

Université Mohamed Boudiaf M'sila



Administration des bases de données

Master 2 IDO

Année 2022/2023

Réalisé par Dr M.Sahraoui

Contenu

1. Le métier d'administration de base de données.

2. Composants de l'architecture d'Oracle.

3. Gestion d'une instance Oracle.

4. Administration physique des BDs: structures physiques de stockage, structures logiques de stockage, tuning logique des BDs.

5. Gestion de la sécurité.

6. Sauvegarde et restauration.

7. Optimisation des requêtes: Sauvegarde et restauration

1. Le métier d'administration de base de données.

- L'Administrateur bases de données conçoit, gère et administre les **systemes de gestion de données de l'entreprise**, en assurant la cohérence, la qualité et la sécurité.
- Autres appellations en français
 - ✓ Gestionnaire de base de données
 - ✓ Administrateur de base de données
 - ✓ Ingénieur datawarehouse

- Les métiers autour des bases de données
 - 1) Administrateur
 - 2) Responsable de la sécurité
 - 3) Administrateur réseaux
 - 4) Développeurs d'application
 - 5) Administrateurs d'application
 - 6) Utilisateurs : modifier les données, créer des rapports
- Dans des environnements de petite taille, l'administrateur peut jouer quasiment tous les rôles.

- Activités principales
 - 1) Concevoir et spécifier les paramètres et l'architecture de la base de données en lien avec l'architecte SI (structure, paramètres, modélisation)
 - 2) Installer le Serveur de données et les applications clientes
 - 3) Administrer la base de données (suivi de la qualité des données, suivi du niveau de service, gestion des accès utilisateurs et de la sécurité)
 - 4) Gérer les évolutions, migrations, sauvegarde et restauration.
 - 5) Mettre à jour les outils d'exploitation
 - 6) Assurer un support technique aux utilisateurs et aux équipes de développement
 - 7) Garantir auprès des utilisateurs internes et externes la confidentialité, l'intégrité et la disponibilité des données

- Système de gestion de base de données (SGBD)



- Système de gestion de base de données (SGBD)

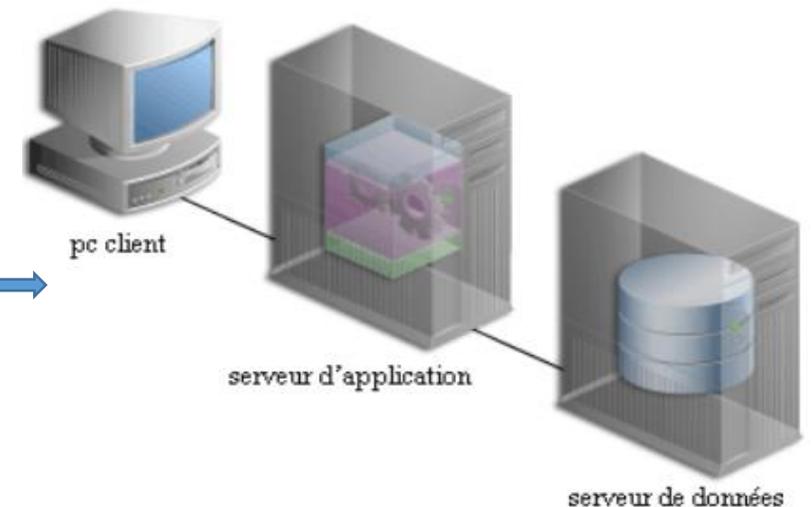
Un SGBD est un ensemble de programmes qui permettent la gestion et l'accès à une base de données.

- Un SGBD possède son propre système de fichier.
- Un SGBD assure la reprise en cas de panne.
- Un SGBD doit permettre la sauvegarde et la restauration d'une BD.
- Un SGBD doit permettre une gestion des rôles et droits.
- Une des fonctions importante des SGBD modernes est d'autoriser les utilisateurs d'effectuer des opérations simultanées (concurrentes) sur des données partagées de la BD.
- Un SGBD peut être mono-utilisateur ou multi-utilisateur.

- Système de gestion de base de données (SGBD)

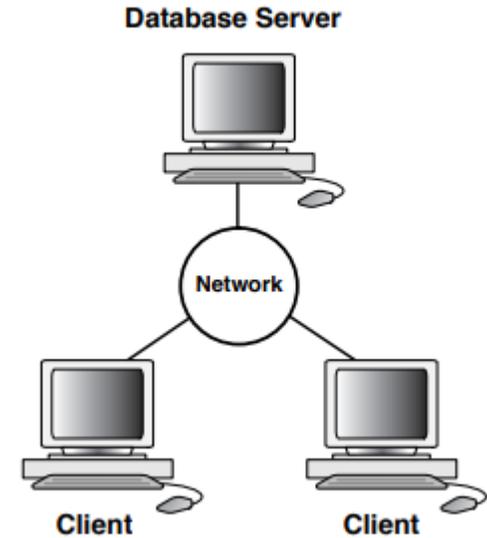
Trois architectures de multi-utilisateurs possibles:

- Architecture locale: Tout est sur le même serveur matériel, programme client et serveur de données
- client/serveur : des applications clientes envoient les requêtes SQL et PL/SQL à un serveur distant, via le réseau.
- Multitier : des serveurs d'application allègent la charge du serveur en réalisant certains accès pour les clients. Ex: Architecture 3-tiers

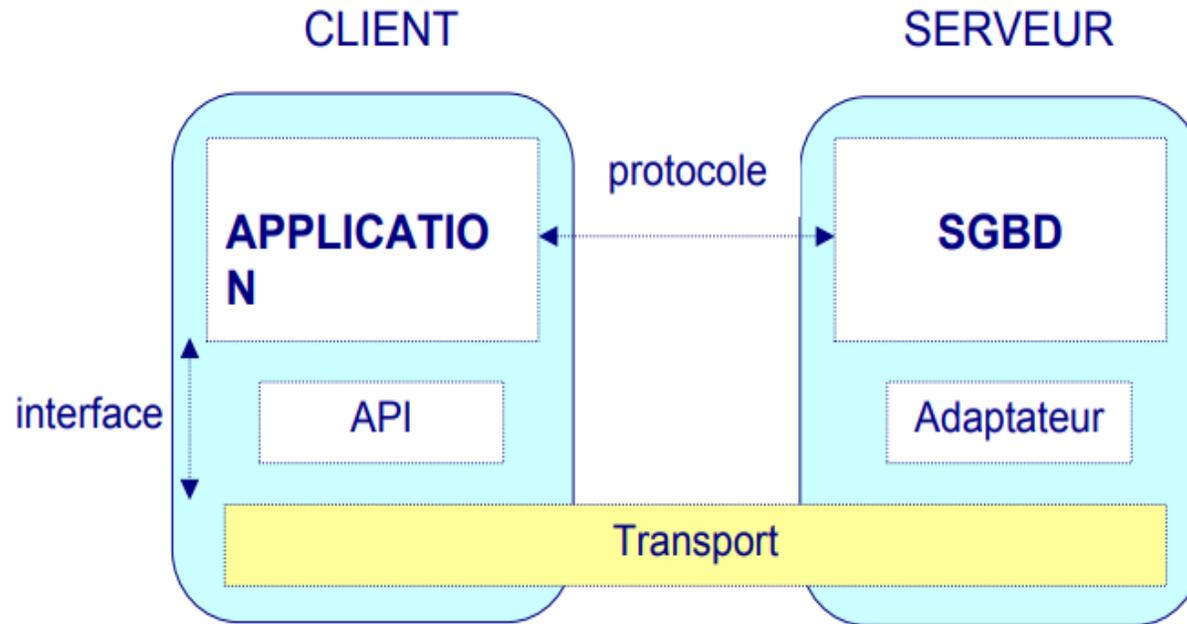


- Architecture Client/Serveur

- L'architecture client/serveur désigne un mode de communication entre plusieurs ordinateurs à double niveau d'hierarchie.
- Le logiciel client peut envoyer des requêtes à un serveur via un protocole de communication à travers un support (réseau).
- Le serveur est initialement passif à l'écoute des requêtes clients sur un port déterminé. Dès qu'une requête lui parvient, il décide de la traiter ou de la mettre en attente et envoie une réponse.



- Architecture Client Serveur

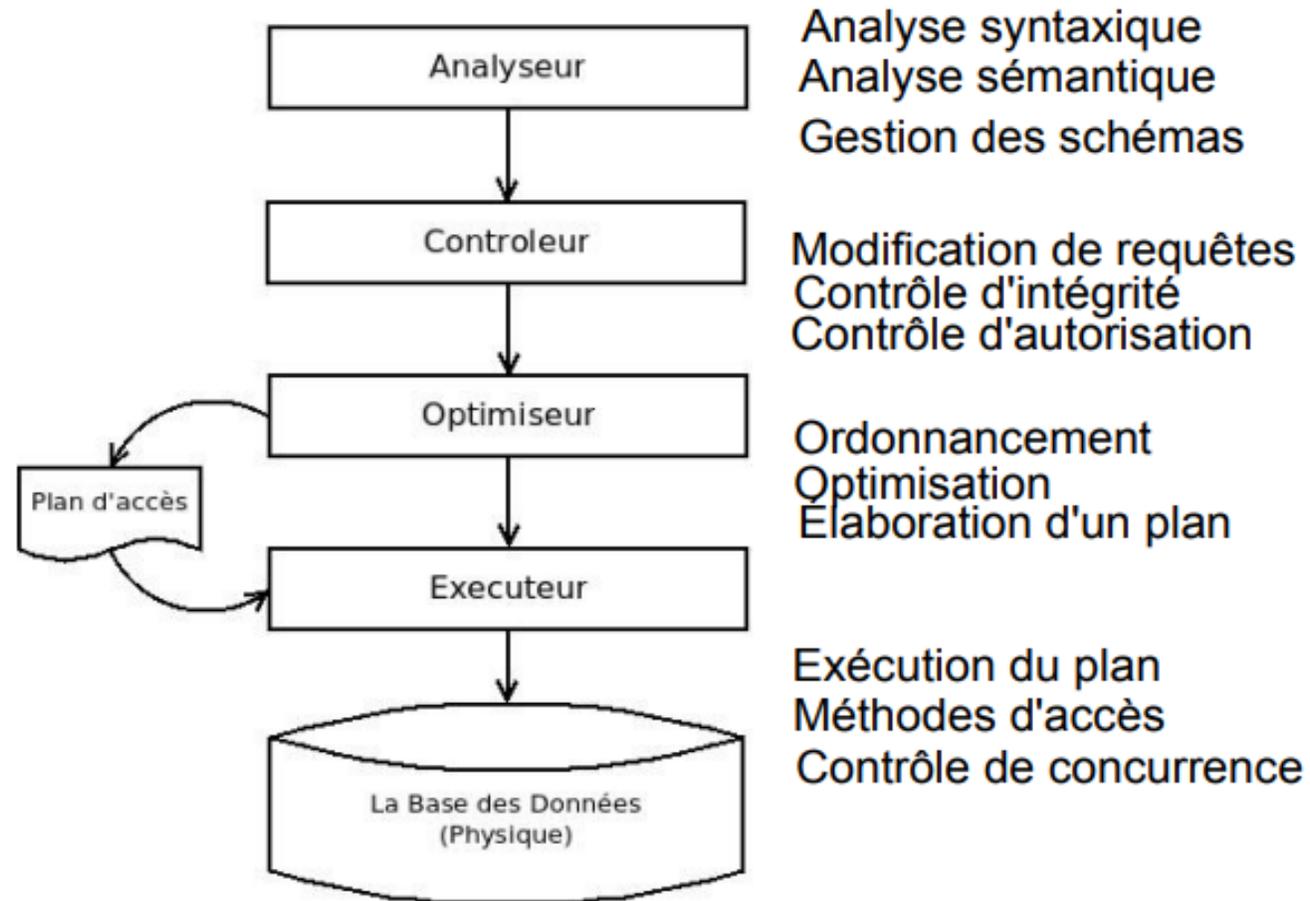


- **API (Application Programming Interface):** Bibliothèque de fonctions qui peut être propriétaire au SGBD ou indépendante, permettant de développer des applications client serveur de telle sorte que les programmes clients envoient leurs requêtes au serveur par des appels à des fonctions contenues dans l'API
Exemple: OCI propriétaire d'Oracle, ODBC permet l'accès à des SGBD différents de Microsoft

- Architecture Client/Serveur

- Un processus utilisateur est créé quand un utilisateur lance une application cliente
- Une connexion va être créée avec l'instance de l'SGBD, l'utilisateur va ouvrir une session
- Un processus serveur va analyser et exécuter les requêtes, retourner les données
 - Mode dédié : un processus serveur pour un processus client
 - Mode partagé : les clients partagent un groupe de processus serveurs

- Architecture générale des SGBDs



- Architecture générale des SGBDs

- Décrire les données

- indépendamment des applications (de manière intrinsèque)

- ⇒ langage de définition des données « DATA DEFINITION LANGUAGE (DDL) ». Ex: create, alter, rename, drop

- Manipuler les données

- Mis à jour les données sans préciser d'algorithme d'accès.

- Interroger et mettre à jour les données, ex: quels sont les noms des produits de prix < 100F ?

- langage de requêtes déclaratif,

- ⇒ langage d'interrogation et de manipulation des données « DATA MANIPULATION LANGUAGE (DML) and DATA INTERROGATION LANGUAGE (DIL) ». Ex: insert, update, delete, select

- Architecture générale des SGBDs

- Contrôler les données

- ❑ intégrité vérification de contraintes d'intégrité

- ex.: le salaire doit être compris entre 400F et 20000F

- ❑ confidentialité contrôle des droits d'accès, autorisation

- ⇒ langage de contrôle des données « DATA CONTROL LANGUAGE (DCL) ». Ex: grant, revoke

- Partage une BD est partagée entre plusieurs utilisateurs en même temps ⇒ contrôle des accès concurrents (Transactions)

- ❑ L'exécution d'une transaction doit préserver la cohérence de la BD

- ⇒ « TRANSACTION CONTROL LANGUAGE (TCL) ». Ex: commit, rollback

- Architecture générale des SGBDs

- Sécurité: reprise après panne, journalisation
- Performances d'accès: index (hashage, arbres balancés ...)
- Indépendance physique: Pouvoir modifier les structures de stockage ou les index sans que cela ait de répercussion au niveau des applications
 - ❑ Les disques, les méthodes d'accès, les modes de placement, le codage des données ne sont pas apparents
- Indépendance logique: Permettre aux différentes applications d'avoir des vues différentes des mêmes données c-à-d permettre au DBA de modifier le schéma logique sans que cela ait de répercussion au niveau des applications.