CPSC 304 Project Cover Page

Milestone #: 2
Date: Sept 28 2024

Group Number: 61

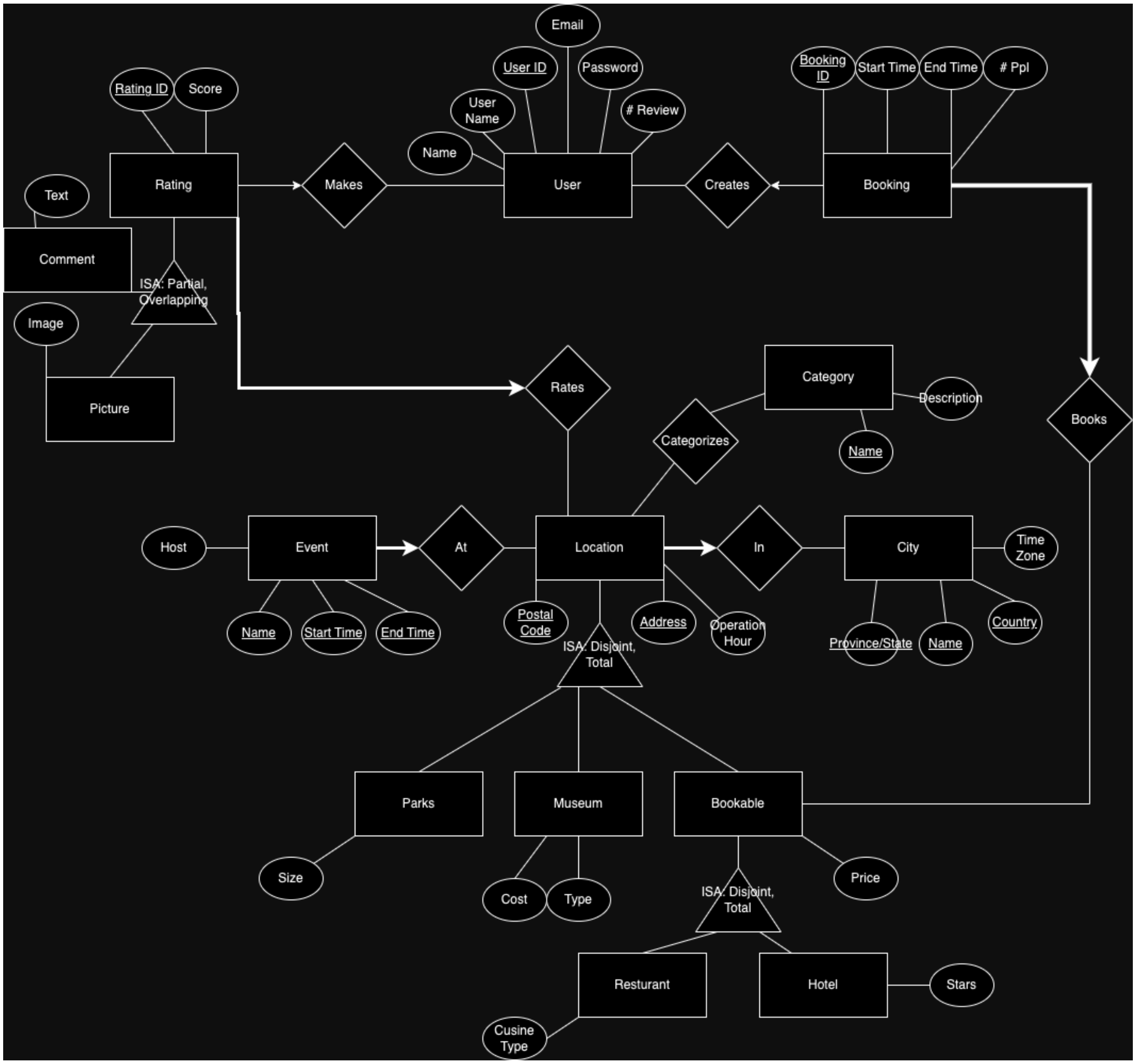
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Gregory Liu	94330628	k5o7g	gregoryliu123@gmail.com
Tony Gao	23782675	o7d0t	tonygao742@gmail.com
Richard Zhou	48759534	h5x5l	richardzhou1688@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. A brief (~2-3 sentences) summary of your project. Many of your TAs are managing multiple projects so this will help them remember details about your project.

Our project is a tourist attraction application, focusing on helping users discover, review, and book experiences in major cities, starting with Vancouver. The application will provide information on local tourist locations, acting as a central hub for travellers to plan and organize their trips.

3. The ER diagram you are basing your item #3 (below) on.



4. The schema derived from your ER diagram (above). For the translation of the ER diagram to the relational model, follow the same instructions as in your lectures. The process should be reasonably straightforward.

For each table:
a. List the table definition (e.g., Table1(attr1: domain1, attr2: domain2, ...)). Make sure to include the domains for each attribute.
b. Specify the primary key (PK), candidate key, (CK) foreign keys (FK), and other constraints (e.g., not null, unique, etc.) that the table must maintain.

```
User (
  UserID: char[10],
  Name: char[255] not null,
  Username: char[255] not null,
  Password: char[255] not null,
  # Reviews: int,
  Email: char[255] UNIQUE,
  Primary Key(UserID)
)

Rating (
  RatingID: char[10],
  Score: float,
  UserID: char[10],
  PostalCode: char[255] not null,
  Address: char[255] not null,
  Primary Key(RatingID),
  Foreign Key(UserID) references User(UserID),
  Foreign Key(PostalCode, Address) references Location(PostalCode, Address)
)

Comment (
  RatingID: char[10],
  Text: char[255],
  fk(RatingID) references Rating(RatingID)
)

Picture (
  RatingID: char[10],
  Image: image,
  fk(RatingID) references Rating(RatingID)
)
```

```
Booking (
  BookingID: char[10],
  StartTime: char[255],
  EndTime: char[255],
  # of People: int,
  UserID: char[10],
  PostalCode: char[255] not null,
  Address: char[255] not null,
  Primary Key(BookingID),
  Foreign Key(UserID) references User(UserID),
  Foreign Key(PostalCode, Address) references Bookable(PostalCode, Address)
)

Categorizes (
  Name: char[255],
  PostalCode: char[255],
  Address: char[255],
  Primary Key(Name, PostalCode, Address),
  Foreign Key(Name) references Category(Name),
  Foreign Key(PostalCode, Address) references Location(PostalCode, Address)
)

Category (
  Name: char[255],
  Description: char[255],
  Primary Key(Name)
)

Location (
  PostalCode: char[255],
  Address: char[255],
  OperationHours: char[255],
  ProvinceState: char[255] not null,
  Primary Key(PostalCode, Address),
  Foreign Key(ProvinceState) references City(ProvinceState)
)

Parks (
  PostalCode: char[255],
  Address: char[255],
  Size: int,
  Foreign Key(PostalCode, Address) references Location(PostalCode, Address)
)
```

```
Museums (
  PostalCode: char[255],
  Address: char[255],
  Cost: float,
  Type: char[255],
  Foreign Key(PostalCode, Address) references Location(PostalCode, Address)
)

Bookable (
  PostalCode: char[255],
  Address: char[255],
  Price: float,
  Foreign Key(PostalCode, Address) references Location(PostalCode, Address)
)

Restaurant (
  PostalCode: char[255],
  Address: char[255],
  CuisineType: char[255],
  Foreign Key(PostalCode, Address) references Bookable(PostalCode, Address)
)

Hotel (
  PostalCode: char[255],
  Address: char[255],
  Price: float,
  Stars: int,
  Foreign Key(PostalCode, Address) references Bookable(PostalCode, Address)
)

Event (
  Name: char[255],
  StartTime: char[255],
  EndTime: char[255],
  Host: char[255],
  PostalCode: char[255],
  Address: char[255],
  Primary Key(Name, StartTime, EndTime),
  Foreign Key(PostalCode, Address) references Location(PostalCode, Address)
)

City (
  ProvinceState: char[255],
  Name: char[255],
  Country: char[255],
  TimeZone: char[255],
  Primary Key(ProvinceState, Name, Country)
)
```

5. Functional Dependencies (FDs)
a. Identify the functional dependencies in your relations, including the ones involving all candidate keys (including the primary key).

PKs and CKs are considered functional dependencies and should be included in the list of FDs. You do not need to include trivial FDs such as A → A.

Note: In your list of FDs, there must be some kind of valid FD other those identified by a PK or CK. If you observe that no relations have FDs other than the PK and CK(s), then you will have to intentionally add some (meaningful) attributes to show valid FDs. We want you to get a good normalization exercise. Your design must go through a normalization process. You do not need to have a non-PK/CK FD for each relation but be reasonable. If your TA feels that some non-PK/CK FDs have been omitted, your grade will be adjusted accordingly.

1. User (UserID, Name, Username, Password, # Reviews, Email)
    a. UserID -> Name, Username, Password, #Reviews, Email
2. Rating(Rating ID, Score, **UserID**, **Postal Code**, **Address**)
    a. Rating ID -> Score, UserID, Postal Code, Address
3. Category(Name, Description)
    a. Name -> Description
4. Picture(**Rating ID**, Image)
    a. Rating ID -> Image
5. Comment(**Rating ID**, Text)
    a. Rating ID -> Text
6. Booking(Booking ID, Start Time, End Time, # of People, **User ID**, **Postal Code**, **Address**)
    a. Booking ID -> Start Time, End Time, # of People, User ID, Postal Code, Address
    b. For this relation, we can add the attribute “Duration” because some businesses like restaurants does not let the user choose end time. As a result, we get the FD:
    c. Start Time, # of Hours -> End Time
7. Categorizes(**Name**, **Postal Code**, **Address**)
8. Location(Postal Code, Address, Operation hours, **Province/State**)
    a. Postal Code, Address -> Operation Hour, Province/State
9. Parks(**Postal Code**, **Address**, Size)

    a. Postal Code, Address -> Size

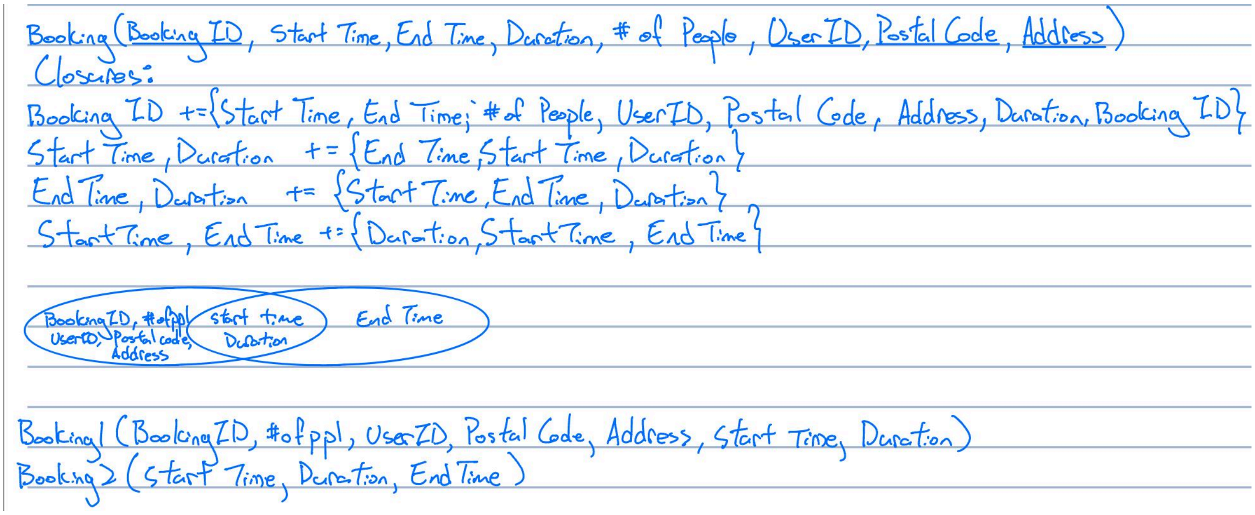
10. Museums(**Postal Code**, **Address**, Cost, Type)
    a. Postal Code, Address -> Cost, Type
11. Bookable(**Postal Code**, **Address**, Price)
    a. Postal Code, Address -> Price
12. Restaurant(**Postal Code**, **Address**, Cuisine Type)
    a. Postal Code, Address -> Cuisine Type
13. Hotel(**Postal Code**,**Address**, Stars)
    a. Postal Code, Address-> Stars
14. Event(Name, Start Time, End Time, Host, **Postal Code**, **Address**)
    a. Name, Start Time, End Time -> Host, Postal Code, Address
    b. Similarily, we can add duration so the event planner can specify the length of the event rather than end time.
    c. Start Time, # of hours -> End Time
    d. Event -> Host
15. City(Province/State, Name, Country, Time Zone)
    a. Province/State, Name, Country -> Time Zone
    b. Province/State, Country -> TimeZone

6. Normalization

a. Normalize each of your tables to be in 3NF or BCNF. Give the list of tables, their primary keys, their candidate keys, and their foreign keys after normalization. You should show the steps taken for the decomposition in a manner similar to that done in class. Should there be errors, and no work is shown, no partial credit can be awarded without steps shown. The format should be the same as Step 3, with tables listed similar to Table1(attr1:domain1, attr2:domain2, …). ALL Tables must be listed, not only the ones post normalization.

1. User (UserID, Name, Username, Password, # Reviews, Email)
    a. UserID -> Name, Username, Password, #Reviews, Email //This covers entire relation so it does not to be decomposed.
2. Rating(Rating ID, Score, **UserID**, **Postal Code**, **Address**)
    a. Rating ID -> Score, UserID, Postal Code, Address //Rating ID is key so this FD can not be decomposed anymore
3. Category(Name, Description)
    a. Name -> Description //Name is key so this FD can not be decomposed anymore
4. Picture(**Rating ID**, Image)
    a. Rating ID -> Image //Rating ID is key so this FD can not be decomposed anymore
5. Comment(**Rating ID**, Text)
    a. Rating ID -> Text //Rating ID is key so this FD can not be decomposed anymore

6. Booking(Booking ID, Start Time, End Time, Duration # of People, **User ID**, **Postal Code**, **Address**)
    a. Booking ID -> Start Time, End Time, # of People, User ID, Postal Code, Address, Duration //Booking ID is key so it can not be decomposed anymore
    b. Start Time, Duration -> End Time //Start Time, and Duration does not cover entire relation so relation needs to be decomposed
    c. End Time, Duration -> Start Time //Solved after first decomposition
    d. Start Time, End Time -> Duration //Solved after first decompostion



Booking(Booking ID, Start Time, End Time, Duration, # of People, User ID, Postal Code, Address)

Closures:

Booking ID +={Start Time, End Time; # of People, UserID, Postal Code, Address, Duration, Booking ID}

Start Time, Duration += {End Time, Start Time, Duration}

End Time, Duration += {Start Time, End Time, Duration}

Start Time, End Time +={Duration, Start Time, End Time}

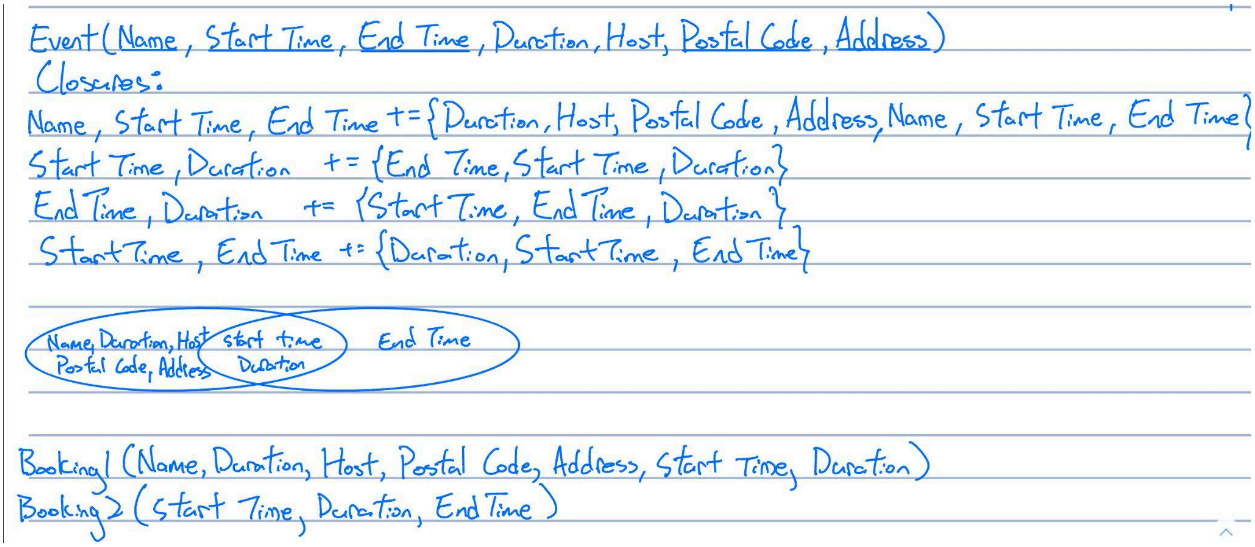
BookingID, #ofppl UserID, Postal code, Address
start time Duration
End Time

Booking1 (BookingID, #of ppl, UserID, Postal Code, Address, start Time, Duration)

Booking2 (start Time, Duration, End Time)

7. Categorizes(**Name**, **Postal Code**, **Address**)
    a. There is no FDs for this relation
8. Location(Postal Code, Address, Operation hours, **Province/State**)
    a. Postal Code, Address -> Operation Hour, Province/State //Postal Code and Address are keys, so it can not be decomposed
9. Parks(**Postal Code**, **Address**, Size)
    a. Postal Code, Address -> Size //Postal Code and Address are keys so it can't be decomposed
10. Museums(**Postal Code**, **Address**, Cost, Type)
    a. Postal Code, Address -> Cost, Type //Postal Code and Address are keys so it can not be decomposed
11. Bookable(**Postal Code**, **Address**, Price)
    a. Postal Code, Address -> Price //Postal Code and Address are keys so it can not be decomposed
12. Restaurant(**Postal Code**, **Address**, Cuisine Type)
    a. Postal Code, Address -> Cuisine Type //Postal Code and Address are keys so it can not be decomposed
13. Hotel(**Postal Code**,**Address**, Stars)
    a. Postal Code, Address-> Stars //Postal Code and Address are keys so it can not be decomposed
14. Event(Name, Start Time, End Time, Duration, Host, **Postal Code**, **Address**)

a. Name, Start Time, End Time -> Host, Postal Code, Address, Duration //Name, StartTime, and End Time are keys, so this FD does not violate FD.
b. Start Time, Duration -> End Time //Start Time, and Duration does not cover entire relation so relation needs to be decomposed
c. End Time, Duration -> Start Time //Solved after first decomposition
d. Start Time, End Time -> Duration //Solved after first decomposition



Event(Name, Start Time, End Time, Duration, Host, Postal Code, Address)

Closures:

Name, Start Time, End Time += {Duration, Host, Postal Code, Address, Name, Start Time, End Time}

Start Time, Duration += {End Time, Start Time, Duration}

End Time, Duration += {Start Time, End Time, Duration}

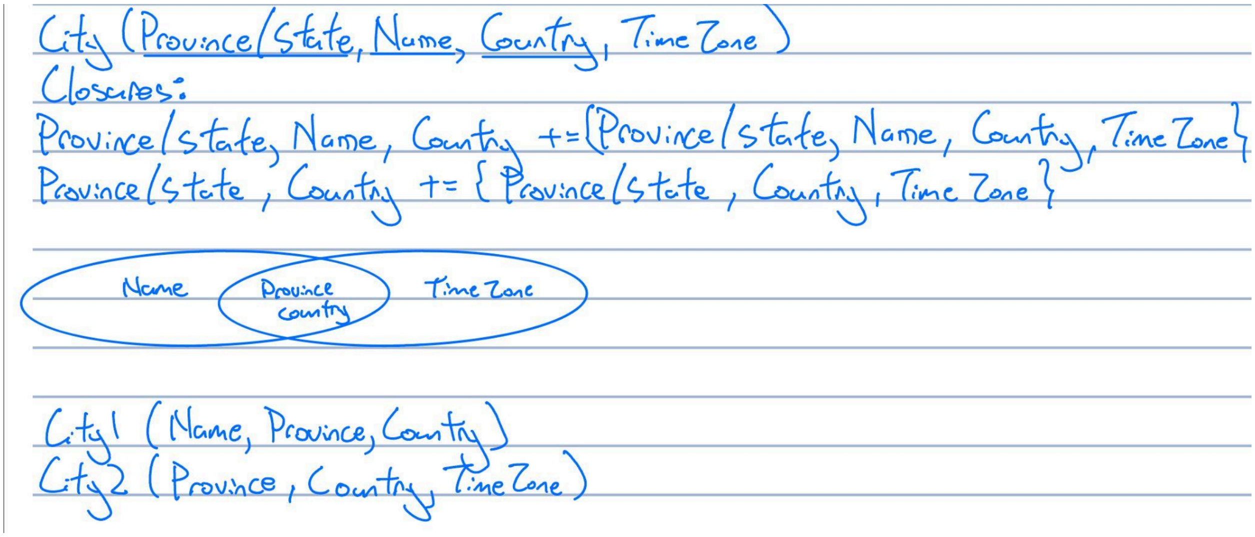
StartTime, End Time += {Duration, StartTime, End Time}

Name, Duration, Host
Postal Code, Address
start time
Duration
End Time

Booking1 (Name, Duration, Host, Postal Code, Address, Start Time, Duration)

Booking2 (Start Time, Duration, End Time)

15. City(Province/State, Name, Country, Time Zone)
a. Province/State, Name, Country -> Time Zone //Province/State, Name, and Country are keys, so this FD does not violate FD.
b. Province/State, Country -> TimeZone //Province/State and Country does not cover entire relation so relation needs to be decomposed



City (Province/State, Name, Country, Time Zone)

Closures:

Province/state, Name, Country += {Province/state, Name, Country, Time Zone}

Province/state, Country += {Province/state, Country, Time Zone}

Name
Province
country
Time Zone

City1 (Name, Province, Country)

City2 (Province, Country, Time Zone)

7. The SQL DDL statements required to create all the tables from item #6. The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc. Unless you know that you will always have exactly x characters for a given character, it is better to use the VARCHAR data type as opposed to a CHAR(Y). For example, UBC courses

always use four characters to represent which department offers a course. In that case, you will want to use CHAR(4) for the department attribute in your SQL DDL statement. If you are trying to represent the name of a UBC course, you will want to use VARCHAR as the number of characters in a course name can vary greatly.

```
CREATE TABLE User (
  UserID CHAR(10) PRIMARY KEY,
  Name VARCHAR NOT NULL,
  Username VARCHAR NOT NULL,
  Password VARCHAR NOT NULL,
  NumReviews INT,
  Email VARCHAR UNIQUE
);

CREATE TABLE Rating (
  RatingID CHAR(10) PRIMARY KEY,
  Score FLOAT,
  UserID CHAR(10),
  PostalCode VARCHAR NOT NULL,
  Address VARCHAR NOT NULL,
  FOREIGN KEY (UserID) REFERENCES User(UserID),
  FOREIGN KEY (PostalCode, Address) REFERENCES Location(PostalCode, Address)
);

CREATE TABLE Comment (
  RatingID CHAR(10),
  Text VARCHAR,
  FOREIGN KEY (RatingID) REFERENCES Rating(RatingID)
);

CREATE TABLE Picture (
  RatingID CHAR(10),
  Image IMAGE,
  FOREIGN KEY (RatingID) REFERENCES Rating(RatingID)
);

CREATE TABLE Time_of_Booking(
  StartTime VARCHAR,
  EndTime VARCHAR,
  Duration VARCHAR,
  PRIMARY KEY(Start Time, Duration)
 );

CREATE TABLE Booking_Details (
  BookingID CHAR(10) PRIMARY KEY,
  StartTime VARCHAR,
  Duration VARCHAR,
  NumPeople INT,
  UserID CHAR(10),
  PostalCode VARCHAR NOT NULL,
  Address VARCHAR NOT NULL,
  FOREIGN KEY (UserID) REFERENCES User(UserID),
  FOREIGN KEY (PostalCode, Address) REFERENCES Bookable(PostalCode, Address)
);

CREATE TABLE Categorizes (
  Name VARCHAR,
  PostalCode VARCHAR,
  Address VARCHAR,
  PRIMARY KEY (Name, PostalCode, Address),
  FOREIGN KEY (Name) REFERENCES Category(Name),
  FOREIGN KEY (PostalCode, Address) REFERENCES Location(PostalCode, Address)
);

CREATE TABLE Category (
  Name VARCHAR PRIMARY KEY,
  Description VARCHAR
);

CREATE TABLE Location (
  PostalCode VARCHAR,
  Address VARCHAR,
  OperationHours VARCHAR,
  ProvinceState VARCHAR NOT NULL,
  PRIMARY KEY (PostalCode, Address),
  FOREIGN KEY (ProvinceState) REFERENCES City(ProvinceState)
);

CREATE TABLE Parks (
  PostalCode VARCHAR,
  Address VARCHAR,
  Size INT,
  FOREIGN KEY (PostalCode, Address) REFERENCES Location(PostalCode, Address)
);

CREATE TABLE Museums (
  PostalCode VARCHAR,
  Address VARCHAR,
  Cost FLOAT,
  Type VARCHAR,
  FOREIGN KEY (PostalCode, Address) REFERENCES Location(PostalCode, Address)
);

CREATE TABLE Bookable (
  PostalCode VARCHAR,
  Address VARCHAR,
  Price FLOAT,
  FOREIGN KEY (PostalCode, Address) REFERENCES Location(PostalCode, Address)
);

CREATE TABLE Restaurant (
  PostalCode VARCHAR,
  Address VARCHAR,
  CuisineType VARCHAR,
  FOREIGN KEY (PostalCode, Address) REFERENCES Bookable(PostalCode, Address)
);

CREATE TABLE Hotel (
  PostalCode VARCHAR,
  Address VARCHAR,
  Price FLOAT,
  Stars INT,
  FOREIGN KEY (PostalCode, Address) REFERENCES Bookable(PostalCode, Address)
);

CREATE TABLE Event_Time (
  StartTime VARCHAR,
  EndTime VARCHAR,
  Duration VARCHAR,
  PRIMARY KEY (StartTime, Duration),
);

CREATE TABLE Event_Details (
  Name VARCHAR,
  StartTime VARCHAR,
  Duration VARCHAR,
  Host VARCHAR,
  PostalCode VARCHAR,
  Address VARCHAR,
  Duration VARCHAR,
  PRIMARY KEY (Name, StartTime, Duration)
  FOREIGN KEY (PostalCode, Address) REFERENCES Location(PostalCode, Address)
```

```
);
```

```
CREATE TABLE City_Details (
  ProvinceState VARCHAR NOT NULL,
  Name VARCHAR NOT NULL,,
  PRIMARY KEY (ProvinceState, Name)
);
```

```
CREATE TABLE Country_TimeZone (
  ProvinceState VARCHAR NOT NULL,
  Country VARCHAR NOT NULL,
  TimeZone VARCHAR,
  PRIMARY KEY (ProvinceState, Country)
);
```

8. INSERT statements to populate each table with at least 5 tuples. You will likely want to have more than 5 tuples so that you can have meaningful queries later. Note: Be consistent with the names used in your ER diagram, schema, and FDs. Make a note if the name has been intentionally changed.

```
-- Insert data into the User table
INSERT INTO User (UserID, Name, UserName, Password, NumberOfReviews, Email)
VALUES (1248484739, 'Greg', 'JumpingMan', 'aKask3*k', 4, 'GregDuck@gmail.com');

INSERT INTO User (UserID, Name, UserName, Password, NumberOfReviews, Email)
VALUES (4445424789, 'Tony', 'ImAnIdiot', 'asfKek!', 5, 'tao0811@gmail.com');

INSERT INTO User (UserID, Name, UserName, Password, NumberOfReviews, Email)
VALUES (3048594739, 'Lacardo', 'AlwaysHungry', 'EEiadkaj*7', 9, 'popo@gmail.com');

INSERT INTO User (UserID, Name, UserName, Password, NumberOfReviews, Email)
VALUES (1348590735, 'Jeff', 'Poep', 'aasdfiei2', 5, 'pldieh@gmail.com');

INSERT INTO User (UserID, Name, UserName, Password, NumberOfReviews, Email)
VALUES (1142484569, 'Tyler', 'peoet', 'aasffkek$$!', 5, 'tchillae@gmail.com');

-- Insert data into the Booking table
INSERT INTO Time_of_Booking (StartTime, EndTime, Duration)
VALUES ('2025-03-11 10:00', '2025-03-11 12:00', '2 hours');
```

```sql
INSERT INTO Time_of_Booking (StartTime, EndTime, Duration)
VALUES ('2025-04-15 11:00', '2025-04-15 13:00', '2 hours');

INSERT INTO Time_of_Booking (StartTime, EndTime, Duration)
VALUES ('2025-05-20 12:00', '2025-05-20 15:00', '3 hours');

INSERT INTO Time_of_Booking (StartTime, EndTime, Duration)
VALUES ('2025-06-10 09:00', '2025-06-10 17:00', '8 hours');

INSERT INTO Time_of_Booking (StartTime, EndTime, Duration)
VALUES ('2025-07-22 07:00', '2025-07-22 10:00', '3 hours');


-- Insert data into the Booking_Details table
INSERT INTO Booking_Details (BookingID, StartTime, Duration, NumPeople, UserID,
PostalCode, Address)
VALUES ('B443284754', '2025-03-11 10:00', '2 hours', 2, 'U248484739', 'V7S 9KS', '2019
e24th ave');

INSERT INTO Booking_Details (BookingID, StartTime, Duration, NumPeople, UserID,
PostalCode, Address)
VALUES ('B443284754', '2025-03-12 12:00', '3 hours', 3, 'U445424789', 'V7S 9KS', '2013
e25th ave');

INSERT INTO Booking_Details (BookingID, StartTime, Duration, NumPeople, UserID,
PostalCode, Address)
VALUES ('B443284754', '2025-03-13 09:00', '1 hour', 1, 'U048594739', 'V7S 9KS', '2018
e26th ave');

INSERT INTO Booking_Details (BookingID, StartTime, Duration, NumPeople, UserID,
PostalCode, Address)
VALUES ('B443284754', '2025-03-14 15:00', '2 hours', 4, 'U348590735', 'V7S 9KS', '2017
e26th ave');

INSERT INTO Booking_Details (BookingID, StartTime, Duration, NumPeople, UserID,
PostalCode, Address)
VALUES ('B443284754', '2025-03-15 11:00', '1.5 hours', 2, 'U142484569', 'V7S 9KS', '2016
e24th ave');
```

```
-- Insert data into the Rating table
INSERT INTO Rating (RatingID, Score, UserID, PostalCode, Address)
VALUES ('R939394423', 4.5, 1248484739, 'V7S 9KS', '2019 e24th ave');

INSERT INTO Rating (RatingID, Score, UserID, PostalCode, Address)
VALUES ('R002304040', 3.8, 4445424789, 'V7S 9KS', '2013 e25th ave');

INSERT INTO Rating (RatingID, Score, UserID, PostalCode, Address)
VALUES ('R044034332', 5.0, 3048594739, 'V7S 9KS', '2018 e26th ave');

INSERT INTO Rating (RatingID, Score, UserID, PostalCode, Address)
VALUES ('R443245004', 4.2, 1348590735, 'V7S 9KS', '2017 e26th ave');

INSERT INTO Rating (RatingID, Score, UserID, PostalCode, Address)
VALUES ('R043032554', 4.0, 1142484569, 'V7S 9KS', '2016 e24th ave');

-- Insert data into the Comment table
INSERT INTO Comment (RatingID, Text)
VALUES ('R939394423', 'Great experience, would recommend!');

INSERT INTO Comment (RatingID, Text)
VALUES ('R002304040', 'It was okay, but there were some issues.');

INSERT INTO Comment (RatingID, Text)
VALUES ('R044034332', 'Absolutely fantastic! Loved every moment.');

INSERT INTO Comment (RatingID, Text)
VALUES ('R443245004', 'Good service but the food was average.');

INSERT INTO Comment (RatingID, Text)
VALUES ('R043032554', 'Nice atmosphere, would visit again.');

-- Insert data into the Picture table
INSERT INTO Picture (RatingID, Image)
VALUES ('R939394423', 'image_data_1');

INSERT INTO Picture (RatingID, Image)
VALUES ('R002304040', 'image_data_2');

INSERT INTO Picture (RatingID, Image)
VALUES ('R044034332', 'image_data_3');
```

```sql
INSERT INTO Picture (RatingID, Image)
VALUES ('R443245004', 'image_data_4');

INSERT INTO Picture (RatingID, Image)
VALUES ('R043032554', 'image_data_5');

-- Insert data into the Location table
INSERT INTO Location (PostalCode, Address, OperationHours, ProvinceState)
VALUES ('V7S 9KS', '2019 e24th ave', '09:00 - 21:00', 'British Columbia');

INSERT INTO Location (PostalCode, Address, OperationHours, ProvinceState)
VALUES ('V7S 9KS', '2013 e25th ave', '10:00 - 22:00', 'British Columbia');

INSERT INTO Location (PostalCode, Address, OperationHours, ProvinceState)
VALUES ('V7S 9KS', '2018 e26th ave', '08:00 - 20:00', 'British Columbia');

INSERT INTO Location (PostalCode, Address, OperationHours, ProvinceState)
VALUES ('V7S 9KS', '2017 e26th ave', '09:00 - 21:00', 'British Columbia');

INSERT INTO Location (PostalCode, Address, OperationHours, ProvinceState)
VALUES ('V7S 9KS', '2016 e24th ave', '10:00 - 22:00', 'British Columbia');

-- Insert data into the Category table
INSERT INTO Category (Name, Description)
VALUES ('Food', 'All types of food-related establishments');

INSERT INTO Category (Name, Description)
VALUES ('Retail', 'Stores selling various goods');

INSERT INTO Category (Name, Description)
VALUES ('Services', 'Service-oriented businesses');

INSERT INTO Category (Name, Description)
VALUES ('Entertainment', 'Places for entertainment and leisure');

INSERT INTO Category (Name, Description)
VALUES ('Technology', 'Tech-related businesses and services');

-- Insert data into the Categorizes table
INSERT INTO Categorizes (Name, PostalCode, Address)
VALUES ('Food', 'V7S 9KS', '2019 e24th ave');

INSERT INTO Categorizes (Name, PostalCode, Address)
VALUES ('Retail', 'V7S 9KS', '2013 e25th ave');
```

```
INSERT INTO Categorizes (Name, PostalCode, Address)
VALUES ('Services', 'V7S 9KS', '2018 e26th ave');

INSERT INTO Categorizes (Name, PostalCode, Address)
VALUES ('Entertainment', 'V7S 9KS', '2017 e26th ave');

INSERT INTO Categorizes (Name, PostalCode, Address)
VALUES ('Technology', 'V7S 9KS', '2016 e24th ave');

-- Insert data into the Parks table
INSERT INTO Parks (PostalCode, Address, Size)
VALUES ('V7S 9KS', '2019 e24th ave', 500);

INSERT INTO Parks (PostalCode, Address, Size)
VALUES ('V7S 9KS', '2013 e25th ave', 750);

INSERT INTO Parks (PostalCode, Address, Size)
VALUES ('V7S 9KS', '2018 e26th ave', 600);

INSERT INTO Parks (PostalCode, Address, Size)
VALUES ('V7S 9KS', '2017 e26th ave', 850);

INSERT INTO Parks (PostalCode, Address, Size)
VALUES ('V7S 9KS', '2016 e24th ave', 400);

-- Insert data into the Museums table
INSERT INTO Museums (PostalCode, Address, Cost, Type)
VALUES ('V7S 9KS', '2019 e24th ave', 15.50, 'Art');

INSERT INTO Museums (PostalCode, Address, Cost, Type)
VALUES ('V7S 9KS', '2013 e25th ave', 20.00, 'History');

INSERT INTO Museums (PostalCode, Address, Cost, Type)
VALUES ('V7S 9KS', '2018 e26th ave', 12.00, 'Science');

INSERT INTO Museums (PostalCode, Address, Cost, Type)
VALUES ('V7S 9KS', '2017 e26th ave', 25.75, 'Natural History');

INSERT INTO Museums (PostalCode, Address, Cost, Type)
VALUES ('V7S 9KS', '2016 e24th ave', 18.00, 'Technology');


-- Insert data into the Restaurant table
```

```sql
INSERT INTO Restaurant (PostalCode, Address, CuisineType)
VALUES ('V7S 9KS', '2019 e24th ave', 'Italian');

INSERT INTO Restaurant (PostalCode, Address, CuisineType)
VALUES ('V7S 9KS', '2013 e25th ave', 'Chinese');

INSERT INTO Restaurant (PostalCode, Address, CuisineType)
VALUES ('V7S 9KS', '2018 e26th ave', 'Mexican');

INSERT INTO Restaurant (PostalCode, Address, CuisineType)
VALUES ('V7S 9KS', '2017 e26th ave', 'Japanese');

INSERT INTO Restaurant (PostalCode, Address, CuisineType)
VALUES ('V7S 9KS', '2016 e24th ave', 'Indian');



-- Insert data into the bookable table
INSERT INTO Bookable (PostalCode, Address, Price)
VALUES ('V7S 9KS', '2019 e24th ave', 150.00);

INSERT INTO Bookable (PostalCode, Address, Price)
VALUES ('V7S 9KS', '2013 e25th ave', 200.00);

INSERT INTO Bookable (PostalCode, Address, Price)
VALUES ('V7S 9KS', '2018 e26th ave', 180.00);

INSERT INTO Bookable (PostalCode, Address, Price)
VALUES ('V7S 9KS', '2017 e26th ave', 220.00);

INSERT INTO Bookable (PostalCode, Address, Price)
VALUES ('V7S 9KS', '2016 e24th ave', 160.00);

-- Insert data into the Hotel table

INSERT INTO Hotel (PostalCode, Address, Price, Stars)
VALUES ('V7S 9KS', '2019 e24th ave', 250.00, 4);

INSERT INTO Hotel (PostalCode, Address, Price, Stars)
VALUES ('V7S 9KS', '2013 e25th ave', 300.00, 5);

INSERT INTO Hotel (PostalCode, Address, Price, Stars)
VALUES ('V7S 9KS', '2018 e26th ave', 180.00, 3);
```

```
INSERT INTO Hotel (PostalCode, Address, Price, Stars)
VALUES ('V7S 9KS', '2017 e26th ave', 220.00, 4);

INSERT INTO Hotel (PostalCode, Address, Price, Stars)
VALUES ('V7S 9KS', '2016 e24th ave', 160.00, 3);



-- Insert data into the Event table
INSERT INTO Event (Name, StartTime, EndTime, Host, PostalCode, Address)
VALUES ('Music Festival', '2025-03-11 10:00', '2025-03-11 18:00', 'John Doe', 'V7S 9KS',
'2019 e24th ave');

INSERT INTO Event (Name, StartTime, EndTime, Host, PostalCode, Address)
VALUES ('Art Exhibition', '2025-04-15 11:00', '2025-04-15 20:00', 'Jane Smith', 'V7S 9KS',
'2013 e25th ave');

INSERT INTO Event (Name, StartTime, EndTime, Host, PostalCode, Address)
VALUES ('Food Fair', '2025-05-20 12:00', '2025-05-20 19:00', 'Mike Johnson', 'V7S 9KS',
'2018 e26th ave');

INSERT INTO Event (Name, StartTime, EndTime, Host, PostalCode, Address)
VALUES ('Tech Conference', '2025-06-10 09:00', '2025-06-10 17:00', 'Sarah Lee', 'V7S 9KS',
'2017 e26th ave');

INSERT INTO Event (Name, StartTime, EndTime, Host, PostalCode, Address)
VALUES ('Charity Run', '2025-07-22 07:00', '2025-07-22 12:00', 'Tom Brown', 'V7S 9KS',
'2016 e24th ave');


-- Insert data into the City table

INSERT INTO City (ProvinceState, Name, Country, TimeZone)
VALUES ('British Columbia', 'Vancouver', 'Canada', 'PST');

INSERT INTO City (ProvinceState, Name, Country, TimeZone)
VALUES ('British Columbia', 'Victoria', 'Canada', 'PST');

INSERT INTO City (ProvinceState, Name, Country, TimeZone)
VALUES ('Ontario', 'Toronto', 'Canada', 'EST');

INSERT INTO City (ProvinceState, Name, Country, TimeZone)
VALUES ('Quebec', 'Montreal', 'Canada', 'EST');
```

```
INSERT INTO City (ProvinceState, Name, Country, TimeZone)
VALUES ('California', 'Los Angeles', 'USA', 'PST');
```