

Национальный исследовательский университет ИТМО, г. Санкт-Петербург

Факультет ПиИКТ



Информационные системы и базы данных

Лабораторная работа № 2

Вариант: 874803

Работу выполнили:

Велюс Арина Костас

Группа: № Р33151

Преподаватель:

Байрамова Хумай Бахруз Кызы

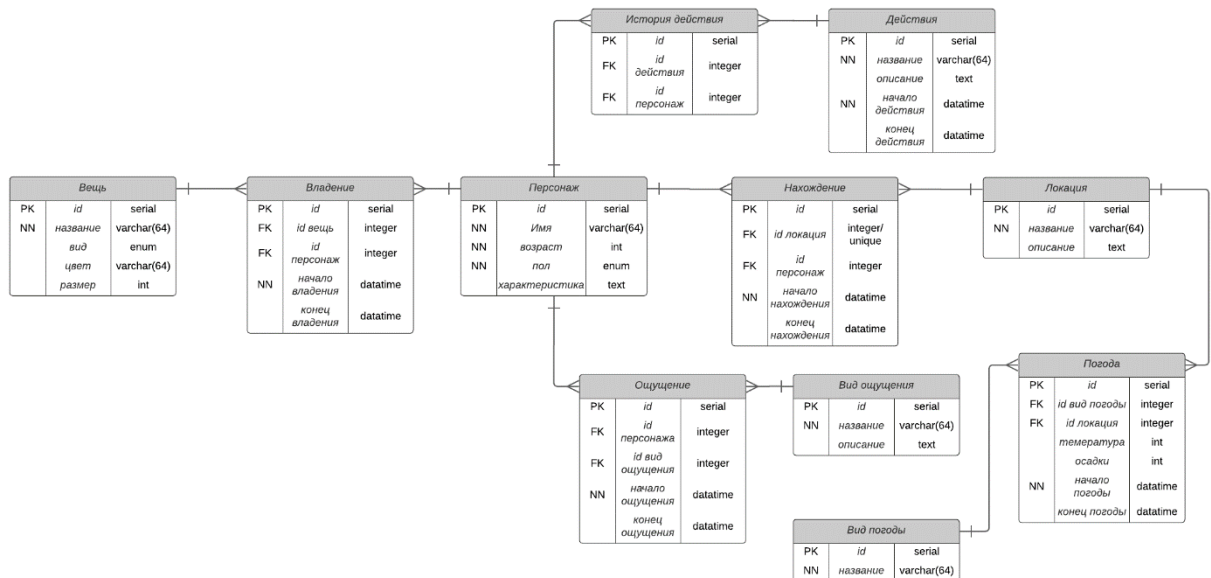
г. Санкт-Петербург

2023

Задание:

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 основе 3NF (как минимум). Постройте схему на основе полученных отношений;
- опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе 3NF;
- преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;
- какие денормализации будут полезны для вашей схемы? Приведите подробное описание;



Функциональные зависимости:

Таблица person:

id -> name_person, age, gender, characteristics

Таблица thing:

id -> name_thing, type_thing, color, size_thing

Таблица action:

id -> name_action, description_action, start_time, end_time

Таблица location:

id -> name_location, description_location

Таблица type_weather:

id -> name_weather

Таблица type_feeling:

id -> name_feeling, description_feeling

Таблица weather:

id -> type_weather_id, location_id, temperature, precipitation, start_time, end_time

Таблица feeling:

id -> person_id, type_feeling_id, start_time, end_time

Таблица history_actions:

id -> action_id, person_id

Таблица presence:

id -> person_id, location_id, start_time, end_time

Таблица possession:

id -> thing_id, person_id, start_time, end_time

1 нормальная форма

Условие:

1. Отношение, на пересечении каждой строки и столбца - одно значение.

У полученной модели все значения атрибутов всех сущностей не больше одного значения на пересечении строк и столбцов => модель находится в первой нормальной форме

2 нормальная форма

Условие:

1. 1НФ.
2. Атрибуты, не входящие в первичный ключ, в полной функциональной зависимости от первичного ключа отношений.

Полная функциональная зависимость: поле A2 находится в полной функциональной зависимости от составного поля A1, если A1 -> A2 (A2 функциональной зависит от A1), но не зависит от подмножеств поля A1.

! Все неключевые атрибуты должны зависеть от первичного ключа. Если первичный ключ составной, то мы не можем исключить один из атрибутов, иначе мы теряем функциональную зависимость

Доказательство:

Детерминантом всех функциональных зависимостей является один атрибут. Значит модель находится в 2НФ.

3 нормальная форма

Условие:

1. 1НФ.
2. 2НФ
3. Не имеет атрибутов, которые не входят в первичный ключ и не находятся в транзитивной функциональной зависимости от первичного ключа.

Транзитивность: $A1 \rightarrow A2$ и $A2 \rightarrow A3$, то $A1 \rightarrow A3$ ($A3$ транзитивно зависит от $A1$)

Доказательство:

В множестве зависимостей нет транзитивных функциональных зависимостей, следовательно, модель находится в 3НФ. (неключевого поле не зависит от другого неключевого поля)

Нормальная форма Бойса-Кодда

Условие:

1. 1НФ, 2НФ, 3НФ
2. Каждый детерминант отношения — потенциальный ключ

Доказательство:

Так как в множестве отношений нет составных первичных ключей и таблицы находятся в 3НФ, то они находятся и в НФБК.

Денормализация

1. Мы можем объединить таблицы «Нахождение» и «Ощущение», если нам часто требуется информация о местоположении и чувствах человека. Объединение может упростить запросы. (Нарушение 2НФ)

```
CREATE TABLE presence_and_feeling (  
    id serial PRIMARY KEY,  
    person_id integer,  
    FOREIGN KEY (person_id) REFERENCES person(id),  
    location_id integer,  
    FOREIGN KEY (location_id) REFERENCES location(id),  
    start_time TIMESTAMP NOT NULL,
```

```

end_time TIMESTAMP,
type_feeling_id integer,
FOREIGN KEY (type_feeling_id) REFERENCES type_feeling(id),
feeling_start_time TIMESTAMP NOT NULL,
feeling_end_time TIMESTAMP
);

```

Запрос – вернуть все записи о нахождении и ощущении для персонажа с id = 1

Запрос ДЮ:

```

SELECT
    person.id AS person_id,
    person.name_person,
    presence.id AS presence_id,
    presence.location_id,
    presence.start_time AS presence_start_time,
    presence.end_time AS presence_end_time,
    feeling.id AS feeling_id,
    feeling.type_feeling_id,
    feeling.start_time AS feeling_start_time,
    feeling.end_time AS feeling_end_time
FROM person
LEFT JOIN presence ON person.id = presence.person_id
LEFT JOIN feeling ON person.id = feeling.person_id
WHERE person.id = 1;

```

```

----- QUERY PLAN -----
Nested Loop Left Join  (cost=0.15..64.59 rows=1 width=198) (actual time=0.029..0.032 rows=1 loops=1)
  Join Filter: (person.id = feeling.person_id)
  -> Nested Loop Left Join  (cost=0.15..36.38 rows=1 width=174) (actual time=0.017..0.018 rows=1 loops=1)
    Join Filter: (person.id = presence.person_id)
    -> Index Scan using person_pkey on person  (cost=0.15..8.17 rows=1 width=150) (actual time=0.006..0.007 rows=1 loops=1)
        Index Cond: (id = 1)
    -> Seq Scan on presence  (cost=0.00..28.12 rows=7 width=28) (actual time=0.007..0.008 rows=1 loops=1)
        Filter: (person_id = 1)
        Rows Removed by Filter: 1
  -> Seq Scan on feeling  (cost=0.00..28.12 rows=7 width=28) (actual time=0.010..0.011 rows=1 loops=1)
        Filter: (person_id = 1)
        Rows Removed by Filter: 1
Planning Time: 1.674 ms
Execution Time: 0.105 ms
(14 строк)

```

Запрос ПОСЛЕ:

```

SELECT presence_and_feeling.*
FROM presence_and_feeling
WHERE presence_and_feeling.person_id = 1;

```

```

----- QUERY PLAN -----
Seq Scan on presence_and_feeling  (cost=0.00..23.38 rows=5 width=48) (actual time=0.005..0.005 rows=0 loops=1)
  Filter: (person_id = 1)
Planning Time: 0.278 ms
Execution Time: 0.035 ms
(4 строки)

```

2. Мы можем объединить таблицы «Владение» и «История действий», если нам часто требуется информация о действиях и вещах персонажа, которыми он владеет. (Нарушение 2НФ)

```

CREATE TABLE possession_and_actions (
    id serial PRIMARY KEY,
    thing_id integer,
    FOREIGN KEY (thing_id) REFERENCES thing(id),
    person_id integer,
    FOREIGN KEY (person_id) REFERENCES person(id),
    start_time TIMESTAMP NOT NULL,
    end_time TIMESTAMP,
    action_id integer,
    FOREIGN KEY (action_id) REFERENCES action(id)
);

```

Запрос – вернуть все владения и действия для персонажа с id = 1

Запрос ДЮ:

```

SELECT
    person.id AS person_id,
    person.name_person,
    possession.id AS possession_id,
    possession.start_time AS possession_start_time,
    possession.end_time AS possession_end_time,
    thing.id AS thing_id,
    thing.name_thing,
    thing.type_thing,
    thing.color,
    thing.size_thing,
    history_actions.id AS history_action_id,
    history_actions.action_id,
    action.name_action,
    action.description_action,
    action.start_time AS action_start_time,
    action.end_time AS action_end_time
FROM person
LEFT JOIN possession ON person.id = possession.person_id
LEFT JOIN thing ON possession.thing_id = thing.id
LEFT JOIN history_actions ON person.id = history_actions.person_id
LEFT JOIN action ON history_actions.action_id = action.id
WHERE person.id = 1;

```

```

QUERY PLAN
-----
Nested Loop Left Join (cost=0.44..80.56 rows=1 width=676) (actual time=0.047..0.051 rows=1 loops=1)
-> Nested Loop Left Join (cost=0.29..76.77 rows=1 width=482) (actual time=0.030..0.033 rows=1 loops=1)
    Join Filter: (person.id = history_actions.person_id)
    -> Nested Loop Left Join (cost=0.29..41.15 rows=1 width=474) (actual time=0.025..0.027 rows=1 loops=1)
        -> Nested Loop Left Join (cost=0.15..36.38 rows=1 width=174) (actual time=0.019..0.020 rows=1 loops=1)
            Join Filter: (person.id = possession.person_id)
            -> Index Scan using person_pkey on person (cost=0.15..8.17 rows=1 width=150) (actual time=0.007..0.007 rows=1 loops=1)
                Index Cond: (id = 1)
            -> Seq Scan on possession (cost=0.00..28.12 rows=7 width=28) (actual time=0.009..0.010 rows=1 loops=1)
                Filter: (person_id = 1)
                Rows Removed by Filter: 1
        -> Index Scan using thing_pkey on thing (cost=0.14..4.73 rows=1 width=304) (actual time=0.004..0.004 rows=1 loops=1)
            Index Cond: (id = possession.thing_id)
    -> Seq Scan on history_actions (cost=0.00..35.50 rows=10 width=12) (actual time=0.004..0.005 rows=1 loops=1)
        Filter: (person_id = 1)
        Rows Removed by Filter: 1
-> Index Scan using action_pkey on action (cost=0.15..3.76 rows=1 width=198) (actual time=0.015..0.015 rows=1 loops=1)
    Index Cond: (id = history_actions.action_id)
Planning Time: 2.125 ms
Execution Time: 0.177 ms
(20 строк)

```

Запрос ПОСЛЕ:

```

SELECT possession_and_actions.*
FROM possession_and_actions
WHERE possession_and_actions.person_id = 1;

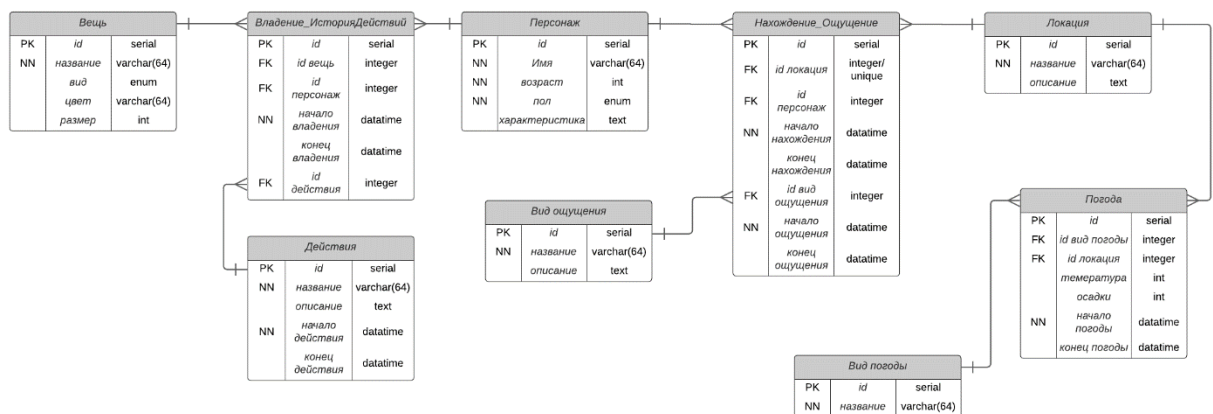
```

```

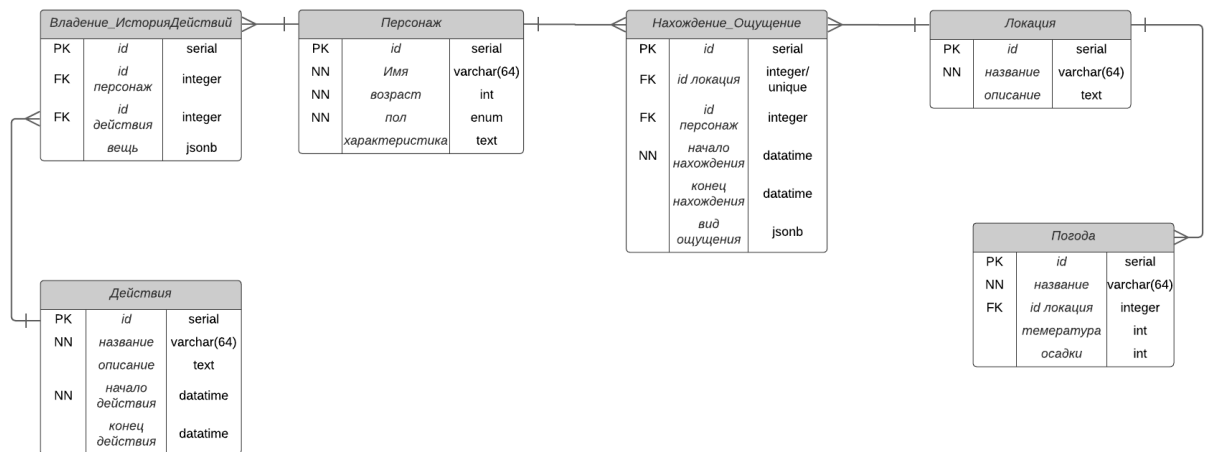
QUERY PLAN
-----
Seq Scan on possession_and_actions (cost=0.00..27.00 rows=7 width=32) (actual time=0.005..0.006 rows=0 loops=1)
    Filter: (person_id = 1)
Planning Time: 0.094 ms
Execution Time: 0.028 ms
(4 строки)

```

Даталогическая модель после денормализации:



Также мы можем нарушить 1НФ. Используя тип jsonb, мы избавляемся от «Вид ощущения» и «Вещи», сделав их jsonb-атрибутами во «Владение-ИсторияДействий» и «Нахождение_Ощущение».



Вывод:

Выполняя данную лабораторную работу, я узнала о 3НФ и BCNF. Также, с помощью нормализации проверила свою модель и убедилась в её эффективности. Мы можем быть уверены, что данные целостны и не избыточны.