

# TP final (1)

Up

## TP FINAL (part 1)

# LA BIBLIOTHEQUE

Auteur : *M. Léonard, H. Luu*



SUJET



QUESTION



CORRECTION

## SUJET: LA BIBLIOTHEQUE

On s'intéresse à la gestion d'une bibliothèque pour laquelle on veut que le système d'information permette de consulter la liste des livres de la bibliothèque, d'enregistrer un emprunt ou une réservation, d'enregistrer l'adhésion ou la démission d'une personne etc.

On s'intéresse plus particulièrement à la gestion des livres de la bibliothèque. La bibliothèque achète régulièrement un certain nombre de livres. Après avoir été enregistrés dans la base, ces livres sont disponibles pour être empruntés. Un emprunt ne doit pas dépasser un mois. Un adhérent ne peut garder plus 5 livres à la fois. Un livre peut être emprunté sans réserver.

***1. La modélisation relationnelle suivante représente la structure statique de la base de données de la bibliothèque.***

ADHERENT(noAdh// nom, prénom, adresse, dateAdh)

$\{ \langle na, n, p, a, d \rangle \in \text{ADHERENT} \Leftrightarrow \text{l'adhérent de numéro } na \text{ a pour nom } n \text{ et pour prénom } pn. \text{ Il}$

habite à l'adresse  $a$  et a adhéré à la bibliothèque à la date  $d$ .)

CATALOGUE(titre // nomAut, prénomAut, anEd)

{ $\langle t, n, p, e \rangle \in \text{CATALOGUE} \Leftrightarrow$  la bibliothèque possède l'œuvre de titre  $t$  dont l'auteur a pour nom  $n$  et pour prénom  $p$ . Cette œuvre a été éditée l'année  $e$ .}

FONDS(cote // titre)

{ $\langle c, t \rangle \in \text{FONDS} \Leftrightarrow$  la bibliothèque a l'exemplaire (livre) de l'œuvre de titre  $t$  sous la cote  $c$ .}

RESERVATION(titre, noAdh, dateRes // dateEmp)

{ $\langle t, na, dr, de \rangle \in \text{RESERVATION} \Leftrightarrow$  l'adhérent  $na$  réserve un œuvre  $t$  à la date  $dr$ . Il l'emprunte à la date  $de$ .

EMPRUNT(cote, dateEmp // noAdh, dateRet)

{ $\langle c, de, na, dret \rangle \in \text{EMPRUNT} \Leftrightarrow$  l'exemplaire  $c$  est emprunté à la date  $de$  par l'adhérent  $na$ . Il le rend à la date  $dret$ .}

## II. Les attributs retenus :

Numéro-adhérent	NoAdh	CHAR(7)
Nom de l'adhérent	nom	CHAR(12)
Prénom de l'adhérent	prenom	CHAR(30)
Adresse de l'adhérent	adresse	VARCHAR(50)
Date-adhérence	dateAdh	DATE
Titre-catalogue	titre	CHAR(30)
Nom de l'auteur	nomAut	CHAR(12)
Prénom de l'auteur	prenomAut	CHAR(30)
An-édition	anEd	entier(4)

Cote-exemplaire	cote	CHAR(8)
Date-réservation	dateRes	DATE
Date-emprunt	dateEmp	DATE
Date-retour	dateRet	DATE

### III. *Les règles d'intégrité :*

1. Tous les attributs d'un adhérent ne peuvent prendre des valeurs inconnues.
2. Tous les attributs d'un catalogue ne peuvent prendre des valeurs inconnues. L'an d'édition doit être après 1950.
3. Tous les attributs d'un fonds ne peuvent prendre des valeurs inconnues.
4. Toute réservation, la date de réservation doit être précédente de celle d'emprunt.
5. Tout emprunt, la date d'emprunt doit être précédente de celle de retour.
6. On ne peut réserver ou emprunter un titre qu'après avoir adhéré à la bibliothèque. (La date d'adhérent est précédente de celle de réservation ou d'emprunt)
7. Un emprunt ne doit pas dépasser un mois.
8. Un adhérent ne peut emprunter qu'au plus 5 titres à la fois et un seul exemplaire pour chaque titre.
9. Lorsqu'un emprunt est fait à partir d'une réservation, on enregistre la date d'emprunt à la réservation.

## QUESTION:

### Part I.

1. Classifier les RI suivantes selon la taxonomie apprise.
  - a. Déterminer quelles RI peuvent être implémentées grâce aux RI « standards » supportées par Oracle.
  - b. Pour autres RI, spécifier leur portée.

2. Ecrire des instructions SQL pour créer des tables et des RI standards.

## CORRECTION

1. **Classifier les RI suivantes selon la taxonomie apprise.**  
 a. **Déterminer quelles RI qui peuvent être implémentées grâce aux RI « standards » supportées par Oracle.**

Ü Réponse

**RI1** : Intra-relation (ADHERENT), Individuel, Mono-attribut : NOT NULL

**RI2** : Intra-relation (CATALOGUE), Individuel, Mono-attribut : NOT NULL, CHECK (anEd  $\geq$  1950)

**RI3** : Intra-relation (FONDS), Individuel, Mono-attribut : NOT NULL

**RI4** : Intra-relation (RESERVATION), Individuel, Multi-attributs : CHECK(dateRes  $\leq$  dateEmp)

**RI5** : Intra-relation (EMPRUNT), Individuel, Multi-attributs : CHECK(dateEmp $\leq$ dateRet)

**RI7** : Intra-relation (EMPRUNT), Individuel, Multi-attributs : CHECK(dateRet < add\_month(dateEmp,1))

- b. **Pour autres RI, spécifier leur portée.**

Ü Réponse

**RI6** : *Inter-relations* : **ADHERENT, RESERVATION, EMPRUNT**

- ADHERENT :

i. Modifier la date d'adhésion (dateAdh) : Refuser la modification et afficher une erreur s'il existe déjà des réservations ou des emprunts de cet adhérent dont la date est antérieure à la nouvelle date.

- RESERVATION :

i. Créer une réservation : Si la date de réservation

(dateRes) est antérieure la date d'adhésion (dateAdh), affecter dateAdh à dateRes.

ii. Modifier la date de réservation : Si la nouvelle date ne satisfait pas la règle, refuser la modification et afficher une erreur.

- **EMPRUNT**

i. Créer un emprunt : Si la date d'emprunt (dateEmp) est antérieure la date d'adhésion (dateAdh), affecter dateAdh à dateRes.

ii. Modifier la date d'emprunt : Refuser la modification et afficher une erreur si la nouvelle date ne satisfait pas la règle.

**RI8 :**

**RI8.1.** Un adhérent ne peut emprunter qu'au plus 5 exemplaires à la fois

***Intra-relation (EMPRUNT), Ensembliste***

- **EMPRUNT :**

i. Créer un emprunt : Refuser cette insertion et afficher une erreur si le nombre des emprunts qui ne sont pas encore rendus est déjà 5.

ii. Modifier la date de retour : Refuser la modification et afficher une erreur si elle transgresse la règle.

**RI8.2.** un seul exemplaire pour chaque titre  
***Inter-relations (EMPRUNT, FONDS)***

- **EMPRUNT :**

i. Créer un emprunt : Refuser cette insertion et afficher une erreur si cet adhérent a emprunté (pas encore rendu) un exemplaire du même titre.

ii. Modifier la date de retour : Refuser la

modification et afficher une erreur si elle transgresse la règle.

- FONDS :

- i. Modifier le titre : Refuser la modification et afficher une erreur s'il existe déjà des emprunts de cet exemplaire.

**RI9 : *Inter-relations (FONDS, RESERVATION, EMPRUNT)***

- FONDS :

- i. Modifier le titre : Refuser la modification et afficher une erreur s'il existe déjà des emprunts de cet exemplaire.

- ii. Supprimer un fonds : *Refuser la suppression.*  
(*Cette opération est déjà vérifiée grâce à la clé étrangère*)

- RESERVATION :

- i. Créer une réservation : la date d'emprunt doit être inconnue lorsqu'une nouvelle réservation est faite (insérée).

- ii. Modification la date d'emprunt : refuser la modification et afficher une erreur si elle transgresse la règle.

- EMPRUNT :

- i. Modifier l'adhérent : Refuser la modification et afficher une erreur si elle transgresse la règle.

- ii. Supprimer un emprunt : Si cet emprunt concerne une réservation, refuser la suppression et afficher une erreur.



## 2. ***Ecrire des instructions SQL pour créer les tables avec les RI standards.***

```
===== Création des tables =====
CREATE TABLE adherent (
  noadh CHAR(5) PRIMARY KEY,
  nom CHAR(20) NOT NULL,
  prenom CHAR(30) NOT NULL,
```

```

    adresse VARCHAR(50) NOT NULL,
    dateadh DATE NOT NULL
);

CREATE TABLE catalogue (
    titre CHAR(30) PRIMARY KEY,
    nom CHAR(20) NOT NULL,
    prenom CHAR(30) NOT NULL,
    anEd NUMBER(4) NOT NULL CONSTRAINT chk_anEd CHECK (anEd>=1950)
);

CREATE TABLE fonds (
    cote CHAR(10) PRIMARY KEY,
    titre CHAR(30) NOT NULL REFERENCES catalogue
);

CREATE TABLE emprunt (
    cote CHAR(10),
    dateemp DATE,
    noadh CHAR(5) NOT NULL,
    dateret DATE,
    CONSTRAINT pk_emprunt PRIMARY KEY(cote, dateemp),
    CONSTRAINT fk_adherent FOREIGN KEY (noadh) REFERENCES adherent,
    CONSTRAINT fk_fonds FOREIGN KEY(cote) REFERENCES fonds,
    CONSTRAINT chk_dateemp CHECK(dateemp<=dateret),
    CONSTRAINT chk_dateret CHECK(dateret<=add_months(dateemp,1))
);

CREATE TABLE reservation(
    titre CHAR(30) REFERENCES catalogue,
    noadh CHAR(5) REFERENCES adherent,
    dateres DATE,
    dateemp DATE,
    CONSTRAINT pk_reservation PRIMARY KEY (titre, noadh, dateres),
    CONSTRAINT chk_dateres CHECK (dateres <= dateemp)
);

===== INSERTION DES DONNEES =====
insert into ADHERENT values ('A001','RENAULT','Robert','Lausanne','08/04/01');
insert into ADHERENT values ('A002','VOLVO','Robert','Geneve','01/01/02');
insert into ADHERENT values ('A003','AQUA','Robert','Sion','25/06/01');

insert into CATALOGUE values ('Base de donnees','ADIBA','Michel',1995);
insert into CATALOGUE values ('Base des objets','COLLAND','Rollet',1990);
insert into CATALOGUE values ('OO DMBS','ADIBA','Michel',1998);
insert into CATALOGUE values ('ORACLE SQL','LINDEN','Brian',2000);
insert into CATALOGUE values ('SQL*Plus reference','LINDEN','Brian',2001);
insert into CATALOGUE values ('Web Database','BUYENS','Jim',2000);

```

```
insert into FONDS values('BDD_01','Base de donnees');
insert into FONDS values('BDD_02','Base de donnees');
```

```
insert into FONDS values('BD0_01','Base des objets');
```

```
insert into FONDS values('OOMS_01','OO DMBS');
insert into FONDS values('OOMS_02','OO DMBS');
```

```
insert into FONDS values('ORA.SQL_01','ORACLE SQL');
```

```
insert into FONDS values('SQLP_01','SQL*Plus reference');
insert into FONDS values('WDB_00','Web Database');
```

```
insert into RESERVATION values('Base de donnees','A001','09/04/01',NULL);
insert into RESERVATION values('Base de donnees','A002','10/01/02', NULL);
insert into RESERVATION values('OO DMBS','A002','02/01/02',NULL);
```

```
insert into EMPRUNT values('BDD_02','21/04/01','A001', '03/05/01');
insert into EMPRUNT values('BDD_01','01/06/02','A001', NULL);
insert into EMPRUNT values('BD0_01','03/06/02','A001', NULL);
insert into EMPRUNT values('OOMS_02','03/06/02','A001', NULL);
insert into EMPRUNT values('ORA.SQL_01','01/06/02','A001', NULL);
insert into EMPRUNT values('SQLP_01','01/06/02','A001', NULL);
```

```
insert into EMPRUNT values('OOMS_02','05/06/02','A002', NULL);
insert into EMPRUNT values('OOMS_01','05/01/02','A002', '10/01/02');
```

```
update RESERVATION set dateEmp='21/04/01' where titre='Base de donnees' and
noAdh='A001' and dateRes='09/04/01';
update RESERVATION set dateEmp='05/01/02' where titre='OO DMBS' and
noAdh='A002' and dateRes='02/01/02';
```