

1. Buatlah stored procedure/function totalSalary untuk menghitung total gaji yang diterima oleh employee jika total gajinya sama dengan akumulasi gaji awal ditambah dengan tambahan dana \$100 per dependent yang dimiliki oleh employee. Stored procedure/function ini menerima 3 buah argumen, yaitu nama employee yang terbagi menjadi fname, minit, lname. Contoh: pemanggilan **totalSalary('Franklin', 'T', 'Wong')** akan mengembalikan nilai 40300 karena Franklin T Wong memiliki 3 dependent dengan salary awal sebesar \$40,000.00. Maka salary yang diterima Franklin T Wong =  $\$40,000.00 + (3 * \$100) = \$40,300.00$ . Setelah stored procedure dibuat, lakukan pengujian dengan memanggil: **SELECT totalSalary('Franklin', 'T', 'Wong')**

```
CREATE OR REPLACE FUNCTION totalSalary(FnameArg
VARCHAR(15), MinitArg VARCHAR(1), LNameArg VARCHAR(15))
RETURNS INTEGER AS
$$
DECLARE
    salaryTotal INTEGER;
    currentSalary INTEGER;
    totalDependent INTEGER;
    employeeSsn VARCHAR(9);
BEGIN
    SELECT E.Salary into currentSalary
    FROM EMPLOYEE E
    WHERE E.Fname = FnameArg AND E.Minit = MinitArg AND
    E.LName = LNameArg;

    SELECT E.Ssn into employeeSsn
    FROM EMPLOYEE E
    WHERE E.Fname = Fnamearg AND E.Minit = Minitarg AND
    E.LName = LNamearg;

    SELECT COUNT(*) into totalDependent
    FROM DEPENDENT D
    WHERE D.Essn = employeeSsn;

    salaryTotal := currentSalary + 100*totalDependent;
```

```
        return salaryTotal;
END;
$$
```

```
LANGUAGE plpgsql;
```

2. Berdasarkan peraturan yang dibuat oleh company, tanggungan (dependent) yang dimiliki oleh employee tidak boleh lebih dari 3 orang. Oleh karena itu, buatlah stored procedure/function dan trigger **checkTotalDependent** untuk memastikan jumlah tanggungan yang dimiliki oleh employee tidak lebih dari 3 dependent. Perhatikan event apa saja yang perlu mengaktifkan trigger yang Anda buat. Setelah stored procedure & trigger dibuat, lakukan pengujian untuk perintah SQL berikut ini: **INSERT INTO DEPENDENT VALUES ('123456789', 'Nate', NULL, NULL, 'Son');**

Stored Procedure/Function:

```
CREATE OR REPLACE FUNCTION checkTotalDependent()
RETURNS TRIGGER AS
$$
DECLARE
    totalDependent INTEGER;
BEGIN
    SELECT COUNT(*) INTO totalDependent
    FROM DEPENDENT
    WHERE DEPENDENT.Essn = NEW.Essn;

    if(totalDependent >= 3) THEN
        RAISE EXCEPTION 'DEPENDENT TIDAK BOLEH LEBIH DARI 3
ORANG';

    END IF;

    RETURN NEW;
END;

$$
language plpgsql;
```

Trigger:

```
CREATE TRIGGER triggerCheckTotalDependent
BEFORE INSERT ON DEPENDENT
FOR EACH ROW EXECUTE PROCEDURE checkTotalDependent();
```

3. Buatlah stored procedure/function checkSalary dan trigger salaryViolation untuk memastikan bahwa gaji yang diterima oleh employee tidak lebih besar dari gaji yang diterima manajer departemen. Perhatikan event apa saja yang perlu untuk mengaktifkan trigger yang Anda buat. Setelah stored procedure & trigger dibuat, lakukan pengetesan untuk perintah SQL berikut ini: **UPDATE employee SET salary = 45000 WHERE ssn = '123456789';**

Stored procedure/function:

```
CREATE OR REPLACE FUNCTION checkSalary()
RETURNS TRIGGER AS
$$
DECLARE
    employeeSalary INTEGER;
    managerSsn Varchar(9);
    managerSalary INTEGER;
BEGIN
    SELECT E.Super_ssn into managerSsn
    FROM EMPLOYEE E
    WHERE E.Ssn = New.Ssn;

    SELECT E.Salary into managerSalary
    FROM EMPLOYEE E
    WHERE E.Ssn = managerSsn;

    if(NEW.Salary > managerSalary) THEN
        RAISE EXCEPTION 'Gaji employee tidak dapat lebih dari
gaji manager';
    END IF;

    RETURN NEW;

END;
```

\$\$

```
language plpgsql;
```

Trigger:

```
CREATE TRIGGER triggerCheckSalary  
BEFORE INSERT OR UPDATE ON EMPLOYEE  
FOR EACH ROW EXECUTE PROCEDURE checkSalary();
```