**--SQL Scripts From One of My projects for a Hotel System**

DROP DATABASE IF EXISTS CAPSTONE5HOTELDB

CREATE DATABASE CAPSTONE5HOTELDB

--#endregion

--#region2----------------------CREATION OF TABLES

/\*Creation of the BOOKING TABLE. Please note that we have two foregin keys in the booking

table. RoomType is a foreign key referencing the RoomType in the Room Table.

Also RequestId is a a foreign key referencing the RequestId in the Request Table \*/

DROP TABLE IF EXISTS Booking;

CREATE TABLE Booking(

BookingId INT NOT NULL PRIMARY KEY,

Room VARCHAR(50),

RoomPrefix VARCHAR(50),

RoomType VARCHAR(50),

StartDate DATE,

EndDate DATE,

RequestId INT)

--The script below is to view the booking table to confirm if we have the proper title and

--framework for the table.

SELECT \* FROM Booking

--The script below gives us the opportunity to delete the table if we choose to.

DROP TABLE Booking

/\*Creation of the MENU TABLE. Please note that we do not have foreigns keys in the table \*/

DROP TABLE IF EXISTS Menu;

CREATE TABLE Menu(

MenuId INT NOT NULL PRIMARY KEY,

MenuName VARCHAR(50),

Price MONEY,

Category VARCHAR(50))

--The script below is to view the menu table to confirm if we have the proper title and

--framework for the table.

SELECT \* FROM Menu

--The script below gives us the opportunity to delete the table if we choose to.

DROP TABLE Menu

/\*Creation of the FOODORDERS TABLE. Please note that we have two foregin keys in the

booking table. Bill\_RoomType is a foreign key referencing the RoomType in the Room Table.

Also MenuId is a a foreign key referencing the MenuId in the Menu Table \*/

DROP TABLE IF EXISTS FoodOrders;

CREATE TABLE FoodOrders(

DestRoom VARCHAR(50),

BillRoom VARCHAR(50),

Bill\_roomPrefix VARCHAR(50),

Bill\_RoomType VARCHAR(50),

Date DATE,

Time TIME,

OrdersNumbers INT,

MenuId INT)

--The script below is to view the FoodOrders table to confirm if we have the proper title

--and framework for the table.

SELECT \* FROM FoodOrders

--The script below gives us the opportunity to delete the table if we choose to.

DROP TABLE FoodOrders

/\*Creation of the REQUEST TABLE. Please note that we have one foregin key in the request

table. RoomType is a foreign key referencing the RoomType in the Room Table.

\*/

DROP TABLE IF EXISTS Request;

CREATE TABLE Request(

RequestId INT NOT NULL PRIMARY KEY,

ClientName VARCHAR(50),

RoomType VARCHAR(50),

RequestType VARCHAR(50),

StartDate DATE,

EndDate DATE,

NumAdults INT,

NumChildren INT)

--The script below is to view the Request table to confirm if we have the proper title and

--framework for the table.

SELECT \* FROM Request

--The script below gives us the opportunity to delete the table if we choose to.

DROP TABLE Request

/\*Creation of the ROOM TABLE. Please note that we do not have any foreign keys in the room

table \*/

DROP TABLE IF EXISTS Room;

CREATE TABLE Room(

RoomId INT,

PricePerDay MONEY,

Capacity INT,

RoomType VARCHAR(50) NOT NULL PRIMARY KEY,

RoomPrefix TEXT)

--The script below is to view the Room table to confirm if we have the proper title and

--framework for the table.

SELECT \* FROM Room

--The script below gives us the opportunity to delete the table if we choose to.

DROP TABLE Room

--#endregion2

--#region3----------------------ALTERING THE TABLES TO SELECT THE FOREIGN KEYS

--The below script it to alter the BOOKING table to select the two foreign keys in the

--table and point them to their respective primary keys

ALTER TABLE Booking

ADD FOREIGN KEY(RoomType)REFERENCES Room(RoomType);

ALTER TABLE Booking

ADD FOREIGN KEY(RequestId)REFERENCES Request(RequestId);

--The below script it to alter the FOODORDERS table to select the two foreign keys in the

--table and point them to their respective primary keys

ALTER TABLE FoodOrders

ADD FOREIGN KEY(Bill\_RoomType)REFERENCES Room(RoomType);

ALTER TABLE FoodOrders

ADD FOREIGN KEY(MenuId)REFERENCES Menu(MenuId);

--The below script it to alter the REQUEST table to select the foreign key in the Request

--table and point it to its respective primary key in the Room table

ALTER TABLE Request

ADD FOREIGN KEY(RoomType)REFERENCES Room(RoomType);

--#endregion3

--#region4----------------------INSERTING DATA INTO ALL THE TABLES THAT HAVE BEEN CREATED

--Inserting data into the BOOKING TABLE. Please note that the FROM should point to the

--folder in your computer where you saved your files.

BULK INSERT Booking

FROM 'C:\Users\Edward Omovudu\Downloads\bookings (3) (1).csv'

WITH (

FORMAT = 'CSV',

FIRSTROW = 2

)

--The script below is to view the Booking table to confirm if the data has been loaded into

--the table

SELECT \* FROM Booking

--Inserting data into the MENU TABLE. Please note that the FROM should point to the folder

--in your computer where you saved your files.

BULK INSERT Menu

FROM 'C:\Users\Edward Omovudu\Downloads\Menu.csv'

WITH (

FORMAT = 'CSV',

FIRSTROW = 2

)

--The script below is to view the Menu table to confirm if the data has been loaded into

--the table

SELECT \* FROM Menu

--Inserting data into the FOOD ORDERS TABLE. Please note that the FROM should point to the

--folder in your computer where you saved your files.

BULK INSERT FoodOrders

FROM 'C:\Users\Edward Omovudu\Downloads\food\_orders (2) (1).csv'

WITH (

FORMAT = 'CSV',

FIRSTROW = 2

)

--The script below is to view the FoodOrders table to confirm if the data has been loaded

--into the table

SELECT \* FROM FoodOrders

--Inserting data into the REQUEST TABLE. Please note that the FROM should point to the

--folder in your computer where you saved your files.

BULK INSERT Request

FROM 'C:\Users\Edward Omovudu\Downloads\requests (3) (1).csv'

WITH (

FORMAT = 'CSV',

FIRSTROW = 2

)

--The script below is to view the Request table to confirm if the data has been loaded

--into the table

SELECT \* FROM Request

--Inserting data into the ROOM TABLE. Please note that the FROM should point to the folder

--in your computer where you saved your files.

BULK INSERT Room

FROM 'C:\Users\Edward Omovudu\Downloads\Rooms.csv'

WITH (

FORMAT = 'CSV',

FIRSTROW = 2

)

--The script below is to view the Room table to confirm if the data has been loaded into

--the table

SELECT \* FROM Room

--#endregion

--#region5-----------------------ALTERING TABLE FOODORDERS to have a primary key. Somehow

--when this step was attempted earlier, it threw up an error message

ALTER TABLE FoodOrders

ADD FoodOrdersId INT PRIMARY KEY IDENTITY(1, 1)

--The script below is to view the foodorders table to find out if our script above achieved

--what we want to achieve.

SELECT \* FROM FoodOrders

--The script below (which has been commented on) gives up the option to drop the

--FoodOrdersId column on the FoodOrder table.

ALTER TABLE FoodOrders DROP COLUMN FoodOrdersId

--#endregion

--#region6-----------------------VIEWING TOP THREE(3) FROM ALL THE TABLES

SELECT TOP 3 \* FROM Booking

SELECT TOP 3 \* FROM Menu

SELECT TOP 3 \* FROM FoodOrders

SELECT TOP 3 \* FROM Request

SELECT TOP 3 \* FROM Room

--#endregion

FROM Menu

LEFT JOIN FoodOrders ON Menu.MenuId = FoodOrders.MenuId

LEFT JOIN Booking ON FoodOrders.BillRoom = Booking.Room

LEFT JOIN Request ON Booking.RequestId =  Request.RequestId

LEFT JOIN Room ON Request.RoomType = Room.RoomType

--#endregion

--#region10 Showing the ones that fell outside the request---51583

SELECT Booking.BookingId,Booking.Room, Booking.RoomType, Booking.StartDate, Booking.EndDate,FoodOrders.Date ,FoodOrders.Time ,FoodOrders.OrdersNumbers ,FoodOrders.MenuId,Menu.MenuName, Menu.Price, Menu.Category, Request.RequestId, Request.ClientName, Request.RequestType, Request.NumAdults, Request.NumChildren, Room.PricePerDay, Room.Capacity

---INTO FulltableWithFoodOrders1

FROM Menu

LEFT JOIN FoodOrders ON Menu.MenuId = FoodOrders.MenuId

LEFT JOIN Booking ON FoodOrders.BillRoom = Booking.Room

LEFT JOIN Request ON Booking.RequestId =  Request.RequestId

LEFT JOIN Room ON Request.RoomType = Room.RoomType

WHERE FoodOrders.Date NOT BETWEEN Booking.StartDate and Booking.EndDate

--#endregion

--#region11 To select into table

SELECT column1, column2, ...

INTO new\_table

FROM old\_table

WHERE condition;

---To update tables

UPDATE TableName

SET ColumnName = NULL;

SELECT \* FROM FulltableWithFoodOrders

SELECT \* FROM FulltableWithFoodOrders1

---To update the table and remove the foodorder values

UPDATE FulltableWithFoodOrders1

SET Date = NULL,

Time = NULL,

Ordersnumbers = NULL,

MenuId = NULL,

MenuName = NULL,

Price = NULL,

Category = NULL

SELECT \* FROM FulltableWithFoodOrders1

--

--Using UNION ALL gives us back our complete table

SELECT \*

INTO Capstone5HotelPBI

FROM FulltableWithFoodOrders

UNION ALL

SELECT \* FROM Capstone5HotelPBI

---Using UNION gave us double the rows- not want we want

SELECT \* FROM FulltableWithFoodOrders

UNION

SELECT \* FROM FulltableWithFoodOrders1

SELECT TOP 5 \* FROM Booking

SELECT TOP 5 \* FROM Request

--#endregion

--#region12 Our Tables

SELECT \* FROM Menu

SELECT \* FROM FoodOrders

SELECT \* FROM Booking

SELECT \* FROM Request

SELECT \* FROM Room

--#endregion

--#region Manipulations