



Université Sidi Mohamed Ben Abdellah
Ecole Nationale des Sciences Appliquées Fès ENSAF
Département Génie Industriel

Bases de Données et Systèmes de Gestion des Bases de Données BD & SGBD

Filières : GI-GMT

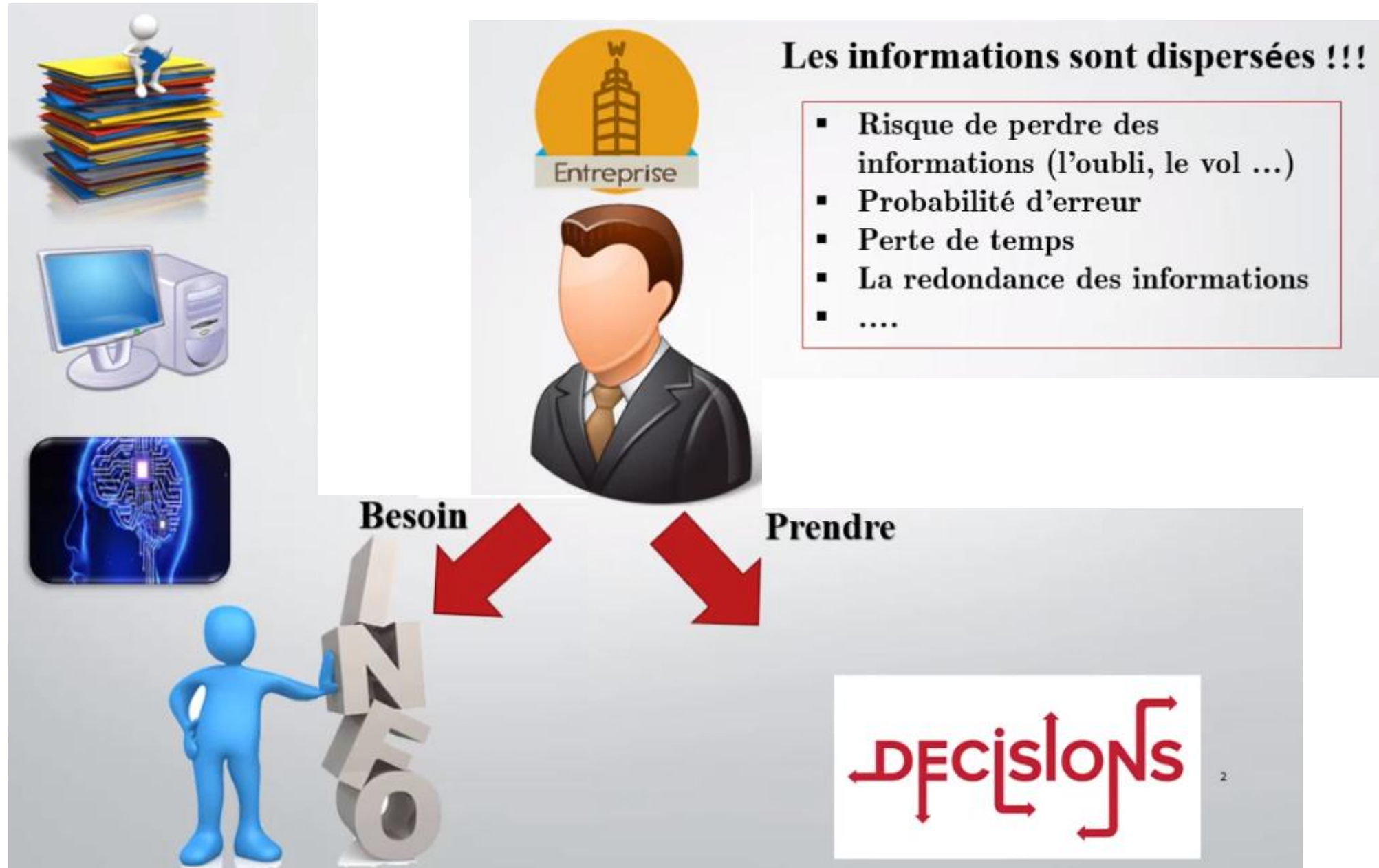
Pr : Mhamed SAYYOURI

□ Système d'information et Base de données

- 2 éléments du module :
 - Analyse et Conception des Système d'Information (32H de cours et TD)
 - Base de données et SGBD (20H de TP)

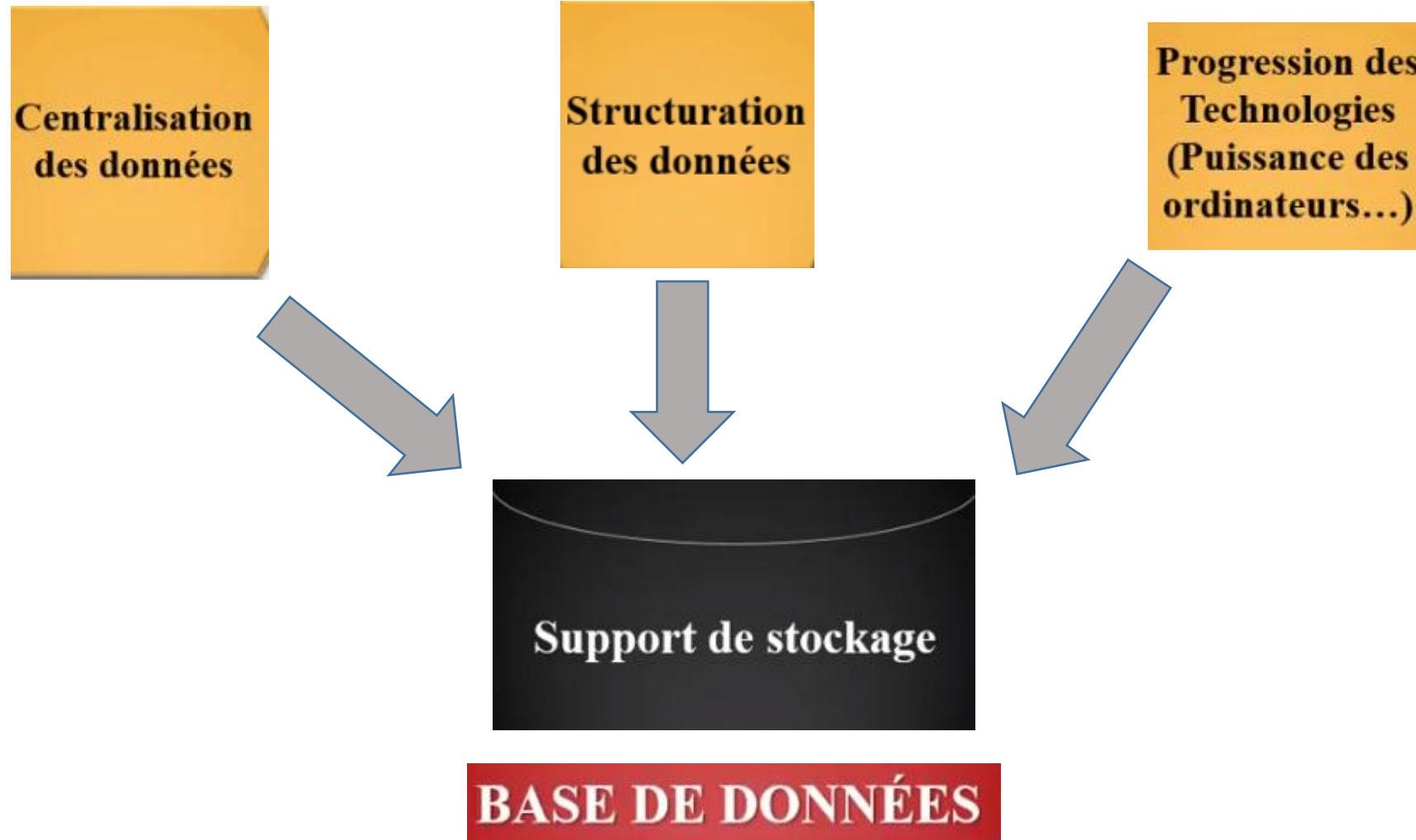
- ❑ Introduction
- ❑ Qu'est ce qu'une base de données?
 - Définition de BD
 - SGBD: Fonctionnalités et Structure
- ❑ Base de données relationnelles (BDR)
 - Le modèle relationnel
 - Le langage SQL

□ Introduction



□ Introduction

LA SOLUTION A ENVISAGER



□ Définition:

- Une Base de données est un ensemble de données **structurées** avec le **minimum de redondance**, mémorisées sur un support **permanent** et qui peut être **partagée** par plusieurs applications et interrogeables de façon **sélective** par plusieurs utilisateurs.
- Une BD est gérée par un **Système de Gestion de Bases de Données (SGBD)**: Assure la **structuration**, le **stockage**, la **consultation** et la **mise à jour** des données.

❖ Exemples:

- Gestion des étudiants, internat, inscription, cours, notes, emploi du temps, ... d'une école (université).
- Gestion des livres, emprunteurs, emprunts d'une Bibliothèque

❑ Caractéristique d'une BD

- Minimum de redondance de données.
 - Partage entre plusieurs applications
 - Contrôle d'intégrité.
 - Indépendances des données et des traitements
 - Sécurité des données et d'accès.
 - Ensemble compact de **tables** contenant les données de la BD. Ces tables sont physiquement des **fichiers**.
- ☛ Tout cela grâce au SGBD

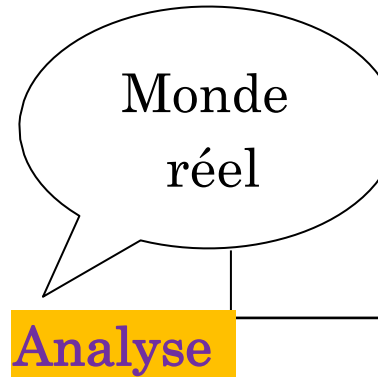
❑ Les modèles des bases de données

On distingue plusieurs modèles de bases de données ; la différence entre ces modèles est la représentation des liens entre les données de la base.



Qu'est ce qu'une base de données?

□ Etapes de conception



Conception

Transformation
logique

Implémentation

Le niveau
externe

Le niveau
conceptuel



Monde réel

Personne
Voiture

Concepteur

Schéma conceptuel

Personne — conduit — Voiture

Table Personne
Table Voiture
Table Conduit
...

Schéma logique

BD

Utilisateurs

LMD

Personne
Dupont
Durant
Rochat
...

Voiture
Ford KA
Peugeot 106
Toyota yaris
...

Le niveau
logique

□ Types d'utilisateurs:

❖ L'administrateur de la BD:

- Créer la BD en précisant la structure ainsi que les détails de stockage sur disque.
- Contrôler l'accès à la BD afin de le permettre aux applications et aux personnes qui y ont droit.
- Conserver une bonne performance d'accès aux données (bonne organisation de stockage, index,)
- Sauvegarder la BD et assurer les reprises après panne.

❖ Le programmeur:

- Développe des applications qui traitent les données de la base.

❖ L'utilisateur final:

- N'a accès qu'aux données qui lui sont utiles via une **application** ou via une **interface du SGBD** afin d'interroger **directement** les tables auxquelles il a droit.

❑ Cycle de Vie d'une BD :

1. Conception de la base

Déterminer les informations qu'il conviendra de mettre dans la base de données ainsi que sa structure.

2. Implantation des données

Transmettre la structure de la BD et les contraintes d'intégrité au SGBD choisi (au moyen du LDD)

3. Utilisation

Interroger et mettre à jour les données (au moyen du LMD)

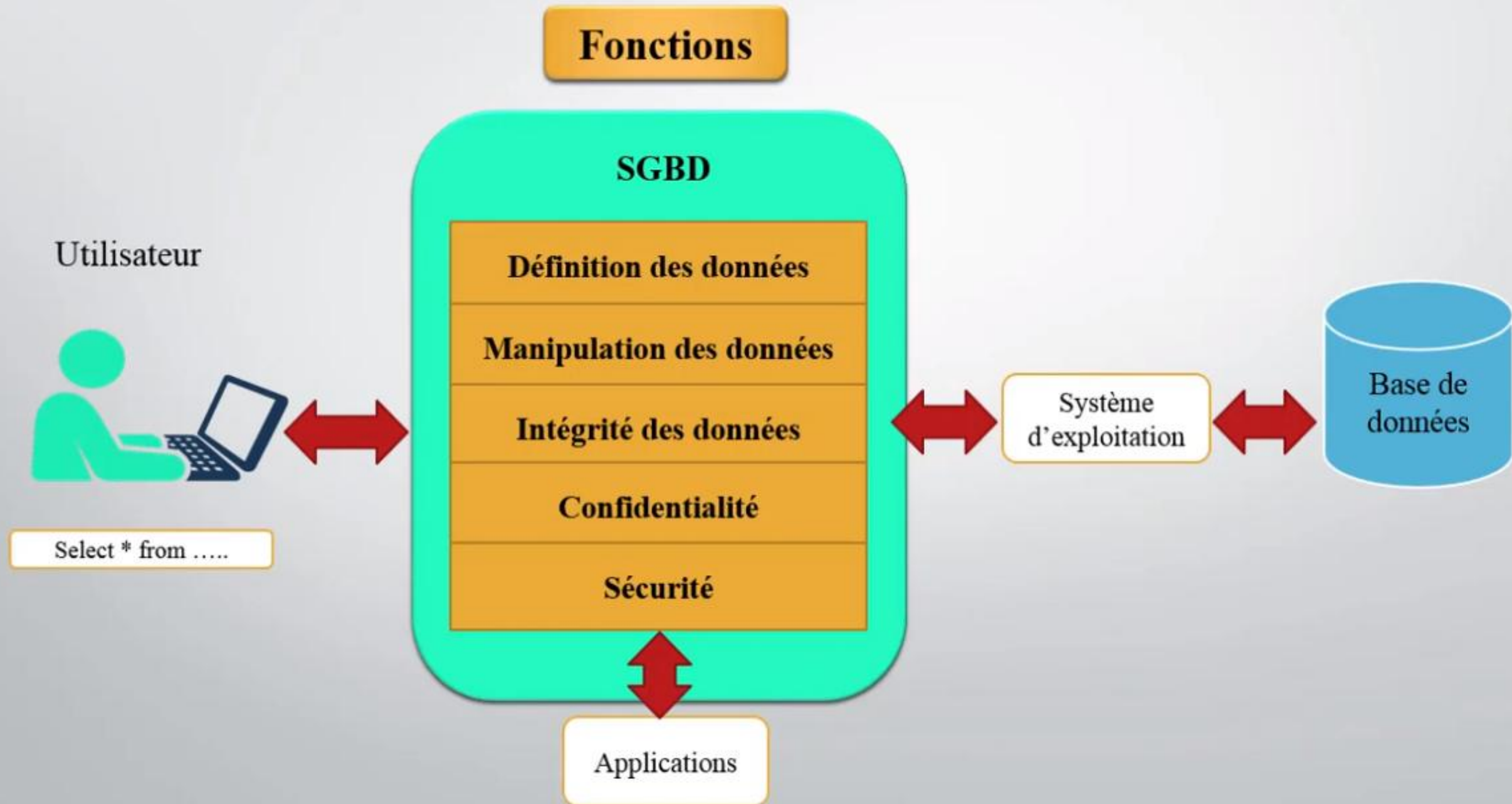
4. Maintenance

Correction, évolution

□ Définition

La gestion de la base de données se fait grâce à **un système** appelé **SGBD** (système de gestion de bases de données) ou en anglais **DBMS** (Database management system). Le SGBD est un ensemble de services (applications logicielles) permettant la manipulation des bases de données.

- Un Système de Gestion de Base de Données peut être défini comme un **ensemble de logiciels prenant en charge la structuration, le stockage, la mise à jour et la maintenance des données**.
- Autrement dit, il permet de **décrire, modifier, interroger et administrer** les données.
- C'est, en fait, **l'interface entre la base de données et les utilisateurs**.



□ La définition des données

Le SGBD offre un **Langage de Description de Données (LDD)**, qui permet de décrire :

- **La liste des relations (Tables).**
- **La liste des attributs de chaque table.**
- **Les liens entre les tables.**
- **Les contraintes** (Une règle obligatoire qui peut s'appliquer sur une table, un attribut ou un lien).

❖ Exemple :

- Liste des tables : Client, Location, Voiture
- Liste des attributs : NomClient, VilleClient
- Contrainte : Un client ne peut pas louer plus qu'une voiture à la fois.

□ La manipulation des données

Le SGBD offre un **Langage de Manipulation de Données (LMD)**, qui permet, en utilisant des requêtes, de réaliser :

- Des recherches de données,
- Des créations de données,
- Des modifications de données,
- Des suppressions de données.

❖ Exemple :

- Insertion/ajout de nouvelles voitures.
- Modification de l'adresse d'un client.
- Suppression d'une voiture.
- Recherche des clients fidèles (qui louent souvent des voitures)

□ L'intégrité des données

L'intégrité permet d'assurer la cohérence et la fiabilité des données en respectant certaines règles appelées contraintes d'intégrité.

❖ Exemple :

- L'attribut note de la BD d'une « Ecole » doit contenir des valeurs comprises entre 0 et 20.
- Un matricule d'une voiture est unique.

□ La confidentialité des données

Le SGBD assure la confidentialité des données par le biais de mots de passe et de privilèges d'accès.

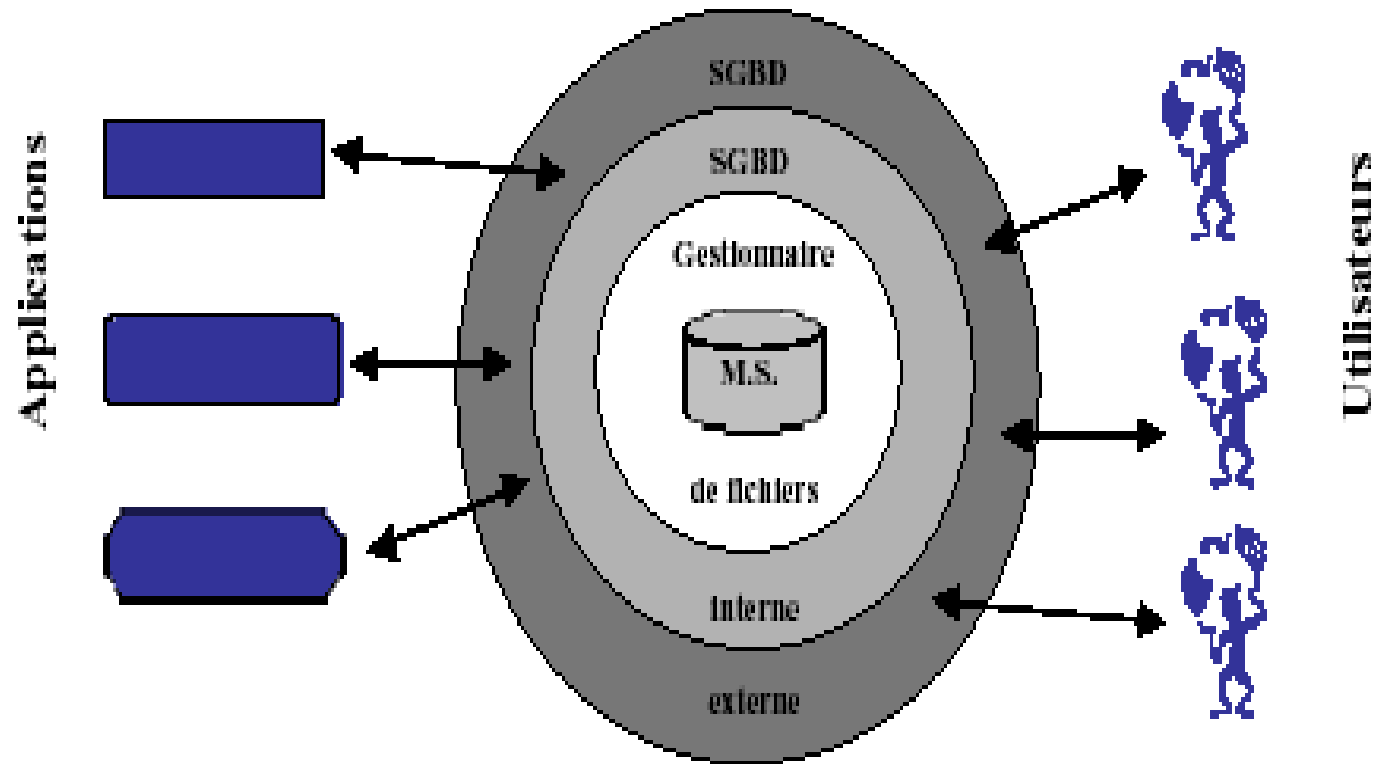
❖ Exemple : seul le directeur peut modifier l'affectation d'un étudiant.

□ La sécurité des données

Le SGBD offre des mécanismes permettant de remettre rapidement la BD dans un état opérationnel en cas d'incident matériel ou logiciel.

❖ Exemple : Sauvegarde de la BD une fois par semaine ; Restauration de la BD en cas de panne.

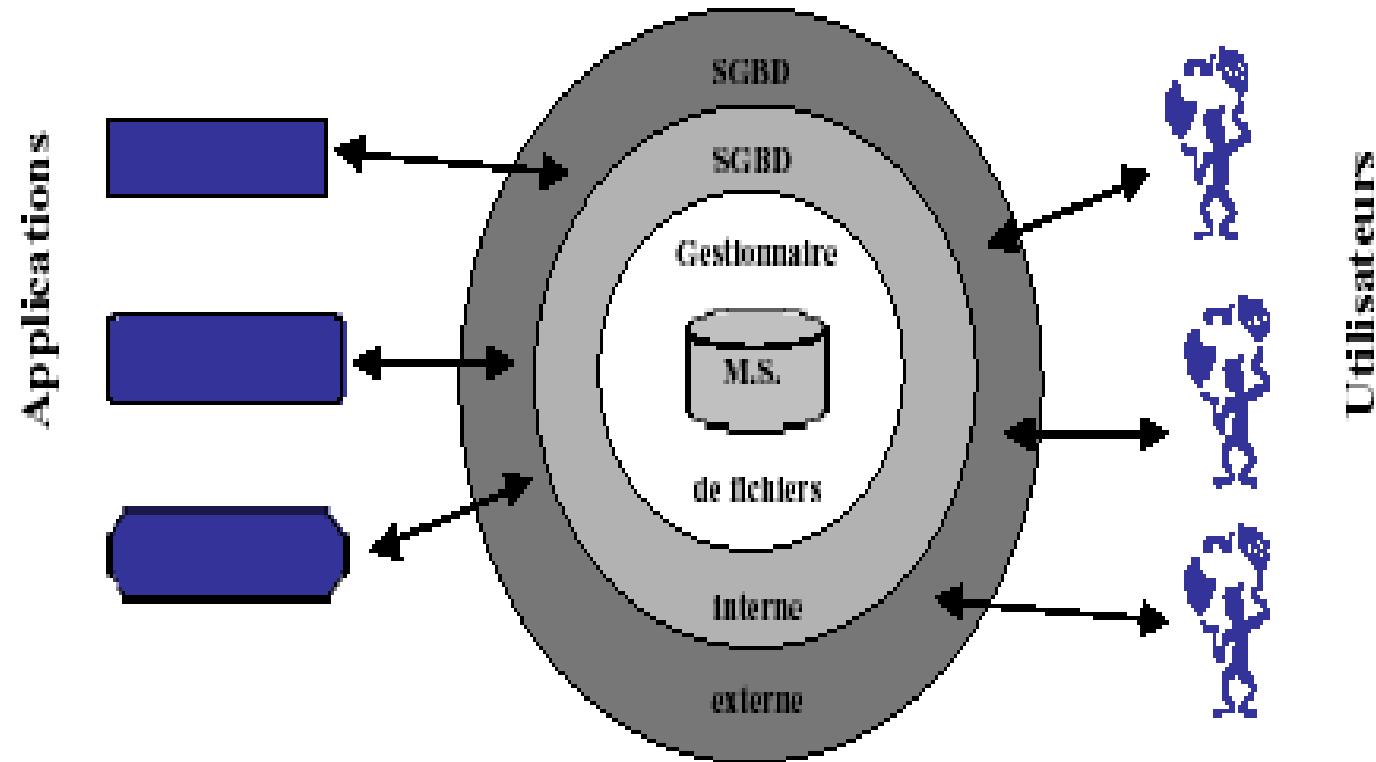
□ Architecture :



➤ Gestionnaire de fichiers

- Stockage des données sur des supports physique.
- Mécanisme de recherche par le contenu.
- Optimisation du stockage.

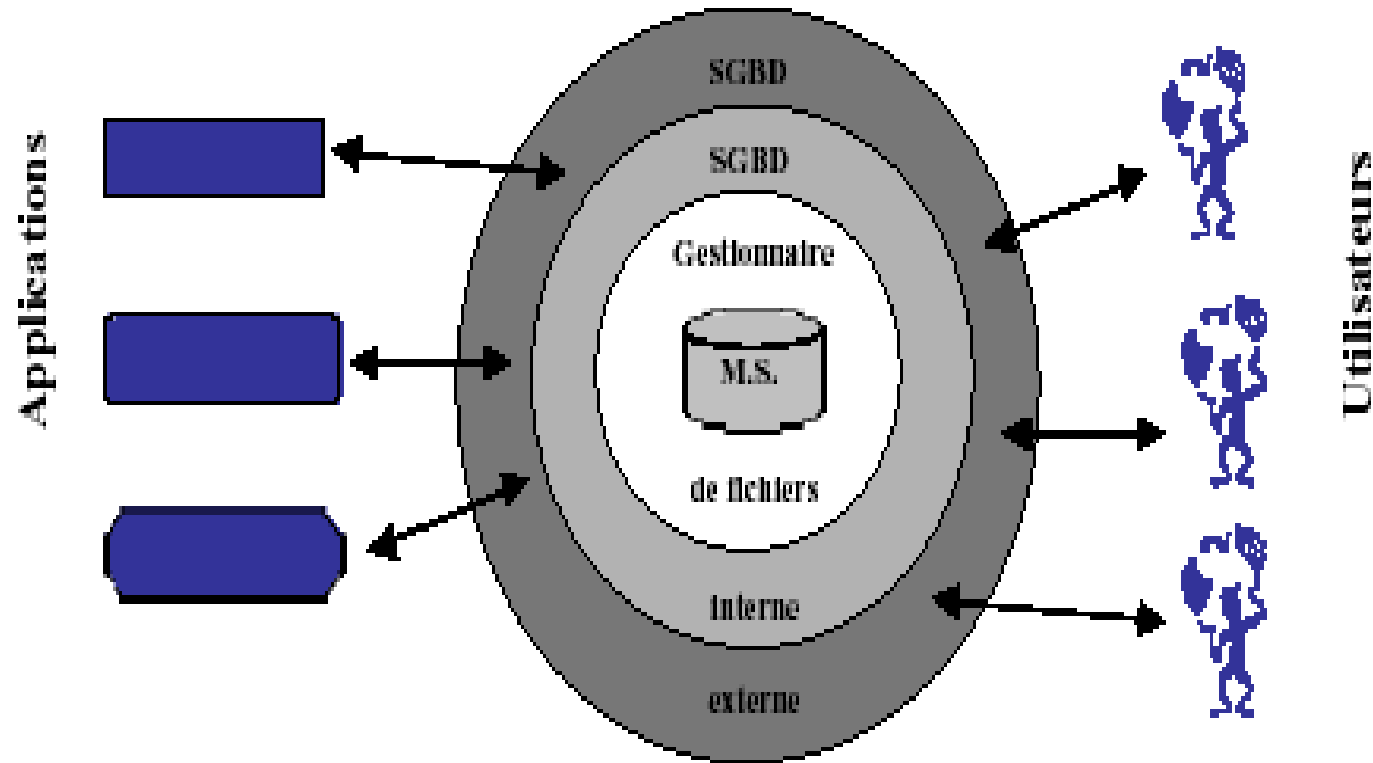
□ Architecture :



➤ SGBD interne (système d'accès au données)

- Assemblage des tables à partir des données dans les fichiers.
- Placement des tables dans les fichiers.
- Gère les liens et l'accès rapide.

□ Architecture :



➤ SGBD externe

- Analyse et interprétation des requêtes des applications et des utilisateurs interactifs.
- Mise en forme des données pour les échanges avec le monde extérieur.

□ Le Système de Gestion de Base de Données Relationnelles (SGBDR)

❖ Définition :

Un SGBDR est un système de gestion des bases de données relationnelles où l'information est organisée dans des tableaux à deux dimensions appelés des relations ou tables.

❖ Exemple



❑ Le schéma relationnel

Le schéma d'une relation (d'une table) est l'ensemble constitué du nom de la relation, suivi de la liste de tous les attributs sur lesquels elle est définie. Par convention, on souligne les clés primaires et on fait précéder les clés étrangères d'un dièse # dans la description des attributs de la table.

❖ Exemple :

Client (CinClient, NomClient, PrénomClient, AdresseClient, VilleClient)

Commande (N°Cmd, #CinClient, dateCmd)

❑ Le schéma relationnel d'une BD

Le schéma d'une base de données est définie par l'ensemble des schémas des relations qui la composent.

❖ Exemple : Base de données - Bibliothèque

Adherent (N°Adh, sexe_adh, nom_adh, prenom_adh, date_naissance, tel_adh)

Auteur (N°Aut, nom_aut, prenom_aut, pays_aut)

Ouvrage (Ref, #N°Auteur, #N°Rayon, titre, date_edition)

Emprunt (#N°Adh, #Ref, date_emp)

❑ Définition

Microsoft Access (officiellement Microsoft Office Access) est un logiciel de gestion de base de données relationnelle éditée par Microsoft.

- Ce logiciel fait partie de la suite Microsoft Office.
- Access est un **SGBDR** permettant de gérer des données en masse.
- Il permet de créer des bases de données plus rapidement et de les gérer plus facilement.
- En outre, Access fournit des outils qui facilitent la communication et la collaboration.



❑ Les objets

- ❖ Les **TABLES** servent à **STOCKER** les informations. Ce sont des fichiers contenant un ensemble d'informations autour d'un même thème ou concept.
- ❖ Les **REQUÊTES** servent à **FILTRE** les données en fonction de critères précis. Elles permettent également de réaliser des **ACTIONS** sur ces données, (calculs, modifications, suppressions...)
- ❖ Les **FORMULAIRES** permettent la **SAISIE** et la **MODIFICATION** d'informations dans les tables de manière plus conviviale que dans les tables : intégrer des cases à cocher, des zones de texte , des images...



❑ Les objets

- ❖ Les **ÉTATS** servent à **IMPRIMER** les données, et permettent de présenter un même fichier de données de façons différentes.
- ❖ Les **MACROS** permettent **d'AUTOMATISER** certaines actions, en programmant des boutons de commande.
- ❖ Les **MODULES** servent à **PROGRAMMER** de manière beaucoup plus pointue que les macros.
- ❖ Le **LANGAGE SQL** est le langage informatique universel qui permet de **MANIPULER** les objets et les données des bases de données.



Les REQUÊTES servent à FILTRER les données en fonction de critères précis. Elles permettent également de réaliser des ACTIONS sur ces données, comme d'effectuer des calculs, des modifications, des suppressions...

REQUÊTE

Les requêtes interrogent les informations contenues dans les champs d'une ou plusieurs tables. Une requête peut être basée sur une seule table, ou sur plusieurs, dans ce cas, il est nécessaire que ces tables soient liées entre elles.

Types de requêtes

Sélection	Action	Analyse croisée
<ul style="list-style-type: none">• Calculée• Paramétrée• Regroupement	<ul style="list-style-type: none">• Ajouter• Modifier• Supprimer• Création Table	<ul style="list-style-type: none">• Présenter les données de manière très synthétique

1. Introduction

Le langage SQL (Structured Query Language) peut être considéré comme le langage d'accès normalisé aux bases de données. Il est supporté par la plupart des produits commerciaux que ce soit par les systèmes de gestion de bases de données micro tel que Access ou par les produits plus professionnels tels que Oracle.

Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma relationnel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution.

De manière synthétique, on peut dire que SQL est un langage relationnel de manipulation de bases de données.

2. Présentation

2.1. Objectifs :

- Créer la structure de la base de données et de ses tables.
- Exécuter les tâches de la gestion des données, telle que l'insertion, la modification et la suppression de données des tables.
- Effectuer des requêtes simples ou complexes.

2.2. Définition

Le SQL (Structured Query Language) est un langage permettant la manipulation et la communication avec une base de données.

L'intérêt de ce langage est qu'il est très puissant et très bien documenté. Il permet, via des requêtes, de créer, interroger, modifier, supprimer et manipuler les données.

2. Présentation

2.3. Fonctionnalités

SQL est un langage composé de sous-ensembles.

- **LDD** : Le Langage de Définition de Données pour créer et supprimer des objets dans la base de données (tables, contraintes d'intégrité, etc...). **Exemple de commandes** : CREATE DROP ALTER
- **LMD** : Le Langage de Manipulation de Données pour la recherche, l'insertion, la mise à jour et la suppression de données. Le LMD est basé sur les opérateurs relationnels, auxquels sont ajoutés des fonctions de calcul d'agrégats et des instructions pour réaliser les opérations d'insertion, mise à jour et suppression. **Exemple de commandes** : INSERT UPDATE DELETE SELECT
- **LCD** : Le Langage de Contrôle de Données pour gérer les droits sur les objets de la base (création des utilisateurs et affectation de leurs droits). **Exemple de commandes** : GRANT REVOKE

3. Le langage de manipulation de données (LMD)

3.1. La projection

La principale commande du langage de manipulation de données est la commande SELECT.

➤ La commande SELECT

L'utilisation la plus courante de SQL consiste à lire des données issues de la base de données. Cela s'effectue grâce à la commande SELECT, qui retourne des enregistrements dans un tableau de résultat. Cette commande peut sélectionner une ou plusieurs colonnes d'une table.

Syntaxe : L'utilisation basique de cette commande s'effectue de la manière suivante.

SELECT nom_Champ1,nom_Champ2,..... **FROM** nom_Table

- Cette requête va sélectionner (SELECT) les champs «nom_du_champ1, nom_du_champ1,... » provenant (FROM) de la table appelée «nom_du_tableau»
- Il est possible de retourner automatiquement toutes les colonnes d'un tableau sans avoir à connaître le nom de toutes les colonnes. Au lieu de lister toutes les colonnes, il faut utiliser le caractère « * ».

SELECT * FROM nom_Table

3. Le langage de manipulation de données (LMD)

➤ La commande **DISTINCT** : Eliminer les redoublons

L'utilisation de la commande SELECT en SQL permet de lire toutes les données d'une ou plusieurs colonnes. Cette commande peut potentiellement afficher des lignes en doubles. Pour éviter des redondances dans les résultats il faut ajouter DISTINCT après le mot SELECT.

Syntaxe : **SELECT DISTINCT** nom_Champ **FROM** nom_Table

Exemple

identifiant	prenom	nom
1	Pierre	Dupond
2	Sabrina	Bernard
3	David	Durand
4	Pierre	Leroy
5	Marie	Leroy

SELECT nom FROM Personne

nom
Dupond
Bernard
Durand
Leroy
Leroy

SELECT DISTINCT nom FROM Personne

nom
Dupond
Bernard
Durand
Leroy

3. Le langage de manipulation de données (LMD)

3.2. La Restriction

Une restriction consiste à sélectionner les lignes satisfaisant à une condition logique effectuée sur leurs attributs. En SQL, les restrictions s'expriment à l'aide de la clause **WHERE** suivie d'une condition logique.

➤ La commande WHERE

La commande WHERE dans une requête SQL permet d'extraire les lignes d'une base de données qui respectent une condition. Cela permet d'obtenir uniquement les informations désirées.

Syntaxe : **SELECT** **nom_Champ** **FROM** **nom_Table** **WHERE** **Condition**

➤ **Opérateurs :** Il existe plusieurs opérateurs à utiliser avec la clause WHERE. La liste ci-jointe présente les opérateurs les plus couramment utilisés.

Opérateurs logiques :

- AND
- OR
- NOT

Comparateur logique :

- IN
- BETWEEN
- LIKE

Comparateurs arithmétiques :

- =
- !=
- >
- <
- >=
- <=
- ::

Opérateur arithmétiques :

- +
- -
- *
- /
- %
- ...

3. Le langage de manipulation de données (LMD)

- Les opérateurs logiques AND et OR peuvent être utilisées au sein de la commande WHERE pour combiner des conditions afin de filtrer les données souhaitées.

Syntaxe: **SELECT** nom_Champ **FROM** nom_Table **WHERE** Condition1 **AND** Condition2

SELECT nom_Champ **FROM** nom_Table **WHERE** Condition1 **OR** Condition2

Ces opérateurs peuvent être combinés :

Syntaxe: **SELECT** nom_Champ **FROM** nom_Table **WHERE** Condition1 **AND** (Condition2 **OR** Condition3)

Exemples :

Pour illustrer les prochaines commandes, nous allons considérer la table « Produit » suivante

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
2	clavier	informatique	32	35
3	souris	informatique	16	30
4	crayon	fourniture	147	2

3. Le langage de manipulation de données (LMD)

```
SELECT * FROM produit
WHERE categorie = 'informatique' AND stock < 20
```

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
3	souris	informatique	16	30

```
SELECT * FROM produit
WHERE nom = 'ordinateur' OR nom = 'clavier'
```

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
2	clavier	informatique	32	35

```
SELECT * FROM produit
WHERE ( categorie = 'informatique' AND prix >= 35 )
OR ( categorie = 'fourniture' AND stock < 200 )
```

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
2	clavier	informatique	32	35
4	crayon	fourniture	147	2

3. Le langage de manipulation de données (LMD)

➤ L'Opérateur IN:

L'opérateur logique IN dans SQL s'utilise avec la commande WHERE pour vérifier si une colonne est égale à une des valeurs comprise dans un ensemble déterminé de valeurs. C'est une méthode simple pour vérifier si une colonne est égale à une valeur OU une autre valeur OU une autre valeur et ainsi de suite, sans avoir à utiliser de multiple fois l'opérateur OR.

Syntaxe **SELECT** nom Champ **FROM** nom Table **WHERE** nom Champ **IN** (valeur1, valeur2, valeur3,)

A savoir : entre les parenthèses il n'y a pas de limite du nombre d'arguments.

Il est possible d'ajouter encore d'autres valeurs. Cette syntaxe peut être associée à l'opérateur NOT pour recherche toutes les lignes qui ne sont pas égales à l'une des valeurs stipulées.

3. Le langage de manipulation de données (LMD)

➤ L'Opérateur BETWEEN:

L'opérateur BETWEEN est utilisé dans une requête SQL pour sélectionner un intervalle de données dans une requête utilisant WHERE. L'intervalle peut être constitué de chaînes de caractères, de nombres ou de dates.

Syntaxe **SELECT * FROM Table WHERE nom colonne BETWEEN 'valeur1' AND 'valeur2'**

La requête suivante retournera toutes les lignes dont la valeur de la colonne « nom_colonne » est comprise entre valeur1 et valeur2.

3. Le langage de manipulation de données (LMD)

➤ L'Opérateur LIKE:

L'opérateur LIKE est utilisé dans la clause WHERE des requêtes SQL. Ce mot-clé permet d'effectuer une recherche sur un modèle particulier. Il est par exemple possible de rechercher les enregistrements dont la valeur d'une colonne commence par telle ou telle lettre.

Syntaxe : `SELECT * FROM table WHERE colonne LIKE modèle`

Un modèle permet de définir un critère de filtrage des données selon sa structure.

Un modèle ressemble généralement à l'un des exemples suivants :

LIKE 'a' : le caractère « % » est un caractère joker qui remplace tous les autres caractères. Ainsi, ce modèle permet de rechercher toutes les chaînes de caractère qui se terminent par un « a ».

LIKE 'a%' : ce modèle permet de rechercher toutes les lignes de « colonne » qui commencent par un « a »

LIKE '%a%' : ce modèle est utilisé pour rechercher tous les enregistrements qui utilisent le caractère « a ».

LIKE 'pa%on' : ce modèle permet de rechercher les chaînes qui commencent par « pa » et qui se terminent par « on », comme « pantalon » ou « pardon ».

3. Le langage de manipulation de données (LMD)

➤ L'Opérateur IS NULL / IS NOT NULL:

Dans le langage SQL, l'opérateur IS permet de filtrer les résultats qui contiennent la valeur NULL. Cet opérateur est indispensable car la valeur NULL est une valeur inconnue et ne peut par conséquent pas être filtrée par les opérateurs de comparaison (égal, inférieur, supérieur ou différent).

Syntaxe **SELECT * FROM 'table' WHERE nom colonne IS NULL**

A l'inverse, pour filtrer les résultats et obtenir uniquement les enregistrements qui ne sont pas nuls

Syntaxe **SELECT * FROM 'table' WHERE nom colonne IS NOT NULL**

3. Le langage de manipulation de données (LMD)

3.3 Le tri des résultats

➤ La commande ORDER BY

La commande ORDER BY permet de trier les lignes d'une requête SQL. Il est possible de trier les données sur une ou plusieurs colonnes, par ordre ascendant ou descendant.

Syntaxe : **SELECT** colonne1, colonne2 **FROM** table **ORDER BY** colonne1

Par défaut les résultats sont classés par ordre ascendant, toutefois il est possible d'inverser l'ordre en utilisant le suffixe DESC après le nom de la colonne. Par ailleurs, il est possible de trier sur plusieurs colonnes en les séparant par une virgule.

Syntaxe : **SELECT** colonne1, colonne2 **FROM** table **ORDER BY** colonne1 DESC, colonne 2 ASC

3. Le langage de manipulation de données (LMD)

3.4 Regroupement des résultats

➤ La commande GROUP BY

Il peut être intéressant de regrouper des résultats afin de faire des opérations par groupe (opérations statistiques par exemple). Cette opération se réalise à l'aide de la clause GROUP BY, suivie du nom de chaque colonne sur laquelle on veut effectuer des regroupements.

Syntaxe : **SELECT** colonne1, fonction(colonne2) **FROM** table **GROUP BY** colonne1

Fonctions :

AVG	Calcule la moyenne d'un set de valeurs
COUNT	Calcule le nombre de lignes
MAX	Récupérer la valeur maximale
MIN	Récupérer la valeur minimale
SUM	Calcule la somme d'un set de valeurs

3. Le langage de manipulation de données (LMD)

3.4 Regroupement des résultats

➤ La commande ORDER BY

Exemple:

id	client	tarif	date_achat
1	Pierre	102	2012-10-23
2	Simon	47	2012-10-27
3	Marie	18	2012-11-05
4	Marie	20	2012-11-14
5	Pierre	160	2012-12-03

```
SELECT client, SUM(tarif)
FROM achat
GROUP BY client
```

client	SUM(tarif)
Pierre	262
Simon	47
Marie	38

4. Le langage de définition de données (LDD)

Le Langage de Définition des Données est la partie de SQL qui permet de décrire les tables et autres objets manipulés par les SGBD.

- La commande CREATE
- La commande ALTER TABLE
- La commande DROP TABLE

4. Le langage de définition de données (LDD)

L'ordre CREATE TABLE permet de créer une table en définissant le nom, le type de chacune des colonnes de la table.

CREATE TABLE *table* (*colonne₁* *type₁*, *colonne₂* *type₂*, ..., *colonne_n*, *type_n*);

Table est le nom que l'on donne à la table. *Colonne_i* est le nom d'une colonne.

Type_i est le type des données contenues dans *Colonne_i*

Exemple :

```
CREATE TABLE STAGE
```

```
    (Num-Stage NUMBER(3) NOT NULL PRIMARY KEY,
```

```
    Libellé-Stage VARCHAR(15) NOT NULL,
```

```
    Nb-jours NUMBER(2),
```

```
    Type-Stage VARCHAR(15),
```

```
    Num-Cat NUMBER);
```

4. Le langage de définition de données (LDD)

Cette commande permet de modifier la définition d'une table.

ALTER TABLE *table*

ADD / MODIFY / DROP (*colonne₁ type₁, colonne₂ type₂, ..., colonne_n, type_n*)

Exemples :

ALTER TABLE STAGIAIRE

ADD NomEntr VARCHAR(20);



Ajout d'un attribut

ALTER TABLE STAGIAIRE

MODIFY NomEntr VARCHAR(30);



Modification dans le type d'un attribut

ALTER TABLE STAGIAIRE

DROP NomEntr;



Suppression d'un attribut

Attention : Certaines modifications peuvent poser des problèmes d'intégrité dans la base !

4. Le langage de définition de données (LDD)

La commande **DROP TABLE** *table* permet de supprimer une table de la base de données. Les lignes de la table et la définition elle-même sont détruites. L'espace occupé par la table est libéré.

Exemple : **DROP TABLE** STAGIAIRE;