*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Программная инженерия,*

*Системное и прикладное программное обеспечение*

**Лабораторная работа по №1**

по дисциплине

«Базы данных»

**Вариант №3139349**

Выполнил:

Ткачев Денис Владимирович

Группа P3111

Преподаватели:

Харитонова Анастасия Евгеньевна

Оглавление

[Задание 2](#_Toc87370215)

[Ход работы 3](#_Toc87370216)

[Текст исходной программы 3](#_Toc87370217)

[Описание программы 3](#_Toc87370218)

[Трассировка 4](#_Toc87370219)

[Вариант программы с меньшим количеством команд 4](#_Toc87370220)

[Трассировка с данными числами 5](#_Toc87370221)

[Вывод 5](#_Toc87370222)

[Список литературы 5](#_Toc87370223)

## Задание

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.

## Описание предметной области

«Роясь в барахле. **Грант** не обнаружил никакой **лодки**, но зато обнаружил в металлическом **ящике** на стене комплект **карт**, которые от сырости покоробились и покрылись **плесенью**. Грант расстелил карты на **полу** и отбросил в сторону большого **паука**. Он рассматривал их очень долго.»

Из текста можно выделить такие сущности, как: Человек Грант (Person); Лодка; Ящик, в которой затем содержались карты. У карт есть состояние, повреждение (Покорбленные, покрытые плесенью). У ящика есть материал (металлический) из которого он сделан. Грант из изначально лодку, но затем находит ящик, в которой содержится комплект карт, которые покрылись плесенью. Расстелает карты на пол, который имеет цвет и материал. Попутно в это же время грант отбрасывает в сторону большого паука, у которого есть размер, вид, местонахождение.

#### Список сущностей:

**Стержневые**:

* *Карта*
* *Человек*
* Ящик
* Лодка
* Паук

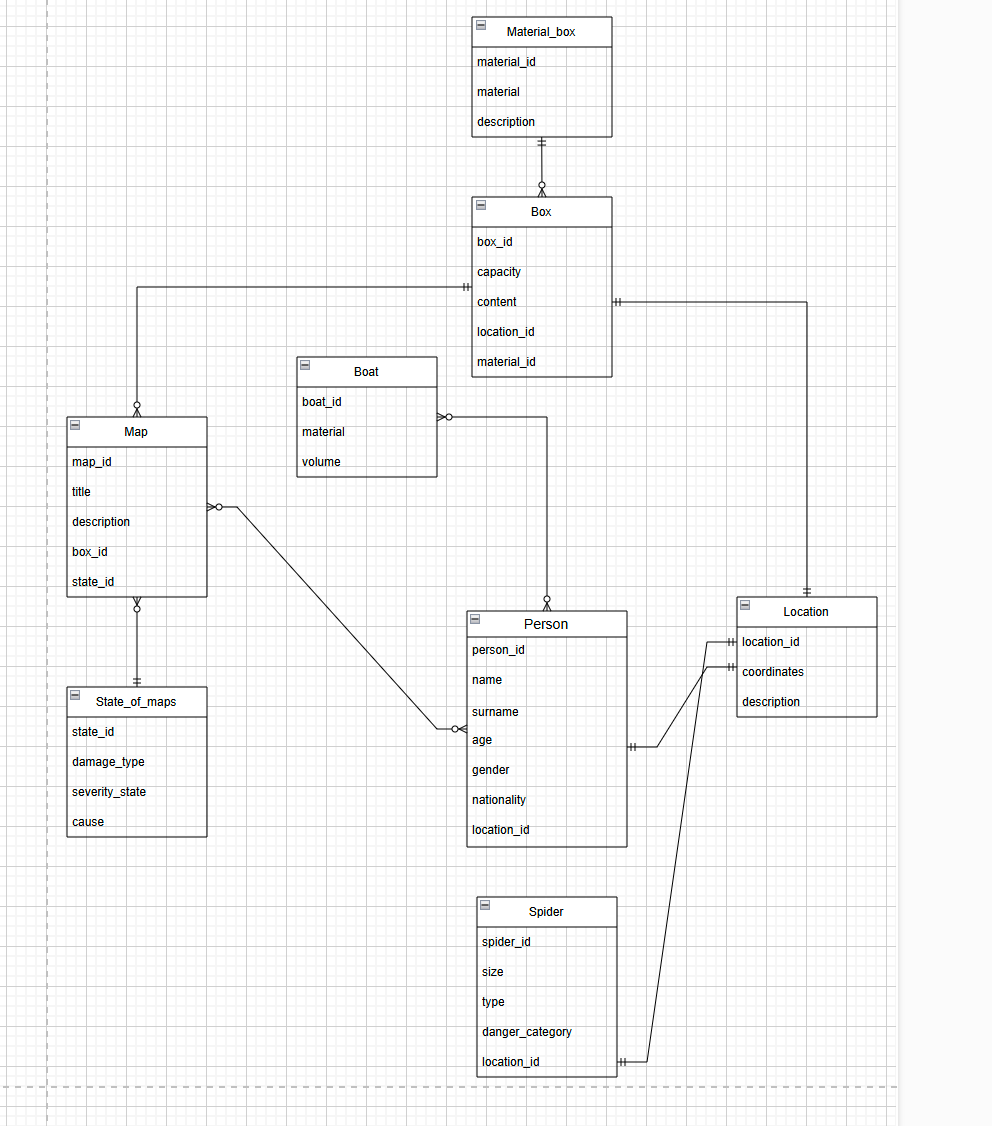
**Ассоциативные**:

* *Поиск* – Человек <-> Поиск <-> Лодка, Карты
* Нахождение в коробке (Items\_in\_box) <–> Ящик <-> Items\_in\_box <-> Карты

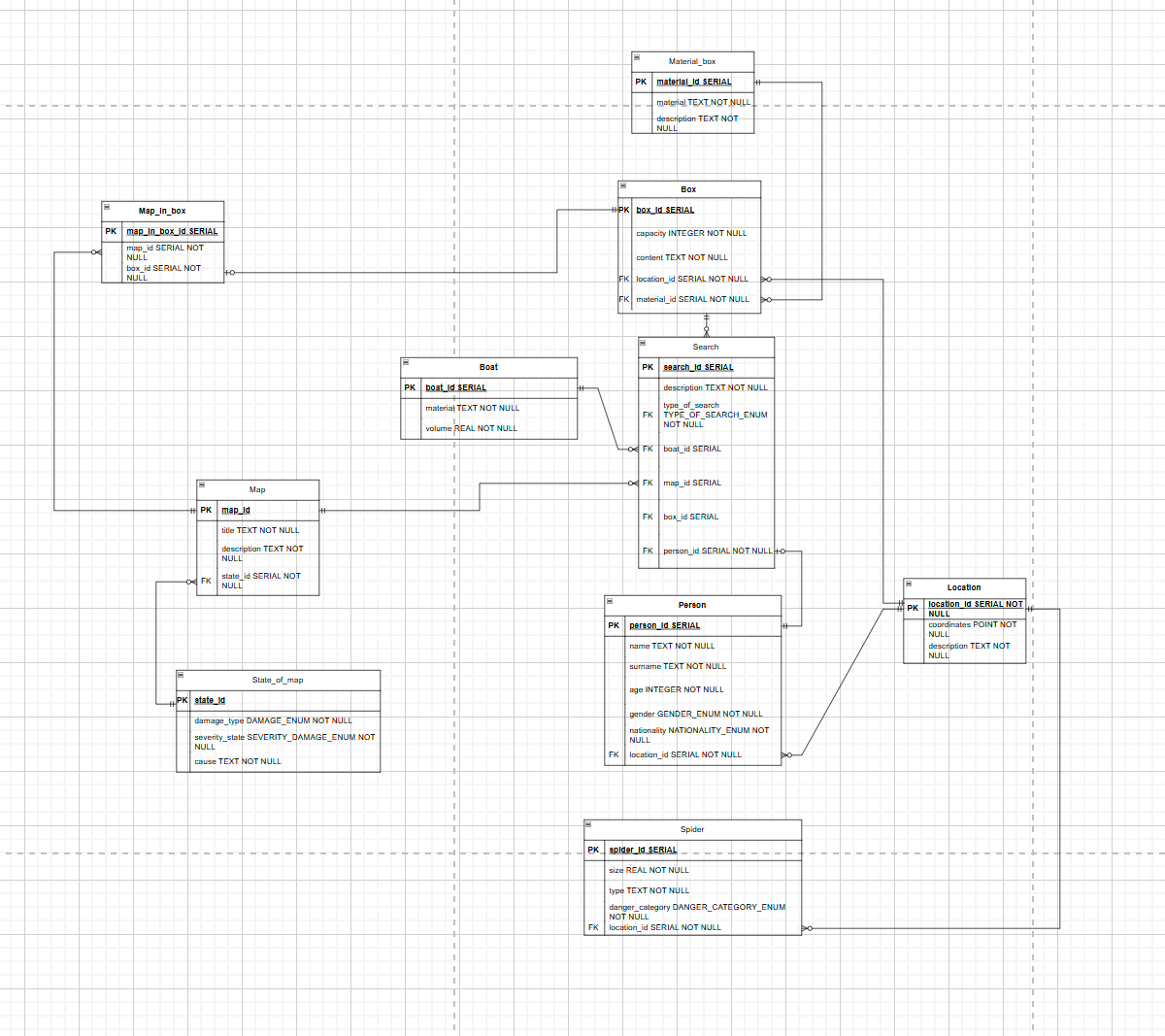
**Характеристические**:

* *Состояние карт*
* *Местоположение*
* *Материал ящика*

## **Инфологическая модель**



## **Даталогическая модель**



## **Реализация на PostgreSQL**

1. -- dropping enum types

2. DROP TYPE IF EXISTS nationality\_enum CASCADE;

3. DROP TYPE IF EXISTS danger\_category\_enum CASCADE;

4. DROP TYPE IF EXISTS damage\_enum CASCADE;

5. DROP TYPE IF EXISTS severity\_damage\_enum CASCADE;

6. DROP TYPE IF EXISTS gender\_enum CASCADE;

7. DROP TYPE IF EXISTS type\_of\_search\_enum CASCADE;

8.

9. -- dropping tables

10. DROP TABLE IF EXISTS person CASCADE;

11. DROP TABLE IF EXISTS search CASCADE;

12. DROP TABLE IF EXISTS box CASCADE;

13. DROP TABLE IF EXISTS material\_box CASCADE;

14. DROP TABLE IF EXISTS location CASCADE;

15. DROP TABLE IF EXISTS spider CASCADE;

16. DROP TABLE IF EXISTS state\_of\_map CASCADE;

17. DROP TABLE IF EXISTS map CASCADE;

18. DROP TABLE IF EXISTS boat CASCADE;

19. DROP TABLE IF EXISTS map\_in\_box CASCADE;

20.

21. -- create enums

22. CREATE TYPE nationality\_enum AS ENUM ('American', 'British', 'Canadian', 'Chinese', 'French', 'German', 'Indian', 'Japanese', 'Russian', 'Spanish');

23. CREATE TYPE danger\_category\_enum AS ENUM ('low', 'moderate', 'high', 'critical', 'unknown');

24. CREATE TYPE gender\_enum AS ENUM ('male', 'female', 'indefinite');

25. CREATE TYPE damage\_enum AS ENUM ('warped', 'moldy', 'torn', 'faded', 'stained', 'burnt', 'water\_damaged','insect\_damage', 'missing\_parts', 'nothing');

26. CREATE TYPE severity\_damage\_enum AS ENUM ('minor', 'severe', 'critical', 'intact');

27. CREATE TYPE type\_of\_search\_enum AS ENUM ('boat', 'map', 'box');

28.

29. -- create tables

30.

31. CREATE TABLE location (

32.     location\_id SERIAL PRIMARY KEY,

33.     coordinates POINT NOT NULL,

34.     description TEXT NOT NULL

35. );

36.

37. CREATE TABLE person (

38.     person\_id SERIAL PRIMARY KEY,

39.     name TEXT NOT NULL,

40.     surname TEXT NOT NULL,

41.     age INTEGER NOT NULL CHECK (age > 0),

42.     gender gender\_enum NOT NULL,

43.     nationality nationality\_enum NOT NULL,

44.     location\_id INTEGER NOT NULL REFERENCES location(location\_id) ON DELETE SET NULL

45. );

46.

47. CREATE TABLE material\_box (

48.     material\_id SERIAL PRIMARY KEY,

49.     description TEXT NOT NULL

50. );

51.

52. CREATE TABLE box (

53.     box\_id SERIAL PRIMARY KEY,

54.     capacity INTEGER NOT NULL CHECK (capacity > 0),

55.     content TEXT NOT NULL,

56.     location\_id SERIAL NOT NULL REFERENCES location(location\_id),

57.     material\_id SERIAL NOT NULL REFERENCES material\_box(material\_id)

58. );

59.

60. CREATE TABLE state\_of\_map (

61.     state\_id SERIAL PRIMARY KEY,

62.     damage\_type damage\_enum NOT NULL,

63.     severity\_state severity\_damage\_enum NOT NULL,

64.     cause TEXT NOT NULL

65. );

66.

67. CREATE TABLE map (

68.     map\_id SERIAL PRIMARY KEY,

69.     description TEXT NOT NULL,

70.     box\_id SERIAL NOT NULL REFERENCES box(box\_id),

71.     state\_id SERIAL NOT NULL REFERENCES state\_of\_map(state\_id)

72. );

73.

74. CREATE TABLE map\_in\_box (

75.     map\_in\_box\_id SERIAL PRIMARY KEY,

76.     map\_id SERIAL NOT NULL REFERENCES map(map\_id),

77.     box\_id SERIAL NOT NULL REFERENCES box(box\_id)

78. );

79.

80. CREATE TABLE boat (

81.     boat\_id SERIAL PRIMARY KEY,

82.     material TEXT NOT NULL,

83.     volume REAL NOT NULL CHECK(volume > 0)

84. );

85.

86. CREATE TABLE search (

87.     search\_id SERIAL PRIMARY KEY,

88.     description TEXT NOT NULL,

89.     type\_of\_search type\_of\_search\_enum NOT NULL,

90.     map\_id INTEGER REFERENCES map(map\_id) ON DELETE SET NULL,

91.     boat\_id INTEGER REFERENCES boat(boat\_id) ON DELETE SET NULL,

92.     box\_id INTEGER REFERENCES box(box\_id) ON DELETE SET NULL,

93.     person\_id INTEGER NOT NULL REFERENCES person(person\_id) ON DELETE CASCADE

94. );

95.

96. CREATE TABLE spider (

97.     spider\_id SERIAL PRIMARY KEY,

98.     size REAL NOT NULL CHECK(size > 0 AND size < 5),

99.     type TEXT NOT NULL,

100.     danger\_category danger\_category\_enum NOT NULL,

101.     location\_id SERIAL NOT NULL REFERENCES location(location\_id)

102. );

103.

104. -- INSERT INTO tables

105.

106. INSERT INTO location (coordinates, description) VALUES (POINT(0.0, 0.0), 'Центр огромной тёмной комнаты');

107.

108. INSERT INTO location (coordinates, description) VALUES (POINT(3.0, 5.0), 'Висит на стене огромной тёмной комнаты');

109.

110. INSERT INTO location (coordinates, description) VALUES (POINT(3.2, 5.5), 'Рядом с ящиком на стене висит павутина');

111.

112. INSERT INTO person (name, surname, age, gender, nationality, location\_id) VALUES ('Ivan', 'Ivanov', 10, 'male', 'Russian', 1);

113. INSERT INTO person (name, surname, age, gender, nationality, location\_id) VALUES ('Petr', 'Petrov', 18, 'male', 'Russian', 3);

114.

115. INSERT INTO material\_box (description) VALUES ('Алюминиевый сплав');

116.

117. INSERT INTO box (capacity, content, location\_id, material\_id) VALUES ('1200', 'Карты, книги, монеты', 2, 1);

118.

119. INSERT INTO state\_of\_map (damage\_type, severity\_state, cause) VALUES ('moldy', 'severe', 'Долго лежала в сыром месте');

120.

121. INSERT INTO map (description, box\_id, state\_id) VALUES ('Карта сокровищ', 1, 1);

122.

123. INSERT INTO map\_in\_box (map\_id, box\_id) VALUES (1, 1);

124.

125. INSERT INTO boat (material, volume) VALUES ('Wood', 2500);

126.

127. INSERT INTO search (description, type\_of\_search, boat\_id, person\_id) VALUES ('Ищу лодку, затем нашёл коробку, а в ней карту', 'boat', 1, 1);

128. INSERT INTO search (description, type\_of\_search, box\_id, person\_id) VALUES ('Ищу ящик с id = 1', 'box', 1, 3);

129.

130. INSERT INTO spider (size, type, danger\_category, location\_id) VALUES (4, 'Большой павук с размером с длинным пальцем', 'high', 3);

131.

## Вывод

В ходе проделанной лабораторной работы были изучены виды представления реляционной базы данных, такие как инфологическая и даталогическая модели. На основе предметной области была построена собственная база данных с разными классами сущностей и разными видами связей между ними. Кроме того, были изучены основы языка запросов SQL.