

Secteur Tertiaire Informatique
Filière « Etude et développement »

Séquence « Développer des composants d'accès
aux données »

**Solutions Cas PAPYRUS MySQL :
Programmer des fonctions et des procédures**

Apprentissage

Mise en pratique

Evaluation

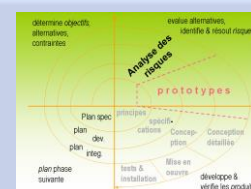
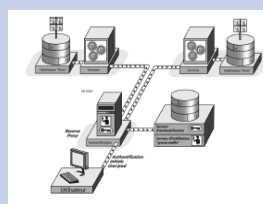


TABLE DES MATIERES

Table des matières	3
1. fonction fn_DateFormat()	5
1.1 Création de la fonction	5
1.2 Utilisation dans la requête	5
2. fonction fn_Satisfaction().....	5
2.1 Création de la fonction	5
2.2 Utilisation dans la requête	6
3. fonction fn_CA_Fournisseur()	6
3.1 Création de la fonction	6
3.2 Utilisation	6
4. Test de LA FONCTION FOUND_ROWS()	7
4.1 Création de la table Fournis_ind.....	7
4.2 Création de la fonction fn_Compte().....	8
4.3 Utilisation de la fonction COUNT_ROWS() directement	8
5. procédure prc_LST_Fournis sans paramètre	9
5.1 Création de la procédure avec une boucle LOOP	9
5.2 Utilisation dans la requête	9
6. procédure prc_LST_Commande_1 avec un paramètre en entrée....	10
6.1 Création de la procédure.....	10
6.2 Utilisation dans la requête	10
7. procédure prc_CA_Fournisseur avec des paramètres en entrée et en sortie.....	11
7.1 Création de la procédure sans message d'erreur	11
7.2 Utilisation dans la requête	11
7.3 Avec un message d'erreur	12
7.4 Utilisation dans la requête	12

Préambule

Ce document livre des solutions types aux besoins de développement de fonctions et de procédures stockées pour le cas Papyrus.

Comme il peut exister plusieurs bonnes solutions à un besoin, nous vous conseillons de confronter vos solutions à ce corrigé et à celles des autres stagiaires.

Objectifs

Méthodologie

1. FONCTION FN_DATEFORMAT()

1.1 CREATION DE LA FONCTION

```
DELIMITER //
```

```
CREATE FUNCTION `fn_DATEFORMAT`(`pdate` DATE, `psep` VARCHAR(5))  
RETURNS varchar(20)  
BEGIN  
    RETURN CONCAT(EXTRACT(DAY FROM pdate), psep, EXTRACT(MONTH FROM pdate),  
                  psep, EXTRACT(YEAR FROM pdate) );  
END//
```

```
DELIMITER ;
```

1.2 UTILISATION DANS LA REQUETE

« Quelles sont les commandes passées au mois de mars et avril ? »

(Affichage numéro de commande, date de commande)

```
SELECT numcom, fn_DateFormat(datcom, '/') AS "Date Commande"  
FROM entcom  
WHERE EXTRACT(MONTH FROM datcom) IN (3,4);
```

2. FONCTION FN_SATISFACTION()

2.1 CREATION DE LA FONCTION

```
DELIMITER //
```

```
CREATE FUNCTION `fn_Satisfaction`(`indice` INT)  
RETURNS varchar(20)  
BEGIN  
    RETURN CASE  
        WHEN indice IS NULL THEN 'Sans commentaire'  
        WHEN indice BETWEEN 1 AND 2 THEN 'Mauvais'  
        WHEN indice BETWEEN 3 AND 4 THEN 'Passable'  
        WHEN indice BETWEEN 5 AND 6 THEN 'Moyen'  
        WHEN indice BETWEEN 7 AND 8 THEN 'Bon'  
        WHEN indice >= 9 THEN 'Excellent'  
    END;  
END//
```

```
DELIMITER ;
```

2.2 UTILISATION DANS LA REQUETE

Affichage de l'indice de satisfaction des fournisseurs.

(On rajoute la sélection et on ne renvoie qu'une colonne).

```
SELECT nomfou, fn_Satisfaction(satisf)
FROM fournis;
```

3. FONCTION FN_CA_FOURNISSEUR()

3.1 CREATION DE LA FONCTION

```
DELIMITER //

CREATE FUNCTION `fn_CA_Fournisseur`(`v_numfou` DECIMAL(10,0), `annee` INT)
RETURNS DECIMAL(10,2)
BEGIN
    DECLARE ca DECIMAL(10,2);

    SELECT SUM(qtecde*priuni*1.2060) INTO ca
    FROM entcom
        INNER JOIN ligcom
            ON entcom.numcom = ligcom.numcom
    WHERE EXTRACT(YEAR FROM datcom) = annee AND entcom.numfou = v_numfou;

    RETURN ca;
END//

DELIMITER ;
```

3.2 UTILISATION

```
SELECT fn_ca_fournisseur(120, 2014) FROM DUAL;
```

4. TEST DE LA FONCTION FOUND_ROWS()

4.1 CREATION DE LA TABLE FOURNIS_IND

```
CREATE TABLE fournis_ind (  
    NUMFOU DECIMAL(10,0) NOT NULL,  
    NOMFOU VARCHAR(30) NOT NULL,  
    RUEFOU VARCHAR(30) NOT NULL,  
    POSFOU VARCHAR(5) NOT NULL,  
    VILFOU VARCHAR(30) NOT NULL,  
    CONFOU VARCHAR(15) NOT NULL,  
    SATISF DECIMAL(3,0) DEFAULT NULL,  
    PRIMARY KEY (NUMFOU))  
ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;  
  
--Astuce pour dupliquer une table et ses données :  
CREATE TABLE fournis_ind AS SELECT * FROM fournis;  
  
--Dupliquer uniquement la structure :  
CREATE TABLE fournis_ind AS SELECT * FROM fournis WHERE 1=2;  
  
--Dupliquer la table avec uniquement les données souhaitées :  
CREATE TABLE fournis_ind AS  
    SELECT * FROM fournis  
    WHERE UPPER(fn_Satisfaction(satisf)) = 'BON';  
  
--Attention ces deux astuces ne dupliquent pas toutes les contraintes ex: les  
'PRIMARY KEY'  
  
INSERT INTO fournis_ind  
    SELECT * FROM fournis  
    WHERE UPPER(fn_Satisfaction(satisf)) = 'BON';
```

4.2 CREATION DE LA FONCTION FN_COMPTE()

```
DELIMITER //
```

```
CREATE FUNCTION `fn_Compte`()
  RETURNS INT
BEGIN
  DECLARE iNB INT;
  SELECT COUNT(*) INTO iNB FROM fournis_ind;
  RETURN iNB;
END//
```

```
DELIMITER ;
```

```
-- Utilisation de la fonction fn_Compte
SELECT fn_Compte() AS "NB_Bons_Fournisseurs" FROM DUAL;
```

4.3 UTILISATION DE LA FONCTION COUNT_ROWS() DIRECTEMENT

Depuis la console de MySQL exécutez les 2 instructions suivantes :

```
SELECT * FROM fournis_ind;
SELECT FOUND_ROWS();
```

Remarque : La fonction FOUND_ROWS() ne peut pas être utilisée dans un SELECT !

Cette fonction est utile notamment avec le langage PHP car on peut récupérer sa valeur directement après avoir lancé et exécuté une requête via PHP.

5. PROCEDURE PRC_LST_FOURNIS SANS PARAMETRE

5.1 CREATION DE LA PROCEDURE AVEC UNE BOUCLE LOOP

```
DELIMITER //  
  
CREATE PROCEDURE prc_LST_FOURNIS()  
  
BEGIN  
    DECLARE vnumfou INT;  
    DECLARE done INT DEFAULT 0;  
    DECLARE c_numfou CURSOR FOR  
        SELECT DISTINCT numfou FROM entcom;  
  
    DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;  
  
    OPEN c_numfou;  
    FETCH c_numfou INTO vnumfou;  
    WHILE done = 0 DO  
        SELECT vnumfou;  
        FETCH c_numfou INTO vnumfou;  
    END WHILE;  
END//  
  
DELIMITER ;
```

5.2 UTILISATION DANS LA REQUETE

```
Call prc_Lst_fournis();
```

6. PROCEDURE PRC_LST_COMMANDE_1 AVEC UN PARAMETRE EN ENTREE

6.1 CREATION DE LA PROCEDURE

```
DELIMITER //
```

```
CREATE PROCEDURE prc_LST_Commande_1 (IN libelle VARCHAR(50))  
BEGIN  
    DECLARE var INT DEFAULT 1;  
    DECLARE done INT DEFAULT 0;  
    DECLARE l_numcom decimal;  
    DECLARE f_nomfou varchar(50);  
    DECLARE p_libart varchar(50);  
    DECLARE c_commande cursor for  
        SELECT distinct 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 CONCAT('%',libelle,'%');  
    DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;  
  
    OPEN c_commande;  
    FETCH c_commande INTO l_numcom,f_nomfou,p_libart;  
    WHILE done = 0 DO  
        select l_numcom,f_nomfou,p_libart;  
        fetch c_commande into l_numcom,f_nomfou,p_libart;  
    END WHILE;  
END//  
  
DELIMITER ;
```

6.2 UTILISATION DANS LA REQUETE

```
CALL `prc_Lst_Commande_1`('urgent');
```

7. PROCEDURE PRC_CA_FOURNISSEUR AVEC DES PARAMETRES EN ENTREE ET EN SORTIE

7.1 CREATION DE LA PROCEDURE SANS MESSAGE D'ERREUR

```
DELIMITER //
```

```
CREATE PROCEDURE prc_CA_Fournisseur (  
    IN iNumfou DECIMAL(10,0),  
    IN iAnnee INT,  
    OUT iCA FLOAT)  
BEGIN  
    DECLARE nbFou INT;  
    SELECT COUNT(*) INTO nbFou FROM Fournis WHERE numfou = iNumfou;  
    IF (nbFou = 1) THEN  
        SELECT SUM(qtecde * priuni *1.2060) INTO iCA  
        FROM entcom  
            INNER JOIN ligcom  
                ON entcom.numcom = ligcom.numcom  
        WHERE EXTRACT(YEAR FROM datcom)= iAnnee  
            AND numfou = iNumfou;  
    ELSE  
        SET iCA = -100;  
    END IF;  
END//
```

```
DELIMITER ;
```

7.2 UTILISATION DANS LA REQUETE

```
CALL prc_CA_Fournisseur(120,2014,@ca); --Fournisseur existant  
SELECT CONCAT('CA:', @ca);  
CALL prc_CA_Fournisseur(121,2014,@ca); --Fournisseur inconnu  
SELECT CONCAT('CA:', @ca);
```

7.3 AVEC UN MESSAGE D'ERREUR

```
DELIMITER //
```

```
CREATE PROCEDURE prc_CA_Fournisseur2 (  
    IN iNumfou DECIMAL(10,0),  
    IN iAnnee INT,  
    OUT iCA FLOAT)  
BEGIN  
    DECLARE nbFou INT;  
    DECLARE fournisseur_absent CONDITION FOR SQLSTATE '45001';  
    SELECT COUNT(*) INTO nbFou FROM Fournis WHERE numfou = iNumfou;  
    IF (nbFou = 1) THEN  
        SELECT SUM(qtecde * priuni *1.2060) INTO iCA  
        FROM entcom  
            INNER JOIN ligcom  
                ON entcom.numcom = ligcom.numcom  
        WHERE EXTRACT(YEAR FROM datcom)= iAnnee  
            AND numfou = iNumfou;  
    ELSE  
        SIGNAL fournisseur_absent  
        SET MESSAGE_TEXT = 'Fournisseur absent',  
            MYSQL_ERRNO = 9000;  
    END IF;  
END//
```

```
DELIMITER ;
```

7.4 UTILISATION DANS LA REQUETE

```
CALL prc_CA_Fournisseur(120,2014,@ca); --Fournisseur existant  
SELECT CONCAT('CA:', @ca);  
CALL prc_CA_Fournisseur(121,2014,@ca); --Fournisseur inconnu  
SELECT CONCAT('CA:', @ca);
```

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)

D. Bin - Formateur

B. Hézard – Formateur

E. Cattaneo - Formateur

Ch. Perrachon – Ingénieure de formation

Date de mise à jour : 7/10/15

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. »

Solutions Cas Papyrus MySQL – Programmer

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