FRAPPER
Colin

Database theory
2DV513

Assignment 2 - Tasks

Note 1 : I'm sorry about my english.
Note 2 : The E/R diagrams are realised with draw.io, it's an open source.

Task 1. Relational algebra

Suppose relations R and S have n and m tuples, respectively. Give the minimum and maximum numbers of tuples that the results of the following expressions can have:

1. R U S
   minimum : it's the maximum between n and m, it happens when all tuples in one table are in the other table.
   maximum : n + m, if all the tuples are differents

2. R ⋈ S
   minimum : 0, if R and S don't have tuples in commun
   maximum : it's the minimum between m and n, it's happen when all tuples in one table are in the other table

3. $\sigma_{(c)}$ (R) x S

   minimum : 0, if the result of the selection return 0 lines
   maximum : it's m x n, it's happen when the result of the selection return all the line of R.

4. $\pi_{(L)}$ (R) \ (S) (set difference)

   minimum : 0, if all the result of the projection are in S
   maximum : n, if all the tuples of the result of the projection aren't in S.

Task 2. Normalization

For each of the following relational schemas and sets of functional dependencies:

1. R(A,B,C,D,E) with functional dependencies AB -> C, DE->C, and B -> D.

1.1 Indicate all the BCNF violations. Do not forget to consider dependencies that are not in the given set but follow from them.

First, if we look more on the FD, we notice that there is an additional FD ; BE -> C, we can found it by doing B->D, BE-> DE (aug with E), DE ->C, so BE -> C.

Let's look if R is in BCNF ;

Definition : We say a relation R is in BCNF if whenever X -> A is a nontrivial FD that holds in R, X is a superkey.

AB -> C : The closure of AB is {A,B,C}, but the left side of this FD is not a superkey, so it's a BCNF violation.

DE -> C : The closure of DE is {D,E,C}, but the left side of this FD is not a superkey, so it's a BCNF violation.

B -> D : The closure of B is {B,D}, but the left side of this FD is not a superkey, so it's a BCNF violation.

BE->C. The closure of BE is {B,E,C}, but the left side of this FD is not a superkey, so it's a BCNF violation.

1.2. Decompose the relations, as necessary into collectionsof relations that are in BCNF.

.

Decompostion :

Rules :

Decompose R using X -> B

- Replace R by relations with schemas :

1. R1 = X^+
2. R2 = (R-$X^{+)}$ U X

Let's try with the first BCNF violation AB -> C. AB^+ = {A,B,C}.

R1 = {A,B,C}

R2 = {A,B,D,E}

Project given FD's F onto the two new relations.

Rules :

1. Compute the closure of F = all nontrivial FD's that follow from F
2. Use only those FD's whose attributes are all in R1 or all in R2.

In R1, the only FD whose attributes are all in R1 is AB->C. So there is BCNF.

In R2, the FD B->C is in BCNF violation ;

Decomposition :

R_{11} = {B,C}

R_{22} = {A,B,E}

In R_{11}, the only FD whose attributes are all in R1 is B->C. So there is in BCNF

In R_{22}, if we compute {A,B,E} : $\{A,B,E\}^+$ = {A,B,C,D,E}, so there is in BCNF.

Finally the union give : {B,C},{A,B,C},{A,B,E}

1.3 Indicate all 3NF violation

Definition : An FD X->A violates 3NF if and only if X is not a superkey, and A is not a prime

All the Fds are in violation of 3NF because none of the left hand sides is a superkey and on the right hand side, there is no part of a key

1.4 Decompose the relations, as necessary into collections of relations that are in 3NF.

Let's try to decompose by B->D this time. ;
R1 = {B,D}
R2 = {A,B,C,E}

In R2 the FD AB->C and BE->C hold.
Both are in violation of 3NF.

Decompose by AB->C ;

R_{11} = {A,B,C}
R_{22} = {A,B,E}
Each of these relations are in 3NF.

Finally the union of the realtion are {B,D},{A,B,C},{A,B,E}

2. R(A,B,C,D,E) with AB ->C, C->D, D->B and D -> E

2.1 Indicate all the BCNF violations. Do not forget to consider dependencies that are not in the given set but follow from them.

Let's look if R is in BCNF
Definition : We say a relation is in BCNF if whenever X -> A is a nontrivial FD that holds in , is a superkey.

AB -> C : The closure of AB is {A,B,C,D,E}*, {A,B} is a superkey

* 1. {A,B} = {A,B}
2. AB->C, so {A,B,C}
3. C->D , so {A,B,C,D}
4. D->E, so {A,B,C,D,E}
5. It's stable, {A,B} is a superkey.

C->D : The closure of DE is {C,D,B,E}, but the left side of this FD is not a superkey, so it's a BCNF violation.

D->B : The closure of B is {D,E,B}, but the left side of this FD is not a superkey, so it's a BCNF violation.

D->E : The closure of B is {D,E,B}, but the left side of this FD is not a superkey, so it's a BCNF violation.

There is also two more FD not given C->B and C->E

2.2. Decompose the relations, as necessary into collections of relations that are in BCNF.

Decompostion :

Decompose R using X -> B
- Replace R by relations with schemas :
1. R1 = X^+
2. R2 = (R-X^+) U X

Let's try with the first BCNF violation C -> D. C^+ = {C,D,B,E}.
R1 = {C,D,B,E}
R2 = {C,A}

In R2, if we compute {C,A} ; $\{C,A\}^+$ = {A,B,C,D,E} so BCNF.
In R1, D->E is in BCNF violation
Decomposition :

R_{11} = {D,E,B}
R_{22} = {D,A,C}

In R_{11}, the only FDs whose attributes are all in R_{11} are D->B, D->E, si there is BCNF violation
In R_{22} if we compute {D,A,C} ; $\{D,A,C\}^+$ = {A,B,C,D,E}, so there is no violation

Finally the union give : {C,A} {D,E,B} {D,A,C}

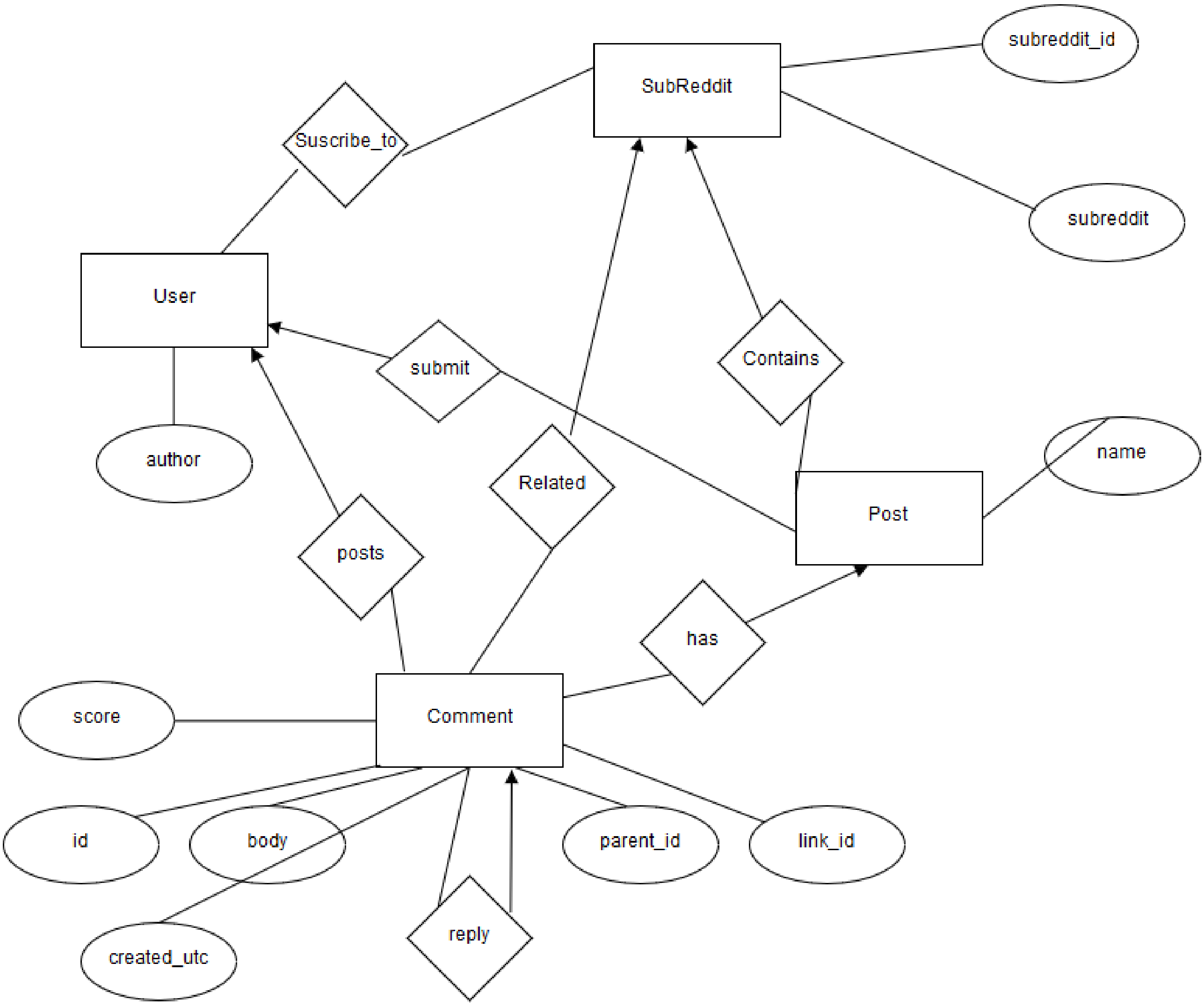
2.3. Indicate all the 3NF violations

Definition : An FD X->A violates 3NF if and only if X is not a superkey, and A is not a prime

AB->C, AB is a superkey, C is a prime. NO 3NF violation
C->D, C is not a superkey. 3NF violation
D->B, D->E, 3NF violation

Task 3 : Setting up the Reddit database

Task 3.1 : Provide an E/R diagram for your design as well as schemas with types.



Explication : An user(s) can suscribe to subreddits, it's a group who are made specifically for certain topics.
Each of these subreddits can be subscribed to by multiple people and multiple people can subscribe to the same subreddit, that's why I choose to make a many to many relationship.
Each Subreddit allows users to post submissions to them. Each subreddit can have many posts, but a specific post can only be in one subreddit.
Also, a user does not need to be subscribed to a subreddit to create a post in it, that's why I also created a separate relationship directly to the post from the user (submit).
A user can edit many posts, but a post is related to only one user. Honnestly I don't think the table Post is necessary here, because in the Json file you gave us, there is only comment, but I wanted to design it like this because this is the most realistic way to do it.

Each post has a comments section. In the comments section, users can post their own comments, but they can also reply to other comments in the comment section. This is why I have created a one-to-many self-relationship for comment to another comment.
I used a one-to-many because a comment can have many sub-comments, but a sub-comment can only be in reply to one comment.
I also created a relation between comment and subreddit because (I think, I'm not sure), but it's possible to have a subreddit without any posts and only comment.

So, I know this isn't the optimal shema that I can design according to this assignment, if I wanted to do it more optimal ; the thing is that you gave us a file (Json file) who contains only Comment, so I can also only create one table comment, but I choose to make my E/R diagram the more realistic as I can and it's also avoid redondancy about the author or subreddit, ...

The schema :

```
User(author TEXT PRIMARY KEY)
```

Note : I suppose that you can't choose the same name as another person (and this is the truth).

```
SubReddit(subreddit_id TEXT PRIMARY KEY,
          subreddit TEXT UNIQUE)
```

Note : I din't decode the base36 for the subreddit_id, also the subreddit name is unique (two subreddits can have the same name).

```
Suscribe_to(s_subreddit TEXT ,s_author TEXT ,
            FOREIGN KEY(s_subreddit) REFERENCES SubReddit(subreddit_id),
            FOREIGN KEY(s_author) REFERENCES User(author))
```

Note : This is the result of the many to many relationship between User and SubReddit, one table with two foreign keys, but again with the file you gave us I can't create this table correctly because I don't have the information about if the user just post a comment in this subreddit, or if he suscribed to it.

```
Post(name_post TEXT PRIMARY KEY,
     post_subreddit TEXT,
     post_author TEXT,
     FOREIGN KEY(post_subreddit) REFERENCES SubReddit(subreddit_id),
     FOREIGN KEY(post_author) REFERENCES User(author))
```

Note : Ok here as I said before this table is kind of useless for this assignment, it contains the id_post (name_post here) as a primary key, then the subreddit where the post as been posted, and the name of the author who edit this post, but again I can't create this table correctly because not enough information.

```
Comment(id_comment INTEGER PRIMARY KEY,
        parent_id TEXT,
        link_id TEXT,
        created_utc TEXT NOT NULL,
        body TEXT,
        score INTEGER,
        author_comment TEXT,
        subreddit_comment TEXT,
        post_comment TEXT,
        FOREIGN KEY(subreddit_comment) REFERENCES SubReddit(subreddit_id),
        FOREIGN KEY(author_comment) REFERENCES User(author)
        FOREIGN KEY(post_comment) REFERENCES Post(name_post))
```

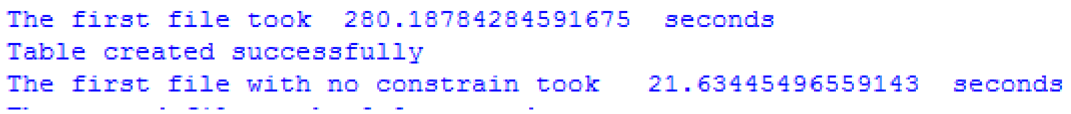
Note : This is the main table, it contains almost all the information to answer to the sql question later, the id_comment is decode in base36, that's why it's an integer.

Task 4 : Importing data

How do these constraints affect the import time. Measure and experiment with turning these on and off. Report measured times and an discuss why you think the constraints affected the times.

Ok, so to mesure the time I used the timer on python, and indeed the constraints affect the import, this is because of the key, as it said in the slides « An index can be dense or sparse. A dense index contains a value for each row in the table, while a sparse does not have to. Indices created for primary keys or unique values are always dense ».

So this is the time for the first json file ;
With the constraints : 280 seconds (4 Minutes and 40 Seconds)
Without the constraints : 21 seconds



```
The first file took  280.18784284591675  seconds
Table created successfully
The first file with no constrain took   21.63445496559143  seconds
```

You can try it by running my python file.

I also try to run the other file, the second took around 25 minutes, I didn't try the third one.

Would it be reasonable to import and turn on constraints after? When?

Yes I think so, especially the foreign key, you can do this with ALTER TABLE ? NOCHECK CONSTRAINT ALL, but it's « impossible » to turn primary key or unique constraint, but I think we can do it for the database who contains few rows, but when the database become really bigger, the table with large rows benefits more from indices.

I also find a little article about constraints performance :
« Constraints have performance implications, both improving performance in some cases and consuming resources in others. Most of the time, the costs are worth it in that we can improve the quality of our data without the need to build complex application logic or database scripts that try to keep data in order. A well thought out database design will often answer constraint questions up-front. This makes it so that we do not need to address technical debt years later when we suddenly realize that one column should always have been populated, or that dinosaurs shouldn't be assigned to non-existent species. Constraints are valuable tools, and ones that can improve documentation, performance, data integrity, and make sense of what would otherwise be complex data. »

Source : http://www.sqlshack.com/the-benefits-costs-and-documentation-of-database-constraints/

Task 5 : Queries

You have two choices : you can install an add from firefox
https://addons.mozilla.org/en/firefox/addon/sqlite-manager/
and then go to firefox -> tools -> Sqlite manager, then create a file example.db, change in my python file the path and then in Sqlite manager -> Execut SQL and copy paste the request

OR

create an example.db and just remove the comment in the python file and run it.

The queries has been realised with the first file.

1. How many comments have a specific user posted?

Specific user :

```
SELECT COUNT (*) FROM Comment WHERE author_comment = 'igiveyoumylife'
```

Note : In the table comment there is all the comments, with the name of the author, even the redondancy so we just have to count the number of line where author = a specific user (here igiveyoumylife).

Random user :

```
SELECT COUNT(*),author_comment FROM Comment WHERE author_comment =
(SELECT author_comment FROM Comment ORDER BY RANDOM() LIMIT 1)
```

Note, the same as the previous one except that to choose a random name we use ORDER BY RANDOM LIMIT 1, it will choose a random name.

2. How many comments does a specific subreddit get per day?

Specific subreddit :

```
SELECT (COUNT (*)/30) AS COMMENT_PER_DAY FROM Comment JOIN
SubReddit ON Comment.subreddit_comment = SubReddit.subreddit_id WHERE subreddit =
'reddit.com'
```

Note : I must say I was surprised about the question especially the « per day », this doesn't make any sense, I mean this is a middle, sometimes, the week end for example there is a lot more comments than on a week day, anyway I just count the comments in a month and divide it by 30.

Random subreddit :

```
SELECT (COUNT(*) / 30),subreddit FROM Comment JOIN SubReddit ON
Comment.subreddit_comment = SubReddit.subreddit_id WHERE subreddit =(SELECT
subreddit FROM SubReddit ORDER BY RANDOM() LIMIT 1)
```

Note : same as part 2 question 1

3. How many comments include the word ‘lol’?

```
SELECT COUNT(*)FROM Comment WHERE body LIKE '%lol%'
```

Note : we must use the « attribute » LIKE.

4. Users that commented on a specific link has also posted to which subreddits?

Specific link :

```
SELECT DISTINCT subreddit FROM SubReddit JOIN Comment ON
SubReddit.subreddit_id = Comment.subreddit_comment WHERE author_comment IN (Select
author_comment FROM Comment WHERE link_id = 't3_2zvjj')
```

Note : The attribute « DISTINCT » permit to avoid the redondancy, here the specific link is t3_2zvjj, I also use the attribute IN to make some test to see which one is the best between IN and =

Random link :

```
SELECT DISTINCT subreddit FROM SubReddit JOIN Comment ON  SubReddit.subreddit_id =
Comment.subreddit_comment WHERE author_comment IN (Select author_comment FROM
Comment WHERE link_id =(SELECT link_id FROM Comment ORDER BY RANDOM() LIMIT
1))
```

5. Which users have the highest and lowest combined scores? (combined as the sum of all scores)

highest :

```
SELECT  author_comment, SUM(score) AS highscores FROM Comment GROUP BY
author_comment Order by highscores DESC LIMIT 2
```

Note : Here, I used SUM to make the sum of the score, also use GROUP BY on author, and Order by to sort the result the column highscores. LIMIT 2 permit to return only the two first line (Don't take in consideration [deleted])

lowest :

```
SELECT  author_comment, SUM(score) AS lowestscore FROM Comment GROUP BY
author_comment Order by lowestscore LIMIT 1
```

Note : Same as the previous question

6. Which subreddits have the highest and lowest scored comments?

Highest :

```
SELECT subreddit, SUM(score) AS highscore FROM SubReddit JOIN Comment ON
SubReddit.subreddit_id = Comment.subreddit_comment GROUP BY subreddit ORDER BY
highscore DESC LIMIT 1
```

Note : same as previous one

lowest :

```
SELECT subreddit, SUM(score) AS lowestscore FROM SubReddit JOIN Comment ON
SubReddit.subreddit_id = Comment.subreddit_comment GROUP BY subreddit ORDER BY
lowestscore  LIMIT 1
```

7. Given a specific user, list all the users he or she has potentially interacted with (i.e., everyone who as commented on a link that the specific user has commented on).

Specific user :

```
SELECT DISTINCT author_comment FROM Comment Where  link_id IN (SELECT
link_id FROM Comment WHERE author_comment ='igiveyoumylife'
```

Note : We need to check the author and find a commun link_id with the other author

Random user :

```
SELECT DISTINCT author_comment FROM Comment Where  link_id IN (SELECT
link_id FROM Comment WHERE author_comment = (SELECT author FROM User ORDER BY
RANDOM() LIMIT 1))
```

8. Which users has only posted to a single subreddit?

```
SELECT author_comment FROM Comment GROUP BY author_comment HAVING COUNT(*) = 1
```

Note : Here we use the attribute HAVING, so we're looking for all the author where the line appears only one time.

Optionnal question : Which subreddits share no users, i.e., have no users that have posted to the others.

```
SELECT subreddit FROM Comment JOIN SubReddit ON Comment.subreddit_comment = SubReddit.subreddit_id WHERE author_comment IN (SELECT author_comment FROM Comment GROUP BY author_comment HAVING COUNT(*) = 1) AND subreddit_comment IN (SELECT subreddit_comment FROM Comment GROUP BY subreddit_comment HAVING COUNT(*) = 1))
```

Note : This one is tricky and I'm not sure if it's working but I used the attribute AND,we need to have an author who post a single comment.