

## Modèle relationnel

**Exercice 1** On souhaite modéliser un annuaire téléphonique simple dans lequel chaque personne (identifiée par son nom et son prénom) est associée à son numéro de téléphone. Proposer le modèle Conceptuel de Données (MCD) de cet annuaire.

**Exercice 2** Donner la modélisation relationnelle (MCD) d'un bulletin scolaire. Cette dernière doit permettre de mentionner

- des élèves, possédants un numéro d'étudiant alphanumérique unique
- un ensemble de matières fixées, mais qui ne sont pas données
- au plus une note sur 20, par matière et par élève.

On prendra soin de préciser toutes les contraintes utilisateurs qui ne peuvent être inscrites dans les schémas des relations.

**Exercice 3** On considère la solution donnée pour l'exercice 0. Dire si chacun des ensembles est une relation valide pour le schéma Annuaire.

1. {}
2. {{ 'Titi', 'Toto', '0128456789' }}
3. {{ ('Titi', 'Toto', '0123456789'), ('Doe, John', '0123456789') }}
4. {{ ('Titi, Toto, '0123456789'), ('Titi, 'Toto', '987654343210') }}
5. {{ ('Titi', 'Toto', '0128456789'), ('Doe', 'John') }}
6. {{ ('Titi', 'Toto', 42) }}

**Exercice 4** On considère la solution donnée pour l'exercice 141. Dire si chacun des ensembles est une relation valide pour le schéma de la base de données du bulletin de notes.

1.
  - Eleve={}
  - Matiere = {}
  - Note={}
2.
  - Eleve = {{('AB56789', 'Titi', 'Toto')}}
  - Matiere = {{('0, NS1'), (1, 'Sport')}}
  - Note = {{('AB56789', 1, 17)}}
3.
  - Eleve = {{('AB56789', 'Titi', 'Toto')}}
  - Matiere = {{('0, 'NSI')}}
  - Note = {{('AB56789', 1, 17)}}
4.
  - Eleve = {{(' AB56789', 'Titi', 'Toto', )}}
  - Matiere = {{('0, 'NSI')}}
  - Note = {{('AB56789', 0, 17), (7 AB56789', 0, 18)}}
5.
  - Eleve = {{(' AB56789', 'Titi', 'Toto', )}}
  - Matiere = {{('0, 'NSI'), (1, 'Sport')}}
  - Note = {{('AB56789', 0, 17), ('AB56789', 1, 17)}}

## Base de données relationnelle

**Exercice 5** Pour chacune des séquences d'ordres SQL suivantes, dire quelle instruction provoque une erreur. On suppose que la base de données ne contient aucune table au début de chaque séquence.

1.

```
DROP TABLE client;
```

```
CREATE TABLE client (cid INT PRIMARY KEY, nom VARCHAR (100),prenom VARCHAR (100),points_fidelite INT NOT NULL );
```

2.

```
CREATE TABLE client (cid INT PRIMARY KEY,nom VARCHAR (100),prenom VARCHAR (100),points_fidelite INT NOT NULL,);
```

```
CREATE TABLE commande (cid INT REFERENCES client (cid),pid INT REFERENCES produit (pid),date DATE NOT NULL);
```

```
CREATE TABLE produit (pid INT PRIMARY KEY,nom VARCHAR (100),prix DECIMAL(10,2));
```

3.

```
CREATE TABLE client (cid INT PRIMARY KEY, nom VARCHAR (100),prenom VARCHAR (100),points_fidelite INT NOT NULL,);
```

```
CREATE TABLE produit (pid INT PRIMARY KEY,nom VARCHAR (100),prix DECIMAL(10,2));
```

```
CREATE TABLE commande (cid INT REFERENCES client(cid), nom VARCHAR(100) REFERENCES produit (nomp) ,
                        date DATE NOT NULL);
```

**Ex7**

**Soit la base de données suivantes**

Client (Codecli, Nom, Prenom, Adresse, CP, Ville, Tel)

Commande (Num, Date, #Codecli)

Extrait de la table Client							Extrait de la table Commande		
Codecli	Nom	Prenom	Adresse	CP	Ville	Tel	Num	Date	Codecli
41100035	Bacquet	Adeline	10 rue platini	62170	Saint-Josse	0321459631	C20050215	23/03/2005	41100094
41100094	Baheux	Sosthène	296 rue des jumeaux	59500	Douai	0328627431	C20050183	26/06/2005	41100035
41100095	Baillieu	Anastasia	277 boulevard danton	59600	Maubeuge	0328639248	C20050341	18/11/2005	41100094
41100097	Mori	Jean	1 rue des Etournaux	32110	Moulins	489170469			
41100126	Renault	Marie	3 boulevard Victor Hug	59500	Douai	336373900			

Afficher numéros de commande de Sosthène Baheux.

Utiliser une jointure interne syntaxe WHERE idem ensuite avec la syntaxe INNER JOIN

### Ex 7 Extrait- de sujet de bac

Cet exercice porte sur les bases de données.

Dans cet exercice, on pourra utiliser les mots clés suivants du langage SQL :

SELECT, FROM, WHERE, JOIN, ON, INSERT INTO, VALUES, MIN, MAX, OR, AND.

Les fonctions d'agrégation MIN(propriete) et MAX(propriete) renvoient, respectivement, la plus petite et la plus grande valeur de l'attribut propriete pour les enregistrements sélectionnés.

Des acteurs ayant joué dans différentes pièces de théâtre sont recensés dans une base de données `Theatre` dont le schéma relationnel est donné ci-dessous :

- `Piece` (`idPiece`, `titre`, `langue`)
- `Acteur` (`idActeur`, `nom`, `prenom`, `anneeNaiss`)
- `Role` (`#idPiece`, `#idActeur`, `nomRole`)

Dans ce schéma, les clés primaires sont soulignées et les clés étrangères sont précédées du symbole #.

L'attribut `idPiece` de la relation `Role` est une clé étrangère faisant référence à l'attribut `idPiece` de la relation `Piece`.

L'attribut `idActeur` de la relation `Role` est une clé étrangère faisant référence à l'attribut `idActeur` de la relation `Acteur`.

Tous les attributs dont le nom est préfixé par `id` sont des nombres entiers ainsi que l'attribut `anneeNaiss`. Les autres attributs sont des chaînes de caractères.

1. Expliquer pourquoi il n'est pas possible d'insérer une entrée dans la relation `Role` si les relations `Piece` et `Acteur` sont vides.

2. Dans la pièce « Le Tartuffe », l'acteur Micha Lescot a joué le rôle de Tartuffe.

L'identifiant de Micha Lescot est 389761 et celui de cette pièce est 46721.

Écrire une requête SQL permettant d'ajouter ce rôle dans la table (ou relation) `Role`.

3. Expliquer ce que fait la requête SQL suivante.

```
UPDATE Piece
SET langue = "Anglais"
WHERE langue = "Américain" OR langue = "Britannique";
```

4. Pour chacun des quatre items suivants, écrire une requête SQL permettant d'extraire les informations demandées.

a. Le nom et prénom des artistes nés après 1990.

b. L'année de naissance du plus jeune artiste.

c. Le nom des rôles joués par l'acteur Vincent Macaigne.

d. Le titre des pièces écrites en Russe dans lesquelles l'actrice Jeanne Balibar a joué.