

Bases de données et SQL

Ambre Le Berre

2025/2026

Agenda

1. Introduction et vocabulaire
2. Requêtes simples sur les bases de données
3. Opérations sur plusieurs tables
4. Agrégats

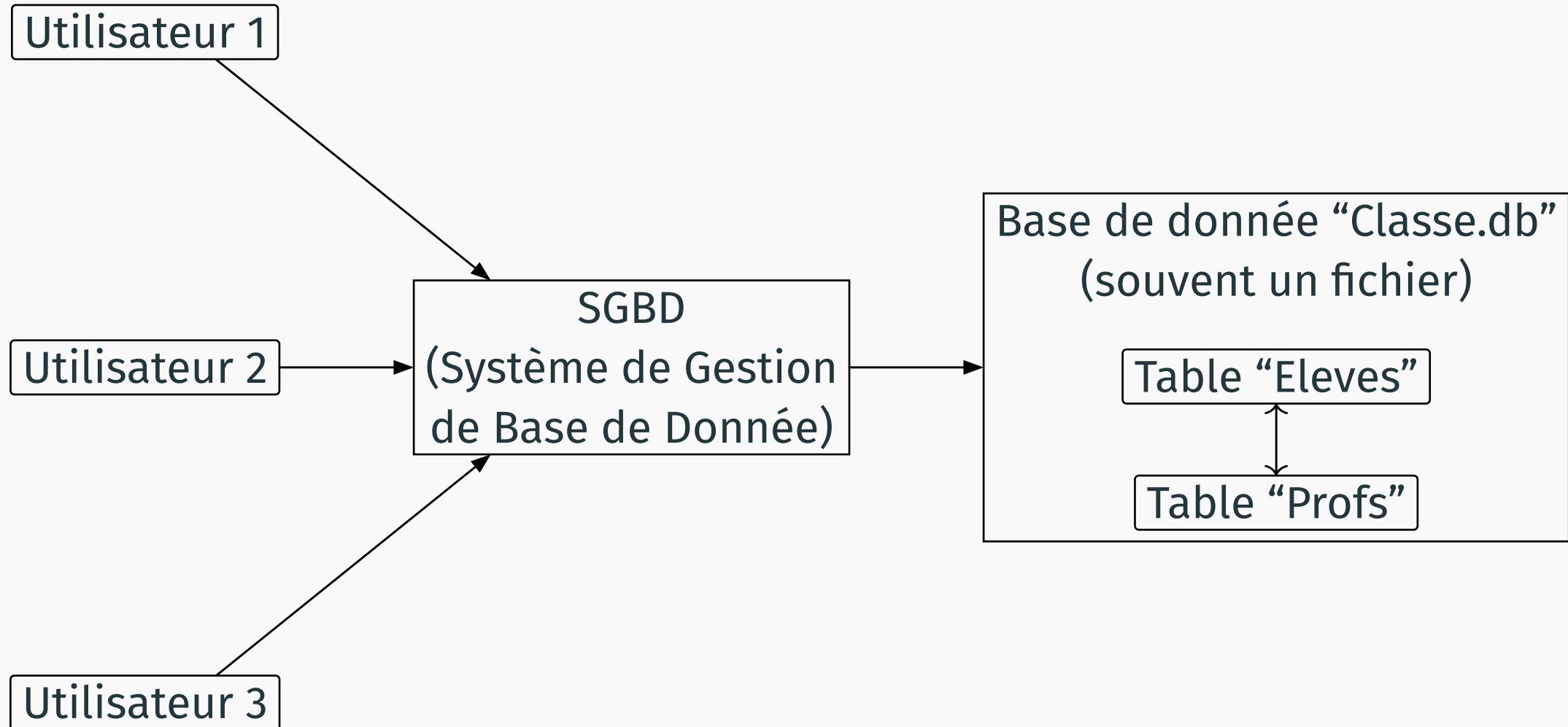
Introduction et vocabulaire

Introduction

Les bases de données servent à ... **stocker des données.**

- Stocker beaucoup de données
- Pouvoir y accéder et faire des opération dessus rapidement

SGBD



Exemple de table (ou entité)

Entité “Eleves”

Prenom	Nom	Classe	Age	Ville
Steph	ANE	MPSI	17	Paris
Marc	IMBUT	PCSI	18	Marseille
Jo	NID	MPSI	19	Montpellier
Rayan	AIR	MP	19	Paris
Tom	DESAVOIE	PCSI	18	Caen
Jerry	CANE	PSI	19	Paris
Paul	AINE	MP	19	Marseille
Sam	EDI	MP	20	Caen
Jean	NAIMAR	PCSI	18	Toulouse
Sam	OURAIL	PC	20	Paris

Exemple de table (ou entité)

Prenom	Nom	Classe	Age	Ville
Steph	ANE	MPSI	17	Paris
Marc	IMBUT	PCSI	18	Marseille
Jo	NID	MPSI	19	Montpellier
Rayan	AIR	MP	19	Paris
Tom	DESAVOIE	PCSI	18	Caen
Jerry	CANE	PSI	19	Paris
Paul	AINE	MP	19	Marseille
Sam	EDI	MP	20	Caen
Jean	NAIMAR	PCSI	18	Toulouse
Sam	OURAIL	PC	20	Paris

← Enregistrement (n-uplet)

Exemple de table (ou entité)

Prenom	Nom	Classe	Age	Ville	← Attribut
Steph	ANE	MPSI	17	Paris	
Marc	IMBUT	PCSI	18	Marseille	← Enregistrement (n-uplet)
Jo	NID	MPSI	19	Montpellier	
Rayan	AIR	MP	19	Paris	
Tom	DESAVOIE	PCSI	18	Caen	
Jerry	CANE	PSI	19	Paris	
Paul	AINE	MP	19	Marseille	
Sam	EDI	MP	20	Caen	
Jean	NAIMAR	PCSI	18	Toulouse	
Sam	OURAIL	PC	20	Paris	

Domaine d'un Attribut : valeurs possibles qu'il peut prendre. Dans notre cas, ce sera entier, flottant, ou chaîne de caractères.

Exemple de table (ou entité)

<u>Prenom</u>	<u>Nom</u>	Classe	Age	Ville	← Attribut
Steph	ANE	MPSI	17	Paris	
Marc	IMBUT	PCSI	18	Marseille	← Enregistrement (n-uplet)
Jo	NID	MPSI	19	Montpellier	
Rayan	AIR	MP	19	Paris	
Tom	DESAVOIE	PCSI	18	Caen	
Jerry	CANE	PSI	19	Paris	
Paul	AINE	MP	19	Marseille	
Sam	EDI	MP	20	Caen	
Jean	NAIMAR	PCSI	18	Toulouse	
Sam	OURAIL	PC	20	Paris	

←
Clé Primaire = identifiant unique d'un enregistrement

Clé étrangère

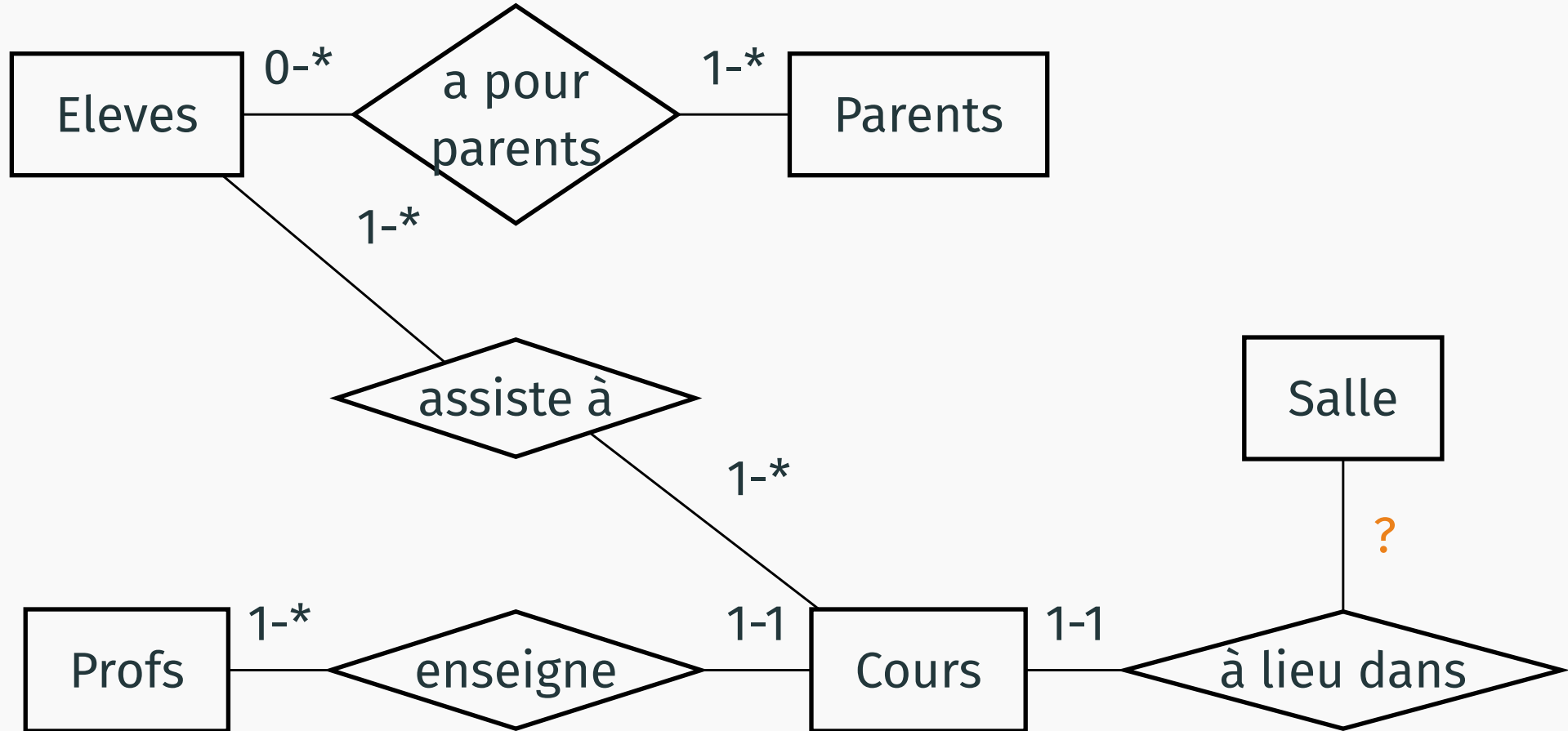
Clé étrangère

<u>Prenom</u>	<u>Nom</u>	<u>Classe</u>	Age	Ville
Sam	EDI	MP	20	Caen
Jean	NAIMAR	PCSI	18	Toulouse
Sam	OURAIL	PC	20	Paris

<u>Titre</u>	<u>Prenom</u>	<u>Nom</u>	<u>Classe</u>	<u>Salle</u>
Mme	Sylvie	DEFOU	MP	B18
Mr	Georges	AITTE	PC	C15
Mme	Marie	AGE	PSI	E8

Modèle Entité-Association

Permet de représenter un schéma de base de donnée.



Requêtes simples sur les bases de données

Le langage SQL



- Language simple mais puissant
- Permet d'accéder, de trier, de regrouper, de croiser ... Les données dans une base de donnée

Projection

La première opération sur les bases de données est la **projection**. Cela permet de sélectionner seulement certains attributs (colonnes) d'une table.

Projection

La première opération sur les bases de données est la **projection**.
Cela permet de sélectionner seulement certains attributs (colonnes) d'une table.

```
SELECT * FROM Eleves
```

Projection

La première opération sur les bases de données est la **projection**. Cela permet de sélectionner seulement certains attributs (colonnes) d'une table.

```
SELECT * FROM Eleves
```

```
SELECT Prenom, Nom FROM Eleves
```


Projection : options

On peut ajouter plusieurs options aux projections :

Projection : options

On peut ajouter plusieurs options aux projections :

- Renommer des colonnes

```
SELECT Prenom AS P FROM Eleves
```

Projection : options

On peut ajouter plusieurs options aux projections :

- Renommer des colonnes

```
SELECT Prenom AS P FROM Eleves
```

- Trier les données

```
SELECT * FROM Eleves ORDER BY Age DESC
```

Projection : options

On peut ajouter plusieurs options aux projections :

- Renommer des colonnes

```
SELECT Prenom AS P FROM Eleves
```

- Trier les données

```
SELECT * FROM Eleves ORDER BY Age DESC
```

- Afficher seulement certaines données

```
SELECT * FROM Eleves LIMIT 3 OFFSET 2
```

Projection : options

On peut ajouter plusieurs options aux projections :

- Renommer des colonnes

```
SELECT Prenom AS P FROM Eleves
```

- Trier les données

```
SELECT * FROM Eleves ORDER BY Age DESC
```

- Afficher seulement certaines données

```
SELECT * FROM Eleves LIMIT 3 OFFSET 2
```

- Supprimer les doublons

```
SELECT DISTINCT Age FROM Eleves
```

Si la projection permettait de sélectionner des colonnes, la sélection, elle, permet de sélectionner des enregistrements, soit des lignes.

```
SELECT * FROM Eleves WHERE Age = 19
```

Opérateurs de sélection

Les opérateurs de sélection sont les suivants :

- = et ! = pour l'égalité
- >, <, ≥, ≤ pour les comparaisons
- AND, OR et NOT pour combiner des opérateurs

Ecrivez des requêtes qui :

- Renvoie le nom et prenom des élèves de MP qui ont moins de 20 ans.
-

Ecrivez des requêtes qui :

- Renvoie le nom et prenom des élèves de MP qui ont moins de 20 ans.
- Renvoie la liste des ages des élèves de spé, sans doublons.

Opérations sur plusieurs tables

Jointures

Les jointures permettent de croiser les données de plusieurs table, en général en fonction d'une clé étrangère.

Entité "Eleves"				
<u>Prenom</u>	<u>Nom</u>	Classe	Age	Ville
Sam	EDI	MP	20	Caen
Jean	NAIMAR	PCSI	18	Toulouse
Sam	OURAIL	PC	20	Paris

★ -1

Entité "Profs"				
Titre	<u>Prenom</u>	<u>Nom</u>	<u>Classe</u>	Salle
Mme	Sylvie	DEFOU	MP	B18
Mr	Georges	AITTE	PC	C15
Mme	Marie	AGE	PSI	E8

Opérateur JOIN ON

```
SELECT Eleves.Prenom, Eleves.Nom, Profs.Salle FROM Eleves  
JOIN Profs ON  
Eleves.Classe = Profs.Classe
```

Que fait cette requête à votre avis ?

Opérateur JOIN ON

```
SELECT Eleves.Prenom, Eleves.Nom, Profs.Salle FROM Eleves  
JOIN Profs ON  
Eleves.Classe = Profs.Classe
```



```
SELECT E.Prenom, E.Nom, P.Salle FROM Eleves AS E  
JOIN Profs AS P ON  
E.Classe = P.Classe
```

Jointure plus complexes, autojointures

- On peut faire une jointure sur autre chose qu'une clé :

```
SELECT * FROM Eleves JOIN Profs ON Eleves.Nom = Profs.Nom
```

Jointure plus complexes, autojointures

- On peut faire une jointure sur autre chose qu'une clé :

```
SELECT * FROM Eleves JOIN Profs ON Eleves.Nom = Profs.Nom
```

- On peut joindre une table avec elle-même (c'est alors une "autojointure")

```
SELECT * FROM Eleves AS E1 JOIN Eleves AS E2 ON  
      (E1.Classe = E2.Classe) AND (E1.Age = E2.Age) AND  
(E1.Prenom ≠ E2.Prenom)
```

Requêtes imbriquées

Le résultat d'une requête étant une table, on peut imbriquer les requêtes :

```
SELECT DISTINCT Prenom FROM (SELECT DISTINCT Nom, Age FROM Eleves  
WHERE Age < 19)
```


Ecrivez des requêtes pour :

- Afficher le nom, le prénom, et la salle de chaque élève.
-

Ecrivez des requêtes pour :

- Afficher le nom, le prénom, et la salle de chaque élève.
- Afficher toutes les paires d'élèves qui ont le même prénom.

Autres opérations ensemblistes

Les opérateurs ensemblistes classiques sont disponibles sur les tables. Ces opérateurs s'utilisent seulement sur des tables avec **les mêmes attributs**. Ils s'utilisent en général entre deux SELECT :

```
(SELECT Nom FROM Eleves) UNION (SELECT Nom FROM Profs)
```

Les opérateurs sont **UNION**, **INTERSECT** et **EXCEPT**. On peut aussi faire un produit cartésien avec **CROSS JOIN**.

Agrégats

Agrégats

Les agrégats servent à rassembler des enregistrements qui ont un attribut en commun.

```
SELECT * FROM Eleves GROUP BY Age
```

Agrégats

Les agrégats servent à rassembler des enregistrements qui ont un attribut en commun.

```
SELECT * FROM Eleves GROUP BY Age
```

On a également des **fonctions d'agrégation** qui deviennent disponibles :

```
SELECT Age, COUNT() AS Nombre FROM Eleves GROUP BY Age
```

Les fonctions d'agrégation

Les fonctions d'agrégation disponibles sont :

- `COUNT()`
- `MIN(col) / MAX(col)`
- `SUM(col)`
- `AVG(col)`

Ordre des opérateurs

1. SELECT (DISTINCT)
2. FROM
3. JOIN .. ON
4. WHERE
5. GROUP BY
6. ORDER BY
7. LIMIT .. OFFSET ..

Comment faire une restriction sur le résultat d'un agrégat ?

HAVING

1. SELECT (DISTINCT)
2. FROM
3. JOIN .. ON
4. WHERE
5. GROUP BY
6. HAVING
7. ORDER BY
8. LIMIT .. OFFSET ..

```
SELECT Nom, Prenom FROM Eleves GROUP BY Age HAVING COUNT() > 2
```

Ecrivez des requêtes pour :

- Afficher le nombre d'élèves par salle
- Afficher l'élève le plus âgé de chaque prof.