

Introduction aux Bases de Données

Ulysse COUTAUD
coutaudu@gmail.com

Base de données (BDD) ?

Ensemble d'informations:

- Stockées.
- Consultées.
- Modifiées.

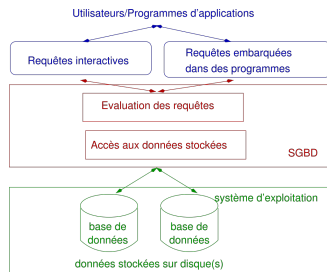
Exemples:

- Carnet d'adresses (quelques kilobytes).
- Comptes bancaires.
- Registre des personnels, des accès.
- Réservations des places dans les trains.
- Suivi des stocks, de la production, des commandes.
- Données de supervisions, programmes d'usages, ...
- World Data Centre for Climate (6 petabytes).

Système de Gestion de Base de Données (SGBD) ?

Ensemble d'outils permettant **l'organisation, le contrôle, la consultation et la modification** d'une base de données.

- Accès et manipulation standardisés.
 - Indépendance physique des données
 - Indépendance logique des données
 - Réduire la complexité
 - Stockage et accès optimisés.
- Contrôle de cohérence des données
- Sécurité des données
- Accès ou transactions (massivement) concurrentes.
- Fiabilité (sauvegardes, restaurations, ...)



Un exemple

Tracabilité et contrôle de flux dans l'industrie automobile.

Dans ce cours

Rudiments des Bases de Données:

- Le modèle "relationnel".
- Le langage SQL.
- PostgreSQL.
- Lire les données.
- Modifier les données.
- Concevoir une base de données simple.

Un peu de vocabulaire

- **BDD** Base De Données
 - **DB** Database
- **SGBD** Système de Gestion de Base de Données
 - **DBMS** Data Base Management System
- **SGBDR** Système de Gestion de Base de Données Relationnelles
 - **RDBMS** Relational Database Management System
- **SQL** Structured Query Language : Norme¹ de langage de manipulation de données relationnelles.

¹ISO/IEC 9075-1:2023

Les principaux acteurs

- **ORACLE®** :

- Oracle DB

-  MySQL

- Microsoft

- MS SQL:  Microsoft SQL Server

- Microsoft Access: 

- IBM: IBM DB2 

- SQLite:  SQLite



PostgreSQL

- **PostgreSQL**

- Open source

- Université de Berkeley

- Populaire et répandu ^a

^ahttps://db-engines.com/en/ranking_trend/

relational+dbms

Le modèle relationnel

Une TABLE = une "RELATION"

Clé primaire

Colonne/attribut

Employés(prénom,salaire,adresse,service)			
<u>Prénom</u>	Salaire	Adresse	Service
John	120	Randwick	Production
Mary	130	Wollolong	Maintenance
Peter	110	Randwick	Stock
Tom	120	Botany Bay	Production

Schéma de la relation

Ligne/"enregistrement"

Le modèle relationnel

Une TABLE = une "RELATION"

Clé primaire

Colonne/attribut

Schéma de la relation

Ligne/"enregistrement"

Employés(prénom,salaire,adresse,service)			
<u>Prénom</u>	Salaire	Adresse	Service
John	120	Randwick	Production
Mary	130	Wollolong	Maintenance
Peter	110	Randwick	Stock
Tom	120	Botany Bay	Production

Horaires(service,début,fin)		
<u>Service</u>	Début	Fin
Production	5	13
Maintenance	8	16
Stock	8	16

L'ensemble des schémas de relations + liens entre eux = le schéma de la base de données.

Se connecter à la base de données

TABLE Employes(prenom,salaire,adresse,service)				TABLE Horaires(service,debut,fin)		
Prenom	Salaire	Adresse	Service	Service	Debut	Fin
John	120	Randwick	Production	Production	5	13
Mary	130	Wollolong	Maintenance	Maintenance	8	16
Peter	110	Randwick	Stock	Stock	8	16
Tom	120	Botany Bay	Production			

Figure: BDD "atelier"

- Ouvrir une console (Win + cmd) et taper la commande "psql atelier postgres",
- Ou bien double cliquer sur le script Documents\SGBDR_L3PAII\fabriqueBDDConnectCLI.bat .
- Entrer le mot de passe "postgres".
- Lister les tables de la BDD courante \dt.

Clause SELECT

Afficher toute la table

```
SELECT * FROM nom_table;
```

Afficher certaines colonnes

```
SELECT colonne1, colonne2 FROM nom_table;
```

BDD "atelier":

TABLE Employes(prenom,salaire,adresse,service)			
<u>Prenom</u>	Salaire	Adresse	Service
John	120	Randwick	Production
Mary	130	Wollolong	Maintenance
Peter	110	Randwick	Stock
Tom	120	Botany Bay	Production

TABLE Horaires(service,debut,fin)		
<u>Service</u>	Debut	Fin
Production	5	13
Maintenance	8	16
Stock	8	16

Afficher les prénoms et salaires des employés.

Afficher toutes les informations des employés.

Clause SELECT avec contraintes

Afficher les lignes validant la condition

```
SELECT colonne1, colonne2  
FROM nom_table  
WHERE condition1;
```

Nombres:

=, !=, <, <=, >, >=,
BETWEEN nombre1 AND
nombre2,
IN (nombre1, nombre2, nombre3)

Texte:

colonne1 = 'texte' (!= <>)
LIKE (caractère joker: %)
IN ('mot1', 'mot2', 'mot3')

BDD "atelier": employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

Afficher les prénoms des employés du service production.

Afficher les prénoms des employés qui gagne 110 ou moins.

Afficher les prénoms des employés dont l'adresse contient la chaîne "an".

Afficher les prénoms et salaires des employés du service Production et Maintenance.

Filtrer les résultats

Clause DISTINCT

```
SELECT DISTINCT colonne1 FROM nom_table;
```

BDD "atelier": employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

Afficher sans répétition la liste des salaires.

Afficher sans répétition les horaires de débuts de services.

Trier les résultats

Clause ORDER BY

```
SELECT colonne1  
FROM nom_table  
ORDER BY colonne1 ASC/DESC;
```

BDD "atelier": employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

Afficher toutes les informations sur les employes en les triants par ordre alphabétique sur les prénoms.

Afficher prénoms et adresses des employes en triant par salaires décroissants.

Limiter les résultats

Clauses LIMIT et OFFSET

```
SELECT colonne1  
FROM table  
LIMIT num_limit OFFSET num_offset;
```

BDD "atelier": employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

Afficher l'employé le moins bien payé.

Afficher le deuxième employé le moins bien payé.

Joindre des tables

Clause INNER JOIN

```
SELECT colonne1  
FROM nom_table1  
INNER JOIN nom_table2  
ON table1.colonneX=table2.colonneY;
```

Remarque

- 1: Le INNER JOIN est le JOIN par défaut, le mot clé "INNER" n'est pas nécessaire. Il est même très souvent omis.
- 2: En cas d'ambiguïté sur le nom de colonne, on ajoute en préfixe la table visée (ex: table1.colonne1). On peut également renommer la colonne résultat avec le mot clé "AS" (ex: table1.colonne1 AS T1C1)

BDD "atelier": `employees(prenom,salaire,adresse,service)` `horaires(service,debut,fin)`

Afficher la jonction complète des tables employees et horaires.

Afficher les horaires de John.

Afficher les noms et horaires de tous les employés du service production.

Joindre des tables

LEFT JOIN

Il y aura forcément chacune des lignes de table1. Si aucune correspondance avec table2: NULL.

RIGHT JOIN

Il y aura forcément chacune des lignes de table2. Si aucune correspondance avec table1: NULL.

Valeur NULL

Se filtre avec IS NULL ou IS NOT NULL

BDD "atelier": employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

Afficher la jonction de toutes les informations de la BDD

Afficher les prénoms des employés qui n'ont pas d'horaires.

Les expressions

```
SELECT colonne1, colonne2 * 10 AS colonne2_pourcent
FROM nom_table;
SELECT colonne1, colonne2 FROM nom_table
WHERE colonne2 * 10 > 80;
```

BDD "atelier": employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

Les salaires de la base sont journaliers. Afficher les prénoms et salaires mensuels (20 jours travaillés).

Les agrégats

```
SELECT COUNT(colonne1) from table;
```

Les fonctions d'agrégats:

COUNT, MIN, MAX, AVG, SUM.

BDD "atelier": employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

Afficher le nombre d'employés.

Afficher le salaire minimal.

Afficher le salaire moyen.

Afficher la masse salariale totale.

Les agrégats

Clause GROUP BY

SELECT colonne1, SUM(colonne2) FROM table GROUP BY colonne1; Applique la fonction d'aggrégat à chaque groupe ayant la même valeur en colonne1.

BDD "atelier": employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

Afficher la masse salariale de chaque service.

Afficher le salaire moyen par adresse.

Clause HAVING

Equivalent de WHERE pour les expressions.

Afficher le salaire moyen des groupes d'employés ayant tous le même salaire à une même adresse.

Votre nouveau meilleur ami:

`https://sql.sh/`