

Examen 1 (Corrigé type) : Concepts de bases de données avancées

Durée de l'examen : 1.5 heure

Date : 15/01/2023

Exercice 1 (4 points):

1. Quels sont les atouts du modèle relationnel-objet par rapport au modèle relationnel ?
 - a. **Support des attributs composés.....(1pt)**
 - b. **L'héritage permet une optimisation dans l'utilisation des tables,(1pt)**

Parmi les assertions suivantes, lesquelles expriment des limites des SGBDR qui justifient l'évolution du modèle relationnel classique vers le modèle relationnel-objet ?(1pt)

- a) **Le principe d'atomicité des attributs d'une relation (première forme normale) empêche de disposer de propriétés structurant plusieurs valeurs dans un type complexe.**
- b) **La séparation des données et des traitements empêche l'intégration des méthodes au modèle.**
- c) **La représentation de données complexes conduit à une fragmentation importante de l'information en de multiples relations et à des chutes de performance.**
- d) **Le modèle relationnel ne permet pas d'exécuter correctement des transactions en réseau.**
- e) **Il n'est pas possible de créer des types de données personnalisées.**
- f) **Il n'est pas possible d'exécuter des instructions SQL1 ou SQL2 à partir d'un langage objet tel que Java, alors que c'est possible avec SQL3.**
- g) **L'héritage n'est pas représentable directement.**

2. Selon l'utilisation dans votre TP, PostgreSQL est un SGBD de type :
 - a. Hiérarchique
 - b. orienté objet
 - c. XML/RDF
 - d. Mixte
 - a. **SGBD relationnel avec des extensions objet comme les classes, l'héritage,(1pt).**
 - b. objet-relationnel

Exercice 2 (5 point):

Soit la relation suivante :

CarnetDeVoyage (numAuteur, nomAuteur, prenomAuteur, numVille, nomVille, nomPays, description)

Avec l'ensemble de dépendances fonctionnelles :

- numAuteur → nomAuteur
- numAuteur → prenomAuteur
- numVille → nomVille
- numVille → nomPays
- numAuteur, numVille → description

1. Rappeler la définition formelle d'une clé et donnez la ou les clés existantes.
2. Pour chaque clé, énoncer les DF qui prouvent que c'est une clé.
3. Dites en quelle forme normale est la relation (montrez pourquoi).
4. Proposez un schéma normalisé en 3NF, sans perte d'information, en faisant apparaître les clés.
5. Donnez les requêtes SQL permettant de créer les différentes tables de cette BD.

Solution 2 (5 points)

1. Clé: Une clé est un groupe d'attributs **minimal** qui **détermine tous les attributs de la relation.....(1pt)**

2. Il y a une unique clé (numAuteur, numVille).....(1pt)

- numAuteur, numVille → nomAuteur

- numAuteur, numVille → prenomAuteur
- numAuteur, numVille → nomVille
- numAuteur, numVille → nomPays
- numAuteur, numVille → description

3. La forme normale :

La relation est en 1NF, on a identifié une clé et les attributs sont atomiques. Elle n'est pas en 2NF car des attributs faisant partie de la clé déterminent d'autres attributs, par exemple : numAuteur → nomAuteur.....(1pt)

4. Proposez un schéma normalisé en 3NF, sans perte, en faisant apparaître les clés.....(1pt)

Auteur(#numAuteur:integer, nomAuteur:string, prenomAuteur:string)

Ville(#numVille:integer, nomVille:string, nomPays:string)

CarnetDeVoyage (#numAuteur=>Auteur(numAuteur), #numVille=>Ville(numVille), description:string)

5. Création de tables.....(1pt)

```
CREATE TABLE Auteur (
    numAuteur INTEGER PRIMARY KEY,
    nomAuteur VARCHAR,
    prenomAuteur VARCHAR
);
CREATE TABLE Ville (
    numVille INTEGER PRIMARY KEY,
    nomVille VARCHAR,
    nomPays VARCHAR
);
CREATE TABLE CarnetDeVoyage (
    numAuteur INTEGER REFERENCES Auteur(numAuteur),
    numVille INTEGER REFERENCES Ville(numVille),
    description VARCHAR,
    PRIMARY KEY (numAuteur, numVille)
);
```

Exercice 3 (5 points):

Soit le schéma relationnel suivant gérant le fonctionnement d'une agence de location d'appartements.

```
APPARTEMENT(code_appt, adresse, type, prix_loyer)
LOCATAIRE(code_loc, nom, prenom)
LOCATION(#code_loc, #code_appt)
PAIEMENT_LOYER(#code_loc, #code_appt, date_payement, prix_paye)
```

Donner les requêtes SQL pour:

1. Créer les différentes tables de cette relation.
2. Insérer trois instances dans chaque table.
3. Afficher tous les montants payés par un locataire ayant le code 51.
4. Donner l'adresse de l'appartement loué par un locataire ayant le code 160.

Solution 3 (5 points)

1. La création de tables(1.5pt)

```
CREATE TABLE appartement (
code_appt INTEGER PRIMARY KEY,
adresse VARCHAR,
```

```
type VARCHAR,  
prix_loyer REAL  
);
```

```
CREATE TABLE LOCATAIRE(  
code_loc INTEGER PRIMARY KEY,  
nom VARCHAR,  
prenom VARCHAR  
);
```

```
CREATE TABLE LOCATION(  
#code_loc INTEGER REFERENCES LOCATAIRE(code_loc),  
#code_appt INTEGER REFERENCES appartement (code_appt),  
PRIMARY KEY (#code_loc,#code_appt)  
);
```

```
CREATE TABLE PAIEMENT_LOYER(  
#code_loc INTEGER REFERENCES LOCATAIRE(code_loc),  
#code_appt INTEGER REFERENCES appartement (code_appt),  
date_payement DATE PRIMARY KEY,  
prix_paye REAL  
);
```

2. Insertion des instances(1.5pt)

```
INSERT INTO appartement (code_appt, adresse, type, prix_loyer)VALUES (15,  
'khenchela','F4', 155.55 );  
INSERT INTO appartement (code_appt, adresse, type, prix_loyer)VALUES (16, 'Batna','F3',  
255.55 );  
INSERT INTO appartement (code_appt, adresse, type, prix_loyer)VALUES (17, 'Setif','F5',  
365.55 );
```

```
INSERT INTO LOCATAIRE(code_loc, nom, prenom) VALUES (51, 'Ahmed','BenAli') ;  
INSERT INTO LOCATAIRE(code_loc, nom, prenom) VALUES (130, 'Asmaa','Haddad') ;  
INSERT INTO LOCATAIRE(code_loc, nom, prenom) VALUES (160,'Fouad','Brahimi') ;
```

```
INSERT INTO LOCATION(#code_loc, #code_appt) VALUES (51,15) ;  
INSERT INTO LOCATION(#code_loc, #code_appt) VALUES (14,16) ;  
INSERT INTO LOCATION(#code_loc, #code_appt) VALUES (160,17) ;
```

```
INSERT INTO PAIEMENT_LOYER(#code_loc, #code_appt, date_payement, prix_paye) VALUES  
(51,15,'31-12-2022',155.23) ;  
INSERT INTO PAIEMENT_LOYER(#code_loc, #code_appt, date_payement, prix_paye) VALUES  
(14,16,'25-03-2019',256.33) ;  
INSERT INTO PAIEMENT_LOYER(#code_loc, #code_appt, date_payement, prix_paye) VALUES  
(160,17,'02-01-2001',258.65) ;
```

3. Afficher tous les montants payés par un locataire ayant le code 51.(1pt)

```
SELECT prix_paye  
FROM PAIEMENT_LOYER  
WHERE code_loc=51
```

4. En algèbre relationnelle et en SQL afficher l'adresse de l'appartement loué par un locataire avec le code 160.(1pt)

```
SELECT adresse  
FROM LOCATION l JOIN APPARTEMENT a
```

```
ON l.code_appt=a.code_appt
WHERE l.code_loc='160'
```

Exercice 4 (6 points):

Utilisant les fonctionnalités du relationnel objet, modélisez une base de données de gestion d'emprunts de ressources par des abonnés. Les **ressources** empruntées peuvent être des **disques** ou des **livres**. Ces ressources sont identifiées par leurs numéros ISBN et possèdent un titre et une date de parution. Une personne peut être un éditeur, un abonné ou auteur, il est identifié par son nom et son prénom avec un identifiant. Chaque ressource est éditée par un **éditeur**, dont on stocke l'**adresse** composée de nom de la rue, code postal et ville. Une ressource est écrite par un seul **auteur**. Les livres ont un numéro d'édition et un nombre de pages. Les disques peuvent être des supports **CD** ou **DVD** et possèdent un nombre de pistes. On stocke pour chaque **auteur** son numéro_auteur dans l'ordre des auteurs. On mémorise également pour chaque abonné la date d'inscription des abonnés, ainsi que leur numéro de téléphone et leur adresse afin de pouvoir les contacter. Lorsqu'un abonné **emprunte** une ressource, on mémorise le numéro d'emprunt, la date d'emprunt et la date de restitution.

1. Donner les requêtes permettant de :
 - a. Créer la base de données emprunt_DB,
 - b. Définir tous les types nécessaires. Prendre en compte toutes associations qui existent.
 - c. Définir les tables nécessaires à la base de données.
 - d. Insérer au moins trois lignes par table, en renseignant toutes les colonnes.
2. Donner les requêtes suivantes :
 - a. Quels sont les livres écrits par « George Gardarin » ?
 - b. Quels sont les abonnés qui habitent à la ville d'Alger ?

Solution 4:

1. Création de la base de données emprunt_DB(1pt)

```
Create database emprunt_DB ;
```

2. Définition des types(1pt)

```
Create type T_adresse as (
Rue VARCHAR,
code postale INTEGER,
vile VARCHAR
) ;
```

```
CREATE TYPE Support as ENUM('CD', 'DVD');
```

3. Création des tables(2pt)

```
CREATE TABLE Ressource (
ISBN INTEGER PRIMARY KEY,
Titre TEXT,
date_parution DATE,
editeur INTEGER REFERENCES editeur(id),
id_Auteur INTEGER REFERENCES auteur(id)
);
```

```
CREATE TABLE Disque(
Nbre_piste INTEGER,
Support Support,
) INHERITS (Ressource);
```

```
CREATE TABLE Livre(
Num_edi Text,
```

```
nbre_pages INTEGER
) INHERITS (Ressource);
```

```
CREATE TABLE Personne (
  Id serial not null primary key,
  Nom VARCHAR,
  Prénom VARCHAR
);
```

```
CREATE TABLE Editeur (
  Adresse T_adresse
) INHERITS (Personne);
```

```
CREATE TABLE Auteur(
  num_auteur INTEGER
) INHERITS (Personne);
```

```
CREATE TABLE Abonne(
  date_inscription Date,
  telephone Text,
  Adresse T_adresse
) INHERITS (Personne);
```

```
CREATE TABLE Edition(
  #id_ress INTEGER REFERENCES Ressource(ISBN),
  id_auteur INTEGER REFERENCES Auteur(Id)
);
```

```
CREATE TABLE Emprunt (
  num_epm serial not null primary key,
  date_emp Date,
  #id_abonnée INTEGER REFERENCES Abonne(id),
  id_ressource INTEGER REFERENCES Ressource(ISBN),
  date_retour Date
);
```

4. Insertion des instances(1pt)

```
INSERT INTO Disque(ISBN, Titre ,date_parution, editeur, id_Auteur, Nbre_piste,
Support) values(1234,'Les réseaux',01-01-2001, 'Les pages blues',25,450,'CD');
INSERT INTO Disque(ISBN, Titre ,date_parution, editeur, id_Auteur, Nbre_piste,
Support) values(5467,'Anglais', 02-02-2003,'Chiheb',45,350,'DVD');
INSERT INTO Disque(ISBN, Titre ,date_parution, editeur, id_Auteur, Nbre_piste,
Support) values(3751,'Histoire',05-11-2020, 'Jessour', 34,430,'CD');
```

```
INSERT INTO Livre(ISBN, Titre ,date_parution, editeur, id_Auteur, Nbre_piste,
Support, Num_edi, nbre_pages) values();
INSERT INTO Livre(ISBN, Titre ,date_parution, editeur, id_Auteur, Nbre_piste,
Support, Num_edi, nbre_pages) values();
INSERT INTO Livre(ISBN, Titre ,date_parution, editeur, id_Auteur, Nbre_piste,
Support, Num_edi, nbre_pages) values();
```

```
INSERT INTO Editeur (Id, Nom, Prénom, Adresse) values ();
INSERT INTO Editeur (Id, Nom, Prénom, Adresse) values ();
INSERT INTO Editeur (Id, Nom, Prénom, Adresse) values ();
```

```
INSERT INTO Auteur (Id, Nom, Prénom, num_auteur) values ();
INSERT INTO Auteur (Id, Nom, Prénom, num_auteur) values ();
```

```
INSERT INTO Auteur (Id, Nom, Prénom, num_auteur) values () ;
```

```
INSERT INTO Abonne (Id, Nom, Prénom, date_inscription, telephone, Adresse)  
values () ;
```

```
INSERT INTO Abonne (Id, Nom, Prénom, date_inscription, telephone, Adresse)  
values () ;
```

```
INSERT INTO Abonne (Id, Nom, Prénom, date_inscription, telephone, Adresse)  
values () ;
```

```
INSERT INTO Edition(id_ress, id_auteur) values ();
```

```
INSERT INTO Edition(id_ress, id_auteur) values ();
```

```
INSERT INTO Edition(id_ress, id_auteur) values ();
```

```
INSERT INTO Emprunt (num_epm, date_emp, id_abonnée, id_ressource,date_retour)  
values ();
```

```
INSERT INTO Emprunt (num_epm, date_emp, id_abonnée, id_ressource,date_retour)  
values ();
```

```
INSERT INTO Emprunt (num_epm, date_emp, id_abonnée, id_ressource,date_retour)  
values ();
```

Requêtes

2.A(0.5pt)

```
SELECT titre FROM livre l JOINT Auteur A ON l.id_Auteur = A.id WHERE  
A.nom = Guy AND A.prenom = Pujolle;
```

2.B.(0.5pt)

```
Select Nom, prenom from Abonne where (Adresse).ville ='Alger';
```

بالتوفيق