

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

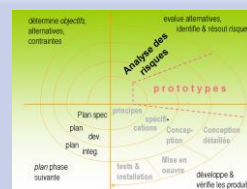
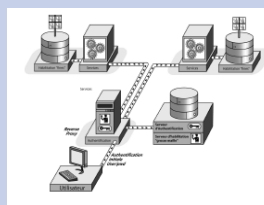


TABLE DES MATIERES

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

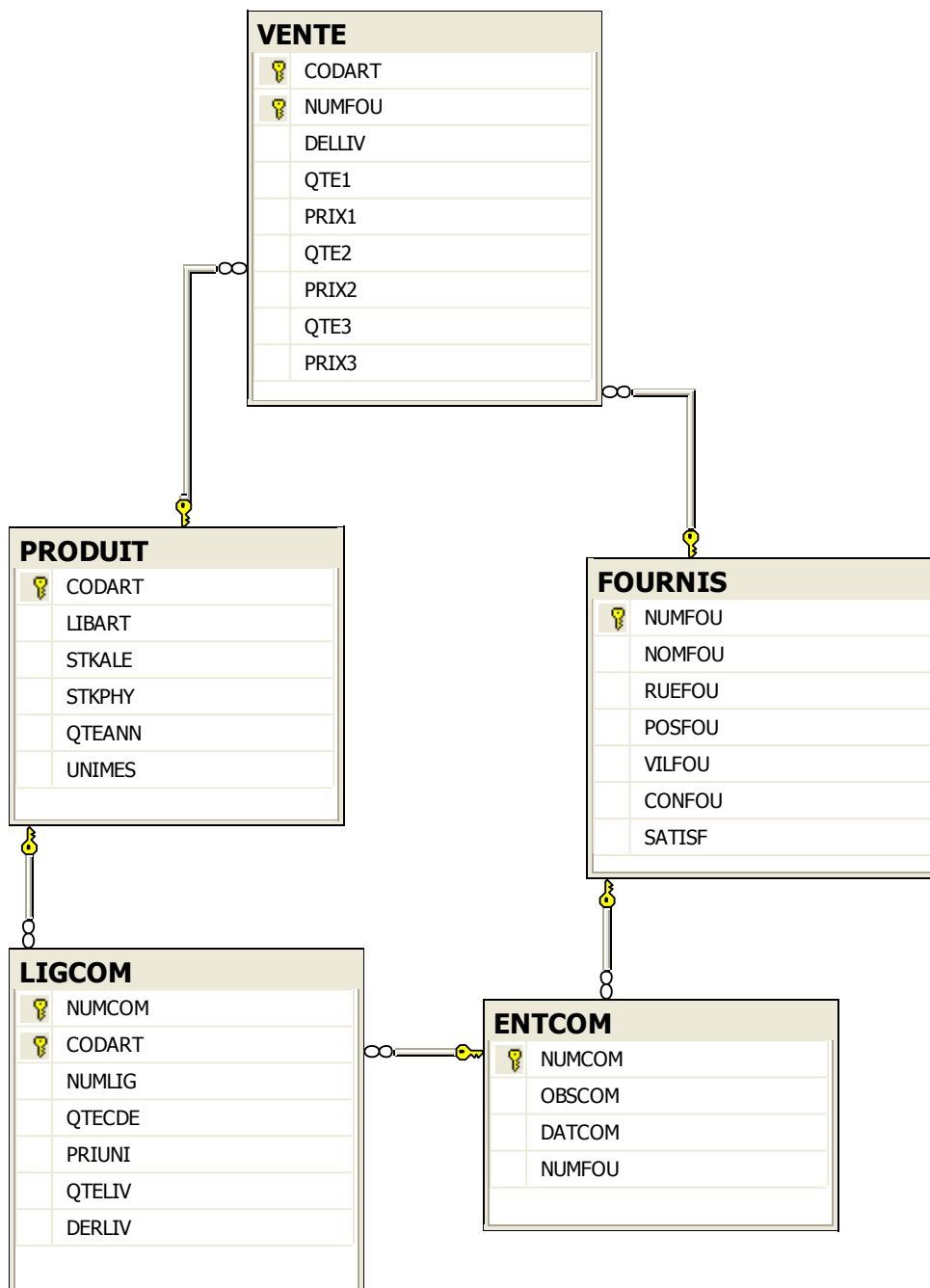
Mettre en pratique la programmation du SGBD MySQL tant au niveau des fonctions que des procédures stockées.

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, PR1X1, QTE2, PR1X2, QTE3, PR1X3)

Le jeu d'essai a été constitué lors d'une séance précédente.

Cas Papyrus MySQL - Programmer dans le langage du SGBD

Afpa © 2015 – Section Tertiaire Informatique – Filière « Etude et développement »

2. LES FONCTIONS

2.1 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 or REPLACE FUNCTION FN_DATEFORMAT (pdate DATE, psep
                                         VARCHAR)
RETURN VARCHAR2 AS
BEGIN
    RETURN EXTRACT (DAY FROM pdate) || psep || EXTRACT (MONTH
        FROM pdate) || psep || EXTRACT (YEAR FROM pdate);
END FN_DATEFORMAT;
```

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

Utilisez la ensuite 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 EXTRACT(MONTH FROM datcom) IN (3,4)
```

2.2 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.

Cas Papyrus MySQL - Programmer dans le langage du SGBD

Afpa © 2015 – Section Tertiaire Informatique – Filière « Etude et développement »

2.3 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.

Besoin : « Calculer le chiffre d'affaires par fournisseur pour l'année en cours sachant que les prix indiqués sont HT et que le taux de TVA est 20,00% ».

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

```
SELECT fournis.numfou, nomfou, SUM(qtecde * priuni * 1,20) AS CA
FROM fournis
  INNER JOIN entcom
ON fournis.numfou = entcom.numfou
  INNER JOIN ligcom
ON entcom.numcom = ligcom.numcom
WHERE EXTRACT(YEAR FROM datcom) = EXTRACT(YEAR FROM SYSDATE)
GROUP BY fournis.numfou, nomfou
```

2.4 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()`.

3. LES PROCEDURES STOCKEES

3.1 CRÉATION D'UNE PROCÉDURE STOCKÉE SANS PARAMÈTRE

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.

3.2 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' || '%';
```

3.3 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 EXTRACT(YEAR FROM 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.

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

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

Cas Papyrus MySQL - Programmer dans le langage du SGBD

Afpa © 2015 – Section Tertiaire Informatique – Filière « Etude et développement »

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

Date de mise à jour : 12/10/15

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. »

Cas Papyrus MySQL - Programmer dans le langage du SGBD

Afpa © 2015 – Section Tertiaire Informatique – Filière « Etude et développement »