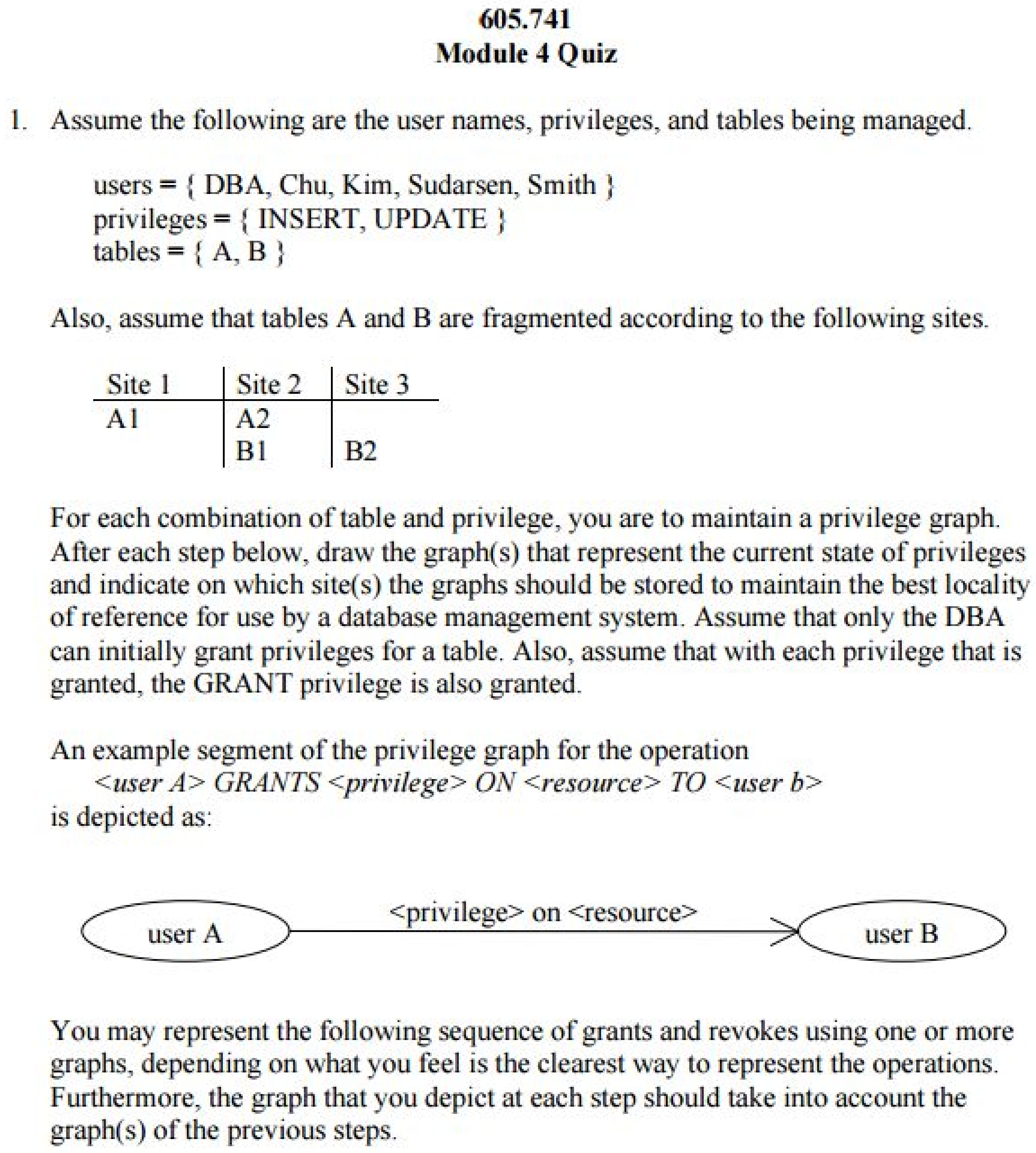
Xiang Xu 2/18/2017



605.741
Module 4 Quiz

1. Assume the following are the user names, privileges, and tables being managed.

users = { DBA, Chu, Kim, Sudarsen, Smith }
privileges = { INSERT, UPDATE }
tables = { A, B }

Also, assume that tables A and B are fragmented according to the following sites.

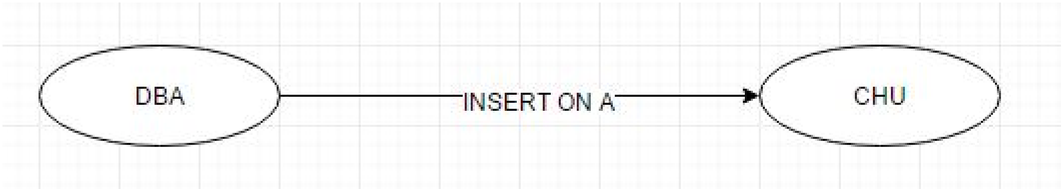
Site 1	Site 2	Site 3
A1	A2	
	B1	B2

For each combination of table and privilege, you are to maintain a privilege graph. After each step below, draw the graph(s) that represent the current state of privileges and indicate on which site(s) the graphs should be stored to maintain the best locality of reference for use by a database management system. Assume that only the DBA can initially grant privileges for a table. Also, assume that with each privilege that is granted, the GRANT privilege is also granted.

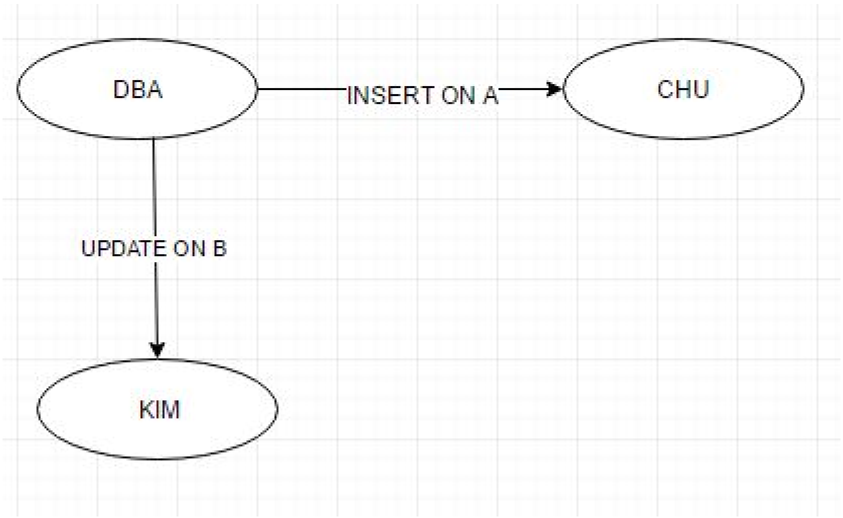
An example segment of the privilege graph for the operation
<user A> GRANTS <privilege> ON <resource> TO <user b>
is depicted as:

You may represent the following sequence of grants and revokes using one or more graphs, depending on what you feel is the clearest way to represent the operations. Furthermore, the graph that you depict at each step should take into account the graph(s) of the previous steps.

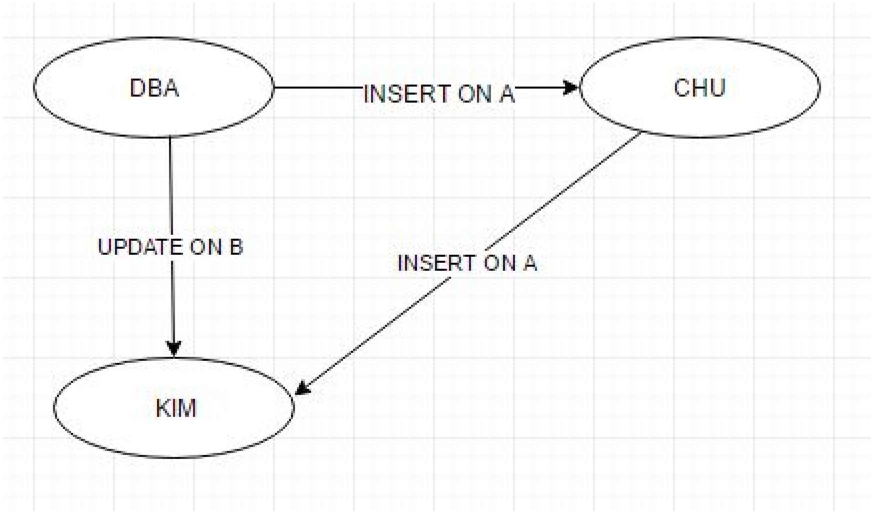
1) DBA issues: GRANT INSERT ON A TO CHU



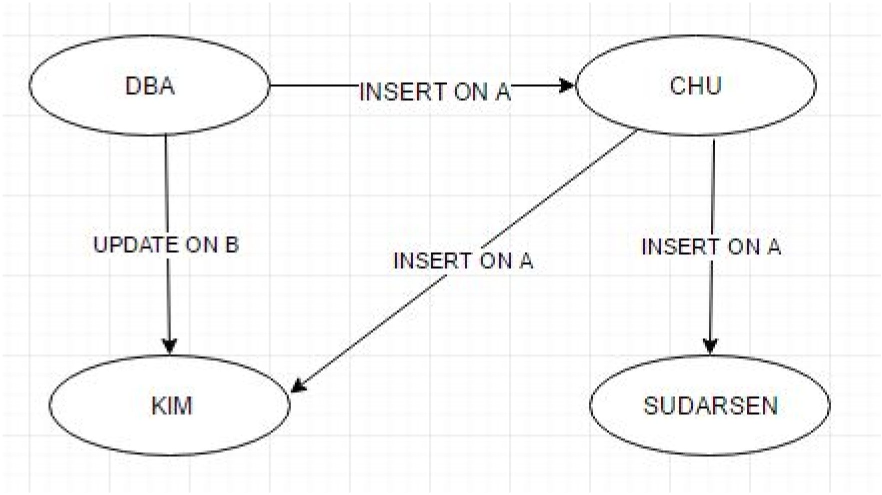
2) DBA issues: GRANT UPDATE ON B TO KIM



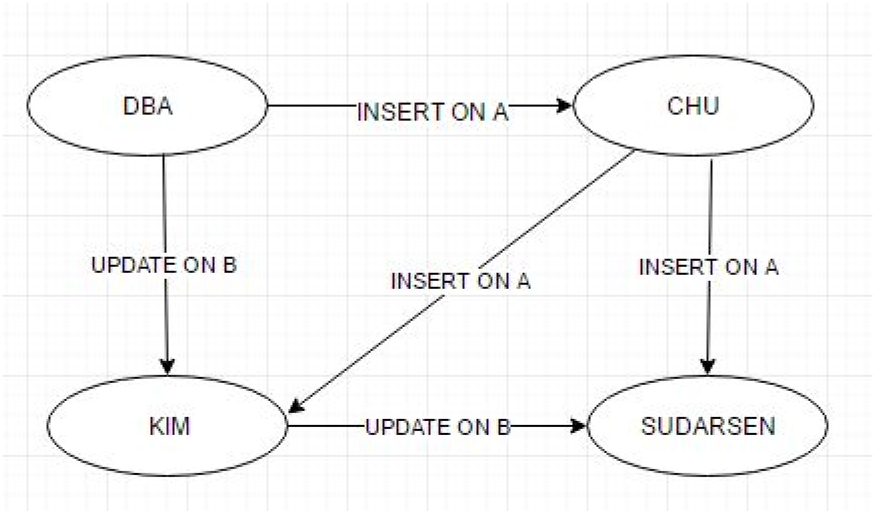
3) CHU issues: GRANT INSERT ON A TO KIM



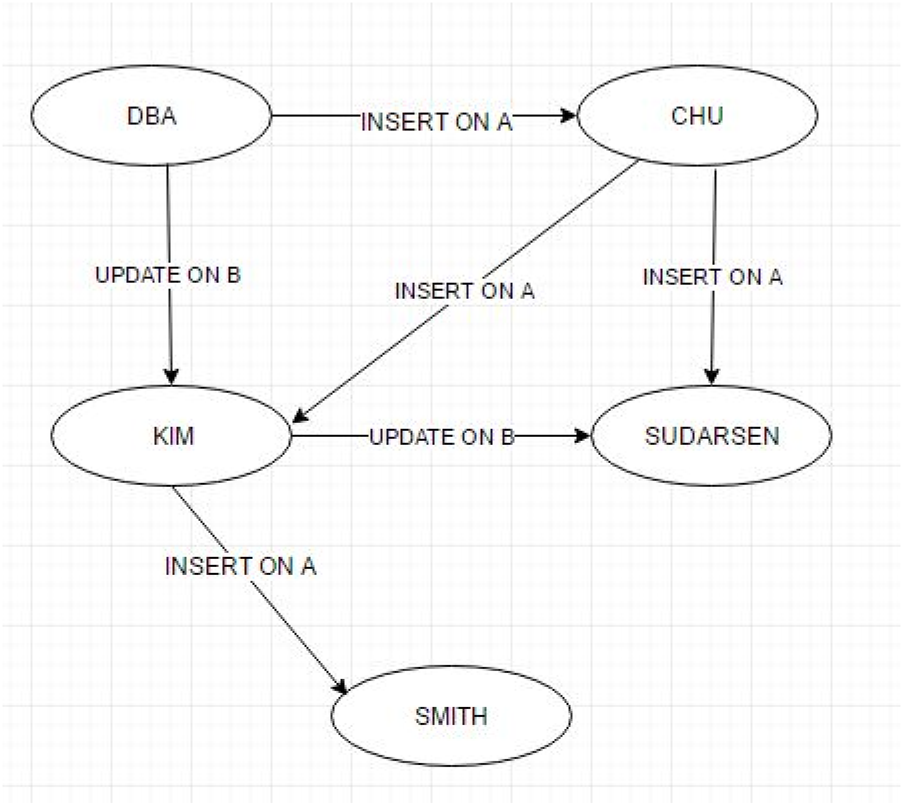
4) CHU issues: GRANT INSERT ON A TO SUDARSEN



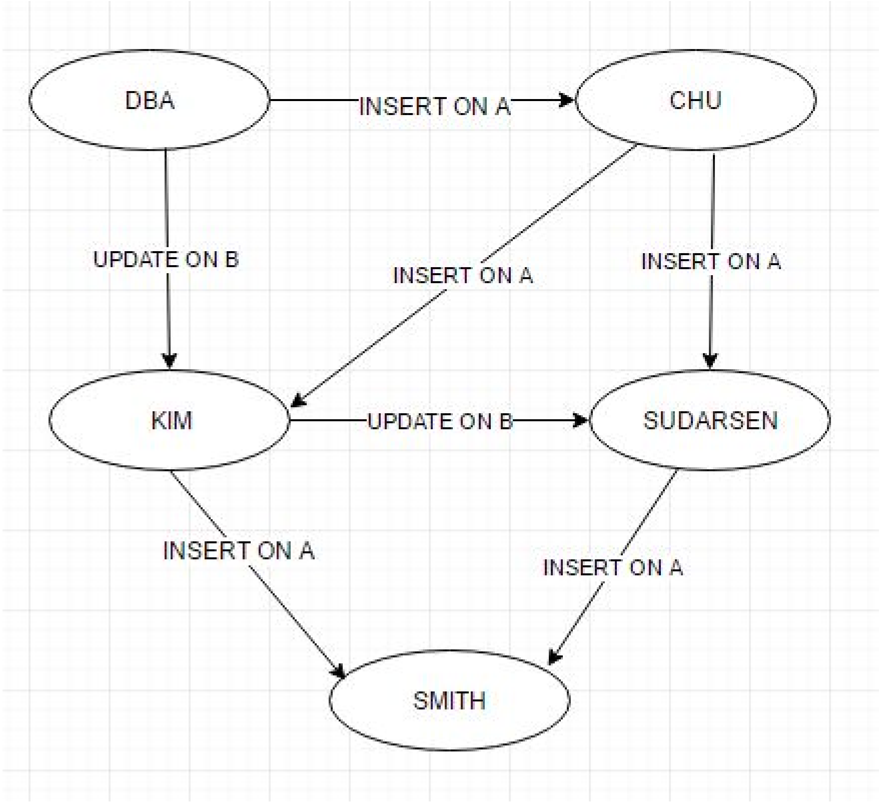
5) KIM issues: GRANT UPDATE ON B TO SUDARSEN



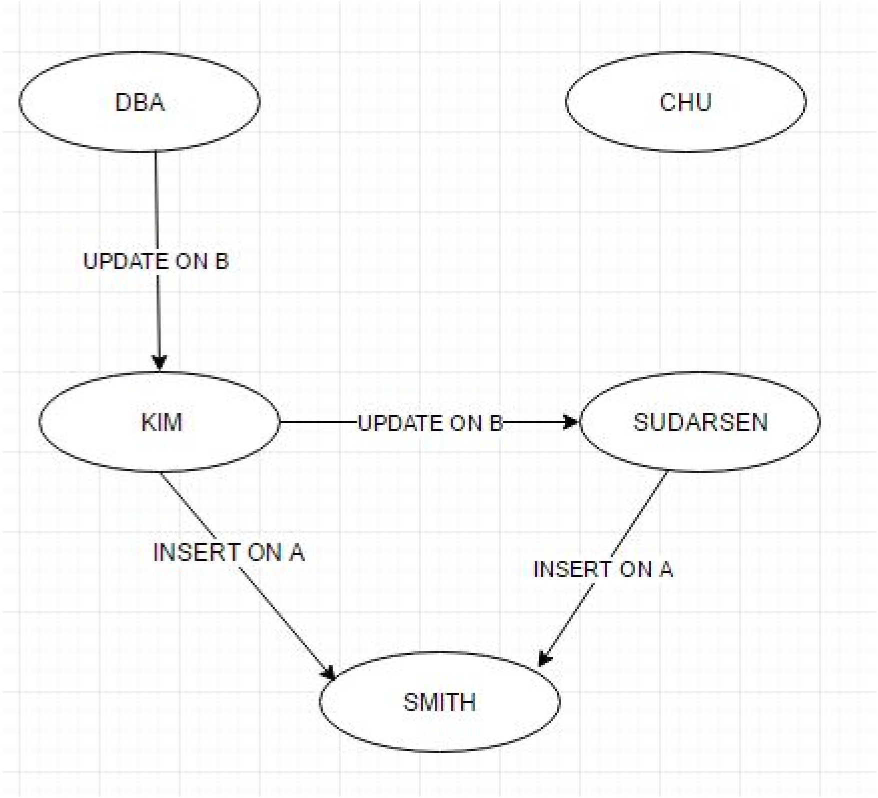
6) KIM issues: GRANT INSERT ON A TO SMITH



7) SUDARSEN issues: GRANT INSERT ON A TO SMITH

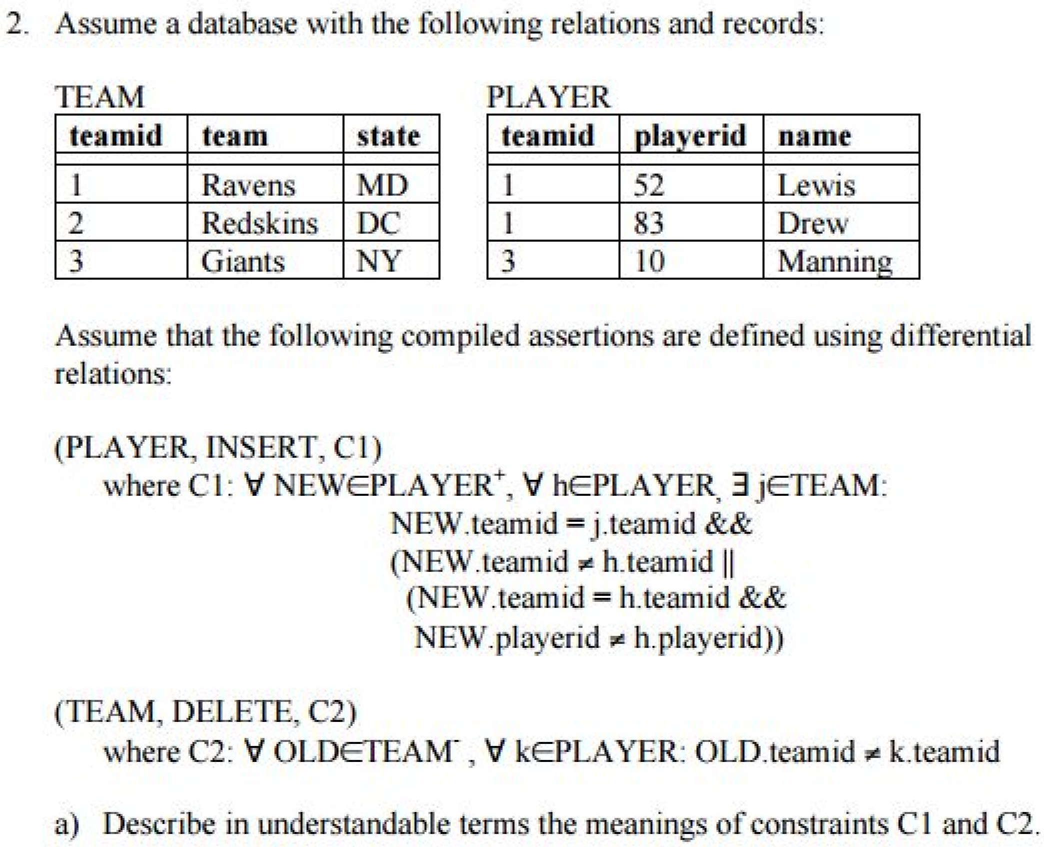


8) DBA issues: REVOKE INSERT ON A TO CHU



9) CHU issues: GRANT INSERT ON A TO SMITH

Because we assumed only DBA can initially grant privileges, thus, CHU can not grant privileges to SMITH, since he doesn’t have the permissions from DBA



2. Assume a database with the following relations and records:

TEAM

teamid	team	state
1	Ravens	MD
2	Redskins	DC
3	Giants	NY

PLAYER

teamid	playerid	name
1	52	Lewis
1	83	Drew
3	10	Manning

Assume that the following compiled assertions are defined using differential relations:

(PLAYER, INSERT, C1)

where C1: $\forall$ NEW$\in$PLAYER^{+}, $\forall$ h$\in$PLAYER, $\exists$ j$\in$TEAM:

$$\begin{aligned}&\text{NEW.teamid} = \text{j.teamid}\ \&\&\\&(\text{NEW.teamid} \neq \text{h.teamid}\ \|\\&\ (\text{NEW.teamid} = \text{h.teamid}\ \&\&\\&\ \ \text{NEW.playerid} \neq \text{h.playerid}))\end{aligned}$$

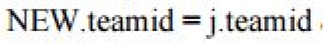
(TEAM, DELETE, C2)

where C2: $\forall$ OLD$\in$TEAM^{-} , $\forall$ k$\in$PLAYER: OLD.teamid $\neq$ k.teamid

a) Describe in understandable terms the meanings of constraints C1 and C2.

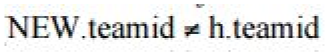
C1

For each new row is going to be inserted,



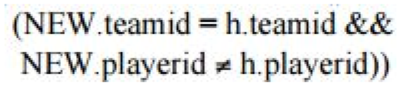
NEW.teamid = j.teamid

(1)We want to make sure the team is already exists in the TEAM relation, (we can find a corresponding teamid in TEAM relation),



NEW.teamid $\neq$ h.teamid

(2)Secondly, the new teamid is not exist at all in the PLAYER



(NEW.teamid = h.teamid &&

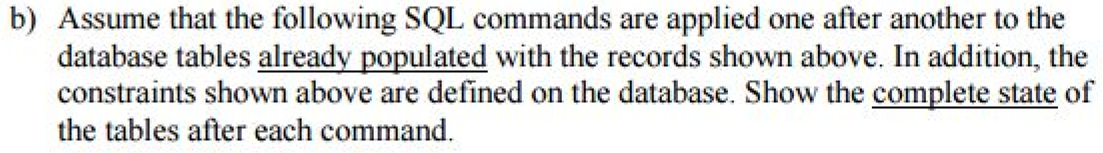
NEW.playerid $\neq$ h.playerid))

(3)if the player is already in a team, then we are not insert the player more than once.

(1) && ((2)||(3))

An insertion will perform when the new teamid can be found in TEAM relation first, then we want to check the new teamid is not exist at all in the PLAYER relation, if we find the new teamid exist in the PLAYER relation, then we want to make sure the playerid didn’t exist for that team in PLAYER relation.

C2
Before delete a teamid from TEAM relation, we want to make sure all related teamids are already deleted in PLAYER id



b) Assume that the following SQL commands are applied one after another to the database tables already populated with the records shown above. In addition, the constraints shown above are defined on the database. Show the complete state of the tables after each command.

1) INSERT INTO TEAM VALUES (4, ‘Steelers’, ‘PA’)

Team

teamid	team	state
1	Ravens	MD
2	Redskins	DC
3	Giants	NY
4	Steeler	PA

Player

teamid	playerid	name
1	52	Lewis
1	83	Drew
3	10	Manning

2) DELETE FROM TEAM WHERE teamid = 1

Team

teamid	team	state
1	Ravens	MD
2	Redskins	DC

3	Giants	NY
4	Steeler	PA

Player

teamid	playerid	name
1	52	Lewis
1	83	Drew
3	10	Manning

3) INSERT INTO PLAYER VALUES (2, 26, 'Portis')

Team

teamid	team	state
1	Ravens	MD
2	Redskins	DC
3	Giants	NY
4	Steeler	PA

Player

teamid	playerid	name
1	52	Lewis
1	83	Drew
3	10	Manning
2	26	Potris

4) DELETE FROM PLAYER WHERE playerid = 83

Team

teamid	team	state
1	Ravens	MD
2	Redskins	DC
3	Giants	NY
4	Steeler	PA

Player

teamid	playerid	name
1	52	Lewis
3	10	Manning
2	26	Potris

5) INSERT INTO PLAYER VALUES (2, 26, ‘Dockery’)

Team

teamid	team	state
1	Ravens	MD
2	Redskins	DC
3	Giants	NY
4	Steeler	PA

Player

teamid	playerid	name
1	52	Lewis
3	10	Manning
2	26	Potris