

ПРАКТИЧЕСКАЯ РАБОТА 1. ЧАСТЬ 1

Практическая работа №1 выполняется индивидуально по методическим указаниям и включает в себя несколько заданий. Практическая работа №1 состоит из 2х частей. В данном документе представлено задание для выполнения первой части практической работы №1.

По практической работе №1 формируется итоговый отчет, содержащий результаты выполнения 2х частей работы. Итоговый отчет должен содержать:

- Титульный лист
- Цель работы.
- Задачи, решаемые при выполнении работы.
- Исходные данные.
- Выполнение работы: Краткое описание процесса выполнения всех задач по шагам (при наличии нескольких шагов) со скриншотами.
- Выводы и анализ результатов работы. Обобщение результатов выполнения всех задач работы: чего должны были достичь, чего фактически достигли и каким образом, с какими трудностями столкнулись, какие проблемы на каких этапах выполнения возникли и как именно были решены.

Задание 1. Установка необходимого ПО

1. Необходимо скачать ПО (СУБД PostgreSQL) по следующим ссылкам:

- Официальный источник ПО: <https://www.postgresql.org/download/>
- Прямая ссылка на дистрибутивы: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

2. Далее необходимо установить ПО, пройдя по шагам загрузчика (рисунок 1.1 и 1.2):

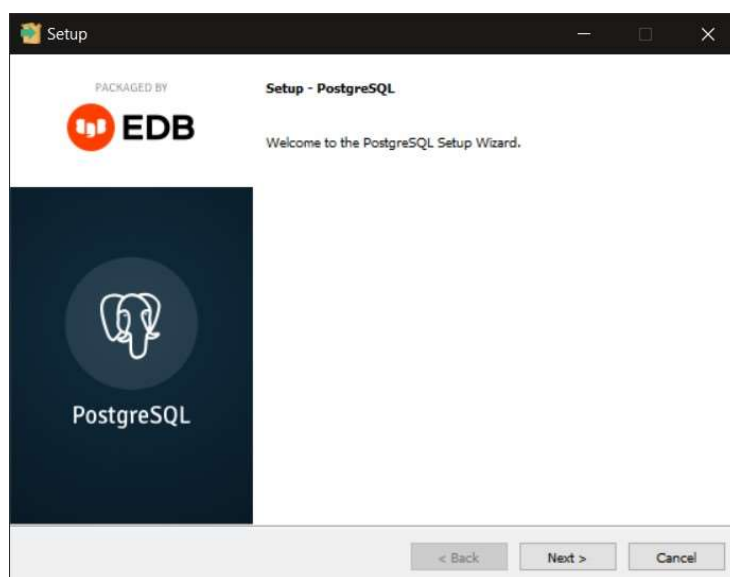


Рисунок 1.1 – Установка СУБД PostgreSQL

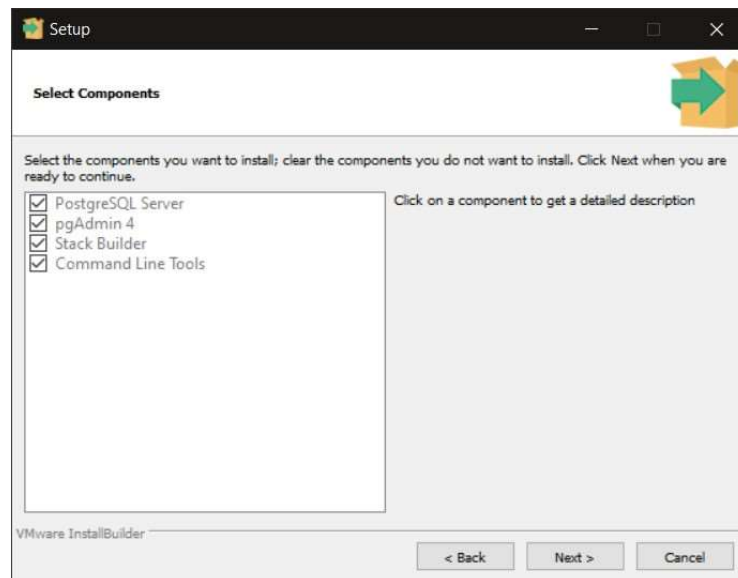


Рисунок 1.2 – Выбор компонентов для установки

Обратите внимание, что стоит галка напротив pgAdmin – это IDE (интегрированная среда разработки), клиентская часть для работы с СУБД PostgreSQL, которая рекомендуется к использованию. При желании можно использовать любую другую IDE, но настройку необходимо будет выполнить самостоятельно.

Все остальные шаги можно оставить без изменения. При желании можно сменить директорию, куда будет установлен экземпляр сервера (рисунок 1.3).

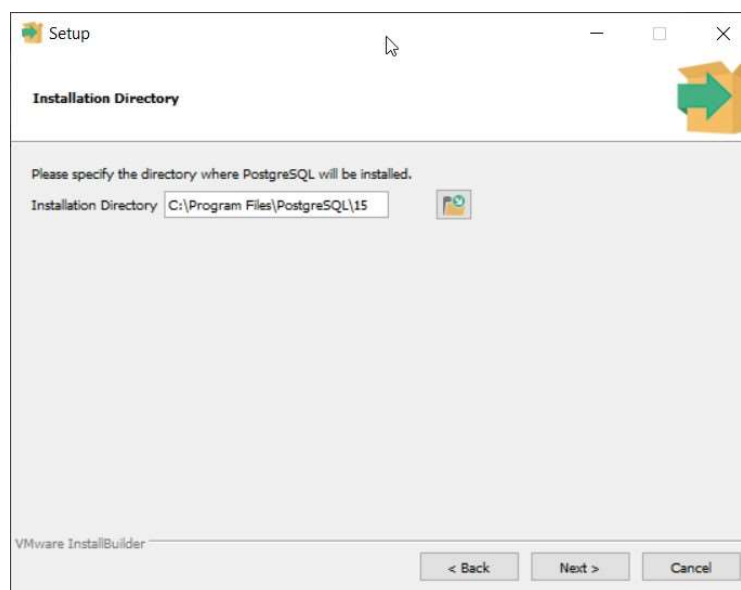


Рисунок 1.3 – Изменение директории для установки сервера

Обратите внимание, что пароль (рисунок 1.4), который вы будете придумывать для суперадмина необходимо запомнить – он вам понадобится для выполнения других лабораторных работ.

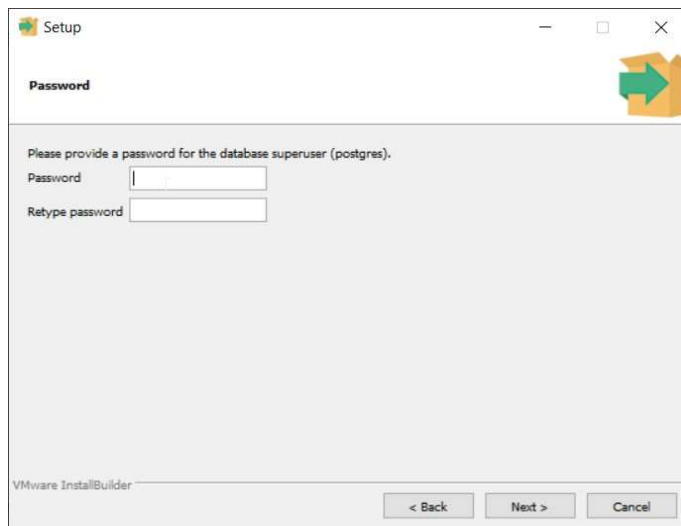


Рисунок 1.4 – Установка пароля для суперадмина.

Порт и все остальные вкладки оставляем без изменения (рисунки 1.5, 1.6, 1.7).

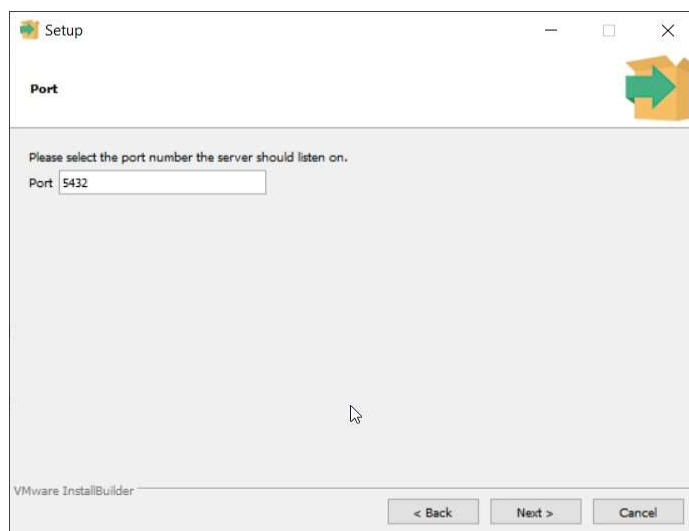


Рисунок 1.5 – Установка номера порта.

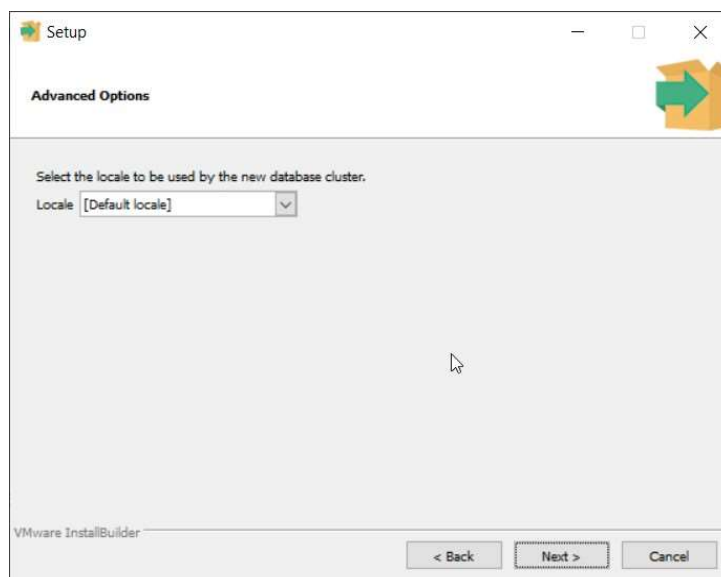


Рисунок 1.6 – Дополнительные опции.

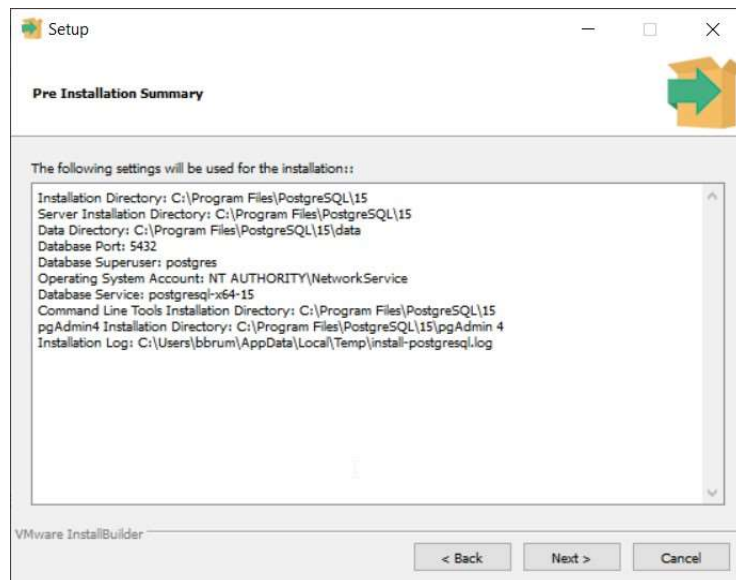


Рисунок 1.7 – Предварительная информация об устанавливаемых компонентах.

После завершения инсталляции снимите галочку «Stack builder...» в заключительном окне установщика (рисунок 1.8) и нажмите Finish. Данная утилита используется для загрузки и установки дополнительных инструментов, драйверов и приложений, и нам не нужна для завершения инсталляции PostgreSQL.

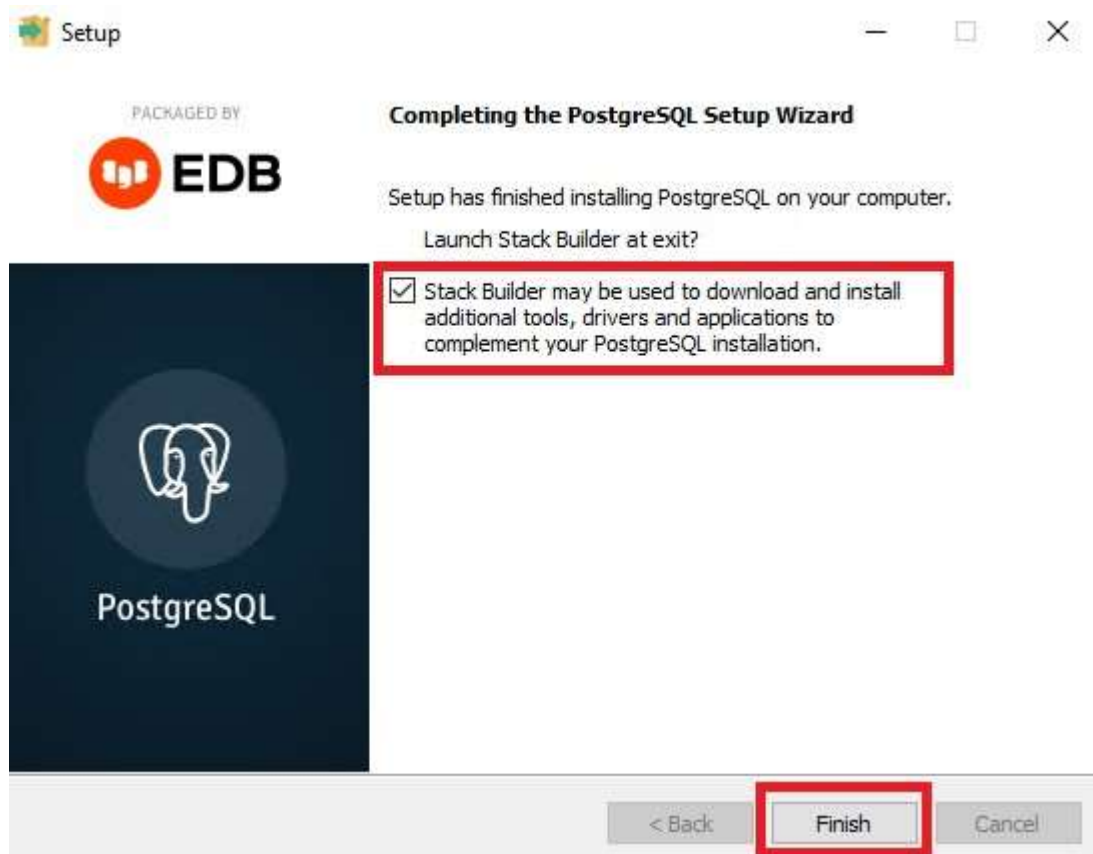


Рисунок 1.8 – Заключительное окно установщика.

Если галку вы забыли снять и запустился установщик Stack Builder, то нажмите на кнопку «Отмена».

3. Запустите IDE pgAdmin, которая установилась вместе с серверной частью (рисунок 1.9).

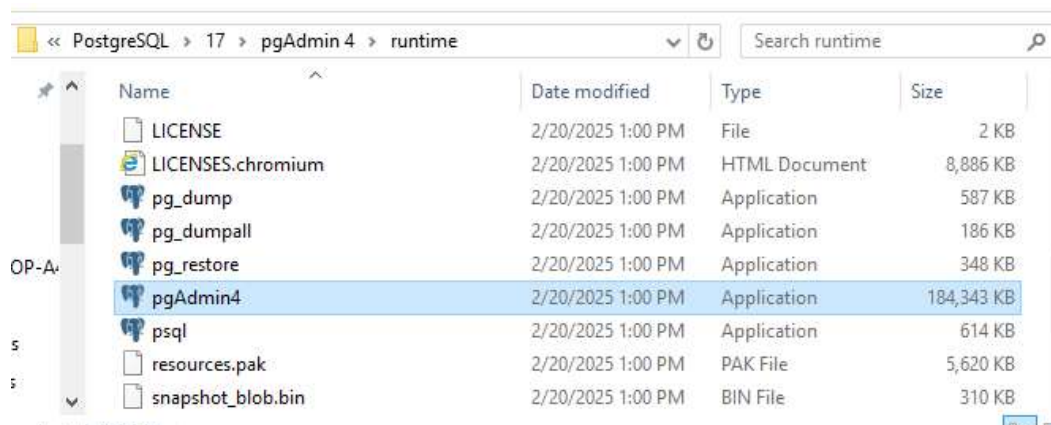


Рисунок 1.9 – Выбор инструмента IDE pgAdmin в диспетчере файлов.

При первом входе необходимо будет выбрать сервер СУБД (рисунок 1.10)

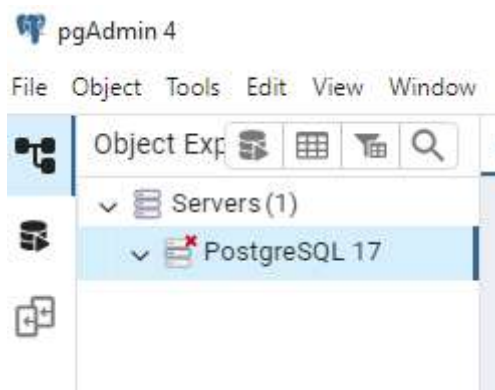


Рисунок 1.10 – Выбор сервера СУБД в IDE pgAdmin.

И ввести пароль (рисунок 1.11), указанный вами при установке. Рекомендуется поставить галочку для его сохранения системой.

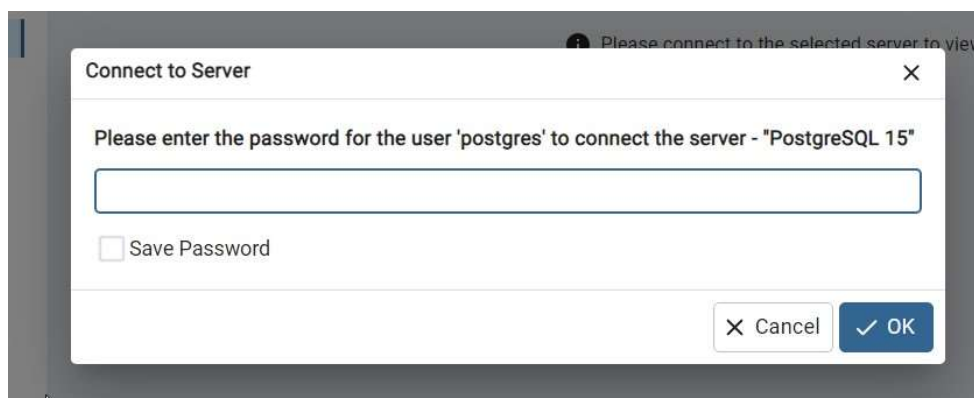


Рисунок 1.11 – Ввод пароля суперадмина.

4. Программное обеспечение готово к работе (рисунок 1.12)

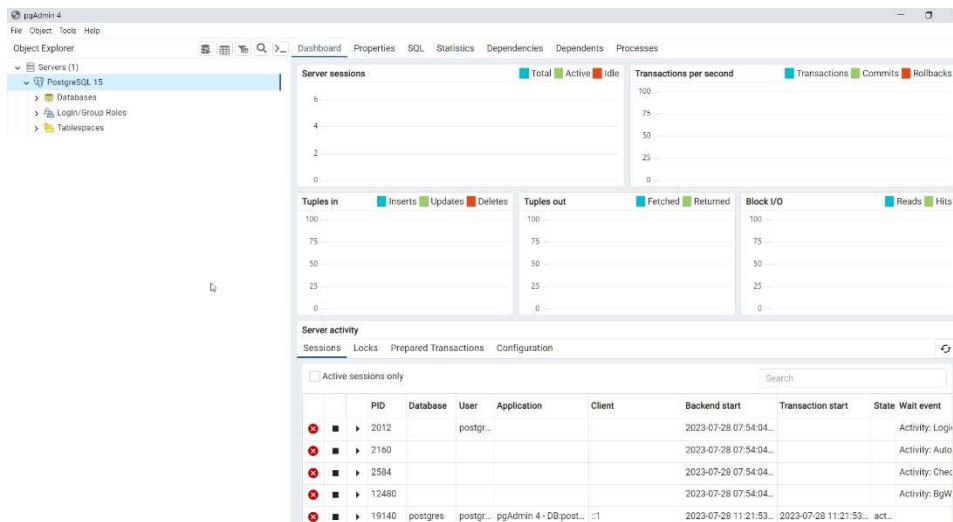


Рисунок 1.12 – Сервер подключен.

5. Для удобства можно переключить интерфейс pgAdmin на русский язык. Для этого выберите меню File->Preferences, в появившемся окне в меню слева выберите Miscellaneous -> User Language, а затем в поле User Language справа «Russian».

6. Проверьте работоспособность. В обозревателе объектов (слева) раскройте узел «Базы данных», нажмите правой кнопкой мыши (далее по тексту ПКМ) на базе «postgres» - Запросник.

В открывшемся окне напишите запрос проверки версии сервера:

`SELECT version();`

Нажмите кнопку «play» или F5, посмотрите результат (рисунок 1.13).

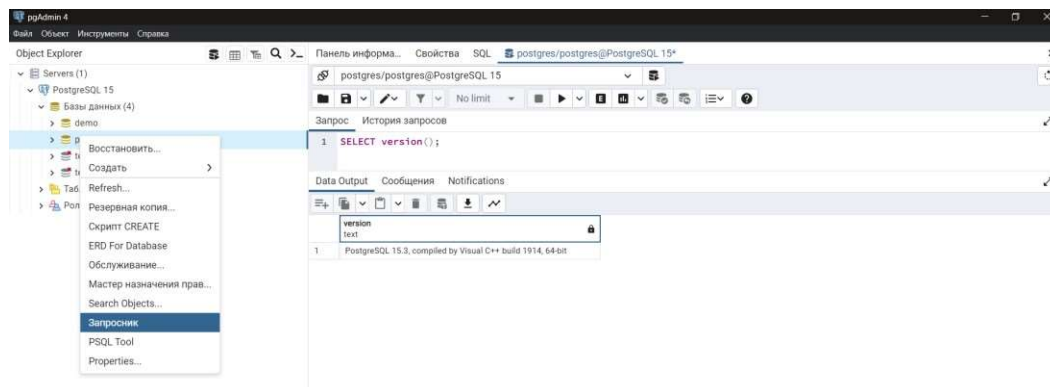


Рисунок 1.13 – Демонстрация работы «Запросника».

Задание 2. Подключение учебной базы данных

1. «demo-medium.zip» — это учебная БД авиаперелётов (рисунок 2.1).

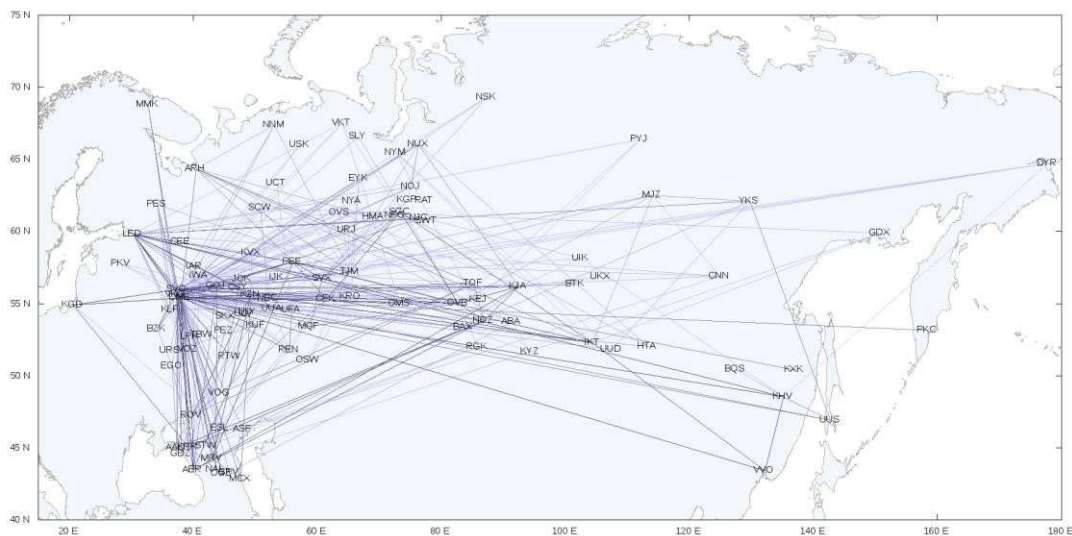


Рисунок 2.1 – Учебная БД авиаперелётов.

Скачайте версию учебной БД по следующему ресурсу: <https://postgrespro.ru/education/demodb>

Есть варианты этой учебной БД разных размеров:

- demo-small.zip (21 МБ) — данные по полетам за один месяц (размер БД примерно 300 МБ),
- demo-medium.zip (62 МБ) — данные по полетам за три месяца (размер БД примерно 700 МБ),
- demo-big.zip (232 МБ) — данные по полетам за год (размер БД примерно 2,5 ГБ).

Обратите внимание, для лабораторных работ мы будем использовать demo-medium версии
15.08.2017.

2. Подключение учебной базы данных выполняется из командной строки (рисунок 2.2). Для этого запустите командную строку любым из следующих вариантов:

- Сочетание клавиш WIN+R, затем нужно вписать cmd и нажать Enter;
- Значок поиска около кнопки «Пуск» – cmd – запуск;

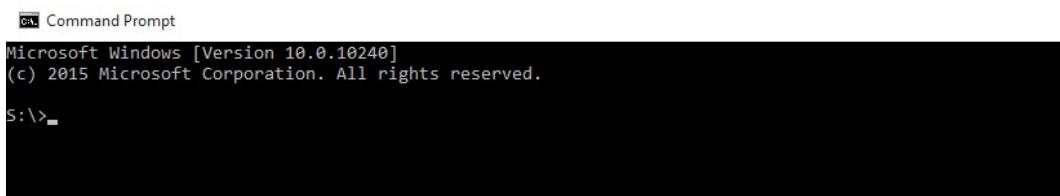


Рисунок 2.2 – Подключение учебной базы из командной строки.

3. Перенесите архив с учебной БД в удобную для вас папку (в текущем примере это c:\Postgres\l) и разархивируйте его. Вы увидите sql файл скрипта для развертывания БД. Не стоит его запускать напрямую в pgAdmin – IDE не справится с размером.

4. В командной строке смените директорию на ту, в которой лежит sql файл БД (рисунок 2.3):

- Если требуется сменить диск – Для смены на диск C - команда «с:»
- Если требуется сменить каталог на диске – через команду cd

Например, cd Postgres\demo-medium

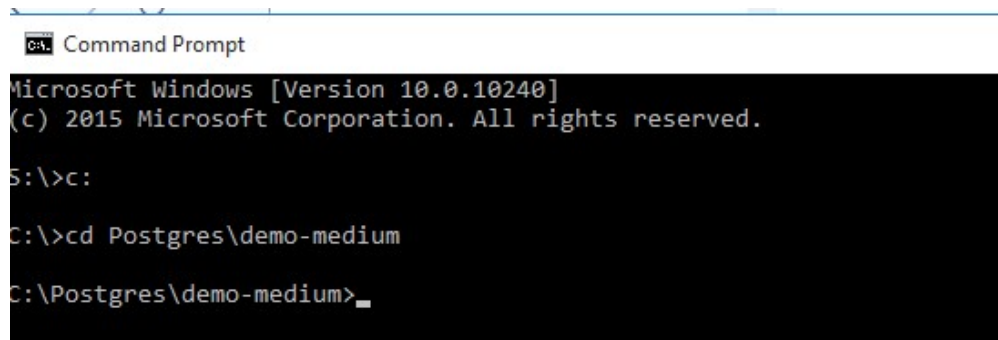


Рисунок 2.3 – Смена директории для запуска ddl-скрипта БД.

5. Скопируйте путь установки серверного программного обеспечения PostgreSQL (рисунок 2.4). В нашем случае это

- «C:\Program Files\PostgreSQL\17\bin\»

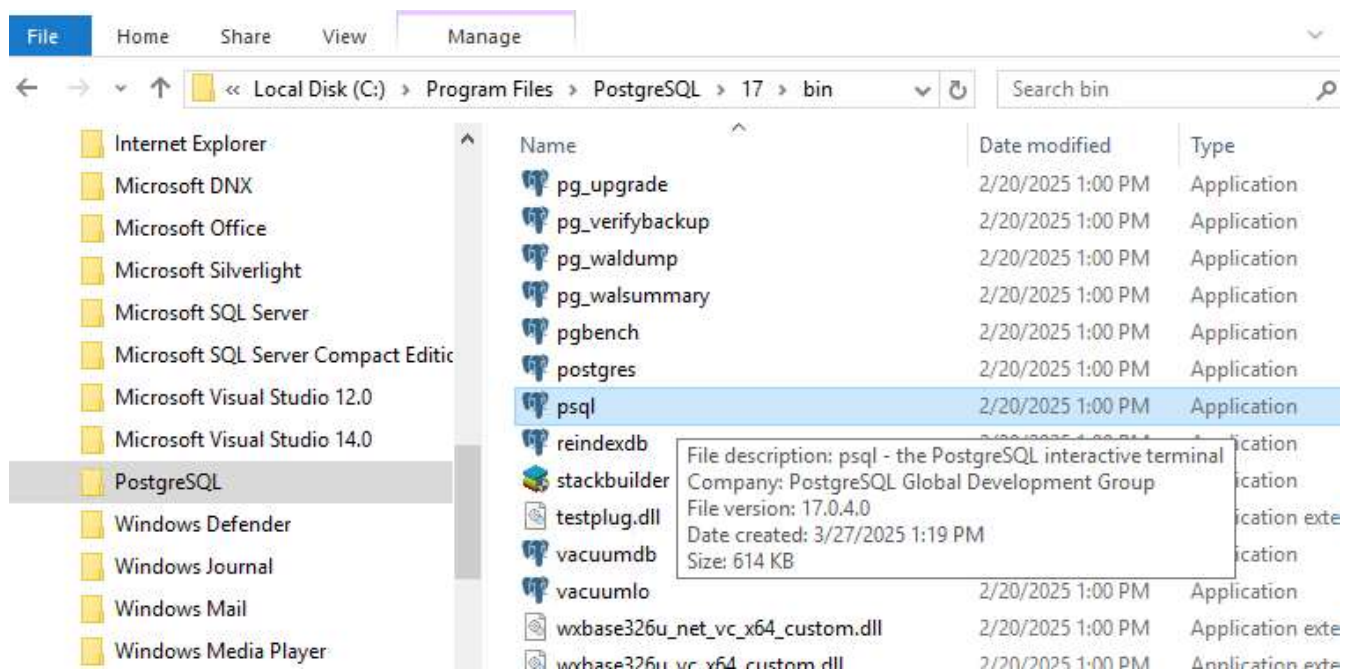


Рисунок 2.4 – Определение пути серверного программного обеспечения PostgreSQL в диспетчере файлов.

6. Запустите в командной строке следующую команду:

- "C:\Program Files\PostgreSQL\17\bin\psql" -U postgres -f demo-medium-20170815.sql

, где

- C:\Program Files\PostgreSQL\17\bin\psql – указание на файл PostgreSQL;
- -U postgres – указание на пользователя (без изменения);
- -f demo-medium-20170815.sql – указание на название скрипта БД.

ВВ! Первый путь должен быть **в кавычках**

Также вам потребуется пароль, который вы придумали для суперпользователя PostgreSQL (рисунок 2.5).

```
Command Prompt - "C:\Program Files\PostgreSQL\17\bin\psql" -U postgres -f demo-medium-20170815.sql
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

S:\>c:

C:\>cd Postgres\demo-medium

C:\Postgres\demo-medium>"C:\Program Files\PostgreSQL\17\bin\psql" -U postgres -f demo-medium-20170815.sql
Password for user postgres: _
```

Рисунок 2.5 – Ввод пароля суперпользователя.

7. Ожидайте порядка 10 минут.

8. Далее можно проверить, что все развернулось успешно. Для этого обновите в pgAdmin узел «Базы данных» - ПКМ на узле – обновить. Должна появиться новая БД «Демо».

9. Откройте запросник (ПКМ на базу - Запросник) на базе данных Демо.

10. Проверьте работоспособность, выполнив следующий скрипт (рисунок 2.6):

1 SELECT * FROM bookings.aircrafts_data;
2

Data OutputСообщенияNotifications

	aircraft_code [PK] character	model jsonb	range integer
1	773	{"en": "Boeing 777-300", "ru": "Боинг 777-300"}	11100
2	763	{"en": "Boeing 767-300", "ru": "Боинг 767-300"}	7900
3	SU9	{"en": "Sukhoi Superjet-100", "ru": "Сухой Суперджет-10..."}	3000
4	320	{"en": "Airbus A320-200", "ru": "Аэробус A320-200"}	5700
5	321	{"en": "Airbus A321-200", "ru": "Аэробус A321-200"}	5600
6	319	{"en": "Airbus A319-100", "ru": "Аэробус A319-100"}	6700
7	733	{"en": "Boeing 737-300", "ru": "Боинг 737-300"}	4200
8	CN1	{"en": "Cessna 208 Caravan", "ru": "Сессна 208 Караван"}	1200
9	CR2	{"en": "Bombardier CRJ-200", "ru": "Бомбардье CRJ-200"}	2700

Рисунок 2.6 – Проверка работоспособности БД в «Запроснике».

11. Учебная база данных готова к работе.

NB! Для углубления знаний, полученных в нашем курсе, можно воспользоваться курсами, разработанными компанией Postgres Professional <https://postgrespro.ru/education/courses>.

Задание 3. Ознакомление с учебной БД

Ознакомьтесь с описанием учебной БД. Если у вас отличная версия от той, которая приложена к курсу, это можно сделать в разделе «Описание» по ссылке:

<https://postgrespro.ru/education/demodb> **Схема БД** (рисунок 3.1):

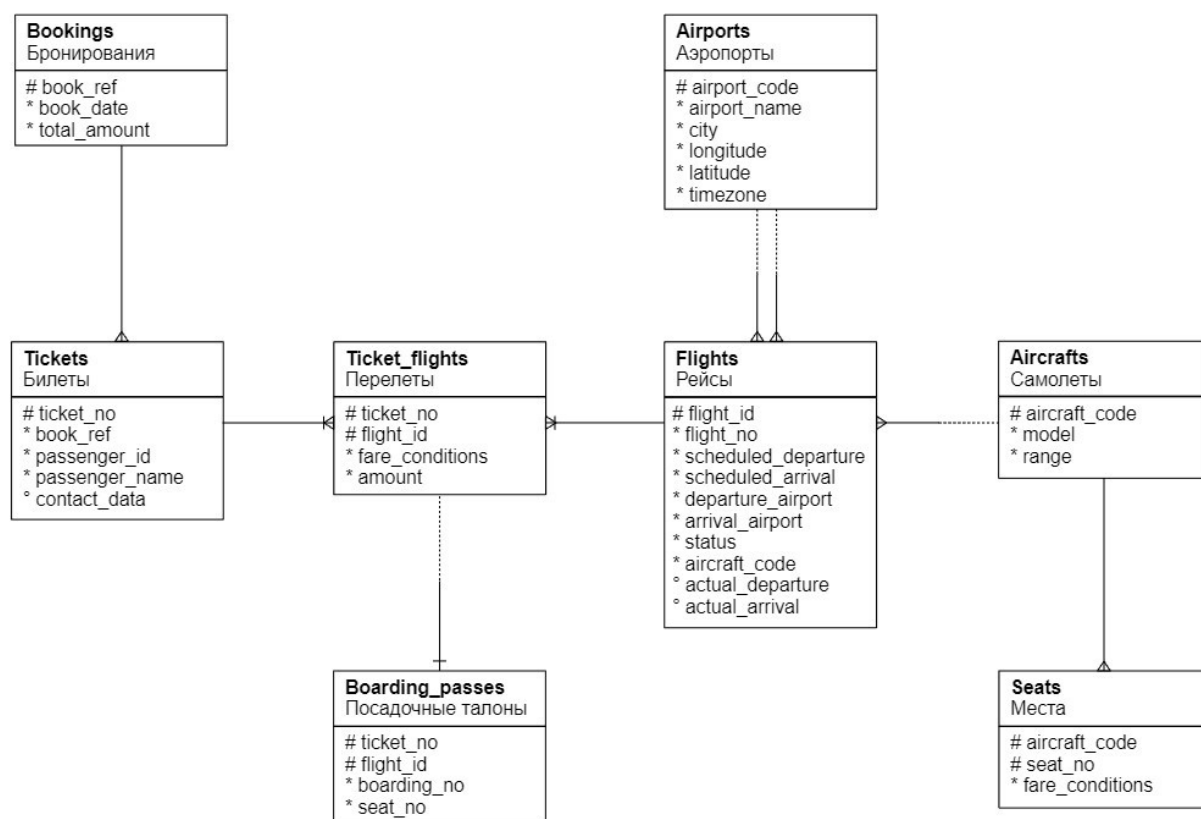


Рисунок 3.1 – Схема БД.

Описание схемы представлено в таблицах 1-9:

Основной сущностью является бронирование (bookings).

В одно бронирование можно включить несколько пассажиров, каждому из которых выписывается отдельный билет (tickets). Билет имеет уникальный номер и содержит информацию о пассажире. Как таковой пассажир не является отдельной сущностью. Как имя, так и номер документа пассажира могут меняться с течением времени, так что невозможно однозначно найти все билеты одного человека; для простоты можно считать, что все пассажиры уникальны.

Билет включает один или несколько перелетов (ticket_flights). Несколько перелетов могут включаться в билет в случаях, когда нет прямого рейса, соединяющего пункты отправления и назначения (полет с пересадками), либо когда билет взят «туда и обратно». В схеме данных нет жёсткого ограничения, но предполагается, что все билеты в одном бронировании имеют одинаковый набор перелетов.

Каждый рейс (flights) следует из одного аэропорта (airports) в другой. Рейсы с одним номером имеют одинаковые пункты вылета и назначения, но будут отличаться датой отправления.

При регистрации на рейс пассажиру выдаётся посадочный талон (boarding_passes), в котором указано место в самолете. Пассажир может зарегистрироваться только на тот рейс, который есть

у него в билете. Комбинация рейса и места в самолете должна быть уникальной, чтобы не допустить выдачу двух посадочных талонов на одно место.

Количество мест (seats) в самолете и их распределение по классам обслуживания зависит от модели самолета (aircrafts), выполняющего рейс. Предполагается, что каждая модель самолета имеет только одну компоновку салона. Схема данных не контролирует, что места в посадочных талонах соответствуют имеющимся в самолете (такая проверка может быть сделана с использованием табличных триггеров или в приложении).

Таблица 1. Описание таблиц

Имя	Тип	Small	Medium	Big	Описание
aircrafts	таблица	16 kB	16 kB	16 kB	Самолеты
airports	таблица	48 kB	48 kB	48 kB	Аэропорты
boarding_passes	таблица	31 MB	102 MB	427 MB	Посадочные талоны
bookings	таблица	13 MB	30 MB	105 MB	Бронирования
flights	таблица	3 MB	6 MB	19 MB	Рейсы
flights_v	представление	0 kb	0 kb	0 kb	Рейсы
routes	мат. предст.	136 kB	136 kB	136 kB	Маршруты
seats	таблица	88 kB	88 kB	88 kB	Места
ticket_flights	таблица	64 MB	145 MB	516 MB	Перелеты
tickets	таблица	47 MB	107 MB	381 MB	Билеты

Таблица 2. Bookings.aircrafts

Столбец	Тип	Модификаторы	Описание
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
model	text	NOT NULL	Модель самолета
range	integer	NOT NULL	Максимальная дальность полета, км

Индексы:

PRIMARY KEY, btree (aircraft_code)

Ограничения-проверки:

CHECK (range > 0)

Таблица 3. Bookings.airports

Столбец	Тип	Модификаторы	Описание
airport_code	char(3)	NOT NULL	Код аэропорта
airport_name	text	NOT NULL	Название аэропорта
city	text	NOT NULL	Город
longitude	float	NOT NULL	Координаты аэропорта: долгота
latitude	float	NOT NULL	Координаты аэропорта: широта
timezone	text	NOT NULL	Часовой пояс аэропорта

Индексы:

PRIMARY KEY, btree (airport_code)

Таблица 4. Bookings.boarding_passes

Столбец	Тип	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
flight_id	integer	NOT NULL	Идентификатор рейса
boarding_no	integer	NOT NULL	Номер посадочного талона
seat_no	varchar(4)	NOT NULL	Номер места

Индексы:

PRIMARY KEY, btree (ticket_no, flight_id)

UNIQUE CONSTRAINT, btree (flight_id, boarding_no)

UNIQUE CONSTRAINT, btree (flight_id, seat_no)

Ограничения внешнего ключа:

FOREIGN KEY (ticket_no, flight_id)

REFERENCES ticket_flights(ticket_no, flight_id)

Таблица 5. Bookings.bookings

Столбец	Тип	Модификаторы	Описание
book_ref	char(6)	NOT NULL	Номер бронирования
book_date	timestampz	NOT NULL	Дата бронирования
total_amount	numeric(10,2)	NOT NULL	Полная сумма бронирования

Индексы:

PRIMARY KEY, btree (book_ref)

Таблица 6. Bookings.flights

Столбец	Тип	Модификаторы	Описание
flight_id	serial	NOT NULL	Идентификатор рейса
flight_no	char(6)	NOT NULL	Номер рейса
scheduled_departure	timestampz	NOT NULL	Время вылета по расписанию
scheduled_arrival	timestampz	NOT NULL	Время прилёта по расписанию
departure_airport	char(3)	NOT NULL	Аэропорт отправления
arrival_airport	char(3)	NOT NULL	Аэропорт прибытия
status	varchar(20)	NOT NULL	Статус рейса
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
actual_departure	timestampz		Фактическое время вылета
actual_arrival	timestampz		Фактическое время прилёта

Индексы:

```
PRIMARY KEY, btree (flight_id)
UNIQUE CONSTRAINT, btree (flight_no, scheduled_departure)
```

Ограничения-проверки:

```
CHECK (scheduled_arrival > scheduled_departure)
CHECK ((actual_arrival IS NULL)
OR ((actual_departure IS NOT NULL AND actual_arrival IS NOT NULL)
AND (actual_arrival > actual_departure)))
CHECK (status IN ('On Time', 'Delayed', 'Departed',
'Arrived', 'Scheduled', 'Cancelled'))
```

Ограничения внешнего ключа:

```
FOREIGN KEY (aircraft_code)
REFERENCES aircrafts(aircraft_code)
FOREIGN KEY (arrival_airport)
REFERENCES airports(airport_code)
FOREIGN KEY (departure_airport)
REFERENCES airports(airport_code)
```

Таблица 7. Bookings.seats

Столбец	Тип	Модификаторы	Описание
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
seat_no	varchar(4)	NOT NULL	Номер места
fare_conditions	varchar(10)	NOT NULL	Класс обслуживания

Индексы:

```
PRIMARY KEY, btree (aircraft_code, seat_no)
```

Ограничения-проверки:

```
CHECK (fare_conditions IN ('Economy', 'Comfort', 'Business'))
```

Ограничения внешнего ключа:

```
FOREIGN KEY (aircraft_code)
REFERENCES aircrafts(aircraft_code) ON DELETE CASCADE
```

Таблица 8. Bookings.ticket_flights

Столбец	Тип	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
flight_id	integer	NOT NULL	Идентификатор рейса
fare_conditions	varchar(10)	NOT NULL	Класс обслуживания
amount	numeric(10,2)	NOT NULL	Стоимость перелета

Индексы:
PRIMARY KEY, btree (ticket_no, flight_id)

Ограничения-проверки:
CHECK (amount >= 0)
CHECK (fare_conditions IN ('Economy', 'Comfort', 'Business'))

Ограничения внешнего ключа:
FOREIGN KEY (flight_id) REFERENCES flights(flight_id)
FOREIGN KEY (ticket_no) REFERENCES tickets(ticket_no)

Таблица 9. Bookings.tickets

Столбец	Тип	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
book_ref	char(6)	NOT NULL	Номер бронирования
passenger_id	varchar(20)	NOT NULL	Идентификатор пассажира
passenger_name	text	NOT NULL	Имя пассажира
contact_data	jsonb		Контактные данные пассажира

Индексы:
PRIMARY KEY, btree (ticket_no)

Ограничения внешнего ключа:
FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)

Задание 4. Создание учебной базы данных HR средствами pgAdmin

1. Нажмите ПКМ на узле «Базы данных» и выберите меню «Создать» - «База данных» (рисунок 4.1):

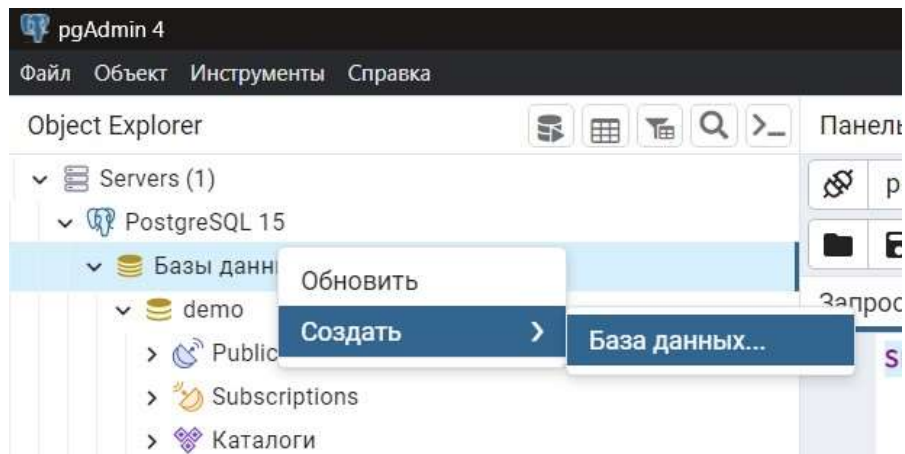


Рисунок 4.1 – Обзор объектов. Создание БД.

2. В открывшемся окне на вкладке General заполните название БД как HR
3. На вкладке Определение укажите параметр Максимальное число подключений как 1
4. Посмотрите и изучите вкладку SQL
5. Нажмите на кнопку «Сохранить»
6. Убедитесь через Обзор объектов в том, что БД создана
7. Удалите БД путем нажатия на нее ПКМ.

NB! Все подключения к этой БД (на случай, если вы отдельно открыли) должны быть выключены.

Задание 5. Создание учебной базы данных HR средствами PL/pgSQL

1. Откройте запросник на любой БД (как пример, штатная postgres)
2. Напишите код создания БД (рисунок 5.1):

```
CREATE DATABASE "HR"
WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'Russian_Russia.1251'
  LC_CTYPE = 'Russian_Russia.1251'
  LOCALE_PROVIDER = 'libc'
  TABLESPACE = pg_default
  CONNECTION LIMIT = 1
  IS_TEMPLATE = False;
```

Рисунок 5.1 – Создание БД DDL оператором CREATE.

Параметры LC_ нужны для правильного сопоставления строковых данных.
Если у вас возникла ошибка (рисунок 5.2):

```
Сообщения
ERROR: new collation (Russian_Russia.1251) is incompatible with the
collation of the template database (en-US)
HINT: Use the same collation as in the template database, or use template0
as template.
```

Рисунок 5.2 – Сообщение об ошибке.

Добавьте в оператор создания БД «HR» (CREATE DATABASE) параметр

- TEMPLATE template0

3. Убедитесь через Обзорщик объектов в том, что БД создана

Задание 6. Создание схем

1. Разверните вкладку для создания схем (рисунок 6.1):

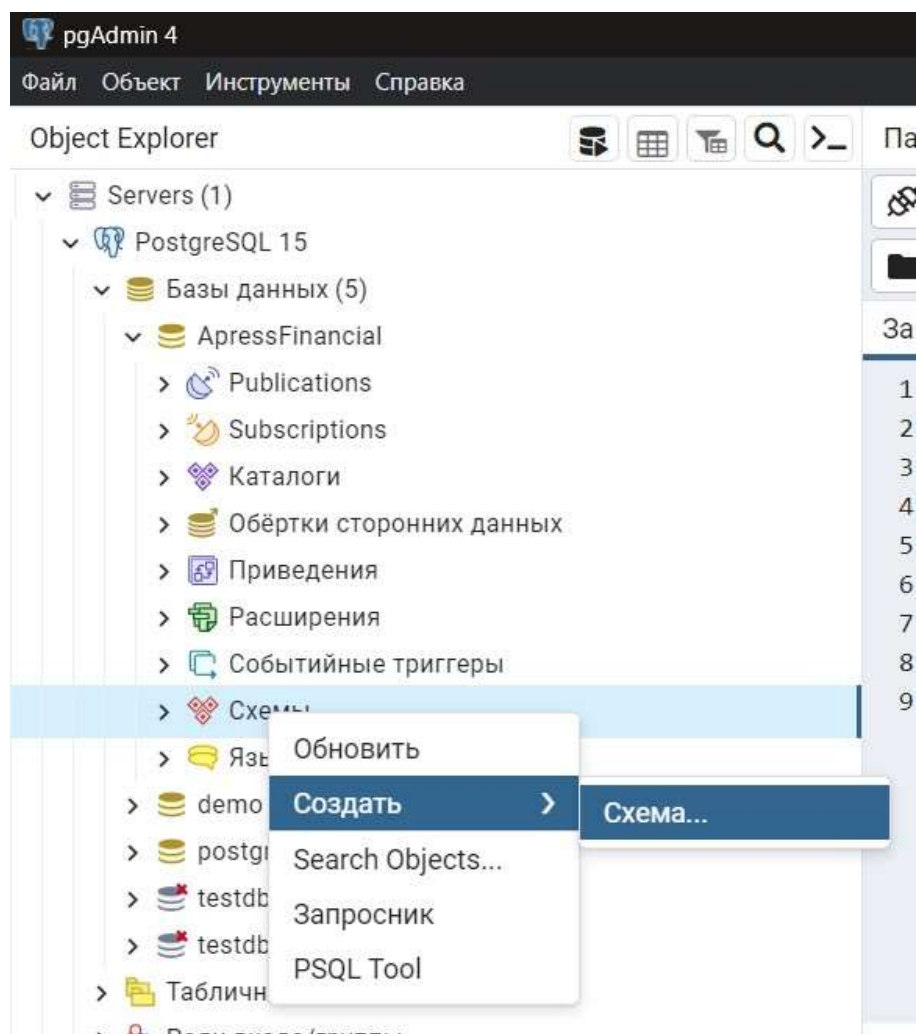


Рисунок 6.1 – Обзорщик объектов. Создание схемы.

2. Создайте схему EmployeesDepartments
3. Убедитесь через Обзорщик объектов в том, что схема создана
4. Для создания второй схемы необходимо открыть запросник к базе HR
5. Создайте схему, используя следующий код (рисунок 6.2):

```
CREATE SCHEMA IF NOT EXISTS "countries"  
AUTHORIZATION postgres;
```

Рисунок 6.2 – Создание схемы DDL оператором CREATE.

6. Убедитесь через Обзорщик объектов в том, что схема создана.