

# Exercices corrigés (1)

## A- Modélisation

Une compagnie d'assurance veut utiliser un SGBD pour stocker ses contrats d'assurances de voiture. Une police d'assurance est souscrite par une seule personne mais peut concerner plusieurs véhicules. Chaque véhicule doit avoir un conducteur principal qui peut être différent de l'assuré lui-même. Chaque véhicule peut être assuré sous un régime particulier (tous risques, tiers...). Le coefficient de bonus est propre au véhicule pour une police d'assurance donnée. Une personne ne peut pas être conducteur principal de plus d'une voiture pour une même police d'assurance.

On propose la relation universelle suivante.

**Assurance** (NumAssurance, NumPersonne, Nom, Prenom, Adresse, NumAssuré, NumCond, NumImmat, TypeAss, Bonus)

où NumAss est le numéro de la police d'assurance; NumPersonne, Nom, Prénom et Adresse les coordonnées de toutes les personnes connues par la compagnie, assurés ou conducteurs; NumAssuré et NumCond sont respectivement les numéros des personnes en tant qu'Assuré ou en tant que conducteur; NumImmat le numéro d'immatriculation d'une voiture; TypeAss et Bonus le type d'assurance et le bonus du véhicule pour une police particulière.

Dans un premier temps, nous supposons que les numéros de conducteur et d'assurés correspondent de façon univoque au numéro de personne (et inversement) mais qu'ils peuvent avoir des valeurs différentes.

- Donner la liste des dépendances fonctionnelles en les validant par les hypothèses de l'énoncé ou par des hypothèses supplémentaires que vous ne manquerez pas de préciser. En particulier, vous vous interrogerez sur les points suivants, à savoir si un assuré peut ou non souscrire plusieurs polices d'assurances différentes, si une voiture peut ou non être assurée plusieurs fois sous la même police d'assurance, sous des polices d'assurance différentes, etc.
- Donner une clef de la relation Assurance.
- Proposer une décomposition en 3 FN, sans perte et qui préserve les DFs.
- On suppose maintenant que les personnes ont des numéros identiques en tant que personne, assuré ou conducteur. Comment pouvez vous alors simplifier votre schéma. Auriez vous obtenu la même décomposition en supprimant les attributs NumCond et NumAssuré de la relation universelle. Donnez une explication.

## B- requêtes relationnelles

Pour avoir le droit d'accès à une machine Unix de l'école, il faut être individuellement déclaré comme ayant un droit d'accès à cette machine, ou bien appartenir à un groupe d'utilisateurs (dit net-group) qui a lui même accès à la machine. Bien entendu, ces possibilités ne sont pas exclusives l'une de l'autre.

Les informations concernant ces droits d'accès sont stockées dans le schéma relationnel suivant:

**host** (hostid, hostname)

**user** (userid, login, name, firstname)

**netgroup** (netgrpid, userid)

**Accessgroup** (netgrpid, hostid)

**Accessuser** (userid, hostid)

## B1- Requêtes relationnelles

Répondre en SQL aux questions suivantes :

- Quels sont les utilisateurs qui ont accès a "erebe" ?
- Quels sont les utilisateurs qui ont à la fois un accès groupé et un accès individuel à "erebe"?
- Quels sont les utilisateurs ayant accès à toutes les machines?

Donner l'arbre algébrique pour la 3eme question.

## B2- Vues

- Créer la vue indiquant les machines qui ne sont accessibles à personne ?  
Peut on la mettre à jour ? Pourquoi ?
- Créer la vue qui donne les machines dont le nombre d'utilisateurs autorisés est inférieur à 20. On prendra garde à ne compter chaque utilisateur qu'une seule fois.  
Peut on mettre à jour cette vue ? Pourquoi ?

## C- questions diverses

Répondez en deux lignes aux questions suivantes :

- Etant données une relation universelle, et une décomposition de cette relation universelle qui préserve les DFs. Si cette décomposition n'est pas sans perte, que suffit il de faire pour la rendre sans perte ?
- Etant données une relation universelle, et une décomposition de cette relation universelle qui est sans perte, cette décomposition préserve-t-elle obligatoirement les DFS ?

## Correction

**Assurance** (NumAssurance, NumPersonne, Nom, Prenom, Adresse, NumAssuré, NumCond, NumImmat, TypeAss, Bonus)

### Partie 1

- Une police d'assurance est souscrite par une seule personne

NumAssurance -> NumAssure

Hyp. supp. Mais on suppose qu'une personne peut souscrire plusieurs polices d'assurance.

- mais peut concerner plusieurs vehicules

On n'a donc pas la DF numAssurance -> NumImmat

- Chaque véhicule doit avoir un conducteur principal qui peut être différent de l'assuré lui-même.

Hyp. supp. Un véhicule peut être assuré plusieurs fois, mais à chaque fois sous des polices d'assurances différentes.

NumAssurance, NumImmat -> NumCond

- Chaque véhicule peut être assuré sous un régime particulier (tous risques, tiers...).

NumAssurance, NumImmat -> TypeAss

- Le coefficient de bonus est propre au véhicule pour une police d'assurance donnée.

NumAssurance, NumImmat -> Bonus

- Une personne ne peut pas être conducteur principal de plus d'une voiture pour une même police d'assurance.

NumCond, NumAssurance -> NumImmat

- Plus les hypothese classiques

NumPers -> Nom, Prénom, Adresse

Donc on a les DF suivantes :

DF1: NumPers -> Nom, Prénom, Adresse

DF2: NumCond, NumAssurance -> NumImmat

DF3: NumAssurance, NumImmat -> Bonus, TypeAss, NumCond

DF4: NumAssurance -> NumAssure

Il y a en outre dépendance entre NumCond et NumPers, NumAss et NumPers. Les assurés sont des personnes, Idem pour les conducteurs.

DF5: NumAssure -> NumPers

DF6: NumCond -> NumPers

La clé de la relation universelle est donc (NumAssurance, NumImmat).

Algo 3FN qui preserve les DFs

Personne(NumPers, Nom, Prénom, Adresse)

VehiculeAssure(NumAssurance, NumImmat, Bonus, TypeAss, NumCond)

Assurance (NumAssurance, NumAssure)

Assure(NumAssure, NumPers)

Conducteur(NumCond, NumPers)

Si les domaines de NumCond, NumAssure et NumPers sont identiques, alors les dépendances fonctionnelles 5 et 6 deviennent des dépendances d'inclusion, les deux dernières relations sont superflues et les relations précédentes deviennent :

VehiculeAssure(NumAssurance, NumImmat, Bonus, TypeAss, NumPers)

Assurance (NumAssurance, NumPers)

## Partie 2

On désignera par les synonymes U, G, AU, AG, H respectivement les relations User, NetGroup, AccessUser, AccessGrp, Host.

### Q1:

```
SELECT login FROM U, AU, H WHERE U.userid=AU.userid AND AU.hostid=H.hostid AND  
H.hostname='erebe'
```

UNION

```
SELECT login FROM U, G, AG, H WHERE U.userid=G.userid AND G.netgrpid=AG.netgrpid AND  
AG.hostid=H.hostid AND H.hostname='erebe'
```

### Q2:

idem avec INTERSECTS au lieu d' UNION

### Q3:

```
SELECT login FROM U WHERE NOT EXISTS
```

```
(SELECT * FROM H WHERE hostid NOT IN
```

```
(SELECT AU.hostid FROM AU WHERE H.hostid=AU.hostid AND AU.userid=U.userid
```

UNION

```
SELECT AG.hostid FROM AG,G WHERE H.hostid=AG.hostid AND AG.netgrpid=G.netgrpid AND  
G.userid = U.userid)
```

## Partie 3

### Q1:

```
CREATE VIEW private_host AS
```

```
SELECT * FROM H WHERE hostid NOT IN
```

```
(SELECT AU.hostid FROM AU UNION SELECT AG.hostid FROM AG, G WHERE
```

```
AG.netgrpid=G.netgrpid
```

```
) WITH CHECK OPTION
```

/\* peut être mise à jour, en insertion, pourvu que le hostid ne soit pas déjà dans un tuple de AU ou de AG \*/

### Q2:

```
CREATE VIEW acces AS
```

```
SELECT * FROM AU UNION SELECT AG.hostid, G.userid FROM AG, G WHERE
```

```
AG.netgrpid=G.netgrpid;
```

/\* cette vue ne peut être directement mise à jour \*/

```
CREATE VIEW low_access_host AS
```

```
SELECT hostid, hostname FROM acces
GROUP BY hostid, hostname HAVING COUNT(userid) < 20
/* cette vue, construite par agrégation de tuples, ne peut être mise à jour */
```

## Exercices corrigés (2)

### A. Normalisation

#### Préambule:

Vous savez tous désormais ce qu'est une opération de jointure relationnelle. Si on remplace les attributs de jointures alphanumériques par des attributs spatiaux (type complexe représentant par exemple une suite de points, de lignes, etc) et l'opérateur (=, >, <, etc) par un opérateur spatial (inclusion, intersection, adjacence, etc), on parle alors de jointure spatiale. Cette opération de jointure spatiale est une opération assez critique, dans la mesure où elle fait intervenir des données volumineuses et des opérateurs assez complexes.

Nous développons à l'école une plateforme d'évaluation de performance des jointures spatiales.

Pour ce faire, Nous définissons des types de données spatiales (DT: datatype), qui correspondent à (i) un certain espace (un rectangle défini par longueur et sa largeur), (ii) un certain nombre d'objets spatiaux et (iii) une distribution statistique. Ces types peuvent être instanciés autant de fois qu'on le souhaite, on obtient alors autant de jeux de données spatiaux identifiés par un nom de fichier (DSfile: Data Sample file) et correspondant chacun à un type DT.

De la même manière, nous définissons des modèles de tests (TM), chaque modèle de test est la spécification d'une jointure spatiale sur deux types de données spatiales, avec un opérateur spatial particulier.

? l'instanciation de ces modèles de tests sur les "data samples" correspond un nombre de solution (solnb). Les instances de ces modèles de tests peuvent être exécutées autant de fois qu'on le souhaite à un temps donné et sur une machine donnée (time, host), avec un algorithme donné (algo) et permettent de recenser le nombre de lectures, écritures (read, write) réalisées.

#### Sujet:

**Nous proposons la relation universelle suivante pour stocker dans une base de données l'ensemble des résultats de notre plateforme.**

**Universelle** (TMid, DTid1, longueur1, largeur1, nb1, distri1, DSfile1, DTid2, longueur2, largeur2, nb2, distri2, DSfile2, op, algo, host, time, solnb, read, write)

- Donnez une couverture minimale des DFs et proposez une clé pour cette relation.
- Proposez une décomposition en 3FN. Cette décomposition est elle en BCNF ?
- Y a-t-il redondance quand à l'information sur les types de données spatiales ? Comment expliquez vous cette redondance ? proposez un moyen intuitif de la supprimer.

## B. Requêtes SQL

On dispose du schema relationnel :

- Assurance (**numassurance**, numassuré)
- Véhicule (**numassurance**, **numvehicule**, numconducteur, typeassurance, bonus)
- Personne (**numpersonne**, nom, prénom, adresse)

sur lequel on veut poser les requêtes suivantes:

- Quelles sont les voitures assurées par Monsieur Dupont ? (SQL **et algèbre relationnelle** )
- Y a-t-il des assurés qui ont plusieurs contrats d'assurance ?

Vous donnerez deux expressions de cette requête en SQL, une qui utilise des fonctions agrégats, l'autre pas.

- Quels sont les assurés qui ne sont pas conducteur principal d'un vehicule qu'ils assurent ?

- Quels sont les assurés qui sont conducteur principal d'un des véhicules qu'ils assurent ?
- Trouvez les assurés qui sont conducteur principal de tous les véhicules qu'ils assurent.
- Quels sont les assurés dont tous les véhicules ont un bonus supérieur à 0.5 ?
- Créez la vue numassurance, numvehicule, numcond, typeassurance, bonus) des véhicules qui sont assurés plusieurs fois sous des numéros d'assurances différents.

Est il possible de mettre cette vue à jour ? La réponse à cette question dépend-elle de la façon dont la vue a été définie ?

- Parmi les véhicules assurés plusieurs fois, quels sont ceux qui sont toujours assurés par la même personne.

### C. Questions générales

- Est il exact de dire qu'un schema relationnel est une instance du modèle relationnel ?
- En vous inspirant du vocabulaire relationnel, indiquez si le langage HTML est un schéma ou un modèle ?

## Correction

### A. Normalisation

#### Préambule:

**Vous savez tous désormais ce qu'est une opération de jointure relationnelle. Si on remplace les attributs de jointures alphanumériques par des attributs spatiaux (type complexe représentant par exemple une suite de points, de lignes, etc) et l'opérateur (=, >, <, etc) par un opérateur spatial (inclusion, intersection, adjacence, etc), on parle alors de jointure spatiale.**

Cette opération de jointure spatiale est une opération assez critique, dans la mesure où elle fait intervenir des données volumineuse et des opérateurs assez complexes.

Nous développons à l'école une plateforme d'évaluation de performance des jointures spatiales.

Pour ce faire, Nous définissons des types de données spatiales (DT: datatype), qui correspondent à (i) un certain espace (un rectangle défini par longueur et sa largeur), (ii) un certain nombre d'objets spatiaux et (iii) une distribution statistique. Ces types peuvent être instanciés autant de fois qu'on le souhaite, on obtient alors alors autant de jeux de données spatiaux identifiés par un nom de fichier (DSfile: Data Sample file) et correspondant chacun à un type DT.

De la même manière, nous définissons des modèles de tests (TM), chaque modèle de test est la spécification d'une jointure spatiale sur deux types de données spatiales, avec un opérateur spatial particulier.

? l'instanciation de ces modèles de tests sur les "data samples" correspond un nombre de solution (solnb). Les instances de ces modèles de tests peuvent être exécutées autant de fois qu'on le souhaite à un temps donné et sur une machine donnée (time, host), avec un algorithme donné (algo) et permettent de recenser le nombre de lectures, écritures (read, write) réalisées.

### Sujet:

**Nous proposons la relation universelle suivante pour stocker dans une base de données l'ensemble des résultats de notre plateforme.**

**Universelle** (TMid, DTid1, longueur1, largeur1, nb1, distri1, DSfile1, DTid2, longueur2, largeur2, nb2, distri2, DSfile2, op, algo, host, time, solnb, read, write)

- Donnez une couverture minimale des DFs et proposez une clé pour cette relation.
  - o DTid1-> longueur1, largeur1, nb1, distri1
  - o DTid2-> longueur2, largeur2, nb2, distri2
  - o TMid -> DTid1, DTid2, op  
et éventuellement  
DTid1, DTid2, op -> TMid (\*)
  - o DSfile1->DTid1
  - o DSfile2->DTid2
  - o DSid1, DSid2, op -> solnb
  - o DSid1, DSid2, op, algo, host, time -> read, write  
Il y a des variantes possibles à cette dernière DF suivant les hypothèses faites. Par exemple, on peut supposer que l'on ne peut pas lancer plus d'un test à la fois sur un host à un instant donné. Dans ce cas, la DF devient :  
host,time -> DSid1, DSid2, op, algo, read, write.

Une clé possible est (TMid, DSfile1, DSfile2, algo, host, time)

Une autre clé possible si on ajoute la DF (\*) est (DSfile1, DSfile2, op, algo, host, time)

- Proposez une décomposition en 3FN. Cette décomposition est-elle en BCNF ?
  - o DT1 (**DTid1**, longueur1, largeur1, nb1, distri1)
  - o DT2 (**DTid2**, longueur2, largeur2, nb2, distri2)



- o TM (**TMid**, DTid1, DTid2, op)
- o DS1 (**DSfile1**, DTid1)
- o DS2 (**DSfile2**, DTid2)
- o TI (**DSid1, DSid2, op**, solnb)  
si (\*) TI (**DSid1, DSid2, op**, solnb, TMid)
- o TE (**DSid1, DSid2, op, algo, host, time**, read, write)

Elle est en 3FN par construction. Elle est également en BCNF car, quelle que soit la relation, il n'y a pas de DF qui ne parte d'un ensemble d'attributs ne formant pas une clé.

- Y a-t-il redondance quand à l'information sur les types de données spatiales ? Comment expliquez vous cette redondance ? proposez un moyen intuitif de la supprimer.

On peut remplacer DT1, DT2 par la relation DT et idem pour DS1, DS2.

- o DT (**DTid**, longueur, largeur, nb, distri)
- o DS (**DSfile**, DTid)

On a alors des dépendances d'inclusion entre TM.DTid1-2 et DT.DTid, TI.DSfile1-2 et DS.DSfile.

.

## B. Requêtes SQL

On dispose du schema relationnel :

- Assurance (**numassurance**, numassuré)
- Véhicule (**numassurance, numvehicule**, numconducteur, typeassurance, bonus)
- Personne (**numpersonne**, nom, prénom, adresse)

sur lequel on veut poser les requêtes suivantes:

- Quelles sont les voitures assurées par Monsieur Dupont ? (SQL **et algèbre relationnelle** )

```
select numvehicule from Assurance A, Vehicule V, Personne P
where A.numassurance=V.numassurance and
A.numassure=P.numpersonne and
nom='Dupont';
```

- Y a-t-il des assurés qui ont plusieurs contrats d'assurance ?

Vous donnerez deux expressions de cette requête en SQL, une qui utilise des fonctions agrégats, l'autre pas.

```
select numassure from Assurance A
group by numassuré having count(numassurance) >1;
```

```
select A1.numassure from Assurance A1, Assurance A2
where A1.numassure=A2.numassure and
A1.numassurance <> A2.numassurance;
```

- Quels sont les assurés qui ne sont pas conducteur principal d'un vehicule qu'ils assurent ?

La question est ambiguë. On ne sait pas si on recherche les assurés qui ne sont conducteur principal d'aucun véhicule qu'ils assurent ou si on recherche qui ne sont pas conducteur principal d'au moins un véhicule qu'ils assurent.

Aucun véhicule

```
select numassure from Assurance
numassure not in (select numassuré from Assurance A, Vehicule V
where A.numassurance=V.numassurance and
A.numassure=V.numcond);
```

Au moins un véhicule

```
select numassuré from Assurance A, Vehicule V
where A.numassurance=V.numassurance and
A.numassure<>V.numcond;
```

- Quels sont les assurés qui sont conducteur principal d'un des vehicules qu'ils assurent ?

```
select numassuré from Assurance A, Vehicule V
where A.numassurance=V.numassurance and
A.numassure=V.numcond;
```

- Trouvez les assurés qui sont conducteur principal de tous les véhicules qu'ils assurent.

```
select numassure from Assurance
numassure not in (select numassuré from Assurance A, Vehicule V
where A.numassurance=V.numassurance and
A.numassure <> V.numcond);
```

- Quels sont les assurés dont tous les véhicules ont un bonus supérieur à 0.5 ?

```
select numassure from Assurance
numassure not in (select numassuré from Assurance A, Vehicule V
where A.numassurance=V.numassurance and
V.bonus < 0.5);
```

- Créez la vue numassurance, numvehicule, numcond, typeassurance, bonus) des véhicules qui sont assurés plusieurs fois sous des numéros d'assurances différents.

```
create view AV as
select numassurance, numvehicule, numcond, typeassurance, bonus
from Vehicule
where numvehicule in (select numvehicule from Vehicule
group by numvehicule having count(numassurance)>1);
```

Est il possible de mettre cette vue à jour ? La réponse à cette question dépend-elle de la façon dont la vue a été définie ?

Telle que définie, on peut mettre la vue a jour. Si oin l'avait defini par une simple cjointure declarative, cela aurait ete impossible.

- Parmi les véhicules assurés plusieurs fois, quels sont ceux qui sont toujours assurés par la même personne.

```
select numvehicule
from Assurance A, AV
where A.numassurance=AV.numassurance
group by numvehicule
having count(distinct numassure) =1;
```

## C. Questions générales

- Est il exact de dire qu'un schema relationnel est une instance du modèle relationnel ?

oui. les tables raltionnelles sont une instance d'un sceh'ma relationnel, lui-même instance du modèle relationnel

- En vous inspirant du vocabulaire relationnel, indiquez si le langage HTML est un schéma ou un modèle ?

Par rapport au lien d'instanciation, HTML est une instance de SGML. Les instances de HTML sont des données. En comparant avec le modèle relationnel, on peut donc dire que SGML est le modèle de données et HTML est un schéma de données.

En revanche, si on s'attache à la rigueur de la structure de données, on peut difficilement défendre l'idée que HTML est un schéma. En effet, les documents HTML n'ont aucune structure rigide et peuvent tous avoir leur propre structure, différente de celle d'un autre. HTML n'impose rien en ce sens. C'est pourquoi on dit parfois en parlant de bases de données documentaires de bases de données sans schéma.

## Exercices corrigés (3)

### A- requêtes relationnelles

Les informations concernant une base de données bibliographique sont stockées dans le schéma relationnel suivant:

**Livres** (numlivre, titre, annee)

**Motsclefs** (numlivre, motclef)

**Coauteurs** (numlivre, numaut)

**Auteurs** (numaut, nomaut, prenoma, info)

### A1- Requêtes relationnelles

Répondre en SQL aux questions suivantes :

- Quels sont les titres des livres dont les auteurs ne sont pas référencés dans la BD ?
- Combien d'auteurs ont signé un livre parlant de "relational data bases" ?
- Donner les titres des livres qui parlent à la fois de "relational join" et de "clustering" ?
- Donner les noms des auteurs qui n'ont jamais écrit un livre seul ?
- supprimer les auteurs qui n'ont écrit aucun livre.

### A2- Vues

Créer une vue qui contient tous les auteurs qui ont écrit (ou coécrit) au moins un livre en 1997.

Peut-on mettre à jour cette vue (insertion, suppression)? Pourquoi ? Cette caractéristique est-elle liée à la façon dont la vue a été définie ?

Parmi ces auteurs, lesquels ont écrit plus de dix livres sur l'ensemble de leur carrière ?

### B- Modélisation

Cet exercice est indépendant du précédent, même s'il en reprend le contexte.

Le fond documentaire décrit dans la partie A peut maintenant être utilisé au sein d'un service de prêts. On souhaite donc enrichir son schéma relationnel pour la gestion des adhérents de la bibliothèque.

Il y a deux types d'adhérents, enseignant ou élève. La durée standard d'un prêt est de 4 semaines. Un élève peut bénéficier d'un prêt de 3 mois, mais il doit alors faire garantir sa demande par un enseignant.

Pour créer cette base de données, on propose la relation universelle suivante :

RELUNIV (numlivre, numens, nomens, prenomens, adresseens, numeleve, nomeleve, prenomeleve, adresseleve, datesortie, dateretour)

En cas d'emprunt d'un livre par un enseignant, seuls les attributs numlivre, numens, nomens, prenomens, adresseens, datesortie sont renseignés.

En cas d'emprunt standard d'un livre par un élève, seuls les attributs numlivre, numeleve, nomeleve, prenomeleve, adresseleve, datesortie sont renseignés. En cas d'emprunt longue durée par un élève, les attributs numens, nomens, prenomens, adresseens sont aussi renseignés.

dateretour est mis à jour au moment où le document est rendu.

**Hypothèse supplémentaire** : on souhaite garder un historique des emprunts.

- Donner la liste des dépendances fonctionnelles en les validant par les hypothèses de l'énoncé ou par des hypothèses supplémentaires que vous préciserez.
- En déduire une cle de la relation RELUNIV.
- Proposer une décomposition en 3 FN, sans perte et qui préserve les DFs.
- Cette décomposition vous paraît elle perfectible ? Si oui, comment ?
- Un élève LAMBDA ayant suivi le cours de dominante informatique émet quelques critiques sur le schéma relationnel de la bibliothèque et suggère de partir plutôt d'une nouvelle liste d'attributs.

RELUNIV2 (numlivre, numpers, nompers, prenompers, adrpers, typepers, numgarant, datesortie, dateretour)

où typepers indique si la personne est enseignant ou élève et numgarant est le numéro de l'enseignant qui, en cas de prêt longue durée d'un élève, garantit ce prêt.

A votre avis, LAMBDA a-t-il brillamment réussi le module de Bases de Données ? Expliquez pourquoi. (max 15 lignes et quelques DFs)

### C- questions diverses

Répondez en quelques lignes aux questions suivantes :

- SQL est il un langage déclaratif ou procédural ? Expliquez votre réponse. (5 lignes max)
- Supposons qu'on ait déclaré une contrainte de clé référentielle entre l'attribut numauteur de la relation Coauteurs et l'attribut numauteur de la relation Auteurs. Quelles sont les deux manipulations permettant de vérifier que la contrainte est bien contrôlée par le SGBD ? Peut on vérifier une telle contrainte à l'aide de vues (définies avec des check option) ? (10 lignes max)

# Correction

## A- requêtes relationnelles

Les informations concernant une base de données bibliographique sont stockées dans le schéma relationnel suivant:

**Livres** (numlivre, titre, annee)

**Motsclefs** (numlivre, motclef)

**Coauteurs** (numlivre, numaut)

**Auteurs** (numaut, nomaut, prenomaut, info)

## A1- Requêtes relationnelles

Répondre en SQL aux questions suivantes :

- Quels sont les titres des livres dont les auteurs ne sont pas référencés dans la BD ?

```
select titre from Livres
where numlivre not in (select numlivre from Coauteurs);
```

- Combien d'auteurs ont signé un livre parlant de "relational data bases" ?

```
select count(numaut)
from Coauteurs, Motsclefs
where Motscles.numlivre=Coauteurs.numlivre
and motclef = 'relational data bases';
```

- Donner les titres des livres qui parlent à la fois de "relational join" et de "clustering" ?

```
select titre from Livres
where
numlivre in (select numlivre from Motsclefs where motclef='relational join')
and
numlivre in (select numlivre from Motsclefs where motclef='clustering');
```

- Donner les noms des auteurs qui n'ont jamais écrit un livre seul ?

```
select nomauteur from Auteurs
where numauteur not in (
select C1.numauteur from Coauteurs C1
where C1.numlivre not in (
select C2.numlivre from Coauteurs C2
where C2.numauteur <> C1.numauteur [and C2.numlivre=C1.numlivre]));
```

```
select nomauteur from Auteurs
where numauteur not in (
select C1.numauteur from Coauteurs C1, Coauteurs C2
where (C1.numlivre, C2.numauteur) not in (
select numlivre, numauteur from Coauteurs));
```

```
select nomauteur from Auteurs
where numauteur not in (
select numauteur from Coauteurs
where numlivre not in (
select C1.numlivre from Coauteurs C1, Coauteurs C2
where C2.numauteur <> C1.numauteur and C2.numlivre=C1.numlivre));
```

```
select nomauteur from Auteurs
where numauteur not in (
select numauteur from Coauteurs
where numlivre in (
select numlivre from Coauteurs
group by numlivre
having count(numauteur)=1 ));
```

```
select nomauteur from Auteurs
where numauteur not in (
select numauteur from Coauteurs
where numlivre not in (
select numlivre from Coauteurs
group by numlivre
having count(numauteur)>=2 ));
```

- supprimer les auteurs qui n'ont écrit aucun livre.

```
delete from Auteurs
where numauteur not in (select numauteur from Coauteurs);
```

## A2- Vues

Créer une vue qui contient tous les auteurs qui ont écrit (ou coécrit) au moins un livre en 1997.

```
Create view Auteurs97 as
select * from Auteurs
where numauteur in (
select numauteur from Coauteurs, Livres
where annee=1997 and Coauteurs.numlivre=Livres.numlivre)
```

Peut on mettre à jour cette vue (insertion, suppression)? Pourquoi ? Cette caractéristique est elle liée à la façon dont la vue a été définie ?

Oui en insertion. Mais le tuple ne sera pas visible dans la vue si Coauteurs n'a pas été mis à jour.



Oui en suppression, mais cela ne supprime pas les tuples de Coauteurs.

Dépend de la syntaxe de création de la vue.

Autre syntaxe ne supportant pas de mise à jour

Create view Auteurs97 as

```
select A.* from Auteurs A, Coauteurs C, Livres L
```

```
where A.numauteur=C.numauteur
```

```
annee=1997 and C.numlivre=L.numlivre
```

Parmi ces auteurs, lesquels ont écrit plus de dix livres sur l'ensemble de leur carrière ?

```
select nomauteur from Auteurs97 A97, Coauteurs C
```

```
where A97.numauteur=C.numauteur
```

```
group by A97.numauteur, nomauteur
```

```
having count (numlivre ) > 10
```

## B- Modélisation

Cet exercice est indépendant du précédent, même s'il en reprend le contexte.

Le fond documentaire décrit dans la partie A peut maintenant être utilisé au sein d'un service de prêts. On souhaite donc enrichir son schéma relationnel pour la gestion des adhérents de la bibliothèque.

Il y a deux types d'adhérents, enseignant ou élève, La durée standard d'un prêt est de 4 semaines. Un élève peut bénéficier d'un prêt de 3 mois, mais il doit alors faire garantir sa demande par un enseignant.

Pour créer cette base de données, on propose la relation universelle suivante :

RELUNIV (numlivre, numens, nomens, prenomens, adresseens, numeleve, nomeleve, prenomeleve, adresseleve, datesortie, dateretour)

En cas d'emprunt d'un livre par un enseignant, seuls les attributs numlivre, numens, nomens, prenomens, adresseens, datesortie sont renseignés.

En cas d'emprunt standard d'un livre par un élève, seuls les attributs numlivre, numeleve, nomeleve, prenomeleve, adresseleve, datesortie sont renseignés. En cas d'emprunt longue durée par un élève, les attributs numens, nomens, prenomens, adresseens sont aussi renseignés.

dateretour est mis à jour au moment où le document est rendu.

**Hypothèse supplémentaire** : on souhaite garder un historique des emprunts.

- Donner la liste des dépendances fonctionnelles en les validant par les hypothèses de l'énoncé ou par des hypothèses supplémentaires que vous préciserez.

numens-> nomens, prenomens, adressens

numeleve -> nomeleve, prenomeleve, adresseleve

numlivre,datesortie -> numens, numeleve, dateretour

Ne prend pas en compte qu'un livre peut être emprunté pour une journée.

- En déduire une clef de la relation RELUNIV.

numlivre, datesortie

- Proposer une décomposition en 3 FN, sans perte et qui préserve les DFs.

Enseignant(numens, nomens, prenomens, adressens)

Eleve(numeleve, nomeleve, prenomeleve, adresseleve)

Emprunt(numlivre, datesortie, numens, numeleve, dateretour)

- Cette décomposition vous paraît-elle perfectible ? Si oui, comment ?

En factorisant Eleve et enseignant

Personne (numpers, nompers, prenompers, adressepers)

- Un élève LAMBDA ayant suivi le cours de dominante informatique émet quelques critiques sur le schéma relationnel de la bibliothèque et suggère de partir plutôt d'une nouvelle liste d'attributs.

RELUNIV2 (numlivre, numpers, nompers, prenompers, adrpers, typepers, numgarant, datesortie, dateretour)

où typepers indique si la personne est enseignant ou élève et numgarant est le numéro de l'enseignant qui, en cas de prêt longue durée d'un élève, garantit ce prêt.

A votre avis, LAMBDA a-t-il brillamment réussi le module de Bases de Données ? Expliquez pourquoi. (max 15 lignes et quelques DFs)

un tel schéma ne peut pas être analysé simplement à l'aide de DFs. En particulier, le lien entre numgarant et numpers est une dépendance d'inclusion et ne peut pas être pris en compte dans une décomposition basée sur la théorie de la normalisation.

## C- questions diverses

Répondez en quelques lignes aux questions suivantes :

- SQL est-il un langage déclaratif ou procédural ? Expliquez votre réponse. (5 lignes max)

Oui et non. Oui car inspiration du calcul relationnel. Non car la négation implique imbrication et donc procéduralité.

- Supposons qu'on ait déclaré une contrainte de clé référentielle entre l'attribut numauteur de la relation Coauteurs et l'attribut numauteur de la relation Auteurs. Quelles sont les deux manipulations permettant de vérifier que la contrainte est bien contrôlée par le SGBD ? Peut on vérifier une telle contrainte à l'aide de vues (définies avec des check option) ? (10 lignes max)

Une insertion sur Coauteurs et une suppression sur Auteurs

Insertion vérifiable par vue + check option , mais pas suppression.