

### Série d'exercices du chapitre 1- Solution

#### Exercice 3 :

Soit la relation suivante :

COMPTECLIENT(numCli, nomCli, nbComptes)

1. Ecrire le trigger qui empêche un client d'avoir plus de 10 comptes en banque

```
CREATE TRIGGER verifieNbComptes
BEFORE INSERT ON COMPTECLIENT
FOR EACH ROW
DECLARE
    nbComptes int;
BEGIN
    SELECT nbComptes INTO nbComptes
    FROM COMPTECLIENT
    WHERE numCli = :new.numCli;

    IF (nbComptes > 10) THEN
        RAISE_APPLICATION_ERROR(-20556 'Ce client a deja trop de comptes');
    END IF;
END;
```

#### Exercice5 : écriture de triggers

1. Ecrire un trigger qui permet de Contrôler les salaires : ils doivent être plus grand que le plus petit des salaires et plus petit que le plus grand des salaire

```
CREATE TRIGGER verifieSalaire
BEFORE INSERT OR UPDATE ON employer
FOR EACH ROW
DECLARE
    minSal int;
    maxSal int;
BEGIN
    SELECT MIN(salaire), MAX(salaire) INTO minsal, maxsal
    FROM employer

    IF (new.salaire < minSal OR new.salaie > maxSal) THEN
        RAISE_APPLICATION_ERROR(-20556 'Ce client a deja trop de comptes');
    END IF;
END;
```

2. Écrire un trigger qui permet de mettre le salaire à 5000\$ si un employé n'a pas de salaire.

**Employee(Name, Salary, Department)**

```
CREATE TRIGGER deffaultSalary
BEFORE INSERT OR UPDATE ON employee
FOR EACH ROW
BEGIN
    IF (new.salary == null)
        new.salary = 5000
    END IF;
END;
```

3. Soit une table cumul sert à enregistrer le cumul des augmentations dont ont bénéficié les employés d'une entreprise.
  - a. Ecrire le trigger qui fait la mise à jour automatique d'une table cumul qui totalise les augmentations de salaire de chaque employé.

```
CREATE TRIGGER totalAugmentation
AFTER UPDATE ON employee
FOR EACH ROW
BEGIN
    UPDATE cumul
    Set augmentation = augmentation + (new.salaire - old.salaire)
    Where matricule = old.matricule
END;
```

- b. Ecrire un trigger qui ajoute une ligne dans la table cumul quand un employé est créé

```
CREATE TRIGGER totalAugmentation
AFTER INSERT ON employee
FOR EACH ROW
BEGIN
    INSERT INTO cumul (matricula, augmentation)
    VALUES (new.matricule, 0)
END;
```