





Question 3:

Dictionnaire de données

Table UTILISATEUR

Champ	Type	Taille	Description	Contrainte
user_id	INT	-	Identifiant unique de l'utilisateur	Clé primaire, Auto-incrémenté
nom	STRING (VARCHAR)	50	Nom de l'utilisateur	NOT NULL
prenom	STRING (VARCHAR)	50	Prénom de l'utilisateur	NOT NULL
email	STRING (VARCHAR)	100	Email de l'utilisateur	NOT NULL, UNIQUE
mot_de_pass	STRING (VARCHAR)	255	Mot de passe de l'utilisateur	NOT NULL
telephone	STRING (VARCHAR)	15	Numéro de téléphone	NOT NULL
adresse	STRING (VARCHAR)	255	Adresse de l'utilisateur	NOT NULL

Table COMMANDE

Champ	Type	Taille	Description	Contrainte
commande_id	INT	-	Identifiant unique de la commande	Clé primaire, Auto-incrémenté
user_id	INT	-	Référence vers l'utilisateur	Clé étrangère
date_commande	DATE	-	Date de la commande	NOT NULL
mode_recuperation	STRING (VARCHAR)	20	Mode de récupération de la commande	NOT NULL
status	BOOLEAN	-	État de la commande	NOT NULL
prix_total	DECIMAL	(10,2)	Prix total de la commande	NOT NULL
nombre_article	INT	-	Nombre total d'articles	NOT NULL

Table LIGNE_COMMANDE

Champ	Type	Taille	Description	Contrainte
ligne_commande_id	INT	-	Identifiant unique de la ligne	Clé primaire, Auto-incrémenté
commande_id	INT	-	Référence vers la commande	Clé étrangère
article_id	INT	-	Référence vers l'article	Clé étrangère
quantite	INT	-	Quantité commandée	NOT NULL
nombre_commande	INT	-	Numéro de la commande	NOT NULL

Table ARTICLE

Champ	Type	Taille	Description	Contrainte
article_id	INT	-	Identifiant unique de l'article	Clé primaire, Auto-incrémenté
user_id	INT	-	Référence vers l'utilisateur	Clé étrangère
titre	STRING (VARCHAR)	100	Titre de l'article	NOT NULL
description	STRING (VARCHAR)	255	Description de l'article	NOT NULL
prix	INT	-	Prix de l'article	NOT NULL

| image | STRING (VARCHAR) | 255 | Chemin de l'image | NULL |

Notes supplémentaires :

- Toutes les clés primaires sont auto-incrémentées
- Les champs marqués NOT NULL sont obligatoires
- Les types STRING sont implémentés en VARCHAR avec une taille maximale définie
- Les prix sont stockés en centimes pour éviter les problèmes d'arrondi

Question 4:

```
-- Création de la table UTILISATEUR
CREATE TABLE UTILISATEUR (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    nom VARCHAR(50) NOT NULL,
    prenom VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    mot_de_pass VARCHAR(255) NOT NULL,
    telephone VARCHAR(15) NOT NULL,
    adresse VARCHAR(255) NOT NULL
);

-- Création de la table ARTICLE
CREATE TABLE ARTICLE (
    article_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    titre VARCHAR(100) NOT NULL,
    description VARCHAR(255) NOT NULL,
    prix INT NOT NULL,
    image VARCHAR(255),
    FOREIGN KEY (user_id) REFERENCES UTILISATEUR(user_id)
);

-- Création de la table COMMANDE
CREATE TABLE COMMANDE (
    commande_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    date_commande DATE NOT NULL,
    mode_recuperation VARCHAR(20) NOT NULL,
    status BOOLEAN NOT NULL,
    prix_total DECIMAL(10,2) NOT NULL,
    nombre_article INT NOT NULL,
    FOREIGN KEY (user_id) REFERENCES UTILISATEUR(user_id)
);

-- Création de la table LIGNE_COMMANDE
CREATE TABLE LIGNE_COMMANDE (
    ligne_commande_id INT PRIMARY KEY AUTO_INCREMENT,
    commande_id INT NOT NULL,
    article_id INT NOT NULL,
    quantite INT NOT NULL,
    nombre_commande INT NOT NULL,
```

```
FOREIGN KEY (commande_id) REFERENCES COMMANDE(commande_id),  
FOREIGN KEY (article_id) REFERENCES ARTICLE(article_id)  
);  
  
-- Ajout des index pour optimiser les performances  
CREATE INDEX idx_user_email ON UTILISATEUR(email);  
CREATE INDEX idx_commande_user ON COMMANDE(user_id);  
CREATE INDEX idx_ligne_commande ON LIGNE_COMMANDE(commande_id, article_id);
```

Test sql

-- 1. D'abord, assurons-nous que la base de données est vide

```
DELETE FROM LIGNE_COMMANDE;  
DELETE FROM COMMANDE;  
DELETE FROM ARTICLE;  
DELETE FROM UTILISATEUR;
```

-- 2. Insérer des utilisateurs de test

```
INSERT INTO UTILISATEUR (nom, prenom, email, mot_de_pass, telephone, adresse) VALUES  
( 'Dupont', 'Jean', 'jean.dupont@email.com', 'motdepasse123', '0123456789', '123 rue de Paris'),  
( 'Martin', 'Marie', 'marie.martin@email.com', 'password456', '0987654321', '456 avenue des  
Fleurs');
```

-- 3. Insérer des articles

```
INSERT INTO ARTICLE (user_id, titre, description, prix, image) VALUES  
(1, 'Gâteau au chocolat', 'Délicieux gâteau au chocolat noir', 2500, 'gateau-chocolat.jpg'),  
(1, 'Tarte aux pommes', 'Tarte aux pommes traditionnelle', 2000, 'tarte-pommes.jpg'),  
(2, 'Éclair au café', 'Éclair avec crème au café', 300, 'eclair-cafe.jpg');
```

-- 4. Insérer des commandes

```
INSERT INTO COMMANDE (user_id, date_commande, mode_recuperation, status, prix_total,  
nombre_article) VALUES  
(2, '2024-03-15', 'livraison', true, 5000, 2),  
(1, '2024-03-16', 'sur_place', false, 2300, 3);
```

-- 5. Insérer des lignes de commande

```
INSERT INTO LIGNE_COMMANDE (commande_id, article_id, quantite, nombre_commande)  
VALUES  
(1, 1, 2, 1),  
(2, 2, 1, 1),  
(2, 3, 2, 2);
```

-- 6. Requêtes de test pour vérifier que tout fonctionne

-- Test 1: Afficher tous les articles avec leur créateur

```
SELECT a.titre, a.prix, u.nom, u.prenom  
FROM ARTICLE a  
JOIN UTILISATEUR u ON a.user_id = u.user_id;
```

-- Test 2: Afficher le détail d'une commande avec les articles

```
SELECT
```

```
c.commande_id,  
u.nom AS client,  
a.titre AS article,  
lc.quantite,  
a.prix * lc.quantite AS sous_total  
FROM COMMANDE c  
JOIN UTILISATEUR u ON c.user_id = u.user_id  
JOIN LIGNE_COMMANDE lc ON c.commande_id = lc.commande_id  
JOIN ARTICLE a ON lc.article_id = a.article_id  
WHERE c.commande_id = 1;
```

-- Test 3: Calculer le total des ventes par utilisateur

```
SELECT  
    u.nom,  
    u.prenom,  
    COUNT(DISTINCT c.commande_id) as nombre_commandes,  
    SUM(c.prix_total) as total_ventes  
FROM UTILISATEUR u  
LEFT JOIN COMMANDE c ON u.user_id = c.user_id  
GROUP BY u.user_id;
```

-- Test 4: Vérifier les articles les plus vendus

```
SELECT  
    a.titre,  
    SUM(lc.quantite) as quantite_totale  
FROM ARTICLE a  
LEFT JOIN LIGNE_COMMANDE lc ON a.article_id = lc.article_id  
GROUP BY a.article_id  
ORDER BY quantite_totale DESC;
```