

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙ-  
СКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Анализ данных**

**Отчет по лабораторной работе №2.20(С Duckdb)**

**Тема: «Изучение Duckdb»**

Выполнил студент группы

ИВТ-б-о-21-1

Уланбекова А.У « »\_\_\_\_\_20\_\_г.

Подпись студента\_\_\_\_\_

Работа защищена « »\_\_\_\_\_20\_\_г.

Проверил доцент

Кафедры инфокоммуникаций, старший  
преподаватель

Воронкин Р.А.

\_\_\_\_\_  
(подпись)

Ставрополь 2023

**Цель работы:** исследовать возможности Duckdb.

**Ход работы:**

**1. Создал репозиторий в GitHub**, дополнил правила в .gitignore для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на компьютер и организовал в соответствии с моделью ветвления git-flow.

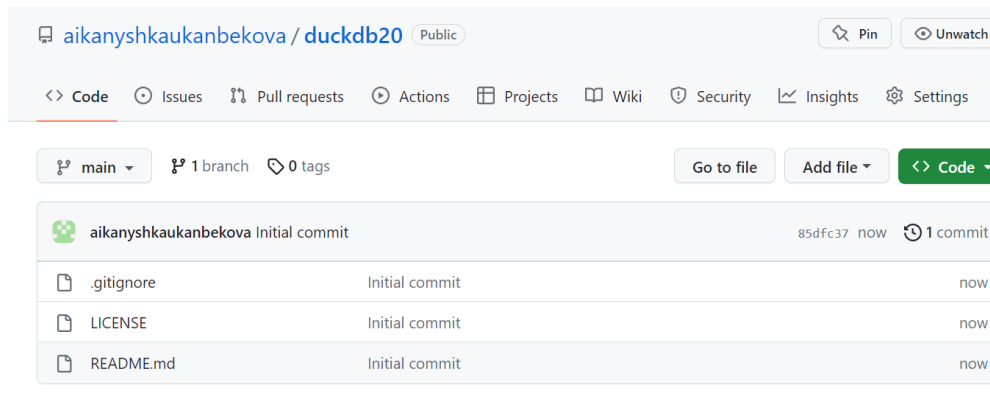


Рисунок 1.1 – Созданный репозиторий

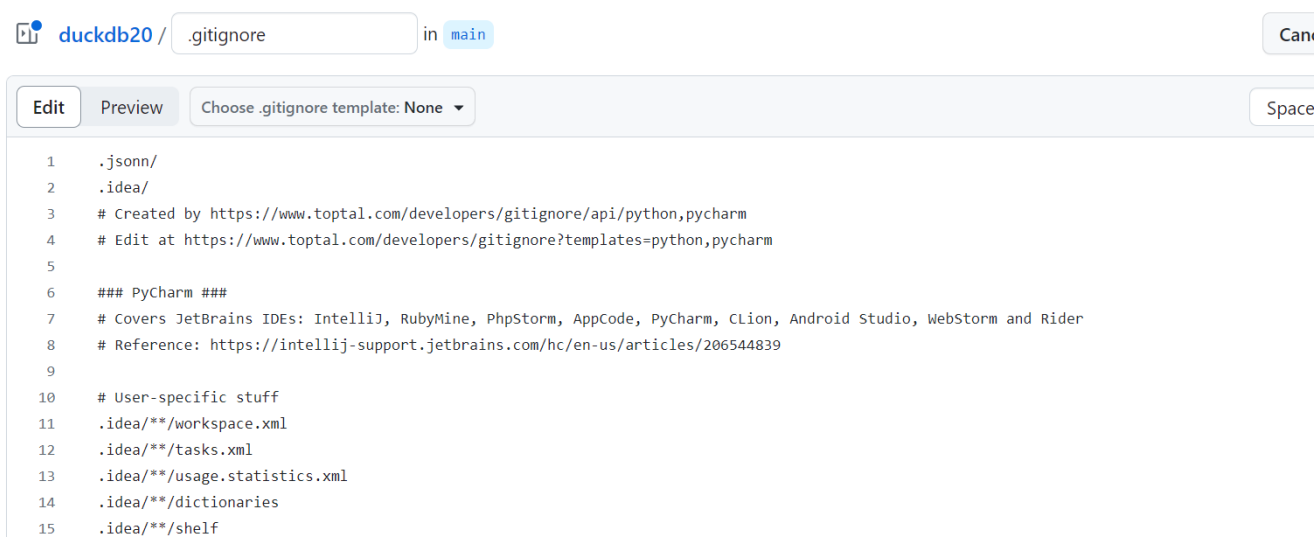


Рисунок 1.2 – Изменения в .gitignore

```
Администратор: Git CMD

C:\Users\User>cd C:\Users\User\Desktop\2.1\20

C:\Users\User\Desktop\2.1\20>git clone https://github.com/aikanyshkaukanbekova/duckdb20.git
Cloning into 'duckdb20'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), 4.49 KiB | 765.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.

C:\Users\User\Desktop\2.1\20>
```

Рисунок 1.3 – Клонирование репозитория

## №2. Работа с duckdb.

Первое, что делаем необходимо сделать, это открываем консоль powershell и устанавливаем scoop. После установки scoop ставим пакет duckdb.

```
PS C:\Users\Admin> Set-ExecutionPolicy RemoteSigned -scope CurrentUser

Изменение политики выполнения
Политика выполнения защищает компьютер от ненадежных сценариев. Изменение политики выполнения может поставить под угрозу безопасность системы, как описано в разделе справки, вызываемом командой about_Execution_Policies и расположенном по адресу https://go.microsoft.com/fwlink/?LinkID=135170 . Вы хотите изменить политику выполнения?
[Y] Да - Y [A] Да для всех - A [N] Нет - N [L] Нет для всех - L [S] Приостановить - S [?] Справка
(значением по умолчанию является "N"):Y
PS C:\Users\Admin> iwr -useb get.scoop.sh | iex
Initializing...
Downloading ...
Creating shim...
Adding ~\scoop\shims to your path.
Scoop was installed successfully!
Type 'scoop help' for instructions.
PS C:\Users\Admin> install duck.db
install : Имя "install" не распознано как имя командлета, функции, файла сценария или выполняемой программы. Проверьте
правильность написания имени, а также наличие и правильность пути, после чего повторите попытку.
строка:1 знак:1
+ install duck.db
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (install:String) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Admin> scoop install duck.db
couldn't find manifest for 'duck.db'
PS C:\Users\Admin> scoop install duckdb
Installing 'duckdb' (0.8.0) [64bit] from main bucket
duckdb_cli-windows-amd64.zip (8,5 MB) [=====] 100%
Checking hash of duckdb_cli-windows-amd64.zip ... ok.
Extracting duckdb_cli-windows-amd64.zip ... done.
Linking ~\scoop\apps\duckdb\current => ~\scoop\apps\duckdb\0.8.0
Creating shim for 'duckdb'.
'duckdb' (0.8.0) was installed successfully!
PS C:\Users\Admin>
```

Рисунок 1 – Установка duckdb

```
PS C:\Windows\system32> duckdb country
v0.8.0 e8e4cea5ec
Enter ".help" for usage hints.
D CREATE TABLE country AS SELECT * FROM read_csv_auto(c:\users\admin\desktop\downloads\report_2018-2019.csv)
>
```

Рисунок 1.1 – Создание базы данных и импорт данных в созданную таблицу

```
D CREATE TABLE country AS SELECT * FROM read_csv_auto('C:\Users\Admin\Downloads\report_2018-2019.csv');
D SELECT count(*) FROM country;
```

| count_star() |
|--------------|
| int64        |
| 312          |

Рисунок 2 – Запрос №1

```
D COPY (SELECT count(*) FROM country) TO 'req1.json';
D COPY (SELECT count(*) FROM country) TO 'req1.csv' (HEADER, DELIMITER ',');
D
```

Рисунок 3 – Сохранение запроса №1

```
D SELECT "Country or region", Year FROM country WHERE Score=(select Min(Score) FROM country);
```

| Country or region<br>varchar | Year<br>int64 |
|------------------------------|---------------|
| South Sudan                  | 2019          |

```
D COPY (SELECT "Country or region", Year FROM country WHERE Score=(select Min(Score) FROM country)) TO 'req3.json';
D COPY (SELECT "Country or region", Year FROM country WHERE Score=(select Min(Score) FROM country)) TO 'req3.csv' (HEADER, DELIMITER ',');
```

Рисунок 4 – Запрос №2 и сохранение его

```
D SELECT "Country or region", Year, Score FROM country where "GDP per capita" = (select max("GDP per capita") FROM country);
```

| Country or region<br>varchar | Year<br>int64 | Score<br>double |
|------------------------------|---------------|-----------------|
| United Arab Emirates         | 2018          | 6.774           |

```
D COPY (SELECT "Country or region", Year, Score FROM country where "GDP per capita" = (select max("GDP per capita") FROM country)) TO 'req4.json';
D COPY (SELECT "Country or region", Year, Score FROM country where "GDP per capita" = (select max("GDP per capita") FROM country)) TO 'req4.csv' (HEADER, DELIMITER ',');
```

Рисунок 5 – Запрос №3 и сохранение его

Рисунок 6 – Запрос №4 и сохранение

```
D SELECT "Country or region", year, Score FROM country where year in ('2018') ORDER BY 3 DESC limit 10;
```

| Country or region<br>varchar | Year<br>int64 | Score<br>double |
|------------------------------|---------------|-----------------|
| Finland                      | 2018          | 7.632           |
| Norway                       | 2018          | 7.594           |
| Denmark                      | 2018          | 7.555           |
| Iceland                      | 2018          | 7.495           |
| Switzerland                  | 2018          | 7.487           |
| Netherlands                  | 2018          | 7.441           |
| Canada                       | 2018          | 7.328           |
| New Zealand                  | 2018          | 7.324           |
| Sweden                       | 2018          | 7.314           |
| Australia                    | 2018          | 7.272           |

10 rows 3 columns

```
D COPY (SELECT "Country or region", year, Score FROM country where year in ('2018') ORDER BY 3 DESC limit 10) TO 'req5.json';
D COPY (SELECT "Country or region", year, Score FROM country where year in ('2018') ORDER BY 3 DESC limit 10) TO 'req5.csv' (HEADER, DELIMITER ',');
```

Рисунок 7 – Запрос №5 и его сохранение












|   |         |                  |                        |          |
|---|---------|------------------|------------------------|----------|
|  | country | 28.05.2023 22:07 | Файл "DB"              | 1 036 КБ |
|  | req5    | 28.05.2023 22:06 | Microsoft Excel Com... | 1 КБ     |
|  | req5    | 28.05.2023 22:06 | JSON File              | 1 КБ     |
|  | req4    | 28.05.2023 22:04 | Microsoft Excel Com... | 1 КБ     |
|  | req4    | 28.05.2023 22:03 | JSON File              | 1 КБ     |
|  | req3    | 28.05.2023 22:01 | Microsoft Excel Com... | 1 КБ     |
|  | req3    | 28.05.2023 22:01 | JSON File              | 1 КБ     |
|  | req2    | 28.05.2023 21:58 | Microsoft Excel Com... | 1 КБ     |
|  | req2    | 28.05.2023 21:58 | JSON File              | 1 КБ     |
|  | req1    | 28.05.2023 21:56 | Microsoft Excel Com... | 1 КБ     |
|  | req1    | 28.05.2023 21:55 | JSON File              | 1 КБ     |

Рисунок 8 – Сохранённые запросы и база данных

**Вывод:** в результате выполнения лабораторной работы были исследованы на практике базовые возможности системы управления базами данных Duckdb.