COMP1204 Data management: Coursework 2

Kritagya Gurung

May 23, 2019

Student ID: 30220238

1 The Relational Model

1.1 EX1

Attribute name	data type	Attribute name	data type	Attribute name	data type
Hotel ID	int	Overall rating	float	Avg. Price	int
URL	string	Author	string	Content	string
Date	string	No. Reader	int	No. Helpful	int
Overall	int	Value	int	Rooms	int
Location	int	Cleanliness	int	Check in/Front desk	int
Service	int	Business service	int		

Relation Schema:
(R1)Review(Hotel ID, Overall rating, Avg. Price, URL, Author, Content, Date, No. Reader, No. Helpful, Overall, Value, Rooms, Location, Cleanliness, Check in/Front desk, Service, Business service).
The primary key consists of Hotel ID, Author and date. This is because all other attributes rely on the attributes mentioned and in order to write a review, you must have an author, the date that review was written on and what hotel the review is for.

1.2 EX2

Hotel ID $\rightarrow$ Overall rating, Avg. Price, URL
Hotel ID is the determinant for Overall rating, Avg. Price and URL. This is because each hotel has their own distinct properties.
Author, Date $\rightarrow$ Content, No. Reader, No. Helpful, Overall, Value, Rooms, Location, Cleanliness, Check in/Front desk, Service, Business service.
Author and Date should be the determinant for all other attributes since it is the author who decides the scores and the content of the review on a hotel. Additionally, the author must stay at the hotel thus date is used to show when the hotel was reviewed.
Candidate keys:
Hotel ID, Author, Date

1.3 EX3

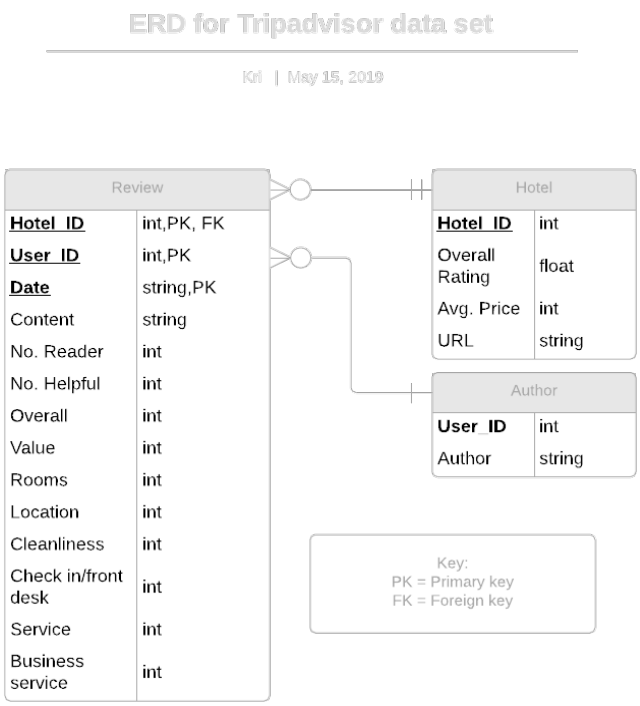
Attribute name	data type	Attribute name	data type	Attribute name	data type
Hotel ID	int	Overall rating	float	Avg. Price	int
URL	string	Author	string	Content	string
Date	string	No. Reader	int	No. Helpful	int
Overall	int	Value	int	Rooms	int
Location	int	Cleanliness	int	Check in/Front desk	int
Service	int	Business service	int	User ID	int

Hotel(Hotel ID, Avg. Price, Date)
Review(Hotel ID, User ID, Content, Date, No. Reader, No. Helpful, Overall, Value, Rooms, Location, Cleanliness, Check in/Front desk, Service, Business service)
Author(User ID, Author)
Primary Key : Hotel ID, User ID, Date

Foreign Key : Hotel ID, User ID

2 Entity-Relationship Diagramming

2.1 EX4



3 Relational Algebra

3.1 EX5

$$\sigma_{User\_ID=given\_ID}(Review) \tag{1}$$

3.2 EX6

$$\sigma_{No.reviewsgiven>2}({}_{User\_ID}\gamma_{count(User\_ID)\rightarrow No.reviewsgiven}(Review)) \bowtie Author \tag{2}$$

3.3 EX7

$$\sigma_{No.reviewsreceived>10}({}_{Hotel\_ID}\gamma_{count(Hotel\_ID)\rightarrow No.reviewsreceived}(Review)) \tag{3}$$

3.4 EX8

$$\sigma_{Averagecleanliness \geq 4.5 \wedge Overallrating > 3}((_{Hotel\_ID}\gamma_{avg(Cleanliness) \rightarrow Averagecleanliness}(Review)) \bowtie (Hotel)) \quad (4)$$

4 SQL Queries

4.1 EX9

CREATE TABLE HotelReviews(Hotel_ID INTEGER, Overall_rating REAL, Average_price INTEGER, URL TEXT, Author TEXT, Content TEXT, Date TEXT, Number_of_readers INTEGER, Number_of_helpful INTEGER, Overall INTEGER, Value INTEGER, Rooms INTEGER, Location INTEGER, Cleanliness INTEGER, Check_in_and_Front_desk INTEGER, Service INTEGER, Business_service INTEGER, PRIMARY KEY(Hotel_ID, Author, Date));

4.2 EX10

Bash scripting not available.

4.3 EX11

CREATE TABLE Hotel (Hotel_ID INTEGER PRIMARY KEY, Overall_rating REAL, Average_price INTEGER, URL TEXT);
CREATE TABLE Author (User_ID INTEGER PRIMARY KEY, Author TEXT);
CREATE TABLE Review (Hotel_ID INTEGER, User_ID TEXT, Content TEXT, Date TEXT, Number_of_readers INTEGER, Number_of_helpful INTEGER, Overall INTEGER, Value INTEGER, Rooms INTEGER, Location INTEGER, Cleanliness INTEGER, Check_in_and_Front_desk INTEGER, Service INTEGER, Business_service INTEGER, FOREIGN KEY (Hotel_ID) REFERENCES Hotel(Hotel_ID), FOREIGN KEY (User_ID) REFERENCES Author(User_ID), PRIMARY KEY (Hotel_ID,User_ID,Date));

4.4 EX12

As the bash scripting does not work as intented, I have made my custom test data. Here is the test data set I used. INSERT INTO Hotel(Hotel_ID,Overall_rating,Average_price,URL) VALUES(11111,2.0,13,"test url1"), (11112,4,123,"test url2");
INSERT INTO Author(User_ID,Author) VALUES(1,"testAuthor1"), (2,"testAuthor2"), (3,"testAuthor3");
INSERT INTO Review(Hotel_ID, User_ID, Content, Date,Number_of_readers, Number_of_helpful, Overall, Value, Rooms, Location, Cleanliness, Check_in_From_desk, Service, Business_service) VALUES(11111,1,"contentTest","dateTest",1,2,3,4,5,6,7,8,9,10), (11112,2,"contentTest","dateTest",1,2,3,4,5,6,7,8,9,10), (11111,3,"contentTest","dateTest",1,2,3,4,5,6,7,8,9,10), (11112,2,"contentTest","dateTest2",1,2,3,4,5,6,7,8,9,10), (11111,2,"contentTest","dateTest",1,2,3,4,5,6,7,8,9,10), (11112,3,"contentTest","dateTest",1,2,3,4,5,6,7,8,9,10), (11112,3,"contentTest","dateTest2",1,2,3,4,5,6,7,8,9,10), (11112,3,"contentTest","dateTest3",1,2,3,4,5,6,7,8,9,10), (11112,3,"contentTest","dateTest4",1,2,3,4,5,6,3,8,9,10), (11112,3,"contentTest","dateTest5",1,2,3,4,5,6,7,8,9,10), (11112,3,"contentTest","dateTest6",1,2,3,4,5,6,7,8,9,10), (11112,3,"contentTest","dateTest7",1,2,3,4,5,6,3,8,9,10), (11112,3,"contentTest","dateTest8",1,2,3,4,5,6,7,8,9,10), (11112,3,"contentTest","dateTest9",1,2,3,4,5,6,7,8,9,10),

(11112,3,"contentTest","dateTest10",1,2,3,4,5,6,7,8,9,10), (11112,3,"contentTest","dateTest11",1,2,3,4,5,6,3,8,9,10);
DROP TABLE IF EXISTS HotelReviews;

4.5 EX13

I would added indexes to Hotel_ID and User_ID attributes as they are used as the main attributes in table joins, they are used in multiple queries and are foreign keys.
CREATE INDEX Hotel_ID_Index ON Hotel(Hotel_ID);
CREATE INDEX User_ID_Index ON Author(User_ID);

4.6 EX14

ex5.sql
SELECT * FROM Review WHERE User_ID=2;
(replace 2 with a given id)
ex6.sql
SELECT Hotel_ID,count(*) AS Number_of_reviews_recieved FROM Review GROUP BY Hotel_ID HAVING count(*); > 10;
ex7.sql
SELECT Review.User_ID,Author.Author , count(*) AS Number_of_reviews_given FROM Review INNER JOIN Author ON Author.User_ID = Review.User_ID GROUP BY Review.User_ID HAVING count(*) > 2;
ex8.sql
SELECT Review.Hotel_ID, Hotel.Overall_rating, Hotel.Average_price, Hotel.URL , avg(Review.Cleanliness) AS Average_cleanliness FROM Review INNER JOIN Hotel ON Review.Hotel_ID = Hotel.Hotel_ID GROUP BY Review.Hotel_ID HAVING Average_cleanliness $\geq$ 4.5 AND Overall_rating > 3;

5 Conclusions

5.1 EX15