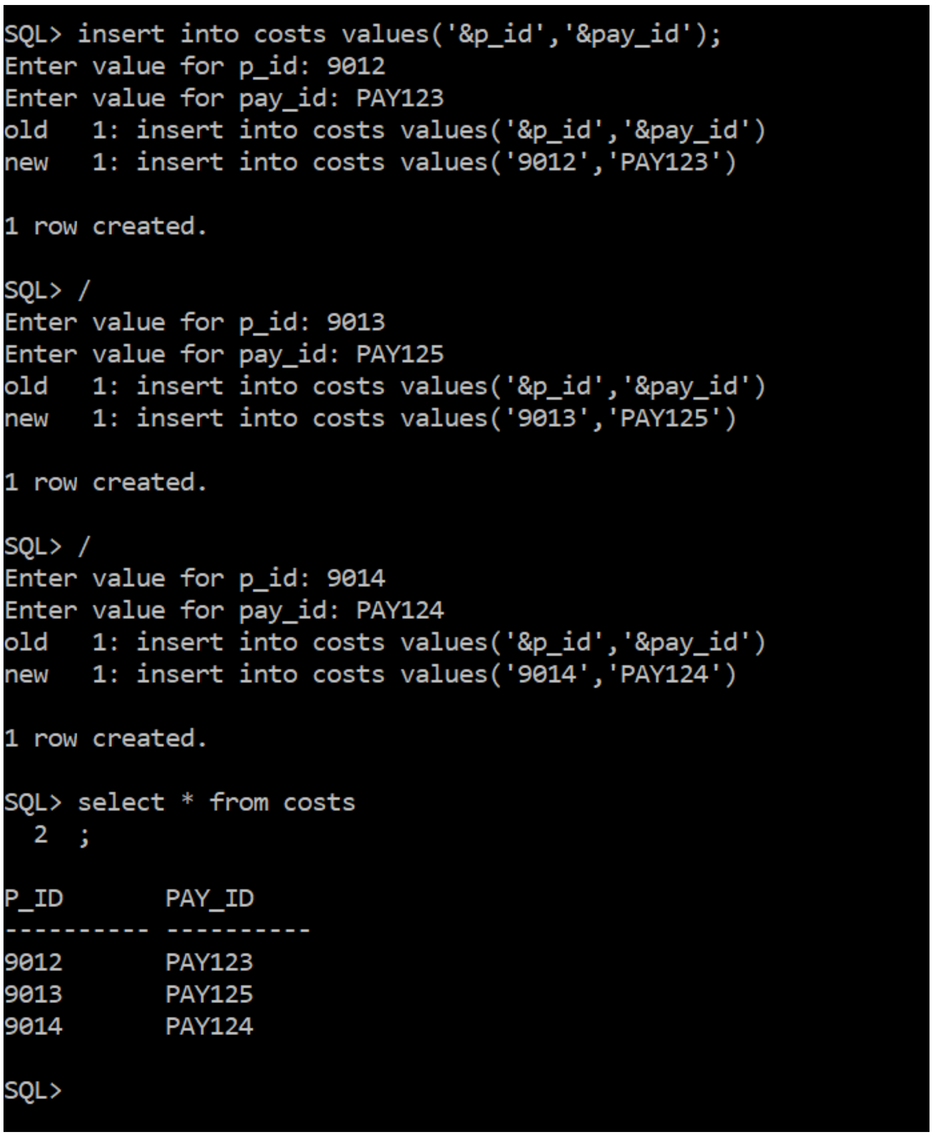


```
SQL> select * from passengers;

P_ID       P_F_NAME                       P_L_NAME
---------- ------------------------------ ------------------------------
P_COUNTRY            P_PHONE
-------------------- ----------
9012       Omkara                         Oberoi
India                9078690786

9013       Anika                          Choudary
India                7865078659

9014       Vikram                         Rana
India                6789067890


SQL>
```

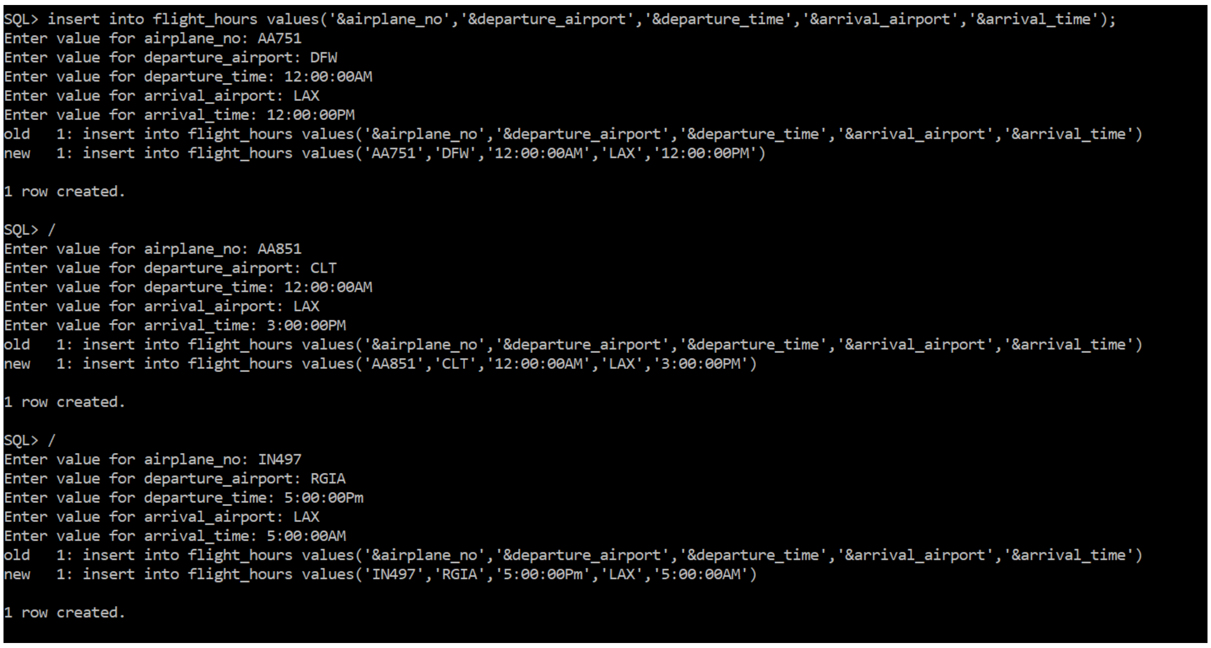

```
SQL> insert into reserves values('&varchar2',&number);
Enter value for varchar2: 9012
Enter value for number: 67
old   1: insert into reserves values('&varchar2',&number)
new   1: insert into reserves values('9012',67)

1 row created.

SQL> /
Enter value for varchar2: 9013
Enter value for number: 45
old   1: insert into reserves values('&varchar2',&number)
new   1: insert into reserves values('9013',45)

1 row created.

SQL> /
Enter value for varchar2: 9014
Enter value for number: 56
old   1: insert into reserves values('&varchar2',&number)
new   1: insert into reserves values('9014',56)

1 row created.

SQL> select * from reserves;

P_ID          SEAT_NO
---------- ----------
9012               67
9013               45
9014               56

SQL>
```



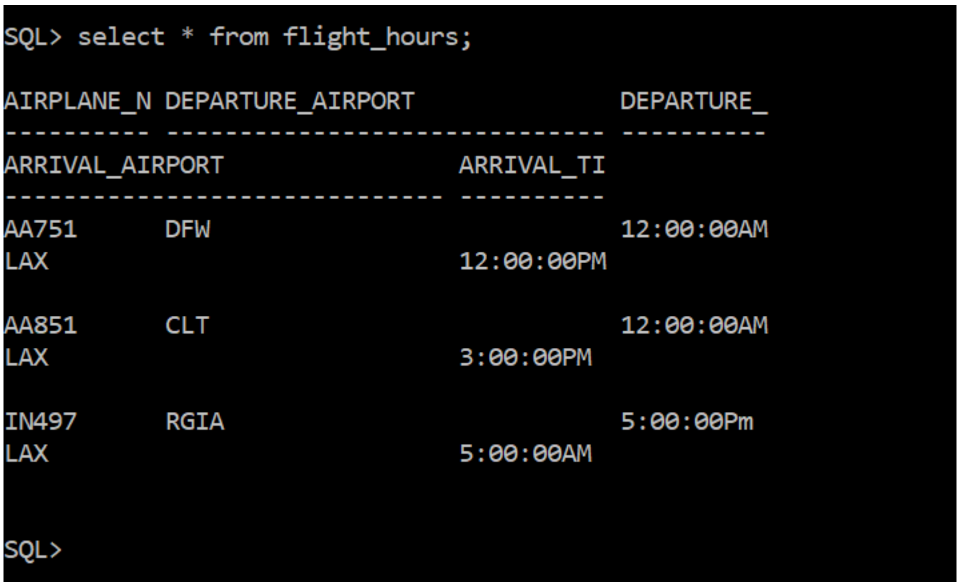
```
SQL> insert into payments values('&pay_id',&amount,&discount,&tax,&total_amount);
Enter value for pay_id: PAY123
Enter value for amount: 50000
Enter value for discount: 10
Enter value for tax: 90
Enter value for total_amount: 45090
old   1: insert into payments values('&pay_id',&amount,&discount,&tax,&total_amount)
new   1: insert into payments values('PAY123',50000,10,90,45090)

1 row created.

SQL> /
Enter value for pay_id: PAY124
Enter value for amount: 40000
Enter value for discount: 20
Enter value for tax: 100
Enter value for total_amount: 32100
old   1: insert into payments values('&pay_id',&amount,&discount,&tax,&total_amount)
new   1: insert into payments values('PAY124',40000,20,100,32100)

1 row created.

SQL> /
Enter value for pay_id: PAY125
Enter value for amount: 25000
Enter value for discount: 10
Enter value for tax: 99
Enter value for total_amount: 22599
old   1: insert into payments values('&pay_id',&amount,&discount,&tax,&total_amount)
new   1: insert into payments values('PAY125',25000,10,99,22599)

1 row created.

SQL> select * from payments;

PAY_ID         AMOUNT   DISCOUNT        TAX TOTAL_AMOUNT
---------- ---------- ---------- ---------- ------------
PAY123          50000         10         90        45090
PAY124          40000         20        100        32100
PAY125          25000         10         99        22599
```



```
SQL> insert into costs values('&p_id','&pay_id');
Enter value for p_id: 9012
Enter value for pay_id: PAY123
old   1: insert into costs values('&p_id','&pay_id')
new   1: insert into costs values('9012','PAY123')

1 row created.

SQL> /
Enter value for p_id: 9013
Enter value for pay_id: PAY125
old   1: insert into costs values('&p_id','&pay_id')
new   1: insert into costs values('9013','PAY125')

1 row created.

SQL> /
Enter value for p_id: 9014
Enter value for pay_id: PAY124
old   1: insert into costs values('&p_id','&pay_id')
new   1: insert into costs values('9014','PAY124')

1 row created.

SQL> select * from costs
  2  ;

P_ID       PAY_ID
---------- ----------
9012       PAY123
9013       PAY125
9014       PAY124

SQL>
```



```
SQL> insert into flight_hours values('&airplane_no','&departure_airport','&departure_time','&arrival_airport','&arrival_time');
Enter value for airplane_no: AA751
Enter value for departure_airport: DFW
Enter value for departure_time: 12:00:00AM
Enter value for arrival_airport: LAX
Enter value for arrival_time: 12:00:00PM
old   1: insert into flight_hours values('&airplane_no','&departure_airport','&departure_time','&arrival_airport','&arrival_time')
new   1: insert into flight_hours values('AA751','DFW','12:00:00AM','LAX','12:00:00PM')

1 row created.

SQL> /
Enter value for airplane_no: AA851
Enter value for departure_airport: CLT
Enter value for departure_time: 12:00:00AM
Enter value for arrival_airport: LAX
Enter value for arrival_time: 3:00:00PM
old   1: insert into flight_hours values('&airplane_no','&departure_airport','&departure_time','&arrival_airport','&arrival_time')
new   1: insert into flight_hours values('AA851','CLT','12:00:00AM','LAX','3:00:00PM')

1 row created.

SQL> /
Enter value for airplane_no: IN497
Enter value for departure_airport: RGIA
Enter value for departure_time: 5:00:00Pm
Enter value for arrival_airport: LAX
Enter value for arrival_time: 5:00:00AM
old   1: insert into flight_hours values('&airplane_no','&departure_airport','&departure_time','&arrival_airport','&arrival_time')
new   1: insert into flight_hours values('IN497','RGIA','5:00:00Pm','LAX','5:00:00AM')

1 row created.
```



```
SQL> select * from flight_hours;

AIRPLANE_N DEPARTURE_AIRPORT              DEPARTURE_
---------- ------------------------------ ----------
ARRIVAL_AIRPORT                ARRIVAL_TI
------------------------------ ----------
AA751      DFW                            12:00:00AM
LAX                            12:00:00PM

AA851      CLT                            12:00:00AM
LAX                            3:00:00PM

IN497      RGIA                           5:00:00Pm
LAX                            5:00:00AM


SQL>
```

IMPLEMENTATION

Front end programs and its connectivity

Java Database Connectivity (**JDBC**) is an application programming interface (API) for

the programming language Java, which defines how a client may access a database. It is a

Java-based data access technology used for Java database connectivity. It is part of the

Java Standard Edition platform, from Oracle Corporation. It provides methods to query

and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
public void connectToDB()

{

       try

      {
        connection=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","ashwini","vasavi");

                          statement = connection.createStatement();


                    }

                    catch (SQLException connectException)

                    {

                      System.out.println(connectException.getMessage());

                      System.out.println(connectException.getSQLState());

                      System.out.println(connectException.getErrorCode());

                      System.exit(1);

                    }

            }
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used

for updating tables in the database directly.

Code:

```
package DBMS;

import java.awt.*;

import java.awt.event.*;

import java.sql.*;

//1602-19-737-068 M Ashwini

@SuppressWarnings("serial")

public class CreateTables extends Frame implements ActionListener

{

        MenuBar mb;

        MenuItem
m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17,m18,m19,m20;


        Menu employees,passengers,airport,flight_hours,payments;

        Button insertButton, submit;


        TextField
e_idText,e_nameText,e_ageText,e_phoneText,e_roleText,e_salaryText;//employees

        TextField
p_idText,p_f_nameText,p_l_nameText,p_countryText,p_phoneText;//passengers
```

```
        TextField
a_idText,a_nameText,a_countryText,a_stateText,a_cityText,a_zipText;//airport

        TextField
airplane_noText,departure_airportText,departure_timeText,arrival_airportText,arrival_timeT
ext;//flight_hours

        TextField pay_idText,amountText,discountText,taxText,total_amountText;//payments


        TextArea errorText;

        Connection connection;

        Statement statement;




        //For updates

        Button modify;

        List employeesList,passengersList,airportList,flight_hoursList,paymentsList;


        ResultSet rs;




                //For delete

                Button deleteRowButton;


                public CreateTables()

                {

                        try

                        {

                                Class.forName ("oracle.jdbc.driver.OracleDriver");

                        }
                        catch (Exception e)

                        {

                                System.err.println("Unable to find and load driver");

                                System.exit(1);

                        }

                        connectToDB ();

                }


                public void connectToDB()

            {

                        try

                        {

connection=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","ashwini",
"vasavi");

                  statement = connection.createStatement();


                        }

                        catch (SQLException connectException)

                        {

                  System.out.println(connectException.getMessage());

                  System.out.println(connectException.getSQLState());

                  System.out.println(connectException.getErrorCode());

                  System.exit(1);

                        }

            }

                public void buildFrame()

                {
```

```
        //Basic Frame Properties

        setTitle("Airlines Management System");

        setSize(500, 600);

        setVisible(true);


        //menubar

        mb = new MenuBar();

        setMenuBar(mb);

setSize(550,500);

setLayout(null);

setVisible(true);


//Employees

employees=new Menu("Employees");


 m1=new MenuItem("Insert Employees");

 m2=new MenuItem("Update Employees");

 m3=new MenuItem("Delete Employees");

 m4=new MenuItem("View Employees");


employees.add(m1);

employees.add(m2);

employees.add(m3);

employees.add(m4);


mb.add(employees);
```

```
//Passengers
passengers=new Menu("Passengers");
m5=new MenuItem("Insert Passengers");
m6=new MenuItem("Update Passengers");
m7=new MenuItem("Delete Passengers");
m8=new MenuItem("View Passengers");

passengers.add(m5);
passengers.add(m6);
passengers.add(m7);
passengers.add(m8);

mb.add(passengers);

//Airport
airport=new Menu("Airport");
m9=new MenuItem("Insert Airport");
m10=new MenuItem("Update Airport");
m11=new MenuItem("Delete Airport");
m12=new MenuItem("View Airport");

airport.add(m9);
airport.add(m10);
airport.add(m11);
airport.add(m12);

mb.add(airport);
```

```
//Flight Hours

flight_hours=new Menu("Flight_Hours");

m13=new MenuItem("Insert Flight_Hours");

m14=new MenuItem("Update Flight_Hours");

m15=new MenuItem("Delete Flight_Hours");

m16=new MenuItem("View Flight_Hours");


flight_hours.add(m13);

flight_hours.add(m14);

flight_hours.add(m15);

flight_hours.add(m16);


mb.add(flight_hours);


//Payments

payments=new Menu("Payments");


m17=new MenuItem("Insert Payments");

m18=new MenuItem("Update Payments");

m19=new MenuItem("Delete Payments");

m20=new MenuItem("View Payments");


payments.add(m17);

payments.add(m18);

payments.add(m19);

payments.add(m20);
```

```
  mb.add(payments);

//Registering action Listeners
  m1.addActionListener(this);
  m2.addActionListener(this);
  m3.addActionListener(this);
  m4.addActionListener(this);
  m5.addActionListener(this);
  m6.addActionListener(this);
  m7.addActionListener(this);
  m8.addActionListener(this);
  m9.addActionListener(this);
  m10.addActionListener(this);
  m11.addActionListener(this);
  m12.addActionListener(this);
  m13.addActionListener(this);
  m14.addActionListener(this);
  m15.addActionListener(this);
  m16.addActionListener(this);
  m17.addActionListener(this);
  m18.addActionListener(this);
  m19.addActionListener(this);
  m20.addActionListener(this);

      }
      public void actionPerformed(ActionEvent ae)
```

```
{
        String arg = ae.getActionCommand();
        if(arg.equals("Insert Employees"))
                this.buildGUIEmployees();
        if(arg.equals("Update Employees"))
                this.updateEmployeesGUI();
        if(arg.equals("Delete Employees"))
                this.deleteGUIEmployees();
        if(arg.equals("View Employees"))
                this.viewEmployeesGUI();
        if(arg.equals("Insert Passengers"))
                this.buildGUIPassengers();
        if(arg.equals("Update Passengers"))
                this.updatePassengersGUI();
        if(arg.equals("Delete Passengers"))
                this.deleteGUIPassengers();
        if(arg.equals("View Passengers"))
                this.viewPassengersGUI();
        if(arg.equals("Insert Airport"))
                this.buildGUIAirport();
        if(arg.equals("Update Airport"))
                this.updateAirportGUI();
        if(arg.equals("Delete Airport"))
                this.deleteGUIAirport();
        if(arg.equals("View Airport"))
                this.viewAirportGUI();
        if(arg.equals("Insert Flight_Hours"))
```

```
                this.buildGUIFlight_Hours();
        if(arg.equals("Update Flight_Hours"))
                this.updateFlight_HoursGUI();
        if(arg.equals("Delete Flight_Hours"))
                this.deleteGUIFlight_Hours();
        if(arg.equals("View Flight_Hours"))
                this.viewFlight_HoursGUI();
        if(arg.equals("Insert Payments"))
                this.buildGUIPayments();
        if(arg.equals("Update Payments"))
                this.updatePaymentsGUI();
        if(arg.equals("Delete Payments"))
                this.deleteGUIPayments();
        if(arg.equals("View Payments"))
                this.viewPaymentsGUI();
}
public void buildGUIEmployees()
{
        removeAll();
        //Handle Insert Account Button
        insertButton = new Button("Submit");
        insertButton.addActionListener(new ActionListener()
        {
                public void actionPerformed(ActionEvent e)
                {
                        try
                        {
```

```
                                        String query= "INSERT INTO employees
VALUES('" + e_idText.getText() + "', '" + e_nameText.getText() + "','" + e_ageText.getText()
+ "','" + e_phoneText.getText() + "','" + e_roleText.getText()+ "','" +e_salaryText.getText()+
"')";
                                          int i = statement.executeUpdate(query);
                                          errorText.append("\nInserted " + i + " rows
successfully");
                                        }
                                        catch (SQLException insertException)
                                        {
                                          displaySQLErrors(insertException);
                                        }
                                }
                        });
                e_idText = new TextField(15);
                e_nameText = new TextField(15);
                e_ageText = new TextField(15);
                e_phoneText= new TextField(15);
                e_roleText= new TextField(15);
                e_salaryText= new TextField(15);


                errorText = new TextArea(10, 40);
                errorText.setEditable(false);

                Panel first = new Panel();
                first.setLayout(new GridLayout(4, 2));
                first.add(new Label("Employee ID:"));
first.add(e_idText);

first.add(new Label("Employee Name:"));

first.add(e_nameText);

first.add(new Label("Employee Age:"));

first.add(e_ageText);

first.add(new Label("Employee Phone Number:"));

first.add(e_phoneText);

first.add(new Label("Employee Role:"));

first.add(e_roleText);

first.add(new Label("Employee Salary:"));

first.add(e_salaryText);

first.setBounds(125,90,200,100);


Panel second = new Panel(new GridLayout(4, 1));

second.add(insertButton);

second.setBounds(125,220,150,100);


Panel third = new Panel();

third.add(errorText);

third.setBounds(125,320,300,200);


//setLayout(null);


add(first);

add(second);

add(third);
```

```
                setLayout(new FlowLayout());
                setVisible(true);


        }
        public void buildGUIPassengers()
        {
                removeAll();
                //Handle Insert Account Button
                insertButton = new Button("Submit");
                insertButton.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {
                                try
                                {
                                        String query= "INSERT INTO passengers
VALUES('" + p_idText.getText() + "', '" + p_f_nameText.getText() + "','" +
p_l_nameText.getText() + "','" + p_countryText.getText() + "','" +
p_phoneText.getText()+"')";
                                         int i = statement.executeUpdate(query);
                                         errorText.append("\nInserted " + i + " rows
successfully");
                                        }
                                        catch (SQLException insertException)
                                        {
                                         displaySQLErrors(insertException);
                                        }
                                }
        });
p_idText = new TextField(15);
p_f_nameText = new TextField(15);
p_l_nameText = new TextField(15);
p_countryText= new TextField(15);
p_phoneText= new TextField(15);



errorText = new TextArea(10, 40);
errorText.setEditable(false);

Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Passenger ID:"));
first.add(p_idText);
first.add(new Label("Passenger First Name:"));
first.add(p_f_nameText);
first.add(new Label("Passenger Last Name:"));
first.add(p_l_nameText);
first.add(new Label("Passenger Country:"));
first.add(p_countryText);
first.add(new Label("Passenger Phone Number:"));
first.add(p_phoneText);

first.setBounds(125,90,200,100);

Panel second = new Panel(new GridLayout(4, 1));
```

```
        second.add(insertButton);
        second.setBounds(125,220,150,100);

        Panel third = new Panel();
        third.add(errorText);
        third.setBounds(125,320,300,200);

        //setLayout(null);

        add(first);
        add(second);
        add(third);

        setLayout(new FlowLayout());
        setVisible(true);

    }
    public void buildGUIAirport()
    {
        removeAll();
        //Handle Insert Account Button
        insertButton = new Button("Submit");
        insertButton.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                try
```

```
                                        {
                                              String query= "INSERT INTO airport
VALUES('" + a_idText.getText() + "', '" + a_nameText.getText() + "','" +
a_countryText.getText() + "','" + a_stateText.getText() + "','" + a_cityText.getText()+"',
'"+a_zipText.getText()+"')";
                                                int i = statement.executeUpdate(query);
                                                errorText.append("\nInserted " + i + " rows
successfully");
                                              }
                                              catch (SQLException insertException)
                                              {
                                                displaySQLErrors(insertException);
                                              }
                                        }
                                  });
                            a_idText = new TextField(15);
                            a_nameText = new TextField(15);
                            a_countryText = new TextField(15);
                            a_stateText= new TextField(15);
                            a_cityText= new TextField(15);
                            a_zipText= new TextField(15);



                            errorText = new TextArea(10, 40);
                            errorText.setEditable(false);

                            Panel first = new Panel();
                            first.setLayout(new GridLayout(4, 2));
first.add(new Label("Airport ID:"));

first.add(a_idText);

first.add(new Label("Airport Name:"));

first.add(a_nameText);

first.add(new Label("Airport Country:"));

first.add(a_countryText);

first.add(new Label("Airport State:"));

first.add(a_stateText);

first.add(new Label("Airport City:"));

first.add(a_cityText);

first.add(new Label("Airport Zip:"));

first.add(a_zipText);


first.setBounds(125,90,200,100);


Panel second = new Panel(new GridLayout(4, 1));

second.add(insertButton);

second.setBounds(125,220,150,100);


Panel third = new Panel();

third.add(errorText);

third.setBounds(125,320,300,200);


//setLayout(null);


add(first);

add(second);
```

```
		add(third);


		setLayout(new FlowLayout());
		setVisible(true);


	}


	public void buildGUIFlight_Hours()
	{
		removeAll();
		//Handle Insert Account Button
		insertButton = new Button("Submit");
		insertButton.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent e)
			{
				try
				{
					String query= "INSERT INTO flight_hours
VALUES('" + airplane_noText.getText() + "', '" + departure_airportText.getText() + "','" +
departure_timeText.getText() + "','" + arrival_airportText.getText() + "','" +
arrival_timeText.getText()+"')";
					  int i = statement.executeUpdate(query);
					  errorText.append("\nInserted " + i + " rows
successfully");
					}
					catch (SQLException insertException)
					{
```

```
                                displaySQLErrors(insertException);
                            }
                    }
            });
airplane_noText = new TextField(15);
departure_airportText = new TextField(15);
departure_timeText = new TextField(15);
arrival_airportText= new TextField(15);
arrival_timeText= new TextField(15);



errorText = new TextArea(10, 40);
errorText.setEditable(false);

Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Airplane No:"));
first.add(airplane_noText);
first.add(new Label("Departure Airport:"));
first.add(departure_airportText);
first.add(new Label("Departure Time:"));
first.add(departure_timeText);
first.add(new Label("Arrival Airport:"));
first.add(arrival_airportText);
first.add(new Label("Arrival Time:"));
first.add(arrival_timeText);
```

```
		first.setBounds(125,90,200,100);

		Panel second = new Panel(new GridLayout(4, 1));
		second.add(insertButton);
		second.setBounds(125,220,150,100);

		Panel third = new Panel();
		third.add(errorText);
		third.setBounds(125,320,300,200);

		//setLayout(null);

		add(first);
		add(second);
		add(third);

		setLayout(new FlowLayout());
		setVisible(true);

	}

	public void buildGUIPayments()
	{
		removeAll();
		//Handle Insert Account Button
		insertButton = new Button("Submit");
		insertButton.addActionListener(new ActionListener()
                    {
                        public void actionPerformed(ActionEvent e)
                        {
                            try
                            {
                                String query= "INSERT INTO payments
VALUES('" + pay_idText.getText() + "', '" + amountText.getText() + "','" +
discountText.getText() + "','" + taxText.getText() + "','" + total_amountText.getText()+ "')";
                                  int i = statement.executeUpdate(query);
                                  errorText.append("\nInserted " + i + " rows
successfully");
                                }
                                catch (SQLException insertException)
                                {
                                  displaySQLErrors(insertException);
                                }
                            }
                        });
                    pay_idText = new TextField(15);
                    amountText = new TextField(15);
                    discountText = new TextField(15);
                    taxText= new TextField(15);
                    total_amountText= new TextField(15);

                    errorText = new TextArea(10, 40);
                    errorText.setEditable(false);

                    Panel first = new Panel();
```

```
first.setLayout(new GridLayout(4, 2));

first.add(new Label("Payment ID:"));

first.add(pay_idText);

first.add(new Label("Payment Amount:"));

first.add(amountText);

first.add(new Label("Payment discount:"));

first.add(discountText);

first.add(new Label("Payment tax:"));

first.add(taxText);

first.add(new Label("Payment total amount:"));

first.add(total_amountText);

first.setBounds(125,90,200,100);


Panel second = new Panel(new GridLayout(4, 1));

second.add(insertButton);

second.setBounds(125,220,150,100);


Panel third = new Panel();

third.add(errorText);

third.setBounds(125,320,300,200);


//setLayout(null);


add(first);

add(second);

add(third);
```

```
        {

                setLayout(new FlowLayout());
                setVisible(true);

        }

        private void loadEmployees()
        {
                try
                {
                  rs = statement.executeQuery("SELECT * FROM employees");
                  while (rs.next())
                  {
                        employeesList.add(rs.getString("e_id"));
                  }
                }
                catch (SQLException e)
                {
                  displaySQLErrors(e);
                }
        }

        private void loadPassengers()
        {
                try
                {
                  rs = statement.executeQuery("SELECT * FROM passengers");
                  while (rs.next())
```

```
              {
                      passengersList.add(rs.getString("p_id"));
              }
            }
            catch (SQLException e)
            {
              displaySQLErrors(e);
            }
}

private void loadAirport()
{
            try
            {
              rs = statement.executeQuery("SELECT * FROM airport");
              while (rs.next())
              {
                      airportList.add(rs.getString("a_id"));
              }
            }
            catch (SQLException e)
            {
              displaySQLErrors(e);
            }
}

private void loadFlight_Hours()
```

```
{
        try
        {
          rs = statement.executeQuery("SELECT * FROM flight_hours");
          while (rs.next())
          {
                  flight_hoursList.add(rs.getString("airplane_no"));
          }
        }
        catch (SQLException e)
        {
          displaySQLErrors(e);
        }
}

private void loadPayments()
{
        try
        {
          rs = statement.executeQuery("SELECT * FROM payments");
          while (rs.next())
          {
                  paymentsList.add(rs.getString("pay_id"));
          }
        }
        catch (SQLException e)
        {
              displaySQLErrors(e);
            }
        }
        public void updateEmployeesGUI()
        {
            removeAll();
            employeesList = new List(6);
            loadEmployees();
            add(employeesList);

            //When a list item is selected populate the text fields
            employeesList.addItemListener(new ItemListener()
            {
                public void itemStateChanged(ItemEvent e)
                {
                    try
                    {
                        rs = statement.executeQuery("SELECT *
FROM employees");
                        while (rs.next())
                        {
                            if
(rs.getString("e_id").equals(employeesList.getSelectedItem()))
                            break;
                        }
                        if (!rs.isAfterLast())
                        {
                            e_idText.setText(rs.getString("e_id"));
        e_nameText.setText(rs.getString("e_name"));
                                                        e_ageText.setText(rs.getString("e_age"));

        e_phoneText.setText(rs.getString("e_phone"));

        e_roleText.setText(rs.getString("e_role"));

        e_salaryText.setText(rs.getString("e_salary"));
                                                }
                                        }
                                        catch (SQLException selectException)
                                        {
                                                displaySQLErrors(selectException);
                                        }
                                }
                        });
                        //Handle Update Employees Button
                        modify = new Button("Modify");
                        modify.addActionListener(new ActionListener()
                        {
                                public void actionPerformed(ActionEvent e)
                                {
                                        try
                                        {
                                                Statement statement =
connection.createStatement();
                                                int i = statement.executeUpdate("UPDATE
employees "
```

```
                                                        + "SET e_salary=" + e_salaryText.getText()
+",e_name='"+e_nameText.getText()+"',e_age="+e_ageText.getText()+",e_phone="+e_phon
eText.getText()+",e_role='"+e_roleText.getText()+"'"

                                                        + " WHERE e_id = '" +
employeesList.getSelectedItem() + "'");

                                                        errorText.append("\nUpdated " + i + " rows
successfully");

                                                        employeesList.removeAll();

                                                        loadEmployees();

                                              }

                                              catch (SQLException insertException)

                                              {

                                                        displaySQLErrors(insertException);

                                              }

                                    }

                          });

                          e_idText = new TextField(15);

                          //e_idText.setEditable(false);

                          e_nameText = new TextField(15);

                          //e_nameText.setEditable(false);

                          e_salaryText = new TextField(15);

                          e_ageText = new TextField(15);

                          //e_ageText.setEditable(false);

                          e_roleText = new TextField(15);

                          //e_roleText.setEditable(false);

                          e_phoneText = new TextField(15);

                          //e_phoneText.setEditable(false);
```

```
errorText = new TextArea(10, 40);

errorText.setEditable(false);


Panel first = new Panel();

first.setLayout(new GridLayout(4, 2));

first.add(new Label("Employee ID:"));

first.add(e_idText);

first.add(new Label("Employee Name:"));

first.add(e_nameText);

first.add(new Label("Employee Age"));

first.add(e_ageText);

first.add(new Label("Employee Phone Number:"));

first.add(e_phoneText);

first.add(new Label("Employee Role:"));

first.add(e_roleText);

first.add(new Label("Employee Salary:"));

first.add(e_salaryText);


Panel second = new Panel(new GridLayout(4, 1));

second.add(modify);


Panel third = new Panel();

third.add(errorText);


add(first);


add(second);
```

```
		add(third);

		//setTitle("Update ....");
		//setSize(500, 600);
		setLayout(new FlowLayout());
		setVisible(true);

	}

	public void updatePassengersGUI()
	{
		removeAll();
		passengersList = new List(6);
		loadPassengers();
		add(passengersList);

		//When a list item is selected populate the text fields
		passengersList.addItemListener(new ItemListener()
		{
			public void itemStateChanged(ItemEvent e)
			{
				try
				{
					rs = statement.executeQuery("SELECT *
FROM passengers");
					while (rs.next())
					{
```

```
                                                                        if
(rs.getString("p_id").equals(passengersList.getSelectedItem()))
                                                                        break;
                                                                }
                                                                if (!rs.isAfterLast())
                                                                {
                                                                        p_idText.setText(rs.getString("p_id"));

        p_f_nameText.setText(rs.getString("p_f_name"));

        p_l_nameText.setText(rs.getString("p_l_name"));

        p_countryText.setText(rs.getString("p_country"));

        p_phoneText.setText(rs.getString("p_phone"));
                                                                }
                                                        }
                                                        catch (SQLException selectException)
                                                        {
                                                                displaySQLErrors(selectException);
                                                        }
                                                }
                                        });
                                        //Handle Update Employees Button
                                        modify = new Button("Modify");
                                        modify.addActionListener(new ActionListener()
                                        {
                                                public void actionPerformed(ActionEvent e)
                                                {
```

```
                        try
                        {
                                Statement statement =
connection.createStatement();
                                int i = statement.executeUpdate("UPDATE
passengers "
                                + "SET p_f_name='" + p_f_nameText.getText()
+"',p_l_name='"+p_l_nameText.getText()+"',p_country='"+p_countryText.getText()+"',p_ph
one="+p_phoneText.getText()
                                + " WHERE p_id = '" +
passengersList.getSelectedItem() + "'");
                                errorText.append("\nUpdated " + i + " rows
successfully");
                                passengersList.removeAll();
                                loadPassengers();
                        }
                        catch (SQLException insertException)
                        {
                                displaySQLErrors(insertException);
                        }
                }
        });
        p_idText = new TextField(15);
        p_f_nameText = new TextField(15);
        p_l_nameText = new TextField(15);
        p_countryText = new TextField(15);
        p_phoneText = new TextField(15);

        errorText = new TextArea(10, 40);
```

```
errorText.setEditable(false);


Panel first = new Panel();

first.setLayout(new GridLayout(4, 2));

first.add(new Label("Passenger ID:"));

first.add(p_idText);

first.add(new Label("Passenger First Name:"));

first.add(p_f_nameText);

first.add(new Label("Passenger Last Name"));

first.add(p_l_nameText);

first.add(new Label("Passenger Country:"));

first.add(p_countryText);

first.add(new Label("Passenger Phone Number:"));

first.add(p_phoneText);


Panel second = new Panel(new GridLayout(4, 1));

second.add(modify);


Panel third = new Panel();

third.add(errorText);


add(first);


add(second);

add(third);


//setTitle("Update ....");
```

```
                //setSize(500, 600);

                setLayout(new FlowLayout());

                setVisible(true);

        }


        public void updateAirportGUI()

        {

                removeAll();

                airportList = new List(6);

                loadAirport();

                add(airportList);


                //When a list item is selected populate the text fields

                airportList.addItemListener(new ItemListener()

                {

                        public void itemStateChanged(ItemEvent e)

                        {

                                try

                                {

                                        rs = statement.executeQuery("SELECT *
FROM airport");

                                        while (rs.next())

                                        {

                                                if
(rs.getString("a_id").equals(airportList.getSelectedItem()))

                                                break;

                                        }

                                        if (!rs.isAfterLast())
                                        {
                                             a_idText.setText(rs.getString("a_id"));

          a_nameText.setText(rs.getString("a_name"));

          a_countryText.setText(rs.getString("a_country"));

          a_stateText.setText(rs.getString("a_state"));

          a_cityText.setText(rs.getString("a_city"));
                                             a_zipText.setText(rs.getString("a_zip"));
                                        }
                                   }
                                   catch (SQLException selectException)
                                   {
                                        displaySQLErrors(selectException);
                                   }
                              }
                         });
                         //Handle Update Employees Button
                         modify = new Button("Modify");
                         modify.addActionListener(new ActionListener()
                         {
                              public void actionPerformed(ActionEvent e)
                              {
                                   try
                                   {
                                        Statement statement =
     connection.createStatement();
```

```
                        int i = statement.executeUpdate("UPDATE
airport "
                        + "SET a_country='" + a_countryText.getText()
+"',a_name='"+a_nameText.getText()+"',a_state='"+a_stateText.getText()+"',a_city='"+a_cit
yText.getText()+"',a_zip='"+a_zipText.getText()+"'"
                        + " WHERE a_id = '" +
airportList.getSelectedItem() + "'");
                        errorText.append("\nUpdated " + i + " rows
successfully");
                        airportList.removeAll();
                        loadAirport();
                    }
                    catch (SQLException insertException)
                    {
                        displaySQLErrors(insertException);
                    }
                }
            });
            a_idText = new TextField(15);
            a_nameText = new TextField(15);
            a_countryText = new TextField(15);
            a_stateText = new TextField(15);
            a_cityText = new TextField(15);
            a_zipText = new TextField(15);

            errorText = new TextArea(10, 40);
            errorText.setEditable(false);

            Panel first = new Panel();
```

```
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Airport ID:"));
first.add(a_idText);
first.add(new Label("Airport Name:"));
first.add(a_nameText);
first.add(new Label("Airport Country"));
first.add(a_countryText);
first.add(new Label("Airport State:"));
first.add(a_stateText);
first.add(new Label("Airport City:"));
first.add(a_cityText);
first.add(new Label("Airport zip code:"));
first.add(a_zipText);

Panel second = new Panel(new GridLayout(4, 1));
second.add(modify);

Panel third = new Panel();
third.add(errorText);

add(first);


add(second);
add(third);

//setTitle("Update ....");
//setSize(500, 600);
```

```
                setLayout(new FlowLayout());
                setVisible(true);


        }


        public void updateFlight_HoursGUI()
        {
                removeAll();
                flight_hoursList = new List(6);
                loadFlight_Hours();
                add(flight_hoursList);


                //When a list item is selected populate the text fields
                flight_hoursList.addItemListener(new ItemListener()
                {
                        public void itemStateChanged(ItemEvent e)
                        {
                                try
                                {
                                        rs = statement.executeQuery("SELECT *
FROM flight_hours");
                                        while (rs.next())
                                        {
                                                if
(rs.getString("airplane_no").equals(flight_hoursList.getSelectedItem()))
                                                break;
                                        }
                                        if (!rs.isAfterLast())
                                        {

    airplane_noText.setText(rs.getString("airplane_no"));

    departure_airportText.setText(rs.getString("departure_airport"));

    departure_timeText.setText(rs.getString("departure_time"));

    arrival_airportText.setText(rs.getString("arrival_airport"));

    arrival_timeText.setText(rs.getString("arrival_time"));
                                        }
                                    }
                                    catch (SQLException selectException)
                                    {
                                        displaySQLErrors(selectException);
                                    }
                                }
                            });
                            //Handle Update Employees Button
                            modify = new Button("Modify");
                            modify.addActionListener(new ActionListener()
                            {
                                public void actionPerformed(ActionEvent e)
                                {
                                    try
                                    {
                                        Statement statement =
connection.createStatement();
```

```
                                                    int i = statement.executeUpdate("UPDATE
flight_hours "
                                                    + "SET departure_airport='" +
departure_airportText.getText()
+"',departure_time='"+departure_timeText.getText()+"',arrival_airport='"+arrival_airportText
.getText()+"',arrival_time='"+arrival_timeText.getText()+"'"
                                                    + " WHERE airplane_no = '" +
flight_hoursList.getSelectedItem() + "'");
                                                    errorText.append("\nUpdated " + i + " rows
successfully");
                                                    flight_hoursList.removeAll();
                                                    loadFlight_Hours();
                                            }
                                            catch (SQLException insertException)
                                            {
                                                    displaySQLErrors(insertException);
                                            }
                                    }
                            });
                            airplane_noText = new TextField(15);
                            departure_airportText = new TextField(15);
                            departure_timeText = new TextField(15);
                            arrival_airportText= new TextField(15);
                            arrival_timeText= new TextField(15);

                            errorText = new TextArea(10, 40);
                            errorText.setEditable(false);

                            Panel first = new Panel();
```

```
first.setLayout(new GridLayout(4, 2));

first.add(new Label("Airplane No:"));

first.add(airplane_noText);

first.add(new Label("Departure Airport:"));

first.add(departure_airportText);

first.add(new Label("Departure Time:"));

first.add(departure_timeText);

first.add(new Label("Arrival Airport:"));

first.add(arrival_airportText);

first.add(new Label("Arrival Time:"));

first.add(arrival_timeText);

first.setBounds(125,90,200,100);


Panel second = new Panel(new GridLayout(4, 1));

second.add(modify);


Panel third = new Panel();

third.add(errorText);


add(first);


add(second);

add(third);


//setTitle("Update ....");

//setSize(500, 600);

setLayout(new FlowLayout());
```

```
				setVisible(true);
			}

			public void updatePaymentsGUI()
			{
				removeAll();
				paymentsList = new List(6);
				loadPayments();
				add(paymentsList);

				//When a list item is selected populate the text fields
				paymentsList.addItemListener(new ItemListener()
				{
					public void itemStateChanged(ItemEvent e)
					{
						try
						{
							rs = statement.executeQuery("SELECT *
FROM payments");
							while (rs.next())
							{
								if
(rs.getString("pay_id").equals(paymentsList.getSelectedItem()))
								break;
							}
							if (!rs.isAfterLast())
							{
          pay_idText.setText(rs.getString("pay_id"));

          amountText.setText(rs.getString("amount"));

          discountText.setText(rs.getString("discount"));
                                                            taxText.setText(rs.getString("tax"));

          total_amountText.setText(rs.getString("total_amount"));
                                                  }
                                        }
                                        catch (SQLException selectException)
                                        {
                                                  displaySQLErrors(selectException);
                                        }
                              }
                    });
                    //Handle Update Employees Button
                    modify = new Button("Modify");
                    modify.addActionListener(new ActionListener()
                    {
                              public void actionPerformed(ActionEvent e)
                              {
                                        try
                                        {
                                                  Statement statement =
connection.createStatement();
                                                  int i = statement.executeUpdate("UPDATE
payments "
```

```
                                        + "SET amount=" + amountText.getText()
+",discount="+discountText.getText()+",tax="+taxText.getText()+",total_amount="+total_a
mountText.getText()+""

                                        + " WHERE pay_id = '" +
paymentsList.getSelectedItem() + "'");

                                        errorText.append("\nUpdated " + i + " rows
successfully");

                                        paymentsList.removeAll();

                                        loadPayments();

                                   }

                                   catch (SQLException insertException)

                                   {

                                        displaySQLErrors(insertException);

                                   }

                              }

                         });

                         pay_idText = new TextField(15);

                         amountText = new TextField(15);

                         discountText = new TextField(15);

                         taxText= new TextField(15);

                         total_amountText= new TextField(15);


                         errorText = new TextArea(10, 40);

                         errorText.setEditable(false);


                         Panel first = new Panel();

                         first.setLayout(new GridLayout(4, 2));

                         first.add(new Label("Payment ID:"));
first.add(pay_idText);

first.add(new Label("Payment Amount:"));

first.add(amountText);

first.add(new Label("Payment discount:"));

first.add(discountText);

first.add(new Label("Payment tax:"));

first.add(taxText);

first.add(new Label("Payment total amount:"));

first.add(total_amountText);

first.setBounds(125,90,200,100);


Panel second = new Panel(new GridLayout(4, 1));

second.add(modify);


Panel third = new Panel();

third.add(errorText);


add(first);


add(second);

add(third);


//setTitle("Update ....");

//setSize(500, 600);

setLayout(new FlowLayout());

setVisible(true);
```

```
                    }

                    public void deleteGUIEmployees()
                    {
                              removeAll();
                       employeesList = new List(10);
                              loadEmployees();
                              add(employeesList);

                              //When a list item is selected populate the text fields

                              employeesList.addItemListener(new ItemListener()
                              {
                                        public void itemStateChanged(ItemEvent e)
                                        {
                                                  try
                                                  {
                                                            rs = statement.executeQuery("SELECT *
FROM employees");
                                                            while (rs.next())
                                                            {
                                                                      if
(rs.getString("e_id").equals(employeesList.getSelectedItem()))
                                                                      break;
                                                            }
                                                            if (!rs.isAfterLast())
                                                            {
                                                                      e_idText.setText(rs.getString("e_id"));
```

```
	e_nameText.setText(rs.getString("e_name"));
						e_ageText.setText(rs.getString("e_age"));

	e_phoneText.setText(rs.getString("e_phone"));

	e_roleText.setText(rs.getString("e_role"));

	e_salaryText.setText(rs.getString("e_salary"));
					}
				}
				catch (SQLException selectException)
				{
					displaySQLErrors(selectException);
				}
			}
		});

		//Handle Delete employees Button
		deleteRowButton = new Button("Delete Row");
		deleteRowButton.addActionListener(new ActionListener()
		{
			public void actionPerformed(ActionEvent e)
			{
				try
				{
					Statement statement =
connection.createStatement();
```

```
                                        int i = statement.executeUpdate("DELETE
FROM employees WHERE e_id = '" + employeesList.getSelectedItem()+"'");
                                        errorText.append("\nDeleted " + i + " rows
successfully");
                                        e_idText.setText(null);
                                        e_nameText.setText(null);
                                        e_ageText.setText(null);
                                        e_phoneText.setText(null);
                                        e_roleText.setText(null);
                                        e_salaryText.setText(null);
                                        employeesList.removeAll();
                                        loadEmployees();
                                }
                                catch (SQLException deleteException)
                                {
                                        displaySQLErrors(deleteException);
                                }
                        }
                });

                e_idText = new TextField(15);
                e_nameText = new TextField(15);
                e_ageText = new TextField(15);
                e_phoneText = new TextField(15);
                e_roleText = new TextField(15);
                e_salaryText = new TextField(15);

                errorText = new TextArea(10, 40);
```

```
errorText.setEditable(false);


e_idText.setEditable(false);

e_nameText.setEditable(false);

e_ageText.setEditable(false);

e_phoneText.setEditable(false);

e_roleText.setEditable(false);

e_salaryText.setEditable(false);



Panel first = new Panel();

first.setLayout(new GridLayout(4, 2));

first.add(new Label("Employee ID:"));

first.add(e_idText);

first.add(new Label("Employee Name:"));

first.add(e_nameText);

first.add(new Label("Employee Age:"));

first.add(e_ageText);

first.add(new Label("Employee Phone Number:"));

first.add(e_phoneText);

first.add(new Label("Employee Role:"));

first.add(e_roleText);

first.add(new Label("Employee Salary:"));

first.add(e_salaryText);


Panel second = new Panel(new GridLayout(4, 1));

second.add(deleteRowButton);
```

```
        Panel third = new Panel();
        third.add(errorText);

        add(first);
        add(second);
        add(third);


        setLayout(new FlowLayout());
        setVisible(true);

}

public void deleteGUIPassengers()
{
        removeAll();
   passengersList = new List(10);
        loadPassengers();
        add(passengersList);

        //When a list item is selected populate the text fields

        passengersList.addItemListener(new ItemListener()
        {
                public void itemStateChanged(ItemEvent e)
                {
```

```
                                        try
                                        {
                                                rs = statement.executeQuery("SELECT *
FROM passengers");
                                                while (rs.next())
                                                {
                                                        if
(rs.getString("p_id").equals(passengersList.getSelectedItem()))
                                                        break;
                                                }
                                                if (!rs.isAfterLast())
                                                {
                                                        p_idText.setText(rs.getString("p_id"));

        p_f_nameText.setText(rs.getString("p_f_name"));

        p_l_nameText.setText(rs.getString("p_l_name"));

        p_countryText.setText(rs.getString("p_country"));

        p_phoneText.setText(rs.getString("p_phone"));
                                                }
                                        }
                                        catch (SQLException selectException)
                                        {
                                                displaySQLErrors(selectException);
                                        }
                                }
                        });
```

```
                    //Handle Delete employees Button
                    deleteRowButton = new Button("Delete Row");
                    deleteRowButton.addActionListener(new ActionListener()
                    {
                            public void actionPerformed(ActionEvent e)
                            {
                                    try
                                    {
                                            Statement statement =
connection.createStatement();
                                            int i = statement.executeUpdate("DELETE
FROM passengers WHERE p_id = '" + passengersList.getSelectedItem()+"'");
                                            errorText.append("\nDeleted " + i + " rows
successfully");
                                            p_idText.setText(null);
                                            p_f_nameText.setText(null);
                                            p_l_nameText.setText(null);
                                            p_phoneText.setText(null);
                                            p_countryText.setText(null);
                                            passengersList.removeAll();
                                            loadPassengers();
                                    }
                                    catch (SQLException deleteException)
                                    {
                                            displaySQLErrors(deleteException);
                                    }
                            }
                    });

                    p_idText = new TextField(15);

                    p_f_nameText = new TextField(15);

                    p_l_nameText = new TextField(15);

                    p_countryText = new TextField(15);

                    p_phoneText = new TextField(15);


                    errorText = new TextArea(10, 40);

                    errorText.setEditable(false);


                    p_idText.setEditable(false);

                    p_f_nameText.setEditable(false);

                    p_l_nameText.setEditable(false);

                    p_phoneText.setEditable(false);

                    p_countryText.setEditable(false);


                    Panel first = new Panel();

                    first.setLayout(new GridLayout(4, 2));

                    first.add(new Label("Passenger ID:"));

                    first.add(p_idText);

                    first.add(new Label("Passenger First Name:"));

                    first.add(p_f_nameText);

                    first.add(new Label("Passenger Last Name"));

                    first.add(p_l_nameText);

                    first.add(new Label("Passenger Country:"));

                    first.add(p_countryText);

                    first.add(new Label("Passenger Phone Number:"));
```

```
        first.add(p_phoneText);


        Panel second = new Panel(new GridLayout(4, 1));

        second.add(deleteRowButton);


        Panel third = new Panel();

        third.add(errorText);


        add(first);

        add(second);

        add(third);




        setLayout(new FlowLayout());

        setVisible(true);


}

public void deleteGUIAirport()

{

        removeAll();

   airportList = new List(10);

        loadAirport();

        add(airportList);


        //When a list item is selected populate the text fields


        airportList.addItemListener(new ItemListener()
                    {
                         public void itemStateChanged(ItemEvent e)
                         {
                              try
                              {
                                   rs = statement.executeQuery("SELECT *
FROM airport");
                                   while (rs.next())
                                   {
                                        if
(rs.getString("a_id").equals(airportList.getSelectedItem()))
                                        break;
                                   }
                                   if (!rs.isAfterLast())
                                   {
                                        a_idText.setText(rs.getString("a_id"));

     a_nameText.setText(rs.getString("a_name"));

     a_countryText.setText(rs.getString("a_country"));

     a_stateText.setText(rs.getString("a_state"));

     a_cityText.setText(rs.getString("a_city"));
                                        a_zipText.setText(rs.getString("a_zip"));


                                   }
                              }
                              catch (SQLException selectException)
                    {
                        displaySQLErrors(selectException);
                    }
                }
            });

            //Handle Delete employees Button
            deleteRowButton = new Button("Delete Row");
            deleteRowButton.addActionListener(new ActionListener()
            {
                public void actionPerformed(ActionEvent e)
                {
                    try
                    {
                        Statement statement =
connection.createStatement();
                        int i = statement.executeUpdate("DELETE
FROM airport WHERE a_id = '" + airportList.getSelectedItem()+"'");
                        errorText.append("\nDeleted " + i + " rows
successfully");
                        a_idText.setText(null);
                        a_nameText.setText(null);
                        a_countryText.setText(null);
                        a_stateText.setText(null);
                        a_cityText.setText(null);
                        a_zipText.setText(null);
                        airportList.removeAll();
                        loadAirport();
                }
                catch (SQLException deleteException)
                {
                        displaySQLErrors(deleteException);
                }
        }
});

a_idText = new TextField(15);
a_nameText = new TextField(15);
a_countryText = new TextField(15);
a_stateText = new TextField(15);
a_cityText = new TextField(15);
a_zipText = new TextField(15);

errorText = new TextArea(10, 40);
errorText.setEditable(false);

a_idText.setEditable(false);
a_nameText.setEditable(false);
a_countryText.setEditable(false);
a_stateText.setEditable(false);
a_cityText.setEditable(false);
a_zipText.setEditable(false);

Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
```

```
        first.add(new Label("Airport ID:"));

        first.add(a_idText);

        first.add(new Label("Airport Name:"));

        first.add(a_nameText);

        first.add(new Label("Airport Country"));

        first.add(a_countryText);

        first.add(new Label("Airport State:"));

        first.add(a_stateText);

        first.add(new Label("Airport City:"));

        first.add(a_cityText);

        first.add(new Label("Airport zip code:"));

        first.add(a_zipText);


        Panel second = new Panel(new GridLayout(4, 1));

        second.add(deleteRowButton);


        Panel third = new Panel();

        third.add(errorText);


        add(first);

        add(second);

        add(third);



        setLayout(new FlowLayout());

        setVisible(true);

}
```

```
                public void deleteGUIFlight_Hours()
                {
                        removeAll();
                   flight_hoursList = new List(10);
                        loadFlight_Hours();
                        add(flight_hoursList);

                        //When a list item is selected populate the text fields

                        flight_hoursList.addItemListener(new ItemListener()
                        {
                                public void itemStateChanged(ItemEvent e)
                                {
                                        try
                                        {
                                                rs = statement.executeQuery("SELECT *
FROM flight_hours");
                                                while (rs.next())
                                                {
                                                        if
(rs.getString("airplane_no").equals(flight_hoursList.getSelectedItem()))
                                                        break;
                                                }
                                                if (!rs.isAfterLast())
                                                {

        airplane_noText.setText(rs.getString("airplane_no"));
departure_airportText.setText(rs.getString("departure_airport"));

departure_timeText.setText(rs.getString("departure_time"));

arrival_airportText.setText(rs.getString("arrival_airport"));

arrival_timeText.setText(rs.getString("arrival_time"));
                                        }
                              }
                              catch (SQLException selectException)
                              {
                                        displaySQLErrors(selectException);
                              }
                    }
          });

          //Handle Delete employees Button
          deleteRowButton = new Button("Delete Row");
          deleteRowButton.addActionListener(new ActionListener()
          {
                    public void actionPerformed(ActionEvent e)
                    {
                              try
                              {
                                        Statement statement =
connection.createStatement();
                                        int i = statement.executeUpdate("DELETE
FROM flight_hours WHERE airplane_no = '" + flight_hoursList.getSelectedItem()+"'");
```

```
                                        errorText.append("\nDeleted " + i + " rows
successfully");

                                        airplane_noText.setText(null);

                                        departure_airportText.setText(null);

                                        departure_timeText.setText(null);

                                        arrival_airportText.setText(null);

                                        arrival_timeText.setText(null);

                                        flight_hoursList.removeAll();

                                        loadFlight_Hours();

                                   }

                                   catch (SQLException deleteException)

                                   {

                                        displaySQLErrors(deleteException);

                                   }

                         }

                    });


                    airplane_noText = new TextField(15);

                    departure_airportText = new TextField(15);

                    departure_timeText = new TextField(15);

                    arrival_airportText = new TextField(15);

                    arrival_timeText = new TextField(15);


                    errorText = new TextArea(10, 40);

                    errorText.setEditable(false);


                    airplane_noText.setEditable(false);
```

```
departure_airportText.setEditable(false);

departure_timeText.setEditable(false);

arrival_airportText.setEditable(false);

arrival_timeText.setEditable(false);


Panel first = new Panel();

first.setLayout(new GridLayout(4, 2));

first.add(new Label("Airplane No:"));

first.add(airplane_noText);

first.add(new Label("Departure Airport:"));

first.add(departure_airportText);

first.add(new Label("Departure Time:"));

first.add(departure_timeText);

first.add(new Label("Arrival Airport:"));

first.add(arrival_airportText);

first.add(new Label("Arrival Time:"));

first.add(arrival_timeText);



Panel second = new Panel(new GridLayout(4, 1));

second.add(deleteRowButton);


Panel third = new Panel();

third.add(errorText);


add(first);

add(second);
```

```
		add(third);



		setLayout(new FlowLayout());
		setVisible(true);
	}

	public void deleteGUIPayments()
	{
		removeAll();
	   paymentsList = new List(10);
		loadPayments();
		add(paymentsList);

		//When a list item is selected populate the text fields

		paymentsList.addItemListener(new ItemListener()
		{
			public void itemStateChanged(ItemEvent e)
			{
				try
				{
					rs = statement.executeQuery("SELECT *
FROM payments");
					while (rs.next())
					{
						if
(rs.getString("pay_id").equals(paymentsList.getSelectedItem()))
```

```
                                        break;
                                    }
                                    if (!rs.isAfterLast())
                                    {

pay_idText.setText(rs.getString("pay_id"));

amountText.setText(rs.getString("amount"));

discountText.setText(rs.getString("discount"));
                                        taxText.setText(rs.getString("tax"));

total_amountText.setText(rs.getString("total_amount"));
                                    }
                                }
                                catch (SQLException selectException)
                                {
                                    displaySQLErrors(selectException);
                                }
                            }
                        });

                        //Handle Delete employees Button
                        deleteRowButton = new Button("Delete Row");
                        deleteRowButton.addActionListener(new ActionListener()
                        {
                            public void actionPerformed(ActionEvent e)
                            {
```

```
                try
                {
                        Statement statement =
connection.createStatement();
                        int i = statement.executeUpdate("DELETE
FROM payments WHERE pay_id = '" + paymentsList.getSelectedItem()+"'");
                        errorText.append("\nDeleted " + i + " rows
successfully");
                        pay_idText.setText(null);
                        amountText.setText(null);
                        discountText.setText(null);
                        taxText.setText(null);
                        total_amountText.setText(null);
                        paymentsList.removeAll();
                        loadPayments();
                }
                catch (SQLException deleteException)
                {
                        displaySQLErrors(deleteException);
                }
        }
});

pay_idText = new TextField(15);
amountText = new TextField(15);
discountText = new TextField(15);
taxText = new TextField(15);
total_amountText = new TextField(15);
```

```
errorText = new TextArea(10, 40);
errorText.setEditable(false);

pay_idText.setEditable(false);
amountText.setEditable(false);
discountText.setEditable(false);
taxText.setEditable(false);
total_amountText.setEditable(false);

Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Payment ID:"));
first.add(pay_idText);
first.add(new Label("Payment Amount:"));
first.add(amountText);
first.add(new Label("Payment discount:"));
first.add(discountText);
first.add(new Label("Payment tax:"));
first.add(taxText);
first.add(new Label("Payment total amount:"));
first.add(total_amountText);

Panel second = new Panel(new GridLayout(4, 1));
second.add(deleteRowButton);

Panel third = new Panel();
```

```
        third.add(errorText);

        add(first);
        add(second);
        add(third);


        setLayout(new FlowLayout());
        setVisible(true);
    }

    public void viewEmployeesGUI()
    {
        removeAll();
        employeesList = new List(6);
        loadEmployees();
        add(employeesList);

        //When a list item is selected populate the text fields
        employeesList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent e)
            {
                try
                {
                    rs = statement.executeQuery("SELECT *
FROM employees");
```

```
                                                while (rs.next())
                                                {
                                                        if
(rs.getString("e_id").equals(employeesList.getSelectedItem()))
                                                        break;
                                                }
                                                if (!rs.isAfterLast())
                                                {
                                                        e_idText.setText(rs.getString("e_id"));

        e_nameText.setText(rs.getString("e_name"));
                                                        e_ageText.setText(rs.getString("e_age"));

        e_phoneText.setText(rs.getString("e_phone"));

        e_roleText.setText(rs.getString("e_role"));

        e_salaryText.setText(rs.getString("e_salary"));
                                                }
                                        }
                                        catch (SQLException selectException)
                                        {
                                                displaySQLErrors(selectException);
                                        }
                                }
                        });
                        //Handle Update Menu Button
                        modify = new Button("Update Employees");
                        modify.addActionListener(new ActionListener()
                        {
                                public void actionPerformed(ActionEvent e)
                                {
                                        try
                                        {
                                                Statement statement = 
connection.createStatement();
                                                int i = statement.executeUpdate("UPDATE 
employees "
                                                + "SET e_salary=" + e_salaryText.getText()
                                                + " WHERE e_id = '" + 
employeesList.getSelectedItem() + "'");
                                                errorText.append("\nUpdated " + i + " rows 
successfully");
                                                employeesList.removeAll();
                                                loadEmployees();
                                        }
                                        catch (SQLException insertException)
                                        {
                                                displaySQLErrors(insertException);
                                        }
                                }
                        });

                        e_idText = new TextField(15);
                        e_idText.setEditable(false);
                        e_nameText = new TextField(15);
                        e_nameText.setEditable(false);
```

```
e_ageText = new TextField(15);
e_ageText.setEditable(false);
e_phoneText = new TextField(15);
e_phoneText.setEditable(false);
e_roleText = new TextField(15);
e_roleText.setEditable(false);
e_salaryText = new TextField(15);
e_salaryText.setEditable(false);

errorText = new TextArea(10, 40);
errorText.setEditable(false);

Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Employee ID:"));
first.add(e_idText);
first.add(new Label("Employee Name:"));
first.add(e_nameText);
first.add(new Label("Employee Age:"));
first.add(e_ageText);
first.add(new Label("Employee Phone Number:"));
first.add(e_phoneText);
first.add(new Label("Employee Role:"));
first.add(e_roleText);
first.add(new Label("Employee Salary:"));
first.add(e_salaryText);
```

```
        Panel second = new Panel(new GridLayout(4, 1));
        //second.add(modify);

        Panel third = new Panel();
        third.add(errorText);

        add(first);

        add(second);
        add(third);

        //setTitle("Update ....");
        //setSize(500, 600);
        setLayout(new FlowLayout());
        setVisible(true);

    }

    public void viewPassengersGUI()
    {
        removeAll();
        passengersList = new List(6);
        loadPassengers();
        add(passengersList);

        //When a list item is selected populate the text fields
        passengersList.addItemListener(new ItemListener()
                        {
                                public void itemStateChanged(ItemEvent e)
                                {
                                        try
                                        {
                                                rs = statement.executeQuery("SELECT *
FROM passengers");
                                                while (rs.next())
                                                {
                                                        if
(rs.getString("p_id").equals(passengersList.getSelectedItem()))
                                                        break;
                                                }
                                                if (!rs.isAfterLast())
                                                {
                                                        p_idText.setText(rs.getString("p_id"));

        p_f_nameText.setText(rs.getString("p_f_name"));

        p_l_nameText.setText(rs.getString("p_l_name"));

        p_countryText.setText(rs.getString("p_country"));

        p_phoneText.setText(rs.getString("p_phone"));
                                                }
                                        }
                                        catch (SQLException selectException)
                                        {
                                                displaySQLErrors(selectException);
```

```
                }
            }
        });
        //Handle Update Menu Button
        modify = new Button("Update Passengers");
        modify.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                try
                {
                    Statement statement =
connection.createStatement();
                    int i = statement.executeUpdate("UPDATE
passengers "
                    + "SET p_f_name='" +
p_f_nameText.getText()+"'"
                    + " WHERE p_id = '" +
passengersList.getSelectedItem() + "'");
                    errorText.append("\nUpdated " + i + " rows
successfully");
                    passengersList.removeAll();
                    loadPassengers();
                }
                catch (SQLException insertException)
                {
                    displaySQLErrors(insertException);
                }
            }
```

```
});

p_idText = new TextField(15);
p_idText.setEditable(false);
p_f_nameText = new TextField(15);
p_f_nameText.setEditable(false);
p_l_nameText = new TextField(15);
p_l_nameText.setEditable(false);
p_phoneText = new TextField(15);
p_phoneText.setEditable(false);
p_countryText = new TextField(15);
p_countryText.setEditable(false);



errorText = new TextArea(10, 40);
errorText.setEditable(false);

Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Passenger ID:"));
first.add(p_idText);
first.add(new Label("Passenger First Name:"));
first.add(p_f_nameText);
first.add(new Label("Passenger Last Name"));
first.add(p_l_nameText);
first.add(new Label("Passenger Country:"));
first.add(p_countryText);
```

```
		first.add(new Label("Passenger Phone Number:"));
		first.add(p_phoneText);

		Panel second = new Panel(new GridLayout(4, 1));
		//second.add(modify);

		Panel third = new Panel();
		third.add(errorText);

		add(first);

		add(second);
		add(third);

		//setTitle("Update ....");
		//setSize(500, 600);
		setLayout(new FlowLayout());
		setVisible(true);

	}

	public void viewAirportGUI()
	{
		removeAll();
		airportList = new List(6);
		loadAirport();
		add(airportList);
```

```
				//When a list item is selected populate the text fields
				airportList.addItemListener(new ItemListener()
				{
					public void itemStateChanged(ItemEvent e)
					{
						try
						{
							rs = statement.executeQuery("SELECT *
FROM airport");
							while (rs.next())
							{
								if
(rs.getString("a_id").equals(airportList.getSelectedItem()))
								break;
							}
							if (!rs.isAfterLast())
							{
								a_idText.setText(rs.getString("a_id"));

	a_nameText.setText(rs.getString("a_name"));

	a_countryText.setText(rs.getString("a_country"));

	a_stateText.setText(rs.getString("a_state"));

	a_cityText.setText(rs.getString("a_city"));
								a_zipText.setText(rs.getString("a_zip"));
							}
```

```
                    }
                    catch (SQLException selectException)
                    {
                        displaySQLErrors(selectException);
                    }
                }
            });
            //Handle Update Airport Button
            modify = new Button("Update Airport");
            modify.addActionListener(new ActionListener()
            {
                public void actionPerformed(ActionEvent e)
                {
                    try
                    {
                        Statement statement =
connection.createStatement();
                        int i = statement.executeUpdate("UPDATE
airport "
                        + "SET a_name='" + a_nameText.getText()+"'"
                        + " WHERE a_id = '" +
airportList.getSelectedItem() + "'");
                        errorText.append("\nUpdated " + i + " rows
successfully");
                        airportList.removeAll();
                        loadAirport();
                    }
                    catch (SQLException insertException)
                    {
                        displaySQLErrors(insertException);
                    }
                }
            });

            a_idText = new TextField(15);
            a_idText.setEditable(false);
            a_nameText = new TextField(15);
            a_nameText.setEditable(false);
            a_countryText = new TextField(15);
            a_countryText.setEditable(false);
            a_stateText = new TextField(15);
            a_stateText.setEditable(false);
            a_cityText = new TextField(15);
            a_cityText.setEditable(false);
            a_zipText = new TextField(15);
            a_zipText.setEditable(false);

            errorText = new TextArea(10, 40);
            errorText.setEditable(false);

            Panel first = new Panel();
            first.setLayout(new GridLayout(4, 2));
            first.add(new Label("Airport ID:"));
            first.add(a_idText);
            first.add(new Label("Airport Name:"));
```

```
        first.add(a_nameText);

        first.add(new Label("Airport Country"));

        first.add(a_countryText);

        first.add(new Label("Airport State:"));

        first.add(a_stateText);

        first.add(new Label("Airport City:"));

        first.add(a_cityText);

        first.add(new Label("Airport zip code:"));

        first.add(a_zipText);


        Panel second = new Panel(new GridLayout(4, 1));

        //second.add(modify);


        Panel third = new Panel();

        third.add(errorText);


        add(first);


        add(second);

        add(third);


        //setTitle("Update ....");

        //setSize(500, 600);

        setLayout(new FlowLayout());

        setVisible(true);

    }
```

```
                public void viewFlight_HoursGUI()
                {
                        removeAll();
                        flight_hoursList = new List(6);
                        loadFlight_Hours();
                        add(flight_hoursList);

                        //When a list item is selected populate the text fields
                        flight_hoursList.addItemListener(new ItemListener()
                        {
                                public void itemStateChanged(ItemEvent e)
                                {
                                        try
                                        {
                                                rs = statement.executeQuery("SELECT *
FROM flight_hours");
                                                while (rs.next())
                                                {
                                                        if
(rs.getString("airplane_no").equals(flight_hoursList.getSelectedItem()))
                                                        break;
                                                }
                                                if (!rs.isAfterLast())
                                                {

        airplane_noText.setText(rs.getString("airplane_no"));

        departure_airportText.setText(rs.getString("departure_airport"));
```

```
          departure_timeText.setText(rs.getString("departure_time"));

          arrival_airportText.setText(rs.getString("arrival_airport"));

          arrival_timeText.setText(rs.getString("arrival_time"));
                                                  }
                                        }
                                        catch (SQLException selectException)
                                        {
                                                  displaySQLErrors(selectException);
                                        }
                              }
                    });
                    //Handle Update Menu Button
                    modify = new Button("Update Flight Hours");
                    modify.addActionListener(new ActionListener()
                    {
                              public void actionPerformed(ActionEvent e)
                              {
                                        try
                                        {
                                                  Statement statement =
connection.createStatement();
                                                  int i = statement.executeUpdate("UPDATE
flight_hours "
                                                  + "SET departure_airport='" +
departure_airportText.getText()+"'"
                                                  + " WHERE airplane_no = '" +
flight_hoursList.getSelectedItem() + "'");
```

```
                        errorText.append("\nUpdated " + i + " rows
successfully");

                        flight_hoursList.removeAll();

                        loadFlight_Hours();

                    }

                    catch (SQLException insertException)

                    {

                        displaySQLErrors(insertException);

                    }

                }

            });


            airplane_noText = new TextField(15);

            airplane_noText.setEditable(false);

            departure_airportText = new TextField(15);

            departure_airportText.setEditable(false);

            departure_timeText = new TextField(15);

            departure_timeText.setEditable(false);

            arrival_airportText = new TextField(15);

            arrival_airportText.setEditable(false);

            arrival_timeText = new TextField(15);

            arrival_timeText.setEditable(false);


            errorText = new TextArea(10, 40);

            errorText.setEditable(false);


            Panel first = new Panel();
```

```
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Airplane No:"));
first.add(airplane_noText);
first.add(new Label("Departure Airport:"));
first.add(departure_airportText);
first.add(new Label("Departure Time:"));
first.add(departure_timeText);
first.add(new Label("Arrival Airport:"));
first.add(arrival_airportText);
first.add(new Label("Arrival Time:"));
first.add(arrival_timeText);

Panel second = new Panel(new GridLayout(4, 1));
//second.add(modify);

Panel third = new Panel();
third.add(errorText);

add(first);


add(second);
add(third);

//setTitle("Update ....");
//setSize(500, 600);
setLayout(new FlowLayout());
setVisible(true);
```

```
		}

		public void viewPaymentsGUI()
		{
			removeAll();
			paymentsList = new List(6);
			loadPayments();
			add(paymentsList);

			//When a list item is selected populate the text fields
			paymentsList.addItemListener(new ItemListener()
			{
				public void itemStateChanged(ItemEvent e)
				{
					try
					{
						rs = statement.executeQuery("SELECT *
FROM payments");
						while (rs.next())
						{
							if
(rs.getString("pay_id").equals(paymentsList.getSelectedItem()))
							break;
						}
						if (!rs.isAfterLast())
						{

	pay_idText.setText(rs.getString("pay_id"));
```

```
          amountText.setText(rs.getString("amount"));

          discountText.setText(rs.getString("discount"));
                                                            taxText.setText(rs.getString("tax"));

          total_amountText.setText(rs.getString("total_amount"));
                                                  }
                                        }
                                        catch (SQLException selectException)
                                        {
                                                  displaySQLErrors(selectException);
                                        }
                              }
                    });
                    //Handle Update Menu Button
                    modify = new Button("Update Payments");
                    modify.addActionListener(new ActionListener()
                    {
                              public void actionPerformed(ActionEvent e)
                              {
                                        try
                                        {
                                                  Statement statement =
connection.createStatement();
                                                  int i = statement.executeUpdate("UPDATE
payments "
                                                  + "SET amount='" + amountText.getText()+"'"
                                        + " WHERE pay_id = '" +
paymentsList.getSelectedItem() + "'");
                                        errorText.append("\nUpdated " + i + " rows
successfully");
                                        paymentsList.removeAll();
                                        loadPayments();
                                   }
                                   catch (SQLException insertException)
                                   {
                                        displaySQLErrors(insertException);
                                   }
                              }
                         });

                         pay_idText = new TextField(15);
                         pay_idText.setEditable(false);
                         amountText = new TextField(15);
                         amountText.setEditable(false);
                         discountText = new TextField(15);
                         discountText.setEditable(false);
                         taxText = new TextField(15);
                         taxText.setEditable(false);
                         total_amountText = new TextField(15);
                         total_amountText.setEditable(false);

                         errorText = new TextArea(10, 40);
                         errorText.setEditable(false);
```

```
Panel first = new Panel();

first.setLayout(new GridLayout(4, 2));

first.add(new Label("Payment ID:"));

first.add(pay_idText);

first.add(new Label("Payment Amount:"));

first.add(amountText);

first.add(new Label("Payment discount:"));

first.add(discountText);

first.add(new Label("Payment tax:"));

first.add(taxText);

first.add(new Label("Payment total amount:"));

first.add(total_amountText);


Panel second = new Panel(new GridLayout(4, 1));

//second.add(modify);


Panel third = new Panel();

third.add(errorText);


add(first);


add(second);

add(third);


//setTitle("Update ....");

//setSize(500, 600);

setLayout(new FlowLayout());
```

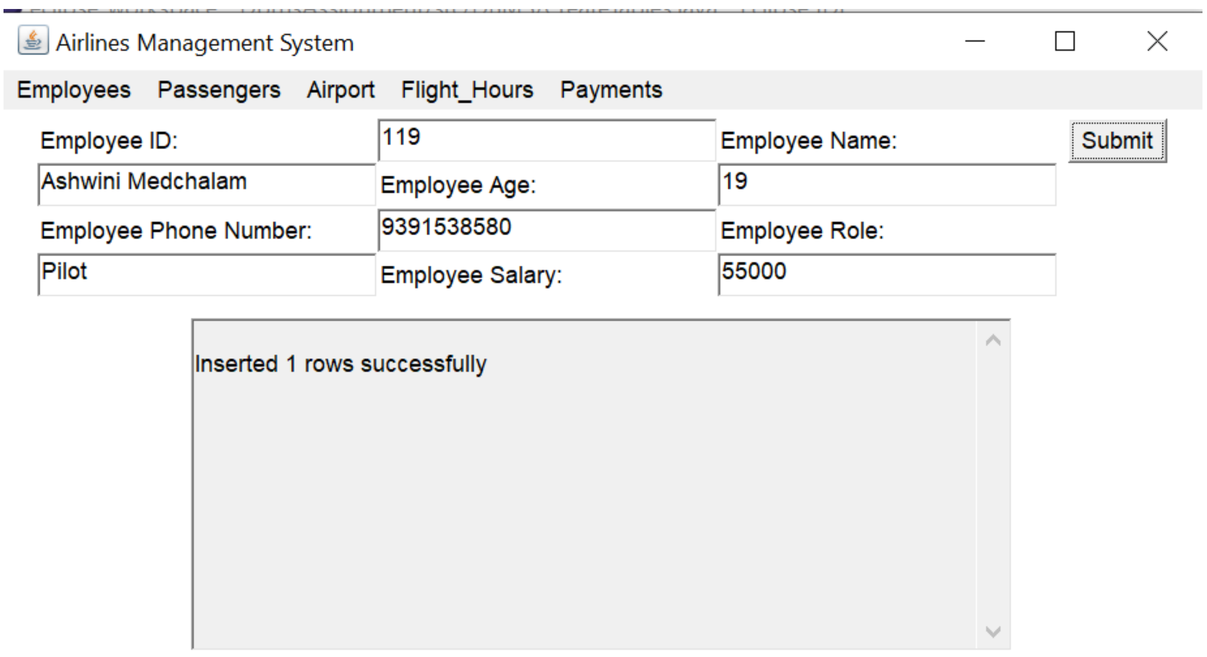
```
                setVisible(true);
        }

        public void displaySQLErrors(SQLException e)
        {
                errorText.append("\nSQLException: " + e.getMessage() + "\n");
                errorText.append("SQLState:     " + e.getSQLState() + "\n");
                errorText.append("VendorError:  " + e.getErrorCode() + "\n");
        }
        public static void main(String[] args)
        {
                CreateTables it = new CreateTables();
                it.addWindowListener(new WindowAdapter(){
                  public void windowClosing(WindowEvent e)
                  {
                        System.exit(0);
                  }
                });
                it.buildFrame();
        }
}
```

Github links and folder structure:

Link: https://github.com/ashu025/AIRLINES-DATABASE-MANAGEMENT-SYSTEM

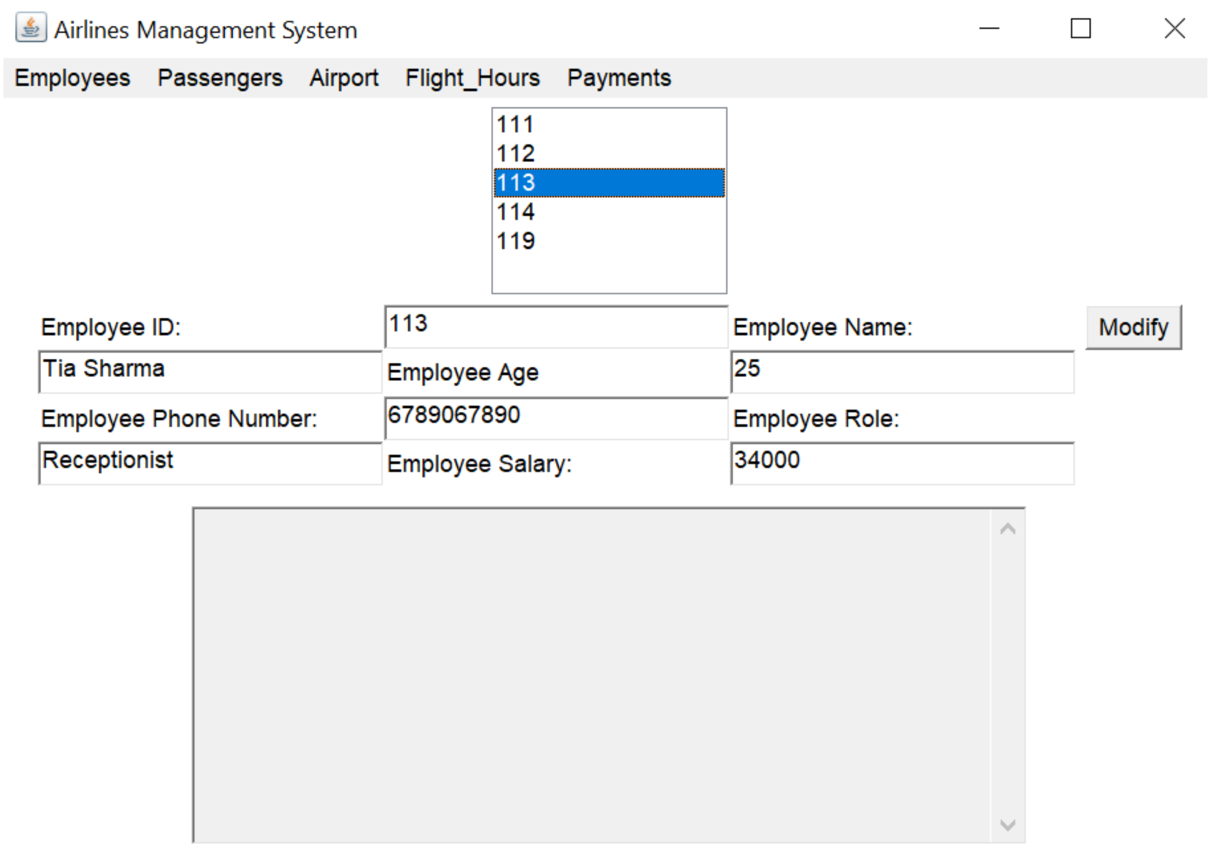
TESTING

OUTPUT SCREENSHOTS:

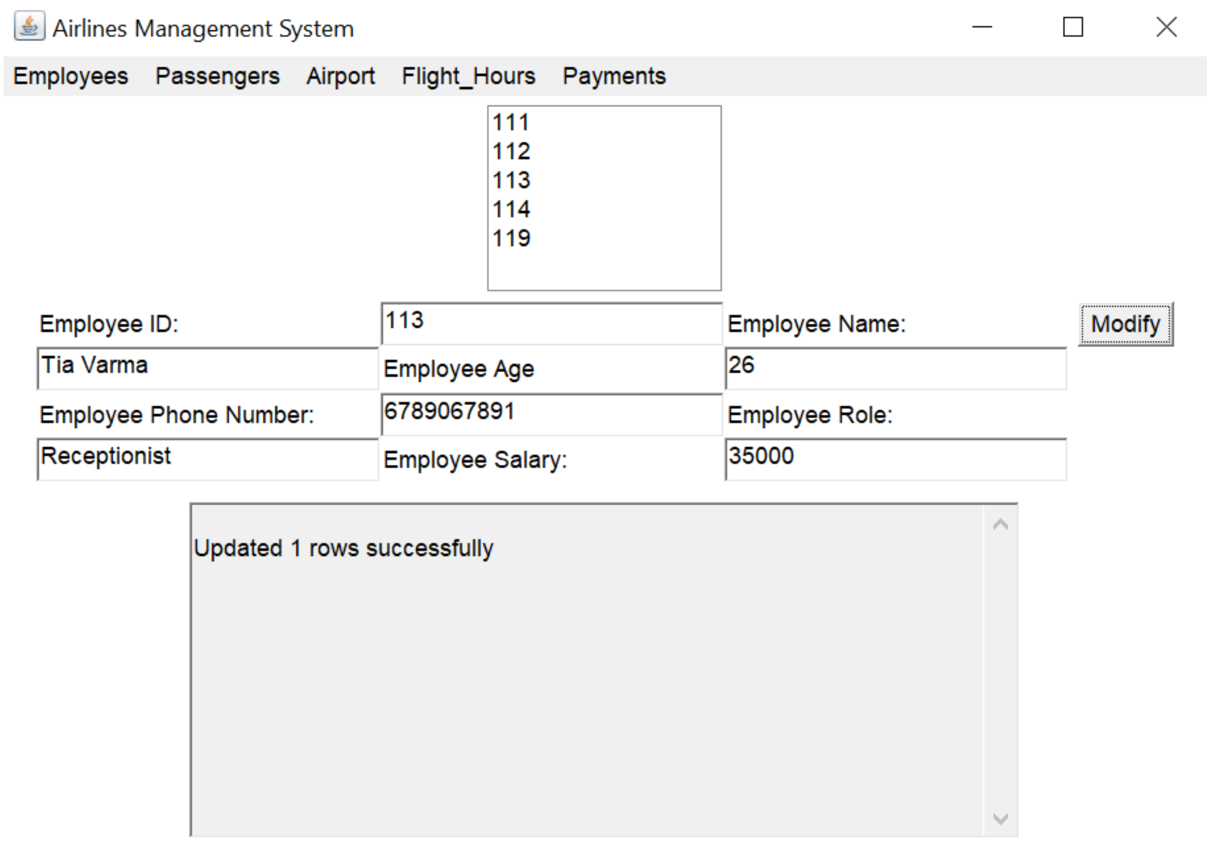


Employee ID:	119	Employee Name:	Ashwini Medchalam
Employee Age:	19	Employee Phone Number:	9391538580
Employee Role:	Pilot	Employee Salary:	55000

```
Inserted 1 rows successfully
```



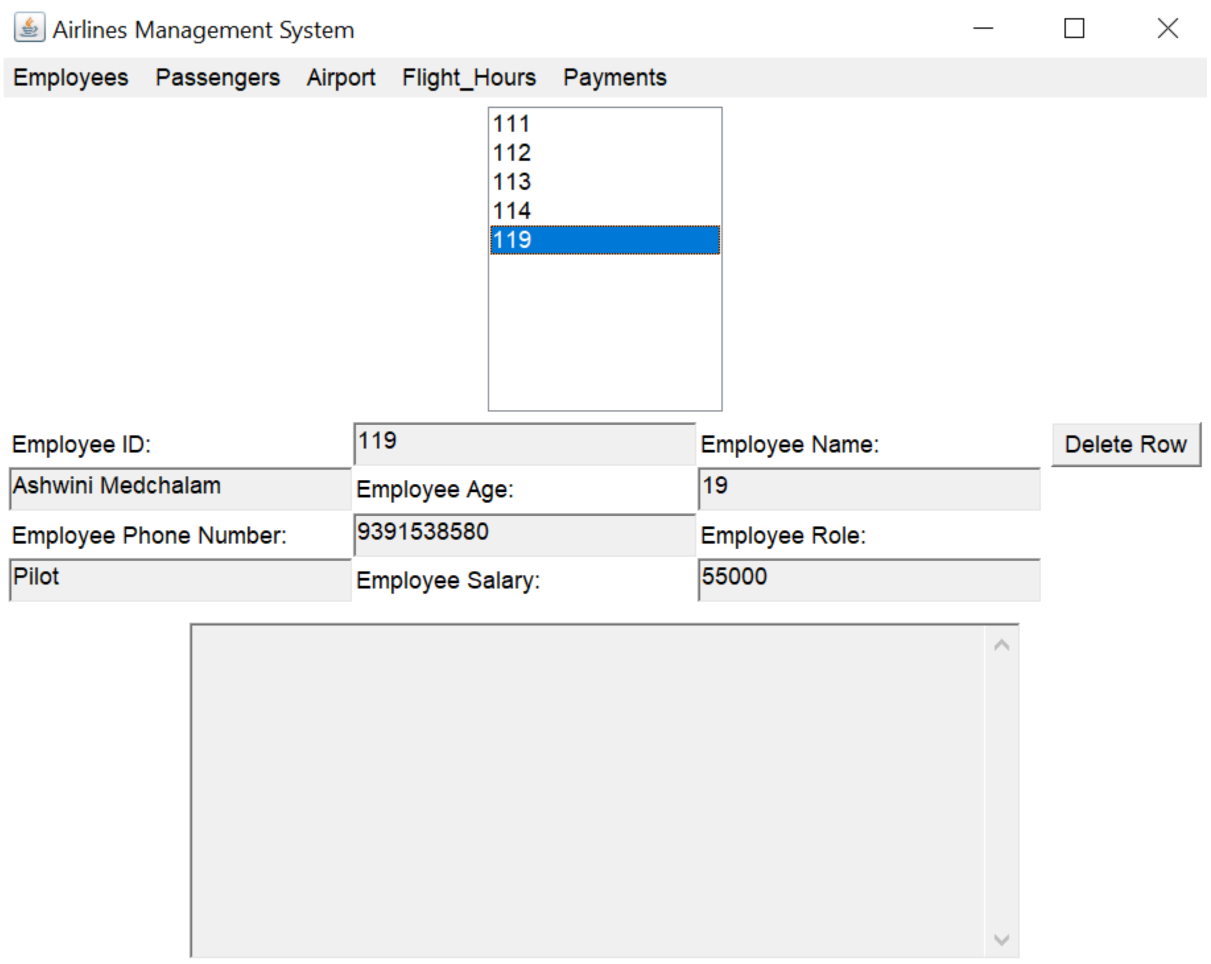
Employee ID:	113	Employee Name:	Tia Sharma
Employee Age	25	Employee Phone Number:	6789067890
Employee Role:	Receptionist	Employee Salary:	34000



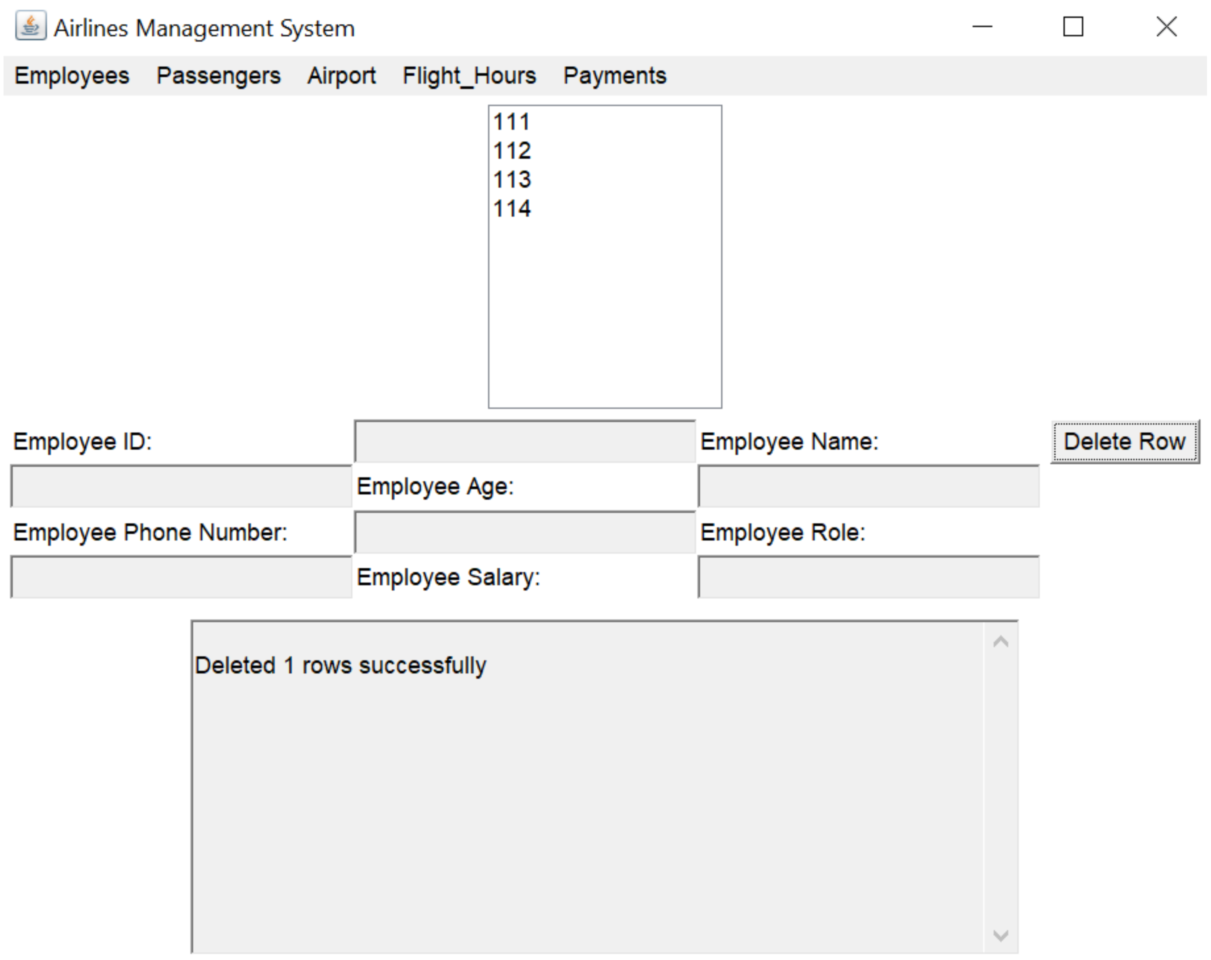
```
111
112
113
114
119
```

Employee ID:	113	Employee Name:	
Tia Varma	Employee Age	26	
Employee Phone Number:	6789067891	Employee Role:	
Receptionist	Employee Salary:	35000	

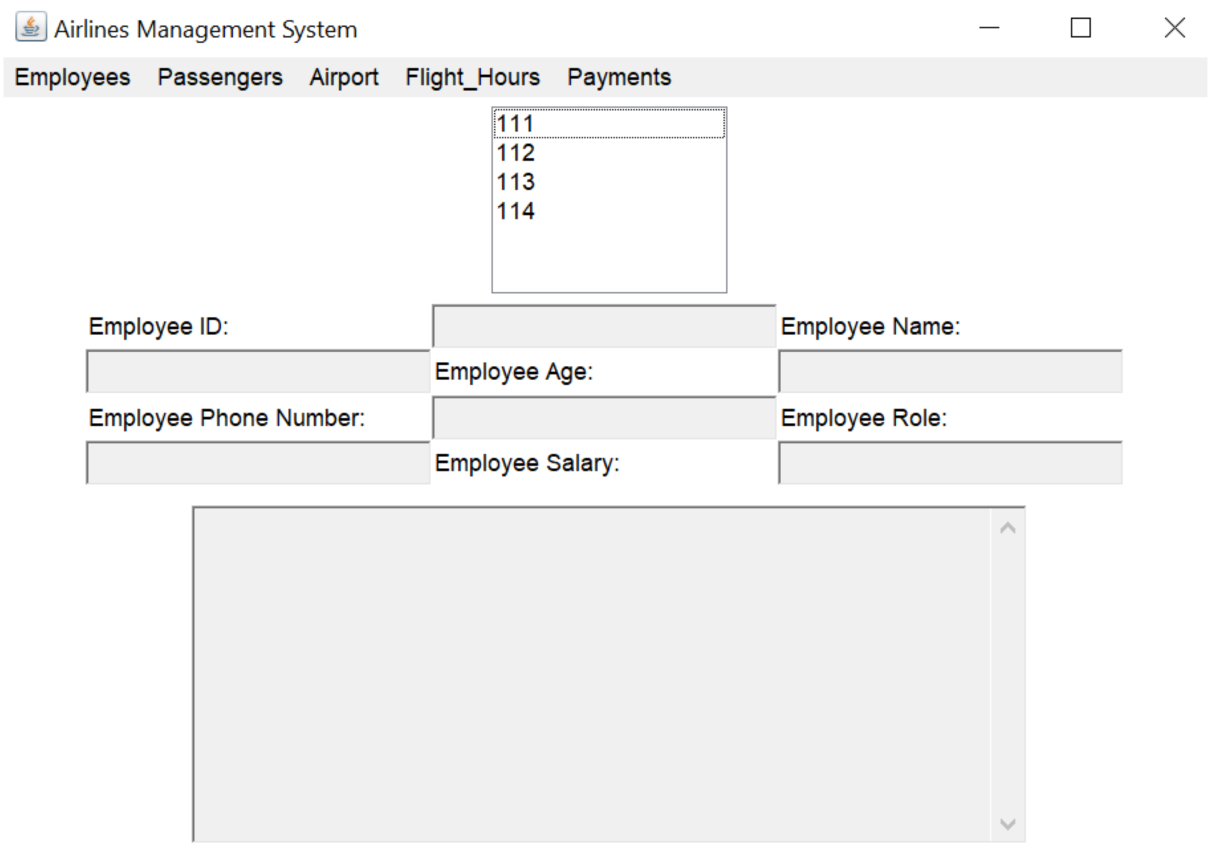
```
Updated 1 rows successfully
```



Employee ID:	119	Employee Name:	
Ashwini Medchalam	Employee Age:	19	
Employee Phone Number:	9391538580	Employee Role:	
Pilot	Employee Salary:	55000	



```
Deleted 1 rows successfully
```

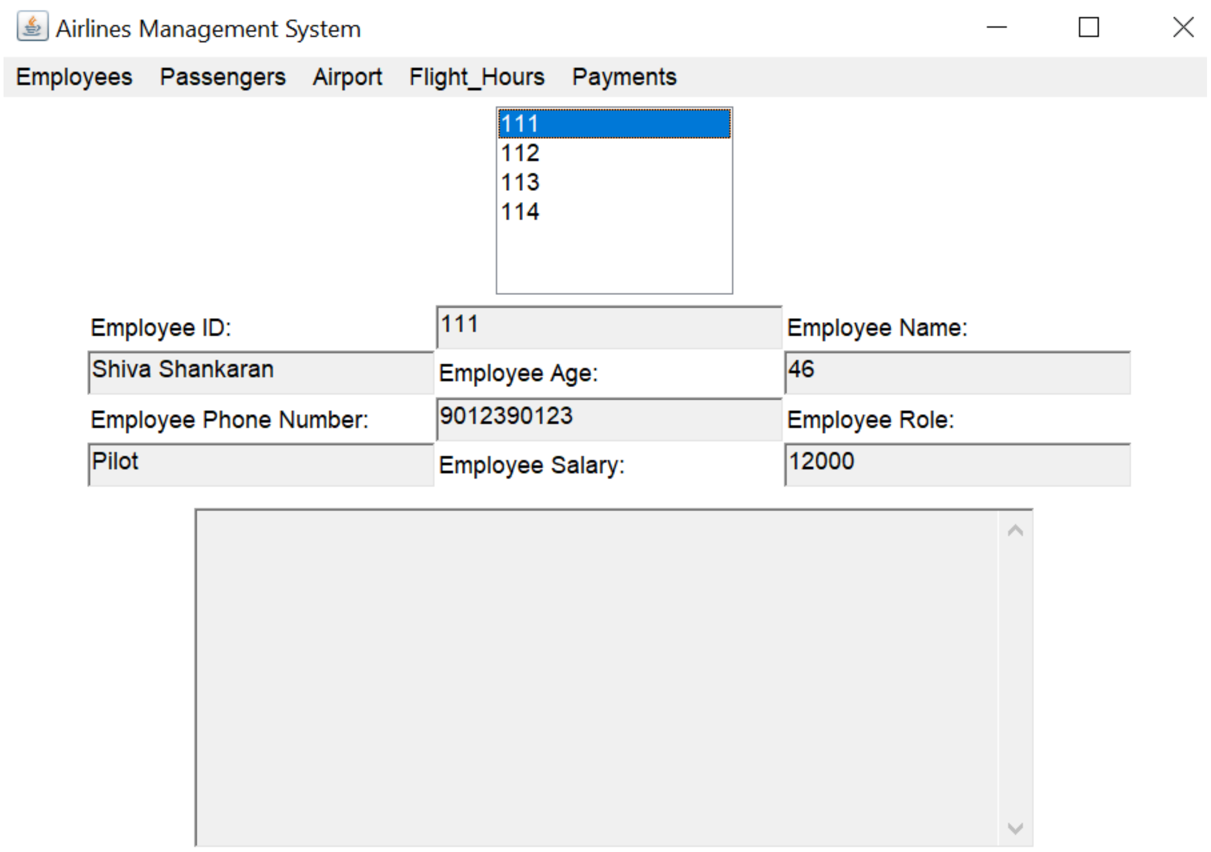


Airlines Management System

Employees Passengers Airport Flight_Hours Payments

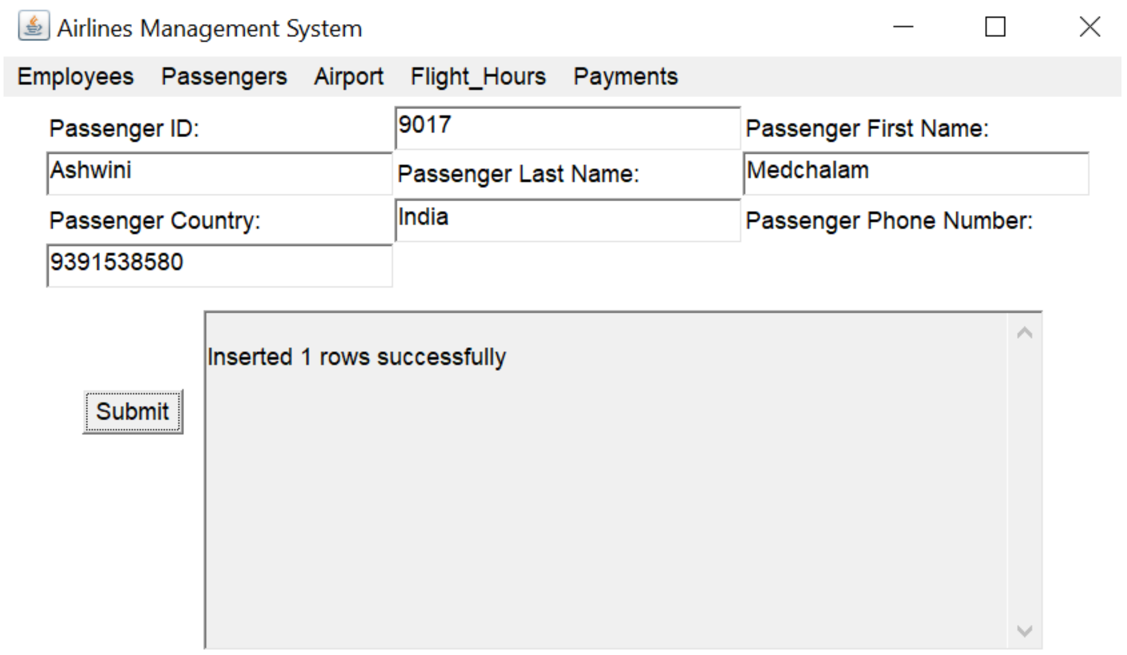
111
112
113
114

Employee ID:
Employee Name:
Employee Age:
Employee Phone Number:
Employee Role:
Employee Salary:


Airlines Management System
Employees Passengers Airport Flight_Hours Payments

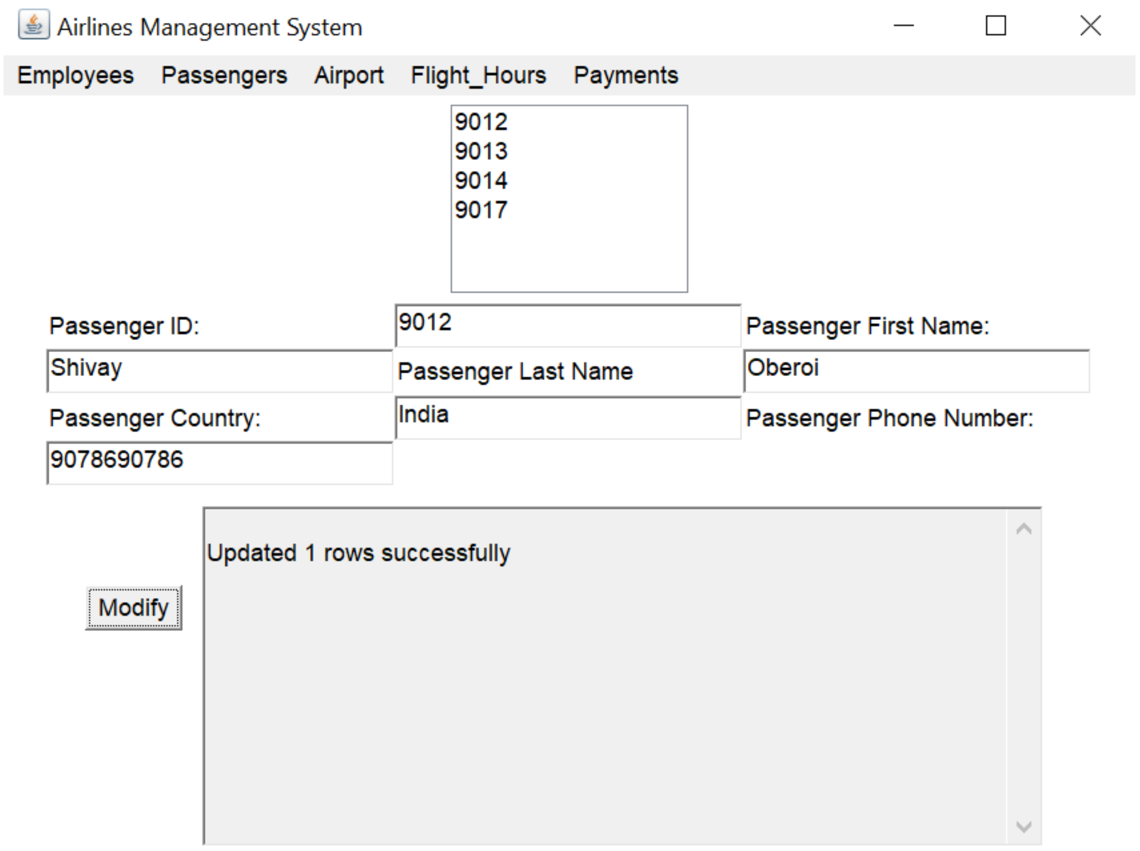
```
111
112
113
114
```

Employee ID:	111	Employee Name:	
Shiva Shankaran	Employee Age:	46	
Employee Phone Number:	9012390123	Employee Role:	
Pilot	Employee Salary:	12000	


Airlines Management System
Employees Passengers Airport Flight_Hours Payments
Submit

Passenger ID:	9017	Passenger First Name:	
Ashwini	Passenger Last Name:	Medchalam	
Passenger Country:	India	Passenger Phone Number:	
9391538580			

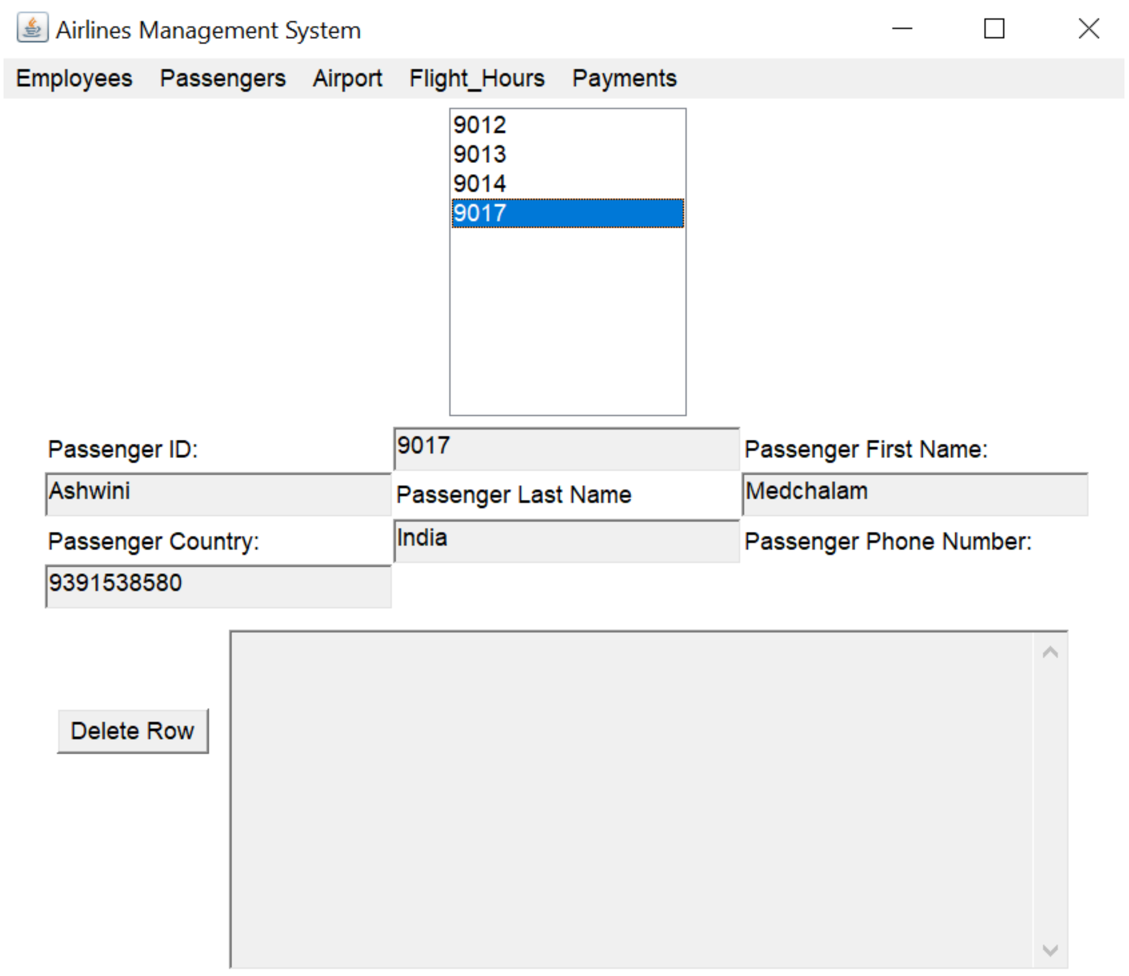
```
Inserted 1 rows successfully
```



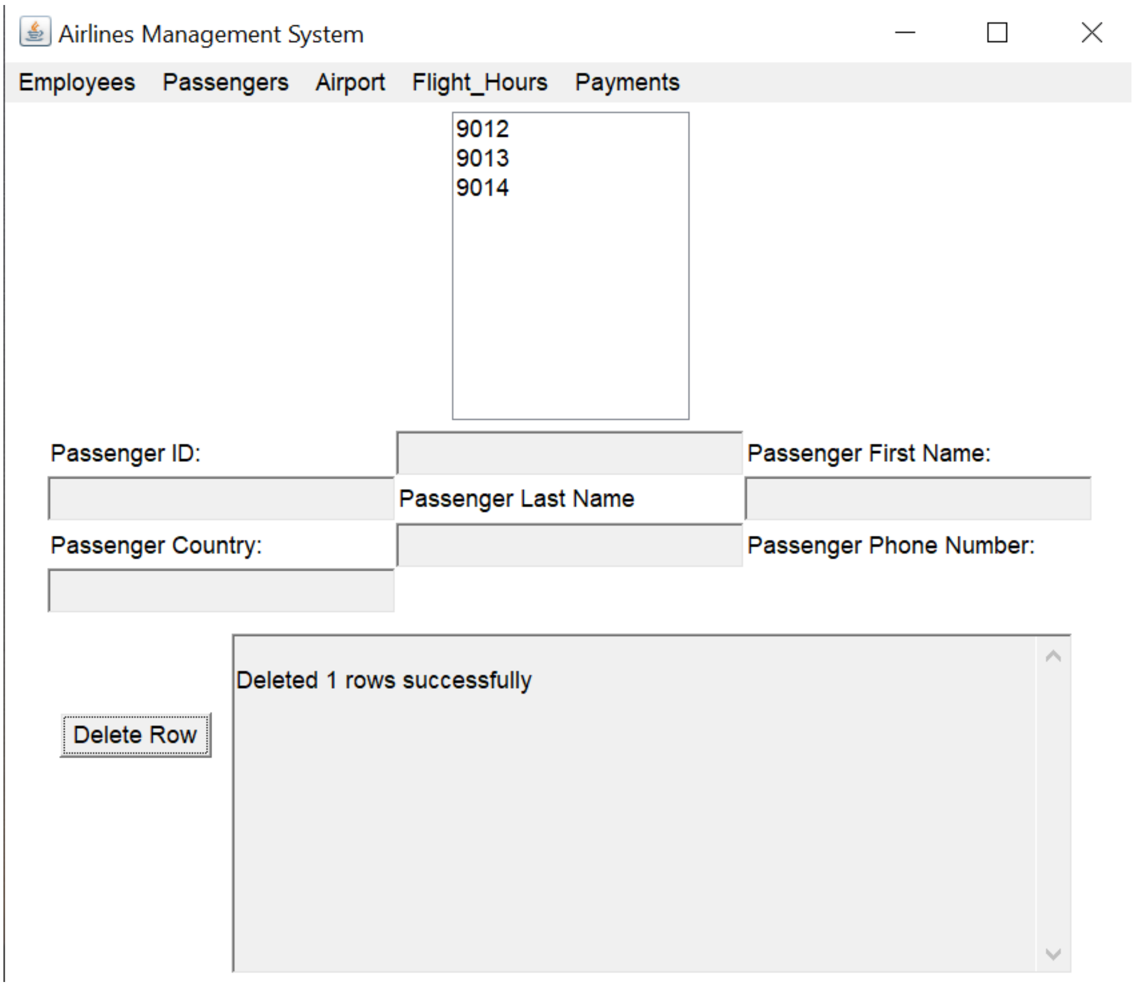
9012
9013
9014
9017

Passenger ID:	9012	Passenger First Name:
Shivay	Passenger Last Name	Oberoi
Passenger Country:	India	Passenger Phone Number:
9078690786		

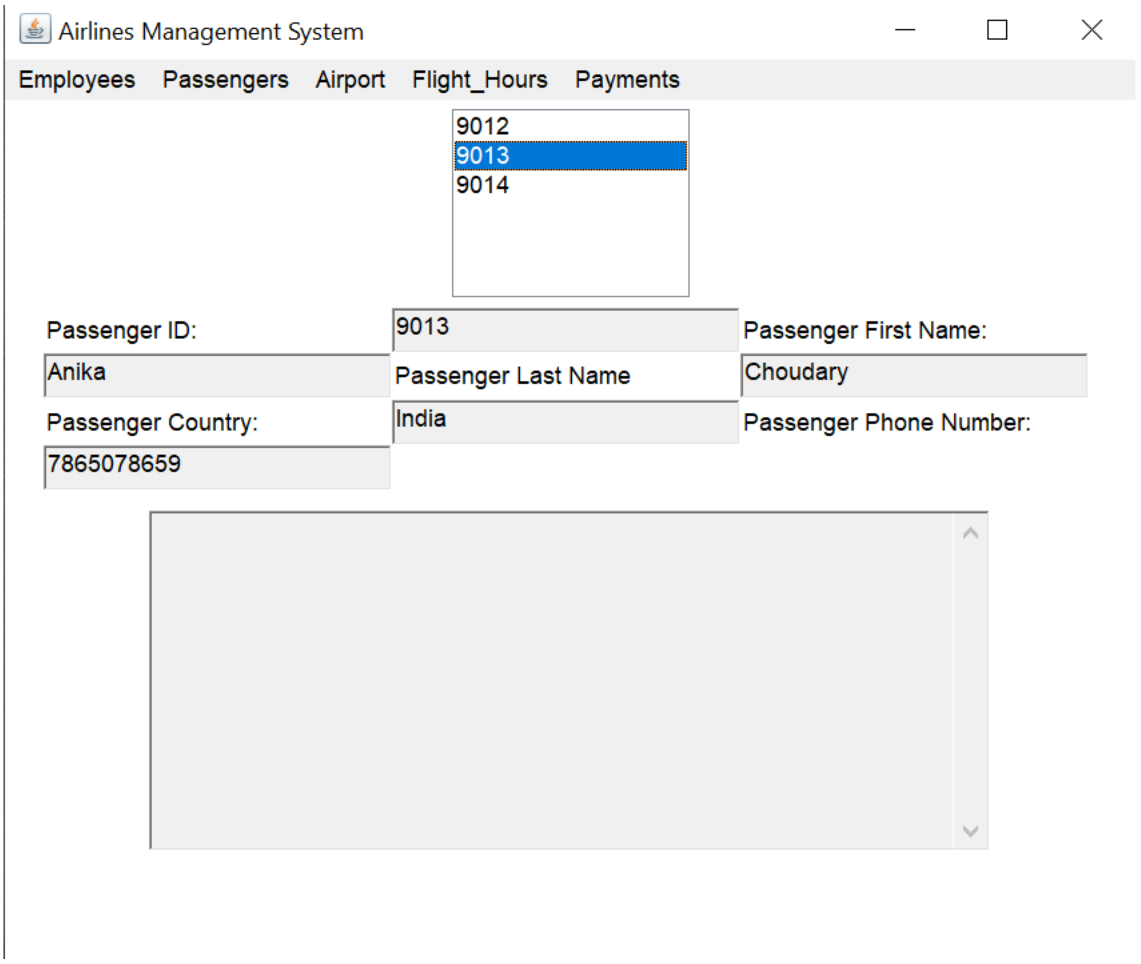
```
Updated 1 rows successfully
```



Passenger ID: 9017
Passenger First Name: Ashwini
Passenger Last Name Medchalam
Passenger Country: India
Passenger Phone Number: 9391538580

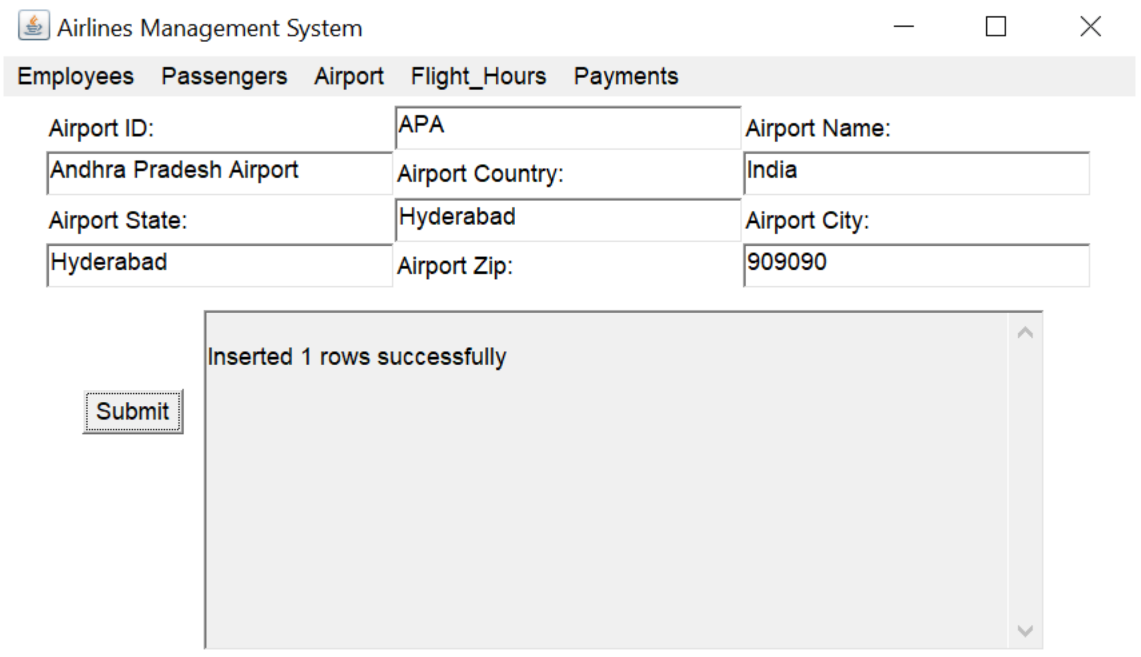


Deleted 1 rows successfully


Airlines Management System
Employees Passengers Airport Flight_Hours Payments

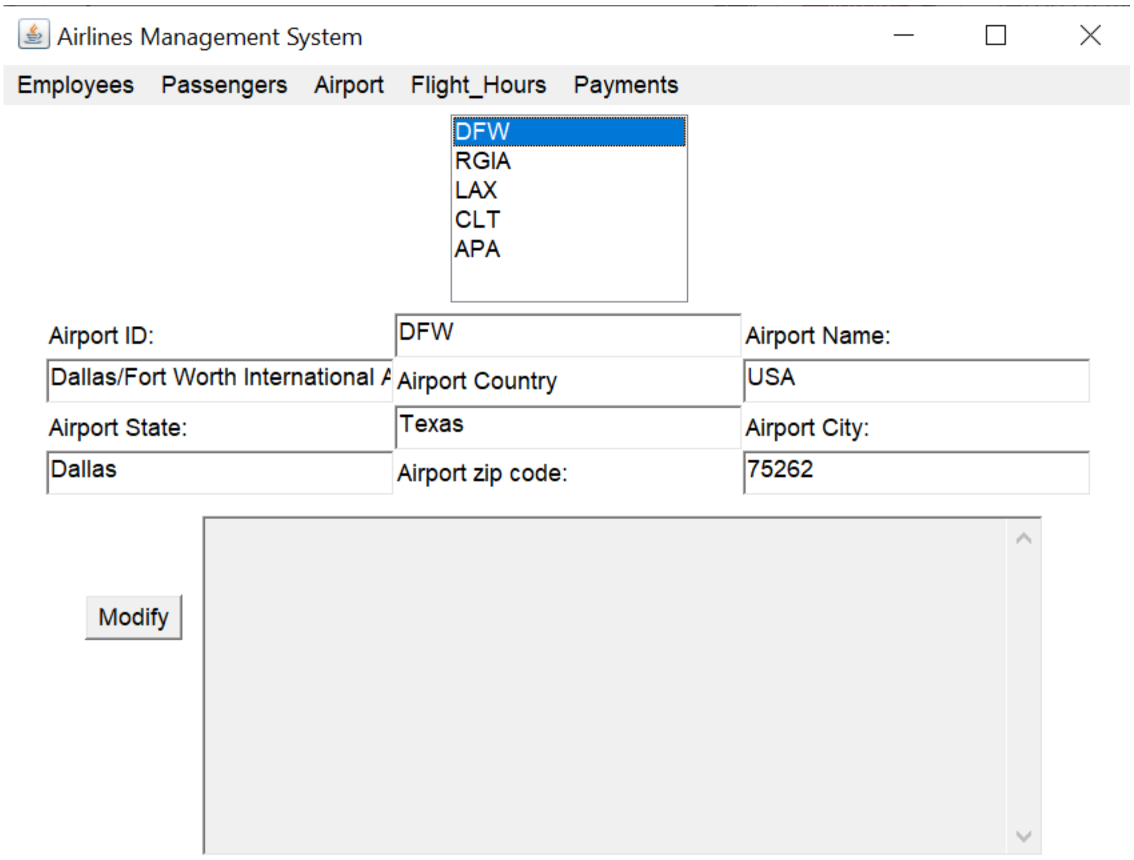
9012
9013
9014

Passenger ID:	9013	Passenger First Name:
Anika	Passenger Last Name	Choudary
Passenger Country:	India	Passenger Phone Number:
7865078659		


Airlines Management System
Employees Passengers Airport Flight_Hours Payments
Submit

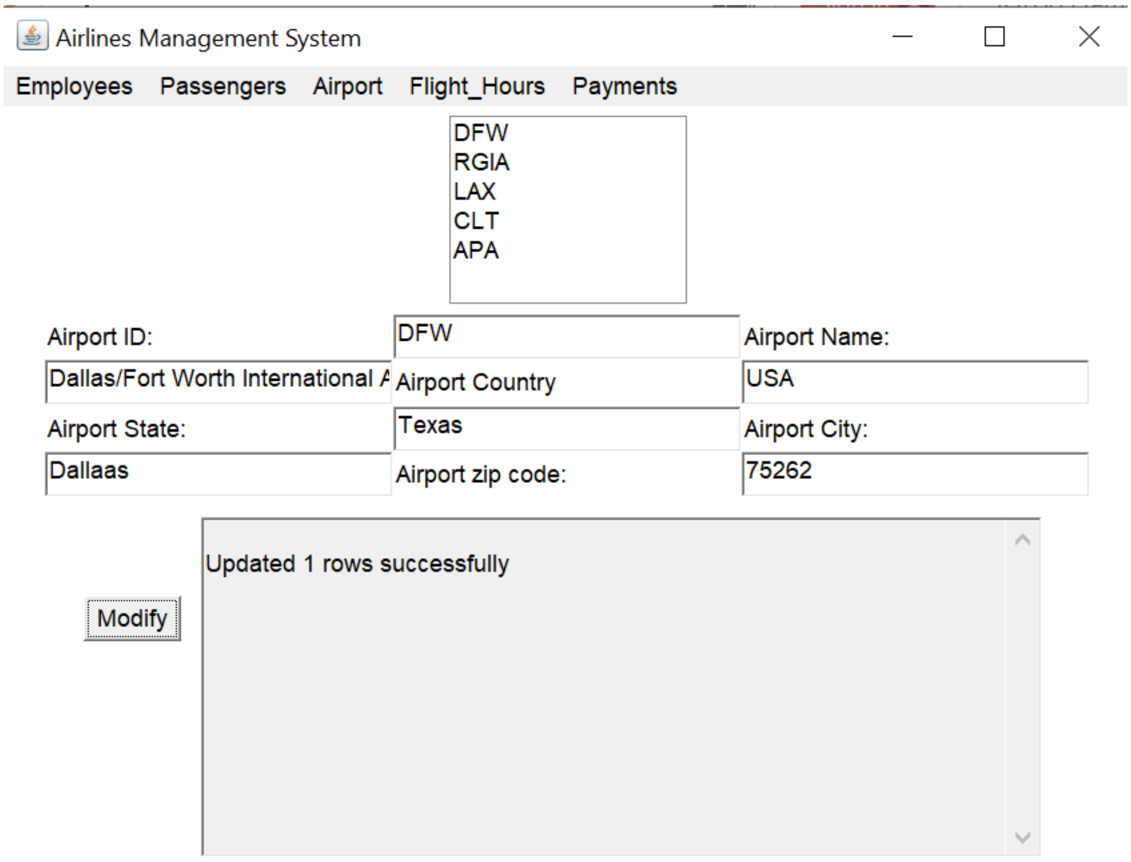
Airport ID:	APA	Airport Name:
Andhra Pradesh Airport	Airport Country:	India
Airport State:	Hyderabad	Airport City:
Hyderabad	Airport Zip:	909090

```
Inserted 1 rows successfully
```


Airlines Management System

DFW
RGIA
LAX
CLT
APA

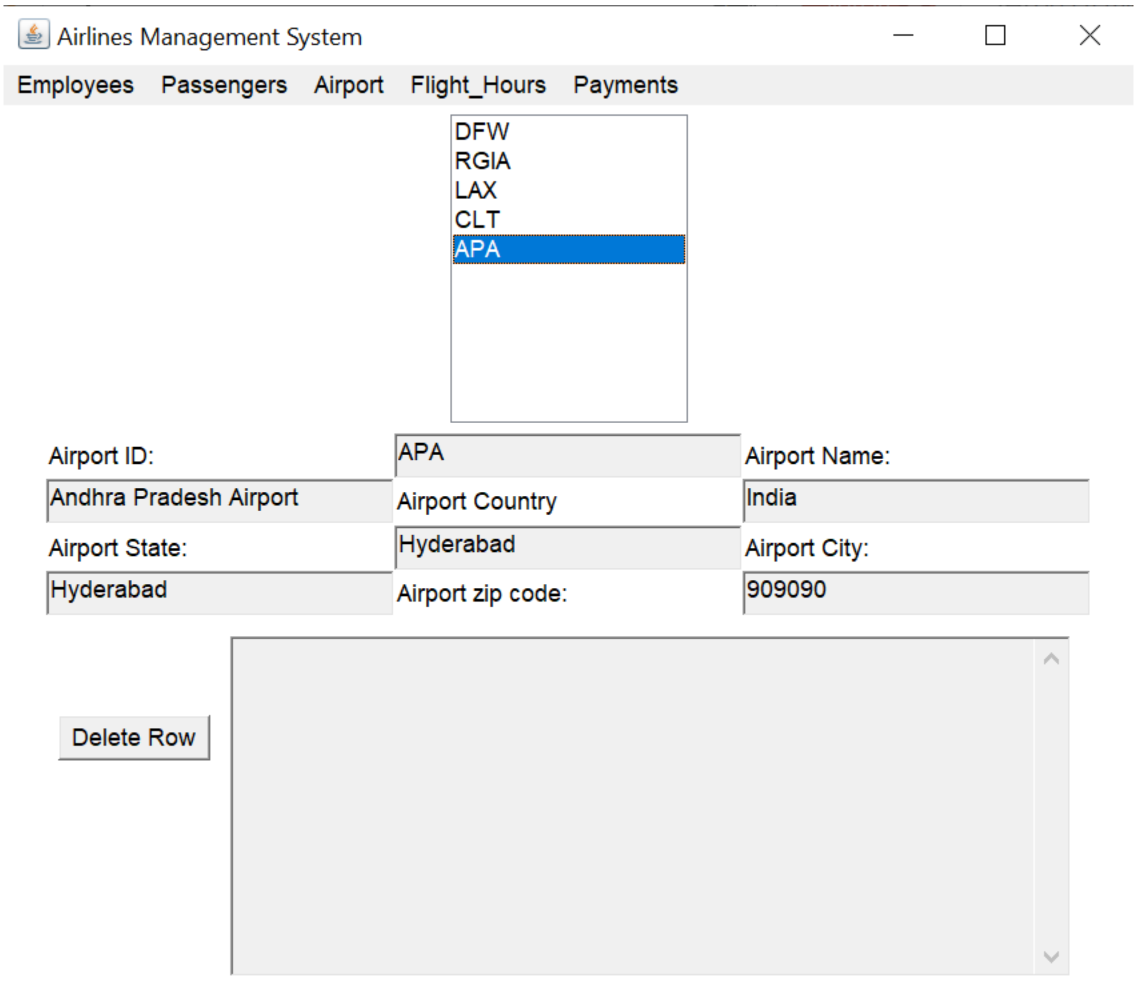
Airport ID:	DFW	Airport Name:	
Dallas/Fort Worth International A	Airport Country	USA	
Airport State:	Texas	Airport City:	
Dallas	Airport zip code:	75262	



```
DFW
RGIA
LAX
CLT
APA
```

Airport ID:	DFW	Airport Name:	
Dallas/Fort Worth International A	Airport Country	USA	
Airport State:	Texas	Airport City:	
Dallaas	Airport zip code:	75262	

```
Updated 1 rows successfully
```



Airport ID: APA

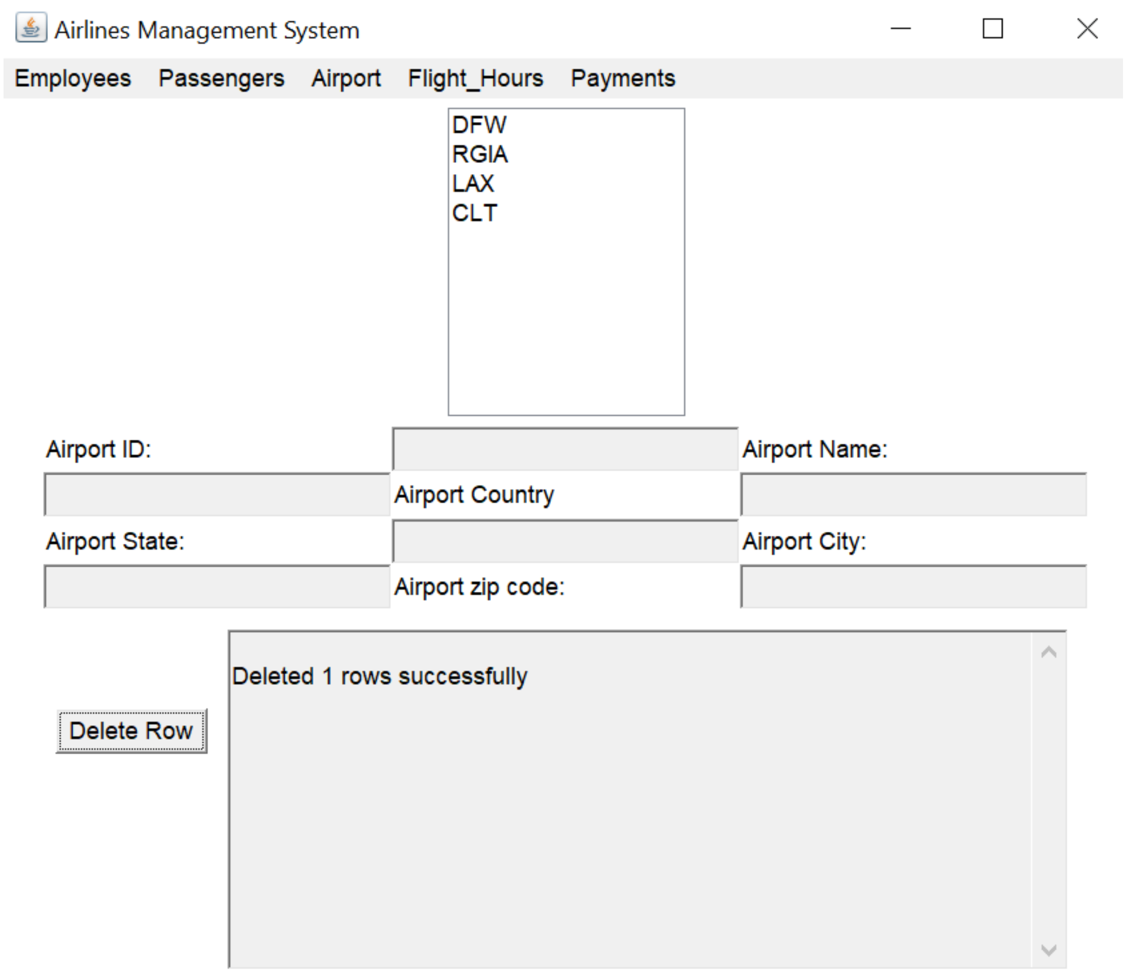
Airport Name: Andhra Pradesh Airport

Airport Country: India

Airport State: Hyderabad

Airport City: Hyderabad

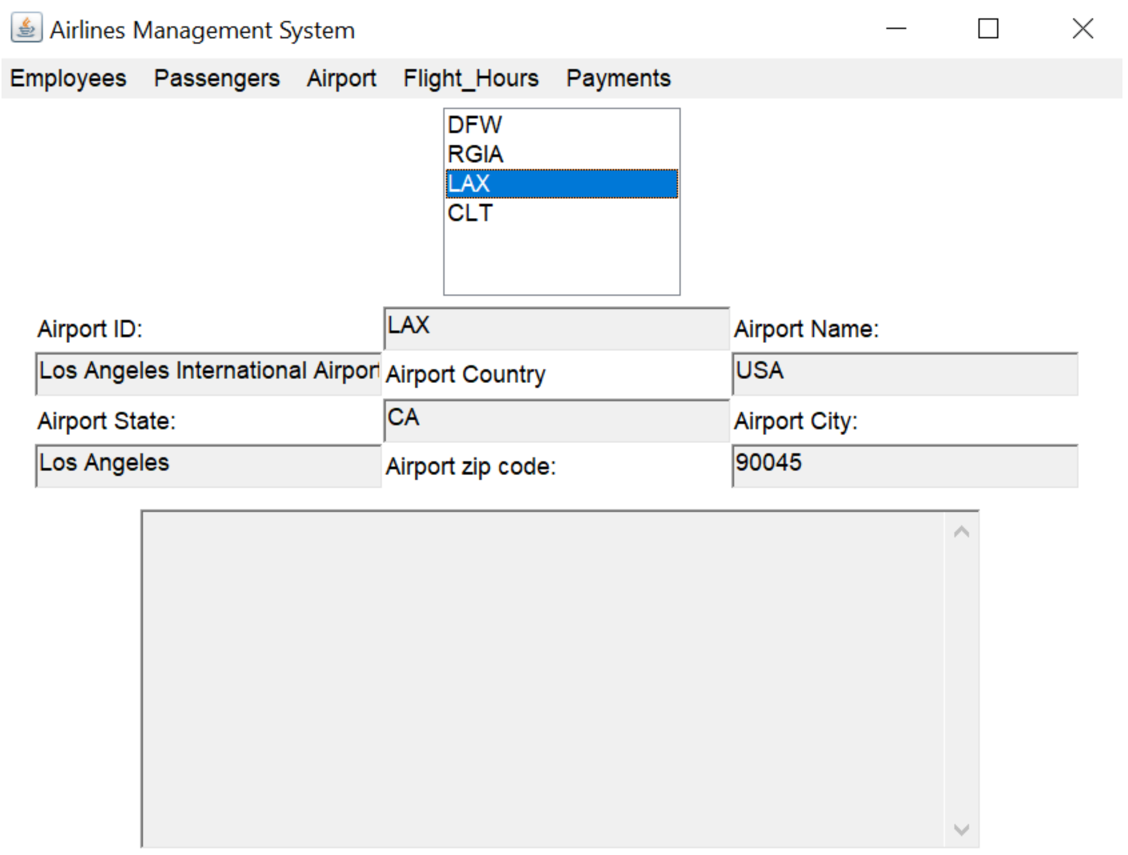
Airport zip code: 909090



DFW
RGIA
LAX
CLT

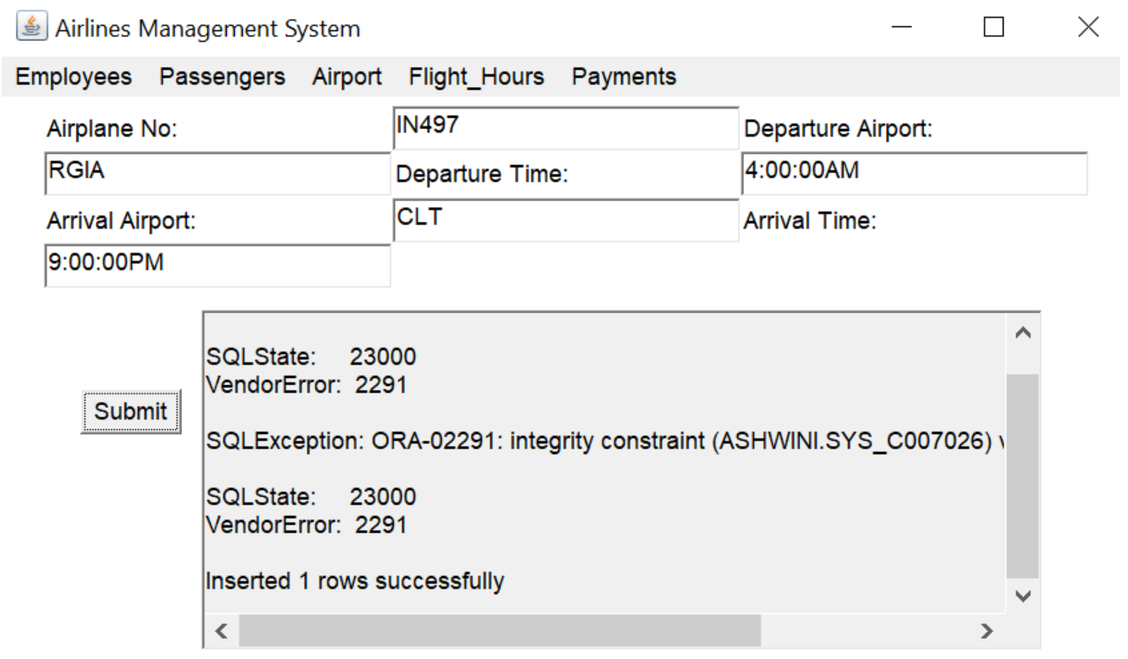
Airport ID:
Airport Name:
Airport Country
Airport State:
Airport City:
Airport zip code:

```
Deleted 1 rows successfully
```



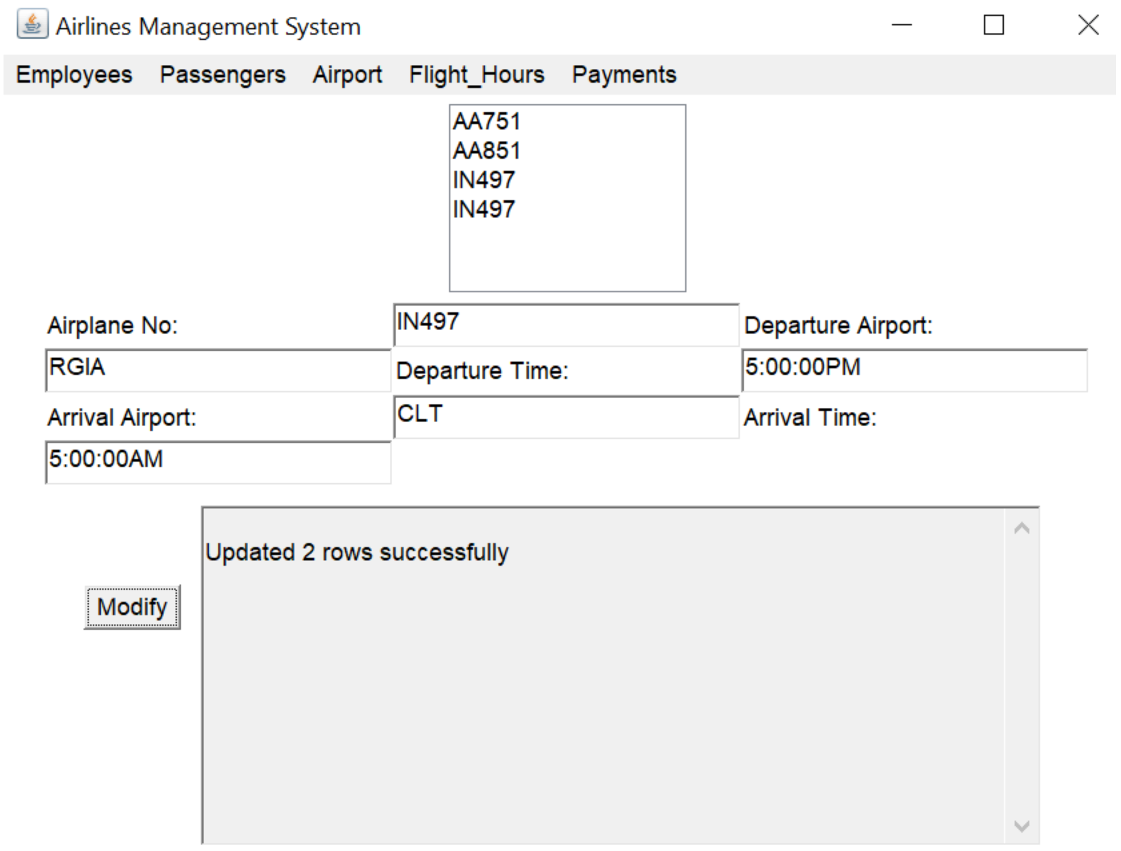
DFW
RGIA
LAX
CLT

Airport ID:	LAX	Airport Name:	Los Angeles International Airpor
Airport Country	USA	Airport State:	CA
Airport City:	Los Angeles	Airport zip code:	90045



Airplane No:	IN497	Departure Airport:	RGIA
Departure Time:	4:00:00AM	Arrival Airport:	CLT
Arrival Time:	9:00:00PM		

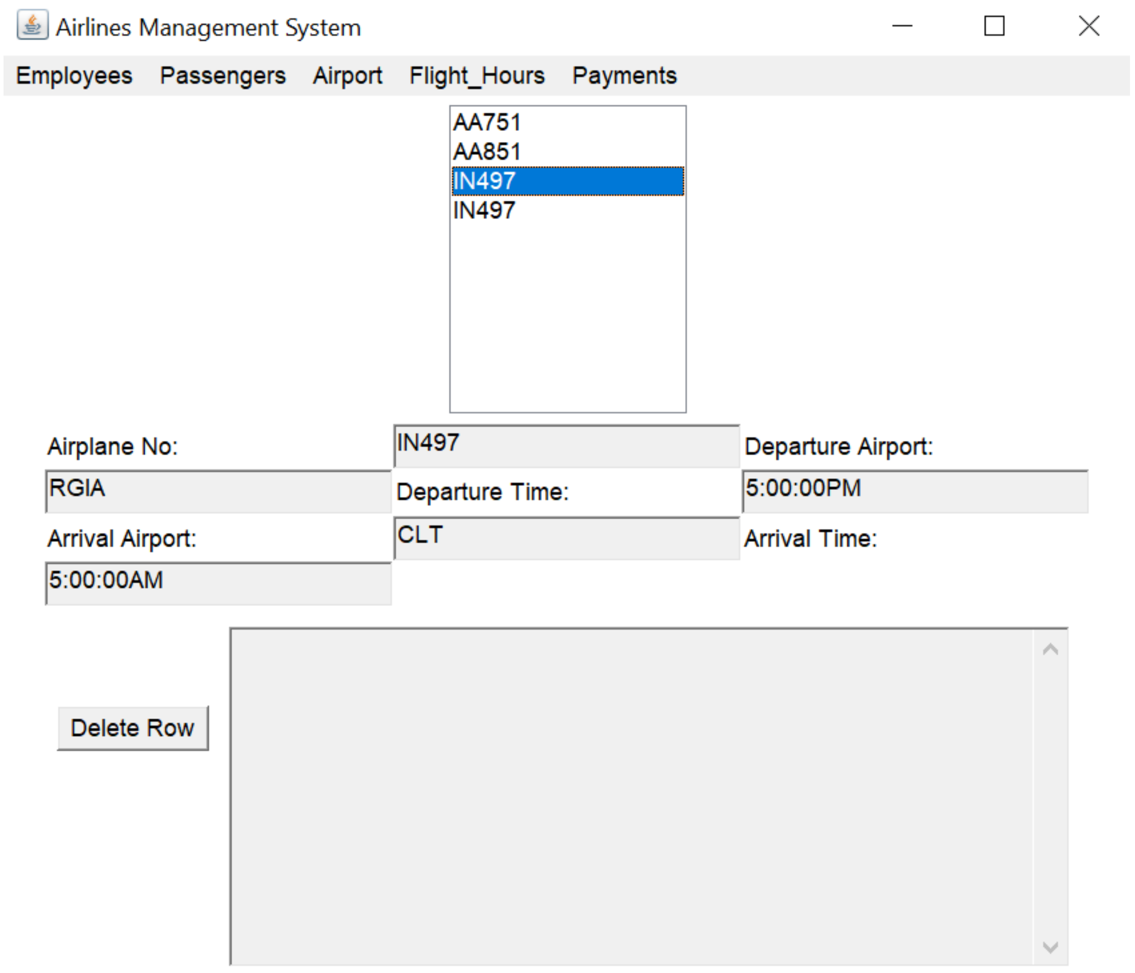
```
SQLState:    23000
VendorError:  2291

SQLException: ORA-02291: integrity constraint (ASHWINI.SYS_C007026)

SQLState:    23000
VendorError:  2291

Inserted 1 rows successfully
```



AA751
AA851
IN497
IN497

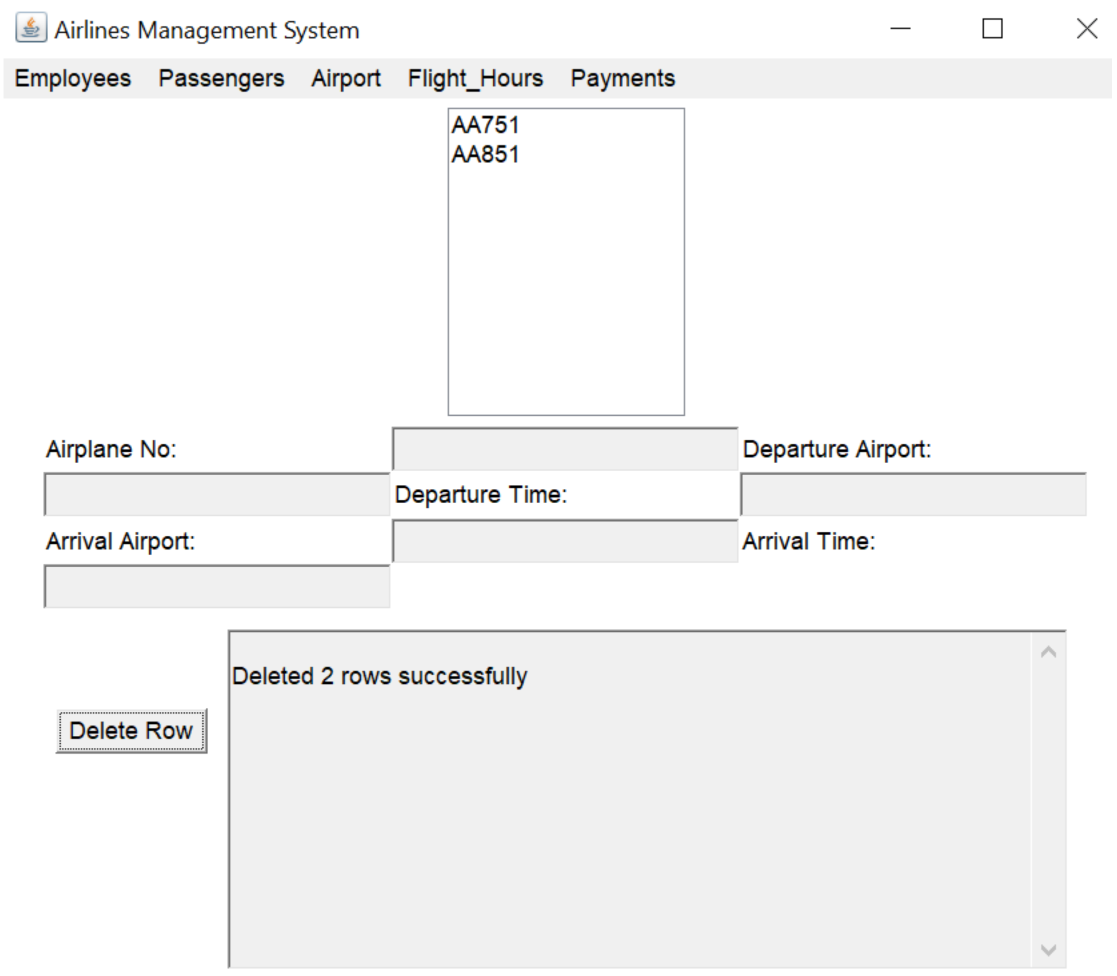
Airplane No:	IN497	Departure Airport:
RGIA	Departure Time:	5:00:00PM
Arrival Airport:	CLT	Arrival Time:
5:00:00AM		

```
Updated 2 rows successfully
```



AA751
AA851
IN497
IN497

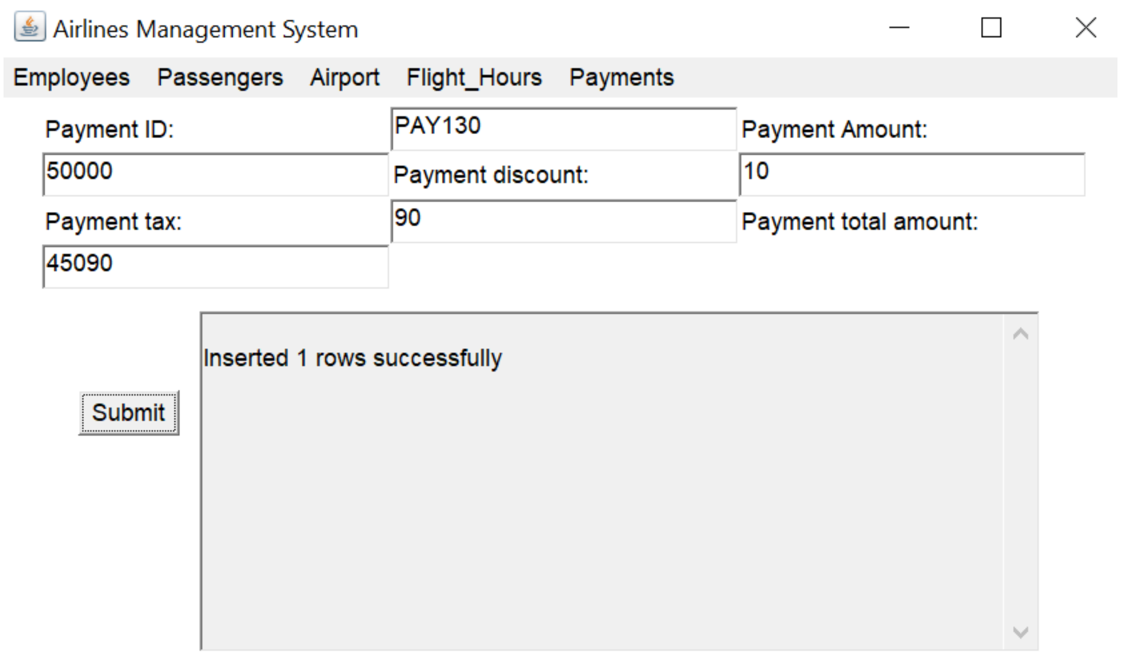
Airplane No: IN497
Departure Airport: RGIA
Departure Time: 5:00:00PM
Arrival Airport: CLT
Arrival Time: 5:00:00AM



AA751
AA851

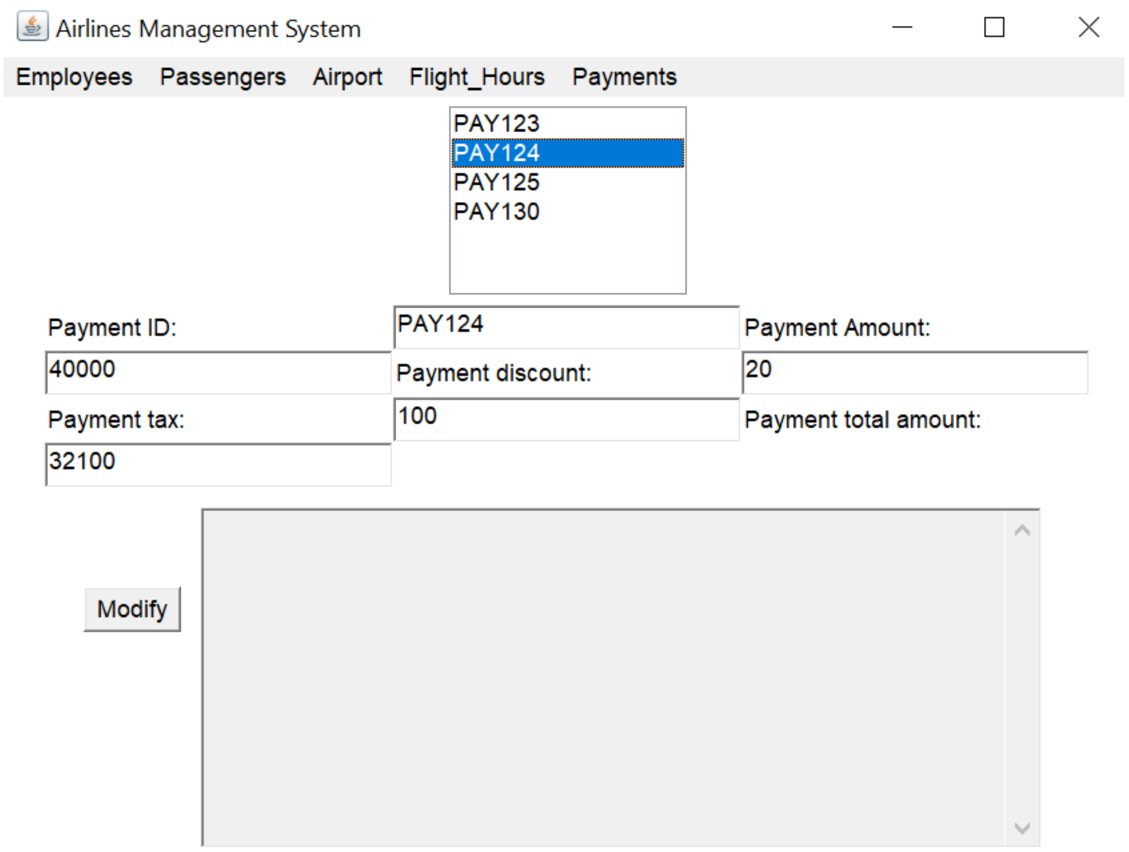
Airplane No:
Departure Airport:
Departure Time:
Arrival Airport:
Arrival Time:

Deleted 2 rows successfully



Payment ID: PAY130
Payment Amount: 50000
Payment discount: 10
Payment tax: 90
Payment total amount: 45090

Inserted 1 rows successfully



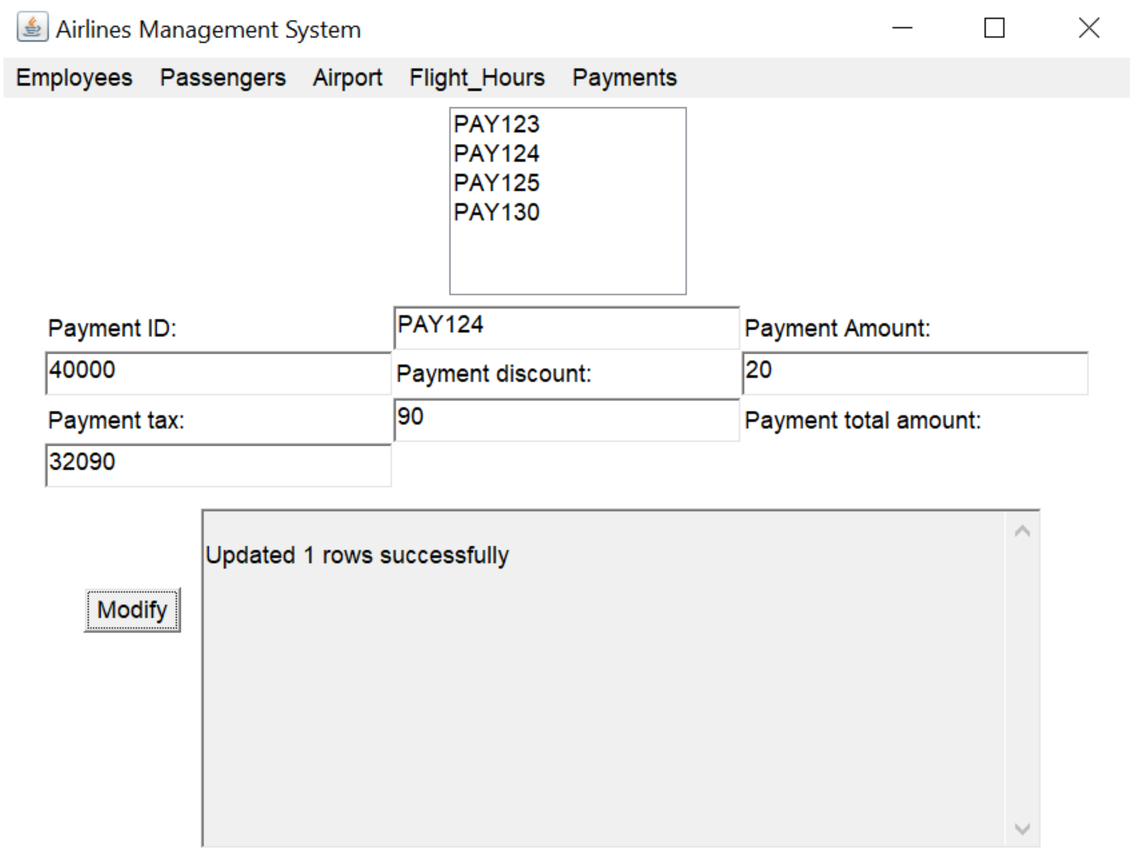
Airlines Management System

Employees Passengers Airport Flight_Hours Payments

PAY123
PAY124
PAY125
PAY130

Payment ID:	PAY124	Payment Amount:
40000	Payment discount:	20
Payment tax:	100	Payment total amount:
32100		

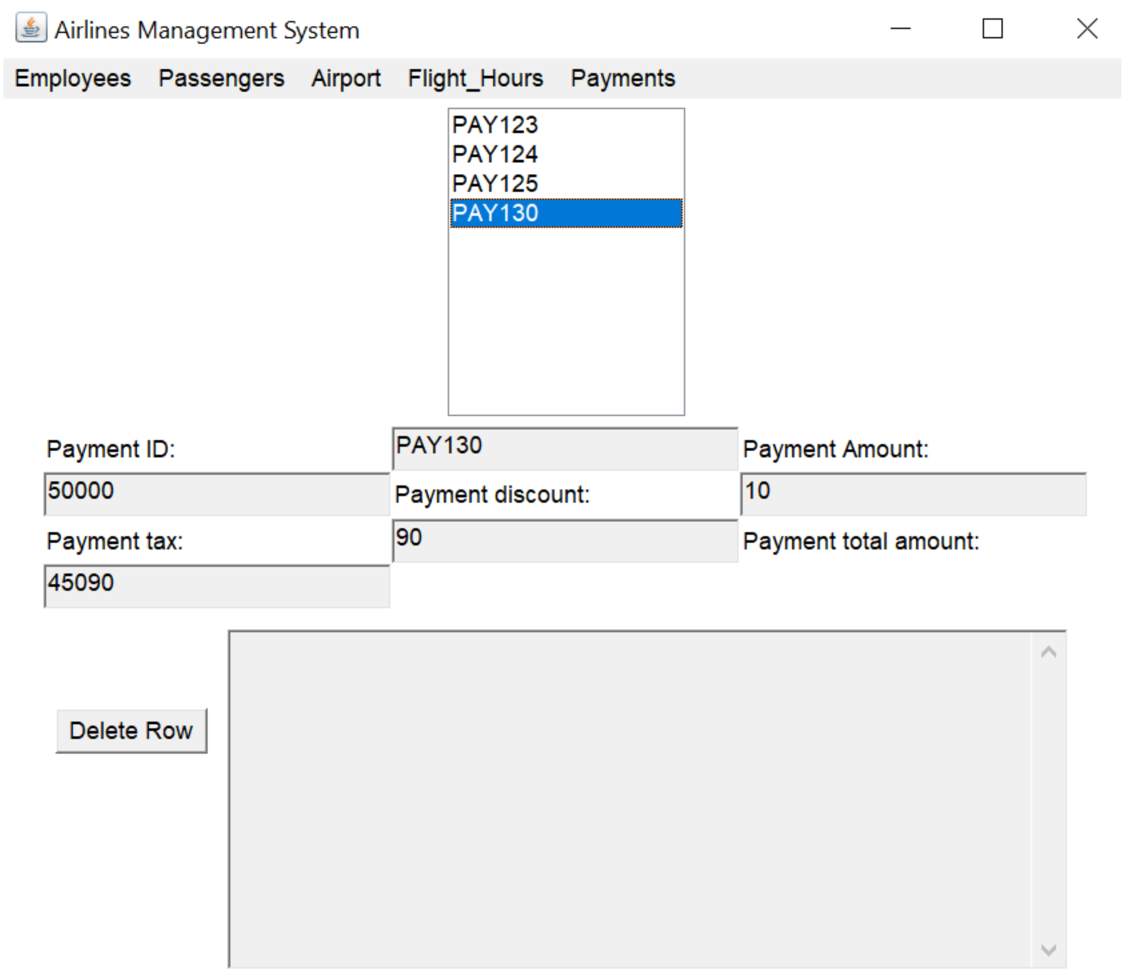
Modify



PAY123
PAY124
PAY125
PAY130

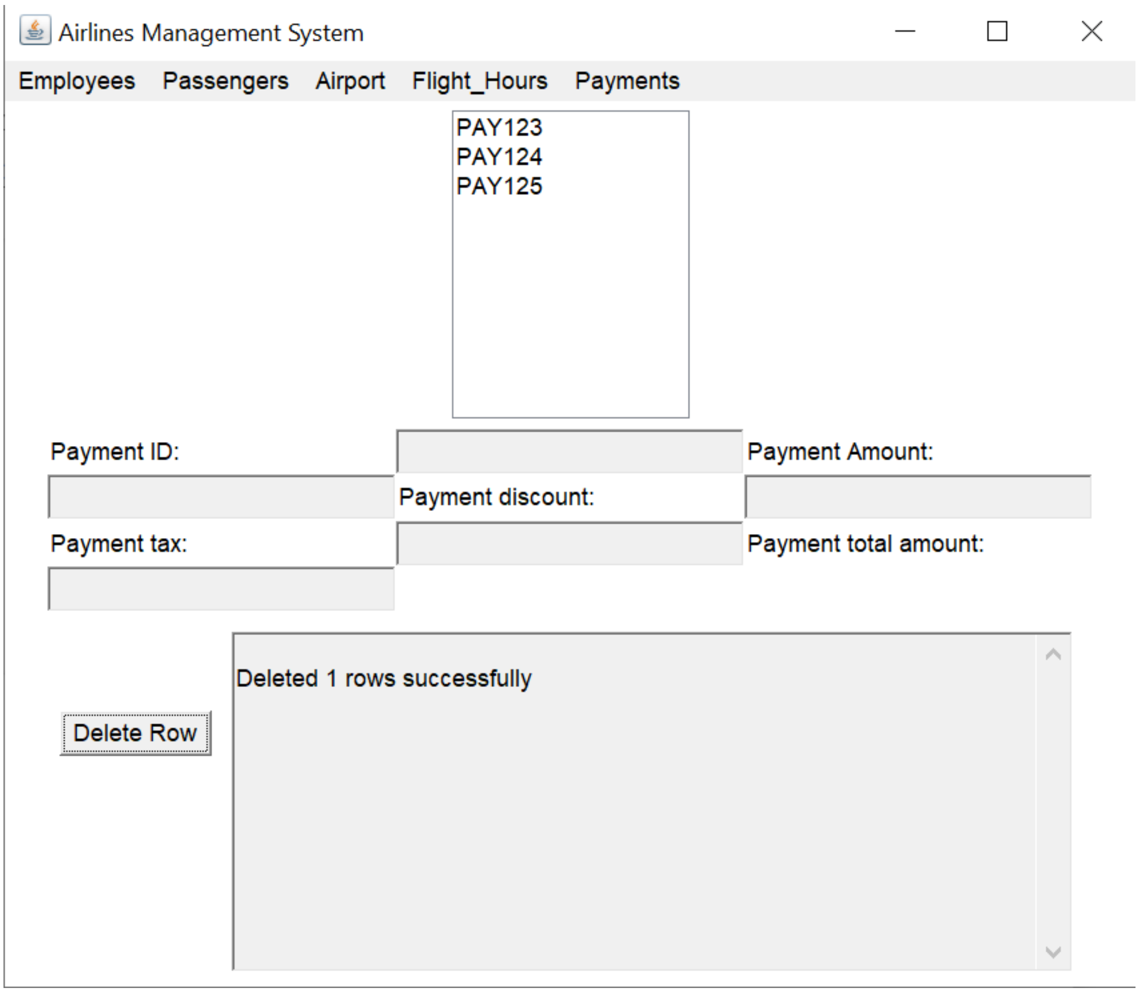
Payment ID:	PAY124	Payment Amount:
40000	Payment discount:	20
Payment tax:	90	Payment total amount:
32090		

```
Updated 1 rows successfully
```



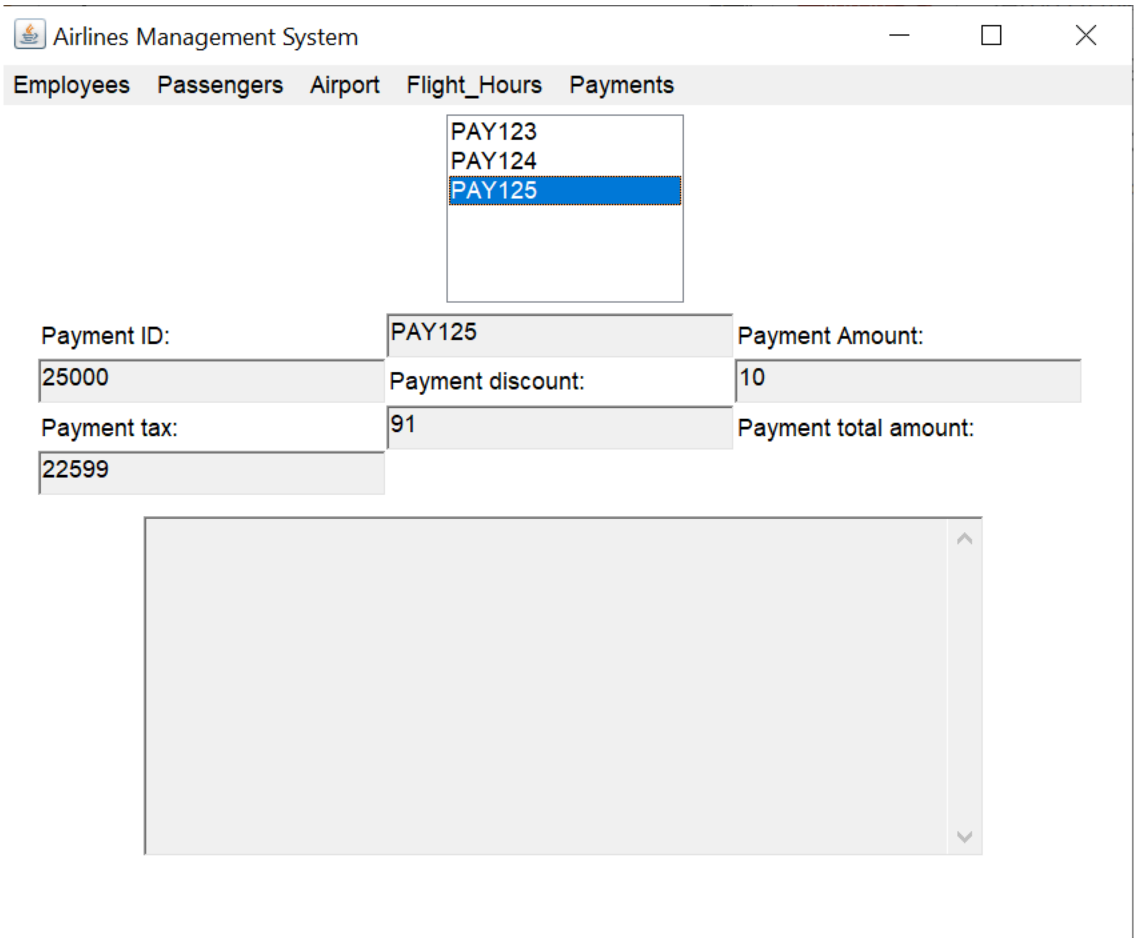
PAY123
PAY124
PAY125
PAY130

Payment ID: PAY130
Payment Amount: 50000
Payment discount: 10
Payment tax: 90
Payment total amount: 45090



```
PAY123
PAY124
PAY125
```

```
Deleted 1 rows successfully
```



PAY123
PAY124
PAY125

Payment ID:	PAY125	Payment Amount:	
25000	Payment discount:	10	
Payment tax:	91	Payment total amount:	
22599			

REFERENCES:

1. Abraham Silberschatz, Henry F Korth, S. Sudarshan, Database System Concepts, 6th Edition,

McGraw-Hill International Edition, 2010.

2. https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT001