

Санкт-Петербургский колледж информационных технологий

ОТЧЕТ

по работе "Создание базы данных в RedisGraph"

по МДК07.01 "Управление и автоматизация баз данных"

Специальность 09.02.07

"Информационные системы и программирование"

Специализация:

"Администратор баз данных"

Выполнил:

студент 4-го курса группы 404 Колинко Роман

Преподаватель:

Фомин Александр Валерьевич

Санкт-Петербург, 2024 год

ВВЕДЕНИЕ

Целью работы является создание базы данных в RedisGraph.

Работа выполнена в ОС Linux в дистрибутиве Ubuntu 20.04.6 LTS.

ВЫПОЛНЕНИЕ

Были разработаны граф и ER-диаграмма базы данных генеалогического древа (рисунок 1-3), а также словарь данных (таблица 1-3). База данных состоит из таблиц данных о человеке (ID, имя, дата рождения и смерти, пол), его семейном положении (ID, муж, жена, дата свадьбы и конец женитьбы) и его детях (ID, ребенок, родители).

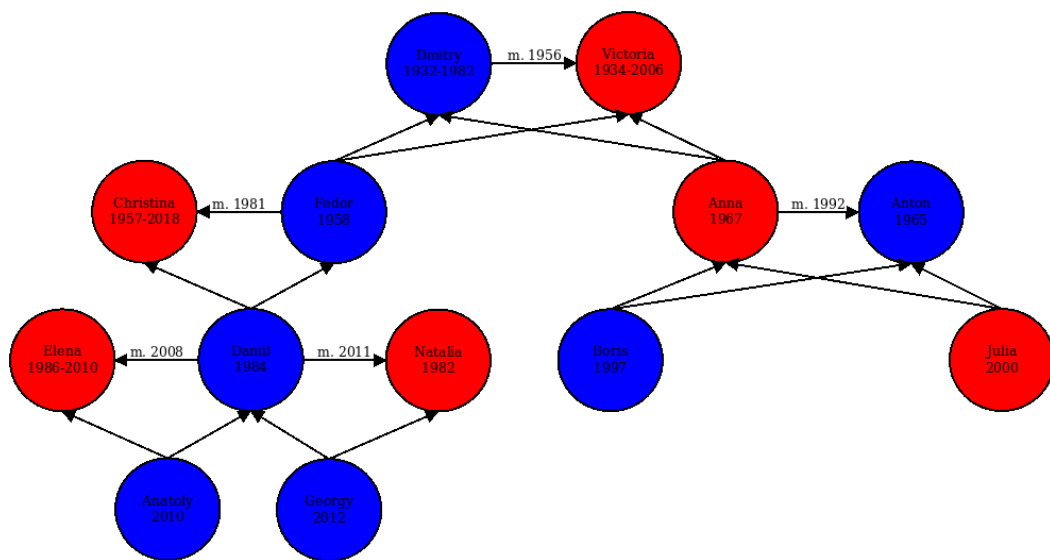


Рисунок 1 – Граф базы данных

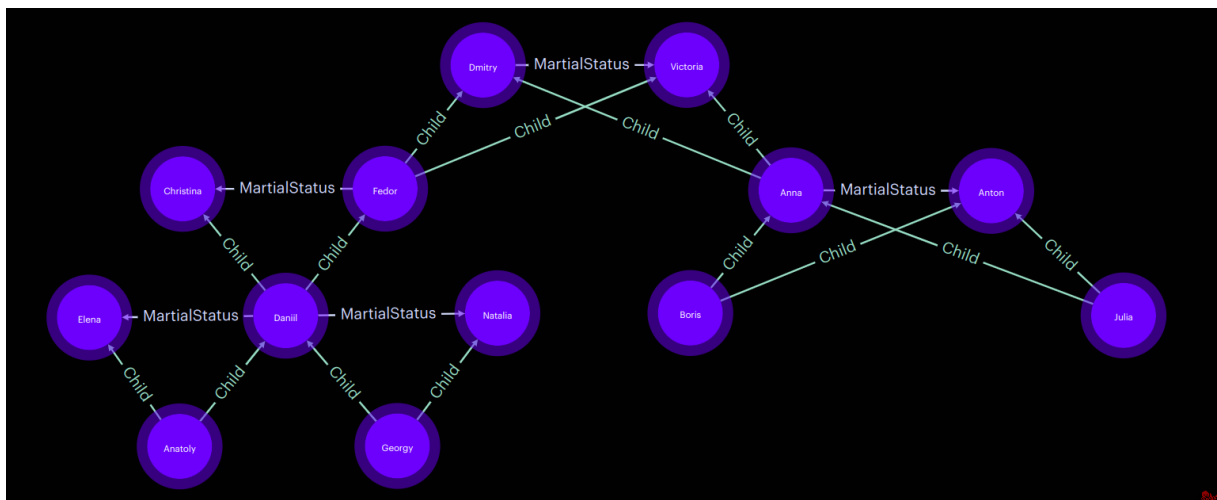


Рисунок 2 – Граф базы данных в RedisInsight

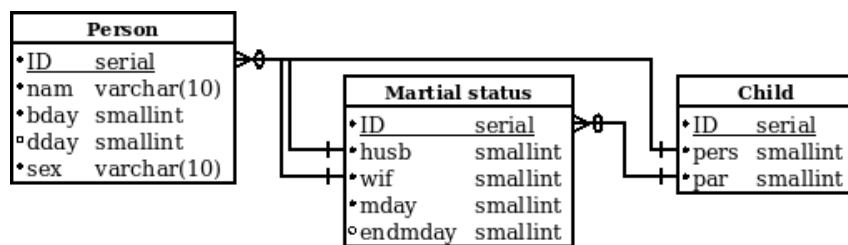


Рисунок 3 – ER-диаграмма

Таблица 1 – Человек

Описание – Данные человека

#	Столбец	Название	Тип данных	Нулевое	Первичный ключ	Внешний ключ	Описание
1	ID	ID	serial	Нет	Да		Идентификатор
2	nam	Имя	varchar(10)	Нет	Нет		Имя человека
3	bday	День рождения	smallint	Нет	Нет		Дата рождения
4	dday	День смерти	smallint	Да	Нет		Дата смерти
5	sex	Пол	varchar(10)	Нет	Нет		Пол

Таблица 2 – Семейное положение

Описание – Данные семейного положения человека

#	Столбец	Название	Тип данных	Нулевое	Первичный ключ	Внешний ключ	Описание
1	ID	ID	serial	Нет	Да		Идентификатор
2	husb	Муж	smallint	Нет	Нет	Person.ID	Муж
3	wif	Жена	smallint	Нет	Нет	Person.ID	Жена
4	mday	Дата свадьбы	smallint	Нет	Нет		День свадьбы
5	endmday	Конец женитьбы	smallint	Да	Нет		Конец женитьбы

Таблица 3 – Ребенок

Описание – Данные детей человека

#	Столбец	Название	Тип данных	Нулевое	Первичный ключ	Внешний ключ	Описание
1	ID	ID	serial	Нет	Да		Идентификатор
2	pers	Человек	smallint	Нет	Нет	Person.ID	Ребенок
3	par	Родители	smallint	Нет	Нет	'Marital status'.ID	Родители

Была создана база данных в PostgreSQL и RedisGraph путем написания скриптов на языках SQL и Cypher соответственно (рисунок 4-5).

```
romankolin=# CREATE DATABASE "Family tree";
CREATE DATABASE
romankolin=# \c "Family tree";
You are now connected to database "Family tree" as user "romankolin".
Family tree=# \i '/home/romankolin/Downloads/2 Создание базы данных в RedisGraph/Script.sql'
CREATE TABLE
CREATE TABLE
INSERT 0 13
INSERT 0 5
INSERT 0 7
```

Рисунок 4 – База данных в Postgresql

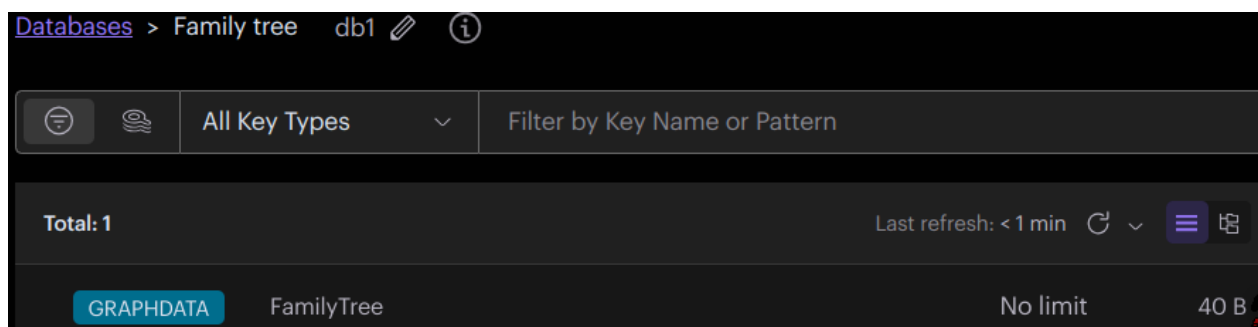


Рисунок 5 – База данных в RedisGraph

Далее были написаны запросы к базам данных в PostgreSQL и RedisGraph на выборку всех людей с датой рождения в хронологическом порядке, родившихся до 1969 года людей с датой рождения в хронологическом порядке, живых людей, умерших людей и их продолжительности жизни, средней продолжительности жизни умерших людей, года смерти Кристины, возраста вступления в брак, среднего возраста вступления в брак, матери Георгия, количества детей у Антона, предков Анатолия, потомков Виктории, двоюродных родственников Даниила (рисунок 6-18).

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(family:Person) RETURN family.name AS Name, family.bday AS Birthday ORDER BY family.bday"
1) 1) "Name"
2) 2) "Birthday"
3) 1) 1) "Dmitry"
2) 2) "1932"
4) 1) "Victoria"
2) 2) "1934"
5) 1) "Christina"
2) 2) "1957"
6) 1) "Fedor"
2) 2) "1958"
7) 1) "Anton"
2) 2) "1965"
8) 1) "Anna"
2) 2) "1965"
9) 1) "Natalia"
2) 2) "1982"
10) 1) "Daniil"
2) 2) "1984"
11) 1) "Elena"
2) 2) "1986"
12) 1) "Boris"
2) 2) "1997"
13) 1) "Julia"
2) 2) "2000"
14) 1) "Anatoly"
2) 2) "2010"
15) 1) "Georgy"
2) 2) "2012"
16) 1) "Cached execution: 1"
17) 2) "Query internal execution time: 0.218400 milliseconds"
```

```
romankolin@romankolinUB: ~
Family tree=# SELECT nam AS "Name", bday AS "Birthday" FROM Person ORDER BY bday;
-----+-----
Name      | Birthday
-----+-----
Dmitry    | 1932
Victoria  | 1934
Christina | 1957
Fedor     | 1958
Anton     | 1965
Anna      | 1965
Natalia   | 1982
Daniil    | 1984
Elena     | 1986
Boris     | 1997
Julia     | 2000
Anatoly   | 2010
Georgy    | 2012
( 13 rows )
Family tree=#
```

Рисунок 6 – Люди с датой рождения в хронологическом порядке

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(family:Person) WHERE family.bday < 1969 RETURN family.name AS Name, family.bday AS Birthday ORDER BY family.bday"
1) 1) "Name"
2) 2) "Birthday"
3) 1) 1) "Dmitry"
2) 2) "1932"
4) 1) "Victoria"
2) 2) "1934"
5) 1) "Christina"
2) 2) "1957"
6) 1) "Fedor"
2) 2) "1958"
7) 1) "Anton"
2) 2) "1965"
8) 1) "Anna"
2) 2) "1967"
9) 1) "Cached execution: 1"
10) 2) "Query internal execution time: 0.163779 milliseconds"
```

```
romankolin@romankolinUB: ~
Family tree=# SELECT nam AS "Name", bday AS "Birthday" FROM Person WHERE bday < 1969 ORDER BY bday;
-----+-----
Name      | Birthday
-----+-----
Dmitry    | 1932
Victoria  | 1934
Christina | 1957
Fedor     | 1958
Anton     | 1965
Anna      | 1967
( 6 rows )
```

Рисунок 7 – Родившиеся до 1969 года люди с датой рождения в хронологическом порядке

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(alive:Person) WHERE alive.dday IS null RETURN alive.name AS Name"
1) 1) "Name"
2) 1) "Fedor"
3) 2) "Anna"
4) 1) "Daniil"
5) 2) "Anatoly"
6) 1) "Natalia"
7) 1) "Georgy"
8) 1) "Anton"
9) 1) "Boris"
10) 1) "Julia"
11) 1) "Cached execution: 0"
12) 2) "Query internal execution time: 0.212161 milliseconds"
```

```
romankolin@romankolinUB: ~
Family tree=# SELECT nam AS "Name" FROM Person WHERE dday IS NULL;
-----
Fedor
Anna
Daniil
Anatoly
Natalia
Georgy
Anton
Boris
Julia
( 9 rows )
```

Рисунок 8 – Живые люди

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(death:Person) WHERE death.dday IS NOT null RETURN death.name AS Name, death.dday-death.bday AS Age"
1) 1) "Name"
2) 2) "Age"
3) 1) 1) "Dmitry"
2) 2) "58"
4) 1) "Victoria"
2) 2) "72"
5) 1) "Christina"
2) 2) "61"
6) 1) "Elena"
2) 2) "24"
7) 1) "Cached execution: 0"
8) 2) "Query internal execution time: 4.696429 milliseconds"
```

```
romankolin@romankolinUB: ~
Family tree=# SELECT nam AS "Name", dday-bday AS "Age" FROM Person WHERE dday IS NOT NULL;
-----+-----
Name      | Age
-----+-----
Dmitry    | 58
Victoria  | 72
Christina | 61
Elena     | 24
( 4 rows )
```

Рисунок 9 – Умершие люди и их продолжительность жизни

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(average:Person) WHERE average.dday IS NOT null RETURN ROUND(AVG(average.dday-average.bday)) AS 'Average age'"
1) 1) "Average age"
2) 1) "52"
3) 1) "Cached execution: 0"
2) "Query internal execution time: 0.320134 milliseconds"

Family tree=# SELECT ROUND( AVG( dday-bday )) AS "Average age" FROM Person WHERE dday IS NOT NULL;
Average age
-----
52
(1 row)
```

Рисунок 10 – Средняя продолжительность жизни умерших людей

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(deathday:Person) WHERE deathday.name='Christina' RETURN deathday.dday AS Deathday"
1) 1) "Deathday"
2) 1) "2018"
3) 1) "Cached execution: 0"
2) "Query internal execution time: 0.264223 milliseconds"

Family tree=# SELECT dday AS "Deathday" FROM Person WHERE nam='Christina';
Deathday
-----
2018
(1 row)
```

Рисунок 11 – Год смерти Кристины

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(male:Person)-[marriage:MaritalStatus]->(female:Person) RETURN male.name AS Husband, female.name AS Wife, marriage.mday-male.bday AS 'Husband marriage age', marriage.mday-female.bday AS 'Wife marriage age'"
1) 1) "Husband"
2) 1) "Wife"
3) 1) "Husband marriage age"
4) 1) "Wife marriage age"
2) 1) 1) "Dmitry"
2) 2) "Victoria"
3) 3) "24"
4) 4) "22"
2) 1) "Fedor"
2) 2) "Christina"
3) 3) "23"
4) 4) "24"
3) 1) "Anna"
2) 2) "Anton"
3) 3) "25"
4) 4) "27"
4) 1) "Danill"
2) 2) "Elena"
3) 3) "24"
4) 4) "22"
5) 1) "Danill"
2) 2) "Natalia"
3) 3) "27"
4) 4) "29"
3) 1) "Cached execution: 1"
2) "Query internal execution time: 4.089159 milliseconds"

Family tree=# SELECT nam AS "Name", mday-bday AS "Marriage age" FROM Person JOIN "Marital status" ON Person.ID="Marital status".husb UNION SELECT nam AS "Name", mday-bday AS "Marriage age" FROM Person JOIN "Marital status" ON Person.ID="Marital status".wif;
Name | Marriage age
-----+-----
Anton | 27
Danill | 27
Elena | 22
Fedor | 23
Christina | 24
Anna | 25
Natalia | 29
Danill | 24
Victoria | 24
Dmitry | 24
(10 rows)

Family tree=#
```

Рисунок 12 – Возраст вступления в брак

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(male:Person)-[marriage:MaritalStatus]->(female:Person) RETURN ROUND((AVG(marriage.mday-male.bday + marriage.mday-female.bday))/2) AS 'Average marriage age'"
1) 1) "Average marriage age"
2) 1) "25"
3) 1) "Cached execution: 0"
2) "Query internal execution time: 0.685833 milliseconds"

Family tree=# WITH avermarriage AS (SELECT AVG(mday-bday) AS "Average marriage age" FROM Person JOIN "Marital status" ON Person.ID="Marital status".husb UNION SELECT AVG(mday-bday) AS "Average marriage age" FROM Person JOIN "Marital status" ON Person.ID="Marital status".wif) Family tree=# SELECT ROUND(AVG("Average marriage age")) AS "Average marriage age" FROM avermarriage;
Average marriage age
-----
25
(1 row)
```

Рисунок 13 – Средний возраст вступления в брак

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(child:Person)-[:Child]->(mother:Person) WHERE child.name='Georgy' AND mother.sex='Female' RETURN mother.name AS Name"
1) 1) "Name"
2) 1) "Natalia"
3) 1) "Cached execution: 0"
2) "Query internal execution time: 31.924521 milliseconds"

Family tree=# SELECT nam AS "Name" FROM Person JOIN "Marital status" ON Person.ID="Marital status".wif JOIN Child ON "Marital status".ID=Child.par WHERE pers=(SELECT ID FROM Person WHERE nam='Georgy');
Name
-----
Natalia
(1 row)
```

Рисунок 14 – Мать Георгия

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(child:Person)-[:Child]->(father:Person) WHERE father.name='Anton' RETURN COUNT(child) AS 'Number of children'"
1) 1) "Number of children"
2) 1) "Fedor"
3) 1) "2"
4) 1) "Cached execution: 1"
5) 2) "Query internal execution time: 0.129872 milliseconds"
```

```
Family tree=# SELECT COUNT(Child.ID) AS "Number of children" FROM Child JOIN "Marital status" ON Child.par="Marital status".ID JOIN Person ON "Marital status".husb=Person.ID WHERE "Marital status".husb=(SELECT ID FROM Person WHERE nam='Anton');
Number of children
-----
2
(1 row)
```

Рисунок 15 – Количество детей у Антона

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(child:Person)-[:Child1..]->(ancestor:Person) WHERE child.name='Anatoly' RETURN ancestor.name AS Name"
1) 1) "Name"
2) 1) "Danil"
3) 1) "Fedor"
4) 1) "Dmitry"
5) 1) "Victoria"
6) 1) "Christina"
7) 1) "Elena"
8) 1) "Cached execution: 0"
9) 2) "Query internal execution time: 0.357183 milliseconds"
```

```
Family tree=# CREATE TEMPORARY TABLE maritalstatus(husb smallint, wif smallint, child smallint);
CREATE TABLE
Family tree=# INSERT INTO maritalstatus SELECT DISTINCT "Marital status".husb, "Marital status".wif, Child.pers FROM Person JOIN "Marital status" ON (Person.ID="Marital status".husb OR Person.ID="Marital status".wif) JOIN Child ON "Marital status".ID=Child.par ORDER BY husb;
INSERT 0 7
Family tree=# WITH wife AS (WITH husband AS (WITH RECURSIVE ancestor AS (SELECT child, husb, wif FROM maritalstatus WHERE child=(SELECT ID FROM Person WHERE nam='Anatoly') UNION SELECT maritalstatus.child, maritalstatus.husb, maritalstatus.wif FROM maritalstatus JOIN ancestor ON (maritalstatus.child=ancestor.husb OR maritalstatus.child=ancestor.wif)) SELECT * FROM ancestor) SELECT child, child.nam AS "Child", child.husb, child.wif FROM Person JOIN child ON Person.ID=child.child) SELECT husband, "Child", nam AS "Husband", husband.wif FROM Person JOIN husband ON Person.ID=husband.husb) SELECT wife, "Child", wife, "Husband", nam AS "Wife" FROM Person JOIN wife ON Person.ID=wife.wif;
Child | Husband | Wife
-----+-----+-----
Anatoly | Danil | Elena
Danil | Fedor | Christina
Fedor | Dmitry | Victoria
(3 rows)
```

Рисунок 16 – Предки Анатолия

```
[db1] > GRAPH.QUERY FamilyTree "MATCH(child:Person)-[:Child1..]->(descendant:Person) WHERE descendant.name='Victoria' RETURN child.name AS Name"
1) 1) "Name"
2) 1) "Fedor"
3) 1) "Danil"
4) 1) "Anatoly"
5) 1) "Georgy"
6) 1) "Anna"
7) 1) "Boris"
8) 1) "Julia"
9) 1) "Cached execution: 0"
10) 2) "Query internal execution time: 0.392850 milliseconds"
```

```
Family tree=# CREATE TEMPORARY TABLE maritalstatus(husb smallint, wif smallint, child smallint);
CREATE TABLE
Family tree=# INSERT INTO maritalstatus SELECT DISTINCT "Marital status".husb, "Marital status".wif, Child.pers FROM Person JOIN "Marital status" ON (Person.ID="Marital status".husb OR Person.ID="Marital status".wif) JOIN Child ON "Marital status".ID=Child.par ORDER BY husb;
INSERT 0 7
Family tree=# WITH child AS (WITH wife AS (WITH husband AS (WITH RECURSIVE descendant AS (SELECT husb, wif, child FROM maritalstatus WHERE wif=(SELECT ID FROM Person WHERE nam='Victoria') UNION SELECT maritalstatus.husb, maritalstatus.wif, maritalstatus.child FROM maritalstatus JOIN descendant ON (maritalstatus.husb=descendant.child OR maritalstatus.wif=descendant.child)) SELECT * FROM descendant) SELECT husband.husb, husband.nam AS "Husband", husband.wif, husband.child FROM Person JOIN husband ON Person.ID=husband.husb) SELECT wife, "Husband", nam AS "Wife", wife, child FROM Person JOIN wife ON Person.ID=wife.wif) SELECT child, "Husband", child, "Wife", nam AS "Child" FROM Person JOIN child ON Person.ID=child.child;
Husband | Wife | Child
-----+-----+-----
Dmitry | Victoria | Fedor
Dmitry | Victoria | Anna
Fedor | Christina | Danil
Anton | Anna | Boris
Anton | Anna | Julia
Danil | Elena | Anatoly
Danil | Natalia | Georgy
(7 rows)
```

Рисунок 17 – Потомки Виктории

```
[db1] > GRAPH.QUERY FamilyTree "MATCH ((child:Person)-[:Child]->(fousin:Person)-[:Child]->(family:Person)-[:Child]->(scousin:Person)-[:Child]->(child:Person) WHERE (fchild.name = 'Danil' AND fousin=scousin) RETURN child.name AS Name"
1) 1) "Name"
2) 1) "Boris"
3) 1) "Julia"
4) 1) "Cached execution: 1"
5) 2) "Query internal execution time: 0.462416 milliseconds"
```

```
Family tree=# CREATE TEMPORARY TABLE maritalstatus(husb smallint, wif smallint, child smallint);
CREATE TABLE
Family tree=# INSERT INTO maritalstatus SELECT DISTINCT "Marital status".husb, "Marital status".wif, Child.pers FROM Person JOIN "Marital status" ON (Person.ID="Marital status".husb OR Person.ID="Marital status".wif) JOIN Child ON "Marital status".ID=Child.par ORDER BY husb;
INSERT 0 7
Family tree=# WITH cousin AS (WITH cousin2 AS (WITH cousin1 AS (WITH cousin AS (SELECT * FROM maritalstatus WHERE child=(SELECT ID FROM Person WHERE nam='Danil')) SELECT husb, wif, child FROM maritalstatus WHERE child=(SELECT husb FROM cousin1) SELECT child FROM maritalstatus WHERE wif=(SELECT wif FROM cousin1)) SELECT child FROM maritalstatus WHERE wif IN (SELECT child FROM cousin2)) SELECT nam AS "Name" FROM Person JOIN cousin ON Person.ID=cousin.child;
Name
-----
Boris
Julia
(2 rows)
```

Рисунок 18 – Двоюродные родственники Даниила