

1 TD

On veut créer une base de données servant à faire l'emploi du temps d'une université. Le service concerné a envoyé aux informaticiens de l'université le schéma suivant pour qu'il soit implémenté.

ELEVE(Mid,Nom,Prénom,Cid)

COURS(Cid,DateDebut,DateFin,HoraireDebut,HoraireFin,Durée,Pnom,Pprénom,Pgrade,PUFR,Groupe,UFR,responsable,année,cursus,matière,semestre, NumSalle,batiment,equipementSalle)

où Duree est la durée horaire du cours, par séance; Pnom,Pprenom sont les noms et prénoms du professeur, Pgrade son grade (maître de conférence, professeur, moniteur, ATER), PUFR l'UFR qui l'emploi; UFR est l'*unité de recherche et de formation* de la matière. Par exemple voici quatre tuples de la table COURS :

(1,12-01-2015,17-04-2015,13h30,14h30,1h,'de Ruyg-Altherre','Nicolas','ATER',1,'Mathématiques','Mathématiques','Boucheron',2015,'ISIFAR','BDD',2,2004,'SG','Linux')

(1,12-01-2015,17-04-2015,14h30,15h30,2h,'de Ruyg-Altherre','Nicolas','ATER',1,'Mathématiques','Mathématiques','Boucheron',2015,'ISIFAR','BDD',2,2016,'SG','tableau noir, projecteur')

(1,12-01-2015,17-04-2015,13h00,15h00,2h,'Capelli','Florent','Moniteur',2,'Informatique','Mathématiques','Boucheron',2015,'L3 MIASS','BDD',2,2015,'SG','Tableau noir')

(1,12-01-2015,17-04-2015,15h00,16h00,2h,'Capelli','Florent','Moniteur',2,'Informatique','Mathématiques','Boucheron',2015,'L3 MIASS','BDD',2,2005,'SG','Mac')

1. Cette base de données est mal faite. Dites pourquoi en donnant des exemples.

Solution: Il y a des répétitions (prof par exemple), des problèmes de mises à jour (si la date de début change on doit la changer plusieurs fois), des redondances (durée peut se calculer) et équipement ne permet qu'une entrée, donc difficile à exploiter.

2. Proposer un schéma qui efface ces problèmes. Précisez bien les clefs.

Solution: Beaucoup de solutions peuvent être proposées. Par exemple ;

```

ELEVE(Mid, Nom, Prénom)
INSCRIT(Mid,Cid)
COURS(Cid,HoraireDebut,HoraireFin,Durée,salle,prof,matiere)
SALLE(NumSalle,equipementSalle)
PROF(Pnom,Pprénom,Pgrade,PUFR)
MATIERE(dateDebut,dateFin,UFR,année,cursus,matière,semestre)

```

mais des choses peuvent encore être factorisées : UFR, grade, cursus, équipement salle etc.

3. Écrivez ce schéma avec des pattes de corbeaux. Mettez bien les contraintes de cardinaux.
4. Proposer des dépendances fonctionnelles pour ce nouveau schéma.

Solution: Beaucoup de solutions peuvent être proposées. Par exemple ;

```

HoraireDebut,HoraireFin -> durée
Pnom,Pprenom->Pgrade,PUFR
matière,année,cursus>responsable,UFR,semestre
cursus->UFR

```

```

matière,année,UFR,cursus->semestre,responsable
salle,batiment->équipement
groupe,matière,année -> Prof
année,salle,horaire->Prof,matière
DateDebut->année
Datefin->année

```

2 TP

2.1 Révisions

1. Donnez la liste des tournois dont aucun match s'est fini par un retrait.

Solution:

```

(SELECT tid FROM atp.tournament_big)
EXCEPT
(SELECT tid FROM atp.game_big NATURAL JOIN
      atp.retired_match_big);

```

2. Donnez la liste des tournois où au moins deux matchs se sont finis par des retraits.

Solution:

```

SELECT tid FROM atp.game_big NATURAL JOIN
      atp.retired_match_big
GROUP BY tid
HAVING COUNT(mid) > 1;

```

2.2 Créer, insérer, modifier

Dans les deux tableaux ci-dessous, nous rappelons les différentes commandes pour manipuler les données et la structure des tables.

Commande	Description
INSERT INTO t VALUES ('v1',v2'...)	Insère dans la table t la ligne indiquée, dans l'ordre des colonnes.
INSERT INTO t VALUES (col1='v1', col2='v2' ...)	Idem, où l'on a nommé explicitement les colonnes.
INSERT INTO t (col1, col2 ...) Req	Insère dans la table t le résultat de la requête Req. Il faut bien entendu que les colonnes de Req correspondent à (col1, col2 ...).
DELETE FROM t WHERE condition	Supprime toutes les lignes de la table t qui satisfont la condition.
UPDATE t SET col1='v1', col2='v2' ... WHERE condition	Modifie les colonnes comme indiqué pour chaque ligne qui satisfait la condition.

Commande	Description
CREATE TABLE t (col1 type1 contrainte 1, col2 typ2 contrainte2...)	Crée la table t avec les colonnes col1, col2 qui sont de type type1, type2 avec leurs contraintes.
CREATE TABLE t AS (requête)	Crée la table t et la remplit selon la requête (un peu comme lorsque vous créez une vue, sauf que là les données sont vraiment copiées).
DROP TABLE t	Supprime la table t.
ALTER TABLE t DROP COLUMN col	Supprime la colonne col de la table t.
ALTER TABLE t ALTER COLUMN col type contrainte	Modifie la colonne col de la table t en changeant son type et sa contrainte.

2.3 Un exercice

On voudrait améliorer un peu la gestion des tournois dans la base de données ATP. Dans sa version actuelle, la table `tournament` fait la distinction entre le French Open de 2006 et celui de 2009, même si ces deux tournois sont simplement deux éditions différentes du même tournoi. On va découper la table `tournament` en deux tables : une table `tournaments` qui contiendra la liste des différents tournois ATP et une table `tournament_editions` qui contiendra les informations sur ces tournois qui changent chaque année. Chacune de ces tables auront une clé primaire : `tournament_id` pour la table `tournaments` et `tid` pour `tournament_editions`.

1. Pour chaque colonne de la table `tournament`, décidez si elle doit faire partie de la table `tournaments` ou `tournament_editions`.

	tournaments	tournament_editions
Solution: tid		✓
name	✓	
location	✓	
startdate		✓
enddate		✓
numrounds	✓	
ttype	✓	
surface	✓	
outlier	✓	

2. Quelles sont les nouvelles colonnes des tables `tournaments` et `tournament_editions`? Quelles contraintes a-t-on sur ces colonnes?

Solution: Les deux tables ont une nouvelle colonne `tournament_id`. C'est une clé primaire pour `tournaments` et on a une contrainte de référence : `tournament_editions.tournament_id` fait référence à `tournaments.tournament_id`.

3. On va implémenter ces deux tables en SQL. Vous n'avez les droits d'écriture que dans votre schéma donc tout se fera là.

(a) Commencez par copier deux fois la structure de la table `atp.tournament` dans votre schéma avec les commandes

```
CREATE TABLE entid.tournament_edition AS
    SELECT * FROM atp.tournament_big WHERE FALSE ;
CREATE TABLE entid.tournaments AS
    SELECT * FROM atp.tournament_big WHERE FALSE ;
```

(b) En ajoutant/supprimant des colonnes, retrouvez la structure décrite à la question précédente. Ajoutez les contraintes nécessaires.

Solution:

```
ALTER TABLE entid.tournaments DROP COLUMN startdate ;
ALTER TABLE entid.tournaments DROP COLUMN enddate ;
ALTER TABLE entid.tournaments DROP COLUMN tid ;
ALTER TABLE entid.tournaments ADD COLUMN tournament_id SERIAL ;
ALTER TABLE entid.tournaments ADD PRIMARY KEY (tournament_id) ;

ALTER TABLE entid.tournament_edition DROP COLUMN ttype ;
ALTER TABLE entid.tournament_edition DROP COLUMN name;
ALTER TABLE entid.tournament_edition DROP COLUMN surface;
ALTER TABLE entid.tournament_edition DROP COLUMN numrounds;
ALTER TABLE entid.tournament_edition DROP COLUMN location;
ALTER TABLE entid.tournament_edition ADD COLUMN tournament_id
    integer REFERENCES tournaments.tournament_id ;
ALTER TABLE entid.tournament_edition ADD CONSTRAINT
    PRIMARY KEY (tid) ;
```

4. Importez les données de la base ATP dans vos deux tables.

Solution:

```
INSERT INTO public.tournaments (name,location,numrounds,ttype,surface)
SELECT DISTINCT ON(name,location)
    name, location, numrounds, ttype,surface
FROM atp.tournament ;

INSERT INTO public.tournament_edition
(at.tid, tournament_id, startdate, enddate)
SELECT tid,tournament_id,startdate,enddate
FROM atp.tournament at JOIN public.tournaments pt
ON(at.name=pt.name AND at.location=PT.location) ;
```