LAU
School of Arts and Sciences
Department of Computer Science & Mathematics

Fly Haraty

More than an Airline

A Database Design for The Fly Haraty Company By ***Combinatorial Solutions Inc***

By
Gaëlle Loutfi (Group Leader)
gaelle.loutfi@lau.edu
Hadi Al Hassan
hadi.alhassan@lau.edu
Siraj Al mabsout
siraj.almabsout@lau.edu
Sari Abdul Ghani
sari.abdulghani@lau.edu

A REPORT
Submitted to Dr. *Ramzi. A. Haraty* in partial fulfillment of the requirements of phase IV of the database project for the course CSC/BIF 375 – Database Management Systems
November 2023

Contents

This document is Proudly Typed in LaTeX, Using 1916 Lines of code and 25 Libraries
It is the result of several all-nighters and gallons of caffeine

We hope you enjoy it professor :)

PS: Sorry for using your name

- Hadi, Sari, Gaëlle and Siraj

Combinatorial Solutions Inc

Combinatorial Solutions began its journey as a simple contracting company that provided its problem-solving expertise to clients and grew until it became the largest holder of assets in the MENA region.

It is the proud owner of several companies in the Middle East, including Fly Haraty.

Fly Haraty

Fly Haraty, named after its founder *Ramzi. A. Haraty*, was founded in 1986. The son of a pilot, Ramzi always dreamed of having his own plane and worked towards his vision.
Fly Haraty started as a simple airline company with 5 planes and one office.

Logging operations began with a small black ledger, then moved to an online manifest.
Ramzi soon discovered that he couldn't keep track of his flights using a simple pen and paper, which was when he contacted Combinatorial Solutions (*CS*) Inc for help in designing a database for his company.

A few years later, Fly Haraty found itself as one of the largest airlines in the world with the largest customer base in the MENA Region.

CS often found itself adding new features to the company and would eventually acquire it in 2006, in a multi-billion dollar deal. Ramzi remains a board member to this day.

Combinatorial being their nature, *CS* expanded Fly Haraty to make a customer's flight a nuisance-free experience. Fly Haraty transformed into a full package travel company, where customers would book flights and hotel rooms, and transport between hotels and the airport. In light of the recent improvements, Fly Haraty changed its slogan to the current "*More than an airline*".

And the logo underwent some renovation too, to highlight the modernity of Fly Haraty.

1 Introduction

This database report outlines the requirements for implementing a database system for Fly Haraty. We will incorporate these requirements into an Entity Relationship Diagram using the Chen Notation. Additionally, we will provide explanations for each entity and its attributes, as well as the relationships that connect them. Furthermore, we will translate our ER-diagram into relational schemas following the seven steps outlined in the El Masri textbook.

2 Copyright Notice and Authorization

The Consent of *Ramzi. A. Haraty* was asked for prior to the design of the fictitious airline.

3 Tools used to draw the ER

From the variety of softwares available, we selected *Software Ideas Modeler* ©,
a free, non commercial version of the software to model our ER Diagram.

4 ER Diagram Notations

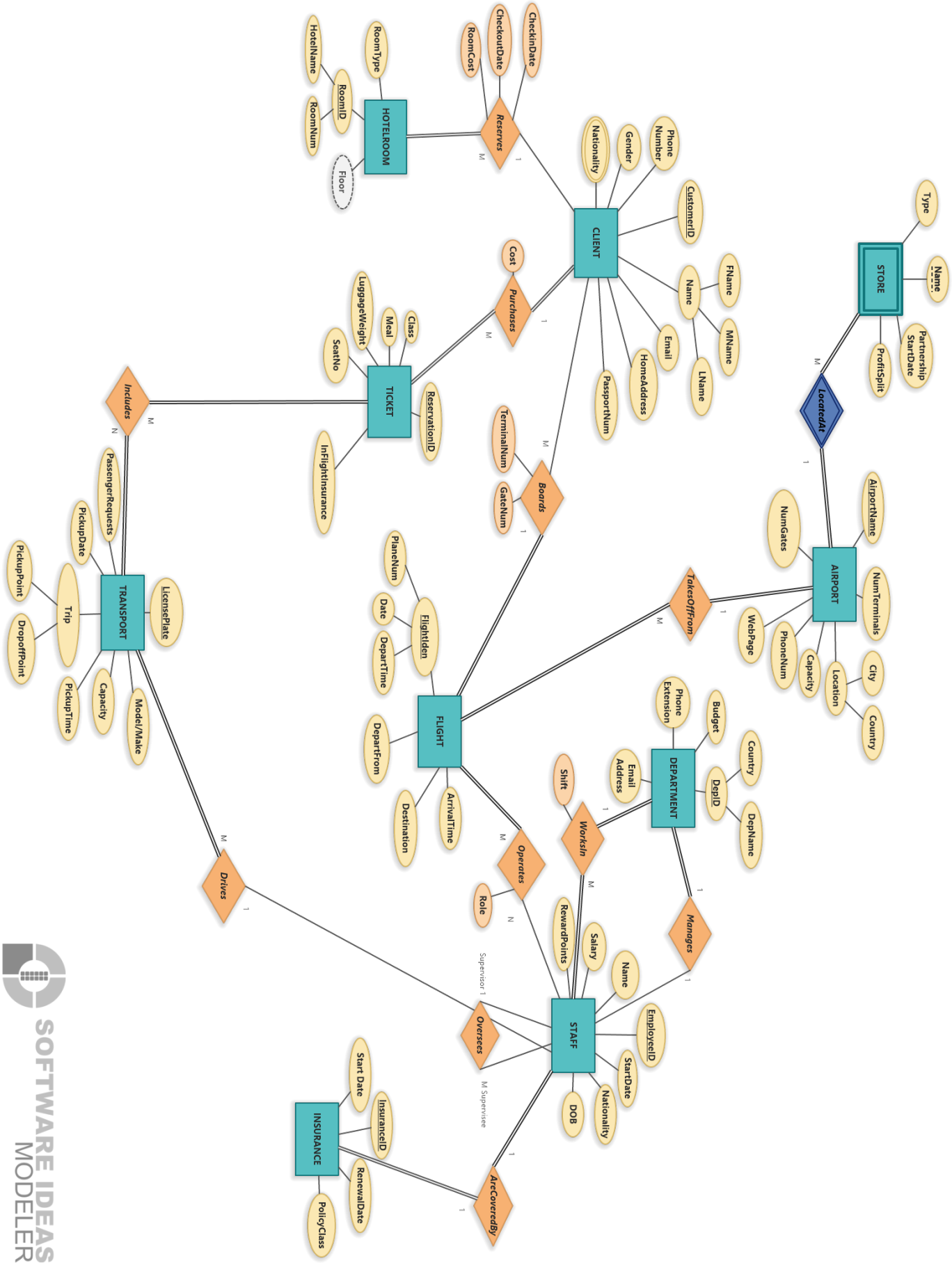
We will be using the Chen notation for our Entity Relationship Diagram.

Notation / Shape	ER Component	Description
	Entity	Rectangle
	Weak Entity	Double Border Rectangle
	Attribute	Oval
	Key Attribute	Oval with Underlined Text
	Partial Key Attribute	Oval with Dashed Underlined Text.
	Multivalued Attribute	Double Border Oval.
	Derived Attribute	Dashed Border Oval.
	Relationship	Rhombus
	Identifying Relationship	Double Border Rhombus.

5 System Description and Requirements

- We need to be able to store clients in our database the first time they purchase a ticket. We must store relevant information about our clients, such as their Name, Nationality, and Gender. For communication, we require their Phone Number and Email. To make the ticket purchase process smooth we should store the client's passport details. And in the case of lost luggage, we need the client's home address to ship the luggage back to them. We also require their ID to run an extensive background check routinely. Each customer will have their own unique Customer ID for our system.
- Clients purchase tickets, and have the luxury to customize their flight options whether it is the seating arrangement or their meal choice. The ticket should have all relevant details of the flight.
- Clients board flights. Details pertaining to the flight must be stored for tracking and logging purposes.
- Every ticket includes transport from and to the airport. Details such as the driver assigned, the car the client is being taken in, and the journey of the trip must be recorded.
- Every flight will take off and land at an airport.
- We should store information about each airport Fly Haraty owns, to make them unique.
- We must store details about our employees. Crucial details like their Name, Date of birth, Nationality, ID for background checks, the date they started working, the reward points the employee accumulates throughout his service, as well as the salary of the employee to keep financial matters clear. Each employee will be assigned a unique Employee identifier to distinguish them.
- Staff members all report to a department and are not allowed to work for more than one department.
- Every department should have only one manager, and they cannot manage another department.
- Some staff members will operate our flight and transportation services.
- We distinguish our departments through their identification, which is composed of the country and name of the department.
- Every airport must have a store inside to accommodate the customer's needs. A branch of the store in an airport is not distinguishable alone by name and should be related to the airport it is in.

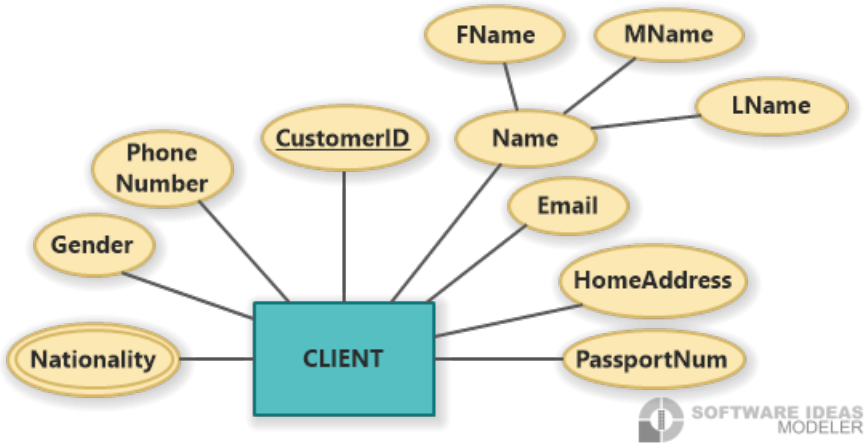
6 Complete ER Diagram for Fly Haraty



[1]For A clearer view Reference the A1 Paper submitted with the written report.

6.1 Entity Types and their Attributes

6.1.1 Client Entity

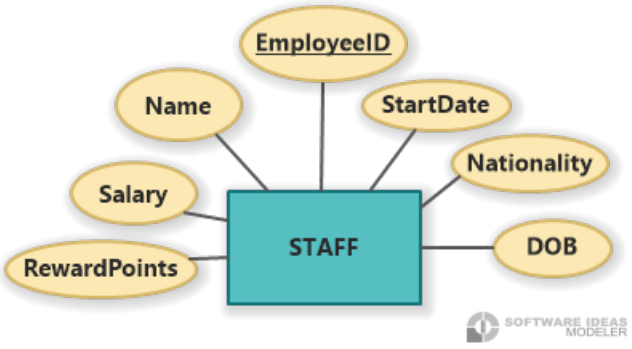


For us at Fly Haraty, the customer always comes first. Even in our diagram.

The Client is the forefront of any business. Hence, it is an entity in our database.

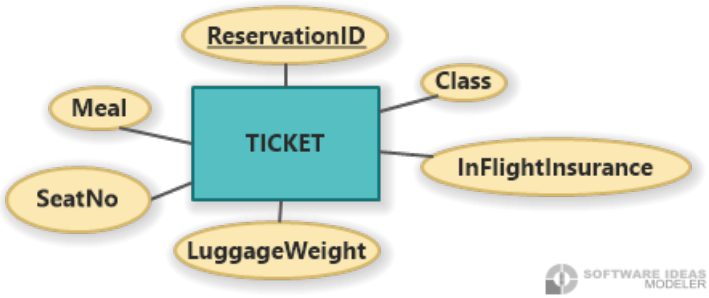
Each client is a unique individual, holding a **name**, **gender**, **identification number (ID)** and **nationality**. Every customer's ID is unique and therefore we set it as our primary key. To better service our client, We require their **phone number** and **email address** to contact them for confirming reservations among other purposes. To make the ticket reservation process smoother, we store a client's **passport details** to ensure they can book a flight with minimal effort. In the case our client loses their luggage during a flight, we store their **home address** to send the luggage if found.

6.1.2 Staff Entity



Our staff is the backbone of Fly Haraty. Every employee can be identified by their unique **employee ID**. Which we set as our primary key in this entity. A staff member holds attributes like their **name**, **date of birth**, **nationality** and **start date**. To run background checks on our employees (then applicants), we require their ID. Each employee has a **salary** and is given **reward points** for each year of service.

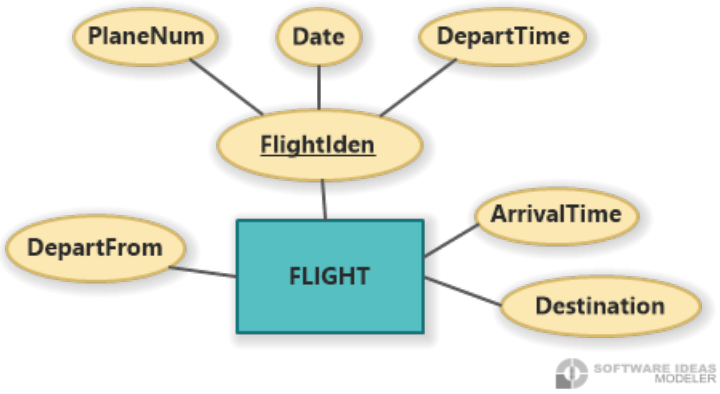
6.1.3 Ticket Entity



A ticket is a client's key to travel. We identify each ticket uniquely by its reservation ID, which we set as our key. A ticket contains all the information pertinent to a client's itinerary.

The ticket also includes the maximum luggage weight, which class, seat, and meal option the client selected when the ticket was purchased. Depending on the selections the client made concerning the class, luggage size, meal, and seating option, the ticket cost is assigned. At Fly Haraty, the customer's safety and comfort is our top priority. Which is why each ticket has an in-flight insurance that covers the medical costs of any emergency that occurs while the client is using our services.

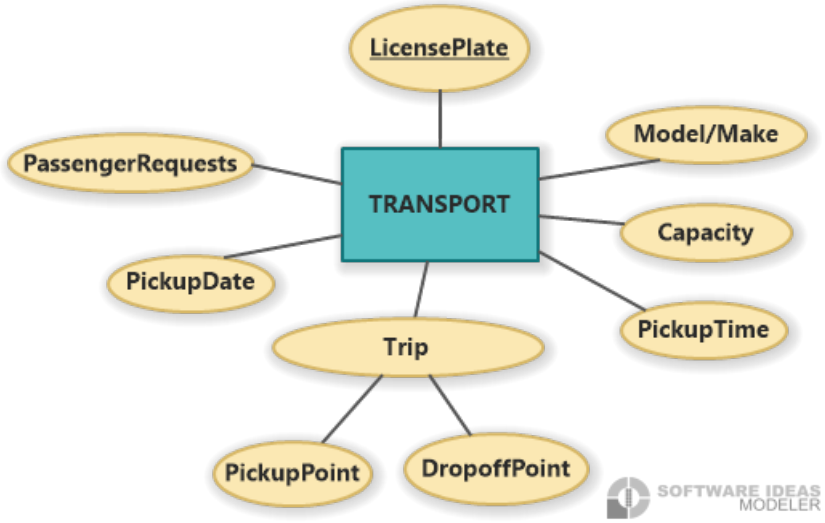
6.1.4 Flight Entity



Every ticket is associated with a flight, and so we set it as an entity in our database, and it has the following attributes:

- **FlightIden** We uniquely identify each flight using our custom identifier, which is a composite attribute consisting of the **plane number**, **flight date**, and the **time of departure**. Together, they form our primary key.
- **ArrivalTime**: The time the plane lands.
- **DepartFrom**: The airport our flight takes off from.
- **Destination**: The airport our plane lands at.

6.1.5 Transportation Entity

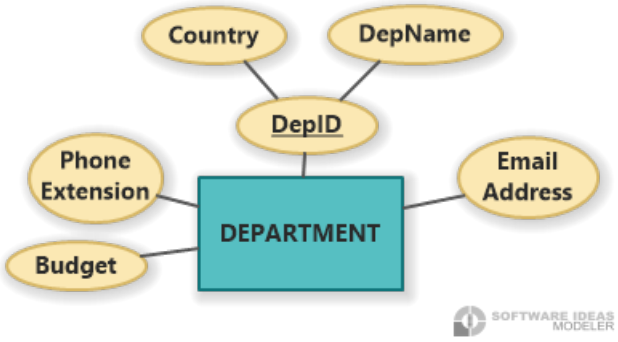


Fly Haraty prides itself in its transportation services offered to our clients, factored into the cost of every ticket is a ride to and from the airport.

Each transport car has the following attributes:

- **LicensePlate** Every Car is unique through its License plate number, and so we set it as our key.
- **Model/make**: The Brand and type of the car.
- **Capacity** How many individuals the vehicle can hold.
- **Trip**: Every car's journey is composed of a pickup and drop-off point.
- **PickupDate**: Date of pickup.
- **PickupTime**: Time our client is scheduled to be picked up at.
- **PassengerRequests**: Any request our customer might have.

6.1.6 Department Entity



Every department entity contains the following:

- **DepID** We differentiate each department through its unique identification, which is a combination of the Country and Department name, and so it is set as our key.
- **DepName**: Fly Haraty is divided into several departments. We assign each department entity a name (Management, HR, Security, Operations . . .).
- **EmailAddress**: Each department has its own email inbox for communication.
- **PhoneExtension**: Each department has its own phone extension for interdepartmental communication.
- **Budget**: Each department is allocated a yearly sum for its operations (salaries, projects . . .)

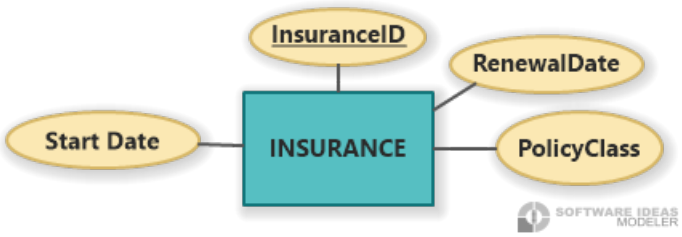
6.1.7 Airport Entity

Fly Haraty is now an international airline, with airports in the Middle East and Europe, and expanding annually.

An airport is the hub of all flights, and so it needs its own entity in our system.
Our entity has the following attributes:

- **AirportName** A moniker assigned to each airport.
  We set it as our key, as each airport's name is unique.
- **Location**: Composite attribute that contains the country and city of the airport.
- **NumGates**: The number of gates.
- **NumTerminals**: The number of terminals.
- **WebPage**: Every Fly Haraty airport has its own website.
- **Capacity**: Estimated number of customers the airport can accommodate.
- **PhoneNum**: Landline of each airport.

6.1.8 Insurance Entity

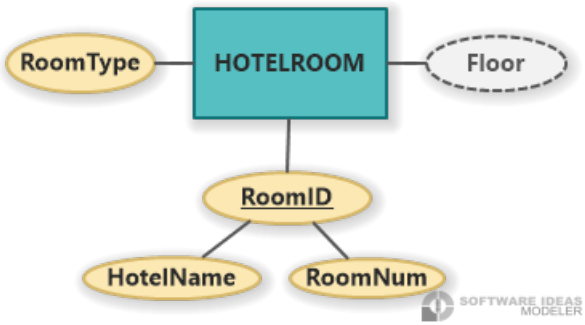


Health insurance is an important right for every employee, ensuring their well-being and safety. Every staff member at Fly Haraty is provided with insurance, which starts as a basic package. Staff members have the option to upgrade their insurance class if they choose to do so.

Our insurance entity is defined by the following attributes:

- **InsuranceID** Each employee's policy is assigned a unique ID for HR purposes. And it is the key of this entity.
- **StartDate**: The start date of the policy
- **RenewalDate**: The date of renewal for the policy.
- **PolicyClass**: The coverage class selected by the employee

6.1.9 Hotel Room Entity



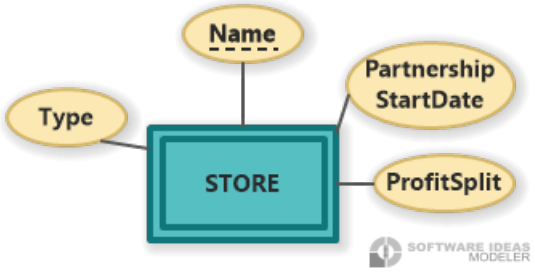
Fly Haraty allows the customer to reserve a hotel room during their stay.

Every hotel room contains:

- **RoomID** Each hotel room is identified using the hotel name and room number, and we set this attribute as our primary key.
- **RoomType**: What suite the client has requested (single, double, master . . .).

  we also have a derived attribute floor, which is identified through the room number.

6.1.10 Store Entity

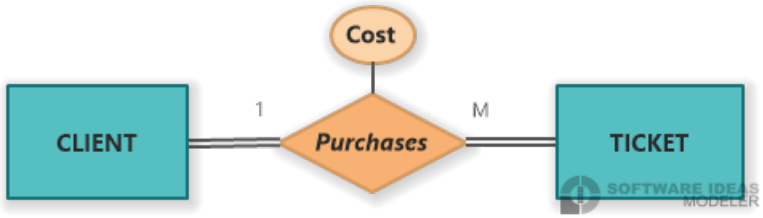


No airport is complete without one. Since we have many branches of the same store in more than one airport, we cannot uniquely identify the store, making it a weak entity in our database.
Our store has the following attributes:

- **Name**: Title of the franchise present at the airport, it cannot be used alone to identify each store uniquely, so it will be our partial key.
- **Type**: What commodity the store offers (Such as Retail, Fast Food. . .)
- **PartnershipStartDate**: The date of the store's opening.
- **ProfitSplit**: The percentage of the profit Fly Haraty receives from the store.

6.2 Relationships and their Explanation

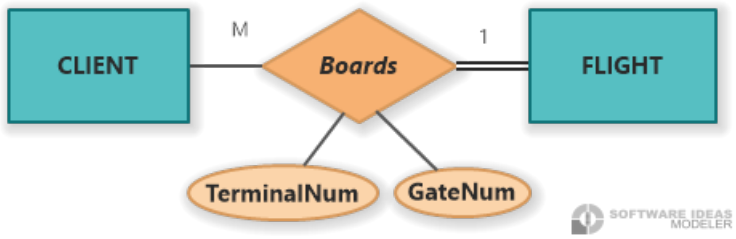
6.2.1 CLIENT *purchases* TICKET



It is a 1:M relationship, given that a client can purchase many tickets, and a ticket can be owned by only one client.

We note total participation for both CLIENT and TICKET since a client should purchase at least one ticket to be in our system, and every ticket must be associated with a client. This relationship also has an attribute cost, which is assigned depending on the selection the client made when purchasing the ticket.

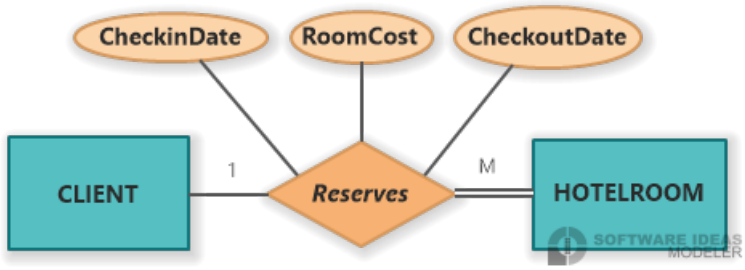
6.2.2 CLIENT *boards* FLIGHT



It is a 1:M relationship, since a client boards a flight and a flight can be boarded by many clients.

We note partial participation for the client entity since a client could have purchased a ticket, but ended up not boarding the flight, and total participation for the flight entity since every flight must be boarded by a client. The relationship includes two attributes TerminalNum and GateNum to illustrate that the client will board the flight from a certain terminal and gate.

6.2.3 CLIENT *reserves* HOTELROOM



It is a 1:M relationship, as a client can reserve many hotel rooms, and a hotel room can be reserved by only one client.

We note partial participation for the CLIENT entity since not every client will reserve a hotel room, and total participation for the HOTELROOM entity as every hotel room must be associated with a client.

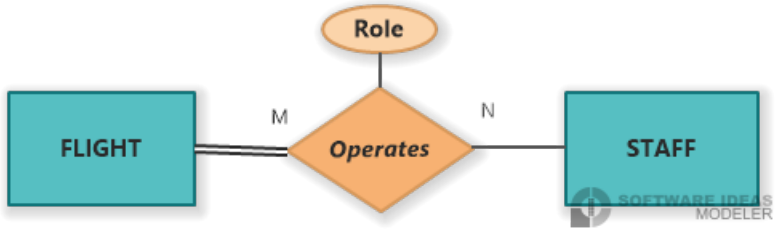
The relationship also includes the attributes check in date, check out date, and the room cost.

6.2.4 TICKET *includes* TRANSPORT

It is a N:M relationship, since a Ticket can include several transport cars (to and from the airport) and a transport can be included by several tickets. In other words, the same transport car can take several clients.

We note total participation for both, since every ticket has transportation included and every transport car must be associated with at least one ticket.

6.2.5 STAFF *operates* FLIGHT



It is a N:M relationship, since a staff member can operate many flights and a flight is operated by many staff.

We note partial participation for the STAFF entity since not all staff are associated with flight operations (office workers), and total participation for the FLIGHT entity since every flight needs to be operated by staff. The relationship also includes an attribute role that specifies the aspect each staff members contributes to each flight.

6.2.6 STAFF *drives* TRANSPORT

It is a 1:M relationship, since a staff member drives many transport vehicles and a transport car can only be driven by one staff member.

We note partial participation for the STAFF entity, similar to the *operates* relationship, not all staff are associated with driving a transport car,

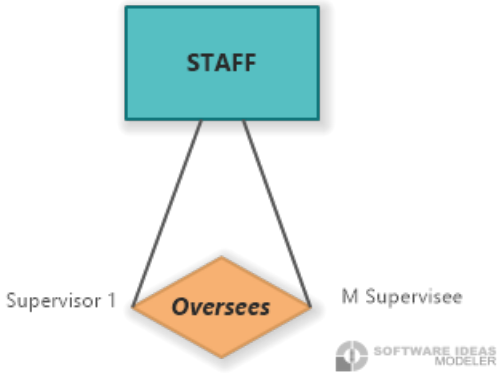
and we note total participation for the TRANSPORT entity, as every vehicle must driven by a staff member.

6.2.7 STAFF *arecoveredby* INSURANCE

It is a 1:1 relationship, as a staff member can be covered by only one insurance policy and a policy can only cover a single employee.

We note total participation for both STAFF and INSURANCE entities, as every client needs to be covered by an insurance policy.

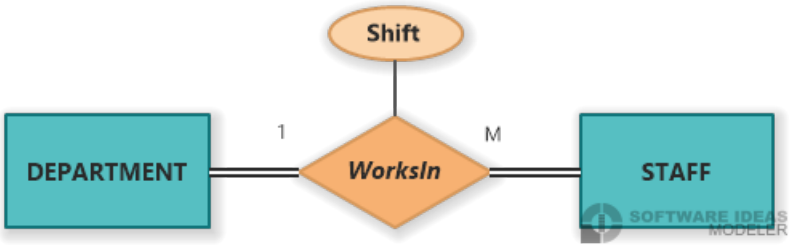
6.2.8 STAFF *oversees* STAFF



It is a 1:M relationship, as a manager oversees many staff and staff members report to only one manager.

We note partial participation for both supervisor and supervisee, as not all staff members are managers and not all managers have a manager.

6.2.9 STAFF *worksin* DEPARTMENT



It is a 1:M relationship, as a staff member can work for only one department and a department can have many staff members.

We note total participation for both entities, as a staff worker must be assigned to a department and a department must have staff.

The relationship includes an attribute shift to indicate which crew the staff is in (day or night shift).

6.2.10 STAFF *manages* DEPARTMENT

It is a 1:1 relationship, as a department can only have one manager.

We note total participation for the DEPARTMENT entity, as a department must have a manager, and partial participation for the STAFF entity, as not all staff are managers.

6.2.11 FLIGHT *takesofffrom* AIRPORT

It is a 1:M relationship, as a flight takes off from one airport and an airport has many flights take off from it.

We note total participation for both as every flight needs to take off from an airport, and every airport needs to have a flight take off from it.

6.2.12 STORE *locatedat* AIRPORT

It is a 1:M relationship, as each branch of this franchise can be at one airport, and every airport can have many Stores. We note total participation for both as every airport needs to have a store, and every store must be in an airport.

7 ER to Relational Data Model Mapping

In this part of the report, we translate our ER Diagram into schema constructs, as per the 7 step algorithm.

7.1 Mapping of Strong Entity Types

In this step, every entity in our ER model will be transformed into a relation composed of its simple attributes. Each relation must be composed of a unique key called primary key. In the case of a composite key, the relation will include the simple attributes composing this key as a primary key (they will be all underlined). Likewise, each composite attribute will be included in the relation as the simple attributes composing it. We will discuss multivalued in Step 6.

7.1.1 Client Entity

CustomerID	FName	MName	LName	PassportNum	PhoneNumber	HomeAddress	Email	Gender

The client relation includes the nine attributes provided with it, like their identification number (denoted CustomerID) which we underline as our key, name, passport number (denoted PassportNum), phone number, home address, email and gender.
It is worth noting that the client entity includes another attribute, nationality, which is not present here as it is a multivalued attribute. Please check section 7.6.1 for more information.

7.1.2 Staff Entity

EmployeeID	Name	DOB	Nationality	Salary	Rewardpoints	StartDate

The staff relation contains seven simple attributes: their identification number (denoted EmployeeID) which we underline as our key, name (which is a composite attribute), date of birth (denoted DOB), nationality, salary, reward points accumulated, and the date they began their service with Fly Haraty.

7.1.3 Ticket Entity

ReservationID	SeatNo	Class	Meal	InFlightInsurance	LuggageWeight

This entity consists of seven simple attributes: the reservation ID of the ticket, seat number (denoted SeatNo), class, meal, in-flight insurance coverage, and luggage weight.

7.1.4 Flight Entity

PlaneNum	Date	DepartTime	ArrivalTime	DepartFrom	Destination

This relation consists of 3 simple attributes, the location of departure, location of arrival, and the arrival time, and 1 composite attribute which is our key that contains 3 simple attributes: plane number (denoted PlaneNum), date of the flight, and departure time.

7.1.5 Hotel Room Entity

HotelName	RoomNum	Roomtype

This relation consists of 1 simple attribute, the class of the room reserved, and 1 composite attribute which is underlined as the key of the relation, which is the name of the hotel and room number.

7.1.6 Transport Entity

LicensePlate	Model/Make	Capacity	PassengerRequests	PickupDate	PickupTime	PickupPoint	DropoffPoint

This relation consists of 1 composite attribute, which is composed of two simple attributes, the pickup and drop off point for the client, and 6 simple attributes, the license plate of the car (which is set as our key), the model/make of the car, its capacity, the date and time of the pickup, in addition to requests made by the passenger.

7.1.7 Department Entity

DepName	Country	EmailAddress	PhoneExtension	Budget

This relation consists of 1 composite attribute (which is set as our key), which is composed of the country and name of the department (denoted DepName), the email address of the department, the phone extension, and the yearly budget of the employees.

7.1.8 Airport Entity

AirportName	ACountry	City	Capacity	NumGates	NumTerminals	PhoneNumber	WebPage

This relation consists of 6 simple attributes, the name of the airport (which we set as our key), the number of terminals (denoted as NumTerminals), the number of gates (denoted as NumGates), the capacity of the airport, phone number, and the webpage of the airport, and 1 composite attribute that contains 2 simple attributes, the city and country of the airport.

7.1.9 Insurance Entity

InsuranceID	PolicyClass	StartDate	RenewalDate

This relations consists of 4 simple attributes, the insurance id (which we set as our key), the policy class, the day the insurance kicks in, and the renewal date.

7.2 Mapping of Weak Entity Types

7.2.1 Store Entity

StoreName	**AirportName**	**Type**	**ProfitSplit**	**PartnershipStartDate**

The Store entity consists of 4 simple attributes, having Name as the partial key. Transforming it into the relational entity, the primary key will be the partial key (StoreName) concatenated with the Primary key of the identifying entity (AirportName).

7.3 Mapping of Binary 1:1 Relationship Types

In our database we map our 1:1 relationships using the foreign key approach.

7.3.1 *Arecoveredby* Relationship

EmployeeID	**Name**	**DOB**	**Nationality**	**Salary**	**Rewardpoints**	**StartDate**	**InsuranceNumber**

Since we have total participation for both staff and insurance entities, we are free to choose either of the entities to have the foreign key of the other. We decided that it is better to have the primary key of Insurance (InsuranceNumber) as the foreign key in the staff relation.

7.3.2 *Manages* Relationship

DepName	**Country**	**EmailAddress**	**PhoneExtension**	**Budget**	**ManagerID**

In this 1:1 relation, we have a total participation at the side of department, and partial participation at the staff side. Thus, we include the primary key of Staff (EmployeeID) as a foreign key in the department relationship. To illustrate, for each department, we have one manager. Thus, a department would include the ID of the manager in its relation

7.4 Mapping of Binary 1:N Relationship Types

7.4.1 *purchases* Relationship

ReservationID	SeatNo	Class	Meal	InFlightInsurance	LuggageWeight	PassengerID	Cost

Since a client can purchase many tickets, we include the CustomerID as a foreign key in Ticket relation, where a ticket is assigned to a single customer. We also add the cost attribute of the relationship into the ticket relation.

7.4.2 *reserves* Relationship

HotelName	RoomNum	Roomtype	PsngrID	CheckInDate	CheckoutDate	Roomcost

Since a client can reserve many hotel rooms, we place the primary key of the client relation as a foreign key in the hotel room relation. We also add the attributes of the reserves relationship (check in date, check out date, room cost).

7.4.3 *drives* Relationship

LicensePlate	Model/Make	Capacity	PassengerRequests	PickupDate	PickupTime	PickupPoint	DropoffPoint
DriverID							

In our model, a staff member can drive many transport cars, so the transport relation includes the primary key of the staff relation (DriverID) as the foreign key in it.

7.4.4 *boards* Relationship

CustomerID	FName	MName	LName	PassportNum	PhoneNumber	HomeAddress	Email	Gender
TerminalNum	GateNum	PNum	FltDate	Dtime				

Since a flight is boarded by many clients we add the primary key of the flight relation (consists of three attributes, PlaneNum, Date, and DepartTime) as a foreign key in the client relation. We also add the attributes of the boards relationship (TerminalNum, GateNum) to the client relation.

7.4.5 *takesofffrom* Relationship

PlaneNum	Date	DepartTime	ArrivalTime	DepartFrom	Destination
ArptName					

Since an airport has many flights take off from it, we add the primary key of the airport relation as the foreign key of the flight relation.

7.4.6 *oversees* Relationship

EmployeeID	Name	DOB	Nationality	Salary	Rewardpoints	StartDate	InsuranceNumber
SupID							

Since it is a self-referencing relationship, we add the primary key of the Staff relation (EmployeeID) in this same relation as AdministratorID, where this foreign key represents the supervisor of some staff member.

7.4.7 *worksin* Relationship

EmployeeID	Name	DOB	Nationality	Salary	Rewardpoints	StartDate	InsuranceNumber
SupID	Shift	Dcountry	DName				

Since a department can have many staff members working for it, we include the DepID as a foreign key in the Staff relation (Where each staff member will have the Department name and country they are working in). Moreover, we add the shift attribute of Worksin relationship into the client relation.

7.5 Mapping of Binary M:N Relationship Types

In this mapping type, we create a new relation that includes the primary key of the first relation combined with the primary key of the second relation as our new primary key, then we add any attributes of the relationship into the new relation we created.

7.5.1 *includes* Relationship

ResID	**PlateNum**

This new relation contains the LicensePlate (primary key of Transport, denoted ResID) and the ReservationID (primary key of Ticket) to be the Primary key of this relation. Thus, a unique key will be formed (having the license plate concatenated with the ticket number).

7.5.2 *operates* Relationship

OperatorID	**PlaneNo**	**FDate**	**DepTime**	**Role**

We combine here the flight number with the employee's ID to get a unique value, representing the new relation's primary key. We also add the role attribute to this relation.

7.6 Mapping of Multivalued Attributes

7.6.1 Nationality Multivalued Attribute

CustomerID	**Nationalty**

A client's nationality is a multivalued attribute in the client entity. For this reason, it is mapped as a relation, having the client's identification number (denoted ClientID) and their corresponding nationality.

7.7 Mapping of N-ary Relationship Types

Since we have no n-ary relationships, this step is skipped.

8 Creating Our Database

To create and query our database, we use an edition of Oracle 11g.

To avoid logical errors in Oracle, we created the tables first, and then later on added the foreign key constraints.

However, for clarity and comfort, we included the foreign key statements directly in each SQL statement. Ignoring the errors that are inherently present by directly placing them there.

8.1 Client Relation

```
CREATE TABLE Client (
    CustomerID INT NOT NULL,
    Fname VARCHAR2(20) NOT NULL,
    Mname VARCHAR2(20) NOT NULL,
    Lname VARCHAR2(20) NOT NULL,
    PassportNum VARCHAR2(15) NOT NULL,
    PhoneNumber VARCHAR2(10),
    HomeAddress VARCHAR2(150),
    Email VARCHAR2(320) NOT NULL,
    Gender CHAR(1) NOT NULL,
    TerminalNum INT NOT NULL,
    GateNum VARCHAR2(2) NOT NULL,
    PNum VARCHAR2(20),
    FltDate DATE,
    DTime TIMESTAMP,
    PRIMARY KEY (CustomerID),
    FOREIGN KEY (PNum, FltDate, DTime)
    REFERENCES Flight (PLANENUM,FLIGHTDATE,DEPARTTIME)
);
```

8.2 Ticket Relation

```
CREATE TABLE Ticket (
    ReservationID INT NOT NULL,
    SeatNo VARCHAR(10) NOT NULL,
    Class VARCHAR(10) NOT NULL,
    Meal VARCHAR(20),
    InFlightInsurance VARCHAR(10) NOT NULL,
    LuggageWeight INT NOT NULL,
    PassengerID INT NOT NULL,
    Cost INT NOT NULL,
    PRIMARY KEY (ReservationID),
    FOREIGN KEY (PassengerID)
    REFERENCES Client(CustomerID)
);
```

8.3 Hotel Room Relation

```
CREATE TABLE HotelRoom (
    HotelName VARCHAR(20) NOT NULL,
    RoomNum INT NOT NULL,
    RoomType VARCHAR(10) NOT NULL,
    PsngrID INT NOT NULL,
    CheckInDate DATE NOT NULL,
    CheckoutDate DATE NOT NULL,
    RoomCost INT NOT NULL,
    PRIMARY KEY(HotelName),
    FOREIGN KEY (PSNGRID)
    REFERENCES CLIENT (CustomerID)
);
```

8.4 Flight Relation

```
CREATE TABLE Flight (
    PlaneNum VARCHAR(10) NOT NULL,
    FlightDate DATE NOT NULL,
    DepartTime TIMESTAMP NOT NULL,
    ArrivalTime TIMESTAMP NOT NULL,
    DepartFrom VARCHAR(50) NOT NULL,
    Destination VARCHAR(50) NOT NULL,
    ArptName VARCHAR(50) NOT NULL,
    PRIMARY KEY (PlaneNum,FlightDate,DepartTime),
    FOREIGN KEY (ArptName)
    REFERENCES Airport(AirportName);
);
```

8.5 Transport Relation

```
CREATE TABLE Transport (
    LicensePlate VARCHAR(15) NOT NULL,
    ModelMake VARCHAR(50) NOT NULL,
    Capacity INT NOT NULL,
    PassengerRequests VARCHAR(100),
    PickupDate DATE NOT NULL,
    PickupTime TIMESTAMP NOT NULL,
    PickupPoint VARCHAR(50) NOT NULL,
    DropoffPoint VARCHAR(50) NOT NULL,
    DriverID INT NOT NULL,
    PRIMARY KEY (LicensePlate),
    FOREIGN KEY (DriverID)
    REFERENCES Staff(EmployeeID);
);
```

8.6 Staff Relation

```
CREATE TABLE Staff (
    EmployeeID INT NOT NULL,
    Name VARCHAR(15) NOT NULL,
    DOB DATE NOT NULL,
    Nationality VARCHAR(10) NOT NULL,
    Salary INT NOT NULL,
    Rewardpoints INT,
    StartDate DATE NOT NULL,
    InsuranceNumber VARCHAR(15),
    SupID INT NOT NULL,
    Shift VARCHAR(10) NOT NULL,
    Dcountry VARCHAR(15) NOT NULL,
    DName VARCHAR(15) NOT NULL,

    PRIMARY KEY (EmployeeID),

    FOREIGN KEY (SupID)
    REFERENCES Staff(EmployeeID),

    FOREIGN KEY (Dcountry, DName)
    REFERENCES Department(Country, DepName),

    FOREIGN KEY (InsuranceNumber)
    REFERENCES Insurance(InsuranceID)
);
```

8.7 Insurance Relation

```
CREATE TABLE Insurance (
    InsuranceID VARCHAR(30) NOT NULL,
    PolicyClass VARCHAR(30) NOT NULL,
    StartDate DATE NOT NULL,
    RenewalDate DATE NOT NULL,
    PRIMARY KEY (InsuranceID)
);
```

8.8 Department Relation

```
CREATE TABLE Department (
    DepName VARCHAR(15) NOT NULL,
    Country VARCHAR(30) NOT NULL,
    EmailAddress VARCHAR(320),
    PhoneExtension VARCHAR(10) NOT NULL,
    Budget INT NOT NULL,
    ManagerID INT NOT NULL,
    PRIMARY KEY (DepName, Country),
    FOREIGN KEY (ManagerID)
    REFERENCES Staff(EmployeeID);
);
```

8.9 Airport Relation

```
CREATE TABLE Airport (
    AirportName VARCHAR(50) NOT NULL,
    ACountry VARCHAR(30) NOT NULL,
    City VARCHAR(20) NOT NULL,
    Capacity INT NOT NULL,
    NumGates INT NOT NULL,
    NumTerminals INT NOT NULL,
    PhoneNumber VARCHAR(20),
    WebPage VARCHAR(100) NOT NULL,
    PRIMARY KEY (AirportName)
);
```

8.10 Store Relation

```
CREATE TABLE Store (
    StoreName VARCHAR(35) NOT NULL,
    AName VARCHAR(50) NOT NULL,
    Type VARCHAR(20) NOT NULL,
    ProfitSplit INT NOT NULL,
    PartnershipStartDate DATE NOT NULL,
    PRIMARY KEY (StoreName, AName),
    FOREIGN KEY (AName) REFERENCES AIRPORT(AirportName)
);
```

8.11 Includes Relation

```
CREATE TABLE Includes (
    ResID INT NOT NULL,
    PlateNum VARCHAR(15) NOT NULL,
    PRIMARY KEY (ResID, PlateNum),
    FOREIGN KEY (ResID) REFERENCES Ticket(ReservationID),
    FOREIGN KEY (PlateNum) REFERENCES Transport(LicensePlate)
);
```

8.12 Operates Relation

```
CREATE TABLE OPERATES (
    OperatorID INT NOT NULL,
    PlaneNo VARCHAR2(50) NOT NULL,
    Fdate DATE NOT NULL,
    DepTime TIMESTAMP NOT NULL,
    Role VARCHAR(20) NOT NULL,
    PRIMARY KEY (OperatorID, PlaneNo, Fdate, DepTime),
    FOREIGN KEY (OperatorID) REFERENCES Staff(EmployeeID),
    FOREIGN KEY (PlaneNo, Fdate, DepTime)
    REFERENCES Flight(PLANENUM, FLIGHTDATE, DEPARTTIME)
);
```

8.13 Nationality Relation

```
CREATE TABLE Nationality (
    CID INT NOT NULL,
    CusNationalty VARCHAR(15),
    PRIMARY KEY (CID, CusNationality)
);
```

9 Insertion into Tables

In order to avoid the software throwing errors, we performed the insert statements in a specific order. Additionally, we had to disable the foreign key constraints for 2 tables, since they reference each other, and re-enabled them after inserting them.

9.1 Airport Insertion

```
INSERT ALL
INTO AIRPORT VALUES ('Haraty International Airport','Lebanon',
'Beirut',5000000,7,2,'01596922','www.FlyHaraty.com.lb')
INTO AIRPORT VALUES ('SkyHarbor International Airport - New York',
'United States','New York', 10000000 , 12 , 3,'555-692-375',
'www.flyharatynewyork.com')
INTO AIRPORT VALUES ('Queen Bee International Airport - Manchester',
'United Kingdom','Manchester',10000000,12,3,'(0161) 111 1111',
'www.flyharatymanchester.com')
INTO AIRPORT VALUES ('Maple Leaf International Airport','Canada',
'Ontario',12000000,12,3,'+1 604 12345678',
'www.flyharatyontario.com.lb')
INTO AIRPORT VALUES ('Jagersdorf International Airport',
'Germany','Hamburg', 12000000 , 15 , 5,'+49 40 12345678',
'www.flyharatyhamburg.com')
SELECT * FROM dual;
```

9.2 Store Insertion

```
INSERT ALL
INTO STORE VALUES ('Noreen''s Corner', 'Haraty International Airport',
'Burger Joint', 50, TO_DATE('2014-11-11', 'YYYY-MM-DD'))
INTO STORE VALUES ('Ahmad''s Botique',
'SkyHarbor International Airport - New York', 'Shoe Retail', 50,
 TO_DATE('2012-10-31', 'YYYY-MM-DD'))
INTO STORE VALUES ('Burger King',
'Queen Bee International Airport - Manchester',
 'Fast Food', 60,
 TO_DATE('2010-05-05', 'YYYY-MM-DD'))
INTO STORE VALUES ('BBQ House','Maple Leaf International Airport',
'Restuarant',30,
TO_DATE('2015-04-12', 'YYYY-MM-DD'))
INTO STORE VALUES ('Patchi','Jagersdorf International Airport',
'Chocolate Shop',40,
TO_DATE('2010-07-09','YYYY-MM-DD'))
INTO STORE VALUES ('Bershka','Jagersdorf International Airport',
'Clothing Retail',35,
TO_DATE('2016-04-28','YYYY-MM-DD'))
INTO STORE VALUES ('Haraty caffe','Haraty International Airport',
'Caffe Shop',70,
TO_DATE('2010-09-05','YYYY-MM-DD'))
INTO STORE VALUES ('Klein','Maple Leaf International Airport','Mini Market',
40,TO_DATE('2013-09-03','YYYY-MM-DD'))
INTO STORE VALUES ('TakeGrill','SkyHarbor International Airport - New York',
'Sandwich Shop',35,
TO_DATE('2014-06-15','YYYY-MM-DD'))
INTO STORE VALUES ('Gifto','Queen Bee International Airport - Manchester',
 'Gift Shop',25,
TO_DATE('2017-02-06','YYYY-MM-DD'))
INTO STORE VALUES ('Flytech','Queen Bee International Airport - Manchester',
'Electronic stores',40,
TO_DATE('2013-10-14','YYYY-MM-DD'))
INTO STORE VALUES ('japanese food',
'SkyHarbor International Airport - New York','noodles store',50,
TO_DATE('2012-08-03','YYYY-MM-DD'))

SELECT * FROM dual;
```

9.3 Flight Insertion

```
INSERT ALL
INTO FLIGHT VALUES ('FH123', TO_DATE('2023-03-11', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-03-11 09:00:00', 'yyyy-mm-dd hh24:mi:ss'),
TO_TIMESTAMP('2023-03-12 00:00:01', 'yyyy-mm-dd hh24:mi:ss'), 'Manchester',
'Beirut', 'Queen Bee International Airport - Manchester')
INTO FLIGHT VALUES ('FH456', TO_DATE('2023-06-12', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-06-12 06:30:00', 'yyyy-mm-dd hh24:mi:ss'),
TO_TIMESTAMP('2023-06-12 13:00:00', 'yyyy-mm-dd hh24:mi:ss'),
'Hamburg', 'Ontario', 'Jagersdorf International Airport')
INTO FLIGHT VALUES ('FH726',
TO_DATE('2023-07-25', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-07-25 21:15:00', 'yyyy-mm-dd hh24:mi:ss'),
TO_TIMESTAMP('2023-07-26 04:00:00', 'yyyy-mm-dd hh24:mi:ss'),
'Beirut', 'Hamburg', 'Haraty International Airport')
INTO FLIGHT VALUES ('FH679', TO_DATE('2023-11-11', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-11-11 11:00:00', 'yyyy-mm-dd hh24:mi:ss'),
TO_TIMESTAMP('2023-11-11 21:30:00', 'yyyy-mm-dd hh24:mi:ss'),
'Beirut', 'Manchester', 'Haraty International Airport')
INTO FLIGHT VALUES ('FH629', TO_DATE('2023-11-11', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-11-11 11:00:00', 'yyyy-mm-dd hh24:mi:ss'),
TO_TIMESTAMP('2023-11-11 18:00:00', 'yyyy-mm-dd hh24:mi:ss'),
'Ontario', 'Newyork', 'Maple Leaf International Airport
')
SELECT * FROM dual;
```

9.4 Client Insertion

```
INSERT ALL
INTO CLIENT VALUES (202301348,'Hadi','Wissam','Al Hassan','LR-1654321',
'81769033','Anderson Street apartment 4','Hadi.alhassan@gmail.com','M',1,
'A1','FH123',TO_DATE('2023-03-11','YYYY-MM-DD'),
TO_TIMESTAMP('2023-03-11 09:00:00', 'yyyy-mm-dd hh24:mi:ss'))
INTO CLIENT VALUES (202301349,'Sari','Mahmoud','Abdul Ghani','LR-6543684',
'76500432','Saida Apartment 3','Sari.Aghani@gmail.com','M',1,'A1','FH123',
TO_DATE('2023-03-11','YYYY-MM-DD'),
TO_TIMESTAMP('2023-03-11 09:00:00', 'yyyy-mm-dd hh24:mi:ss'))
INTO CLIENT VALUES (202301390,'Siraj','Imad','Al Mabsout','LR-5368967',
'78811997','Cola street zhairi building apartment 6','Siraj.Mabsout@gmail.com',
'M',2,
'B2','FH456',
TO_DATE('2023-06-12','YYYY-MM-DD'),
TO_TIMESTAMP('2023-06-12 06:30:00','yyyy-mm-dd hh24:mi:ss'))
INTO CLIENT VALUES (202301391,'Gaelle','Elie','Loutfi','LR-1256546',
'78043235','kora street garden building apartment 5',
'Gaelle.Loutfi@gmail.com','F',2,'B2','FH456',
TO_DATE('2023-06-12','YYYY-MM-DD'),
TO_TIMESTAMP('2023-06-12 06:30:00', 'yyyy-mm-dd hh24:mi:ss'))
INTO CLIENT VALUES (202301432,'Ahmad','Ramzi','Haraty','F95094567',
'03464635','Chouf street borj building apartment 3','Ahmad.Haraty@outlook.com','M',2,
'C1','FH726',
TO_DATE('2023-07-25', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-07-25 21:15:00', 'yyyy-mm-dd hh24:mi:ss'))
INTO CLIENT VALUES (202301433,'Noreen','Ramzi','Haraty','123654798',
'03252648','Chouf street borj building apartment 3','Noreen.Haraty@outlook.com','F',
4,'C1','FH726',TO_DATE('2023-07-25', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-07-25 21:15:00', 'yyyy-mm-dd hh24:mi:ss'))
TO_DATE('2023-07-25', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-07-25 21:15:00', 'yyyy-mm-dd hh24:mi:ss'))
INTO CLIENT VALUES (202301502,'Ramez','Ali','ElMasri','567483903',
'71408321','kaslik street baraket building apartment 7','Ramez.Masri@outlook.com',
'M',1,'D4','FH679',
TO_DATE('2023-11-11','yyyy-mm-dd'),
TO_TIMESTAMP('2023-11-11 11:00:00','yyyy-mm-dd hh24:mi:ss'))
SELECT * FROM dual;
```

Client Insertion (Contd)

```
INSERT ALL
INTO CLIENT VALUES (202301503,'Wael','Emily','Kfouri','397856023',
'70895654','Tripoli street hazem building apartment 7','Wael.Kfouri@hotmail.com',
'M',3,'D3','FH679', TO_DATE('2023-11-11','yyyy-mm-dd'),
TO_TIMESTAMP('2023-11-11 11:00:00','yyyy-mm-dd hh24:mi:ss'))
INTO CLIENT VALUES (202301520,'Bob','Johny','Anderson','365748350',
'81450760','Saida street ramez Apartment 7','Bob.A@gmail.com','M',3,'D4','FH629',
TO_DATE('2023-11-11', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-11-11 11:00:00', 'yyyy-mm-dd hh24:mi:ss'))

INTO CLIENT VALUES (202301521,'Marshal','ken','Mathers','239758402',
'03765498','Sour street majid Apartment 6','Marshal.Math@outlook.com','M',3,'A2',
'FH629',
TO_DATE('2023-11-11', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-11-11 11:00:00', 'yyyy-mm-dd hh24:mi:ss'))
SELECT * FROM dual;
```

9.5 Hotelroom Insertion

```
INSERT ALL
INTO HOTELROOM VALUES ('Gefinor Rotana-Beirut', 1004, 'Suite', 202301348,
TO_DATE('2023-03-11', 'YYYY-MM-DD'),
 TO_DATE('2023-03-12', 'YYYY-MM-DD'), 600)
INTO HOTELROOM VALUES ('Radinsson Blu Martinez-Beirut', 702, 'Deluxe', 202301349,
TO_DATE('2023-03-11', 'YYYY-MM-DD'),
 TO_DATE('2023-03-12', 'YYYY-MM-DD'), 300)
INTO HOTELROOM VALUES ('The Lowell-New York', 804, 'Deluxe', 202301520,
TO_DATE('2023-11-11', 'YYYY-MM-DD'),
 TO_DATE('2023-11-12', 'YYYY-MM-DD'), 450)
INTO HOTELROOM VALUES ('The William Vale-New York', 1204, 'Suite', 202301521,
TO_DATE('2023-11-11', 'YYYY-MM-DD'),
 TO_DATE('2023-11-12', 'YYYY-MM-DD'), 650)
INTO HOTELROOM VALUES ('The Midland Hotel-Manchester', 700, 'Deluxe', 202301502,
TO_DATE('2023-11-11', 'YYYY-MM-DD'),
 TO_DATE('2023-11-12', 'YYYY-MM-DD'), 400)
INTO HOTELROOM VALUES ('Hotel Gotham-Manchester', 103, 'Standard', 202301503,
TO_DATE('2023-11-11', 'YYYY-MM-DD'),
 TO_DATE('2023-11-12', 'YYYY-MM-DD'), 100)
INTO HOTELROOM VALUES ('Four Season Hotel-Toronto', 703, 'Deluxe', 202301390,
TO_DATE('2023-06-12', 'YYYY-MM-DD'),
 TO_DATE('2023-06-13', 'YYYY-MM-DD'), 400)
INTO HOTELROOM VALUES ('Fairmont Royal York-Toronto', 209, 'Standard', 202301391,
TO_DATE('2023-06-12', 'YYYY-MM-DD'),
 TO_DATE('2023-06-13', 'YYYY-MM-DD'), 150)
INTO HOTELROOM VALUES ('The Ritz-Carlton-Hamburg', 1010, 'Suite', 202301432,
TO_DATE('2023-07-26', 'YYYY-MM-DD'),
 TO_DATE('2023-07-27', 'YYYY-MM-DD'), 700)
INTO HOTELROOM VALUES ('Hotel Adlon Kempinski-Hamburg', 303, 'Standard', 202301433,
TO_DATE('2023-07-26', 'YYYY-MM-DD'),
 TO_DATE('2023-07-27', 'YYYY-MM-DD'), 50)
SELECT * FROM dual;
```

9.6 Ticket Insertion

```
INSERT ALL
INTO TICKET VALUES(1001, 25, 'Economy', 'Caesar Salad',
 'Class C', 40, 202301348, 350)
INTO TICKET VALUES(2002, 23, 'Economy', 'Caesar Salad',
 'Class C', 40, 202301349, 350)
INTO TICKET VALUES(3003, 10, 'Business', 'Gourmet Seafood',
 'Class B', 55, 202301390, 750)
INTO TICKET VALUES(4004, 12, 'Business', 'Rice and Meat',
 'Class B', 55, 202301391, 750)
INTO TICKET VALUES(5005, 4, 'First', 'Omelet with sausage',
 'Class A', 70, 202301432, 1500)
INTO TICKET VALUES(6006, 5, 'First', 'Muffins',
 'Class A', 70, 202301433, 1500)
INTO TICKET VALUES(7007, 15, 'Business', 'Vegetable Salad',
'Class B', 55, 202301502, 750)
INTO TICKET VALUES(8008, 3, 'First', 'Granola bars',
 'Class A', 70, 202301503, 1500)
INTO TICKET VALUES(9009, 27, 'Economy', 'Cheese and Crackers',
 'Class C', 40, 202301520, 350)
INTO TICKET VALUES(10001, 24, 'Economy', 'Pasta and Meat',
'Class C', 40, 202301521, 350)

SELECT * FROM dual;
```

9.7 Insurance Insertion

```
INSERT ALL
INTO INSURANCE VALUES ('859314267', 'ClassA',
 TO_DATE('2023-01-01', 'YYYY-MM-DD'), TO_DATE('2024-05-01', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('426718935', 'ClassB',
 TO_DATE('2022-05-15', 'YYYY-MM-DD'), TO_DATE('2025-03-15', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('973512468', 'ClassC',
 TO_DATE('2023-08-20', 'YYYY-MM-DD'), TO_DATE('2024-12-20', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('631845279', 'ClassA',
 TO_DATE('2022-11-10', 'YYYY-MM-DD'), TO_DATE('2026-11-10', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('248697513', 'ClassB',
 TO_DATE('2023-03-05', 'YYYY-MM-DD'), TO_DATE('2024-09-05', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('512384679', 'ClassC',
 TO_DATE('2022-07-18', 'YYYY-MM-DD'), TO_DATE('2025-11-18', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('394752186', 'ClassA',
 TO_DATE('2023-02-28', 'YYYY-MM-DD'), TO_DATE('2024-08-28', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('827163594', 'ClassB',
 TO_DATE('2022-09-12', 'YYYY-MM-DD'), TO_DATE('2026-12-12', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('649218753', 'ClassC',
 TO_DATE('2023-06-25', 'YYYY-MM-DD'), TO_DATE('2024-06-25', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('175936842', 'ClassA',
 TO_DATE('2022-12-15', 'YYYY-MM-DD'), TO_DATE('2025-12-15', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('540529328', 'ClassC',
 TO_DATE('2022-09-14', 'YYYY-MM-DD'), TO_DATE('2023-07-15', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('685930503', 'ClassC',
 TO_DATE('2022-06-18', 'YYYY-MM-DD'), TO_DATE('2024-12-15', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('329582914', 'ClassC',
 TO_DATE('2022-04-03', 'YYYY-MM-DD'), TO_DATE('2025-11-07', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('594922138', 'ClassC',
 TO_DATE('2022-12-23', 'YYYY-MM-DD'), TO_DATE('2023-07-04', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('796934029', 'ClassC',
 TO_DATE('2022-03-27', 'YYYY-MM-DD'), TO_DATE('2024-04-21', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('583970383', 'ClassC',
 TO_DATE('2022-04-20', 'YYYY-MM-DD'), TO_DATE('2024-06-10', 'YYYY-MM-DD'))
SELECT * FROM dual;
```

Insurance Insertion (Contd)

```
INSERT ALL
INTO INSURANCE VALUES ('124968324', 'ClassC',
 TO_DATE('2022-07-12', 'YYYY-MM-DD'), TO_DATE('2025-07-10', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('968393834', 'ClassC',
 TO_DATE('2022-03-05', 'YYYY-MM-DD'), TO_DATE('2024-06-07', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('867958330', 'ClassC',
 TO_DATE('2022-02-07', 'YYYY-MM-DD'), TO_DATE('2023-11-17', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('348875898', 'ClassC',
 TO_DATE('2022-08-04', 'YYYY-MM-DD'), TO_DATE('2024-09-13', 'YYYY-MM-DD'))
INTO INSURANCE VALUES ('233982939', 'ClassC',
 TO_DATE('2022-11-21', 'YYYY-MM-DD'), TO_DATE('2024-03-21', 'YYYY-MM-DD'))
 SELECT * FROM dual;
```

9.8 Department Insertion

```
INSERT ALL
INTO DEPARTMENT VALUES ('Flight Operations','Lebanon',
'flightops@flyharaty.com.lb','101',750000,1)
INTO DEPARTMENT VALUES ('Finance','Lebanon',
'finance@flyharaty.com.lb','103',500000,2)
INTO DEPARTMENT VALUES ('Flight Operations','Germany',
'@flightopsGRlyharaty.com','111',900000,6)
INTO DEPARTMENT VALUES ('Finance','Canada',
'finance@flyharaty.com.fr','113',900000,10)
INTO DEPARTMENT VALUES ('Customer Care','United Kingdom',
'customercareUK@flyharaty.com','114',450000,4)
INTO DEPARTMENT VALUES ('Maintenance','United States',
'maintenanceUS@flyharaty.com.','122',1200000,5)
INTO DEPARTMENT VALUES ('Flight Operations','Canada',
'flightopsCA@flyharaty.com.lb','922',750000,11)
INTO DEPARTMENT VALUES ('Flight Operations','United Kingdom',
'flightopsUK@flyharaty.com.lb','666',750000,15)
INTO DEPARTMENT VALUES ('Flight Operations','United States',
'flightopsUS@flyharaty.com.lb','711',750000,12)

SELECT * FROM dual;
```

We should note that, due to a circular dependency in the Staff and Department tables, we had to disable the foreign key constraints in both tables, then re-enable them again. We will probably have to address this is in the coming phase.

9.9 Staff Insertion

```
INSERT ALL
INTO STAFF VALUES (1,'Ramzi R Haraty',TO_DATE('1961-10-01','yyyy-mm-dd'),
'Lebanese',10000,1000,
TO_DATE('1986-01-23','yyyy-mm-dd'),'859314267',NULL,'AM','Lebanon','Flight Operations')
INTO STAFF VALUES (2,'Joseph Rebehmed',TO_DATE('1980-06-04','yyyy-mm-dd'),
'Lebanese',8000,400,TO_DATE('1998-03-12','yyyy-mm-dd'),'426718935',NULL,'AM','Lebanon',
'Finance')
INTO STAFF VALUES (3,'Faisal Abu Khzam',
TO_DATE('1968-05-08','yyyy-mm-dd'),'American',5000,0,
TO_DATE('2002-04-09','yyyy-mm-dd'),'973512468',1,'PM','Lebanon','Flight Operations')
INTO STAFF VALUES (4,'Nadine Abbas',
TO_DATE('1996-11-05','yyyy-mm-dd'),'Saudi',5000,500,
TO_DATE('2011-11-24','yyyy-mm-dd'),'631845279',NULL,'PM','United Kingdom','Customer Care')
INTO STAFF VALUES (5,'Nader El Khatib',
TO_DATE('1984-04-07','yyyy-mm-dd'),'Lebanese',5000,800,
TO_DATE('2015-01-19','yyyy-mm-dd'),'248697513',NULL,'PM','United States','Maintenance')
INTO STAFF VALUES (6,'Ayoub Saab',
TO_DATE('1993-05-12','yyyy-mm-dd'),'Lebanese',5000,560,
TO_DATE('2012-03-18','yyyy-mm-dd'),'512384679',NULL,'AM','Germany','Flight Operations')
INTO STAFF VALUES (7,'Saul Goodman',
TO_DATE('1985-12-12','yyyy-mm-dd'),'American',9000,240,
TO_DATE('2012-05-14','yyyy-mm-dd'),'394752186',6,'PM','Germany','Flight Operations')
INTO STAFF VALUES (8,'Gordon Ramsay',
TO_DATE('1966-11-08','yyyy-mm-dd'),'British',6000,680,
TO_DATE('2004-09-10','yyyy-mm-dd'),'827163594',5,'PM','United States','Maintenance')
INTO STAFF VALUES (9,'Arthur Morgan',
TO_DATE('1994-05-23','yyyy-mm-dd'),'American',5500,450,
TO_DATE('2004-12-15','yyyy-mm-dd'),'649218753',2,'PM','Lebanon','Finance')
INTO STAFF VALUES (10,'Trevor Philips',
TO_DATE('1970-07-07','yyyy-mm-dd'),'Canadian',7000,430,
TO_DATE('2006-04-14','yyyy-mm-dd'),'175936842',NULL,'AM','Canada','Finance')
INTO STAFF VALUES (11,'Ramez Albilo',
TO_DATE('1987-11-11','YYYY-MM-DD'),'English',5000,500,
TO_DATE('2000-01-13','YYYY-MM-DD'),'540529328',NULL,'AM','Canada','Flight Operations')
INTO STAFF VALUES (12,'Mithad roge',
TO_DATE('1970-07-09','YYYY-MM-DD'),'American',6500,300,
TO_DATE('2010-01-17','YYYY-MM-DD'),'685930503',NULL,'AM',
'United States','Flight Operations')
SELECT * FROM dual;
```

Staff Insertion (Contd)

```
INSERT ALL
INTO STAFF VALUES (13,'Fareed Atrash',
TO_DATE('1990-06-04','YYYY-MM-DD'),'American',5000,500,
TO_DATE('2011-11-01','YYYY-MM-DD'),'329582914',12,'PM',
'United States','Flight Operations')
INTO STAFF VALUES (14,'George Svin',
TO_DATE('1996-09-23','YYYY-MM-DD'),'English',5000,750,
TO_DATE('2015-10-21','YYYY-MM-DD'),'594922138',11,'PM',
'Canada','Flight Operations')
INTO STAFF VALUES (15,'Kinda Halm',
TO_DATE('1987-11-11','YYYY-MM-DD'),'English',4000,700,
TO_DATE('2014-11-03','YYYY-MM-DD'),'796934029',NULL,'AM',
'United Kingdom','Flight Operations')
INTO STAFF VALUES (16,'Jad Karm',
TO_DATE('1977-03-18','YYYY-MM-DD'),'English',3000,500,
TO_DATE('2012-03-04','YYYY-MM-DD'),'583970383',15,'PM',
'United Kingdom','Flight Operations')
INTO STAFF VALUES (17, 'Majed Room',
 TO_DATE('1997-01-22', 'yyyy-mm-dd'), 'Lebanese', 3000, 300,
 TO_DATE('2012-06-13', 'yyyy-mm-dd'), '124968324', 1, 'AM',
 'Lebanon', 'Flight Operations')
INTO STAFF VALUES (18, 'Mazen Kolk',
 TO_DATE('1987-11-11', 'yyyy-mm-dd'), 'American', 4000, 450,
 TO_DATE('2011-03-10', 'yyyy-mm-dd'), '968393834', 12, 'AM',
 'United States', 'Flight Operations')
INTO STAFF VALUES (19, 'Darla Deen',
 TO_DATE('1990-04-22', 'yyyy-mm-dd'), 'English', 4000, 670,
  TO_DATE('2010-07-23', 'yyyy-mm-dd'), '867958330', 15, 'AM',
  'United Kingdom', 'Flight Operations')
INTO STAFF VALUES (20, 'Ramen Loon', TO_DATE('1987-03-04', 'yyyy-mm-dd'),
 'German', 3000, 400,
TO_DATE('2013-05-03', 'yyyy-mm-dd'), '348875898', 6, 'AM',
'Germany', 'Flight Operations')
INTO STAFF VALUES (21, 'Amjad Mahmoud',
 TO_DATE('1987-06-26', 'yyyy-mm-dd'), 'Lebanese', 2000, 300,
 TO_DATE('2000-06-02', 'yyyy-mm-dd'), '233982939', 11, 'AM',
  'Canada', 'Flight Operations')
SELECT * FROM dual;
```

9.10 Transport Insertion

```
INSERT ALL
INTO TRANSPORT VALUES ('G100404', 'Nissan Altima (2018)', 4,
'Please get me a coffee before pickup',
 TO_DATE('2023-03-12', 'YYYY-MM-DD'),
  TO_TIMESTAMP('2023-03-12 12:00:05', 'YYYY-MM-DD HH24:MI:SS'),
 'Haraty International Airport', 'Gefinor Rotana-Beirut', 3)
INTO TRANSPORT VALUES ('T208469', 'Honda Accord (2019)', 4,
'Please adjust the air conditioning to a comfortable temperature before pickup',
 TO_DATE('2023-03-12', 'YYYY-MM-DD'),
  TO_TIMESTAMP('2023-03-12 12:00:10', 'YYYY-MM-DD HH24:MI:SS'),
   'Haraty International Airport', 'Gefinor Rotana-Beirut', 3)

INTO TRANSPORT VALUES ('374ADH', 'Honda Accord (2017)', 4, NULL,
TO_DATE('2023-06-12', 'YYYY-MM-DD'),
 TO_TIMESTAMP('2023-06-12 13:00:00', 'YYYY-MM-DD HH24:MI:SS'),
 'Maple Leaf International Airport', 'Fairmont Royal York-Toronto', 14)

INTO TRANSPORT VALUES ('323ASH', 'Nissan Altima (2020)', 4, NULL,
 TO_DATE('2023-06-12', 'YYYY-MM-DD'),
  TO_TIMESTAMP('2023-06-12 13:00:00', 'YYYY-MM-DD HH24:MI:SS'),
  'Maple Leaf International Airport', 'Four Season Hotel-Toronto', 14)

INTO TRANSPORT VALUES ('CDZ-120', 'Honda Accord (2017)', 4, NULL,
 TO_DATE('2023-07-26', 'YYYY-MM-DD'),
  TO_TIMESTAMP('2023-07-26 4:00:00', 'YYYY-MM-DD HH24:MI:SS'),
  'Jagersdorf International Airport', 'The Ritz-Carlton-Hamburg', 7)

SELECT * FROM dual;
```

Transportation Insertion (Contd)

```
INSERT ALL
INTO TRANSPORT VALUES ('ARK-440', 'Chevrolet Impala (2020)', 4,

'Please get me a a sick bag before pickup
because I get nauseous on flights',
TO_DATE('2023-07-26', 'YYYY-MM-DD'),

 TO_TIMESTAMP('2023-07-26 4:00:00', 'YYYY-MM-DD HH24:MI:SS'),
 'Jagersdorf International Airport', 'Hotel Adlon Kempinski-Hamburg', 7)

INTO TRANSPORT VALUES ('BFD-220', 'Toyota Camry (2019)', 4, NULL,
TO_DATE('2023-11-11', 'YYYY-MM-DD'),
 TO_TIMESTAMP('2023-11-11 21:30:00', 'YYYY-MM-DD HH24:MI:SS'),
 'Queen Bee International Airport', 'Hotel Gotham-Manchester', 16)

INTO TRANSPORT VALUES ('LLR-320', 'Ford Fusion (2018)', 4, NULL,
TO_DATE('2023-11-11', 'YYYY-MM-DD'),
 TO_TIMESTAMP('2023-11-11 21:30:00', 'YYYY-MM-DD HH24:MI:SS'),
 'Queen Bee International Airport',
 'The Midland Hotel-Manchester', 16)

INTO TRANSPORT VALUES ('NY-9012', 'Nissan Altima (2018)', 4,
 'Please turn on seat warmers before pickup',
TO_DATE('2023-11-11', 'YYYY-MM-DD'),
 TO_TIMESTAMP('2023-11-11 18:00:00', 'YYYY-MM-DD HH24:MI:SS'),
 'SkyHarbor International Airport', 'The Lowell-New York', 13)

INTO TRANSPORT VALUES ('NY-3456', 'Toyota Camry (2019)', 4,
'Please get me a coffee before pickup',
 TO_DATE('2023-11-11', 'YYYY-MM-DD'),
  TO_TIMESTAMP('2023-11-11 18:00:00', 'YYYY-MM-DD HH24:MI:SS'),
'SkyHarbor International Airport', 'The William Vale-New York', 13)
SELECT * FROM dual;
```

9.11 Nationality Insertion

```
INSERT ALL
INTO NATIONALITY VALUES (202301348,'Palestinian')
INTO NATIONALITY VALUES (202301349,'Palestinian')
INTO NATIONALITY VALUES (202301349,'Ukranian')
INTO NATIONALITY VALUES (202301390,'Lebanese')
INTO NATIONALITY VALUES (202301391,'Lebanese')
INTO NATIONALITY VALUES (202301433,'Lebanese')
INTO NATIONALITY VALUES (202301502,'Lebanese')
INTO NATIONALITY VALUES (202301503,'Lebanese')
INTO NATIONALITY VALUES (202301432,'Lebanese')
INTO NATIONALITY VALUES (202301520,'American')
INTO NATIONALITY VALUES (202301521,'American')
SELECT * FROM dual;
```

9.12 Includes Insertion

```
INSERT ALL
INTO INCLUDES VALUES (1001, 'G100404')
INTO INCLUDES VALUES (2002, 'G100404')
INTO INCLUDES VALUES (3003, '374ADH')
INTO INCLUDES VALUES (4004, '323ASH')
INTO INCLUDES VALUES (5005, 'CDZ-120')
INTO INCLUDES VALUES (6006, 'ARK-440')
INTO INCLUDES VALUES (7007, 'BFD-220')
INTO INCLUDES VALUES (8008, 'BFD-220')
INTO INCLUDES VALUES (9009, 'NY-9012')
INTO INCLUDES VALUES (10001, 'NY-3456')
SELECT * FROM dual;
```

9.13 Operates Insertion

```
INSERT ALL
INTO OPERATES VALUES(1,'FH123', TO_DATE('2023-03-11', 'yyyy-mm-dd'),
 TO_TIMESTAMP('2023-03-11 09:00:00', 'yyyy-mm-dd hh24:mi:ss'),'Pilot')
INTO OPERATES VALUES(6,'FH726', TO_DATE('2023-07-25', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-07-25 21:15:00', 'yyyy-mm-dd hh24:mi:ss'),'Pilot')
INTO OPERATES VALUES(11,'FH456', TO_DATE('2023-06-12', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-06-12 06:30:00', 'yyyy-mm-dd hh24:mi:ss'),'Pilot')
INTO OPERATES VALUES(15,'FH679', TO_DATE('2023-11-11', 'yyyy-mm-dd'),
 TO_TIMESTAMP('2023-11-11 11:00:00', 'yyyy-mm-dd hh24:mi:ss'),'Pilot')
INTO OPERATES VALUES(12,'FH629', TO_DATE('2023-11-11', 'yyyy-mm-dd'),
TO_TIMESTAMP('2023-11-11 11:00:00', 'yyyy-mm-dd hh24:mi:ss'),'Pilot')
SELECT * FROM dual;
```

10 Snapshot of the Database

10.1 Client

FNAME	MNAME	LNAME	PASSPORTNUM	HONENUMBE	HOMEADDRESS	EMAIL	GENDER	RMINALNU	GATENUM	PNUM	FLTDATE	DTIME
Hadi	Wissam	Al Hassan	LR-1654321	81769033	Anderson Street apartment 4	Hadi.alhassan@gmail.com	M	1	A1	FH123	3/11/2023	11-MAR-23 09.00.00.000000 AM
Sari	Mahmoud	Abdul Ghani	LR-6543684	76500432	Saida Apartment 3	Sari.Aghani@gmail.com	M	1	A1	FH123	3/11/2023	11-MAR-23 09.00.00.000000 AM
Siraj	Imad	Al Mabsout	LR-5368967	78811997	Cola street zhairi building apartment 6	Siraj.Mabsout@gmail.com	M	2	B2	FH456	6/12/2023	12-JUN-23 06.30.00.000000 AM
Gaelle	Elie	Loutfi	LR-1256546	78043235	kora street garden building apartment 5	Gaelle.Loutfi@gmail.com	F	2	B2	FH456	6/12/2023	12-JUN-23 06.30.00.000000 AM
Noreen	Ramzi	Haraty	123654798	3252648	Chouf street borj building apartment 3	Noreen.Haraty@outlook.com	F	4	C1	FH726	7/25/2023	25-JUL-23 09.15.00.000000 PM
Ramez	Ali	ElMasri	567483903	71408321	kaslik street baraket building apartment 7	Ramez.Masri@outlook.com	M	1	D4	FH679	11/11/2023	11-NOV-23 11.00.00.000000 AM
Wael	Emily	Kfouri	397856023	70895654	Tripoli street hazem building apartment 7	Wael.Kfouri@hotmail.com	M	3	D3	FH679	11/11/2023	11-NOV-23 11.00.00.000000 AM
Ahmad	Ramzi	Haraty	395094567	3464635	Chouf street borj building apartment 3	Ahmad.Haraty@outlook.com	M	2	C1	FH726	7/25/2023	25-JUL-23 09.15.00.000000 PM
Bob	Johny	Anderson	365748350	81450760	Saida street ramez Apartment 7	Bob.A@gmail.com	M	3	D4	FH629	11/11/2023	11-NOV-23 11.00.00.000000 AM
Marshal	ken	Mathers	239758402	3765498	Sour street majid Apartment 6	Marshal.Math@outlook.com	M	3	A2	FH629	11/11/2023	11-NOV-23 11.00.00.000000 AM

10.2 Nationality

CID	CUSNATIONALITY
202301348	Palestinian
202301349	Palestinian
202301349	Ukranian
202301390	Lebanese
202301391	Lebanese
202301432	Lebanese
202301433	Lebanese
202301502	Lebanese
202301503	Lebanese
202301520	American
202301521	American

10.3 Ticket

RESERVATIONID	SEATNO	CLASS	MEAL	INFLIGHTINSURANCE	LUGGAGEWEIGHT	PASSENGERID	COST
1001	25	Economy	Caesar Salad	Class C	40	202301348	350
2002	23	Economy	Caesar Salad	Class C	40	202301349	350
3003	10	Business	Gourmet Seafood	Class B	55	202301390	750
4004	12	Business	Rice and Meat	Class B	55	202301391	750
5005	4	First	Omelet with sausage	Class A	70	202301432	1500
6006	5	First	Muffins	Class A	70	202301433	1500
7007	15	Business	Vegetable Salad	Class B	55	202301502	750
8008	3	First	Granola bars	Class A	70	202301503	1500
9009	27	Economy	Cheese and Crackers	Class C	40	202301520	350
10001	24	Economy	Pasta and Meat	Class C	40	202301521	350

10.4 Flight

PLANENUM	FLIGHTDATE	DEPARTTIME	ARRIVALTIME	DEPARTFROM	DESTINATION	ARPTNAME
FH123	03/11/2023	11-MAR-23 09.00.00.000000 AM	12-MAR-23 12.00.01.000000 AM	Manchester	Beirut	Queen Bee International Airport - Manchester
FH456	06/12/2023	12-JUN-23 06.30.00.000000 AM	12-JUN-23 01.00.00.000000 PM	Hamburg	Ontario	Jagersdorf International Airport
FH726	07/25/2023	25-JUL-23 09.15.00.000000 PM	26-JUL-23 04.00.00.000000 AM	Beirut	Hamburg	Haraty International Airport
FH679	11/11/2023	11-NOV-23 11.00.00.000000 AM	11-NOV-23 09.30.00.000000 PM	Beirut	Manchester	Haraty International Airport
FH629	11/11/2023	11-NOV-23 11.00.00.000000 AM	11-NOV-23 06.00.00.000000 PM	Ontario	Newyork	Maple Leaf International Airport

10.5 Department

DEPNAME	COUNTRY	EMAILADDRESS	PHONEEXTENSION	BUDGET	MANAGERID
Flight Operations	Lebanon	flightops@flyharaty.com.lb	101	750000	1
Finance	Lebanon	finance@flyharaty.com.lb	103	500000	2
Flight Operations	Germany	@flightopsGRlyharaty.com	111	900000	6
Finance	Canada	finance@flyharaty.com.fr	113	900000	10
Customer Care	United Kingdom	customercareUK@flyharaty.com	114	450000	4
Maintenance	United States	maintenanceUS@flyharaty.com.	122	1200000	5
Flight Operations	Canada	flightopsCA@flyharaty.com.lb	922	750000	11
Flight Operations	United Kingdom	flightopsUK@flyharaty.com.lb	666	750000	15
Flight Operations	United States	flightopsUS@flyharaty.com.lb	711	750000	12

10.6 Staff

EMPLOYEEID	NAME	DOB	NATIONALITY	SALARY	REWARDPOINTS	STARTDATE	INSURANCENUMBER	SUPID	SHIFT	DCOUNTRY
1	Ramzi R Haraty	10/01/1961	Lebanese	10000	1000	01/23/1986	859314267	-	AM	Lebanon
2	Joseph rebehmed	06/04/1980	Lebanese	8000	400	03/12/1998	426718935	-	AM	Lebanon
3	Faisal Abu Khzam	05/08/1968	American	5000	0	04/09/2002	973512468	1	PM	Lebanon
4	Nadine Abbas	11/05/1996	Saudi	5000	500	11/24/2011	631845279	-	PM	United Kingdom
5	Nader El Khatib	04/07/1984	Lebanese	5000	800	01/19/2015	248697513	-	PM	United States
6	Ayoub Saab	05/12/1993	Lebanese	5000	560	03/18/2012	512384679	-	AM	Germany
7	Saul Goodman	12/12/1985	American	9000	240	05/14/2012	394752186	6	PM	Germany
8	Gordon Ramsay	11/08/1966	British	6000	680	09/10/2004	827163594	5	PM	United States
9	Arthur Morgan	05/23/1994	American	5500	450	12/15/2004	649218753	2	PM	Lebanon
10	Trevor Philips	07/07/1970	Canadian	7000	430	04/14/2006	175936842	-	AM	Canada
11	Ramez Albilo	11/11/1987	English	5000	500	01/13/2000	540529328	-	AM	Canada
12	Mithad roge	07/09/1970	American	6500	300	01/17/2010	685930503	-	AM	United States
13	Fareed Atrash	06/04/1990	American	5000	500	11/01/2011	329582914	12	PM	United States
14	George Svin	09/23/1996	English	5000	750	10/21/2015	594922138	11	PM	Canada
15	Kinda Halm	11/11/1987	English	4000	700	11/03/2014	796934029	-	AM	United Kingdom

10.7 Insurance

INSURANCEID	POLICYCLASS	STARTDATE	RENEWALDATE
540529328	ClassC	09/14/2022	07/15/2023
685930503	ClassC	06/18/2022	12/15/2024
329582914	ClassC	04/03/2022	11/07/2025
594922138	ClassC	12/23/2022	07/04/2023
796934029	ClassC	03/27/2022	04/21/2024
583970383	ClassC	04/20/2022	06/10/2024
124968324	ClassC	07/12/2022	07/10/2025
968393834	ClassC	03/05/2022	06/07/2024
867958330	ClassC	02/07/2022	11/17/2023
348875898	ClassC	08/04/2022	09/13/2024
233982939	ClassC	11/21/2022	03/21/2024
859314267	ClassA	01/01/2023	05/01/2024
426718935	ClassB	05/15/2022	03/15/2025
973512468	ClassC	08/20/2023	12/20/2024
631845279	ClassA	11/10/2022	11/10/2026

INSURANCEID	POLICYCLASS	STARTDATE	RENEWALDATE
248697513	ClassB	03/05/2023	09/05/2024
512384679	ClassC	07/18/2022	11/18/2025
394752186	ClassA	02/28/2023	08/28/2024
827163594	ClassB	09/12/2022	12/12/2026
649218753	ClassC	06/25/2023	06/25/2024
175936842	ClassA	12/15/2022	12/15/2025

10.8 Transport

LICENSEPLATE	MODELMAKE	CAPACITY	PASSENGERREQUESTS	PICKUPDATE	PICKUPTIME	PICKUPPOINT	DROPOFFPOINT	DRIVERID
G100404	Nissan Altima (2018)	4	Please get me a coffee before pickup	03/12/2023	12-MAR-23 12.00.05.000000 PM	Haraty International Airport	Gefinor Rotana-Beirut	3
T208469	Honda Accord (2019)	4	Please adjust the air conditioning to a comfortable temperature before pickup	03/12/2023	12-MAR-23 12.00.10.000000 PM	Haraty International Airport	Gefinor Rotana-Beirut	17
374ADH	Honda Accord (2017)	4	-	06/12/2023	12-JUN-23 01.00.00.000000 PM	Maple Leaf International Airport	Fairmont Royal York-Toronto	14
323ASH	Nissan Altima (2020)	4	-	06/12/2023	12-JUN-23 01.00.00.000000 PM	Maple Leaf International Airport	Four Season Hotel-Toronto	21
CDZ-120	Honda Accord (2017)	4	-	07/26/2023	26-JUL-23 04.00.00.000000 AM	Jagersdorf International Airport	The Ritz-Carlton-Hamburg	7
ARK-440	Chevrolet Impala (2020)	4	Please get me a a sick bag before pickup because I get nauseous on flights	07/26/2023	26-JUL-23 04.00.00.000000 AM	Jagersdorf International Airport	Hotel Adlon Kempinski-Hamburg	20
BFD-220	Toyota Camry (2019)	4	-	11/11/2023	11-NOV-23 09.30.00.000000 PM	Queen Bee International Airport	Hotel Gotham-Manchester	16
LLR-320	Ford Fusion (2018)	4	-	11/11/2023	11-NOV-23 09.30.00.000000 PM	Queen Bee International Airport	The Midland Hotel-Manchester	19
NY-9012	Nissan Altima (2018)	4	Please turn on seat warmers before pickup	11/11/2023	11-NOV-23 06.00.00.000000 PM	SkyHarbor International Airport	The Lowell-New York	13
NY-3456	Toyota Camry (2019)	4	Please get me a coffee before pickup	11/11/2023	11-NOV-23 06.00.00.000000 PM	SkyHarbor International Airport	The William Vale-New York	18

10.9 Airport

AIRPORTNAME	ACOUNTRY	CITY	CAPACITY	NUMGATES	NUMTERMINALS	PHONENUMBER	WEBPAGE
Haraty International Airport	Lebanon	Beirut	5000000	7	2	01596922	www.FlyHaraty.com.lb
SkyHarbor International Airport - New York	United States	New York	10000000	12	3	555-692-375	www.flyharatynewyork.com
Queen Bee International Airport - Manchester	United Kingdom	Manchester	10000000	12	3	(0161) 111 1111	www.flyharatymanchester.com
Maple Leaf International Airport	Canada	Ontario	12000000	12	3	+1 604 12345678	www.flyharatyontario.com.lb
Jagersdorf International Airport	Germany	Hamburg	12000000	15	5	+49 40 12345678	www.flyharatyhamburg.com

10.10 Store

STORENAME	ANAME	TYPE	PROFITSPLIT	PARTNERSHIPSTARTDATE
Noreen's Corner	Haraty International Airport	Burger Joint	50	11/11/2014
Ahmad's Botique	SkyHarbor International Airport - New York	Shoe Retail	50	10/31/2012
Burger King	Queen Bee International Airport - Manchester	Fast Food	60	05/05/2010
BBQ House	Maple Leaf International Airport	Restuarant	30	04/12/2015
Patchi	Jagersdorf International Airport	Chocolate Shop	40	07/09/2010
Bershka	Jagersdorf International Airport	Clothing Retail	35	04/28/2016
Haraty caffe	Haraty International Airport	Caffe Shop	70	09/05/2010
Klein	Maple Leaf International Airport	Mini Market	40	09/03/2013
TakeGrill	SkyHarbor International Airport - New York	Sandwich Shop	35	06/15/2014
Gifto	Queen Bee International Airport - Manchester	Gift Shop	25	02/06/2017
Flytech	Queen Bee International Airport - Manchester	Electronic stores	40	10/14/2013
japanese food	SkyHarbor International Airport - New York	noodles store	50	08/03/2012

10.11 Hotel Room

HOTELNAME	ROOMNUM	ROOMTYPE	PSNGRID	CHECKINDATE	CHECKOUTDATE	ROOMCOST
Gefinor Rotana-Beirut	1004	Suite	202301348	03/11/2023	03/12/2023	600
Radinsson Blu Martinez-Beirut	702	Deluxe	202301349	03/11/2023	03/12/2023	300
The Lowell-New York	804	Deluxe	202301520	11/11/2023	11/12/2023	450
The William Vale-New York	1204	Suite	202301521	11/11/2023	11/12/2023	650
The Midland Hotel-Manchester	700	Deluxe	202301502	11/11/2023	11/12/2023	400
Hotel Gotham-Manchester	103	Standard	202301503	11/11/2023	11/12/2023	100
Four Season Hotel-Toronto	703	Deluxe	202301390	06/12/2023	06/13/2023	400
Fairmont Royal York-Toronto	209	Standard	202301391	06/12/2023	06/13/2023	150
The Ritz-Carlton-Hamburg	1010	Suite	202301432	07/26/2023	07/27/2023	700
Hotel Adlon Kempinski-Hamburg	303	Standard	202301433	07/26/2023	07/27/2023	50

10.12 Includes

RESID	PLATENUM
1001	G100404
2002	G100404
3003	374ADH
4004	323ASH
5005	CDZ-120
6006	ARK-440
7007	BFD-220
8008	BFD-220
9009	NY-9012
10001	NY-3456

10.13 Operates

OPERATORID	PLANENO	FDATE	DEPTIME	ROLE
1	FH123	03/11/2023	11-MAR-23 09.00.00.000000 AM	Pilot
6	FH726	07/25/2023	25-JUL-23 09.15.00.000000 PM	Pilot
11	FH456	06/12/2023	12-JUN-23 06.30.00.000000 AM	Pilot
15	FH679	11/11/2023	11-NOV-23 11.00.00.000000 AM	Pilot
12	FH629	11/11/2023	11-NOV-23 11.00.00.000000 AM	Pilot

11 Utilizing SQL Queries

We now present several cases that demonstrate the use of the Structured Query Language (SQL).

11.1 Partnership Divorce

Fly Haraty wants to reconsider its partnerships with companies that produce a lower profit margin than their other stores. They decide to collect a list of all stores whose profit split with Fly Haraty is at most 40% of the store's income.

Solution: We will utilize an SQL query to collect the name, airport name, and country of the story whose profit split is 40% or less, like so:

```
1 SELECT STORENAME, ANAME,AIRPORTNAME,ACOUNTRY
2 FROM STORE,AIRPORT
3 WHERE ANAME = AIRPORTNAME
4 AND ProfitSplit <= 40;
```

STORENAME	ANAME	AIRPORTNAME	ACOUNTRY
BBQ House	Maple Leaf International Airport	Maple Leaf International Airport	Canada
Patchi	Jagersdorf International Airport	Jagersdorf International Airport	Germany
Bershka	Jagersdorf International Airport	Jagersdorf International Airport	Germany
Klein	Maple Leaf International Airport	Maple Leaf International Airport	Canada
TakeGrill	SkyHarbor International Airport - New York	SkyHarbor International Airport - New York	United States
Gifto	Queen Bee International Airport - Manchester	Queen Bee International Airport - Manchester	United Kingdom
Flytech	Queen Bee International Airport - Manchester	Queen Bee International Airport - Manchester	United Kingdom

11.2 Hit-man on the Loose

Fly Haraty was shocked to discover that a murder had taken place during one of their flights!
After canvasing the areas and collecting initial data, Fly haraty has deduced that the murder took place on flight *FH123*. After interviewing the witnesses, one of them mentioned that they saw a suspicious individual on the plane who slipped something into the deceased victim's drink. The witness could not remember much of the individual's appearance, except that they had ordered a caeser salad. Working with this tiny lead, Fly Haraty wishes to assist public law enforcement with a query that helps them narrow down their suspect list.

Solution: We will write an SQL query to collect the names of people who were on the flight and had ordered a caeser salad. Like so:

```
SELECT FNAME, MNAME, LNAME,CLIENT.CUSTOMERID,
 PASSPORTNUM, PHONENUMBER,HOTELNAME ,ROOMNUM
FROM CLIENT, HOTELROOM, TICKET
WHERE CLIENT.PNUM = 'FH123'
AND CUSTOMERID = PSNGRID
AND PASSENGERID = CUSTOMERID
AND MEAL = 'Caesar Salad'
```

FNAME	MNAME	LNAME	CUSTOMERID	PASSPORTNUM	PHONENUMBER	HOTELNAME	ROOMNUM
Hadi	Wissam	Al Hassan	202301348	LR-1654321	81769033	Gefinor Rotana-Beirut	1004
Sari	Mahmoud	Abdul Ghani	202301349	LR-6543684	76500432	Radinsson Blu Martinez-Beirut	702

I bet the first guy did it.

11.3 Rude Driver

One of our customers reported that her driver was rude and inappropriate during the pickup, we do not tolerate such actions at Fly Haraty, and so we will lookup the name of the employee assigned to the vehicle through the license plate the complaint issuer recorded, to immediately order an internal review and decide pending action.

Solution: We will run the license plate of the driver's car, and retrieve the driver log to compare with the date of the complaint and pickup time recorded by the client to identify the driver.

```
SELECT PICKUPDATE, PICKUPTIME, NAME, EMPLOYEEID,
 SALARY*0.75 AS REDUCED_SALARY
FROM TICKET, INCLUDES, TRANSPORT, STAFF
WHERE PASSENGERID = '202301432'
AND RESERVATIONID = RESID
AND PLATENUM = LICENSEPLATE
AND LICENSEPLATE = 'CDZ-120'a
AND EMPLOYEEID = DRIVERID;
```

PICKUPDATE	PICKUPTIME	NAME	EMPLOYEEID	REDUCED_SALARY
07/26/2023	26-JUL-23 04.00.00.000000 AM	Saul Goodman	7	6750

11.4 Men's Day Discount

Fly Haraty wants to offer a promotion in celebration of International Men's Day, by reducing the ticket and room price by 70% and 50% respectively. However, to compensate for the reduced revenue, and it being a men's day, Fly Haraty has decided to accordingly bump up the price of women's ticket and room price by 20% and 50% respectively.

Solution: We design a query to go over our clients, and adjust their room and ticket cost based on gender. Like so:

```
SELECT
    FNAME,
    MNAME,
    LNAME,
    HOTELNAME,
    CASE WHEN GENDER = 'M' THEN ROOMCOST * 0.3
    ELSE ROOMCOST * 1.2 END AS ADJUSTED_ROOMCOST,
    PNUM,
    CASE WHEN GENDER = 'M' THEN COST * 0.5
    ELSE COST * 1.5 END AS ADJUSTED_COST
FROM
    CLIENT
JOIN
    HOTELROOM ON CUSTOMERID = PSNGRID
JOIN
    TICKET ON CUSTOMERID = PASSENGERID;
```

FNAME	MNAME	LNAME	HOTELNAME	ADJUSTED_ROOMCOST	PNUM	ADJUSTED_COST
Hadi	Wissam	Al Hassan	Gefinor Rotana-Beirut	180	FH123	175
Sari	Mahmoud	Abdul Ghani	Radinsson Blu Martinez-Beirut	90	FH123	175
Bob	Johny	Anderson	The Lowell-New York	135	FH629	175
Marshal	ken	Mathers	The William Vale-New York	195	FH629	175
Ramez	Ali	ElMasri	The Midland Hotel-Manchester	120	FH679	375
Wael	Emily	Kfouri	Hotel Gotham-Manchester	30	FH679	750
Siraj	Imad	Al Mabsout	Four Season Hotel-Toronto	120	FH456	375
Gaelle	Elie	Loutfi	Fairmont Royal York-Toronto	180	FH456	1125
Ahmad	Ramzi	Haraty	The Ritz-Carlton-Hamburg	210	FH726	750
Noreen	Ramzi	Haraty	Hotel Adlon Kempinski-Hamburg	60	FH726	2250

11.5 Reward for Night Shift Employees

It has been a rough month for our night shift staff, and we have decided to reward them for their perseverance through the long hours of the night, with a 50% raise and 300 employee points for the month.

Solution: We design an SQL query to increase the salary and reward point count for any employee on the night shift.

```
SELECT DISTINCT
    NAME,
    SALARY * 1.5 AS ADJUSTED_SALARY,
    REWARDPOINTS + 300 AS ADJUSTED_REWARDPOINTS
FROM
    STAFF
JOIN
    DEPARTMENT ON DEPNAME = 'Finance'
WHERE
    SHIFT = 'PM';
```

NAME	ADJUSTED_SALARY	ADJUSTED_REWARDPOINTS
Saul Goodman	13500	540
Fareed Atrash	7500	800
George Svin	7500	1050
Faisal Abu Khzam	7500	300
Arthur Morgan	8250	750
Nadine Abbas	7500	800
Nader El Khatib	7500	1100
Gordon Ramsay	9000	980
Jad Karm	4500	800

11.6 Microsoft Partnership

Microsoft is teaming up with Fly Haraty! And to kick off the partnership, they have generously offered to pay for the trips of any client using an "@outlook.com" email.

Solution: We design an SQL query to comb through the emails of our clients and return those whose email ends with "@outlook.com", using the `REGEXP_LIKE()` method.

```
SELECT FName, LName, PHONENUMBER
FROM CLIENT
WHERE REGEXP_LIKE(Email, '^[a-zA-Z0-9]+\.[a-zA-Z0-9]+@outlook\.com$');
```

FNAME	LNAME	PHONENUMBER
Noreen	Haraty	03252648
Ramez	ElMasri	71408321
Ahmad	Haraty	03464635
Marshal	Mathers	03765498

Congratulations to the winners!

11.7 Who's the Boss

A quick department survey wants the records of employees working under Lebanon Flight Ops manager, Ramzi Haraty.

Solution: We write an SQL query to retrieve the names of employees who are stationed in Lebanon and work in the "Flight Operations" department. We search for employees whose supervisor has no supervisor (i.e the head of the department).

```
SELECT NAME, EMPLOYEEID
FROM STAFF
WHERE SUPID in (SELECT EMPLOYEEID
FROM STAFF
WHERE SUPID IS NULL
AND DCOUNTRY = 'Lebanon'
AND DNAME = 'Flight Operations');
```

NAME	EMPLOYEEID
Faisal Abu Khzam	3
Majed Room	17

11.8 Abu Salim Hates the 10th Floor

"Abu salim", one of Fly Haraty's drivers, hates picking up clients that regularly stay at the 10th floor of Beirut hotels, that are rude to him during the ride, and have him carry their luggage on his bad back, without even giving him a tip. He asks for us to do him a favor today, and put him on the shift opposite to the time pickups for which the "10th floor folks" are scheduled.

Solution: Accommodating Abu Salim's request, we design an SQL query to get the pickup time of individuals who booked a suite on the 10th floor today, and will place him on a different shift to avoid the cheapskates.

```
SELECT Distinct FNAME, PICKUPTIME
FROM HOTELROOM, CLIENT, TICKET, INCLUDES, TRANSPORT
WHERE CUSTOMERID = PSNGRID
    AND CUSTOMERID = PASSENGERID
    AND RESERVATIONID = RESID
    AND LICENSEPLATE = PLATENUM
    AND ROOMNUM BETWEEN 1000 AND 1099
    AND CHECKINDATE = TO_DATE('2023-03-11', 'YYYY-MM-DD');
```

FNAME	PICKUPTIME
Hadi	12-MAR-23 12.00.05.000000 PM

11.9 Bomb Threat

The morning of June 12, Fly Haraty have received an alarming anonymous phone call that a client is planning to bring an explosive into their German airport, and will be placed in terminal 2, gate B2.

Solution: we hurriedly write a query to gather the names of all suspects who will be passing through the specific location and take them into questioning.

```
SELECT C.FNAME, C.MNAME, C.LNAME, C.PHONENUMBER
FROM AIRPORT A
JOIN FLIGHT F ON A.AIRPORTNAME = F.ARPTNAME
JOIN CLIENT C ON F.PLANENUM = C.PNUM
AND AIRPORTNAME = 'Jagersdorf International Airport'
AND F.FLIGHTDATE = TO_DATE('2023-06-12','yyyy-mm-dd')
AND C.TERMINALNUM = 2
AND C.GATENUM = 'B2';
```

FNAME	MNAME	LNAME	PHONENUMBER
Siraj	Imad	Al Mabsout	78811997
Gaelle	Elie	Loutfi	78043235

And now, comes arguably the team's hardest query yet.

11.10 A Wishy Washy Case

As part of a case study correlating income with insurance, Fly Haraty are asked to give a list of employees who have class A insurance and their salaries. Sounds pretty difficult.

Solution: The team writes a query to retrieve all the names, department, and salary of all those holding class A insurance.

```
1 SELECT DISTINCT NAME,DNAME,SALARY
2 FROM STAFF,INSURANCE
3 WHERE   INSURANCENUMBER = INSURANCEID
4 AND POLICYCLASS in ('ClassA')
```

NAME	DNAME	SALARY
Ramzi R Haraty	Flight Operations	10000
Saul Goodman	Flight Operations	9000
Nadine Abbas	Customer Care	5000
Trevor Philips	Finance	7000

12 Database Normalization

In this section we map functional dependencies and normalize our database.

12.1 First Normal From (1NF)

The first normal form is a property of a relation in a relational database that does not allow tables/arrays as values, therefore maintaining the atomicity property of the database. All our tables pass the 1NF, as no value in a tuple includes multiple values.

12.2 Second Normal Form (2NF)

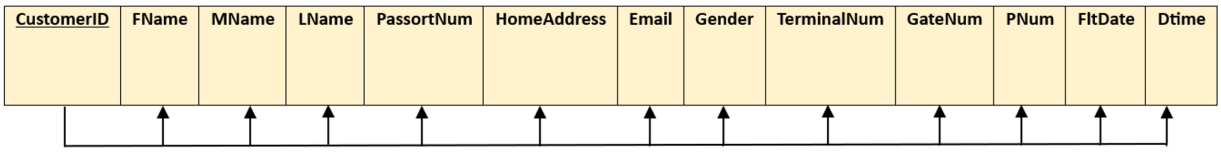
A relation is said to be in second normal form if and only if the relation is in first normal form but has no non-prime attribute functionally dependent on any candidate key's proper subset in a relation. All our tables pass the 2NF, as no proper subset of the primary key functionally determines a non-prime attribute.

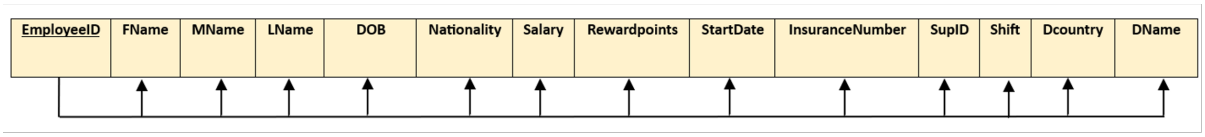
12.3 Third Normal Form (3NF)

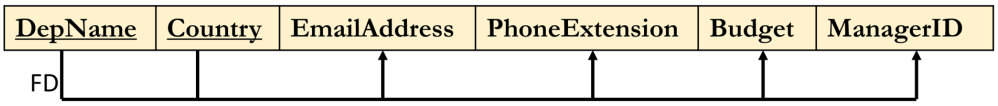
A relation is said to be in third normal form if and only if the relation is in second normal form but has no transitive partial dependency, in other words, no non-prime attribute transitively dependent on the primary key. Our tables are in 3NF, as no non-prime functionally determines any non-prime attributes.

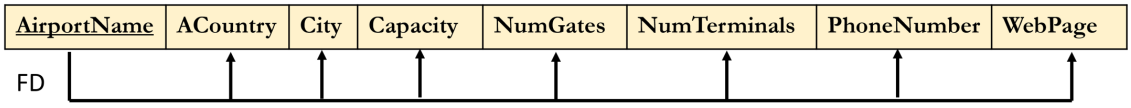
12.4 Boyce Codd Normal Form (BCNF or 3.5NF)

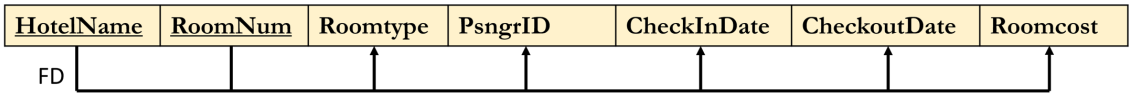
The Boyce-Codd normal form adds a condition of the third normal form, where, for all functional dependencies $X \rightarrow Y$, X is either the super key, or the candidate key. In other words, no non-prime attribute functionally depends a prime attribute. Our databases are in BCNF, as no such contradiction to the BCNF property exists.

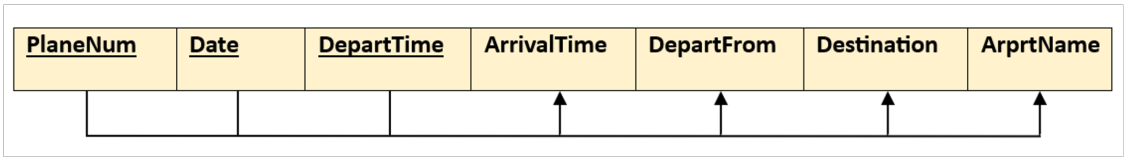

CustomerID
FName
MName
LName
PassortNum
HomeAddress
Email
Gender
TerminalNum
GateNum
PNum
FltDate
Dtime

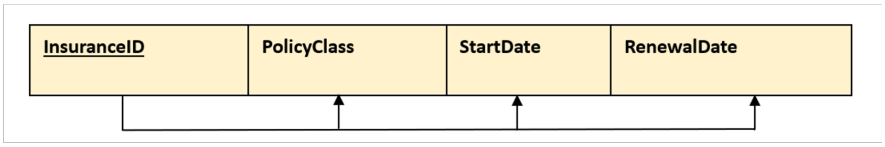

EmployeeID
FName
MName
LName
DOB
Nationality
Salary
Rewardpoints
StartDate
InsuranceNumber
SupID
Shift
Dcountry
DName

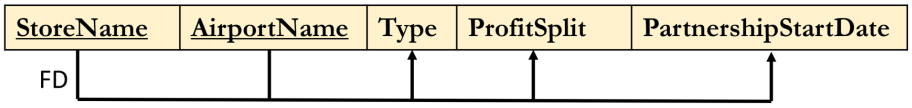

DepName
Country
EmailAddress
PhoneExtension
Budget
ManagerID
FD

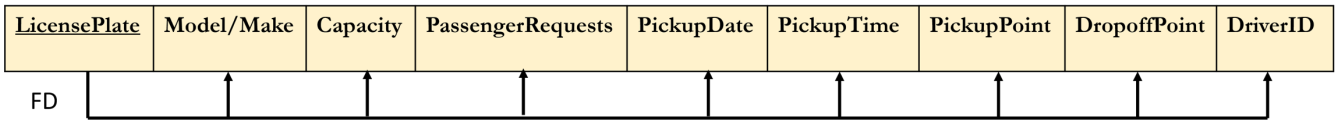

AirportName
ACountry
City
Capacity
NumGates
NumTerminals
PhoneNumber
WebPage
FD

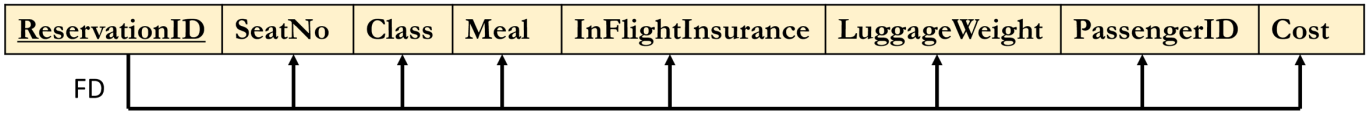

HotelName
RoomNum
Roomtype
PsngrID
CheckInDate
CheckoutDate
Roomcost
FD

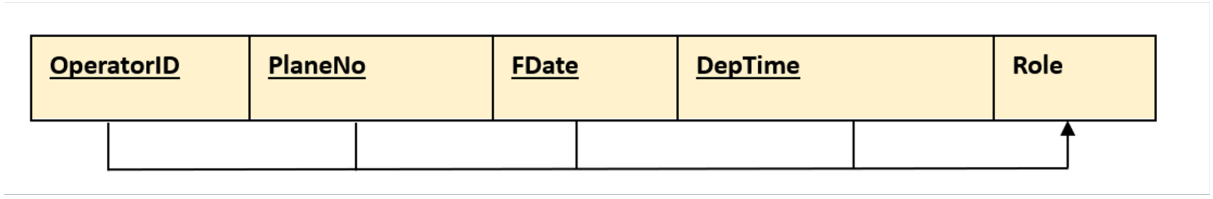

PlaneNum
Date
DepartTime
ArrivalTime
DepartFrom
Destination
ArprtName

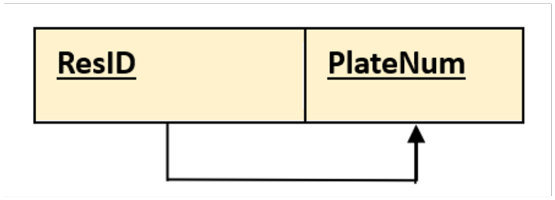

InsuranceID
PolicyClass
StartDate
RenewalDate

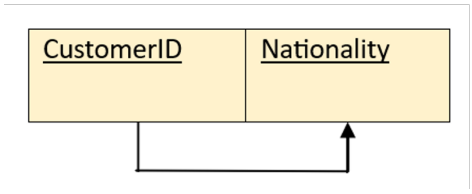

StoreName
AirportName
Type
ProfitSplit
PartnershipStartDate
FD


LicensePlate
Model/Make
Capacity
PassengerRequests
PickupDate
PickupTime
PickupPoint
DropoffPoint
DriverID
FD


ReservationID
SeatNo
Class
Meal
InFlightInsurance
LuggageWeight
PassengerID
Cost
FD


OperatorID
PlaneNo
FDate
DepTime
Role


ResID
PlateNum


CustomerID
Nationality

13 Conclusion

In phase I , we introduced Fly Haraty and described the system requirements for the company's database system. We drew the ER diagram of our database, explaining the entities and relationships between them.

In phase II, we mapped said entities and relationships to tables.

In Phase III, we built our database by creating tables and inserting data, in addition to performing queries in SQL.

In this phase, we normalized our database, and ensured it is in Boyce-Codd Normal Formal (BCNF).

14 Instructor's Feedback

-
-
-
-
-
-
-
-
-

Personal notes / Comments :)

Ratings:

Presentation: 1 - 2 - 3 - 4 - 5

Style: 1 - 2 - 3 - 4 - 5

Creativity: 1 - 2 - 3 - 4 - 5

Structure and logic: 1 - 2 - 3 - 4 - 5

Following Feedback: 1 - 2 - 3 - 4 - 5