Chapitre 1 : Architecture générale d'un SGBD relationnel

1. Définitions :

Définition d'une Base de données :

Une base de données (BD ou BDD) est un ensemble structuré et organisé de grandes quantités de données mémorisées sur des supports accessibles par un ordinateur pour satisfaire simultanément plusieurs utilisateurs.
Les BD constituent le cœur du système d'information.

Définition d'un Système de Gestion de Base de données Relationnel:

Un système de gestion de base de données (SGBD) est un logiciel qui permet la création, la manipulation et le stockage d'une base de données.

Un SGBD Relationnel (SGBDR) est un SGBD dont la base de données est organisée sous forme de tables relationnelles suivant le modèle relationnel.

Exemples de SGBDR :

-SYSTEM-R de IBM : 1er SGBDR (1975) prouvant la faisabilité du modèle relationnel

-ORACLE d'oracle corporation (1ère version années 70). Oracle ne cesse de progresser en incluant les nouvelles technologies : Orienté objet, Internet, Big data, Cloud …

-Microsoft SQL Server (appelé aussi SQL Server ou MSSQL) de Microsoft corporation

-MySQL, 1ère version 1995, de MySQL AB (et Oracle corporation)

-PostgreSQL, outil libre, 1ère version 1996

2. Objectifs d'un SGBDR :

a) Définition des données : Le SGBD doit être muni d'un langage de définition de données (LDD) permettant de définir les données.

b) Manipulation de données : Le SGBD doit être muni d'un langage de manipulation de données (LMD) permettant d'insérer, consulter, modifier, supprimer des données.

c) Evolution de la BD : Le système doit permettre l'évolution de le BD en permettant le changement de son schéma (ajout/suppression de tables, ajout/suppression d'attributs d'une table….).

d) Indépendance logique : le niveau conceptuel doit pouvoir être modifié sans remettre en cause le niveau physique, c'est-à-dire que l'administrateur de la base de données doit pouvoir la faire évoluer la BD sans gêner les utilisateurs (les programmes d'applications)

e) Indépendance physique : le niveau physique peut être modifié indépendamment du niveau conceptuel. Cela signifie que tous les aspects matériels de la base de données

f) Cohérence des données : les données doivent être cohérentes entre elles, de plus lorsque des éléments font référence à d'autres, ces derniers doivent être présents. Ceci est possible grâce à la définition de contraintes d'intégrité.

g) Limitation de la redondance : le SGBD doit pouvoir éviter dans la mesure du possible des informations redondantes afin d'éviter le gaspillage d'espace mémoire ainsi que des erreurs.

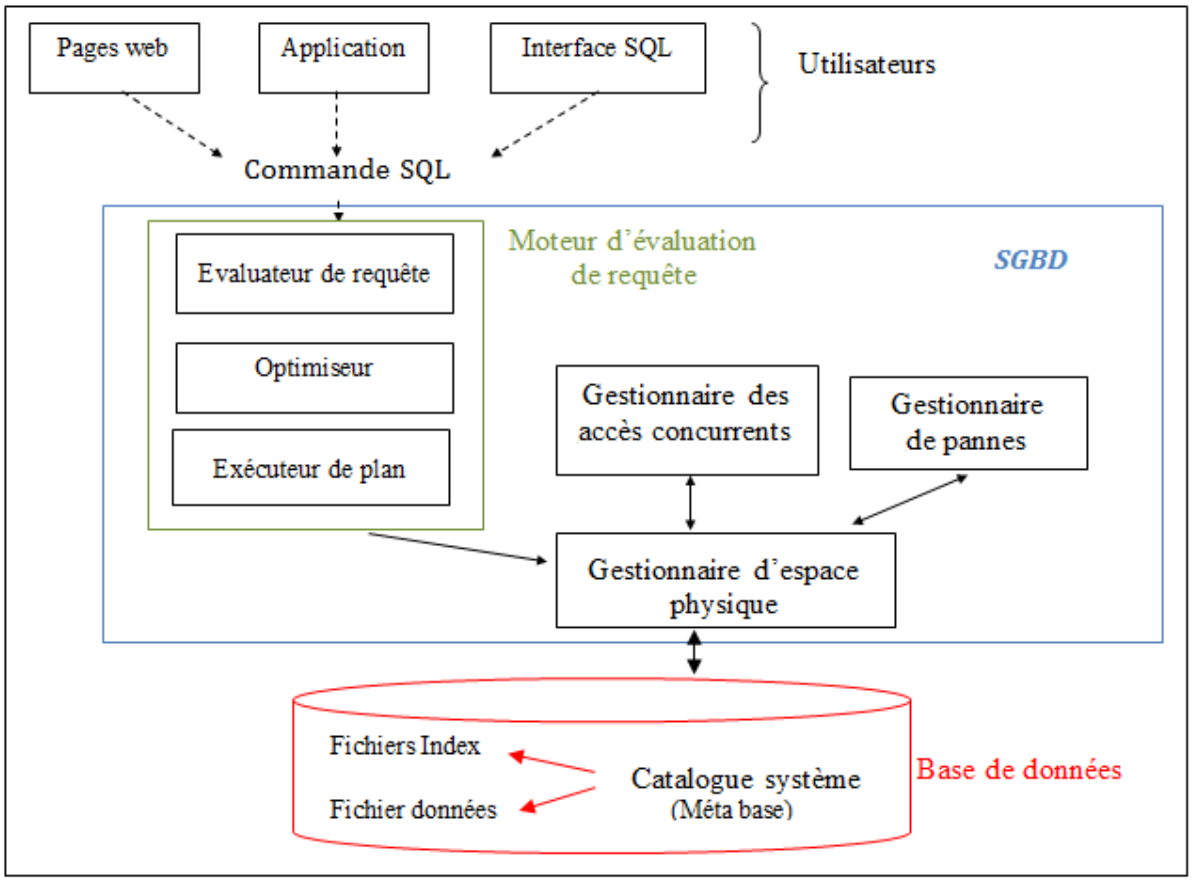
h) Efficacité des accès aux données : le système doit pouvoir fournir les réponses aux requêtes le plus rapidement possible, cela implique des algorithmes de recherche rapides et l'utilisation d'index.

i) Partageabilité : le SGBD doit permettre l'accès simultané à la base de données par plusieurs utilisateurs/ applications. Pour cela, il doit gérer les conflits d'accès (les détecter et les résoudre). Ceci est possible grâce à la notion de transaction et de définition d'algorithmes de gestion de la concurrence.

j) Confidentialité : le SGBD doit présenter des mécanismes permettant de gérer les droits d'accès aux données selon les utilisateurs

k) Sécurité : la BD doit être sécurisée contre les pannes.

3. Architecture fonctionnelle d'un SGBDR :



- **Le moteur d'évaluation de requête :**

Une requête est généralement analysée selon les étapes suivantes :

-Analyse (évaluateur de requête) : La requête est vérifiée syntaxiquement et sémantiquement puis elle est transformée en arbre algébrique où les opérateurs sont les nœuds et les opérandes les feuilles

-Optimisation (optimiseur) : Une séquence SQL peut donner lieu à plusieurs interprétations pouvant conduire à différents temps d'exécution. D'où la nécessité de trouver des techniques qui permettent de trouver le meilleur ordre d'application des opérateurs, les meilleurs chemins d'accès aux relations. Ces techniques qui améliorent les performances du système se retrouvent au niveau de l'optimiseur.

-Génération d'un plan d'exécution (exécuteur de plan) : L'exécuteur de plan fournit le résultat de la requête en exécutant les opérateurs contenus dans l'arborescence à l'aide de procédure associée à ces opérateurs.

- **Le gestionnaire d'espace physique** : permet une organisation efficace des données et un accès rapide grâce à l'implémentation de méthodes d'accès appropriées.

- **Le gestionnaire des accès concurrents et le gestionnaire de panne** concernent la gestion de transactions et la reprise après panne. Ils offrent des mécanismes assurant l'accès à la base de données par plusieurs utilisateurs en parallèle et sa sécurité contre les pannes.

- **Le Catalogue système ou Méta-base**, appelé également dictionnaire de données, contient la description des données de la base (relations, index, vues…), ainsi que des relations systèmes.

4. Fonctions d'un SGBDR :

4.1 Création / modification

a) Création d'une table :

La structure de données primaire dans les systèmes relationnels est la relation. La forme générale de la requête permettant la création d'une relation ou table dans SQL est la suivante :

CREATE TABLE <nom de table> (définition d'attribut [, définition d'attribut]…) ;

Où définition d'attribut est : <nom-attribut type-donnée>

Les principaux types de données prédéfinis (dans Oracle) sont :

-CHAR(n) : chaîne de longueur fixe de n car (par défaut 1 car au maximum 255 car)
-VARCHAR2(n) : chaîne de longueur variable (par défaut 1car au maximum 2000 car)
-DATE : type date format par défaut jj/mm/aa
-Number(x[,y]) : x nombre total de chiffres et y nombre de chiffres après la virgule

Exemple :

Soit la relation : BATEAU (nbat, nombat, sponsor)

Un bateau a un numéro unique (nbat), un nom (nombat) et un sponsor (sponsor).
Nbat : entier de trois chiffres. **Nombat et sponsor :** chaines de 40 caractères maximum

Pour créer la relation BATEAU, on écrit :
CREATE TABLE BATEAU (nbat NUMBER(3), nombat CHAR(40), sponsor CHAR(40)) ;

Remarques :

-Les ordres CREATE TABLE peuvent être émis à tout moment dans la vie de la base.

-Pour la clé primaire, il est recommandé de la définir en ajoutant PRIMARY KEY dès la création de table. Ainsi pour certains SGBDR, les données seront stockées selon l'ordre de cette clé.

Exemple : CREATE TABLE BATEAU (nbat NUMBER(3) **PRIMARY KEY**, nombat CHAR(40), sponsor CHAR(40)) ;

b) Mise à jour des données :

- **Insertion de lignes :**

Après la création des tables, l'utilisateur peut insérer des données :

-Pour insérer un tuple :

INSERT INTO <nom table> [<liste_attributs>] **VALUES** (liste_valeurs);

- Pour insérer plusieurs tuples à partir d'une autre relation

INSERT INTO <nom table> [<liste_attributs>] <Ordre SELECT>;

Dans ce cas, le nombre de colonnes et le type des colonnes dans le INSERT doivent correspondre à ceux du résultat du SELECT

Exemple :

```
INSERT INTO BATEAU (nbat, nombat, sponsor) values (102, 'TASSILI', 'DJEZZY');
INSERT INTO BATEAU (nbat, nombat, sponsor) values (103, 'EL BAHDJA', 'BNA');
INSERT INTO BATEAU  values (104, 'LA COLOMBE', 'NEDJMA');
INSERT INTO BATEAU  values (105, 'HOGGAR', 'BNA');
```

On obtient:

BATEAU

Nbat	Nombat	Sponsor
102	TASSILI	DJEZZY
103	EL BAHDJA	BNA
104	LA COLOMBE	NEDJMA
105	HOGGAR	BNA

Considérons la relation BATEAU_BNA (numb, nomb)

Où numb : entier de trois chiffres. Nomb : chaines de 40 caractères max

-A partir de la table Bateau, insérer dans la table BATEAU_BNA les bateaux sponsorisés par BNA :

```
INSERT INTO BATEAU_BNA (numb, nomb)
    SELECT nbat, nombat FROM BATEAU WHERE sponsor = 'BNA' ;
```

Cette instruction insère dans BATEAU_BNA les 2 tuples (103, 'EL BAHDJA') et (105,'HOGGAR')

- **Modification de valeurs :**

Pour la modification des données d'une relation nous utilisons la commande UPDATE.

UPDATE <nom de table>
SET <nom attribut1> = nouvelle valeur, [nom attribut2 = nouvelle valeur, …]
[WHERE < conditions >] ;

Par exemple, pour modifier le sponsor du bateau n°104, on écrit :

```
UPDATE BATEAU SET SPONSOR = 'OOREDOO' WHERE NBAT = 104;
```

-Pour modifier la valeur d'une colonne d'une relation à partir des données d'une autre relation, nous utilisons la commande UPDATE comme suit :

UPDATE < nom de table 1 >
SET <nom attribut> = (SELECT expression
FROM < nom de table 2 >
WHERE <conditions>)
[WHERE < conditions>] ;

- **Suppression de ligne :**

Pour supprimer des n-uplets d'une relation nous utilisons la commande :

DELETE FROM <nom table> **[WHERE** <prédicat>] ;

L'instruction supprime les tuples qui satisfont le prédicat.

Par exemple, pour supprimer les bateaux sponsorisés par la BNA, on écrit :

DELETE FROM BATEAU WHERE SPONSOR = 'BNA';

c) Modification de schéma :

La modification de schéma d'une BD consiste à :

- ajouter une nouvelle relation (par CREATE TABLE)
- supprimer une relation par la commande : **DROP TABLE <nom table> ;**
- modifier le schéma d'une relation.

Modification de schéma d'une relation:

- Modifier un attribut (colonne) :

ALTER TABLE <nom table> MODIFY (< nom attribut> type de donnée) ;

Exemple : ALTER TABLE BATEAU MODIFY NOMBAT (CHAR (60)) ;

- Ajouter un nouvel attribut (afin de prendre en compte une nouvelle information) :

ALTER TABLE <nom table> ADD (< nom attribut> type de donnée) ;

Exemple : ALTER TABLE BATEAU ADD (ETAT CHAR(30));

Cette requête ajoute une colonne ETAT à la relation BATEAU.

- Supprimer un attribut :

ALTER TABLE <nom table> **DROP COLUMN** < nom attribut> ;

Exemple : ALTER TABLE BATEAU DROP COLUMN ETAT ;

4.2. Chemins d'accès ou INDEX

Un index est une structure de données qui permet d'accélérer les recherches dans une relation (table) en associant à une clé d'index (la liste des attributs indexés) l'emplacement physique du (des) tuple (s) (enregistrements) correspondant(s) sur le disque.

En SQL, la commande de création d'un index est de la forme :

CREATE INDEX <nom Index> **ON** < nom Relation> (attribut [ordre] [, attribut [ordre]..]);

Où ordre est soit **ASC** (ascendant) ou **DESC** (descendant) (ASC par défaut).

Remarque : La plupart des SGBD créent automatiquement un index pour chaque clé (primaire ou candidate) permettant ainsi la vérification de la contrainte d'unicité.

Exemple :

Pour définir des index sur la relation BATEAU :

CREATE INDEX IdxBAT1 ON BATEAU (Nbat) ; (index sur la clé primaire)

CREATE INDEX IdxBAT2 ON BATEAU (Sponsor) ; (index sur un attribut non clé)

L'instruction de suppression d'un index est : **DROP INDEX** <nom index> ;

Remarque : En permettant l'exécution à tout moment des instructions CREATE INDEX et DROP INDEX, le SGBD assure l'indépendance des programmes vis à vis des chemins d'accès (indépendance physique).

4.3. Catalogues (méta base)

Un SGBD relationnel conserve les informations concernant les objets créés par les utilisateurs (tables, vues, index…) dans des catalogues qui sont eux-mêmes des relations stockées dans la base. Le SGBD permet aux utilisateurs de consulter ces catalogues, grâce au langage d'interrogation.

Exemples de catalogues:

- **Relation :** ce catalogue contient des descriptions de toutes les relations (tables) de la base de données. Chaque n-uplet correspond à une relation. On y trouve des informations sur le nom de la relation, son degré, certaines statistiques sur son contenu etc.…
- **Attribut :** pour chaque relation (table) décrite dans le catalogue Relation, le catalogue Attribut comporte autant de lignes qu'il y a d'attributs dans la relation. Chaque n-uplets décrit un attribut : son nom, son type, sa longueur, etc.…
- **Index :** on y trouve la description des chemins d'accès définis sur les relations de la base. Pour chaque index, par exemple, on y trouve la relation, les colonnes d'index etc.…

D'autres catalogues existent relatifs à d'autres aspects (vues, privilèges, contraintes…)

Exemples d'extension des catalogues :

Catalogue RELATION :

Nom Relation	Type Relation	Idf Relation	Taille tuple	Cardinalité	Degré	date de création	…
RELATION	S	1		100			
ATTRIBUT	S	2		1002	5		
…							
BATEAU	B	100	83	60	3		

Chaque tuple décrit une relation qui peut être de 3 types : Base, Système, ou Vue.

Catalogue ATTRIBUT :

Nom Attribut	Idf attribut	Idf Relation	Type attribut	Longueur	…
NOMRELATION	1	1	Char	50	
TYPERELATION	2	1	Char	1	
…					
NBAT	1000	100	Int	2	
NOMBAT	1001	100	Char	40	
SPONSOR	1002	100	Char	40	

Chaque tuple décrit un attribut d'une relation système, de base ou d'une vue

Catalogue INDEX :

Nom index	Idf-rel	Col index	….
…			
IDXBAT1	100	NBAT	
IDXBAT2	100	SPONSOR	
…			

Chaque tuple décrit un index créé sur une relation de base ou système

D'autres fonctions du SGBDR existent telles que :

- assurer la cohérence des informations stockées par rapport à leur signification : contrainte d'intégrité ou intégrité interne,
- Assurer l'accès concurrent tout en gardant la cohérence de la base : synchronisation des accès concurrents,
- Assurer la sûreté de fonctionnement en cas de panne : mécanisme de reprise,
- Assurer la sûreté d'utilisation : droit d'accès

….

Ces dernières seront étudiées dans les chapitres suivants

– Fin chapitre 1 –