

**MARINE MARGACO.**

## **DEFINITIONS DES MOTS CLES**

Base\_de\_données: Grande quantité de données stockées, organisées et structurées pour être interrogé par un logiciel.

SGBDR: Système de Gestion de Bases de Données Relationnelles.

Une\_table: Ensemble de **données** organisées sous forme d'un tableau où les colonnes correspondent à des catégories d'information (une colonne peut stocker des numéros de téléphone, une autre des noms...) et les lignes à des enregistrements, également appelés entrées.

Une\_colonne: L'objectif d'une base de données orientée colonnes est d'optimiser l'écriture et la lecture de données vers et depuis le disque, afin d'accélérer le renvoi des résultats d'une requête.

Une\_ligne/\_Un\_enregistrement: Une base de données relationnelle est constituée d'un ensemble de tables. Une table est constituée de lignes et de colonnes. Chaque colonne correspond à un champ de données. Chaque ligne correspond à un enregistrement.

Clé\_primaire: Permet d'identifier chaque enregistrement dans une table de base de données. Chaque enregistrement de cette clé primaire doit être UNIQUE et ne doit pas contenir de valeur NULL.

Clé étrangère: Permet de gérer des relations entre plusieurs tables, et garantissent la cohérence des données.

Une\_contrainte:

Un\_index: les index sont des ressources très utiles qui permettent d'accéder plus rapidement aux données. Cette page explique le fonctionnement des index et leurs intérêts pour accroître les performances de lectures des données.

CRUD: Create, Read, Update, Delete (désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données.)

Une jointure: Les jointures en SQL permettent d'associer plusieurs tables dans une même requête. Cela permet d'exploiter la puissance des bases de données relationnelles pour obtenir des résultats qui combinent les données de plusieurs tables de manière efficace.

## QUESTIONS

1. Qu'est ce qu'une transaction, un commit et un rollback ?
2. Citez un exemple concret issu de votre projet chef d'oeuvre dans lequel une transaction pourrait s'appliquer.
3. Qu'est-ce qu'une sous-requête? Proposez une requête contenant une sous-requête
4. Qu'est-ce qu'une vue ?
5. Que sont MyISAM et innnoDB ? Dressez un comparatif de ces deux concepts

## REPONSES :

**1)** Une transaction est un ensemble d'une ou plusieurs requêtes SQL regroupées au sein d'un bloc qui est exécuté sur un jeu de données. Le début d'une transaction est marqué par le mot clef BEGIN et la fin, en fonction de ce qu'on souhaite faire, par le mot clef ROLLBACK ou COMMIT. Tout les personnes ayant utilisé un serveur de base de données ont déjà fait une transaction. Une requête SQL est la plus petite transaction possible pour laquelle les mots clef de début et fin sont rajoutés de manière implicite par le serveur de base de données.

- ROLLBACK : Restaure/Annule toutes les modifications effectuées depuis le début de la transaction.
- COMMIT : Valide toutes les modifications effectuées depuis le début de la transaction.

<https://bioinfo-fr.net/sql-tips-les-transactions>

**2)** Mon projet chef d'oeuvre : Site regroupant des events avec création de compte utilisateur, possibilité de réserver des places. Une transaction qui pourrait s'appliquer sur mon projet : faire une requête pour récupérer les données d'un utilisateur lors de la création de son compte.

**3)** Dans le langage SQL une sous-requête (aussi appelé "requête imbriquée" ou "requête en cascade") consiste à exécuter une requête à

l'intérieur d'une autre requête. Une requête imbriquée est souvent utilisée au sein d'une clause WHERE ou de HAVING pour remplacer une ou plusieurs constante.

<https://sql.sh/cours/sous-requete>

Exemple : dans une table de produits pour un site marchand qui vend des téléphones.

```
SELECT * FROM products WHERE brand = apple.
```

**4)** Une vue dans une base de données est une synthèse d'une requête d'interrogation de la base. On peut la voir comme une table virtuelle, définie par une requête.

Les avantages des vues sont :

- d'éviter de taper une requête très longue : la vue sert à donner un nom à la requête pour l'utiliser souvent,
- de masquer certaines données à certains utilisateurs. En **SQL** les protections d'une vue ne sont pas forcément les mêmes que celles des tables sous-jacentes.

[https://fr.wikipedia.org/wiki/Vue\\_\(base\\_de\\_donn%C3%A9es\)](https://fr.wikipedia.org/wiki/Vue_(base_de_donn%C3%A9es))

**5)** MyISAM et InnoDB sont des moteurs de stockage proposés par MySQL.

MyISAM est un moteur non-transactionnel, il ne gère ni les relations, ni les transactions.

InnoDB est un moteur relationnel, il offre une bonne sécurité en ce qui concerne la validité des données et est donc un moteur fortement conseillé pour stocker des données relationnelles ou nécessitant des transactions classiques.

**MyISAM :**

Les avantages :

- Très rapide pour les requêtes de type SELECT ou INSERT
- Il supporte les index fulltext : permet d'effectuer des recherches sur des mots en se basant sur un index spécifique, accélérant ainsi les recherches
- Il gère les conflits d'accès (ou lock) : permet de verrouiller une

table pour des opérations bien précises

- Très facile à administrer : possibilité de recopier directement les fichiers d'un serveur vers un autre, support des tables compressées...

Les\_inconvénients\_:

- Il ne supporte pas les transactions
- Il ne supporte pas les clés étrangères

**InnoDB:**

Les\_avantages\_:

- Il supporte ACID : permet d'assurer que chaque enregistrement sera complètement réussi, ou complètement échoué, ainsi les risques d'erreurs sont impossibles, même en cas de panne
- Il gère les transactions (instructions sql BEGIN, COMMIT, ROLLBACK...)
- Il supporte les clés étrangères et les intégrités référentielles
- Il possède un système de récupération automatique en cas de crash

Les\_inconvénients\_:

- Il ne permet pas d'index fulltext (sauf dans les versions récentes)
- Son administration est un peu plus complexe (gestion de tablespace, paramètres supplémentaires dans le my.cnf...)
- Le moteur de stockage est parfois plus lent que d'autres et gourmand en ressources mémoires
- InnoDB prend de 1,5 à 2 fois la place qu'occupe une table

MyISAM, ce qui n'est pas négligeable dans certains cas

### COMPARATIF :

- MyISAM : à privilégier lorsqu'il est surtout nécessaire d'effectuer des requêtes pour lire ou insérer des données. Pour des sites web, c'est souvent la solution recommandée.
- InnoDB : à privilégier pour les systèmes qui ne doivent pas comporter d'erreurs et qui nécessitent des clés étrangères. Pour une application gérant des données importantes, tel une application bancaire, cela peut se révéler le moteur de stockage recommandé.

<https://sql.sh/1548-mysql-innodb-myisam>