Федеральное государственное автономное образовательное учреждение высшего образования
"Национальный Исследовательский Университет ИТМО"
Мегафакультет Компьютерных Технологий и Управления
Факультет Программной Инженерии и Компьютерной Техники

Дополнительное задание к лабораторной работе №1

по дисциплине

'Информационные системы и базы данных'

Вариант 3893

Выполнил Студент группы P32101
Лапин Алексей Александрович
Преподаватель:
Сагайдак Алина Алексеевна

г. Санкт-Петербург
2023г.

Содержание

1 Текст задания. **3**
1.1 Первая таска: . . . 3
1.2 Вторая таска: . . . 3

2 Задание 1. **3**
2.1 Query: . . . 3
2.2 Result: . . . 4
2.3 Code: . . . 4

3 Задание 2. **11**
3.1 Инфологическая модель . . . 11
3.2 Даталогическая модель . . . 12

4 Реализация даталогической модели на SQL. **13**

5 Вывод **17**

1 Текст задания.

1.1 Первая таска:

Запрос: выведите имя человека, сила воли которого может быть сдержанной, устойчивой или самоуправленческой, который живет в квартире в Питере. При этом, он должен находиться в дружеских отношениях с человеком, имя которого начинается с "А"и содержит не больше 7 букв. По мимо того, у этого человека (у первого) должны быть планы в активном статусе, описание шагов которых больше 10 символов и эти шаги трудные!
Если таких людей несколько, то отсортируйте их по длине имени в убывающем порядке.

1.2 Вторая таска:

Ваша предметная область: университет
Сами задания:

1. Придумать ER модель и нарисовать даталогическую модель для предложенной предметной области (минимум 8 сущностей, в одной из сущностей минимум 6 атрибутов, связь вида многие-ко-многим, один-к-одному)
2. Привести DDL для реализации сущностей, которые участвуют в связях многие-ко-многим, один-к-одному (связи тоже должны присутствовать в коде)
3. Привести пример использования языка DML

2 Задание 1.

2.1 Query:

```sql
SELECT p.name
FROM people p
JOIN personLiveInHouse plh ON p.id = plh.personId
JOIN houses h ON plh.houseId = h.id
JOIN relationships r ON p.id = r.personId1
JOIN people p2 ON r.personId2 = p2.id
JOIN plans pl ON p.id = pl.ownerId
JOIN steps s ON pl.id = s.planId
WHERE p.willpower IN (1, 2, 3) -- Assuming 1, 2, 3 correspond to restrained, stable, or self-governing willpower
AND h.location LIKE 'St. Petersburg%'
AND h.houseTypeId = (SELECT id FROM houseTypes WHERE name = 'apartments')
AND r.relationTypeId = (SELECT id FROM relationshipTypes WHERE name = 'friend')
AND p2.name LIKE 'A%' AND LENGTH(p2.name) <= 7
AND pl.statusId = (SELECT id FROM statuses WHERE name = 'active')
AND LENGTH(s.description) >= 10
AND s.difficultyid = (SELECT id FROM difficulties WHERE name = 'hard')
GROUP BY p.id, p.name
```

```
18 ORDER BY LENGTH(p.name) DESC;
```

2.2 Result:

```
 name
-----------
 Alexander
 Amelia
 Alexey
(3 строки)
```

2.3 Code:

Создание таблиц:

```
1  CREATE TABLE IF NOT EXISTS people (
2      id              serial          PRIMARY KEY,
3      name            varchar(32)     NOT NULL,
4      willpower       integer         NOT NULL CHECK ( willpower > 0),
5      gender          varchar(32)     NOT NULL CHECK(gender IN ('MALE',
           'FEMALE'))
6  );
7
8  CREATE TABLE IF NOT EXISTS houseTypes (
9      id      serial          PRIMARY KEY,
10     name    varchar(32)     NOT NULL UNIQUE
11 );
12
13 CREATE TABLE IF NOT EXISTS houses (
14     id              serial          PRIMARY KEY,
15     location        varchar(128)    NOT NULL UNIQUE,
16     houseTypeId     integer         REFERENCES houseTypes(id) ON DELETE
           SET NULL
17 );
18
19 CREATE TABLE IF NOT EXISTS personLiveInHouse(
20     personId    integer     REFERENCES people(id) ON DELETE CASCADE,
21     houseId     integer     REFERENCES houses(id) ON DELETE CASCADE,
22     PRIMARY KEY (personID, houseId)
23 );
24
25 CREATE TABLE IF NOT EXISTS relationshipTypes (
26     id      serial          PRIMARY KEY,
27     name    varchar(32)     NOT NULL UNIQUE
28 );
29
30 CREATE TABLE IF NOT EXISTS relationships (
31     personId1       integer     REFERENCES people(id) ON DELETE CASCADE,
32     personId2       integer     REFERENCES people(id) ON DELETE CASCADE,
33     relationTypeId  integer     REFERENCES relationshipTypes(id) ON DELETE
           CASCADE,
34     lastEditedDate  date,
```

```
    PRIMARY KEY (personId1, personId2),
    CHECK ( personId1 <> personId2 )
);

CREATE TABLE IF NOT EXISTS statuses (
    id      serial              PRIMARY KEY,
    name    varchar(32)         NOT NULL UNIQUE
);

CREATE TABLE IF NOT EXISTS goal (
    id      serial              PRIMARY KEY,
    name    varchar(128)         NOT NULL UNIQUE
);

CREATE TABLE IF NOT EXISTS difficulties (
    id      serial          PRIMARY KEY,
    name    varchar(32)     NOT NULL UNIQUE,
    power   integer         NOT NULL CHECK ( power > 0)
);

CREATE TABLE IF NOT EXISTS plans (
    id          serial      PRIMARY KEY,
    ownerId     integer     REFERENCES people(id) ON DELETE CASCADE,
    statusId    integer     REFERENCES statuses(id) ON DELETE SET NULL
);

CREATE TABLE IF NOT EXISTS goalInPlan (
    planId      integer    REFERENCES plans(id) ON DELETE CASCADE,
    goalId      integer    REFERENCES goal(id) ON DELETE CASCADE,
    PRIMARY KEY (planId, goalId)
);

CREATE TABLE IF NOT EXISTS steps (
    id              serial          PRIMARY KEY,
    planId          integer         REFERENCES plans(id) ON DELETE CASCADE,
    difficultyId    integer         REFERENCES difficulties(id) ON DELETE
       CASCADE,
    description     varchar(128)
);

CREATE TABLE IF NOT EXISTS support(
    supporterId     integer     REFERENCES people(id) ON DELETE CASCADE,
    planId          integer     REFERENCES plans(id) ON DELETE CASCADE,
    supportPower    integer,
    PRIMARY KEY (supporterId, planId)
)
```

Удаление таблиц:

```
DROP TABLE IF EXISTS goalInPlan;
DROP TABLE IF EXISTS goal;
DROP TABLE IF EXISTS steps;
```

```
DROP TABLE IF EXISTS difficulties;
DROP TABLE IF EXISTS relationships;
DROP TABLE IF EXISTS relationshipTypes;
DROP TABLE IF EXISTS support;
DROP TABLE IF EXISTS personliveinhouse;
DROP TABLE IF EXISTS plans;
DROP TABLE IF EXISTS people;
DROP TABLE IF EXISTS statuses;
DROP TABLE IF EXISTS houses;
DROP TABLE IF EXISTS houseTypes;
```

Заполнение тестовых значений:

```
INSERT INTO people values (DEFAULT, 'Floyd', 1, 'MALE'); -- 1
INSERT INTO people values (DEFAULT, 'Anna', 2, 'FEMALE'); -- 2
INSERT INTO people values (DEFAULT, 'Halvorsen', 3, 'MALE'); -- 3
INSERT INTO people values (DEFAULT, 'Sirenis',4, 'FEMALE'); -- 4
INSERT INTO people values (DEFAULT, 'Alexander', 1, 'MALE'); -- 5
INSERT INTO people values (DEFAULT, 'Amelia', 2, 'FEMALE'); -- 6
INSERT INTO people values (DEFAULT, 'Adam', 3, 'MALE'); -- 7
INSERT INTO people values (DEFAULT, 'Alexey', 2, 'MALE'); -- 8

INSERT INTO houseTypes values (DEFAULT, 'apartments') ON CONFLICT DO
   NOTHING;
INSERT INTO houseTypes values (DEFAULT, 'detached house') ON CONFLICT DO
   NOTHING;
INSERT INTO houseTypes values (DEFAULT, 'cottage') ON CONFLICT DO NOTHING;
INSERT INTO houseTypes values (DEFAULT, 'townhouse') ON CONFLICT DO
   NOTHING;

INSERT INTO houses values (DEFAULT, 'St. Petersburg Kronverksky Pr. 49',
   (SELECT housetypes.id FROM houseTypes WHERE name='apartments'));
INSERT INTO houses values (DEFAULT, 'St. Petersburg Lomonosova Street, 9',
   (SELECT housetypes.id FROM houseTypes WHERE name='detached house'));
INSERT INTO houses values (DEFAULT, 'Moscow Tchaykovskogo St, 11/2',
   (SELECT housetypes.id FROM houseTypes WHERE name='cottage'));
INSERT INTO houses values (DEFAULT, 'Moscow Grivtsova Lane, 14', (SELECT
   housetypes.id FROM houseTypes WHERE name='townhouse'));
INSERT INTO houses values (DEFAULT, 'Murmansk Birzhevaya Line, 14',
   (SELECT housetypes.id FROM houseTypes WHERE name='apartments'));


DO $$
    DECLARE NUM_OF_PEOPLE INT = 8;
    DECLARE NUM_OF_PLANS INT = 0;
    DECLARE NUM_OF_STEPS INT = 0;
BEGIN
    FOR i IN 2..NUM_OF_PEOPLE LOOP
        INSERT INTO personliveinhouse VALUES (i, 1);
    END LOOP;
    FOR i IN 1..NUM_OF_PEOPLE LOOP
        IF i NOT IN (2, 3, 4) THEN
```

```
32              INSERT INTO personliveinhouse VALUES (i, 2);
33          END IF;
34      END LOOP;
35      FOR i IN 1..NUM_OF_PEOPLE LOOP
36          IF i NOT IN (4, 6, 7) THEN
37              INSERT INTO personliveinhouse VALUES (i, 3);
38          END IF;
39      END LOOP;
40      FOR i IN 1..NUM_OF_PEOPLE LOOP
41          IF i NOT IN (8, 2, 1) THEN
42              INSERT INTO personliveinhouse VALUES (i, 4);
43          END IF;
44      END LOOP;
45      FOR i IN 1..NUM_OF_PEOPLE LOOP
46          IF i NOT IN (1, 2, 3) THEN
47              INSERT INTO personliveinhouse VALUES (i, 5);
48          END IF;
49      END LOOP;
50
51      INSERT INTO relationshiptypes values (DEFAULT, 'friend');
52      INSERT INTO relationshiptypes values (DEFAULT, 'enemy');
53
54      FOR i IN 1..NUM_OF_PEOPLE LOOP
55          FOR j IN 1..NUM_OF_PEOPLE LOOP
56              IF i != j AND i NOT IN (2, 4, 1, 3) AND j NOT IN (2, 4, 1, 3)
                   THEN
57                  INSERT INTO relationships values (
58                                                      i,
59                                                      j,
60                                                      1,
61                                                      '2019-01-01'
62                                                  );
63
64              ELSIF (i != j) AND (i IN (2, 4, 1, 3) OR j IN (2, 4, 1, 3))
                   THEN
65                  INSERT INTO relationships values (
66                                                      i,
67                                                      j,
68                                                      2,
69                                                      '2020-03-12'
70                                                  );
71              END IF;
72          END LOOP;
73      END LOOP;
74
75      INSERT INTO difficulties values (DEFAULT, 'easy', 100);
76      INSERT INTO difficulties values (DEFAULT, 'medium', 200);
77      INSERT INTO difficulties values (DEFAULT, 'hard', 300);
78
79      INSERT INTO statuses values (DEFAULT, 'active');
80      INSERT INTO statuses values (DEFAULT, 'done');
```

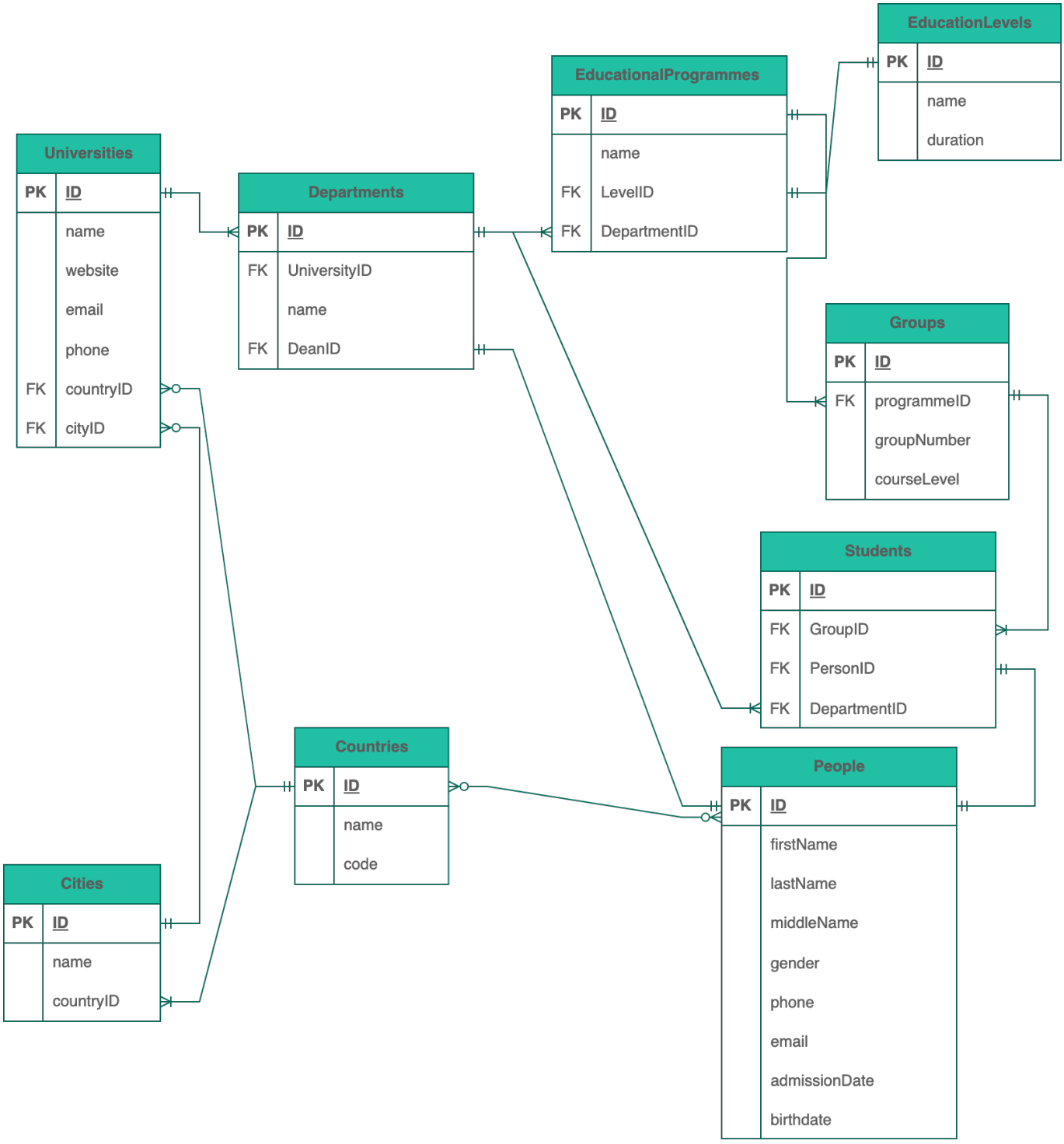
```
81      INSERT INTO statuses values (DEFAULT, 'failed');
82
83      FOR i IN 1..NUM_OF_PEOPLE LOOP
84          INSERT INTO plans values (
85                                      DEFAULT,
86                                      i,
87                                      1
88                                  );
89          NUM_OF_PLANS := NUM_OF_PLANS + 1;
90      END LOOP;
91
92
93
94      FOR i IN 1..NUM_OF_PEOPLE LOOP
95          IF i NOT IN (2, 4, 1, 3) THEN
96              INSERT INTO plans values (
97                                          DEFAULT,
98                                          i,
99                                          2
100                                     );
101             NUM_OF_PLANS := NUM_OF_PLANS + 1;
102         END IF;
103     END LOOP;
104
105
106     FOR i IN 1..NUM_OF_PEOPLE LOOP
107         IF i NOT IN (5, 6, 2, 8) THEN
108             INSERT INTO plans values (
109                                         DEFAULT,
110                                         i,
111                                         3
112                                     );
113             NUM_OF_PLANS := NUM_OF_PLANS + 1;
114         END IF;
115     END LOOP;
116
117     NUM_OF_STEPS := 12;
118
119     FOR i IN 1..NUM_OF_PLANS LOOP
120         FOR j IN 1..NUM_OF_STEPS LOOP
121             IF i NOT IN (2, 4, 1, 3, 5, 6, 7, 8) THEN
122                 INSERT INTO steps values (
123                                             DEFAULT,
124                                             i,
125                                             1,
126                                             'Do something special'
127                                         );
128             ELSIF i IN (2, 4) THEN
129                 INSERT INTO steps values (
130                                             DEFAULT,
131                                             i,
```

```
                                        2,
                                        'Do it'
                                    );
            ELSIF i IN (7) THEN
                INSERT INTO steps values (
                                        DEFAULT,
                                        i,
                                        3,
                                        'Do it'
                                    );
            ELSE
                INSERT INTO steps values (
                                        DEFAULT,
                                        i,
                                        3,
                                        'Do something amazing'
                                    );
            END IF;
        END LOOP;
    END LOOP;


    INSERT INTO goal values (
                             DEFAULT,
                             'Become a millionaire from billionaire'
                            );
    INSERT INTO goal values (
                             DEFAULT,
                             'Graduate from university'
                            );
    INSERT INTO goal values (
                             DEFAULT,
                             'Conquer the world'
                            );
    INSERT INTO goal values (
                             DEFAULT,
                             'Submit the laboratory work'
                            );
    INSERT INTO goal values (
                             DEFAULT,
                             'Go to sleep'
                            );

    FOR i IN 1..NUM_OF_PLANS LOOP
        IF i NOT IN (2, 4, 1, 3, 5, 6, 7, 8) THEN
            INSERT INTO goalinplan values (
                                           i,
                                           (SELECT id FROM goal WHERE
                                              name='Become a millionaire
                                              from billionaire')
                                           );
```

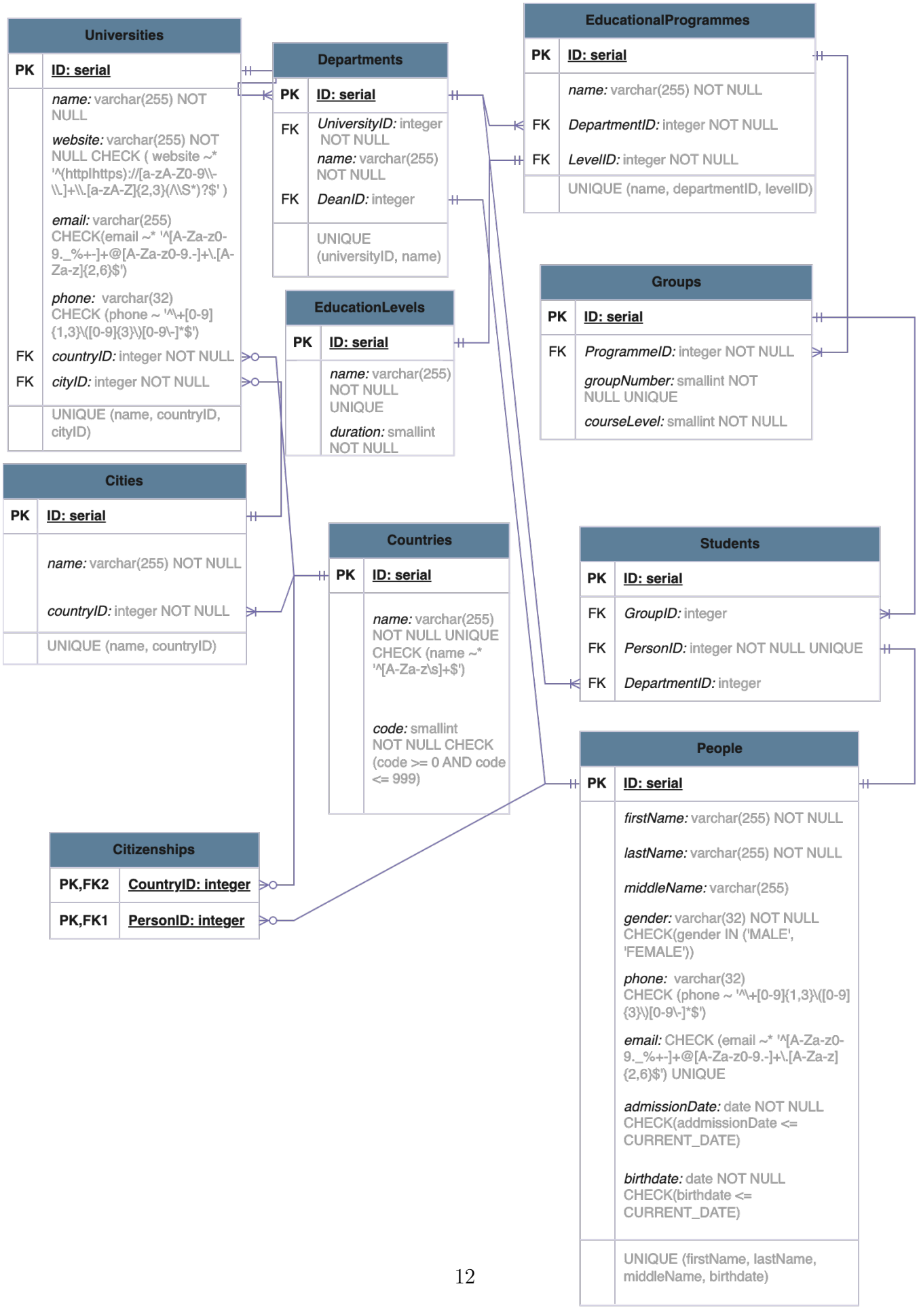
```
181         ELSIF i IN (2, 4) THEN
182             INSERT INTO goalinplan values (
183                                             i,
184                                              (SELECT id FROM goal WHERE
                                                    name='Graduate from
                                                    university')
185                                             );
186         ELSE
187             INSERT INTO goalinplan values (
188                                             i,
189                                              (SELECT id FROM goal WHERE
                                                    name='Conquer the world')
190                                             );
191         END IF;
192     END LOOP;
193
194     FOR i IN 1..NUM_OF_PEOPLE LOOP
195         FOR j IN 1..NUM_OF_PLANS LOOP
196             IF (SELECT relationtypeid FROM relationships WHERE
197                                                              personid1 = i
198                                                          AND personid2 =
                                                             (SELECT
                                                             ownerid FROM
                                                             plans WHERE id
                                                             = j)) = 1 THEN
199                 INSERT INTO support values (
200                                              i,
201                                              j,
202                                              100
203                                             );
204             ELSE
205                 INSERT INTO support values (
206                                              i,
207                                              j,
208                                              -100
209                                             );
210             END IF;
211         END LOOP;
212     END LOOP;
213 END $$
```

3 Задание 2.

3.1 Инфологическая модель



3.2 Даталогическая модель



4 Реализация даталогической модели на SQL.

Создание таблиц:

```
CREATE TABLE IF NOT EXISTS Countries(
    id              serial          PRIMARY KEY,
    name            varchar(255)    NOT NULL UNIQUE CHECK (name ~*
        '^[A-Za-z\s]+$'),
    code            smallint        NOT NULL CHECK (code >= 0 AND code <=
        999)
);

CREATE TABLE IF NOT EXISTS Cities(
    id              serial          PRIMARY KEY,
    name            varchar(255)    NOT NULL,
    countryID       integer         NOT NULL REFERENCES Countries(id) ON
        DELETE CASCADE ON UPDATE CASCADE,
    UNIQUE (name, countryID)
);

CREATE TABLE IF NOT EXISTS Universities(
    id              serial          PRIMARY KEY,
    name            varchar(255)    NOT NULL,
    website         varchar(255)    NOT NULL CHECK ( website ~*
        '^(http|https)://[a-zA-Z0-9\\-\\.]+\.[a-zA-Z]{2,3}(/\S*)?$' ),
    email           varchar(255)    CHECK(email ~*
        '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}$'),
    phone           varchar(32)     CHECK (phone ~
        '^\+[0-9]{1,3}\([0-9]{3}\)[0-9\-]*$'),
    countryID       integer         REFERENCES Countries(id) ON DELETE SET
        NULL ON UPDATE CASCADE NOT NULL,
    cityID          integer         REFERENCES Cities(id) ON DELETE SET
        NULL ON UPDATE CASCADE NOT NULL,
    UNIQUE (name, countryID, cityID)
);

CREATE TABLE IF NOT EXISTS People(
    id              serial          PRIMARY KEY,
    firstName       varchar(255)    NOT NULL,
    lastName        varchar(255)    NOT NULL,
    middleName      varchar(255),
    gender          varchar(32)     NOT NULL CHECK(gender IN ('MALE',
        'FEMALE')),
    phone           varchar(32)     CHECK (phone ~*
        '^\+[0-9]{1,3}\([0-9]{3}\)[0-9\-]*$'),
    email           varchar(255)    CHECK (email ~*
        '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}$') UNIQUE,
    addmissionDate  date            NOT NULL CHECK(addmissionDate <=
        CURRENT_DATE),
    birthdate       date            NOT NULL CHECK(birthdate <=
        CURRENT_DATE),
```

```
35      UNIQUE (firstName, lastName, middleName, birthdate)
36  );
37
38  CREATE TABLE IF NOT EXISTS Citizenships(
39      countryID       integer         NOT NULL REFERENCES Countries(id) ON
            DELETE CASCADE ON UPDATE CASCADE,
40      personID        integer         NOT NULL REFERENCES People(id) ON
            DELETE CASCADE ON UPDATE CASCADE
41  );
42
43
44  CREATE TABLE IF NOT EXISTS Departments(
45      id              serial          PRIMARY KEY,
46      universityID    integer         NOT NULL REFERENCES Universities(id)
            ON DELETE CASCADE ON UPDATE CASCADE,
47      name            varchar(255)    NOT NULL,
48      deanID          integer         REFERENCES People(id) ON DELETE SET
            NULL ON UPDATE CASCADE,
49      UNIQUE (universityID, name)
50  );
51
52  CREATE TABLE IF NOT EXISTS EducationLevels(
53      id              serial          PRIMARY KEY,
54      name            varchar(255)    NOT NULL UNIQUE,
55      duration        smallint        NOT NULL
56  );
57
58  CREATE TABLE IF NOT EXISTS EducationalProgrammes(
59      id              serial          PRIMARY KEY,
60      name            varchar(255)    NOT NULL,
61      departmentID    integer         REFERENCES Departments(id) ON DELETE
            CASCADE ON UPDATE CASCADE NOT NULL,
62      levelID         integer         REFERENCES EducationLevels(id) ON
            DELETE CASCADE ON UPDATE CASCADE NOT NULL,
63      UNIQUE (name, departmentID, levelID)
64  );
65
66  CREATE TABLE IF NOT EXISTS Groups(
67      id              serial          PRIMARY KEY,
68      programmeID     integer         REFERENCES EducationalProgrammes(id)
            ON DELETE CASCADE ON UPDATE CASCADE NOT NULL,
69      groupNumber     smallint        NOT NULL UNIQUE,
70      courseLevel     smallint        NOT NULL
71  );
72
73  CREATE TABLE IF NOT EXISTS Students(
74      id              serial          PRIMARY KEY,
75      groupID         integer         REFERENCES Groups(id) ON DELETE SET
            NULL ON UPDATE CASCADE,
76      personID        integer         REFERENCES People(id) ON DELETE
            CASCADE ON UPDATE CASCADE NOT NULL UNIQUE ,
```

```
77      departmentID    integer         REFERENCES Departments(id) ON DELETE
            SET NULL ON UPDATE CASCADE
78  );
```

Удаление таблиц:

```
1  DROP TABLE IF EXISTS Students;
2  DROP TABLE IF EXISTS Groups;
3  DROP TABLE IF EXISTS EducationalProgrammes;
4  DROP TABLE IF EXISTS EducationLevels;
5  DROP TABLE IF EXISTS Departments;
6  DROP TABLE IF EXISTS Universities;
7  DROP TABLE IF EXISTS Citizenships;
8  DROP TABLE IF EXISTS Cities;
9  DROP TABLE IF EXISTS People;
10 DROP TABLE IF EXISTS Countries;
```

Заполнение тестовых значений:

```
1  INSERT INTO countries values (DEFAULT,'Russia',7);
2  INSERT INTO countries values (DEFAULT, 'United States', 1);
3  INSERT INTO countries values (DEFAULT, 'China', 86);
4
5  INSERT INTO cities values (DEFAULT, 'Saint Petersburg', 1);
6
7  INSERT INTO universities values (
8                                   DEFAULT,
9                                   'Национальный исследовательский университет ИТМО',
10                                  'https://itmo.ru/',
11                                  'abit@itmo.ru',
12                                  '+7(812)480-04-80',
13                                  1,
14                                  1
15                                 );
16 INSERT INTO people values (
17                             DEFAULT,
18                             'Павел',
19                             'Кустарев',
20                             'Валерьевич',
21                             'MALE',
22                              '+7(123)4567890',
23                             'kustarev1@ifmo.com',
24                             '1989-01-01',
25                             '1970-07-20'
26                            );
27 INSERT INTO people values (
28                             DEFAULT,
29                             'Анатолий',
```

```
30                              'Карпов',
31                              'Валерьевич',
32                              'MALE',
33                               '+7(193)4567890',
34                              'antoshka@ifmo.com',
35                              '2020-01-01',
36                              '2000-07-20'
37                             );
38
39 INSERT INTO departments values (
40                                  DEFAULT,
41                                  1,
42                                  'факультет программной инженерии и компьюте
                                       рной техники',
43                                  (SELECT id FROM people WHERE firstname =
                                       'Павел'
44                                                           AND lastname =
                                                              'Кустарев'
45                                                           AND middlename =
                                                              'Валерьевич'
46                                                         AND birthdate =
                                                            '1970-07-20')
47                                );
48 INSERT INTO educationlevels values (DEFAULT, 'Бакалавриат', 4);
49 INSERT INTO educationalprogrammes values (
50                                            DEFAULT,
51                                            'Системное и прикладное программн
                                                 ое обеспечение',
52                                            1,
53                                            1
54                                           );
55 INSERT INTO groups values (
56                              DEFAULT,
57                              1,
58                              3110,
59                              1
60                             );
61
62 INSERT INTO students values (DEFAULT, 1, 2, 1);
63
64 INSERT INTO citizenships values (1, 1);
65 INSERT INTO citizenships values (1, 2);
66
67 UPDATE citizenships SET countryid = 3 WHERE personid = 2;
68
69 DELETE FROM countries WHERE code = 1;
```

5 Вывод

В ходе выполнения дополнительного задания к лабораторной работе были изучены составление больших запросов, инфологическая модель, даталогическая модель, основы PostgreSQL.