



Institut Préparatoire aux Etudes d'Ingénieurs de Tunis

Chapitre IV

INTRODUCTION AUX BASES DE DONNÉES

Mme Nesrine Ayed

PLAN

- Limites des fichiers.
- Introduction aux bases de données.
 - Définition.
 - SGBD.
 - Architecture SGBD.
 - Notions de base.
- ▶ L'algèbre relationnelle.
- ▶ Le langage SQL.
- ▶ la pratique en python :SQLite.

INTRODUCTION

- L'informatique repose sur le traitement automatique de l'information
 - Stocker des données,
 - Accéder aux données
 - Utiliser des données
 - Manipuler et traiter des données...
- Représenter et structurer les données dans des structures de données, des conteneurs de données.
 - Exemple : fichier

LIMITES DES FICHIERS

- L'utilisation du fichier exige la connaissance.
 - Du mode d'accès.
 - De la structure des enregistrement (la manière avec laquelle les données sont stockées.
 - L'emplacement du fichier....
- La modification d'une donnée peut entraîner une modification sur plusieurs lignes.
- La suppression d'une donnée peut entraîner une perte d'autres données.

LIMITES DES FICHIERS

- Redondance des données.
- Toute modification de la structure des enregistrements engendre la réécriture de tous les programmes manipulant ces fichier.

L'utilisation des fichiers engendre des systèmes d'information constitués :

- de programmes inflexibles et couteux.
- de données inaccessibles à tout non spécialiste.

BD: DÉFINITION

- Une base de données est un ensemble structuré de données:
 - enregistré sur des supports accessibles par l'ordinateur,
 - représentant des informations en rapport avec un thème ou une activité
 - pouvant être interrogé et mis à jour par une communauté d'utilisateurs.
- Schématiquement, une base de données est un ensemble de *tables contenant des données reliées entre elles par des relations.*
- *On accède à l'information par le biais de requêtes exprimées dans un langage spécifique.*

EXEMPLE DE BD D'UNE BIBLIOTHÈQUE

Livres			
id	auteur	titre	id_genre
1	1	Salambô	1
2	2	Quatre-vingt-treize	1
3	3	L'immortaliste	1
4	4	Aphorismes	2
5	1	Madame Bovary	1
...

Auteurs		
id	nom	prénom
1	Flaubert	Gustave
2	Hugo	Victor
3	Gide	André
4	wilde	Oscar
...

IPEIT - 2021/2022

Genre	
id_genre	genre
1	Roman
2	littérature
...	...

Emprunteurs

Id	nom	prénom	tel
1	Ben brahim	Nader	98201458
2	Alaya	Myriam	22515478
3	jilani	Amira	21548963
4	kilani	Youssef	55487265
5	khmir	Dorra	67458100
6	Ayari	mouna	
...	

Emprunts			
qui	quoi	date	rendu
1	1	10-sep-16	15-oct-16
6	1	12-sep-16	29-sep-16
2	1	20-sep-16	20-oct-16
2	2	29-sep-16	5-oct-16
6	2	2-oct-16	15-oct-16
3	5	9-oct-16	20-oct-16
1	5	9-oct-16	NULL
5	3	12-oct-16	29-oct-16
...

SGBD

- Un système de gestion de base de donnée (SGBD) est un ensemble de programmes qui assure la création, la gestion et l'accès à une base de donnée
- Un SGBD doit permettre l'ajout, la suppression, la modification et la recherche de données.
- L'interaction avec un SGBD se fait par l'intermédiaire de requêtes exprimées dans le langage SQL (*Structured Query Language*).

SGBD

○ Un SGBD permet d'assurer:

- **Un accès aux données en un temps raisonnable :** L'accès aux données se fait par l'intermédiaire d'un Langage de requêtes).
- **Une Non-redondance des données :** Pour éviter les problèmes lors des mises à jour, chaque donnée ne doit être présente qu'une seule fois dans la base.
- **Une Indépendance des données:** la manière dont les informations sont présentées aux utilisateurs diffère de la manière dont elles sont organisées, et celle-ci diffère de la manière dont les informations sont enregistrées.

SGBD

- **Une Cohérence des données :** Les données sont vérifiées automatiquement à chaque insertion, modification ou suppression.
- **Un Partage des données :** plusieurs utilisateurs peuvent accéder aux mêmes données au même moment de manière transparente.
- **une sécurité des données :** Les données doivent pouvoir être protégées contre les accès non autorisés.
- **Une résistance aux pannes :** Il faut pouvoir récupérer une base dans un état « sain » si une panne survient .

EXEMPLE DE SGBD

- Quelques SGBD
 - **Propriétaires:**
 - Informix,
 - Oracle,
 - Microsoft SQL Server...
 - **Libre(gratuit)**
 - PostgreSQL
 - MySql
 - SQLite...

ARCHITECTURE SGBD

- ▶ Une application est censée manipuler et traiter des données pour fournir un résultat (texte, courbe, fenêtre, bouton...)
- ▶ En fonction du type de l'application et de la nature des données à manipuler différents types d'architecture matérielles et logicielles existent:
 - **Architecture à un niveau**
 - **Architecture à 2 niveaux**
 - **Architecture à 3 niveaux**

ARCHITECTURE À UN NIVEAU

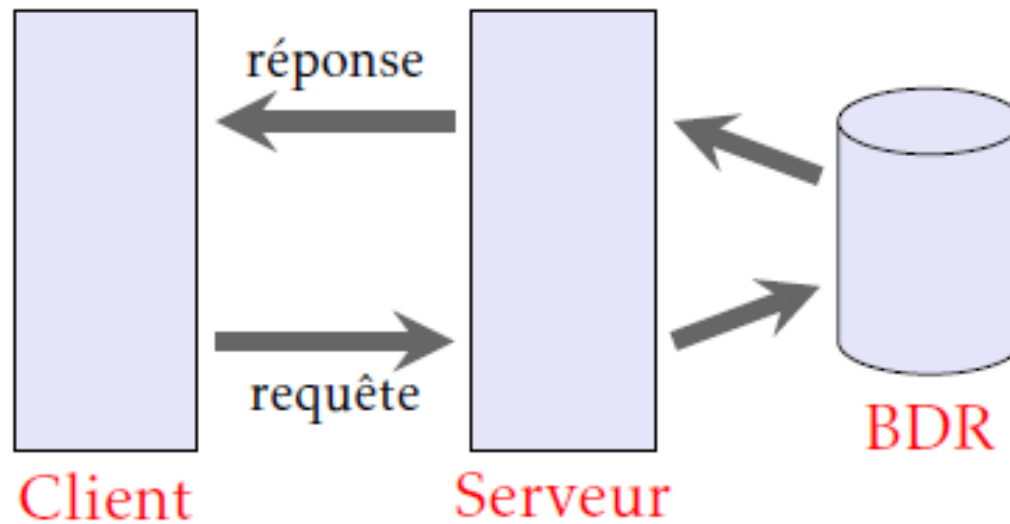
- ▶ Une base de donnée fonctionnant en local sur l'ordinateur de l'utilisateur.
- ▶ Des applications dont les données n'ont pas à être partagées avec d'autres utilisateurs.
- ▶ Exemple: Logiciels de messagerie

ARCHITECTURE À DEUX NIVEAUX

► Architecture client serveur

- Composée de N clients et un serveur
- Le client est celui qui envoie des requêtes(poste de travail où est installée une partie simple de l'application).
- Le serveur est celui qui renvoie les données.(machine puissante où sont installées une grande partie de l'application et la base de donnée prise en charge.
- Les clients sont connectés au serveur par divers procédés et protocoles.

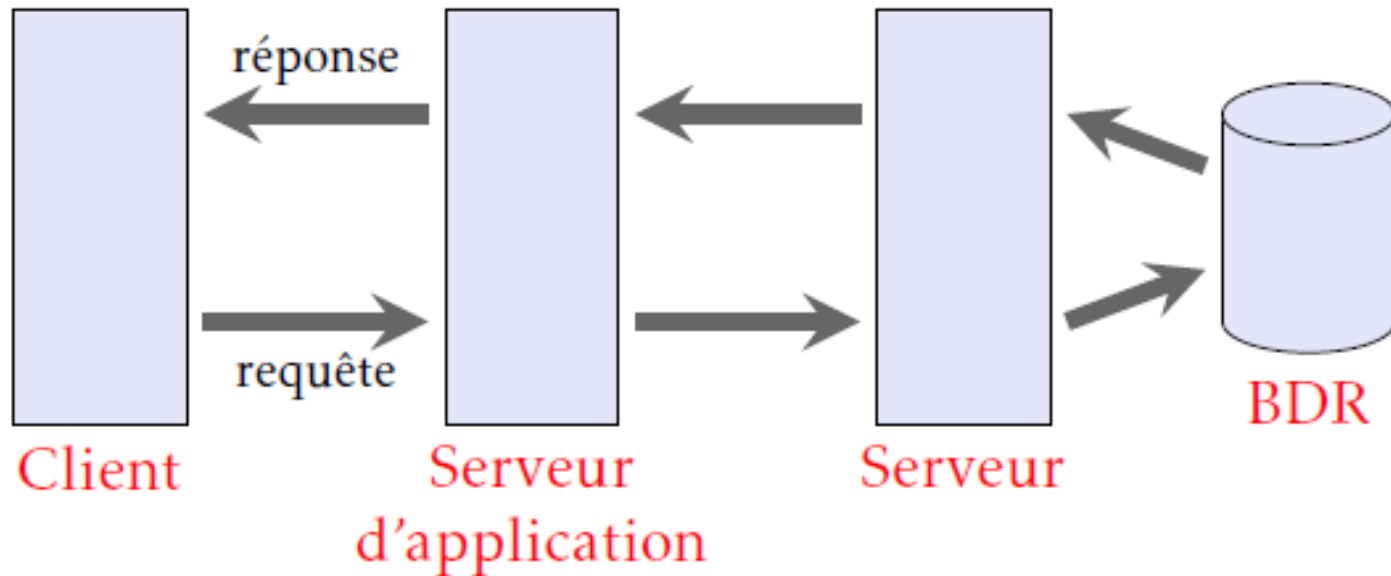
ARCHITECTURE À DEUX NIVEAUX



ARCHITECTURE À TROIS NIVEAUX

- ▶ Architecture classique rencontrée sur Internet, elle est composée par:
 - N clients
 - Un serveur d'application qui va traiter la demande du client.
 - Un serveur de base de données qui va traiter les requêtes et envoyer les données nécessaires.
- ▶ Meilleure répartition de charge entre l'accès aux données et le traitements des données

ARCHITECTURE À TROIS NIVEAUX



BD: NOTIONS DE BASE

▶ Relation

- Les données sont organisées en tables à deux dimensions appelées relation
- Une relation est notée R

▶ Attributs/champs: $A=\{A_1, A_2, \dots, A_n\}$

- les nom attribués aux colonnes d'une table
- Dans la base de données **bibliothèque**, la relation **livres** a comme attributs: $A=\{id, auteur, titre, genre\}$
- Noté $R.A_i$

▶ Domaine

- L'ensemble des valeurs d'un attribut A_i noté $D(A_i)$
- $D(id)=int$, $D(auteur)=int$, $D(titre)=string$, $D(genre)=int$

BD: NOTIONS DE BASE

► Schéma d'une relation

- défini par
 - Un ensemble fini d'attributs et de leur domaine
 - $S = \{(A_1, D(A_1)), (A_2, D(A_2)), \dots, (A_n, D(A_n))\}$
 - $\text{Livre} = \{(\text{id}, \text{int}), (\text{auteur}, \text{int}), (\text{titre}, \text{string}), (\text{genre}, \text{int})\}$

- chaque ligne d'une relation R est un **n-uplet** appelé aussi enregistrement.
 - un élément de $D(A_1) \times D(A_2) \times D(A_3) \times \dots \times D(A_n)$

BD: NOTIONS DE BASE

► Clé

- ensemble fini d'attribut qui permet d'identifier de manière unique un n-uplet

► Clé primaire

- Un choix d'une clé
- Exemple: id est une clé primaire pour les relations: Livres, Auteurs, Emprunteurs et Genre
- Clé primaire de la relation emprunts ??

► Clé étrangère

- Un attribut (ou groupe d'attributs) d'une relation R1 qui fait référence à la clé primaire de R2.
- Sert à relier 2 tables entre elles.
- Exemple:
 - id_genre de la relation Livres est une clé étrangère, elle fait référence à id_genre de la relation Genre
 - auteur de la relation Livres est une clé étrangère, elle fait référence à id de la relation Auteurs

BD: SCHÉMA RELATIONNEL

- Livres(id, **auteur**, titre, **id_genre**)
 - Genre(id_genre, genre)
 - Auteurs(id, nom, prenom)
 - Emprunteurs(id, nom, prenom, tel)
 - Emprunts(**qui**, **quoi**, date, rendu)
-
- Les clés primaires sont soulignées par un trait continu
 - Les clés étrangères sont en rouge

BD: NOTIONS DE BASE

► Contrainte d'intégrité

- permet de garantir la cohérence des données lors des mises à jour de la base
- une assertion vérifiée par les données de la base, à tout moment

INTÉGRITÉ DE DOMAINE

- Les valeurs d'une colonne de relation doivent appartenir au domaine correspondant

INTÉGRITÉ DE CLÉ

- La valeur de la clés primaire doit être :
 - ▢ unique
 - ▢ non NULLE

INTÉGRITÉ RÉFÉRENCIELLE

- La valeur de la clé étrangère est soit 'NULLE' soit une valeur de la clé à laquelle elle fait référence.

ALGÈBRE RELATIONNELLE

- Une théorie mathématique proche de la théorie des ensembles qui définit des opérations qui peuvent être effectuées sur des *relations*
- Constituée d'un ensemble d'opérations formelles sur les relations.
- Les principes de l'algèbre relationnelle sont beaucoup utilisés de nos jours par les SGBD pour la gestion des bases de données informatiques

OPÉRATEURS RELATIONNELS : PROJECTION

- ▶ Notation: $\Pi_X(R)$
- ▶ Données : Une relation R d'un schéma S et X une sous partie du schéma
- ▶ Résultat : une Relation R' où on ne considère que les attributs du schéma X
- ▶ Exemple1
 - $\Pi_{\text{auteur,titre}}(\text{Livres})$

auteur	titre
1	Salambô
2	Quatre-vingt-treize
3	L'immortaliste
4	Aphorismes
1	Madame Bovary
...	...

OPÉRATEURS RELATIONNELS : PROJECTION

► Exemple 2

- $\Pi_{\text{id_genre}}(\text{Livres})$

id_genre
1
2
...

- La projection d'une relations $R(S)$ selon les attributs A_i donne la relation restante après conservation des colonnes des attributs A_i et suppression des n-uplets devenus doubles

OPÉRATEURS RELATIONNELS : SÉLECTION

- ▶ Notation: $\sigma_c(R)$
- ▶ Données : Une relation R d'un schéma S et un critère c
- ▶ Résultat : $\{(r(S) / r \in R, c(r))\}$
- ▶ Exemple1
 - : $\sigma_{\text{id_genre}=2}(\text{Livres})$

id	auteur	titre	id_genre
4	4	Aphorismes	2
...

OPÉRATEURS RELATIONNELS : SÉLECTION

▶ Exemple2

◦ : $\sigma_{\text{date}='9\text{-oct-16'}}(\text{Emprunts})$

qui	quoi	date	rendu
3	5	9-oct-16	20-oct-16
1	5	9-oct-16	NULL
...

▶ La sélection d'une relations $R(S)$ selon le critère $c = a$ (avec $a \in D(A)$) „,donne la relation restante après suppression des n-uplets ne satisfaisants pas le critère c

OPÉRATEURS RELATIONNELS : SÉLECTION

- ▶ On peut utiliser de nombreux critères pour les sélections:
 - Opérateurs de comparaison
 - Opérateurs logiques
- Détails: avec le langage SQL.

OPÉRATEURS ENSEMBLISTES: UNION

- ▶ Notation: $R1 \cup R2$
- ▶ Données : une relations $R1$ et une relation $R2$ de même schéma S ,
- ▶ Résultat : $R = \{ (r(S) / r(S) \in R1 \text{ ou } r(S) \in R2) \}$
- ▶ Les doublons sont éliminés
- ▶ Exemple: $voiture1 \cup voiture2$

Voiture 1
modèle
fiesta
fusion
focus

Voiture2
modèle
fiesta
focus
Range Rover
ka

modèle
fiesta
fusion
focus
Range Rover
ka

OPÉRATEURS ENSEMBLISTES:

INTERSECTION

- ▶ Notation: $R1 \cap R2$
- ▶ Données : une relations R1 et une relation R2 de même schéma S,
- ▶ Résultat : $R = \{ (r(S) / r(S) \in R1 \text{ et } r(S) \in R2) \}$
- ▶ Exemple: $\text{voiture1} \cap \text{voiture2}$

Voiture 1
modèle
fiesta
fusion
focus

Voiture2
modèle
fiesta
focus
Range Rover
ka

modèle
fiesta
focus

OPÉRATEURS ENSEMBLISTES:

DIFFÉRENCE

- ▶ Notation : $R1-R2$
- ▶ Données : une relations $R1$ et une relation $R2$ de même schéma S ,
- ▶ Résultat : $R = \{r(S) / r(S) \in R1 \text{ et } r(S) \notin R2\}$
- ▶ Les doublons sont éliminés
- ▶ Exemple: voiture1- voiture2

Voiture 1
modèle
fiesta
fusion
focus

Voiture2
modèle
fiesta
focus
Range Rover
ka

modèle
fusion

OPÉRATEURS RELATIONNELS: PRODUIT CARTÉSIEN

- ▶ Notation: $R1 \times R2$
- ▶ Données : deux relations R1 et R2 de schémas respectifs S1 et S2
- ▶ Résultat : $R3 = R1 \times R2 \{r3 = (r1, r2) / r1 \in R1, r2 \in R2\}$
- ▶ $Card(R1 \times R2) = card(R1) \times card(R2)$
- ▶ Exemple1
 - Voiture \times teinte

Teinte	
ref	Couleur
100	Noir
102	Bleu
103	Rouge

Voiture	
id	modèle
1	fiesta
2	fusion
3	focus

OPÉRATEURS RELATIONNELS: PRODUIT CARTÉSIEN

id	modèle	ref	couleur
1	fiesta	100	Noir
1	fiesta	102	Bleu
1	fiesta	103	Rouge
2	fusion	100	Noir
2	fusion	102	Bleu
2	fusion	103	Rouge
3	focus	100	Noir
3	focus	102	Bleu
3	focus	103	Rouge

OPÉRATEURS RELATIONNELS: RENOMMAGE

- ▶ Notation: $\rho_{A_i \rightarrow B_i}(R)$
- ▶ Données : une relations R schéma S , un attribut $A_i \in S$ et un attribut/ $B_i \in S'$ et $B_i \notin S$
- ▶ Résultat : un relation R avec A_i rebaptisé B_i
- ▶ Le renommage $\rho_{A_i \rightarrow B_i}(R)$ d'un schéma S donne une relation identique à la relation de départ, mis à part le schéma de relation qui a été changé pour intégrer le nouveau nom d'attribut B_i .
- ▶ On peut renommer plusieurs attributs en une seule opération

OPÉRATEURS RELATIONNELS: RENOMMAGE

► Exemple

- $\rho_{\text{titre} \rightarrow \text{titre_livre}}(\text{Livres})$

id	auteur	titre_livre	id_genre
1	1	Salambô	1
2	2	Quatre-vingt-treize	1
3	3	L'immortaliste	1
4	4	Aphorismes	2
5	1	Madame Bovary	1
...

OPÉRATEURS RELATIONNELS : JOINTURE

- ▶ Notation: $R1 \bowtie_c R2$
- ▶ Données : deux relations $R1$ et $R2$ de schémas respectifs $S1$ et $S2$ et un critère c portant sur un attribut de $R1$ et un autre de $R2$
- ▶ Résultat : $R3 = \sigma_c(R1 \times R2)$
- ▶ La jointure $R1 \bowtie_{(A_{1i} = A_{2j})} R2$ des schémas $S1$ et $S2$ selon (A_{1i}, A_{2j}) donne le produit cartésien $R1 \times R2$ suivi de l'élimination des n -uplets ne satisfaisant pas la condition sur A_{1i} et A_{2j} .

OPÉRATEURS RELATIONNEL : JOINTURE

- Exemple: $\text{Livres} \bowtie_{\text{Livres.id_genre}=\text{Genre.id_genre}} \text{Genre} =$
 $\sigma_{\text{Livres.id_genre}=\text{Genre.id_genre}}(\text{Livres} \times \text{Genre})$

Livres			
id	auteur	titre	id_genre
1	1	Salambô	1
2	2	Quatre-vingt-treize	1
3	3	L'immortaliste	1
4	4	Aphorismes	2
5	1	Madame Bovary	1
...

Genre	
id_genre	genre
1	Roman
2	littérature
...	...

OPÉRATEUR RELATIONNEL :

JOINTURE

► Exemple: $\text{Livres} \bowtie_{\text{Livres.id_genre}=\text{Genre.id_genre}} \text{Genre}$

id	auteur	titre	id_genre	id_genre	genre
1	1	Salambô	1	1	Roman
2	2	Quatre-vingt-treize	1	1	Roman
3	3	L'immortaliste	1	1	Roman
4	4	Aphorismes	2	2	littérature
5	1	Madame Bovary	1	1	Roman
...		

OPÉRATEURS RELATIONNEL : AGRÉGATION

- ▶ Notation: $A_1, A_2, \dots, A_n \gamma f_1(B_1) f_2(B_2) \dots f_n(B_n)(R)$
- ▶ Données : une Relation R de schémas $S(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_n)$, et f_1, f_2, \dots, f_n des fonction d'agrégation
- ▶ Résultat : R' obtenue en
 - ▶ Groupant les n -uplets de R dont les attributs A_1, A_2, \dots, A_n sont identiques en groupe G_1, G_2, \dots, G_n
 - ▶ Ajoutant de nouveaux attributs $f_1(B_1) f_2(B_2) \dots f_n(B_n)$ aux enregistrements $\Pi_{A_1, A_2, \dots, A_n}(R)$ dont les valeurs sont $f_i(G_i.B_i)$

OPÉRATEURS RELATIONNELS : AGRÉGATION

- Exemple: `id_genre y compte(titre)(Livres)`

Livres			
id	auteur	titre	Id_genre
1	1	Salambô	1
2	2	Quatre-vingt-treize	1
3	3	L'immortaliste	1
4	4	Aphorismes	2
5	1	Madame Bovary	1
...

Id_genr e	Compte(titre)
1	4
2	1
...	...

OPÉRATEURS RELATIONNE : AGRÉGATION

► Fonctions d'agrégation

- compte: $f(e_1, \dots, e_n) = n$
- somme: $f(e_1, \dots, e_n) = e_1 + \dots + e_n$
- moyenne $f(e_1, \dots, e_n) = (e_1 + \dots + e_n)/n$
- maximum: $f(e_1, \dots, e_n) = \max(e_1, \dots, e_n)$
- minimum: $f(e_1, \dots, e_n) = \min(e_1, \dots, e_n)$

OPÉRATEURS RELATIONNELS : DIVISION

- ▶ Notation: $R1 \div R2$
- ▶ Données : deux relations $R1$ et $R2$ de schémas respectifs $S1$ et $S2$ tels que $S2 \subseteq S1$
- ▶ Résultat :
 - ▶ une relation $R3$ regroupant toutes les parties de la relation $R1$ qui sont associées à toutes les occurrences de la relation $R2$.
 - ▶ la division de $R1$ par $R2$ ($R1 \div R2$) génère une relation qui regroupe tous les n -uplets qui, concaténés à chacun des n -uplets de $R2$, donne toujours un n -uplet de $R1$.
 - ▶ **$R1 \div R2$:**
 - ▶ $T1 = \prod_C(R1)$ avec $C = S1 - S2$
 - ▶ $T2 = \prod_C((R2 \times T1) - R1)$
 - ▶ $R3 = T1 - T2$

OPÉRATEURS RELATIONNELS :DIVISION

Enseignement

Enseignant	Nom_Etudiant
Ktata	Karim alani
Ayed	Sami moussa
Latrech	Haifa haji
Ktata	Sami moussa
Ktata	Amina ali
Latrech	Sami moussa
Ayed	Karim alani

Etudiant

Nom_Etudiant
Sami moussa
Karim alani

Enseignement ÷ Etudiant donne:

Enseignant

Ayed
Ktata

Les enseignants qui enseignent à tous les étudiants de la table Etudiants

LANGUAGE SQL

- ▶ Le langage SQL est le langage de programmation qui permet de communiquer avec une BD.
- ▶ Il permet de tout faire en ligne de commande, sans passer par l'interface graphique.
 - Création et alimentation d'une BD
 - Interrogation d'une BD

LANGAGE SQL : INTERROGATION D'UNE BASE

► Projection

```
SELECT A1,A2,...An FROM R
```

```
SELECT * from R
```

- L'opérateur **distinct** avant un attribut ou plusieurs attributs permet de ne considérer que les résultats distincts.

► Exemple

```
SELECT auteur,titre FROM LIVRES
```

```
SELECT distinct id_genre FROM  
Livres
```

LANGUAGE SQL : INTERROGATION D'UNE BASE

- ▶ Sélection

```
SELECT * FROM R WHERE c
```

- ▶ Exemple

```
SELECT * FROM LIVRES WHERE  
id_genre=2
```

LANGAGE SQL : INTERROGATION D'UNE BASE

► Sélection: opérateur de comparaison

Opérateur	Désignation
<	Inferieur strictement
>	Supérieur strictement
<=	Inférieur ou égal
>=	Supérieur ou égal
!= ou <>	différent
=	égal
IS NULL	Est nul
IS NOT NULL	Est non nul

LANGAGE SQL : INTERROGATION D'UNE BASE

► Sélection: Autres opérateurs

Opérateur	Désignation
OR/AND	OU/ET
DISTINCT	Afficher les résultats distincts
IN /NOT IN	Vérifier l'égalité à l'une des valeurs comprises dans une liste donnée ou le résultat d'une sous-requête.
BETWEEN a AND b	Vérifier un critère entre deux valeur a et b
EXISTS/NOT EXISTS	Tester l'existence ou l'absence des données dans une sous requête

LANGAGE SQL : INTERROGATION D'UNE BASE

► Sélection: Autres opérateurs

Opérateur	Désignation
LIKE	Comparaison partielle(recherche sur un modèle plutôt qu'une valeur exacte)
LIKE '%ab%'	La chaine qui contient la chaine "ab"
LIKE "%a"	La chaine qui se termine par "a"
LIKE "a%"	La chaine qui commence par "a"
LIKE "a%b"	La chaine qui commence par "a" et se termine par b

LANGUAGE SQL : INTERROGATION D'UNE BASE

► Produit cartésien

```
SELECT *  
FROM  
(SELECT A1,...An FROM R1),  
(SELECT B1,...Bn FROM R2)
```

Ou bien:

```
SELECT * FROM R1,R2
```

LANGAGE SQL : INTERROGATION D'UNE BASE

► Exemple

```
SELECT *  
FROM  
(SELECT id,modèle FROM Voiture) ,  
(SELECT ref,couleur FROM Teinte)
```

```
SELECT * FROM Voiture,Teinte
```

LANGUAGE SQL : INTERROGATION D'UNE BASE

- ▶ Renommage: R est une relation de schéma $(A_1, \dots, A_n, C_1, \dots, C_m)$

```
SELECT A1 AS B1, ..., An AS Bn, C1, ..., Cm FROM  
R
```

IPEIT - 2021/2022

- ▶ Exemple

```
SELECT id, auteur, titre AS "titre_livre", id_genre  
FROM Livres
```

LANGAGE SQL : INTERROGATION D'UNE BASE

- ▶ Jointure

```
SELECT * FROM R1 JOIN R2 ON A1i = A2j
```

```
SELECT * FROM R1,R2 WHERE A1i = A2j
```

- ▶ Exemple

```
SELECT * FROM Livres JOIN Genre  
ON Livres.id_genre=Genre. id_genre
```

LANGUAGE SQL : INTERROGATION D'UNE BASE

▸ Agrégation

```
SELECT A1,A2,...An, f1(B1) ,f2(B2),...,fn(Bn) FROM R GROUP BY A
```

Avec $A=(A_1,A_2,...A_n)$

▸ Exemple

```
SELECT id_genre, COUNT(titre) FROM Livres  
GROUP BY id_genre
```

LANGAGE SQL : INTERROGATION D'UNE BASE

► Agrégation : fonctions usuelles

SQL	Désignation
COUNT	comptage
SUM	somme
AVG	moyenne
MIN	minimum
MAX	maximum

LANGAGE SQL : INTERROGATION D'UNE BASE

- ▶ Il est possible de coupler le calcul d'une agrégation à une projection / sélection / renommage.
- ▶ Sélection:
 - ▶ Sélection en amont: une agrégation sur le résultat d'une sélection : **WHERE C GROUP BY A_i**
 - ▶ Sélection en aval: une sélection sur le résultat d'une agrégation: **GROUP BY A_i HAVING C**

LANGUAGE SQL : INTERROGATION D'UNE BASE

► Intersection

```
SELECT * FROM R1 INTERSECT SELECT *  
FROM R2
```

IPFIT - 2021/2022

► Exemple

```
SELECT * FROM Voiture1 INTERSECT SELECT  
*FROM  
Voiture2
```

LANGUAGE SQL : INTERROGATION D'UNE BASE

▶ Union

```
SELECT * FROM R1 UNION SELECT * FROM  
R2
```

IPÉIT - 2021/2022

▶ Exemple

```
SELECT * FROM Voiture1 UNION SELECT * FROM  
Voiture2
```

LANGUAGE SQL : INTERROGATION D'UNE BASE

► Différence

```
SELECT * FROM R1 EXCEPT SELECT *  
FROM R2;
```

IP/EIT - 2021/2022

► Exemple

```
SELECT * FROM Voiture1 EXCEPT SELECT * FROM  
Voiture2
```

LANGUAGE SQL : INTERROGATION D'UNE BASE

- ▶ Organisation d'une requête

```
ORDER BY Ai ASC  
ORDER BY Ai DESC
```

IP/EIT - 2021/2022

- ▶ Exemple

```
SELECT * FROM Voiture1 ORDER BY modèle ASC
```

LANGAGE SQL : INTERROGATION D'UNE BASE

‣ Division

- Il n'existe pas en SQL d'équivalent direct à la division !!

‣ Autre solution ?

- Exemple: Les enseignants qui enseignent à tous les étudiants de la table Etudiants.
- Il suffit de chercher les enseignants pour qui il n'existe pas d'étudiants qu'ils n'enseignent pas (double négation)

SQL:DIVISION

Enseignement

Enseignant	Nom_Etudiant
Ktata	Karim alani
Ayed	Sami moussa
Latrech	Haifa haji
Ktata	Sami moussa
Ktata	Amina ali
Latrech	Sami moussa
Ayed	Karim alani

Etudiant

Nom_Etudiant
Sami moussa
Karim alani

Enseignant

Ayed
Ktata

LANGAGE SQL : INTERROGATION D'UNE BASE

Enseignement (enseignant, nom_etudiant)
Etudiant(nom_etudiant)

```
SELECT distinct enseignant FROM Enseignement as  
EN1 WHERE NOT EXISTS  
( SELECT * FROM Etudiant  
WHERE NOT EXISTS  
( SELECT * from Enseignement as EN2 WHERE  
EN1.enseignant=EN2.enseignant AND  
EN2.nom_etudiant=Etudiant.nom_etudiant))
```


LANGAGE SQL : CRÉATION ET ALIMENTATION D'UNE BD

Contraintes d'intégrité:

- ▶ Clé primaire:
 - PRIMARY KEY(clé primaire)
- ▶ Clé étrangère:
 - FOREIGN KEY(clé étrangère) REFERENCES nom_table (clé primaire)
- ▶ Contrainte de vérification
 - CHECK (critère)
- ▶ Contrainte de non nullité
 - NOT NULL
- ▶ Contrainte d'unicité
 - UNIQUE

LANGAGE SQL : CRÉATION ET ALIMENTATION D'UNE BD

► Création d'une table:

```
CREATE TABLE if not exists nom_table ( attribut1  
type_att1,attribut2 type_att2.....,attributn type_attn,PRIMARY KEY  
clé primaire), FOREIGN KEY (clé étrangère) REFERENCES  
table_cle_primaire (clé primaire),.....)
```

IPEP, 2021/2022

► Type:

- CHAR(n) : chaine de taille fixe de longueur maximale n
- VARCHAR(n):chaine de taille variable de longueur maximale n
- INT ou INTEGER
- FLOAT
- DATE
- TIME
- TIMESTAMP

LANGAGE SQL : CRÉATION ET ALIMENTATION D'UNE BD

- ▶ Destruction d'une table:

```
DROP TABLE nom_table
```

- ▶ Renommage d'une table:

```
ALTER TABLE ancien_nom RENAME TO nouveau_nom
```

LANGAGE SQL : CRÉATION ET ALIMENTATION D'UNE BD

▶ Alimentation d'une table:

- ▶ Insertion d'un n-uplet (enregistrement)

```
INSERT INTO nom_table VALUES (val1,val2.....valn)
```

- ▶ Insertion de quelques attributs.:

```
INSERT INTO nom_table (attribut1,attribut3) VALUES (val1,val3)
```

LANGAGE SQL : CRÉATION ET ALIMENTATION D'UNE BD

- ▶ Suppression d'un n-uplet:

```
DELETE FROM nom_table WHERE attribut1=val1 And  
attribut2=val2
```

- ▶ mettre à jour les valeurs d'un ou plusieurs attributs

```
UPDATE nom_table SET attribut1=nouv_val1 WHERE C
```

C : est un critère qui assure des mises à jour sur des lignes bien particulières

LA PRATIQUE: SQLITE3/PYTHON

- `import sqlite3` *#importe la bibliothèque sqlite de python3*
- `nom_conn=sqlite3.connect("chemin_base.sqlite3")` *#connexion avec la base*
- `nom_cur=nom_conn.cursor()` *#crée un curseur à partir d'une connexion à la BD pour stocker une version temporaire du travail*
- `nom_cur.execute("requête sql de définition/manipulation de données")`
création d'une table, alimentation, mise à jour, suppression....
- `nom_con.commit()` *# validation*

LA PRATIQUE: SQLITE3/PYTHON

- `nom_cur.execute("requête d'interrogation")`
- `nom_cur.fetchone()` #récupération d'une ligne
- `nom_cur.fetchall()` #récupération de toutes les lignes dans une liste
- `for i in nom_cur: #récupération ligne par ligne`
 `print(i)`
- `com_cur.close()` # fermeture du curseur
- `nom_con.close()` # fermeture de la connexion

LA PRATIQUE: SQLITE3/PYTHON

- Insertion de n-uplets à partir d'une liste L remplie de tuples (un tuple représente un n-uplet)

```
for x in L:  
    nom_cur.execute(" INSERT INTO nom_table VALUES  
    (?,?....,?)",x)
```

Ou bien:

```
nom_cur.executemany(" INSERT INTO nom_table VALUES  
    (?,?....,?)",L)
```


LA PRATIQUE: SQLITE3/PYTHON

- Une fonction python qui permet d'exécuter une requête sql

```
def requete_sql (nom_cur, R):  
    nom_cur.execute(R)
```

LA PRATIQUE: SQLITE3/PYTHON

- Une fonction python qui permet d'exécuter une requête d'interrogation R et de retourner le résultat sous forme d'une liste de tuples où chaque tuple représente un n-uplet du résultat de la requête

```
def requete_sql (nom_cur, R):  
    nom_cur.execute(R)  
    resultat=[ x for x in nom_cur ]  
    return resultat
```

- Ou bien

```
def requete_sql (nom_cur, R):  
    nom_cur.execute(R)  
    return nom_cur.fetchall()
```

APPLICATION

- Soit le schéma relationnel de la base de donnée assurant la gestion des chambres d'un hôtel suivant :
 - Chambres(id_chambre, numero, etage, nb_place, vue, tarif)
 - Sejours(id_sejour, date_deb, date_fin, id_chambre, id_client)
 - Clients(id_client, nom, prenom, adresse, cp, ville, tel, email)

APPLICATION

- Le numéro et l'étage de la chambre attribuée au client "jean Martin" dont la date d'arrivée est prévue pour le 10/04/2019

```
select numero,etage from chambres where  
Id_chambre=(select id_chambre from sejours where  
id_client=(select id _client from clients where nom =  
'martin' and prenom='jean' )and date_deb="10/04/2019 ")
```

APPLICATION

- Le nombre de séjours effectué par Jean Martin dans cet hôtel

```
select count (id_client) from sejours where id_client=(
select id_client from clients where nom='martin' and
prenom='jean' )
```

- Ou bien

```
select count(id_client) from (select * from sejours join clients on
sejours.id_client=clients.id_client) where nom='martin' and
prenom='jean'
```

APPLICATION

- Le numéro et l'étage des chambres libres pour une éventuelle réservation d'une chambre pour 4 personnes du 13 /05/2019 au 17/05/2019.

```
select numero,etage from chambres where nb_places=4  
and id_chambre not in (select id_chambre from sejours  
where date_deb BETWEEN "13/05/2019 "and "16/05/2019"  
or date_fin BETWEEN "14/05/2019" and "17/05/2019")
```

- N.B. date_deb et date_fin sont de type DATE