

TP3 Base de Données

GOURGAUD Kévin / PUP Hugo

L2 Groupe 2

25/11/2019

Partie II

1)

```
/*Question 1*/  
  
DROP FUNCTION IF EXISTS `INSERT_PAYS`//  
  
CREATE FUNCTION INSERT_PAYS (nom VARCHAR(32), abreviation VARCHAR(4), capitale VARCHAR(35), province VARCHAR(32), superficie INT, population INT, independance DATE, regime VARCHAR(120))  
RETURNS INT  
BEGIN  
DECLARE EXIT HANDLER FOR SQLSTATE '23000'  
  
BEGIN  
RETURN -1;  
END ;  
  
INSERT INTO country(Name,Code,Capital,Province,Area,Population) VALUES (nom,abreviation,capitale,province,surface,population);  
INSERT INTO politics(Country,Independence,Government) VALUES (abreviation,independance,regime);  
RETURN 1;  
END //
```

```
DELIMITER ;  
  
DELETE FROM country WHERE Name = 'Hugoslavia';  
DELETE FROM politics WHERE Country = 'HP';  
SELECT INSERT_PAYS('Hugoslavia','HP','Issou','Toutencarton',547030,60000000,'1997-12-27','dictature');  
SELECT * FROM country WHERE Name = 'Hugoslavia';  
SELECT * FROM politics WHERE Country = 'HP';
```

Ici, nous créons la fonction INSERT_PAYS qui prend en paramètre le nom d'un pays, son abréviation, sa capitale, le nom de sa province, sa superficie, sa population, sa date d'indépendance (s'il y en a une) et son régime politique. S'il le pays qu'on souhaite rajouter est déjà dans notre table, la fonction fait une erreur, quitte et renvoie -1. Sinon il ajoute le pays dans la table country et la date d'indépendance et le régime politique du pays dans la table politics et renvoie 1 pour montrer que cela a été fait.

Nous avons testé notre fonction avec un exemple de pays (imaginaire), voici le résultat de la fonction sur notre exemple :

```
+-----+
| INSERT_PAYS('Hugoslavia','HP','Issou','Toutencarton',547030,60000000,'1997-12-27','dictature') |
+-----+
|                                                                                               1 |
+-----+
1 row in set (0,01 sec)

+-----+
| Name      | Code | Capital | Province      | Area   | Population | OldCap | ChangeDate |
+-----+
| Hugoslavia | HP   | Issou   | Toutencarton  | 547030 | 60000000  | NULL   | NULL        |
+-----+
1 row in set (0,00 sec)

+-----+
| Country | Independence | Government |
+-----+
| HP      | 1997-12-27   | dictature  |
+-----+
1 row in set (0,00 sec)
```

2)

```
CREATE TABLE LISTCAP (  
Pays VARCHAR(32) PRIMARY KEY,  
Capitale VARCHAR (35)  
);  
  
DROP PROCEDURE IF EXISTS `PAYS_CAP`;  
  
DELIMITER //  
  
/*Procédure List_cap*/  
CREATE PROCEDURE PAYS_CAP (IN code VARCHAR(4), OUT erreur INT)  
BEGIN  
DECLARE var_pays VARCHAR(32);  
DECLARE var_cap VARCHAR(35);  
DECLARE EXIT HANDLER FOR SQLSTATE '23000'  
BEGIN  
SET erreur = -1;  
END;  
SELECT Name,Capital INTO var_pays, var_cap FROM country WHERE Code = code;  
INSERT INTO LISTCAP VALUES(var_pays,var_cap);  
SET erreur = 1;  
END //  
  
DELIMITER ;  
  
/*Procédure Charge_cap*/  
DROP PROCEDURE IF EXISTS `CHARGE_CAP`;  
  
DELIMITER //  
  
CREATE PROCEDURE CHARGE_CAP()  
BEGIN  
DECLARE boucle INTEGER DEFAULT 0;  
DECLARE code_pays VARCHAR(4);  
DECLARE erreur INTEGER DEFAULT 1;  
DECLARE curseur CURSOR FOR SELECT Code FROM country;  
DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET boucle = 1;  
OPEN curseur;  
WHILE boucle = 0 DO  
FETCH FROM curseur INTO code_pays;  
CALL PAYS_CAP(code_pays,erreur);  
END WHILE;  
CLOSE curseur;  
END //
```

Pour les procédure PAYS_CAP et CHARGE_CAP, nous avons eu besoin de créer la table LISTCAP contenant des pays et leurs capitales respectives. Du côté de la procédure PAYS_CAP, avec l'abréviation passé en paramètre, la procédure cherche le nom et la capitale du pays correspondant dans la table country et l'insert dans la table LISTCAP. Si le pays est déjà présent, la procédure renvoie une erreur valant -1 sinon elle renvoie 1. La procédure CHARGE_CAP va quant à elle déclarer un curseur qui va parcourir la totalité de la table country et appeler la procédure PAYS_CAP en utilisant le Code en paramètre pour chaque pays de notre table.

```
ERROR 1172 (42000): Result consisted of more than one row
Empty set (0,00 sec)
```

3)

```
/*Question3*/

ALTER TABLE country DROP COLUMN OldCap;
ALTER TABLE country DROP COLUMN ChangeDate;
ALTER TABLE country ADD COLUMN OldCap VARCHAR(20), ADD COLUMN ChangeDate DATE;

DROP TRIGGER IF EXISTS `changepap`;

DELIMITER //

CREATE TRIGGER changepap BEFORE UPDATE
ON country FOR EACH ROW
BEGIN
IF OLD.Capital <> NEW.Capital
THEN
SET NEW.ChangeDate = NOW();
SET NEW.OldCap = OLD.Capital;
END IF;
END //

DELIMITER ;
SELECT * FROM country WHERE Code = 'HP';
UPDATE country SET Capital = 'MontMartre' WHERE Code = 'HP';
SELECT * FROM country WHERE Code = 'HP';
```

On commence par modifier la table country en ajoutant les colonnes Oldcap et ChangeDate. On utilise ensuite un déclencheur (TRIGGER) qui détecte une mise à jour de la table country pour chaque tuples. Ce déclencheur met à jour la colonne OldCap avec l'ancienne capitales et met à jour la date auquelle à eu lieu le changement.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| Name      | Code | Capital | Province | Area  | Population | OldCap | ChangeDate |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Hugoslovakia | HP   | Issou   | Toutencarton | 547030 | 600000000 | NULL   | NULL       |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0,01 sec)

Query OK, 1 row affected (0,00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

+-----+-----+-----+-----+-----+-----+-----+-----+
| Name      | Code | Capital   | Province | Area  | Population | OldCap | ChangeDate |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Hugoslovakia | HP   | MontMartre | Toutencarton | 547030 | 600000000 | Issou   | 2019-11-26 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

4)

```
/*Question 4*/  
  
DROP PROCEDURE IF EXISTS `AJOUT_POPULA`;  
  
DELIMITER //  
  
CREATE PROCEDURE AJOUT_POPULA (IN X INT, IN nom_pays VARCHAR(32))  
BEGIN  
UPDATE country SET Population = Population + (Population *X/100) WHERE country.Name = nom_pays;  
END //  
  
DELIMITER ;  
  
SELECT * FROM country WHERE Code='HP';  
CALL AJOUT_POPULA(6,'Hugoslavia');  
SELECT * FROM country WHERE Code='HP';
```

Nous avons écrit cette procédure afin d'ajouter un pourcentage X de la population à la population d'un pays, ce pourcentage est passé en paramètre tout comme le nom du pays concerné. Nous avons repris notre exemple de pays (imaginaire) et avons ajouté 6% de la population à la population totale précédente. Voici le résultat :

Name	Code	Capital	Province	Area	Population	OldCap	ChangeDate
Hugoslavia	HP	Issou	Toutencarton	547030	63600000	NULL	NULL

5)

```
DROP PROCEDURE IF EXISTS `SOMME_POP_CONTINENT`;

DELIMITER //

CREATE PROCEDURE SOMME_POP_CONTINENT()
BEGIN
SELECT Continent, SUM(Population) AS Population_continent
FROM country, encompasses
WHERE country.Code = encompasses.Country
GROUP BY Continent;
END //

DELIMITER ;
CALL SOMME_POP_CONTINENT();
```

Cette procédure, à la manière d'une requête, permet d'effectuer la somme des populations des pays par continent. Ci-après, le résultat obtenu :

5. Ecrire une procédure qui fait

résultats.

Continent	Population_continent
Africa	730937485
America	778769518
Asia	3701335697
Australia/Oceania	28386834
Europe	791702262

¹ Rappel : MySQL 5.0 ne peut pas faire de jointures dans une procédure.

6)

7)

```
/*Question 7*/
DROP PROCEDURE IF EXISTS `Long_Frontieres`;

DELIMITER //

CREATE PROCEDURE Long_Frontieres(IN continent VARCHAR(20))
BEGIN
SELECT Continent,country.Name AS Nom_Pays
FROM country,encompasses
WHERE country.Code = encompasses.Country
AND encompasses.Continent = continent;
END//

DELIMITER ;

CALL Long_Frontieres('America');
CALL Long_Frontieres('Europe');
CALL Long_Frontieres('Asia');
```

Cette

procédure permet de faire la somme des frontières des pays d'un continent passé en paramètres.

Continent	Nom_Pays
America	Antigua and Barbuda
America	Barbados
America	Bolivia
America	Brazil
America	Bahamas
America	Belize
America	Cuba
America	Canada
America	Colombia
America	Costa Rica
America	Dominican Republic
America	Ecuador
America	El Salvador
America	French Guiana
America	Guatemala
America	Guyana
America	Honduras
America	Jamaica
America	Saint Kitts and Nevis
America	Mexico
America	Nicaragua
America	Panama
America	Peru
America	Paraguay
America	Argentina
America	Chile
America	Haiti
America	Uruguay
America	Suriname
America	Trinidad and Tobago
America	United States
America	Dominica
America	Grenada
America	Saint Lucia
America	Saint Vincent and the Grenadines
America	Venezuela

Nous obtenons seulement la liste des pays du continent appelé. Notre procédure ne fonctionne pas.