SGBD : BASES DE DONNÉES AVANCÉES [M3106C]

TD $N^0 5$ - NORMALISATION DE DONNÉES

Objectifs

– Règles de gestions et Dépendances Fonctionnelles
– Erreurs et Normalisation

Corrigés

Exercice I :

Question 1.1.

(1) clé primaire : deux lignes identiques
(2) dépendance `nom` ⟶ `prenom` : même nom associé à plusieurs prénoms différents.
(3) dépendance `nom matiere` ⟶ `salle` : même couple (nom, matière) associé à plusieurs salles différentes.

Question 1.2.

Lors deux mises à jour concurrentes (UPDATE 1,UPDATE 2) ou (DELETE 1, UPDATE 2) : UPDATE 2 est perdue.

Question 1.3.

```
 insert into ems1
   select distinct nom,prenom from ems;

insert into ems2
   select nom,matiere,salle from ems;
```

Question 1.4.

```
drop table ems;

create view ems as
       select e2.nom ,prenom,matiere,salle,e2.ctid
  from ems1 e1,ems2 e2
  where e1.nom=e2.nom;
```

Date: 30 septembre 2014.
Hocine ABIR - IUT Villetaneuse .

Question 1.5.

```
-- insert rules
CREATE RULE ins_ems2
  AS ON insert TO ems
  WHERE NOT EXISTS (SELECT * FROM ems2 WHERE NEW.nom=nom
             and NEW.matiere=matiere)
  DO INSTEAD
    INSERT INTO ems2 values(NEW.Nom, NEW.matiere, NEW.Salle);

CREATE RULE no_ins_ems2
  AS ON insert TO ems
   DO INSTEAD
     NOTHING;

CREATE RULE ins_ems1
  AS ON insert TO ems
  WHERE NOT EXISTS (SELECT * FROM ems1 WHERE NEW.nom=nom)
  DO INSTEAD
    INSERT INTO ems1 values(NEW.Nom, NEW.Prenom);

CREATE RULE no_ins_ems1
  AS ON insert TO ems
   DO INSTEAD
     NOTHING;
```

Question 1.6.

```
alter table ems2
   drop constraint ems2_nom_fkey;
alter table ems2
   ADD FOREIGN KEY(nom) REFERENCES EMS1(nom)
   ON DELETE CASCADE
   DEFERRABLE INITIALLY DEFERRED;

-- delete rules
CREATE or replace RULE del_ems1
  AS ON delete TO ems
  WHERE (SELECT count(*)=1 FROM ems2 WHERE OLD.nom=nom)
  DO INSTEAD
      delete from ems1 where nom=OLD.nom;

 CREATE or replace RULE del_ems2
  AS ON delete TO ems
  DO INSTEAD
      delete from ems2 where nom=OLD.nom
```

```
        and    matiere=old.matiere;
```

Question 1.7.

```
-- update rules
-- ems1 avant ems2 : sinon ctid disparait
-- de la vue
CREATE or replace RULE upd_ems
  AS ON update TO ems
  WHERE  NEW.nom=OLD.nom
     and NEW.matiere=OLD.matiere
  DO INSTEAD
  (
    UPDATE ems1 set prenom=NEW.prenom
        WHERE nom=OLD.Nom;
    UPDATE ems2 set salle=NEW.salle
        WHERE nom=OLD.Nom and matiere=OLD.matiere;
  );

-- insert avant delete
CREATE or replace RULE updr_ems
  AS ON update TO ems
  DO INSTEAD
  (
     INSERT INTO ems (nom,prenom,matiere,salle)
         values(NEW.nom,NEW.prenom,NEW.matiere,NEW.salle);
     DELETE FROM ems
        WHERE nom=OLD.nom and matiere=OLD.matiere;
  );
```

Question 1.8.

```
grant select on ems to public;
alter function  ems_ins(varchar,varchar,varchar,varchar) SECURITY DEFINER
alter function  ems_del(tid) SECURITY DEFINER
alter function  ems_upd(varchar,varchar,varchar,varchar,tid) SECURITY DEFINER
grant execute on function ems_ins(varchar,varchar,varchar,varchar) to public;
grant execute on function ems_del(tid) to public;
grant execute on function ems_upd(varchar,varchar,varchar,varchar,tid) to pub
```

Question 1.9.

```
alter table ems2
   add column owner name;
```

```
-- Modifier les fonctions !! OU les règles
create or replace function ems_ins(varchar,varchar,varchar,varchar)
returns void as $$
  insert into ems (nom,prenom,matiere,salle,owner)
             values(case when ($1!=’’ and upper($1)!=’NULL’) then $1 end,
           case when ($2!=’’ and upper($2)!=’NULL’) then $2 end,
  case when ($3!=’’ and upper($3)!=’NULL’) then $3 end,
     case when ($4!=’’ and upper($4)!=’NULL’) then $4 end,
                current_user);
$$ language SQL;

create or replace function ems_del(tid)
returns void as $$
  delete from ems where ctid=$1
     and owner=current_user;
$$ language SQL;

create or replace function ems_upd(varchar,varchar,varchar,varchar,tid)
returns void as $$
  update ems set
         nom=case when ($1!=’’ and upper($1)!=’NULL’) then $1 end,
         prenom=case when ($2!=’’ and upper($2)!=’NULL’) then $2 end,
   matiere=case when ($3!=’’ and upper($3)!=’NULL’) then $3 end,
      salle= case when ($4!=’’ and upper($4)!=’NULL’) then $4 end
  where ctid= $5 and owner=current_user;
$$ language SQL;
```