



**«Московский государственный технический
университет имени Н.Э. Баумана»**

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ КАФЕДРА
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

Отчет

по лабораторной работе №2

Дисциплина: Базы данных

Название лабораторной работы: Основы языка SQL. SELECT запросы.

Вариант 5

Студент

ИУ6-34Б

(Группа)

А.И.Гарифуллин

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

М.А. Скворцова

(Подпись, дата)

(И.О. Фамилия)

Постановка целей и задач

Цели:

Данная лабораторная работа призвана сформировать у студента понимание особенностей хранения данных приложения в РСУБД, а также настройка и поддержка хранения данных.

Задачи:

- Получить теоретические знания по концептуальным картам.
- Ознакомиться с нормализацией в БД.
- Изучение типов связей.
- Ознакомиться с DDL операторами.
- Изучение типов данных.
- Научится добавлять записи в таблицы.
- Научиться удалять и изменять записи в таблице.
- Научиться контролю целостности данных.
- Ознакомиться с механизмами контроля, транзакциями и триггерами.

Часть 1. Проектирование схемы базы данных

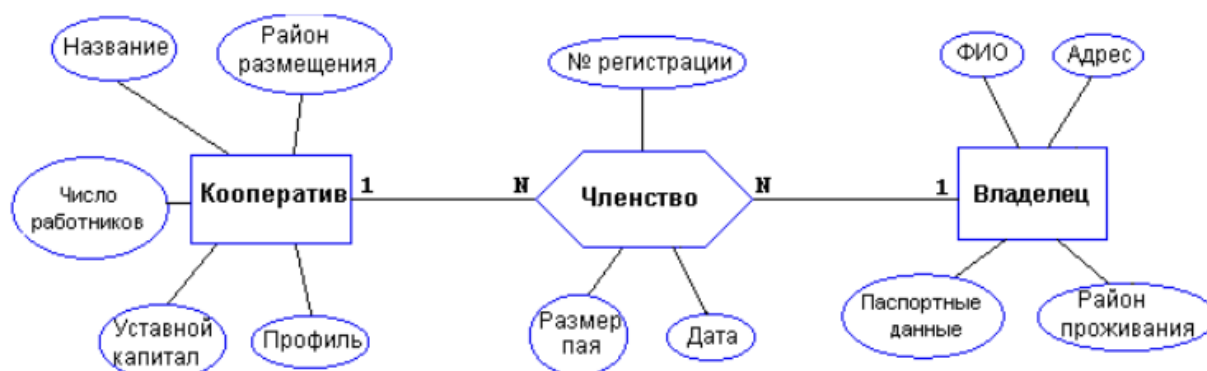
Первое практическое задание связано с проектированием схемы базы данных для работы приложения (WEB/Mobile/Desktop). Каждый индивидуальный вариант содержит предметную область, из которой должна быть проектируемая база данных. К данной предметной области необходимо добавить не менее 2-х дополнительных таблиц (сущностей), необходимых для детального решения поставленной задачи. Задачей студента является решить, для чего будет использоваться создаваемая база данных, и, исходя из этого, построить её концептуальную схему. Результатом данной части лабораторной работы является схема базы данных (в виде ER-диаграммы, содержащей таблицы и связи между ними, с уточнением типов столбцов, с описанием внешних и первичных ключей). При сдаче задания студент должен обосновать соответствие созданной схемы поставленной задаче.

Исходная схема:

Вариант №5

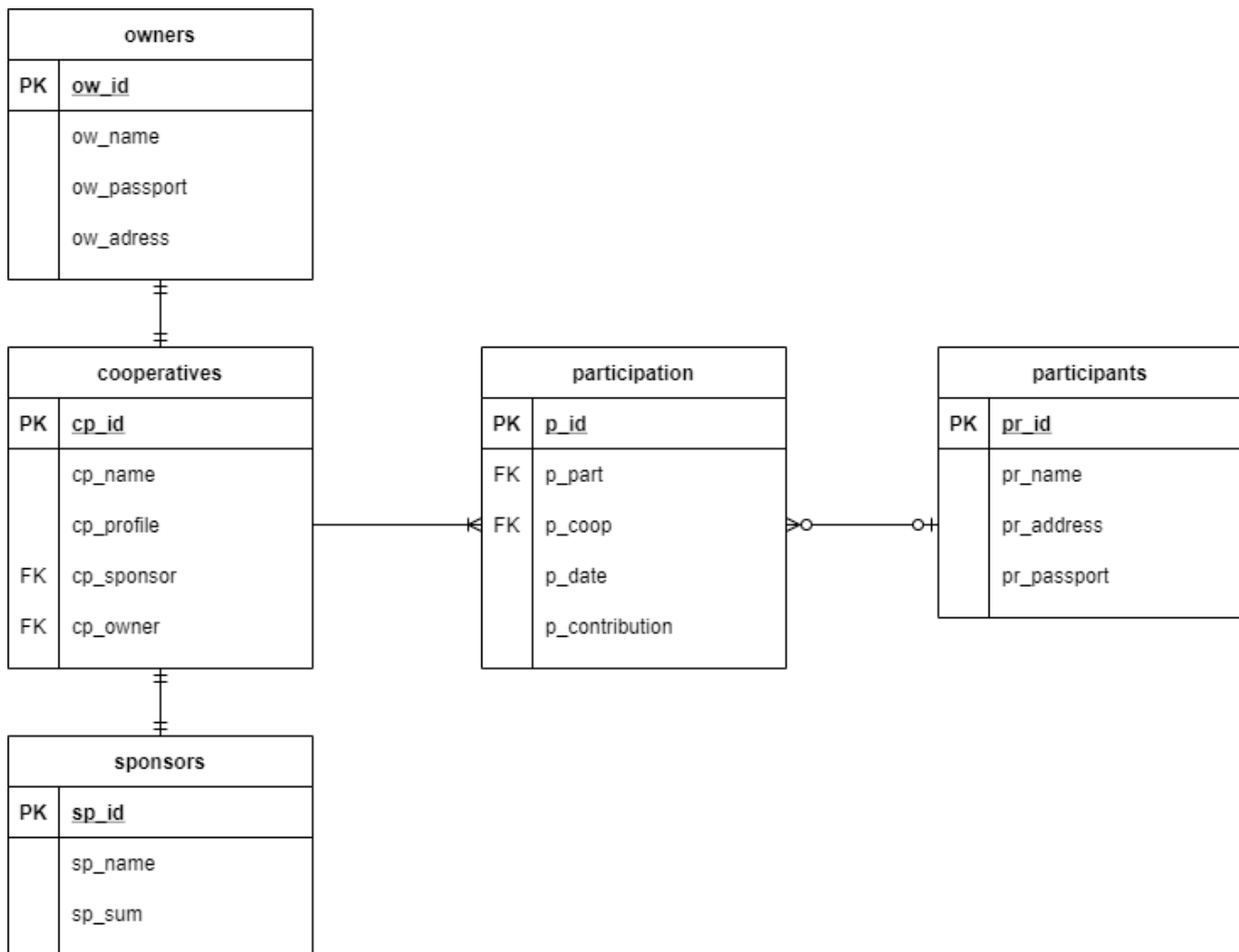
Предметная область для практических заданий №2.* №3.*: **Кооперативы**

Пример схемы для задания №2.1, от которого можно отталкиваться:



Примечание: профиль - продуктовый, галантерейный, канцелярский и т.п.

ER-диаграмма предметной области - Научные конференции:



Часть 2. Создание и заполнения таблиц.

Второе практическое задание заключается в подготовке SQL-скрипта для создания таблиц согласно схеме, полученной в предыдущем задании (с уточнением типов столбцов). Необходимо определить первичные и внешние ключи, а также декларативные ограничения целостности (возможность принимать неопределенное значение, уникальные ключи, проверочные ограничения и т. д.). Таблицы следует создавать в отдельной базе данных. Кроме того, нужно подготовить данные для заполнения созданных таблиц. Кроме того, нужно подготовить данные для заполнения созданных таблиц. Объем подготовленных данных должен составлять не менее 10 экземпляров для каждой из стержневых сущностей и 1000 экземпляров для целевой

сущности. На основе этих данных необходимо создать SQL-скрипт для вставки соответствующих строк в таблицы БД.

С помощью средств pgAdmin 4 создал данные таблицы и наложил ограничения на столбцы.

Также добавил связи, указанные на схеме, указав первичные и внешние ключи

Подготовил данные для таблиц и заполнил их с помощью SQL-скриптов.

```
INSERT INTO sponsors (sp_name, sp_sum) VALUES
('RSHB', 10000000),
('Scanner', 8000000),
('Moy Capital', 8300000),
('Aricapital', 5400000),
('Pervaya', 6500000),
('BSPB', 9000000),
('Raiffayzen', 11000000),
('Index', 7000000),
('Alpha', 7700000),
('Gazprom', 10500000);
```

```
INSERT INTO owners (ow_name, ow_passport, ow_adress) VALUES
('John Snow', 1224, 'Neque St.'),
('Emilie Spencer', 8431, 'Sun Av.'),
('Mona Ortega', 8351, 'Snow St.'),
('David Beckham', 5482, 'Gates Av.'),
('Peter Parker', 6394, 'Madison St.'),
('Justin Horfman', 9518, 'Atlantic Av.'),
('Jim Kerry', 1522, 'Hall St.'),
('Mark Star', 7047, 'Snow St.'),
('Alex Mask', 3183, 'Din St.'),
('Andrey Makarov', 7777, 'Lubyanskaya');
```

```
INSERT INTO cooperatives (cp_name, cp_profile, cp_sponsor, cp_owner) VALUES
('OneProduct', 'product', 3, 2),
('PenPen', 'clerical', 4, 1),
('Gallery', 'haberdashery', 8, 3),
('Eating God', 'product', 1, 7),
('TTM', 'building', 5, 4),
('PPP', 'product', 2, 6),
('Empire Build', 'building', 9, 5),
('All Clean', 'processing', 6, 10),
('Clerc', 'clerical', 10, 8),
('Only Comfort', 'housing', 7, 9);
```

Часть 3. Операторы манипулирования

В ходе выполнения задания необходимо:

- Подготовить 3-4 выборки, которые имеют осмысленное значение для предметной области, и также составить для них SQL-скрипты.

- Сформулировать 3-4 запроса на изменение и удаление из базы данных. Запросы должны быть сформулированы в терминах предметной области. Среди запросов обязательно должны быть такие, которые будут вызывать срабатывание ограничений целостности. Составить SQL-скрипты для выполнения этих запросов.

Выборки, значение и SQL-скрипты:

Вывести все кооперативы и их спонсоров с суммой инвестиций более 6.5 млн

Query		Query History	
1	SELECT	cp_name as name_of_cooperative, sp_name as name_of_sponsor, sp_sum as investments	
2	FROM	cooperatives cc, sponsors s	
3	WHERE	cc.cp_sponsor = s.sp_id AND s.sp_sum > 6500000	
4			

Data Output			Messages	Notifications
	name_of_cooperative text	name_of_sponsor text	investments integer	
1	OneProduct	Moy Capital	8300000	
2	Gallery	Index	7000000	
3	Eating God	RSHB	10000000	
4	PPP	Scanner	8000000	
5	Empire Build	Alpha	7700000	
6	All Clean	BSPB	9000000	
7	Clerc	Gazprom	10500000	
8	Only Comfort	Raiffayzen	11000000	

Вывести имена, названия кооперативов и даты вступления тех, кто вступил в интервале от 01.10.21 до 01.11.21

Query

Query History

1

```
SELECT pr_name as name_of_participant, cp_name as name_of_cooperative, p_date as date_of_entry
FROM participants pr, cooperatives cc, participation pp
WHERE pp.p_part = pr.pr_id AND pp.p_coop = cc.cp_id AND pp.p_date > '2021-10-01' AND pp.p_date < '2021-11-01'
```

4

Data Output

Messages

Notifications

	name_of_participant text	name_of_cooperative text	date_of_entry date
1	Xanthus Hicks	Gallery	2021-10-07
2	Dean Velasquez	Empire Build	2021-10-24
3	Gisela Frederick	PenPen	2021-10-11
4	Hedley Sampson	Only Comfort	2021-10-31
5	Merrill Watkins	All Clean	2021-10-25
6	Michael Gallagher	All Clean	2021-10-05
7	Deanna Hammond	PenPen	2021-10-08
8	Tanisha Morrow	PPP	2021-10-14
9	Cassandra Reyes	Only Comfort	2021-10-20
10	Cameron Camacho	PenPen	2021-10-29
11	Yael Cervantes	All Clean	2021-10-28
12	Tad Rojas	All Clean	2021-10-07
13	Blake Ramos	Gallery	2021-10-12
14	Barbara Chavez	OneProduct	2021-10-05
15	Demetria Haley	TTM	2021-10-07
16	Yoshio Mann	Empire Build	2021-10-19

Активация Window
Чтобы активировать Windows
"Параметры".

Вывести имена, адрес и кооперативы, в которые вступили люди, проживающие по адресу, содержащий в названии “Dictum”

Query

Query History

1

```
SELECT pr_name as name_of_participant, pr_address as address, cp_name as cooperative
FROM participants pr, participation pp, cooperatives cc
WHERE pr.pr_address LIKE '%Dictum%' AND pr.pr_id = pp.p_part AND pp.p_coop = cc.cp_id
```

4

Data Output

Messages

Notifications

	name_of_participant text	address text	cooperative text
1	Hollee Elliott	1976 Dictum Av.	Gallery
2	Mari Landry	P.O. Box 356, 7101 Dictum St.	PPP
3	Ezekiel Ware	P.O. Box 225, 3676 Dictum Street	Eating God

Изменим участника с номером 333

До обновления

```

1 SELECT *
2 from participants pp
3 where pp.pr_id = 333
4

```

Data Output Messages Notifications

	pr_id [PK] integer	pr_name text	pr_address text	pr_passport integer
1	333	Erasmus ...	6387 Sit St.	3875

UPDATE-запрос:

```

UPDATE participants pp set
pr_name = 'Amir Garifullin',
pr_address = '199454 Pr.Vernadskogo',
pr_passport = 9217
WHERE pr_id = 333;

```

Данные после обновления:

```

1 select *
2 from participants
3 where pr_id = 333

```

Data Output Messages Notifications

	pr_id [PK] integer	pr_name text	pr_address text	pr_passport integer
1	333	Amir Gari...	199454 Pr....	9217

Удаление данных:

Удалим все членства с 20.10.21 по 30.10.21:

Query Query History

```

1 DELETE FROM participation pp
2 WHERE pp.p_date >= '2021-10-20' AND pp.p_date <= '2021-10-30'

```

Data Output Messages Notifications

DELETE 7

Query returned successfully in 152 msec.

Удалим спонсоров, которые инвестировали меньше 8млн:

До удаления:

```
1 select *
2 from sponsors s
3 where s.sp_sum < 8000000
```

Data Output Messages Notifications

	sp_id [PK] integer	sp_name text	sp_sum integer
1	4	Aricapital	5400000
2	5	Pervaya	6500000
3	8	Index	7000000
4	9	Alpha	7700000

Удаление:

```
1 delete from sponsors s
2 where s.sp_sum < 8000000
```

Data Output Messages Notifications

ERROR: ОШИБКА: UPDATE или DELETE в таблице "sponsors" нарушает ограничение внешнего ключа "fk_sponsor" таблицы "cooperatives"
DETAIL: На ключ (sp_id)=(4) всё ещё есть ссылки в таблице "cooperatives".

SQL state: 23503

При удалении возникает ошибка нарушения ограничения внешнего ключа.

Часть 4. Контроль целостности данных

Четвертое практическое задание посвящено контролю целостности данных, который производится с помощью механизма транзакций и триггеров. Транзакции позволяют рассматривать группу операций как единое целое, либо отрабатывают все операции, либо ни одной. Это позволяет избегать несогласованности данных. Триггеры позволяют проверять целостность данных в момент выполнения транзакций, поддерживать целостность, внося изменения, и откатывать транзакции, приводящие к потере целостности.

Необходимо подготовить SQL-скрипты для проверки наличия аномалий (потерянных изменений, грязных чтений, неповторяющихся чтений, фантомов) при параллельном исполнении транзакций на различных уровнях изолированности SQL/92 (READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE). Подготовленные скрипты должны работать с одной из таблиц, созданных в практическом задании №2.1. Для проверки наличия аномалий потребуются два параллельных сеанса, операторы в которых выполняются пошагово:

- Установить в обоих сеансах уровень изоляции READ UNCOMMITTED. Выполнить сценарии проверки наличия аномалий потерянных изменений и грязных чтений.

- Установить в обоих сеансах уровень изоляции READ COMMITTED.

Выполнить сценарии проверки наличия аномалий грязных чтений и неповторяющихся чтений.

- Установить в обоих сеансах уровень изоляции REPEATABLE READ.

Выполнить сценарии проверки наличия аномалий неповторяющихся чтений и фантомов.

- Установить в обоих сеансах уровень изоляции SERIALIZABLE.

Выполнить сценарий проверки наличия фантомов.

Необходимо составить скрипт для создания триггера, а также подготовить несколько запросов для проверки и демонстрации его полезных свойств:

- Изменение данных для сохранения целостности.
- Проверка транзакций и их откат в случае нарушения целостности.

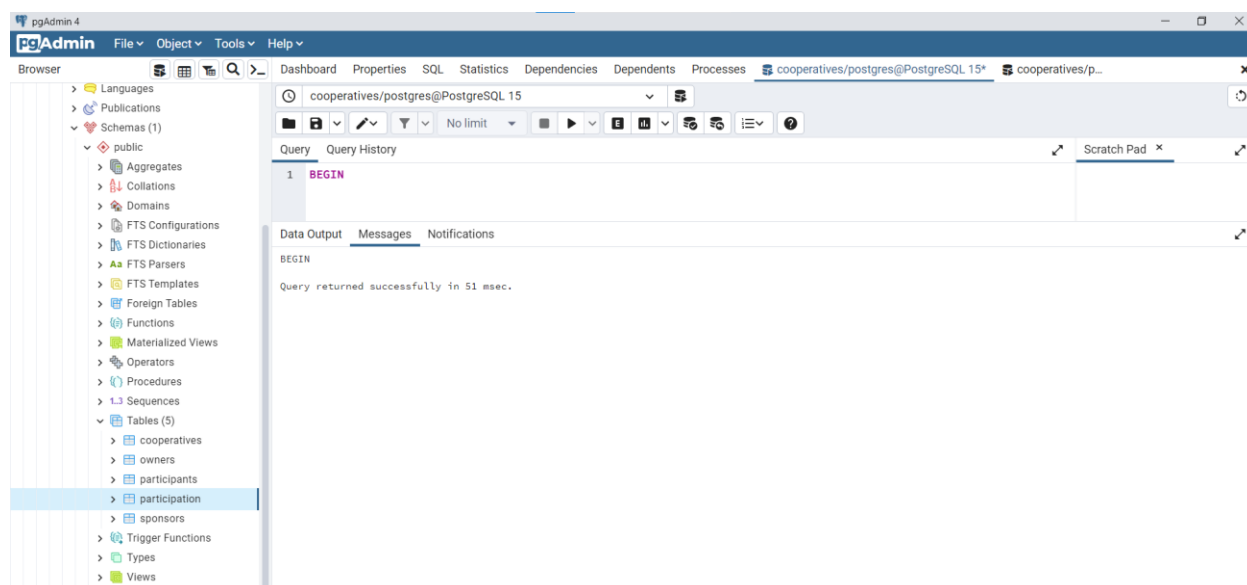
Read Uncommitted:

В PostgreSQL реализация уровня изоляции READ UNCOMMITTED более строгая, чем того требует стандарт языка SQL. Фактически этот уровень тождественен уровню изоляции READ COMMITTED. Поэтому будем считать эксперимент, проведенный для уровня изоляции READ COMMITTED, выполненным и для уровня READ UNCOMMITTED.

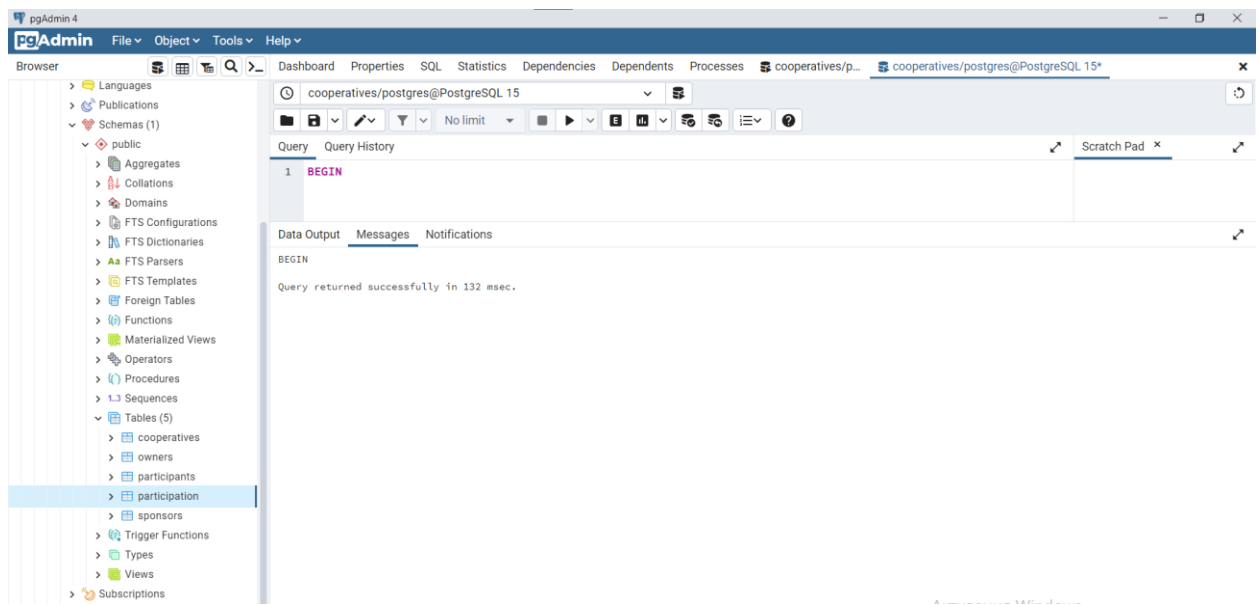
Read Committed:

Проверка на грязное чтение:

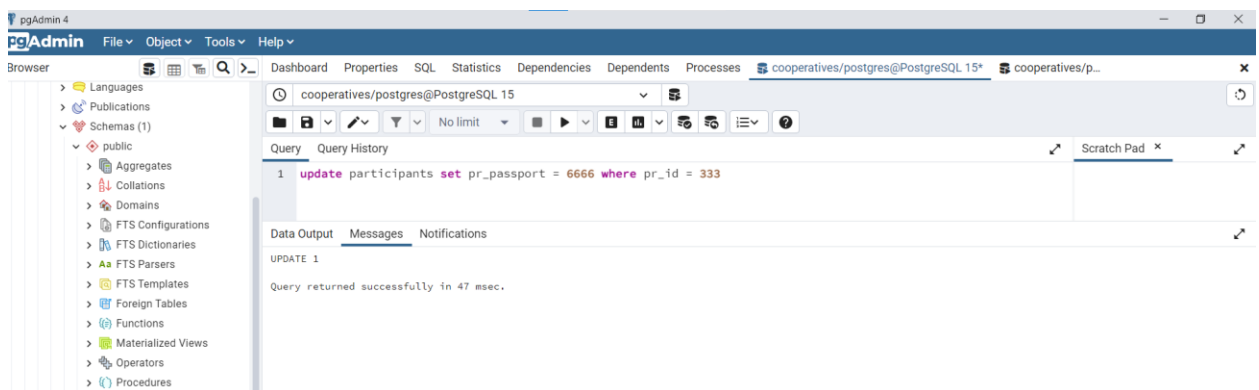
Активация 1-й транзакции:



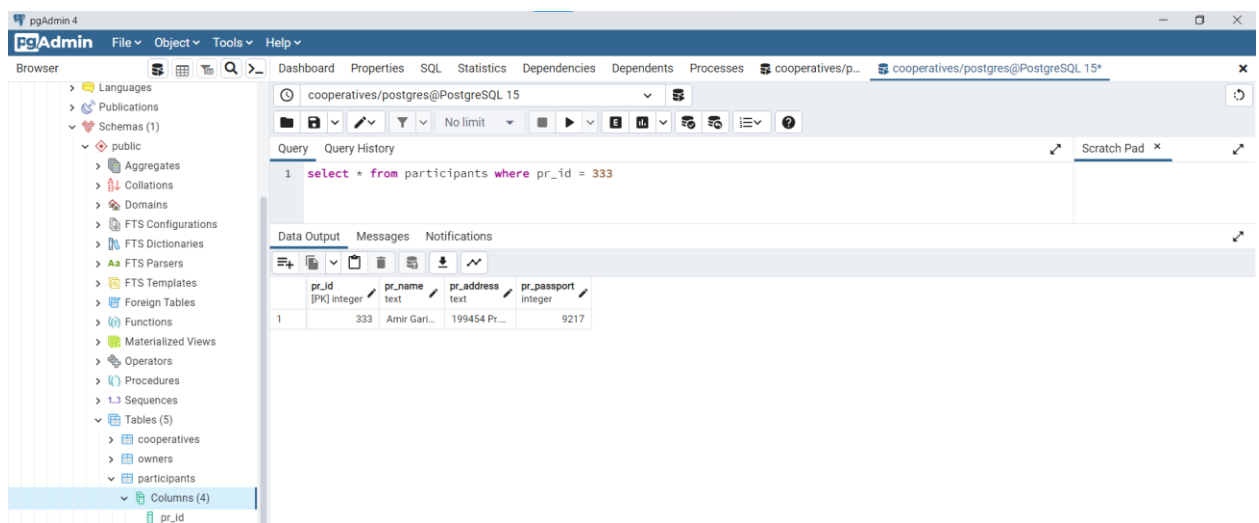
Активация 2-й транзакции:



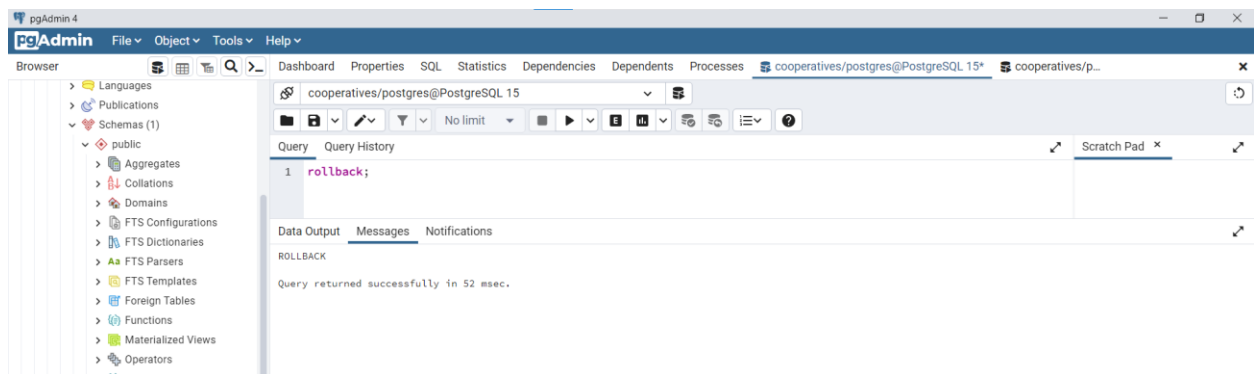
Попытка обновления записи таблицы в первой транзакции:



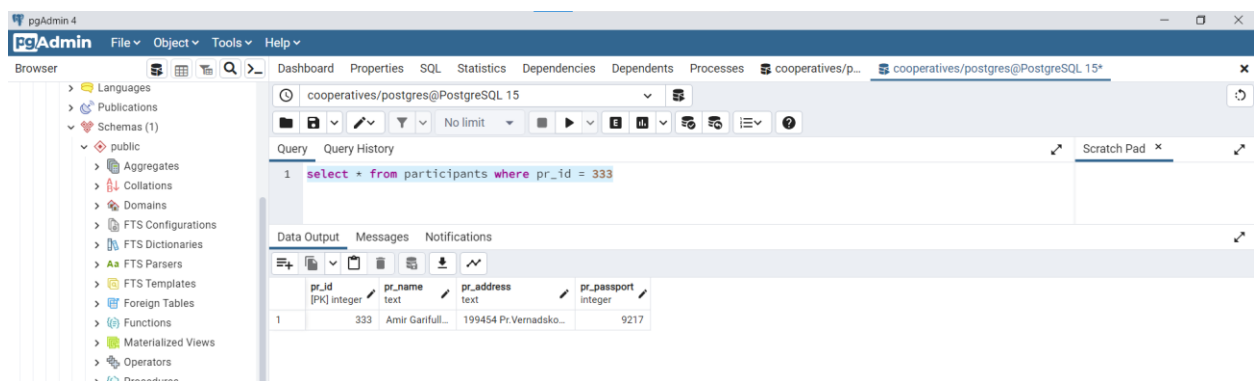
Проверка обновлений во второй транзакции:



Откат первой транзакции:



Повторный запрос на чтение во второй транзакции:

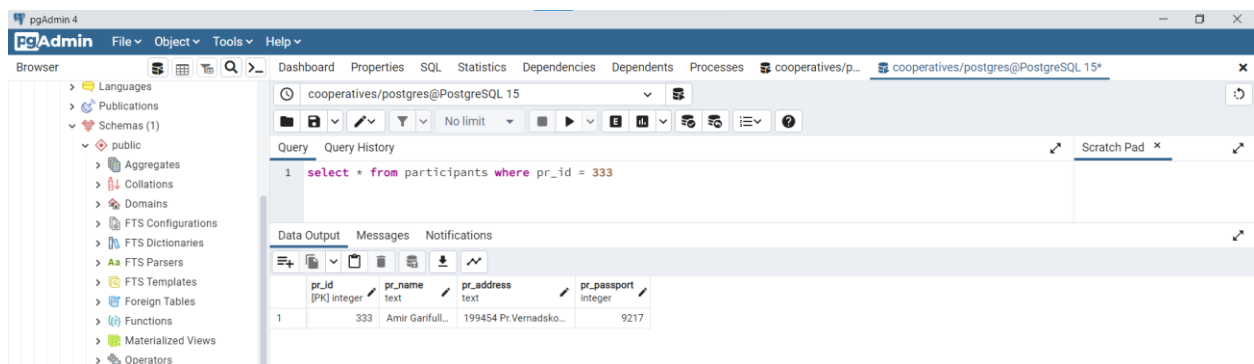


Видно, что грязное чтение не прошло.

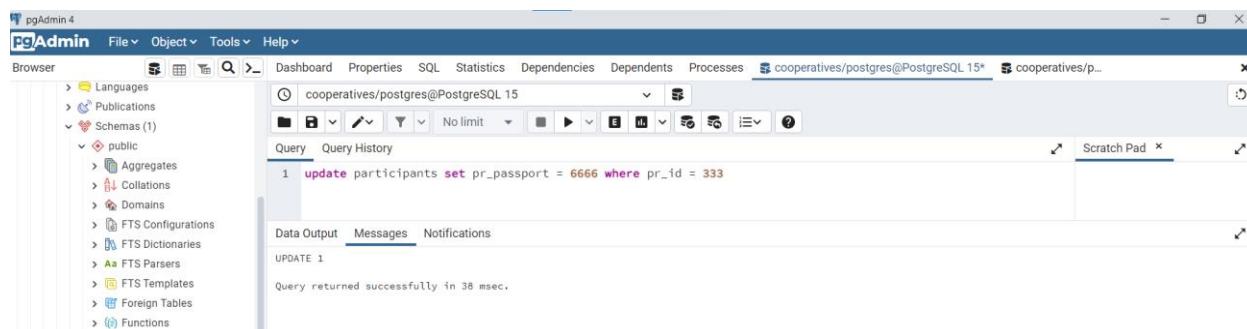
Проверка на неповторяющееся чтение:

Активация обеих транзакций (без скриншотов)

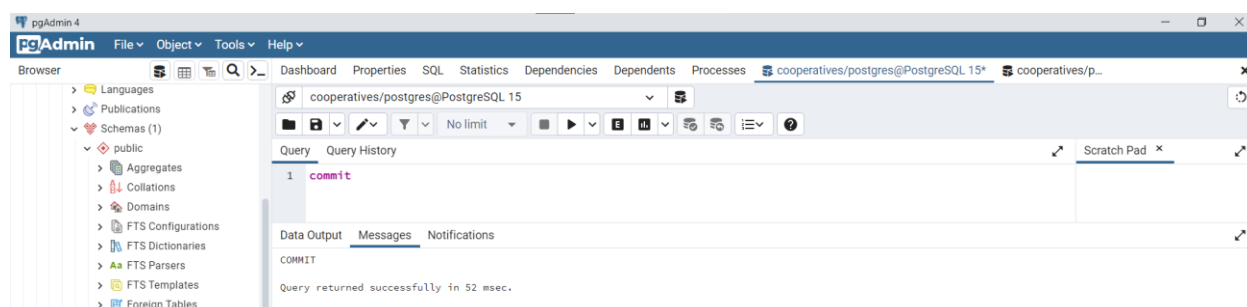
Чтение записи во 2-й транзакции:



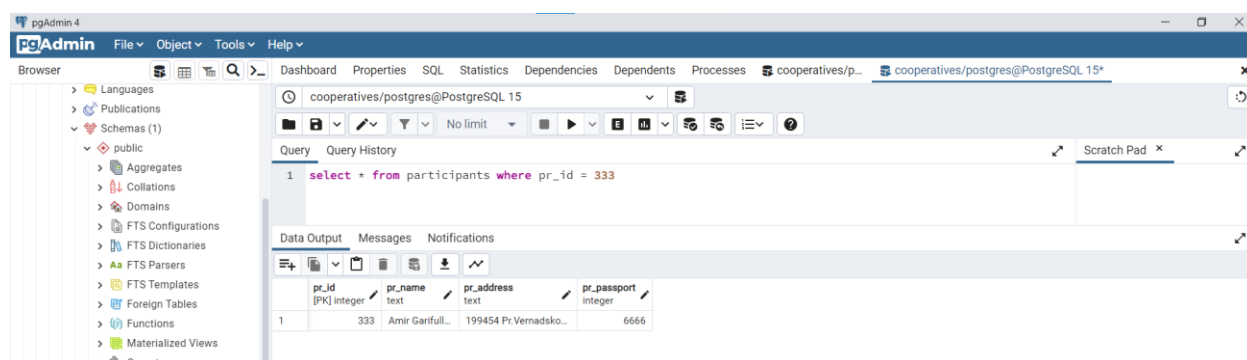
Обновление значения в 1-й транзакции:



Выход из 1-й транзакции:



Аналогичный запрос чтения записи во 2-й таблице:



Видно, что значение поменялось. Значит аномалия неповторяющегося чтения произошла.

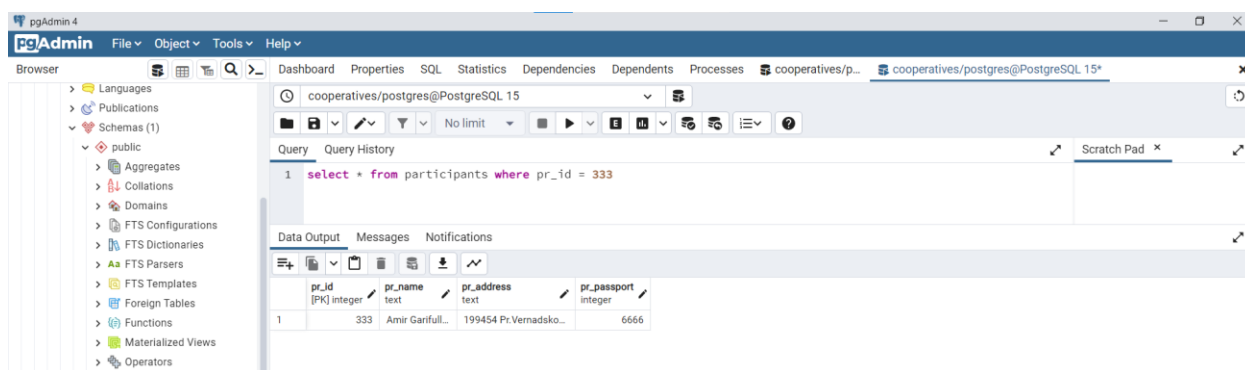
Таким образом при READ COMMITTED, не выполняется аномалия грязного чтения, а аномалия неповторяющихся значений выполняется.

REPEATABLE READ

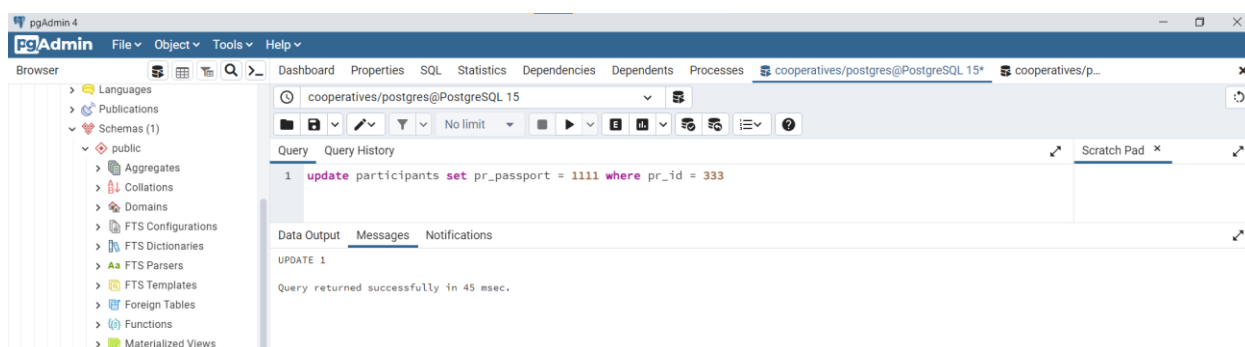
Проверка на неповторяющееся чтение:

Активация обеих транзакций командой BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ; (без скриншотов).

Запрос на чтение записи во второй транзакции:

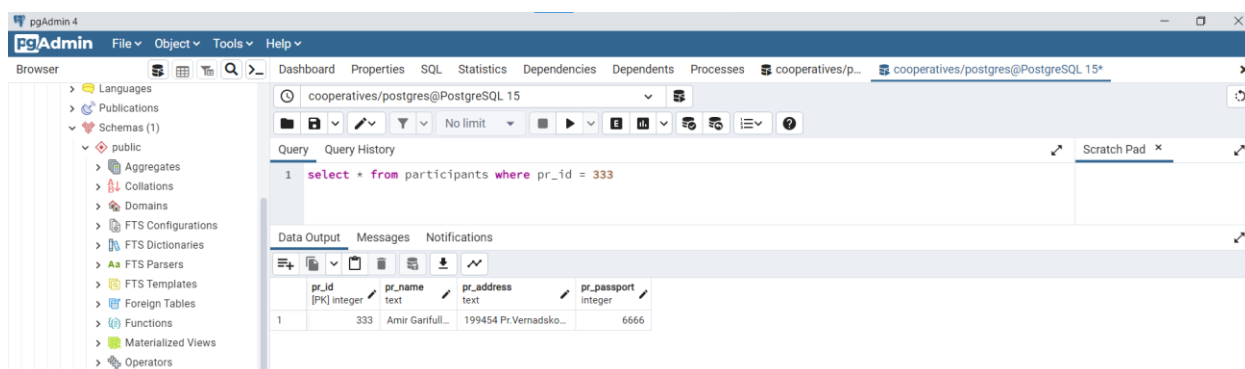


Обновление значения этой записи в первой транзакции:



Завершение 1-й транзакции (без скриншота).

Запрос на чтение во второй транзакции:

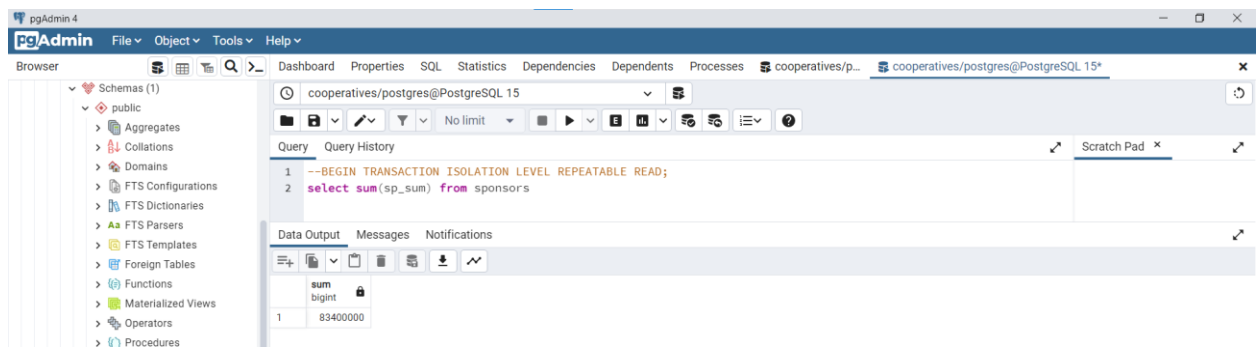


Видно, что запись не поменялась, а значит аномалия неповторяющегося чтения не прошла.

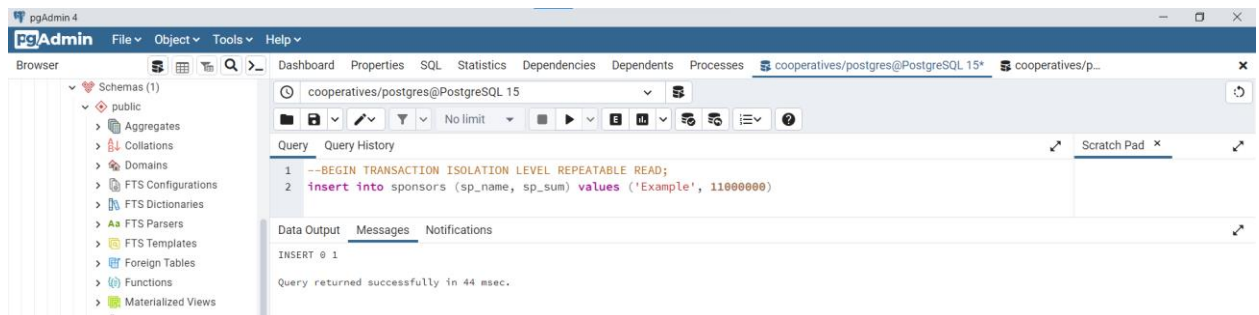
Проверка фантомов:

Активация 2-х параллельных транзакций (без скриншотов).

Нахождение суммарного количества инвестиций всех спонсоров:

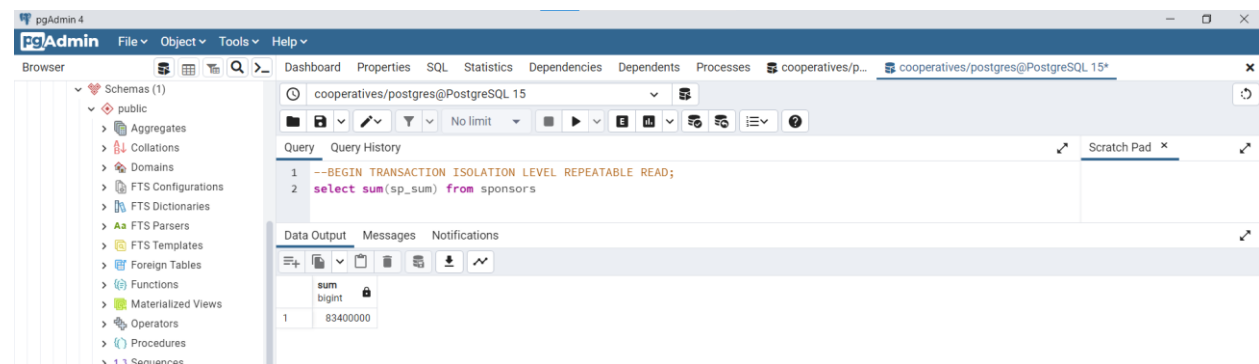


Добавление новых значений в 1-й транзакции:



Завершение 1-й транзакции (без скриншота).

Повторный запрос на чтение во 2-й транзакции:



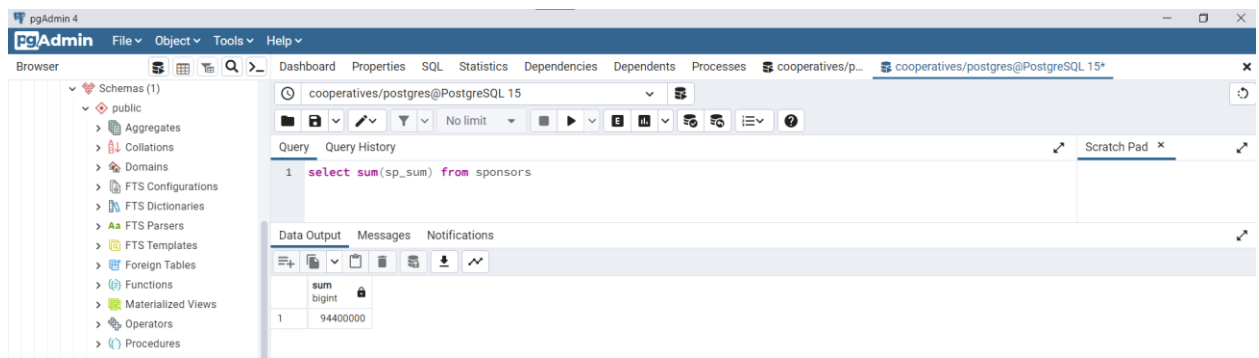
Значение не поменялось. Всё потому что PGAdmin блокирует фантомное чтение.

Аномалия фантомного чтения не прошла.

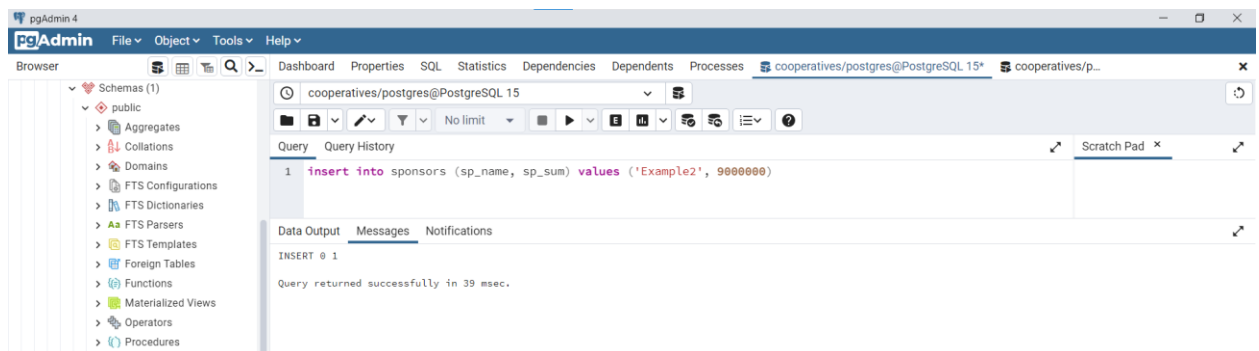
SERIALIZABLE:

Активация транзакций командой BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE (без скриншотов).

Запрос на чтение во 2-й транзакции:

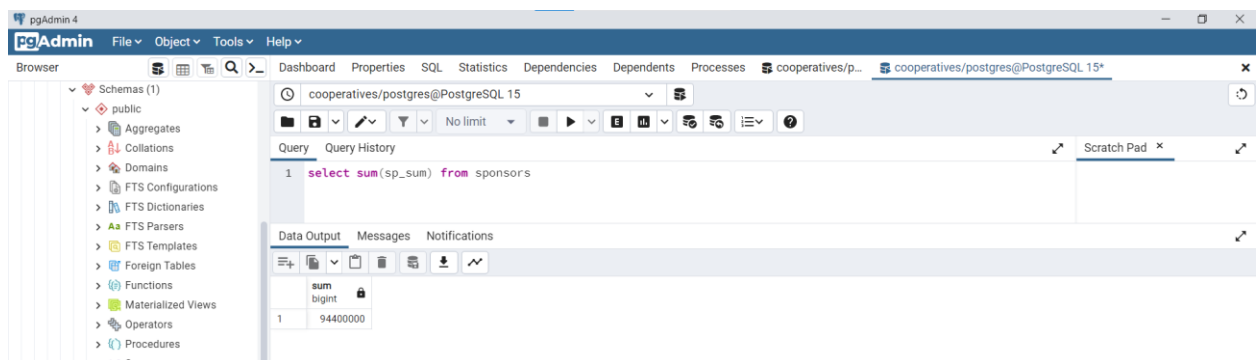


Добавление новой записи в 1-й транзакции:



Завершение 1-й транзакции (без скриншота).

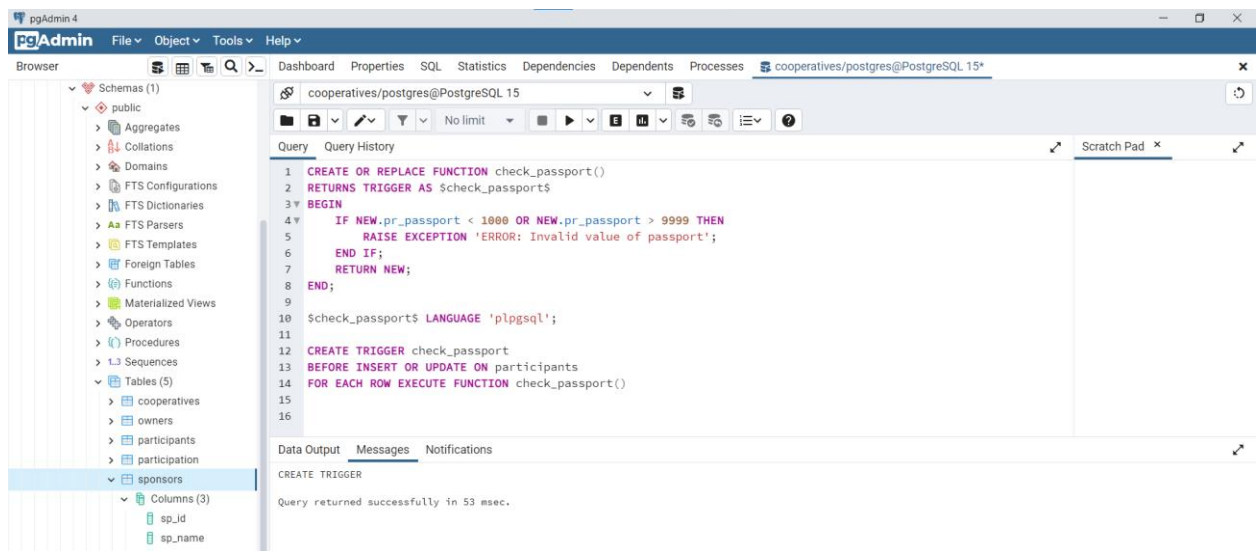
Повторный запрос на чтение во второй транзакции:



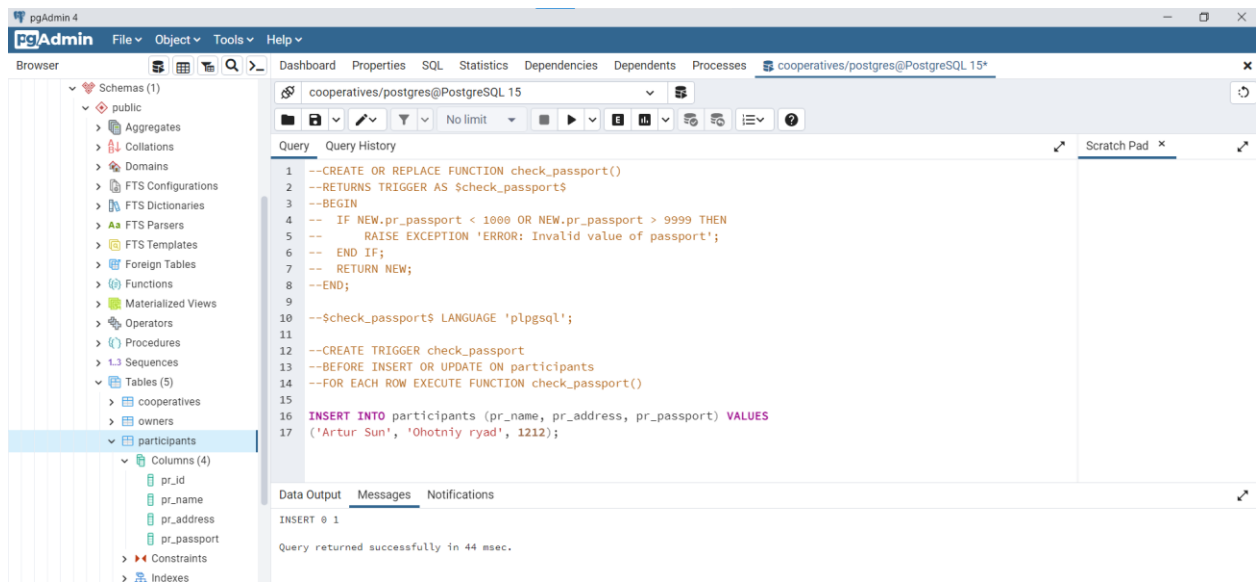
Как мы видим, значение не поменялось, а значит фантомного чтения не произошло.

Триггеры:

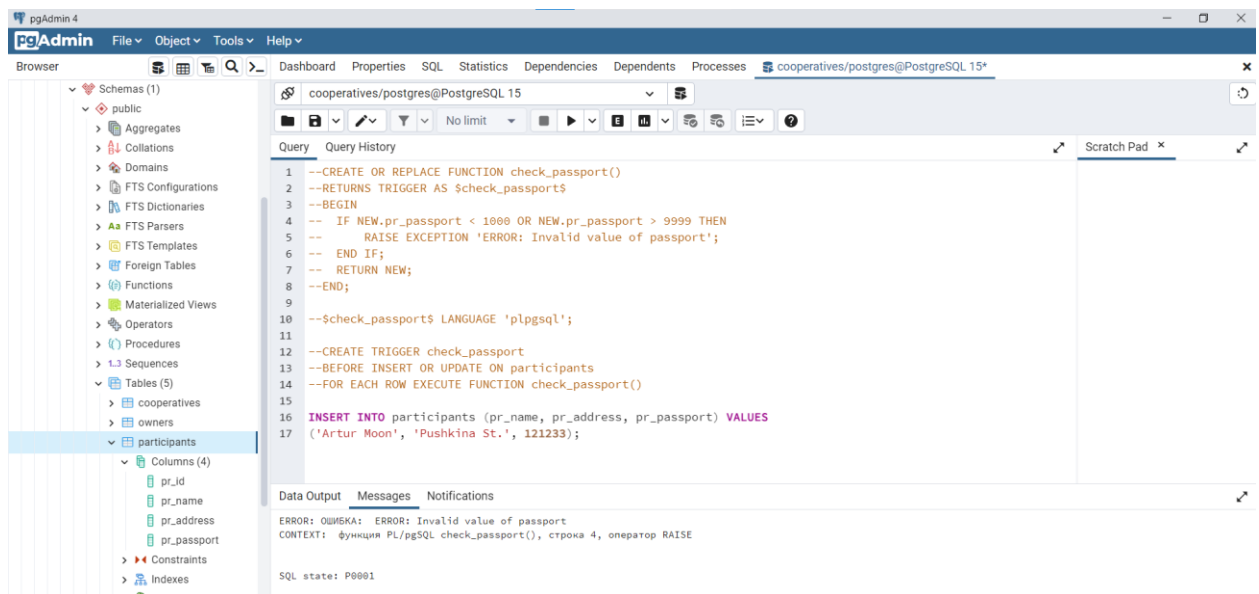
Создание триггера, проверяющего корректность добавляемых и обновляемых значений в таблице participants, а в частности значение столбца pr_passport должно соответствовать промежутку [1000; 9999]:



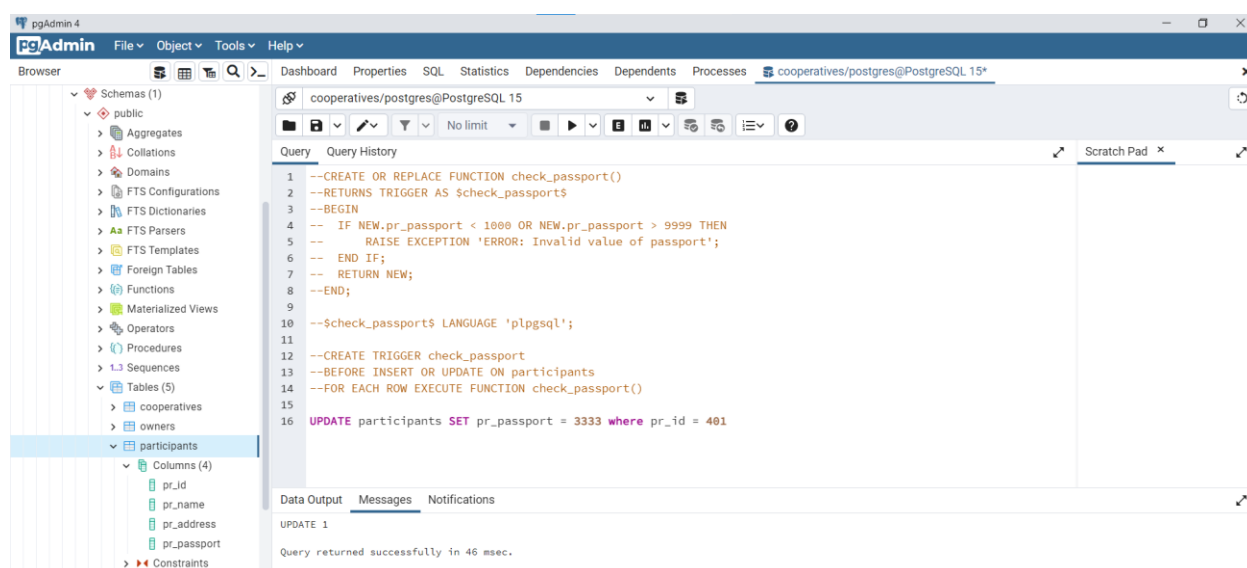
Попытка добавления с корректными данными:



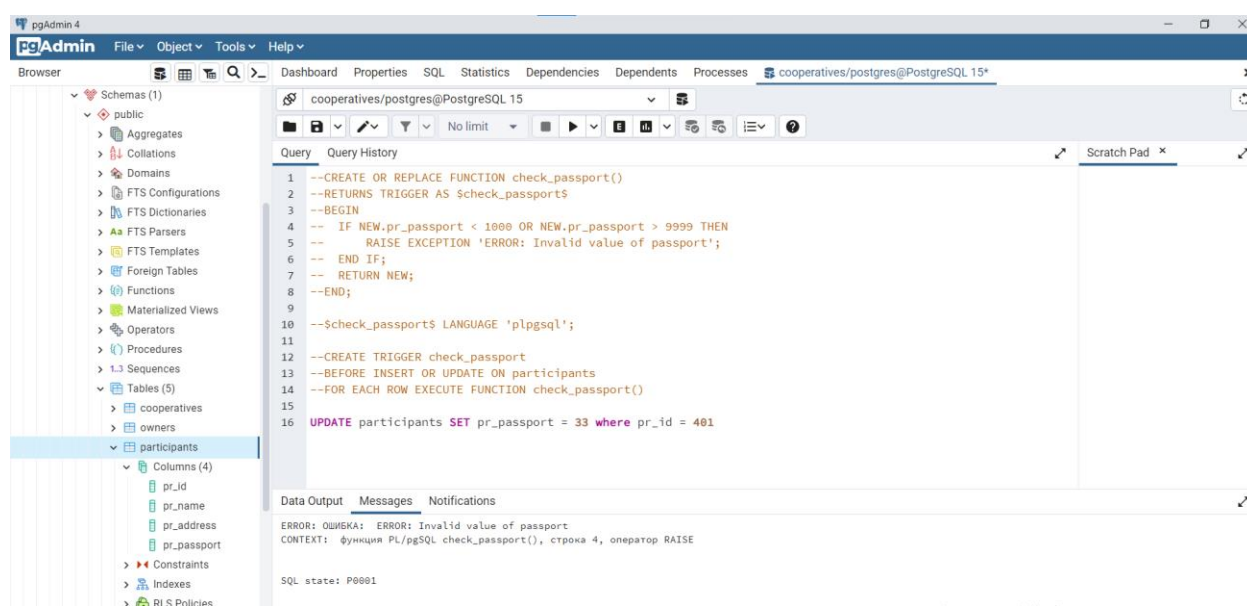
Попытка добавления с некорректными данными:



Попытка изменения существующей записи (корректные данные):



Попытка изменения существующей записи (некорректные данные):



Триггер успешно защищает от добавления некорректных данных, тем самым сохраняя целостность БД.

Вывод: в ходе проделанной лабораторной работы, были приобретены навыки проектирования базы данных, её создания на PostgreSQL, а также изучены связи между различными таблицами (внешний и первичный ключи). Были приобретены навыки оперирования между связанными таблицами. Изучены

различные уровни изолирования и способы предотвращения аномалий.
Получены и успешно использованы знания о триггерах.