

Corrigé type - Examen "Architecture et Fonctionnement des SGBD"

Question 1 : Architecture en couches d'un SGBD moderne (4 points)

L'architecture d'un SGBD moderne se décompose généralement en plusieurs couches interactives, chacune ayant un rôle spécifique :

1. **Couche externe (interface utilisateur)** :

Cette couche gère l'interaction avec les utilisateurs. Elle inclut les interfaces graphiques (GUI), les interfaces en ligne de commande (CLI), ou les API pour les applications. Son rôle est de recevoir les requêtes et d'afficher les résultats.

2. **Couche conceptuelle (logique)** :

Elle représente la structure logique de la base de données, indépendamment de la manière dont les données sont stockées physiquement. C'est ici que sont définies les tables, les relations, les vues et les contraintes d'intégrité.

3. **Couche interne (physique)** :

Cette couche gère le stockage réel des données sur le disque. Elle inclut les structures de fichiers, les index, la gestion de l'espace disque et les techniques de compression.

Interactions :

Les couches communiquent entre elles pour répondre aux requêtes. Par exemple, une requête SQL est traitée par la couche externe, analysée par la couche conceptuelle, puis traduite en opérations physiques par la couche interne.

Question 2 : Comparaison des modèles de données (5 points)

1. **Modèle hiérarchique** :

- **Avantages** : Simple, efficace pour les données structurées de manière arborescente (ex : systèmes de fichiers).
- **Inconvénients** : Rigide, difficulté à composer des données complexes.
- **Exemple** : Systèmes de gestion de fichiers XML.

2. **Modèle réseau** :

- **Avantages** : Plus flexible que le modèle hiérarchique, permet des relations complexes.
- **Inconvénients** : Complexe à implémenter et à maintenir.
- **Exemple** : Systèmes de gestion d'inventaire avec relations complexes.

3. **Modèle relationnel** :

- **Avantages** : Normalisé, facile à utiliser avec SQL, adapté à une large gamme d'applications.
- **Inconvénients** : Peut être moins performant pour les données non structurées.
- **Exemple** : Bases de données d'entreprise (MySQL, PostgreSQL).

4. **Modèle orienté objet** :

- **Avantages** : Idéal pour les données complexes et les objets. Supporte l'héritage et

l'encapsulation.

- **Inconvénients** : Moins standardisé, complexité accrue.
- **Exemple** : Bases de données pour les applications géographiques ou de simulation.

Question 3 : Processus de transaction et propriétés ACID (5 points)

Processus de transaction :

Une transaction est une séquence d'opérations exécutées comme une seule unité logique. Elle commence avec `BEGIN` et se termine par `COMMIT` ou `ROLLBACK`.

Propriétés ACID :

1. **Atomicité** : La transaction est entière ou annulée.
2. **Cohérence** : La transaction maintient l'intégrité de la base de données.
3. **Isolation** : Les transactions concurrentes ne s'interfèrent pas.
4. **Durabilité** : Les résultats d'une transaction validée sont permanents.

Exemple :

Une transaction bancaire où un transfert d'argent doit être complet ou annulé pour éviter des incohérences financières.

Question 4 : Optimisation des requêtes (3 points)

Définition :

L'optimisation des requêtes vise à améliorer les performances en minimisant le temps d'exécution et l'utilisation des ressources.

Techniques :

1. **Indexation** : Création d'index sur les colonnes fréquemment interrogées.
2. **Réécriture des requêtes** : Simplification ou réorganisation des requêtes SQL.
3. **Plan d'exécution** : Analyse des plans d'exécution pour identifier les goulots d'étranglement.
4. **Partitionnement** : Division des tables en sous