

Biraveen
SIVAHARAN

Compte rendu TD12 SLAM 3

Exercice n°1 :

Schéma relationnel :

Vehicule (id,nom,idContrat#)

contrat(id,nom)

sinistre(id,nom)

Impliquer(idVehicule#,idSinistre#)

Création des tables :

1/ create table vehicule (id integer PRIMARY KEY,
marque text,
idContrat integer references contrat(id));

2/ create table contrat (id integer PRIMARY KEY,
nom text);

3/ create table sinistre(id integer PRIMARY KEY,
nom text);

4/ create table impliquer(id integer PRIMARY KEY,
idVehicule integer references vehicule(id),
idSinistre integer references sinistre(id));

Insertion des données

```
INSERT INTO contrat VALUES ( '1', 'Tout-risque');  
INSERT INTO vehicule VALUES ( '1', 'Mercedes', '1');  
INSERT INTO impliquer VALUES ( '1', '1', '1');
```

```
INSERT INTO vehicule VALUES ( '2', 'BMW');  
INSERT INTO contrat VALUES ( '2', 'tiers');  
UPDATE vehicule SET idContrat = '2';
```

Procédure stockée / Création de la fonction :

```
create function verif()returns trigger as $$
declare
assure integer ;
begin
Select idContact into assure from vehicule where id=NEW.idVehicule;
if assure = null then
raise exception 'Le véhicule est pas assuré';
end if;
return NEW;
end;
$$LANGUAGE plpGSQL;
```

```
postgres=# create function verif()returns trigger as $$
postgres$# declare
postgres$# assure integer ;
postgres$# begin
postgres$# Select idContact into assure from vehicule where id=NEW.idVehicule;
postgres$# if assure = null then
postgres$# raise exception 'Le véhicule est pas assuré';
postgres$# end if;
postgres$# return NEW;
postgres$# end;
postgres$# $$LANGUAGE plpGSQL;
CREATE FUNCTION
```

Création du TRIGGER :

Create TRIGGER vehicule1 BEFORE INSERT on impliquer
FOR EACH ROW EXECUTE PROCEDURE verif());

```
postgres=#
postgres=#
postgres=# Create TRIGGER vehicule1 BEFORE INSERT on impliquer
postgres=# FOR EACH ROW EXECUTE PROCEDURE verif());
CREATE TRIGGER
postgres=#
```

Exercice n°2 :

Schéma relationnel :

Professeur ("id", nom, prenom);

Matiere("id",libelle);

Classe ("id",numSalle, nbrEleves, etage);

Enseigner ("id",idMatiere#,idSalle#,idProfesseur#);

Création des tables :

```
create table Professeur ( id integer PRIMARY KEY,  
nom text,  
prenom text,  
);
```

```
create table Matiere (id integer PRIMARY KEY,  
libelle text);
```

```
create table Classe ( id integer PRIMARY KEY,  
nbrEleves integer,  
etages integer);
```

```
create table Enseigner ( id integer PRIMARY KEY,  
idMatiere integer references matiere(id),  
idProfesseur integer references professeur(id));
```

```
create table qualifie( id integer PRIMARY KEY,  
idProfesseur integer references professeur(id),  
idMatiere integer references matiere(id)  
);
```

Insertion des données

```
INSERT INTO Professeur VALUES ('1','Monsieur DUPONT');  
INSERT INTO Matiere VALUES ('1','technologie');  
INSERT INTO Matiere VALUES ('2','technologie');  
INSERT INTO classe VALUES ('1','23','2');  
INSERT INTO Enseigner VALUES ('1','1','1');  
INSERT INTO Enseigner VALUES ('2','2','1');
```

