Le modèle Relationnel :

Le modèle relationnel (mis de l'avant par Edgar Frank Codd**)** est le plus implémenté et le plus stable des modèles actuellement utilisés. Dans ce modèle la gestion des données est simplifiée grâce aux contraintes d'intégrité. Le système de récupération des données permet à l'utilisateur de visualiser la base de données à travers des structures **de tables relationnelles.**

L'unité de stockage élémentaire est la **« table ».** En général, c'est aux tables que les utilisateurs font référence pour accéder aux données. **Une base de données est constituée de plusieurs tables reliées entre elles.**

Une table est constituée de lignes et de colonnes. Chaque ligne représente un **enregistrement** et chaque colonne représente un **attribut**.

Exemple de table

Voici la table **Livres** qui contient des informations par rapport à des livres de notre bibliothèque.

NumeroLivre	TitreLivre	Auteur	Langue
101	Ainsi parlait Zarathoustra	Friedrich Nietzsche	Allemand
102	La paix des profondeurs	Aldous Huxley	Anglais
103	Les misérables	Victor Hugo	Français
104	La liberté n'est pas une marque de yogourt	Pierre Falardeau	Français

Remarquez

1- Chaque colonne définit un attribut. Un attribut est appelé également champ de la table. Les champs servent à donner la définition d'une table.
2- Chaque ligne définit une instance (ou une occurrence) de la table Livres. Une ligne représente également un enregistrement d'une table.
3- Une table est donc un ensemble d'enregistrements. Pour définir un enregistrement on utilise des attributs.

Définitions :

Termes	Définitions et représentation
Domaine	Ensemble de valeur caractérisé par un nom. Exemple le domaine de noms de jeux vidéo peux être Jeux {Word of Warcraft, Starcraft, Vampire, Civilization,Fallout } Sexe {F, M}
Relation	Sous-ensemble du produit cartésien d'une liste de domaines caractérisé par un nom exemple
Schéma d'une relation	Nom de la relation suivi de la liste des attributs : Etudiant (Nom, Prenom, Code_permanent, adresse)
Attribut	Sous-groupe d'information à l'intérieure d'une relation Code_permanent, Nom Un attribut peut être considéré comme une colonne d'une table ou d'une relation
Valeur d'un attribut	La valeur de l'attribut en rapport avec le domaine de valeurs. Exemple valeur de l'attribut Nom est Martin
Occurrence	Instance d'une entité ou d'une relation. Enregistrement
Identifiant	Attribut permettant d'identifier les occurrences d'une entité ou d'une relation d'une manière unique
Table	Objet d'une base de données relationnelle. Ensemble de lignes et de colonnes. (ce concept sera défini en détail dans le chapitre suivant)
Clé primaire	Identifiant dans une table. Attribut permettant d'identifier chaque occurrence d'une table de marinière unique. Ou encore attribut permettant d'identifier chaque ligne d'une table d'une manière unique.
Clé étrangère	Attribut d'une table (table A) qui est clé primaire d'une table (table B)

Présentation du SGBD Oracle.

Architecture :

Par définition un système de gestion des bases de données est un ensemble de programmes destinés à gérer les fichiers

Oracle est constitué essentiellement des couches suivantes :

1. Le noyau : dont le rôle est de
    a. Optimisation dans l'exécution des requêtes
    b. Gestion des accélérateurs (index et Clusters)
    c. Stockage des données
    d. Gestion de l'intégrité des données
    e. Gestion des connexions à la base de données
    f. Exécution des requêtes
2. Le dictionnaire de données : le dictionnaire de données d'oracle est une métabase qui décrit d'une façon dynamique la base de données. Il permet ainsi de décrire les objets suivants :
    a. Les objets de la base de données (TABLES, SYNONYMES, VUES, COLONNES, …)
    b. Les utilisateurs accédant à la base de données avec leurs privilèges (CONNECT, RESOURCE et DBA).

Ainsi toute opération qui affecte la structure de la base de données provoque automatiquement une mise à jour du dictionnaire.

3. La couche SQL : cette couche joue le rôle d'interface entre le noyau et les différents outils d'oracle. Ainsi tout accès à la base de données est exprimé en langage SQL. Le rôle de cette couche est d'interpréter les commandes SQL , de faire la vérification syntaxique et sémantique et de les soumettre au noyau pour exécution
4. La couche PL/SQL : cette couche est une extension de la couche SQL puis que le PL/SQL est une extension procédurale du SQL.

Objets manipulés par Oracle

Oracle supporte plusieurs types d'objets, en voici quelques-uns :

Les tables : objets contenant les données des utilisateurs ou appartenant au système. Une table est composée de colonnes (Champs ou attributs) et de lignes (enregistrements ou occurrences)

Les vues : Une vue est une table virtuelle issue du modèle externe contenant une partie d'une ou plusieurs tables. Les vues sont utilisées pour ne permettre aux utilisateurs d'utiliser uniquement les données dont ils ont besoin

User : Utilisateurs du système oracle. Chaque usager est identifié par un nom d'usager et un mot de passe.

Les séquences : génératrices de numéro unique

Synonymes : autre nom donné aux objets Table, vue, séquence et schéma.

Les procédures : programme PL/SQL prêt à être exécuté.

Les déclencheurs : procédure déclenchée avant toute modification de la base de données (TRIGGER)

Conseils Généraux :

- ✓ SQL n'est pas sensible à la casse, cependant il est conseillé d'utiliser les mots réservés (commandes, le type de données …) en majuscules
- ✓ Il ne faut pas oublier le point-virgule à la fin de chaque ligne de commande.
- ✓ Utiliser les deux traits -- pour mettre une ligne en commentaire
- ✓ Utiliser /* et */ pour mettre plusieurs lignes en commentaire
- ✓ Utiliser des noms significatifs pour les objets que vous créez
- ✓ Ne pas utiliser de mots réservés comme noms d'objets (tables, vue, colonne.)
- ✓ Mettre une clé primaire pour chacune des tables que vous créez
- ✓ Si vous avez à contrôler l'intégrité référentielle, alors il faudra déterminer l'ordre dans lequel vous allez détruire vos tables

Les commandes SQL

Il existe principalement trois types de commandes SQL

1. Les commandes de définition des données (LDD, langage de définition des données)

   Ces commandes permettent de créer ou de modifier un objet de la base de données : CREATE, ALTER, DROP, MODIFY,

2. Les commandes de manipulation des données (LMD,Langage de Manipulation des Données)

Ces commandes permettent de faire la mise à jour ou la consultation des données dans une table de la base de données.

SELECT, UPDATE, INSERT, DELETE

3. Les commandes de control de données (LCD) :

Exemple : GRANT et REVOKE

La commande SELECT

La commande SELECT est la commande le plus simple à utiliser avec SQL. Cette commande n'affecte en rien la base de données et permet d'extraire des données d'une ou plusieurs tables. La syntaxe simplifiée n'utilise pas de jointure et elle se présente comme suit :

Convention d'écriture

Les <> indique une obligation
Les () pourra être répété plusieurs fois, il faut juste les séparer par une virgule
Les [] indique une option.

```
SELECT <nom_de colonne1,...nom_de_colonnen>
FROM <nom_de table>
[WHERE <condition>]
[ORDER BY <nom_de_colonne>];
```

- ✓ La clause ORDER BY spécifie le tri des données après extraction. Si l'ordre de tri n'est pas précisé alors le tri est par défaut croissant.
- ✓ Pour avoir un tri décroissant il faut ajouter l'option DESC.
- ✓ Le tri peut se faire selon plusieurs colonnes, il faut les séparer par des virgules
- ✓ Dans la commande SELECT, le joker * indique que toutes les colonnes seront sélectionnées.
- ✓ L'option AS permet de changer le nom de colonnes pour l'affichage uniquement.
- ✓ Dans la clause WHERE :

- Une valeur de type caractère (CHAR ou VARCHAR2) doit être mise entre apostrophes. Si la chaine de caractère contient des apostrophes, ceux-ci doivent être doublés.
- Le type numérique (NUMBER) est saisi en notation standard. La virgule décimale est remplacée par un point lors de la saisie
- Le type date doit être mis entre apostrophes et doit respecter le format DATE du SGBD ou être converti avec la fonction TO_DATE

Exemples

SELECT * FROM employes;

SELECT empno, ename, job

FROM EMPLOYES ORDER BY ename, job;

SELECT empno, ename, job

FROM EMPLOYES ORDER BY ename DESC, job;

SELECT * FROM EMPLOYES ORDER BY 1,2;

Le tri se fait selon la colonne 1, puis la colonne 2

Liste des opérateurs utilisés dans la condition (WHERE)

Opérateurs	Signification	Exemple
=	Égalité	SELECT empno, ename, job FROM EMPLOYES WHERE SAL =2500
<> ou != ou ^=	Inégalité ou différent	
>	Plus grand	SELECT empno, ename, job FROM EMPLOYES WHERE SAL >2500 ORDER BY ename ;

<	Plus petit	
>=	Plus grand ou égal	
<=	Plus petit ou égal	
LIKE	Si la valeur est comme une chaine de caractères. Le % est utilisé pour débuter ou compléter la chaine de caractère.	Select * from employes WHERE nom like 'Faf%';
NOT LIKE	Si la valeur n'est pas comme une chaine de caractères. Le % est utilisé pour débuter ou compléter la chaine de caractère.	
IN	Égal à une valeur dans une liste	Select * from employes WHERE nom IN ('Fafar', 'Simpson','Lebeau');
NOT IN	N'est pas égal à une valeur dans une liste	
IS NULL	Si la valeur retournée est NULL	Select * from employes WHERE salaire is NULL;
IS NOT NULL	Si la valeur retournée est n'est pas NULL	
BETWEEN x AND Y	Si la valeur est comprise entre x et y	Select * from employes WHERE salaire BETWEEN 9000 AND 16000 ;
NOT BETWEEN x AND y	Si la valeur n'est pas comprise entre x et y	
ANY	Si au moins une valeur répond à la comparaison	
ALL	Si toutes les valeurs répondent à la comparaison	
EXISTS	Si la colonne existe	
NOT EXISTS	Si la colonne n'existe pas	