

# Bases de données

Lycée Fénelon - BCPST

## Bases de données

Architecture clients-serveur

## Tables, attributs et enregistrements

Vocabulaire des BDD

Commandes SQL de manipulation de tables

Exercice 1

Fonction d'agrégations

## Produit cartésien et jointure de tables

# Base de données. Architecture clients-serveur

- L'utilisation de fichiers pour stocker des données est trop limitée pour gérer un grand volume de données. On utilise plutôt une base de données qui manipulera des fichiers mais organisera les données de façon à les gérer de façon optimale.

# Base de données. Architecture clients-serveur

- L'utilisation de fichiers pour stocker des données est trop limitée pour gérer un grand volume de données. On utilise plutôt une base de données qui manipulera des fichiers mais organisera les données de façon à les gérer de façon optimale.
- Par exemple, dans les sites de e-commerce, le catalogue des produits est stocké dans une base de donnée.

# Base de données. Architecture clients-serveur

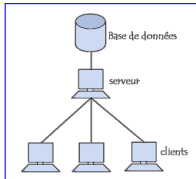
- L'utilisation de fichiers pour stocker des données est trop limitée pour gérer un grand volume de données. On utilise plutôt une base de données qui manipulera des fichiers mais organisera les données de façon à les gérer de façon optimale.
- Par exemple, dans les sites de e-commerce, le catalogue des produits est stocké dans une base de donnée. **Toutes les données administratives de la population française sont stockées dans des bases de données.**

# Base de données. Architecture clients-serveur

- L'utilisation de fichiers pour stocker des données est trop limitée pour gérer un grand volume de données. On utilise plutôt une base de données qui manipulera des fichiers mais organisera les données de façon à les gérer de façon optimale.
- Par exemple, dans les sites de e-commerce, le catalogue des produits est stocké dans une base de donnée. Toutes les données administratives de la population française sont stockées dans des bases de données.
- Une base de donnée est le système permettant le stockage des données, de manière structurée, en général sur un serveur (ordinateur relié à un réseau, internet par exemple).

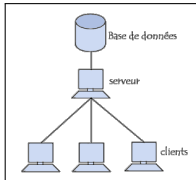
# Base de données. Architecture clients-serveur

- L'utilisation de fichiers pour stocker des données est trop limitée pour gérer un grand volume de données. On utilise plutôt une base de données qui manipulera des fichiers mais organisera les données de façon à les gérer de façon optimale.
- Par exemple, dans les sites de e-commerce, le catalogue des produits est stocké dans une base de donnée. Toutes les données administratives de la population française sont stockées dans des bases de données.
- Une base de donnée est le système permettant le stockage des données, de manière structurée, en général sur un serveur (ordinateur relié à un réseau, internet par exemple). Elle est conçue selon une architecture clients-serveurs :



# Base de données. Architecture clients-serveur

- L'utilisation de fichiers pour stocker des données est trop limitée pour gérer un grand volume de données. On utilise plutôt une base de données qui manipulera des fichiers mais organisera les données de façon à les gérer de façon optimale.
- Par exemple, dans les sites de e-commerce, le catalogue des produits est stocké dans une base de donnée. Toutes les données administratives de la population française sont stockées dans des bases de données.
- Une base de donnée est le système permettant le stockage des données, de manière structurée, en général sur un serveur (ordinateur relié à un réseau, internet par exemple). Elle est conçue selon une architecture clients-serveurs :



- Le serveur reçoit des **requêtes** de clients pour :
  - créer/modifier la base de donnée (seulement les administrateurs)
  - Interroger la base de donnée (tous les utilisateurs autorisés).



# Une base de donnée qu'est que c'est ?

- Un bon exemple de base de données est un carnet d'adresse. Dans un carnet d'adresse chaque entrée a un prénom, nom, numéro de téléphone, adresse, etc... Toutes ces données sont en relation : une adresse est reliée à un nom-prénom, etc...

# Une base de donnée qu'est que c'est ?

- Un bon exemple de base de données est un carnet d'adresse. Dans un carnet d'adresse chaque entrée a un prénom, nom, numéro de téléphone, adresse, etc... Toutes ces données sont en relation : une adresse est reliée à un nom-prénom, etc...
- La recherche dans un carnet d'adresse se fait en général par le nom. Mais par exemple un commercial pourra rechercher plutôt par adresse afin de gérer ses déplacements ; une centrale d'appel par numéro de téléphone pour optimiser ses coûts de frais d'appel ; une personne dont le véhicule est en panne pourra y rechercher plutôt un garagiste, etc...

# Une base de donnée qu'est que c'est ?

- Un bon exemple de base de données est un carnet d'adresse. Dans un carnet d'adresse chaque entrée a un prénom, nom, numéro de téléphone, adresse, etc... Toutes ces données sont en relation : une adresse est reliée à un nom-prénom, etc...
- La recherche dans un carnet d'adresse se fait en général par le nom. Mais par exemple un commercial pourra rechercher plutôt par adresse afin de gérer ses déplacements ; une centrale d'appel par numéro de téléphone pour optimiser ses coûts de frais d'appel ; une personne dont le véhicule est en panne pourra y rechercher plutôt un garagiste, etc...
- Une base de donnée répond à des requêtes formulées dans un langage structuré de requêtes : ce langage permettra de créer/modifier ou d'accéder aux informations pertinentes à l'aide d'un langage approprié. Par exemple : "Quels sont les garagistes dans mon carnet d'adresse situés sur la commune de mon lieu de panne?".

# Base de donnée - Vocabulaire

- Illustrons ces propos en imaginant quelle pourrait être une base de donnée pour gérer tous ses contacts.

# Base de donnée - Vocabulaire

- Illustrons ces propos en imaginant quelle pourrait être une base de donnée pour gérer tous ses contacts.
- On pourrait utiliser un carnet d'adresse pour stocker toutes les informations concernant ses amis : prénom, nom, numéro(s) de téléphone (1 ou plusieurs), adresse, date anniversaire.

# Base de donnée - Vocabulaire

- Illustrons ces propos en imaginant quelle pourrait être une base de donnée pour gérer tous ses contacts.
- On pourrait utiliser un carnet d'adresse pour stocker toutes les informations concernant ses amis : prénom, nom, numéro(s) de téléphone (1 ou plusieurs), adresse, date anniversaire.
- Et un autre carnet d'adresse pour stocker toutes les informations concernant ses contacts professionnels : prénom, nom, numéros(s) de téléphone, adresse, société, position.

# Base de donnée - Vocabulaire

- Illustrons ces propos en imaginant quelle pourrait être une base de donnée pour gérer tous ses contacts.
- On pourrait utiliser un carnet d'adresse pour stocker toutes les informations concernant ses amis : prénom, nom, numéro(s) de téléphone (1 ou plusieurs), adresse, date anniversaire.
- Et un autre carnet d'adresse pour stocker toutes les informations concernant ses contacts professionnels : prénom, nom, numéros(s) de téléphone, adresse, société, position.
- Ces deux carnets d'adresse constitueront deux tables dans la base de donnée ; la table est définie par son nom et son schéma de relation : c'est l'ensemble des champs (attributs) qui la constituent (nom, prénom, téléphone, etc...).

# Base de donnée - Vocabulaire

- Illustrons ces propos en imaginant quelle pourrait être une base de donnée pour gérer tous ses contacts.
- On pourrait utiliser un carnet d'adresse pour stocker toutes les informations concernant ses amis : prénom, nom, numéro(s) de téléphone (1 ou plusieurs), adresse, date anniversaire.
- Et un autre carnet d'adresse pour stocker toutes les informations concernant ses contacts professionnels : prénom, nom, numéros(s) de téléphone, adresse, société, position.
- Ces deux carnets d'adresse constitueront deux tables dans la base de donnée ; la table est définie par son nom et son schéma de relation : c'est l'ensemble des champs (attributs) qui la constituent (nom, prénom, téléphone, etc...).
- Les deux tables et leurs schémas relationnels :

| Amis   |     |           |         |              |
|--------|-----|-----------|---------|--------------|
| Prénom | Nom | Téléphone | Adresse | Anniversaire |

| Contacts Pro. |     |           |         |         |       |
|---------------|-----|-----------|---------|---------|-------|
| Prénom        | Nom | Téléphone | Adresse | Société | Poste |



# Base de donnée - Vocabulaire

- Chaque table va regrouper des contacts,  $n$ -uplets de valeurs. Ils ont appelés enregistrement.

# Base de donnée - Vocabulaire

- Chaque table va regrouper des contacts,  $n$ -uplets de valeurs. Ils ont appelés enregistrement.

Par exemple on a ajouté un enregistrement. :

| Amis   |        |            |                      |              |
|--------|--------|------------|----------------------|--------------|
| Prénom | Nom    | Téléphone  | Adresse              | Anniversaire |
| Paul   | Dupond | 0102030405 | 13 av. de la lumière | 27 mars 1990 |

# Base de donnée - Vocabulaire

- Chaque table va regrouper des contacts,  $n$ -uplets de valeurs. Ils ont appelés enregistrement.

Par exemple on a ajoute un enregistrement. :

| Amis   |        |            |                      |              |
|--------|--------|------------|----------------------|--------------|
| Prénom | Nom    | Téléphone  | Adresse              | Anniversaire |
| Paul   | Dupond | 0102030405 | 13 av. de la lumière | 27 mars 1990 |

- Cet enregistrement est constitué des valeurs d'attributs : Paul, Dupond, 0102030405, etc...

# Base de donnée - Vocabulaire

- Chaque table va regrouper des contacts,  $n$ -uplets de valeurs. Ils ont appelés enregistrement.

Par exemple on a ajoute un enregistrement. :

| Amis   |        |            |                      |              |
|--------|--------|------------|----------------------|--------------|
| Prénom | Nom    | Téléphone  | Adresse              | Anniversaire |
| Paul   | Dupond | 0102030405 | 13 av. de la lumière | 27 mars 1990 |

- Cet enregistrement est constitué des valeurs d'attributs : Paul, Dupond, 0102030405, etc...
- Chaque attribut a un domaine ; c'est essentiellement son type : prénom, nom et adresse seront stockés comme chaîne de caractère ; téléphone comme un entier et anniversaire comme une date.

# Base de donnée - Vocabulaire

- Dans une table on doit pouvoir accéder facilement à un enregistrement à l'aide de la valeur de l'un de ses attributs.

# Base de donnée - Vocabulaire

- Dans une table on doit pouvoir accéder facilement à un enregistrement à l'aide de la valeur de l'un de ses attributs.
- Un ou plusieurs attributs donnant accès à un unique élément dans une table s'appelle une clé.

# Base de donnée - Vocabulaire

- Dans une table on doit pouvoir accéder facilement à un enregistrement à l'aide de la valeur de l'un de ses attributs.
- Un ou plusieurs attributs donnant accès à un unique élément dans une table s'appelle une clé.
- Par exemple le couple (nom, prénom) est un assez bon choix de clé. On parle de clé composite quand elle est constituée de plusieurs attributs.

# Base de donnée - Vocabulaire

- Dans une table on doit pouvoir accéder facilement à un enregistrement à l'aide de la valeur de l'un de ses attributs.
- Un ou plusieurs attributs donnant accès à un unique élément dans une table s'appelle une clé.
- Par exemple le couple (nom, prénom) est un assez bon choix de clé. On parle de clé composite quand elle est constituée de plusieurs attributs.
- La clé choisie s'appelle la clé primaire. Elle doit être aussi simple que possible.



# Base de donnée - Vocabulaire

- Dans une table on doit pouvoir accéder facilement à un enregistrement à l'aide de la valeur de l'un de ses attributs.
- Un ou plusieurs attributs donnant accès à un unique élément dans une table s'appelle une clé.
- Par exemple le couple (nom, prénom) est un assez bon choix de clé. On parle de clé composite quand elle est constituée de plusieurs attributs.
- La clé choisie s'appelle la clé primaire. Elle doit être aussi simple que possible.
- Les autres clés possibles sont les clés secondaires.

# Vocabulaire des BDD

|                           |                                                                                         |
|---------------------------|-----------------------------------------------------------------------------------------|
| Une <b>Base de donnée</b> | est constituée de tables, contenant les données organisées selon un schéma relationnel. |
|---------------------------|-----------------------------------------------------------------------------------------|

# Vocabulaire des BDD

|                                        |                                                                                         |
|----------------------------------------|-----------------------------------------------------------------------------------------|
| Une <b>Base de donnée</b>              | est constituée de tables, contenant les données organisées selon un schéma relationnel. |
| Une <b>Table</b><br>ou <b>Relation</b> | est un ensemble de données organisées vérifiant un même schéma relationnel.             |

# Vocabulaire des BDD

|                                        |                                                                                         |
|----------------------------------------|-----------------------------------------------------------------------------------------|
| Une <b>Base de donnée</b>              | est constituée de tables, contenant les données organisées selon un schéma relationnel. |
| Une <b>Table</b><br>ou <b>Relation</b> | est un ensemble de données organisées vérifiant un même schéma relationnel.             |
| Un <b>schéma relationnel</b>           | est une suite finie d'attributs (colonnes d'une table).                                 |
|                                        |                                                                                         |

# Vocabulaire des BDD

|                                        |                                                                                         |
|----------------------------------------|-----------------------------------------------------------------------------------------|
| Une <b>Base de donnée</b>              | est constituée de tables, contenant les données organisées selon un schéma relationnel. |
| Une <b>Table</b><br>ou <b>Relation</b> | est un ensemble de données organisées vérifiant un même schéma relationnel.             |
| Un <b>schéma relationnel</b>           | est une suite finie d'attributs (colonnes d'une table).                                 |
| Un <b>Attribut</b>                     | définit la colonne d'une table : nom (identifiant) et type.                             |

# Vocabulaire des BDD

|                                        |                                                                                         |
|----------------------------------------|-----------------------------------------------------------------------------------------|
| Une <b>Base de donnée</b>              | est constituée de tables, contenant les données organisées selon un schéma relationnel. |
| Une <b>Table</b><br>ou <b>Relation</b> | est un ensemble de données organisées vérifiant un même schéma relationnel.             |
| Un <b>schéma relationnel</b>           | est une suite finie d'attributs (colonnes d'une table).                                 |
| Un <b>Attribut</b>                     | définit la colonne d'une table : nom (identifiant) et type.                             |
| Un <b>Enregistrement</b>               | est un élément d'une table (une ligne).                                                 |

# Vocabulaire des BDD

|                                        |                                                                                         |
|----------------------------------------|-----------------------------------------------------------------------------------------|
| Une <b>Base de donnée</b>              | est constituée de tables, contenant les données organisées selon un schéma relationnel. |
| Une <b>Table</b><br>ou <b>Relation</b> | est un ensemble de données organisées vérifiant un même schéma relationnel.             |
| Un <b>schéma relationnel</b>           | est une suite finie d'attributs (colonnes d'une table).                                 |
| Un <b>Attribut</b>                     | définit la colonne d'une table : nom (identifiant) et type.                             |
| Un <b>Enregistrement</b>               | est un élément d'une table (une ligne).                                                 |
| Une <b>Clé</b>                         | est un ou plusieurs attributs donnant accès à un seul élément dans une table.           |
| La <b>Clé primaire</b>                 | est la clé choisie parmi toutes les clés possibles.                                     |

- Dans une base de donnée nommée classe : figure une table eleves ayant pour attributs : nom, prénom, adresse, telephone, date de naissance :



- Dans une base de donnée nommée classe : figure une table `elevés` ayant pour attributs : nom, prénom, adresse, telephone, date de naissance :

| Nom                                                                                                     |                        | Type                     |
|---------------------------------------------------------------------------------------------------------|------------------------|--------------------------|
| ▼  Tables (1)          |                        |                          |
| ▼  <code>elevés</code> |                        |                          |
|                        | <code>nom</code>       | <code>VARCHAR(20)</code> |
|                        | <code>prenom</code>    | <code>VARCHAR(18)</code> |
|                        | <code>adresse</code>   | <code>VARCHAR(50)</code> |
|                        | <code>telephone</code> | <code>INT(10)</code>     |
|                        | <code>naissance</code> | <code>DATE</code>        |

- Dans une base de donnée nommée classe : figure une table `elevés` ayant pour attributs : nom, prénom, adresse, telephone, date de naissance :

| Nom                   | Type        |
|-----------------------|-------------|
| ▼  Tables (1)         |             |
| ▼ <code>elevés</code> |             |
| nom                   | VARCHAR(20) |
| prenom                | VARCHAR(18) |
| adresse               | VARCHAR(50) |
| telephone             | INT(10)     |
| naissance             | DATE        |

- Nous avons saisi manuellement les enregistrements suivants dans la table `elevés` :

|   | nom      | prenom | adresse             | telephone  | naissance  | langue |
|---|----------|--------|---------------------|------------|------------|--------|
|   | Filter   | Filter | Filter              | Filter     | Filter     | Filter |
| 1 | Dupond   | Tom    | 3 place de la gare  | 609080706  | 1995-03-15 | esp    |
| 2 | Charron  | Ella   | 10 rue de la gare   | 0607080908 | 1994-12-10 | ita    |
| 3 | Cuvelier | Paul   | 11 rue Paul Bert    | 0665544332 | 1995-09-12 | all    |
| 4 | Dumas    | Anne   | 5 rue de l'oiseau   | 0667778778 | 1994-10-01 | rus    |
| 5 | Henri    | Thomas | 3 quai du port      | 0607677878 | 1995-01-24 | all    |
| 6 | Laurin   | Theo   | 4 place de l'Ormeau | 0605040321 | 1995-06-07 | esp    |

- On peut alors afficher la table grâce à la requête :

```
SELECT * FROM eleves
```

qui produit pour résultat :

|   | nom      | prenom | adresse             | telephone | naissance  | langue |
|---|----------|--------|---------------------|-----------|------------|--------|
| 1 | Dupond   | Tom    | 3 place de la gare  | 609080706 | 1995-03-15 | esp    |
| 2 | Charron  | Ella   | 10 rue de la gare   | 607080908 | 1994-12-10 | ita    |
| 3 | Cuvelier | Paul   | 11 rue Paul Bert    | 665544332 | 1995-09-12 | all    |
| 4 | Dumas    | Anne   | 5 rue de l'oiseau   | 667778778 | 1994-10-01 | rus    |
| 5 | Henri    | Thomas | 3 quai du port      | 607677878 | 1995-01-24 | all    |
| 6 | Laurin   | Theo   | 4 place de l'Ormeau | 605040321 | 1995-06-07 | esp    |

- On peut alors afficher la table grâce à la requête :

```
SELECT * FROM eleves
```

qui produit pour résultat :

|   | nom      | prenom | adresse             | telephone | naissance  | langue |
|---|----------|--------|---------------------|-----------|------------|--------|
| 1 | Dupond   | Tom    | 3 place de la gare  | 609080706 | 1995-03-15 | esp    |
| 2 | Charron  | Ella   | 10 rue de la gare   | 607080908 | 1994-12-10 | ita    |
| 3 | Cuvelier | Paul   | 11 rue Paul Bert    | 665544332 | 1995-09-12 | all    |
| 4 | Dumas    | Anne   | 5 rue de l'oiseau   | 667778778 | 1994-10-01 | rus    |
| 5 | Henri    | Thomas | 3 quai du port      | 607677878 | 1995-01-24 | all    |
| 6 | Laurin   | Theo   | 4 place de l'Ormeau | 605040321 | 1995-06-07 | esp    |

- On peut ne sélectionner qu'un enregistrement :

```
SELECT * FROM eleves WHERE nom = 'Charron'
```

|   | nom     | prenom | adresse           | telephone | naissance  | langue |
|---|---------|--------|-------------------|-----------|------------|--------|
| 1 | Charron | Ella   | 10 rue de la gare | 607080908 | 1994-12-10 | ita    |

- ou ne sélectionner que certains attributs :

```
SELECT nom, prénom FROM eleves
```

|   | nom      | prénom |
|---|----------|--------|
| 1 | Dupond   | Tom    |
| 2 | Charron  | Ella   |
| 3 | Cuvelier | Paul   |
| 4 | Dumas    | Anne   |
| 5 | Henri    | Thomas |
| 6 | Laurin   | Theo   |

- ou ne sélectionner que certains attributs :

```
SELECT nom, prénom FROM eleves
```

|   | nom      | prénom |
|---|----------|--------|
| 1 | Dupond   | Tom    |
| 2 | Charron  | Ella   |
| 3 | Cuvelier | Paul   |
| 4 | Dumas    | Anne   |
| 5 | Henri    | Thomas |
| 6 | Laurin   | Theo   |

- ... ORDER BY... permet de les ordonner selon une valeur d'attribut :

```
SELECT nom, prénom, naissance FROM eleves ORDER BY nom
```

|   | nom      | prénom | naissance  |
|---|----------|--------|------------|
| 1 | Charron  | Ella   | 1994-12-10 |
| 2 | Cuvelier | Paul   | 1995-09-12 |
| 3 | Dumas    | Anne   | 1994-10-01 |
| 4 | Dupond   | Tom    | 1995-03-15 |
| 5 | Henri    | Thomas | 1995-01-24 |
| 6 | Laurin   | Theo   | 1995-06-07 |

# Commandes SQL de manipulation de tables

```
SELECT * FROM ma_table
```

Afficher tous les enregistrements  
de la table `ma_table`

# Commandes SQL de manipulation de tables

|                                               |                                                                            |
|-----------------------------------------------|----------------------------------------------------------------------------|
| <code>SELECT * FROM ma_table</code>           | Afficher tous les enregistrements de la table <code>ma_table</code>        |
| <code>SELECT A1, ..., Ap FROM ma_table</code> | Projection : afficher certaines colonnes de la table <code>ma_table</code> |
|                                               |                                                                            |



# Commandes SQL de manipulation de tables

|                                                  |                                                                            |
|--------------------------------------------------|----------------------------------------------------------------------------|
| <code>SELECT * FROM ma_table</code>              | Afficher tous les enregistrements de la table <code>ma_table</code>        |
| <code>SELECT A1, ..., Ap FROM ma_table</code>    | Projection : afficher certaines colonnes de la table <code>ma_table</code> |
| <code>SELECT ... FROM ... WHERE condition</code> | Sélection : sélectionne les enregistrements vérifiant une condition        |
|                                                  |                                                                            |

# Commandes SQL de manipulation de tables

|                                                  |                                                                            |
|--------------------------------------------------|----------------------------------------------------------------------------|
| <code>SELECT * FROM ma_table</code>              | Afficher tous les enregistrements de la table <code>ma_table</code>        |
| <code>SELECT A1, ..., Ap FROM ma_table</code>    | Projection : afficher certaines colonnes de la table <code>ma_table</code> |
| <code>SELECT ... FROM ... WHERE condition</code> | Sélection : sélectionne les enregistrements vérifiant une condition        |
| <code>SELECT ... ORDER BY Ak [DESC]</code>       | Ordonner l'affichage selon un attribut                                     |
|                                                  |                                                                            |

# Commandes SQL de manipulation de tables

|                                                  |                                                                            |
|--------------------------------------------------|----------------------------------------------------------------------------|
| <code>SELECT * FROM ma_table</code>              | Afficher tous les enregistrements de la table <code>ma_table</code>        |
| <code>SELECT A1, ..., Ap FROM ma_table</code>    | Projection : afficher certaines colonnes de la table <code>ma_table</code> |
| <code>SELECT ... FROM ... WHERE condition</code> | Sélection : sélectionne les enregistrements vérifiant une condition        |
| <code>SELECT ... ORDER BY Ak [DESC]</code>       | Ordonner l'affichage selon un attribut                                     |
| <code>SELECT ... LIMIT n</code>                  | Limitation aux <code>n</code> premiers éléments                            |
|                                                  |                                                                            |

# Commandes SQL de manipulation de tables

|                                                  |                                                                            |
|--------------------------------------------------|----------------------------------------------------------------------------|
| <code>SELECT * FROM ma_table</code>              | Afficher tous les enregistrements de la table <code>ma_table</code>        |
| <code>SELECT A1, ..., Ap FROM ma_table</code>    | Projection : afficher certaines colonnes de la table <code>ma_table</code> |
| <code>SELECT ... FROM ... WHERE condition</code> | Sélection : sélectionne les enregistrements vérifiant une condition        |
| <code>SELECT ... ORDER BY Ak [DESC]</code>       | Ordonner l'affichage selon un attribut                                     |
| <code>SELECT ... LIMIT n</code>                  | Limitation aux <code>n</code> premiers éléments                            |
| <code>SELECT ... LIMIT m, n</code>               | Limitation aux éléments de <code>m</code> à <code>n</code>                 |
|                                                  |                                                                            |

# Commandes SQL de manipulation de tables

|                                                  |                                                                            |
|--------------------------------------------------|----------------------------------------------------------------------------|
| <code>SELECT * FROM ma_table</code>              | Afficher tous les enregistrements de la table <code>ma_table</code>        |
| <code>SELECT A1, ..., Ap FROM ma_table</code>    | Projection : afficher certaines colonnes de la table <code>ma_table</code> |
| <code>SELECT ... FROM ... WHERE condition</code> | Sélection : sélectionne les enregistrements vérifiant une condition        |
| <code>SELECT ... ORDER BY Ak [DESC]</code>       | Ordonner l'affichage selon un attribut                                     |
| <code>SELECT ... LIMIT n</code>                  | Limitation aux <code>n</code> premiers éléments                            |
| <code>SELECT ... LIMIT m, n</code>               | Limitation aux éléments de <code>m</code> à <code>n</code>                 |
| <code>SELECT A1 AS nom FROM ma_table</code>      | Sélection/projection avec renommage.                                       |

# Exercice 1

Ouvrir la table `nobel` à l'aide du logiciel SQLlite.

1. Lancer une requête qui affiche tout le contenu de la table.
2. Lancer une requête qui affiche tous les lauréats du nobel de physique, et leur année d'obtention.
3. Lancer une requête qui affiche tous les lauréats du nobel de l'année 1956.
4. Lancer une requête qui affiche toutes les disciplines du nobel.
5. Lancer une requête qui affiche tous les nobels obtenus par Marie Curie.

# Fonctions d'agrégations

- Une fonction d'agrégation est une application qui a une table associe un nombre. Le plus souvent pour compter, faire la moyenne d'un attribut, etc..., selon une colonne.

# Fonctions d'agrégations

- Une fonction d'agrégation est une application qui a une table associe un nombre. Le plus souvent pour compter, faire la moyenne d'un attribut, etc..., selon une colonne.
- Les principales fonctions d'agrégation :

|       |                                                             |
|-------|-------------------------------------------------------------|
| COUNT | pour compter le nombre d'enregistrements d'une table        |
| AVG   | valeur moyenne d'une colonne de type numérique d'une table  |
| MIN   | valeur minimale d'une colonne de type numérique d'une table |
| MAX   | valeur maximale d'une colonne de type numérique d'une table |
| SUM   | somme d'une colonne de type numérique d'une table           |



# Fonctions d'agrégations

- Une fonction d'agrégation est une application qui a une table associe un nombre. Le plus souvent pour compter, faire la moyenne d'un attribut, etc..., selon une colonne.
- Les principales fonctions d'agrégation :

|       |                                                             |
|-------|-------------------------------------------------------------|
| COUNT | pour compter le nombre d'enregistrements d'une table        |
| AVG   | valeur moyenne d'une colonne de type numérique d'une table  |
| MIN   | valeur minimale d'une colonne de type numérique d'une table |
| MAX   | valeur maximale d'une colonne de type numérique d'une table |
| SUM   | somme d'une colonne de type numérique d'une table           |

- Exemple : Compter les élèves (en paramètre les attributs à compter, les cases vides ne sont pas comptabilisés)

```
SELECT COUNT(*) FROM eleves
```

| COUNT(*) |   |
|----------|---|
| 1        | 6 |

La table contient 6 enregistrements (= élèves).

# Fonctions d'agrégations

- Une fonction d'agrégation retourne une seule ligne. Avec la commande GROUP BY on retourne plusieurs lignes en regroupant les mêmes valeurs d'un attribut et en effectuant l'agrégation sur chaque groupe :

```
SELECT COUNT(*) FROM eleves GROUP BY langue
```

On compte le nombre d'élèves par groupe de langue.

|   | COUNT(*) | langue |
|---|----------|--------|
| 1 | 2        | all    |
| 2 | 2        | esp    |
| 3 | 1        | ita    |
| 4 | 1        | rus    |

# Fonctions d'agrégations

- Une fonction d'agrégation retourne une seule ligne. Avec la commande GROUP BY on retourne plusieurs lignes en regroupant les mêmes valeurs d'un attribut et en effectuant l'agrégation sur chaque groupe :

```
SELECT COUNT(*) FROM eleves GROUP BY langue
```

On compte le nombre d'élèves par groupe de langue.

|   | COUNT(*) | langue |
|---|----------|--------|
| 1 | 2        | all    |
| 2 | 2        | esp    |
| 3 | 1        | ita    |
| 4 | 1        | rus    |

```
SELECT COUNT(*) FROM eleves GROUP BY langue WHERE langue = 'esp'
```

ne comptera que le groupe des élèves faisant espagnol.

- Supposons que notre BDD contienne une seconde table notes contenant les notes des élèves :

|                                                                                            |              |                         |
|--------------------------------------------------------------------------------------------|--------------|-------------------------|
|  nom      | VARCHAR(20)  | `nom` VARCHAR(20)       |
|  prenom   | VARCHAR(18)  | `prenom` VARCHAR(18)    |
|  math     | DECIMAL(4,2) | `math` DECIMAL(4,2)     |
|  physique | DECIMAL(4,2) | `physique` DECIMAL(4,2) |
|  français | DECIMAL(4,2) | `français` DECIMAL(4,2) |
|  anglais  | DECIMAL(4,2) | `anglais` DECIMAL(4,2)  |

- Supposons que notre BDD contienne une seconde table **notes** contenant les notes des élèves :

|                                                                                            |              |                         |
|--------------------------------------------------------------------------------------------|--------------|-------------------------|
|  nom      | VARCHAR(20)  | `nom` VARCHAR(20)       |
|  prenom   | VARCHAR(18)  | `prenom` VARCHAR(18)    |
|  math     | DECIMAL(4,2) | `math` DECIMAL(4,2)     |
|  physique | DECIMAL(4,2) | `physique` DECIMAL(4,2) |
|  français | DECIMAL(4,2) | `français` DECIMAL(4,2) |
|  anglais  | DECIMAL(4,2) | `anglais` DECIMAL(4,2)  |

|   | nom      | prenom | math   | physique | français | anglais |
|---|----------|--------|--------|----------|----------|---------|
|   | Filter   | Filter | Filter | Filter   | Filter   | Filter  |
| 1 | Dupond   | Tom    | 10     | 12       | 14       | 15      |
| 2 | Dumas    | Anne   | 10     | 12       | 14       | 15      |
| 3 | Laurin   | Theo   | 11     | 8        | 13       | 12      |
| 4 | Charron  | Ella   | 9      | 10       | 11       | 7       |
| 5 | Henri    | Thomas | 8      | 11       | 7        | 6       |
| 6 | Cuvelier | Paul   | 5      | 10       | 8        | 12      |

- La moyenne en maths s'obtient par la requête :

```
SELECT AVG(math) FROM notes
```

- La moyenne en maths s'obtient par la requête :

```
SELECT AVG(math) FROM notes
```

- La moyenne générale de la classe s'obtient par la requête :

```
SELECT (AVG(math)+AVG(physique)+AVG(français)+AVG(anglais)) / 4.0  
FROM notes
```

- La moyenne en maths s'obtient par la requête :

```
SELECT AVG(math) FROM notes
```

- La moyenne générale de la classe s'obtient par la requête :

```
SELECT (AVG(math)+AVG(physique)+AVG(français)+AVG(anglais)) / 4.0  
FROM notes
```

On affiche la moyenne générale d'un élève par :

```
SELECT nom, (math+physique+français+anglais)/4.0 FROM notes  
WHERE nom = 'Dumas'
```



- La moyenne en maths s'obtient par la requête :

```
SELECT AVG(math) FROM notes
```

- La moyenne générale de la classe s'obtient par la requête :

```
SELECT (AVG(math)+AVG(physique)+AVG(français)+AVG(anglais)) / 4.0  
FROM notes
```

On affiche la moyenne générale d'un élève par :

```
SELECT nom, (math+physique+français+anglais)/4.0 FROM notes  
WHERE nom = 'Dumas'
```

La moyenne générale des élèves s'obtient par la requête :

```
SELECT nom, (math+physique+français+anglais)/4.0 FROM notes
```

|   | nom      | (math+physique+français+anglais)/4 |
|---|----------|------------------------------------|
| 1 | Dupond   | 12                                 |
| 2 | Dumas    | 12                                 |
| 3 | Laurin   | 11                                 |
| 4 | Charron  | 9                                  |
| 5 | Henri    | 8                                  |
| 6 | Cuvelier | 8                                  |

Ou encore, pour un résultat plus esthétique effectuer un renommage avec AS :

```
SELECT nom, (math+physique+français+anglais)/4.0 AS moyenne FROM notes
```

|   | nom      | moyenne |
|---|----------|---------|
| 1 | Dupond   | 12      |
| 2 | Dumas    | 12      |
| 3 | Laurin   | 11      |
| 4 | Charron  | 9       |
| 5 | Henri    | 8       |
| 6 | Cuvelier | 8       |

Ou encore, pour un résultat plus esthétique effectuer un renommage avec AS :

```
SELECT nom, (math+physique+français+anglais)/4.0 AS moyenne FROM notes
```

|   | nom      | moyenne |
|---|----------|---------|
| 1 | Dupond   | 12      |
| 2 | Dumas    | 12      |
| 3 | Laurin   | 11      |
| 4 | Charron  | 9       |
| 5 | Henri    | 8       |
| 6 | Cuvelier | 8       |

- Le nom du major de promo s'obtient par la requête :

```
SELECT nom, (math+physique+français+anglais)/4.0 AS moyenne  
FROM notes ORDER BY moyenne DESC LIMIT 1
```

## Exercice 2

Dans la table `nobel` :

1. Sur quelle période s'étale la table ? (obtenir l'année du premier et du dernier nobel dans la table).
2. Obtenir le nombre d'années dans la table.
3. Obtenir le nombre de lauréats des nobel.
4. Obtenir le nombre des lauréats des nobel de Médecine.
5. Obtenir le nombre des lauréats des nobels de chaque discipline.
6. Afficher tous les lauréats et pour chacun le nombre de nobel obtenus, classés par nombre décroissant.
7. Afficher tous lauréats ayant reçu plusieurs nobels.

# Produit cartésien de tables

- Pour effectuer un produit cartésien : `SELECT * FROM table1, table2`

```
SELECT * FROM eleves, notes
```

# Produit cartésien de tables

- Pour effectuer un produit cartésien : `SELECT * FROM table1, table2`

```
SELECT * FROM eleves, notes
```

|    | nom     | prenom | adresse            | telephone | naissance  | langue | nom      | prenom | math | physique | français | anglais |
|----|---------|--------|--------------------|-----------|------------|--------|----------|--------|------|----------|----------|---------|
| 1  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Dupond   | Tom    | 10   | 12       | 14       | 15      |
| 2  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Dumas    | Anne   | 10   | 12       | 14       | 15      |
| 3  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Laurin   | Theo   | 11   | 8        | 13       | 12      |
| 4  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Charron  | Ella   | 9    | 10       | 11       | 7       |
| 5  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Henri    | Thomas | 8    | 11       | 7        | 6       |
| 6  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Cuvelier | Paul   | 5    | 10       | 8        | 12      |
| 7  | Charron | Ella   | 10 rue de la gare  | 607080908 | 1994-12-10 | ita    | Dupond   | Tom    | 10   | 12       | 14       | 15      |
| 8  | Charron | Ella   | 10 rue de la gare  | 607080908 | 1994-12-10 | ita    | Dumas    | Anne   | 10   | 12       | 14       | 15      |
| 9  | Charron | Ella   | 10 rue de la gare  | 607080908 | 1994-12-10 | ita    | Laurin   | Theo   | 11   | 8        | 13       | 12      |
| 10 | Charron | Ella   | 10 rue de la gare  | 607080908 | 1994-12-10 | ita    | Charron  | Ella   | 9    | 10       | 11       | 7       |

36 enregistrements ramenés depuis `SELECT * FROM eleves, notes;` (en 4ms)

# Produit cartésien de tables

- Pour effectuer un produit cartésien : `SELECT * FROM table1, table2`

```
SELECT * FROM eleves, notes
```

|    | nom     | prenom | adresse            | telephone | naissance  | langue | nom      | prenom | math | physique | français | anglais |
|----|---------|--------|--------------------|-----------|------------|--------|----------|--------|------|----------|----------|---------|
| 1  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Dupond   | Tom    | 10   | 12       | 14       | 15      |
| 2  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Dumas    | Anne   | 10   | 12       | 14       | 15      |
| 3  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Laurin   | Theo   | 11   | 8        | 13       | 12      |
| 4  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Charron  | Ella   | 9    | 10       | 11       | 7       |
| 5  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Henri    | Thomas | 8    | 11       | 7        | 6       |
| 6  | Dupond  | Tom    | 3 place de la gare | 609080706 | 1995-03-15 | esp    | Cuvelier | Paul   | 5    | 10       | 8        | 12      |
| 7  | Charron | Ella   | 10 rue de la gare  | 607080908 | 1994-12-10 | ita    | Dupond   | Tom    | 10   | 12       | 14       | 15      |
| 8  | Charron | Ella   | 10 rue de la gare  | 607080908 | 1994-12-10 | ita    | Dumas    | Anne   | 10   | 12       | 14       | 15      |
| 9  | Charron | Ella   | 10 rue de la gare  | 607080908 | 1994-12-10 | ita    | Laurin   | Theo   | 11   | 8        | 13       | 12      |
| 10 | Charron | Ella   | 10 rue de la gare  | 607080908 | 1994-12-10 | ita    | Charron  | Ella   | 9    | 10       | 11       | 7       |

36 enregistrements ramenés depuis `SELECT * FROM eleves, notes;` (en 4ms)

- Un produit cartésien est rarement utile ; une jointure l'est beaucoup plus.

- Pour effectuer une jointure symétrique simple :  
(Condition : égalité de certains attributs) :

```
SELECT * FROM table1 JOIN table2  
      ON table1.attribut = table2.attribut
```



- Pour effectuer une jointure symétrique simple :  
(Condition : égalité de certains attributs) :

```
SELECT * FROM table1 JOIN table2  
ON table1.attribut = table2.attribut
```

Par exemple, seulement sur quelques attributs (projection) :

```
SELECT eleves.nom, eleves.prenom, naissance, langue,  
(math+physique+français+anglais)/4.0 FROM eleves JOIN notes  
ON eleves.nom = notes.nom
```

|   | nom      | prenom | naissance  | langue | (math+physique+français+anglais)/4.0 |
|---|----------|--------|------------|--------|--------------------------------------|
| 1 | Dupond   | Tom    | 1995-03-15 | esp    | 12.75                                |
| 2 | Charron  | Ella   | 1994-12-10 | ita    | 9.25                                 |
| 3 | Cuvelier | Paul   | 1995-09-12 | all    | 8.75                                 |
| 4 | Dumas    | Anne   | 1994-10-01 | rus    | 12.75                                |
| 5 | Henri    | Thomas | 1995-01-24 | all    | 8.0                                  |
| 6 | Laurin   | Theo   | 1995-06-07 | esp    | 11.0                                 |

Pour obtenir le classement : avec ORDER BY et AS (renommage) :

```
SELECT eleves.nom, eleves.prenom, naissance, langue,  
       (math+physique+français+anglais)/4.0 AS moyenne  
FROM eleves JOIN notes ON eleves.nom = notes.nom  
ORDER BY moyenne DESC
```

Pour obtenir le classement : avec ORDER BY et AS (renommage) :

```
SELECT eleves.nom, eleves.prenom, naissance, langue,  
       (math+physique+français+anglais)/4.0 AS moyenne  
FROM eleves JOIN notes ON eleves.nom = notes.nom  
ORDER BY moyenne DESC
```

|   | nom      | prenom | naissance  | langue | moyenne |
|---|----------|--------|------------|--------|---------|
| 1 | Dupond   | Tom    | 1995-03-15 | esp    | 12.75   |
| 2 | Dumas    | Anne   | 1994-10-01 | rus    | 12.75   |
| 3 | Laurin   | Theo   | 1995-06-07 | esp    | 11.0    |
| 4 | Charron  | Ella   | 1994-12-10 | ita    | 9.25    |
| 5 | Cuvelier | Paul   | 1995-09-12 | all    | 8.75    |
| 6 | Henri    | Thomas | 1995-01-24 | all    | 8.0     |

- Récupérer les noms et n° de téléphone des élèves n'ayant pas la moyenne :

```
SELECT eleves.nom, telephone  
FROM eleves JOIN notes ON eleves.nom = notes.nom  
WHERE (math+physique+français+anglais)/4.0 < 10
```

- Récupérer les noms et n° de téléphone des élèves n'ayant pas la moyenne :

```
SELECT eleves.nom, telephone  
FROM eleves JOIN notes ON eleves.nom = notes.nom  
WHERE (math+physique+français+anglais)/4.0 < 10
```

|   | nom      | telephone |
|---|----------|-----------|
| 1 | Charron  | 607080908 |
| 2 | Cuvelier | 665544332 |
| 3 | Henri    | 607677878 |

- Récupérer les noms et n° de téléphone des élèves n'ayant pas la moyenne :

```
SELECT eleves.nom, telephone  
FROM eleves JOIN notes ON eleves.nom = notes.nom  
WHERE (math+physique+français+anglais)/4.0 < 10
```

|   | nom      | telephone |
|---|----------|-----------|
| 1 | Charron  | 607080908 |
| 2 | Cuvelier | 665544332 |
| 3 | Henri    | 607677878 |

- Récupérer les noms et moyenne des élèves nés après 1995 :

```
SELECT eleves.nom, (math+physique+français+anglais)/4.0  
AS moyenne FROM eleves JOIN notes  
ON eleves.nom = notes.nom  
WHERE naissance >= '1995-01-01'
```

- Récupérer les noms et n° de téléphone des élèves n'ayant pas la moyenne :

```
SELECT eleves.nom, telephone
FROM eleves JOIN notes ON eleves.nom = notes.nom
WHERE (math+physique+français+anglais)/4.0 < 10
```

|   | nom      | telephone |
|---|----------|-----------|
| 1 | Charron  | 607080908 |
| 2 | Cuvelier | 665544332 |
| 3 | Henri    | 607677878 |

- Récupérer les noms et moyenne des élèves nés après 1995 :

```
SELECT eleves.nom, (math+physique+français+anglais)/4.0
AS moyenne FROM eleves JOIN notes
ON eleves.nom = notes.nom
WHERE naissance >= '1995-01-01'
```

|   | nom      | moyenne |
|---|----------|---------|
| 1 | Dupond   | 12.75   |
| 2 | Cuvelier | 8.75    |
| 3 | Henri    | 8.0     |
| 4 | Laurin   | 11.0    |

## Exercice 3 : Base de donnée movie.sqlite

- La base de donnée movie.sqlite contient 3 tables vérifiant les schémas suivants :

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
        votes INTEGER, director INTEGER)
```

Elle porte sur tous les films de cinema, leur casting, acteurs et réalisateurs jusqu'à l'année 2000. La table actor contient les acteurs mais aussi les réalisateurs.



## Exercice 3 : Base de donnée movie.sqlite

- La base de donnée movie.sqlite contient 3 tables vérifiant les schémas suivants :

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
        votes INTEGER, director INTEGER)
```

Elle porte sur tous les films de cinema, leur casting, acteurs et réalisateurs jusqu'à l'année 2000. La table actor contient les acteurs mais aussi les réalisateurs.

**Question 1.** Comment obtenir les titres des films de l'année 1989 ?

## Exercice 3 : Base de donnée movie.sqlite

- La base de donnée movie.sqlite contient 3 tables vérifiant les schémas suivants :

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
        votes INTEGER, director INTEGER)
```

Elle porte sur tous les films de cinema, leur casting, acteurs et réalisateurs jusqu'à l'année 2000. La table actor contient les acteurs mais aussi les réalisateurs.

**Question 1.** Comment obtenir les titres des films de l'année 1989 ?

**Réponse :** (sélection)

```
SELECT title FROM movie WHERE yr = 1989
```

## Exercice 3 : Base de donnée movie.sqlite

- La base de donnée movie.sqlite contient 3 tables vérifiant les schémas suivants :

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
        votes INTEGER, director INTEGER)
```

Elle porte sur tous les films de cinema, leur casting, acteurs et réalisateurs jusqu'à l'année 2000. La table actor contient les acteurs mais aussi les réalisateurs.

**Question 2.** Comment obtenir la note moyenne de tous les films ?

## Exercice 3 : Base de donnée movie.sqlite

- La base de donnée movie.sqlite contient 3 tables vérifiant les schémas suivants :

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
        votes INTEGER, director INTEGER)
```

Elle porte sur tous les films de cinema, leur casting, acteurs et réalisateurs jusqu'à l'année 2000. La table actor contient les acteurs mais aussi les réalisateurs.

**Question 2.** Comment obtenir la note moyenne de tous les films ?

**Réponse :** (sélection avec agrégation)

```
SELECT AVG(score) FROM movie
```

## Exercice 3 : Base de donnée movie.sqlite

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
        votes INTEGER, director INTEGER)
```

**Question 3.** Obtenir les noms de tous les réalisateurs.

## Exercice 3 : Base de donnée movie.sqlite

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
        votes INTEGER, director INTEGER)
```

**Question 3.** Obtenir les noms de tous les réalisateurs.

**Réponse :** (sélection avec jointure)

```
SELECT name FROM actor JOIN movie ON id = director
```

# Base de donnée movie.sqlite

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
        votes INTEGER, director INTEGER)
```

**Question 4.** Obtenir nom des réalisateurs et notes moyennes de leur films.

# Base de donnée movie.sqlite

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
       votes INTEGER, director INTEGER)
```

**Question 4.** Obtenir nom des réalisateurs et notes moyennes de leur films.

**Réponse :** (jointure avec agrégation par groupe)

```
SELECT name, AVG(score) FROM actor JOIN movie  
       ON actor.id = movie.director GROUP BY movie.director
```



# Base de donnée movie.sqlite

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
        votes INTEGER, director INTEGER)
```

**Question 5.** Comment obtenir le casting du film Star Wars ?

# Base de donnée movie.sqlite

```
actor ( id INTEGER, name VARCHAR(35))  
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)  
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,  
       votes INTEGER, director INTEGER)
```

**Question 5.** Comment obtenir le casting du film Star Wars ?

**Réponse :** (jointure à 3 tables)

```
SELECT actor.name FROM actor JOIN casting JOIN movie  
      ON actor.id = casting.actorid  
      AND movie.id=casting.movieid  
      AND movie.title="Star Wars"
```