Федеральное государственное автономное образовательное высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники
Направление подготовки: 09.03.01 – Информатика и вычислительная техника

Отчет
по лабораторной работе №3
по дисциплине «Базы данных»

Вариант 313122

Выполнил:

Колмаков Дмитрий Владимирович, P3131

Преподаватель:

Наумова Надежда Александровна

г. Санкт-Петербург, 2023 г.

Оглавление

Задание

Текст задания

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

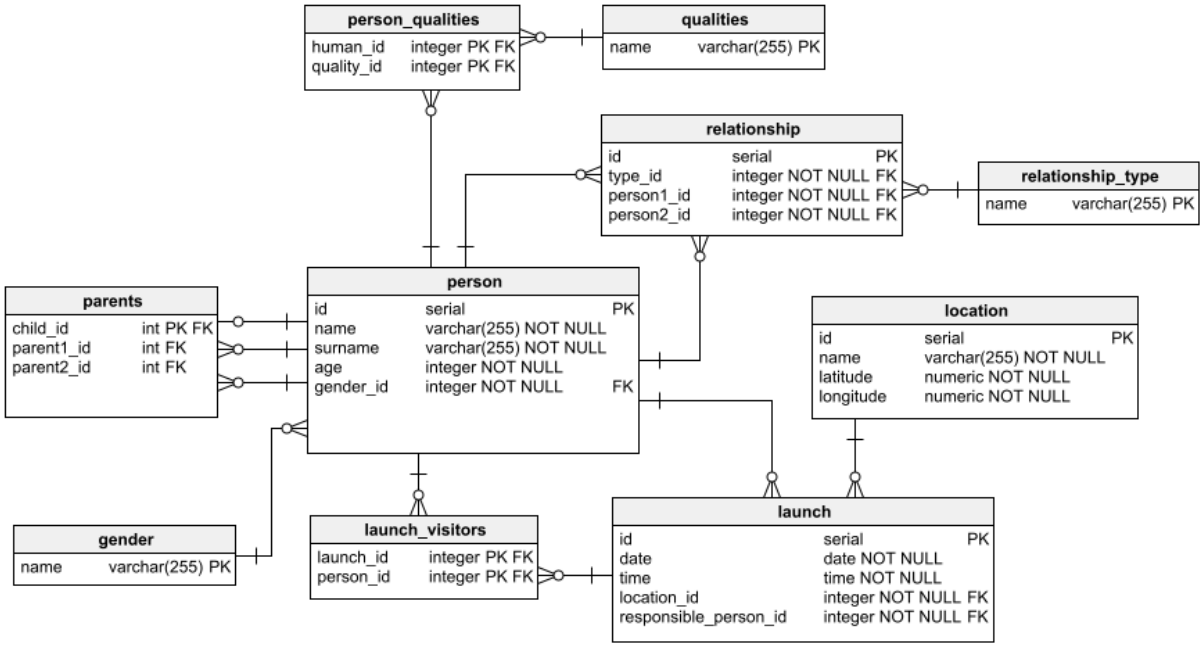
- опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
- приведите отношения в 3NF (как минимум). Постройте схему на основеNF (как минимум);
- опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему 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.

Ход работы



Функциональные зависимости

person: id → (name, surname, age, gender_id)

gender: name → ()

parents: child_id → (parent1_id, parent2_id)

qualities: name → ()

person_qualities: (human_id, quality_id) → ()

relationship_type: name → ()

relationship: id → (type_id, person1_id, person2_id)

location: id → (name, latitude, longitude)

launch: id → (date, time, location_id, responsible_person_id)

launch_visitors: (launch_id, person_id) → ()

Нормальные формы

1NF: Отношение находится в 1NF, если все атрибуты атомарны и нет повторяющихся атрибутов с одинаковым смыслом. Модель находится в 1NF, так как все атрибуты различны и хранят единственное значение.

2NF: Отношение находится в 2NF, если оно находится в 1NF и все его неключевые атрибуты полностью функционально зависят от первичного ключа. Модель находится в 2NF, так как во всех отношениях есть первичный ключ и все неключевые атрибуты полностью функционально зависят от первичных ключей, а не от его частей.

3NF: Отношение находится в 3NF, если оно находится в 2NF и не содержит транзитивных зависимостей. Модель находится в 3NF, так как все неключевые атрибуты зависят напрямую от первичных ключей, и не содержат транзитивных зависимостей.

BCNF: Отношение находится в BCNF, если оно находится в 3NF и ключевые атрибуты составного ключа не зависят от неключевых атрибутов. Модель находится в BCNF, так как оба условия выполнены для всех отношений.

Денормализация

Объединение связанных таблиц

В некоторых случаях, объединение таблиц может уменьшить количество операций JOIN и ускорить обработку запросов. Например, можно рассмотреть объединение таблиц person и parents, если часто запрашиваются данные о человеке и его родителях одновременно.

Добавление избыточных атрибутов

В некоторых случаях добавление избыточных атрибутов может улучшить производительность запросов. Например, если часто запрашивается количество посетителей запуска, можно добавить атрибут visitors_count в таблицу launch. Это позволит избежать операций подсчета при каждом запросе, однако необходимо будет обновлять этот атрибут при добавлении или удалении посетителей.

Триггеры

```
-- Триггер, который будет выводить ФИ родителей человека,
-- информация о котором в таблице person была обновлена

-- Функция, выполняющая поиск родителей по имени и фамилии человека
CREATE OR REPLACE FUNCTION find_parents (name_param VARCHAR(255),
surname_param VARCHAR(255))
    RETURNS TABLE (parent_name VARCHAR(255), parent_surname VARCHAR(255)) AS
$$
BEGIN
    RETURN QUERY SELECT parent.name, parent.surname FROM person parent
    INNER JOIN parents ON (parent.id = parents.parent1_id OR parent.id =
parents.parent2_id)
    INNER JOIN person child ON child.id = parents.child_id
    WHERE child.name = name_param
    AND child.surname = surname_param;
END;
$$ LANGUAGE plpgsql;

-- Функция, выводящая ФИ родителей по имени и фамилии человека
CREATE OR REPLACE FUNCTION print_find_parents(name_param VARCHAR(255),
surname_param VARCHAR(255))
    RETURNS VOID AS $$
DECLARE
    result RECORD;
BEGIN
    RAISE NOTICE 'Имя, Фамилия родителей человека % %:', name_param,
surname_param;
    FOR result IN SELECT * FROM find_parents(name_param, surname_param) LOOP
        RAISE NOTICE '- % %', result.parent_name, result.parent_surname;
    END LOOP;
END;
$$ LANGUAGE plpgsql;

-- Функция, которая вызывается при обновлении person
CREATE OR REPLACE FUNCTION update_person_trigger_function()
    RETURNS TRIGGER AS $$
BEGIN
    PERFORM print_find_parents(NEW.name, NEW.surname);
    RETURN NEW;
END;
```

```
$$ LANGUAGE plpgsql;

-- Удаляем существующие триггеры, если они существуют
DROP TRIGGER IF EXISTS update_person_trigger ON person;

-- Создаем триггер, который вызывает функцию update_person_trigger_function
-- при вставке записи в таблицу person
CREATE TRIGGER update_person_trigger
    AFTER UPDATE ON person
    FOR EACH ROW
EXECUTE FUNCTION update_person_trigger_function();

-- Проверяем функциональность триггеров и функций
UPDATE person SET name = 'Дэвид' WHERE name = 'Дэвид';
```

Результат

```
studs.s368328> UPDATE person SET name = 'Дэвид' WHERE name = 'a'
Имя, Фамилия родителей человека Дэвид Смирнов:
Мама Смирнова
Папа Смирнов
```

Вывод

При выполнении лабораторной работы я познакомился с понятием нормализации и денормализации. Научился определять функциональные зависимости модели, а также анализировать последнюю на соответствие различным нормальным формам. Изучил эффективные способы денормализации схемы базы данных и ситуации, в которых возможно их применение.