

Cours SQL : Les Jointures

1. **Contexte** : Gestion des affectations des étudiants aux promotions de cours.

Un étudiant décrit par un nom, prénom, email est affecté à une promotion identifiée par un nom, diplôme et une salle.

- **Création du MCD :**
- **Elaboration du MLDR : application des règles 1 et 2**

Promotion (idpromotion, nom, diplome, salle)

Etudiant (idetudiant, nom, prenom, email, idpromotion #)

- **Script de la Base de données**

```
create database cfa_266_23 ;
use cfa_266_23 ;

create table promotion (
    idpromotion int(3) not null auto_increment,
    nom varchar(50),
    diplome varchar(50),
    salle varchar(50),
    primary key (idpromotion)
);

create table etudiant (
    idetudiant int(3) not null auto_increment,
    nom varchar(50),
    prenom varchar(50),
    email varchar(50),
    idpromotion int(3) not null,
    primary key(idetudiant),
    foreign key(idpromotion) references promotion (idpromotion)
);
```

- **Insertion des données**

```
insert into promotion values (null, "Promo 266", "BTS SIO SLAM", "Salle 1.D");
insert into promotion values (null, "Promo 267", "BTS SIO SISR", "Salle 4.D");
insert into etudiant values (null, "Mohamed", "Lucas", "a@gmail.com", 1);
insert into etudiant values (null, "Paul", "Olivier", "p@gmail.com", 1);
insert into etudiant values (null, "Marie", "Julia", "m@gmail.com", 2);
insert into etudiant values (null, "Marine", "Garance", "g@gmail.com", 2);
```

Résultats des insertions :

```
[mysql> select * from promotion;
```

idpromotion	nom	diplome	salle
1	Promo 266	BTS SIO SLAM	Salle 1.D
2	Promo 267	BTS SIO SISR	Salle 4.D

```
2 rows in set (0,01 sec)
```

```
[mysql> select * from etudiant ;
```

idetudiant	nom	prenom	email	idpromotion
1	Mohamed	Lucas	a@gmail.com	1
2	Paul	Olivier	p@gmail.com	1
3	Marie	Julia	m@gmail.com	2
4	Marine	Garance	g@gmail.com	2

```
4 rows in set (0,00 sec)
```

- Requêtes de jointure entre les deux tables :

Afficher la liste des étudiants (nom, prénom, email) avec le nom de leur promotion et la salle de cours.

```
select etudiant.nom, etudiant.prenom, etudiant.email, promotion.nom, promotion.salle
from etudiant, promotion
where promotion.idpromotion = etudiant.idpromotion ;
```

```
mysql> select etudiant.nom, etudiant.prenom, etudiant.email, promotion.nom, promotion.salle
-> from etudiant, promotion
-> where promotion.idpromotion = etudiant.idpromotion ;
```

nom	prenom	email	nom	salle
Mohamed	Lucas	a@gmail.com	Promo 266	Salle 1.D
Paul	Olivier	p@gmail.com	Promo 266	Salle 1.D
Marie	Julia	m@gmail.com	Promo 267	Salle 4.D
Marine	Garance	g@gmail.com	Promo 267	Salle 4.D

```
4 rows in set (0,09 sec)
```

On peut améliorer le résultat de la requête en utilisant des alias :

```
select etudiant.nom, etudiant.prenom, etudiant.email, promotion.nom as promotion,
promotion.salle
from etudiant, promotion
where promotion.idpromotion = etudiant.idpromotion ;
```

```
mysql> select etudiant.nom, etudiant.prenom, etudiant.email, promotion.nom as promotion, promotion.salle
-> from etudiant, promotion
-> where promotion.idpromotion = etudiant.idpromotion ;
```

nom	prenom	email	promotion	salle
Mohamed	Lucas	a@gmail.com	Promo 266	Salle 1.D
Paul	Olivier	p@gmail.com	Promo 266	Salle 1.D
Marie	Julia	m@gmail.com	Promo 267	Salle 4.D
Marine	Garance	g@gmail.com	Promo 267	Salle 4.D

```
4 rows in set (0,01 sec)
```

On peut réduire le texte de la requête en créant des alias sur les noms des tables :

```
select e.nom, e.prenom, e.email, p.nom as promotion, p.salle
from etudiant e, promotion p
where p.idpromotion = e.idpromotion ;
```

```
mysql> select e.nom, e.prenom, e.email, p.nom as promotion, p.salle
-> from etudiant e, promotion p
-> where p.idpromotion = e.idpromotion ;
```

nom	prenom	email	promotion	salle
Mohamed	Lucas	a@gmail.com	Promo 266	Salle 1.D
Paul	Olivier	p@gmail.com	Promo 266	Salle 1.D
Marie	Julia	m@gmail.com	Promo 267	Salle 4.D
Marine	Garance	g@gmail.com	Promo 267	Salle 4.D

```
4 rows in set (0,01 sec)
```

Série de requêtes de jointure :

- **Afficher la liste des étudiants qui sont dans la salle Salle 1.D.**

```
Select e.nom, e.prenom, e.email
from etudiant e, promotion p
where p.idpromotion = e.idpromotion
and p.salle = "Salle 1.D" ;
```

```
Select e.nom, e.prenom, e.email
from etudiant e, promotion p
where p.idpromotion = e.idpromotion
and p.salle like "Salle 1%" ;
```

```
mysql> Select e.nom, e.prenom, e.email
-> from etudiant e, promotion p
-> where p.idpromotion = e.idpromotion
-> and p.salle = "Salle 1.D" ;
```

nom	prenom	email
Mohamed	Lucas	a@gmail.com
Paul	Olivier	p@gmail.com

```
2 rows in set (0,08 sec)
```

- **Afficher la liste des étudiants (nom, prénom) qui prépare un diplôme de BTS.**

```
select e.nom, e.prenom
from etudiant e, promotion p
where p.idpromotion = e.idpromotion
and p.diplome like "BTS%";
```

```
mysql> select e.nom, e.prenom
-> from etudiant e, promotion p
-> where p.idpromotion = e.idpromotion
-> and p.diplome like "BTS%";
```

nom	prenom
Mohamed	Lucas
Paul	Olivier
Marie	Julia
Marine	Garance

4 rows in set (0,13 sec)

- **Afficher le nom de la promotion et la salle de cours de l'étudiant ayant pour mail : a@gmail.com.**

```
select p.nom, p.salle from etudiant e , promotion p
where p.idpromotion = e.idpromotion
and e.email = "a@gmail.com";
```

```
mysql> select p.nom, p.salle from etudiant e , promotion p
-> where p.idpromotion = e.idpromotion
-> and e.email = "a@gmail.com";
```

nom	salle
Promo 266	Salle 1.D

1 row in set (0,12 sec)

- **Quel est le nombre d'étudiants qui sont dans la salle Salle 1.D ?**

```
select count(e.idetudiant) as nbEtudiants
from promotion p, etudiant e
where p.idpromotion = e.idpromotion
and p.salle like "Salle 1%";
```

```
mysql> select count(e.idetudiant) as nbEtudiants
-> from promotion p, etudiant e
-> where p.idpromotion = e.idpromotion
-> and p.salle like "Salle 1%";
```

nbEtudiants
2

1 row in set (0,01 sec)