

Epreuve théorique N°1 / NSI Corrigé

Bases de données

Question 1 :

Les champs sont : Prénom_client ; Nom_client ; mail_client ; Nom_produit ; Prix_produit_unitaire ; Date_commande et Nombre_produit_commande.

Question 2 :

Trois inconvénients majeurs :

- Problèmes de sécurité : on peut effacer n'importe quelle donnée et même le fichier en entier ! La perte d'un fichier client est une catastrophe pour une entreprise.
- Problèmes de redondances : la répétition de données est source d'erreurs.
- Problèmes d'organisation : le nombre important de champs rend la manipulation de données complexe et peu efficace.

Question 3 :

L'attribut id_cl est la clé primaire de la table *Client* et a pour rôle d'identifier de manière unique un enregistrement.

Question 4 :

Cette requête entre les caractéristiques de clients issues du fichier .CSV dans la table *Client*.

Question 5 :

Une clé primaire peut être entrée manuellement ou être en auto-incrémentation : la base de données se charge elle-même d'attribuer sa valeur à chaque enregistrement (valeur entière).

Question 6 :

On peut proposer comme requête:

```
SELECT * from Client;
```

Question 7 :

Voici la table Produit :

CHAR id_pr (clé primaire en chaîne de caractères car pas d'opérations à effectuer)

CHAR nom_pr

FLOAT prix_pr

Question 8 :

En s'appuyant sur la question 5/, on peut proposer comme requête :

```
INSERT INTO Produit (id_pr, nom_pr, prix_pr)
VALUES ('1', 'Tubes de colle', 2.2),
('2', 'Ciseaux', 4.2),
('3', 'Feuilles calque', 3.8),
('4', 'Copies doubles', 1.9) ;
```

Question 9 :

Une clé étrangère permet de relier plusieurs tables entre elles. Elle est toujours reliée à la clé primaire d'une autre table.

Question 10 :

Voici la table Commande :

```
INT id_com (clé primaire en auto-incrémentation)
CHAR id_pr (clé étrangère reliant la table Produit)
INT id_cl (clé étrangère reliant la table Client)
DATE date
INT nombre_com
```

Question 11 :

On peut proposer comme requête :

```
SELECT nom_cl
FROM Client
JOIN Commande ON Client.id_cl = Commande.id_cl
WHERE Commande.date = '10/12/2020' ;
```

Autre possibilité :

```
SELECT nom_cl
FROM Client, Commande
WHERE Client.id_cl = Commande.id_cl AND Commande.date = '10/12/2020' ;
```

On obtient 'Dupond' répété trois fois.

Remarque : Il faudrait utiliser le mot clé DISTINCT pour éviter les répétitions.

```
SELECT DISTINCT nom_cl
FROM Client
JOIN Commande ON Client.id_cl = Commande.id_cl
WHERE Commande.date = '10/12/2020' ;
```

Question 12 :

On peut proposer comme requête :

```
SELECT nom_pr
FROM Produit
JOIN Commande ON Produit.id_pr = Commande.id_pr
JOIN Client ON Client.id_cl = Commande.id_cl
WHERE Client.nom_cl = 'Talon' AND Client.prénom_cl = 'Marie' ;
```

Autre possibilité :

```
SELECT Client.nom_cl
```

```
FROM Client, Commande, Produit
```

```
WHERE Client.id_cl = Commande.id_cl AND Produit.id_pr = Commande.id_pr  
      AND Client.nom_cl = 'Talon' AND Client.prénom_cl = 'Marie' ;
```