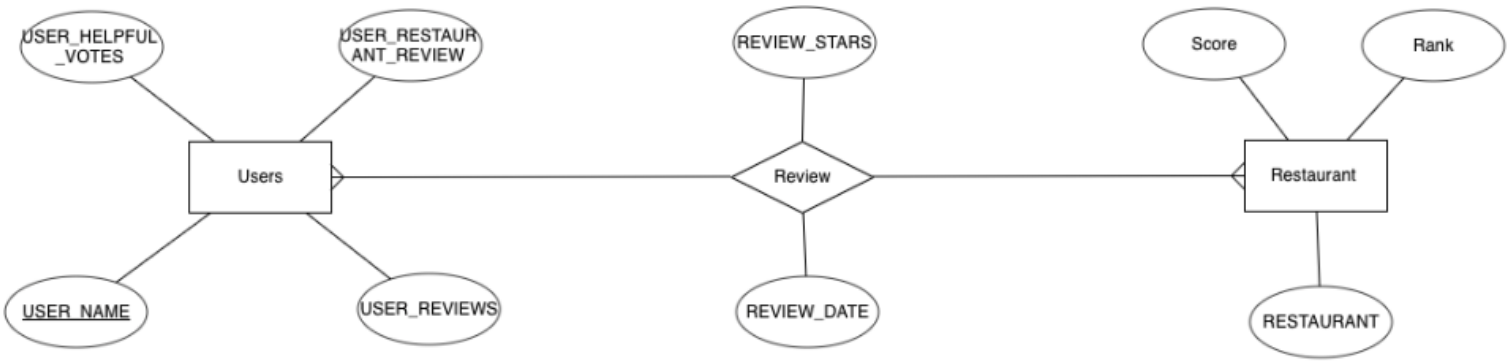
Hemin Patel

Professor Imielinski

Prin Info & Data Mgt

26 October 2021

1) For the attached table TripAdvisor.csv reconstruct reasonable entities, relationships and attributes and draw ER diagrams.



---

2) Create the table TripAdvisor in mysql on your local mysql server and upload the TripAdvisor.csv to it.

```
CREATE TABLE `tripadvisor` (
 `RESTAURANT` text,
 `RANK` int DEFAULT NULL,
 `SCORE` double DEFAULT NULL,
 `USER_NAME` text,
 `REVIEW_STARS` int DEFAULT NULL,
 `REVIEW_DATE` text,
```

```
  `USER_REVIEWS` int DEFAULT NULL,
  `USER_RESTAURANT_REVIEWS` int DEFAULT NULL,
  `USER_HELPFUL_VOTES` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

---

3) Identify reasonable functional dependencies

Restaurant -> rank, score

```
SELECT DISTINCT 'True' as 'nonempty'
From hw4.tripadvisor t1
Where Not Exists (SELECT * from hw4.tripadvisor t1, hw4.tripadvisor t2 where
t1.RESTAURANT = t2.RESTAURANT and (t1.RANK != t2.RANK or t1.SCORE !=
t2.SCORE))
UNION
SELECT DISTINCT 'False' as 'nonempty'
FROM hw4.tripadvisor t1
Where Exists (SELECT * from hw4.tripadvisor t1, hw4.tripadvisor t2 where
t1.RESTAURANT = t2.RESTAURANT and (t1.RANK != t2.RANK or t1.SCORE !=
t2.SCORE))
```

Restaurant, user_name, review_date -> user_review, user_restaurant_review, user_helpful_votes, review_stars

```
SELECT DISTINCT 'True' as 'nonempty'
From hw4.tripadvisor t1
```

```
Where Not Exists (SELECT * from hw4.tripadvisor t1, hw4.tripadvisor t2 where
(t1.RESTAURANT = t2.RESTAURANT) and (t1.USER_NAME = t2.USER_NAME)
and (t1.REVIEW_DATE = t2.REVIEW_DATE) and (t1.USER_REVIEWS !=
t2.USER_REVIEWS and (t1.USER_RESTAURANT_REVIEWS !=
t2.USER_RESTAURANT_REVIEWS) and (t1.USER_HELPFUL_VOTES !=
t2.USER_HELPFUL_VOTES) and t1.REVIEW_STARS != t2.REVIEW_STARS))
UNION
SELECT DISTINCT 'False' as 'nonempty'
FROM hw4.tripadvisor t1
Where Exists (SELECT * from hw4.tripadvisor t1, hw4.tripadvisor t2 where
(t1.RESTAURANT = t2.RESTAURANT) and (t1.USER_NAME = t2.USER_NAME)
and (t1.REVIEW_DATE = t2.REVIEW_DATE) and (t1.USER_REVIEWS !=
t2.USER_REVIEWS and (t1.USER_RESTAURANT_REVIEWS !=
t2.USER_RESTAURANT_REVIEWS) and (t1.USER_HELPFUL_VOTES !=
t2.USER_HELPFUL_VOTES) and t1.REVIEW_STARS != t2.REVIEW_STARS))
```

---

4) Show that TripAdvisor is not in the BCNF. This requires identifying functional dependency which violates BCNF. Decompose TripAdvisor to BCNF.

Restaurant table

```
Create table hw4.Rest
As (Select distinct t.RESTAURANT, t.RANK, t.SCORE #Made restaurant primary: PRIMARY KEY (`RESTAURANT`(45))
From hw4.tripadvisor t);
```

User table

```
Create table hw4.Users
As (Select distinct t.USER_NAME, SUM(t.USER_REVIEWS) as USER_REVIEWS ,
SUM(USER_RESTAURANT_REVIEWS) as USER_RESTAURANT_REVIEWS,
SUM(t.USER_HELPFUL_VOTES) as USER_HELPFUL_VOTES
From hw4.tripadvisor t Group BY USER_NAME)

DELETE from hw4.Users WHERE USER_NAME = '0';
```

Review table

```
Create table hw4.Reviews
As Select t.RESTAURANT, t.USER_NAME, MAX(REVIEW_DATE) as
REVIEW_DATE, AVG(REVIEW_STARS) as REVIEW_STARS
From hw4.tripadvisor t
Group BY RESTAURANT, USER_NAME;
```

---

5) Download their content to separate csv files and submit them as well.

I have attached them to the zip file.

---

6) Demonstrate that your decomposition has lossless join property.

```
select REST.USER_NAME, REVIEW.SCORE from REST, REVIEW
where REST.RESTAURANT = REVIEW.RESTAURANT
```