

Helwan University  
Faculty of Engineering

# DBMS

## Airlines System

# Database

Dr. Hadeer Ahmed

Group No.07

Submission Date  
24/12/2022

Prepared By:

Mohamed Mahmoud Abdel Aziz Elsehetry	21116
Mohamed Hassan Kamal	21066
Abdullah Fawzi Abdel Satar Salamah	21041
Moataz Mohamed Megahed	21047

## **Introduction**

Airlines is having more transactions, causing day to day expanding business activities hard to manage its operations. To ensure more flexible service for customers, it's highly recommended to implement Airlines Reservation System (ARS), a computerized system that will help manage all information related to flight, passengers, their contact details, reservation, transactions, schedule publishing, air far payments etc.

## **Purpose and Method**

ADBS is operating on-spot airline reservation, and flight booking services with help of several Airports. Though having branches in multiple cities targeting a high range of customers.

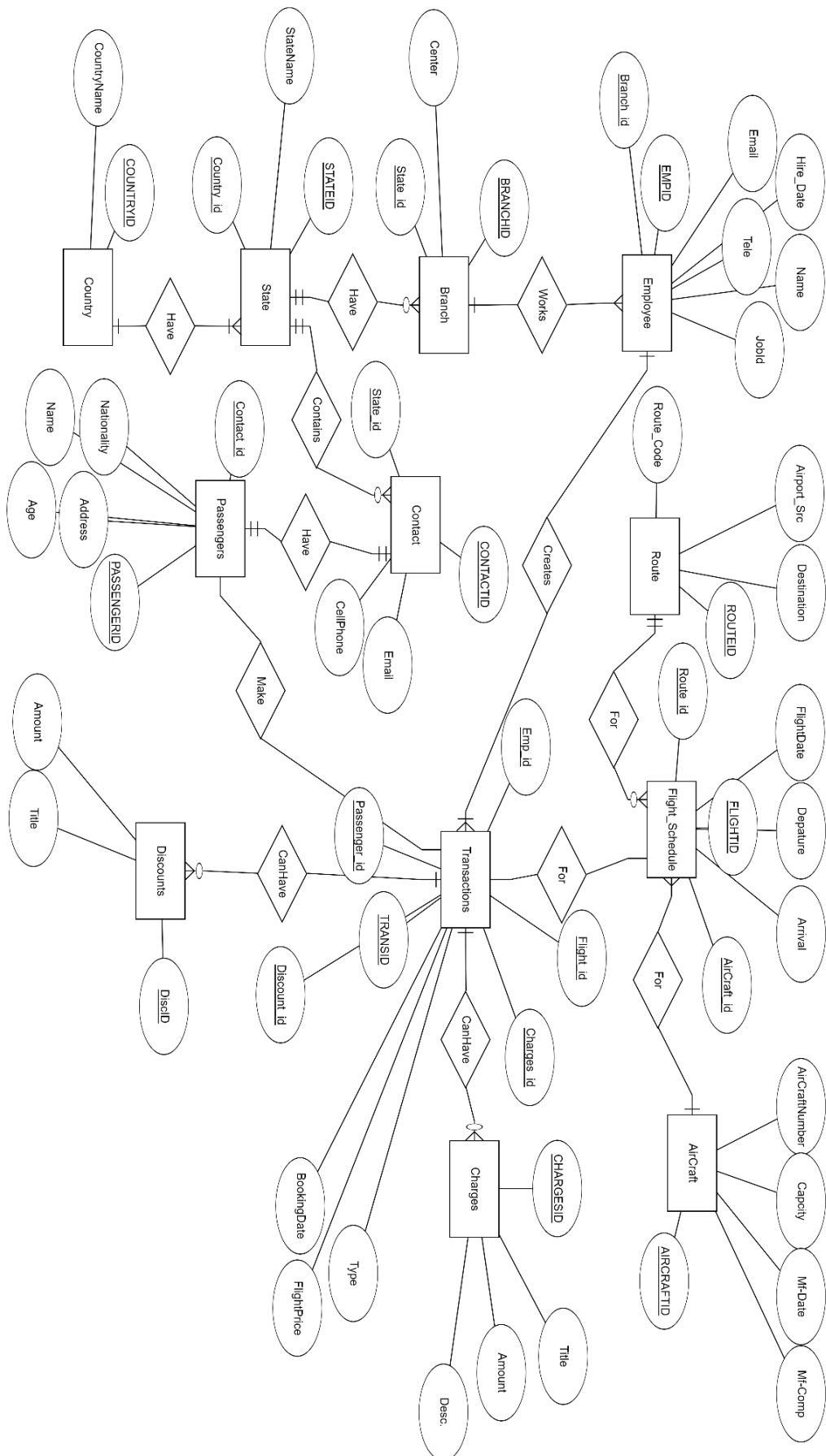
The purpose of the system is to facilitate reservation operations, link databases to each other accurately, and avoid human errors because one employee sometimes fails to satisfy customers in remote areas when they must travel for reservation and also when they need to travel for cancellation or flight day extension. The followings below are the system Entities, Attributes, and Relations.

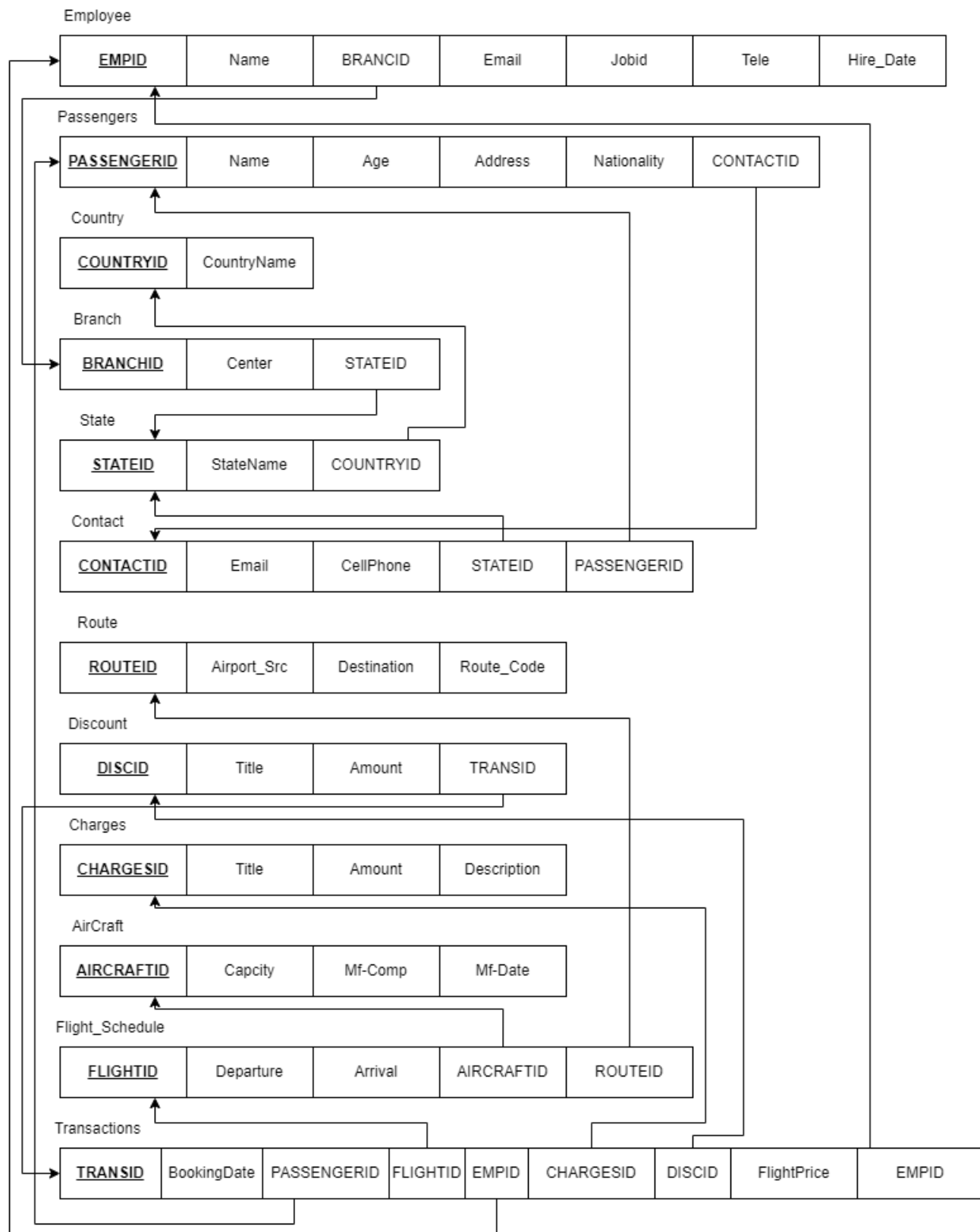
## **Database Design for ADBS**

### **Entities List:**

- AirCraft
- Branch
- Charges
- Contact
- Country
- Discounts
- Employee
- Flight\_Schedule
- Passengers
- Route
- State
- Transactions

# Database ER Diagram



**Database Relational Table Diagram**

## Tables Creation with SQL

### AirCraft Table

```
CREATE TABLE IF NOT EXISTS "AirCraft" (
  "AircraftId" INTEGER,
  "Aircraft_Number" nVarchar(32),
  "Capacity" INTEGER,
  "Mf_Comp" nVarchar(32),
  "Mf_Date" Date,
  PRIMARY KEY("AircraftId"));
```

### Charges Table

```
CREATE TABLE IF NOT EXISTS "Charges" (
  "ChargeId" INTEGER,
  "Title" nVarchar(32),
  "Amount" INTEGER,
  "Descr" TEXT,
  PRIMARY KEY("ChargeId"));
```

### Country Table

```
CREATE TABLE IF NOT EXISTS "Country" (
  "CountryId" INTEGER,
  "CountryName" nVarChar(32),
  PRIMARY KEY("CountryId"));
```

### Employee Table

```
CREATE TABLE IF NOT EXISTS "Employee" (
  "EmpId" INTEGER,
  "Name" nVarChar(32),
  "Branch_id" INTEGER,
  "Email" nVarChar(32),
  "JobId" nVarChar(32),
  "Tel" nVarChar(32),
  "Hire_date" Date,
  PRIMARY KEY("EmpId"),
  FOREIGN KEY("Branch_id") REFERENCES
  "Branch"("BranchId"));
```

### Passengers Table

```
CREATE TABLE IF NOT EXISTS "Passengers" (
  "PassengerId" INTEGER,
  "Name" nVarChar(32),
  "Age" INTEGER,
  "Address" nVarChar(64),
  "Nationality" nVarChar(32),
  "Contact_id" INTEGER,
  PRIMARY KEY("PassengerId"),
```

```
FOREIGN KEY("Contact_id") REFERENCES
  "Contact"("ContactId"));
```

### Route Table

```
CREATE TABLE IF NOT EXISTS "Route" (
  "RouteId" INTEGER,
  "Airport_Src" nVarChar(64),
  "Destination" nVarChar(64),
  "Route_Code" TEXT UNIQUE,
  PRIMARY KEY("RouteId"));
```

### Contact Table

```
CREATE TABLE IF NOT EXISTS "Contact" (
  "ContactId" INTEGER,
  "Email" nVarchar(64),
  "CellPhone" nVarchar(64),
  "State_id" INTEGER,
  PRIMARY KEY("ContactId"));
```

### Charges Table

```
CREATE TABLE IF NOT EXISTS "Branch" (
  "BranchId" INTEGER,
  "Center" nVarchar(32),
  "State_id" INTEGER,
  PRIMARY KEY("BranchId"));
```

### Charges Table

```
CREATE TABLE IF NOT EXISTS "State" (
  "StateId" INTEGER,
  "StateName" TEXT,
  "Country_id" INTEGER,
  PRIMARY KEY("StateId"),
  FOREIGN KEY("Country_id") REFERENCES
  "Country"("CountryId"));
```

**Transactions Table**

```
CREATE TABLE IF NOT EXISTS "Transactions" (  
  "TransId" integer NOT NULL,  
  "BookingDate" datetime,  
  "Passenger_id" integer,  
  "Flight_id" integer,  
  "Type" bit,  
  "Employee_id" integer,  
  "Charges_id" integer,  
  "Discount_id" integer,  
  "FlightPrice" numeric,  
  PRIMARY KEY("TransId"),  
  FOREIGN KEY("Employee_id") REFERENCES  
    "Employee"("EmpId"),  
  FOREIGN KEY("Passenger_id") REFERENCES  
    "Passengers"("PassengerId"),  
  FOREIGN KEY("Discount_id") REFERENCES  
    "Discounts"("DiscId"),  
  FOREIGN KEY("Charges_id") REFERENCES  
    "Charges"("ChargeId"), FOREIGN KEY("Flight_id")  
  REFERENCES "Flight_Schedule"("FlightId"));
```

**Discounts Table**

```
CREATE TABLE IF NOT EXISTS "Discounts" (  
  "DiscId" INTEGER,  
  "Title" nVarChar(32),  
  "Amount" TEXT,  
  PRIMARY KEY("DiscId"));  
  
CREATE TABLE IF NOT EXISTS "Flight_Schedule" (  
  "FlightId" integer NOT NULL,  
  "Flight_Date" datetime,  
  "Departure" nvarchar(52) COLLATE NOCASE,  
  "Arrival" nvarchar(52) COLLATE NOCASE,  
  "AirCraft_id" integer,  
  "Route_id" integer,  
  PRIMARY KEY("FlightId"),  
  FOREIGN KEY("AirCraft_id") REFERENCES  
    "AirCraft"("AircraftId"),  
  FOREIGN KEY("Route_id") REFERENCES  
    "Route"("RouteId"));
```

## Prepared SQL Queries

1. View passengers on certain flight (Foreign passengers Or All passengers option)

- **SQL Query (Foreign passengers):**

```
SELECT P.PassengerId, P.Name, P.Age, P.Address, P.Nationality,
       P.Contact_id, C.Email, C.CellPhone, C.State_id
FROM   Transactions AS T, Passengers AS P, Contact AS C
WHERE  T.Passenger_id=P.PassengerId AND P.Contact_id=C.ContactId AND
       T.Flight_id selectedFlightID AND P.Nationality NOT LIKE 'Egy%'
```

- **Output:**

The screenshot shows the Airlines DBMS web application. The 'Perform Query' button is visible. The query results are displayed in a table with the following data:

	PassengerId	Name	Age	Address	Nationality	Contact_id	Email	CellPhone	State_id
1	11	Ilieff, Mr. Ylio	15	2140 Euclid ...	Qatar	22	maryHoney@...	(432) 889-8707	15
2	3	Myles, Mr. ...	62	24 Second ...	Qatar	10	eadesmichell...	(214)771-4611	13
3	24	Williams, Mr. ...	21	800 Middle ...	Tunis	6	dollface0908...	(214) 739-5600	26
4	19	Ilmakangas, ...	27	19 West 39th ...	Qatar	19	krissyh@att.net	(956) 668-0234	13
5	19	Ilmakangas, ...	27	19 West 39th ...	Qatar	19	krissyh@att.net	(956) 668-0234	13
6	16	del Carlo, Mrs...	24	1161 South ...	Tunis	16	k.gold@live.c...	(210)216-4480	22

## 2. View Flights on (certain Day) Going to (Certain Airport)

- **SQL Query:**

```
SELECT DISTINCT F.FlightId, F.Flight_Date, F.Departure,
                F.Arrival, R.Airport_Src, R.Destination, R.Route_Code
FROM   Flight_Schedule AS F, Route AS R
WHERE  F.Route_id=R.RouteId AND F.Flight_Date=SelectedDate AND
R.Destination=SelectedDistination
```

- **Output:**

Nesma Airlines DBMS

DB Browser

Perform Query

Query 1

Query 2

Query 3

Query 4

Query 5

Query 6

View Flights on Day

2022-12-23

Going to

Jeddah King Abdul Aziz International Airport

Perform Query

	FlightId	Flight_Date	Departure	Arrival	Airport_Src	Destination	Route_Code
1	1	2022-12-23	12:30:00	14:30:00	Cairo ...	Jeddah King ...	NE-170
2	13	2022-12-23	7:00:00	10:00:00	HBE Alexandria ...	Jeddah King ...	NE-201
3	14	2022-12-23	10:00:00	13:00:00	Tunis(TUN) ...	Jeddah King ...	NE-202



3. Calculate sum of tickets prices booked in (Certain Day)

- **SQL Query:**

```
SELECT COUNT(FlightPrice) AS 'Number of Flights', sum(FlightPrice) AS  
        'TOTAL INCOME'  
FROM    Transactions AS T  
WHERE   T.BookingDate = SelectedDate
```

- **Output:**

**Nesma Airlines DBMS**

DB Browser   Perform Query

Query 1   Query 2   Query 3   Query 4   Query 5   Query 6

Calculate sum of tickets prices booked in day : 2022-09-14

Perform Query

	TOTAL INCOME
1	11325

## 4. View Flights flying from (Certain Airport)

- **SQL Query:**

```
SELECT F.FlightId, F.Flight_Date, F.Departure, F.Arrival, R.Airport_Src,
       R.Destination, R.Route_Code
FROM Flight_Schedule AS F, Route AS R
WHERE F.Route_id=R.RouteId AND R.Airport_Src = selectedFlightID
```

- **Output:**

Nesma Airlines DBMS							
DB Browser		Perform Query					
Query 1		Query 2	Query 3	Query 4	Query 5	Query 6	
View flights flying from <input type="text" value="Cairo International Airport"/>							Perform Query
	FlightId	Flight_Date	Departure	Arrival	Airport_Src	Destination	Route_Code
1	1	2022-12-23	12:30:00	14:30:00	Cairo ...	Jeddah King ...	NE-170
2	2	2022-12-24	3:20:00	5:40:00	Cairo ...	SSH Sharm El ...	NE-176
3	3	2022-12-25	1:00:00	2:00:00	Cairo ...	SSH Sharm El ...	NE-176
4	6	2022-12-28	20:00:00	21:00:00	Cairo ...	Doha ...	NE-185
5	15	2023-01-06	13:00:00	17:00:00	Cairo ...	Doha ...	NE-185
6	16	2023-01-07	14:00:00	18:00:00	Cairo ...	Doha ...	NE-185
7	19	2023-01-10	21:20:00	24:00:00	Cairo ...	SSH Sharm El ...	NE-176
8	20	2023-01-11	20:00:00	23:00:00	Cairo ...	SSH Sharm El ...	NE-176
9	23	2023-01-14	9:00:00	15:00:00	Cairo ...	Jeddah King ...	NE-170
10	24	2023-01-15	11:00:00	19:00:00	Cairo ...	Jeddah King ...	NE-170

## 5. View Passengers with applied Discounts

- **SQL Query:**

```
SELECT Passengers.Name, Transactions.Passenger_id, Transactions.Flight_id,
       Transactions.FlightPrice, Transactions.Discount_id, Discounts.Title,
       Discounts.Amount, (FlightPrice - (FlightPrice*Amount)) AS 'price after
       discount'
```

```
FROM Transactions INNER JOIN Discounts INNER JOIN Passengers
ON Transactions.Discount_id=Discounts.DiscId AND
Transactions.Passenger_id=Passengers.PassengerId ORDER BY
Transactions.Discount_id
```

- **Output:**

Nesma Airlines DBMS

DB Browser

Perform Query

Query 1

Query 2

Query 3

Query 4

Query 5

Query 6

Show Passengers with applied discount

Perform Query

	Name	Passenger_id	Flight_id	FlightPrice	Discount_id	Title	Amount	rice after discoun
1	Hirvonen, Mrs. ...	5	1	5147	1	Childrens	0.12	4529.36
2	Williams, Mr. ...	24	2	5118	1	Childrens	0.12	4503.84
3	Flegenheim, Mr...	23	4	5906	1	Childrens	0.12	5197.28
4	Rothschild, Mr. ...	21	1	1062	2	Old People	0.13	923.94
5	Flegenheim, Mr...	23	1	4683	2	Old People	0.13	4074.21
6	Ilieff, Mr. Ylio	11	2	5218	2	Old People	0.13	4539.66
7	Assaf, Mr. Gerios	18	3	2154	2	Old People	0.13	1873.98
8	Myles, Mr. ...	3	2	4290	3	Occassions	0.1	3861.0
9	Howard, Mr. ...	14	4	6621	3	Occassions	0.1	5958.9
10	Robins, Mr. ...	26	4	4252	4	SpecialCustomer	0.20	3401.6

6. View Passenger who reserved tickets from (Certain Date) to (Certain Date)

- **SQL Query:**

```
SELECT T.BookingDate, T.FlightPrice, P.PassengerId, P.Name, P.Age,
       P.Address, P.Nationality\
FROM   Transactions AS T, Passengers AS P\
WHERE  T.Passenger_id=P.PassengerId AND T.BookingDate >= Date1 AND
       T.BookingDate <= Date2 ORDER BY T.BookingDate
```

- **Output:**

# Nesma Airlines DBMS

DB Browser

Perform Query

Query 1

Query 2

Query 3

Query 4

Query 5

Query 6

View passengers who reserved Tickets From date

2022-09-10

To

2022-09-10

Perform Query

	BookingDate	FlightPrice	PassengerId	Name	Age	Address	Nationality
1	2022-09-10	6096	15	Chaffee, Mrs. ...	47	1500 Valencia ...	Qatar
2	2022-09-10	1062	21	Rothschild, Mr. ...	55	2013 Avenue of ...	Egypt
3	2022-09-10	2456	23	Flegenheim, Mr...	10	500 Arbor Road	Qatar

## 7. View Minimum and Maximum Flight Prices

- **SQL Query:**

**SELECT \***

**FROM Transactions**

**Where FlightPrice IN (Select max(FlightPrice) From Transactions)**

**Or FlightPrice IN (Select min(FlightPrice) From Transactions);**

- **Output:**

Airlines DBMS									
DB Browser		Perform Query							
Query 1	Query 2	Query 3	Query 4	Query 5	Query 6	Query 7	Query 8	Query 9	
View Maximum and Minimum Flight Prices									Perform Query
	TransId	BookingDate	Passenger_id	Flight_id	Type	Employee_id	Charges_id	Discount_id	FlightPrice
1	3	2022-09-12	8	1	1	17	2	None	6991
2	21	2022-09-30	28	4	0	2	None	None	1042

## 8. View Special Customers who (Have 3 or more Transactions)

- **SQL Query:**

**Select \***

**From** Transactions T **JOIN** Passengers p **ON** T.Passenger\_id == p.PassengerId

**JOIN** Discounts d **ON** d.DiscId == T.Discount\_id

**GROUP BY** T.Passenger\_id

**Having** Count(T.Passenger\_id) >=3 **AND** Max(T.TransId);

- **Output:**

Airlines DBMS

DB Browser

Perform Query

Query 1

Query 2

Query 3

Query 4

Query 5

Query 6

Query 7

Query 8

Query 9

View Special Customers ( Have 3 or more Transactions)

Perform Query

	TransId	BookingDate	Passenger_id	Flight_id	Type	Employee_id	Charges_id	Discount_id	FlightPrice	PassengerId	Name
1	31	2022-09-13	11	5	0	25	None	4	6856	11	Ilieff, Mr. Ylio
2	30	2022-09-24	18	6	0	25	None	4	2369	18	Assaf, Mr. Gerios
3	20	2022-10-04	19	3	0	22	None	4	2154	19	Ilmakangas, Mi...

9. A Query to Find where each branch Center is located

- **SQL Query:**

```
SELECT B.BranchId,B.Center,S.StateId, S.StateName
FROM Branch AS B LEFT OUTER JOIN State AS S
ON B.State_id = S.StateId
```

- **SQL Query:**

Airlines DBMS				
DB Browser		Perform Query		
Query 1	Query 2	Query 3	Query 4	Query 5
Query 6	Query 7	Query 8	Query 9	
Show Branches Locations				Perform Query
BranchId	Center	StateId	StateName	
1	Titan Airways	21	Sfax	
2	Tradewind ...	1	Alexandria	
3	Comlux Aviatio...	9	Al Jawf	
4	Master Top ...	11	Jeddah	
5	Flair Airlines Ltd.	18	Umm Salal	
6	Swift Air, LLC	3	Aswan	
7	DCA	17	Umm Sa	
8	ACM AIR ...	11	Jeddah	
9	Interisland ...	1	Alexandria	
10	Polar Airlines ...	11	Jeddah	
11	JetClub AG	27	Jendouba	
12	Vision Airlines	20	Al Wakrah ...	
13	Metropix UK, ...	28	Madanin	
14	Multi-Aero, Inc.	10	Al Madinah	
15	Open Skies	14	Al Jumaliyah	

## 10. View Flight Schedule for Certain Aircraft Number

- **SQL Query:**

**SELECT** A.Aircraft\_Number

**From** Aircraft as A

(**Where** AircraftId = Any **SELECT** F.Aircraft\_id **FROM** Flight\_Schedule F

**Where** AirCraft\_id = 1)