Question 1 (4 points) :
L'architecture en couches d'un SGBD moderne est une structure sophistiquée qui assure une séparation claire des responsabilités :

1. Niveau externe (vue utilisateur) :
- Interface utilisateur et applications
- Gestion des vues et des droits d'accès
- Adaptation aux besoins spécifiques des utilisateurs
- Abstraction des données pour chaque utilisateur
- Gestion des formulaires et des requêtes

2. Niveau conceptuel (vue logique) :
- Structure globale de la base de données
- Modèle de données
- Relations entre les entités
- Contraintes d'intégrité
- Schéma conceptuel
- Règles métier

3. Niveau interne (vue physique) :
- Stockage physique des données
- Gestion des fichiers et des index
- Optimisation des performances
- Gestion de la mémoire cache
- Stratégies de stockage

Les interactions entre ces couches se font de manière hiérarchique, avec le niveau conceptuel servant d'intermédiaire entre les vues externes et le stockage physique. Cette architecture permet une maintenance facilitée et une évolution indépendante de chaque couche.

Question 2 (5 points) :
Les différents modèles de données présentent des caractéristiques distinctes et des cas d'utilisation spécifiques :

1. Modèle hiérarchique :
- Structure en arbre avec des relations parent-enfant
- Avantages :
  * Simple à comprendre et à implémenter
  * Rapide pour les requêtes simples
  * Efficace pour les données naturellement hiérarchiques
- Inconvénients :
  * Structure rigide
  * Difficulté de modification
  * Redondance des données
- Exemple : Structure d'une entreprise avec départements et employés

2. Modèle réseau :
- Structure en graphe avec des relations complexes
- Avantages :
  * Plus flexible que le hiérarchique
  * Supporte les relations many-to-many
  * Adapté aux données complexes
- Inconvénients :
  * Complexe à maintenir
  * Requêtes complexes
  * Performance variable
- Exemple : Réseau de transport avec stations et connexions

3. Modèle relationnel :
- Structure en tables avec relations
- Avantages :
  * Flexible et normalisé
  * Facile à maintenir
  * Support complet des contraintes
  * Standardisé (SQL)
- Inconvénients :
  * Peut être plus lent pour certaines opérations
  * Complexité des jointures
  * Overhead de normalisation
- Exemple : Système de gestion de bibliothèque

4. Modèle orienté objet :
- Structure en objets avec héritage
- Avantages :
  * Adapté aux données complexes
  * Support de l'héritage
  * Intégration avec OOP
- Inconvénients :
  * Plus complexe à implémenter
  * Moins standardisé
  * Performance variable
- Exemple : Système de gestion de documents

Question 3 (5 points) :
Le processus de transaction dans un SGBD est un mécanisme crucial qui assure l'intégrité des données :

1. Propriétés ACID :
- Atomicité : La transaction est indivisible (tout ou rien)
- Cohérence : La base reste dans un état valide
- Isolation : Les transactions sont indépendantes
- Durabilité : Les modifications sont permanentes

2. Phases d'une transaction :
- Début de transaction
- Exécution des opérations
- Validation ou annulation
- Fin de transaction

3. Gestion des erreurs :
- Rollback en cas d'erreur
- Commit en cas de succès
- Points de sauvegarde

Exemple pratique : Un système bancaire
- Transaction de transfert entre deux comptes
- Vérification des soldes
- Mise à jour des comptes
- Confirmation de la transaction

Question 4 (3 points) :
L'optimisation des requêtes est un processus complexe qui utilise plusieurs techniques avancées :

1. Techniques d'indexation :

- Index B-tree
- Index bitmap
- Index composites
- Index partiels

2. Optimisation du stockage :
- Partitionnement des tables
- Clustering des données
- Compression des données
- Gestion du cache

3. Optimisation des requêtes :
- Analyse des plans d'exécution
- Réécriture des requêtes
- Utilisation de vues matérialisées
- Mise en cache des résultats

Question 5 (3 points) :
La concurrence dans les SGBD est un défi majeur qui nécessite une gestion sophistiquée :

1. Problèmes de concurrence :
- Lectures sales (dirty reads)
- Lectures non répétables
- Écritures perdues
- Incohérences de données

2. Solutions :
- Verrouillage optimiste
- Verrouillage pessimiste
- Gestion des transactions isolées
- Gestion des deadlocks

3. Niveaux d'isolation :
- Read uncommitted
- Read committed
- Repeatable read
- Serializable