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

Nom(s) et prénom(s) : Rafael Carvalho Matias & Jérôme Ouellet 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*).*

SELECT \* FROM produit;

1. Insérer les données dans la table Produit (3 pts).

INSERT INTO produit (pro\_id, pro\_code, pro\_nom, pro\_qte, pro\_prix)

VALUES

(1001, 'CRA', 'Crayon Rouge', 5000, 1.23),

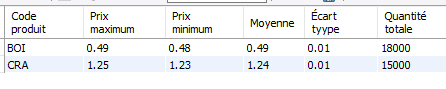
(1002, 'CRA', 'Crayon Bleu', 8000, 1.25),

(1003, 'CRA', 'Crayon Noir', 2000, 1.25),

(1004, 'BOI', 'Boite 2B', 10000, 0.48),

(1005, 'BOI', 'Boite 2H', 8000, 0.49);

1. (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 :



SELECT

pro\_code,

MAX(pro\_prix) AS prix\_max,

MIN(pro\_prix) AS prix\_min,

AVG(pro\_prix) AS prix\_moyen,

STD(pro\_prix) AS ecart\_type\_prix,

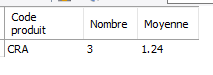
SUM(pro\_qte) AS quantite\_totale

FROM produit

GROUP BY pro\_code

ORDER BY pro\_code;

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).



SELECT

pro\_code as "Code produit",

COUNT(pro\_nom) as "nombre",

format(AVG(pro\_prix),2) as "Moyenne"

FROM produit

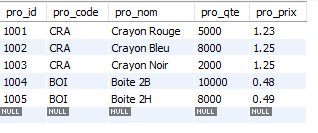
Group by pro\_code

Having COUNT(pro\_nom) > 2

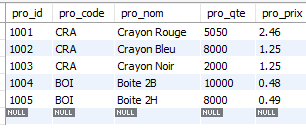
Order by pro\_code;

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



SET AUTOCOMMIT = 0;

SET SQL\_SAFE\_UPDATES = 0;

UPDATE produit

SET pro\_qte = pro\_qte + 50,

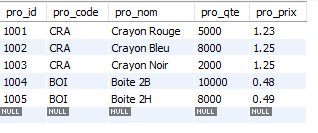
pro\_prix = pro\_prix + 1.23

WHERE pro\_nom = "Crayon Rouge";

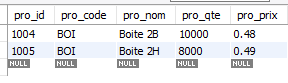
COMMIT;

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

Avant suppression



Après suppression



Set autocommit = 0;

Delete From produit

where pro\_nom like "Crayon%";

select \* from produit;

ROLLBACK;

select \* from produit;

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 |

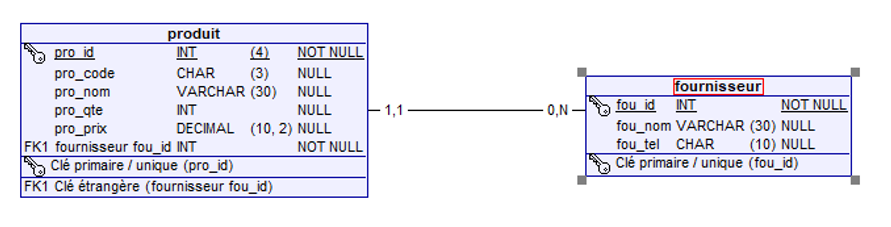
CREATE TABLE fournisseur (

fou\_id INT primary key NOT NULL,

fou\_nom varchar(30) NULL,

fou\_tel CHAR(10) NULL

);



1. Insérer les données dans la table *Fournisseur (3 pts).*

insert INTO fournisseur (fou\_id, fou\_nom, fou\_tel) VALUES

(501, "ABC Vente", 4181112222),

(502, "XYZ Compagnie", 5143334444),

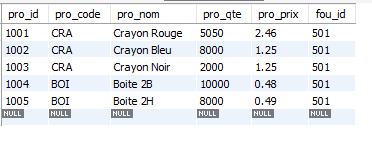
(503, "QQ Coop", 6131118888),

(504, "QU Quebec", 4185557777);

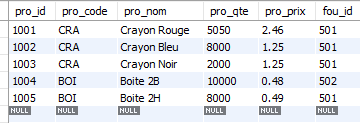
1. Chaque produit a un seul fournisseur (10 pts).
   1. Ajouter dans la table produit, la **CLÉ ÉTRANGÈRE** *fou\_id*, en utilisant :

*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.



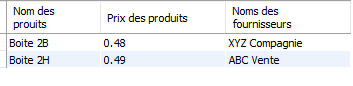
ALTER TABLE produit ADD COLUMN pro\_fou\_id INT, ADD CONSTRAINT fk\_pro\_fou\_id foreign key (pro\_fou\_id) REFERENCES fournisseur (fou\_id);

update produit set pro\_fou\_id = 501;

update produit set pro\_fou\_id = 502 where pro\_id = 1004;

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!



select pro\_nom as "Nom des produits",

pro\_prix as "Prix produits",

fou\_nom as "Nom des fournisseur"

from produit

inner join fournisseur on pro\_fou\_id = fou\_id

where pro\_prix < 1;

select pro\_nom as "Nom des produits",

pro\_prix as "Prix produits",

fou\_nom as "Nom des fournisseur"

from produit, fournisseur

where pro\_fou\_id = fou\_id

and pro\_prix < 1;

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!).

Elle ne devrait pas avoir lieu car la cle etrangere 506 n’existe pas

Il faudrait ajouter un fournisseur avec cette clé :

INSERT INTO fournisseur (fou\_id, fou\_nom, fou\_tel) VALUES (506, 'Naji (best prof ever)', '123456789');

INSERT INTO produit (pro\_id, pro\_code, pro\_nom, pro\_qte, pro\_prix, pro\_fou\_id)

VALUES (1006, 'BOI', 'Boite 2C', 7000, 2.23, 506);

1. Supprimer la colonne *fou\_id* de la table *produit*, en utilisant ALTER TABLE (4 pts).

alter table produit drop foreign key fk\_pro\_fou\_id;

alter table produit drop column pro\_fou\_id;

1. Supprimer le produit no 1006 (2 pts).

delete from produit where pro\_id = 1006;

1. 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 |

CREATE TABLE contact (

pro\_id INT,

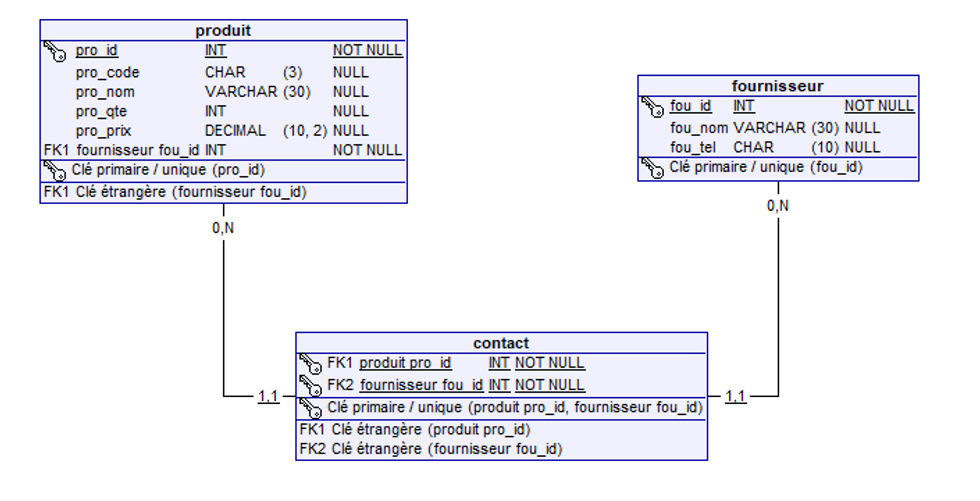
fou\_id INT,

PRIMARY KEY (pro\_id, fou\_id),

FOREIGN KEY (pro\_id) REFERENCES produit(pro\_id),

FOREIGN KEY (fou\_id) REFERENCES fournisseur(fou\_id)

);



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).

INSERT INTO contact (pro\_id, fou\_id)

VALUES

(1001, 501),

(1002, 501),

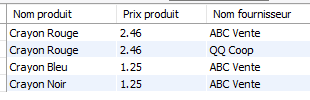
(1003, 501),

(1004, 502),

(1001, 503);

1. 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)



select p.pro\_nom as 'Nom Produit', p.pro\_prix as 'Prix produit', f.fou\_nom as 'Nom des fournisseurs'

from produit p

inner join contact c on p.pro\_id = c.pro\_id

inner join fournisseur f on c.fou\_id = f.fou\_id

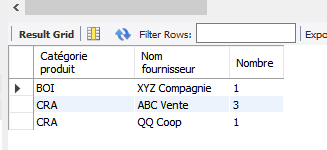
where pro\_prix > 1;

select p.pro\_nom as 'Nom Produit', p.pro\_prix as 'Prix produit', f.fou\_nom as 'Nom des fournisseurs'

from produit p, contact c, fournisseur f

where p.pro\_id = c.pro\_id and f.fou\_id = c.fou\_id and pro\_prix > 1;

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



select p.pro\_code as 'Categorie produit', f.fou\_nom as 'nom fournisseur', count(\*) as nombre

from produit p, fournisseur f, contact c

where f.fou\_id = c.fou\_id and p.pro\_id = c.pro\_id

group by p.pro\_code, f.fou\_nom

order by p.pro\_code, nombre desc;

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



select p.pro\_code as 'Categorie produit', f.fou\_nom as 'nom fournisseur', count(\*) as nombre

from produit p, fournisseur f, contact c

where f.fou\_id = c.fou\_id and p.pro\_id = c.pro\_id

group by p.pro\_code, f.fou\_nom

having count(\*) > 2

order by p.pro\_code, nombre desc;

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).



select fou\_nom

from fournisseur

where fou\_id not in (select fou\_id from contact);

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

Utiliser *ALTER TABLE* …. (2 pts)

alter table produit add column pro\_date\_achat date;

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.

UPDATE produit

SET pro\_date\_achat = '2024-02-25'

WHERE pro\_id IN (1001, 1002, 1003);

* 1. *2024-04-10*, pour les produits : 1004 et 1005. La mise à jour doit se faire en une seule opération.

UPDATE produit

SET pro\_date\_achat = '2024-04-10'

WHERE pro\_id IN (1004, 1005);

1. 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

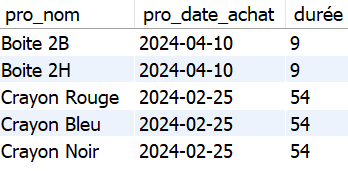
Description générée automatiquement

select \* from produit

where pro\_date\_achat between '2024-03-15' and (select curdate())

order by pro\_id desc;

* 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.



select pro\_nom, pro\_date\_achat,DATEDIFF(CURDATE(), pro\_date\_achat) as 'nombre'

from produit

order by pro\_date\_achat desc;

**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**