Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ» Кафедра информатики

Отчет по лабораторной работе №6

Построение рекурсивных SQL запросов

Выполнил: студент гр. 953501 Кременевский Владислав Серегеевич

Руководитель: доцент Чащин С. В.

Цель работы

Получить практические навыки построения рекурсивных запросов.

Задача

- 1. Построить, заполнить таблицу ParentOf(parent next, child text) и найти
 - Предков Магу
 - Потомков Carol
- 2. Построить и заполнить таблицы Employee(ID int, salary int), Manager(mID int, eID int), Project(name text, mgrID int) и используя рекурсивный запрос найти совокупную зарплату на проекте.
- 3. Используя рекурсивный запрос, вывести первые 10 чисел Фибоначчи

Практическая часть

- 1. Построить, заполнить таблицу ParentOf(parent next, child text) и найти
 - Предков Магу
 - Потомков Carol

```
CREATE TABLE ParentOf(
     parent VARCHAR(20) NOT NULL,
     child VARCHAR(20) NOT NULL,
INSERT INTO ParentOf
     VALUES
          ('Alice', 'Carol'),
  ('Bob', 'Carol'),
  ('Carol', 'Dave'),
  ('Carol', 'George'),
  ('Dave', 'Mary'),
  ('Eve', 'Mary'),
  ('Mary', 'Frank');
WITH Rec(parent, child)
     AS (
          SELECT parent, child
          FROM ParentOf
          WHERE child = 'Mary'
          UNION ALL
          SELECT par.parent, par.child
          FROM ParentOf as par
          JOIN Rec as rec
          ON rec.parent = par.child
     )
— Предки Магу
SELECT parent FROM REC;
```

Результат предков Магу:

| 1 | Dave |
|---|-------|
| 2 | Eve |
| 3 | Carol |
| 4 | Bob |
| 5 | Alice |

Для Carol:

```
WITH Rec(parent, child)
AS (

SELECT parent, child
FROM ParentOf
WHERE parent = 'Carol'
UNION ALL
SELECT par.parent, par.child
FROM ParentOf as par
JOIN Rec as rec
ON rec.child = par.parent
)
```

Результат потомков Carol:

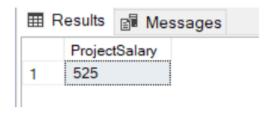
SELECT * FROM REC;

| 1 | Dave |
|---|--------|
| 2 | George |
| 3 | Mary |
| 4 | Frank |
| | |

2. Построить и заполнить таблицы Employee(ID int, salary int), Manager(mID int, eID int), Project(name text, mgrID int) и используя рекурсивный запрос найти совокупную зарплату на проекте.

```
CREATE TABLE Employee(
                             ID INT NOT NULL,
                             salary INT NOT NULL
                         CREATE TABLE Manager(
                             mID INT NOT NULL,
                             eID INT NOT NULL
                         CREATE TABLE Project(
                             name VARCHAR(30),
                             mgrID INT
                         );
                    INSERT INTO Employee
                        VALUES
                             (123, 100),
                             (890, 110),
                             (234, 90),
                             (345, 80),
                             (456, 70),
                             (567, 60),
                             (111, 200),
                             (222, 150),
                             (333, 120),
                             (098, 15);
                    INSERT INTO Manager
                        VALUES
                             (123, 234),
                             (111, 222),
                             (234, 345),
                             (234, 456),
                             (345, 567),
                             (123, 890),
                             (890, 098);
                    INSERT INTO Project
                        VALUES
                             ('X', 123),
('Y', 234),
('Z', 456);
WITH PRemps(ID) AS (SELECT mgrID AS ID FROM Project WHERE name LIKE @Proj
                        UNION ALL
                        SELECT eID as ID
                        FROM Manager M, PRemps X
                        WHERE M.mID = X.ID)
SELECT SUM (salary) AS PRojectSalary
FROM Employee
WHERE ID IN (SELECT ID FROM PRemps);
```

Результат:



3. Используя рекурсивный запрос, вывести первые 10 чисел Фибоначчи

```
WITH FibbonacciNumbers (RecursionLevel, FibbonacciNumber, NextNumber)
AS (

SELECT 0 AS RecursionLevel,
0 AS FibbonacciNumber,
1 AS NextNumber
UNION ALL
SELECT a.RecursionLevel + 1 AS RecursionLevel,
a.NextNumber AS FibbonacciNumber
a.FibbonacciNumber + a.NextNumber AS NextNumber
FROM FibbonacciNumbers a
WHERE a.RecursionLevel < 10

SELECT 'F' + CAST( fn.RecursionLevel AS VARCHAR) AS FibonacciOrdinal,
fn.FibbonacciNumber,
fn.NextNumber
FROM FibbonacciNumbers fn
G0
```

Результат:

| | FibonacciOrdinal | FibonacciNumber | NextNumber |
|----|------------------|-----------------|------------|
| 1 | F0 | 0 | 1 |
| 2 | F1 | 1 | 1 |
| 3 | F2 | 1 | 2 |
| 4 | F3 | 2 | 3 |
| 5 | F4 | 3 | 5 |
| 6 | F5 | 5 | 8 |
| 7 | F6 | 8 | 13 |
| 8 | F7 | 13 | 21 |
| 9 | F8 | 21 | 34 |
| 10 | F9 | 34 | 55 |
| 11 | F10 | 55 | 89 |