

Secteur Tertiaire Informatique
Filière « Etude et développement »

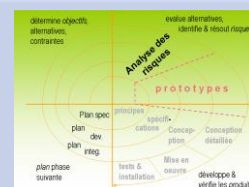
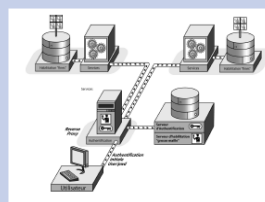
Séquence « Développer des composants d'accès
aux données »

Cas PAPYRUS MySQL :
Programmer dans le langage du SGBD

Apprentissage

Mise en pratique

Evaluation



Dates de mise à jour :

05/02/23 – Damien Bin – Ordre des exercices

30/01/24 – Damien Bin – Ajout des résultats attendus pour chaque exercice

TABLE DES MATIERES

Table des matières	3
1. L'EXISTANT	5
2. LES PROCEDURES STOCKEES	6
2.1 Création d'une procédure stockée sans paramètre	6
2.2 Création d'une procédure stockée avec un paramètre en entrée	6
2.3 Création d'une procédure stockée avec des paramètres en entrée et en sortie	6
2.4 Créez le message d'erreur : « Fournisseur inexistant »	7
3. LES FONCTIONS	7
3.1 Fonction <code>fn_DateFormat()</code>	7
3.2 Création d'une fonction <code>fn_Satisfaction()</code>	7
3.3 Création d'une fonction <code>fn_CA_Fournisseur()</code>	8
3.4 Test de la fonction <code>FOUND_ROWS()</code>	8

Préambule

Ce document expose les besoins de programmation du SGBD relatifs au Cas Papyrus pour lequel il a été créé une base de données.

Objectifs

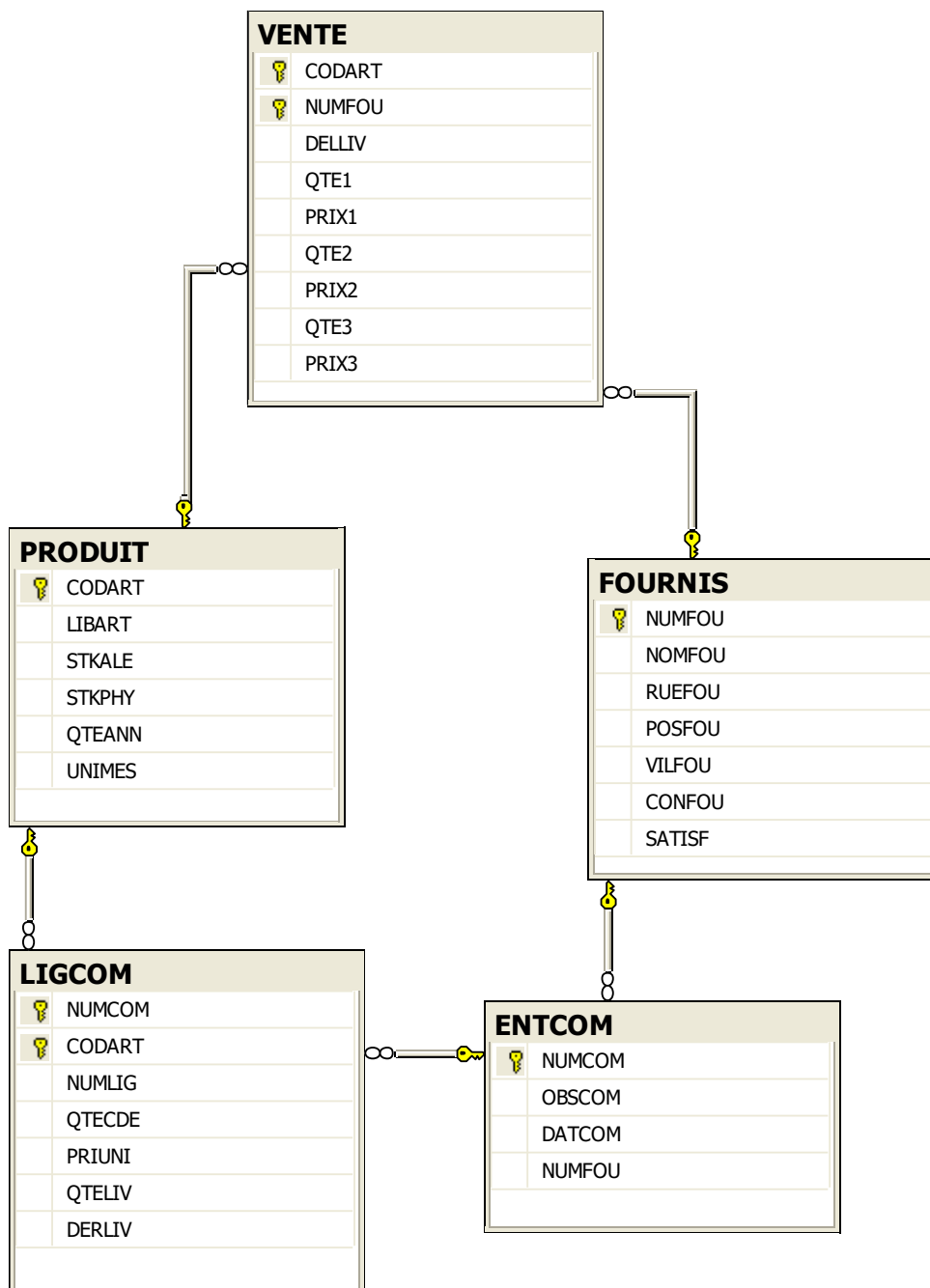
Mettre en pratique la programmation du SGBD MySQL tant au niveau des fonctions que des procédures stockées.

Méthodologie

Réalisez les exercices progressivement, après étude des ressources de formation concernant chacun des chapitres, programmation des fonctions et programmation des procédures stockées.

1. L'EXISTANT

La base de données relationnelle PAPYRUS est constituée des relations suivantes :



PRODUIT (CODART, LIBART, STKALE, STKPHY, QTEANN, UNIMES)

ENTCOM (NUMCOM, OBSCOM, DATCOM, NUMFOU)

LIGCOM (NUMCOM, CODART, NUMLIG, QTECDE, PRIUNI, QTELIV, DERLIV)

FOURNIS (NUMFOU, NOMFOU, RUEFOU, POSFOU, VILFOU, CONFOU, SATISF)

VENTE (CODART, NUMFOU, DELLIV, QTE1, PRIX1, QTE2, PRIX2, QTE3, PRIX3)

2. LES PROCEDURES STOCKEES

2.1 CREATION D'UNE PROCEDURE STOCKEE SANS PARAMETRE

Créez la procédure stockée **prc_LST_fournis** correspondant à la requête

« Afficher le code des fournisseurs pour lesquels une commande a été passée ».

```
SELECT DISTINCT numfou FROM entcom;
```

Exécutez-la pour vérifier qu'elle fonctionne conformément à votre attente.

(Résultat : 6 NUMFOU)

2.2 CREATION D'UNE PROCEDURE STOCKEE AVEC UN PARAMETRE EN ENTREE

Créer la procédure stockée **prc_LST_Commandes**, qui liste les commandes ayant un libellé particulier dans le champ OBSCOM (exemple avec « urgent »)

```
SELECT ligcom.numcom, fournis.nomfou, libart
FROM entcom
  INNER JOIN fournis
    ON entcom.numfou = fournis.numfou
  INNER JOIN ligcom
    ON entcom.numcom = ligcom.numcom
  INNER JOIN produit
    ON ligcom.codart = produit.codart
WHERE obscom LIKE '%urgent%';
```

(Résultat : 3 commandes en urgence)

2.3 CREATION D'UNE PROCEDURE STOCKEE AVEC DES PARAMETRES EN ENTREE ET EN SORTIE

Créer la procédure stockée **prc_CA_Fournisseur**, qui pour un code fournisseur et une année entrés en paramètre, calcule et restitue le CA potentiel de ce fournisseur pour l'année souhaitée. (Exemple avec le fournisseur 9180 et l'année 2015)

```
SELECT SUM(priuni*qtecde) AS CA
FROM entcom
  INNER JOIN ligcom ON entcom.numcom = ligcom.numcom
WHERE numfou= 9180
  AND YEAR(datcom) = '2015';
```

Le chiffre d'affaires est renvoyé dans le paramètre de sortie. On n'exécutera la requête que si le code fournisseur est valide, c'est-à-dire s'il existe dans la table FOURNIS, sinon on renverra un code d'état égal à -100 dans le CA.

Testez la procédure stockée avec différentes valeurs de paramètres.

2.4 CREEZ LE MESSAGE D'ERREUR : « FOURNISSEUR INEXISTANT »

Créez une nouvelle procédure stockée **prc_CA_Fournisseur2** qui, lorsque le code **FOURNIS** n'existe pas, lève une exception et renvoie un code erreur égal à 9000.

3. LES FONCTIONS

3.1 FONCTION **fn_DateFormat()**

Pour bien comprendre l'utilité d'une fonction, créez, dans votre base de données MySQL Papyrus, la fonction **fn_DateFormat()** codée ci-dessous :

```
CREATE FUNCTION fn_dateFormat (pdate DATE, psep VARCHAR(10))
    RETURNS VARCHAR(50)
BEGIN
    RETURN CONCAT_WS(psep,
                    DAY(pdate), MONTH(pdate), YEAR(pdate));
END;
```

*Le paramètre **psep** donne le caractère de séparation dans la date.*

Utilisez-la dans la requête ci-dessous qui affiche les commandes d'un mois donné, de manière à afficher la date des commandes avec ce format de date (très français avec le séparateur / entre jour et mois et entre mois et année).

Requête SQL réalisée à adapter pour utiliser la fonction **fn_DateFormat() qui affiche les commande passées au mois de mars et avril :**

```
SELECT numcom, datcom
FROM entcom
WHERE MONTH(datcom) IN (3,4)
```

3.2 CREATION D'UNE FONCTION **fn_Satisfaction()**

Créer la fonction scalaire **fn_Satisfaction()**, qui, avec l'indice de satisfaction en entrée, affiche un niveau de satisfaction en clair pour un fournisseur :

- Indice = Null, 'sans commentaire'
- Indice = 1 ou 2, 'Mauvais'
- Indice = 3 ou 4, 'Passable'
- Indice = 5 ou 6, 'Moyen'
- Indice = 7 ou 8, 'Bon'
- Indice = 9 ou 10, 'Excellent'

Adaptez au besoin votre jeu d'essai.

Testez votre fonction, en affichant le niveau de satisfaction des fournisseurs lors d'une nouvelle requête renvoyant le numéro fournisseur, le nom fournisseur et la satisfaction en clair.

3.3 CREATION D'UNE FONCTION `fn_CA_Fournisseur()`

Créer la fonction `fn_CA_Fournisseur()`, qui en fonction d'un code fournisseur et d'une année entrés en paramètre, restituera le CA de ce fournisseur pour l'année souhaitée.

Vous pouvez vous appuyer sur cette requête :

« Calculer le chiffre d'affaires pour le fournisseur 120 pour l'année 2007 sachant que les prix indiqués sont HT et que le taux de TVA est 20,60% ».

Requête SQL réalisée à adapter pour utiliser la fonction `fn_CA_Fournisseur()` :

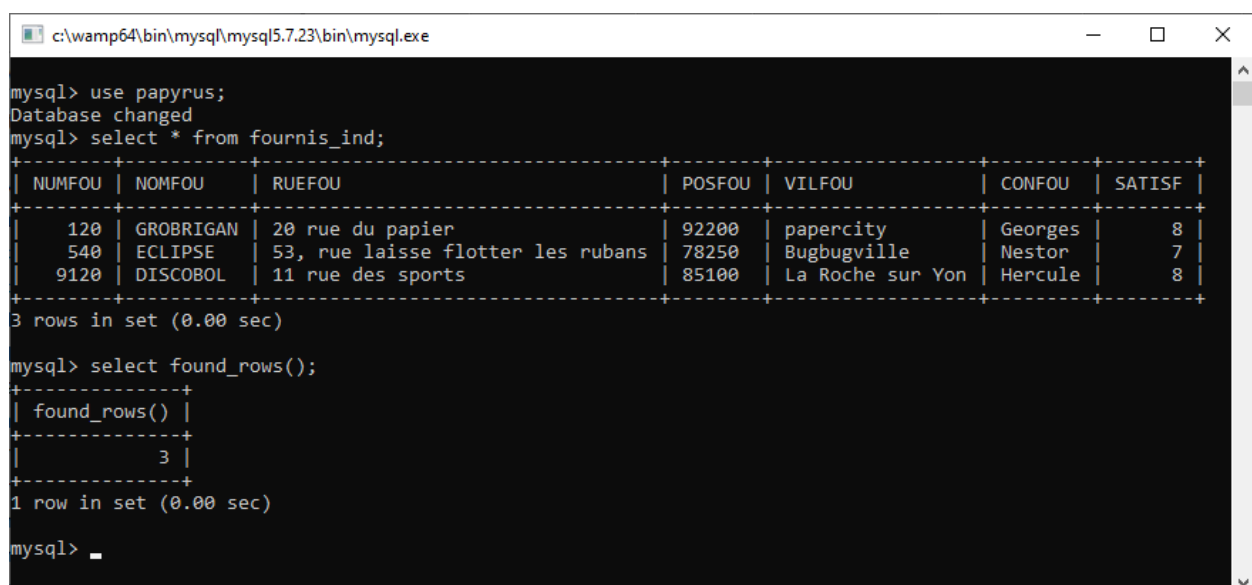
```
SELECT SUM(qtecde*priuni*1.2060) AS ca
FROM entcom
      INNER JOIN ligcom
            ON entcom.numcom = ligcom.numcom
WHERE YEAR(datcom) = 2007 AND entcom.numfou = 120;
```

3.4 TEST DE LA FONCTION `FOUND_ROWS()`

Créez une table `FOURNIS_IND` de structure identique à `FOURNIS` et insérez les lignes de la table `FOURNIS` dont la colonne indice de satisfaction est 'Bon'.

Créez la fonction `fn_Compte()` qui délivre le nombre de lignes de la table `FOURNIS_IND`.

Depuis la console MySQL, exécutez directement la requête qui affiche tous les enregistrements de la table `FOURNIS_IND` puis exécutez la fonction `FOUND_ROWS()`.



```
c:\wamp64\bin\mysql\mysql5.7.23\bin\mysql.exe

mysql> use papyrus;
Database changed
mysql> select * from fournis_ind;
+-----+-----+-----+-----+-----+-----+
| NUMFOU | NOMFOU | RUEFOU | POSFOU | VILFOU | CONFOU | SATISF |
+-----+-----+-----+-----+-----+-----+
| 120 | GROBRIGAN | 20 rue du papier | 92200 | papercity | Georges | 8 |
| 540 | ECLIPSE | 53, rue laisse flotter les rubans | 78250 | Bugbugville | Nestor | 7 |
| 9120 | DISCOBOL | 11 rue des sports | 85100 | La Roche sur Yon | Hercule | 8 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select found_rows();
+-----+
| found_rows() |
+-----+
| 3 |
+-----+
1 row in set (0.00 sec)

mysql> _
```


CREDITS

ŒUVRE COLLECTIVE DE L'AFPA

Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)

B. Hézard - Formateur

Ch. Perrachon – Ingénieure de formation

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »