

# Création de base de donnée

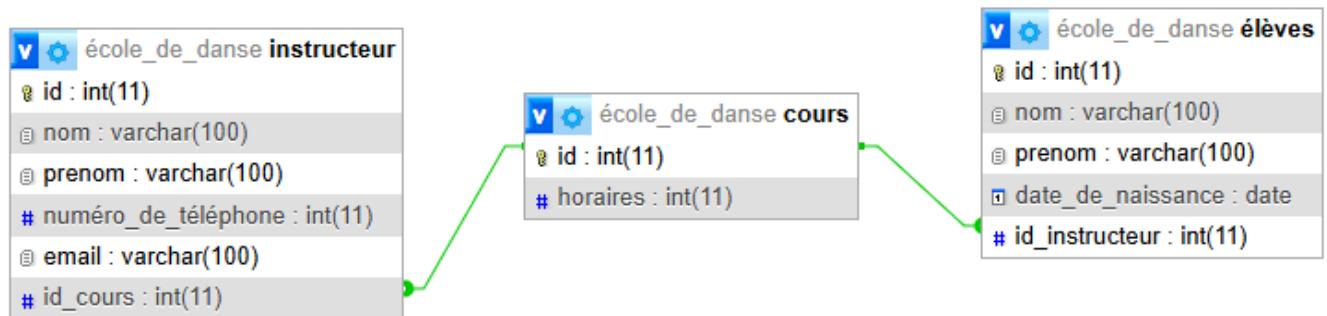


## Sommaire :

<b>Contexte :.....</b>	<b>3</b>
<b>Création de base de donné :.....</b>	<b>4</b>
<b>Insertion d'un jeu d'essai :.....</b>	<b>5</b>
<b>Requête de test :.....</b>	<b>6</b>
<b>Création de vue:.....</b>	<b>8</b>
<b>Création d'utilisateur :.....</b>	<b>9</b>

## Contexte :

J'ai choisi de prendre l'exemple d'une école de danse, les trois acteurs majeurs sont donc (cours, élèves et instructeur). Les élèves auront un id de cours attribuer, les instructeurs auront un id de cours attribuer aussi. Cela donnera quelque chose comme ça:



## Création de base de donn  :

Cette commande permet de cr e la base de donn e

```
Create DATABASE  cole_de_danse ;
```

Les commandes suivantes permettent de créer des tables avec des attributs

```
Create table cours (
    id int primary key,
    horaires int)
engine=INNODB;
```

```
create table élèves (
    id int PRIMARY key,
    nom varchar(100),
    prenom varchar(100),
    date_de_naissance date,
    id_instructeur int,
    foreign key (id_instructeur)REFERENCES cours(id) on delete cascade )
engine=INNODB;
```

```
CREATE TABLE instructeur (
    id INT PRIMARY KEY,
    nom varchar(100),
    prenom VARCHAR(100),
    numéro_de_téléphone INT,
    email VARCHAR(100),
    id_cours INT,
    FOREIGN KEY (id_cours) REFERENCES cours(id) ON DELETE CASCADE
) ENGINE=InnoDB;
```

## Insertion d'un jeu d'essai :

Le but d'insérer un jeu d'essai est de pouvoir tester le liens entre les tables et leur bon fonctionnement

Voici les commandes utilisées :

```
INSERT INTO cours (id, horaires) VALUES
(1, 10),
(2, 14),
(3, 18);
```

```
INSERT INTO instructeur (id, nom, prenom, numéro_de_téléphone, email, id_cours)
VALUES
(1, 'Durand', 'Sophie', 123456789, 'sophie.durand@email.com', 1),
(2, 'Martin', 'Lucas', 987654321, 'lucas.martin@email.com', 2),
(3, 'Bernard', 'Emma', 112233445, 'emma.bernard@email.com', 3);
```

```
INSERT INTO élèves (id, nom, prenom, date_de_naissance, id_instructeur) VALUES  
(1, 'Lemoine', 'Alice', '2010-05-12', 1),  
(2, 'Dupont', 'Thomas', '2009-08-23', 2),  
(3, 'Morel', 'Julie', '2011-03-17', 3),  
(4, 'Garcia', 'Nathan', '2010-11-05', 1),  
(5, 'Petit', 'Laura', '2008-06-30', 2);
```

## Requête de test :

Pour vérifier le fonctionnement de la base avec les données que nous avons entre, nous allons effectuer plusieurs commandes de test et vérifier le résultat.

### Test1:

```
SELECT élèves.nom, élèves.prenom , instructeur.nom  
FROM élèves  
INNER JOIN instructeur ON élèves.id_instructeur = instructeur.id;
```

### Résultat 1:

nom	prenom	nom
Lemoine	Alice	Durand
Dupont	Thomas	Martin
Morel	Julie	Bernard
Garcia	Nathan	Durand
Petit	Laura	Martin

### Test2:

```
SELECT instructeur.nom, instructeur.prenom, cours.horaires  
FROM instructeur  
LEFT JOIN cours ON instructeur.id_cours = cours.id;
```

### Résultat 2:

nom	prenom	horaires
Durand	Sophie	10
Martin	Lucas	14
Bernard	Emma	18

Test3:

```
SELECT *
FROM élèves WHERE date_de_naissance > '2010-01-01';
```

Résultat 3:

id	nom	prenom	date_de_naissance	id_instructeur
1	Lemoine	Alice	2010-05-12	1
3	Morel	Julie	2011-03-17	3
4	Garcia	Nathan	2010-11-05	1

Test4:

```
SELECT *
FROM instructeur ORDER BY nom ASC;
```

Résultat 4:

id	nom	1	prenom	numéro_de_téléphone	email	id_cours
3	Bernard		Emma	112233445	emma.bernard@email.com	3
1	Durand		Sophie	123456789	sophie.durand@email.com	1
2	Martin		Lucas	987654321	lucas.martin@email.com	2

Test5:

```
SELECT instructeur.nom, COUNT(élèves.id) AS NombreÉlèves
FROM instructeur
LEFT JOIN élèves ON instructeur.id = élèves.id_instructeur
GROUP BY instructeur.nom;
```

Résultat 5:

nom	NombreÉlèves
Bernard	1
Durand	2
Martin	2

Test6:

```
SELECT COUNT(*) AS TotalÉlèves FROM élèves;
```

Résultat 6:

TotalÉlèves
5

Test7:

```
SELECT AVG(YEAR(CURDATE()) - YEAR(date_de_naissance)) AS AgeMoyen FROM
élèves;
```

Résultat 7:

AgeMoyen  
15.4000

Test8:

```
SELECT * FROM élèves
WHERE id_instructeur = (
    SELECT id_instructeur FROM élèves WHERE prenom = 'Alice' LIMIT 1
);
```

Résultat 8:

id	nom	prenom	date_de_naissance	id_instructeur
1	Lemoine	Alice	2010-05-12	1
4	Garcia	Nathan	2010-11-05	1

## Création de vue:

Une **vue** est une requête SQL enregistrée qui permet d'afficher un ensemble de données de manière simplifiée et réutilisable. Elle est particulièrement utile pour faciliter l'accès aux informations sans avoir à réécrire des requêtes complexes à chaque consultation.

### Exemple :

La vue suivante affiche un résumé des instructeurs, avec leurs horaires de cours et le nombre d'élèves qu'ils encadrent :

```
CREATE VIEW Vue_Statistiques_Instructeurs AS
SELECT
    instructeur.nom AS NomInstructeur,
    instructeur.prenom AS PrénomInstructeur,
    cours.horaires AS HoraireCours,
    COUNT(élèves.id) AS NombreÉlèves
FROM instructeur
LEFT JOIN cours ON instructeur.id_cours = cours.id
LEFT JOIN élèves ON instructeur.id = élèves.id_instructeur
GROUP BY instructeur.id, cours.horaires;
```

NomInstructeur	PrénomInstructeur	HoraireCours	NombreÉlèves
Durand	Sophie	10	2
Martin	Lucas	14	2
Bernard	Emma	18	1

## Création d'utilisateur :

Voici la commande pour créer un utilisateur "instructeur" qui peut ajouter ou modifier des données dans la table

```
CREATE USER 'instructeur'@'localhost' IDENTIFIED BY '123soleil';
GRANT SELECT, INSERT, UPDATE ON ecole_de_danse.élèves TO 'instructeur'@'localhost';
GRANT SELECT, INSERT, UPDATE ON ecole_de_danse.cours TO 'instructeur'@'localhost';
```

Voici la commande pour créer un utilisateur "élèves" qui peut seulement lire les données

```
CREATE USER 'eleve'@'localhost' IDENTIFIED BY '123soleil';
GRANT SELECT ON ecole_de_danse.élèves TO 'eleve'@'localhost';
GRANT SELECT ON ecole_de_danse.cours TO 'eleve'@'localhost';
```