



Introduction à MySQL

Objectifs

- Introduction aux bases de données
- Introduction à SQL
- Les types de données
- Les requêtes
- les jointures
- les agrégations
- les opérations ensemblistes

Intro Bases de données

Qu'est-ce qu'une base de données ?

Data ?

BDD ?

- * un outil utilisé pour stocker et gérer une collection organisée de données, de manière à ce qu'elle puisse être facilement consultée et gérée.
- * Vous pouvez organiser les données en tables, en lignes, en colonnes et les indexer pour faciliter la recherche d'informations pertinentes.

SGBD ?

- * Les gestionnaires de bases de données (SGBD) créent une base de données de manière à ce qu'un seul ensemble de programmes logiciels donne accès aux données à tous les utilisateurs.

Il existe de nombreuses bases de données disponibles telles que MySQL, Sybase, Oracle, MongoDB, Informix, PostgreSQL, SQL Server, etc.

SQL ?

(Structured Query Language) est utilisé pour manipuler les données stockées dans une base de données. SQL dépend de l'algèbre relationnelle et du calcul relationnel en tuples

SGBD

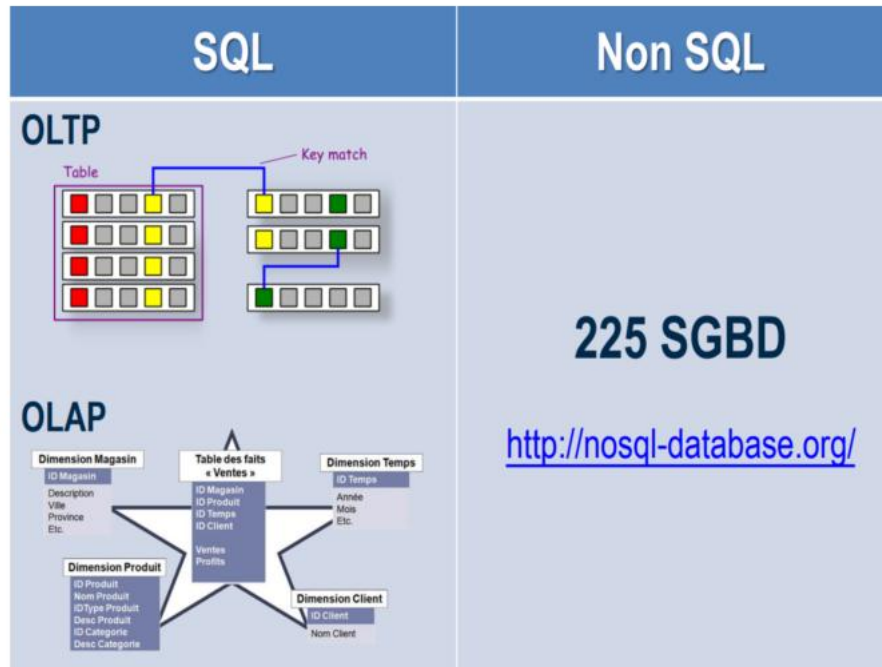
Avantages	Inconvénients
Contrôle de la redondance Il stocke toutes les données dans un seul fichier de base de données, ce qui permet de contrôler la redondance des données.	Taille Il occupe un espace disque important et nécessite une grande mémoire pour fonctionner efficacement.
Partage des données Un utilisateur autorisé peut partager les données entre plusieurs utilisateurs.	Coût Le SGBD nécessite un processeur de données à haute vitesse et une mémoire plus grande pour exécuter le logiciel SGBD, ce qui le rend coûteux.
Sauvegarde Il offre un sous-système de sauvegarde et de récupération. Ce système de récupération crée automatiquement des données en cas de défaillance du système et restaure les données si nécessaire.	Complexité Le SGBD crée une complexité et des exigences supplémentaires.
Interfaces utilisateur multiples Il propose différents types d'interfaces utilisateur comme GUI, interfaces d'application.	

Un SGBDR est un SGBD tabulaire qui maintient la sécurité, l'intégrité, la précision et la cohérence des données.

Les Types de Bases de données

Online Transaction Processing

Online Analytical Processing



Clés-Valeurs

Documents

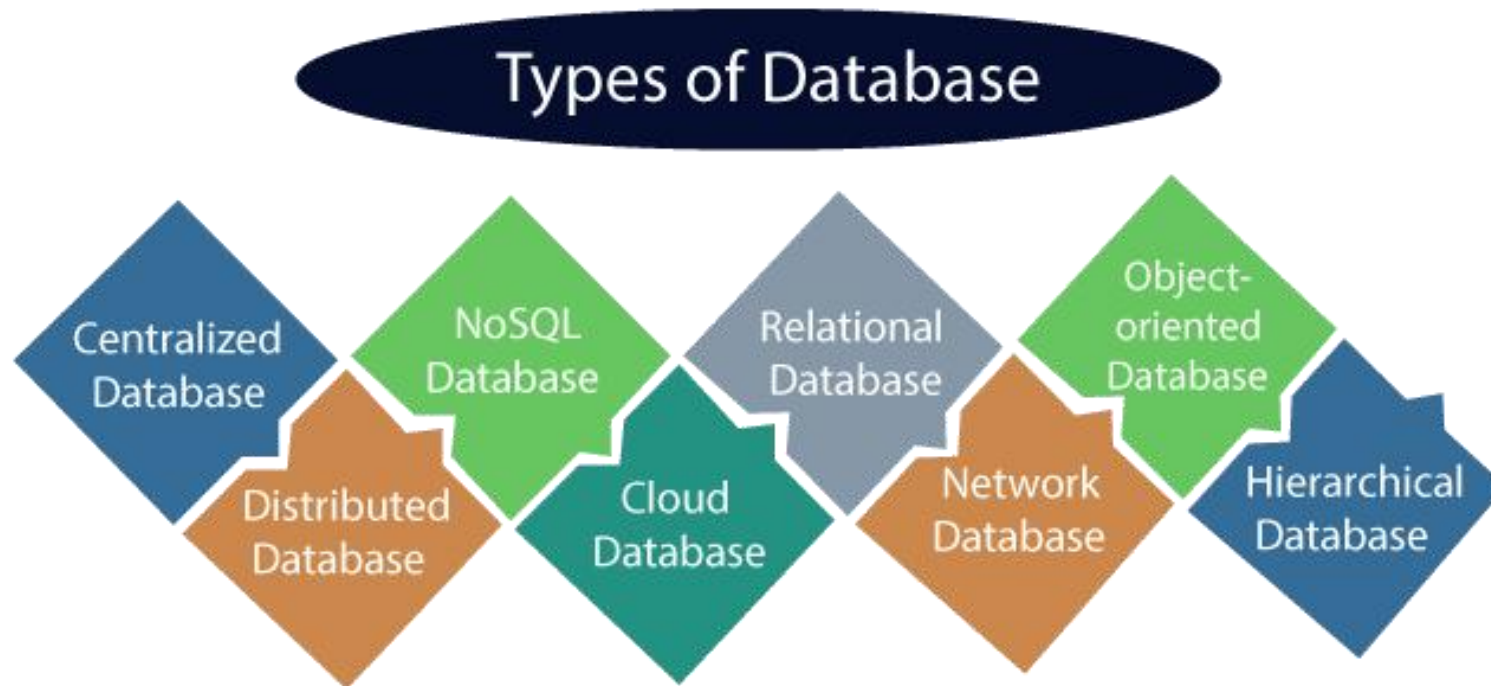
Colonnes

Graphe

<https://www.javatpoint.com/what-is-database>

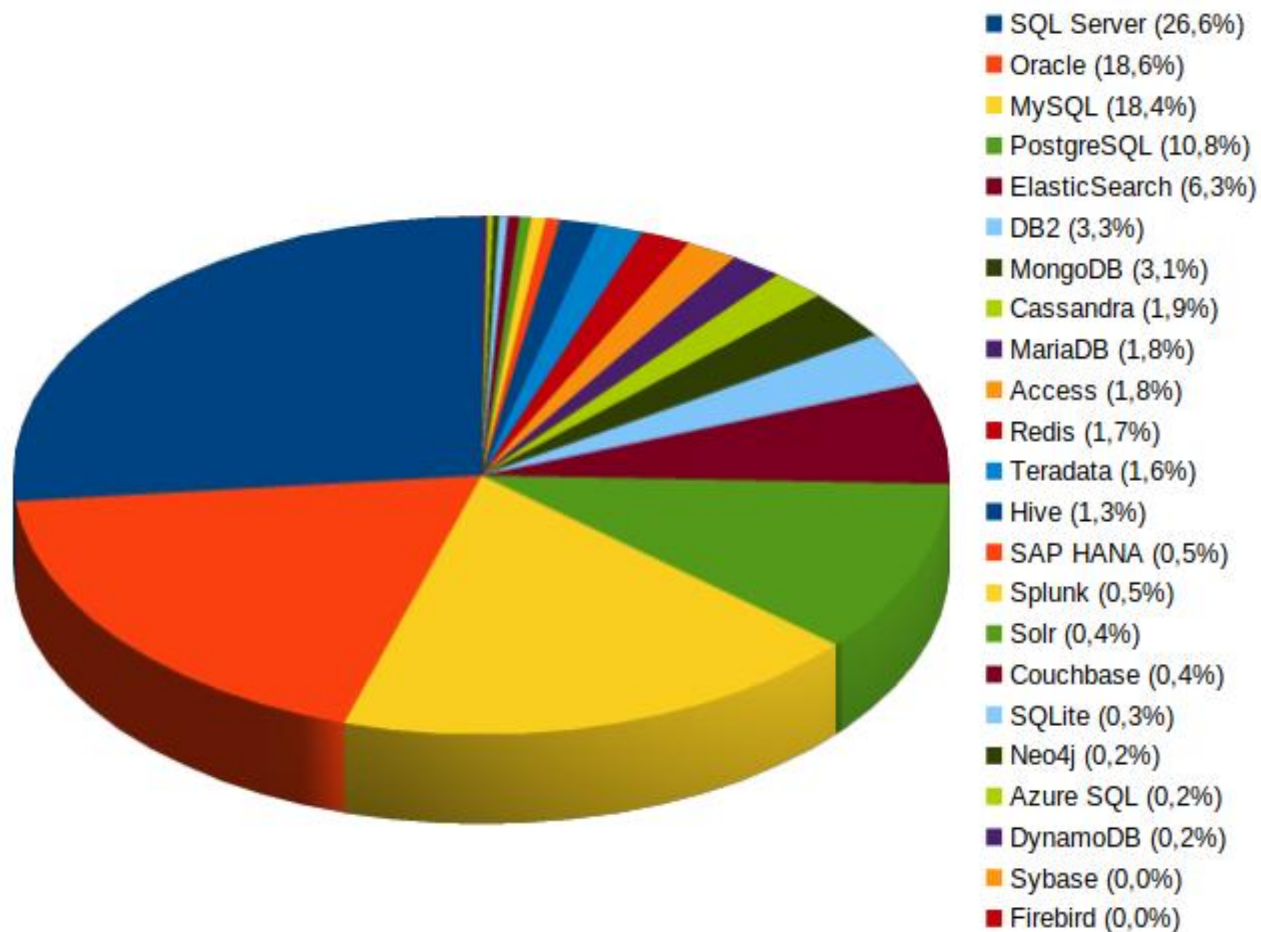
Memcached, Redis, Coherence (key-value store)
MongoDB, CouchDB, (Document-based)
Cassandra (Columns)
Neo4j(Graph)
HBase, Big Table (Tabular)

Les Types de BAses de données

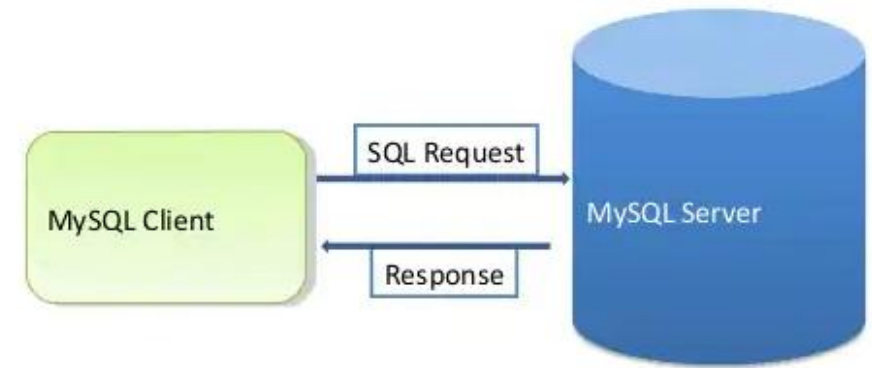


SGBD

Popularité des SGBD dans les offres d'emploi en 2020



SGBDR



Une architecture MySQL doit comprendre :

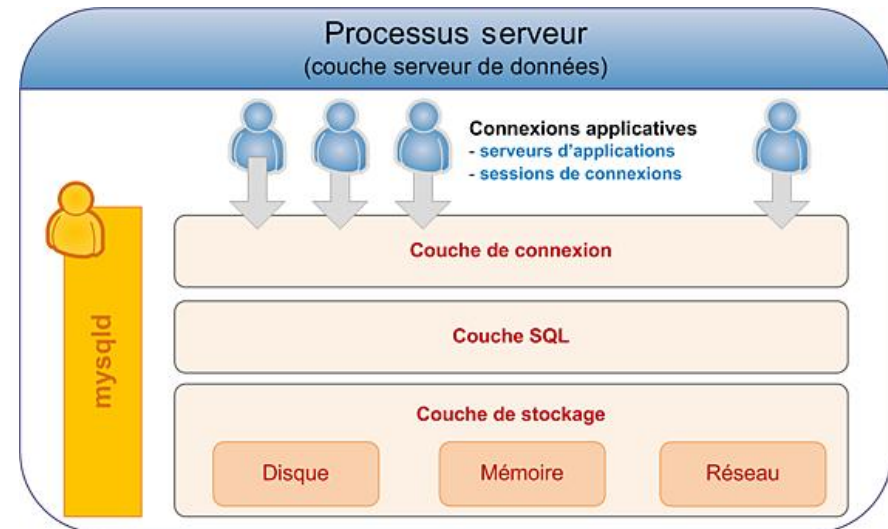
- le serveur de données MySQL Server
- des programmes clients

On peut trouver aussi des programmes non clients d'administration et des utilitaires MySQL.

* Le serveur de données accepte les connexions de clients effectuant des demandes d'extraction ou de mises à jour de données.

* Une communication client/serveur est alors établie entre systèmes d'exploitation identiques ou différents.

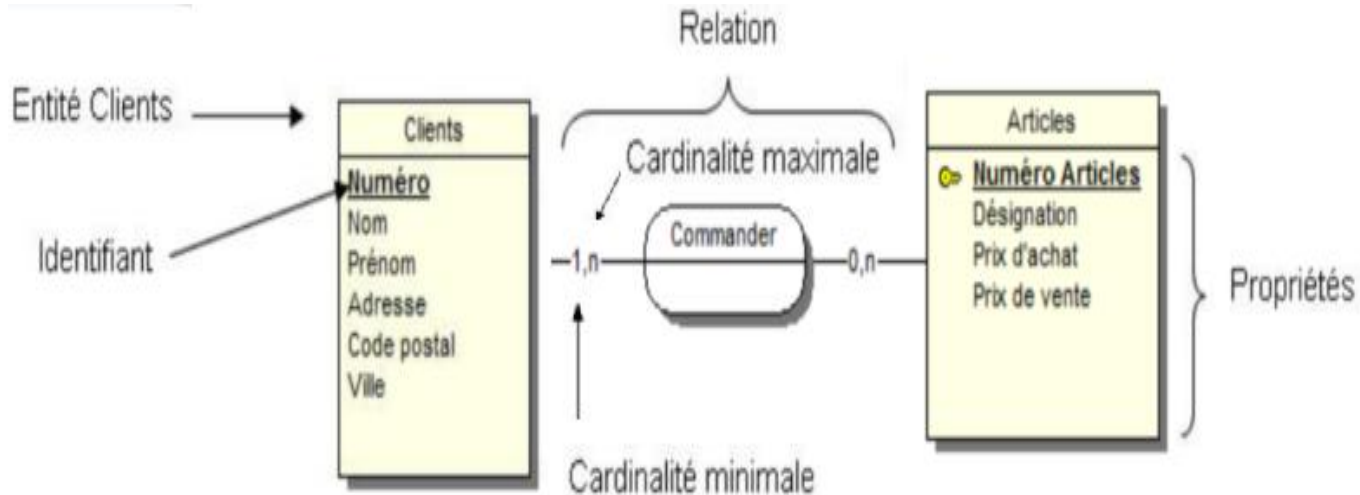
* Les programmes clients se connectent à l'instance MySQL pour accéder ou modifier des données.



Le Modèle Relationnel

Outre le système de gestion, le modèle historique, et encore aujourd'hui largement utilisé comme base de données opérationnelles, est le modèle dit **relationnel**.

Edgar Frank Codd [[voir détails](#)], mathématicien et chercheur chez IBM à la fin des années 1960, est considéré comme le père de ce modèle alors qu'il étudiait de nouvelles méthodes pour gérer de grandes quantités de données. En 1970, il publia un article où il proposait de stocker des données hétérogènes dans des **tables**, permettant d'établir des **relations** entre elles.



Structure de la Base de données

Example taken from world db MySQL



	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP
	UGA	Uganda	Africa	Eastern Africa	241038.00	1962	21778000	42.9	6313.00
	UKR	Ukraine	Europe	Eastern Europe	603700.00	1991	50456000	66.0	42168.00
	UMI	United States Minor ...	Oceania	Micronesia/Caribbean	16.00	NULL	0	NULL	0.00
	URY	Uruguay	South America	South America	175016.00	1828	3337000	75.2	20831.00
	USA	United States	North America	North America	9363520.00	1776	278357000	77.1	8510700.00
	UZB	Uzbekistan	Asia	Southern and Central Asia	447400.00	1991	24318000	63.7	14194.00
	VAT	Holy See (Vatican Ci...	Europe	Southern Europe	0.40	1929	1000	NULL	9.00

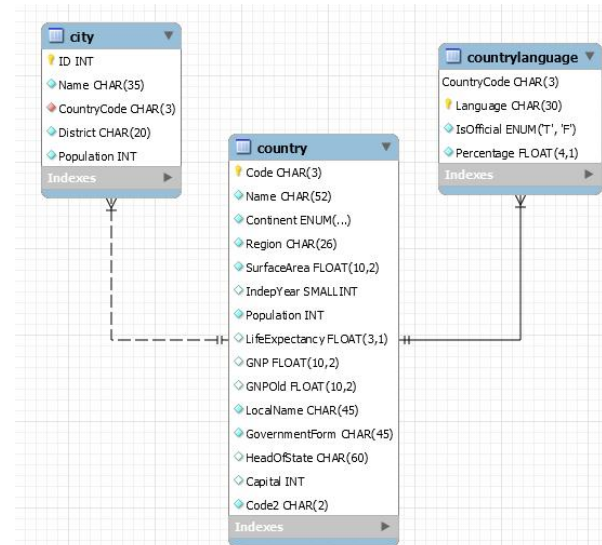
Country
Table

CountryCode	Language	IsOfficial	Percentage
ABW	Dutch	T	5.3
ABW	English	F	9.5
ABW	Papiamentu	F	76.7
ABW	Spanish	F	7.4

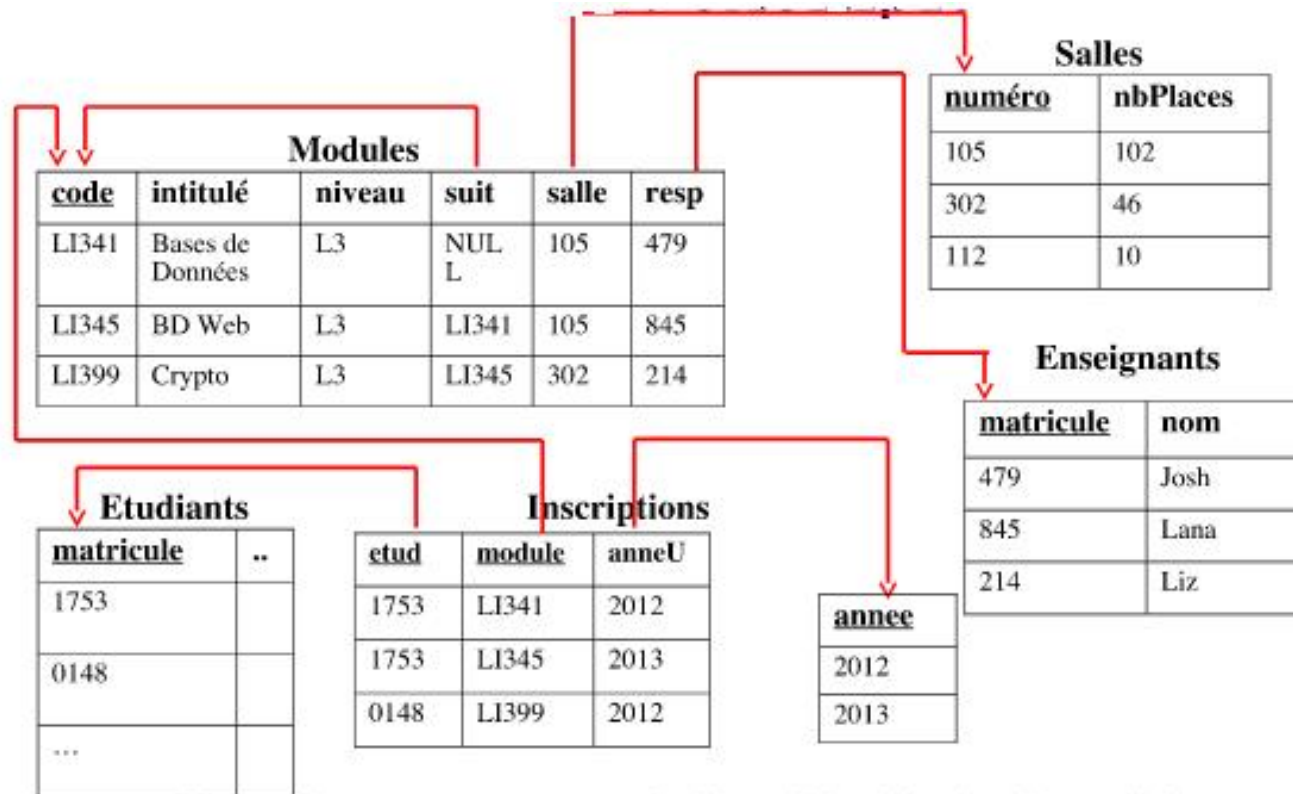
Countrylanguage Table

ID	Name	CountryCode	District	Population
3793	New York	USA	New York	8008278
3794	Los Angeles	USA	California	3694820
3795	Chicago	USA	Illinois	2896016
3796	Houston	USA	Texas	1953631
3797	Philadelphia	USA	Pennsylvania	1517550
3798	Phoenix	USA	Arizona	1321045
3799	San Diego	USA	California	1223400
3800	Dallas	USA	Texas	1188580

City Table



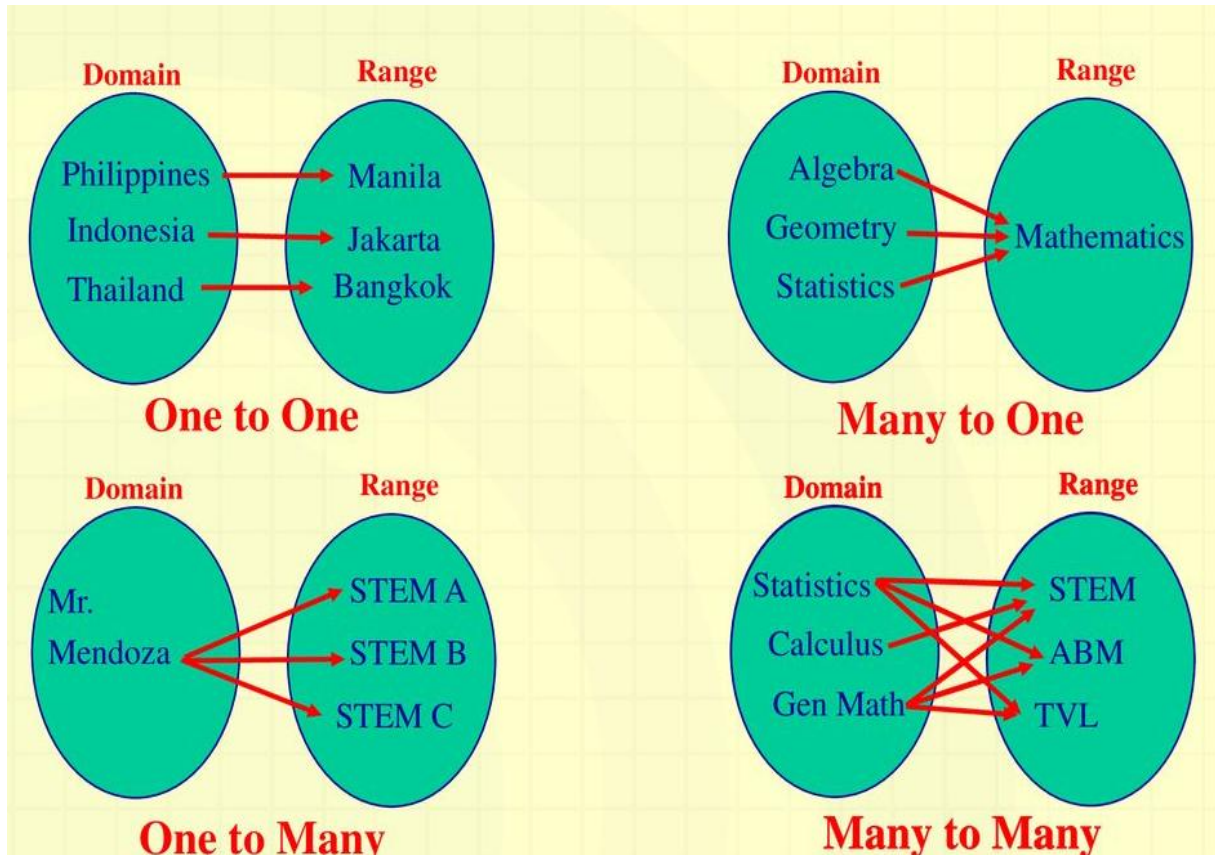
Les Contraintes



Des clés étrangères peuvent composer la clé candidate/primaire d'une relation:

- Exemple: la clé {etud, module} de la table **Inscriptions**

Plusieurs Modèles de Relations



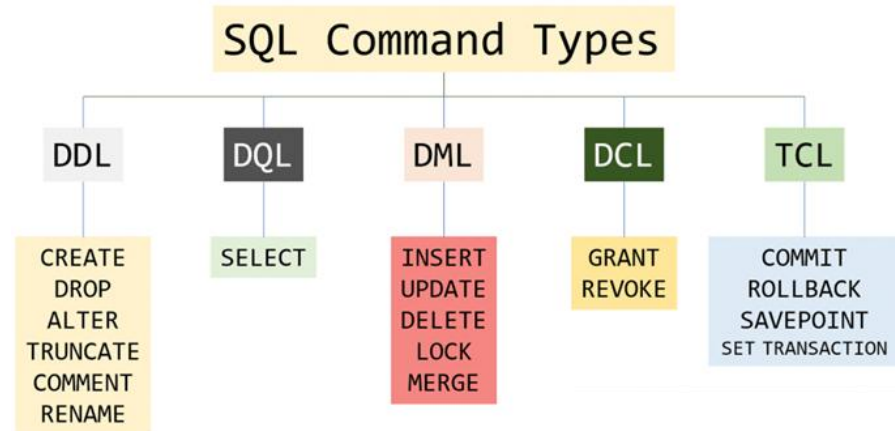
SQL

SQL (Structured Query Language) est un langage de programmation utilisé pour gérer et interagir avec des bases de données relationnelles. Les bases de données relationnelles stockent les données sous forme de **tables**, ce qui permet de gérer facilement des données structurées.

SQL signifie Structured Query Language (langage de requête structuré). Il s'agit d'un langage que vous pouvez utiliser pour accéder aux données de votre base de données, telles que :

- Lecture des données
- Ajout de nouvelles données
- Mise à jour des données existantes
- Suppression de données
- Modification de la structure de votre base de données

CRUD: Create ,Read,Update, Delete

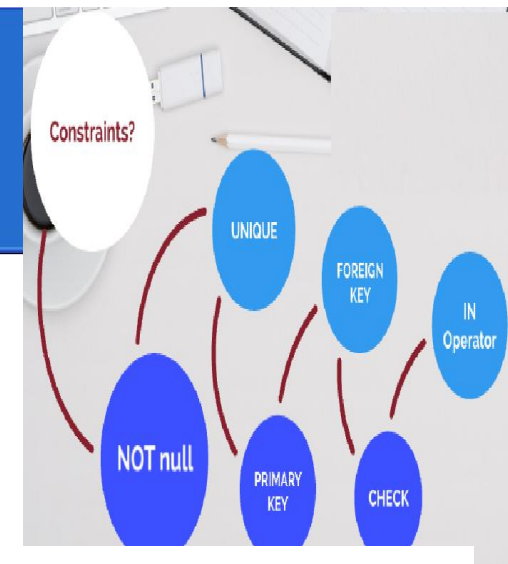


Les contraintes du modèle relationnel

Assurer la cohérence et l'intégrité des données

✎ Principales contraintes :

- ◇ Clé primaire (Primary Key - PK) : Un identifiant unique pour chaque ligne d'une table
- ◇ Clé étrangère (Foreign Key - FK) : Relie une table à une autre
- ◇ Unique : Empêche les doublons (ex : adresse email)
- ◇ Not Null : Certains champs doivent être obligatoirement remplis
- ◇ CHECK (<condition>) : contrôle la validité de la valeur des attributs spécifiés dans la condition dans le cadre d'une restriction de domaine



Produits			
Référence	Nom	En stock	Prix (€)
554871	Confiture de fraise 250g	true	4.80
543154	Gelée de coing 300g	true	3.75
554745	Confiture pêche-menthe 250g	true	5.20
562119	Gelée pomme-rhubarbe	false	4.90
553557	Gelée de fruits rouges	true	3.90

clé primaire

Ventes			
Numéro	Référence produit	Date	Quantité
1	543154	21/08/2017	1
2	562119	21/08/2017	2
3	543154	24/08/2017	5
4	553557	13/09/2017	2

clé primaire

clé étrangère

fait référence à la clé primaire de la relation Produits

enregistrements en relation

concepts Clé: Contraintes

Contraintes d'intégrité sur une table

Les contraintes d'intégrité sur une table sont :

- * PRIMARY KEY (<liste d'attributs>) : définit les attributs de la liste comme la clé primaire

- * UNIQUE (<liste d'attributs>) : interdit que deux tuples de la relation aient les mêmes valeurs pour l'ensemble des attributs de la liste.

- * FOREIGN KEY (<liste d'attributs>) REFERENCES <nom table>(<nom colonnes>) : contrôle l'intégrité référentielle entre les attributs de la liste et la table et ses colonnes spécifiées

- * CHECK (<condition>) : contrôle la validité de la valeur des attributs spécifiés dans la condition dans le cadre d'une restriction de domaine

Clé candidate

La clause UNIQUE NOT NULL sur un attribut ou un groupe d'attributs définit une clé candidate non primaire.

```
CREATE TABLE Personne (  
  N°SS CHAR(13) PRIMARY KEY,  
  Nom VARCHAR(25) NOT NULL,  
  Prenom VARCHAR(25) NOT NULL,  
  Age INTEGER(3) CHECK (Age BETWEEN 18 AND 65),  
  Mariage CHAR(13) REFERENCES Personne(N°SS),  
  Codepostal INTEGER(5),  
  Pays VARCHAR(50),  
  UNIQUE (Nom, Prenom),  
  FOREIGN KEY (Codepostal, Pays) REFERENCES Adresse (CP, Pays)  
);  
  
CREATE TABLE Adresse (  
  CP INTEGER(5) NOT NULL,  
  Pays VARCHAR(50) NOT NULL,  
  Initiale CHAR(1) CHECK (Initiale = LEFT(Pays, 1)),  
  PRIMARY KEY (CP, Pays)  
);
```


Les commandes SQL de base

SQL est utilisé pour définir et gérer la structure des bases de données, y compris la création, la modification et la suppression de tables.

Création de tables

Pour créer une nouvelle table dans une base de données, vous utiliserez la commande CREATE TABLE.

```
CREATE TABLE clients(  
  id INT PRIMARY KEY,  
  nom VARCHAR(50),  
  prenom VARCHAR(50),  
  age INT  
);
```

Cette commande crée une table nommée "clients" avec quatre colonnes : "id", "nom", "prenom" et "age". La colonne "id" est définie un identifiant unique, on appelle ça aussi une clé primaire (PRIMARY KEY).

Suppression de tables

Pour supprimer une table, utilisez la commande DROP TABLE. Attention, cette commande supprime définitivement la table et toutes ses données.

```
DROP TABLE clients;
```

Cela supprime la table "clients" de la base de données.

Modification de tables

* **Ajout de colonnes** à une table existante, avec
ALTER TABLE..ADD

```
ALTER TABLE clients  
ADD email VARCHAR(100);
```

Cela ajoute une colonne "email" à la table "clients".

* **Modification de colonnes** : modifier le type de données ou le nom d'une colonne, avec ALTER TABLE .. MODIFY.

```
ALTER TABLE clients  
MODIFY COLUMN age INT UNSIGNED;
```

Cela modifie la colonne "age" pour qu'elle accepte uniquement des valeurs entières non signées.

Les commandes SQL de base

INSERT INTO - Insertion de données

La commande INSERT INTO permet d'ajouter de nouvelles données à une table.

```
INSERT INTO clients(nom, prenom,age) VALUES (Dupont, Alice,21);
```

Cela insérera un nouvel employé dans la table "client" avec les valeurs spécifiées.

UPDATE - Mise à jour de données

La commande UPDATE permet de modifier des données existantes dans une table.

```
UPDATE clients SET age= 30 WHERE nom = Dupont and prenom=Alice ;
```

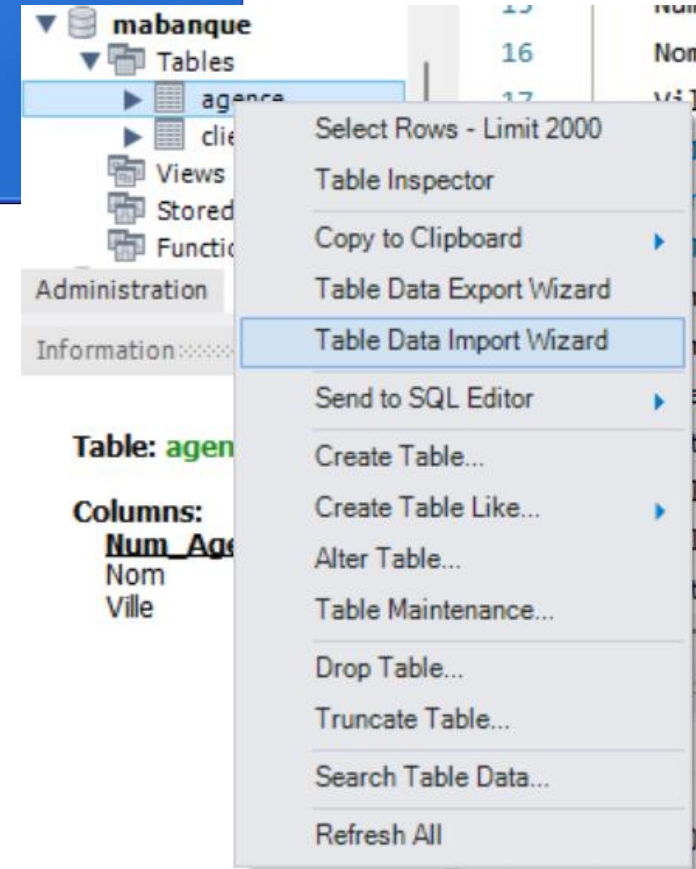
Cela mettra à jour le prenom du client dont le nom est "Dupont"

DELETE - Suppression de données

La commande DELETE permet de supprimer des données d'une table. Par exemple :

```
DELETE FROM clients WHERE nom = Dupont;
```

Cela supprimera tous les enregistrements dans la table "clients" où le nom est "Dupont".



Les commandes SQL de base

SELECT - Récupération de données

La commande SELECT est utilisée pour récupérer des données depuis une table.

```
SELECT * FROM clients;
```

Cela récupérera toutes les données de la table "clients". Vous pouvez spécifier des colonnes spécifiques au lieu de "*", si nécessaire.

```
SELECT nom, prenom FROM clients;
```

WHERE - Filtrage de données

La clause WHERE est utilisée pour filtrer les données dans une requête SELECT, UPDATE ou DELETE. Elle permet de spécifier des conditions pour sélectionner les données appropriées.

```
SELECT * FROM clients WHERE age > 25;
```

Cela récupérera tous les employés dont l'âge est supérieur à 25 ans.

ORDER BY - Tri des résultats

La clause ORDER BY permet de trier les résultats d'une requête selon une ou plusieurs colonnes.

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

Cela triera les résultats par ordre alphabétique croissant du nom

```
SELECT [ ALL | DISTINCT ] { select_list }  
[ INTO new_table ]  
FROM { table_source }  
[ WHERE search_condition ]  
[ GROUP BY group_expression ]  
[ HAVING search_condition ]  
[ ORDER BY order_clause ]
```

concepts Clé: Types

Types de Bases

Types alphanumériques

- *char(n)* ou character
- *varchar(n)* ou varying character

Types Numériques

- *int* ou *integer*, *smallint*
- *numeric(t,d)*, *decimal(t,d)*
- *real*, double precision
- *float(n)*

types/domaines

Etudiants(matricule : *Number*,
nom : *Varchar*,
prenom : *Varchar*,
adresse : *Varchar*)

Modules(code : *Number*,
intitule : *Varchar*,
niveau : *Varchar*,
salle : *Number*)

Salles(numero : *Number*,
capacite : *Number*)

Type of Data	Format	Example
Text	'Single Quotes'	Name Between 'Canton' and 'Denver'
Date, DateTime, Timestamp, Year	'YYYY-MM-DD', 'YYYY-MM-DD HH:MI:SS', 'YYYY' or 'YY'	startDate between '2015-01-01' and '2016-12-31'
number	No quotes	IndepYear > 1900
Boolean/ enum('t', 'f')	'T' 'F' or true / false	isOfficial = 't' ; isOfficial = true

concepts Clé: fonctions

Fonctions SQL

▪ Fonctions d'agrégation

- SQL AVG()
- SQL COUNT()
- SQL MAX()
- SQL MIN()
- SQL SUM()

▪ Fonctions mathématiques / numérique

- SQL RAND()
- SQL ROUND()

▪ Fonctions de chaînes de caractères

- SQL CONCAT()
- SQL LENGTH()
- SQL REPLACE()
- SQL SOUNDEX()
- SQL SUBSTRING()
- SQL LEFT()
- SQL RIGHT()
- SQL REVERSE()
- SQL TRIM()
- SQL LTRIM()
- SQL RTRIM()
- SQL LPAD()
- SQL UPPER()
- SQL LOWER()
- SQL UCASE()
- SQL LCASE()
- SQL LOCATE()
- SQL INSTR()

▪ Fonctions de dates et d'heures

- SQL DATE_FORMAT()
- SQL DATEDIFF()
- SQL DAYOFWEEK()
- SQL MONTH()
- SQL NOW()
- SQL SEC_TO_TIME()
- SQL TIMEDIFF()
- SQL TIMESTAMP()
- SQL YEAR()

Mise en place de MySQL

Installer votre environnement SQL avec Mysql(server) et le client (wysql workbench)

MySQL: <http://dev.mysql.com/downloads/mysql/#downloads>

MySQL Workbench:

<https://dev.mysql.com/downloads/workbench/>

Ressources Utiles :

https://www.youtube.com/watch?v=X_umYKqKaF0https://www.youtube.com/watch?v=X_umYKqKaF0

Mise en place de MySQL

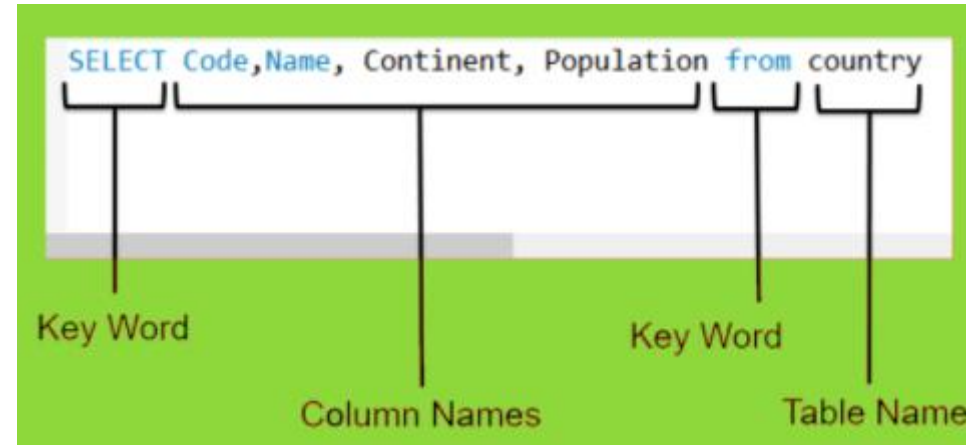
On peut trouver aussi des programmes non clients d'administration et des utilitaires MySQL.

Programme	Description
mysql	Permet d'exécuter des requêtes ou de faire des restaurations logiques de données.
mysqladmin	Permet d'administrer le serveur en ligne de commande sur une instance démarrée.
mysqlcheck	Permet de vérifier l'intégrité des tables d'une base de données en ligne.
mysqldump	Permet de réaliser une sauvegarde logique des données au format de sortie SQL.
mysqlimport	Permet d'importer des fichiers de données issues d'un format de type texte.
mysqlshow	Permet d'afficher des informations sur la base de données, sur les tables et ses colonnes.
mysqlslap	Permet de simuler une charge utilisateur sur une base de données au niveau de l'instance.

SQL Cheat Sheet

The most Important Commands

- CREATE DATABASE - creates a new database
- CREATE TABLE - creates a new table
- INSERT INTO - inserts new data into a database
- ALTER DATABASE - modifies a database
- ALTER TABLE - modifies a table
- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index



Application

- 1- Créer une base de données TP1
- 2- Créer dans TP1, la table clients décrite dans le cours
- 3- Tester les commandes SQL liées au CRUD
- 4- Exploiter les différentes commandes SQL

Pré-requis:

Mysql installé

Application

vous gérez une base de données pour une maison de disques. Vous avez besoin de suivre les artistes et leurs albums pour gérer leur carrière musicale.

La table "**Artistes**" stocke des informations sur les artistes tels que leur nom, leur genre musical et l'année de début de leur carrière.

La table "**Albums**" contient des détails sur les albums, y compris leur titre, l'ID de l'artiste associé et l'année de sortie.

La table "**Chansons**" stocke des informations sur les chansons individuelles, y compris leur titre, l'ID de l'album auquel elles appartiennent et leur durée.

- 1- Créer une base de données 'TP2 pour gérer des informations sur des artistes musicaux et leurs albums.
- 2- Créer dans TP2 ,les tables Artistes et Albums
- 3- Insérer un jeux de données (4 lignes minimum) et afficher es tables
- 4- Ajouter la table Chansons avec au moins Une chanson par album
- 5- Trouvez toutes les chansons d'un album spécifique (id 1 par exemple)

Les requetes simples

Requetage Simple: Projection

Une projection est une sélection de colonnes d'une table, sur la base d'une liste d'attributs placés après le `SELECT` et séparés par une virgule.

Projection

Sélection d'un nombre restreint de colonnes (ou attributs) d'une table

Ex : les noms et prénoms uniquement des étudiants

Attention : il y a risque d'avoir des doublons

Ex : le sexe des étudiants

Table			

Résultat

```
use employees;  
SELECT name,surname  
FROM People;
```

name	surname
Jean	Dupont
François	Bonnevialle
Thierry	Dupont
Michelle	Lestable
Marc	

Requetage Simple: Restriction

Une restriction est une sélection de lignes d'une table, sur la base d'une condition à respecter, définie à la suite du terme WHERE. Cette condition peut être simple {>, =, <=, >=, <>} ou complexe: une combinaison de comparaisons à l'aide de 'AND', de 'OR' et de 'NOT' (attention donc aux parenthèses dans ce cas).

Restriction

Sélection d'un nombre restreint de tuples (ou lignes) d'une table, selon un ou plusieurs critères

Ex : les étudiants de moins de 19 ans

Table

Résultat

```
SELECT *  
FROM People  
WHERE surname = "Dupont"
```

id	name	surname	birth_date	number_of_children
1	Jean	Dupont	1990-08-13	1
3	Thierry	Dupont	1990-03-13	2

Les Operateurs de comparaison

Une restriction est une sélection de lignes d'une table, sur la base d'une condition à respecter, définie à la suite du terme WHERE. Cette condition peut être simple {>, =, <=, >=, <>} ou complexe: une combinaison de comparaisons à l'aide de 'AND', de 'OR' et de 'NOT' (attention donc aux parenthèses dans ce cas).

>	<i>Strictement supérieur</i>
>=	<i>Supérieur ou égal</i>
<	<i>Strictement inférieur</i>
<=	<i>Inférieur ou égal</i>
=	<i>égal</i>
!= (ou <>)	<i>Différent de</i>

```
SELECT *  
FROM People  
WHERE number_of_children <=2;
```

id	name	surname	birth_date	number_of_children
1	Jean	Dupont	1990-08-13	1
2	François	Bonnevialle	1982-09-01	2
3	Thierry	Dupont	1990-03-13	2
5	Marc		1995-08-23	0

Les Operateurs: AND et OR

id	name	surname	birth_date	number_of_children
1	Jean	Dupont	1990-08-13	1
2	François	Bonnevialle	1982-09-01	2
3	Thierry	Dupont	1990-03-13	2
4	Michelle	Lestable	1985-07-12	3
5	Marc		1995-08-23	0

- * Les opérateurs logiques testent le caractère vrai ou faux d'une condition. Les opérateurs logiques, comme les opérateurs de comparaison, retournent un type de données booléen de valeur TRUE, FALSE ou UNKNOWN.
- * Les opérateurs logiques AND et OR peuvent être utilisées avec la commande WHERE pour combiner des conditions.

```
SELECT nom_colonnes  
FROM nom_table  
WHERE condition1 AND condition2
```

```
SELECT *  
FROM People  
WHERE surname = "Dupont" AND number_of_children < 2;
```

id	name	surname	birth_date	number_of_children
1	Jean	Dupont	1990-08-13	1

```
SELECT nom_colonnes FROM nom_table  
WHERE condition1 OR condition2
```

```
SELECT *  
FROM People  
WHERE surname = "Dupont" or number_of_children < 2;
```

id	name	surname	birth_date	number_of_children
1	Jean	Dupont	1990-08-13	1
3	Thierry	Dupont	1990-03-13	2
5	Marc		1995-08-23	0

```
SELECT nom_colonnes FROM nom_table  
WHERE condition1 AND (condition2 OR condition3)
```

Les Operateurs: IN, BETWEEN

La clause IN permet de remplacer une série de OR dans la condition.

La clause BETWEEN permet de définir des intervalles de valeurs et donc eventuellement remplacer des OR et des AND.

```
SELECT *  
FROM People  
WHERE name IN ("Jean", "Thierry", "Marc");
```

id	name	surname	birth_date	number_of_children
1	Jean	Dupont	1990-08-13	1
3	Thierry	Dupont	1990-03-13	2
5	Marc		1995-08-23	0

```
SELECT *  
FROM People  
WHERE number_of_children BETWEEN 1 AND 3;
```

id	name	surname	birth_date	number_of_children
1	Jean	Dupont	1990-08-13	1
2	François	Bonnevialle	1982-09-01	2
3	Thierry	Dupont	1990-03-13	2
4	Michelle	Lestable	1985-07-12	3

Les Operateurs: LIKE

La clause LIKE permet de faire des recherches dans du texte en essayant de trouver les résultats qui correspondent à un pattern donné.

```
SELECT *  
FROM People  
WHERE name LIKE 'M%'
```

id	name	surname	birth_date	number_of_children
4	Michelle	Lestable	1985-07-12	3
5	Marc		1995-08-23	0

'C%'	Retourne toutes les valeurs qui commencent par la lettre C
'%C'	Retourne toutes les valeurs qui finissent par la lettre C
'%C%'	Retourne toutes les valeurs qui contiennent C (que ce soit la première, la dernière ou une lettre au milieu)
'_C%'	Retourne toutes les valeurs qui ont la lettre C en deuxième lettre

ORDER BY

* La clause ORDER BY permet de trier les résultats d'une requête par un ou plusieurs critères. On peut trier dans l'ordre croissant ou décroissant une colonne ou bien combiner les critères de tri.

* La clause AS permet de renommer a posteriori une colonne dans la liste des résultats mais n'affecte en rien les données en base.

```
SELECT name,surname, number_of_children as children
FROM People
ORDER BY children DESC, surname ASC;
```

name	surname	children
Michelle	Lestable	3
François	Bonnevialle	2
Thierry	Dupont	2
Jean	Dupont	1
Marc		0

- Trier par une colonne contenant du texte revient à trier par ordre lexicographique.

- Le tri se fait par ordre d'apparition dans la clause ORDER BY. On va d'abord trier par number_of_children décroissant puis par nom de famille croissant.

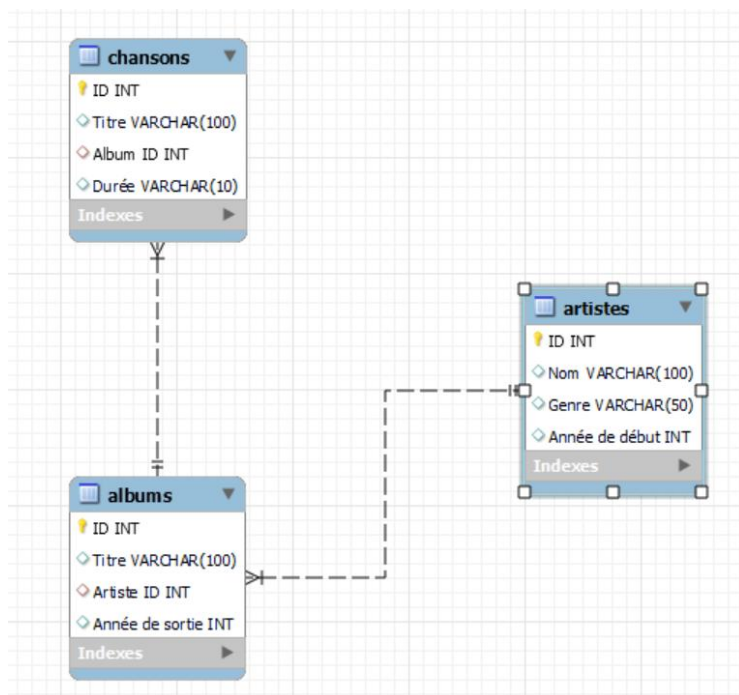
- La clause ASC n'est pas nécessaire, par défaut le tri est

```
SELECT name,surname, number_of_children as children
FROM People
ORDER BY 3 DESC, 2 ASC;
```

Application

On reprend la BDD Artistes , répondre aux requêtes suivantes:

1- Alimenter les 3 tables comme suit:



ID	Titre	Album ID	Durée
1	Billie Jean	1	4:54
2	Beat It	1	4:18
3	Imagine	2	3:03
4	Material Girl	3	3:53
5	Redemption Song	4	3:49
6	Purple Rain	5	8:41
7	Non, je ne regrette rien	6	2:23
8	La Bohème	7	4:05
9	Papaoutai	8	3:51

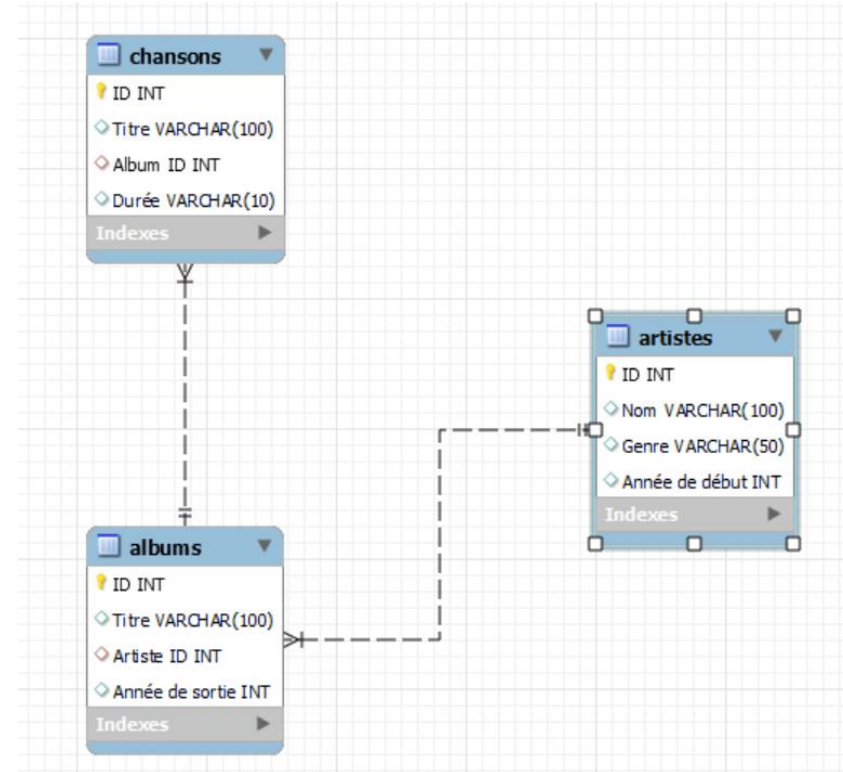
ID	Nom	Genre	Année de début
1	Michael Jackson	Pop	1964
2	The Beatles	Rock	1960
3	Madonna	Pop	1979
4	Bob Marley	Reggae	1963
5	Prince	R&B	1975
6	Édith Piaf	Chanson française	1935
7	Charles Aznavour	Chanson française	1946
8	Stromae	Pop	2000
9	Adele	Pop	2006

ID	Titre	Artiste ID	Année de sortie
1	Thriller	1	1982
2	Abbey Road	2	1969
3	Like a Virgin	3	1984
4	Legend	4	1984
5	Purple Rain	5	1984
6	La Vie en rose	6	1947
7	Sur ma vie	7	1954
8	Racine Carrée	8	2013

Application

2- répondre aux requêtes suivantes:

- Recherchez tous les artistes du genre "Pop"
- Obtenez la liste des albums sortis en 1984
- Trouvez toutes les chansons de l'album d'id 6
- Recherchez toutes les chansons dont la durée est supérieure à 4 minutes
- Listez les artistes par ordre alphabétique de leur nom
- Recherchez les artistes dont le nom commence par "M"
- Affichez les artistes qui ont commencé leur carrière entre 1990 et 2023
- Sélectionnez les artistes de genre "Pop" ou "Rock"
- Retrouvez les artistes de genre "pop" qui ont commencé leur carrière après 1990
- Affichez les artistes qui sont soit de genre "Reggae" soit de genre "Rock"



Hands On

Valider le chapitre 4 de la ressource suivante:

<https://my-learning.w3schools.com/tutorial/sql>

4. Database

0 of 20 lessons completed

Lesson 4.1 - Create DB

Lesson 4.2 - Drop DB

Lesson 4.3 - Backup DB

Lesson 4.4 - Create Table

Lesson 4.5 - Drop Table

Lesson 4.6 - Alter Table

Lesson 4.7 - Constraints

Lesson 4.8 - Not Null

Lesson 4.9 - Unique

Lesson 4.10 - Primary Key

Lesson 4.11 - Foreign Key

Lesson 4.12 - Check

Lesson 4.13 - Default

Lesson 4.14 - Index

Lesson 4.15 - Auto Increment

Lesson 4.16 - Dates

~~Lesson 4.17 - Views~~

~~Lesson 4.18 - Injection~~

~~Lesson 4.19 - Hosting~~

Lesson 4.20 - Data Types

Hands On

Valider les chap 1 et la moitié du 2 de la ressource suivante:

<https://my-learning.w3schools.com/tutorial/sql>

Your progress

✓ 1. Introduction

○ 2. Statements Part 1

- Lesson 2.1 - SELECT
- Lesson 2.2 - SELECT DISTINCT
- Lesson 2.3 - WHERE
- Lesson 2.4 - AND, OR, NOT
- Lesson 2.5 - ORDER BY
- Lesson 2.6 - INSERT INTO
- Lesson 2.7 - NULL Values
- Lesson 2.8 - UPDATE
- Lesson 2.9 - DELETE

Ressources utiles

1. La documentation sql.sh

2. La documentation de [MySQL](http://mysql.com)

3. Tuto SQL et MySQL

<https://www.youtube.com/playlist?list=PLjwdMgw5TTLXXpRlzDZq7d8iS6YXgnsIt>