SET C

Person - Area Database

Person (pno int, name varchar (20), birthdate date, income money)

Area (aid int, aname varchar (20), area_type varchar (5))

The person and area related to many to one relationship. The attribute ‘area_type’ can have values either ‘urban’ or ‘rural’.

Create the above database in PostGreSQL and insert sufficient records.

a. Write a cursor to accept a month as an input parameter from the user and display the names of persons whose birthday falls in that particular month.

b. Write a cursor to display the names of persons living in urban area.

c. Write a cursor to print names of all persons having income between50000 and 100000.

ANSWER:

```
CREATE TABLE Area (
    aid INT PRIMARY KEY,
    aname VARCHAR(20),
    area_type VARCHAR(5) CHECK (area_type IN ('urban', 'rural'))
);

-- Create Person Table
CREATE TABLE Person (
    pno INT PRIMARY KEY,
    name VARCHAR(20),
    birthdate DATE,
    income MONEY,
    aid INT,
    FOREIGN KEY (aid) REFERENCES Area(aid)
);
```

```
INSERT INTO Area VALUES(1, 'Downtown', 'urban');
INSERT INTO Area VALUES(2, 'Suburb', 'urban');
INSERT INTO Area VALUES(3, 'Ruralville', 'rural');
INSERT INTO Area VALUES(4, 'Countryside', 'rural');

select * from Area;

-- Insert records into Person table
INSERT INTO Person VALUES(1, 'John Doe', '1990-03-15', '55000', 1);
INSERT INTO Person VALUES(2, 'Jane Smith', '1985-06-25', '75000', 2);
INSERT INTO Person VALUES(3, 'Alice Brown', '2000-11-10', '62000', 3);
INSERT INTO Person VALUES(4, 'Bob Johnson', '1992-03-30', '80000', 1);
INSERT INTO Person VALUES(5, 'Charlie White', '1995-05-20', '95000', 4);
INSERT INTO Person VALUES(6, 'Eve Black', '1998-07-15', '120000', 2);
```

Q a. Write a cursor to accept a month as an input parameter from the user and display the names of persons whose birthday falls in that particular month.

```
CREATE OR REPLACE FUNCTION get_persons_by_month(month int) RETURNS VOID AS '
DECLARE
    person_name VARCHAR(20);
    person_cursor CURSOR FOR SELECT p.name
        FROM Person p
        WHERE EXTRACT(MONTH FROM p.birthdate) = month;
  BEGIN
    OPEN person_cursor;
    LOOP
        FETCH person_cursor INTO person_name;
```

```
        EXIT WHEN NOT FOUND;
        RAISE NOTICE ' 'Person: %' ', person_name;
    END LOOP;
    CLOSE person_cursor;
END;
' LANGUAGE plpgsql;
```

Call Function:

```
SELECT get_persons_by_month(3);
```

Q b. Write a cursor to display the names of persons living in urban area

```
CREATE OR REPLACE FUNCTION get_persons_in_urban_area() RETURNS VOID AS '
DECLARE
    person_name VARCHAR(20);
    urban_person_cursor CURSOR FOR SELECT p.name
        FROM Person p,Area a WHERE a.area_type = "urban" AND p.aid = a.aid ;
BEGIN
    OPEN urban_person_cursor;
    LOOP
        FETCH urban_person_cursor INTO person_name;
        EXIT WHEN NOT FOUND;
        RAISE NOTICE  ' 'Person: % ' ', person_name;
    END LOOP;
    CLOSE urban_person_cursor;
END;
' LANGUAGE plpgsql;
```

Call Function:

```
SELECT get_persons_in_urban_area();
```

Q c. Write a cursor to print names of all persons having income between50000 and 100000.

```
CREATE OR REPLACE FUNCTION get_persons_by_income_range()RETURNS VOID AS '
DECLARE
   person_name VARCHAR(20);
   income_cursor CURSOR FOR SELECT p.name FROM Person p
      WHERE p.income BETWEEN  "50000"  AND "100000";

BEGIN
   OPEN income_cursor;
   LOOP
      FETCH income_cursor INTO person_name;
      EXIT WHEN NOT FOUND;
      RAISE NOTICE ' 'Person: %' ' , person_name;
   END LOOP;
   CLOSE income_cursor;
END;
' LANGUAGE plpgsql;
```

Call Function:

```
SELECT get_persons_by_income_range();
```