|  |
| --- |
| Travail pratique 4 |
| TP4 – LDD – LMD & Les jointures de MySQL (12%)  /110 pts |

Nom(s) et prénom(s) : Groupe :

Objectif :

* Se familiariser avec les concepts de base de MySQL;
  + Langage de Manipulation de Données (LMD)
  + Langage de Définition de Données (LDD)
  + Les jointures

À remettre :

* + Ce document Word complété avec vos Nom et Prénom, contenant toutes les requêtes des questions de 1 à 29.

Consigne**:**

* Ce TP4 peut se faire en équipe de deux;
* Date de remise : voir MIO.

**Partie I**

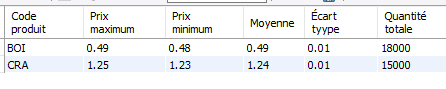
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Base de données: BDTP4votrenom Table: produit  Clé primaire: pro\_id | | | | |
| pro\_id INT | pro\_code CHAR(3) | pro\_nom VARCHAR(30) | pro\_qte INT | pro\_prix DECIMAL(10,2) |
| 1001 | CRA | Crayon Rouge | 5000 | 1.23 |
| 1002 | CRA | Crayon Bleu | 8000 | 1.25 |
| 1003 | CRA | Crayon Noir | 2000 | 1.25 |
| 1004 | BOI | Boite 2B | 10000 | 0.48 |
| 1005 | BOI | Boite 2H | 8000 | 0.49 |

1. Créer la base de données *BDTP4votreNom*et la table *produit* (5 pts).

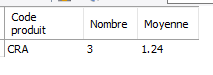
Donner les **requêtes SQL** correspondantes aux questions ci-dessous. Il faut faire attention aux Alias.

À titre d’information, j’ai copié les captures d’écran ci-dessous, le vendredi 19 avril 2024.

1. Afficher la structure de la table *Produit (*1 pt*).*
2. Insérer les données dans la table Produit (3 pts).
3. (5 pts) Afficher pour chaque code produit (on a deux code BOI et CRA, groupé par code produit), le prix maximum, le prix minimum, la moyenne des prix, l’écart type (STD) et la quantité totale. N’oubliez pas de trier la liste selon le code produit comme ceci :

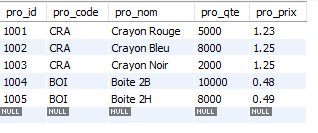


1. Afficher le code du produit, le nombre de produits et la moyenne des prix groupés par code produit. Afficher ceux dont le nombre de produits est supérieur à 2 (4 pts).

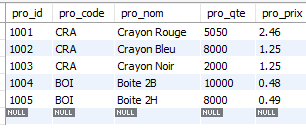


1. Mettre à jour la table produit en ajoutant 50 à la quantité et 1.23 aux prix, à tous les crayons rouges (4 pts).

Avant

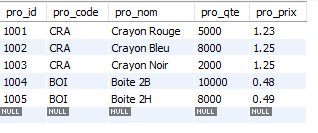


Après

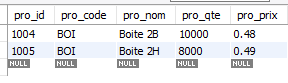


1. Supprimer tous les produits *Crayons*. Ensuite, annuler la suppression avec la commande : ROLLBACK (5 pts).

Avant suppression

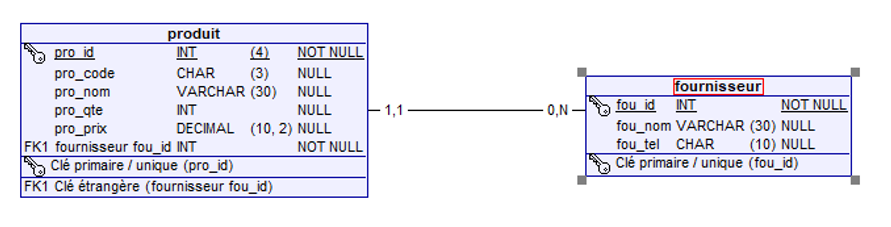


Après suppression



1. Créer la table *Fournisseur* suivante (3 pts):

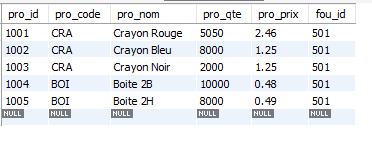
|  |  |  |
| --- | --- | --- |
| Base de données: BDTP4 Table: Fournisseur  Clé primaire: four\_id | | |
| fou\_id INT | fou\_nom VARCHAR (30) | fou\_tel CHAR(10) |
| **501** | ABC Vente | 4181112222 |
| **502** | XYZ Compagnie | 5143334444 |
| **503** | QQ Coop | 6131118888 |
| **504** | QU Quebec | 4185557777 |



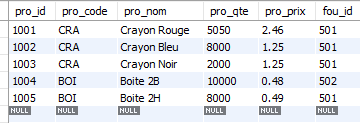
1. Insérer les données dans la table *Fournisseur (3 pts).*
2. Chaque produit a un seul fournisseur (10 pts).
   1. Ajouter dans la table produit, la **CLÉ ÉTRANGÈRE** *fou\_id*, en utilisant :

*ALTER TABLE*…

* 1. Afficher la nouvelle structure de la table *produit*
  2. Mettre à jour (UPDATE) la table *produit*, en ajoutant le fournisseur *fou\_id = 501* à tous les produits.

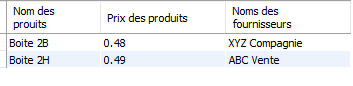


* 1. le produit no *1004* a pour fournisseur *le no 502,* faire la mise à jour de la table produit.



1. En utilisant les jointures, afficher le nom du produit, le prix du produit inférieur à 1 dollar et le nom du fournisseur (Donner deux solutions : SQL-1 et SQL-2) (5 pts).

Attention aux alias!



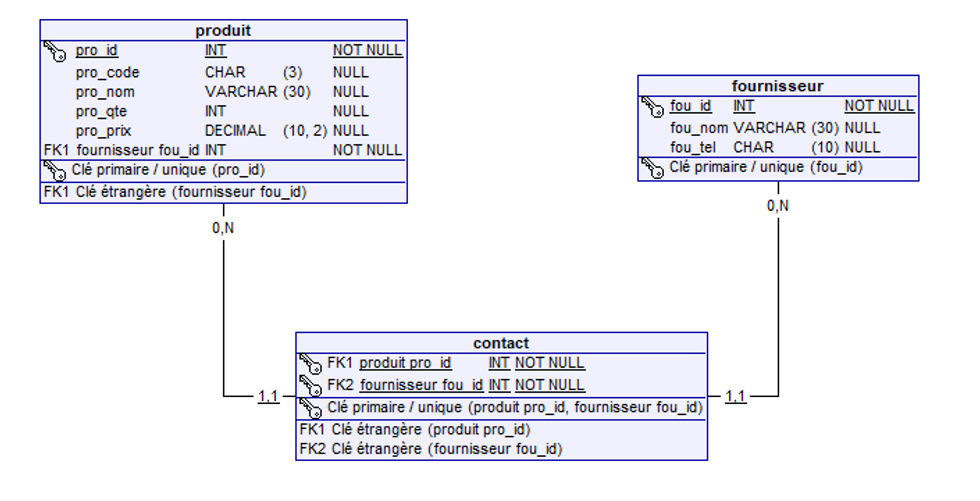
1. Ajouter dans la table produit, l’enregistrement suivant (2 pts):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id produit** | **Code produit** | **Nom produit** | **Quantité produit** | **Prix produit** | Id Fournisseur |
| 1006 | BOI | Boite 2C | 7000 | 2.23 | 506 |

Est-ce normal que l’insertion puisse avoir lieu ? Justifier votre réponse en proposant une solution précise (question très importante!).

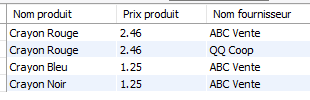
1. Supprimer la colonne *fou\_id* de la table *produit*, en utilisant ALTER TABLE (4 pts).
2. Supprimer le produit no 1006 (2 pts).
3. Créer la table d’intersection entre produit et Fournisseur en la nommant: ***contact*** (5 pts)

|  |  |
| --- | --- |
| Base de données: BDTP4 Table: contact  Clé primaire: pro\_id, fou\_id | |
| pro\_id INT  (Clé étrangère) | fou\_id INT  (Clé étrangère) |
| 1001 | 501 |
| 1002 | 501 |
| 1003 | 501 |
| 1004 | 502 |
| 1001 | 503 |

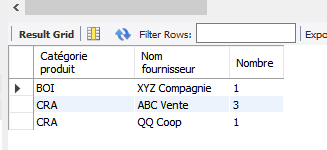


1. Insérer les données dans la table *contact*. Vous aurez peut-être des messages d’erreurs, dans ce cas à vous de trouver des solutions pour faciliter l’insertion des données (2 pts).
2. Afficher la liste des noms des produits, les prix des produits supérieurs à 1$ et les noms des fournisseurs des produits (Jointure impliquant les trois tables).

Proposer deux solutions (SQL-1 et SQL-2)- (6 pts)



1. Afficher les catégories des produits accompagnées des noms des fournisseurs ainsi que le nombre de produits par fournisseur (3 pts).



1. Reprendre la question 18 en affichant seulement les catégories dont le nombre de produits par fournisseur est supérieur à 2 (2 pts).



1. Afficher la liste des fournisseurs qui n’ont pas encore fourni aucun produit (intersection entre deux tables : contact et fournisseur (Sous-requêtes)) (3 pts).



1. Ajouter à la table *Produit*, la colonne suivante : *pro\_date\_achat*.

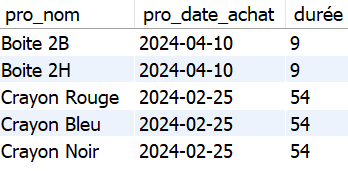
Utiliser *ALTER TABLE* …. (2 pts)

1. Mettre à jour (UPDATE) le champ *pro\_date\_achat*, de la table Produit avec les données suivantes (6 pts):
   1. *2024-02-25*, pour les produits : 1001,1002 et1003. La mise à jour doit se faire en une seule opération.
   2. *2024-04-10*, pour les produits : 1004 et 1005. La mise à jour doit se faire en une seule opération.
2. Faire les opérations suivantes sur les dates (6 pts):
   1. (3 pts) Afficher la liste des produits achetés entre le 2024*-03-15* et la date d’aujourd’hui. Trier le résultat dans l’ordre décroissant des numéros des produits.

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement

* 1. (3 pts) Afficher la liste des produits, les dates d’achat ainsi que la durée en jours depuis les achats, trier le résultat dans l’ordre décroissant de la date d’achat.



**Partie II**

1. Vous allez modifier la base de données *Employe.sql*, que nous avons utilisée depuis le début de la session en ajoutant une table *affectation*. La table *affectation* ne contient que trois colonnes : *emp\_matr*, *dep\_num et aff\_date*. Pour cela d’abord vous allez reproduire le MCD ci-dessous avec OpenMOdelSphere et insérer le ici (3 pts). N’oubliez pas d’ajouter votre nom dans la cartouche.

Une image contenant texte, capture d’écran, ligne, diagramme

Description générée automatiquement

1. Convertir ce MCD en MRD et insérer le ici (2 pts).

Ici vous allez avoir une nouvelle table affectation. Seule cette table *affectation* qui doit permettre de connaitre l’appartenance de **chaque employé à son département.** Maintenantpas de clé étrangère *num departement* dans la table *employe*.

1. Renommer Employe.sql par *EmployeAffectation.sql.* (6 pts)

Faire les changements nécessaires, puis insérer les 14 enregistrements dans la table affectation. Mettre les mêmes dates d’affectation pour *EmployeAffectation.sql* qui étaient les dates d’embauche dans la version Employe.sql.

Insérer le script complet de création de la BD ici.

1. Le commis Armand, matricule 7776 travaille au département 20 depuis le 03-12-2011. Maintenant il va être réaffecté au département 30 à la date d’aujourd’hui. Pour cela vous allez insérer cette nouvelle affectation pour le commis Armand (2 pts). Voici le résultat après l‘insertion de cette nouvelle affectation pour le commis Armand.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

1. *Armand* après un certain temps dans son nouveau département 30, peut-il être réaffecté dans son ancien département le No 20 ?

Si oui insérer la nouvelle affectation *d’Armand*, sinon justifier votre réponse. Cette base de données permet-elle de retracer deux affectations d’un employé ? Justifier votre réponse (2 pts).

1. Trouver la liste des employés qui gagnent plus que Biron (4 pts).

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**FIN du TP4**