

Secteur Tertiaire Informatique
Filière « Etude et développement »

Séquence « Développer des pages Web »

SQL & MySQL

Apprentissage

Mise en pratique

Evaluation



1. EXERCICES DE SQL SUR LA BASE MAGASIN

En utilisant les scripts de création de la base **Magasin** qui vous ont été fourni, vous devrez créer en local votre Base Magasin.

Ensuite, vous devrez répondre aux questions suivantes :

Question 1

Dans la base magasin, sélectionnez les articles dont le prix est inférieur à 1 500 euros.

Question 2

Dans la base magasin, sélectionner les clients qui habitent à Paris

Question 3

Dans la base magasin, sélectionner les clients qui ont un âge supérieur à 40 ans.

Question 4

Dans la base magasin, sélectionner les clients qui n'ont pas d'adresse email

Question 5

Dans la base magasin, sélectionner les clients dont le nom commence par une lettre comprise entre A et H, la lettre H étant non comprise.

Question 6

Dans la base magasin, sélectionner les clients dont l'âge figure dans la liste (18,19,20)

Question 7

Dans la base magasin, sélectionner les clients qui habitent une avenue, et dont l'adresse contient donc la chaîne de caractères « Av » (sensible à la casse).

Question 8

Dans la base magasin, sélectionnez les articles dont le prix est compris entre 100 et 500 euros.

Question 9

Dans la base magasin, sélectionnez tous les articles de marque Nikon (dont la désignation contient ce mot).

Question 10

Dans la base magasin, sélectionnez tous les caméscopes, leur prix et leur référence.

Question 11

Dans la base magasin, sélectionner les clients dont le nom commence par **Ma** à l'aide d'une *expression régulière*.

Question 12

Dans la base magasin, sélectionner les clients qui ont moins de 30 ans et qui habitent à Paris.

Question 13

Dans la base magasin, sélectionner les clients qui habitent à **Lyon** ou dont le nom commence par la lettre **H**.

Question 14

Dans la base magasin, sélectionner les clients qui n'habitent ni à **Lyon** ni à **Paris**.

Question 15

Dans la base magasin, calculer l'âge moyen des clients

Question 16

Dans la base magasin, calculer le nombre de villes différentes de la table client.

Question 17

Dans la base magasin, sélectionnez tous les produits de la catégorie informatique, et affichez leur code, leur désignation et leur prix par ordre décroissant de prix.

Question 18

Dans la base magasin, sélectionnez tous les clients de moins de 40 ans, et ordonnez les résultats par ville en ordre alphabétique.

Question 19

Dans la base magasin, calculez le prix moyen de tous les articles.

Question 20

Dans la base magasin, calculez le nombre d'e-mails non NULL et distincts l'un de l'autre.

Question 21

Dans la base magasin, déterminer l'âge du client le plus jeune.

Question 22

Dans la base magasin, calculer le nombre total des articles commandés

Question 23

Dans la base magasin, calculer l'âge moyen des clients par ville.

Question 24

Dans la base magasin, afficher l'âge minimum, l'âge maximum et l'âge moyen des clients par ville.

Question 25

Dans la base magasin, calculer l'âge moyen des clients qui habitent une ville dont le nom commence par un **L**

Question 26

Dans la base magasin, sélectionnez tous les articles commandés par chaque client.

Question 27

Dans la base magasin, sélectionnez tous les clients dont le montant d'une commande dépasse 1 500 euros.

Question 28

Dans la base magasin, sélectionnez tous les clients dont le montant total de toutes les commandes dépasse 5 000 euros.

Question 29

Dans la base magasin, à l'aide d'une jointure, retrouver toutes les commandes faites par un client. (Donner les deux écritures possibles de cette requête, celle qui utilise WHERE et celle qui utilise INNER JOIN)

Question 30

Dans la base magasin, à l'aide d'une jointure, afficher la liste des articles les plus vendus sur le site et à afficher leur code, leur désignation, leur prix et le nombre total d'articles. (Donner les deux écritures possibles de cette requête, celle qui utilise WHERE et celle qui utilise INNER JOIN)

Question 31

Dans la base magasin et dans la table client ajouter un champ code postal après le champ ville.

Vous avez donc maintenant un nouveau champ code postal dont les valeurs sont nulles pour chaque client.

Créer une contrainte sur le **Code Postal Client** pour que, s'il est non renseigné il soit par défaut fixé à '19240' (Allassac) ou bien compris entre '01000' et '99999'. Vous devrez créer pour cela un **trigger** qui enverra un message d'erreur lorsque cette contrainte ne sera pas respectée avant l'insertion et la modification d'un enregistrement.

Vous pourrez vous référer à ce lien :

<https://dev.mysql.com/doc/refman/8.0/en/signal.html>

... et à la documentation fournie pour en apprendre davantage sur le concept de **trigger** de **MySQL**

Faire un essai avec par exemple :

```
INSERT INTO `client`(`nom`, `prenom`, `age`, `adresse`, `ville`,  
`code_postal`, `mail`) VALUES ("RESTOUEIX", "Sacha", 52, "25 Route de la  
Plaine", "Allassac", "00999", "sacha8milo@gmail.com")
```

```
INSERT INTO `client`(`nom`, `prenom`, `age`, `adresse`, `ville`,  
`code_postal`, `mail`) VALUES ("RESTOUEIX","Sacha",52,"25 Route de la  
Plaine","Allassac","92300","sacha8milo@gmail.com")
```

Question 32

Vous allez ajouter un champs **version** de type **int** à la table **client** de la base **magasin**. Avec par défaut la valeur 0.

Vous allez créer une nouvelle table **historique_client** qui aura la structure suivante :

```
CREATE TABLE historique_client (  
  id BIGINT NOT NULL AUTO_INCREMENT,  
  action ENUM ( 'update' , 'delete' ) DEFAULT NULL,  
  date_action DATETIME DEFAULT NULL,  
  version BIGINT NOT NULL DEFAULT 0,  
  id_original BIGINT NOT NULL DEFAULT 0,  
  nom VARCHAR(50) DEFAULT NULL,  
  PRIMARY KEY (id),  
  KEY (id_original),  
  KEY (action),  
  KEY (date_action),  
  KEY (version)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Vous allez devoir écrire un **Trigger** qui avant chaque **modification** :

– A chaque **modification** d'un enregistrement de la table « **client** », il va recopier les données dans la table « **historique_client** » et gérer les champs qui servent à l'historique. C'est à dire incrémenter le numéro de **version**, renseigner le champ « **action** » à la valeur « **update** » et le champ « **date_action** » à la date et heure courante.

Vous ferez de même après chaque **suppression** d'un **client** :

– A chaque **suppression** d'un enregistrement de la table « **client** », il va recopier les données (qui vont être supprimées dans la table d'origine) et là encore gérer les champs qui servent à l'historique. C'est à dire incrémenter le numéro de **version**, renseigner le champ « **action** » cette fois-ci avec la valeur « **delete** » et le champ « **date_action** » à la date et heure courante.

Vous pourrez tester ensuite qu'à chaque modification d'un client, la table **historique_client** sera alimentée.

Pour la **suppression**, vous ne pourrez bien entendu pas supprimer des clients qui sont engagés dans une **commande** sur le principe de l'intégrité des données portant sur les **clefs étrangères**.