CSE 305

ASSIGNMENT 2

- BOGDAN OLEINIKOV
- NAFIUL AZIM
- SYED JAMAL

Table of Contents

1.Introduction - The Travel Agency Queries

The following document is the documentation of the Travel Agency's Create Database Query and its Create Table Queries that shall define the database system for the proposed travel agency. This document will show the queries that create the entity tables and then briefly describe each entity's attributes while noting the constraints that are included in their create table queries.

2.The Create Database Query

Listed below is the query for creating the travel agency database.

```
DROP DATABASE IF EXISTS Travel_agency;
CREATE DATABASE Travel_agency;
USE Travel_agency;
```

Create Database Query1: Since we didn't have any dedicated database for this project, we created a database named Travel_agency which will contain all the tables we need for this project.

3.The Create Table Queries

Listed Below are the Create Table Queries for the Database.

3.1 Location

```
CREATE TABLE Location (
  Location_ID INT NOT NULL AUTO_INCREMENT,
  Street_Address Varchar(6),
  City Varchar(50),
  State Varchar(50),
  Country Varchar(50),
  PRIMARY KEY (Location_ID)
);
ALTER TABLE Location AUTO_INCREMENT=10001;
```

Create Table Query1: Location represents locations in the database that people will, ultimately, be travelling to and from when using the services of the travel agency.

- **Location**: We have created the Location relation(table) including 5 attributes.
    - **Location_ID**: has integer type it has `NOT NULL` and `AUTO_INCREMENT` as this is our primary key for this table we decided the Location_ID will be an automatically generated integer. As Primary Key can't be NULL we stated this will not contain any NULL value using NOT NULL clause.
    - **Street_Address**: has Varchar(50). The street address of Location can be a string with up to 50 characters.
    - **City**: has Varchar(50). The city of Location can be a string with up to 50 characters.
    - **State**: has Varchar(50). The state of Location can be a string with up to 50 characters.
    - **Country**: has Varchar(50). The country of Location can be a string with up to 50 characters.

      To make the Location_ID look better we are using a 5-digit auto generated integer by using the command displayed below. Assuming the total number of locations this database can hold will be in between the range of 10001=99999.

      ```
      ALTER TABLE Location AUTO_INCREMENT=10001;
      ```

      Here in this line of code we are making sure our counting for location starts from 10001.

3.2 PaymentAccount

```
CREATE TABLE PaymentAccount (
  PaymentAccount_Number INT,
  Payment_Type Varchar(50),
  Card_Expirary_Date date NOT NULL,
  PRIMARY KEY (PaymentAccount_Number, Payment_Type),
  CHECK ( PaymentAccount_Number>=1000000000000000
  AND PaymentAccount_Number<=1000000000000000),
  CONSTRAINT Payment_Type_Const
  CHECK (Payment_Type IN ('Credit', 'Debit', 'PayPal'))
);
```

Create Table Query2: PaymentAccount represents payment accounts stored in the database of people who are using the travel agency to pay for trips that have been booked.

- **PaymentAccount**: contains 3 attributes.
    - **PaymentAccount_Number**: has integer value. The constraints for this is it has to be a 16-digit integer. As we know, all the payment accounts have a 16-digit account number.
    - **Payment_Type**: can be up to 6 characters and is of the VARCHAR type. Since our system will only accept payment from Credit, Debit or from a PayPal account. `CHECK (Payment_Type IN ('Credit', 'Debit', 'PayPal'))` will restrict the user from inputting values other than these 3 provided values.
    - **Card_Expirary_Date**: contains the expiration date of `date` type in MMYYYY format.

      Since PaymentAccount_Number from a different Payment_Type might contain the same account number, alone PaymentAccount_Number cannot be a primary key. For that reason, the primary key we chose is the composite key Payment_Type and PaymentAccount_Number.

3.3 Employee

```
CREATE TABLE Employee (
  Employee_ID INT NOT NULL AUTO_INCREMENT,
  Date_Joined DATETIME DEFAULT CURRENT_TIMESTAMP,
  Employee_Name Varchar(50) NOT NULL,
  Contact Varchar(50),
  Salary FLOAT NOT NULL,
  Supervisor_ID INT,
  PRIMARY KEY (Employee_ID),
  FOREIGN KEY Supervisor_ID references Employee(Employee_ID),
  CHECK( Employee_ID>10000 AND Employee_ID<100000),
  CHECK(Salary>0),
  CHECK( Supervisor_ID>10000 AND Supervisor_ID<100000)
);
ALTER TABLE Employee AUTO_INCREMENT=10001;
```

Create Table Query3: Employee represents the employees that work for the travel agency that are stored in the travel agency database.

- **Employee**: contains 6 attributes.
    - **Employee_ID**: is an autogenerated 5-digit primary key. Which starts incrementing from 10001 using similar technique for counting Location_ID.
    - **Date_Joined**: has the DATETIME type because we want to make default Date_Joined as the day of entry in database. In order to do this, we used the MySQL function called CURRENT_TIMESTAMP. Which has DATETIME return type.
    - **Employee_Name**: has a type of VARCHAR contains at maximum 50-character spaces for Employee_Name. And since it is not a good idea to have NULL in the Employee_Name, the value of this field cannot be NULL.
    - **Contact**: This is a type of VARCHAR contains at maximum 50-character spaces.
    - **Salary**: We can assume an employee can get his payment in dollars and cent for this reason it has a type of FLOAT. Also all the employees has to be paid off for their work So this can't be NULL.
    - **Supervisor_ID**: is a reference to an instance in the employee class has same value and constraints as Employee_ID.

3.4 Trip

```
CREATE TABLE Trip (
  Reservation_Number INT  NOT NULL AUTO_INCREMENT,
  Date_Reserved DATETIME DEFAULT CURRENT_TIMESTAMP ,
  Date_Departure DATETIME DEFAULT CURRENT_TIMESTAMP,
  Date_Return DATETIME DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (Reservation_Number),
  CHECK(Reservation_Number< 100000000000000),
  CHECK(Date_Departure>=Date_Reserved),
  CHECK(Date_Departure<Date_RETURN)

);
ALTER TABLE Trip AUTO_INCREMENT = 10000000000000;
```

Create Table Query4: Trip represents trips stored in the database that people using the travel agency services have booked.

- **Trip**: contains 4 attributes.
    - **Reservation_Number**: is a unique reservation number of type INT for each trip reservation. This is the PRIMARY KEY to Trip. It is designed to be 14-digit key.
    - **Date_Reserved**: is an attribute that has type DATETIME representing the time and date the trip was reserved. This field has constraint which requires the Date_Reserved to be before or on the date and time of date of departure.
    - **Data_Departure**: is an attribute that has type DATETIME representing the time and date of departure. This field has constraint which requires the reservation to be made before or on the date of departure. Also, departure has to be before the date and time of returning from the trip.
    - **Date_Returned**: is an attribute that has type DATETIME representing the date and time of returning from the trip. The constraint checks for the value to be greater than the date and time of departure.

3.5 Passengers

```
CREATE TABLE Passenger (
  Passenger_ID INT NOT NULL AUTO_INCREMENT,
  Passenger_Name Varchar(50),
  Passenger_Gender Varchar(50),
  Passenger_BirthDate DATETIME,
  Passenger_RegisterDate DATETIME DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (Passenger_ID),
  CHECK(Passenger_BirthDate >=1900
  OR Passenger_BirthDate <= Passenger_RegisterDate),
  CONSTRAINT Gender_Const
  CHECK (Passenger_Gender IN ('Male', 'Female', 'Other'))
);
ALTER TABLE Passenger AUTO_INCREMENT = 10000000000000;
```

Create Table Query5: Passenger represents passengers stored in the database they are people that have used the travel agency.

- **Passenger**: contains 5 attributes.
    - **Passenger_ID**: is an 18-digit INT value that represents the unique identification number of the passenger in the database. This is the primary key to this entity.
    - **Passenger_Name**: VARCHAR representing the name of the passenger.
    - **Passenger_Gender**: VARCHAR value representing the gender of the passenger. The gender must be of male, female, or other.
    - **Passenger_BirthDate**: DATETIME value representing the birthdate of the passenger. This value cannot before the year 1900.
    - **Passenger_RegisterDate**: DATETIME value representing the register date into the travel agency databse of the passenger. This value cannot occur before the birthdate of the person.

3.6 PaymentReciept

```
CREATE TABLE PaymentReciept (
  Receipt_Number INT NOT NULL AUTO_INCREMENT,
  Reservation_Number INT,
  Date_Paid DATETIME DEFAULT CURRENT_TIMESTAMP,
  Total_Paid DECIMAL(10,4),
  Trip_Summary Varchar(5000),
  PaymentAccount_Number INT,
  Payment_Type Varchar(50),
  PRIMARY KEY (Receipt_Number),
  FOREIGN KEY(Reservation_Number)
  references Trip(Reservation_Number)
  FOREIGN KEY  (PaymentAccount_Number, Payment_Type)
  references PaymentAccount(PaymentAccount_Number, Payment_Type),
  CHECK(Total_Paid >= 0),
  CHECK(Receipt_Number >= 1000000000000000
  AND Receipt_Number < 10000000000000000),
 );
ALTER TABLE PaymentReciept AUTO_INCREMENT = 1000000000000000;
```

Create Table Query6: PaymentReciept represents the reciepts of trips stored in the travel agency database.

- **PaymentReciept**: This table will contain the information a trip receipt after making a payment(transaction). It contains 7 attributes.
    - **PaymentAccount_Number**: This is a 16 digit autogenerated integer starting from 10000000000000000. The constraint for this has been set in such a way so that it can't be more the 16 digits.
    - **Reservation_Number**: This is a foreign key references to Reservation_Number attribute of table Trip.
    - **Date_Paid**: DATETIME type. It represents when the trip was paid.
    - **Total_Paid**: DECIMAL type. It represents the total amount paid for the trip.
    - **Trip Summary**: is of varchar type. Field contains a simple description of the trip.
    - **PaymentAccount_Number & PaymentAccount_Type**: composite foreign key of PaymentAccount.

3.7 TransportTickets

```
CREATE TABLE TransportTickets (
 TransportTicket_Number INT,
 Source_Location INT,
 Final_Destination INT,
 ModeOfTransport Varchar(50),
 Route_Number INT,
 Company Varchar(50),
 Fare Decimal(10,4),
 PRIMARY KEY (TransportTicket_Number),
 FOREIGN KEY (Source_Location) references Location(Location_ID),
 FOREIGN KEY (Final_Destination) references Location(Location_ID)
);
```

Create Table Query7: TransportTickets represents the transportation tickets of groups stored in the travel agency database.

- **TransportTickets**: This entity has 7 attributes. It represents the transport ticket used for the trip.
    - **TransportTicketNumber**: is the primary key to the TransportTickets entity. This attribute has type INT representing the ticket number.
    - **Source_Location**: INT representing the location_ID for the origin location's primary key of the trip.
    - **Final_Destination**: INT referencing the location_ID for the destination location's primary key of the trip.
    - **ModeOfTransport**: VARCHAR representing the transportation used for the trip.
    - **Route_Number**: INT representing the route number for the origin type of transportation.
    - **Company**: VARCHAR representing the company of the mode of transportation
    - **Fare**: DECIMAL value representing the fare for the ticket. A 10-digit number with 4 decimal places so that prices can be calculated and then rounded.

3.8 Flight

```
CREATE TABLE Flight (
  Flight_ID INT NOT NULL AUTO_INCREMENT,
  Flight_Number INT,
  Flight_Carrier Varchar(50),
  Flight_Fare Decimal(10,4),
  Flight_Class Varchar(10),
  Source_Airport_ID Varchar(50),
  Destination_Airport_ID Varchar(50),
  Flight_Capacity INT,
  Flight_Dep_Time DATETIME,
  Flight_Arv_Time DATETIME,
  PRIMARY KEY (Flight_ID),
  CHECK (Flight_Class IN ('Business', 'Economy')),
  CHECK(Flight_Fare>=0),
  CHECK(Flight_Capacity>0 AND Flight_Capacity<305),
  CHECK(Flight_Dep_Time< Flight_Arv_Time)
);
ALTER TABLE Flight AUTO_INCREMENT = 100000000000000000;
```

Create Table Query8: Flight represents flights stored in the database that people using the travel agency services may look up and book during a trip.

- **Flight**: This entity represents Airplane/Flight as mode of Transport. It has ten attributes.
    - **Flight_ID**: It is an18-digit INT value that uniquely identifies the flight. This is the primary key and will auto increment each time with subsequent entry to the database starting from 100000000000000000.
    - **Flight_Number**: INT value representing the flight number.
    - **Flight_Career**: VARCHAR value representing the flight's career company.
    - **Flight_Fare**: DECIMAL value representing the fare for the flight. A 10-digit number with 4 decimal places so that prices can be calculated and then rounded.
    - **Flight_Class**: VARCHAR representing the luxury level of the flight. It can be either economy of business class.
    - **Source_Airport**: VARCHAR representing the flight's takeoff airport name.
    - **Destination_Airport**: VARCHAR representing the landing airport's name.
    - **Flight_Capacity**: INT representing the maximum number of passengers allowed in the flight. We are limiting the capacity to 305 passenger for a particular flight.
    - **Flight_Dep_Time**: DATETIME representing the timestamp of the departure date and time of the flight.

- **Flight_Arv_Time**: DATETIME representing the timestamp of arrival date/time.

3.9 GroupTrip

```
CREATE TABLE GroupTrip (
  Group_ID INT NOT NULL AUTO_INCREMENT,
  Reservation_Number INT,
  Group_Size INT,
  Purpose Varchar(50),
  PRIMARY KEY (Group_ID),
  FOREIGN KEY (Reservation_Number)
  references Trip(Reservation_Number),
  CHECK(Group_ID< 100000000),
  CHECK(Group_Size>0 AND Group_Size<=6)
);
ALTER TABLE GroupTrip AUTO_INCREMENT = 10000001;
```

Create Table Query9: GroupTrip represents the group of people in a certain trip that is stored in the travel agency database.

- **GroupTrip**: contains 4 attributes.
    - **Group_ID**: Integer type auto incremented 8 digits long primary key identifies each group uniquely. As this is a primary key it can't be NULL.
    - **Reservation_Number**: Is a foreign key refers to the Trip relation. The constraints for this is same as the attribute it refers to.
    - **Group_Size**: Contains the number of people in a group. As most of the travel agencies defines a Group can have no more than 6 people. We are following this idea so the constraint for this attribute is it must be a number in between 1 and 6 (inclusive).
    - **Purpose**: for each tuple it contains 50 character long spaces to describe the purpose of the trip.

3.10 TransportationOfGroup

```
CREATE TABLE TransporationOfGroup (
  Transportation_ID INT NOT NULL AUTO_INCREMENT,
  Group_ID INT,
  TransportTicket_Number INT,
  Class Varchar(50),
  PRIMARY KEY (Transportation_ID),
  FOREIGN KEY (Group_ID) references GroupTrip(Group_ID),
  FOREIGN KEY (TransportTicket_Number)
  references TransportTickets(TransportTicket_Number),
  CHECK(Transportation_ID<100000000),
  CHECK(Group_ID>10000000 AND Group_ID< 100000000),
  CHECK(TransportTicket_Number>10000000
  AND  TransportTicket_Number< 100000000)
);
ALTER TABLE TransportationOfGroup AUTO_INCREMENT = 10000001;
```

Create Table Query10: TransportationOfGroup represents the transportations of a certain group stored in the travel agency database.

- **TransportationOfGroup**: contains 4 attributes.
    - **Transportation_ID**: is an auto incremented 8-digit integer starting from 10000001. The constraints has been set in such a way it can't contain more than 8 digits.
    - **Group_ID**: is of integer type. It is a foreign key which is referencing Group_ID of GroupTrip relation.
    - **TransportTicket_Number**: is another foreign key of integer type. It references TransportTicket_Number of the TransportTickets relation. The constraint for this attribute is that it has to be an 8-digit number starting from 10000001
    - **Class**: is of type VARCAR(50) meaning it can contain string of 50 character spaces.

3.11 Accommodation

```
CREATE TABLE Accommodation (
  Accommodation_ID INT NOT NULL AUTO_INCREMENT,
  Accommodation_Name Varchar(50),
  Accommodation_Type Varchar(50),
  Quantity_Available INT NOT NULL,
  Rate_Per_Night DECIMAL(10, 4) NOT NULL,
  Facilities Varchar(1000),
  PRIMARY KEY (Accommodation_ID),
  CHECK(Rate_Per_Night>=0),
  CHECK( Quantity_Available>=0),
  CHECK( Accommodation_ID< 1000000000000000000)
);
ALTER TABLE Accommodation  AUTO_INCREMENT = 100000000000000000;
```

Create Table Query11: Accommodation represents accommodations stored in the database that people using the travel agency services may look up and book during a trip.

- **Accommodation**: will contain information of each location of accommodation and can have multiple rows for 1 location of different kind of facilities and contains 6 attributes.
    - **Accommodation_ID**: This is an auto incremented number 8-digit integer starting from 1000000000000000. The constraints have been set in such a way it can't contain more than 16 digits.
    - **Accommodation_Name**: will contain the name of the Accommodation and has 50 spaces to hold this information for each row.
    - **Accommodation_Type**:: will contain the type (hotel, motel etc) of the Accommodation and has 50 spaces to hold this information for each row.
    - **Quantity_Available**: contains information of how many more person can the place be able to accommodate at a certain time. It is of integer type. The only restriction for this is it can't be less than 0 and cannot be of NULL value.
    - **Rate_Per_Night**: will contain the payment amount for 1 night. It is of DECIMAL type must be greater than 0 and this can't be null. This will have altogether 10 digits of which last 4 digits are after decimal point (for precision in calculation).
    - **Facilities**: is of varcar(1000), which means it can contain up to 50 characters. It will have the information of facilities that the unique Accommodation_ID provides.

3.12 AccommodationReservation

```
CREATE TABLE AccommodationReservation (
  Confirmation_Number INT NOT NULL AUTO_INCREMENT,
  Accommodation_ID INT, Group_ID INT,
  Check_In_Date DATETIME,
  Check_Out_Date DATETIME,
  Total_Amount DECIMAL(10,4),
  PRIMARY KEY (Confirmation_Number),
  FOREIGN KEY(Accommodation_ID)
  references Accommodation(Accommodation_ID),
  Foreign KEY ( Group_ID) references GroupTrip(Group_ID),
  CHECK(Total_Amount>=0),
  CHECK( Accommodation_ID >= 10000000000000000
  AND Accommodation_ID < 100000000000000000),
  CHECK( Confirmation_Number< 100000000000000000),
  CHECK(Group_ID>100000000 AND Group_ID <100000000),
  CHECK(Check_In_Date< Check_Out_Date)
);
ALTER TABLE AccommodationReservation
AUTO_INCREMENT = 10000000000000000;
```

Create Table Query12: AccommodationReservation represents group reservations for accommodations that are stored in the travel agency database.

- **AccommodationReservation**: This will contain the information of the group's accommodation. This relation contains 6 attributes.
    - **Confirmation_Number**: is a unique 16 digit autogenerated primary key. The constraints cannot be null and must be 16 digits. Using `ALTER TABLE query`
    - **Group_ID**: is a foreign key that refers to Group_ID of GroupTrip. Its constraints abide to its constraints listed in GroupTrip.
    - **Accommodation_ID**: is a foreign key refers to Accommodation_ID of Accommodation. Its constraints abide to its constraints listed in Accmomodation.
    - **Total_Amount**: records the amount of money that must be paid by a group for each accommodation they book. It must be greater than or equal to 0. It is of DECIMAL type.
    - **Check_In_Date**: is DATETIME type. It is the check-in time of accommodation.
    - **Check_Out_Date**: is DATETIME type. It is the check-out time of the accommodation. For constraints, Check_Out_Date must be greater than Check_In_Date.

3.13 Review

```
CREATE TABLE Review (
  Review_Number INT NOT NULL AUTO_INCREMENT,
  Group_ID INT,
  Passenger_ID INT,
  Rating INT,
  Detailed_Review Varchar(1000),
  PRIMARY KEY (Review_Number),
  FOREIGN KEY (Group_ID) references GroupTrip(Group_ID),
  FOREIGN KEY  (Passenger_ID) references Passenger(Passenger_ID),
  CHECK(Passenger_ID >=10000000000000
  AND Passenger_ID <100000000000000),
  CHECK(Group_ID > 100000000 AND Group_ID < 100000000),
  CHECK(Rating >=0 AND Rating < 5)
);
```

Create Table Query13: Review represents a review of a trip written by a passenger that is stored in the travel agency database.

- **Review**: contains 5 attributes.
    - **Review_Number**: is the primary key which is auto generated.
    - **Group_ID**: is a foreign key that references to GroupTrip. It has the same constraints of Group_ID in GroupTrip table.
    - **Passenger_ID**: is of INT type. It is a foreign key that references Passenger. It has the same constraints as Passenger_ID of Passenger. It uniquely identifies a single passenger that belongs to a specified group.
    - **Ratings**: is type INT and cannot be NULL. Passenger can rate ranging between 1 to 5 (inclusive) where higher number means more satisfaction.
    - **Datailed_Review**: is type VARCAR. Passenger can write here of their experience of the trip in a1000 character space.

3.14 Bus

```
CREATE TABLE Bus (
  Bus_ID INT NOT NULL AUTO_INCREMENT,
  Bus_Number INT,
  Bus_Company Varchar(50),
  Bus_Fare Decimal(10,4),
  Source_Bus_Stop Varchar(50) NOT NULL,
  Destination_Bus_Stop Varchar(50) NOT NULL,
  Bus_Capacity INT,
  Bus_Dep_Time DATETIME,
  Bus_Arv_Time DATETIME,
  PRIMARY KEY (Bus_ID),
  CHECK(Bus_Fare >= 0),
  CHECK(Bus_Capacity  >0 AND Bus_Capacity < 50),
  CHECK(Bus_Dep_Time < Bus_Arv_Time)
);
ALTER TABLE Bus AUTO_INCREMENT = 1000000000000000000;
```

Create Table Query14: Bus represents bus trips stored in the database that people using the travel agency services may look up and book during a trip.

- **Bus**: This table will serve as Bus database for the travel-agency. It will be modified by the bus agencies who are the clients of travel agency it has 9 attributes.
    - **Bus_ID**: is an auto-incremented 16-digit integer primary key for the table.
    - **Bus_Number**: this integer value is to hold the physical identification number of each bus (if a specific company has multiple bus). It can be NULL if a company has only 1 bus for a route.
    - **Bus_Company**: will hold up to 50-character long name of a Bus company.
    - **Bus_Fare**: holds the price of a certain bus trip. This attribute is of DECIMAL type must be greater than 0 and this can't be null. And this will have altogether 10 digits of which the last 4 digits are after the decimal point.
    - **Source_Bus_Stop**: is a 50-character long string that holds the address of the starting point of a bus route. This field can't be NULL.
    - **Destination_Bus_Stop**: is a 50-character long string that holds the address of the Destination point of a bus route. This field can't be NULL.
    - **Bus_Capacity**: is an integer that contains the number of seats left in a bus. It must be in between 0 and 50. As it is assumed that bus cannot accommodate more than 50 passengers at one time.
    - **Bus_Dep_Time**: is of DATETIME data type which means it can hold both date and time of a bus departure.
    - **Bus_Arv_Time**: is of DATETIME data type which means it can hold both date and time of a bus arrival at the destination.
      The constraint for these 2 attributes is Bus_Dep_Time must be less than Bus_Arv_Time.

3.15 Cruise

```
CREATE TABLE Cruise (
  Cruise_ID INT NOT NULL AUTO_INCREMENT,
  Route_Number INT,
  Cruise_Company Varchar(50),
  Cruise_Number INT,
  Source_Port Varchar(50),
  Destination_Port Varchar(50),
  Cruise_Fare Decimal(10,4),
  Cruise_Class varchar(10),
  Cruise_Capacity INT,
  Cruise_Dep_Time DATETIME,
  Cruise_Arv_Time DATETIME,
  PRIMARY KEY (Cruise_ID),
  CHECK (Cruise_Class IN ('Business', 'Economy')),
  CHECK(Cruise_Fare >= 0),
  CHECK(Cruise_Capacity > 0 AND Cruise_Capacity < 10000),
  CHECK(Cruise_Dep_Time < Cruise_Arv_Time)
);
ALTER TABLE Cruise AUTO_INCREMENT = 1000000000000000000;
```

Create Table Query15: Cruise represents cruise voyages stored in the database that people using the travel agency services may look up and book during a trip.

- **Cruise**: contains 11 attributes describing the cruise voyage.
    - **Cruise_ID**: INT representing the 18-digit auto incrementing primary key.
    - **Route_Number**: INT representing the route number which the Cruise follows.
    - **Cruise_Company**: VARCHAR representing the company who owns the cruise.
    - **Cruise_Number**: INT representing the cruise number.
    - **Source_Port**: VARCHAR representing the port of origin.
    - **Destination_Port**: VARCHAR representing the port of destination.
    - **Cruise_Class**: VARCHAR value representing the class of the cruise.
    - **Cruise_Fair**: The fair for the cruise as INT
    - **Cruise_Capacity**: INT below 10000 that represents cruise passenger capacity.
    - **Cruise_Dep_Time**: is of DATETIME data type which means it can hold both date and time of a cruise departure.
    - **Cruise_Arv_Time**: is of DATETIME data type which means it can hold both date and time of a cruise arrival at the destination.
      The constraint for these 2 attributes is Cruise_Dep_Time must be less than Cruise_Arv_Time.

3.16 Car

```
CREATE TABLE Car(
  Car_ID INT  NOT NULL AUTO_INCREMENT,
  Car_Number INT,
  Company Varchar(50),
  Park_Addr Varchar(50),
  Primary Key(Car_ID)
);
ALTER TABLE Car AUTO_INCREMENT = 100000000000000000;
```

Create Table Query16: Car represents car rental vehicles stored in the database that people using the travel agency services may look up and rent when booking a trip.

- **Car:** This entity represents Car as mode of Transport for the trip. It has five attributes.
    - **Car_ID**: This is the primary key to this entity. It is an INT value representing the ID of the car. This an 18-digit value starting from 100000000000000000 and auto increments with subsequent entry to the database.
    - **Car_Number**: This is an INT value representing the Number_Plate of the car. For simplicity we are allowing only digit values unlike real world's alphanumeric choice.
    - **Company**: This is the name of the company which the car belongs to. This is of type Varchar(50) meaning, a string value with maximum character of 50.
    - **Park_Addr**: VARCHAR representing the address of the designated parking lot.

3.17 GroupMembers

```
CREATE TABLE GroupMembers(
    Group_ID INT NOT NULL,
    Passenger_ID INT NOT NULL,
    FOREIGN KEY (Group_ID) references GroupTrip(Group_ID),
    FOREIGN KEY  (Passenger_ID) references Passenger(Passenger_ID),
    CHECK(Passenger_ID >=10000000000000
    AND Passenger_ID <100000000000000),
    CHECK(Group_ID>100000000 AND Group_ID <100000000)
);
```

Create Table Query17: GroupMembers represents the passengers that belong to a certain group stored in the travel agency database.

- **GroupMembers**: We have created this table to solve 'set valued' related issue for GroupTrip table. Group can consist of multiple members. But it is not possible to put all members ID in one attribute. For that reason, we created this table where each tuple as whole is a primary key. This relation has 2 attributes.
    - **Group_ID**: is a foreign key that refers to Group_ID of GroupTrip. Its constraints abide to its constraints listed in GroupTrip.
    - **Passenger_ID**: is a foreign key that refers to Passenger_ID of Passenger. Its constraints abide to its constraints listed in Passenger.