Destination Database

Project Report

INFO 210-900, Winter 2020

18 March 2020

Tracy Li (tl842), Matthew Lieb (mjl387),

Yegeon Seo (ys554), Quan Ta (qmt26)

1. Executive Summary

For our project, we chose the Oracle Implementation category, as the members of our group were relatively new to relational databases. After discussing our interests and considering how best to gain a strong understanding of working with databases, we chose to create one that would be utilized by travel agencies to house and search for data about various travel destinations. It would also be utilized by clients of agencies to search as well. We realized there were many factors that went into determining where to go for a vacation, including culture, history, food, and climate. There were also other less inherent factors, such as general price, how well the destination caters to tourists with things like tours, and how well the destination has been received by others. All of these would be included in the database.

Once we had finalized what we wanted to do, we set about researching destinations to include. As some members of our group had travelled to multiple continents, coming up with locations from around the world to include was rather straightforward. At the same time, we also had to look into how to set up the database, and how best to represent the data we would find. Some members of our group had multiple years of experience with object-oriented programming languages and their capabilities, so transitioning to SQL and understanding its limitations required some time and practice, as well as some redesigns of our schema when certain ideas didn’t pan out.

In order to ensure diversity in possible queries without forcing everyone to overwork themselves, we compiled data for 20 unique cities, many of which also include reviews and/or tour packages.

1. Problem Statement

2.1 Goals

The overall goal of the system is to provide clients with detailed information about different vacation sites. Examples of available information include, but are not limited to: city, culture, climate, average expenses, and available tour packages.

2.2 Context

It is critical that travel agencies are able to provide clients an enjoyable vacation. By extension, being able to determine key factors of a client’s vacation in advance is integral to ensuring the trip will go as planned. It is also important to keep track of these pieces of information to give agencies a better understanding of which vacation sites are the highest recommended for different types of trips.

2.3 Scope

2.3.1 In-Scope:

System will include city, culture (which consists of language, cuisine, and historical sites), climate, pricing, client reviews, and available tours.

2.3.2 Out-Scope:

System will not include payment activities such as plane tickets, renting/housing, at-location transportation, and other miscellaneous charges.

1. Requirements

3.1 Analysis of Current System

There are numerous resources on travel agency data models on the Internet. Our plan is to create a simplified version of those models.

3.2 Data requirements

* For cities, the database will keep track of a *CityID*, the *Country* where the city resides, the *Culture*, *Climate*, and an *AvgPrice*. *Culture* and *Climate* will connect with other data tables to join additional data into query results.
* For cultures, the database will keep track of a *CultureID*, as well as a culture’s *PrimLanguage*, *Cuisine*, and a foriegn key to a separate table *HistoricalSites* containing up to three historical sites. It will produce different data depending on the *CultureID*.
* For climates, the database will keep track of a *ClimateID,* a foriegn key to a table *BestMonths* containing up to three of the best months to visit the location, and the average temperatures for each season. It will produce different data depending on the *ClimateID*.
* For clients, the database will keep track of a *ClientID*, and a single character variable *IsGroup* for if the client is traveling in a group or not.
* For reviews, the database will rely on an existing *CityID* and *ClientID* as a composite key, and keep track of a rating.
* For tours, the database will keep track of a *TourID*, starting and ending dates, *Cost*, and a foriegn key to a separate table *Activities* containing up to three activities included in the tour.
* For city tour packages, the database will rely on an existing *CityID* and *TourID* as a composite key.

3.3 Business rules and Logic

* When a new city is created, its name will serve as a unique ID.
* When a new culture is created, it will start with an ID of 1 and increase by order. It will contain distinct information. The ID for the culture’s historical sites will be identical.
  + This also applies to a new climate and its best months. They will share a distinct numberedID.
  + This also applies to the activities linked to tours.
* Each client will have a name that serves as a unique ID. When a review is created, it will have a composite key made up of *CityID*, which connects it to one city, and *ClientID*, which connects it to one client. A city can have multiple reviews, and one client can write multiple reviews.
* Each tour will have a name that serves as a unique ID. This ID will not bear any connection to the number-based ID for its activities. When a tour package for a specific city is set up, it will have a composite key made up of *CityID*, which connects it to one city, and *TourID*, which connects it to one tour. A city can have multiple specific tour packages, and one tour can apply to multiple cities.

3.4 Any other assumption

* Database will be used by travel agencies and clients of the agencies.
* Database will adhere to hardware and software specifications in 3.7.

3.5 The way we solved the problem

We collected all of our research data, and compiled it into a single database. This database can be used by tourists and travel agencies to plan trips based on budget, interest in culture, reception from other database clients, and availability of tour packages.

3.6 The way we obtained the data

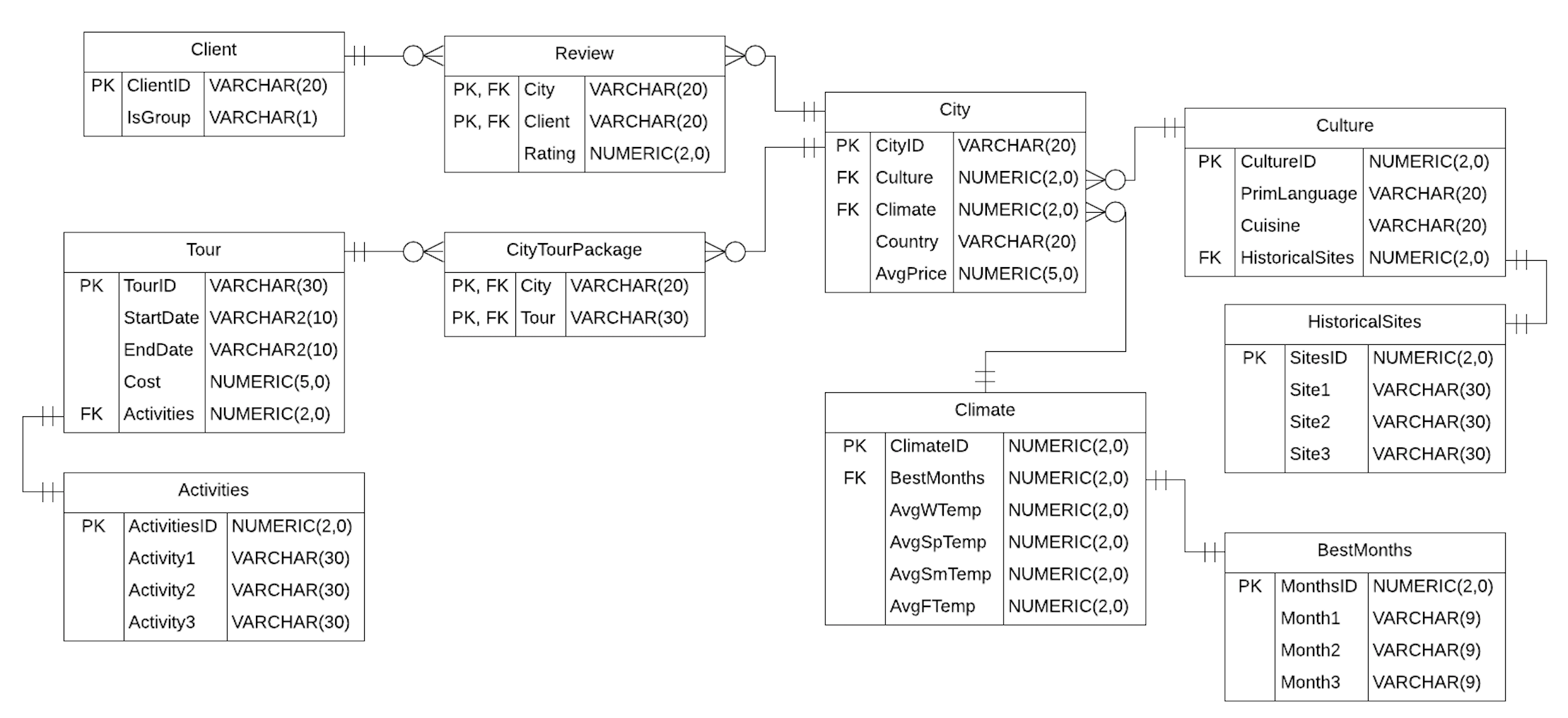
* Quantifiable or easy to find data such as pricing, best months to visit, and climate temperatures, were taken from websites that contain such data for public knowledge.
* Cultural aspects such as historical sites and language were taken from tourism websites and articles.
* Client reviews and available tours were either referenced from existing instances on various tourism websites, or created by group members.

3.7 Hardware, software, and special features

* Database will be created using Oracle SQL Plus.
* Database will be used on both Windows and Mac OS.

1. Conceptual Design

4.1 ERD



4.2 ERD Documentation

* City.AvgPrice: The average price of a week’s vacation in the city.
* Culture.PrimLanguage: The primary language spoken in the city.
* Culture.Cuisine: The cuisine style of the city.
* HistoricalSites table: A list of historical sites present in the city.
* BestMonths table: A list of the best months to visit the city based on its climate.
* Climate.Avg\_Temp: The average temperature of a city’s season (W = Winter, Sp = Spring, Sm = Summer, F = Fall).
* Client.IsGroup: An indicator of whether or not the Client travels in a group when visiting cities. Will always be either ‘Y’ or ‘N’.
* Activities table: A list of activities that are part of the tour.

1. Relational Schema

**City** (CityID, Culture, Climate, Country, AvgPrice)

FOREIGN KEY Culture REFERENCES Culture CultureID

FOREIGN KEY Climate REFERENCES Climate ClimateID

**Culture** (CultureID, PrimLanguage, Cuisine, HistoricalSites)

FOREIGN KEY HistoricalSites REFERENCES HistoricalSites SitesID

**HistoricalSites** (SitesID, Site1, Site2, Site3)

**Climate** (ClimateID, BestMonths, AvgWTemp, AvgSpTemp, AvgSmTemp, AvgFTemp)

FOREIGN KEY BestMonths REFERENCES BestMonths MonthsID

**BestMonths** (MonthsID, Month1, Month2, Month3)

**Client** (ClientID, IsGroup)

**Review** (City, Client, Rating)

FOREIGN KEY City REFERENCES City CityID

FOREIGN KEY Client REFERENCES Client ClientID

**Tour** (TourID, StartDate, EndDate, Cost, Activities)

FOREIGN KEY Activities REFERENCES Activities ActivitiesID

**Activities** (ActivitiesID, Activity1, Activity2, Activity3)

**CityTourPackage** (City, Tour)

FOREIGN KEY City REFERENCES City CityID

FOREIGN KEY Tour REFERENCES Tour TourID

1. Data Dictionary

* City
  + CityID
    - The city’s name.
    - VARCHAR(20)
    - PK
  + Culture
    - A foriegn key to a table containing data about the city’s culture.
    - Only 99 unique cultures can be stored in the database.
    - NUMERIC(2,0)
    - FK
  + Climate
    - A foriegn key to a table containing data about the city’s climate.
    - Only 99 unique climates can be stored in the database.
    - NUMERIC(2,0)
    - FK
  + Country
    - The country the city resides in.
    - VARCHAR(20)
  + AvgPrice
    - The average price of a week’s vacation in the city, in USD.
    - Is assumed to be under $100,000.
    - NUMERIC(5,0)
* Culture
  + CultureID
    - A primary key for connecting a city to its culture.
    - NUMERIC(2,0)
    - PK
  + PrimLanguage
    - The primary language spoken in the city.
    - VARCHAR(20)
  + Cuisine
    - The cuisine style of the city.
    - VARCHAR(20)
  + HistoricalSites
    - A foriegn key to a table serving as a list of historical sites present in the city. It is the same value as CultureID, and follows the same limitations.
    - NUMERIC(2,0)
    - FK
* HistoricalSites
  + SitesID
    - A primary key for connecting a city’s culture to its historical sites.
    - NUMERIC(2,0)
    - PK
  + Site1, Site2, & Site3
    - The three attributes listed are identical in purpose.
    - Historical sites in the city.
    - VARCHAR(30)
* Climate
  + ClimateID
    - A primary key for connecting a city to its climate.
    - NUMERIC(2,0)
    - PK
  + BestMonths
    - A foriegn key to a table serving as a list of the best months to visit the city based on its climate. It is the same value as ClimateID, and follows the same limitations.
    - NUMERIC(2,0)
    - FK
  + AvgWTemp, AvgSpTemp, AvgSmTemp, & AvgFTemp
    - The four attributes listed are identical in purpose.
    - W = Winter, Sp = Spring, Sm = Summer, F = Fall.
    - The average temperature of the city’s seasons in fahrenheit.
    - Is assumed to be under 100.
    - NUMERIC(2,0)
* BestMonths
  + MonthsID
    - A primary key for connecting a city’s climate to its best months.
    - NUMERIC(2,0)
    - PK
  + Month1, Month2, & Month3
    - The three attributes listed are identical in purpose.
    - The best months to visit the city based on its climate.
    - VARCHAR(9)
* Client
  + ClientID
    - The client’s name.
    - VARCHAR(20)
    - PK
  + IsGroup
    - An indicator of whether or not the client travels in a group when visiting cities.
    - Will always be either ‘Y’ or ‘N’.
    - VARCHAR(1)
* Review
  + City
    - The city being reviewed, represented by a City table’s primary key.
    - Serves as part of a composite primary key.
    - VARCHAR(20)
    - PK, FK
  + Client
    - The client making the review, represented by a Client table’s primary key.
    - Serves as part of a composite primary key.
    - VARCHAR(20)
    - PK, FK
  + Rating
    - The client’s rating of the city.
    - Assumed to be 1-10.
    - NUMERIC(2,0)
* Tour
  + TourID
    - The tour’s name.
    - VARCHAR(30)
    - PK
  + StartDate & EndDate
    - The two attributes listed are identical in type.
    - The dates on which the tour starts and ends.
    - Are in YYYY-MM-DD format.
    - VARCHAR2(10)
  + Cost
    - The cost of the tour, in USD.
    - Is assumed to be under $100,000.
    - NUMERIC(5,0)
  + Activities
    - A foriegn key to a table serving as a list of activities that are part of the tour.
    - Only 99 unique lists of activities can be stored in the database.
    - NUMERIC(2,0)
    - FK
* Activities
  + ActivitiesID
    - A primary key for connecting a tour to its activities.
    - NUMERIC(2,0)
    - PK
  + Activity1, Activity2, & Activity3
    - The three attributes listed are identical in type.
    - Different activities that are included in the tour.
    - VARCHAR(30)
* CityTourPackage
  + City
    - The city to which the tour pertains, represented by a City table’s primary key.
    - Serves as part of a composite primary key.
    - VARCHAR(20)
    - PK, FK
  + Tour
    - The tour being used in the city, represented by a Tour table’s primary key.
    - Serves as part of a composite primary key.
    - VARCHAR(30)
    - PK, FK

1. Data
2. INSERT INTO Commands

INSERT INTO HISTORICALSITES VALUES(1, 'Statue of Libery', 'Empire State Building', '9/11 Memorial');

INSERT INTO HISTORICALSITES VALUES(2, 'Colosseum', 'Pantheon', 'Forum');

INSERT INTO HISTORICALSITES VALUES(3, 'Imperial Palace', 'Senso-ji Temple', 'Meiji Shrine');

INSERT INTO HISTORICALSITES VALUES(4, 'Griffith Observatory', 'Hollywood Walk of Fame', 'Venice Beach Boardwalk');

INSERT INTO HISTORICALSITES VALUES(5, 'Eiffel Tower', 'Arc de Triomphe', 'Louvre');

INSERT INTO HISTORICALSITES VALUES(6, 'Buckingham Palace', 'Tower of London', 'Big Ben');

INSERT INTO HISTORICALSITES VALUES(7, 'Brandenburg Gate', 'Reichstag Building', 'Holocaust Memorial');

INSERT INTO HISTORICALSITES VALUES(8, 'Millennium Park', 'Navy Pier', 'Art Institute of Chicago');

INSERT INTO HISTORICALSITES VALUES(9, 'Vieux-Montreal', 'Notre-Dame Basilica', 'Mont-Royal');

INSERT INTO HISTORICALSITES VALUES(10, 'Federation Square', 'Shrine of Remembrance', 'Queen Victoria Market');

INSERT INTO HISTORICALSITES VALUES(11, 'Odysea Aquarium', 'Sonoran Desert', 'Museum of Contemporary Art');

INSERT INTO HISTORICALSITES VALUES(12, 'Chamberlin Observatory', 'Denver Botanic Gardens', 'Colorado Symphony');

INSERT INTO HISTORICALSITES VALUES(13, 'Museo De Oro', 'Ruinas de Ujarras', 'Iglesia de la Merced');

INSERT INTO HISTORICALSITES VALUES(14, 'Grand Palace', 'Giant Swing', 'Terminal 21');

INSERT INTO HISTORICALSITES VALUES(15, 'The Bund', 'Shanghai Museum', 'Zhujiajiao');

INSERT INTO HISTORICALSITES VALUES(16, 'Seoul City Wall', 'Gyeongbokgung Palace', 'Changdeokgung Palace');

INSERT INTO HISTORICALSITES VALUES(17, 'Ibirapuera Park', 'Contemporary Art Museum', 'Se Cathedral');

INSERT INTO HISTORICALSITES VALUES(18, 'Buccaneer Cove', 'Post office Bay', 'The Wall of Tears');

INSERT INTO HISTORICALSITES VALUES(19, 'Arch of Santa Catalina', 'La Merced Church', 'Plaza Central Park');

INSERT INTO HISTORICALSITES VALUES(20, 'Panama Canal', 'Bahai Temple', 'Independence Square');

INSERT INTO CULTURE VALUES(1, 'English', 'American', 1);

INSERT INTO CULTURE VALUES(2, 'Italian', 'Italian', 2);

INSERT INTO CULTURE VALUES(3, 'Japanese', 'Japanese', 3);

INSERT INTO CULTURE VALUES(4, 'English', 'American', 4);

INSERT INTO CULTURE VALUES(5, 'French', 'French', 5);

INSERT INTO CULTURE VALUES(6, 'English', 'British', 6);

INSERT INTO CULTURE VALUES(7, 'German', 'German', 7);

INSERT INTO CULTURE VALUES(8, 'English', 'American', 8);

INSERT INTO CULTURE VALUES(9, 'French/English', 'Quebec', 9);

INSERT INTO CULTURE VALUES(10, 'English', 'Australian', 10);

INSERT INTO CULTURE VALUES(11, 'English', 'American', 11);

INSERT INTO CULTURE VALUES(12, 'English', 'American', 12);

INSERT INTO CULTURE VALUES(13, 'Spanish', 'Costa Rican', 13);

INSERT INTO CULTURE VALUES(14, 'Thai', 'Thai', 14);

INSERT INTO CULTURE VALUES(15, 'Chinese', 'Chinese', 15);

INSERT INTO CULTURE VALUES(16, 'Korean', 'Korean', 16);

INSERT INTO CULTURE VALUES(17, 'Portuguese', 'Brazilian', 17);

INSERT INTO CULTURE VALUES(18, 'Spanish', 'Ecuadorian', 18);

INSERT INTO CULTURE VALUES(19, 'Spanish', 'Guatemalan', 19);

INSERT INTO CULTURE VALUES(20, 'Spanish', 'Panamanian', 20);

INSERT INTO BESTMONTHS VALUES(1, 'May', 'June', 'September');

INSERT INTO BESTMONTHS VALUES(2, 'April', 'May', 'June');

INSERT INTO BESTMONTHS VALUES(3, 'March', 'April', 'October');

INSERT INTO BESTMONTHS VALUES(4, 'March', 'April', 'May');

INSERT INTO BESTMONTHS VALUES(5, 'April', 'May', 'June');

INSERT INTO BESTMONTHS VALUES(6, 'May', 'June', 'July');

INSERT INTO BESTMONTHS VALUES(7, 'February', 'June', 'September');

INSERT INTO BESTMONTHS VALUES(8, 'May', 'June', 'July');

INSERT INTO BESTMONTHS VALUES(9, 'June', 'July', 'August');

INSERT INTO BESTMONTHS VALUES(10, 'January', 'February', 'March');

INSERT INTO BESTMONTHS VALUES(11, 'September', 'October', 'November');

INSERT INTO BESTMONTHS VALUES(12, 'April', 'May', 'September');

INSERT INTO BESTMONTHS VALUES(13, 'February', 'April', 'December');

INSERT INTO BESTMONTHS VALUES(14, 'March', 'November', 'December');

INSERT INTO BESTMONTHS VALUES(15, 'October', 'November', 'December');

INSERT INTO BESTMONTHS VALUES(16, 'March', 'May', 'September');

INSERT INTO BESTMONTHS VALUES(17, 'March', 'May', 'October');

INSERT INTO BESTMONTHS VALUES(18, 'June', 'July', 'November');

INSERT INTO BESTMONTHS VALUES(19, 'April', 'October', 'November');

INSERT INTO BESTMONTHS VALUES(20, 'May', 'July', 'October');

INSERT INTO CLIMATE VALUES(1, 1, 34, 52, 74, 58);

INSERT INTO CLIMATE VALUES(2, 2, 46, 56, 73, 61);

INSERT INTO CLIMATE VALUES(3, 3, 43, 56, 75, 64);

INSERT INTO CLIMATE VALUES(4, 4, 58, 63, 72, 68);

INSERT INTO CLIMATE VALUES(5, 5, 40, 53, 66, 54);

INSERT INTO CLIMATE VALUES(6, 6, 41, 47, 57, 57);

INSERT INTO CLIMATE VALUES(7, 7, 34, 48, 65, 49);

INSERT INTO CLIMATE VALUES(8, 8, 27, 47, 71, 53);

INSERT INTO CLIMATE VALUES(9, 9, 17, 30, 68, 35);

INSERT INTO CLIMATE VALUES(10, 10, 52, 60, 70, 61);

INSERT INTO CLIMATE VALUES(11, 11, 66, 83, 99, 87);

INSERT INTO CLIMATE VALUES(12, 12, 40, 60, 85, 75);

INSERT INTO CLIMATE VALUES(13, 13, 74, 77, 76, 75);

INSERT INTO CLIMATE VALUES(14, 14, 88, 94, 90, 89);

INSERT INTO CLIMATE VALUES(15, 15, 50, 65, 88, 72);

INSERT INTO CLIMATE VALUES(16, 16, 38, 50, 70, 45);

INSERT INTO CLIMATE VALUES(17, 17, 70, 71, 65, 69);

INSERT INTO CLIMATE VALUES(18, 18, 80, 82, 73, 70);

INSERT INTO CLIMATE VALUES(19, 19, 65, 72, 60, 63);

INSERT INTO CLIMATE VALUES(20, 20, 60, 71, 85, 69);

INSERT INTO CITY VALUES('New York City', 1, 1, 'USA', 1676);

INSERT INTO CITY VALUES('Rome', 2, 2, 'Italy', 1121);

INSERT INTO CITY VALUES('Tokyo', 3, 3, 'Japan', 1610);

INSERT INTO CITY VALUES('Los Angeles', 4, 4, 'USA', 1328);

INSERT INTO CITY VALUES('Paris', 5, 5, 'France', 1250);

INSERT INTO CITY VALUES('London', 6, 6, 'UK', 1250);

INSERT INTO CITY VALUES('Berlin', 7, 7, 'Germany', 800);

INSERT INTO CITY VALUES('Chicago', 8, 8, 'USA', 1260);

INSERT INTO CITY VALUES('Montreal', 9, 9, 'Canada', 1190);

INSERT INTO CITY VALUES('Melbourne', 10, 10, 'Australia', 960);

INSERT INTO CITY VALUES('Scottsdale', 11, 11, 'USA', 1607);

INSERT INTO CITY VALUES('Denver', 12, 12, 'USA', 2358);

INSERT INTO CITY VALUES('San Jose', 13, 13, 'Costa Rica', 850);

INSERT INTO CITY VALUES('Bangkok', 14, 14, 'Thailand', 920);

INSERT INTO CITY VALUES('Shanghai', 15, 15, 'China', 2400);

INSERT INTO CITY VALUES('Seoul', 16, 16, 'South Korea', 1942);

INSERT INTO CITY VALUES('Sao Paulo', 17, 17, 'Brazil', 617);

INSERT INTO CITY VALUES('Galapagos Islands', 18, 18, 'Ecuador', 1548);

INSERT INTO CITY VALUES('Antigua', 19, 19, 'Guatemala', 1645);

INSERT INTO CITY VALUES('Panama City', 20, 20, 'Panama', 958);

INSERT INTO CLIENT VALUES('Mark Smith', 'N');

INSERT INTO CLIENT VALUES('James Lee', 'Y');

INSERT INTO CLIENT VALUES('Tom S', 'N');

INSERT INTO CLIENT VALUES('Kim W', 'Y');

INSERT INTO CLIENT VALUES('Justin Park', 'N');

INSERT INTO CLIENT VALUES('Liam Do', 'Y');

INSERT INTO CLIENT VALUES('Julio Hertenez', 'N');

INSERT INTO CLIENT VALUES('Lisa Majors', 'Y');

INSERT INTO CLIENT VALUES('Leo Zhang', 'N');

INSERT INTO CLIENT VALUES('Randy Diep', 'Y');

INSERT INTO CLIENT VALUES('Bernard Convento', 'N');

INSERT INTO REVIEW VALUES('New York City', 'Mark Smith', 9);

INSERT INTO REVIEW VALUES('Los Angeles', 'James Lee', 8);

INSERT INTO REVIEW VALUES('London', 'Tom S', 9);

INSERT INTO REVIEW VALUES('Berlin', 'Kim W', 9);

INSERT INTO REVIEW VALUES('Scottsdale', 'Justin Park', 8);

INSERT INTO REVIEW VALUES('Denver', 'Liam Do', 7);

INSERT INTO REVIEW VALUES('San Jose', 'Julio Hertenez', 7);

INSERT INTO REVIEW VALUES('Bangkok', 'Lisa Majors', 9);

INSERT INTO REVIEW VALUES('Shanghai', 'Leo Zhang', 8);

INSERT INTO REVIEW VALUES('Galapagos Islands', 'Randy Diep', 9);

INSERT INTO REVIEW VALUES('Panama City', 'Bernard Convento', 7);

INSERT INTO ACTIVITIES VALUES(1, 'Times Sqaure', 'Broadway', 'Harlem');

INSERT INTO ACTIVITIES VALUES(2, 'River cruise', '100 stops', 'none');

INSERT INTO ACTIVITIES VALUES(3, 'Yuyuan Garden', 'Disneyland Park', 'Nanjing Road');

INSERT INTO ACTIVITIES VALUES(4, 'Insadong', 'Line Friends', 'Jinhae Cherry Blossom');

INSERT INTO TOUR VALUES('Ultimate NYC Tour', '2020-03-01', '2020-03-07', 100, 1);

INSERT INTO TOUR VALUES('London Hop-On Hop-Off Tour', '2020-03-11', '2020-03-18', 125, 2);

INSERT INTO TOUR VALUES('Shanghai Tour', '2020-04-12', '2020-04-18', 155, 3);

INSERT INTO TOUR VALUES('Seoul Tour', '2020-05-01', '2020-05-07', 138, 4);

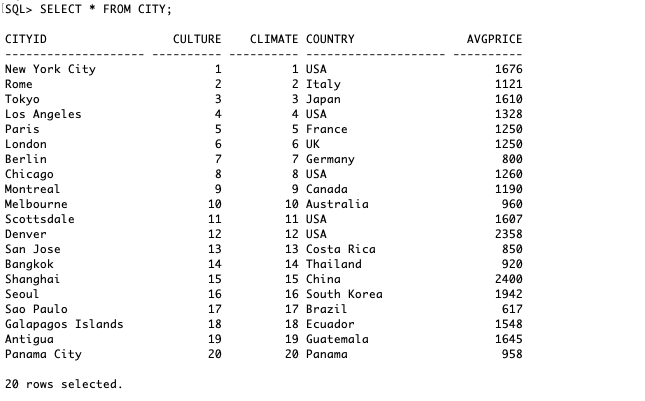
INSERT INTO CITYTOURPACKAGE VALUES('New York City', 'Ultimate NYC Tour');

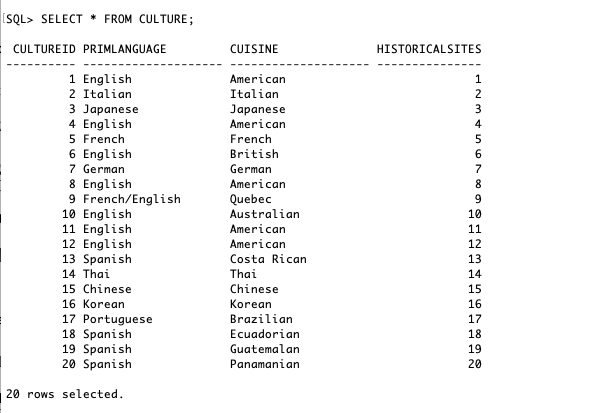
INSERT INTO CITYTOURPACKAGE VALUES('London', 'London Hop-On Hop-Off Tour');

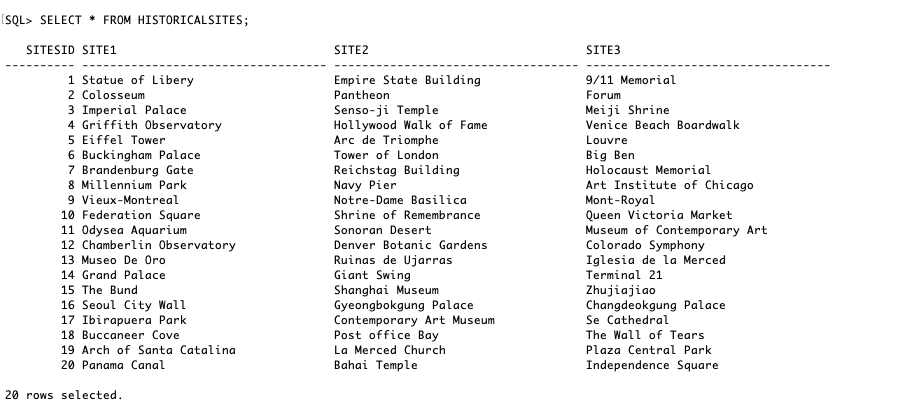
INSERT INTO CITYTOURPACKAGE VALUES('Shanghai', 'Shanghai Tour');

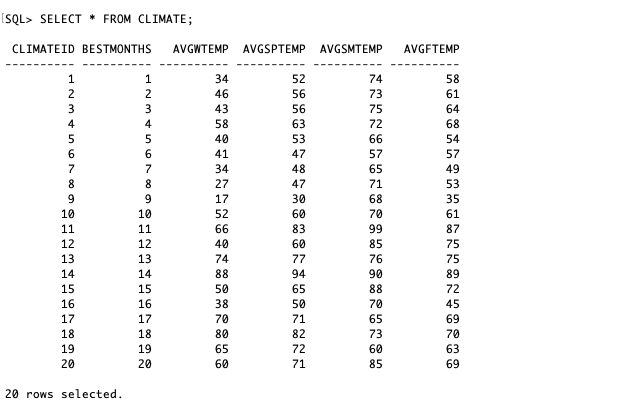
INSERT INTO CITYTOURPACKAGE VALUES('Seoul', 'Seoul Tour');

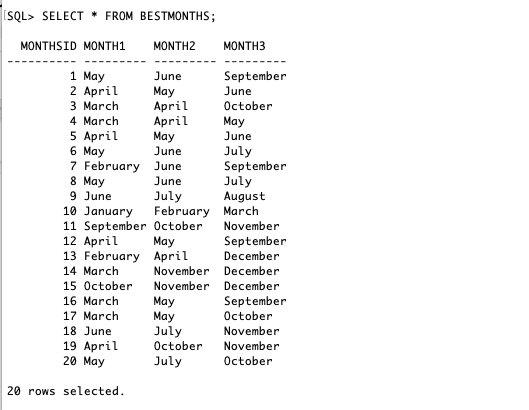
1. Data from SELECT \* FROM *all tables*

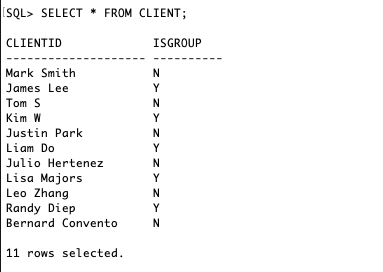


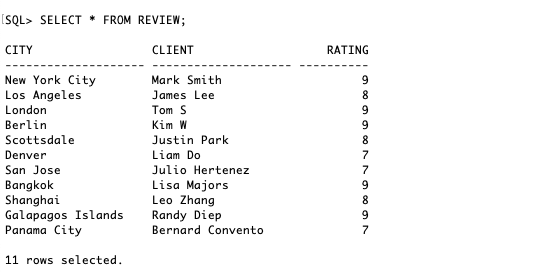


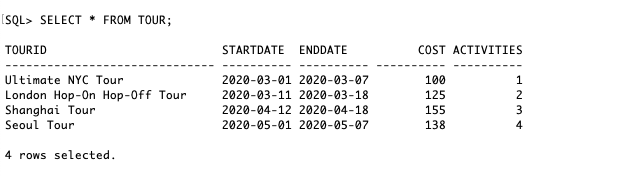


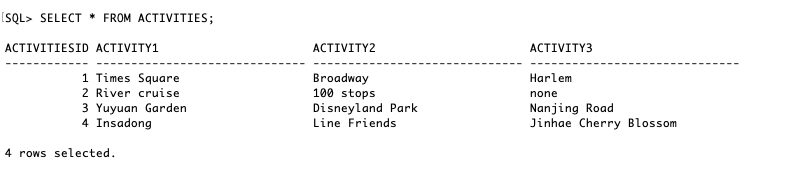


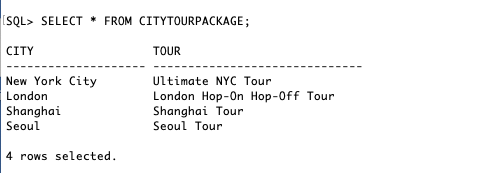






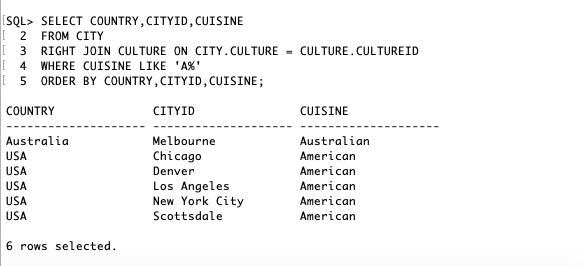




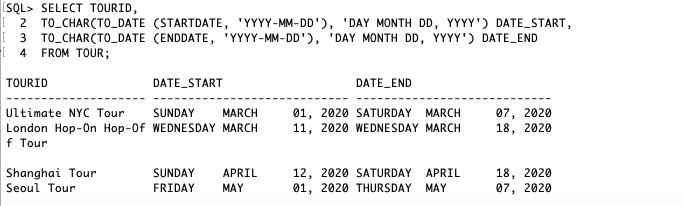


1. Data Queries

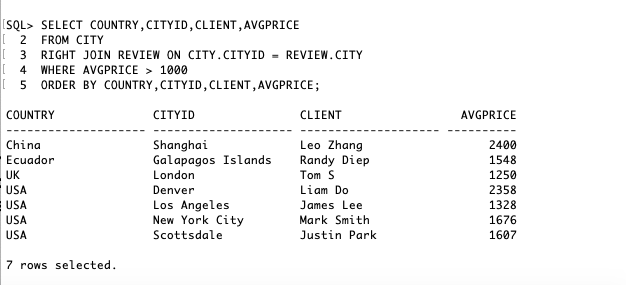
* Queries by Tracy:
  + Selecting all cuisines that start with the letter ‘A’ from the city table. Ordered by country, city, and cuisine name.



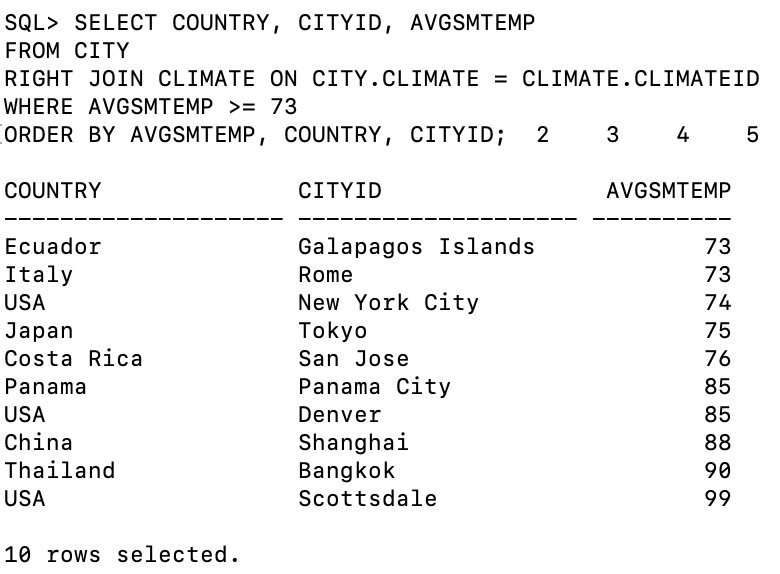
* + Selecting all tour names and listing the start date and end date of the tours reformatted using TO\_CHAR().



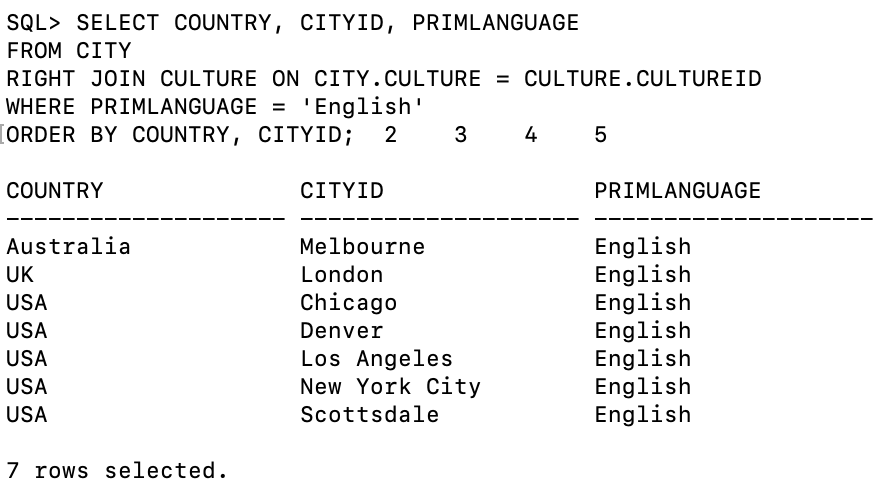
* + Selecting all clients whose reviewed city’s average price is more than $1000. Ordered by country, city, client, and average price.



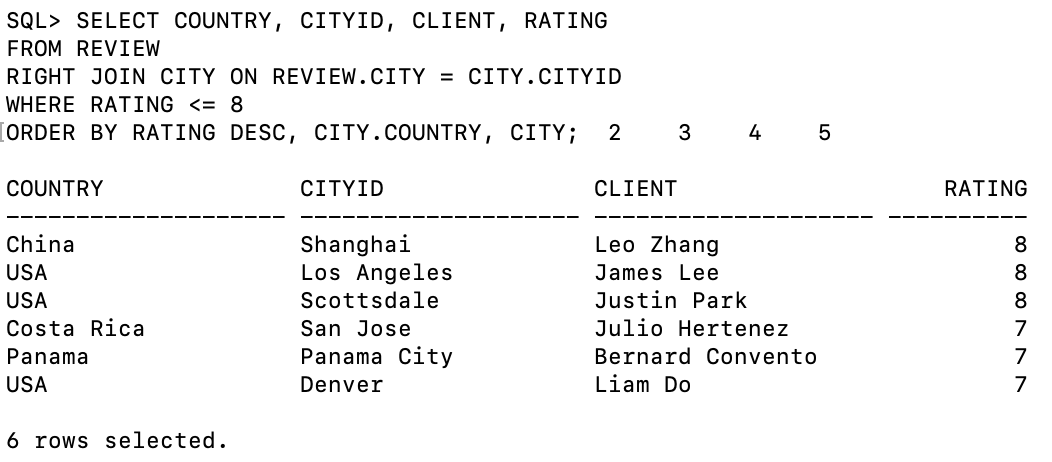
* Queries by Matthew:
  + Selecting all cities with average summer temperatures of at least 73 degrees. The temperature itself is included in the printout to verify accuracy, as is the city’s country. Ordered by temperature, then country, then city.



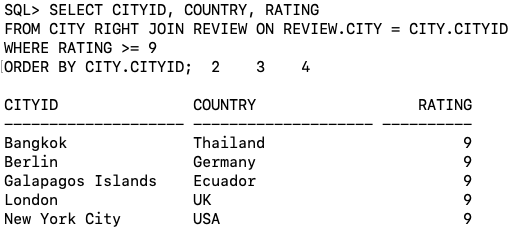
* + Selecting all cities where English is the sole primary language. The language itself is included in the printout to verify accuracy, as is the city’s country. Ordered by country, then city.



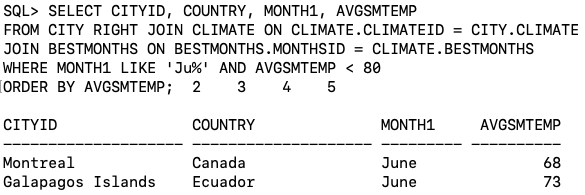
* + Selecting all cities that had reviews with ratings of 8 or less. The rating itself is included in the printout to verify accuracy, as is the review’s client, and the city’s country. Ordered by rating in descending order, then country, then city.



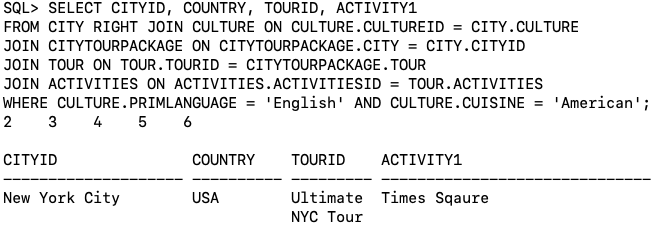
* Queries by Yegeon:
  + Select all cities with rating above or equal to 9. The rows are ordered by the city name.



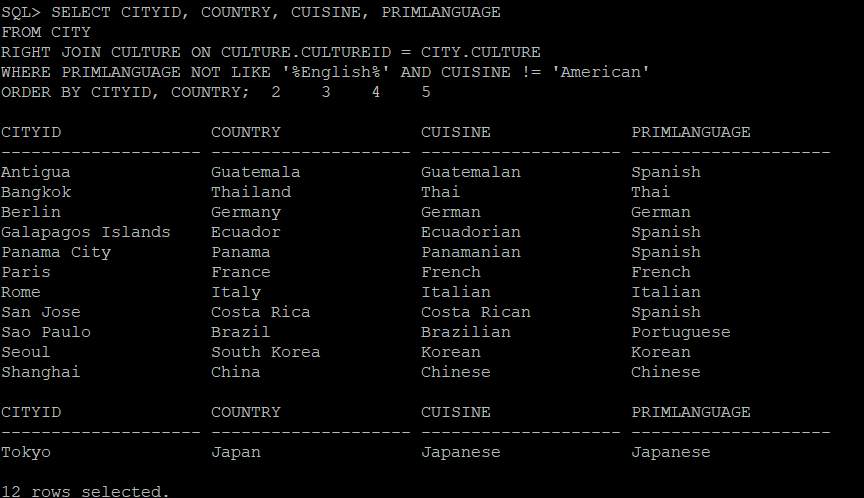
* + Select all cities where the best months are either June or July, and the average summer temperature is less than 80. The table is ordered by average temperature.

****

* + Select all tour packages that have cities with english as primary language, and american as cuisine. It does not have an order.



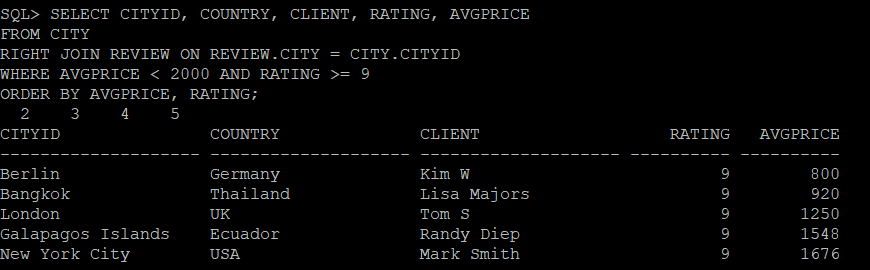
* Queries by Quan:
  + Select all cities that do not have English as primary language and American as cuisine. The table is ordered by city, then country.

****

* + Select all cities that have March as one of their best months. The table is ordered by city, then country.

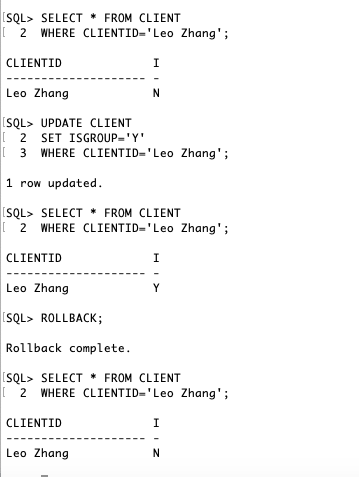


* + Select all cities that have ratings of at least 9, and have an average price of less than $2000. The table is ordered by average price, then rating.

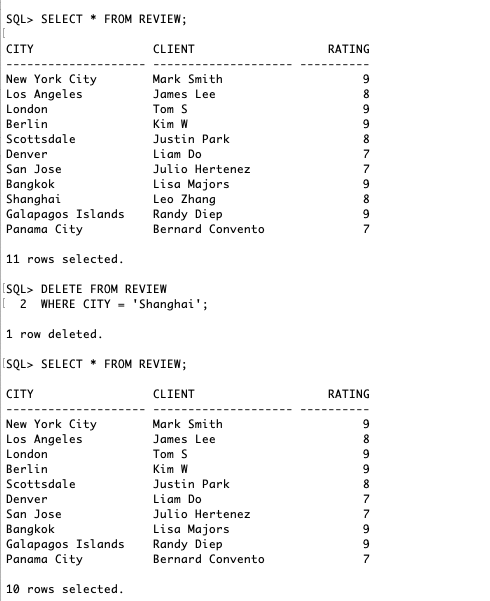


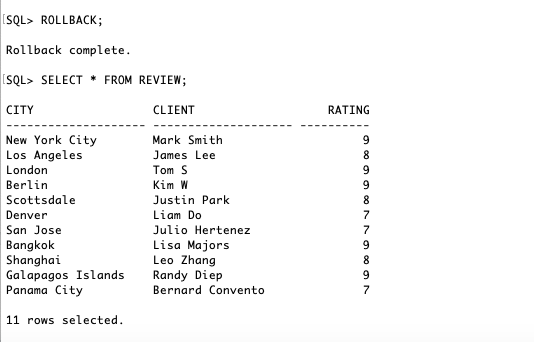
1. Data Manipulation

* Tracy:
  + Update: Changing the IsGroup of the Client ‘Leo Zhang’ to ‘Y’.

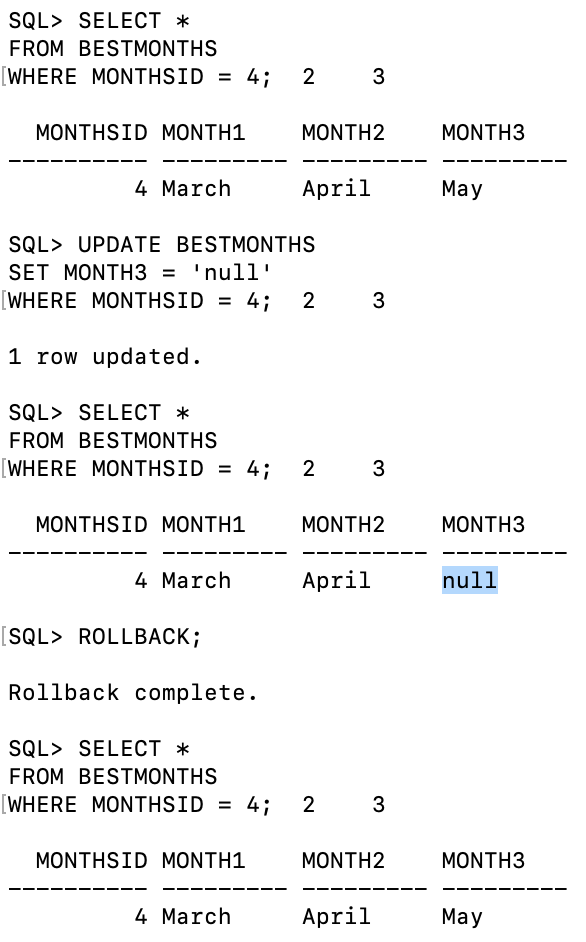


* + Delete: Deleting all reviews of Shanghai from Review.

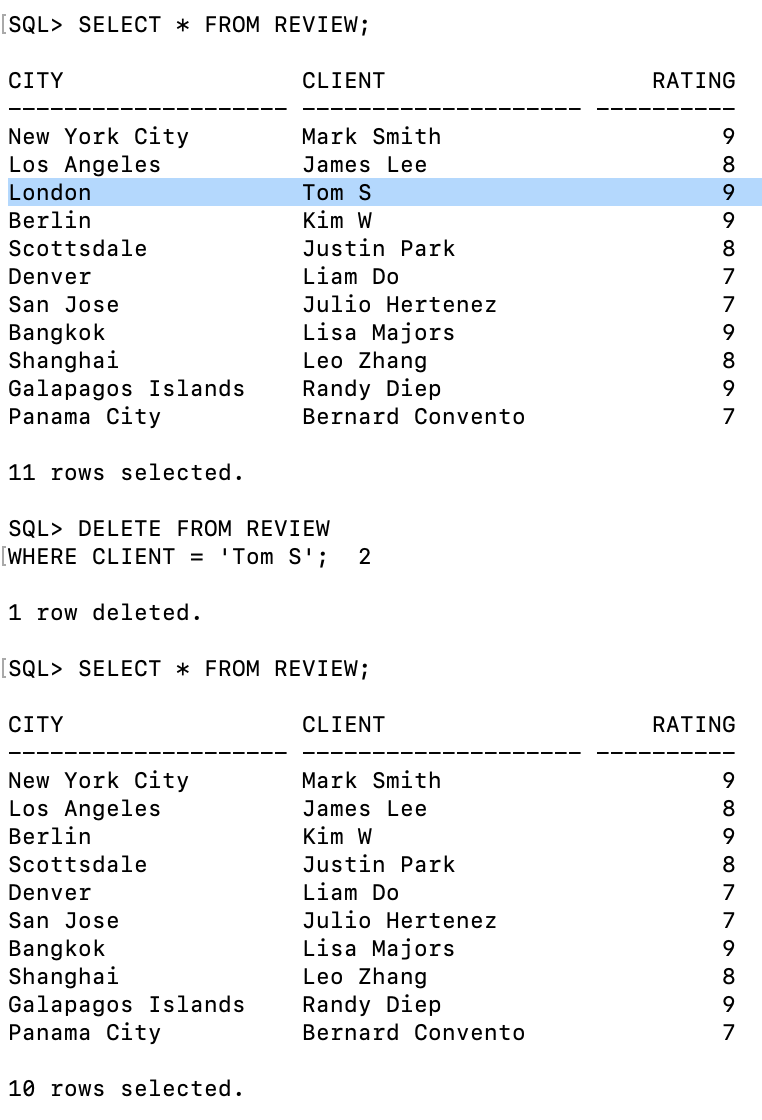


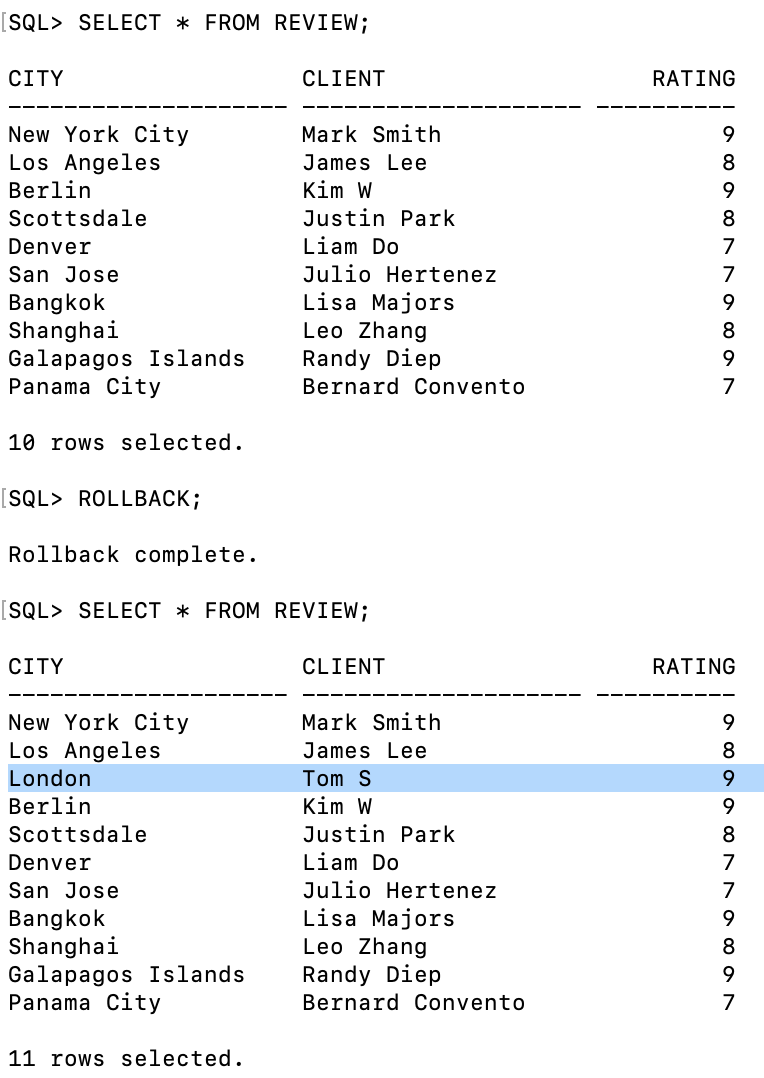


* Matthew:
  + Update: Changing the third month in the BestMonth table with MonthsID of 4 to ‘null’.

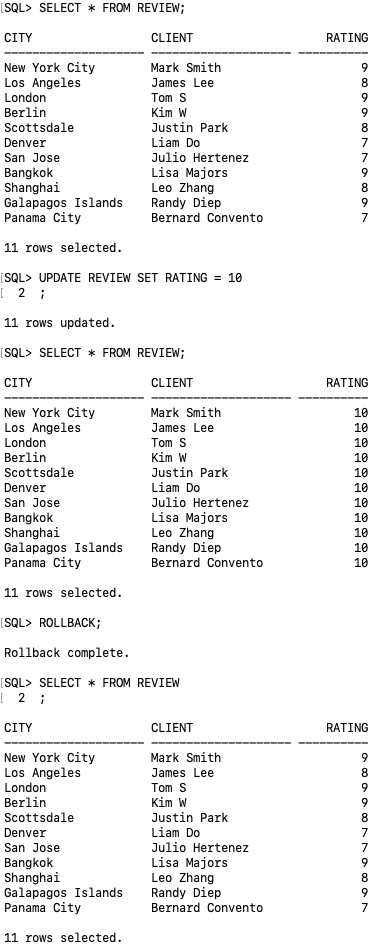


* + Delete: Deleting all reviews by Client ‘Tom S’. There is only one review.





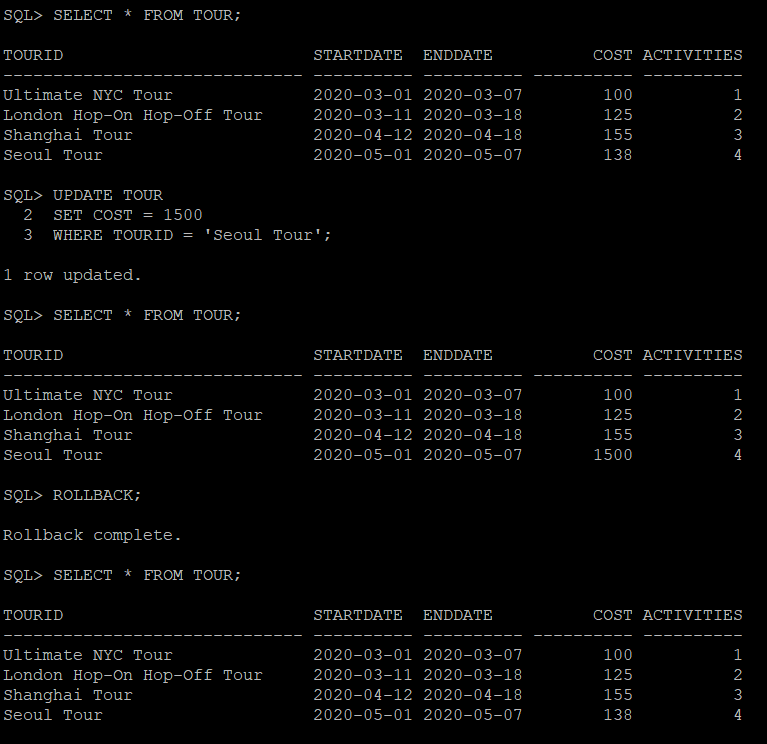
* Yegeon:
  + Update: Changed all ratings in Review to 10.



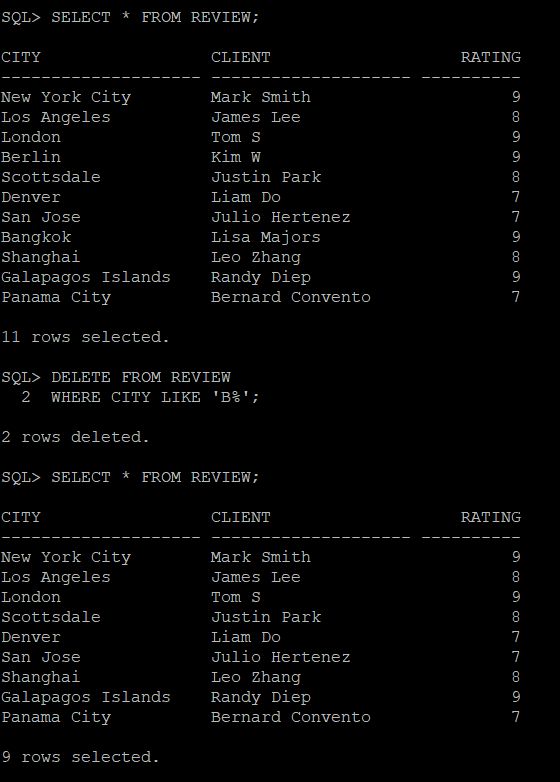
* + Delete: Deleted all Review rows with ratings less than or equal to 7.

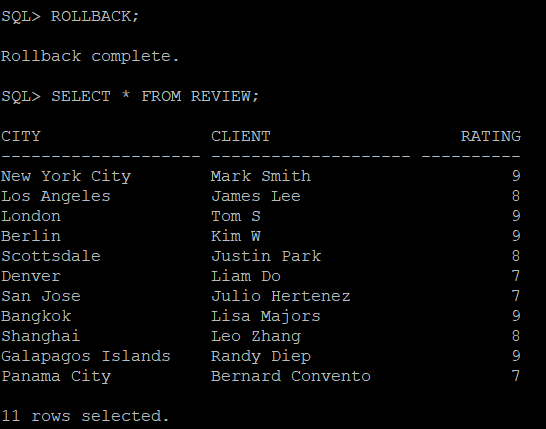


* Quan:
  + Update: Changed the cost of the tour for Seoul to $1,500.



* + Delete: Deleted all Review rows for Cities that start with the letter “B”.





1. Summary

* Tracy: Overall, I learned a lot throughout the entire project. It was interesting learning about creating tables, inserting into tables, updating and deleting them using SQL. I had a nice experience in using SQL to create a database regarding cities, tours, etc. Before starting this project, I had no experience of how SQL works and through this project, it taught me a lot about how it actually works. The labs and lectures from each week helped a lot in accomplishing this project. It was also nice working in a group because it gave each of us an opportunity to ask each other questions and also tackle different responsibilities. If we had more time, I would like to work more on practicing on queries and also creating more tables for our ERD. I hope to use what I learned from this project to develop more in my knowledge of using SQL for databases in future projects.
* Matthew: Two major lessons I learned over the course of the project were managing scope and managing group unity. Over the course of the project, we had a fair number of ideas regarding how we wanted to set up the database, but only some of them turned out the way we hoped. With more prior knowledge of SQL, we might have been able to skip dealing with some of those hurdles. When it came to group unity, many of our members had widely different schedules that stifled our ability to meet in person, which is where we found we could work more diligently and share and act on new ideas. Towards the end of the project, when COVID-19 kept us from meeting at all, ensuring everyone was on the same page and making their contribution was difficult.
* Yegeon: This course project taught me how to work with the database that can be used in real life. Table actions such as creating, inserting, and manipulating data are an essential part of working with a database, and using SQL to perform these actions were interesting. I personally use other languages such as JavaScript or python to work with a database, but realized that SQL can be much simpler and more powerful than some of the technologies I am currently using. The lectures and readings gave me more insights into what I can do with SQL, and I think it was very helpful since there may be a chance to use it in future jobs that require SQL. All of the group members had a busy schedule, but I think Matthew did a really good job organizing and coordinating the project. Overall, this class gave me more insight into how a database works and how to work with it.
* Quan: This entire project is the first time I get to work with databases and SQL in general. Throughout the course, I have multiple hand-ons experience drawing ERD, creating relational schema and working with SQL in Oracle. For someone who knows little to nothing about SQL and databases, this project gives me lots of chances to practice and create a usable database. I also learned a lot from other group members, as they are willing to help with my problem despite any circumstances. If we as a group had more time, we would be able to expand our database with more tables and more effective queries. Thanks to this course and this group project in specific, I am now more confident in working with SQL and I look forward to practicing with it more in the future.

1. Appendix

11.1 Breakdown of division of work

* All Members:
  + Researched five locations, wrote their INSERT statements.
  + Wrote three query statements for Part 8, and an update and delete statement for Part 9.
  + Contributed to non-SQL portions of the final report.
* Tracy:
  + Wrote CREATE statements for Climate and BestMonths tables.
  + Compiled references document for submission (see Appendix 11.4).
* Matthew:
  + Wrote CREATE statements for City, Culture, and HistoricalSites tables.
  + Finalized ERD file for submission.
  + Compiled final CREATE and INSERT files for submission.
* Yegeon:
  + Wrote CREATE statements for Tour, Activities, and TourPackage tables.
* Quan:
  + Wrote CREATE statements for Client and Review tables.

11.2 Sample input/output/program listing:

See Part 8: Data Queries

11.3 Design Decisions

Over the course of the project, the number of tables we planned to create varied significantly. This was due to changing opinions about what to include in the database, as well as feedback from instructors regarding how best to represent certain pieces of data. A major turning point in the project was understanding how SQL could and couldn’t store what we wanted the database to contain. Members of the group with object-oriented programming language backgrounds suggested values such as best months or tour activities could be stored in a list within a table row. However, implementing this did not come to pass, and was instead replaced with the various tables that serve as lists connected to other tables via foreign keys. While these tables are finite in regards to how many “months” or “activities” they can hold, we concluded that they were sufficient in demonstrating the idea of list-esque data.

11.4 References

Several websites were used in researching the locations. They are listed in a separate document included in the submission.