

TD1 & TP1 : bibliothèque des solutions

Les relations de cette base de données sont accessibles en préfixant leur nom par `biblio`. Par exemple :
`select ... from biblio.leCatalogue.`

3 Compréhension et lecture

Question 1 :

Compléter le schéma donné dans l'énoncé par l'expression des contraintes d'intégrité mentionnées dans le texte.

- Faire remarquer que les domaines ne sont pas définis. Les contraintes de typage sont des contraintes après tout !

`domaine(titre) = domaine(nom) = domaine(prénom)`
`domaine(cote) = domaine(noAdh)`
`domaine(anNais) = entiers`
`domaine(datEmp) = domaine(datADh) = date`

- les contraintes d'intégrité référentielles :

`LeFonds[titre] = LeCatalogue[titre]`
`LesEmprunts[cote] ⊆ LeFonds[cote]`
`LesEmprunts[noAdh] ⊆ LesAdhérents[noAdh]`

- une personne ne peut emprunter qu'un seul exemplaire d'un même livre
- le nombre de livres empruntés par un adhérent à un instant donné est inférieur à un nombre donné, le même pour tous les adhérents.

Discuter de "un exemplaire ne peut être emprunté que s'il est disponible" : cette contrainte est prise en compte dans le schéma car `cote` est identifiant dans la relation `LeFonds`.

On souhaite sans doute aussi que dans la relation `LeFonds`, l'attribut `titre` ait toujours une valeur, de même pour les attributs `noAdh`, `datEmp` dans la relation `LesEmprunts`. La valuation obligatoire n'est pas à spécifier pour les attributs participants aux clefs, car dans ce cas elle est implicite. On le verra plus tard, lorsqu'une valeur vide (*absente*) est comparée à une autre valeur, et quelque soit l'opérateur, la valeur retournée est vide. Par suite, il est raisonnable d'interdire des valeurs vides pour des attributs pour lesquels on veut assurer des valeurs sans répétitions.

Question 2 :

Les expressions suivantes sont-elles correctes ? Si non, pourquoi ? Si oui, pour chacune, donnez la spécification de la relation qu'elle construit (attribut, identifiant, prédicat).

- `select titre, nom, prénom from LeCatalogue where anEd=1959;`

Soit $R(\underline{\text{titre}}, \text{nom}, \text{prénom})$ la relation construite par la requête $\langle t, n, p \rangle \in R \iff$ le livre de titre t écrit par l'auteur de prénom p et de nom n a été édité en 1959.

2. `select noAdh from LesEmprunts where cote <> 10;`

Cette requête ne construit pas une relation (c'est-à-dire un ensemble). Un même adhérent pouvant emprunter plusieurs exemplaires dont la cote est différente de 10, son numéro apparaîtra dans le résultat en autant de fois. Il fallait utiliser la clause `select distinct`.

3. `select distinct titre from LeFonds;`

Soit $R(\underline{\text{titre}})$ la relation construite par la requête $\langle t \rangle \in R \iff t$ est le titre d'un exemplaire. Puisque $\text{LeFonds}[\text{titre}] = \text{LeCatalogue}[\text{titre}]$, il s'agit des titres des livres détenus par la bibliothèque. On aurait pu écrire : `select titre from LeCatalogue;`

4. `select titre, anEd from LeFonds;`

Cette requête est incorrecte car l'attribut `anEd` n'est pas défini dans la relation `LeFonds`.

5. `select titre, nom, prénom from LeCatalogue;`

Soit $R(\underline{\text{titre}}, \text{nom}, \text{prénom})$ la relation construite par la requête $\langle t, n, p \rangle \in R \iff$ le livre de titre t a été écrit par l'auteur de nom n et de prénom p .

4 Expression de requêtes

Question 3 :

Pour chaque requête ci-dessous, spécifier la relation construite par le résultat attendu, et donner son expression en SQL. Les requêtes devront construire des résultats sans répétition de valeurs. La clause `distinct` sera utilisée uniquement lorsque nécessaire. Les produits de relation seront exprimés dans la clause `from` des requêtes.

En TD, ne traiter que les requêtes qui utilisent uniquement sélection et projection. Faire réfléchir à plusieurs versions lorsque c'est possible.

Pour le TP, les faire réfléchir aux tests : (1) faire varier les paramètres quand il y en a, (2) ajouter des données pour illustrer des cas limites.

1. Titre et année d'édition des livres du catalogue.

Soit $R(\underline{\text{titre}}, \text{anEd})$ la relation construite par la requête. $\langle t, a \rangle \in R \iff$ le livre de titre t a été édité l'année a .

```
select titre, anEd from LeCatalogue;
```

2. Numéro, nom, prénom et adresse des adhérents nés après 1960.

Soit $R(\underline{\text{noAdh}}, \text{nom}, \text{prénom}, \text{adresse})$ la relation construite par la requête. $\langle a, n, p, v \rangle \in R \iff$ l'adhérent identifié par a est né en 1960 ou après. Il(elle) a pour nom n , prénom p , et il(elle) habite la ville v .

```
select noAdh, nom, prenom, adresse
from LesAdherents
where anNais >= 1960;
```

3. Nom et prénom des auteurs dont la bibliothèque possède au moins un exemplaire.

Soit $R(\underline{\text{nom}}, \underline{\text{prénom}})$ la relation construite par la requête. $\langle n, p \rangle \in R \iff$ la personne de nom n et de prénom p est l'auteur d'un livre dont la bibliothèque possède un exemplaire au moins.

```
select distinct nom, prenom
from LeCatalogue;
```

4. Numéro des adhérents qui n'ont aucun emprunt.

Soit $R(\underline{\text{noAdh}})$ la relation construite par la requête. $\langle a \rangle \in R \iff$ l'adhérent identifié par a n'a aucun emprunt en cours.

```
select noAdh from LesAdherents minus select noAdh from LesEmprunts;
```

Autre solution :

```
select noAdh from LesAdherents
where noAdh not in (select noAdh from LesEmprunts);
```

5. Numéro des adhérents qui n'empruntent aucun des exemplaires dont la cote est 2, 3, ou 8.

Soit $R(\underline{\text{noAdh}})$ la relation construite par la requête. $\langle a \rangle \in R \iff$ l'adhérent de numéro a n'emprunte aucun des exemplaires de cote 2, 3, ou 8.

```
select noAdh from LesEmprunts
minus
select noAdh from LesEmprunts
where cote = 2 or cote = 3 or cote = 8;
```

Autre version :

```
select distinct noAdh from LesEmprunts
where noAdh not in
  (select noAdh from LesEmprunts
   where cote = 2 or cote = 3 or cote = 8);
```

Dans les requêtes ci-dessus, l'expression de sélection $\text{cote} = 2 \text{ or } \text{cote} = 3 \text{ or } \text{cote} = 8$ peut être remplacée par $\text{cote in } (2, 3, 8)$.

6. Cote et titre des exemplaires de livres édités en 1980 ou après.

Soit $R(\underline{\text{cote}}, \text{titre})$ la relation construite par la requête. $\langle c, t \rangle \in R \iff$ l'exemplaire de cote c est un exemplaire d'un livre de titre t et édité en 1980 ou après.

```
select cote, titre
from LeFonds natural join LeCatalogue
where anEd >= 1980;
```

7. Numéro, nom et prénom des adhérents qui empruntent au moins un exemplaire d'un livre écrit par Franck Herbert.

Soit $R(\underline{\text{noAdh}}, \text{nom}, \text{prénom})$ la relation construite par la requête. $\langle a, n, p \rangle \in R \iff$ l'adhérent identifié par a , de nom n et de prénom p , emprunte au moins un exemplaire d'un livre de Franck Herbert.

```
select distinct noAdh, A.nom, A.prenom
from LesAdherents A natural join LesEmprunts E
  natural join LeFonds F
  join LeCatalogue C on (F.titre = C.titre)
where C.nom = 'Herbert' and C.prenom = 'Franck';
```

8. Nom et prénom des adhérents qui sont auteurs d'au moins un livre détenu par la bibliothèque.

Soit $R(\underline{\text{nom}}, \underline{\text{prénom}})$ la relation construite par la requête. $\langle n, p \rangle \in R \iff n$ et p sont les nom et prénom d'adhérents qui sont auteurs d'au moins un livre détenu par la bibliothèque.

Faire remarquer que les adhérents ne sont pas identifiés par leur nom et leur prénom. De même, rien ne garantit qu'un couple de nom et prénom identifie un seul auteur.

```
select nom, prenom from LesAdherents
intersect
select nom, prenom from LeCatalogue;
```

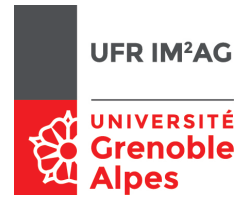
Autre version (noter la clause `select distinct`) :

```
select distinct nom, prenom
from LesAdherents natural join LeCatalogue;
```

9. Couple de titre de livres d'auteurs différents et édités la même année.

Soit $R(\underline{\text{titre1}}, \underline{\text{titre2}})$ la relation construite par la requête. $\langle t1, t2 \rangle \in R \iff t1$ est le titre d'un livre édité la même année que le livre $t2$. $t1$ et $t2$ sont d'auteurs différents.

```
select C1.titre as titre1, C2.titre as titre2
from LeCatalogue C1 join LeCatalogue C2
on (C1.anEd=C2.anEd and (C1.nom<>C2.nom or C1.prenom<>C2.prenom)
and C1.titre < C2.titre);
```



TP1 : découvrir Oracle *des solutions*

L'accès au Système de Gestion de Bases de Données (SGBD) Oracle n'est possible que sur le serveur `im2ag-oracle`.

1 Connexion à Oracle

Se connecter à la machine `im2ag-oracle`. Pour cela, par exemple, à partir d'un autre serveur taper :

```
ssh -X im2ag-oracle
```

Vous devez fournir votre nom d'utilisateur et votre mot de passe (ceux que vous utilisez habituellement sur le réseau de l'université). Vous êtes sous **Unix**, il s'affiche :

```
{1}im2ag-oracle
```

Ensuite, appeler le logiciel **Oracle** en utilisant :

```
{1}im2ag-oracle sqlplus
```

Vous devez fournir votre nom d'utilisateur (le même que précédemment) et le mot de passe associé (celui que vous utilisez habituellement sur le réseau de l'université).

Vous êtes sous **Oracle**, il s'affiche :

```
SQL>
```

Pour quitter l'interprète SQL d'Oracle, utiliser `exit`, qui ramène sous Unix. Ensuite, pour sortir d'Unix, utiliser à nouveau `exit`.

Si besoin, pour travailler à la fois avec Oracle et avec un éditeur, ouvrir deux fenêtres sous `im2ag-oracle`; dans l'une, appeler Oracle par `sqlplus`, et, dans l'autre, appeler un éditeur par `nedit` (par exemple).

La commande `host` permet d'exécuter un shell depuis `sqlplus`. Par exemple :

```
SQL> host  
> gedit &  
> exit  
SQL>
```

2 Exécution d'une requête en ligne

Dans l'environnement Oracle, taper la ligne suivante et analyser le résultat :

```
SQL> select titre from biblio.LeCatalogue;
```

Ensuite, taper la ligne suivante et analyser le résultat :

```
SQL> select titre from LeCatalogue;
```

Pour avoir une description des attributs de la relation `biblio.LeCatalogue`, taper :

```
SQL> desc biblio.LeCatalogue
```

Les relations de la bibliothèque ont été créées dans le schéma `biblio`. Pour basculer vers ce schéma pendant la durée de la session `sql` en cours, il suffit de soumettre la requête : `alter session set current.schema=biblio`. La durée de la bascule est liée à celle de la session `sql`.

3 Exécution d'une requête à partir d'un fichier

La plupart du temps, on prépare les requêtes dans un fichier qu'on demande ensuite à Oracle d'exécuter, en utilisant la commande **start**. Créer un fichier de suffixe **.sql**, par exemple **Req1.sql**, puis dans ce fichier, taper le texte suivant :

```
-- le titre des livres de la bibliothèque
select titre
from biblio.LeCatalogue;
```

La ligne commençant par "--" est une ligne de commentaires, les autres lignes forment une requête SQL. Il ne faut pas placer de commentaires à l'intérieur d'une requête.

Pour exécuter cette requête, taper sous Oracle :

```
SQL> start Req1
```

Le système cherche alors le fichier **Req1.sql** et l'exécute. Le résultat doit être le même que lorsqu'on tape la requête en ligne. Essayer différentes variantes dans le fichier **Req1.sql**, les faire exécuter et analyser le résultat. Si Oracle indique des erreurs, les corriger.

Essayer aussi d'écrire plusieurs requêtes l'une après l'autre dans le même fichier.

Il est aussi possible de procéder par *copier-coller* : copier la requête rédigée avec l'éditeur puis la coller dans la fenêtre où Oracle est exécuté.

4 Stockage des résultats des requêtes dans un fichier

Pour stocker automatiquement les résultats des requêtes dans un fichier, utiliser **spool**, en insérant dans le fichier **Req1.sql** une ligne en haut et une ligne en bas :

```
spool ResReq1
... (ici vos requêtes) ...
spool off;
```

Comme précédemment, taper sous Oracle :

```
SQL> start Req1
```

Un fichier est créé automatiquement, de nom **ResReq1.lst** (le nom que vous avez donné, suivi du suffixe **.lst**). Ouvrir le fichier **ResReq1.lst** pour voir son contenu.

Remarque : on peut remplacer **spool ResReq1** par **spool Req1**, le système ne confond pas les fichiers **Req1.lst** et **Req1.sql** puisqu'ils n'ont pas le même suffixe.

On obtient le même comportement par copier-coller de la requête.

5 Requêtes paramétrées

Dans un fichier de nom par exemple **Req2.sql**, taper le texte suivant (où **accept** et **prompt** sont des mots-clés d'Oracle) :

```
-- Numéro, nom, et prénom des adhérents nés après 1960
-- et qui habitent à Grenoble.
accept lannee prompt 'Donner une annee de naissance : '
accept laville prompt 'Donner un nom de ville : '
select noAdh, nom, prenom
from biblio.LesAdherents
where adresse = '&laville' and anNais > &lannee;
```

Attention à l'utilisation des quotes ! Il faut en mettre pour '**&laville**' (chaîne de caractères) mais pas pour **&leprix** (nombre).

Taper **start Req2**. Le système imprime le texte **Donner une annee de naissance :** (sans quotes). Répondre en donnant un entier. Puis le système imprime le texte **Donner un nom de ville :** (sans quotes). Répondre en donnant une ville (sans quotes). Alors le système exécute la requête.

