

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

Кафедра инфокоммуникаций
«Основы работы с DuckDB»

Отчет по лабораторной работе
по дисциплине «Программирование на Python»

Выполнил студент группы ИВТ-б-о-21-1

Толубаев Рамиль Ахметович

«20» мая 2023г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: исследовать базовые возможности системы управления базами данных DuckDB.

Порядок выполнения работы:

1. Решить задачи:

Задание 1. Выполните команды. Что вернула команда .schema ?

```
D create table customer(name string);
D select * from customer;

name
varchar

0 rows

D .schema customer
CREATE TABLE customer("name" VARCHAR);;
```

Рисунок 1 - Результат выполнения задания 1

Команда .schema customer вывела запрос CREATE, соответствующий таблице customer.

Задание 2. С помощью команды .help найдите команду, которая отвечает за вывод времени выполнения запроса.

```
D .timer on
D select count(*) from read_csv_auto('./city.csv');

count_star()
int64

1117

Run Time (s): real 0.078 user 0.046875 sys 0.000000
```

Рисунок 5 - Результат выполнения задания 2

Задание 3. Загрузить файл city.csv. Выполните запрос. Какое число он вернул?

```
D select max(length(city)) from read_csv_auto('./city.csv');

max(length(city))
int64

25
```

Рисунок 6 - Результат выполнения задания 3

Задание 4. загрузите файл city.csv, но без использования опции --csv.

```
D create table city as select * from read_csv_auto('./city.csv');
D
```

Рисунок 7 - Результат выполнения задания 4

Задание 5. Напишите запрос, который посчитает количество городов для каждого часового пояса в Сибирском и Приволжском федеральных округах. Выведите столбцы `timezone` и `city_count`, отсортируйте по значению часового пояса.

```
D select timezone, count(city) as city_count from read_csv_auto('./city.csv')
> where federal_district in ('Приволжский', 'Сибирский')
> group by timezone
> order by timezone ASC;
```

timezone varchar	city_count int64
UTC+3	101
UTC+4	41
UTC+5	58
UTC+6	6
UTC+7	86
UTC+8	22

Рисунок 8 - Результат выполнения задания 5

Задание 6. Напишите запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию.

```
D create table city as select * from read_csv_auto('./city.csv');
D select
> timezone, count(city) as city_count from city
> group by timezone
> order by city_count DESC;
```

timezone varchar	city_count int64
UTC+3	656
UTC+5	173
UTC+7	86
UTC+4	66
UTC+9	31
UTC+8	28
UTC+10	22
UTC+2	22
UTC+11	17
UTC+12	6
UTC+6	6
11 rows	2 columns

Рисунок 9 - Результат выполнения задания 6

2. Выполните индивидуальное задание. Каждый запрос к базе данных сохраните в файл с расширением sql.

Загрузите в SQLite выбранный Вами датасет в формате CSV. Сформируйте более пяти запросов к таблицам БД. Выгрузите результат выполнения запросов в форматы CSV и JSON.

Запрос 1.

```
D create table covid as select * from read_csv_auto('./covid.csv');
D SELECT Country, CasesLast7d
> FROM covid
> WHERE CasesLast7d BETWEEN 2500 AND 3000;
```

Country varchar	CasesLast7d int64
Colombia	2512

```
D copy(
> SELECT Country, CasesLast7d
> FROM covid
> WHERE CasesLast7d BETWEEN 2500 AND 3000) to 'req1.json';
D copy(
> SELECT Country, CasesLast7d
> FROM covid
> WHERE CasesLast7d BETWEEN 2500 AND 3000) to 'req1.csv' (header, delimiter ',');
```

Рисунок 10 - Результат выполнения запроса 1 индивидуального задания

Запрос 2.

```
D SELECT Country, DeathsLast7d FROM covid WHERE DeathsLast7d == 0 LIMIT 2;
```

Country varchar	DeathsLast7d int64
Falkland Islands	0
French Guiana	0

```
D copy(
> SELECT Country, DeathsLast7d FROM covid WHERE DeathsLast7d == 0 LIMIT 2) to 'req2.json';
D copy(
> SELECT Country, DeathsLast7d FROM covid WHERE DeathsLast7d == 0 LIMIT 2) to 'req2.csv' (header, delimiter ',');
```

Рисунок 11 - Результат выполнения запроса 2 индивидуального задания

Запрос 3.

```
D SELECT Country, Population FROM covid ORDER BY Population ASC LIMIT 10;
```

Country varchar	Population int64
Falkland Islands	3655
French Guiana	312076
Suriname	595733
Guyana	793124
Uruguay	3494576
Paraguay	7284217
Bolivia	11948668
Ecuador	18106806
Chile	19400316
Venezuela	28296430

10 rows 2 columns

```
D copy(  
> SELECT Country, Population FROM covid ORDER BY Population ASC LIMIT 10) to 'req3.json';  
D copy(  
> SELECT Country, Population FROM covid ORDER BY Population ASC LIMIT 10) to 'req3.csv' (header, delimiter ',');  
D
```

Рисунок 12 - Результат выполнения запроса 3 индивидуального задания

Запрос 4.

```
D SELECT Country, CasesPreceding7d, Population FROM covid WHERE CasesPreceding7d BETWEEN 4000 AND 4999 ORDER BY Population DESC LIMIT 3;
```

Country varchar	CasesPreceding7d int64	Population int64
Colombia	4312	51821865
Peru	4901	33766958
Ecuador	4629	18106806

```
D copy(  
> SELECT Country, CasesPreceding7d, Population FROM covid WHERE CasesPreceding7d BETWEEN 4000 AND 4999 ORDER BY Population DESC LIMIT 3) to 'req4.json';  
D copy(  
> SELECT Country, CasesPreceding7d, Population FROM covid WHERE CasesPreceding7d BETWEEN 4000 AND 4999 ORDER BY Population DESC LIMIT 3) to 'req4.csv' (header, delimiter ',');  
D
```

Рисунок 13 - Результат выполнения запроса 4 индивидуального задания

Запрос 5.

```

D SELECT Country, WeeklyCasePerChange, WeeklyDeathPerChange FROM covid WHERE WeeklyCasePerChange <= 0 AND WeeklyDeathPerChange != 0
LIMIT 8;

```

Country varchar	WeeklyCasePerChange int64	WeeklyDeathPerChange int64
Argentina	-50	35
Bolivia	-31	-44
Brazil	-19	-22
Chile	-38	-33
Colombia	-42	-22
Ecuador	-29	-87
Guyana	-20	-100
Paraguay	-34	-42

```

D copy(
> SELECT Country, WeeklyCasePerChange, WeeklyDeathPerChange FROM covid WHERE WeeklyCasePerChange <= 0 AND WeeklyDeathPerChange != 0
LIMIT 8) to 'req5.json';
D copy(
> SELECT Country, WeeklyCasePerChange, WeeklyDeathPerChange FROM covid WHERE WeeklyCasePerChange <= 0 AND WeeklyDeathPerChange != 0
LIMIT 8) to 'req5.csv' (header, delimiter ',');

```

Рисунок 14 - Результат выполнения запроса 5 индивидуального задания

Запрос 6

```

D SELECT Country, CasesLast7d1MPop, CasesPreceding7d, Population FROM covid WHERE CasesLast7d1MPop BETWEEN 0 AND 250 AND CasesPreceding7d != 0 ORDER BY Population DESC;

```

Country varchar	CasesLast7d1MPop int64	CasesPreceding7d int64	Population int64
Colombia	48	4312	51821865
Peru	119	4901	33766958
Venezuela	16	1260	28296430
Ecuador	181	4629	18106806
Bolivia	133	2312	11948668
Paraguay	111	1216	7284217
Guyana	50	50	793124

```

D copy(
> SELECT Country, CasesLast7d1MPop, CasesPreceding7d, Population FROM covid WHERE CasesLast7d1MPop BETWEEN 0 AND 250 AND CasesPreceding7d != 0 ORDER BY Population DESC) to 'req6.json';
D copy(
> SELECT Country, CasesLast7d1MPop, CasesPreceding7d, Population FROM covid WHERE CasesLast7d1MPop BETWEEN 0 AND 250 AND CasesPreceding7d != 0 ORDER BY Population DESC) to 'req6.csv' (header, delimiter ',');

```

Рисунок 15 - Результат выполнения запроса 6 индивидуального задания

Вывод: были исследованы базовые возможности системы управления базами данных DuckDB.

