

**Федеральное государственное автономное образовательное
учреждение высшего образования
Национальный исследовательский университет
"Высшая школа экономики"**

Московский институт электроники и математики имени А. Н. Тихонова
Программа "Прикладная математика"

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

ЗАДАНИЯ №1-4

**Группа БПМ213
Вариант 7**

Выполнил:
Варфоломеева Анастасия Андреевна

Москва, 2024 г.

1 Описание модели

Модель «Заказ билетов» должна содержать информацию о пассажире, билете, рейсе самолета

Использовать следующие справочники и классификаторы:

- типы самолетов
- справочник рейсов
- тип рейса (регулярный, чартер, специальный)

2 Задание 1

Рассматриваемый автоматизируемый бизнес процесс - покупка билета на самолет.

Название роли	Решаемые задачи
Покупатель	Заполнение анкет и документов, выбор предпочтительного рейса, покупка билета
Колл-центр	Оценка правильности заполнения анкет и документов
Специалист по подбору направлений	Составление списка лучших направлений в указанные даты
Специалист по подбору рейсов	Составление списка рейсов по предпочтительным направлениям и ценам
Специалист по бронированию посадочных мест	Бронирование предпочтительного для пассажира места

Таблица 1: Роли пользователей в процессе

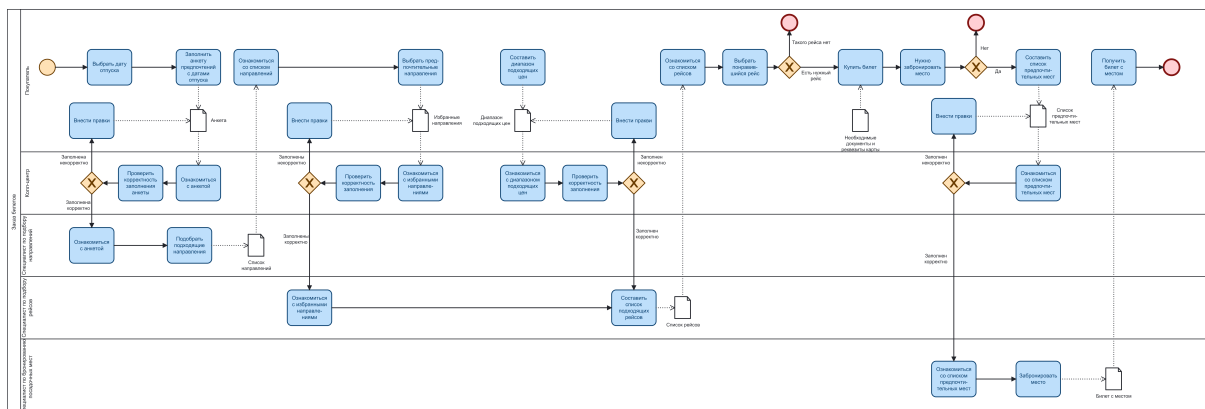


Рис. 1: BPMN диаграмма процесса

3 Задание 2

3.1 Экспорт данных из таблицы

Для того, чтобы извлечь данные из таблицы route использовалась программа с открытым исходным кодом для работы с базами данных pgAdmin.

	id [PK] uuid	from character varying (256)	to character varying (256)
1	1bb415c2-87b1-4a00-b20f-35d6a072a1...	Москва	Иркутск
2	1f78b16e-af60-4c46-af7e-75f1ea442c7a	Москва	Екатеринбург
3	440d26ef-423a-4ac6-a4b9-4331179f66...	Санкт-Петербург	Москва
4	68c76267-0272-4c65-a924-3aedc595fc...	Москва	Санкт-Петербург
5	7f6ceb94-44c1-4d6e-909a-10b0ce815b...	Самара	Москва
6	8c0a9e48-c9a2-4609-9217-748096fc96...	Екатеринбург	Москва
7	b897d00a-0acc-4ab6-adb8-72db8aaa9f...	Москва	Самара
8	bd1b0322-e8d0-406b-9e17-287e2641a...	Иркутск	Москва
9	f1dddfd2-a5b6-4545-bf4e-36371b4e00ee	Ростов-на-Дону	Москва
10	f522e10f-6ad2-49ba-8fc3-e7de6d4dab6e	Москва	Ростов-на-Дону

Рис. 2: Изначальное содержание таблицы

Import/Export data - table 'route'

General
Options
Columns

Import/Export

Import
Export

Filename

/home/varfolomeeva/HSE/db/exp

Format

csv

Encoding

UTF8

i
?

Close
Reset
OK

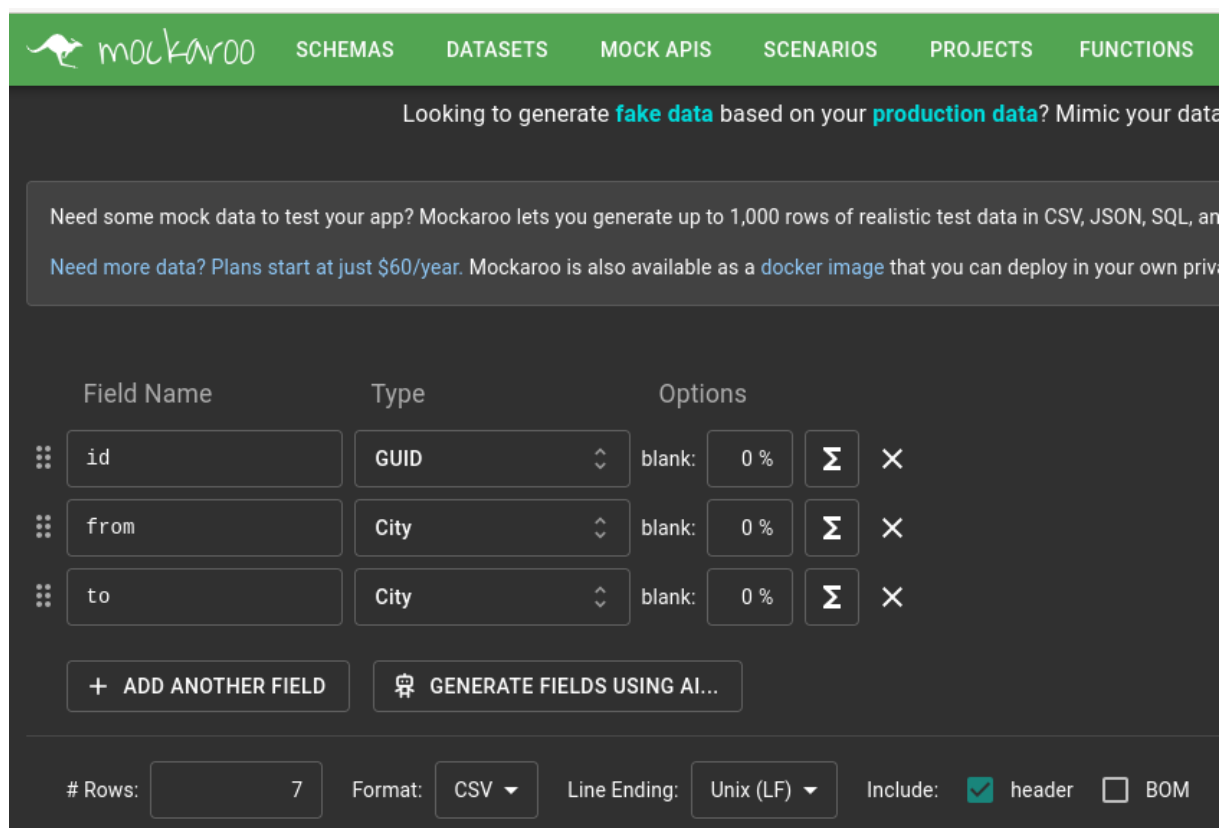
Рис. 3: Импорт данных из таблицы

	A	B	C
1	68c76267-0272-4c65-a924-3aedc595fcb6	Москва	Санкт-Петербург
2	440d26ef-423a-4ac6-a4b9-4331179f6601	Санкт-Петербург	Москва
3	1f78b16e-af60-4c46-af7e-75f1ea442c7a	Москва	Екатеринбург
4	8c0a9e48-c9a2-4609-9217-748096fc9668	Екатеринбург	Москва
5	f522e10f-6ad2-49ba-8fc3-e7de6d4dab6e	Москва	Ростов-на-Дону
6	f1dddfd2-a5b6-4545-bf4e-36371b4e00ee	Ростов-на-Дону	Москва
7	b897d00a-0acc-4ab6-adb8-72db8aaa9f80	Москва	Самара
8	7f6ceb94-44c1-4d6e-909a-10b0ce815b76	Самара	Москва
9	1bb415c2-87b1-4a00-b20f-35d6a072a1eb	Москва	Иркутск
10	bd1b0322-e8d0-406b-9e17-287e2641acf1	Иркутск	Москва

Рис. 4: csv файл с данными из таблицы route

3.2 Создание новых данных

Для добавления новых маршрутов в таблицу route использовался сайт Mockaroo для генерации данных. Был выполнен следующий запрос и получен соответствующий csv файл.



Looking to generate **fake data** based on your **production data**? Mimic your data

Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and more. Need more data? Plans start at just \$60/year. Mockaroo is also available as a [docker image](#) that you can deploy in your own private cloud.

Field Name	Type	Options
id	GUID	blank: 0 % Σ X
from	City	blank: 0 % Σ X
to	City	blank: 0 % Σ X

+ ADD ANOTHER FIELD 🤖 GENERATE FIELDS USING AI...

Rows: 7 Format: CSV Line Ending: Unix (LF) Include: ☒ header ☐ BOM

Рис. 5: Запрос на сайте Mockaroo

	A	B	C
1	id	from	to
2	7adbe677-4c21-5a33-b202-75880aad8911	Szklarska Poreba	Qianjin
3	ea3c1dc6-b43b-5321-ab6d-3a0acae8907c	Otjimbingwe	Tombatu
4	01e83cae-b150-5cc5-81ac-3694a178cc04	Renhe	Vilppula
5	942fe0cc-8a0b-5b61-ba9f-db4adbd3f322	Shanjiang	Honda
6	ff389683-51a6-54d7-a1a8-9fbefc0a8d82	Tococ East	Penja
7	cbf78b3f-7aa5-5556-8f68-21da78625104	Zhichanghe	Napak
8	eb46f97e-ad5c-522c-b484-48448407e00d	Caringin	Serov

Рис. 6: Csv файл с новыми маршрутами для импорта в базу данных

При помощи функции импорта из формата csv в pgAdmin, в таблицу route были добавлены новые рейсы.

	id [PK] uuid	from character varying (256)	to character varying (256)
1	01e83cae-b150-5cc5-81ac-3694a178c...	Renhe	Vilppula
2	1bb415c2-87b1-4a00-b20f-35d6a072a...	Москва	Иркутск
3	1f78b16e-af60-4c46-af7e-75f1ea442c7a	Москва	Екатеринбург
4	440d26ef-423a-4ac6-a4b9-4331179f66...	Санкт-Петербург	Москва
5	68c76267-0272-4c65-a924-3aedc595fc...	Москва	Санкт-Петербург
6	7adbe677-4c21-5a33-b202-75880aad8...	Szklarska Poreba	Qianjin
7	7f6ceb94-44c1-4d6e-909a-10b0ce815b...	Самара	Москва
8	8c0a9e48-c9a2-4609-9217-748096fc96...	Екатеринбург	Москва
9	942fe0cc-8a0b-5b61-ba9f-db4adbd3f3...	Shanjiang	Honda
10	b897d00a-0acc-4ab6-adb8-72db8aaa9f...	Москва	Самара
11	bd1b0322-e8d0-406b-9e17-287e2641a...	Иркутск	Москва
12	cbf78b3f-7aa5-5556-8f68-21da786251...	Zhichanghe	Napak
13	ea3c1dc6-b43b-5321-ab6d-3a0acae89...	Otjimbingwe	Tombatu
14	eb46f97e-ad5c-522c-b484-48448407e0...	Caringin	Serov
15	f1dddfd2-a5b6-4545-bf4e-36371b4e00...	Ростов-на-Дону	Москва
16	f522e10f-6ad2-49ba-8fc3-e7de6d4dab6e	Москва	Ростов-на-Дону
17	ff389683-51a6-54d7-a1a8-9fbefc0a8d82	Tococ East	Penja

Рис. 7: Таблица route после добавление новых маршрутов

3.3 Выгрузка результатов запроса

Также были выгружены результаты одного из запросов, реализованных в домашнем задании по курсу базы данных: получить список рейсов, на которых вылетали пассажиры с заданной фамилией и датой вылета.

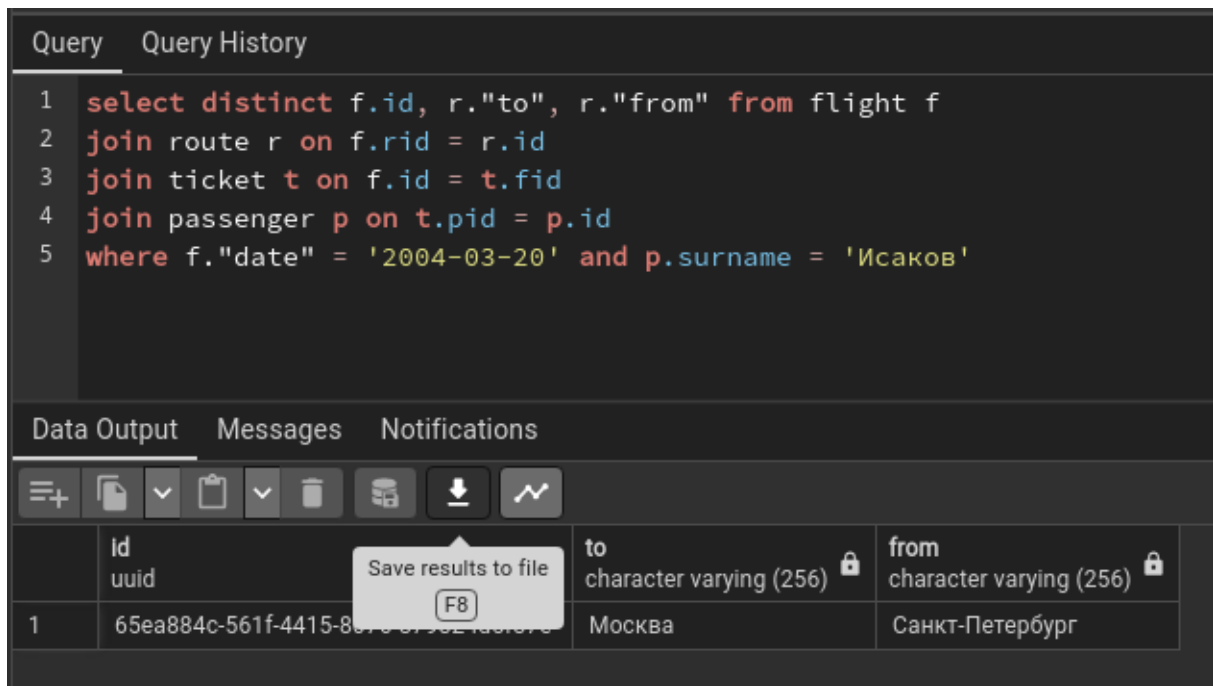


Рис. 8: Экспорт результатов запроса в csv файл

	A	B	C
1	id	to	from
2	65ea884c-561f-4415-8076-879324d5f37e	Москва	Санкт-Петербург

Рис. 9: Csv файл с результатами запроса

4 Задание 3

4.1 Создание таблицы для ведения записей

Прежде чем создать функцию-триггер, которая отслеживает изменения в таблице route, необходимо сделать таблицу route_log, в которой содержится информация об операции, дате ее проведения и имя пользователя, который внес изменения, а так же сама информация о записях в таблице route.

```
1 create table route_log(
2 operation char(6) NOT NULL,
3 "date" timestamp NOT NULL,
4 userid text NOT NULL,
5 "id" varchar(256),
6 "from" varchar(256),
7 "to" varchar(256)
8 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 46 msec.

Рис. 10: Таблица route_log

4.2 Создание триггера

Была создана функция-триггер записывающая изменения из таблицы route в таблицу route_log. Для определения типа операции была использована переменная TG_OP, для обращения к новым данным переменная New, а к старым - Old. Функция была прикреплена к операциям вставки и удаления в таблицы route, и вносила изменения для каждой измененной записи.

```
1 CREATE OR REPLACE FUNCTION log_route() RETURNS TRIGGER AS $log_route$
2▼ BEGIN
3▼ IF (TG_OP = 'DELETE') THEN
4 INSERT INTO route_log SELECT 'DELETE', now(), user, OLD.*;
5 ELSIF (TG_OP = 'UPDATE') THEN
6 INSERT INTO route_log SELECT 'UPDATE', now(), user, NEW.*;
7 END IF;
8 RETURN NULL;
9 END;
10 $log_route$ LANGUAGE plpgsql;
11 CREATE TRIGGER log_route
12 AFTER UPDATE OR DELETE ON route
13 FOR EACH ROW EXECUTE PROCEDURE log_route();
```

Data Output Messages Notifications

CREATE TRIGGER

Query returned successfully in 41 msec.

Рис. 11: Триггер-функция для отслеживания изменений

4.3 Проверка работы триггера

Для проверки работы триггера были выполнены операции вставки и удаления записи.

```
1 UPDATE route
2 SET "from"='Нью-Йорк' WHERE "from"='Москва'
```

Data Output Messages Notifications

UPDATE 5

Query returned successfully in 41 msec.

Рис. 12: Операция изменения

```
1 DELETE FROM route
2 WHERE "id"='b897d00a-0acc-4ab6-adb8-72db8aaa9f80'
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 72 msec.

Рис. 13: Операция удаления

Операции прошли успешно, и в таблице route_log появились записи

```
1 SELECT * FROM public.route_log
2
```

Data Output Messages Notifications

	operation character	date timestamp without time zone	userid text	id character varying (256)	from character varying (256)	to character varying (256)
1	UPDATE	2024-10-22 16:27:17.715834	postgres	68c76267-0272-4c65-a924-3aedc595fcb6	Нью-Йорк	Санкт-Петербург
2	UPDATE	2024-10-22 16:27:17.715834	postgres	1f78b16e-af60-4c46-af7e-75f1ea442c7a	Нью-Йорк	Екатеринбург
3	UPDATE	2024-10-22 16:27:17.715834	postgres	f522e10f-6ad2-49ba-8fc3-e7de6d4dab6e	Нью-Йорк	Ростов-на-Дону
4	UPDATE	2024-10-22 16:27:17.715834	postgres	b897d00a-0acc-4ab6-adb8-72db8aaa9f80	Нью-Йорк	Самара
5	UPDATE	2024-10-22 16:27:17.715834	postgres	1bb415c2-87b1-4a00-b20f-35d6a072a1eb	Нью-Йорк	Иркутск
6	DELETE	2024-10-22 16:28:57.68201	postgres	b897d00a-0acc-4ab6-adb8-72db8aaa9f80	Нью-Йорк	Самара

Рис. 14: Записи в таблице route_log

5 Задание 4

5.1 Создание резервной копии БД

Резервная копия БД была создана через pgAdmin.

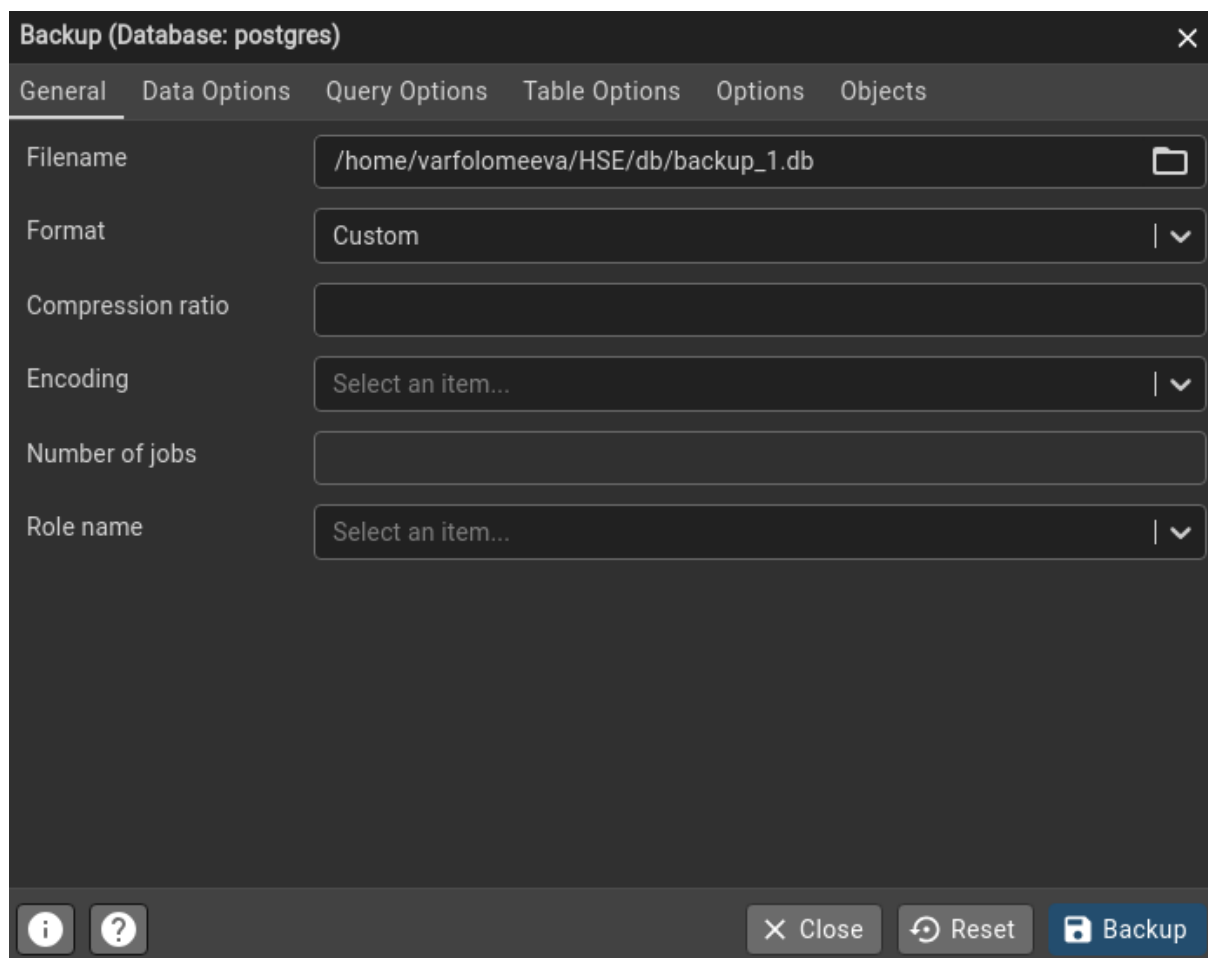


Рис. 15: Создание резервной копии БД

5.2 Удаление данных из БД

При помощи запроса были удалены все записи из БД.

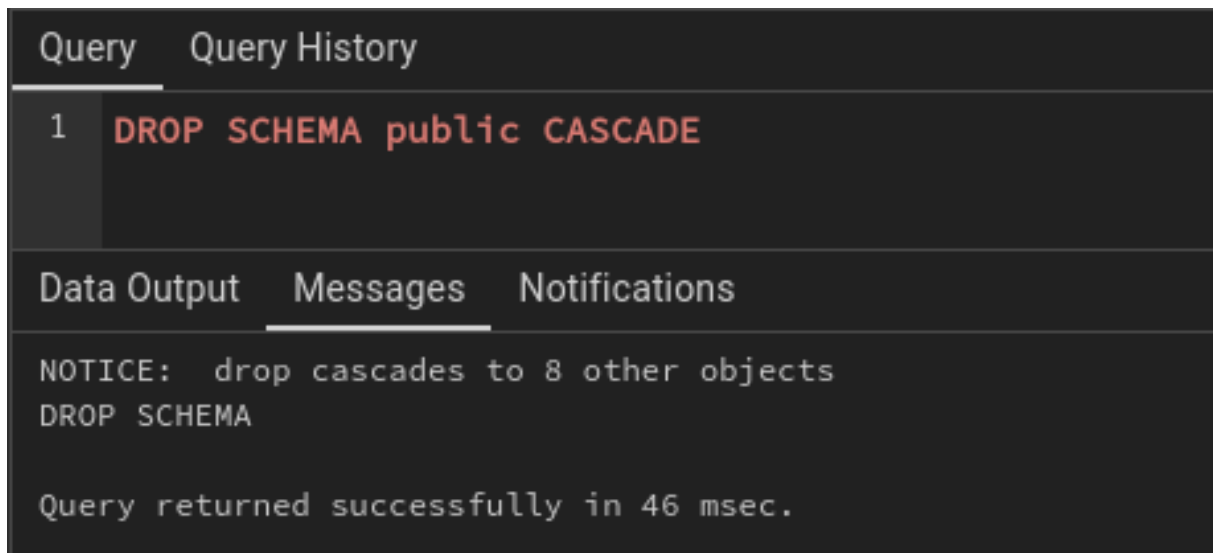


Рис. 16: Запрос для удаления всех таблиц из БД

5.3 Восстановление БД из резервной копии

Затем БД была восстановлена из созданной резервной копии через pgAdmin.

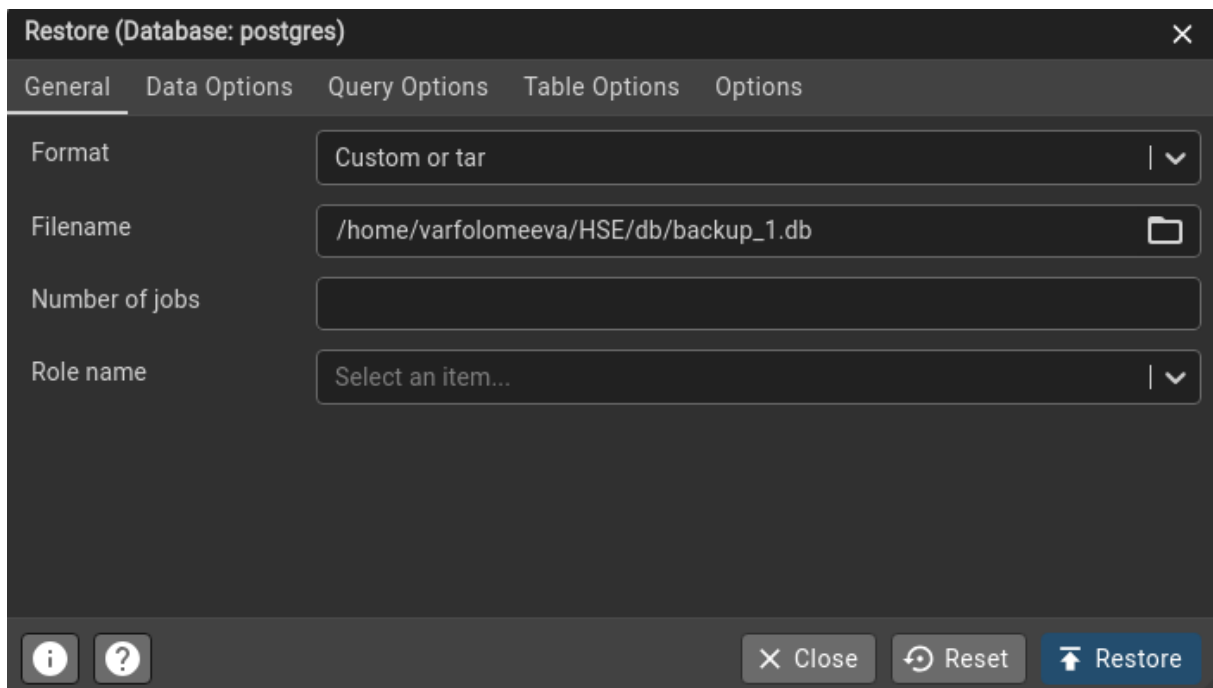


Рис. 17: Восстановление БД из резервной копии


```

create table PASSENGER(
    "id" uuid primary key,
    surname varchar(256),
    "name" varchar(256),
    patronymic varchar(256)
);
create table ROUTE(
    "id" uuid primary key,
    "from" varchar(256),
    "to" varchar(256)
);
create table FLIGHT_TYPE(
    "id" uuid primary key,
    "type" varchar(256)
);
create table AIRPLANE_TYPE(
    "id" uuid primary key,
    "type" varchar(256)
);
create table FLIGHT(
    "id" uuid primary key,
    FTID uuid references FLIGHT_TYPE("id"),
    ATID uuid references AIRPLANE_TYPE("id"),
    RID uuid references ROUTE("id"),
    "date" date
);
create table TICKET(
    "id" uuid primary key,
    seat varchar(16),
    PID uuid references PASSENGER("id"),
    FID uuid references FLIGHT("id")
);

INSERT INTO ROUTE VALUES('68c76267-0272-4c65-a924-3aedc595fcb6', 'Москва', 'Санкт-Петербург');
INSERT INTO ROUTE VALUES('440d26ef-423a-4ac6-a4b9-4331179f6601', 'Санкт-Петербург', 'Москва');
INSERT INTO ROUTE VALUES('1f78b16e-af60-4c46-af7e-75f1ea442c7a', 'Москва', 'Екатеринбург');
INSERT INTO ROUTE VALUES('8c0a9e48-c9a2-4609-9217-748096fc9668', 'Екатеринбург', 'Москва');
INSERT INTO ROUTE VALUES('f522e10f-6ad2-49ba-8fc3-e7de6d4dab6e', 'Москва', 'Ростов-на-Дону');
INSERT INTO ROUTE VALUES('f1dddfd2-a5b6-4545-bf4e-36371b4e00ee', 'Ростов-на-Дону', 'Москва');
INSERT INTO ROUTE VALUES('b897d00a-0acc-4ab6-adb8-72db8aaa9f80', 'Москва', 'Самара');
INSERT INTO ROUTE VALUES('7f6ceb94-44c1-4d6e-909a-10b0ce815b76', 'Самара', 'Москва');
INSERT INTO ROUTE VALUES('1bb415c2-87b1-4a00-b20f-35d6a072a1eb', 'Москва', 'Иркутск');
INSERT INTO ROUTE VALUES('bd1b0322-e8d0-406b-9e17-287e2641acf1', 'Иркутск', 'Москва');
INSERT INTO AIRPLANE_TYPE VALUES('0ec6dbc4-fc69-4506-9dd3-b5d62b35fc7b', 'Airbus A340');
INSERT INTO AIRPLANE_TYPE VALUES('6a916c63-a33f-46f1-9e7c-08ba4ac361d3', 'Airbus A320');
INSERT INTO AIRPLANE_TYPE VALUES('e2f5d080-6697-4df5-97f7-b153a9c1711c', 'Boeing 737');

```

```

INSERT INTO AIRPLANE_TYPE VALUES('d3665acc-8c9a-49d0-a575-c43e28bf4f9a', 'Boeing 777');
INSERT INTO AIRPLANE_TYPE VALUES('7ee6e78e-bce4-48f5-9f3a-3bdf2b1a7409', 'Sukhoi Superjet');
INSERT INTO FLIGHT_TYPE VALUES('b8d60677-c999-46c8-8ce8-59a55d59eb32', 'Регулярный');
INSERT INTO FLIGHT_TYPE VALUES('79877549-08b0-454b-bbf5-277978ca2202', 'Чартерный');
INSERT INTO FLIGHT_TYPE VALUES('1b5e687f-2278-47d8-968e-69b4c5d3c1fc', 'Специальный');
INSERT INTO PASSENGER VALUES('e9c58c6b-2717-4f01-a677-9c2769fc8fda', 'Суворов', 'Остап');
INSERT INTO PASSENGER VALUES('dbd9caa9-7d5c-4485-bfa2-6856ee8b7aaa', 'Киселева', 'Зоя');
INSERT INTO PASSENGER VALUES('400b81ea-f526-4164-8420-810c2513c721', 'Лебедева', 'Антон');
INSERT INTO PASSENGER VALUES('679160c3-eeb7-4699-9b3e-fec324c0b8b5', 'Беляков', 'Амос');
INSERT INTO PASSENGER VALUES('8102ad81-a079-4d08-b44a-4641da87b2a1', 'Лебедева', 'Евдокия');
INSERT INTO PASSENGER VALUES('34e3743f-b7cb-4148-8ddb-adf8abec894e', 'Колесников', 'Евгений');
INSERT INTO PASSENGER VALUES('babf471e-960a-41fc-b333-737f4be3b518', 'Лебедева', 'Евфимий');
INSERT INTO PASSENGER VALUES('c7699198-86c9-4391-be0a-c41409042eb2', 'Исаков', 'Демьян');
INSERT INTO PASSENGER VALUES('fe44906b-0860-4d8d-9e0a-a80545c74134', 'Евдокимов', 'Дарья');
INSERT INTO PASSENGER VALUES('548af5e9-03d3-449e-9714-d6c96329fcce', 'Абрамова', 'Раиса');
INSERT INTO FLIGHT VALUES('65ea884c-561f-4415-8076-879324d5f37e', '79877549-08b0-454b-bbf5-277978ca2202');
INSERT INTO FLIGHT VALUES('ae7bf53d-2b1f-4b58-942d-55a3ef757391', '79877549-08b0-454b-bbf5-277978ca2202');
INSERT INTO FLIGHT VALUES('26201c12-65e3-40bc-b41e-83a99941c217', 'b8d60677-c999-46c8-8ce8-59a55d59eb32');
INSERT INTO FLIGHT VALUES('badb32e0-6f50-4156-97ef-68628b4d46ea', 'b8d60677-c999-46c8-8ce8-59a55d59eb32');
INSERT INTO FLIGHT VALUES('c88911df-c1e5-4f4f-a525-aadf568f90de', '79877549-08b0-454b-bbf5-277978ca2202');
INSERT INTO FLIGHT VALUES('9d41a46e-fb75-434f-a5ff-765a5c77b317', 'b8d60677-c999-46c8-8ce8-59a55d59eb32');
INSERT INTO FLIGHT VALUES('ed1f34fb-b5b1-49fd-8e5f-075f308888ef', 'b8d60677-c999-46c8-8ce8-59a55d59eb32');
INSERT INTO FLIGHT VALUES('e3831f26-0969-4778-82c6-03eda19d32c0', '79877549-08b0-454b-bbf5-277978ca2202');
INSERT INTO FLIGHT VALUES('60812ff6-1667-45a2-814c-0c614aa73e5d', '79877549-08b0-454b-bbf5-277978ca2202');
INSERT INTO FLIGHT VALUES('1f0932dd-9821-4e5e-91c7-71ab78c506fa', 'b8d60677-c999-46c8-8ce8-59a55d59eb32');
INSERT INTO TICKET VALUES('179c2ae2-098a-4a88-9ad0-a825504453f7', '19E', '8102ad81-a079-4d08-b44a-4641da87b2a1');
INSERT INTO TICKET VALUES('ac3756b5-a7f2-4cda-9a0c-f72c5b0b4145', '24B', 'c7699198-86c9-4391-be0a-c41409042eb2');
INSERT INTO TICKET VALUES('024ee9c2-f4e9-4eee-8596-8d4a03465596', '19A', '400b81ea-f526-4164-8420-810c2513c721');
INSERT INTO TICKET VALUES('2418c31c-d693-4f53-8dff-ad1cc49ac3dd', '19A', 'dbd9caa9-7d5c-4485-bfa2-6856ee8b7aaa');
INSERT INTO TICKET VALUES('df065cb4-d1f9-464d-96bc-cc51e197aa40', '23C', 'fe44906b-0860-4d8d-9e0a-a80545c74134');
INSERT INTO TICKET VALUES('b4971d1b-27c0-4c83-83bc-e901ea1e86ec', '17B', NULL, 'ae7bf53d-2b1f-4b58-942d-55a3ef757391');
INSERT INTO TICKET VALUES('9ba57cba-f1c8-43a0-b6d1-d973c306d46f', '18B', 'fe44906b-0860-4d8d-9e0a-a80545c74134');
INSERT INTO TICKET VALUES('405fafaa-23bc-4583-8a81-8f597f5617eb', '24F', 'fe44906b-0860-4d8d-9e0a-a80545c74134');
INSERT INTO TICKET VALUES('ea647803-ec5c-4a10-be63-57bc565039a2', '20A', '679160c3-eeb7-4699-9b3e-fec324c0b8b5');
INSERT INTO TICKET VALUES('e812ccce-b744-4d26-a3fd-616d655d9585', '25C', 'babf471e-960a-41fc-b333-737f4be3b518');

```

6.3 Описание таблиц

PASSENGER			
id	UUID	PRIMARY KEY	Уникальный идентификатор пассажира
surname	VARCHAR(256)	-	Фамилия пассажира
name	VARCHAR(256)	-	Имя пассажира
patronymic	VARCHAR(256)	-	Отчество пассажира

ROUTE			
id	UUID	PRIMARY KEY	Уникальный идентификатор маршрута
from	VARCHAR(256)	-	Город отправления

to	VARCHAR(256)	-	Город прибытия
----	--------------	---	----------------

FLIGHT_TYPE			
id	UUID	PRIMARY KEY	Уникальный идентификатор типа рейса
type	VARCHAR(256)	-	Тип рейса

AIRPLANE_TYPE			
id	UUID	PRIMARY KEY	Уникальный идентификатор типа самолета
type	VARCHAR(256)	-	Тип самолета

FLIGHT			
id	UUID	PRIMARY KEY	Уникальный идентификатор рейса
FTID	UUID	REFERENCES FLIGHT_TYPE(id)	Уникальный идентификатор типа рейса
ATID	UUID	REFERENCES AIRPLANE_TYPE(id)	Уникальный идентификатор типа самолета
RID	UUID	REFERENCES ROUTE(id)	Уникальный идентификатор маршрута
date	DATE	-	Дата рейса

TICKET			
id	UUID	PRIMARY KEY	Уникальный идентификатор страны
seat	VARCHAR(16)	-	Посадочное место
PID	UUID	REFERENCES PASSENGER(id)	Уникальный идентификатор пассажира
FID	UUID	REFERENCES FLIGHT(id)	Уникальный идентификатор рейса