
Introduction bases de données

Eductive

Concepts Fondamentaux

Qu'est-ce qu'une base de données?

Une base de données est une collection organisée et structurée de données qui sont stockées de manière persistante pour être gérées, récupérées et mises à jour facilement. Les bases de données sont utilisées pour stocker des informations telles que des informations personnelles, des transactions commerciales, des stocks, des médias, etc.

Pourquoi avons-nous besoin de bases de données?

1. **Stockage Organisé** : Les bases de données fournissent un moyen structuré de stocker des données. Cela permet d'éviter le chaos et facilite l'accès aux informations.
2. **Accès Rapide** : Les bases de données permettent d'accéder rapidement aux informations en utilisant des requêtes SQL. Cela est particulièrement important pour les entreprises qui doivent récupérer des données en temps réel.
3. **Gestion des Données** : Les bases de données permettent de gérer efficacement de grandes quantités de données. Les opérations de recherche, de mise à jour et de suppression peuvent être automatisées.
4. **Partage Facilité** : Les bases de données permettent à plusieurs utilisateurs d'accéder et de partager les mêmes données en même temps, tout en maintenant l'intégrité des données.
5. **Conservation des Données** : Les données sont conservées de manière permanente, ce qui est essentiel pour l'archivage et le respect des réglementations.
6. **Sécurité** : Les bases de données offrent des mécanismes de sécurité pour protéger les données sensibles contre les accès non autorisés.

Différence entre Données Structurées et Non Structurées

Données Structurées : Ce sont des données organisées sous forme de tableaux, avec des lignes et des colonnes. Les exemples incluent les bases de données relationnelles où les informations sont stockées dans des tables.

Données Non Structurées : Ce sont des données qui ne suivent pas de structure spécifique. Elles peuvent être sous forme de texte, d'images, de vidéos, etc. Les exemples incluent les fichiers multimédias, les documents texte et les données web non formatées.

Termes Clés

Tables : Les tables sont les structures fondamentales d'une base de données relationnelle. Elles ressemblent à des feuilles de calcul avec des lignes et des colonnes. Chaque table a un nom unique et est utilisée pour stocker un type spécifique de données.

Enregistrements (ou Lignes) : Chaque ligne dans une table est appelée un enregistrement. Un enregistrement contient un ensemble de données liées à une entité spécifique, comme un client, un produit, etc.

Champs (ou Colonnes) : Chaque colonne dans une table est appelée un champ. Les champs représentent les attributs spécifiques d'une entité. Par exemple, dans une table "Clients", les champs pourraient être "Nom", "Adresse", "Numéro de téléphone", etc.

Modèles de Données

Les trois modèles de données

Il existe trois modèles de données clés : le modèle hiérarchique, le modèle en réseau et le modèle relationnel.

- **Modèle Hiérarchique:** Imaginez-le comme une structure d'arborescence où chaque élément a un seul parent, comme les dossiers et sous-dossiers dans un système de fichiers. C'était l'un des premiers modèles utilisés, mais il est devenu complexe à gérer pour des structures de données complexes.
- **Modèle en Réseau:** Ce modèle permet à un élément d'avoir plusieurs parents, créant ainsi des relations complexes. Il a introduit la flexibilité, mais la gestion des relations s'est avérée compliquée.
- **Modèle Relationnel:** C'est le modèle le plus largement utilisé de nos jours. Il organise les données en tables avec des lignes et des colonnes, où chaque ligne représente un enregistrement unique. Les tables sont liées par des clés primaires et des clés étrangères pour établir des relations entre elles.

Avantages du modèle relationnel,

Parmi ces modèles, le modèle relationnel s'est imposé comme le standard de facto pour la gestion des données. Voici pourquoi :

- Structure Simplicité : Les tables, les lignes et les colonnes sont intuitives et faciles à comprendre.
- Flexibilité : Vous pouvez étendre et modifier les données sans remodeler toute la base de données.
- Normalisation : Le modèle encourage la normalisation pour éviter la redondance des données.
- Requêtes Puissantes : Le langage SQL permet des requêtes sophistiquées pour récupérer des données spécifiques.

Le modèle relationnel a joué un rôle majeur dans le succès des bases de données modernes en rendant la gestion des données plus efficace et accessible.

Clés primaires et clés étrangères

Lorsque nous parlons de relations entre les tables dans le modèle relationnel, deux concepts clés émergent : les clés primaires et les clés étrangères.

- **Clé Primaire** : C'est un attribut unique qui identifie de manière univoque chaque enregistrement dans une table. Il garantit que chaque ligne a une identité distincte.
- **Clé Étrangère** : C'est une colonne dans une table qui établit une relation avec la clé primaire d'une autre table. Elle permet de lier les enregistrements de différentes tables, créant ainsi des relations significatives.

Pensez aux clés primaires comme aux **numéros d'identification uniques** pour chaque ligne, et aux clés étrangères comme aux **liens qui tissent ensemble les données d'une manière logique et compréhensible**.

Introduction au Langage SQL

Introduction au Langage SQL

Le langage SQL, ou **Structured Query Language**, est un langage de programmation spécialement conçu pour interagir avec les bases de données relationnelles.

Il sert de pont entre nous, les utilisateurs, et les données stockées dans les bases de données. SQL est essentiellement le langage qui nous permet de communiquer avec les bases de données de manière claire et organisée.

Introduction au Langage SQL

En utilisant des instructions SQL, nous pouvons effectuer une variété d'opérations sur les données, telles que:

- L'insertion
- la mise à jour
- la suppression
- la récupération.

C'est un langage déclaratif, ce qui signifie que nous décrivons ce que nous voulons accomplir, et le système de gestion de base de données (**SGBD**) se charge de la manière de le faire.

Quelques exemples de requêtes SQL

Quelques exemples simples de requêtes SQL SELECT pour récupérer des données d'une table:

La requête SQL la plus couramment utilisée est SELECT, qui permet de récupérer des données spécifiques d'une ou plusieurs tables. Voici quelques exemples de requêtes SELECT simples :

Sélectionner toutes les colonnes de la table "Clients" :

```
SELECT * FROM Clients;
```

Quelques exemples de requêtes SQL

Sélectionner seulement les noms et les adresses de la table "Clients" :

```
SELECT Nom, Adresse FROM Clients;
```

Sélectionner tous les enregistrements de la table "Commandes" où le montant est supérieur à 100 :

```
SELECT * FROM Commandes WHERE Montant > 100;
```

Quelques exemples de requêtes SQL

L'une des caractéristiques les plus attrayantes de SQL est sa **lisibilité** et sa **simplicité**. Sa syntaxe ressemble presque à une phrase en anglais, ce qui facilite la compréhension, même pour les débutants. L'approche déclarative de SQL signifie que vous n'avez pas à vous soucier des détails techniques de comment récupérer les données ; vous spécifiez simplement ce que vous voulez.

Quelques exemples de requêtes SQL

Par exemple, regardez ces deux requêtes :

- Sélectionner tous les utilisateurs ayant un âge supérieur à 25 dans le modèle relationnel

```
SELECT * FROM Utilisateurs WHERE Age > 25;
```

Par rapport à du Code Python: pour récupérer les utilisateurs ayant un âge supérieur à 25 (simplifié):

```
for utilisateur in liste_utilisateurs: if utilisateur.age  
> 25: print(utilisateur)
```

Quelques exemples de requêtes SQL

Tutoriel utile: <https://www.javatpoint.com/mysql-queries>

Et le site officiel de MySQL:

<https://dev.mysql.com/doc/mysql-tutorial-excerpt/8.0/en/examples.html>

Quelques exercices

Exercice 1

Données : Table "Employees":

EmployeeID	FirstName	LastName	Department	Salary
1	John	Smith	HR	50000
2	Jane	Doe	Marketing	60000
3	Michael	Johnson	IT	55000
4	Emily	Brown	Sales	52000

Exercice 1

Questions :

- Créez la table avec ses valeurs
- Sélectionnez tous les employés de la table.
- Sélectionnez les prénoms et noms des employés.
- Sélectionnez les employés du département "IT".
- Sélectionnez les employés dont le salaire est supérieur à 55000.

Exercice 2

Données : Table "Products"

ProductID	ProductName	Category	Price	ProductID
1	Laptop	Electronics	800	1
2	T-shirt	Clothing	20	2
3	Book	Books	15	3
4	Smartphone	Electronics	600	4

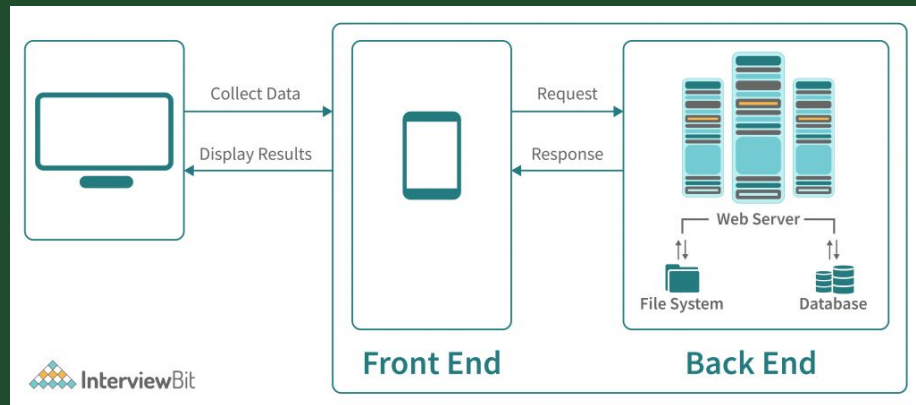
Exercice 2

Questions :

- Créez la table avec ses valeurs
- Sélectionnez les produits dont la catégorie est "Electronics".
- Sélectionnez les noms et prix des produits dont le prix est inférieur à 50.
- Sélectionnez les produits dont le nom commence par "L" et dont le prix est supérieur à 100.
- Sélectionnez les produits dont le nom contient le mot "book" (peu importe la casse).

La BDD et les applis web

Marion Gourlaouen



Pour le 2e cours...

Pour le 2e cours...

- Réfléchissez à des exemples de situations dans lesquelles les bases de données sont utilisées dans la vie quotidienne.
- Explorez quelques requêtes SQL de base pour se familiariser avec la syntaxe et finissez les exercices proposés.
- Exemples d'exercices pour s'entraîner:
 - a. Bons exemples expliqués pour les SELECT:
<https://www.w3resource.com/sql-exercises/sql-retrieve-from-table.php>
 - b. Avec sélecteurs: <https://www.w3resource.com/sql-exercises/sql-boolean-operators.php>
- Installez MySQL et phpmyadmin, et appliquez les exercices directement avec ces outils:
<https://docs.google.com/document/d/1Q9WifKHvXakAV0cZ6gIFBsYN762ovzogylhuGCoK5kR8/edit?usp=sharing>

Liens

Pense bête commandes SQL: <http://coursz.com/les-commandes-sql.html>