

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



Лабораторная работа №1
по дисциплине
'Информационные системы и базы данных'
Вариант 3893

Выполнил Студент группы Р33102
Лапин Алексей Александрович
Преподаватель:
Блохина Елена Николаевна

г. Санкт-Петербург
2023г.

Содержание

1	Текст задания.	3
2	Описание предметной области.	3
3	Список сущностей и их классификацию (стержневая, ассоциация, характеристика).	4
4	Инфологическая модель	5
5	Даталогическая модель	6
6	Реализация даталогической модели на SQL.	6
7	Вывод	13

1 Текст задания.

Для выполнения лабораторной работы №1 необходимо:

1. На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.
2. Составить инфологическую модель.
3. Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.
4. Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.
5. Заполнить созданные таблицы тестовыми данными.

Для создания объектов базы данных у каждого студента есть своя схема. Название схемы соответствует имени пользователя в базе studs (sXXXXXX). Команда для подключения к базе studs:

```
psql -h pg -d studs
```

Каждый студент должен использовать свою схему при работе над лабораторной работой №1 (а также в рамках выполнения 2, 3 и 4 этапа курсовой работы).

Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Описание предметной области.
3. Список сущностей и их классификацию (стержневая, ассоциация, характеристика).
4. Инфологическая модель (ER-диаграмма в расширенном виде - с атрибутами, ключами...).
5. Даталогическая модель (должна содержать типы атрибутов, вспомогательные таблицы для отображения связей "многие-ко-многим").
6. Реализация даталогической модели на SQL.
7. Выводы по работе.

2 Описание предметной области.

Он не стал дожидаться ответа и правильно сделал. Сирэйнис даже не пошевелинулась, но он тотчас же почувствовал, что его тело перестает ему повиноваться. Сила, столкнувшаяся с его волей, оказалась куда более могущественной, чем он ожидал, и это навело его на мысль, что Сирэйнис, возможно, помогало огромное число людей. Беспомощно повлекся он обратно к дому, и на какой-то ужасный момент ему даже подумалось, что великолепный его план провалился.

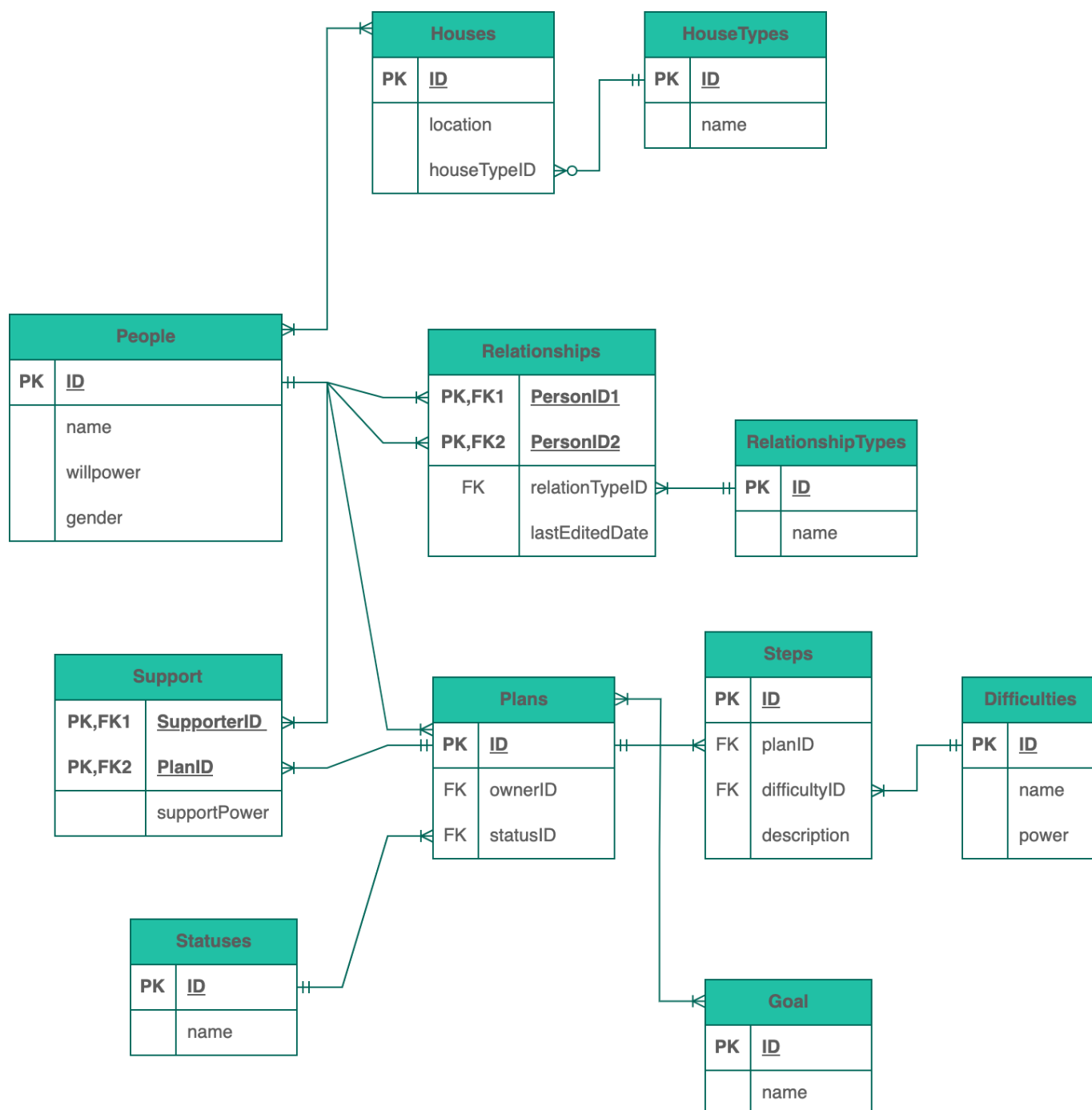
Творческий пересказ:

Люди хотят достичь некоторых целей. Для достижения целей, люди строят планы, которые состоят из определенного набора шагов разной сложности. Чтобы преодолеть сложности на своём пути люди обладают силой воли и поддержкой других людей. Так же у людей есть свой дом или несколько домов, где они живут с другими жильцами, которые могут быть им родственниками, друзьями, а порой даже врагами.

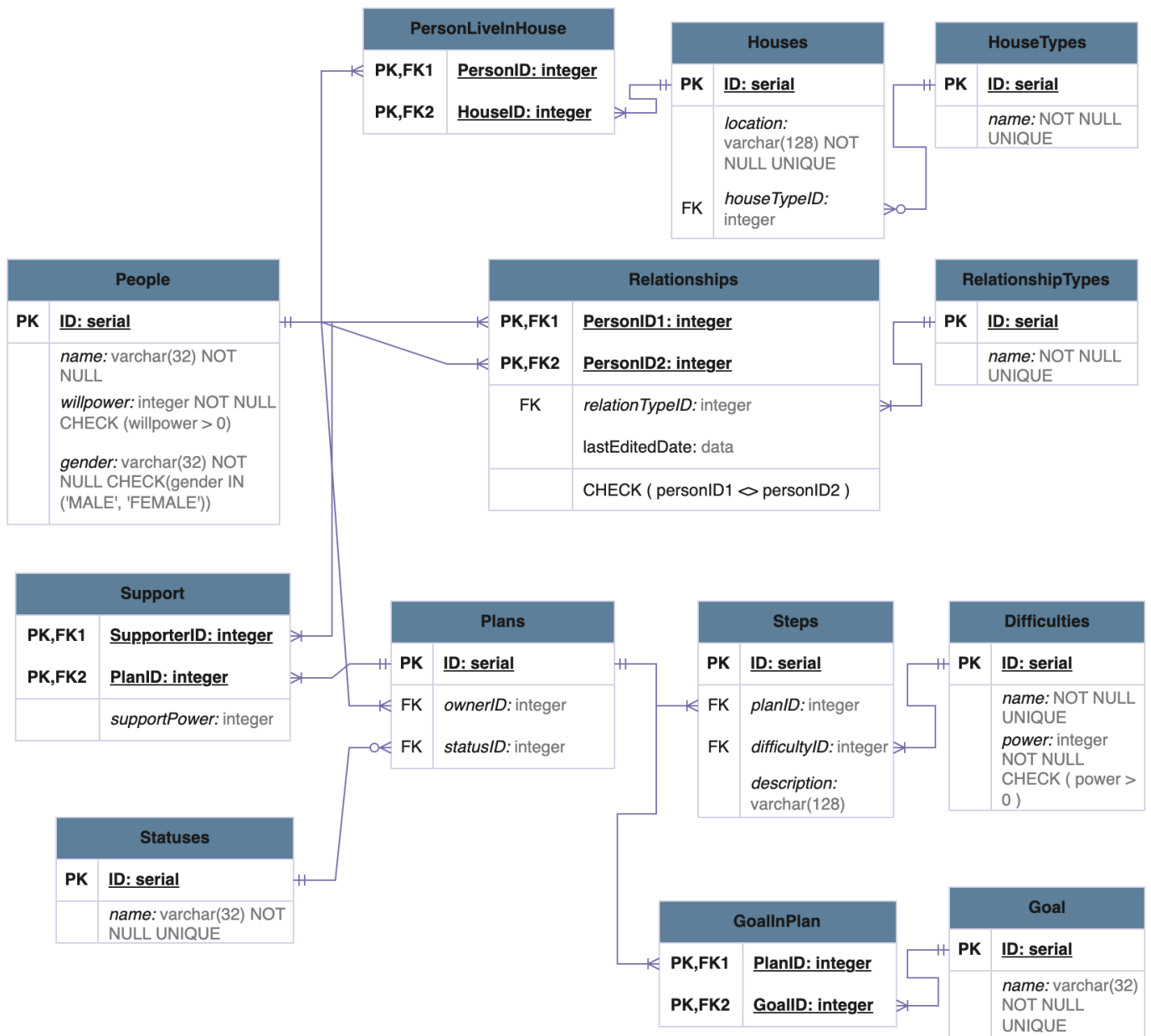
3 Список сущностей и их классификацию (стержневая, ассоциация, характеристика).

Сущность	Аттрибуты	Вид
People	id, name, willpower, gender	стержневая
Houses	id, location, houseType(Id)	стержневая
HouseTypes	id, name	характеристика
Relationships	person1(id), person2(id), relationType(id), lastEditedDate	ассоциация
RelationshipTypes	id, name	характеристика
Plans	id, owner(id), status(id)	стержневая
Steps	id, plan(id), difficulty(id), description	стержневая
Difficulties	id, name, power	характеристика
Goal	id, name	стержневая
Support	supporter(id), plan(id), supportPower	ассоциация
Statuses	id, name	характеристика

4 Инфологическая модель



5 Дatalogическая модель



6 Реализация дatalogической модели на SQL.

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',
6                                     'FEMALE'))
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,
35     PRIMARY KEY (personId1, personId2),
36     CHECK ( personId1 <> personId2 )
37 );
38
39 CREATE TABLE IF NOT EXISTS statuses (
40     id      serial      PRIMARY KEY,
41     name    varchar(32)   NOT NULL UNIQUE
42 );
43
44 CREATE TABLE IF NOT EXISTS goal (
45     id      serial      PRIMARY KEY,
46     name    varchar(128)   NOT NULL UNIQUE
47 );
48
49 CREATE TABLE IF NOT EXISTS difficulties (
50     id      serial      PRIMARY KEY,
51     name    varchar(32)   NOT NULL UNIQUE,
52     power   integer      NOT NULL CHECK ( power > 0)
53 );
54
55 CREATE TABLE IF NOT EXISTS plans (
56     id      serial      PRIMARY KEY,
57     ownerId integer      REFERENCES people(id) ON DELETE CASCADE,

```

```

58     statusId      integer      REFERENCES statuses(id) ON DELETE SET NULL
59 );
60
61 CREATE TABLE IF NOT EXISTS goalInPlan (
62     planId        integer      REFERENCES plans(id) ON DELETE CASCADE,
63     goalId        integer      REFERENCES goal(id) ON DELETE CASCADE,
64     PRIMARY KEY (planId, goalId)
65 );
66
67 CREATE TABLE IF NOT EXISTS steps (
68     id            serial        PRIMARY KEY,
69     planId        integer      REFERENCES plans(id) ON DELETE CASCADE,
70     difficultyId  integer      REFERENCES difficulties(id) ON DELETE
71         CASCADE,
72     description   varchar(128)
73 );
74
75 CREATE TABLE IF NOT EXISTS support(
76     supporterId   integer      REFERENCES people(id) ON DELETE CASCADE,
77     planId        integer      REFERENCES plans(id) ON DELETE CASCADE,
78     supportPower  integer,
79     PRIMARY KEY (supporterId, planId)

```

Удаление таблиц:

```

1 DROP TABLE IF EXISTS goalInPlan;
2 DROP TABLE IF EXISTS goal;
3 DROP TABLE IF EXISTS steps;
4 DROP TABLE IF EXISTS difficulties;
5 DROP TABLE IF EXISTS relationships;
6 DROP TABLE IF EXISTS relationshipTypes;
7 DROP TABLE IF EXISTS support;
8 DROP TABLE IF EXISTS personliveinhouse;
9 DROP TABLE IF EXISTS plans;
10 DROP TABLE IF EXISTS people;
11 DROP TABLE IF EXISTS statuses;
12 DROP TABLE IF EXISTS houses;
13 DROP TABLE IF EXISTS houseTypes;

```

Заполнение тестовых значений:

```

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

```



```

8 INSERT INTO people values (DEFAULT, 'Alexey', 2, 'MALE'); -- 8
9
10 INSERT INTO houseTypes values (DEFAULT, 'apartments') ON CONFLICT DO
    NOTHING;
11 INSERT INTO houseTypes values (DEFAULT, 'detached house') ON CONFLICT DO
    NOTHING;
12 INSERT INTO houseTypes values (DEFAULT, 'cottage') ON CONFLICT DO NOTHING;
13 INSERT INTO houseTypes values (DEFAULT, 'townhouse') ON CONFLICT DO
    NOTHING;
14
15 INSERT INTO houses values (DEFAULT, 'St. Petersburg Kronverksky Pr. 49',
    (SELECT housetypes.id FROM houseTypes WHERE name='apartments'));
16 INSERT INTO houses values (DEFAULT, 'St. Petersburg Lomonosova Street, 9',
    (SELECT housetypes.id FROM houseTypes WHERE name='detached house'));
17 INSERT INTO houses values (DEFAULT, 'Moscow Tchaykovskogo St, 11/2',
    (SELECT housetypes.id FROM houseTypes WHERE name='cottage'));
18 INSERT INTO houses values (DEFAULT, 'Moscow Grivtsova Lane, 14', (SELECT
    housetypes.id FROM houseTypes WHERE name='townhouse'));
19 INSERT INTO houses values (DEFAULT, 'Murmansk Birzhevaya Line, 14',
    (SELECT housetypes.id FROM houseTypes WHERE name='apartments'));
20
21
22 DO $$
23     DECLARE NUM_OF_PEOPLE INT = 8;
24     DECLARE NUM_OF_PLANS INT = 0;
25     DECLARE NUM_OF_STEPS INT = 0;
26 BEGIN
27     FOR i IN 2..NUM_OF_PEOPLE LOOP
28         INSERT INTO personliveinhouse VALUES (i, 1);
29     END LOOP;
30     FOR i IN 1..NUM_OF_PEOPLE LOOP
31         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)
57             THEN
58                 INSERT INTO relationships values (
59                     i,
60                     j,
61                     1,
62                     '2019-01-01'
63                 );
64             ELSIF (i != j) AND (i IN (2, 4, 1, 3) OR j IN (2, 4, 1, 3))
65                 THEN
66                     INSERT INTO relationships values (
67                         i,
68                         j,
69                         2,
70                         '2020-03-12'
71                     );
72             END IF;
73     END LOOP;
74 END LOOP;
75
76 INSERT INTO difficulties values (DEFAULT, 'easy', 100);
77 INSERT INTO difficulties values (DEFAULT, 'medium', 200);
78 INSERT INTO difficulties values (DEFAULT, 'hard', 300);
79
80 INSERT INTO statuses values (DEFAULT, 'active');
81 INSERT INTO statuses values (DEFAULT, 'done');
82 INSERT INTO statuses values (DEFAULT, 'failed');
83
84 FOR i IN 1..NUM_OF_PEOPLE LOOP
85     INSERT INTO plans values (
86         DEFAULT,
87         i,
88         1
89     );
90     NUM_OF_PLANS := NUM_OF_PLANS + 1;
91 END LOOP;
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,
132                 2,
133                 'Do it'
134             );
135         ELSIF i IN (7) THEN
136             INSERT INTO steps values (
137                 DEFAULT,
138                 i,
139                 3,
140                 'Do it'
141             );
142         ELSE
143             INSERT INTO steps values (
144                 DEFAULT,
145                 i,
146                 3,
147                 'Do something amazing'
148             );
149         END IF;
150     END LOOP;

```

```

151     END LOOP;
152
153
154     INSERT INTO goal values (
155         DEFAULT,
156         'Become a millionaire from billionaire'
157     );
158     INSERT INTO goal values (
159         DEFAULT,
160         'Graduate from university'
161     );
162     INSERT INTO goal values (
163         DEFAULT,
164         'Conquer the world'
165     );
166     INSERT INTO goal values (
167         DEFAULT,
168         'Submit the laboratory work'
169     );
170     INSERT INTO goal values (
171         DEFAULT,
172         'Go to sleep'
173     );
174
175     FOR i IN 1..NUM_OF_PLANS LOOP
176         IF i NOT IN (2, 4, 1, 3, 5, 6, 7, 8) THEN
177             INSERT INTO goalinplan values (
178                 i,
179                 (SELECT id FROM goal WHERE
180                     name='Become a millionaire
181                     from billionaire')
182             );
183         ELSIF i IN (2, 4) THEN
184             INSERT INTO goalinplan values (
185                 i,
186                 (SELECT id FROM goal WHERE
187                     name='Graduate from
188                     university')
189             );
190         ELSE
191             INSERT INTO goalinplan values (
192                 i,
193                 (SELECT id FROM goal WHERE
194                     name='Conquer the world')
195             );
196         END IF;
197     END LOOP;
198
199     FOR i IN 1..NUM_OF_PEOPLE LOOP
200         FOR j IN 1..NUM_OF_PLANS LOOP
201             IF (SELECT relationtypeid FROM relationships WHERE

```

197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213	<pre> personid1 = i AND personid2 = (SELECT ownerid FROM plans WHERE id = j)) = 1 THEN INSERT INTO support values (i, j, 100); ELSE INSERT INTO support values (i, j, -100); END IF; END LOOP; END LOOP; END \$\$ </pre>
---	---

7 Вывод

В ходе выполнения лабораторной работы были изучены сущности и их классификация, инфологическая модель, даталогическая модель, основы PostgreSQL.