



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

**Институт информационных технологий (ИИТ)
Кафедра цифровой трансформации (ЦТ)**

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ
по дисциплине «Проектирование баз данных»

Практическое занятие №1

Студенты группы *ИКБО-20-23 Кузнецов Лев Андреевич*

(подпись)

Ассистент *Брайловский А.В.*

(подпись)

Отчет представлен «__» _____ 2025 г.

Москва 2025 г.

СОДЕРЖАНИЕ

ПОСТАНОВКА ЗАДАЧИ	3
ХОД РАБОТЫ	4
ОТВЕТЫ НА ВОПРОСЫ.....	12
ВЫВОД.....	13

ПОСТАНОВКА ЗАДАЧИ

Цель: сформировать и закрепить навыки работы с реляционными базами данных на примере СУБД PostgreSQL.

Постановка задачи: на основе логической модели данных, созданной в предыдущем семестре, выполнить необходимые шаги.

ХОД РАБОТЫ

1. Анализ и описание ограничений целостности

Для данного пункта были составлены две таблицы на примере двух других таблиц: request и app_user, - из логической модели из предыдущего семестра (таблицы 1-2).

Таблица 1. Описание ограничений для таблицы request

Название столбца	Тип данных	Ограничение	Обоснование (Бизнес-правило)
id_request	SERIAL	PRIMARY KEY	Уникальный идентификатор заявки, генерируется автоматически.
id_app_user	INT	FOREIGN KEY (app_user)	Ссылка на пользователя. Заявка не может существовать без пользователя.
id_transaction	INT	FOREIGN KEY (transaction)	Ссылка на транзакцию. Заявка не может существовать без транзакции
description	TEXT	NOT NULL	Описание заявки, включающее в себя всю необходимую информацию о приобретаемом курсе. В каждой заявке обязательное присутствует непустое поле описания.

Таблица 2. Описание ограничений для таблицы app_user

Название столбца	Тип данных	Ограничение	Обоснование (Бизнес-правило)
id_app_user	SERIAL	PRIMARY KEY	Уникальный идентификатор заявки, генерируется автоматически.
personal_data	TEXT	NOT NULL	Личная информация пользователя. Для использования приложения пользователь обязан ввести

			требуемые личные данные, по этой причине данный столбец не может не иметь значения.
--	--	--	---

2. Создание структуры данных

The screenshot shows the dbstud application interface. The main window displays SQL code for creating a database schema. The code includes dropping existing tables, creating a parent table 'app_user', creating a child table 'transaction', adding foreign key constraints, creating another child table 'request', and adding more foreign key constraints. Below the code editor, a 'Статистика 1' (Statistics 1) window is open, showing the execution results of the SQL queries.

```

DROP TABLE IF EXISTS app_user cascade;
DROP TABLE IF EXISTS request cascade;
DROP TABLE IF EXISTS transaction cascade;

-- Сначала создаём родительскую таблицу с различными ограничениями
create table app_user (
  id_app_user serial NOT NULL PRIMARY KEY,
  personal_data text NOT null
);

-- Создание дочерней таблицы
create table transaction (
  id_transaction serial NOT NULL PRIMARY KEY,
  id_app_user integer NOT NULL,
  sum smallint NOT NULL,
  status smallint NOT null
);

-- Внесение ограничений в дочернюю таблицу
alter table transaction
add constraint transaction_id_user_fk
foreign key (id_app_user) references app_user (id_app_user);

-- Создание последней дочерней таблицы с ограничениями
create table request (
  id_request serial NOT NULL PRIMARY KEY,
  id_app_user integer NOT NULL,
  id_transaction integer NOT NULL,
  description text NOT null
);

-- Обновление данных в дочерней таблице
alter table request
add constraint request_id_user_fk
foreign key (id_app_user) references app_user (id_app_user);

alter table request
add constraint request_id_transaction_fk foreign key (id_transaction) references transaction (id_transaction);

```

Name	Value
Queries	9
Updated Rows	0
Execute time	0.075s
Fetch time	0.000s
Total time	0.075s
Start time	2025-09-01 22:53:30.826
Finish time	2025-09-01 22:53:30.905

Рисунок 1 – Кодовый вид запроса и итоговый вывод

3. Заполнение таблиц данными (DML – Data Manipulation Language)

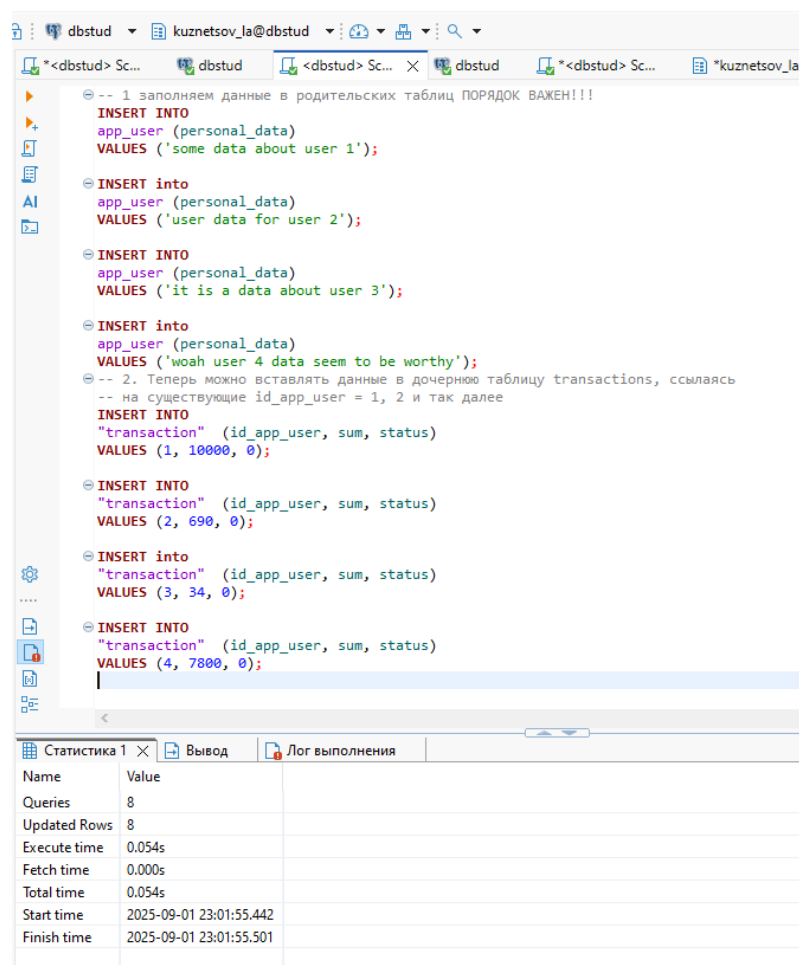


Рисунок 2 – Заполнение данных в итоговых таблицах

4. Составление запросов на выборку

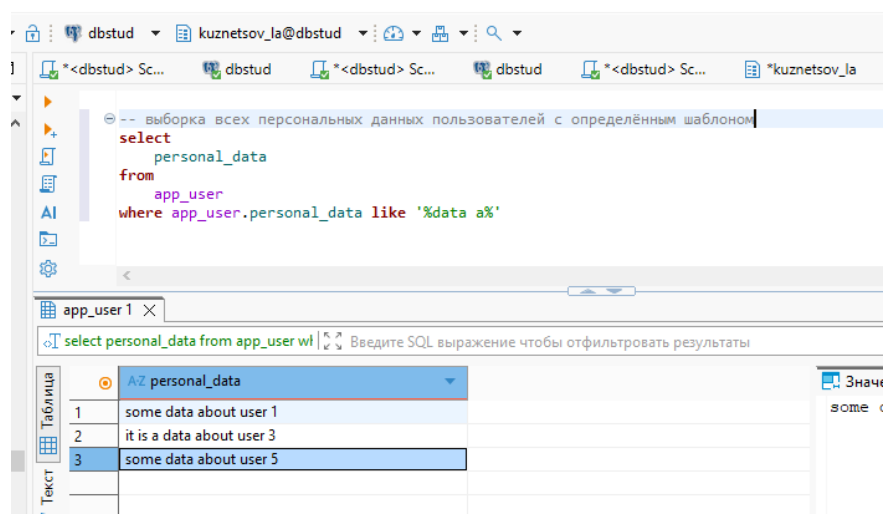


Рисунок 3 – Выборка при помощи %

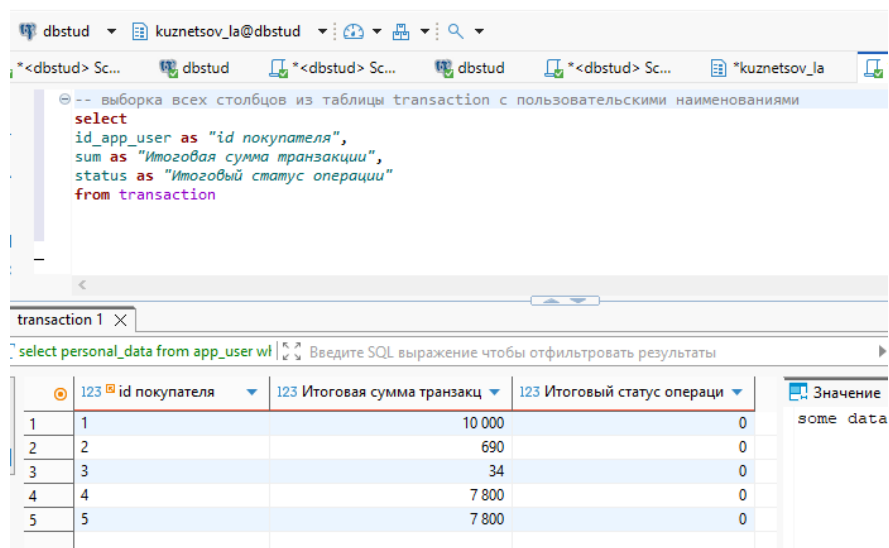


Рисунок 4 – Пользовательское именование столбцов

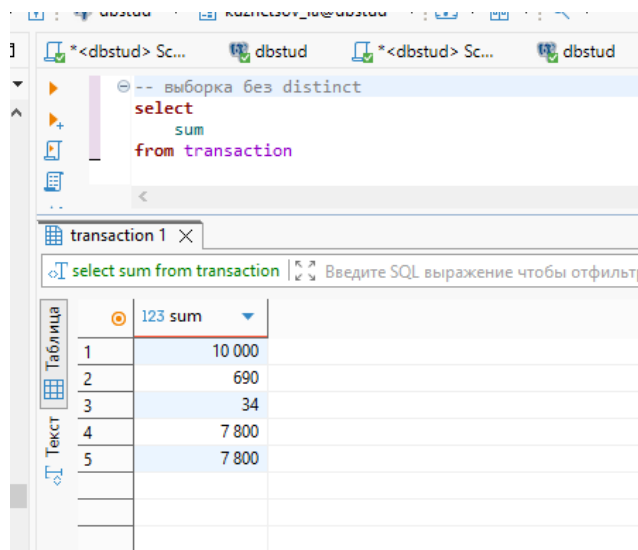


Рисунок 5 – Выборка по столбцу

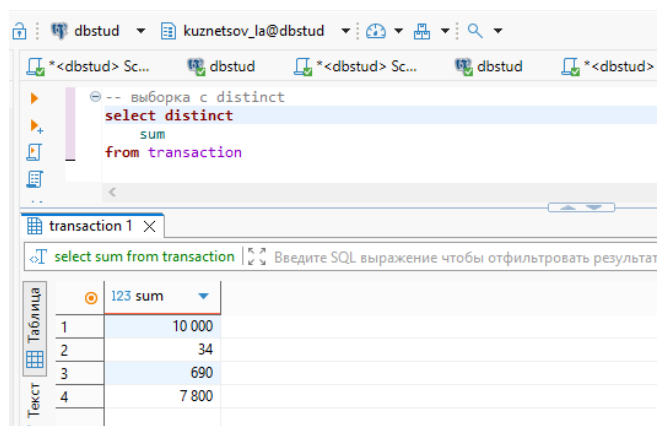


Рисунок 6 – Выборка по неповторяющемуся столбцу

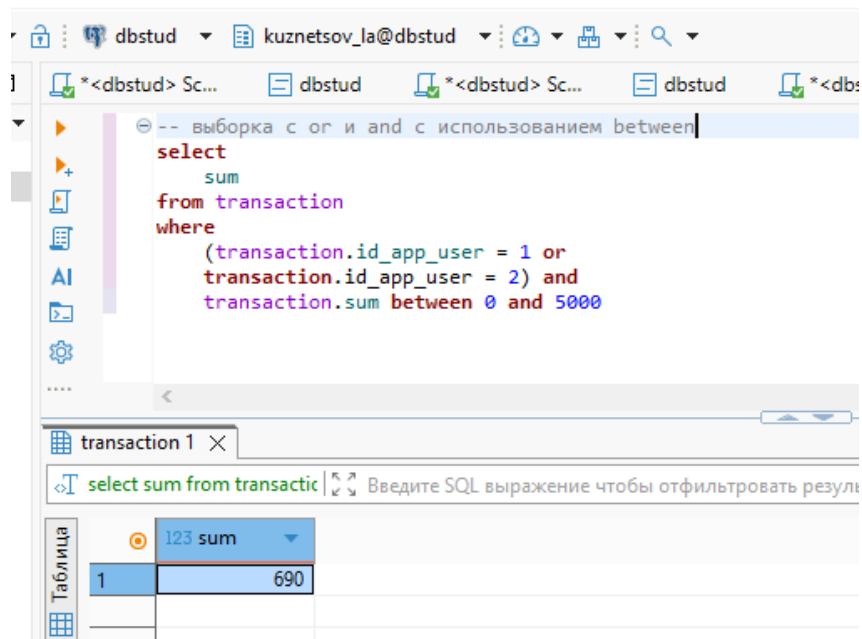


Рисунок 7 – Выборка с использованием or, and и between

На рисунке 7 представлена выборка с использованием between, что наглядно даёт понять принцип работы таких ключевых слов как min и max.

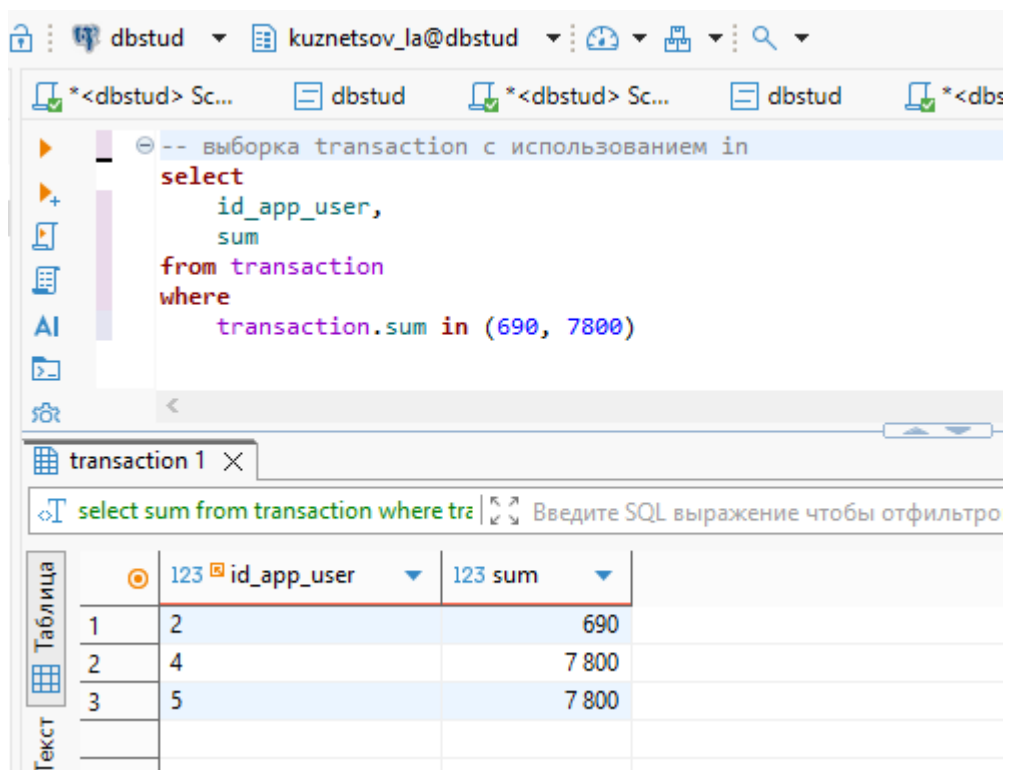


Рисунок 8 – Выборка с использованием in

SQL Query:

```
-- выборка transaction с использованием is not null
select
  id_app_user,
  sum
from transaction
where
  transaction.sum is not null
```

Result Table:

	id_app_user	sum
1	1	10 000
2	2	690
3	3	34
4	4	7 800
5	5	7 800

Рисунок 9 – Выборка при помощи where с is not null

5. Составление запросов на выборку

SQL Query:

```
-- сортировка по возрастанию
select
  id_app_user,
  sum
from transaction
order by
  transaction.sum asc
```

Result Table:

	id_app_user	sum
1	3	34
2	2	690
3	4	7 800
4	5	7 800
5	1	10 000

Рисунок 10 – Сортировка по возрастанию (ключевое слово - asc)

SQL Query:

```
-- сортировка по убыванию
select
  id_app_user,
  sum
from transaction
order by
  transaction.sum desc
```

Result Table:

	id_app_user	sum
1	1	10 000
2	4	7 800
3	5	7 800
4	2	690
5	3	34

Рисунок 11 - Сортировка по убыванию (ключевое слово - desc)

-- комбинация сортировок по возрастанию и убыванию

```
select
    id_app_user,
    sum
from transaction
order by
    transaction.sum desc,
    transaction.id_app_user asc
```

transaction 1 X

select id_app_user, sum from transact | Введите SQL выражение чтобы отфильтровать

	123 id_app_user	123 sum
1	1	10 000
2	4	7 800
3	5	7 800
4	2	690
5	3	34

Рисунок 12 – Комбинация сортировок desc и asc

-- комбинация сортировок по возрастанию и убыванию

```
select
    id_app_user,
    sum
from transaction
order by
    transaction.sum desc,
    transaction.id_app_user desc
```

transaction 1 X

select id_app_user, sum from transact | Введите SQL выражение чтобы отфильтровать

	123 id_app_user	123 sum
1	1	10 000
2	5	7 800
3	4	7 800
4	2	690
5	3	34

Рисунок 13 - Комбинация сортировок desc и desc

Как видно из рисунков 12-13 сначала производится сортировка по sum и при обнаружении дубликатов в sum далее будет проведена сортировка по id_app_user (обратите внимание на 2-ую и 3-ю строки в id_app_user в окне вывода на обоих рисунках).

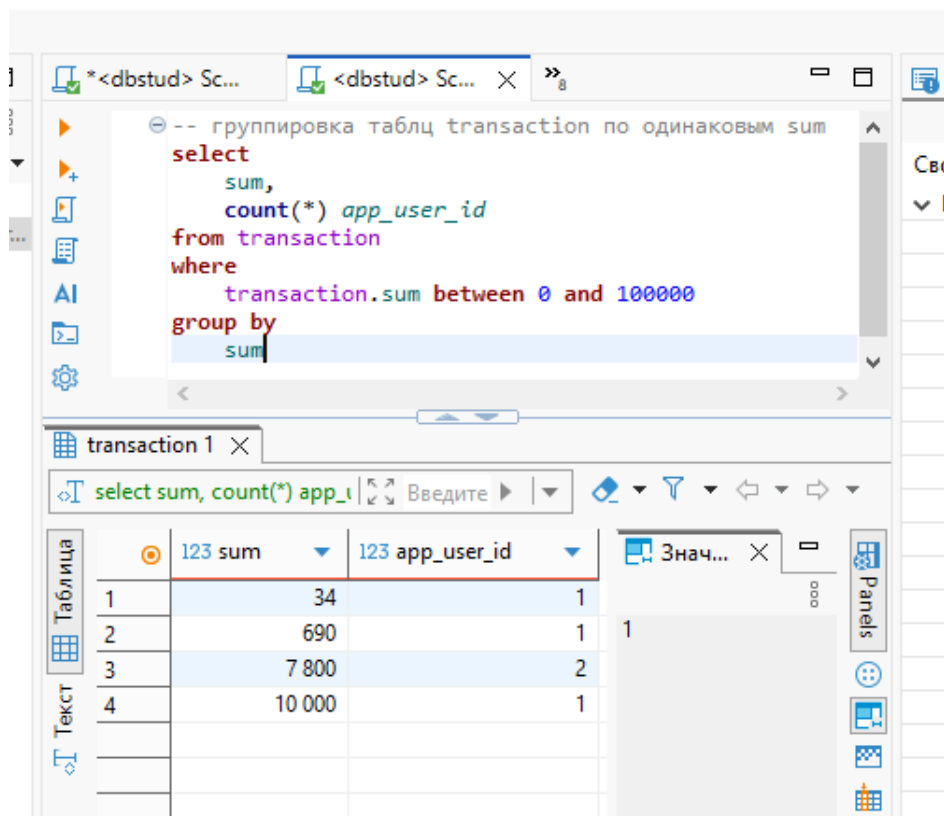


Рисунок 14 – Группировка таблиц по app_user_id

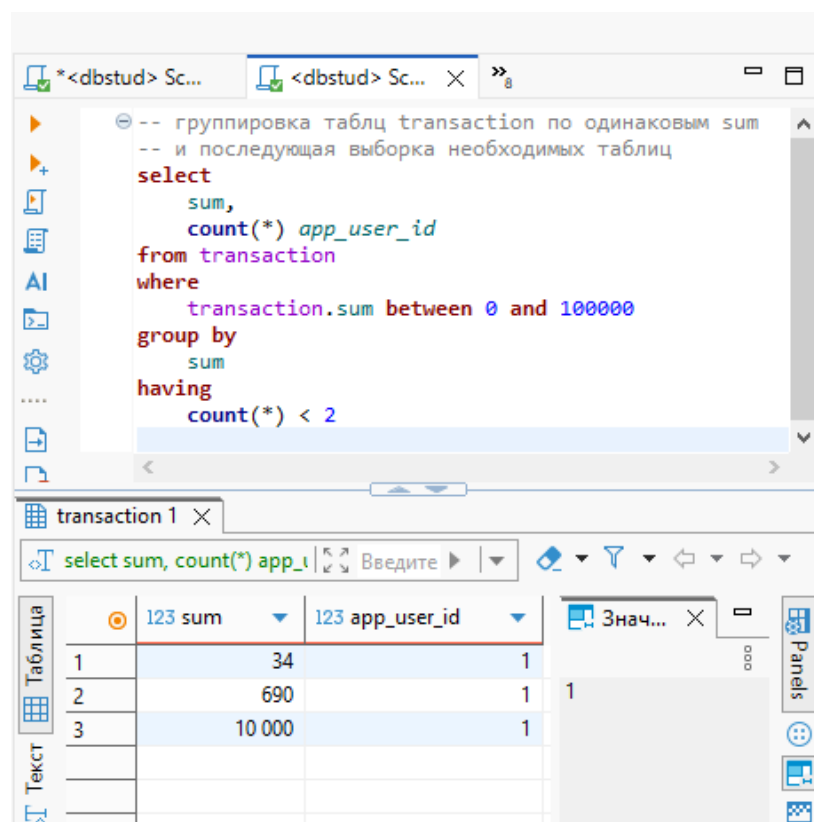


Рисунок 15 - Группировка таблиц по app_user_id с дополнительным условием

ОТВЕТЫ НА ВОПРОСЫ

1. Отличие в том, что primary key – это комбинация unique и not null. primary key не допускает значения null, unique - допускает;
2. Дочерняя таблица по данному foreign key удалит все свои связанные дочерние таблицы;
3. from → where → group by → having → select → order by;
4. Потому что нельзя обратиться к Nil псевдониму;
5. Золотое правило group by – любой столбец указанный в select должен находиться либо в group by, либо в агрегатной функции. Потому что так СУБД сможет сгруппировать столбцы по контролируемому правилу.
6. Отсортировать transaction по count(*) app_user_id Найти все transaction с sum в пределах от 100 до 1000 и при этом итоговое кол-во id_app_user должно быть меньше 2.

ВЫВОД

В ходе работы были получены и закреплены практическим путём знания и навыки по работе с СУБД.