

NFP107 : Systèmes de gestion de bases de données
EXAMEN 2ème session, septembre 2007
 (tout document écrit autorisé)

1 Modèle relationnel (12 points)

Un réseau de garages utilise une base de données relationnelle pour répertorier les informations sur les réparations de voitures effectuées dans les garages du réseau. La base utilisée a le schéma suivant :

- Garage(idGarage, nomGarage, gérant)
- Véhicule(immatriculation, marque, modèle, propriétaire, garageVendeur)
- Réparation(idGarage, immatriculation, date)
- Personne(nom, adresse, âge)

Les clés primaires sont soulignées. Un garage a un identifiant numérique et il est caractérisé par son nom et par le nom de son gérant. Un véhicule est identifié par son immatriculation et caractérisé par la marque du véhicule (ex. Renault), le modèle (ex. Twingo), le nom du propriétaire et, si la voiture a été achetée à un garage, par l'identifiant du garage vendeur. Une réparation, identifiée par le garage qui réalise la réparation et par l'immatriculation de la voiture, est caractérisée par la date à laquelle la réparation a été réalisée. Une personne est identifiée par son nom et caractérisée par son adresse et son âge.

Modèle conceptuel

1. (1 point) Identifiez, à partir de la description ci-dessus, toutes les clés étrangères dans ce schéma relationnel, en précisant la relation référencée par chaque clé étrangère.

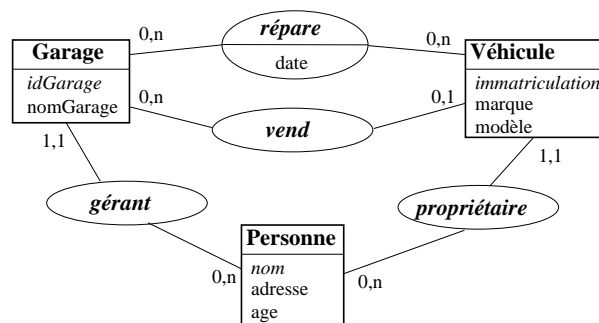
Solution : Dans Garage : gérant clé étrangère → Personne.

Dans Véhicule : propriétaire clé étrangère → Personne et garageVendeur → Garage.

Dans Réparation : idGarage clé étrangère → Garage et immatriculation → Véhicule.

2. (2,5 points) Construisez le schéma conceptuel (entité-association, Merise, etc.) de cette base de données en précisant bien les multiplicités des associations.

Solution :



3. (1 point) En se basant seulement sur le schéma relationnel et indépendamment de la vraisemblance de la situation, est-il possible qu'un véhicule soit réparé plusieurs fois dans le même garage ? Est-il possible qu'il soit réparé dans plusieurs garages ? Justifiez votre réponse.

Solution : Un véhicule ne peut pas être réparé plusieurs fois dans le même garage, même à des dates différentes, car le couple (idGarage, immatriculation) est une clé pour Réparation. Par contre, un même véhicule peut apparaître associé à des garages différents.

Requêtes

1. (1 point) Les dates de réparation de la voiture immatriculée "123ABC99" (SQL et algèbre).

Solution : SQL :

```
select date from Réparation
where immatriculation="123ABC99";
```

Algèbre :

$$\Pi_{date}(\sigma_{immatriculation="123ABC99"}(Réparation)).$$

2. (1,5 points) La marque de voiture du gérant du garage "AutoMax" (SQL et algèbre).

Solution : SQL :

```
select marque from Garage, Véhicule
where nomGarage="AutoMax" and
      Garage.gérant = Véhicule.propriétaire;
```

Algèbre :

$$\Pi_{marque}(Véhicule \bowtie_{propriétaire=gérant} (\sigma_{nomGarage="AutoMax"}(Garage))).$$

3. (2 points) Les garagistes (gérants de garage) qui ne possèdent pas de voiture (SQL et algèbre).

Solution : SQL :

```
select gérant from Garage
where gérant not in (select propriétaire from Véhicule);
```

Algèbre :

$$\Pi_{gérant}(Garage) - \rho_{propriétaire/gérant}(\Pi_{propriétaire}(Véhicule)).$$

4. (1,5 points) Le nom et l'adresse des personnes qui ont réparé leur véhicule dans le garage d'où il a été acheté (SQL).

Solution :

```
select p.nom, p.adresse
from Personne p, Réparation r, Véhicule v
where v.immatriculation=r.immatriculation and
      v.garageVendeur=r.idGarage and
      v.propriétaire=p.nom;
```

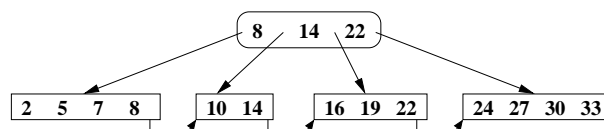
5. (1,5 points) Pour chaque modèle de la marque Renault, le nom du modèle et le nombre de réparations enregistrées pour ce modèle (SQL).

Solution : SQL :

```
select modèle, count(*)
from Véhicule, Réparation
where marque="Renault" and
      Véhicule.immatriculation=Réparation.immatriculation and
group by modèle;
```

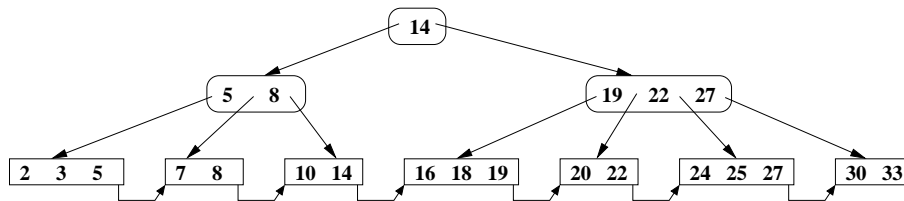
2 Organisation physique et optimisation (5 points)

Index Dans la base de données il existe un index sur l'attribut idGarage de la relation Garage. Cet index est de type arbre B+ d'ordre 2. L'index ne stocke pas les n-uplets de Garage, mais seulement des pointeurs vers ces n-uplets sur disque. A un moment donné, l'index a la structure suivante :



1. (1,5 points) Montrer l'évolution de l'index après insertion successive de garages ayant pour identifiant les valeurs 3, 18, 25 et 20.

Solution :



2. (1 point) Quel est le coût de la première insertion ci-dessus (celle de la valeur 3) dans l'index ? Le coût s'exprime en termes de lectures/écritures de pages disque, sachant que chaque noeud de l'arbre occupe une page. On considère qu'il n'y a pas de cache qui garde plusieurs pages disque en mémoire.

Solution : Coût insertion = coût recherche + coût mise-à-jour arbre. La recherche de la feuille qui doit contenir 3 prend deux lectures (hauteur arbre). L'insertion de 3 dans la feuille provoque son éclatement, donc on écrira deux noeuds (pages). La valeur du milieu (5) remonte, donc il faut une lecture et une écriture du noeud parent. En tout, cela prend $2+2+2=6$ lectures/écritures de pages.

Optimisation Soit la requête suivante, qui retourne les dates et les garages où des véhicules Renault ont été réparés.

```
select date, idGarage from Réparation, Véhicule
where marque="Renault" and
      Véhicule.immatriculation=Réparation.immatriculation;
```

1. (1,5 points) Donnez le plan d'exécution physique pour cette requête s'il existe un index sur l'immatriculation dans Réparation. Expliquez et justifiez ce plan, présenté sous forme d'arbre ou de plan EXPLAIN.

Solution : Jointure par boucles imbriquées avec parcours d'index : parcours séquentiel et sélection sur la marque de Véhicule et parcours d'index sur Réparation.immatriculation.

2. (1 point) On rajoute un index sur la marque des véhicules dans Véhicule, mais l'optimiseur du système indique que le plan d'exécution ne change pas. Donnez une explication vraisemblable pour la non-utilisation de cet index.

Solution : L'index sur la marque de voiture est peu sélectif, il y a probablement une proportion importante de véhicules Renault. Utiliser l'index pour la sélection sur la marque est plus coûteux qu'une sélection séquentielle si le nombre de véhicules Renault est plus grand que le nombre de pages disque de Véhicule.

3 Concurrence (3 points)

Soit l'exécution concurrente de trois transactions suivante, où x et y sont des n -uplets des relations de la base de données.

$$r_1[x]w_2[y]r_3[x]r_1[y]w_3[x]w_1[y]c_2w_3[y]c_3c_1$$

1. (0,5 points) Laquelle des trois transactions pourrait provenir de l'exécution de la requête du sujet "Optimisation" ci-dessus ?

Solution : Aucune, chacune des 3 transactions contient au moins une écriture et la requête en question ne fait que des lectures.

2. (1 point) Vérifier si l'exécution est sérialisable, en construisant son graphe de sérialisation.

Solution : Conflits : sur x : $r_1[x] - w_3[x]$, sur y : $w_2[y] - r_1[y]$, $w_2[y] - w_1[y]$, $w_2[y] - w_3[y]$, $r_1[y] - w_3[y]$, $w_1[y] - w_3[y]$

Le graphe de sérialisation contient les arcs $T_1 \rightarrow T_3$, $T_2 \rightarrow T_1$ et $T_2 \rightarrow T_3$. Il n'y a pas de cycle dans ce graphe, donc l'exécution est sérialisable.

3. (1,5 points) Trouver l'exécution produite par verrouillage à deux phases si les verrous d'une transaction sont relâchés après son *Commit*. Les verrous de lecture sont partageables, ceux d'écriture sont exclusifs. Les opérations bloquées en attente de verrou s'exécutent en priorité quand le verrou devient disponible, en respectant l'ordre de blocage.

Solution : $r_1[x]$ s'exécute en prenant le verrou de lecture sur x

$w_2[y]$ s'exécute en prenant le verrou d'écriture sur y

$r_3[x]$ partage le verrou de lecture sur x avec $r_1[x]$ et s'exécute

$r_1[y]$ bloquée par $w_2[y]$, donc T_1 bloquée

$w_3[x]$ bloquée par $r_1[x]$, donc T_3 bloquée

$w_1[y]$ bloquée, car T_1 bloquée

c_2 s'exécute et relâche les verrous de T_2 sur $y \Rightarrow r_1[y]$ et $w_1[y]$ peuvent s'exécuter en prenant les verrous de lecture/écriture sur y , mais pas $w_3[x]$, qui reste bloquée

$w_3[y]$ et c_3 bloquées, car T_3 bloquée

c_1 s'exécute et relâche les verrous de $T_1 \Rightarrow w_3[x]$, $w_3[y]$ et c_3 s'exécutent

Résultat : $r_1[x]w_2[y]r_3[x]c_2r_1[y]w_1[y]c_1w_3[x]w_3[y]c_3$