

Санкт–Петербургский Национальный Исследовательский Университет
Информационных технологий, Механики и Оптики

Лабораторная работа по дисциплине “Разработка и администрирование баз
данных”
на тему
“Установка СУБД PostgreSQL. Создание базы данных в pgadmin.
Создание таблиц базы данных PostgreSQL. Заполнение таблиц рабочими
данными.”

Выполнил: студент гр. D41421 Мамонова А.А.

г. Санкт–Петербург
2020 г.

Лабораторная работа № 2

Установка СУБД PostgreSQL. Создание БД в pgadmin.

Цель работы: овладеть практическими навыками установки СУБД PostgreSQL 11 и создания базы данных в pgadmin 4, овладеть практическими навыками создания таблиц базы данных PostgreSQL 11, заполнения их рабочими данными, резервного копирования и восстановления БД.

Ход работы:

Установка PostgreSQL.

В рамках обучения на первом семестре магистратуры мною уже был установлен PostgreSQL версии 11.5, рисунок 1.

```
vladiknaska@MacBook-Air-VladikNaska ~/D/s/d/ITMO_IMRIP_DataBases_2020>  
vladiknaska@MacBook-Air-VladikNaska ~/D/s/d/ITMO_IMRIP_DataBases_2020> psql --version  
psql (PostgreSQL) 11.5  
vladiknaska@MacBook-Air-VladikNaska ~/D/s/d/ITMO_IMRIP_DataBases_2020>
```

Рисунок 1. Установленный PostgreSQL

Установка pgadmin4.

Для установки был выбран пакетный менеджер Homebrew, т.к. это удобный способ установки ПО на macOS. Процесс установки показан на рисунке 2.

```
vladiknaska@MacBook-Air-VladikNaska ~/D/s/db> ls 19:31:58  
ITMO_IMRIP_DataBases_2020 schema.sql  
vladiknaska@MacBook-Air-VladikNaska ~/D/s/db> brew cask install pgadmin4 19:31:59  
Updating Homebrew...  
=>  
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-cask'...  
remote: Enumerating objects: 4, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 436656 (delta 0), reused 2 (delta 0), pack-reused 436652  
Receiving objects: 100% (436656/436656), 196.50 MiB | 2.71 MiB/s, done.  
Resolving deltas: 100% (309103/309103), done.  
Tapped 1 command and 3526 casks (3,639 files, 210.9MB).  
=>  
##### 100.0%  
=>  
pgadmin4  
=>  
📦 pgadmin4 was successfully installed!  
vladiknaska@MacBook-Air-VladikNaska ~/D/s/db> 19:40:14
```

Рисунок 2. Установка pgadmin4

Запуск pgAdmin 4 и подключение к серверу PostgreSQL 11

Для начала запущен сервер postgres, рисунок 3.

```
vladiknaska@MacBook-Air-VladikNaska ~/D/s/db> pg_ctl -D /usr/local/var/postgres start
waiting for server to start....2020-05-07 19:12:19.010 MSK [30336] LOG:  listening on IPv6 address ":::1", port 5432
2020-05-07 19:12:19.010 MSK [30336] LOG:  listening on IPv4 address "127.0.0.1", port 5432
2020-05-07 19:12:19.012 MSK [30336] LOG:  listening on Unix socket "/tmp/.s.PGSQL.5432"
2020-05-07 19:12:19.057 MSK [30337] LOG:  database system was shut down at 2020-01-03 16:33:41 MSK
2020-05-07 19:12:19.089 MSK [30336] LOG:  database system is ready to accept connections
done
server started
vladiknaska@MacBook-Air-VladikNaska ~/D/s/db> █
```

Рисунок 3. Запуск сервера

В веб-интерфейсе создано подключение к серверу, см. Рисунок 4.

Create - Server [X]

General **Connection** SSL SSH Tunnel Advanced

Host name/address:

Port:

Maintenance database:

Username:

Password:

Save password? ☐

Role:

Service:

[i] [?] [Cancel] [Reset] [Save]

Рисунок 4. Создание подключения

Создание базы данных

Создана база данных с помощью диалогового окна pgadmin4, рисунок 5.

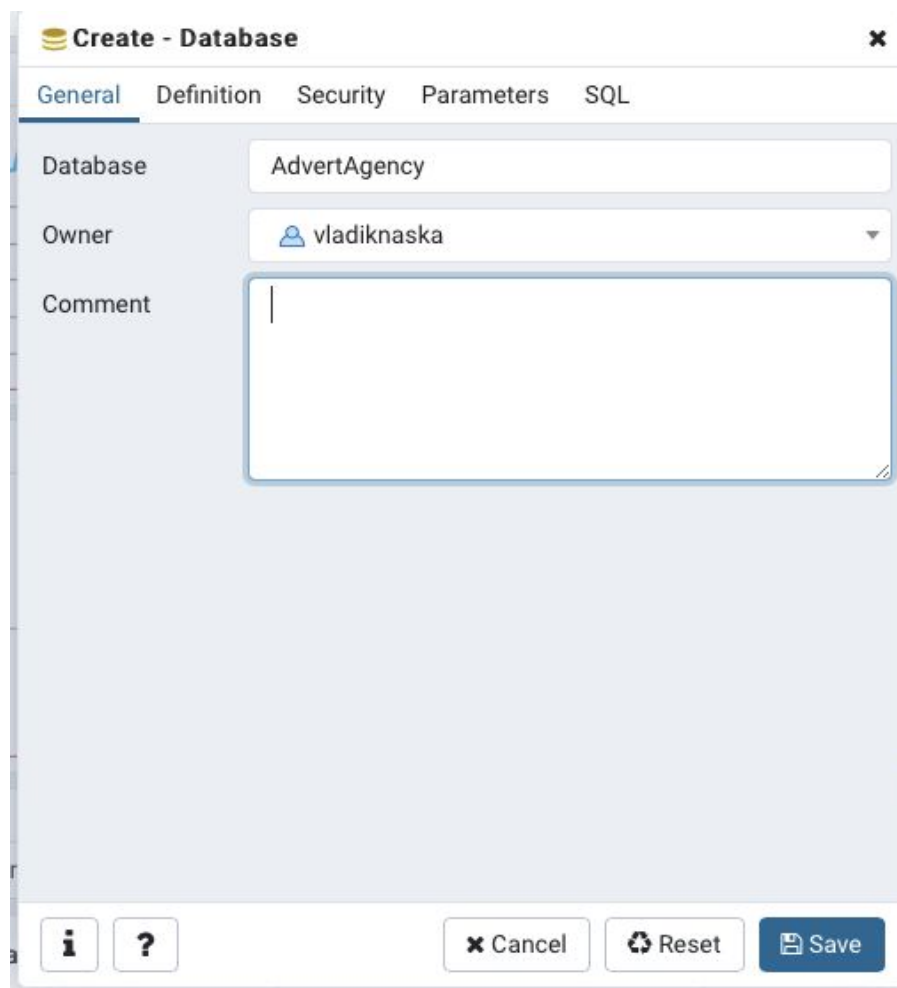


Рисунок 5. Создание БД

Создание таблиц базы данных PostgreSQL. Заполнение таблиц рабочими данными.

Создание схемы БД

Для определения схемы базы данных использовалось диалоговое окно Schema (Схема). Схема – это организационная рабочая область базы данных, похожая на каталоги или пространства имен, рисунок 6.

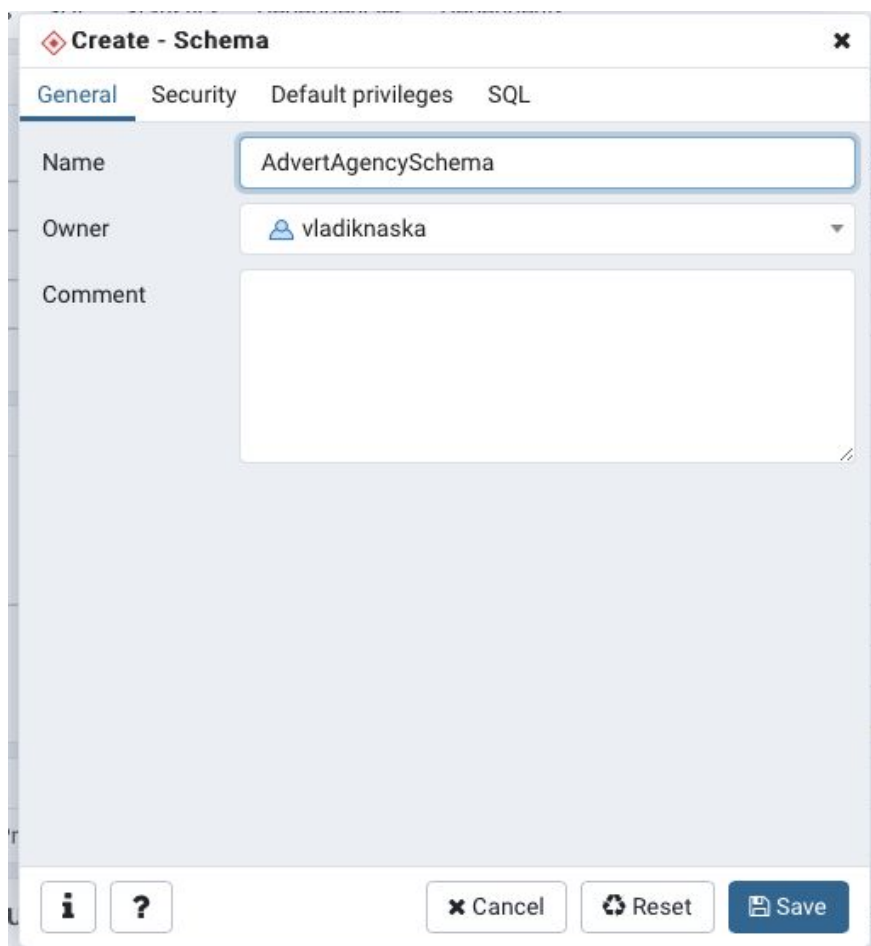


Рисунок 6. Создание схемы БД

Создание или изменение таблиц

PgAdmin 4 предоставляет диалоги, которые позволяют изменять все свойства и атрибуты таблицы. С помощью диалога *Таблица* создана таблица Customer (рисунок 7), Order, Material, OrderContent (рисунок 9), MaterialPerOrder, PriceList, Performer. Также был создан тип Enum для статуса заказа, рисунок 8.

Create - Table [X]

General **Columns** Constraints Advanced Partitions Parameters Security SQL

Inherited from table(s)

Columns +

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	id	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
	full_name	character	256		<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
	phone_number	character	30		<input type="checkbox"/> No	<input type="checkbox"/> No
	email	character			<input type="checkbox"/> No	<input type="checkbox"/> No

[i] [?] [Cancel] [Reset] [Save]

Рисунок 7. Создание таблицы Customer в диалоговом окне

status [X]

General **Definition** Security SQL

Type Enumeration

Enumeration type +

Label

- approved
- denied
- paid
- closed

[i] [?] [Cancel] [Reset] [Save]

Рисунок 8. Создание enum



Рисунок 9. Таблица OrderContent

Задание ограничений для таблицы

С помощью диалогового окна «*Check*» определены проверочные ограничения: наличие знака “@” в поле email, рисунок 10. Ограничение проверки определяет выражение, которое выдает логический результат, которому должны удовлетворять новые или обновленные строки, чтобы операция вставки или обновления выполнялась успешно.

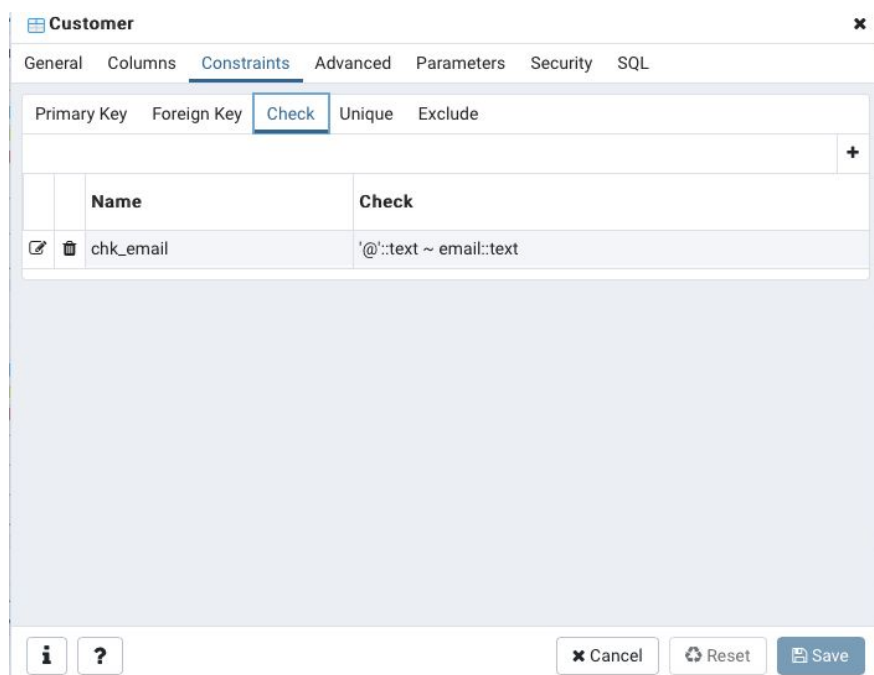


Рисунок 10. Проверка на содержание ‘@’ в email

Также, заданы ограничения Foreign key для всех таблиц, пример для таблицы Orders представлен на рисунке 11.

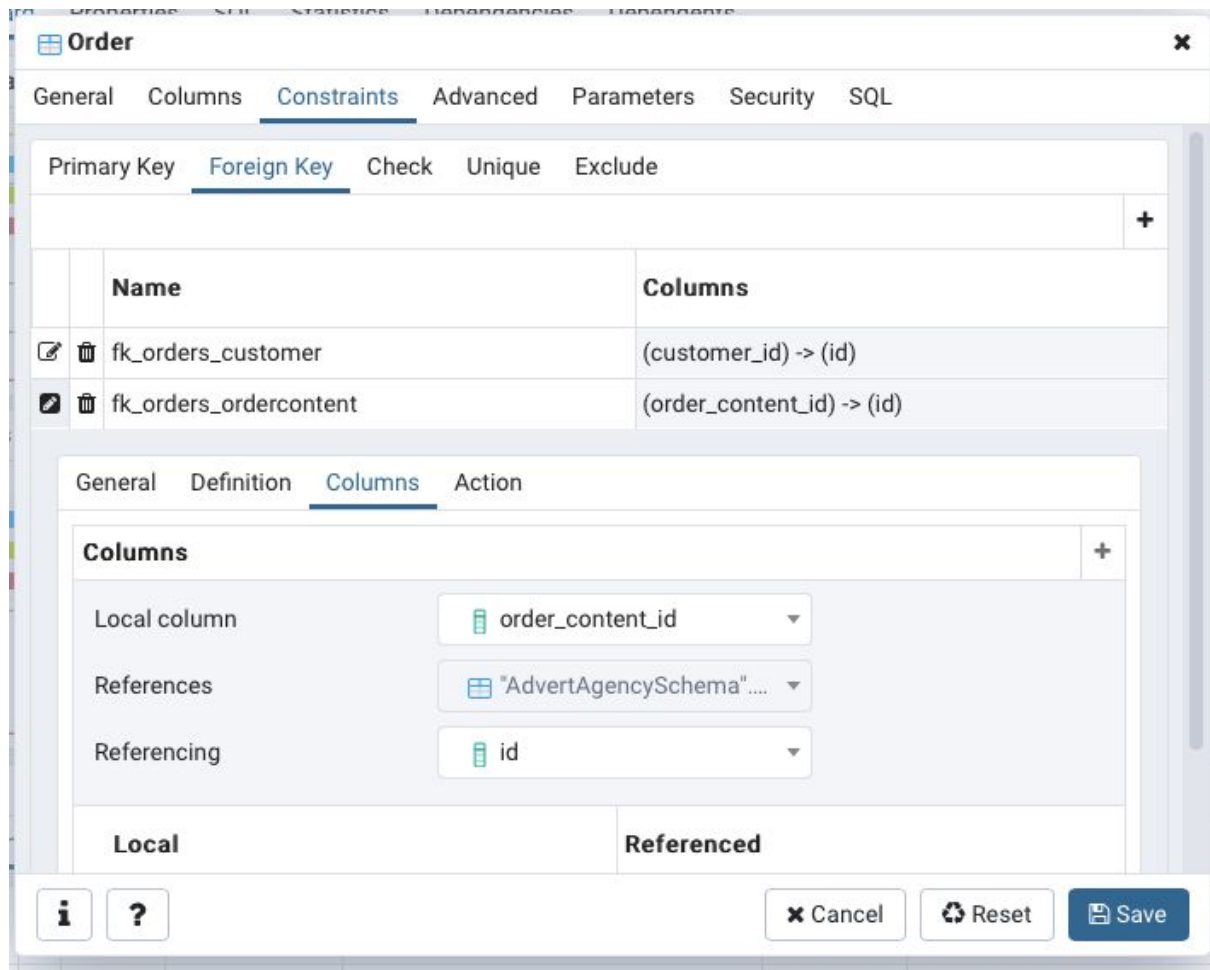
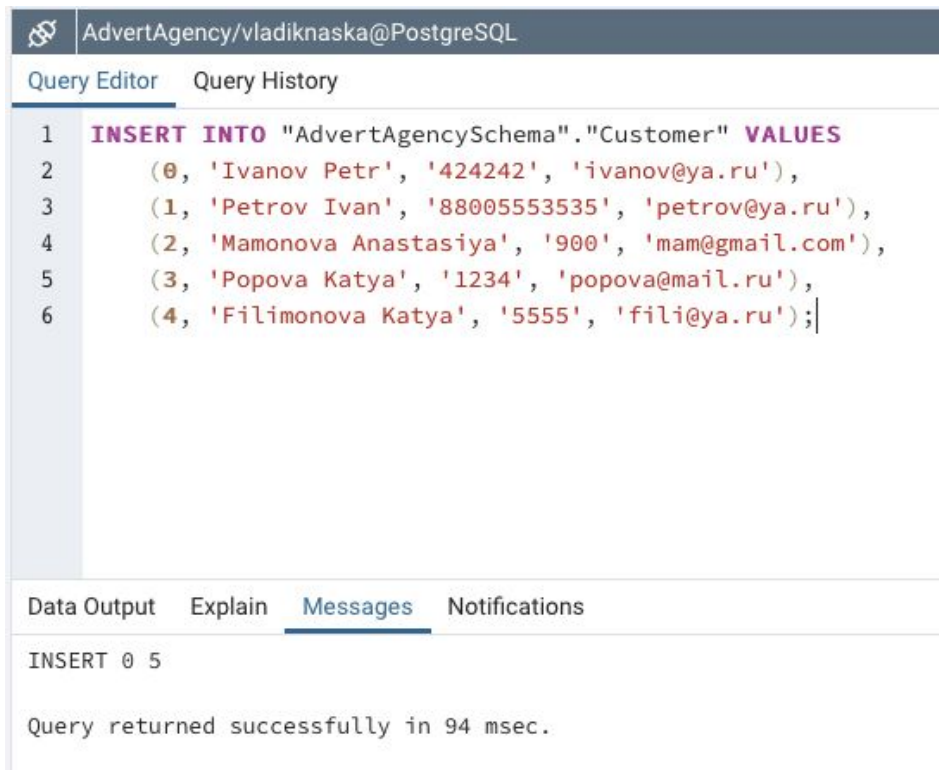


Рисунок 11. Определение внешних ключей для Orders

Заполнение таблиц рабочими данными

По очереди заполнены таблицы Customer (рисунок 12), PriceList (рисунок 13), Performer (рисунок 14), Material (рисунок 15), MaterialPerOrder, OrderContent, Order.



AdvertAgency/vladiknaska@PostgreSQL

Query Editor Query History

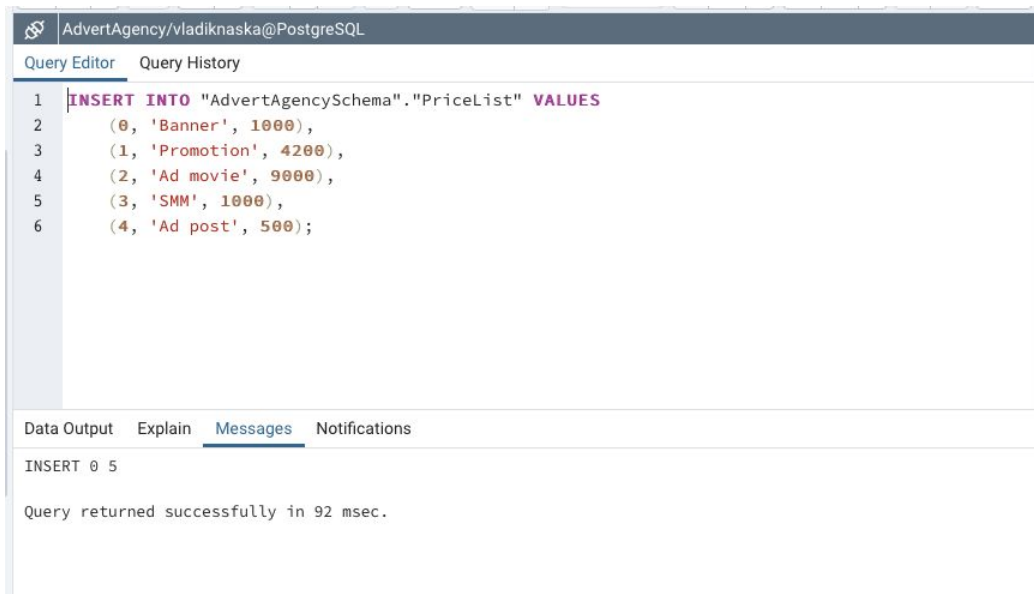
```
1 INSERT INTO "AdvertAgencySchema"."Customer" VALUES
2     (0, 'Ivanov Petr', '424242', 'ivanov@ya.ru'),
3     (1, 'Petrov Ivan', '88005553535', 'petrov@ya.ru'),
4     (2, 'Mamonova Anastasiya', '900', 'mam@gmail.com'),
5     (3, 'Popova Katya', '1234', 'popova@mail.ru'),
6     (4, 'Filimonova Katya', '5555', 'fili@ya.ru');
```

Data Output Explain Messages Notifications

INSERT 0 5

Query returned successfully in 94 msec.

Рисунок 12. Заполнение таблицы Customer



AdvertAgency/vladiknaska@PostgreSQL

Query Editor Query History

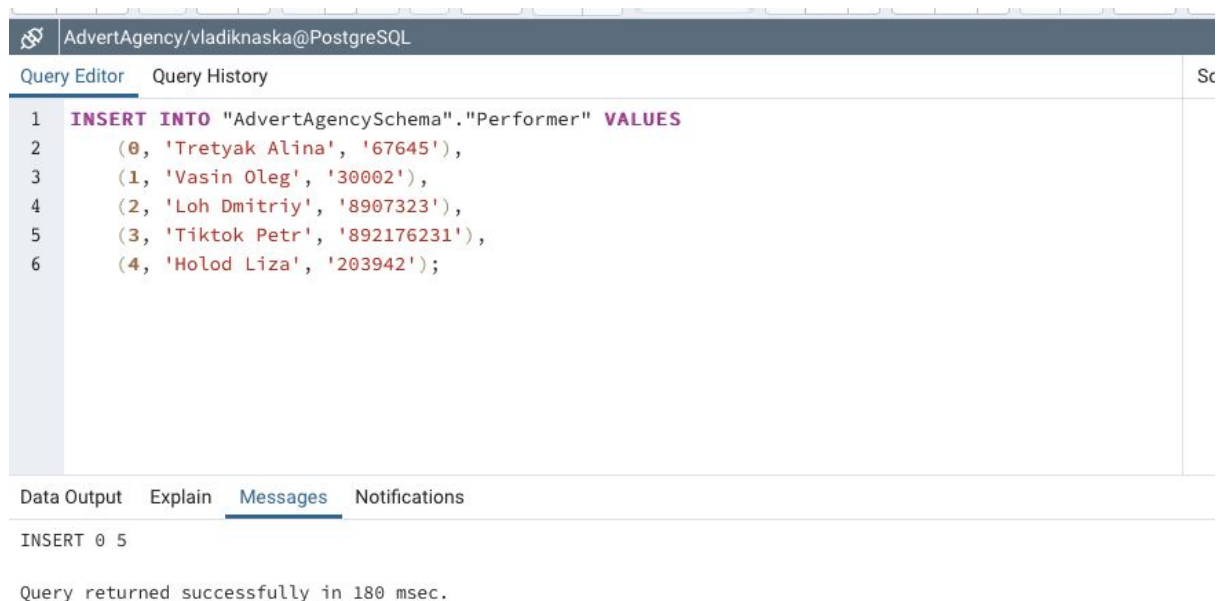
```
1 INSERT INTO "AdvertAgencySchema"."PriceList" VALUES
2     (0, 'Banner', 1000),
3     (1, 'Promotion', 4200),
4     (2, 'Ad movie', 9000),
5     (3, 'SMM', 1000),
6     (4, 'Ad post', 500);
```

Data Output Explain Messages Notifications

INSERT 0 5

Query returned successfully in 92 msec.

Рисунок 13. Заполнение таблицы PriceList



AdvertAgency/vladiknaska@PostgreSQL

Query Editor Query History

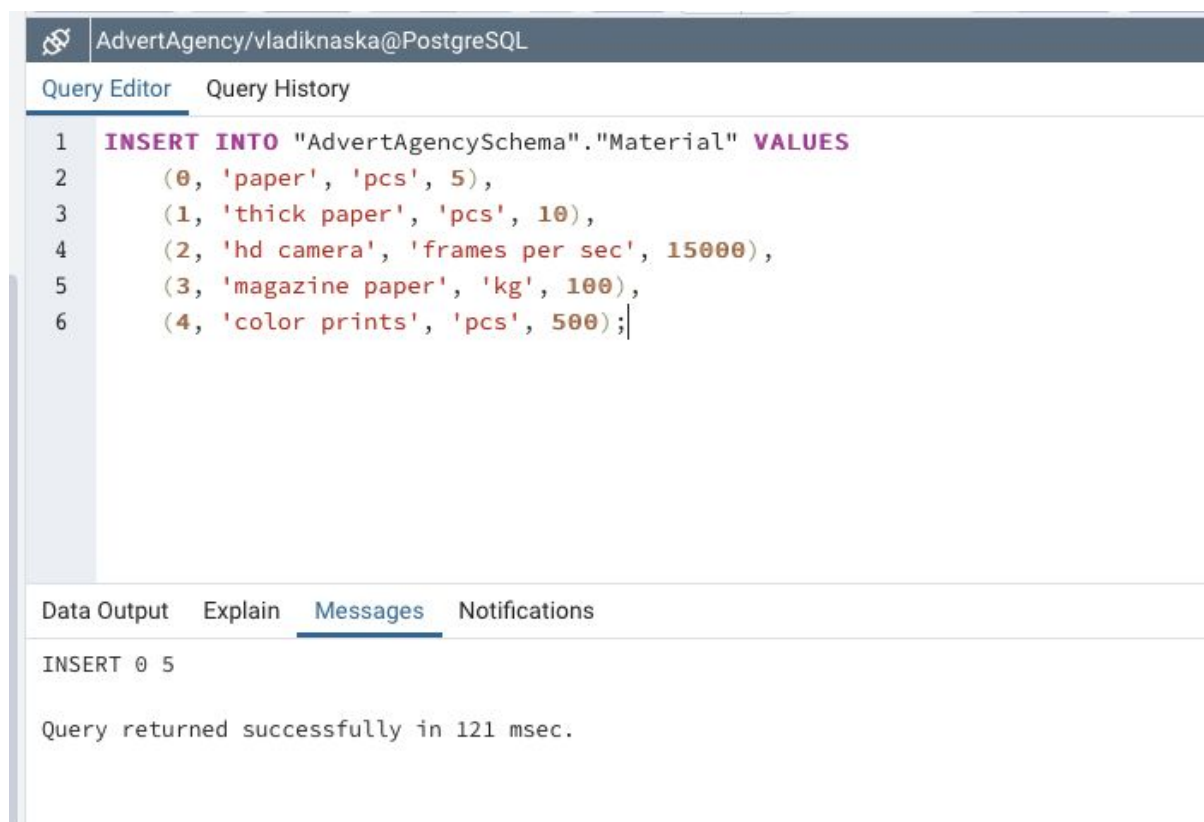
```
1 INSERT INTO "AdvertAgencySchema"."Performer" VALUES
2     (0, 'Tretyak Alina', '67645'),
3     (1, 'Vasin Oleg', '30002'),
4     (2, 'Loh Dmitriy', '8907323'),
5     (3, 'Tiktok Petr', '892176231'),
6     (4, 'Holod Liza', '203942');
```

Data Output Explain Messages Notifications

INSERT 0 5

Query returned successfully in 180 msec.

Рисунок 14. Заполнение таблицы Performer



AdvertAgency/vladiknaska@PostgreSQL

Query Editor Query History

```
1 INSERT INTO "AdvertAgencySchema"."Material" VALUES
2     (0, 'paper', 'pcs', 5),
3     (1, 'thick paper', 'pcs', 10),
4     (2, 'hd camera', 'frames per sec', 15000),
5     (3, 'magazine paper', 'kg', 100),
6     (4, 'color prints', 'pcs', 500);|
```

Data Output Explain Messages Notifications

INSERT 0 5

Query returned successfully in 121 msec.

Рисунок 15. Заполнение таблицы Material

С помощью команд `select * from table_name` проверено все ли было добавлено:

```
select * from "AdvertAgencySchema"."Customer";
```

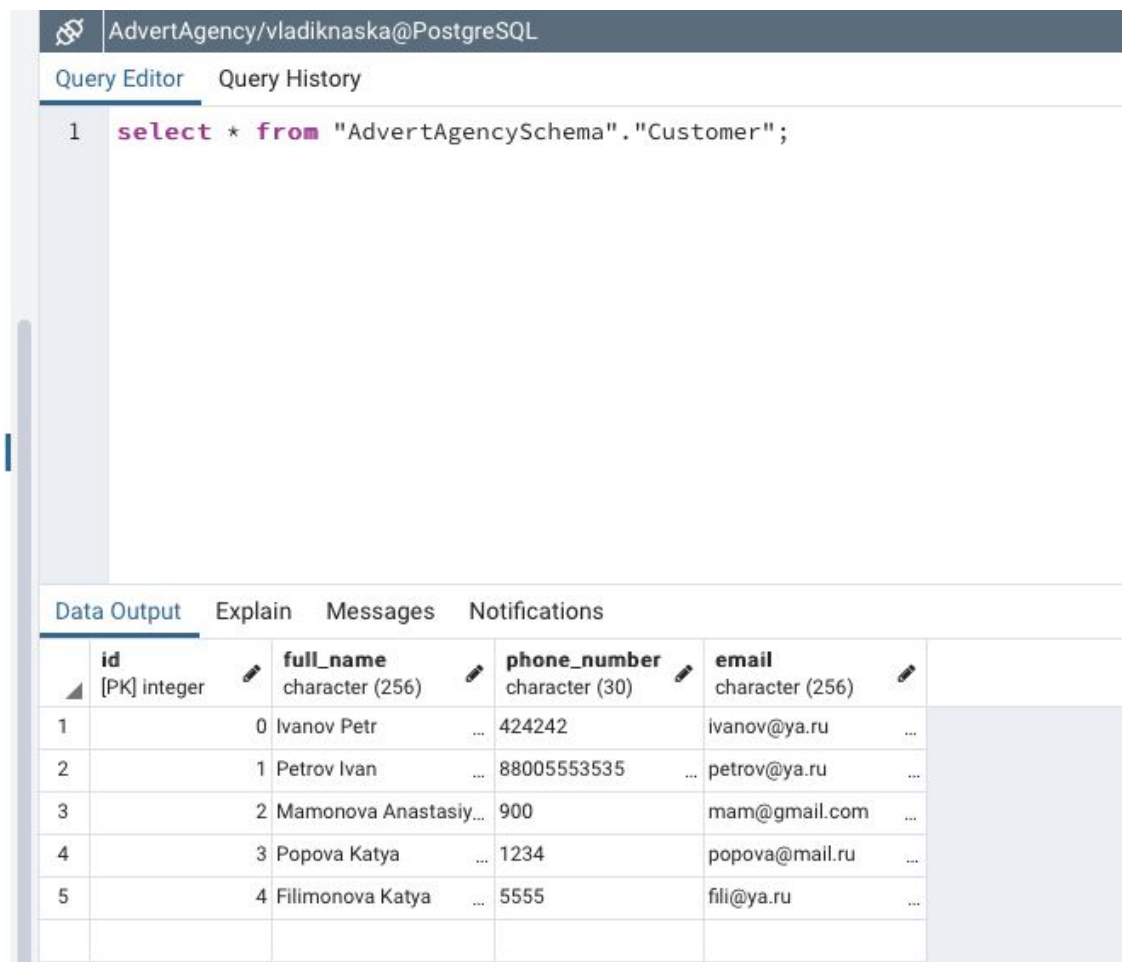
```
select * from "AdvertAgencySchema"."Material";
```

```

select * from "AdvertAgencySchema"."MaterialPerOrder";
select * from "AdvertAgencySchema"."Order";
select * from "AdvertAgencySchema"."OrderContent";
select * from "AdvertAgencySchema"."Performer";
select * from "AdvertAgencySchema"."PriceList";

```

Результаты представлены на рисунках 15-20.



The screenshot shows a PostgreSQL Query Editor interface. At the top, the connection is 'AdvertAgency/vladiknaska@PostgreSQL'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, showing a single SQL query: `1 select * from "AdvertAgencySchema"."Customer";`. Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying the results of the query in a table format. The table has five columns: 'id' (integer, primary key), 'full_name' (character (256)), 'phone_number' (character (30)), and 'email' (character (256)). There are five rows of data, each with an index from 1 to 5 in the first column.

	id [PK] integer	full_name character (256)	phone_number character (30)	email character (256)
1	0	Ivanov Petr	424242	ivanov@ya.ru
2	1	Petrov Ivan	88005553535	petrov@ya.ru
3	2	Mamonova Anastasiy...	900	mam@gmail.com
4	3	Popova Katya	1234	popova@mail.ru
5	4	Filimonova Katya	5555	fili@ya.ru

Рисунок 15. Таблица Customer

Query Editor

Query History

1

select * from "AdvertAgencySchema"."Material";

2

Data Output

Explain

Messages

Notifications

	<div>id</div> <div>[PK] integer</div>	<div>name</div> <div>character (256)</div>	<div>measure</div> <div>character (128)</div>	<div>price_per_item</div> <div>money</div>	
1	0	paper	pcs	\$5.00	
2	1	thick paper	pcs	\$10.00	
3	2	hd camera	frames per sec	\$15,000.00	
4	3	magazine paper	kg	\$100.00	
5	4	color prints	pcs	\$500.00	

Рисунок 16. Таблица Material

AdvertAgency/vladknaska@PostgreSQL

Query Editor

Query History

1

select * from "AdvertAgencySchema"."MaterialPerOrder";

2

Data Output

Explain

Messages

Notifications

	<div><div>id</div><div>[PK] integer</div></div>	<div><div>material_id</div><div>integer</div></div>	<div><div>amount</div><div>double precision</div></div>	
1		0		2
2		1	1	10
3		2	2	100
4		3	3	1
5		4	4	1000

Рисунок 17. Таблица MaterialPerOrder

AdvertAgency/vladiknaska@PostgreSQL

Query Editor Query History

```

1 select * from "AdvertAgencySchema"."Order";
2

```

Data Output Explain Messages Notifications

	id [PK] integer	order_content_id integer	customer_id integer	paid_at date	status character (128)
1	0	0	1	2018-01-01	approved
2	1	1	4	2020-05-01	closed
3	2	2	3	2019-02-01	denied
4	3	3	2	2017-07-01	approved
5	4	4	0	2016-10-01	approved

Рисунок 18. Таблица Order

AdvertAgency/vladiknaska@PostgreSQL

Query Editor Query History

```

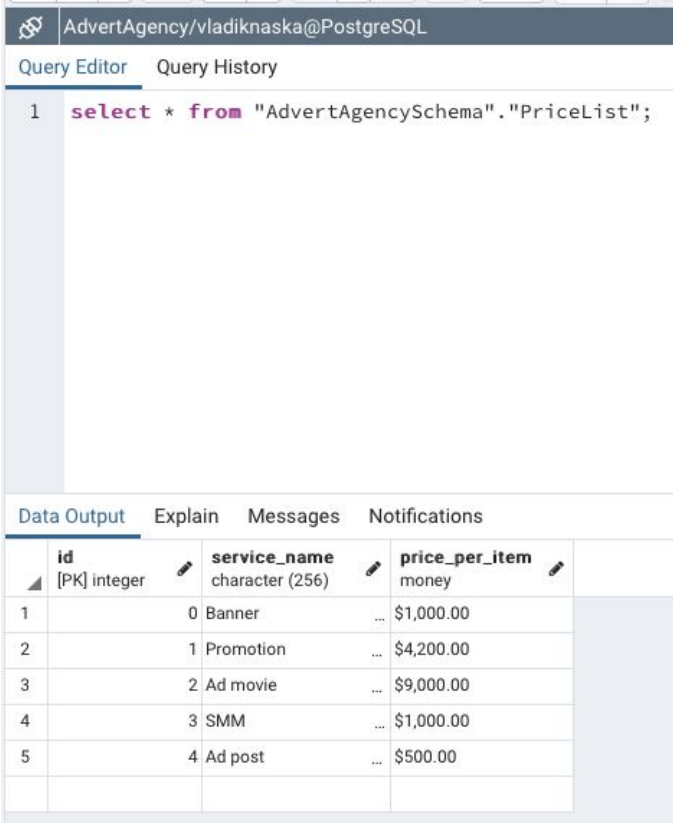
1 select * from "AdvertAgencySchema"."Performer";
2

```

Data Output Explain Messages Notifications

	id [PK] integer	full_name character (256)	phone_number character (30)
1	0	Tretyak Alina	67645
2	1	Vasin Oleg	30002
3	2	Loh Dmitriy	8907323
4	3	Tiktok Petr	892176231
5	4	Holod Liza	203942

Рисунок 19. Таблица Performer



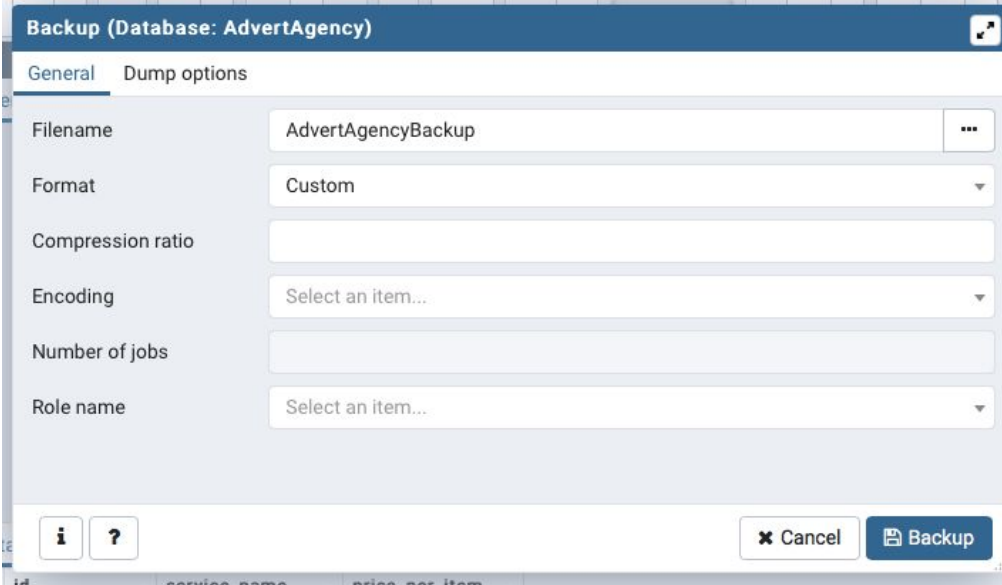
The screenshot shows a PostgreSQL Query Editor window with the connection 'AdvertAgency/vladiknaska@PostgreSQL'. The 'Query Editor' tab is active, displaying the SQL query: `select * from "AdvertAgencySchema"."PriceList";`. Below the query editor, the 'Data Output' tab is selected, showing the results of the query in a table format. The table has three columns: 'id' (integer, primary key), 'service_name' (character (256)), and 'price_per_item' (money). The results are as follows:

id	service_name	price_per_item
0	Banner	\$1,000.00
1	Promotion	\$4,200.00
2	Ad movie	\$9,000.00
3	SMM	\$1,000.00
4	Ad post	\$500.00

Рисунок 20. Таблица PriceList

Создание резервной копии БД

Pgadmin 4 предоставляет удобный интерфейс для создания резервной копии БД. С помощью диалогового окна Create Backup создана резервная копия базы данных AdvertAgency, рисунок 21.



The screenshot shows the 'Backup (Database: AdvertAgency)' dialog box in PgAdmin 4. The 'General' tab is selected, and the 'Dump options' section is visible. The fields are configured as follows:

- Filename: AdvertAgencyBackup
- Format: Custom
- Compression ratio: (empty field)
- Encoding: Select an item...
- Number of jobs: (empty field)
- Role name: Select an item...

At the bottom of the dialog, there are buttons for 'Cancel' and 'Backup'.

Рисунок 21. Создание резервной копии

Восстановление базы данных

Диалог *Restore* обеспечивает простой способ восстановления или обновления базы данных. Диалог *Backup* вызывает опцию утилиты `pg_dump` клиента; диалог *Restore* вызывает опцию утилиты `pg_restore` клиента. На рисунке 22 показано восстановление базы данных AdvertAgency.

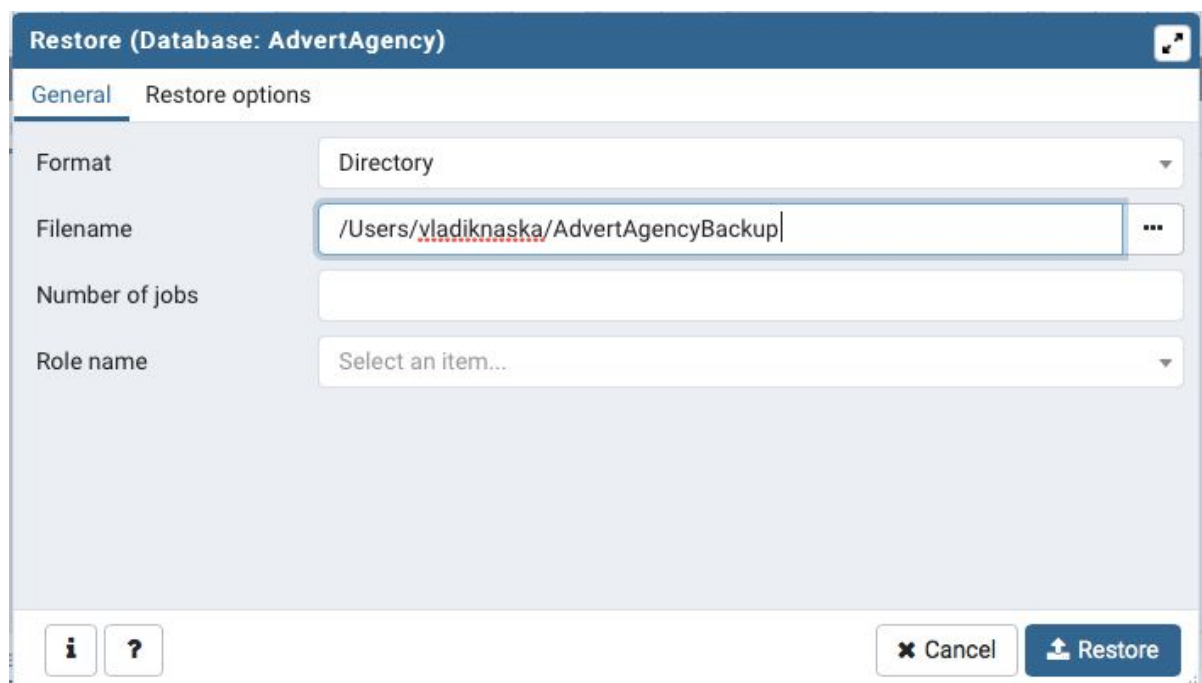


Рисунок 22. Восстановление БД

Вывод: в ходе выполнения данной лабораторной работы получены практические навыки установки СУБД PostgreSQL 11 и создания базы данных в pgadmin 4, создания таблиц базы данных PostgreSQL 11, заполнения их рабочими данными, резервного копирования и восстановления БД.