

06 - Applications - Exercices

Raphaël P. Barazzutti - 25 février 2024

Exercice 1 : Comprendre l'architecture 3-tiers

1. Décrivez brièvement les rôles des trois tiers (présentation, logique applicative, données) dans une architecture 3-tiers.
2. Citez au moins deux avantages qu'apporte cette séparation en trois couches, tels qu'expliqués dans les slides.

présentation : pour l'utilisateur final logique : logique métier - calcul - règle - mise en forme données : db - stockage - sauvegarde et récupération

scalabilité
modularité
évolution indépendante de chaque couche

Exercice 2 : Bases de données relationnelles (SQL)

stockage de données sous forme de tables - elles sont liées entre elle par des relations (clés primaire et clés étrangères)

1. Définissez le principe fondamental d'une base de données relationnelle.
2. Donnez deux exemples de SGBDR (Systèmes de Gestion de Bases de Données Relationnelles) mentionnés dans les slides.
3. Expliquez ce que signifie "ACID" et pourquoi c'est important pour certaines applications.

MySQL - MARIA Db - Oracle

Ce sont les propriétés qui garantissent la fiabilité des transactions dans des db. Atomicité : garanti que l'entier de la transaction ait eu lieu, de part et d'autre des bases de données, même en cas de panne, de crash ou d'autres erreurs. La cohérence garanti que la db n'a pas d'inscription illégale et que rien ne rende la db corrompue, donc que les données soient correctes en tout temps. L'isolation veille à ce que des transactions qui sont faites simultanément, soient traitées comme si elles avaient été faites les unes après les autres. La durabilité veille à ce qu'une transaction terminée ne soient pas oubliée en cas de panne. Elle doit être écrite dans une mémoire non volatile.

Exercice 3 : Bases de données NoSQL

1. Qu'appelle-t-on NoSQL et en quoi ces bases se distinguent-elles des bases SQL classiques ?
2. Nommez deux types de bases NoSQL (parmi clé-valeur, documents, colonnes larges, graphes, etc.) et donnez un exemple concret de système pour chacun de ces types.
3. Quelles limites peuvent avoir les bases NoSQL par rapport aux bases relationnelles ?

Elles n'utilisent pas les schémas relationnels classiques, mais peuvent utiliser des structures de données variées, comme des documents(json), tableaux, graphe, clés-valeurs

Clé-valeurs : Redis - documents : MongoDB - colonnes larges : Cassandra - graphes : Neo4j

Souvent moins cohérent que les SGBDR, donc pas toujours ACID.

Exercice 4 : Sélection du type de base

1. Dans quel cas privilégieriez-vous une base de données NoSQL plutôt qu'une base relationnelle ? Donnez un exemple d'application qui correspond à ce choix, tel que vu dans les slides.
2. À l'inverse, dans quel contexte rester sur un SGBD relationnel serait-il plus judicieux ?

Applications Web à fortes charges (blogs, réseaux sociaux, e-commerce, là où la rapidité et la scalabilité priment sur la cohérence stricte.

là où l'utilisation nécessite une forte cohérence (e-commerce (partie paiement) - finance, analyses structurées, reportings etc.)

Exercice 5 : Le théorème CAP

1. Énoncez ce que représentent les lettres C, A et P du théorème CAP.
2. Expliquez pourquoi, selon ce théorème, il est impossible de satisfaire simultanément ces trois propriétés dans un système distribué.
3. Donnez un exemple concret (mentionné dans le slide sur le CAP) de base de données qui privilégie la disponibilité et la tolérance au partitionnement (A et P), mais pas la consistance stricte.

Consistance - Disponibilité (Availability) - Tolérance au partitionnement . Le principe est que l'on ne peut pas simultanément garantir