

Programmation et Administration de Bases de données --- TP surveillé 2016

Administration

1- Créer trois utilisateurs resp, ut1, et ut2 avec leur login comme mot de passer, et créer une base de données portant aussi leur nom.

```
CREATE USER resp WITH PASSWORD 'resp';
CREATE DATABASE resp;
CREATE USER ut1 WITH PASSWORD 'ut1';
CREATE USER ut2 WITH PASSWORD 'ut2';
CREATE DATABASE ut1;
CREATE DATABASE ut2;
```

2- En tant que resp, créer la table HeuresTravaillees ayant dans l'ordre, un attribut date de type DATE, employeLogin de type VARCHAR(20), et heures de type INT.

S'assurer que que :

- date est non nulle.
- le couple (date, employeLogin) une clé primaire.
- heures est toujours strictement positif.

```
psql -U resp -h localhost -p 5433
```

```
CREATE TABLE HeuresTravaillees (
date DATE NOT NULL,
employeLogin VARCHAR(20),
heures INT,
CONSTRAINT cons_chk_heures CHECK (heures > 0),
CONSTRAINT cons_pk PRIMARY KEY (date,employeLogin));
```

3- On veut donner un moyen aux utilisateurs de consulter leur propres heures travaillées, et l'intervalle de temps sur lequel ils ont travaillé.

Toujours en tant que resp, créer une vue HTView(ht, entre, et) de HeuresTravaillees avec dans l'ordre comme attributs, ht étant la somme des heures travaillées par l'utilisateur courant, entre étant la plus ancienne date où l'utilisateur a travaillé, et et étant la date la plus récente.

```

CREATE VIEW HTView AS
(SELECT sum(heures) AS ht, min(date) AS entre, max(date) AS et
FROM HeuresTravaillees
WHERE employeLogin = CURRENT_USER
);

```

2bis- Ajouter des lignes dans HeuresTravaillees. Par exemple,

```

INSERT INTO HeuresTravaillees VALUES ('2012-03-21', 'ut1', 5);
INSERT INTO HeuresTravaillees VALUES ('2012-04-12', 'ut1', 8);
INSERT INTO HeuresTravaillees VALUES ('2012-12-12', 'ut2', 6);
INSERT INTO HeuresTravaillees VALUES (NULL, 'ut3', 6);
INSERT INTO HeuresTravaillees VALUES ('2012-06-21', 'ut3', -5);
INSERT INTO HeuresTravaillees VALUES ('2012-03-21', 'ut2', 4);
INSERT INTO HeuresTravaillees VALUES ('2012-12-12', 'ut2', 3);
INSERT INTO HeuresTravaillees VALUES ('2013-01-09', 'ut1', 4);

```

```

SELECT * FROM HeuresTravaillees;
   date      | employeLogin | heures
-----+-----+-----
 2012-03-21 | ut1          |      5
 2012-04-12 | ut1          |      8
 2012-12-12 | ut2          |      6
 2012-03-21 | ut2          |      4
 2013-01-09 | ut1          |      4

```

4- Donner les permissions nécessaires et suffisantes à ut1 et ut2, de telle sorte que l'on obtienne les résultats suivants sur la base de données resp :

En tant que ut1 :

```

resp=> SELECT * FROM HTView;
 ht | entre | et
----+-----+-----
 17 | 2012-03-21 | 2013-01-09

```

Et en tant que ut2 :

```

resp=> SELECT * FROM HTView;
 ht | entre | et
----+-----+-----
 10 | 2012-03-21 | 2012-12-12

```

```

GRANT SELECT ON HTView TO ut1;
GRANT SELECT ON HTView TO ut2;

```

5- Créer un utilisateur respassist qui assiste le responsable et qui aura le droit d'insérer et supprimer des lignes dans HeuresTravaillees.

En tant que "root" :

```
CREATE USER respassist WITH PASSWORD 'respassist';  
CREATE DATABASE respassist;
```

En tant que resp :

```
GRANT INSERT, DELETE ON HeuresTravailles TO respassist;
```

La suppression ne devrait pas fonctionner en général.

Contraintes

Epoux(epoux,epouse)



Epouse(epouse,epoux)

Ecrire en détail le code SQL pour :

- créer les relations Epoux et Epouse
- Insérer Epoux(45,13), Epouse(13,45)

```
CREATE TABLE Epoux ( epoux INTEGER PRIMARY KEY, epouse INTEGER);  
CREATE TABLE Epouse ( epouse INTEGER PRIMARY KEY, epoux INTEGER  
REFERENCES Epoux);
```

```
ALTER TABLE Epoux ADD FOREIGN KEY (epouse) REFERENCES Epouse  
(epouse) DEFERRABLE INITIALLY DEFERRED;
```

```
START TRANSACTION;  
INSERT INTO Epoux VALUES (45,13);  
INSERT INTO Epouse VALUES (13,45);  
COMMIT
```

Programmation

1- Créer les tables Compagnie et Pilote :

```
CREATE TYPE typeCie AS ENUM ('A', 'B');
```

```
CREATE TABLE Compagnie  
(compagnieCode VARCHAR(20) PRIMARY KEY,  
type typeCie NOT NULL,  
nomComp VARCHAR(20) NOT NULL);
```

```
CREATE TABLE Pilote  
(brevet VARCHAR(20) PRIMARY KEY,  
nom VARCHAR(20) NOT NULL,  
nbHVol NUMERIC,  
compagnie VARCHAR(3),  
grade VARCHAR(5),  
FOREIGN KEY (compagnie) REFERENCES Compagnie(compagnieCode));
```

2- On va ajouter des contraintes à respecter au moment de l'insertion d'un nouveau pilote. (Rappel : les contraintes CHECK ne sont pas supportées avec MySQL.) Le grade CDB signifie commandant de bord, COPI copilote et INST instructeur.

Toute erreur d'insertion doit signaler l'erreur 'Echec de contrainte CHECK sur Pilote.grade'.

a) C1. Faire en sorte de refuser d'insérer un pilote qui n'a pas un grade 'CDB', 'COPI', ou 'INST'.

b) C2. Ajouter à votre solution le cas particulier d'une **compagnie de type B** : Si on essaye d'insérer un CDB avec moins de 1500h, un COPI avec moins de 500h de vol ou si on essaye d'insérer un INST (quelque soit le nombre d'heures de vol), on refuse l'insertion. On laisse faire sinon.

c) C3. Ajouter à votre solution le cas particulier d'une **compagnie de type A** : Si on essaye d'insérer un CDB avec moins de 500h de vol, on refuse l'insertion. Si on essaye d'insérer un CDB avec au moins 500h de vol mais moins de 1500h, alors le pilote doit être inséré comme COPI. Si on essaye d'insérer un COPI avec moins de 500h de vol ou un INST avec moins de 3000h de vol, on refuse l'insertion. On laisse faire sinon.

```

CREATE OR REPLACE FUNCTION process_InsPilote() RETURNS TRIGGER AS
$TrigBefInsPilote$
BEGIN
    IF (NEW.grade NOT IN ('CDB', 'COPI', 'INST'))
        THEN
            RAISE EXCEPTION 'Echec de contrainte CHECK sur Pilote.grade';
        END IF;

    IF (SELECT type FROM Compagnie WHERE compagnieCode = NEW.compagnie) = 'B'
        THEN
            IF ((NEW.grade = 'CDB' AND (NEW.nbHVol <1500)) OR
                (NEW.grade = 'COPI' AND (NEW.nbHVol <500)) OR
                (NEW.grade = 'INST'))
                THEN
                    RAISE EXCEPTION 'Echec de contrainte CHECK sur Pilote.grade';
                END IF;
        ELSE IF (SELECT type FROM Compagnie WHERE compagnieCode = NEW.compagnie) =
            'A'
            THEN
                IF ((NEW.grade = 'CDB' AND (NEW.nbHVol <500)) OR
                    (NEW.grade = 'COPI' AND (NEW.nbHVol <500)) OR
                    (NEW.grade = 'INST' AND (NEW.nbHVol <3000)))
                    THEN
                        RAISE EXCEPTION 'Echec de contrainte CHECK sur Pilote.grade';
                    END IF;
                IF (NEW.grade = 'CDB' AND (NEW.nbHVol <1500) AND (NEW.nbHVol >=500))
                    THEN
                        SET NEW.grade = 'COPI';
                    END IF;
            END IF;
        END IF;
END IF;
END;

$TrigBefInsPilote$ LANGUAGE plpgsql;

CREATE TRIGGER TrigBefInsPilote
BEFORE INSERT ON Pilote
FOR EACH ROW
EXECUTE PROCEDURE process_InsPilote();

```

Par exemple, on s'attendra à ces résultats.

```

INSERT INTO Compagnie VALUES ('AF', 'A', 'Air France');
INSERT INTO Compagnie VALUES ('ASE', 'B', 'Senart Club');
INSERT INTO Compagnie VALUES ('EM', 'A', 'Emirates');

INSERT INTO Pilote VALUES ('BR-1', 'Daniel Campo', 5200, 'AF', 'CDB');
INSERT INTO Pilote VALUES ('BR-2', 'Elisa Tovalisa', 700, 'AF', 'COPI');
INSERT INTO Pilote VALUES ('BR-3', 'Pierre Filou', 9000, 'AF', 'INST');
INSERT INTO Pilote VALUES ('BR-4', 'Philippe Reigner', 1000, 'AF', 'COPI');

```

```

INSERT INTO Pilote VALUES ('BR-5','Oscar Del',1200,'ASE','CDB');
INSERT INTO Pilote VALUES ('BR-6','Minimo Corti', 1200,'EM','CDB');
INSERT INTO Pilote VALUES ('BR-7','Inigo Stru', 2999,'EM','INST');
INSERT INTO Pilote VALUES ('BR-8','Grade Fake', 2999,'EM','UJF');
INSERT INTO Pilote VALUES ('BR-9','Pico Pepito', 45,'ASE','COPI');
INSERT INTO Pilote VALUES ('BR-10','Instro Inutilo', 9000,'ASE','INST');
INSERT INTO Pilote VALUES ('BR-10','Instro Inutilo', 9000,'ASE','COPI');
INSERT INTO Pilote VALUES ('BR-11','Senarttois Bien', 5700,'ASE','CDB');

```

```

testsab=> INSERT INTO Pilote VALUES ('BR-19','Senarttois Bieng', 0,'ASE','CDB');
ERREUR: Echec de contrainte CHECK sur Pilote.grade

```

```

SELECT * FROM Pilote;

```

```

+-----+-----+-----+-----+-----+
| brevet | nom           | nbHVol | compagnie | grade |
+-----+-----+-----+-----+-----+
| BR-1   | Daniel Campo  | 5200   | AF        | CDB   |
| BR-10  | Instro Inutilo | 9000   | ASE       | COPI  |
| BR-11  | Senarttois Bien | 5700   | ASE       | CDB   |
| BR-2   | Elisa Tovalisa | 700    | AF        | COPI  |
| BR-3   | Pierre Filou   | 9000   | AF        | INST  |
| BR-4   | Philippe Reigner | 1000   | AF        | COPI  |
| BR-6   | Minimo Corti  | 1200   | EM        | COPI  |
+-----+-----+-----+-----+-----+

```