

---

## MODÈLE RELATIONNEL

---

**Exercice 0** On souhaite modéliser un annuaire téléphonique simple dans lequel chaque personne (identifiée par son nom et prénom) est associée à son numéro de téléphone. Propose une modélisation relationnelle de cet annuaire.

**Exercice 1** On considère la modélisation relationnelle d'un bulletin scolaire selon les schémas de base de données suivants :

*Eleve(nom String, prenom String, num String)*

*Matiere(inititule String, m\_id Int)*

*Note(#num String, #m\_id Int, note Float)*

Dis 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 = {('Titi', 'Toto', 'AB56789')}*  
• *Matiere = {('NSI', 0), ('Sport', 1)}*  
• *Note = {('AB56789', 1, 17)}*
3. • *Eleve = {('Titi', 'Toto', 'AB56789')}*  
• *Matiere = {('NSI', 0)}*  
• *Note = {('AB56789', 1, 17)}*
4. • *Eleve = {('Titi', 'Toto', 'AB56789')}*  
• *Matiere = {('NSI', 0)}*  
• *Note = {('AB56789', 0, 17), ('AB56789', 0, 18)}*
5. • *Eleve = {('Titi', 'Toto', 'AB56789')}*  
• *Matiere = {('NSI', 0), ('Sport', 1)}*  
• *Note = {('AB56789', 0, 17), ('AB56789', 1, 17)}*

**Exercice 2** On considère deux relations  $R(a \underline{\text{Int}}, b \underline{\text{Int}}, c \underline{\text{Int}})$  et  $S(\#a \underline{\text{Int}}, e \underline{\text{Int}})$  où l'attribut  $a$  de  $S$  est une clé étrangère faisant référence à  $a$  de  $R$ . Dis si les affirmations suivantes sont vraies ou fausses, en justifiant.

- Les  $a$  de  $R$  sont tous deux à deux distincts.
- Les  $b$  de  $R$  sont tous deux à deux distincts.
- Les  $a$  de  $S$  sont tous deux à deux distincts.
- Les  $e$  de  $S$  sont tous deux à deux distincts.
- $S$  peut être vide alors que  $R$  est non vide.
- $R$  peut être vide alors que  $S$  est non vide.

**Exercice 3** Regroupe ensemble les termes synonymes : colonne, entité, domaine, attribut, schéma, base de données, type, column, row.

**Exercice 4** Reprends la modélisation donnée à l'exercice 1. Donne les ordres SQL permettant de créer les tables correspondantes, avec un maximum de contrainte d'intégrité. Donne les ordres SQL permettant de supprimer ces tables une fois qu'elles existent.

**Exercice 5** On considère les deux tables suivantes :

```
CREATE TABLE joueur( j_id INT PRIMARY KEY,  
                      nom VARCHAR(100) NOT NULL);  
  
CREATE TABLE partie( j1 INT REFERENCES joueur(j_id),  
                     j2 INT REFERENCES joueur(j_id),  
                     score1 INT NOT NULL,  
                     score2 INT NOT NULL,  
                     CHECK ( j1 <> j2 ) );
```

Ces tables stockent des résultats de parties entre des joueurs. Liste toutes les contraintes d'intégrité et pour chacune donne des ordres SQL violant ces contraintes.

**Exercice 6** Modifie les ordres de création de table de l'exercice 5. pour prendre en compte les modifications suivantes :

- La table **partie** contient en plus une colonne **jour** non nulle, indiquant la date à laquelle la partie de jeu à au lieu.
- Les scores ne peuvent pas être négatifs.
- Deux joueurs en peuvent pas jouer plusieurs fois le même jour.