

Table des matières

1. **Introduction à SQL et Gestion de Base de Données**
 - Présentation générale du projet
2. **Chapitre 1 : Création d'une base de données et d'un schéma**
 - Définition : Base de données et schéma
 - Code exemple : Création d'une base et d'un schéma
 - Exercice : Créez une base de données et un schéma
3. **Chapitre 2 : Création de tables**
 - Définition : Tables, clés primaires, relations
 - Code exemple : Création des tables `Categorie` et `Produit`
 - Exercice : Création des tables `Client` et `Commande`
4. **Chapitre 3 : Insertion de données**
 - Définition : Utilisation de `INSERT INTO`
 - Code exemple : Ajout de catégories et de produits
 - Exercice : Ajouter des clients et des commandes
5. **Chapitre 4 : Mise à jour et suppression de données**
 - Définition : Requêtes `UPDATE` et `DELETE`
 - Code exemple : Modification d'un produit, suppression d'un produit
 - Exercice : Mise à jour de clients et suppression de catégories
6. **Chapitre 5 : Alias et concaténation**
 - Définition : Alias et concaténation en SQL
 - Code exemple : Utilisation d'alias et concaténation de colonnes
 - Exercice : Affichage de noms avec alias et concaténation
7. **Chapitre 6 : Création de procédures stockées et de fonctions SQL**
 - Définition : Procédures stockées et fonctions
 - Code exemple : Procédure pour ajouter un produit, fonction pour calculer la TVA
 - Exercice : Création d'une procédure pour ajouter un client et fonction de calcul de total avec TVA
8. **Chapitre 7 : Relations et jointures**
 - Définition : Jointures entre tables en SQL
 - Code exemple : Jointure entre `Produit` et `Categorie`
 - Exercice : Jointure entre `Commande` et `Client`
9. **Chapitre 8 : Exercices supplémentaires**
 - Exercice : Création de la base `Inventaire` avec un schéma `stock`
 - Exercice : Ajout de tables et gestion des stocks
 - Exercice : Procédure pour mise à jour des stocks
 - Exercice : Création d'une fonction pour le total des produits
10. **Conclusion**
 - Récapitulatif des concepts appris
 - Conseils pour aller plus loin avec SQL

Chapitre 1 : Création d'une base de données et d'un schéma

Définition

Une **base de données** est un ensemble structuré d'informations, et un **schéma** est un conteneur logique à l'intérieur d'une base où sont stockées les tables, procédures et autres objets SQL.

Code Exemple

```
-- Création de la base de données
CREATE DATABASE GestionCommerce;

-- Utilisation de la base de données
USE GestionCommerce;

-- Création d'un schéma
CREATE SCHEMA ml;
```

Exercice

1. Créez une base de données nommée **Magasin**.
2. Créez un schéma **admin** dans cette base.

Chapitre 2 : Création de tables

Définition

Une **table** est une structure composée de colonnes (champs) et de lignes (enregistrements) servant à stocker des données. Une **clé primaire** identifie chaque ligne de manière unique.

```
-- Création de la table Categorie
CREATE TABLE ml.Categorie (
    CategorieID INT PRIMARY KEY,
    NomCategorie VARCHAR(50) NOT NULL
);

-- Création de la table Produit avec une relation vers Categorie
CREATE TABLE ml.Produit (
    ProduitID INT PRIMARY KEY,
    NomProduit VARCHAR(100),
    Prix DECIMAL(10, 2),
    CategorieID INT,
    FOREIGN KEY (CategorieID) REFERENCES ml.Categorie(CategorieID)
);
```

Exercice

1. Créez une table **Client** avec les champs : ClientID (clé primaire), Nom, Email.
2. Ajoutez une table **Commande** avec une relation vers la table Client.

Chapitre 3 : Insertion de données

Définition

L'instruction **INSERT INTO** permet d'ajouter des enregistrements dans une table.

Code Exemple

```
-- Insertion de données dans la table Categorie
INSERT INTO ml.Categorie (CategorieID, NomCategorie)
VALUES (1, 'Electronique'), (2, 'Vêtements');

-- Insertion de données dans la table Produit
INSERT INTO ml.Produit (ProduitID, NomProduit, Prix, CategorieID)
VALUES (1, 'Téléphone', 599.99, 1), (2, 'T-shirt', 19.99, 2);
```

Exercice

1. Ajoutez 3 clients avec différents emails.
2. Insérez une commande associée à un client.

Chapitre 4 : Mise à jour et suppression de données

Définition

- **UPDATE** permet de modifier des enregistrements existants.
- **DELETE** supprime des enregistrements d'une table.

Code Exemple

```
-- Mise à jour du prix d'un produit
UPDATE ml.Produit
SET Prix = 549.99
WHERE ProduitID = 1;

-- Suppression d'un produit
DELETE FROM ml.Produit
WHERE ProduitID = 2;
```

Exercice

1. Modifiez le nom d'un client.
2. Supprimez une catégorie.

Chapitre 5 : Alias et concaténation

Définition

Un **alias** permet de donner un nom temporaire à une colonne ou table. **Concaténation** combine plusieurs chaînes de caractères en une seule.

Code Exemple

```
-- Utilisation d'un alias
SELECT NomProduit AS 'Produit', Prix AS 'Prix en €'
FROM ml.Produit;

-- Concaténation de colonnes
SELECT CONCAT(NomProduit, ' - ', Prix, '€') AS 'Description'
FROM ml.Produit;
```

Exercice

1. Créez une requête affichant le nom du client avec un alias.
2. Concaténez le nom et l'email des clients.

Chapitre 6 : Création de procédures stockées et de fonctions

Définition

- Une **procédure stockée** est un ensemble de requêtes SQL enregistrées sous un nom.
- Une **fonction** retourne une valeur en fonction d'une logique SQL.

Code Exemple

```
-- Création d'une procédure stockée pour ajouter un produit
CREATE PROCEDURE AjouterProduit (
    IN pNom VARCHAR(100),
    IN pPrix DECIMAL(10, 2),
    IN pCategorieID INT
)
BEGIN
    INSERT INTO ml.Produit (NomProduit, Prix, CategorieID)
    VALUES (pNom, pPrix, pCategorieID);
END;

-- Création d'une fonction pour calculer la TVA
CREATE FUNCTION CalculerTVA(pPrix DECIMAL(10, 2))
RETURNS DECIMAL(10, 2)
BEGIN
    RETURN pPrix * 0.20;
END;
```

Exercice

1. Créez une procédure stockée pour ajouter un client.
2. Créez une fonction retournant le total d'une commande avec TVA.

Chapitre 7 : Relations et jointures

Définition

Les **jointures** permettent de récupérer des données à partir de plusieurs tables liées.

Code Exemple

```
-- Jointure entre Produit et Categorie
SELECT p.NomProduit, c.NomCategorie
FROM ml.Produit p
JOIN ml.Categorie c ON p.CategorieID = c.CategorieID;
```

Exercice

1. Effectuez une jointure entre Commande et Client.
 2. Affichez toutes les commandes avec le nom du client correspondant.
-

Chapitre 8 : Exercices supplémentaires

1. Créez une base de données `Inventaire` avec un schéma `stock`.
2. Ajoutez des tables pour gérer des stocks de produits.
3. Écrivez une procédure stockée permettant de mettre à jour les quantités de stock.
4. Créez une fonction retournant le nombre total de produits disponibles.