S

Database Design and Development

Using Access and MS SQL Server

Sara Ferreira da Silva

Student Number: 1669329

# Table of Contents

[Table of Contents 2](#_Toc465858937)

[Background for the development of the chosen database system 3](#_Toc465858938)

[Scope for the Database System to be Developed 3](#_Toc465858939)

[Business Rules 3](#_Toc465858940)

[Entity Relationship Diagram (ERD) 4](#_Toc465858941)

[Relational Schema in 3NF 5](#_Toc465858942)

[Functional Dependencies: 6](#_Toc465858943)

[Evaluation of the Database Design 7](#_Toc465858944)

[Appendix (MS Access Database) 7](#_Toc465858945)

[Appendix (MS SQL Server Database) 16](#_Toc465858946)

[Bibliography 22](#_Toc465858947)

# Background for the development of the chosen database system

Celtic travel was originally established in a small location in Washington DC in 1976 as a small family run business. In 1982, David Murphy joined the company and began promoting travel to his friends. David Murphy's extensive marketing and educational background enabled him to grow the business substantially. Celtic Travel became extremely competitive in the local industry. However, the company has still been using spreadsheets and flat file databases to gather and organize its data which has been troublesome and caused several issues. David Murphy believes this has delayed the company’s growth.

# Scope for the Database System to be Developed

He believes a database will make it easier and more efficient to manage the company’s data. The database will keep track of customer’s information as well as agents and employees. It will track hotel and flight bookings as well as availability which should make travel bookings faster for customers and eliminate transaction mistakes. It will serve as a data repository for the company and with the help of queries and reports it will keep all Celtic travel employees correctly informed as well as aid managers decision making process.

# Business Rules

1. Only one traveller and one agent are allowed per booking.
2. The flight table checks to ensure the departure city and arrival city are different. It also ensures the departure date is less than or equal to the arrival date.
3. A person can only have a home number and cell phone per row.
4. Once a reservation availability for any type hits zero a null is returned. Overbooking is not allowed.

# Entity Relationship Diagram (ERD)



# Relational Schema in 3NF

RelationalSchema3NF

# Functional Dependencies:

**Person table:** The Person table stores information about all people associated with the agency from travellers, agents and salary employees.

Functional Dependency: PersonID → lastName, firstName, email

**Agent table:** The agents table stores all data about the agents who sell for Celtic Travel and their commission.

Functional Dependency: PersonID → commissionPercentage

**Employee table:** Employees who do not work by commission are stored here. An example employee type could be office administrator or janitor.

Functional Dependency: PersonID → hourlyWage

**Hotel and HotelBooking tables**: The hotel table stores information about a hotel, the hotel bookings table links a hotel stay to an itinerary. As with the

car rental table, the rooms available will be automatically decremented when booking is added.

Functional Dependencies Hotel: HotelID → location, HotelName, RoomsAvailable

Functional Dependencies Hotel\_Bookings: HotelID, ItineraryID →

**Flight table:** The Flight table is similar to the Hotel table however, it does check to ensure the departure city and arrival city are different. It also ensures the departure date is less than or equal to the arrival date.

Functional Dependencies: FlightNumber → AirlineName, depart, arrive, DepartCity, ArriveCity, Capacity

**Flight Booking table:** The Flight Booking table links flights to itineraries.

Functional Dependencies: FlightNumber, ItineraryID →

**Itinerary table**: The Itinerary table brings everything together. It links a customer to flight and hotel bookings.

Functional Dependencies: ItineraryID → CustomerID, AgentID, BookedDate

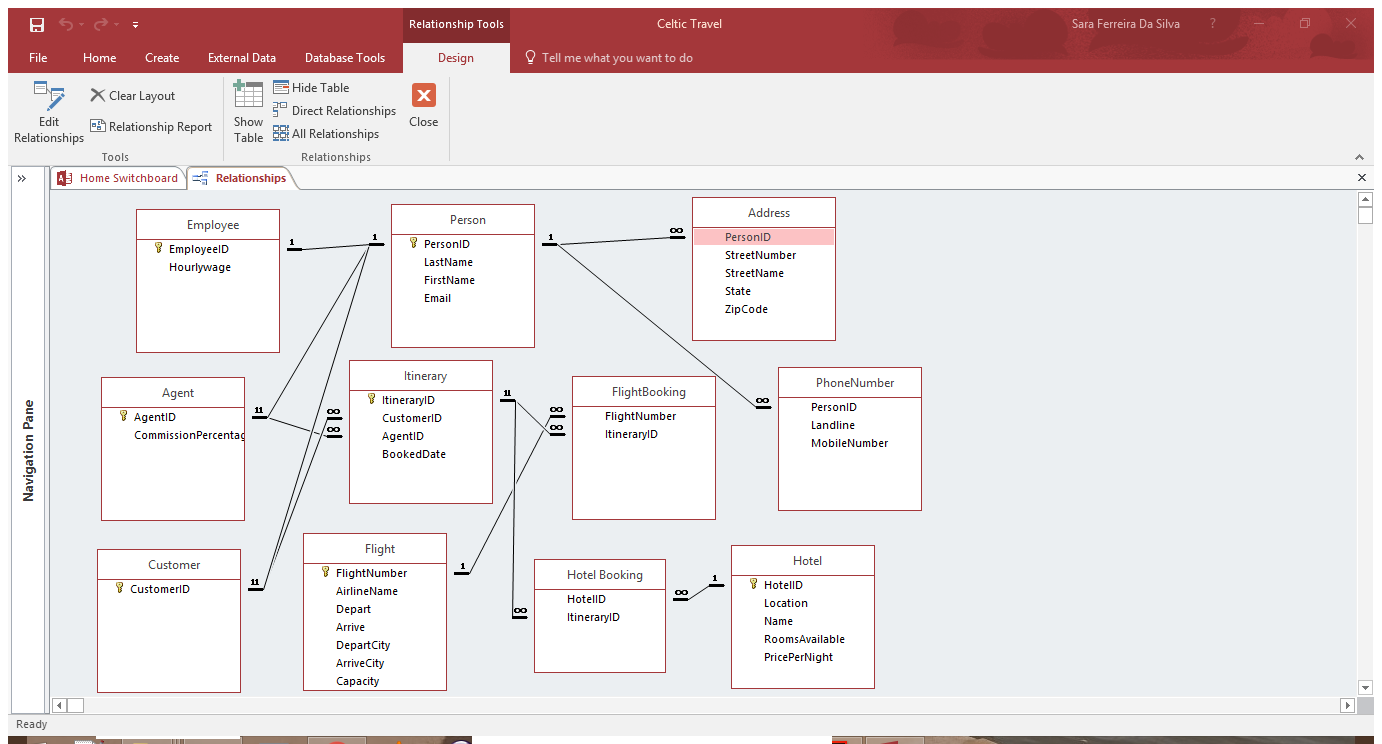
# Evaluation of the Database Design

The Database system developed will greatly help Celtic Travel achieve further growth by acting as a repository through which the company’s information can be manage. However, there is still room for improving the system such as:

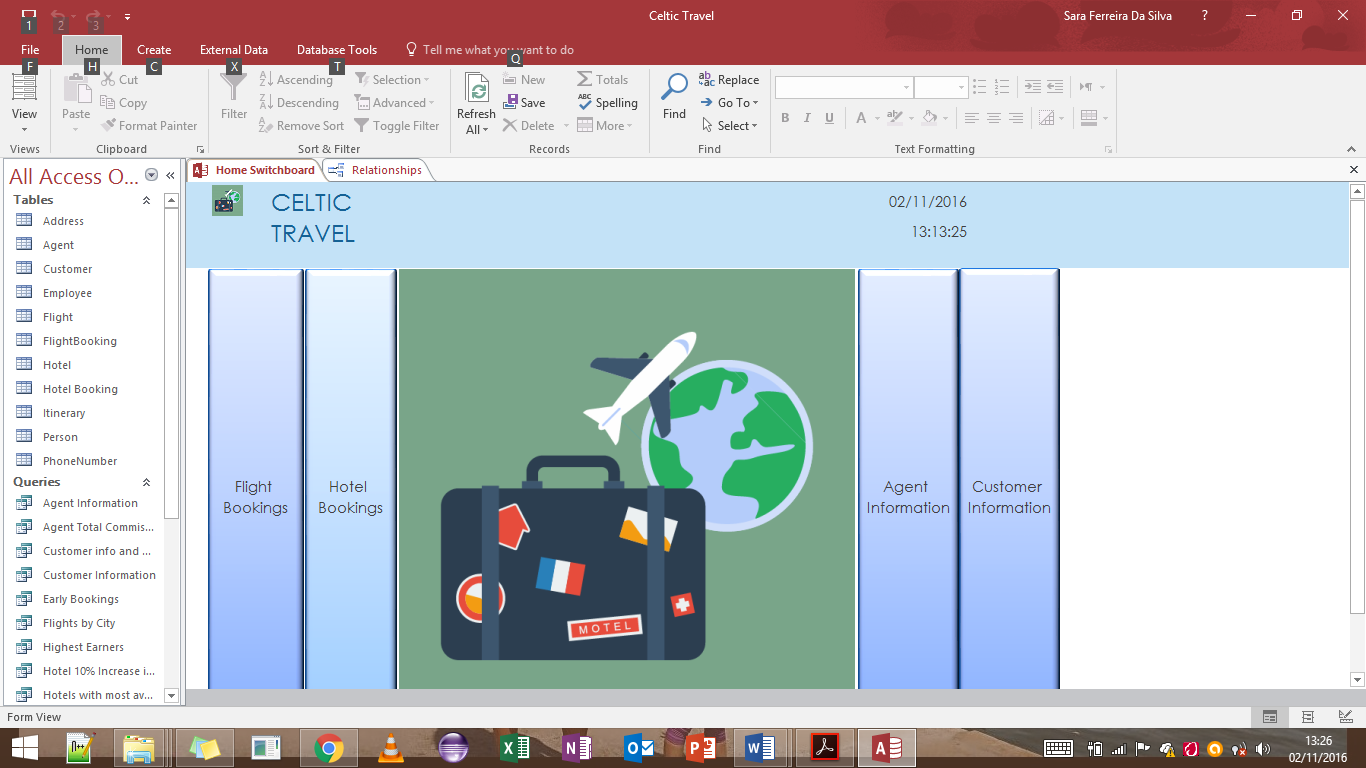
* Creating more queries that will support details of all reservation types;
* Adding transaction values such as purchase amounts so that costs for a trip can be tracked;
* Creating a more complex and detailed reservation system so that tables are not needed in case of future trip types such as cruises, resorts, etc.

# Appendix (MS Access Database)

Database Relationships

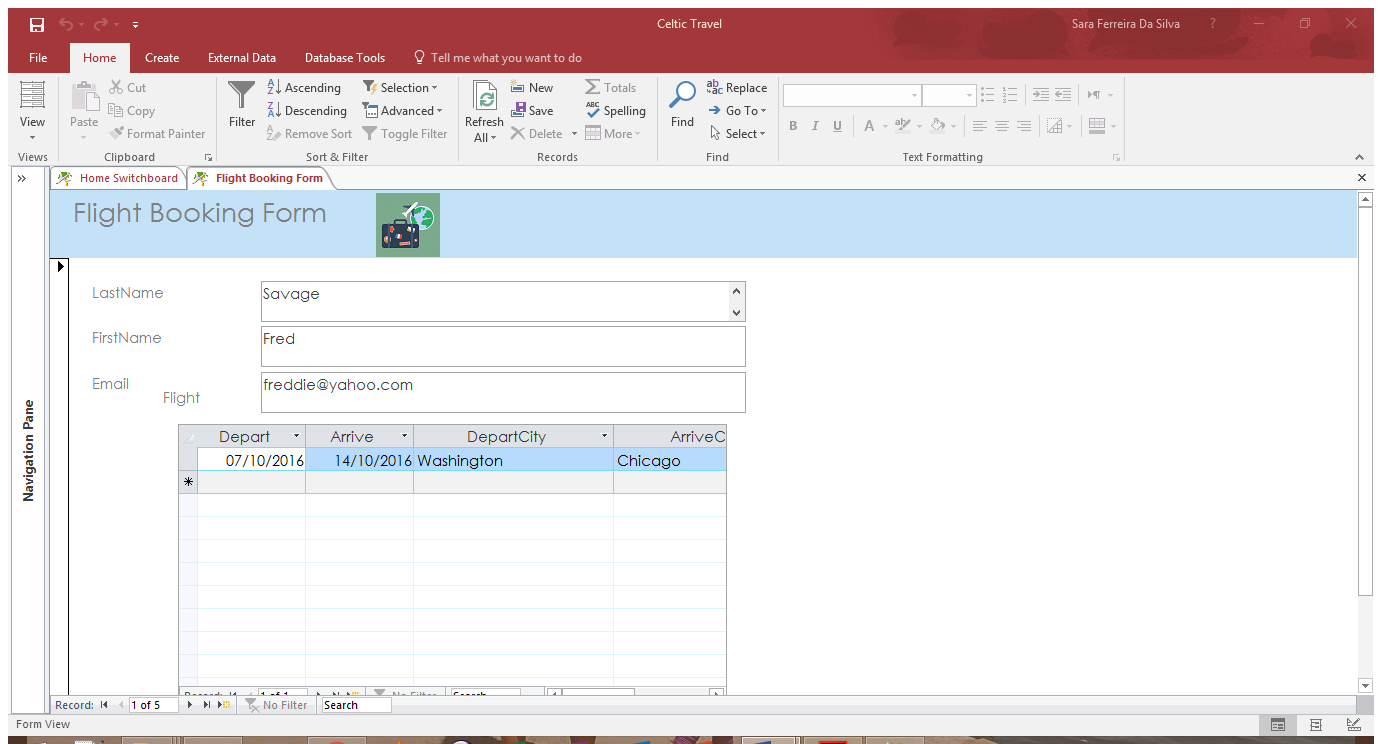


Switchboard

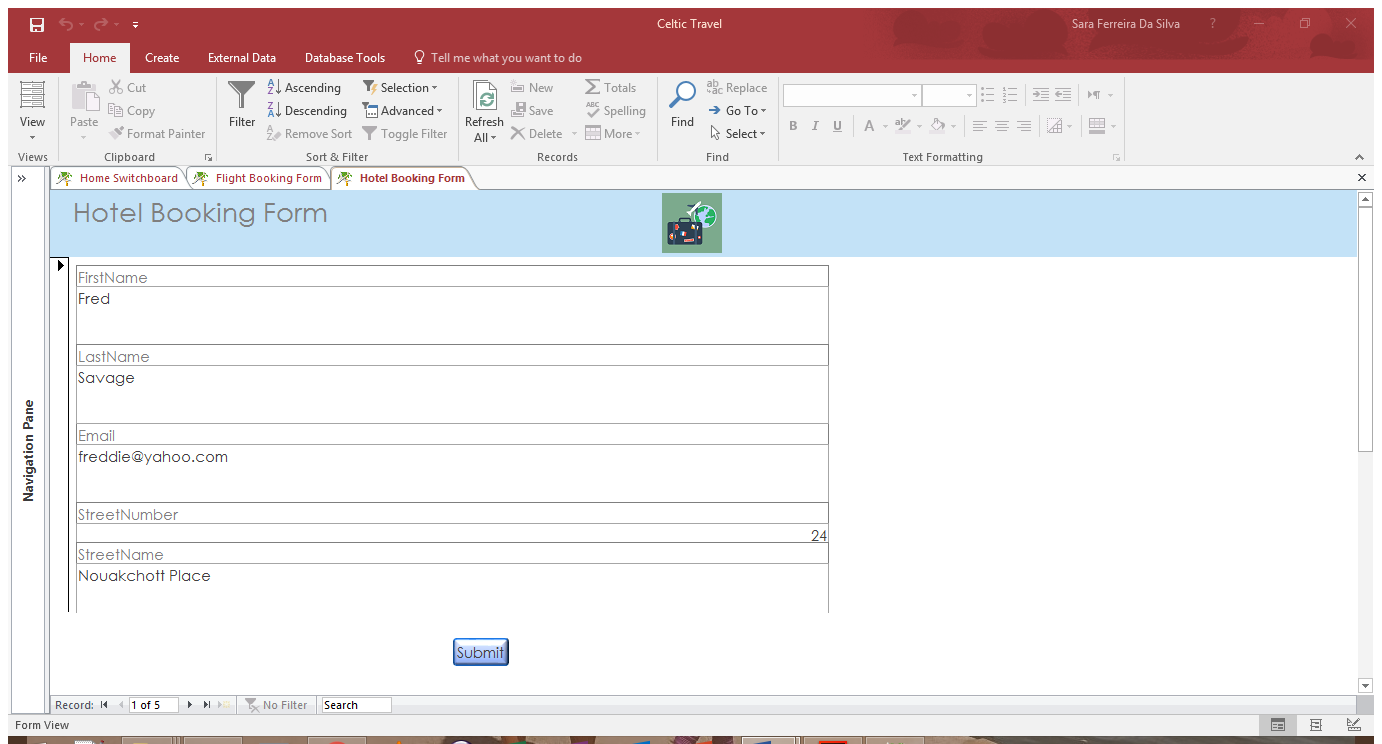


Flight Bookings and Hotel Bookings forms can be accessed through the switchboard as well as Agent Information and Customer Information reports.

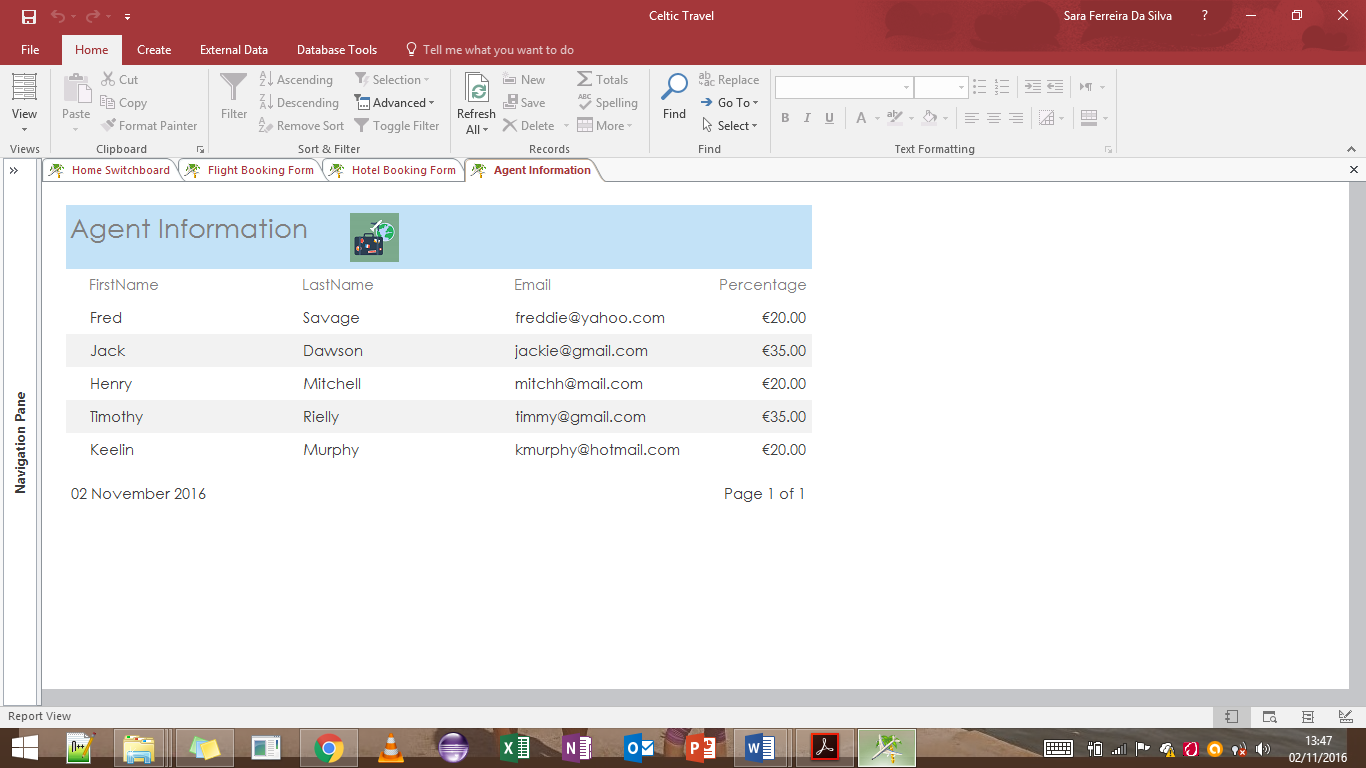
Flight Bookings Form



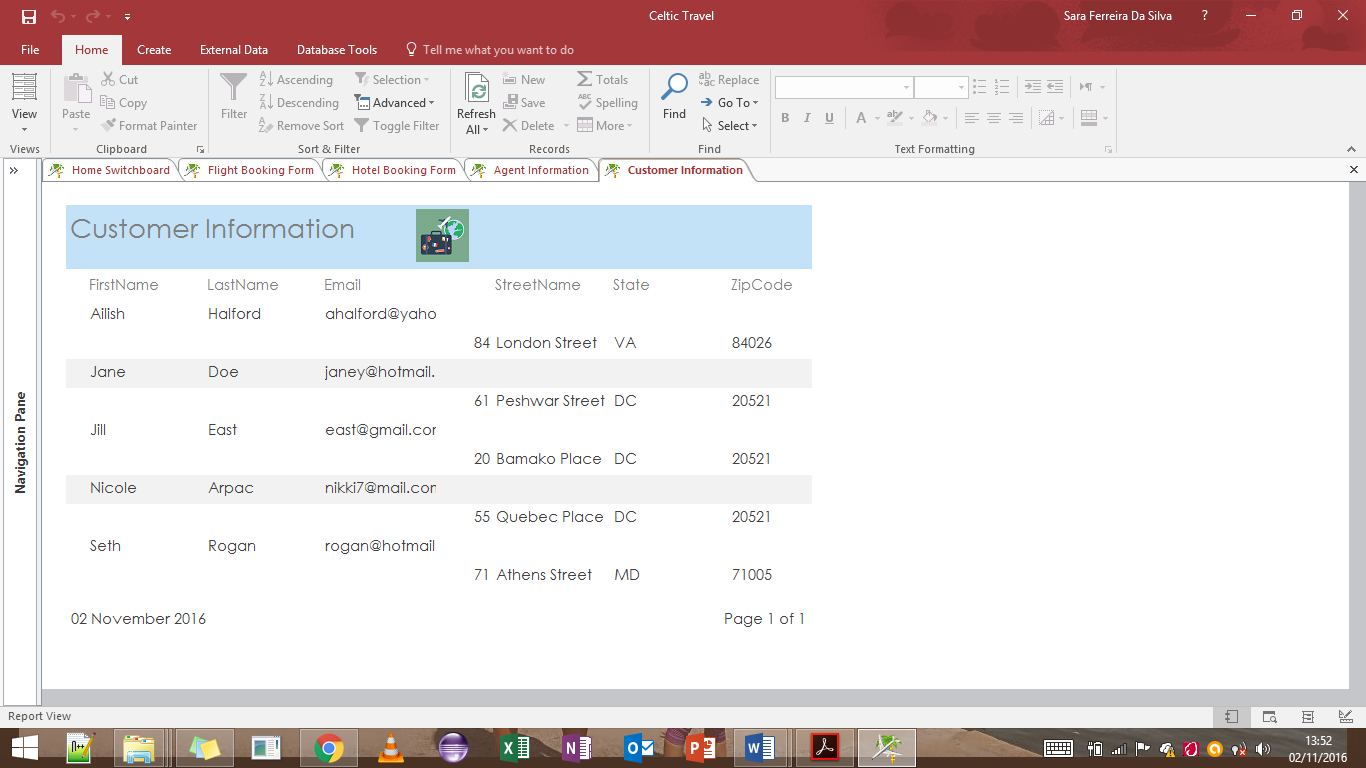
Hotel Bookings Form



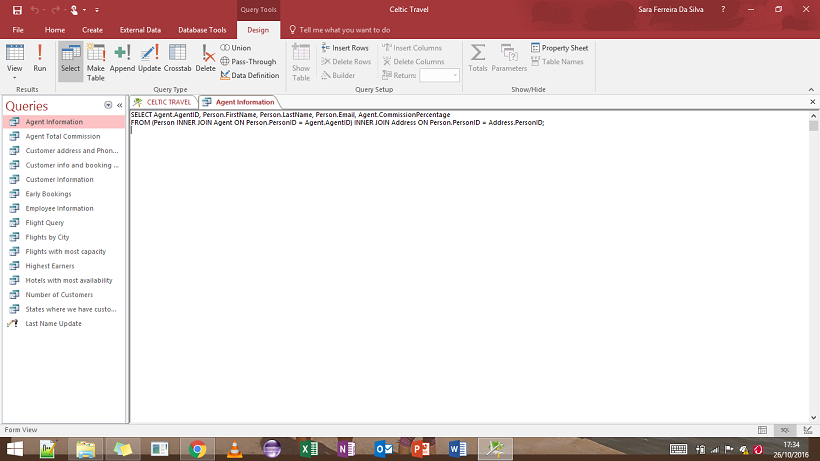
Agent Information Report



Customer Information Report



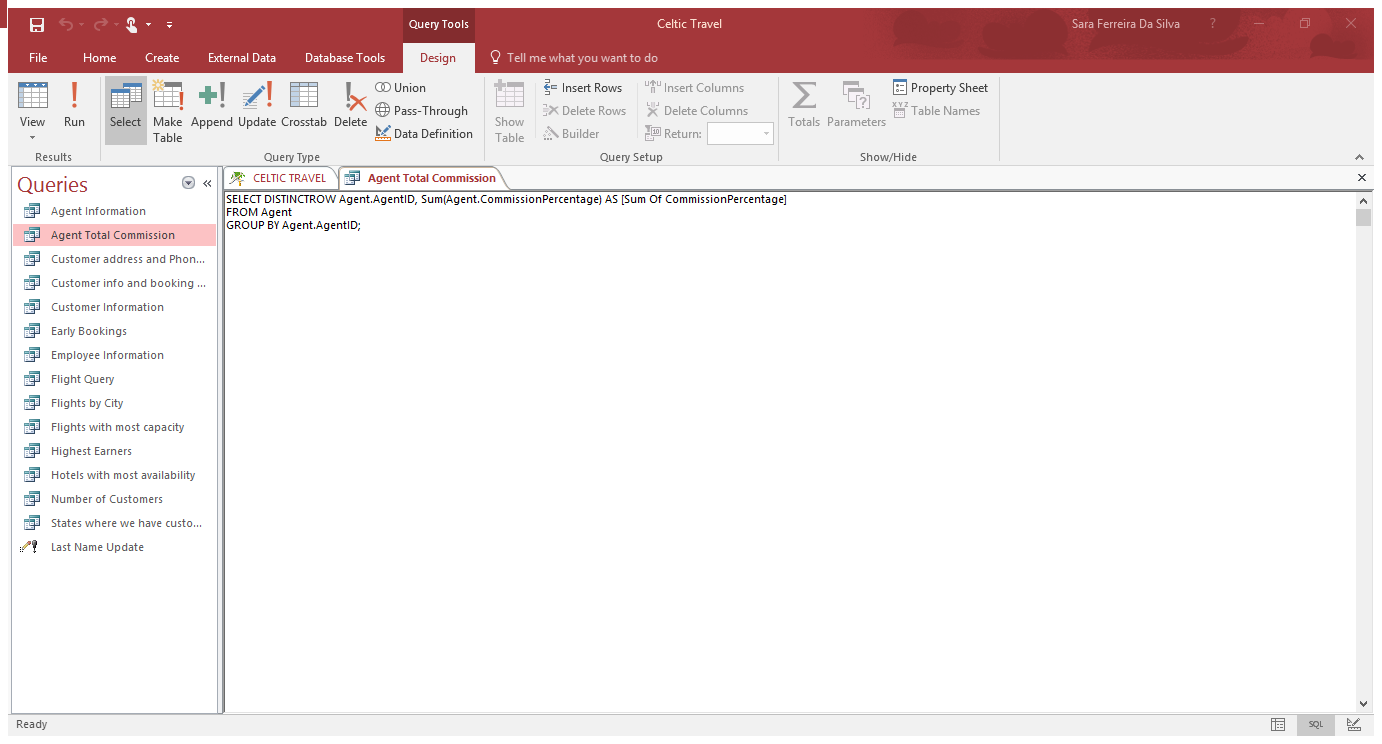
Agent Information Query

****

SELECT Agent.AgentID, Person.FirstName, Person.LastName, Person.Email, Agent.CommissionPercentage

FROM (Person INNER JOIN Agent ON Person.PersonID = Agent.AgentID) INNER JOIN Address ON Person.PersonID = Address.PersonID;

Agent Total Commission Query

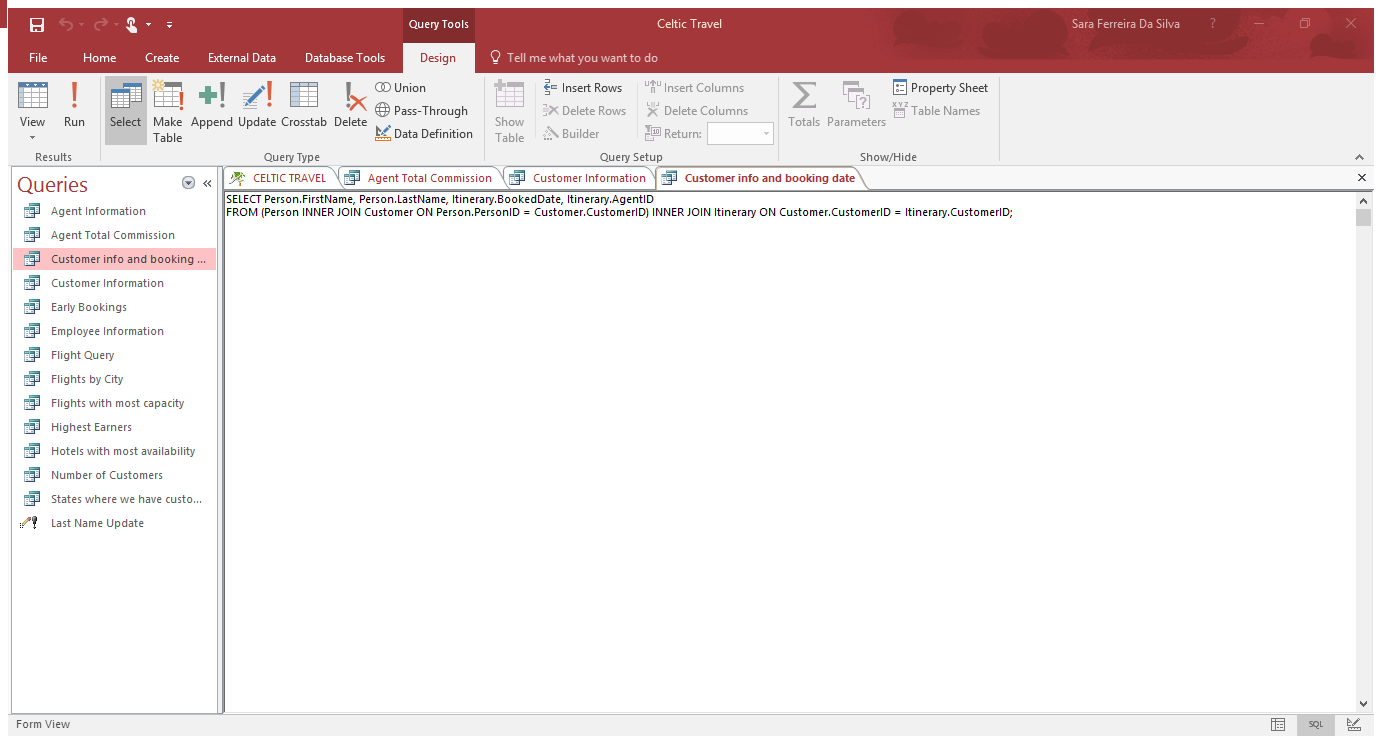


SELECT DISTINCTROW Agent.AgentID, Sum(Agent.CommissionPercentage) AS [Sum Of CommissionPercentage]

FROM Agent

GROUP BY Agent.AgentID;

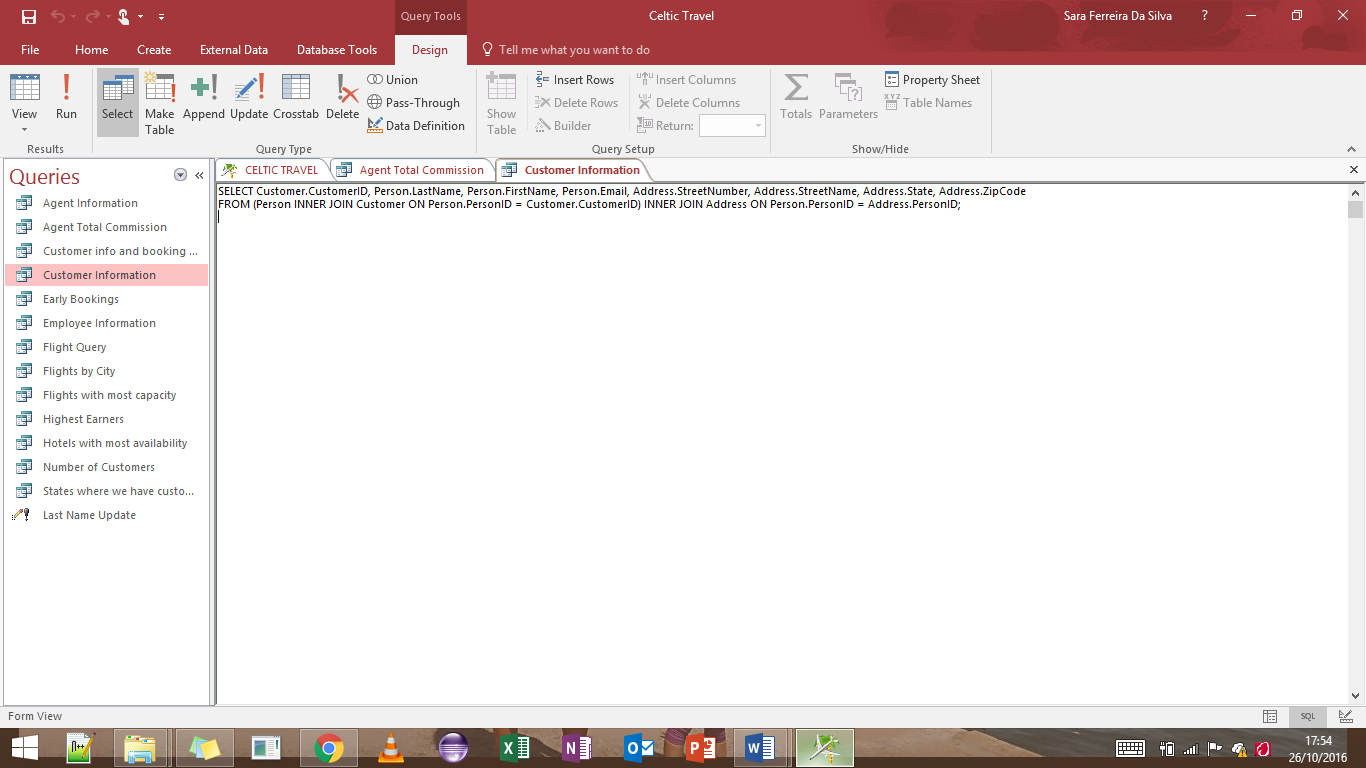
Customer Information and Booking Date Query



SELECT Person.FirstName, Person.LastName, Itinerary.BookedDate, Itinerary.AgentID

FROM (Person INNER JOIN Customer ON Person.PersonID = Customer.CustomerID) INNER JOIN Itinerary ON Customer.CustomerID = Itinerary.CustomerID;

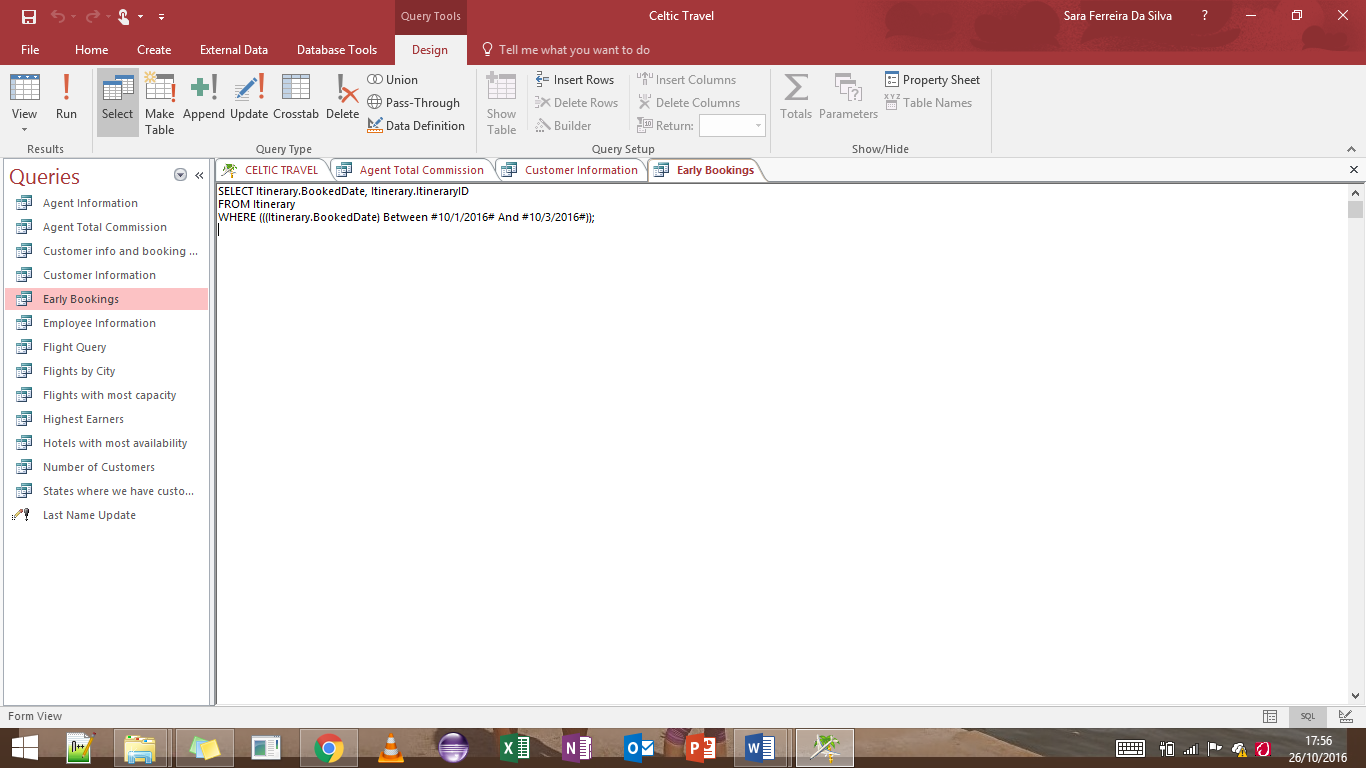
Customer General Information Query



SELECT Customer.CustomerID, Person.LastName, Person.FirstName, Person.Email, Address.StreetNumber, Address.StreetName, Address.State, Address.ZipCode

FROM (Person INNER JOIN Customer ON Person.PersonID = Customer.CustomerID) INNER JOIN Address ON Person.PersonID = Address.PersonID;

Early Bookings Query

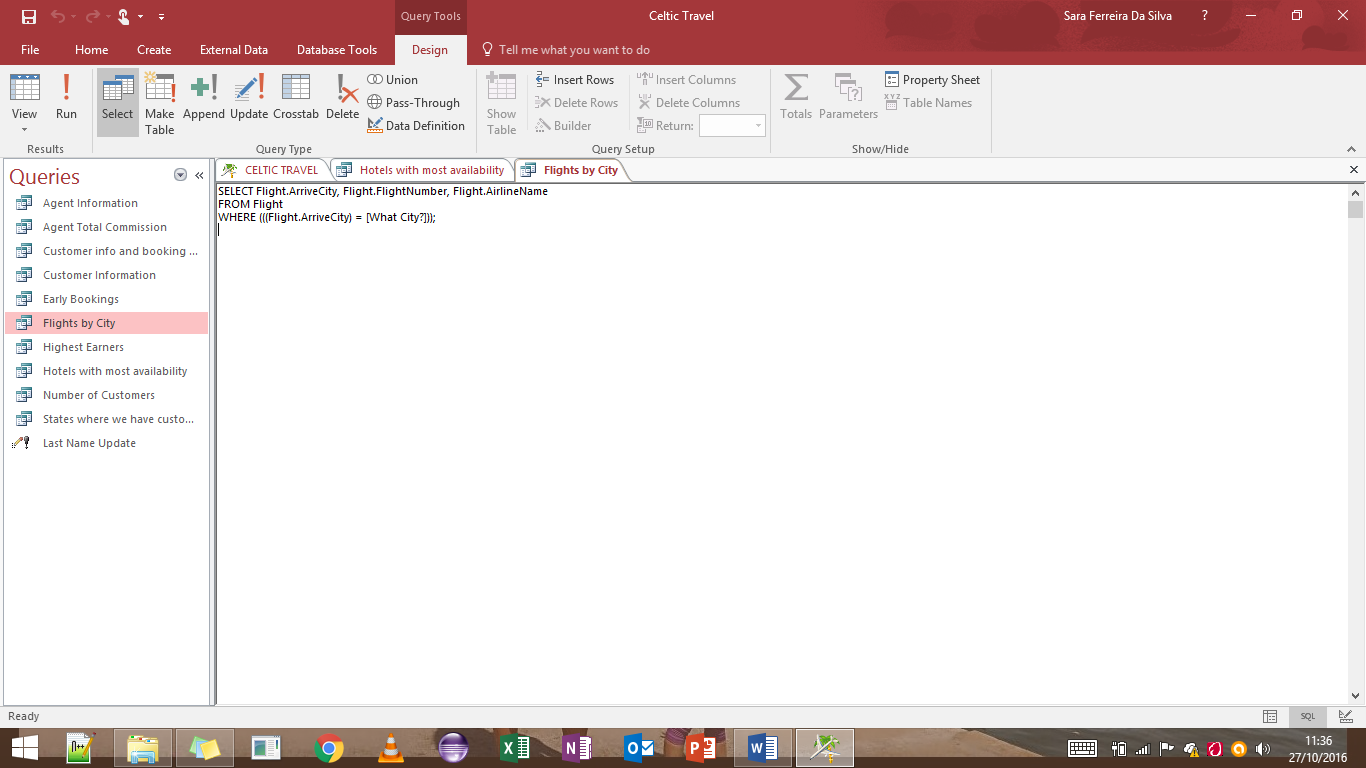


SELECT Itinerary.BookedDate, Itinerary.ItineraryID

FROM Itinerary

WHERE (((Itinerary.BookedDate) Between #10/1/2016# And #10/3/2016#));

Flights by City

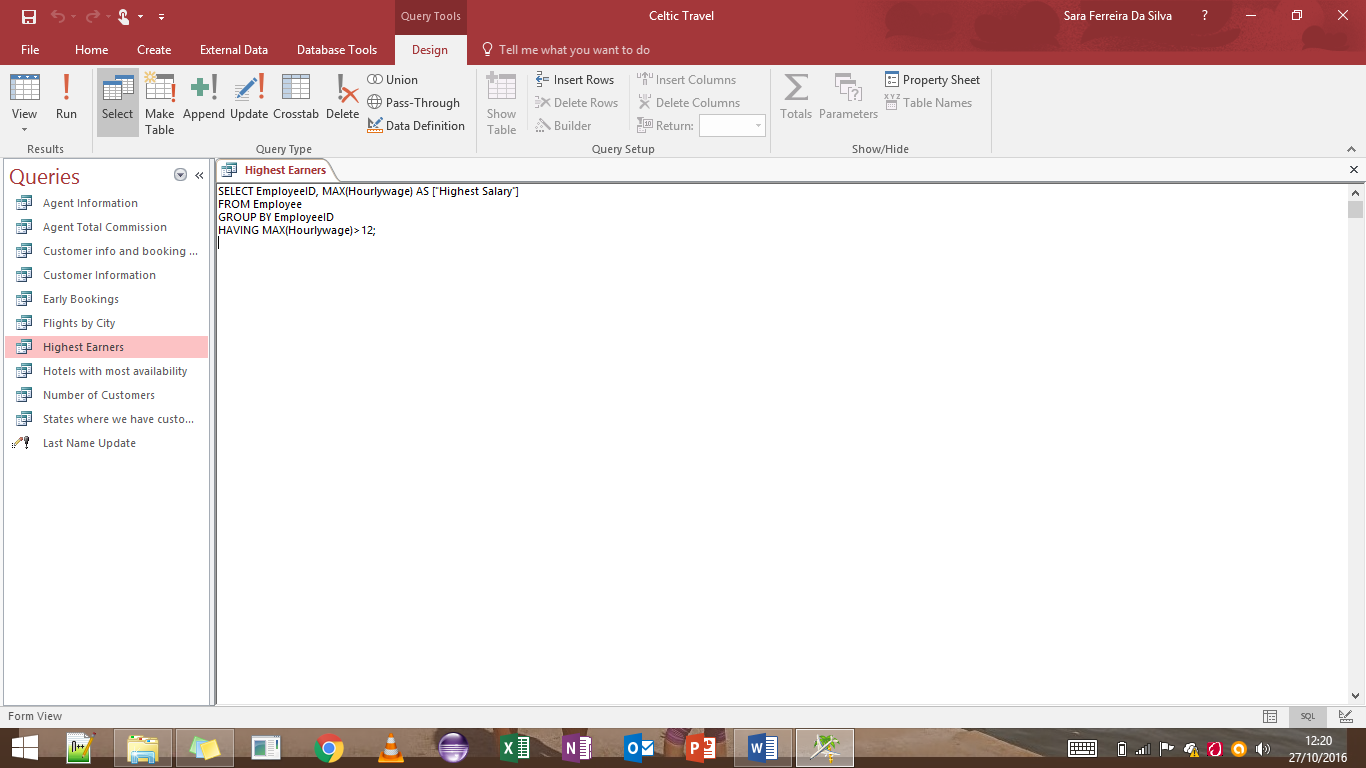
****

SELECT Flight.ArriveCity, Flight.FlightNumber, Flight.AirlineName

FROM Flight

WHERE (((Flight.ArriveCity) = [What City?]));

Highest Earners Query

****

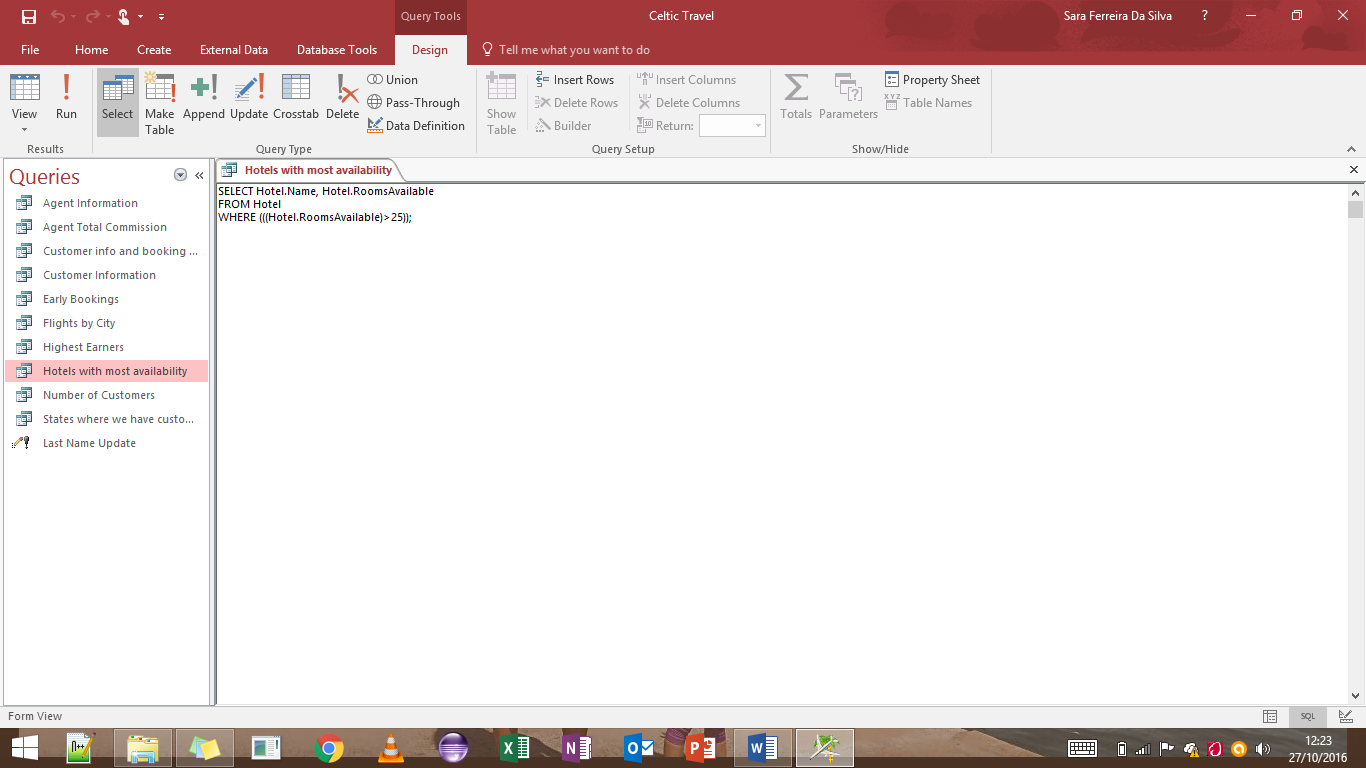
SELECT EmployeeID, MAX(Hourlywage) AS ["Highest Salary"]

FROM Employee

GROUP BY EmployeeID

HAVING MAX(Hourlywage)>12;

Hotels with most availability Query

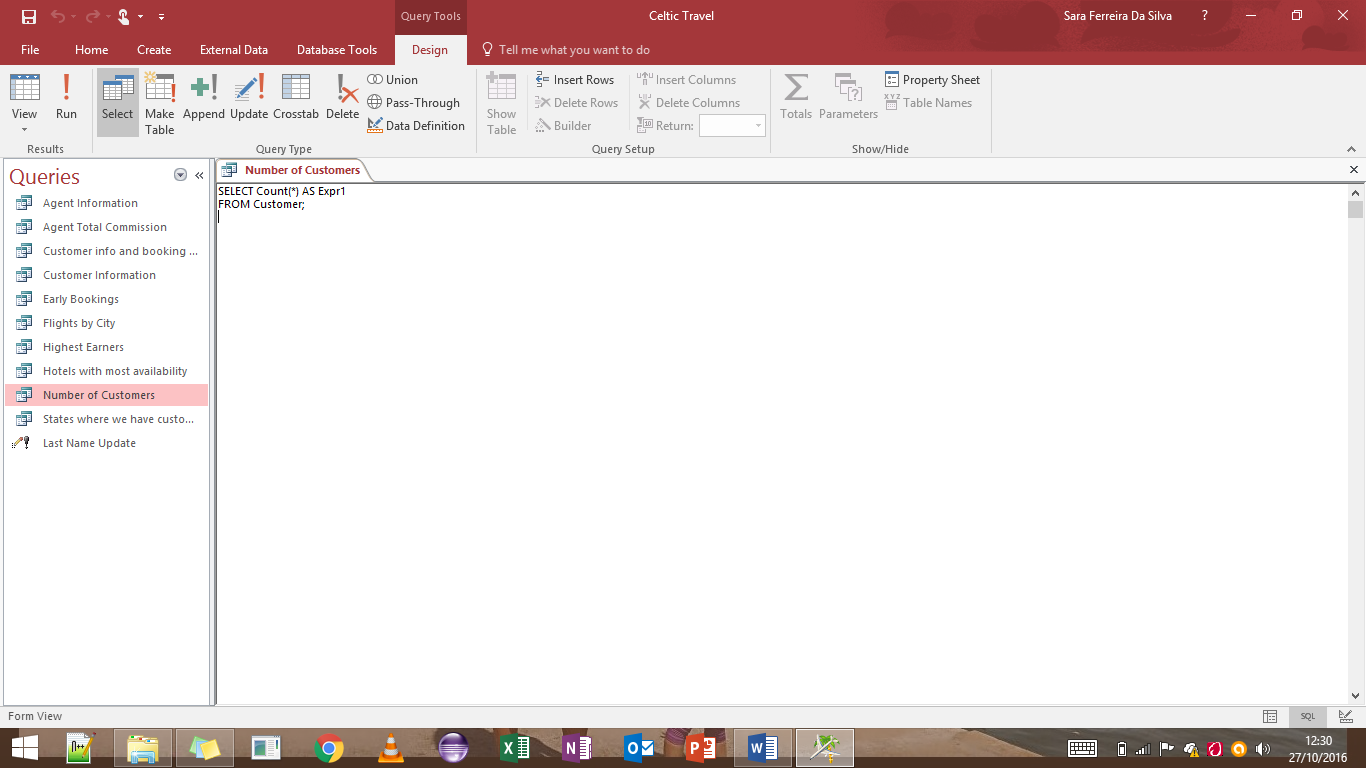
****

SELECT Hotel.Name, Hotel.RoomsAvailable

FROM Hotel

WHERE (((Hotel.RoomsAvailable)>25));

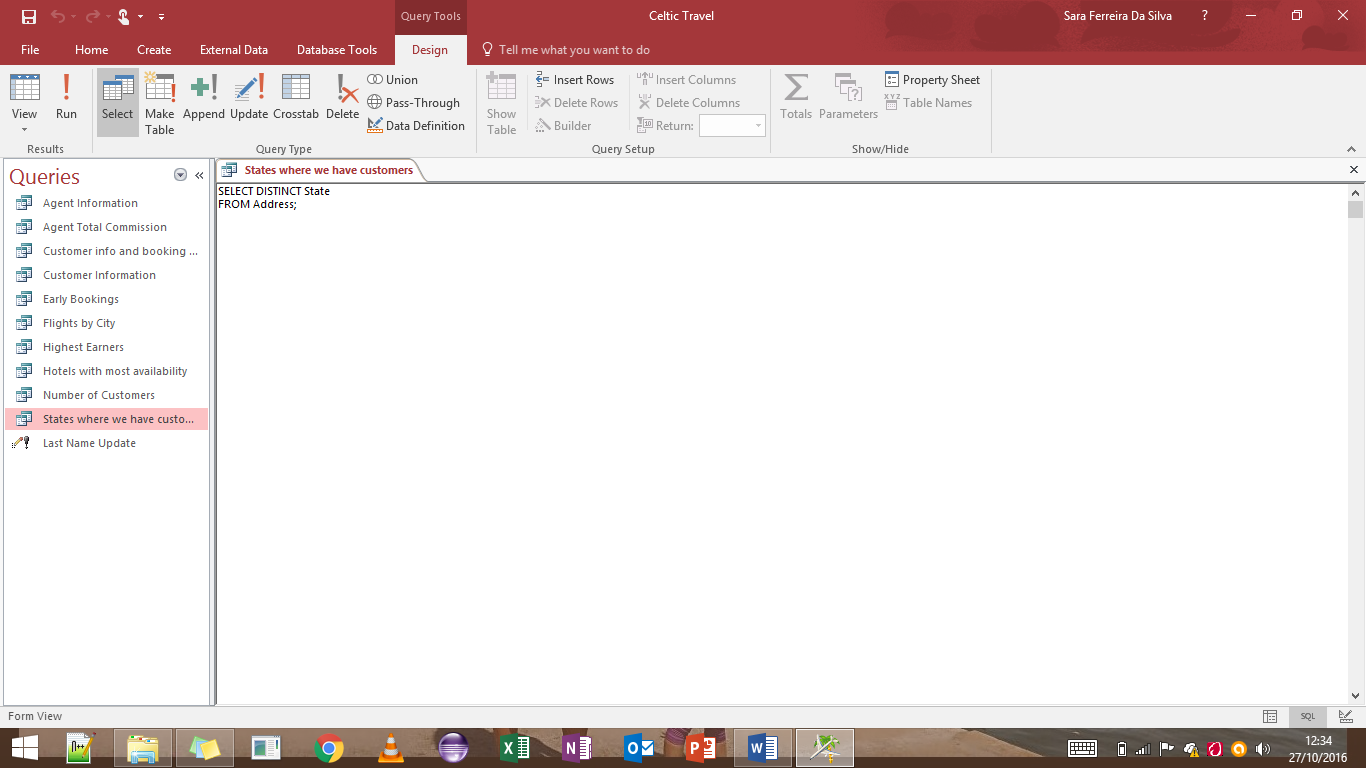
Number of Customers

****

SELECT Count(\*) AS Expr1

FROM Customer;

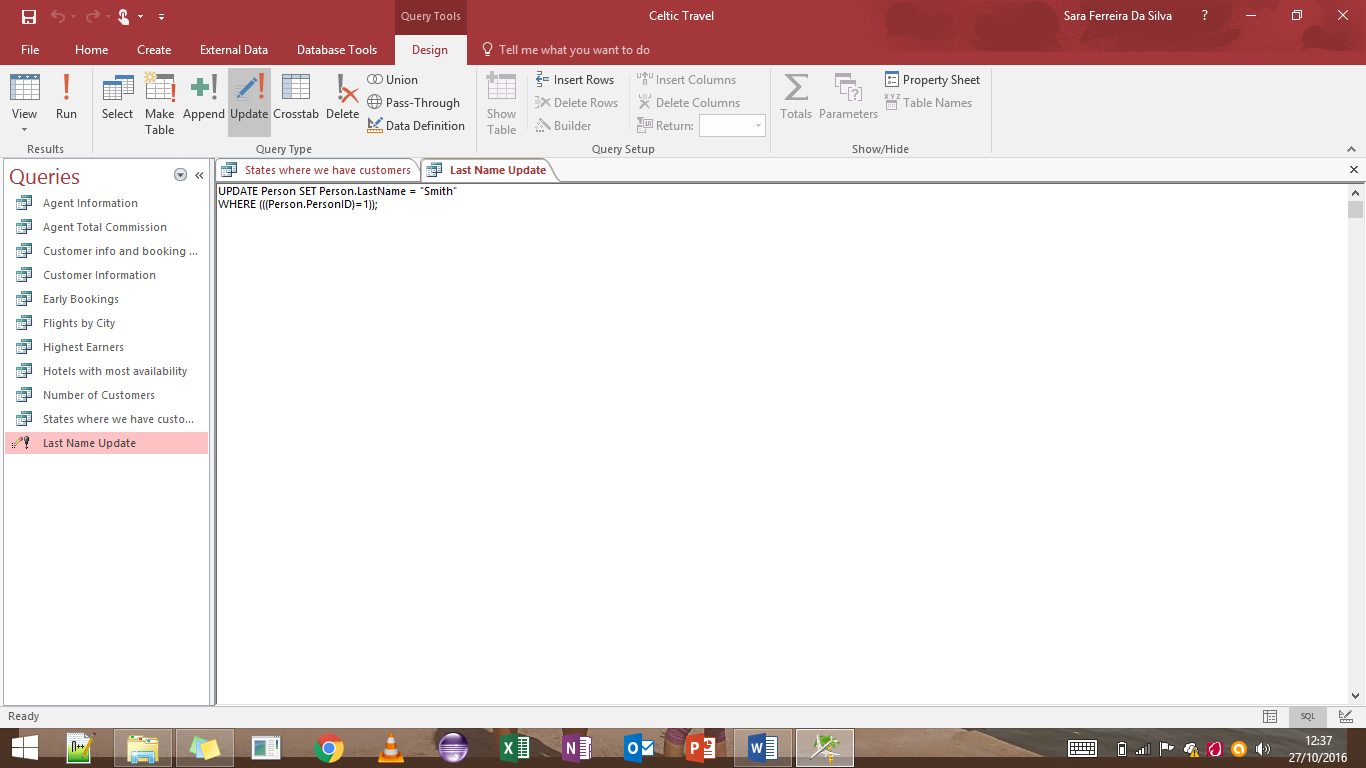
States where we have customers Query

****

SELECT DISTINCT State

FROM Address;

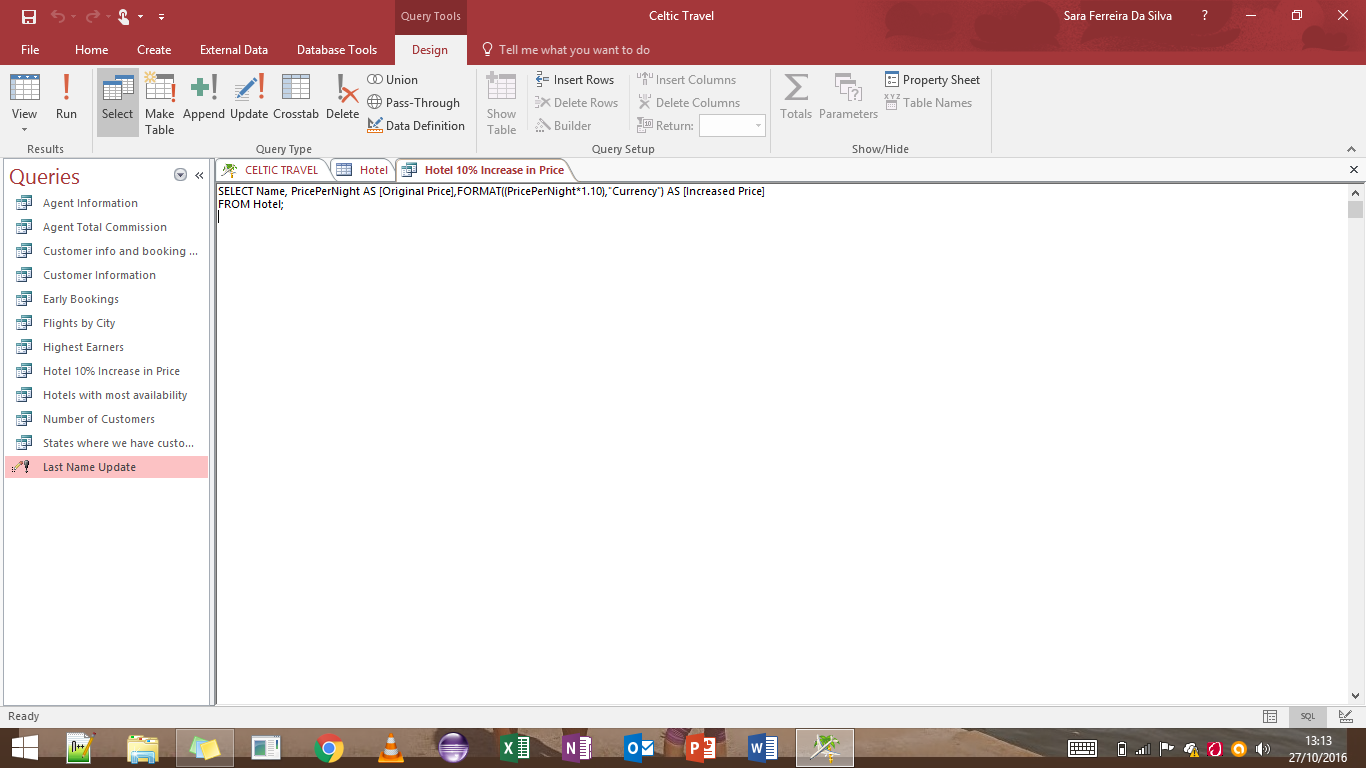
Last name Update Query

****

UPDATE Person SET Person.LastName = "Smith"

WHERE (((Person.PersonID)=1));

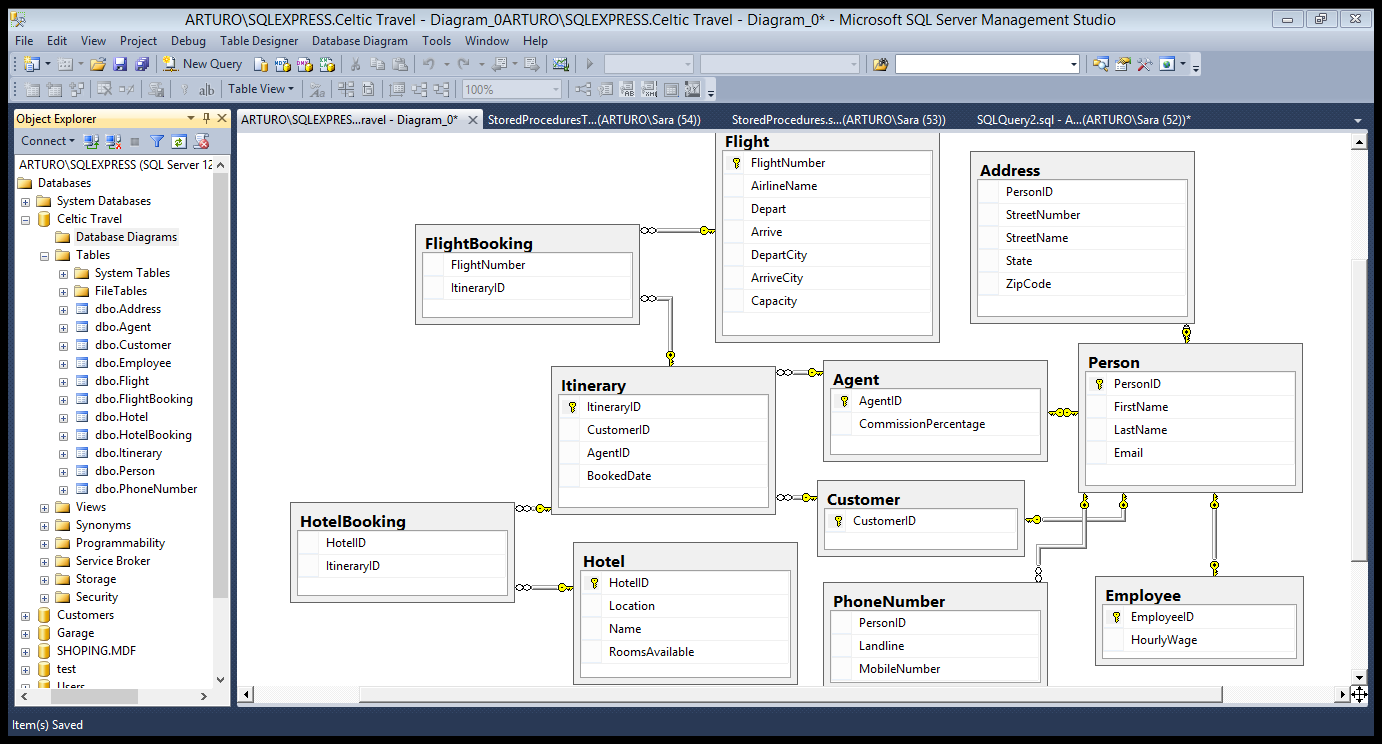
Hotel 10% Increase in Price Query

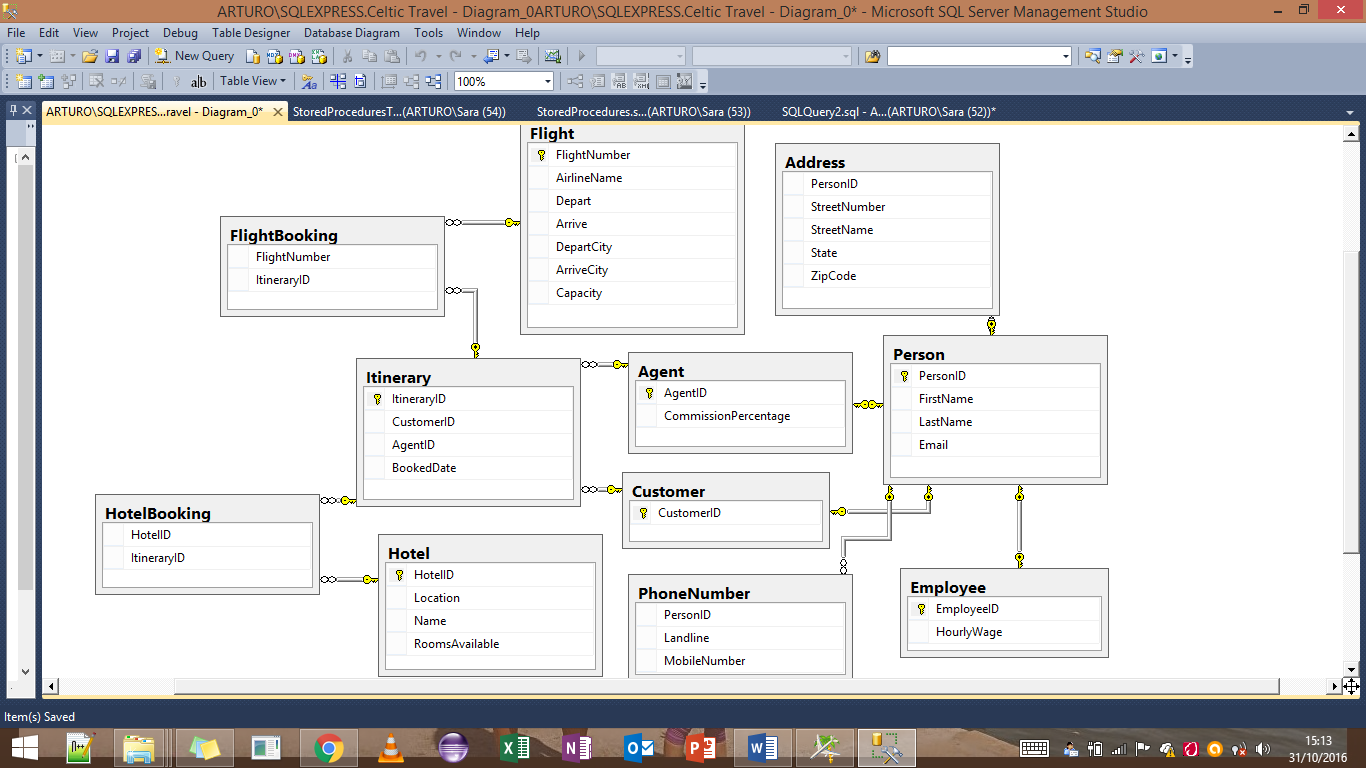
****

SELECT Name, PricePerNight AS [Original Price],FORMAT((PricePerNight\*1.10),"Currency") AS [Increased Price]

FROM Hotel;

# Appendix (MS SQL Server Database)





Code for Creating Tables:

CREATE TABLE Person(

PersonID int IDENTITY(1,1) NOT NULL,

FirstName varchar(20) NOT NULL,

LastName varchar(20) NOT NULL,

Email varchar(255) NOT NULL,

CONSTRAINT pk\_Person PRIMARY KEY (PersonID)

)

CREATE TABLE Agent(

AgentID int NOT NULL,

CommissionPercentage decimal(4,2),

CONSTRAINT pk\_Agent PRIMARY KEY(AgentID)

)

ALTER TABLE Agent

ADD CONSTRAINT fk\_Agent FOREIGN KEY(AgentID)

REFERENCES Itinerary(AgentID)

CREATE TABLE PhoneNumber (

PersonID int NOT NULL,

Landline varchar(20) ,

MobileNumber varchar(20),

CONSTRAINT fk\_PhoneNumber\_Person FOREIGN KEY(PersonID)

REFERENCES Person(PersonID)

)

CREATE TABLE Address (

PersonID int NOT NULL,

StreetNumber varchar(55),

StreetName varchar(20),

State varchar(20),

ZipCode varchar(6),

CONSTRAINT fk\_Address\_Person FOREIGN KEY (PersonID)

REFERENCES Person(PersonID)

)

CREATE TABLE Customer(

CustomerID int NOT NULL,

CONSTRAINT pk\_Customer PRIMARY KEY(CustomerID),

CONSTRAINT fk\_Customer FOREIGN KEY(CustomerID)

REFERENCES Person(PersonID)

)

CREATE TABLE Itinerary(

ItineraryID int IDENTITY(1,1) NOT NULL,

CustomerID int NOT NULL,

AgentID int NOT NULL,

BookedDate date NULL

)

ALTER TABLE Itinerary

ADD CONSTRAINT fk\_Itinerary FOREIGN KEY(AgentID)

REFERENCES Agent(AgentID)

CREATE TABLE Employee(

EmployeeID int NOT NULL,

HourlyWage decimal(4,2);

CONSTRAINT pk\_Employee PRIMARY KEY (EmployeeID)

)

ALTER TABLE Employee

ADD

CONSTRAINT fk\_Employee FOREIGN KEY(EmployeeID)

REFERENCES Person(PersonID)

CREATE TABLE Hotel(

HotelID int IDENTITY(1,1) NOT NULL,

Location varchar(20) NOT NULL,

Name varchar(20) NOT NULL,

RoomsAvailable int NOT NULL,

CONSTRAINT pk\_Hotel PRIMARY KEY(HotelID)

)

CREATE TABLE HotelBooking (

HotelID int NOT NULL,

ItineraryID int NOT NULL,

CONSTRAINT fk\_HotelBooking\_Hotel FOREIGN KEY (HotelID)

REFERENCES Hotel(HotelID),

CONSTRAINT fk\_HotelBooking\_Itinerary FOREIGN KEY (ItineraryID)

REFERENCES Itinerary(ItineraryID)

)

CREATE TABLE Flight(

FlightNumber int NOT NULL,

AirlineName varchar(20),

Depart datetime NOT NULL,

Arrive datetime NOT NULL,

DepartCity varchar(15),

ArriveCity datetime NOT NULL,

Capacity int NOT NULL,

CONSTRAINT pk\_Flight PRIMARY KEY (FlightNumber)

)

CREATE TABLE FlightBooking (

FlightNumber int NOT NULL,

ItineraryID int NOT NULL,

CONSTRAINT fk\_FlightBooking\_Flight FOREIGN KEY (FlightNumber)

REFERENCES Flight(FlightNumber),

CONSTRAINT fk\_FlightBooking\_Itinerary FOREIGN KEY (ItineraryID)

REFERENCES Itinerary(ItineraryID)

)

ALTER TABLE Itinerary

ALTER COLUMN BookedDate date NULL

ALTER TABLE Flight

ALTER COLUMN ArriveCity varchar(15) NOT NULL

Populating the tables using INSERT:

INSERT INTO dbo.Address

(PersonID, StreetNumber, StreetName, State, ZipCode)

VALUES

(1, N'20 ', N'Bamako Place', N'DC', N'20521')

(2, N'71', N'Athens Place', N'MD', N'71005')

(3, N'84', N'London Street', N'VA', N'84026')

(4, N'55', N'Quebec Place', N'DC', N'20521')

(5, N'61', N'Peshwar Street', N'DC', N'20521')

(6, N'24', N'Nouakchott Place', N'DC', N'20521')

(7, N'41', N'Sidney Street', N'MD', N'71005')

(8, N'32', N'Hermosillo Road', N'DC', N'20521')

(9, N'73', N'Easton Road', N'VA', N'84026')

(10, N'90', N'Cedar Place', N'DC', N'20521')

(11, N'27', N'Patrick Street', NULL, NULL)

(12, N'16', N'Mulberry Road', NULL, NULL)

(13, N'85', N'Ashton Gree', NULL, NULL)

(14, N'32', N'Frederick Road', NULL, NULL)

(15, N'44', N'South Place Street', NULL, NULL)

INSERT INTO dbo.Agent

(AgentID, CommissionPercentage)

VALUES

(6, CAST(100.00 AS Decimal(5, 2)))

(7, CAST(120.00 AS Decimal(5, 2)))

(8, CAST(135.00 AS Decimal(5, 2)))

(9, CAST(140.00 AS Decimal(5, 2)))

10, CAST(100.00 AS Decimal(5, 2)))

INSERT INTO dbo.Customer

(CustomerID)

VALUES

(1)

(2)

(3)

(4)

(5)

INSERT INTO dbo.Employee

(EmployeeID, HourlyWage)

VALUES

(11, CAST(15.00 AS Decimal(4, 2)))

(12, CAST(12.00 AS Decimal(4, 2)))

(13, CAST(12.00 AS Decimal(4, 2)))

(14, CAST(15.00 AS Decimal(4, 2)))

(15, CAST(15.00 AS Decimal(4, 2)))

INSERT INTO dbo.Flight

(FlightNumber, AirlineName, Depart, Arrive, DepartCity, ArriveCity, Capacity)

VALUES

(823, N'KLM', CAST(N'2016-10-07' AS Date), CAST(N'2016-10-14' AS Date), N'Washington', N'Chicago', 300)

(834, N'Airlingus', CAST(N'2016-10-08' AS Date), CAST(N'2016-10-15' AS Date), N'Washington', N'New York', 315)

(845, N'TAM', CAST(N'2016-10-09' AS Date), CAST(N'2016-10-16' AS Date), N'Washington', N'London', 330)

(856, N'EASYJET', CAST(N'2016-10-10' AS Date), CAST(N'2016-10-17' AS Date), N'Washington', N'Toronto', 345)

(867, N'IBERIA', CAST(N'2016-10-11' AS Date), CAST(N'2016-10-18' AS Date), N'Washington', N'Chicago', 360)

(878, N'Airlingus', CAST(N'2016-10-12' AS Date), CAST(N'2016-10-19' AS Date), N'Washington', N'New York', 300)

(889, N'TAM', CAST(N'2016-10-13' AS Date), CAST(N'2016-10-20' AS Date), N'Washington', N'London', 315)

(890, N'KLM', CAST(N'2016-10-14' AS Date), CAST(N'2016-10-21' AS Date), N'Washington', N'Toronto', 335)

INSERT INTO dbo.FlightBooking

(FlightNumber, ItineraryID)

VALUES

(823, 3)

(834, 4)

(845, 5)

(856, 6)

(867, 7)

INSERT INTO dbo.Hotel

(HotelID, Location, Name, RoomsAvailable)

VALUES

(1, N'Chicago', N'Sheraton', 40)

(2, N'New York', N'Hilton', 35)

(3, N'London', N'Plaza', 45)

(4, N'Toronto', N'Four Seasons', 25)

(5, N'New York', N'Manhattan', 30)

INSERT INTO dbo.HotelBooking

(HotelID, ItineraryID)

VALUES

(1, 3)

(2, 4)

(3, 5)

(4, 6)

(5, 7)

INSERT INTO dbo.[Itinerary

([ItineraryID], [CustomerID], [AgentID], [BookedDate])

VALUES

(3, 1, 6, CAST(N'2016-01-10' AS Date))

(4, 2, 7, CAST(N'2016-02-10' AS Date))

(5, 3, 8, CAST(N'2016-03-10' AS Date))

(6, 4, 9, CAST(N'2016-04-10' AS Date))

(7, 5, 10, CAST(N'2016-05-10' AS Date))

INSERT INTO dbo.Person

(PersonID, FirstName, LastName, Email)

VALUES

(1, N'Jill', N'East', N'jillEast@hotmail.com')

(2, N'Seth', N'Rogan', N'seth7@gmail.com')

(3, N'Ailish', N'Halford', N'halhal@mail.com')

(4, N'Nicole', N'Smith', N'smiley40@gmail.com')

(5, N'Niall', N'Holmes', N'holN33@yahoo.com')

(6, N'Lilly', N'Dawson', N'lilaD23@gmail.com')

(7, N'Lauren', N'Arpac', N'laureynm@hotmail.com')

(8, N'Olivia', N'Ryan', N'ollyw@yahoo.com')

(9, N'Susan', N'West', N'susy@hotmail.com')

(10, N'Finn', N'Murray', N'finnick@yahoo.com')

(11, N'Timothy', N'Rielly', N'timmy@mail.com')

(12, N'Fred', N'Savage', N'freddie@hotmail.com')

(13, N'Valerie', N'Byrne', N'valval@gmail.com')

(14, N'Alice', N'Redmond', N'alally@hotmail.com')

(15, N'Luke', N'Mitchell', N'luke2016@gmail.com')

INSERT INTO dbo.PhoneNumber

(PersonID, Landline, MobileNumber)

VALUES

(1, N'01 876 5465', N'(085)951-7812')

(2, N'01 533 6479', N'(089)546-5466')

(3, N'01 525 9678', N'(087)369-1241')

(4, N'01 876 8558', N'(085)322-4114')

(5, N'01 876 9889', N'(086)389-7845')

(6, N'01 876 1551', N'(089)952-8664')

(7, N'01 533 7552', N'(089)546-6468')

(8, N'01 876 4212', N'(089)354-5664')

(9, N'01 525 8565', N'(087)357-4162')

(10, N'01 876 4575', N'(085)421-7369')

(11, N'01 876 6543', N'(086)344-5677')

(12, N'01 876 2334', N'(087)234-6321')

(13, N'01 876 4553', N'(085)122-3345')

(14, N'01 876 7642', N'(086)432-5678')

(15, N'01 876 4902', N'(087)322-1214')

Five stored procedures with parameters:

GetAgentInformation

CREATE PROCEDURE GetAgentInformation

@FirstName varchar(15) = NULL

AS

SELECT Agent.AgentID, Person.FirstName, Person.LastName, Person.Email, Agent.CommissionPercentage

FROM (Person INNER JOIN Agent ON Person.PersonID = Agent.AgentID) INNER JOIN Address ON Person.PersonID = Address.PersonID

WHERE FirstName = ISNULL(@FirstName, FirstName)

EXEC GetAgentInformation @FirstName = 'Lilly'

GetCustomerInfoAndBooking

CREATE PROCEDURE GetCustomerInfoAndBooking

@LastName varchar(15) = NULL

AS

SELECT Person.FirstName, Person.LastName, Itinerary.BookedDate, Itinerary.AgentID

FROM (Person INNER JOIN Customer ON Person.PersonID = Customer.CustomerID) INNER JOIN Itinerary ON Customer.CustomerID = Itinerary.CustomerID

WHERE LastName = ISNULL(@LastName, LastName)

EXEC GetCustomerInfoAndBooking @LastName = 'East'

GetHotelAvailabiliy

CREATE PROCEDURE GetHotelAvailability

@Name varchar(15) = NULL

AS

SELECT Hotel.Name, Hotel.RoomsAvailable

FROM Hotel

WHERE Name = ISNULL(@Name, Name)

EXEC GetHotelAvailability @Name = 'Sheraton'

GetCustomerAddress

CREATE PROCEDURE GetCustomerAddress

@LastName varchar(15) = NULL

AS

SELECT Customer.CustomerID, Person.LastName, Person.FirstName, Person.Email, Address.StreetNumber, Address.StreetName, Address.State, Address.ZipCode

FROM (Person INNER JOIN Customer ON Person.PersonID = Customer.CustomerID) INNER JOIN Address ON Person.PersonID = Address.PersonID

WHERE LastName = ISNULL(@LastName, LastName)

EXEC GetCustomerAddress @LastName = 'Rogan'

GetFlightByCity

CREATE PROCEDURE GetFlightByCity

@ArriveCity varchar(15) = NULL

AS

SELECT Flight.ArriveCity, Flight.FlightNumber, Flight.AirlineName

FROM Flight

WHERE ArriveCity = ISNULL(@ArriveCity,ArriveCity)

EXEC GetFlightByCity @ArriveCity = 'Toronto'

# Bibliography

Connolly, T.M., M, T. and Begg, C.E. (1998) Database systems: A practical approach to design, implementation, and management. 2nd edn. Harlow, England: Addison-Wesley.