University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___2____

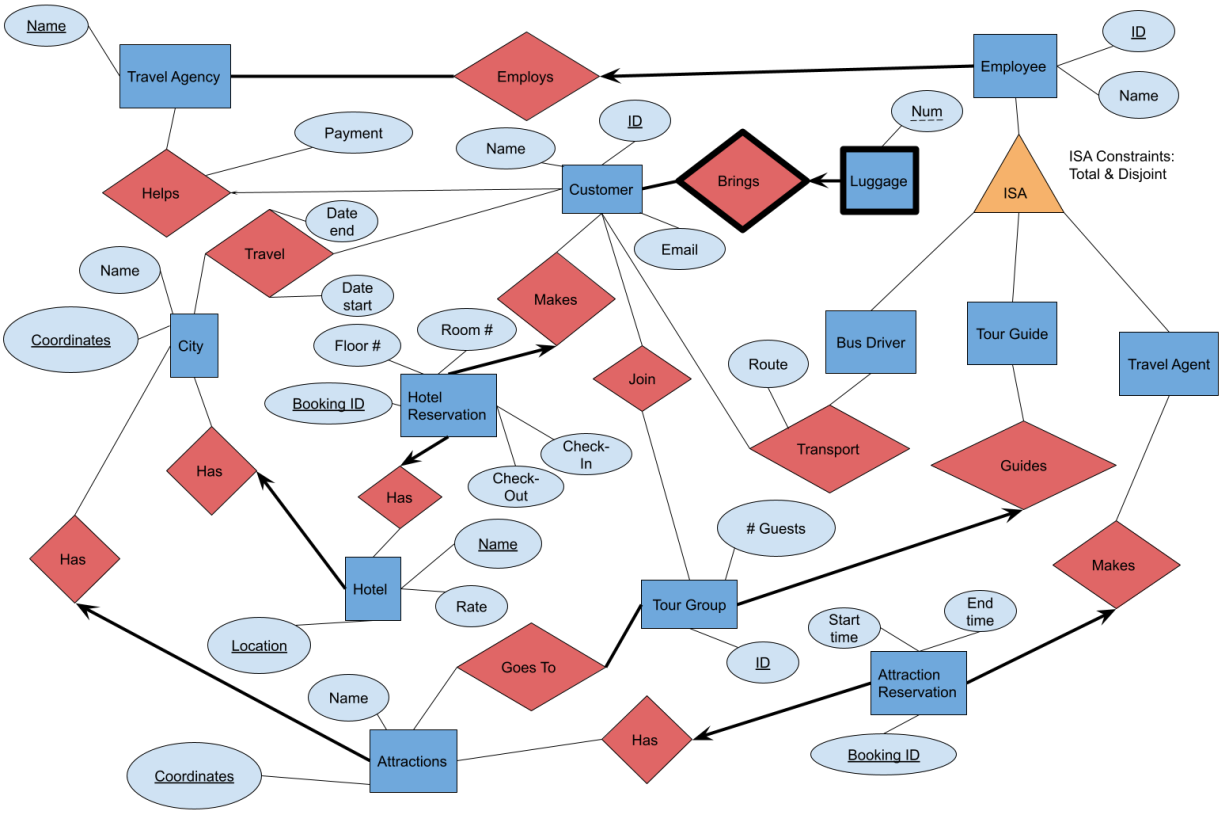
Date: __27/09/2021_____

Group Number: _____48_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Tristen Raab	73694689	u8o2b	tristenraab@gmail.com
Shubhleen Dhaka	94890399	y8v2b	kaurshubhleen18@gmail.com
Daman Sohi	24879397	g7z2b	daman.sohi@ymail.com

University of British Columbia, Vancouver
Department of Computer Science

ER Diagram



Changes:

- Added Floor # as an attribute to HotelReservation entity

Schema

Underlined is a primary key, bolded is a **foreign key.**

Employee(ID:INT, **Agency Name**:CHAR(256) (References TravelAgency(Name))(NOT NULL), EmployeeName:CHAR(256))

BusDriver(**ID**:INT (References Employee(ID)))

TourGuide(**ID**:INT (References Employee(ID)))

TravelAgent(**ID**:INT (References Employee(ID)))

Attractions(Coordinates: CHAR(256), AttractionName:CHAR(256), **CityCoords**: CHAR(256) (References City (NOT NULL))

AttractionReservation(BookingID:CHAR(256), **Coordinates**:CHAR(256) (References Attractions)(NOT NULL), **Agent ID**:CHAR(256) (References TravelAgent) (NOT NULL), Start Time:TIMESTAMP, End Time:TIMESTAMP)

Hotel(Name:CHAR(256), Location:CHAR(256), **CityCoords**:CHAR(256) (References City(Coordinates)) (NOT NULL), Rate:FLOAT)

HotelReservation(BookingID:CHAR(256), Floor:INT, RoomNum:INT, CheckIn:CHAR(256), CHAR(256):CHAR(256), **ID**:INT (References Customer)(NOT NULL), **HotelName**:CHAR(256) (References Hotel)(NOT NULL), **HotelLocation**:CHAR(256) (References Hotel)(NOT NULL))

TourGroupGuides(ID:CHAR(256), **GuideID**:INT (References TourGuide(ID))(NOT NULL), NumGuests:INT)

JoinTour(**CustomerID**:INT (References Customer(ID)), **TourID**:CHAR(256) (References TourGuide(ID)))

DriverTransports(**CID:**INT(References Customer(ID), **BID**:INT(References BusDriver(ID)), Route:CHAR(256))

TourGoesTo(**Coordinates**:CHAR(256) (References Attraction),**ID**:CHAR(256)(References TourGroupGuides(ID))

TravelAgency(Name:CHAR(256))

Customer(ID: INTEGER, Name: CHAR(256), Email:CHAR(256) UNIQUE)

BringsLuggage(**ID:** INTEGER (References Customer(ID)), Num: INTEGER)

HelpsCustomer(**AgencyName**: CHAR(256) (References TravelAgency), **ID**: INTEGER(References Customer(ID)), Payment:FLOAT)

City(Coordinates: CHAR(256), Name:CHAR(256))

TravelCity(**CityCoords**:CHAR(256) (References City(Coordinates)), **ID**: INTEGER (References Customer(ID)), DateEnd: DATE, DateStart: DATE)

University of British Columbia, Vancouver
Department of Computer Science

Functional Dependencies

Employee(ID, AgencyName, Name)

ID -> Name, AgencyName

BusDriver(ID, AgencyName, Name)

ID -> Name, AgencyName

TourGuide(ID, AgencyName, Name)

ID -> Name, AgencyName

TravelAgent(ID, AgencyName, Name)

ID -> Name, AgencyName

Attractions(Coordinates, Name, CityCoords)

Coordinates -> CityCoords, Name

AttractionReservation(BookingID, Coordinates, AgentID, Start Time, End Time)

AttractionBookingID -> AttractionCoordinates, AgentID, StartTime, EndTime

Hotel(Name, Location, CityCoords, Rate)

Name, Location -> CityCoords, Rate

HotelReservation(BookingID, FloorNum, RoomNum, CheckIn, CheckOut, ID, HotelName, HotelLocation)

BookingID -> FloorNum, RoomNum, CheckIn, CheckOut, ID, Name, Location

RoomNum->FloorNum

Decomposition of HotelReservation:

RoomNum+ violates BCNF so we decompose on RoomNum+ and get:

HR_1(BookingID, RoomNum, CheckIn, CheckOut, ID, Name, Location)
and HR_2(RoomNum, FloorNum)

HR_2 is a 2 attribute relation so it is in BCNF

HR_1 only has one valid FD: BookingID -> FloorNum, RoomNum, CheckIn, CheckOut, ID, Name, Location

And it is a key so this relationship is in BCNF and we get as our final answer:

HotelReservation(BookingID:CHAR(256), Room#:INT, CheckIn:TIMESTAMP, CHAR(256):TIMESTAMP, **ID**:INT, **Name**:CHAR(256), **Location**:CHAR(256))

HotelFloors(Room#:INT, Floor:INT)

TourGroupGuides(ID, GuideID, NumGuests)

ID -> GuideID, NumGuests

JoinTour(CustomerID, TourID)

CustomerID, TourID -> CustomerID, TourID

DriverTransports(CID, BID, Route)

CID, BID -> Route

TourGoesTo(Coordinates, ID)

Coordinates, ID -> Coordinates, ID

TravelAgency(Name)

Name -> Name

Customer(ID, Name, Email)

ID -> Name, Email
Name -> Email

Decomposition of Customer:

Name+ Violates BCNF so we decompose on Name+ and then get:

C_1(ID, Name) and C_2(Name, Email)

Since both of these are 2 attribute relations they are both in BCNF. So our final answer is:

Customer(ID: INTEGER, Name: CHAR(256))

CustomerEmails(Name: CHAR(256), Email:CHAR(256))

BringsLuggage(ID, Num)

ID, Num -> ID, Num

HelpsCustomer(Name, ID, Payment)

Name, ID -> Payment

City(Coordinates, Name)

Coordinates -> Coordinates, Name

TravelCity(Coordinates, ID, DateEnd, DateStart)

Coordinates, ID -> DateEnd, DateStart

University of British Columbia, Vancouver
Department of Computer Science

Normalization

All tables are normalized to BCNF

Customer(ID: INTEGER, Name: CHAR(256))

Candidate Key(s): {ID}

CustomerEmails(Name: CHAR(256), Email:CHAR(256))

Candidate Key(s): {Name}

HotelReservation(BookingID:CHAR(256), Room#:INT, CheckIn:TIMESTAMP, CHAR(256):TIMESTAMP, **ID**:INT, **Name**:CHAR(256), **Location**:CHAR(256))

Candidate Key(s): {BookingID}

HotelFloors(Room#:INT, Floor:INT)

Candidate Key(s): {Room#}

Employee(ID:CHAR(256), **Agency Name**:CHAR(256) (References TravelAgency(Name))(NOT NULL), Name:CHAR(256))

Candidate Key(s): {ID}

BusDriver(**ID**:INT (References Employee(ID)))

Candidate Key(s): {ID}

TourGuide(**ID**:INT (References Employee(ID)))

Candidate Key(s): {ID}

TravelAgent(**ID**:INT (References Employee(ID)))

Candidate Key(s): {ID}

Attractions(Coordinates: CHAR(256), Name:CHAR(256), **CityCoords**: CHAR(256) (References City(Coordinates)) (NOT NULL))

Candidate Key(s): {Coordinates}

AttractionReservation(BookingID:CHAR(256), **Coordinates**:CHAR(256) (References Attractions(Coordinates))(NOT NULL), **Agent ID**:CHAR(256) (References TravelAgent(ID))(NOT NULL), Start Time:DATETIME, End Time:DATETIME)

Candidate Key(s): {BookingID}

Hotel(Name:CHAR(256), Location:CHAR(256), **CityCoords**:CHAR(256) (References City(Coordinates)) (NOT NULL), Rate:FLOAT)

Candidate Key(s): {Name,Location}

TourGroup-Guides(ID:CHAR(256), **GuideID**:CHAR(256) (References TourGuide(ID))(NOT NULL), NumGuests)

Candidate Key(s): {ID}

JoinTour(**CustomerID**:CHAR(256) (References Customer(ID)), **TourID**:CHAR(256) (References TourGuide(ID)))

Candidate Key(s): {CustomerID, TourID}

DriverTransports(**CID:**INTEGER(References Customer(ID),**BID**:CHAR(256)(References BusDriver(ID)),Route:CHAR(256))

Candidate Key(s): {CID, BID}

TourGoesTo(**Coordinates**:CHAR(256) (References Attraction(Coordinates),**ID**:CHAR(256)(References TourGroup(ID))

Candidate Key(s): {Coordinates, ID}

TravelAgency(Name:CHAR(256))

Candidate Key(s): {Name}

BringsLuggage(**ID:** INTEGER (References Customer(ID)), Num: INTEGER)

Candidate Key(s): {ID, Num}

HelpsCustomer(**Name**: CHAR(256) (References TravelAgency(Name)), **ID**: INTEGER(References Customer(ID)), Payment:FLOAT)

Candidate Key(s): {Name, ID}

City(Coordinates: CHAR(256), Name:CHAR(256))

Candidate Key(s): {Coordinates}

TravelCity(**CityCoords**:CHAR(256) (References City(Coordinates)), **ID**: INTEGER (References Customer(ID)), DateEnd: DATE, DateStart: DATE)

Candidate Key(s): {CityCoords, ID}

SQL DDL

Please refer to our GitHub with the .sql code
https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_g7z2b_u8o2b_y8v2b/blob/master/project.sql

```
CREATE TABLE Employee (
    ID INT PRIMARY KEY,
   AgencyName CHAR(256) NOT NULL,
        EmployeeName CHAR(256)
);

CREATE TABLE BusDriver (
   ID INT PRIMARY KEY,
   FOREIGN KEY (ID) REFERENCES Employee
      ON DELETE CASCADE
);

CREATE TABLE TourGuide (
   ID INT PRIMARY KEY,
   FOREIGN KEY (ID) REFERENCES Employee
      ON DELETE CASCADE
);

CREATE TABLE TravelAgent (
   ID INT PRIMARY KEY,
   FOREIGN KEY (ID) REFERENCES Employee
      ON DELETE CASCADE
);

CREATE TABLE City (
        CityCoords CHAR(256) PRIMARY KEY,
```

```
        CityName CHAR(256)
);

CREATE TABLE Attractions (
   AttractionCoordinates CHAR(256) PRIMARY KEY,
   AttractionName CHAR(256),
   CityCoords CHAR(256) NOT NULL,
   FOREIGN KEY (CityCoords) REFERENCES City
      ON DELETE CASCADE
);

CREATE TABLE AttractionReservation (
   AttractionBookingID CHAR(256) PRIMARY KEY,
   AttractionCoordinates CHAR(256) NOT NULL,
   StartTime TIMESTAMP,
   EndTime TIMESTAMP,
   AgentID INT NOT NULL,
   FOREIGN KEY (AttractionCoordinates) REFERENCES Attractions
      ON DELETE CASCADE,
   FOREIGN KEY (AgentID) REFERENCES TravelAgent
      ON DELETE CASCADE
);

CREATE TABLE Customer (
        ID INT PRIMARY KEY,
        CustomerName CHAR(256)
);

CREATE TABLE Hotel (
        HotelName CHAR(256),
        HotelLocation CHAR(256),
        CityCoords CHAR(256) NOT NULL,
        Rate INT,
        PRIMARY KEY (HotelName, HotelLocation),
        FOREIGN KEY (CityCoords) REFERENCES City
           ON DELETE CASCADE
);

CREATE TABLE HotelReservation (
        HotelBookingID CHAR(256) PRIMARY KEY,
        RoomNum INT,
        CheckIn TIMESTAMP,
        CHAR(256) TIMESTAMP,
        ID INT NOT NULL,
```

```
  HotelName CHAR(256) NOT NULL,
  HotelLocation CHAR(256) NOT NULL,
  FOREIGN KEY (ID) REFERENCES Customer
    ON DELETE CASCADE,
  FOREIGN KEY (HotelName, HotelLocation) REFERENCES Hotel
    ON DELETE CASCADE
);

CREATE TABLE HotelFloors (
        RoomNum INT PRIMARY KEY,
        FloorNum INT
);

CREATE TABLE TourGroupGuides(
        ID CHAR(256) PRIMARY KEY,
        GuideID INT NOT NULL,
        NumGuests INT,
        FOREIGN KEY (GuideID) REFERENCES TourGuide(ID)
                ON DELETE CASCADE
);

CREATE TABLE JoinTour(
        CustomerID INT,
        TourID CHAR(256),
        PRIMARY KEY(CustomerID, TourID),
        FOREIGN KEY (CustomerID) REFERENCES Customer(ID)
                ON DELETE CASCADE,
        FOREIGN KEY (TourID) REFERENCES TourGroupGuides(ID)
                ON DELETE CASCADE
);

CREATE TABLE DriverTransports(
        CID INT,
        BID INT,
        DriverRoute CHAR(256),
        PRIMARY KEY(CID, BID),
        FOREIGN KEY (CID) REFERENCES Customer(ID)
             ON DELETE CASCADE,
        FOREIGN KEY (BID) REFERENCES BusDriver(ID)
             ON DELETE CASCADE
);

CREATE TABLE TourGoesTo(
        Coordinates CHAR(256),
```

```
        ID CHAR(256),
        PRIMARY KEY(Coordinates, ID),
        FOREIGN KEY (Coordinates) REFERENCES Attractions
                ON DELETE CASCADE,
        FOREIGN KEY (ID) REFERENCES TourGroupGuides(ID)
                ON DELETE CASCADE
);

CREATE TABLE TravelAgency (
        AgencyName CHAR(256) PRIMARY KEY
);

CREATE TABLE CustomerEmails (
        CustomerName CHAR(256) PRIMARY KEY,
        Email CHAR(256) UNIQUE
);

CREATE TABLE BringsLuggage(
        ID INTEGER,
        NUM INTEGER,
        PRIMARY KEY(ID, NUM),
        FOREIGN KEY (ID) REFERENCES Customer
                ON DELETE CASCADE
);

CREATE TABLE HelpsCustomer(
        AgencyName CHAR(256),
        ID INTEGER,
        PAYMENT FLOAT,
        PRIMARY KEY(AgencyName, ID),
        FOREIGN KEY (AgencyName) REFERENCES TravelAgency
          ON DELETE CASCADE,
        FOREIGN KEY (ID) REFERENCES Customer
          ON DELETE CASCADE
);


CREATE TABLE TravelCity(
        CityCoords CHAR(256),
        ID INTEGER,
        DateEnd DATE,
        DateStart DATE,
        PRIMARY KEY(CityCoords, ID),
        FOREIGN KEY (CityCoords) REFERENCES City
```

```
    ON DELETE CASCADE,
        FOREIGN KEY(ID) REFERENCES Customer
    ON DELETE CASCADE
);
```

Populated Tables

Please refer to our GitHub with the .sql code
https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_g7z2b_u8o2b_y8v2b/blob/master/project.sql

Queries

Insert:

Add a new Customer into the TravelAgency database.

Delete:

Delete a planned Tour from the itinerary.

Update:

Update an AttractionReservation with new start and end times.

Selection:

Select all Customers who have a Hotel booked in the next week to remind them.

Projection:

NOT COVERED

Join:

NOT COVERED

Aggregation With Group By:

NOT COVERED

Aggregation With Having:

NOT COVERED

Nested Aggregation With Group By:

NOT COVERED

Division:

NOT COVERED