

1 Modèle relationnel

Planter le Modèle Relationnel suivant :

etudiant(noEtudiant, nomEtudiant, #idGroupe)

groupe(idGroupe, libelleGroupe)

créer un script SQL du type :

```
DROP TABLE IF EXISTS ... ;

DROP TABLE IF EXISTS ... ;

CREATE TABLE ... (

) ;

CREATE TABLE ... (

) ;
```

- Insérer des données

```
INSERT INTO groupe VALUES(1,'S2A1');    -- il id 1
INSERT INTO groupe VALUES(2,'S2A2');    -- id 2
INSERT INTO groupe VALUES(3,'S2bisA1'); -- id 3

INSERT INTO etudiant VALUES(1,'paul',1);
INSERT INTO etudiant VALUES(2,'pierre',2);
INSERT INTO etudiant VALUES(3,'toto',17);
```

Remarque : les valeurs des clés primaires sont imposées mais il est souhaitable de remplacer par la valeur “NULL” par la suite

- On peut créer un étudiant “toto” avec un groupe qui n’existe pas. Il n’y a pas de contrainte d’intégrité. Cette contrainte d’intégrité permet de garantir la cohérence des données.

Article sur toutes les contraintes (plus compliqué).

- Parmi les contraintes d’intégrités on peut citer ;
 - les contraintes structurelles (unicité de clé ; **Contrainte référentielle** ; Contrainte de domaine ; Contrainte de non-nullité)
 - les contraintes non structurelles (dépendances fonctionnelles, ...)

- Cette contrainte d’intégrité référentielle se définit par une(des) clé(s) étrangère(s)

C.I. référentielle : A chaque valeur d’une clé étrangère doit correspondre une valeur de la clé primaire associée

1.1 Définition d’une « clé étrangère »

- Modifier la table **etudiant** avec la contrainte « Foreign Key » et recréer les 2 tables

```
DROP TABLE IF EXISTS etudiant;
CREATE TABLE etudiant (
    noEtudiant INT AUTO INCREMENT,
    nomEtudiant VARCHAR(25),
    idGroupe INT,
    PRIMARY KEY (noEtudiant),
    CONSTRAINT fk_etudiant_groupe
        FOREIGN KEY (idGroupe) REFERENCES groupe(idGroupe)
);
```

- Refaire les commandes « **INSERT** » précédentes.

Faire ATTENTION à l'ordre de suppression et de création des tables quand il y a une contrainte.

- **Dans quel ordre faut-il supprimer ou créer les tables ?**
- Supprimer le groupe créé : `DELETE FROM groupe WHERE idGroupe=1;`
- Modifier un enregistrement : `UPDATE groupe SET libelleGroupe='1D1' WHERE idGroupe=1;`

Conclusion :

- Peut-on créer une contrainte de type référentielle entre un champ « clé étrangère » d'une table et un champ d'une autre table qui n'est pas une clé primaire ?
- Supprimer les 2 mots : “**CONSTRAINT fk_etudiant_groupe**”, MySQL donne-t-il un nom à cette contrainte ? (vérifier avec la commande `SHOW CREATE TABLE etudiant;`)

Un logiciel comme « analyseSI » ou « JMerise » ne sait pas dans quel ordre supprimer ou créer les contraintes sur les clés étrangères, il le fait dans un deuxième temps.

Utiliser les commandes ci-dessous pour supprimer puis remettre la contrainte sur la clé étrangère de la table **etudiant**

```
ALTER TABLE etudiant
    DROP FOREIGN KEY fk_etudiant_groupe;
SHOW CREATE TABLE etudiant;

ALTER TABLE etudiant
    ADD CONSTRAINT fk_etudiant_groupe
        FOREIGN KEY (idGroupe) REFERENCES groupe(idGroupe);
SHOW CREATE TABLE etudiant;
```

Remarques :

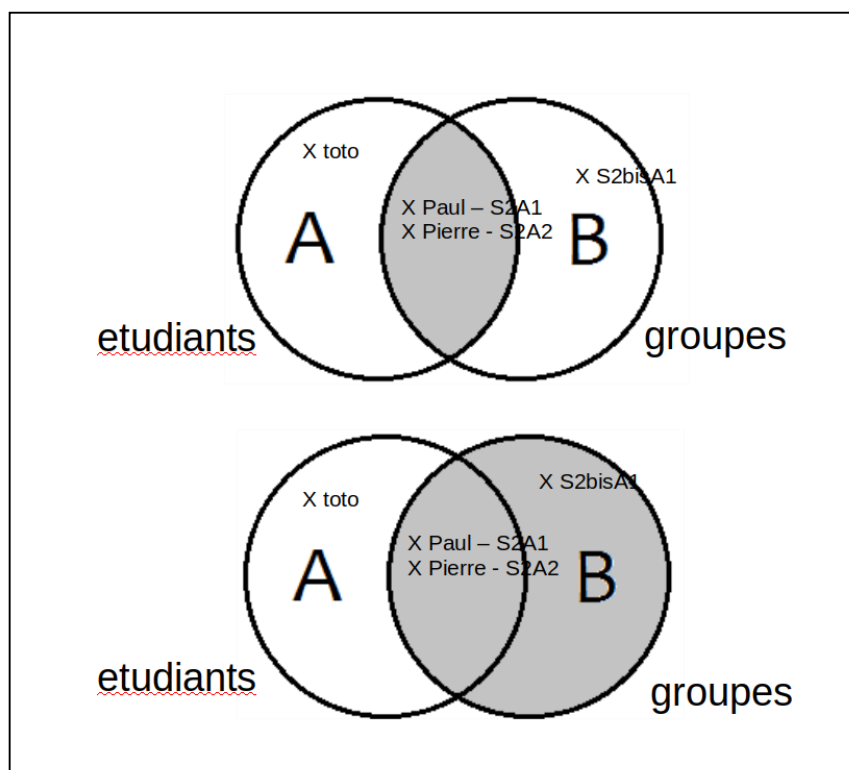
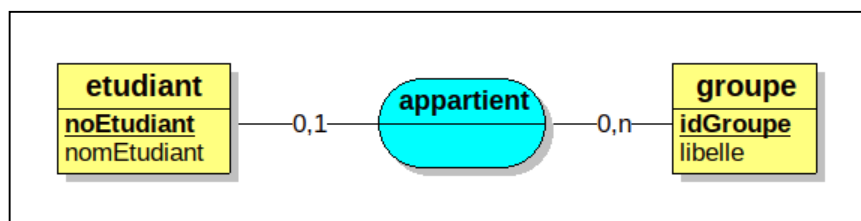
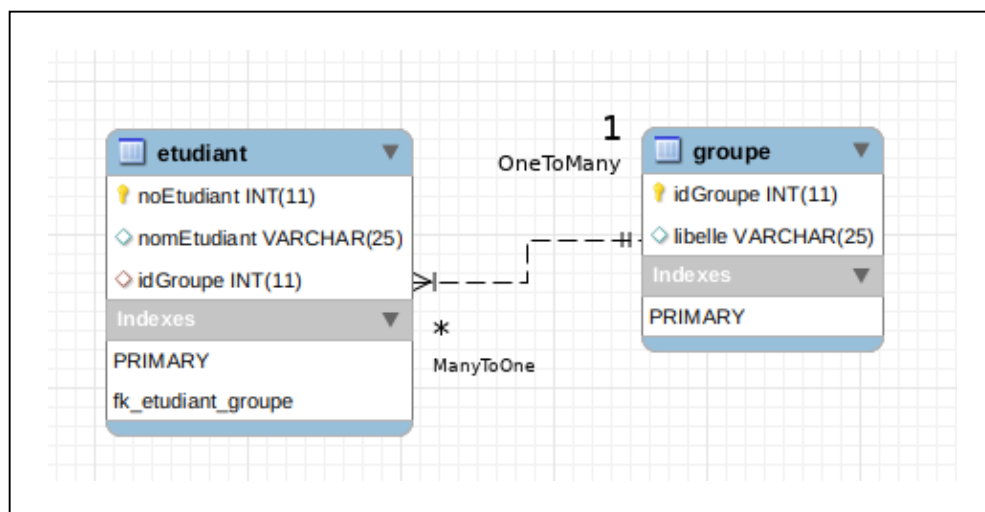
- **Lors de la création des tables** : faire attention à l'ordre, on ne crée pas une table avec une contrainte sur une table qui n'existe pas.
- Si un champ utilise l'attribut « auto_increment », celui-ci doit être une clé primaire.
- **Lors de la suppression des tables** : faire attention à l'ordre, on ne supprime pas une table qui a sa clé primaire utilisée dans une contrainte d'une autre table (clé étrangère dans une autre table).
- **Lors de la création des contraintes** :
 - la table sur laquelle on a une contrainte doit exister
 - les types des 2 champs doivent être identiques (int et unsigned int ne sont pas du même type).

2 Présentation des jointures

L'objectif est d'afficher les étudiants et leur groupe associé :

```
INSERT INTO groupe VALUES(null, 'S2bisA1');
```

- ajouter un groupe :



[Ouvrir ce lien sur Wikipédia qui concerne l'algèbre relationnelle](#)

- Tester les instructions ci-dessous.
- Ajouter en commentaire sous chaque groupe d'instruction le(s) opérateur(s) (Opérateurs relationnels ou ensemblistes) de l'**algèbre relationnelle**.

Les opérateurs sont dans la liste suivante :

Produit cartésien - Sélection – Projection – Jointure

- quelles instructions classeriez vous sous la norme SQL ANSI-92 et sous la norme ANSI-89?

```
SELECT *
FROM etudiant, groupe ;

SELECT *
FROM etudiant, groupe
WHERE
etudiant.idGroupe = groupe.idGroupe;

SELECT etudiant.nomEtudiant, groupe.libell
FROM etudiant, groupe
WHERE
etudiant.idGroupe = groupe.idGroupe;

SELECT *
FROM groupe
INNER JOIN etudiant
ON etudiant.idGroupe = groupe.idGroupe;

SELECT *
FROM groupe
LEFT JOIN etudiant
ON etudiant.idGroupe = groupe.idGroupe;

SELECT *
FROM etudiant
LEFT JOIN groupe
ON etudiant.idGroupe = groupe.idGroupe;

SELECT *
FROM groupe
RIGHT JOIN etudiant
ON etudiant.idGroupe = groupe.idGroupe;

SELECT *
FROM etudiant
RIGHT JOIN groupe
ON etudiant.idGroupe = groupe.idGroupe;
```

3 Exercice

Modifier le script ci-dessous :

- ajouter les clés étrangères
- modifier le script pour supprimer puis ajouter les tables dans le bonne ordre

```
DROP TABLE IF EXISTS Etudiant, xxxx, xxxx, xxxx, xxxx;

CREATE TABLE Etudiant (
  idEtudiant INT AUTO_INCREMENT
, nom VARCHAR(20)
, adresse VARCHAR(40)
, ville VARCHAR(20)
, code_postal VARCHAR(5)
, telephone VARCHAR(10)
, date_naissance DATE
```

```

, sexe VARCHAR(1)
, boursier VARCHAR(1)
, groupe_id INT
, PRIMARY KEY(idEtudiant)
);

CREATE TABLE etudiant_periode (
    etudiant_id INT,
    periode_id INT,
    semestre_id INT,
    PRIMARY KEY (etudiant_id,periode_id,semestre_id)
);

CREATE TABLE Groupe (
    id_groupe INT AUTO_INCREMENT,
    libelle VARCHAR(255),
    PRIMARY KEY (id_groupe)
);

CREATE TABLE Semestre (
    id_semestre INT AUTO_INCREMENT,
    libelle VARCHAR(255),
    PRIMARY KEY (id_semestre)
);

CREATE TABLE Periode (
    id_periode INT AUTO_INCREMENT,
    date_debut DATE,
    date_fin DATE,
    PRIMARY KEY (id_periode)
);

-- 2022
INSERT INTO Etudiant (idEtudiant, nom, adresse, ville, code_postal, telephone, date_naissance, sexe, boursier,
    groupe_id) VALUES
(1, 'BERNARD', '1 rue sous bois', 'Belfort', '90000', '0384545401', '2003-01-01', 'F', 'N', 1),
(19, 'CHAVEAUX', '5 rue du chasseur', 'Strasbourg', '68000', '0384545419', '2003-01-01', 'H', 'N', 2),
(20, 'PRETTOT', '8 rue vilapogo', 'Belfort', '90000', '0384545420', '2003-07-12', 'H', 'O', 2),
(7, 'RIOT', '67 rue pasteur', 'Montbeliard', '25200', '0384545407', '2003-06-06', 'H', 'N', 2),
(8, 'BOISSENIN', '1 rue sous bois', 'Belfort', '90000', '0384545408', '2003-08-09', 'H', 'O', 2),
(2, 'PEQUIGNOT', '2 rue de la liberation', 'Valdoie', '90300', '0384545402', '2002-06-06', 'H', 'O', 1),
(3, 'ZILLIOX', '7 rue du verger', 'Bavilliers', '90120', '0384545403', '2002-01-01', 'H', 'N', 1),
(4, 'MONNIER', '3 rue du boulanger', 'TAILLECOURT', '25400', '0384545404', '2002-02-06', 'H', 'O', 1),
(5, 'BRISCHOUX', '5 rue du chasseur', 'Belfort', '90000', '0384545405', '2002-05-25', 'H', 'N', 2),
(6, 'DUVAL', '8 rue vilapogo', 'Bavilliers', '90120', '0384545406', '2002-03-11', 'H', 'O', 2),
(10, 'FAIVRE', '7 rue des vergers de rioz', 'Valdoie', '90300', '0384545410', '2002-01-05', 'H', 'O', 3),
(17, 'DELANOE', '7 rue du verger', 'Valdoie', '90300', '0384545417', '2002-01-01', 'H', 'N', 1),
(18, 'BONVALOT', '3 rue du boulanger', 'Belfort', '90000', '0384545418', '2002-01-01', 'H', 'O', 1),
(21, 'COULON', '67 rue pasteur', 'Valdoie', '90300', '0384545421', '2002-01-01', 'H', 'N', 2),
(23, 'KENDE', '2 rue de la liberation', 'Bavilliers', '90120', '0384545423', '2002-05-31', 'H', 'N', 3),
(24, 'KLEIN', '7 rue du verger', 'Montbeliard', '25200', '0384545424', '2002-01-01', 'H', 'N', 3),
(25, 'VALZER', '3 rue du boulanger', 'Valdoie', '90300', '0384545425', '2002-01-06', 'H', 'O', 3),
(26, 'PY', '5 rue du chasseur', NULL, '90000', '0384545426', '2002-01-10', 'F', 'N', 3),
(27, 'VERNET', '8 rue vilapogo', NULL, '90120', '0384545427', '2002-02-02', 'H', 'O', 4),
(28, 'BAILLIT', '67 rue pasteur', NULL, '25200', '0384545428', '2002-01-01', 'H', 'N', 4),
(16, 'LUZET', '2 rue de la liberation', 'Belfort', '90000', '0384545416', '2001-01-01', 'H', 'O', 1),
(22, 'VALOT', '1 rue sous bois', '', '90000', '0384545422', '2001-12-12', 'H', 'O', 2),
(29, 'DUPONT', '8 rue vilapogo', '', '90300', '0384545429', '2001-06-06', 'H', 'O', 4),
(11, 'FAIVRE', '3 rue des vergers', 'Cernay', '68000', '0384545411', '2001-01-01', 'F', 'N', 3),
(12, 'DUCHENNE', '5 rue du chasseur', 'Belfort', '90000', '0384545412', '2001-01-01', 'F', 'O', 3),
(13, 'BOULANGER', '8 rue vilapogo', 'Belfort', '90000', '0384545413', '2001-01-01', 'F', 'N', 1),
(9, 'FONTAINE-LEGIOT', '2 rue des vergers', 'Mulhouse', '68000', '0384545409', '2000-01-01', 'H', 'N', 3),
(14, 'MOREAU', '67 rue pasteur', 'Belfort', '90000', '0384545414', '2000-06-01', 'H', 'O', 1),
(15, 'RIGOULOT', '1 rue sous bois', 'Valdoie', '90300', '0384545415', '2000-12-12', 'H', 'N', 1);

INSERT INTO etudiant_periode VALUES (1,1,1),(19,1,2),(20,1,2),(7,1,1),(8,1,1),(2,1,1),(3,1,1),(4,1,1),(5,1,1),
(6,1,1),(10,1,1),(17,1,2),(18,1,2),(21,1,2),(23,1,2),(24,1,2),(25,1,2),(26,1,2),(27,1,3),(28,1,3),(16,1,2),
(22,1,2),(29,1,3),(11,1,1),(12,1,1),(13,1,2),(9,1,1),(14,1,2),(15,1,2);

INSERT INTO Semestre VALUES (NULL, 'S1'), (NULL, 'S2'), (NULL, 'S3'), (NULL, 'S4'), (NULL, 'S5'), (NULL, 'S6');

INSERT INTO Groupe VALUES (NULL, 'A1'), (NULL, 'A2'), (NULL, 'B1'), (NULL, 'B2'), (NULL, 'C1'), (NULL, 'C2'), (NULL, 'D1');

```

```
INSERT INTO Periode VALUES (NULL, '2022-09-01', '2023-01-15'), (NULL, '2023-1-15', '2023-7-1'),
                             (NULL, '2023-09-01', '2024-01-15'), (NULL, '2024-1-15', '2024-7-1'),
                             (NULL, '2024-09-01', '2025-01-15'), (NULL, '2025-1-15', '2025-7-1');
```

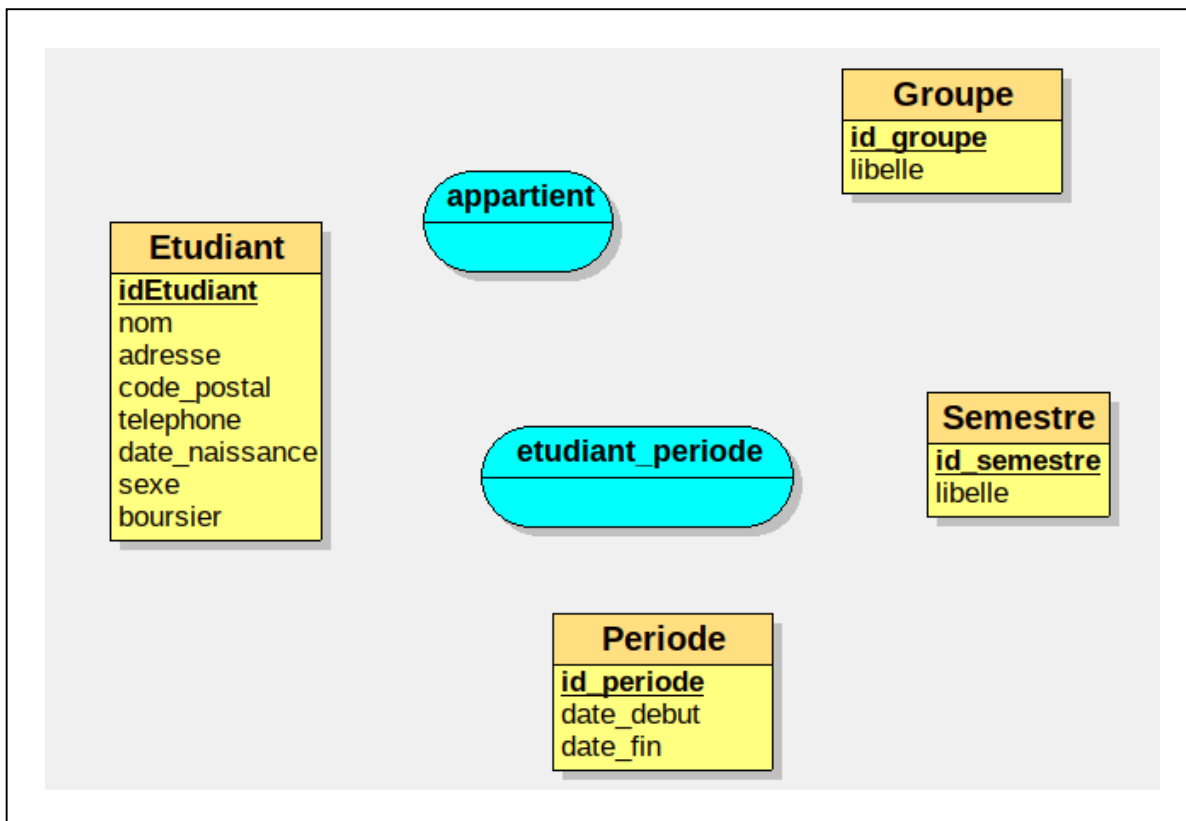
- Réaliser la requête ci-dessous : afficher le nombre d'étudiants dans chaque semestre

Nbre_etudiants	libelle
12	S1
14	S2
3	S3

- Modifier la requête pour afficher les semestres qui n'ont pas d'étudiants

Nbre_etudiants	libelle
12	S1
14	S2
3	S3
0	S4
0	S5
0	S6

4 Réflexion sur le schéma



utiliser le fichier [looping ci-joint](#) et modifier le pour obtenir le bon MLD

- Le schéma réalisé est très discutable, pourquoi ?
- Comment pourrait on ajouter des notes ?

ANNEXE :

```
-- ATTENTION à l'ordre de création et de suppression

DROP TABLE IF EXISTS groupe;
DROP TABLE IF EXISTS etudiant;

CREATE TABLE groupe (
    idGroupe int AUTO_INCREMENT
    , libelle varchar(25)
    , PRIMARY KEY (idGroupe)
);
SHOW CREATE TABLE groupe;

CREATE TABLE etudiant (
    noEtudiant int AUTO_INCREMENT
    , nomEtudiant varchar(25)
    , idGroupe int
    , PRIMARY KEY (noEtudiant)
    , CONSTRAINT fk_etudiant_groupe
        FOREIGN KEY (idGroupe)
        REFERENCES groupe(idGroupe)
);
SHOW CREATE TABLE etudiant;

INSERT INTO groupe VALUES(null,'S2A1'); -- id 1
INSERT INTO groupe VALUES(null,'S2A2'); -- id 2
INSERT INTO groupe VALUES(null,'S2bisA1'); -- id 3
SELECT * from groupe;

INSERT INTO etudiant VALUES(null,'paul',1);
INSERT INTO etudiant VALUES(null,'pierre',2);
-- INSERT INTO etudiant VALUES(null,'toto',17); -- erreur 0k
SELECT * from etudiant;
-- rôle de AUTO_INCREMENT=10

ALTER TABLE etudiant DROP FOREIGN KEY fk_etudiant_groupe;
SHOW CREATE table etudiant;
ALTER TABLE etudiant ADD FOREIGN KEY fk_etudiant_groupe (idGroupe) REFERENCES groupe(idGroupe);
SHOW CREATE table etudiant;
```