

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ
НАПРАВЛЕНИЕ ПРОГРАММНАЯ ИНЖЕНЕРИЯ
ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА СИСТЕМНОЕ И ПРИКЛАДНОЕ
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

курса «Базы данных»

Вариант № 1909

Выполнил студент:

Шубин Егор Вячеславович

группа: Р3109

Преподаватель:

Лектор: Николаев Владимир
Вячеславович,

Практик: Воронина Дарья Сергеевна

Санкт-Петербург, 2025 г.

Содержание

Лабораторная работа № 3.	3
1. Задание варианта № 1909	3
2. Выполнение задания:	4
1. Инфологическая модель:	4
2. Описание функциональных зависимостей:	4
3. Нормализация:	5
4. Нормализованная модель:	5
5. Денормализация	6
6. Функция и триггер на языке PL/pgSQL	7
3. Вывод	8

Лабораторная работа № 3

1. Задание варианта № 1909

Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

- Опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
- Приведите отношения в 3NF (как минимум). Постройте схему на основеNF (как минимум).
- Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основеNF; Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF. Если ваша схема находится уже в BCNF, докажите это;
- Какие денормализации будут полезны для вашей схемы? Приведите подробное описание.

Придумайте триггер и связанную с ним функцию, относящиеся к вашей предметной области, согласуйте их с преподавателем и реализуйте на языке PL/pgSQL.

2. Выполнение задания:

2. 1. Инфологическая модель:

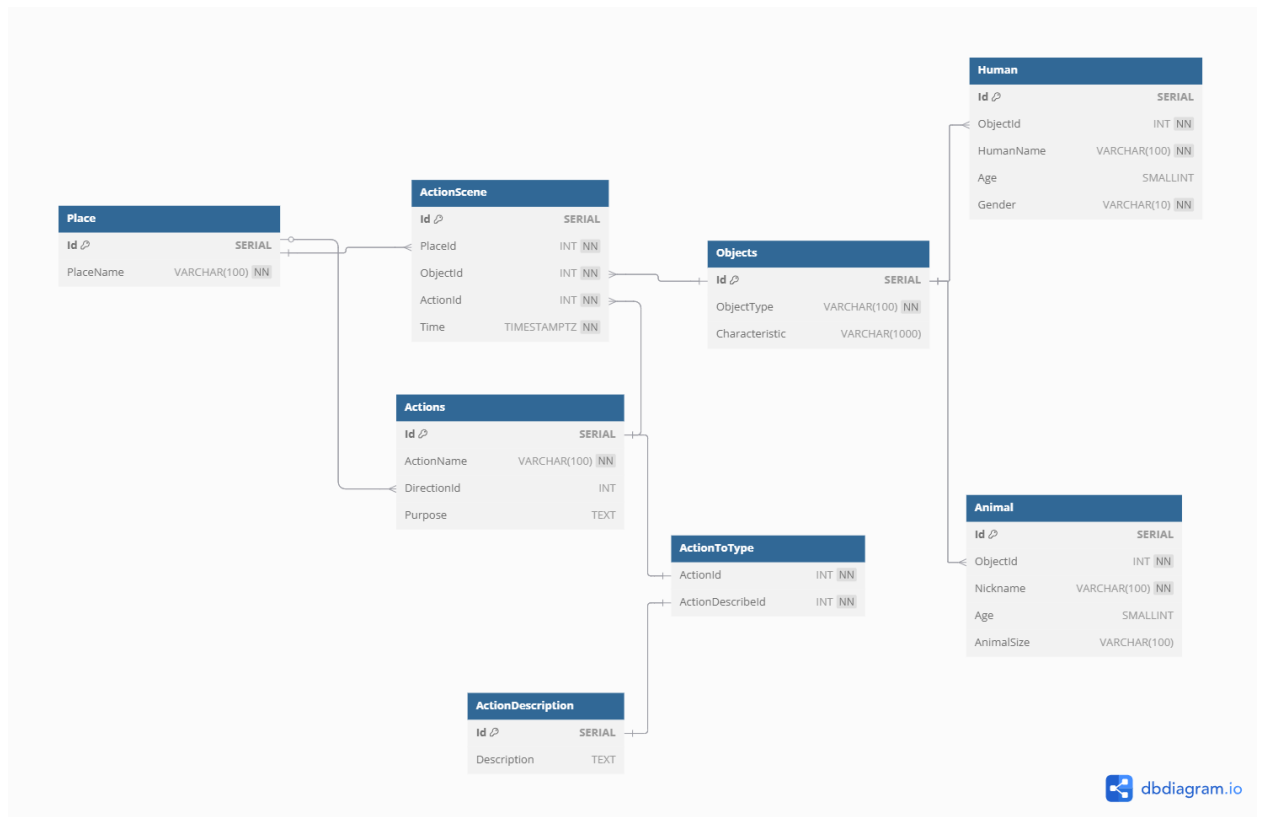


Рис. 1.1: Инфологическая ненормализованная модель

2. 2. Описание функциональных зависимостей:

- ActionScene: $id \rightarrow (PlaceId, ObjectId, ActionId)$
- Place: $id \rightarrow PlaceName$
- Objects: $id \rightarrow ObjectType, Characteristic$
- Actions: $id \rightarrow (ActionName, DirectionId, Purpose)$
- ActionDescription: $id \rightarrow Description$
- ActionToType: $(ActionId, ActionDescribeId) \rightarrow \emptyset$
- Animal: $id \rightarrow (ObjectId, Nickname, Age, AnimalSize)$
- Human: $id \rightarrow (ObjectId, HumanName, Age, Gender)$

2. 3. Нормализация:

- 1NF: Отношения находятся в 1НФ, т.к. все атрибуты являются простыми. Все домены содержат только скалярные значения, повторяющихся доменов в таблице нет.
- 2NF: Отношение находится во 2НФ, т.к. каждый не ключевой атрибут неприводимо зависит от Первичного Ключа(ПК).
- 3NF: В сущности Objects атрибут ObjectType может относиться к разным кортежам в таблице, например значение атрибута "Человек" может относиться к нескольким объектам, нужно вынести ObjectType в отдельную сущность.
- BCNF: Отношение удовлетворяет данной форме, т.к. у сущностей нет составных ключей, у каждой сущности единственный суперключ.

2. 4. Нормализованная модель:

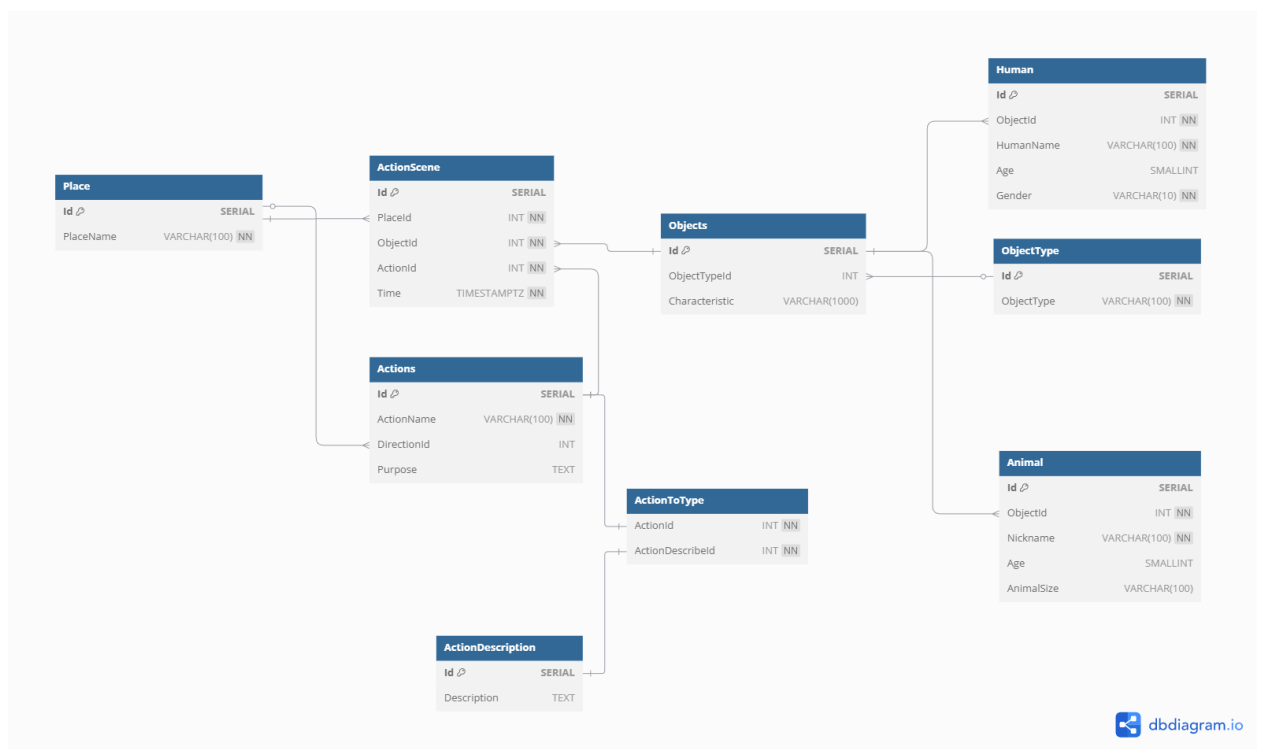


Рис. 1.2: Нормализованная модель

2. 5. Денормализация

Сущности можно денормализовать путем упрощения и соединения нескольких сущностей для того чтобы упростить запросы и увеличить производительность за счет уменьшения JOIN-запросов. Например можно не выносить ObjectType или ActionDescription в отдельную сущность.

2. 6. Функция и триггер на языке PL/pgSQL

```
1 CREATE OR REPLACE FUNCTION trigger_function_calculate_percentage_of_time()
2 RETURNS TRIGGER AS
3 $$
4 DECLARE
5     time_fraction NUMERIC := EXTRACT(EPOCH FROM NEW.EndTime - NEW.StartTime);
6 BEGIN
7     IF (time_fraction <= 0) THEN
8         RAISE EXCEPTION 'Duration of action must be more then % seconds!',
9         time_fraction;
10    END IF;
11    IF TG_OP = 'UPDATE' THEN
12        PERFORM calculate_percentage_of_time(roww.ObjectId, roww.PlaceId)
13        FROM (SELECT * FROM Stats WHERE ObjectId = NEW.ObjectId OR PlaceId =
14        NEW.PlaceId) roww;
15    END IF;
16
17    PERFORM calculate_percentage_of_time(NEW.ObjectId, NEW.PlaceId);
18
19    PERFORM calculate_percentage_of_time(NEW.ObjectId, rec.PlaceId)
20    FROM (SELECT DISTINCT PlaceId FROM Stats WHERE ObjectId = NEW.ObjectId)
21    rec;
22
23    RETURN NEW;
24 END;
25 $$ LANGUAGE plpgsql;
26
27 CREATE OR REPLACE FUNCTION calculate_percentage_of_time(
28     Object INT,
29     Place INT
30 ) RETURNS VOID AS
31 $$
32 DECLARE
33     total_time_seconds NUMERIC;
34     place_time_seconds NUMERIC;
35     percentage NUMERIC;
36 BEGIN
37     SELECT EXTRACT(EPOCH FROM SUM(EndTime - StartTime))
38     INTO total_time_seconds
39     FROM ActionScene
40     WHERE ObjectId = Object;
41
42     SELECT EXTRACT(EPOCH FROM SUM(EndTime - StartTime))
43     INTO place_time_seconds
44     FROM ActionScene
45     WHERE ObjectId = Object AND PlaceId = Place;
46
47     percentage := (place_time_seconds / total_time_seconds) * 100;
48
49     INSERT INTO Stats(ObjectId, PlaceId, Fraction)
50     VALUES (
51         Object,
52         Place,
53         substring((percentage::TEXT) FROM '\d+\.\d\d') || ' %'
54     )
55     ON CONFLICT (ObjectId, PlaceId)
56     DO UPDATE SET Fraction = EXCLUDED.Fraction;
57 END;
58 $$ LANGUAGE plpgsql;
```

```

1 CREATE OR REPLACE TRIGGER calculate_percentage_of_time_trigger_on_insert
2 AFTER INSERT ON ActionScene
3 FOR EACH ROW
4 EXECUTE FUNCTION trigger_function_calculate_percentage_of_time();
5
6 CREATE OR REPLACE TRIGGER calculate_percentage_of_time_trigger_on_update
7 AFTER UPDATE OF StartTime, EndTime, PlaceId, ObjectId ON ActionScene
8 FOR EACH ROW
9 WHEN (
10     (NEW.EndTime != OLD.EndTime) OR
11     (NEW.StartTime != OLD.StartTime) OR
12     (NEW.PlaceId != OLD.PlaceId) OR
13     (NEW.ObjectId != OLD.ObjectId)
14 )
15 EXECUTE FUNCTION trigger_function_calculate_percentage_of_time();

```

3. Вывод

В ходе лабораторной работы была проведена нормализация схемы базы данных до 3NF и BCNF, что устранило избыточность данных и аномалии обновления. Разработаны триггеры на PL/pgSQL для автоматического расчета статистики времени действий, обеспечивающие целостность данных. Выполнена частичная денормализация для оптимизации частых запросов, что улучшило производительность без потери согласованности данных.