

TP SQL

1^{ère} année Informatique

Najib Tounsi

La BD pour une bibliothèque est constituée des tables suivantes (Voir en annexe le script SQL de création sur le SGBD mysql):

Table des livres (Numéro inventaire du livre, titre et auteur du livre, nombre d'exemplaires dans la bibliothèque)

Livre

NumInv	Titre	Auteur	Qte
323	Bases De Données	C.J. Date	3
124	Algorithmes	D. Knuth	2
1110	Programmation	B. Meyer	6
241	Langage C	B. Kernighan	5
321	Unix	K. Thompson	2
216	Systèmes Exploitations	Crocus	1
312	Langage Java	J. Gosling	8

Table des abonné(e)s (Numéro abonné, son nom et prénom)

Abonne

NumAb	Nom	Prénom
12	Benali	Ali
10	BenSaid	Said
32	Karim	Karima
14	Rajaoui	Rajaa

Table des prêts (Numéro inventaire du livre emprunté, Numéro abonné de l'emprunteur et la date d'emprunt)

Pret

NumInv	NumAb	DatePret
323	10	1/11/2007
124	12	2/11/2007
323	12	3/10/2007
241	32	3/10/2007
321	14	13/10/2007
323	14	10/11/2007
312	10	3/10/2007
323	32	13/09/2007

- I. Créer ces tables en Access. Choisir le type entier pour QTE, le type texte de 4 caractères pour NumInv, texte de 3 caractères pour NumAb, type date pour DatePret et texte de 15 caractères pour les champs Nom, Prenom et Auteur.
- II. Alimenter ces tables par les valeurs données.
- III. Ecrire en SQL les requêtes suivantes :

1. Interroger les informations d'une table

I. Rechercher toutes le données

a.

```
mysql> select * from livre;
```

numinv	titre	auteur	qte
323	Bases De Données	C.J. Date	3
124	Algorithmes	D. Knuth	2
1110	Programmation	B. Meyer	6
241	Langage C	B. Kernighan	5
321	Unix	K. Thompson	2
216	Systèmes Exploitati	Crocus	1
312	Langage Java	J. Gosling	8

b.

```
mysql> select * from pret;
```

numinv	numab	datepret
323	10	2007-11-01
124	12	2007-11-02
323	12	2007-10-03
241	32	2007-10-03
321	14	2007-10-13
323	14	2007-11-10
312	10	2007-10-03
323	32	2007-09-13

II. Rechercher certaines lignes

a. « Tous les livres disponibles en quantité égale à 2 »

```
mysql> select *  
-> from livre  
-> where qte = 2;
```

numinv	titre	auteur	qte
124	Algorithmes	D. Knuth	2
321	Unix	K. Thompson	2

+-----+-----+-----+-----+

b. « Tous les livres disponibles en quantité comprise entre 6 et 2 »

```
mysql> select *
-> from livre
-> where qte < 6 and qte > 2;
```

numinv	titre	auteur	qte
323	Bases De Données	C.J. Date	3
241	Langage C	B. Kernighan	5

c. « Tous les livres de titre Unix ou disponibles en quantité comprise entre 6 et 2 »

```
mysql> select *
-> from livre
-> where (qte < 6 and qte > 2) OR titre= "Unix" ;
```

numinv	titre	auteur	qte
323	Bases De Données	C.J. Date	3
241	Langage C	B. Kernighan	5
321	Unix	K. Thompson	2

III. Rechercher certaines colonnes

a. « Le titre et la quantité de tous les livres »

```
mysql> select titre, qte
-> from livre ;
```

titre	qte
Bases De Données	3
Algorithmes	2
Programmation	6
Langage C	5
Unix	2
Systèmes Exploitati	1
Langage Java	8

+-----+-----+

b. « Le titre, la quantité et la quantité augmentée de 5 de tous les livres » Usage des expressions.

```
mysql> select titre, qte, qte+5 as plus5  
      -> from livre;
```

titre	qte	plus5
Bases De Données	3	8
Algorithmes	2	7
Programmation	6	11
Langage C	5	10
Unix	2	7
Systèmes Exploitati	1	6
Langage Java	8	13

(noter la nouvelle colonne dans le résultat (uniquement le résultat)).

c. « Les différentes quantité des livres »

```
mysql> select qte  
      -> from livre;
```

qte
3
2
6
5
2
1
8

Usage de **DISTINCT**..

```
mysql> select distinct qte from livre;
```

qte
3

2
6
5
1
8

d. « Le titre et la quantité des livres disponibles en quantité supérieure à 3»

```
mysql> select titre, qte
      -> from livre
      -> where qte >3 ;
```

titre	qte
Programmation	6
Langage C	5
Langage Java	8

IV. Trier les lignes résultats

« Le titre et l'auteur des livres classés par titre»

a.

```
mysql> select titre, auteur
      -> from livre
      -> order by titre;
```

titre	auteur
Algorithmes	D. Knuth
Bases De Données	C.J. Date
Langage C	B. Kernighan
Langage Java	J. Gosling
Programmation	B. Meyer
Systèmes Exploitati	Crocus
Unix	K. Thompson

b. Ordre décroissant

```
mysql> select titre, auteur
```

```
-> from livre
-> order by titre desc;
```

titre	auteur
Unix	K. Thompson
Systèmes Exploitati	Crocus
Programmation	B. Meyer
Langage Java	J. Gosling
Langage C	B. Kernighan
Bases De Données	C.J. Date
Algorithmes	D. Knuth

c. Uniquement les livre en quantité supérieure à 3

```
mysql> select titre, auteur
-> from livre
-> where qte >3
-> order by titre;
```

order by APRES where

titre	auteur
Langage C	B. Kernighan
Langage Java	J. Gosling
Programmation	B. Meyer

d.

```
mysql> select titre, auteur from livre order by titre
-> where qte >3;
```

```
ERROR 1064: You have an error in your SQL syntax near
'where qte >3' at line 1
```

(Inversion des clauses **order by** et **where**. Le tri vient en fin de recherche ; donc après where.)

V. Calcul sur les dates (!!! adapter à Access. Utiliser date())

a.

```
mysql> select *, curdate() from pret;
```

numinv	numab	datepret	curdate()
--------	-------	----------	-----------

323	10	2007-11-01	2007-11-28
124	12	2007-11-02	2007-11-28
323	12	2007-10-03	2007-11-28
241	32	2007-10-03	2007-11-28
321	14	2007-10-13	2007-11-28
323	14	2007-11-10	2007-11-28
312	10	2007-10-03	2007-11-28
323	32	2007-09-13	2007-11-28

b.

```
mysql> select * from pret
      -> where curdate() -datepret >120;
```

numinv	numab	datepret
323	12	2007-10-03
241	32	2007-10-03
312	10	2007-10-03
323	32	2007-09-13

VI. Recherche sur motifs (Pattern Matching)

a. « Titre et auteurs des livres dont le titre commence par 'L' »

NB. En SQL Access, utiliser '*' eu lieu de '%' et '?' à la place de '_'.

```
mysql> select titre, auteur from livre
      -> where titre like "L%";
```

titre	auteur
Langage C	B. Kernighan
Langage Java	J. Gosling

b. « Titre et auteurs des livres dont la 2^e lettre du titre est un 'a' »

```
mysql> select titre, auteur from livre
      -> where titre like "_a%";
```

titre	auteur
Bases De Données	C.J. Date

Langage C	B. Kernighan
Langage Java	J. Gosling

c. « *Titre et auteurs des livres dont le titre a quatre lettres* »

```
mysql> select * from livre where titre like "____";
```

numinv	titre	auteur	qte
321	Unix	K. Thompson	2

VII. Calcul statistiques (Compter les lignes, moyennes de colonnes etc...)

a. « *Quel est le nombre de livres* »

```
mysql> select count(*) from livre;
```

count(*)
7

b. « *Quel est le nombre de livre disponible en quantité égale à 2* »

```
mysql> select count(*) from livre
-> where qte = 2;
```

count(*)
2

c. « *Renommer la colonne résultat de count* »

```
mysql> select count(*) as nb_livres from livre;
```

nb_livres
7

d. « Quel est, par livre, le nombre d'abonnés l'ayant emprunté »

```
mysql> select numinv, count(*)
      -> from pret
      -> group by numinv;
```

numinv	count(*)
124	1
241	1
312	1
321	1
323	4

e. « Quelle est le livre emprunté par plus de 3 abonnés »

```
mysql> select numinv from pret
      -> group by numinv
      -> having count(*) > 3;
```

numinv
323

f. « Quelle est la quantité maximum disponible pour un livre »

```
mysql> select max(qte) from livre;
```

max(qte)
8

g. « Quelle est le livre disponible en quantité maximum »

```
mysql> select * from livre
      -> where qte = (select max(qte) from livre);
```

numinv	titre	auteur	qte
--------	-------	--------	-----

312	Langage Java	J. Gosling	8

NB. Cette forme de requête n'est pas toujours implémentées.

VIII. Utiliser plusieurs tables (JOIN)

a. « Quel est le numéro des livres emprunté par l'abonné 12 »

```
mysql> select numinv
      -> from pret where numab=12;
```

numinv
124
323

b. « Quel est le titre des livres emprunté par l'abonné 12 ».
Ici on doit interroger aussi la table livre pour chercher le nom du livre.

```
mysql> select titre
      -> from livre, pret
      -> where livre.numinv=pret.numinv
      -> and pret.numab=12;
```

titre
Bases De Données
Algorithme

c. Forme générale d'une jointure

```
Mysql > select *
      -> from livre , pret
      -> where livre.numinv = pret.numinv;
```

numinv	titre	auteur	qte	numinv	numab	datepret
323	Bases De Données	C.J. Date	3	323	10	2007-11-01

323	Bases De Données	C.J. Date	3	323	12	2007-10-03
323	Bases De Données	C.J. Date	3	323	14	2007-11-10
323	Bases De Données	C.J. Date	3	323	32	2007-09-13
124	Algorithme	D. Knuth	2	124	12	2007-11-02
241	LangageC	B. Kernighan	5	241	32	2007-10-03
321	Unix	K.Thompson	2	321	14	2007-10-13
312	LangageJava	J.Gosling	8	312	10	2007-10-03

Qu'on peut faire suivre d'une projection (select champ)

d. « Quels sont les numéros abonné avec le titre des livres empruntés »

```
mysql> select numab, titre
-> from livre , pret
-> where livre.numinv = pret.numinv;
```

numab	titre
10	Bases De Données
12	Bases De Données
14	Bases De Données
32	Bases De Données
12	Algorithmes
32	Langage C
14	Unix
10	Langage Java

Equivalent à

e.

```
mysql> select numab, titre
-> from livre l, pret p
-> where l.numinv = p.numinv;
```

f. « Quels sont les n° abonné et les titres des livres empruntés à la date '2007-10-3' »

```
mysql> select numab, titre
-> from livre l, pret p
-> where l.numinv = p.numinv
-> and datepret = '2007-10-3' ;
```

numab	titre
-------	-------

12	Bases De Données
32	Langage C
10	Langage Java

Essayer aussi

g.

```
mysql> select titre
-> from livre
-> where numinv in (select numinv
-> from pret
-> where datepret ="2007-10-3");
```

titre
Bases De Données
LangageC
LangageJava

Ecriture possible (**requête imbriquée**) quand le résultat final provient d'une seule table (ici, titre dans table livre).

h. Jointure de trois tables

```
mysql> select nom, titre
-> from abonne a, livre l, pret p
-> where a.numab = p.numab
-> and l.numinv = p.numinv;
```

nom	titre
BenSaid	Bases De Données
Benali	Bases De Données
Rajaoui	Bases De Données
Karim	Bases De Données
Benali	Algorithmes
Karim	Langage C
Rajaoui	Unix
BenSaid	Langage Java

- i. Requête imbriquée mettant en oeuvre trois tables.
 « Quel est le nom de l'abonné ayant emprunté un livre sur Unix »

```
mysql> select nom
-> from abonne
-> where numab in (select numab
->                  from pret
->                  where numinv in (select numinv
->                                   from livre
->                                   where titre="Unix"));
```

nom
Rajaoui

- j. « Quelles sont les paires de livres disponibles en quantité égale »
 (Jointure de table avec elle-même)

```
mysql> select l1.titre, l2.titre
-> from livre l1, livre l2
-> where l1.qte = l2.qte;
```

titre	titre
Bases De Données	Bases De Données
Algorithme	Algorithme
Unix	Algorithme
Programmation	Programmation
LangageC	LangageC
Algorithme	Unix
Unix	Unix
Systèmes Exploitati	Systèmes Exploitati
LangageJava	LangageJava

- k. (Astuce, éliminer les cas triviaux)

```
mysql> select l1.titre, l2.titre
-> from livre l1, livre l2
-> where l1.qte = l2.qte and l1.titre > l2.titre;
```

|--|

titre	titre
Unix	Algorithme

IX. Usage de Exists

a. « *Quel est le titre des livres faisant l'objet d'un prêt* »

```
mysql> select titre
      -> from livre
      -> where exists (select *
      ->                  from pret
      ->                  where pret.numinv = livre.numinv);
```

titre
Bases De Données
Algorithme
LangageC
Unix
LangageJava

(les livres 'Programmation' et 'Système Exploitation ne sont empruntés par personne)

b. « *Quel est le titre des livres ne faisant l'objet d'aucun prêt* »

```
mysql> select titre
      -> from livre
      -> where not exists (select *
      ->                  from pret
      ->                  where pret.numinv = livre.numinv);
```

titre
Programmation
Systèmes Exploitati

c. « *Quel est le numéro du livre emprunté par tous les abonnés* » (rep : 323)

```
mysql> select numinv
-> from livre
-> where not exists (select *
->                    from abonne
-> where not exists (select * from pret
->                    where livre.numinv=pret.numinv
->                        and abonne.numab=pret.numab));
```

```
+-----+
| numinv |
+-----+
| 323    |
+-----+
```

OUTER JOIN

- d. « Pour tous les livres afficher les informations le concernant (y compris les prêts) »

La jointure normale donnerait :

```
mysql> select *
-> from livre , pret
-> where livre.numinv=pret.numinv;
```

numinv	titre	auteur	qte	numinv	numab	datepret
323	Bases De Données	C.J. Date	3	323	10	2007-11-01
323	Bases De Données	C.J. Date	3	323	12	2007-10-03
323	Bases De Données	C.J. Date	3	323	14	2007-11-10
323	Bases De Données	C.J. Date	3	323	32	2007-09-13
124	Algorithme	D. Knuth	2	124	12	2007-11-02
241	LangageC	B. Kernighan	5	241	32	2007-10-03
321	Unix	K.Thompson	2	321	14	2007-10-13
312	LangageJava	J.Gosling	8	312	10	2007-10-03

Mais la réponse exacte est :

e.

```
mysql> select *
-> from livre
-> left join pret on livre.numinv=pret.numinv;
```

numinv	titre	auteur	qte	numinv	numab	datepret
323	Bases De Données	C.J. Date	3	323	10	2007-11-01
323	Bases De Données	C.J. Date	3	323	12	2007-10-03

323	Bases De Données	C.J. Date	3	323	14	2007-11-10
323	Bases De Données	C.J. Date	3	323	32	2007-09-13
124	Algorithme	D. Knuth	2	124	12	2007-11-02
1110	Programmation	B. Meyer	6	NULL	NULL	NULL
241	LangageC	B. Kernighan	5	241	32	2007-10-03
321	Unix	K.Thompson	2	321	14	2007-10-13
216	Systèmes Exploitati	Crocus	1	NULL	NULL	NULL
312	LangageJava	J.Gosling	8	312	10	2007-10-03

Ici, les livres 1110 et 216 ne joignent pas avec la table des prêts, mais doivent figurer dans le résultat.

LEFT JOIN, signifie, la jointure se fait de la table à gauche dans from (livre) vers celle à sa droite (pret).

NB : Il y a aussi **RIGHT JOIN**.

f.

```
mysql> select *
-> from livre
-> right join pret on livre.numinv=pret.numinv;
```

numinv	titre	auteur	qte	numinv	numab	datepret
323	Bases De Données	C.J. Date	3	323	10	2007-11-01
124	Algorithme	D. Knuth	2	124	12	2007-11-02
323	Bases De Données	C.J. Date	3	323	12	2007-10-03
241	LangageC	B. Kernighan	5	241	32	2007-10-03
321	Unix	K.Thompson	2	321	14	2007-10-13
323	Bases De Données	C.J. Date	3	323	14	2007-11-10
312	LangageJava	J.Gosling	8	312	10	2007-10-03
323	Bases De Données	C.J. Date	3	323	32	2007-09-13

Ici, la jointure va de la table **pret** vers la table **livre**. Toutes les lignes de **pret** correspondent à une ligne de **livre**. (Ici, c'est le résultat de la jointure habituelle)

X. Commande de mise à jour

a) Ajouter le livre <334, Weaving the Web, Tim Berners-Lee, 4>

```
INSERT INTO Livre VALUES ('334', 'Weaving the Web', 'Tim Berners-Lee', 5)
```

b) Ajouter l'abonné <18, Kacem>. (On ne connaît pas le nom)

```
INSERT INTO Abonne (Numab, Prenom) VALUES ('18', 'Kacem')
```

Noter la désignation explicite des attributs concernés.

c) Supprimer le livre 334

```
DELETE FROM Livre WHERE NumInv= '334'
```

d) Remplacer le nom de l'abonné 18 par Benkacem

```
UPDATE Abonne SET Nom = 'Benkacem' WHERE NumAb='18'
```

e) Augmenter de 2 la quantité du livre 334

```
UPDATE Livre SET Qte = Qte +2 WHERE NumInv='334 '
```

Annexe

BD créée sur mysql par le script

```
[tounsi@khayyam Biblio]$ cat Credb.sql

drop database if exists biblio;

CREATE DATABASE biblio;
use biblio;

drop table if exists livre;
create table livre (
  numinv varchar(4) not null ,
  titre varchar(20),
  auteur varchar(15),
  qte int(2) unsigned,
  primary key (numinv)
);

create index livreind on livre (numinv);

drop table if exists abonne;
create table abonne(
  numab varchar(3) not null ,
  nom varchar(15),
  prenom varchar(15),
  primary key (numab)
);

create index abonneind on abonne (numab);

drop table if exists pret;
create table pret (
  numinv varchar(4) not null ,
  numab varchar(3) not null ,
  datepret date,
  primary key (numinv,numab)
);
create index pretind on pret (numinv,numab);
```