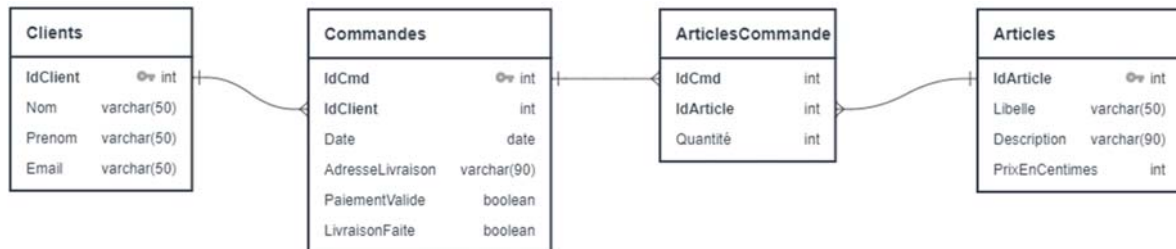


EXERCICE 2 : Base de données d'une plateforme de vente en ligne (4 points)



Cet exercice traite principalement du thème « traitement de données en tables et bases de données ». Nous nous interrogerons dans cet exercice sur la modélisation et l'utilisation des données nécessaires aux fonctionnements de sites de vente en ligne.

Partie A : Modèle relationnel

Voici le modèle relationnel qui sera utilisé dans cet exercice :



Lecture :

- un symbole  identifie une clé primaire
- un trait  entre deux attributs indique qu'ils doivent partager les mêmes valeurs. Ces valeurs sont prises de manière unique dans la relation du côté **+** et zéro, une ou plusieurs fois du côté **≤** à condition d'être présente du côté **+**.

1. Donner les clés primaires des relations **Clients** et **Articles**.
2. Les commandes ci-dessous ont été utilisées pour créer ces deux relations. Donner le domaine (c'est-à-dire le type) des attributs **Email** et **Quantite**.

```
CREATE TABLE Clients (  
  IdClient INT PRIMARY KEY,  
  Nom VARCHAR(50),  
  Prenom VARCHAR(50),  
  Email VARCHAR(50)  
);  
  
CREATE TABLE ArticlesCommande (  
  IdCmd INT,  
  IdArticle INT,  
  Quantite INT,  
  PRIMARY KEY (IdCmd, IdArticle)  
  FOREIGN KEY(IdArticle) REFERENCES Articles(IdArticle)  
);
```

3. En vous inspirant des commandes ci-dessus, recopier et compléter la commande suivante qui permet de créer la relation `Commandes` en précisant sa clé étrangère.

```
CREATE TABLE Commandes (  
  IdCmd INT PRIMARY KEY,  
  IdClient INT,  
  Date DATE,  
  AdresseLivraison VARCHAR(90),  
  PaiementValide BOOLEAN,  
  LivraisonFaite BOOLEAN,  
  FOREIGN KEY(.....) REFERENCES .....  
  (.....)  
);
```

Partie B : Site web

La plateforme de vente en ligne possède un site web pour ses clients qui passent des commandes en remplissant un formulaire.

1. Expliquer pourquoi, lorsqu'un formulaire contient une quantité importante de données, il est préférable d'utiliser la méthode POST plutôt que GET.
2. Pour la validation du paiement, est-il préférable d'utiliser le protocole HTTP ou HTTPS ? Pourquoi ?
3. Expliquer l'intérêt de vérifier, avant la validation du formulaire, le format des informations saisies (par exemple qu'il n'y a pas de chiffre dans le nom ou qu'il y a bien un @ dans l'adresse de courriel).

Partie C : Requêtes SQL

1. Écrire une requête SQL permettant de récupérer l'identifiant et le libellé de tous les articles coûtant moins de 15 euros.
2. Expliquer ce que fait la requête SQL suivante.

```
SELECT u.IdClient, u.Email, v.IdCmd, v.AdresseLivraison  
FROM Clients as u JOIN Commandes as v  
ON u.IdClient = v.IdClient  
WHERE v.PaiementValide = False;
```

3. Écrire une requête SQL permettant de récupérer le libellé des articles de la

commande 1345.

4. On suppose que l'attribut `IdArticle` de la table `Articles` est auto-incrémenté et ne doit donc pas être précisé lors de l'ajout d'un nouvel article. Écrire une requête SQL permettant d'ajouter dans la base ce nouvel article.



« Cet imperméable se replie en forme de pochette. »

Prix : 9,99 euros

Partie D : Adaptation du modèle relationnel

Le propriétaire du site souhaite une adaptation du modèle relationnel afin de :

- comptabiliser le stock pour chaque article
 - pouvoir mémoriser, pour chaque client, une adresse de livraison par défaut de ses commandes.
1. Préciser, pour chaque relation que vous jugez nécessaire de modifier, les attributs ajoutés ainsi que leurs domaines.
 2. À l'arrivée d'une nouvelle commande d'un client, l'algorithme de mise à jour de la base de données ci-dessous est exécuté.
Indiquer l'erreur présente dans cet algorithme.

Algorithme nouvelle_commande (commande)

Début

Pour chaque article de la commande **Faire**

Si Quantité ≤ Stock **Alors**

 Stock ← Stock – 1

Sinon

 annuler l'achat de cet article

FinSi

FinPour

Fin