## Objectifs

- Modéliser et manipuler une base de données adaptée à un site e-commerce.  
- Appliquer des requêtes SQL pour insérer, lire, modifier des données.  
- Travailler avec des concepts avancés comme les contraintes, les jointures, et les transactions.

## Scénario

Vous êtes chargé de concevoir et de gérer une base de données pour un site e-commerce vendant des produits variés.   
La base de données doit gérer les utilisateurs, les produits, les commandes, et les détails des commandes.

## Étapes et défis

### 1. Modélisation de la base de données (20 min)

1. Utilisez MySQL Workbench.  
2. Créez un modèle avec les tables suivantes :  
 - `users` : `id`, `name`, `email`, `password`, `created\_at`.  
 - `products` : `id`, `name`, `price`, `stock\_quantity`, `category`.  
 - `orders` : `id`, `user\_id`, `order\_date`, `total\_price`, `status`.  
 - `order\_details` : `id`, `order\_id`, `product\_id`, `quantity`, `subtotal`.  
3. Ajoutez des relations :  
 - `users` et `orders` (1:N)  
 - `orders` et `order\_details` (1:N)  
 - `products` et `order\_details` (1:N)  
4. Exportez le script SQL pour créer les tables.

### 2. Création et gestion de la base de données (10 min)

1. Créez la base de données appelée `ecommerce`.  
2. Importez le script généré pour créer les tables.

### 3. Insertion de données (20 min)

1. Insérez des données dans les tables. Exemple :

- Table `users` :  
 - (1, "Alice", "alice@example.com", "password123", "2024-01-01")  
 - (2, "Bob", "bob@example.com", "securepass", "2024-02-01")  
 - Table `products` :  
 - (1, "Laptop", 999.99, 10, "Electronics")  
 - (2, "Headphones", 49.99, 50, "Accessories")  
 - Table `orders` :  
 - (1, 1, "2024-03-01", 1099.98, "Completed")  
 - (2, 2, "2024-03-05", 49.99, "Pending")  
 - Table `order\_details` :  
 - (1, 1, 1, 1, 999.99)  
 - (2, 1, 2, 2, 99.98)

### 4. Lecture et filtrage de données (30 min)

1. Écrivez des requêtes SQL pour :  
 - Afficher tous les produits commandés avec le nom de l'utilisateur, triés par `order\_date`.  
 - Filtrer les commandes dont le statut est "Pending".  
 - Trouver le produit le plus cher vendu.  
 - Calculer le total des ventes par catégorie.  
 - Lister les utilisateurs n'ayant passé aucune commande.

### 5. Modification de données (20 min)

1. Modifiez le stock du produit "Laptop" en le diminuant de 1.  
2. Changez le statut de la commande de Bob à "Completed".  
3. Mettez à jour le prix des produits de la catégorie "Accessories" en appliquant une réduction de 10%.

### 6. Les contraintes (15 min)

1. Ajoutez une contrainte UNIQUE sur la colonne `email` de la table `users`.  
2. Ajoutez une contrainte de clé étrangère entre `user\_id` de `orders` et `id` de `users`.  
3. Assurez-vous que la quantité dans `order\_details` ne peut pas être négative.

### 7. Les jointures (20 min)

1. Affichez tous les détails des commandes, avec le nom du produit et l'email de l'utilisateur.  
2. Affichez le total dépensé par chaque utilisateur.  
3. Trouvez tous les produits jamais commandés.

### 8. Transactions et procédures (15 min)

1. Implémentez une transaction qui :  
 - Insère une nouvelle commande pour un utilisateur.  
 - Met à jour le stock du produit commandé.  
 - Annule la transaction si le stock est insuffisant.

2. Créez une procédure stockée pour ajouter un produit et vérifier automatiquement la catégorie.

### 9. Fonction, variables, tables temporaires et vue (20 min)

1. Créer une fonction utilisateur :  
Écrivez une fonction appelée `get\_user\_orders` qui prend l'ID d'un utilisateur en paramètre et retourne tous les détails des commandes passées par cet utilisateur.  
   
2. Utilisation des variables :  
Déclarez une variable pour stocker la quantité totale de produits commandés par un utilisateur spécifique.  
   
3. Créer une table temporaire :  
 Créez une table temporaire pour stocker les produits dont le stock est inférieur à une certaine limite.

4. Créer une vue :  
 Créez une vue pour afficher les utilisateurs et leur total dépensé sur le site e-commerce.