Министерство образования и науки

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Отчет

по лабораторным работам

по дисциплине **«Базы данных»**

Выполнил: **студент группы K3220 И.О. Мороз**

Проверил: **ст. преподаватель И.С. Осетрова**

Санкт-Петербург

2020

# «Создание учебной базы данных»

## Цель работы.

Создание учебной базы данных с использованием SQL запросов. Создание сценария базы данных.

## Задачи, решаемые при выполнении работы.

### Подключение к Microsoft SQL Server.

### Создание учебной базы данных.

### Создание сценария базы данных.

### Удаление учебной базы данных.

### Создание учебной базы данных в Query Editor.

## Объект исследования.

Microsoft SQL Server, учебная база данных.

## Исходные данные.

### Docker-контейнер с SQL Server, DataGrip IDE.

### Методическое пособие.

### Методическое пособие.

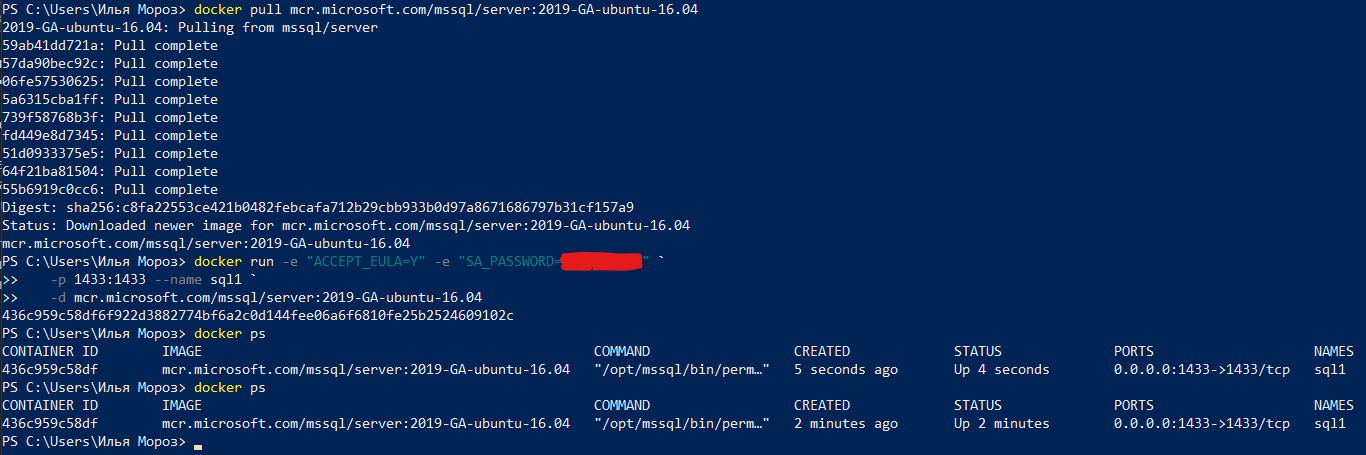
### Методическое пособие.

### Методическое пособие.

## Выполнение работы.

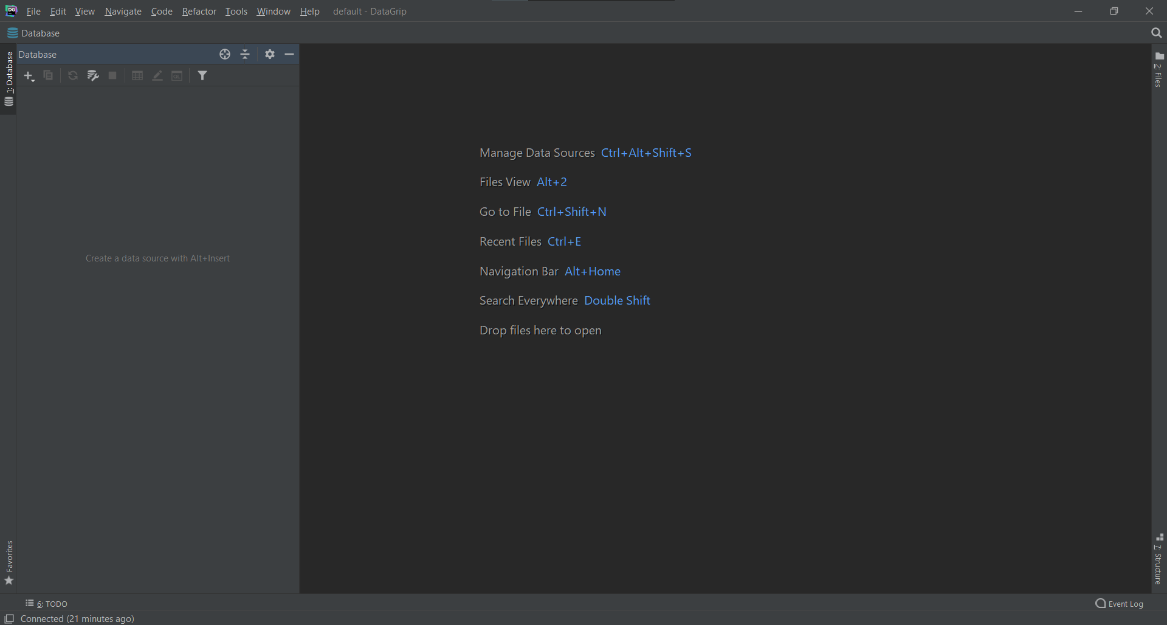
### Подключение к Microsoft SQL Server.

Перед выполнением задания было необходимо поднять SQL Server в локальной сети и подключиться к ней. Для упрощения процесса поднятия SQL Server был использован Docker контейнер от Microsoft (подробнее см. [Краткое руководство. Запуск образов контейнеров SQL Server в Docker](https://docs.microsoft.com/ru-ru/sql/linux/quickstart-install-connect-docker?view=sql-server-ver15&pivots=cs1-cmd))

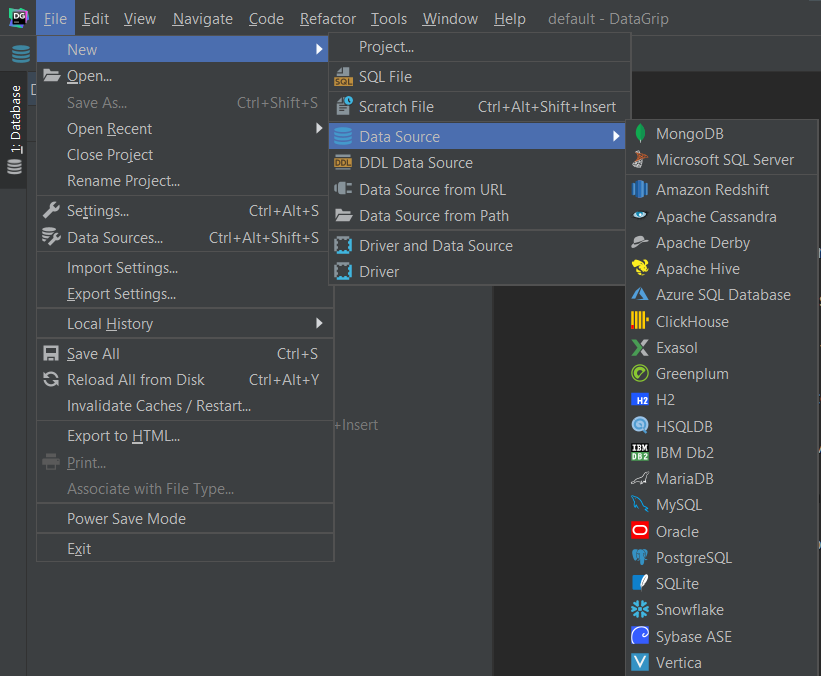


*Рисунок 1 – Запуск Docker контейнера с SQL Server*

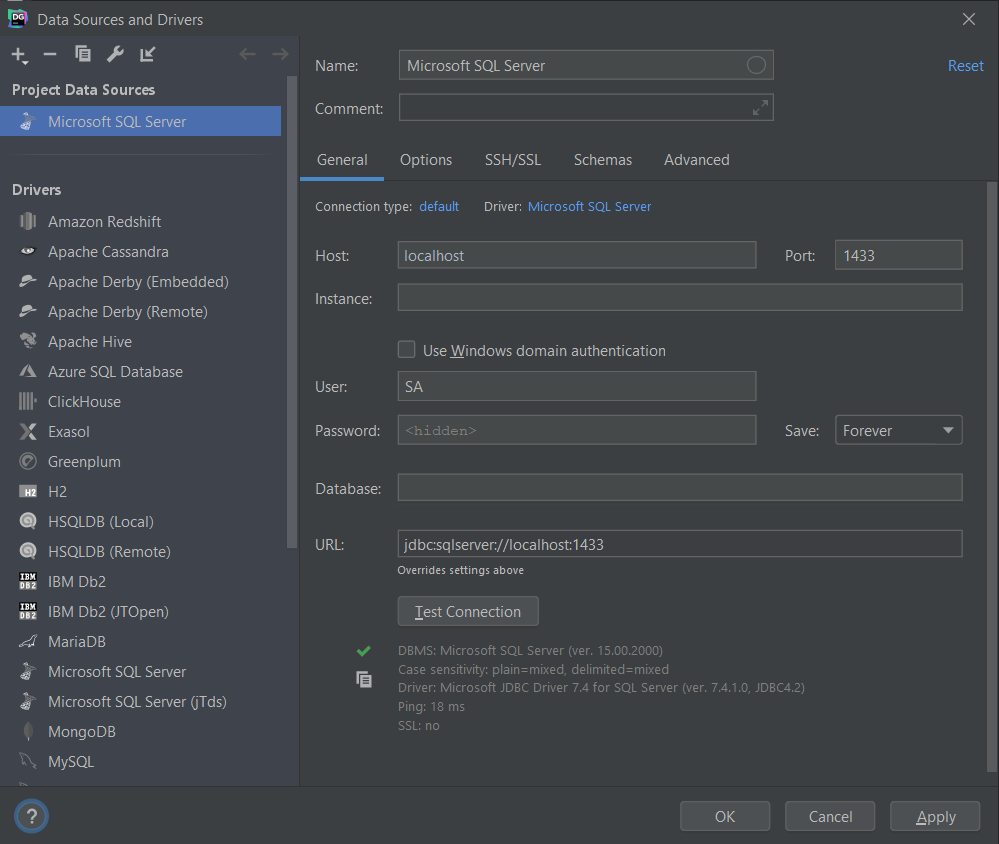
После запуска SQL Server подключение было выполнено с использованием DataGrip:



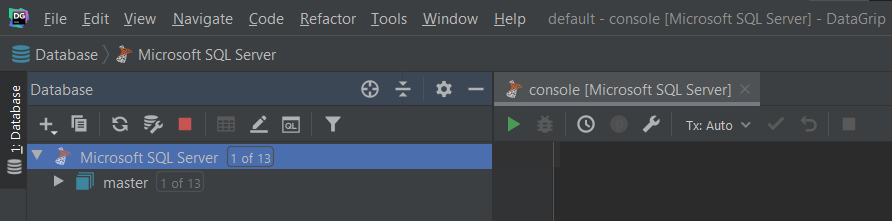
*Рисунок 2 – Окно DataGrip IDE*



*Рисунок 3 – Создание нового Data Source. File – New – Data Source – Microsoft SQL Server*



*Рисунок 4 – Настройка нового подключения*



*Рисунок 5 – Успешное подключение к SQL Server*

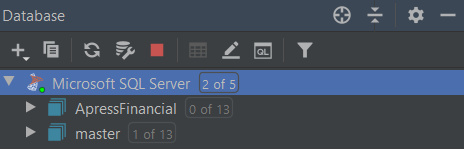
### Создание учебной базы данных.

С помощью SQL запроса была создана новая база ApressFinancial. При создании базы данных был создан вторичный файл данных с логическим именем «ApressFinancial\_act»

CREATE DATABASE ApressFinancial  
ON  
 PRIMARY  
 ( NAME = N'ApressFinancial',  
 FILENAME = N'/var/opt/mssql/data/ApressFinancial.mdf',  
 SIZE = 8192KB,  
 MAXSIZE = UNLIMITED,  
 FILEGROWTH = 1024KB ),  
 FILEGROUP [SECONDARY] DEFAULT  
 ( NAME = N'ApressFinancial\_act',  
 FILENAME = N'/var/opt/mssql/data/ApressFinancial\_act.ndf',  
 SIZE = 8192KB,  
 MAXSIZE = UNLIMITED,  
 FILEGROWTH = 1024KB )  
LOG ON  
 ( NAME = N'ApressFinancial\_log',  
 FILENAME = N'/var/opt/mssql/data/ApressFinancial\_log.ldf',  
 SIZE = 1024KB,  
 MAXSIZE = 2048KB,  
 FILEGROWTH = 10% )  
GO

*Листинг 1 – SQL запрос для создания базы данных «ApressFinancial»*

После выполнения данного SQL запроса была создана база данных.



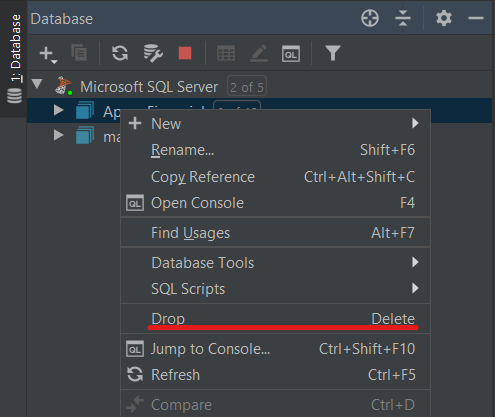
*Рисунок 6 – База данных создана*

### Создание сценария базы данных.

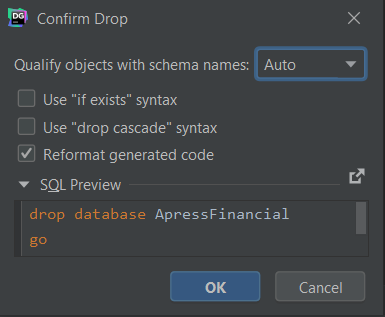
Так как DataGrip IDE поддерживает создание базы данных только с помощью SQL запроса, в предыдущем задании был создан файл сценария базы данных *create\_database.sql*. Листинг этого файла прикреплён в задании 1 (см. Листинг 1).

### Удаление учебной базы данных.

В данном задании было необходимо удалить только что созданную базу данных ApressFinancial.

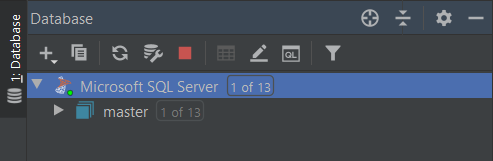


*Рисунок 7 – Команда для удаления базы данных в контекстном меню*



*Рисунок 8 – Подтверждение удаления базы данных*

База данных успешно удалена.

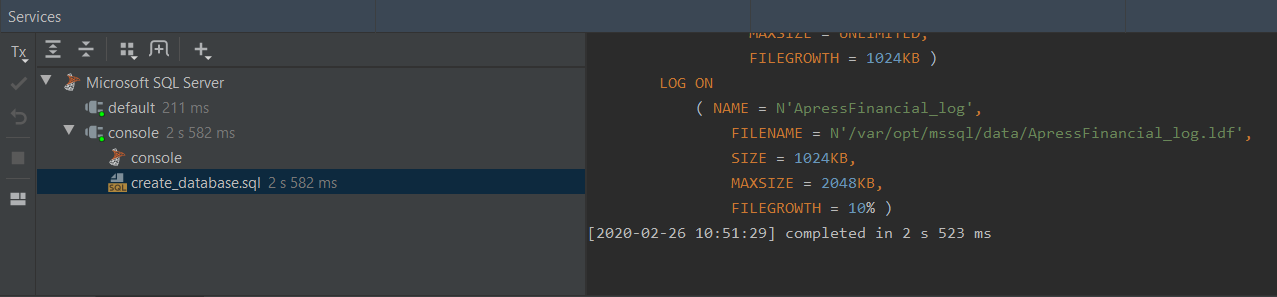


*Рисунок 9 – Структура Microsoft SQL Server после удаления базы данных*

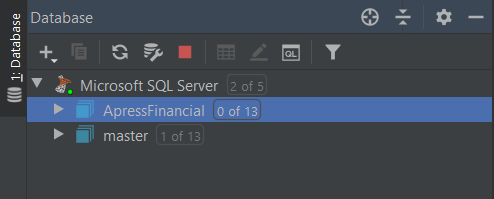
### Создание учебной базы данных в Query Editor.

Для создания учебной базы данных с помощью SQL запроса был открыт уже созданный файл *create\_database.sql*.

Файл был выполнен.



*Рисунок 10 – Результат выполнения файла*

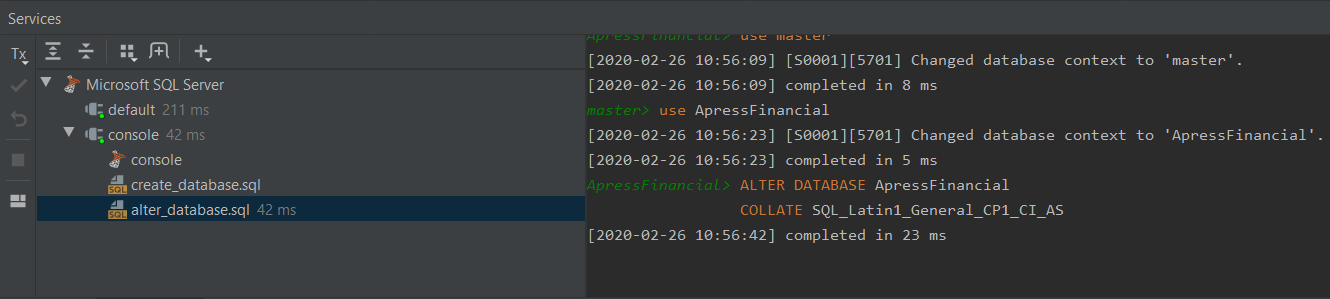


*Рисунок 11 – База данных вновь создана*

Для изменения параметров сортировки для символьных строк был создан файл SQL инструкций *alter\_database.sql*.

ALTER DATABASE ApressFinancial  
COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS  
GO

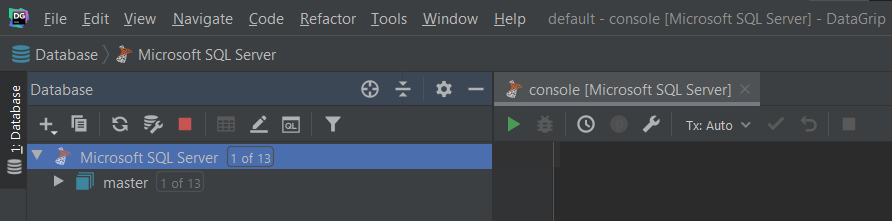
*Листинг 2 – Изменение параметров сортировки для символьных строк*



*Рисунок 12 – Результат выполнения файла*

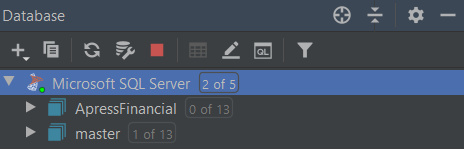
## Окончательные результаты.

### Подключение к базе данных было выполнено.



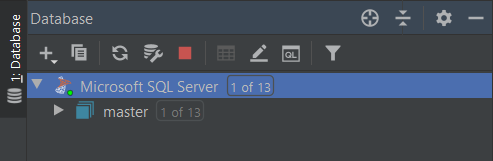
*Рисунок 13 – Успешное подключение к SQL Server*

### Была создана учебная база данных.



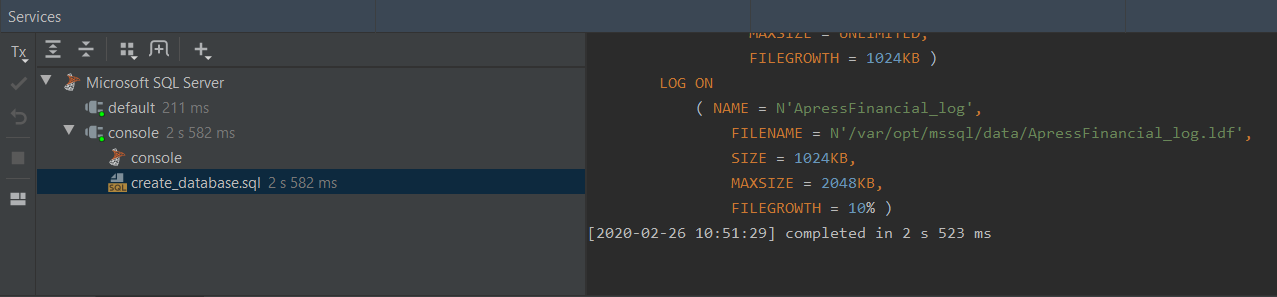
*Рисунок 14 – База данных создана*

### Учебная база данных была удалена с использованием контекстного меню.

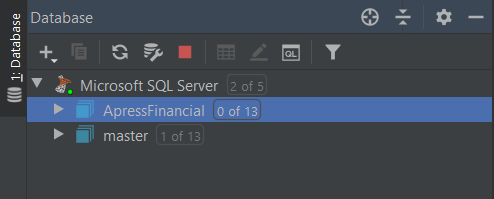


*Рисунок 15 – Структура Microsoft SQL Server после удаления базы данных*

### С помощью Query Editor был создан новый экземпляр учебной базы данных.



*Рисунок 16 – Результат выполнения файла*



*Рисунок 17 – База данных вновь создана*

## Выводы и анализ результатов работы.

В ходе данной лабораторной работы локально был запущен Microsoft SQL Server. При запуске использовался Docker контейнер. К созданному SQL Server было выполнено подключение с использованием DataGrip IDE.

При выполнении заданий были созданы файлы SQL сценариев *create\_database.sql* и *alter\_database.sql*. При выполнении этих файлов была создана учебная база данных ApressFinancial. Для данной базы данных были изменены параметры сортировки для символьных строк.

# «Безопасность: создание схем»

## Цель работы.

Создание схем для базы данных с использованием контекстного меню и Query Editor.

## Задачи, решаемые при выполнении работы.

### Создание новой схемы с использованием контекстного меню.

### Создание новой схемы с использованием Query Editor.

## Объект исследования.

Microsoft SQL Server, учебная база данных, схемы базы данных.

## Исходные данные.

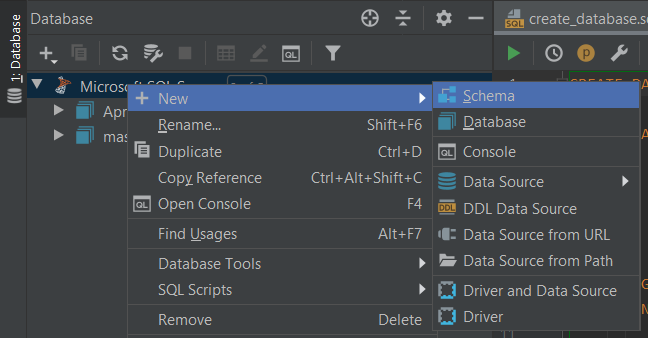
### Методическое пособие, учебная база данных.

### Методическое пособие, учебная база данных.

## Выполнение работы.

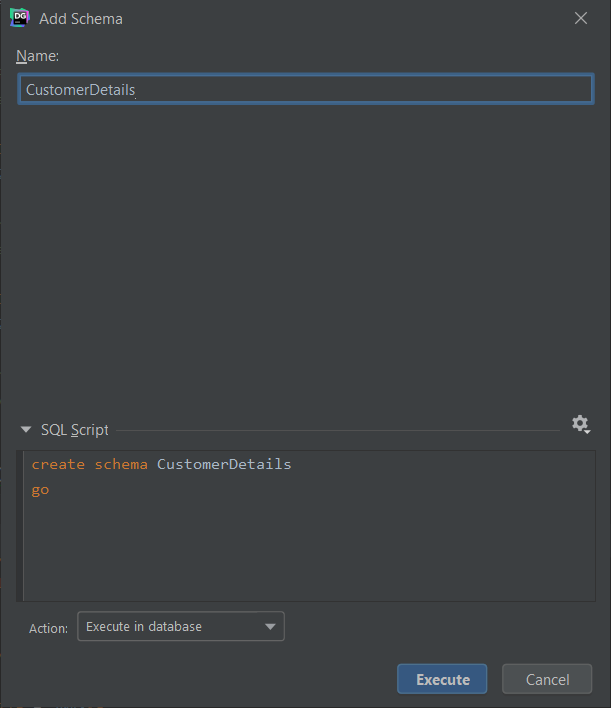
### Создание схемы в DataGrip IDE.

Для создания новой схемы необходимо нажать ПКМ на Data source, выбрать New – Schema.



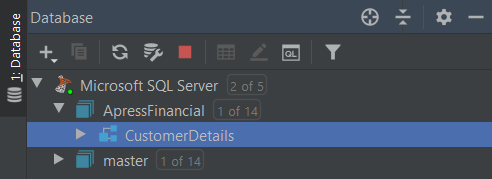
*Рисунок 18 – Контекстное меню создания новой схемы*

В появившемся окне было введено имя схемы «CustomerDetails».



*Рисунок 19 – Окно создания новой схемы*

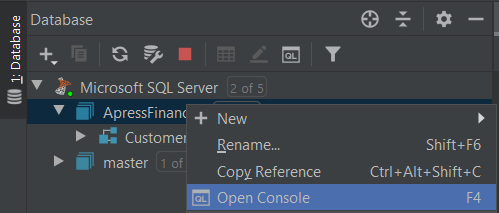
После создания новой схемы она отобразилась у базы данных ApressFinancial.



*Рисунок 20 – Только что созданная схема CustomerDetails*

### Создание схемы с помощью SQL запроса.

Для создания схемы с помощью SQL запроса необходимо открыть консоль для БД ApressFinancial.



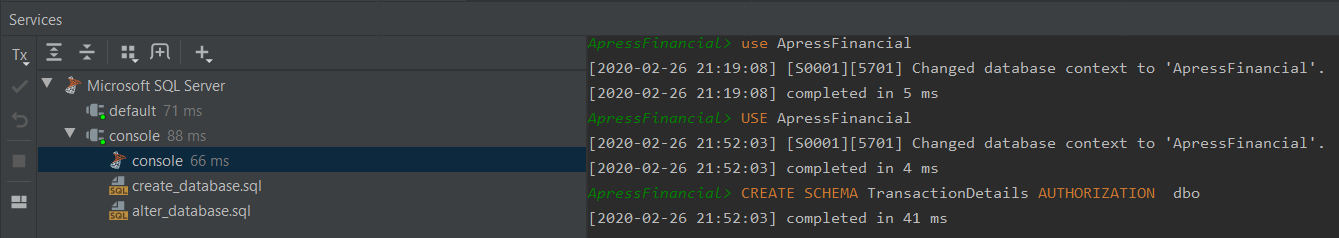
*Рисунок 21 – Открытие консоли*

В консоли был введён следующий SQL запрос.

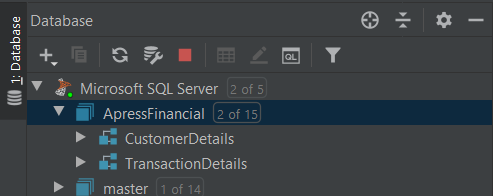
USE ApressFinancial  
GO  
  
CREATE SCHEMA TransactionDetails AUTHORIZATION dbo  
GO

*Листинг 3 – Создание схемы TransactionDetails*

Данный запрос был успешно выполнен.



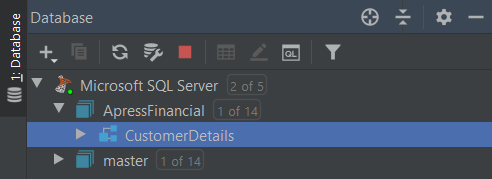
*Рисунок 22 – Успешное выполнение SQL запроса*



*Рисунок 23 – Схема появилась у базы данных*

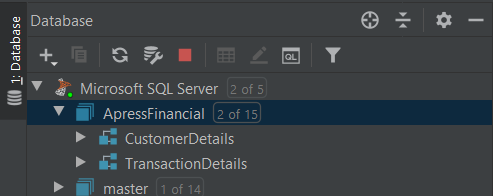
## Окончательные результаты.

### С помощью контекстного меню схема была создана.



*Рисунок 24 – Только что созданная схема CustomerDetails*

### С помощью SQL запроса схема была создана.



*Рисунок 25 – Схема появилась у базы данных*

## Выводы и анализ результатов работы.

В ходе данной лабораторной работы локально были созданы схемы базы данных с помощью контекстного меню DataGrip IDE и с помощью SQL запроса. В результате был получен файл с запросом SQL *create\_schema.sql*.