УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Дисциплина «Информационный системы и базы данных»

**Лабораторная работа №3**

Студент

*Самойлова А.А.*

*P3130*

Преподаватель Шешуков Дмитрий Михайлович

Санкт-Петербург, 2024 г.

Текст задания

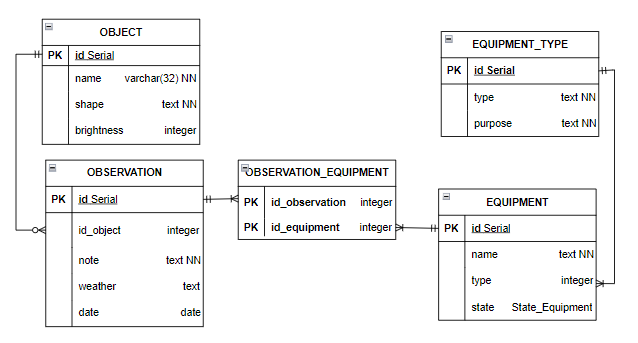
Текст задания

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

Исходная модель



Функциональные зависимости

|  |  |
| --- | --- |
| Object | id 🡪 (name, shape, brightness) |
| Observation | id 🡪 (id\_object, note, weather, date) |
| Equipment | id 🡪 (name, type, state) |
| Observation\_Equipment | (id\_observation, id\_equipment) 🡪 () |
| Equipmen\_Type | id 🡪 (type, purpose) |

Нормализация

**1NF**

1. Нет дублирующихся строк
2. Все атрибуты атомарны
3. Нет повторяющихся атрибутов с одинаковым смыслом

Модель удовлетворяет первой нормальной форме.

**2NF**

1. Отношение в 1NF
2. Есть первичный ключ

2) Атрибуты, не входящие в первичный ключ, в полной

функциональной зависимости от первичного ключа отношения.

Модель удовлетворяет второй нормальной форме.

**3NF**

1. Находится в 2NF

Схема уже находится в 3NF, поскольку каждый атрибут в таблице функционально зависит только от ключа этой таблицы.

**BCNF**

Отношения находится в 3NF каждый ключ является суперключом и нет транзитивных зависимостей. Поэтому она уже находится в BCNF.

Денормализация

1. Добавление представления с именем FULL\_OBSERVATION, которое объединяет таблицы OBSERVATION и OBJECT. Создание представления напрямую не повышает производительность, но может улучшить удобство работы с данными.

CREATE or REPLACE VIEW FULL\_OBSERVATION AS

  SELECT OBSERVATION.id, OBJECT.name AS name\_object, note, weather, date

  FROM OBSERVATION

  INNER JOIN OBJECT

  ON OBSERVATION.id\_object = OBJECT.id;

1. Добавление дополнительного столбца в таблицу OBJECT, который будет содержать информацию о количестве наблюдений, проведенных для каждого объекта.

Добавим столбец observation\_count в таблицу OBJECT, который будет хранить количество наблюдений, проведенных для каждого астрономического объекта. Это позволит быстро получать информацию о том, насколько активно наблюдается каждый объект, без необходимости выполнять дополнительные запросы к таблице OBSERVATION.

ALTER TABLE OBJECT

ADD COLUMN observation\_count INTEGER DEFAULT 0;

Триггер и связанная с ним функция, для автоматического увеличения счетчика наблюдений для соответствующего объекта в таблице OBJECT.

CREATE OR REPLACE FUNCTION update\_observation\_count()

RETURNS TRIGGER AS $$

BEGIN

  IF(TG\_OP = "INSERT") THEN

    UPDATE OBJECT

    SET observation\_count = observation\_count + 1

    WHERE id = NEW.id\_object;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER observation\_insert\_trigger

AFTER INSERT ON OBSERVATION

FOR EACH ROW

EXECUTE PROCEDURE update\_observation\_count();

Такая денормализация поможет ускорить запросы, которые требуют информации о частоте наблюдений для каждого объекта.