IS503 Database Concepts and Applications
Midterm Exam
18/11/2019

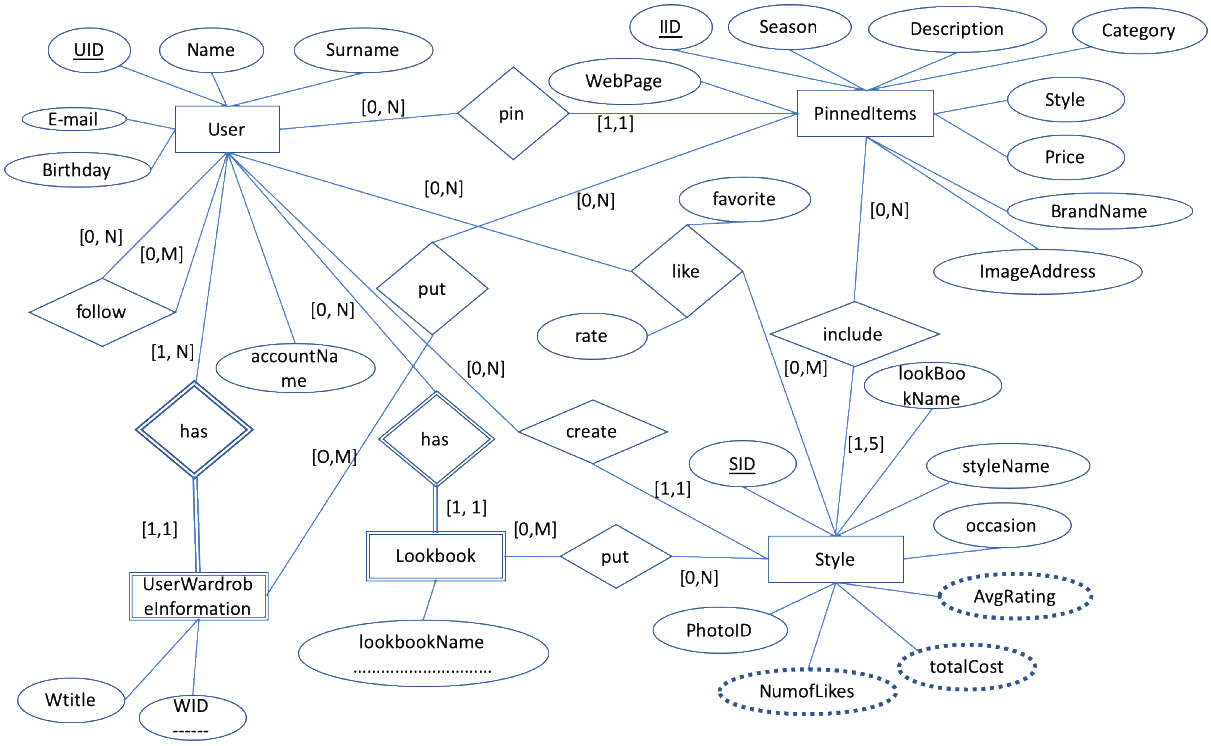
Consider the following case study while answering questions.

The case study is about a lookbook design application. Users are registered to the application by providing an account name, name, surname, e-mail address, and birthday. The system automatically assigns a user ID once the user has completed the registration process. Users can pin items from different shopping web sites to the system where they are required to enter its category (dress, top, tunic, t-shirt, jacket, etc.), price, brand name, web address of the item, image web address, description, season (all, spring/summer, fall/winter) and style (all, dressy, casual or sport). Once all the data is entered, the system assigns a unique ID for the item in the system. Users can browse the uploaded items in the system and they can add any item into one of their wardrobes (aka item gallery). They can have several wardrobes to store items. However, an item can be saved in only one of the wardrobes of the user. Each user has a default lookbook at the beginning but they can create many lookbooks for different occasions. The only constraint is that users can name these lookbooks, but they cannot have two lookbooks with the same name. However, the same name can be used by different users. Users can also create combinations of various items to create a style such as by including a top, shoes, bag, skirt and accessories and these styles can be only saved in lookbooks. A combination (aka style) cannot have more than five items. Different users can name each combination without any constraint i.e. two combinations can have the same name or a combination name might be null. The total cost of the combination is automatically calculated based on each selected item price and displayed. If a user leaves the system for good, their lookbooks and created combinations will be removed from the system automatically. Users can rate combinations and add them as their favourites in their own lookbooks. The system also displays the average rating of combinations. The same item can be used under different lookbooks of the same user. Users can define the occasion (all, evening, daytime, work) and season of their created combination. The image for the created combination is generated automatically by the system. Users can also follow other users. An item can belong to only one category.

Question 1 (30 pts):

Draw the ER/EER diagram of the case study and provide the relational schema of your design. Give any assumptions you have made. Provide min and max constraints in your diagram.

Consider the following partial relations (incomplete according to the above case study) while answering questions 2 and 3. **Do not add new attributes or relations while answering the questions.**



The schema:

User

UID	Name	Surname	E-mail	Birthday	accountName

PinnedItems

IID	ByUID	Category	Price	BrandName	WebPage	ImageAddress	Description	Season	Style

UserWardrobeInformation

UID	WID	Wtitle

UserItemWardrobeGallery

UID	IID	WID

UserLookbook

UID	lookBookName

Styles

SID	byUID	lookBookName	styleName	Occasion	Season	PhotoID	NumofLikes	TotalCost	Ave.Rating

StylesIncludeItems

SID	IID

StylesPutLookbooks

SID	UID	lookBookName

UserLikes

UID	SID	Favorite	Rate

UserFollowers

UID	FUID

PinnedItems

IID	ByUID	Category	Price

UserFollowers

UID	FUID

CreateNewCombination

SID	ByUID	Season	Occasion	TotalPrice

UserLikes

UID	SID	Favorite	Rate

User

UID	Name

IID is the key of the item and SID is the key of the style.

Question 2 (32 pts): (8 pts each)

Write the relational algebra of the following queries.

a. Find the users who pinned the highest number of items both under the category of dress and category of jackets and pinned items from all the categories.
b. Find the users whose none of the styles (combinations) has been marked as favorite by any other users and who have been followed by at least three users.
c. Find the users who **always** create styles for the same occasion and **always** put other users' created styles belonging to that same occasion as favourite. Note that these users will not have any styles for other occasions.
d. Find the users with the highest number of followers who created styles for all types of occasions (such as at least one style for "all", one for "evening", one for "daytime" and one for "work") and none of her created combinations has been marked as favourite. Note that you should first find users who satisfy the latter two conditions and then find the popular user among them.

Answer 2 (32 pts): (8 pts each)

a. $R1(BYUID, COUNTDRESSES) \leftarrow {}_{BYUID}\mathcal{F}_{COUNT\,*}(\sigma_{CLASS="DRESS"}(PINNEDITEMS))$
$R2(BYUID, COUNTJACKETS) \leftarrow {}_{BYUID}\mathcal{F}_{COUNT\,*}(\sigma_{CLASS="JACKET"}(PINNEDITEMS))$
$R3(BYUID, COUNTALLCATEGORIES) \leftarrow {}_{BYUID}\mathcal{F}_{COUNT\,*}(\Pi_{BYUID,\,CATEGORY}(PINNEDITEMS))$
$R11 \leftarrow \mathcal{F}_{MAX\ COUNTDRESSES}(R1) * R1$
$R21 \leftarrow \mathcal{F}_{MAX\ COUNTJACKETS}(R2) * R2$
$R31 \leftarrow \mathcal{F}_{MAX\ COUNTALLCATEGORIES}(R3) * R3$
$RESULT \leftarrow \Pi_{BYUID}(R11) \cap \Pi_{BYUID}(R21) \cap \Pi_{BYUID}(R31)$

b. $R1 \leftarrow \Pi_{UID}(USER) - \Pi_{BYUID}(USERLIKES*CREATENEWCOMBINATION)$
$R2(UID, CNTF) \leftarrow {}_{FUID}\mathcal{F}_{COUNT\,*}(USERFOLLOWERS)$
$R3(UID) \leftarrow \Pi_{UID}(\sigma_{CNTF>=3}(R2))$
$R4(UID) \leftarrow \Pi_{BYUID}(\sigma_{FAVORITE=NULL}(USERLIKES*CREATENEWCOMBINATION))$
$RESULT \leftarrow (R1 \cup R4) \cap R3$

c. $R0(BYUID, OCCASION) \leftarrow \Pi_{BYUID,OCCASION}(CREATENEWCOMBINATION)$
$R1(UID, CNTO) \leftarrow {}_{BYUID}\mathcal{F}_{COUNT\,*}(R0)$
$R2(UID) \leftarrow \Pi_{UID}(\sigma_{CNTO=1}(R1))$
$R00(UID, OCCASION) \leftarrow (\Pi_{UID,OCCASION}(USERLIKES*CREATENEWCOMBINATION))$
$R3(UID,CNTO) \leftarrow {}_{UID}\mathcal{F}_{COUNT\,*}(R00)$
$R4(UID) \leftarrow \Pi_{UID}(\sigma_{CNTO=1}(R3))$
$R3(UID, OCCASION) \leftarrow R2*R4*(\rho_{(UID,OCCASION)}(R0)*R00)$

d. $R1(BYUID) \leftarrow \Pi_{BYUID,OCCASION}(CREATENEWCOMBINATION) \div \Pi_{OCCASION}(CREATENEWCOMBINATION))$
$R2(FUID) \leftarrow \Pi_{BYUID}(CREATENEWCOMBINATION*R1) - \Pi_{BYUID}(CREATENEWCOMBINATION* \sigma_{FAVORITE<>NULL}(USERLIKES))$
$R3(FUID,CNT) \leftarrow {}_{FUID}\mathcal{F}_{COUNT\,*}(USERFOLLOWERS*R2)$
$R4(CNT) \leftarrow \mathcal{F}_{MAX\ CNT}(R3)$
$RESULT \leftarrow R4*R3$

Question 3 (18 pts):

Write the SQL queries of the following queries:

a. Find the users who did not pin any item in the system but created at least three combinations.
b. Find how many styles have been created by each user in the system, each one's most popular season, and the average prices of their created combinations.
c. Find the users who have been followed by at least two people who created all their combinations in "Fall/Winter" season.

Answer 3 (18 pts):

a.
```
(SELECT CB.BYUID
FROM USER U, CREATENEWCOMBINATION CB
WHERE U.UID=CB.BYUID AND
GROUP BY CB.BYUID
HAVING COUNT(* )>2)
MINUS
(SELECT BYUID
FROM PINNEDITEMS)
```

b.
```
(SELECT CB.BYUID, COUNT(*) AS CNT
FROM CREATENEWCOMBINATION CB
GROUP BY CB.BYUID)
JOIN
(SELECT CB.BYUID, AVG(TOTALPRICE) AS TP
FROM CREATENEWCOMBINATION CB
GROUP BY CB.BYUID)
JOIN
(SELECT CB.BYUID, MAX(OCCASION) AS MAXOCC
FROM CREATENEWCOMBINATION CB
GROUP BY CB.BYUID)
```

c.
```
SELECT K.UID
FROM USERFOLLOWERS K, M
WHERE K.FUID AND M.FUID IN
(SELECT US.UID
FROM USER US
WHERE NOT EXISTS (SELECT *
                  FROM CREATENEWCOMBINATION CB
                  WHERE CB.BYUID=US.UID
                  AND OCCASION<>"FALL/WINTER"))
AND K.FUID<>M.FUID
```

Question 4 (10 pts):

Using the above relations, show an example for minimal key, super key, foreign key and derived attribute.

Minimal key: SID (in CreateNewCombination)
Super key: SID, Season (in CreateNewCombination)
Foreign key: SID (in UserLikes)
Derived attribute: TotalPrice

Question 5 (9 pts):

Give examples about how entity integrity, insert and delete anomalies may occur in the case study. Give your reasons. Explain how you can avoid them.

Answer 5 (9 pts):

Recall from our course notes:
Entity integrity: The *primary key attributes* PK of each relation schema R in S **cannot** have NULL values in any tuple of r(R).

- t[PK] ≠ null for any tuple t in r(R)
- If PK has several attributes, null is not allowed in any of these attributes

- This is because primary key values are used to *identify* the individual tuples.
  - If two or more tuples had NULL for their PK, we may not be able to distinguish them. -> no null values to the PK

Referential integrity: A tuple in relation R1 that refers to another relation R2 must refer to an *existing tuple* in R2.

- Integrity constraints should not be violated by the update operations.
- Several update operations may have to be grouped together.

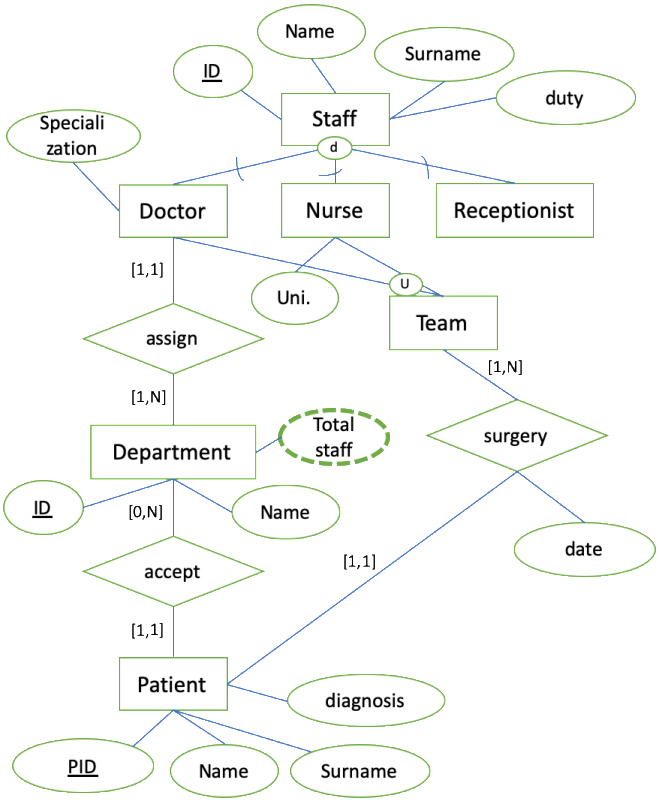
Updates may **propagate** to cause other updates automatically. This may be necessary to maintain integrity constraints.
You should implement how the process will affect other tables while creating your tables: setting null, cascading the effects or restrict.

Question 6 (20 pts):

Consider the following case study: In a hospital, there are doctors, nurses, receptionists and other staff. Patients are accepted to the departments of the hospital. A surgical operation includes doctors, nurses and a patient. Each doctor is assigned to a department, which has a unique department ID, name and the total number of staff (including everyone). Each doctor has a unique ID, name, surname and a specialization area. The system keeps tracks of the surgical operations i.e. when it happened, the team and the patient. Each patient has a patient ID, name, surname and diagnosis information. Nurses have a unique ID, name and graduated university with date. Staff information includes ID, name and duty. Note that all the hospital personnel have different ID.

Draw the EER diagram and its corresponding relational schema. ER solutions will not be accepted.

Answer 6 (20 pts):



The relational schema of the above is expected.