

Corrigé du TD1 : Premières requêtes

Mai/Juin 2023

Exercice 1 : Relevé de notes

Cet exercice est basé sur les relations indiquées page suivante.

Première partie : rétro documentation

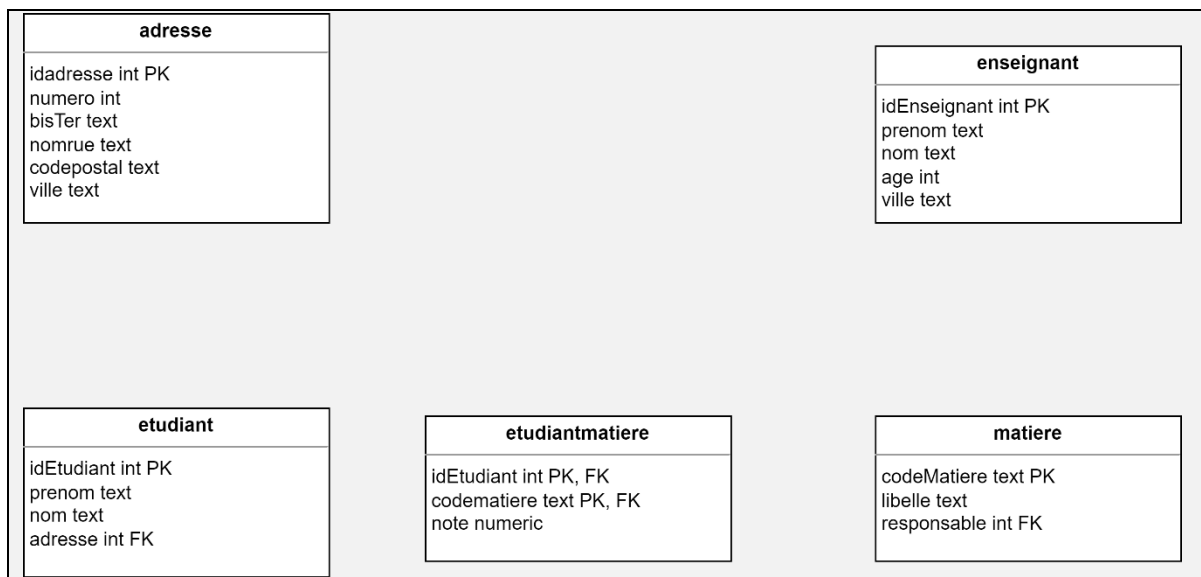
1. A quoi correspondent les champs Adresse(idAdresse) ou matière(codeMatière) ?

Il s'agit des clés primaires. Elles permettent de discriminer chaque ligne dans la table.

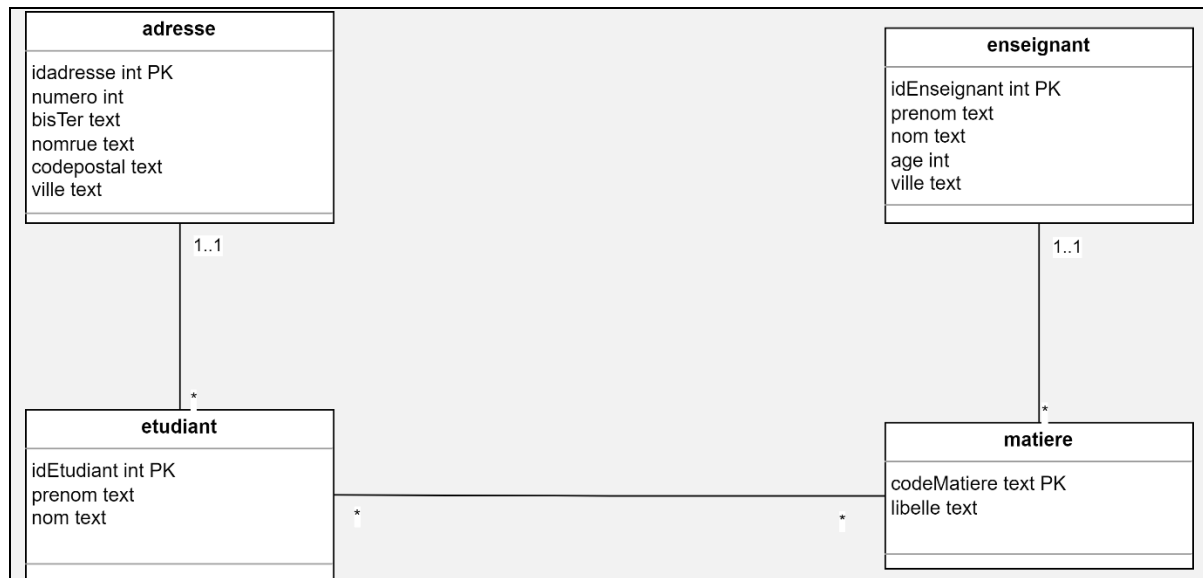
2. A quoi correspondent les champs EtudiantMatiere(idEtudiant) ou Etudiant(idAdresse) ?

Il s'agit de clés étrangères. Il s'agit de la matérialisation d'un lien vers une clé primaire située dans une table distante. C'est ce lien qui permet de faire interagir des données situées dans plusieurs tables.

3. A partir des relations décrites ci-dessus, formalisez le modèle physique de données



4. Déduisez-en le modèle conceptuel



5. Quel est le comportement de la base de données si on souhaite rajouter une note à l'étudiant « 1001 » pour la matière « R207 » ?

La clé primaire de la table etudiantMatiere est composée des deux champs « idEtudiant » et « codeMatiere ». Il ne peut exister qu'une seule occurrence de ce couple de valeurs.

Par conséquent, si on tente de rajouter une nouvelle note avec ce couple de données, l'opération sera refusée par le SGBDR qui va émettre une erreur de violation de clé :

```
INSERT INTO etudiantmatiere (idEtudiant, codeMatiere, note)
VALUES (1001, 'R207', 20) ;
```

ERREUR: la valeur d'une clé dupliquée rompt la contrainte unique « etudiantmatiere_pkey » DETAIL: La clé « (idetudiant, codematiere)=(1001, R207) » existe déjà.

Seconde partie : requêtes SQL simples

1. Ecrivez la déclaration de la table Enseignant en SQL

```
CREATE TABLE enseignant(
    idEnseignant int,
    prenom text,
    Nom text,
    age int,
    Ville text,
    PRIMARY KEY (idEnseignant)
);
```

2. Afficher le numéro et le nom de la rue et la ville de chaque adresse

```
SELECT numero, nomRue, ville FROM adresse ;
```

3. Afficher le nom et prénom de chaque étudiant

```
SELECT nom, prenom FROM etudiant ;
```

4. Afficher le libellé de chaque matière avec le numéro de l'enseignant responsable

```
SELECT libelle, responsable FROM matiere;
```

5. Afficher toutes les informations pour l'étudiant dont le nom est « Lesyeubleu »

```
SELECT * FROM etudiant WHERE nom='Lesyeubleu' ;
```

6. Afficher les numéros d'étudiants qui suivent la matière « R207 »

```
SELECT idEtudiant
FROM etudiantmatiere
WHERE codeMatiere='R207';
```

7. Afficher les noms de rues à St Malo

```
SELECT idEtudiant
FROM etudiantmatiere
WHERE codeMatiere='R207';
```

8. Combien y a-t-il d'enseignants ?

```
SELECT COUNT(*) FROM enseignant ;
```

Troisième partie : requêtes plus complexes

9. Afficher le nom et prénom des étudiants qui suivent les matières « R207 » ou « R115 »

```
-- On doit utiliser DISTINCT pour éviter d'afficher en double les
-- étudiants qui suivent à la fois la matière R207 et R115.
```

```
SELECT DISTINCT etudiant.nom, etudiant.prenom
FROM etudiant INNER JOIN etudiantmatiere USING (idEtudiant)
WHERE etudiantmatiere.codeMatiere='R207'
      OR etudiantmatiere.codeMatiere='R115';
```

```
-- Autre syntaxe :
```

```
SELECT DISTINCT etudiant.nom, etudiant.prenom
FROM etudiant INNER JOIN etudiantmatiere USING (idEtudiant)
WHERE etudiantmatiere.codeMatiere IN ('R207','R115');
```

```
-- Autre syntaxe :
```

```
SELECT DISTINCT etudiant.nom, etudiant.prenom
FROM etudiant, etudiantmatiere
WHERE etudiant.idEtudiant=etudiantmatiere.idEtudiant
      AND etudiantmatiere.codeMatiere IN ('R207','R115');
```

10. Afficher le nom et prénom des étudiants qui suivent les matières « R207 » ou « R115 », en indiquant également le libellé de la matière

```
SELECT DISTINCT etudiant.nom, etudiant.prenom, matiere.libelle
FROM etudiant INNER JOIN etudiantmatiere USING (idEtudiant)
INNER JOIN matiere USING (codeMatiere)
WHERE etudiantmatiere.codeMatiere IN ('R207','R115');
```

11. Afficher toutes les informations pour l'étudiant dont le nom est « Lesyeubleu », en affichant également son adresse

```
SELECT *
FROM etudiant INNER JOIN adresse USING (idAdresse)
WHERE etudiant.nom='Lesyeubleu';
```

12. Afficher le nom, prénom des étudiants, le libellé des matières ainsi que la note obtenue

```
SELECT etudiant.nom, etudiant.prenom,
matiere.libelle, etudiantmatiere.note
FROM etudiant INNER JOIN etudiantmatiere USING (idEtudiant)
INNER JOIN matiere USING (codeMatiere);
```

13. Afficher la note moyenne globale

```
SELECT AVG(note)
FROM etudiantmatiere ;
```

14. Afficher la note moyenne matière par matière

```
SELECT matiere.libelle, AVG(etudiantmatiere.note)
FROM etudiantmatiere INNER JOIN matiere USING (codeMatiere)
GROUP BY matiere.libelle;
```

15. Combien y a-t-il d'enseignants en fonction du nom de la ville ?

```
SELECT ville, COUNT(*)
FROM enseignant
GROUP BY ville ;
```