



Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Отчёт по лабораторной работе 3

Предмет: БД

Варинат: 19191919

Выполнил: студент группы

Р3115 Суворов Станислав Денисович

Проверил: Райла Мартин

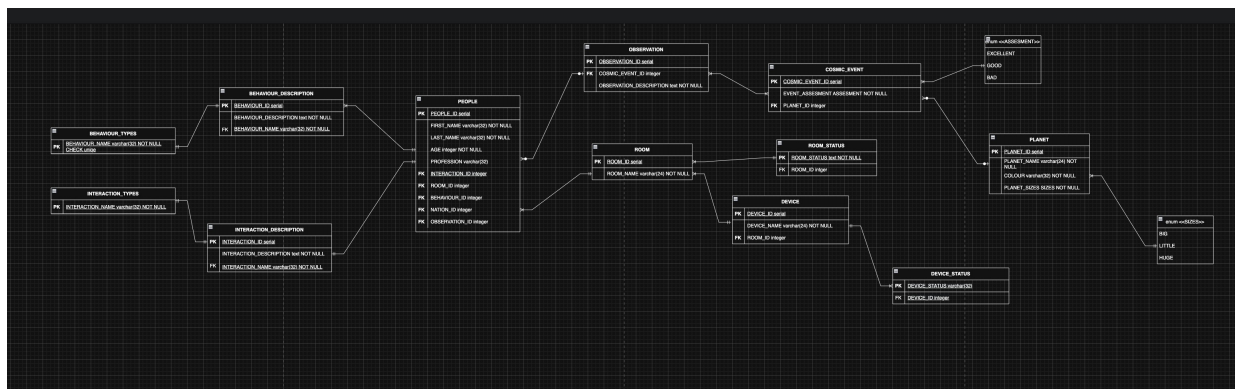
Задание

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 1NF (как минимум).
- Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе 1NF;
- 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.

Даталогическая модель



Описание функциональных зависимостей

- PEOPLE_ID → FIRST_NAME
- PEOPLE_ID → LAST_NAME
- PEOPLE_ID → AGE
- PEOPLE_ID → PROFESSION
- PEOPLE_ID → INTERACTION_ID
- PEOPLE_ID → ROOM_ID
- PEOPLE_ID → BEHAVIOUR_ID
- PEOPLE_ID → NATION_ID
- PEOPLE_ID → OBSERVATION_ID

People

1НФ: Атрибуты атомарны, не содержат составных значений, также содержит первичный ключ, сущность находится в 1НФ.

2НФ: Нет составного РК, поэтому неключевые атрибуты находятся в полной зависимости от РК, и таблица уже находится в 1НФ => таблица находится в 2НФ.

3НФ: Отсутствуют транзитивные зависимости => таблица находится в 3НФ.
НФБК: Соблюдается тк, в нашей таблице суперключом является PEOPLE_ID – от него идут все функциональные зависимости, таблица находится в НФБК.

Behaviour_Types

PK Behaviour_name

1НФ: Атрибут атомарен, не содержит составных значений, первичный ключ есть. Таблица находится в 1НФ

2НФ: нет составного РК, поэтому неключевые атрибуты находятся в полной зависимости от РК.

Находится в 2НФ.

3НФ: Тк не содержит неключевых атрибутов значит нет транзитивных зависимостей.

Находится в 3НФ.

НФБК: Такая же причина

INTERACTION_TYPES

PK INTERACTION_NAME

1НФ: Атрибут атомарен, не содержит составных значений, первичный ключ есть. Таблица находится в 1НФ

2НФ: Нет составного РК, поэтому неключевые атрибуты находятся в полной зависимости от РК. Находится в 2НФ

3НФ: Тк не содержит неключевых атрибутов значит нет транзитивных зависимостей.

Находится в 3НФ.

НФБК: Такая же причина

Behaviours_Description

Функциональные зависимости

- BEHAVIOUR_ID → BEHAVIOUR_DESCRIPTION
- BEHAVIOUR_ID → BEHAVIOUR_NAME

1НФ: Атрибуты атомарны, не содержат составных значений, есть первичный ключ. Находится в 1НФ.

2НФ: Нет составного РК, поэтому неключевые атрибуты находятся в полной зависимости от РК. Находится в 2НФ.

3НФ: Нет связи между атрибутами, каждый зависит только от ID. Следовательно транзитивных зависимостей – нет. Таблица в 3НФ

НФБК: Соблюдается тк. нет зависимостей где ID не ключ.

INTERACTION_DESCRIPTION

- INTERACTION_ID → INTERACTION_DESCRIPTION
- INTERACTION_ID → INTERACTION_NAME

1НФ: Атрибуты атомарны, не содержат составных значений, есть первичный ключ. Находится в 1НФ.

2НФ: Нет составного РК, поэтому неключевые атрибуты находятся в полной зависимости от РК. Находится в 2НФ.

3НФ: Нет связи между атрибутами, каждый зависит только от ID. Следовательно транзитивных зависимостей – нет. Таблица в 3НФ

НФБК: Соблюдается тк. нет зависимостей где ID не ключ.

OBSERVATION

- $OBSERVATION_ID \rightarrow COSMIC_EVENT_ID$
- $OBSERVATION_ID \rightarrow OBSERVATION_DESCRIPTION$
- **1НФ:** Атрибуты атомарны. Не содержат составных значений, есть РК. Находится в 1НФ.
- **2НФ:** РК простой. Неключевые атрибуты полностью зависят от него. . Находится в 2НФ.
- **3НФ:** Неключевые атрибуты зависят только от РК. Нет зависимостей между неключевыми атрибутами. Следовательно, транзитивных зависимостей нет. Таблица в 3НФ.
- **НФБК:** Соблюдается тк. нет зависимостей где ID не ключ.

Cosmic_Event

- $COSMIC_EVENT_ID \rightarrow COSMIC_EVENT_ASSESSMENT$
- $COSMIC_EVENT_ID \rightarrow PLANET_ID$
- $COSMIC_EVENT_ID \rightarrow PLANET_ID \rightarrow COSMIC_EVENT_ASSESSMENT$

1НФ: Атрибуты атомарны. Не содержат составных значений, есть РК. Находится в 1НФ

2НФ: РК простой. Неключевые атрибуты полностью зависят от него. . Находится в 2НФ.

3НФ: Есть транзитивная зависимость.

НФБК: не в 3НФ соответственно не НФБК

ROOM

1. $ROOM_ID \rightarrow ROOM_NAME$

1НФ: Атрибуты атомарны. Не содержат составных значений, есть РК. Находится в 1НФ

2НФ: РК простой. Неключевые атрибуты полностью зависят от него. . Находится в 2НФ.

3НФ: Отсутствуют транзитивные зависимости.

НФБК: Соблюдается тк. нет зависимостей где ID не ключ.

ROOM_STATUS

1. $ROOM_STATUS \rightarrow ROOM_ID$

1НФ: Атрибуты атомарны. Не содержат составных значений, есть РК. Находится в 1НФ
2НФ: РК простой. Неключевые атрибуты полностью зависят от него. . Находится в 2НФ.
3НФ: Отсутствуют транзитивные зависимости.
НФБК: Соблюдается тк. нет зависимостей где ID не ключ.

PLANET

- $PLANET_ID \rightarrow PLANET_NAME$
- $PLANET_ID \rightarrow COLOUR$
- $PLANET_ID \rightarrow PLANET_SIZES$

1НФ: Все атрибуты являются атомарными. Они содержат одиночные, неделимые значения. enum также является атомарным
2НФ: Нет составного РК, поэтому неключевые атрибуты находятся в полной зависимости от РК.
3НФ: Отсутствуют транзитивные зависимости.
НФБК: Соблюдается тк. нет зависимостей где ID не ключ.

Device

- $DEVICE_ID \rightarrow DEVICE_NAME$
- $DEVICE_ID \rightarrow ROOM_ID$
- $DEVICE_ID \rightarrow ROOM_ID \rightarrow DEVICE_NAME$

1НФ: Все атрибуты являются атомарными. Они содержат одиночные, неделимые значения. enum также является атомарным
2НФ: Нет составного РК, поэтому неключевые атрибуты находятся в полной зависимости от РК.
3НФ: Есть транзитивная зависимость.
НФБК: не в 3НФ соответственно не НФБК

Device_Status

- $DEVICE_STATUS \rightarrow DEVICE_ID$

1НФ: Все атрибуты являются атомарными. Они содержат одиночные, неделимые значения. enum также является атомарным
2НФ: Нет составного РК, поэтому неключевые атрибуты находятся в полной зависимости от РК.
3НФ: Отсутствуют транзитивные зависимости.
НФБК: Соблюдается тк. нет зависимостей где ID не ключ.

Денормализация

В целом мы можем ускорить запросы к таблице DEVICE
Путем включения DEVICE_NAME в DEVICE
Не нужен будет join

Триггер

Сложный триггер созданный по мотивам задания из ЕГЭ по информатике про игру в камушки: Триггер проводит учет победителя

```
CREATE TABLE GameSessions (
```

```
    game_session_id SERIAL PRIMARY KEY,
```

```
    game_winning INTEGER,
```

```
    current_game_status VARCHAR(32),
```

```
    number_of_stone INTEGER
```

```
);
```

```
CREATE TABLE Moves (
```

```
    move_id SERIAL PRIMARY KEY,
```

```
    game_session_id INTEGER REFERENCES GameSessions(game_session_id),
```

```
    player_name VARCHAR(32),
```

```
    add_of_stones INTEGER,
```

```
    stones_after_move INTEGER
```

```
);
```

```
CREATE OR REPLACE FUNCTION check_game_winner()
```

```
RETURNS TRIGGER AS $$
```

```
DECLARE
```

```
    game_winning_threshold INTEGER;
```

```
    current_status VARCHAR(32);
```

```
    curr_number_of_stone INTEGER;
```

```
    total_stones INTEGER;
```

```
BEGIN
```

```
    SELECT game_winning, current_game_status, number_of_stone
```

```
    INTO game_winning_threshold, current_status, curr_number_of_stone
```

FROM GameSessions

WHERE game_session_id = NEW.game_session_id;

SELECT COALESCE(SUM(add_of_stones), 0)

INTO total_stones

FROM Moves

WHERE game_session_id = NEW.game_session_id;

total_stones := total_stones + NEW.add_of_stones;

NEW.stones_after_move := total_stones;

IF current_status = 'We play' AND total_stones >= game_winning_threshold THEN

UPDATE GameSessions

SET current_game_status =

CASE NEW.player_name

WHEN 'Petya' THEN 'Petya_win'

WHEN 'Vanya' THEN 'Vanya_win'

ELSE 'Error'

END

WHERE game_session_id = NEW.game_session_id;

END IF;

RETURN NEW;

END;

\$\$ LANGUAGE plpgsql;

CREATE TRIGGER trg_check_game_winner

BEFORE INSERT ON Moves

FOR EACH ROW

EXECUTE FUNCTION check_game_winner();

Вывод

Я изучил нормализацию бд, проанализировал отношения моей модели, определил их НФ.
Написал триггер.