Kuwait University

College Engineering and Petroleum
Computer Engineering Department

CpE 0612341: DATABASE MANAGEMENT SYSTEMS

Semester: Spring 2021/2022

Section No. 01AN/ATTEND

Project Phase 3

Student Name: <Hassan Hussain Ali>

Student Id: <2171115976>

Student Name: <Yassine Nader Serrid>

Student Id: <2181156439>

Student Name: <Yousif Aldhafeeri>

Student Id: <2151117875>

Instructor Name: <Dr. MOHAMMAD JERAGH>

TA Name: <Eng. Anwar El-Ebrahim>

Date: <21/5/2022>

Contents

Table Of Figures:

Tables:

1. Project Description and Requirements

This system is responsible for dealing with airport management. The airport management system primarily deals with employees who operate the airport services and employees who modify flights and tickets. They are responsible for all the management services in the airport, and the second type of users are the passengers. This system helps the airports to be efficient and well organized. This system only deals with the inside work of the airport; it does not deal with outside work like booking flight tickets from websites, flight agencies, or hotels. We chose SQL to design the system because it is efficient at processing queries and joining data across tables, making the complex queries simpler compared to structured data. NoSQL databases lack consistency across products and require more work to query data, especially when the query complexity increases.

Each major city should be identified by the city name as a primary key and the country of that city. Each major city has an airport that is recognized by its name, country, and IATA designator as shown below in table 2. Airports contain airlines and each airline has its own airline ID, name, and an office with a unique 3 digits ID in the airport.

All international airlines operating in different countries around the world have offices in all major cities. Therefore, there are many airline offices at the airport. Airline companies serve flights. Each airline has its own unique code and identifier which is provided by the International Air Transport Association (IATA). A two-letter airline designator is known as an airline code. A three-digit code for the airline is also displayed on the ticket.

The flight serves passengers. Flight carries passengers from the source to the desired destination. Passengers are identifiable by their passenger id and passport number. So, each passenger has details such as name, address, age, gender, and phone.

Each airport has employees, and each employee is identified by EmployeeID, first and last name, job type, age, gender, phone number, and salary. Employees may assist passengers with a variety of services, including reserving a flight ticket, discussing concerns, and so on. At the airport, there is a diversity of available jobs. For the sake of simplicity, our system only considers a few occupations.

Table 1. IATA Designator of Airlines.

Airline Name	IATA Designator (Code) or Airline ID	3-DIGIT CODE
Kuwait Airways	KU	229
SOLINAIR	SP	265
Tunisair	TU	199
American Airlines	AA	001
Turkish Airlines	TK	235

Table 2. IATA Designator of Airports.

Airport Name	IATA Designator (Code)
Kuwait International airport	KWI
Abha International airport	AHB
Doha Airport	OTBD
Abu Dhabi Airport	AUH
Adam Airport	AOM

For a passenger to travel by flight they should provide all their personal information in detail such as the passenger's phone number, first name, last name, age, passport number, and gender. All this information is required to book a flight ticket. Passengers need a ticket; a ticket is used to confirm that the passenger has reserved a seat on a flight. With the ticket, a passenger is allowed to board the flight. An air ticket has information such as the passenger's first and last name, the flight code, date of travel, source, destination, passenger ID, seat number, arrival, and departure time. A flight code is a unique identifier for each flight. A flight code is made up of a four-digit code and an airline code. In this system, all tickets have the same class.

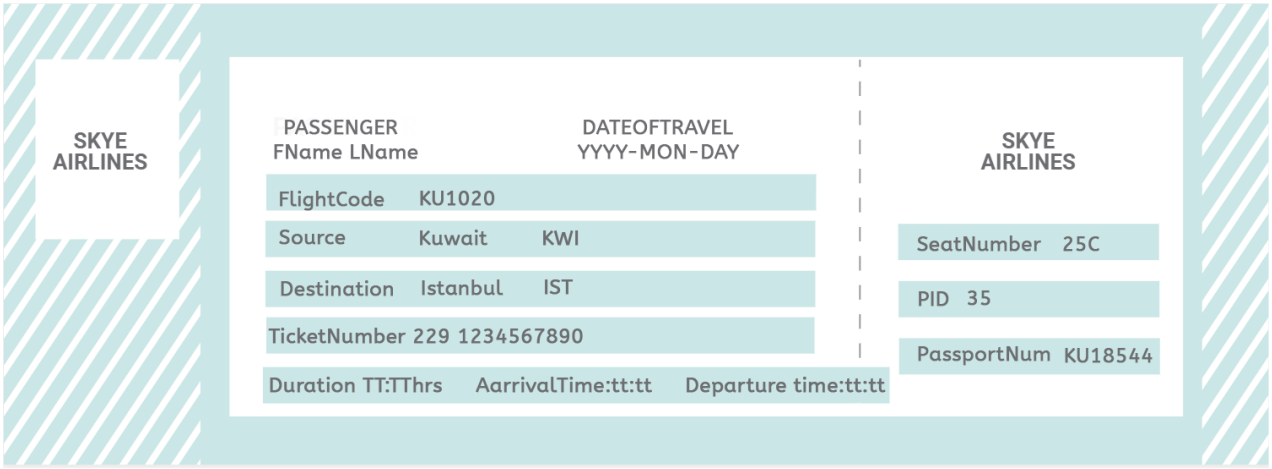


Figure 1. Sample Of a Ticket According to The System Design.

2. Functionalities:

The system will offer a list of functions for both employees and passengers with different interfaces, which will be implemented using Python and then connected to the SQLITE3 database. And all functionalities are linked to the database, so any work done in the interface will be updated automatically in the database. Both interfaces are connected to the same database. For example, any changes made to the employee interface will be updated in the passenger interface. Each type will have an individual interface. We can use parallel processing in Python to run both interfaces.

2.1 Designing The UI:

To design the UI, we used PAGE, which is a cross-platform drag-and-drop GUI generator. It allows us to easily create Python GUI windows containing a selection of Tk and ttk widgets. And after designing each interface individually, we combined all of them into two python code files, one for the employee interface and the other for the passenger interface. We used the Tkinter library in Python to do this task.

2.2 Connecting The Interfaces With The Database:

We used the powerful library, which is sqlite3 library that allows us to execute queries and apply CRUD (Create-Read-Update-Delete) concepts in Python. To link between our interface and the database properly and keep the interface linked and updated frequently for any changes, we used this library to follow the requirements of our design.

2.3 Employee:

For the first type of users, which are the Employees, each employee has an employee id and password which is used to login into the system through the login interface, after the employee logs in to the system it will show him his/her first name, last name Employee ID and salary and then the employee can choose from the admin menu either to book or cancel a passenger ticket or edit passenger information as shown below in Figure 2 and 3.

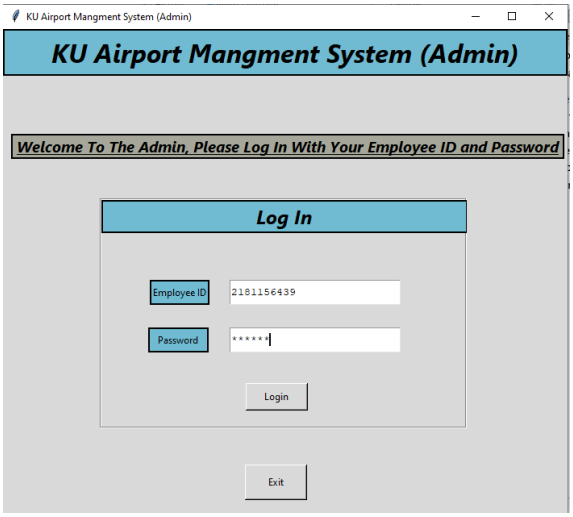


Figure 2. Login Interface For Employees.

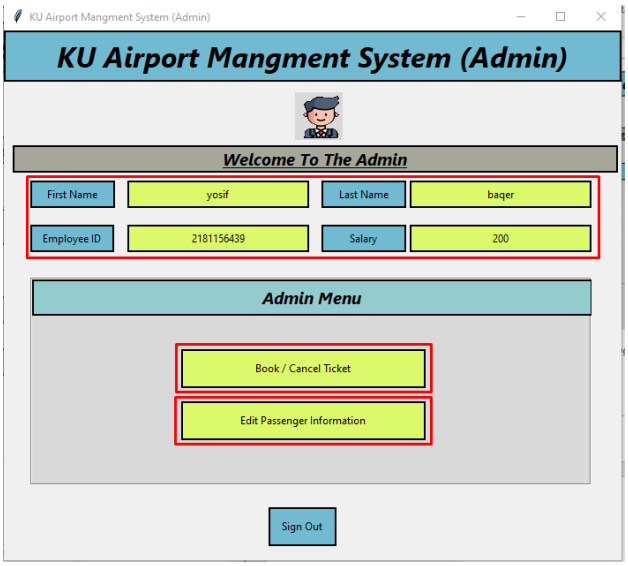


Figure 3. Employee Main Menu.

2.3.1 Book Or Cancel Tickets:

If the employee chooses to book or cancel a ticket from the admin menu, he should enter the passport number for a specific passenger. After entering the passport number for this passenger, the system will display for the employee the passenger's info such as first name, last name, passport number, and age as shown below in Figure 4.

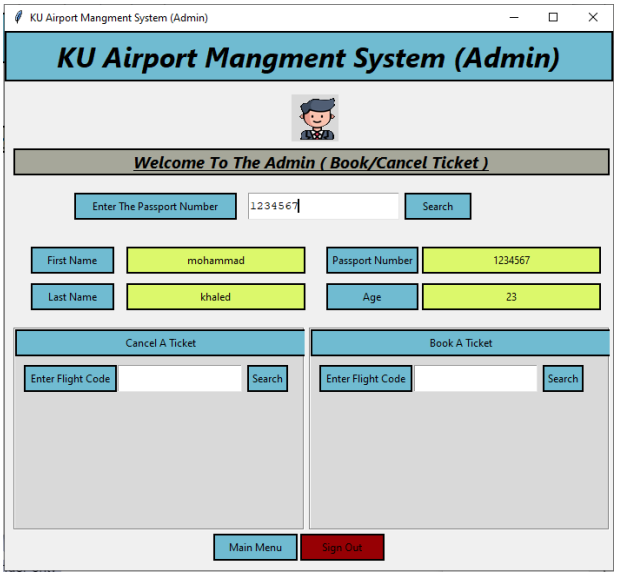


Figure 4. Book/Cancel Ticket.

The employee can book a ticket for this passenger by entering the flight code of the desired flight that the passenger wants to book on it. After entering the flight code, it will show him three scenarios. The first scenario appears when the flight exists, and the employee can book this ticket. After booking the ticket successfully, the system will display that this booking process is done successfully as shown in Figure 5.

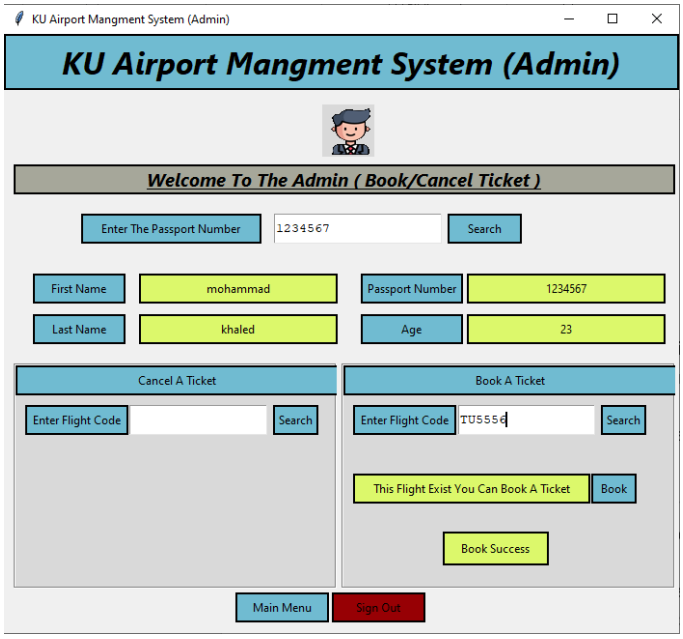


Figure 5. First Scenario: Booking A Ticket For Specific Passenger On Existing Flight

In the second scenario, the system will show the employee that this flight does not exist if the passenger already has a flight ticket on this trip, and the system will display for the employee that the passenger already has this flight ticket as shown in Figure 6.

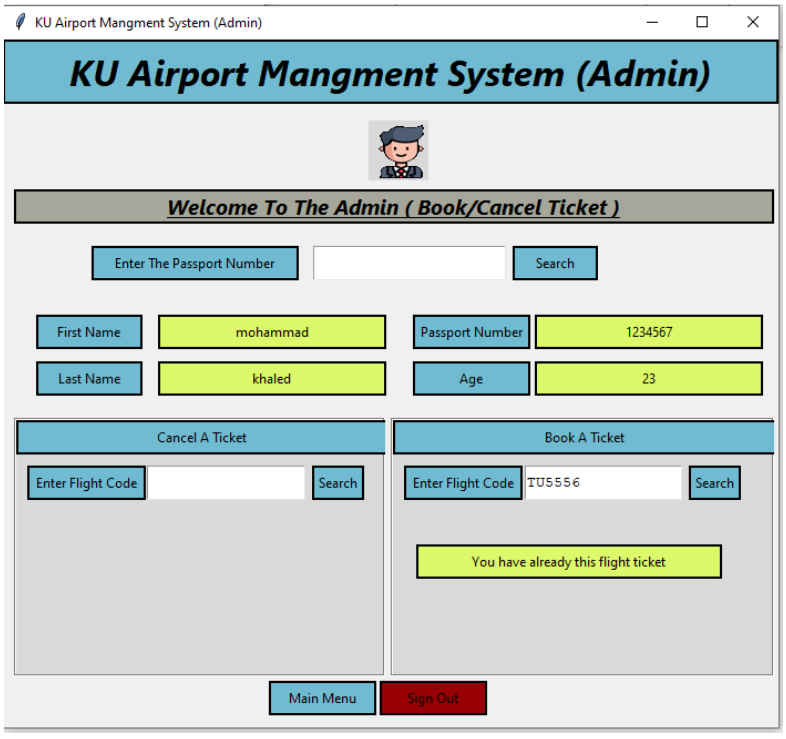


Figure 6. Second Scenario: Booking A Ticket For Specific Passenger On (Condition: The Passenger Already booked it).

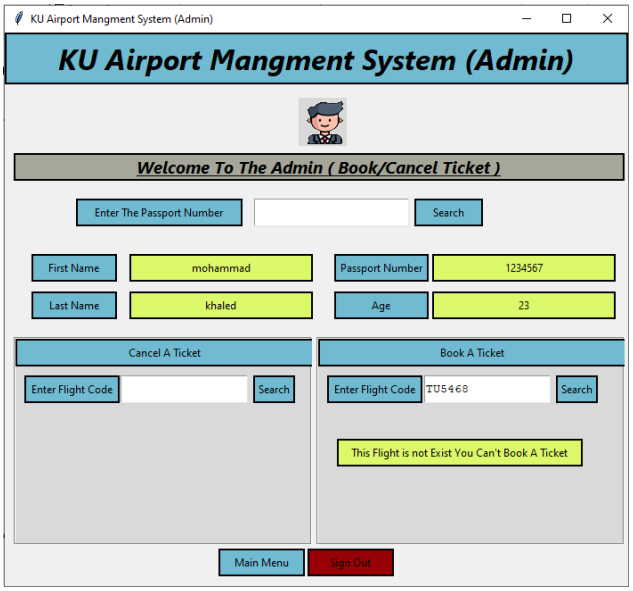


Figure 7. Third Scenario: Booking A Ticket For Specific Passenger and the flight doesn't exist in the DB.

The employee can cancel a ticket for this passenger by entering the flight code of the desired flight that the passenger wants to cancel. After entering the flight code, it will show him two scenarios. The first scenario appears if the passenger already has a ticket on this flight. The employee can cancel the passenger's ticket. After successfully canceling the ticket, the system will display that this canceling process has been completed successfully as shown in Figure 8.

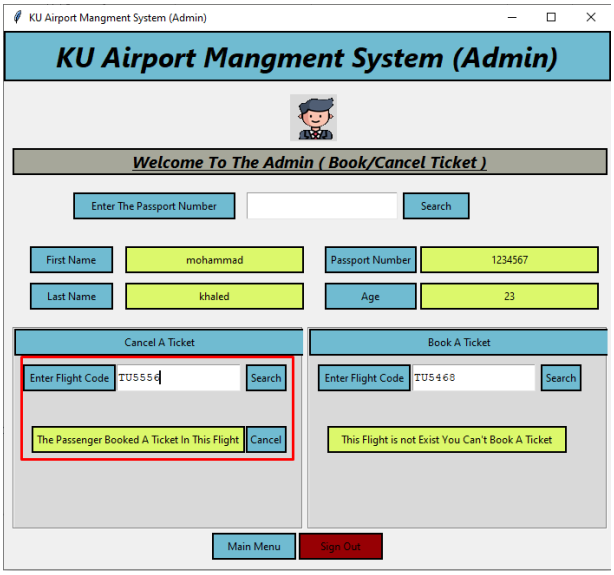


Figure 8. First Scenario: Cancelling A Ticket For Specific Passenger On Existing Flight.

In the second scenario, when a passenger doesn't have a ticket on this flight, the system displays that the passenger doesn't have a ticket on this trip. From the book or cancel interface, the employee has two choices: either to sign out or to go to the main menu interface as show in Figure 9 and 10.

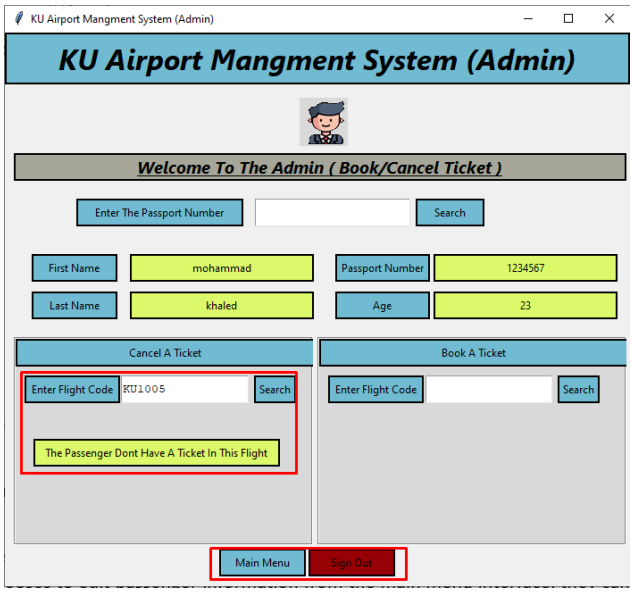


Figure 9. Second Scenario: Cancelling A Ticket For Specific Passenger (The Passenger Don't Have A Ticket In This This Flight).

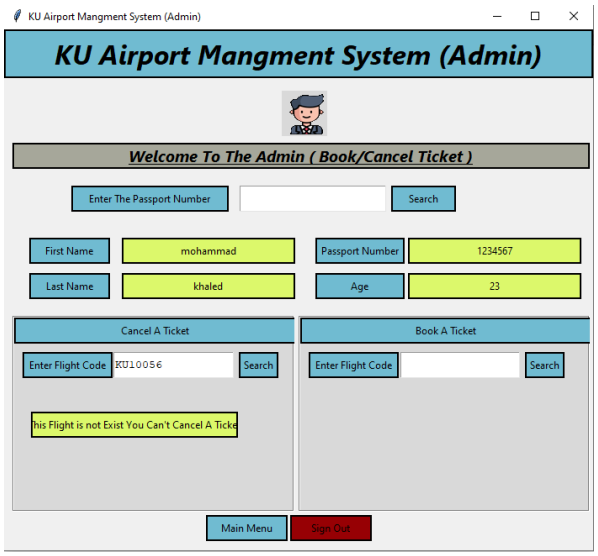


Figure 10. Third Scenario: Cancelling A Ticket For Specific Passenger and the flight doesn't exist in the DB.

2.3.2 Edit Passenger Information:

If the employee chooses to edit passenger information from the main menu interface, they can also edit the information for a particular passenger by entering the passenger's passport number. After entering the passport number for this passenger, the system will display for the employee the passenger's info such as first name, last name, passport number, age, phone number, and his/her age. In the bottom frame, some entries allow the employee to edit the passenger information. The employee can edit just one entry, multiple entries, or all the information. To update the passenger's information in the database, the employee should press the "Update" button. From the edit passenger information interface, the employee has two choices: either to sign out or to go to the main menu interface as shown below in Figure 11 and 12.

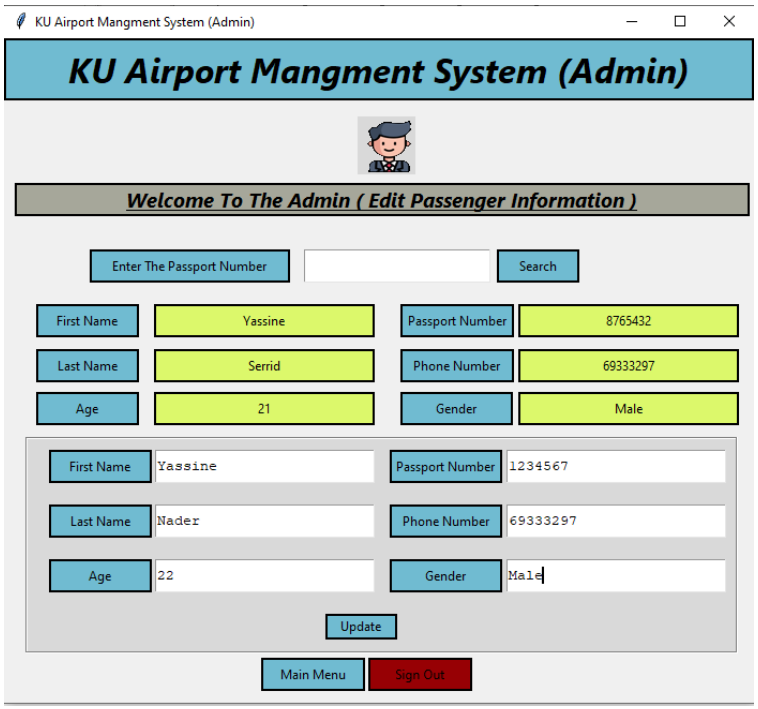


Figure 11. Passenger Info Before Updating.

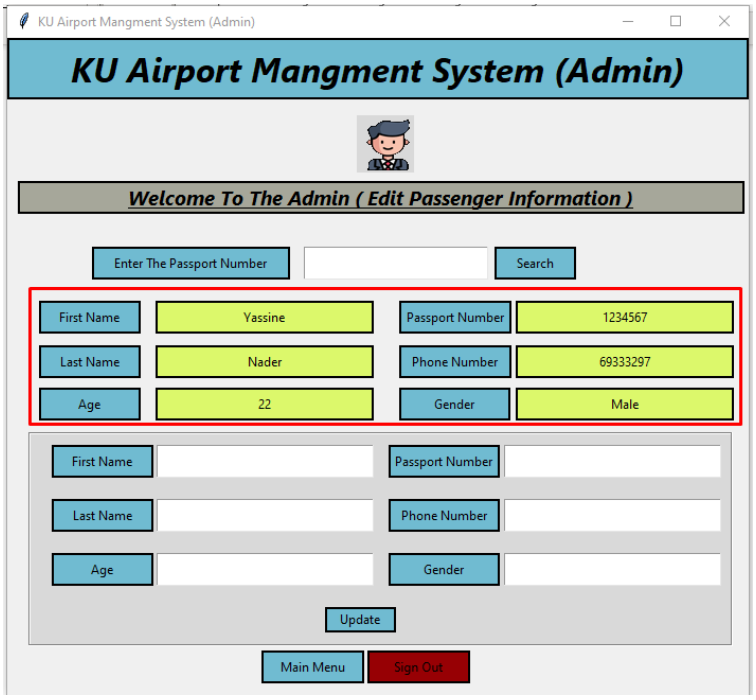


Figure 12. Passenger Info After Updating.

2.4 Passenger:

For the second type of user, which is the passenger, the first interface gives the passenger the ability to show all flights or show his flight (the next flight) as shown in Figure 13.

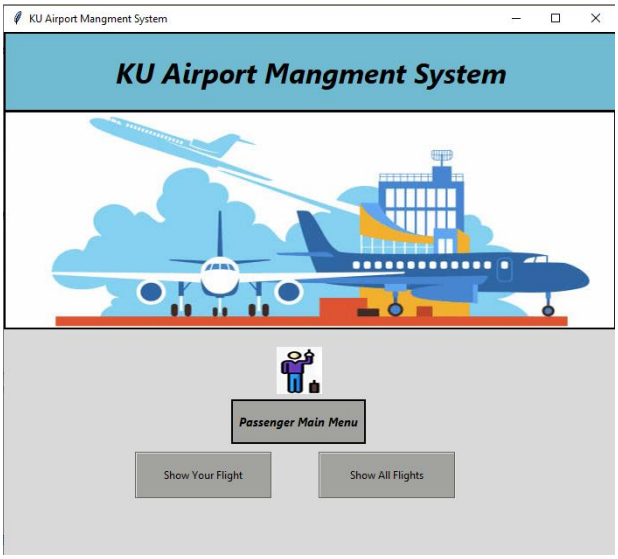


Figure 13. Main Menu Interface For Passengers.

2.4.1 Show All Flights:

Passengers can use the system to see all the flight information, such as source, destination, flight code, arrival time, and departure time. They can only see the next 5 flights as show in Figure 14.

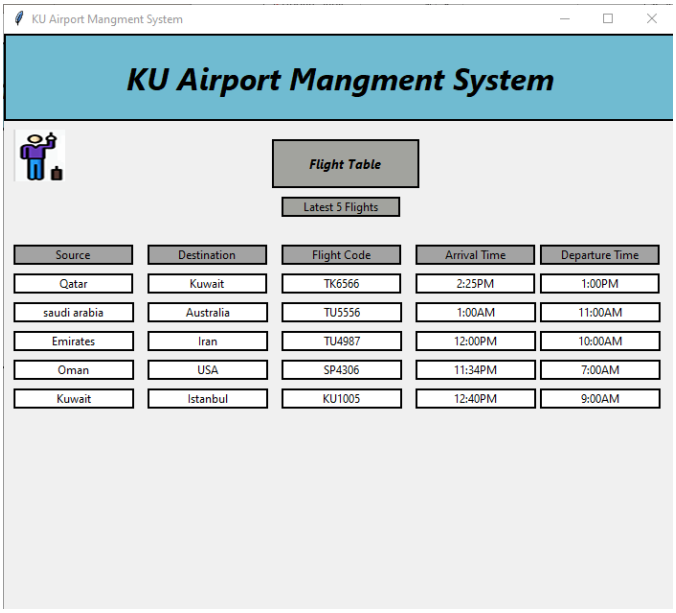


Figure 14. Flight Table For Passengers.

2.4.2 Show Your Flight:

The passenger can see his next flight by entering his passport number. After entering his passport number, the system will show the passenger his/her first name, last name, gender, and phone number. And below his information, passengers can also see their latest flight and its information such as source, destination, flight code, arrival time, and departure time. In this condition the system displays the Ticket status as booked and the booking date as shown below in Figure 15.

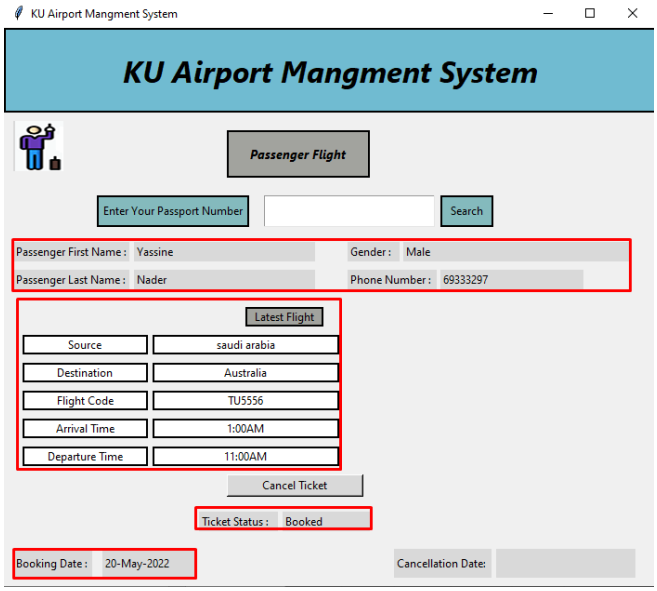


Figure 15. Passenger And Flight Details.

The passengers could cancel their tickets. If the passenger presses the cancel button, the system will display a message that this ticket has been canceled successfully, and it will also display the cancellation date as shown in Figure 16.

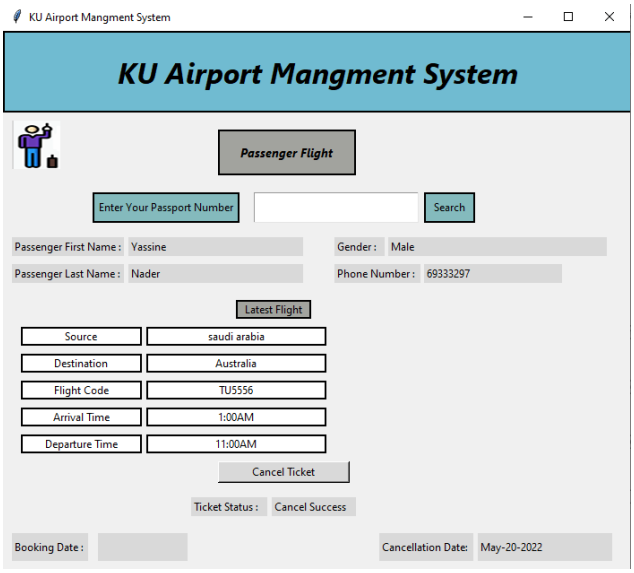


Figure 16. Cancelling Ticket From Passenger's Interface.

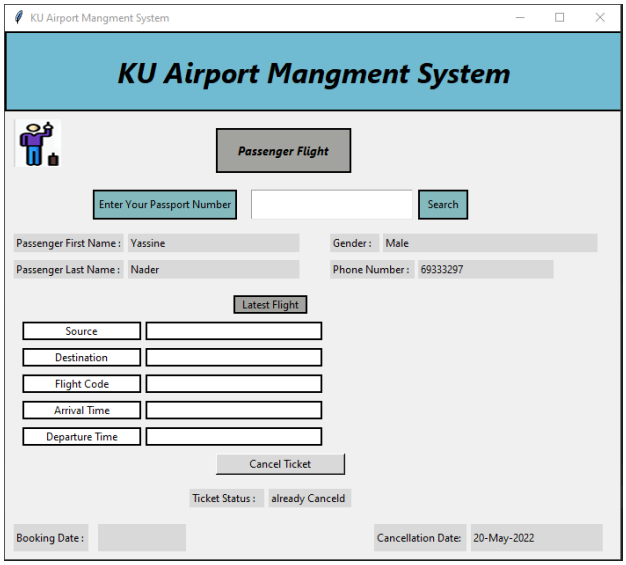


Figure 17. Cancelling Ticket From Passenger's Interface.

If the passenger doesn't have a flight ticket, the system will display a message that no flights have been found. The last scenario is when the passenger only wants to see the details of his flight. The system will display the ticket status as confirmed. The booking date is the day on which he purchases an airline ticket. A passenger can also cancel one or several tickets. The day he cancels an airline ticket is known as the cancellation date as shown above in Figure 17.

3. Entities, ER Diagram and Relational Schema

3.1 Entities

City

CityName	Country

Airport

AirportName	Country	CityName

Airline

AirlineID	OfficeID	ThreeDigitCode	AirlineName

Have

AirportName	AirlineID

Flight

FlightCode	Source	Destination	Duration	DepartureT	ArrivelTime	AirlineID	EmployeeID

Ticket

TicketNumber	DateOfTravel	ArrivalTime	DepartureT	PID	SeatNumber	Source	Destination	EmployeeID

Passenger

PID	Sex	Lname	Fname	Age	PassportNumber	PPhone	FlightCode

Employee

EmployeeID	Sex	Lname	Fname	JobType	Salary	Age	EPhone	AirPortName	Password

Serve

EmployeeID	PID

Books

DateOfBooking	TicketNumber	PID

Cancels

DateOfCancelletion	TicketNumber	PID

Figure 18. Entities Of DB System.

3.2 Relational Schema

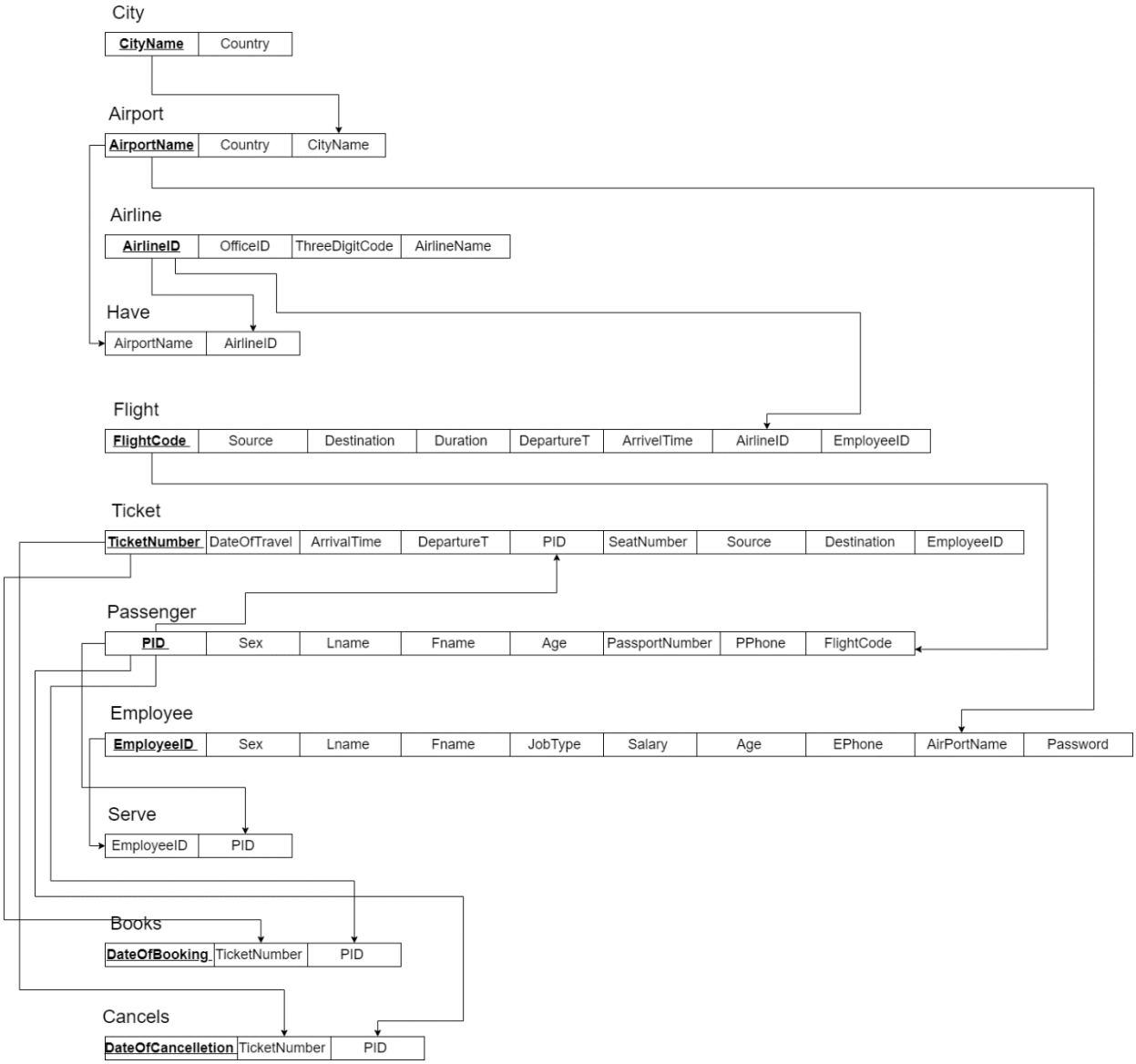


Figure 19. Relational Schema of DB System.

3.3 ER Diagram

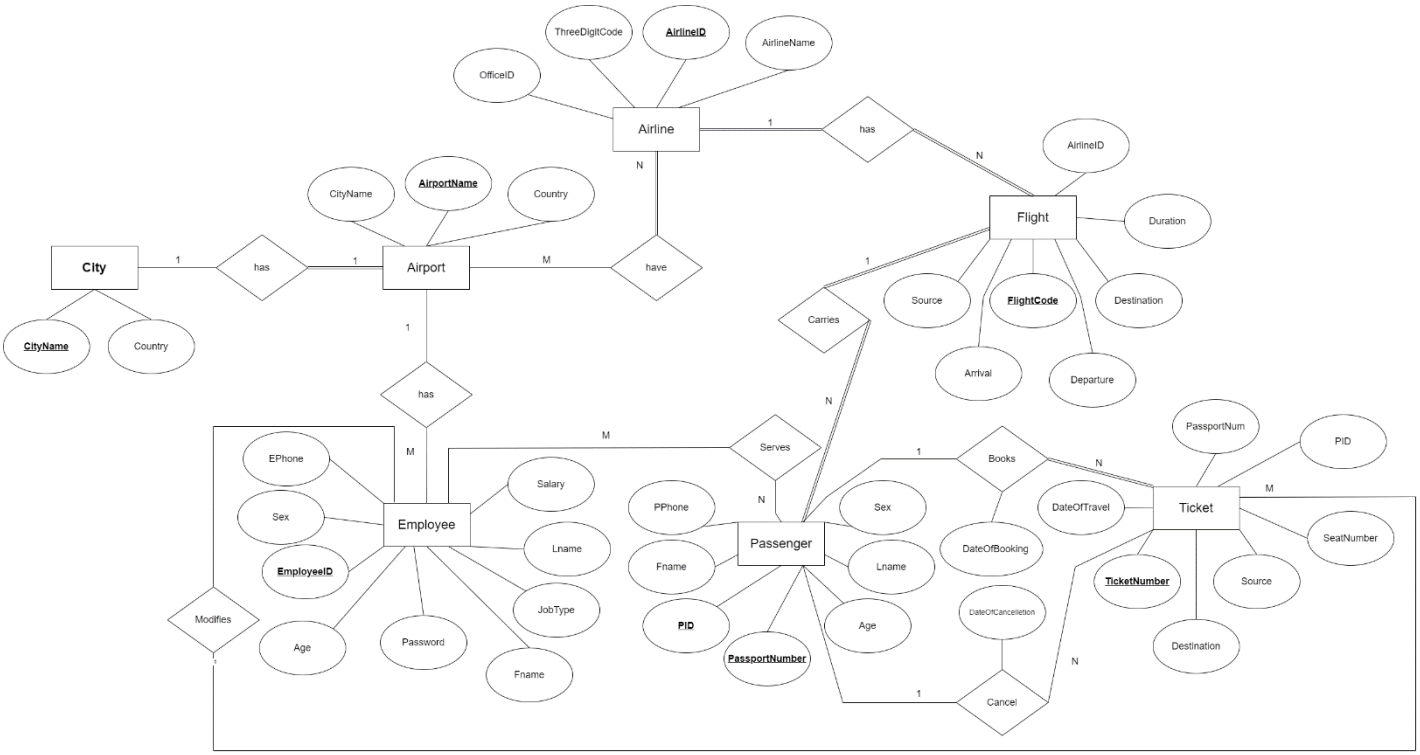


Figure 20. ER Diagram of DB System.

3.3.1 ER/EER Diagram Relationships:

ER diagram contains the following relationships:

Table 3. ER Diagram Relationships.

Entity 1	Name of the Relationship	Entity 2	Cardinality
City	Has	Airport	1:1
Airport	Have	Airline	M: N
Airline	Has	Flight	1: N
Flight	Carries	Passenger	1: N
Passenger	Books	Ticket	1: N
Employee	Serve	Passenger	M: N
Employee	Modify	Ticket	1: N
Passenger	Cancel	Ticket	1: N

Table 4. Types Of the Relationships.

Type of binary relationship	Relationships in the system
One-to-One	1- One city has only one airport.
One-to-Many	1- One Airline has Multiple flights (Many flights belong to the same airlines). 2- One Flight can carry many passengers. 3- One passenger can book one or more tickets. 4- One Employee can modify many tickets. 5- One passenger can cancel one or more tickets.
Many-to-Many	5- Many Employees could serve many passengers.

4. Constraints

Table 5. Constraints Of DB System.

City	1-CityName (Primary Key), (VARCHAR2(50)), NOT NULL. 2-Country (VARCHAR2(50)).
Airport	1-CityName (FOREIGN KEY FROM CITY TABLE), VARCHAR2(50) NOT NULL. 2-AirportName (Primary Key), VARCHAR2(50), NOT NULL. 3-Country (VARCHAR2(50)).
Airline	1- OfficeID (UNIQUE), (Number), (Len == 3), NOT NULL. 2- ThreeDigitCode (UNIQUE), (NUMBER), (LEN == 4), NOT NULL. 3- AirlineID (Primary key), (VARCHAR2(50)), (LEN == 4), NOT NULL. 4-AirlineName (UNIQUE), VARCHAR2(50).
Have	1-AirportName (FOREIGN KEY FROM AIRPORT TABLE), VARCHAR2(50), NOT NULL. 2-AirlineID (FOREIGN KEY FROM AIRLINE TABLE), (VARCHAR2(50)), (LEN == 4), NOT NULL.
Flight	1-Source VARCHAR2(50), NOT NULL. 2-FlightCode Primary Key, VARCHAR2(6), (CHECK LEN == 6), NOT NULL. 3-Destination VARCHA R2(50), NOT NULL. 4-Duration NUMBER. 5-DepartureT VARCHAR2(50), NOT NULL. 6-ArrivelTime VARCHAR2(50), NOT NULL. 7-AirlineID (FOREIGN KEY FROM AIRLINE TABLE), (VARCHAR2(50)), (LEN == 4), NOT NULL. 8- EmployeeID (Number), (Len == 10), NOT NULL.
Ticket	1-DateOfTravel TO_DATE == (‘YYYY-MON-DD’), NOT NULL. 2-ArrivalTime VARCHAR2(50), NOT NULL. 3-DepartureT VARCHAR2(50), NOT NULL 4-PID NUMBER (FOREIGN KEY FROM PASSENGER TABLE), NUMBER (LEN == 10). 5-SeatNumber (UNIQUE), VARCHAR2(3). 6-Source VARCHAR2(50). 7-Destination VARCHAR2(50). 8-TicketNumber (Primary Key), VARCHAR2(13). 9-EmployeeID (Number), (Len == 10), NOT NULL.
Passenger	1-Sex VARCHAR2(20). 2-Lname VARCHAR2(50), NOT NULL. 3-Age Number, CHECK (AGE > 1 AND AGE < 150). 4-PassportNumber VARCHAR2(7) 5-PID Primary Key, NUMBER, (LEN == 10) NOT NULL. 6-Fname VARCHAR2(50), NOT NULL. 7-PPhone NUMBER, (CHECK LEN(PPhone) = 8). 8-FlightCode (FOREIGN KEY FROM FLIGHT TABLE), VARCHAR2(6), NOT NULL.
Employee	1-Salary NUMBER. 2-Lname VARCHAR2(50), NOT NULL. 3-JobType VARCHAR2(50), NOT NULL. 4-Fname VARCHAR2(50), NOT NULL. 5-Age Number, CHECK (AGE > 1 AND AGE < 150). 6-EmployeeID (Primary Key), (Number), (Len == 10), NOT NULL.

	7-Sex VARCHAR2(20). 8-EPhone NUMBER, (CHECK LEN(PPhone) = 8). 9-AirPortName (FOREIGN FROM AIRPORT TABLE), VARCHAR2(50), NOT NULL. 10-Password VARCHAR(15) NOT NULL.
Serves	1-EmployeeID (FOREIGN KEY FROM EMPLOYEE TABLE), (NUMBER) (NOT NULL). 2-PID (FOREIGN KEY FROM PASSENGER TABLE), NUMBER NOT NULL.
Books	1-DateOfBooking ('YYYY-MON-DD'), PRIMARY KEY. 2-TicketNumber (FOREIGN KEY FROM TICKET TABLE), VARCHAR2(13). 3-PID (FOREIGN KEY FROM PASSENGER TABLE), (NUMBER), (NOT NULL).
Cancels	1-DateOfCancelletion ('YYYY-MON-DD'), PRIMARY KEY. 2-TicketNumber (FOREIGN KEY FROM TICKET TABLE), VARCHAR2(13). 3-PID (FOREIGN KEY FROM PASSENGER TABLE), (NUMBER), (NOT NULL).

5. Database Tables

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	OFFICEID	NUMBER	No	(null)	1	(null)
2	THREEDIGITCODE	NUMBER	No	(null)	2	(null)
3	AIRLINEID	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	AIRLINENAME	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)

Figure 21. Airline Table.

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Depende

Sort.. | Filter:

	OFFICEID	THREEDIGITCODE	AIRLINEID	AIRLINENAME
1	1234	229	KU	Kuwait Airways
2	5643	265	SP	SOLINAIR
3	5896	199	TU	Tunisair
4	5986	235	TK	Turkish Airlines

Figure 22. Airline Table with data

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Index

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	AIRPORTNAME	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	COUNTARY	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	CITYNAME	VARCHAR2(50 BYTE)	No	(null)	3	(null)

Figure 23. Airport Table.

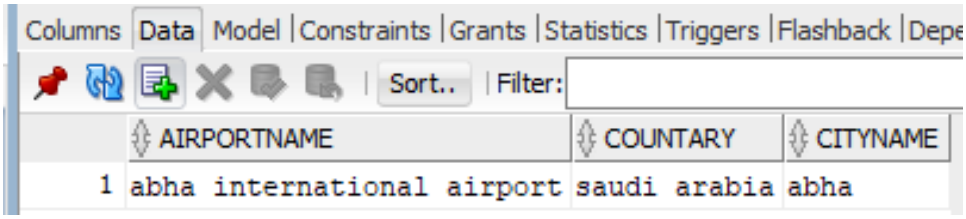


Figure 24. Airport Table with data

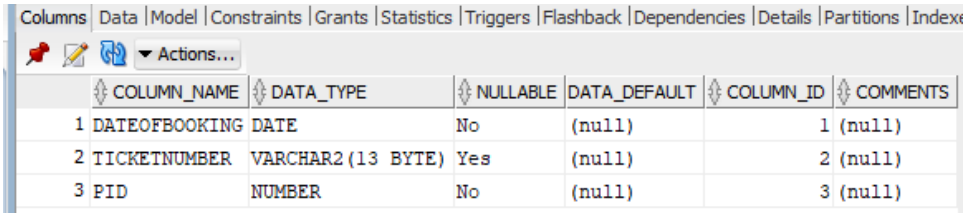


Figure 25. Book A Ticket Table.

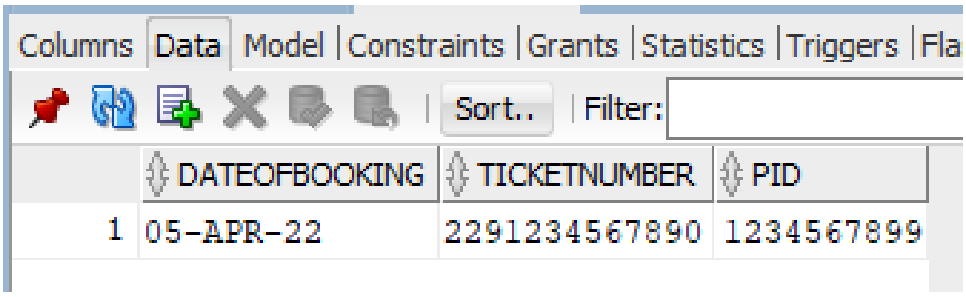


Figure 26. Book A Ticket Table with data

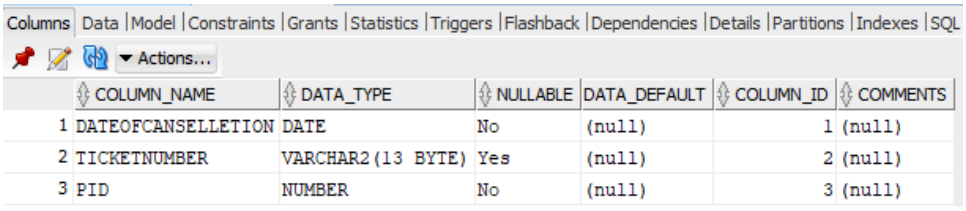


Figure 27. Cancel A Ticket Table.

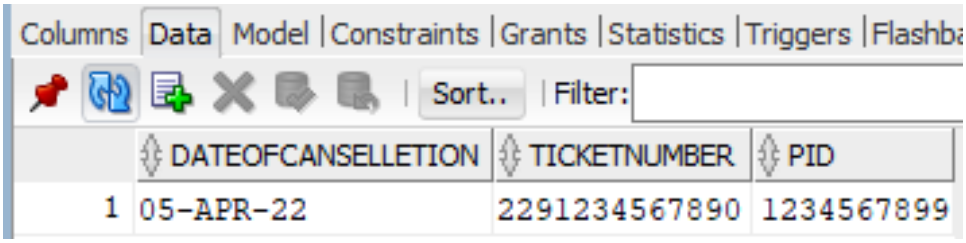


Figure 28. Cancel A Ticket Table with data

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CITYNAME	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	COUNTRY	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)

Figure 29. City Table.

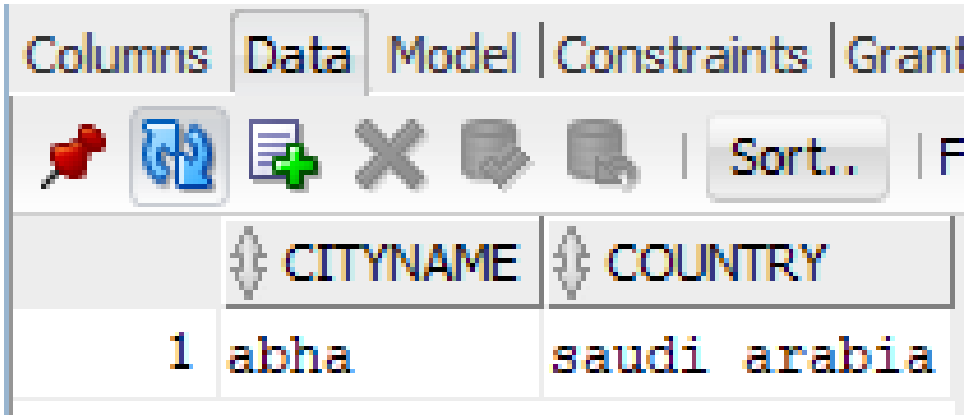


Figure 30. City Table with data

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexe

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SALARY	NUMBER	Yes	(null)	1	(null)
2	LNAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	JOBTYPE	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	FNAME	VARCHAR2(50 BYTE)	No	(null)	4	(null)
5	AGE	DATE	Yes	(null)	5	(null)
6	AIRPORTNAME	VARCHAR2(50 BYTE)	No	(null)	6	(null)
7	EMPLOYEEID	NUMBER	No	(null)	7	(null)
8	SEX	VARCHAR2(20 BYTE)	Yes	(null)	8	(null)
9	EPHONE	NUMBER	Yes	(null)	9	(null)
10	PASSWORD	VARCHAR2(15 BYTE)	No	(null)	10	(null)

Figure 31. Employee Table.

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	S...	LNAME	JOBTYPE	FNAME	AGE	AIRPORTNAME	EMPLOYEEID	SEX	EPHONE	PASSWORD
1	5000	Ahmad	manager	hassan	05-APR-89	abha international airport	2111585893	Male	55446688	123456
2	3000	abdullah	engineer	yaseen	05-AUG-98	abha international airport	5643218795	Male	48956325	123456
3	200	baqer	worker	yosif	07-APR-78	abha international airport	2181156439	Male	69854723	123456

Figure 32. Employee Table with data

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexe

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SOURCE	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	FLIGHTCODE	VARCHAR2(6 BYTE)	No	(null)	2	(null)
3	DESTINATION	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	DURATION	VARCHAR2(50 BYTE)	Yes	(null)	4	(null)
5	DEPARTURET	VARCHAR2(50 BYTE)	No	(null)	5	(null)
6	ARRIVELTIME	VARCHAR2(50 BYTE)	No	(null)	6	(null)
7	AIRLINEID	VARCHAR2(50 BYTE)	No	(null)	7	(null)
8	EMPLOYEEID	NUMBER	No	(null)	8	(null)

Figure 33. Flight Table.

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	SOURCE	FLIGHTCODE	DESTINATION	DURATION	DEPARTURET	ARRIVELTIME	AIRLINEID	EMPLOYEEID
1	Kuwait	KU1005	Istanbul	3:40	9:00AM	12:40PM	KU	2111585893
2	Oman	SP4306	USA	16:34	7:00AM	11:34PM	SP	2111585893
3	Emirates	TU4987	Iran	2:00	10:00AM	12:00PM	TU	2111585893
4	saudi arabia	TU5556	Australia	14:14	11:00AM	1:00AM	TU	2111585893
5	Qatar	TK6566	Kuwait	1:25	1:00PM	2:25PM	TK	2111585893

Figure 34. Flight Table with data

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexe

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	AIRPORTNAME	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	AIRLINEID	VARCHAR2(50 BYTE)	No	(null)	2	(null)

Figure 35. Airport Have Table.

Columns | Data | Model | Constraints | Grants | Statistics | Trigger

Sort.. | Filter:

	AIRPORTNAME	AIRLINEID
1	abha international airport	KU

Figure 36. Airport Have Table with data

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SEX	VARCHAR2(20 BYTE)	Yes	(null)	1	(null)
2	LNAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	FNAME	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	AGE	DATE	Yes	(null)	4	(null)
5	PID	NUMBER	No	(null)	5	(null)
6	PASSPORTNUMBER	VARCHAR2(7 BYTE)	No	(null)	6	(null)
7	PPHONE	NUMBER	Yes	(null)	7	(null)
8	FLIGHTCODE	VARCHAR2(6 BYTE)	No	(null)	8	(null)

Figure 37. Passenger Table.

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	SEX	LNAME	FNAME	AGE	PID	PASSPORTNUMBER	PPHONE	FLIGHTCODE
1	Male	khaled	mohammad	05-APR-99	1234567899	1234567	69676161	KU1005
2	Female	Mnar	Abdullah	06-NOV-45	9966553384	5643218	69583246	SP4306
3	Male	khaled	Ahmad	05-APR-99	4466330022	5632984	69532016	TU4987
4	Male	Ali	Abbas	05-APR-99	8796425349	5698331	69852314	TU5556
5	Male	Abdulgader	Jassem	05-APR-99	8796528430	7744653	98651423	TK6566

Figure 38. Passenger Table with data

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitio

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMPLOYEEID	NUMBER	No	(null)	1	(null)
2	PID	NUMBER	No	(null)	2	(null)

Figure 39. Serves Table.

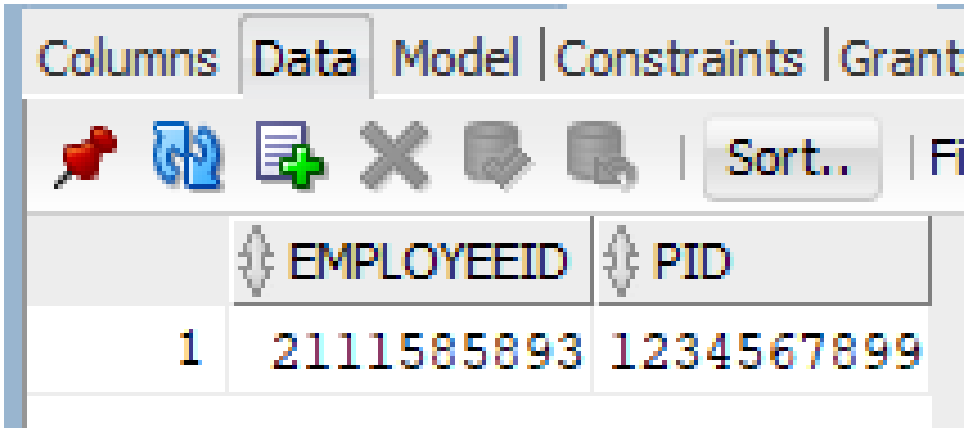
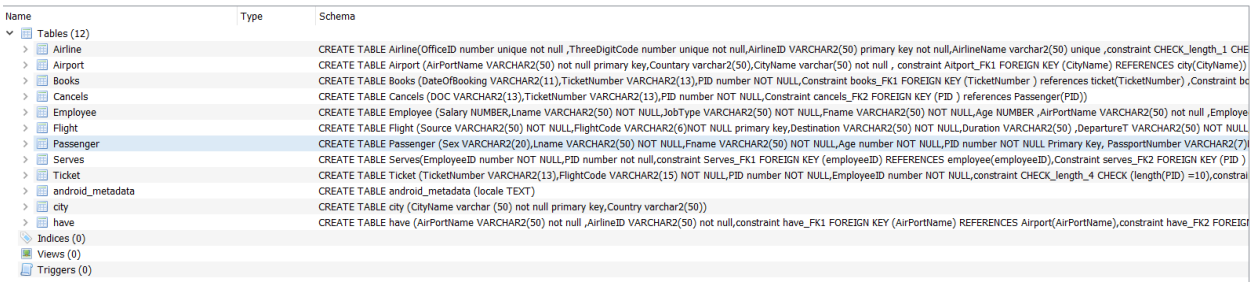


Figure 40. Serves Table with data



Name	Type	Schema
Tables (12)		
Airline		CREATE TABLE Airline(OfficeID number unique not null ,ThreeDigitCode number unique not null,AirlineID VARCHAR2(50) primary key not null,AirlineName varchar2(50) unique ,constraint CHECK_length_1 CHE
Airport		CREATE TABLE Airport (AirPortName VARCHAR2(50) not null primary key,Countary varchar2(50),CityName varchar(50) not null , constraint Aitport_FK1 FOREIGN KEY (CityName) REFERENCES city(CityName))
Books		CREATE TABLE Books (DateOfBooking VARCHAR2(11),TicketNumber VARCHAR2(13),PID number NOT NULL,Constraint books_FK1 FOREIGN KEY (TicketNumber) references ticket(TicketNumber) ,Constraint bo
Cancels		CREATE TABLE Cancels (DOC VARCHAR2(13),TicketNumber VARCHAR2(13),PID number NOT NULL,Constraint cancels_FK2 FOREIGN KEY (PID) references Passenger(PID))
Employee		CREATE TABLE Employee (Salary NUMBER,Lname VARCHAR2(50) NOT NULL,JobType VARCHAR2(50) NOT NULL,Fname VARCHAR2(50) NOT NULL,Age NUMBER ,AirPortName VARCHAR2(50) not null ,Employe
Flight		CREATE TABLE Flight (Source VARCHAR2(50) NOT NULL,FlightCode VARCHAR2(6)NOT NULL primary key,Destination VARCHAR2(50) NOT NULL,Duration VARCHAR2(50) ,DepartureT VARCHAR2(50) NOT NULL
Passenger		CREATE TABLE Passenger (Sex VARCHAR2(20),Lname VARCHAR2(50) NOT NULL,Fname VARCHAR2(50) NOT NULL,Age number NOT NULL,PID number NOT NULL Primary Key, PassportNumber VARCHAR2(7)(
Serves		CREATE TABLE Serves(EmployeeID number NOT NULL,PID number not null,constraint Serves_FK1 FOREIGN KEY (employeeID) REFERENCES employee(employeeID),Constraint serves_FK2 FOREIGN KEY (PID)
Ticket		CREATE TABLE Ticket (TicketNumber VARCHAR2(13),FlightCode VARCHAR2(15) NOT NULL,PID number NOT NULL,EmployeeID number NOT NULL,constraint CHECK_length_4 CHECK (length(PID) =10),constrai
android_metadata		CREATE TABLE android_metadata (locale TEXT)
city		CREATE TABLE city (CityName varchar (50) not null primary key,Country varchar2(50))
have		CREATE TABLE have (AirPortName VARCHAR2(50) not null ,AirlineID VARCHAR2(50) not null,constraint have_FK1 FOREIGN KEY (AirPortName) REFERENCES Airport(AirPortName),constraint have_FK2 FOREIGI
Indices (0)		
Views (0)		
Triggers (0)		

Figure 41. Tables Created Using SQLite3.

6. SQL Code:

SQL CODE

```
/*----------------------CREATING TABLES--------------------*/

CREATE TABLE city (

CityName varchar (50) not null primary key,

Country varchar2(50)

);


create table Airport (

AirPortName VARCHAR2(50) not null primary key,

Countary varchar2(50),

CityName varchar(50)  not null

);

Create table have (

AirPortName VARCHAR2(50) not null ,

AirlineID VARCHAR2(50) not null

);

create table Airline(

OfficeID number unique not null ,

ThreeDigitCode number unique not null,

AirlineID VARCHAR2(50) primary key not null,

AirlineName varchar2(50) unique

);

Create table Flight (

Source VARCHAR2(50) NOT NULL,

FlightCode VARCHAR2(6)NOT NULL primary key,

Destination VARCHAR2(50) NOT NULL,
```

```
Duration VARCHAR2(50) ,
DepartureT VARCHAR2(50) NOT NULL,
ArrivelTime VARCHAR2(50) NOT NULL,
AirlineID VARCHAR2(50)  not null,
EmployeeID number NOT NULL
);

create table Passenger (
Sex VARCHAR2(20),
Lname VARCHAR2(50) NOT NULL,
Fname VARCHAR2(50) NOT NULL,
Age date,
PID number  NOT NULL Primary Key,
PassportNumber  VARCHAR2(7)NOT NULL,
PPhone NUMBER,
FlightCode VARCHAR2(6)NOT NULL
);
create table Ticket (
TicketNumber VARCHAR2(13) Primary Key,
DateOfTravel DATE  NOT NULL,
ArrivelTime VARCHAR2(50) NOT NULL,
DepartureT  VARCHAR2(50) NOT NULL,
SeatNumber VARCHAR2(3) UNIQUE,
Source VARCHAR2(50),
Destination VARCHAR2(50),
```

```
PID number  NOT NULL,
EmployeeID number  NOT NULL
);


create table Employee (
Salary NUMBER,
Lname VARCHAR2(50) NOT NULL,
JobType VARCHAR2(50) NOT NULL,
Fname VARCHAR2(50) NOT NULL,
Age date ,
AirPortName VARCHAR2(50) not null ,
EmployeeID number Primary Key NOT NULL,
Sex VARCHAR2(20),
EPhone NUMBER,
Password VARCHAR2(20) NOT NULL
);


Create table Serves(
EmployeeID number NOT NULL,
PID number not null
);


create table Books (
DateOfBooking DATE primary key,
TicketNumber VARCHAR2(13),
PID number  NOT NULL
```

```
);


create table Cancels (

DateOfCanselletion DATE primary key,

TicketNumber VARCHAR2(13),

PID number  NOT NULL

);

/*-----------------------END OF CREATING TABLES---------------------
*/

/*-----------------------CONSTRAINTS---------------------*/

Alter table Airport

add constraint Aitport_FK1 FOREIGN KEY (CityName) REFERENCES
city(CityName);



Alter table have

add constraint have_FK1 FOREIGN KEY (AirPortName) REFERENCES
Airport(AirPortName);



Alter table have

add constraint have_FK2 FOREIGN KEY (AirlineID) REFERENCES
Airline(AirlineID);



Alter table Airline

add constraint CHECK_length_1 CHECK (length (OfficeID) =4  and
length(ThreeDigitCode)=3 ) ;



Alter table Flight
```

```
add constraint CHECK_length_2 CHECK (length(FlightCode) <6 or
length(FlightCode) =6  );



Alter table Flight

add constraint Flight_FK1 FOREIGN KEY (AirlineID) REFERENCES
Airline(AirlineID);



Alter table Flight

add constraint Flight_FK2 FOREIGN KEY (EmployeeID) REFERENCES
Employee(EmployeeID);



Alter table Passenger

add constraint Passenger_FK1 FOREIGN KEY (FlightCode) REFERENCES
Flight (FlightCode);



Alter table Passenger

add constraint CHECK_length_3 CHECK ( length(PassportNumber) =7 and
length(PPhone)=8 and length(PID) =10);


Alter table Ticket

add constraint CHECK_length_4 CHECK (length(PID) =10);


Alter table Ticket

add constraint Ticket_FK1 FOREIGN KEY (PID) REFERENCES
Passenger(PID);


Alter table Ticket

add constraint ticket_FK2  FOREIGN KEY(EmployeeID) references
employee(EmployeeID);



Alter table Serves
```

```
add constraint Serves_FK1 FOREIGN KEY (employeeID) REFERENCES
employee(employeeID);



Alter table Serves

add Constraint  serves_FK2 FOREIGN KEY (PID ) references
Passenger(PID)  ;



Alter table Employee

add constraint CHECK_length_5 CHECK ( length(EPhone)=8 and
length(EmployeeID)=10) ;



Alter table Employee

add constraint Employee_FK1 FOREIGN KEY (AirPortName) REFERENCES
Airport(AirPortName);



Alter table cancels

add Constraint  cancels_FK1 FOREIGN KEY (TicketNumber ) references
ticket(TicketNumber)  ;



Alter table cancels

add Constraint  cancels_FK2 FOREIGN KEY (PID ) references
Passenger(PID)  ;


Alter table books

add Constraint  books_FK1 FOREIGN KEY (TicketNumber ) references
ticket(TicketNumber)  ;


Alter table books

add Constraint  books_FK2 FOREIGN KEY (PID ) references
Passenger(PID)  ;
```

```
/*----------------------END OF CONSTRAINTS--------------------*/

insert into city (CityName,country) values ('abha','saudi arabia');

insert into Airport (AirPortName,Countary,CityName ) values ('abha
international airport','saudi arabia','abha');

insert into airline(OfficeID  ,ThreeDigitCode ,AirlineID
,AirlineName) values (1234,229,'KU','Kuwait Airways');
insert into airline(OfficeID  ,ThreeDigitCode ,AirlineID
,AirlineName) values (5643,265,'SP','SOLINAIR');
insert into airline(OfficeID  ,ThreeDigitCode ,AirlineID
,AirlineName) values (5896,199,'TU','Tunisair');
insert into airline(OfficeID  ,ThreeDigitCode ,AirlineID
,AirlineName) values (5986,235,'TK','Turkish Airlines');


insert into Employee (Salary,
Lname,JobType,Fname,Age,AirPortName,EmployeeID,Sex,EPhone,Password)
values(5000, 'Ahmad', 'manager','hassan',
TO_DATE('1989/04/05','yyyy/mm/ dd'),'abha international airport',
2111585893, 'Male', 55446688,123456);
insert into Employee (Salary,
Lname,JobType,Fname,Age,AirPortName,EmployeeID,Sex,EPhone,Password)
values(3000, 'abdullah', 'engineer','yaseen',
TO_DATE('1998/08/05','yyyy/mm/ dd'),'abha international airport',
5643218795, 'Male', 48956325,123456);
insert into Employee (Salary,
Lname,JobType,Fname,Age,AirPortName,EmployeeID,Sex,EPhone,Password)
values(200, 'baqer', 'worker','yosif', TO_DATE('1978/04/07','yyyy/mm/
dd'),'abha international airport', 2181156439, 'Male',
69854723,123456);

INSERT INTO flight ( Source ,FlightCode ,Destination
,Duration,DepartureT,ArrivelTime ,AirlineID ,EmployeeID) values
('Kuwait','KU1005','Istanbul','3:40','9:00AM','12:40PM','KU',
2111585893);

INSERT INTO flight ( Source ,FlightCode ,Destination
,Duration,DepartureT,ArrivelTime ,AirlineID ,EmployeeID) values
('Oman','SP4306','USA','16:34','7:00AM','11:34PM','SP', 2111585893);

INSERT INTO flight ( Source ,FlightCode ,Destination
,Duration,DepartureT,ArrivelTime ,AirlineID ,EmployeeID) values
('Emirates','TU4987','Iran','2:00','10:00AM','12:00PM','TU',
2111585893);

INSERT INTO flight ( Source ,FlightCode ,Destination
,Duration,DepartureT,ArrivelTime ,AirlineID ,EmployeeID) values
('saudi arabia','TU5556','Australia','14:14','11:00AM','1:00AM','TU',
2111585893);
```

```
INSERT INTO flight ( Source ,FlightCode ,Destination
,Duration,DepartureT,ArrivelTime ,AirlineID ,EmployeeID) values
('Qatar','TK6566','Kuwait','1:25','1:00PM','2:25PM','TK',
2111585893);

insert into passenger (Sex, Lname,Fname,
Age,PassportNumber,PID,PPhone,FlightCode) values ('Male',
'khaled','mohammad', TO_DATE('1999/04/05','yyyy/mm/ dd'),1234567,
1234567899,69676161, 'KU1005');

insert into passenger (Sex, Lname,Fname,
Age,PassportNumber,PID,PPhone,FlightCode) values ('Female',
'Mnar','Abdullah', TO_DATE('1945/11/06','yyyy/mm/ dd'),5643218,
9966553384,69583246, 'SP4306');

insert into passenger (Sex, Lname,Fname,
Age,PassportNumber,PID,PPhone,FlightCode) values ('Male',
'khaled','Ahmad', TO_DATE('1999/04/05','yyyy/mm/ dd'),5632984,
4466330022,69532016, 'TU4987');

insert into passenger (Sex, Lname,Fname,
Age,PassportNumber,PID,PPhone,FlightCode) values ('Male',
'Ali','Abbas', TO_DATE('1999/04/05','yyyy/mm/ dd'),5698331,
8796425349,69852314,'TU5556');

insert into passenger (Sex, Lname,Fname,
Age,PassportNumber,PID,PPhone,FlightCode) values ('Male',
'Abdulgader','Jassem', TO_DATE('1999/04/05','yyyy/mm/ dd'),7744653,
8796528430,98651423, 'TK6566');

insert into Have (AirportName,AirlineID) values ('abha international
airport','KU' );

insert into ticket (DateOfTravel, ArrivelTime, DepartureT,
SeatNumber, Source,Destination,TicketNumber,EmployeeID, PID) values (
TO_DATE('2022/03/05','yyyy/mm/ dd'),'12:40PM','9:00AM',
'25c','Kuwait','Istanbul','2291234567890', 2111585893, 1234567899 );

insert into serves (EmployeeID, PID) values ( 2111585893,
1234567899);

insert into books (DateOfBooking, TicketNumber,PID) values
(TO_DATE('2022/04/05','yyyy/mm/ dd'), '2291234567890',1234567899 );

insert into cancels (DateOfCanselletion,TicketNumber,PID) values
(TO_DATE('2022/04/05','yyyy/mm/ dd'),'2291234567890',1234567899);
```

7. limitations:

Every project has some limitations, as well our project had some limitations while implementing the database in SQL, but we overcame these limitations in the implementation of the interface, some of the limitations are the login page for the employee which was solved by using python, the employee cannot change the flight time or the airline's information, employees can't change their profile information, employees can only book or cancel tickets or edit passenger information, passengers can see only 5 flights in the flight table or he see his booked flight in a condition that it's the latest flight. Canceling the ticket from the passenger information is not secured because anyone can cancel the flight ticket if he only knows the passport number of the passenger, so the system leaks for some security and privacy, some of these limitations are solved using the interface phase and using the functions that connected the database with the interface.

8. Pros, Cons, And Future Development:

8.1 Development:

The systems still in the first steps and we have intentions to develop the system to make it more secure and have more functionalities and make it more friendly usable for users following HCI concepts of designing UIs.

-Some of development that could make the program more user friendly:

1- Registration interface for both employees and passengers and a login page for passengers to protect the privacy of the users

2- Give the ability for employees to edit their information.

3- Give the option to passengers to book tickets (With Prices and classes for flights)

4- Design different interface with specific functionalities according to the employee's job type.

8.2 Pros:

1- In the implementing process in this project we learned a lot about a very powerful library to design GUI and other library to link the interface with a database.

2- The system does what it's required to do as an Airport database management system.

3- Both interfaces are linked to the same database, so the system is always up to data.

8.3 Cons:

1- The system doesn't take into consideration the number of seats in each flight.

2- The system lacks to privacy and functionalities.

9. Conclusion:

Creating this project enabled team members to gain many skills throughout it phases, it encourages the members to become more responsible and more creative, also we cannot forget the teamwork which was the main feature that help the project to be completed. Through the project phases We learned how to create a more professional interface, and database by using the best available tools, for instance using PAGE tool for creating the interface, using SQL developer and SQL lite to create the database, also some of the skills that we gained are connecting the two interfaces with one database and having a parallel changes for the same database, creating the functions that connect the two interfaces, using a powerful libraries such as Tkinter and sqlite3, perform a more reliable system by solving many of the problems that can occur to the user, basically this system collect all the team members effort, time and precision, so we are thankful that the project came up with the best quality.

10. Resources:

1- scheme, C., & Tech, A. (2017). Converting an ER diagram into relational scheme. Retrieved 27 April 2022, from https://stackoverflow.com/questions/44986305/converting-an-er-diagram-into-relational-scheme

2-tables?, H., & Gao, H. (2009). How to drop all user tables?. Retrieved 27 April 2022, from https://stackoverflow.com/questions/1690404/how-to-drop-all-user-tables

3-IATA. (2022). Retrieved 27 April 2022, from https://www.iata.org/

4- 342 and 341L labs and lectures.

5- PAGE - A Python GUI Generator. (2022). Retrieved 20 May 2022, from http://page.sourceforge.net/#:~:text=PAGE%20is%20a%20cross%2Dplatform,which%20is%20embedded%20in%20tkinter.

6- sqlite3 — DB-API 2.0 interface for SQLite databases — Python 3.10.4 documentation. (2022). Retrieved 20 May 2022, from https://docs.python.org/3/library/sqlite3.html

7- SQLite Python. (2022). Retrieved 20 May 2022, from https://www.sqlitetutorial.net/sqlite-python/