МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт искусственного интеллекта (ИИИ)
Кафедра промышленной информатики (ПИ)

ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

по дисциплине «Разработка баз данных»

Практические работы № 1-4

Студент группы ИКБО-25-22, Ракитин В.А. _____
(подпись)

Преподаватель Благовещенский В.Г.

_____
(подпись)

Отчёт представлен «___»_____2024г.

Москва 2024 г.

Практическая работа № 1

Постановка задачи: создать базу данных и таблицы в ней по выбранной теме, на основе разработанных моделей.

Результат работы:

Для создания базы данных (БД) требуется разработать ее структуру. Для разработки структуры используется диаграмма IDEF1X.

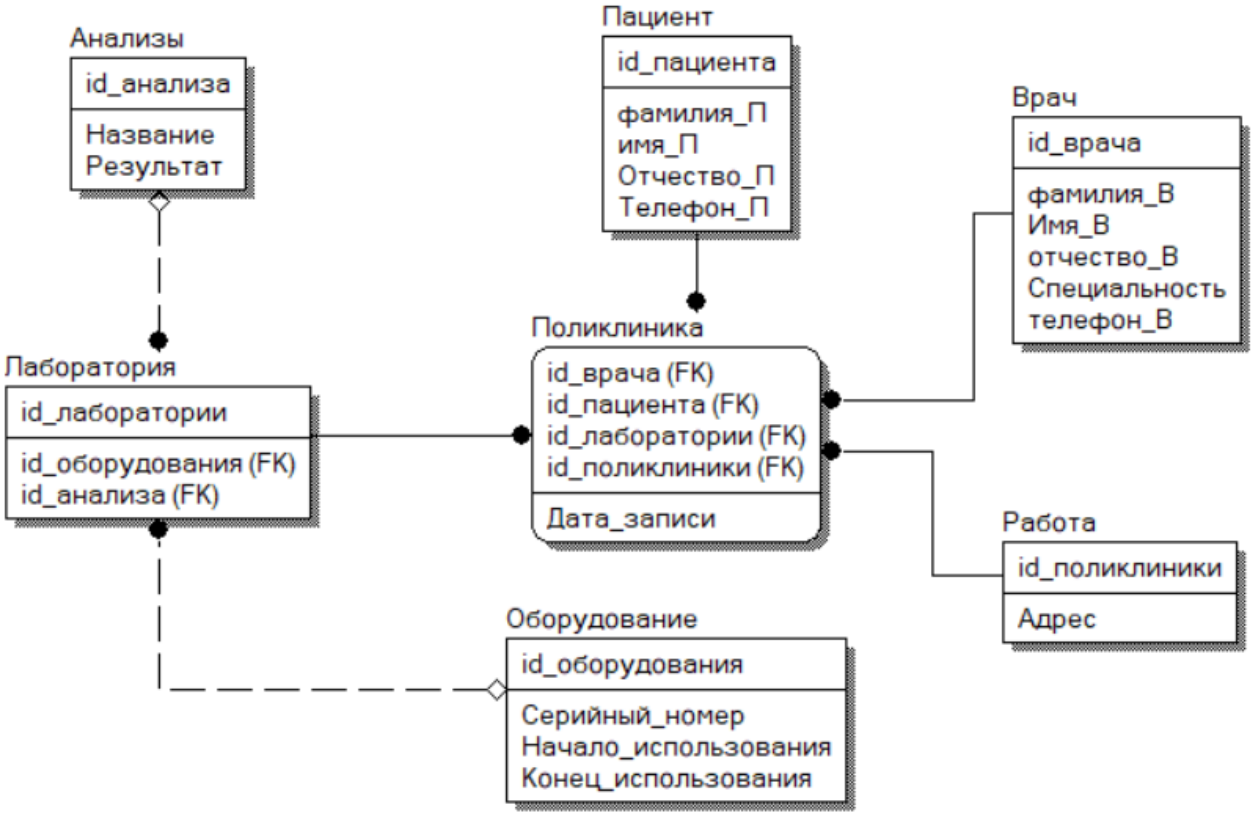

Рисунок 1.1 – Диаграмма IDFE1X по теме «Организация работы поликлиники»

Далее создаем БД с помощью команды create. Чтобы работать в созданной базе данных используем команду use ‘название базы данных’.

```
mysql> create database polyclinic
    -> ;
Query OK, 1 row affected (0.02 sec)
```

Рисунок 1.2 – Создание базы данных

```
mysql> use polyclinic;
Database changed
```

Рисунок 1.3 – Переход в созданную базу данных

Создаем таблицы polyclinic – поликлиника, doctor – врач, work – работа, patient – пациент, laboratory – лаборатория, analyzes – анализы, equipment – оборудование, с помощью метода create table ‘название’ (‘название столбца’ ‘тип

данных’,…). Также чтобы добавить внешние ключи в таблицу используется команда foreign key (‘название столбца’) references ‘название таблицы’ (‘внешний ключ’). Primary key (‘название столбца’,) делает столбец первичном ключом.

```
mysql> create table doctor(id_doctor int auto_increment, name_d varchar(20) not null, surname_d varchar(20) not null, patronymic_d varchar(20) null,
specialization varchar(20) not null, phone_d char not null, primary key(id_doctor));
Query OK, 0 rows affected (0.04 sec)
```

Рисунок 1.4 – Создание таблицы doctor

```
mysql> create table work(id_polyclinic int auto_increment, address varchar(30) not null, primary key(id_polyclinic));
Query OK, 0 rows affected (0.01 sec)
```

Рисунок 1.5 – Создание таблицы work

```
mysql> create table patient(id_patient int auto_increment, name_p varchar(20) not null, surname_p varchar(20) not null, patronymic_p varchar(20) null,
phone_p char not null, primary key(id_patient));
Query OK, 0 rows affected (0.02 sec)
```

Рисунок 1.6 – Создание таблицы patient

```
mysql> create table equipment(id_equipment int auto_increment, serial_number varchar(20) not null, start_dt date not null, end_dt date not null,
primary key(id_equipment));
Query OK, 0 rows affected (0.02 sec)
```

Рисунок 1.7 – Создание таблицы equipment

```
mysql> create table laboratory(id_laboratory int auto_increment, id_equipment int not null, id_analyzes int not null, primary key(id_laboratory));
Query OK, 0 rows affected (0.02 sec)
```

Рисунок 1.8 – Создание таблицы laboratory

```
mysql> create table analyzes(id_analyzes int auto_increment, name_analyzes varchar(20) not null, result varchar(50) not null, primary key(id_analyzes));

Query OK, 0 rows affected (0.01 sec)
```

Рисунок 1.9 – Создание таблицы analyzes

```
mysql> create table polyclinic(id_polyclinic int not null, id_doctor int not null, id_patient int not null, id_laboratory int not null,
record_dt date not null);
Query OK, 0 rows affected (0.02 sec)

mysql> alter table polyclinic add primary key(id_polyclinic, id_doctor, id_patient, id_laboratory);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Рисунок 1.10 – Создание таблицы polyclinic

С помощью команды alter создадим связь между таблицами.

```
mysql> alter table laboratory add foreign key(id_analyzes) references analyzes(id_analyzes);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table laboratory add foreign key(id_equipment) references equipment(id_equipment);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table polyclinic add foreign key(id_laboratory) references laboratory(id_laboratory);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table polyclinic add foreign key(id_polyclinic) references work(id_polyclinic);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table polyclinic add foreign key(id_patient) references patient(id_patient);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table polyclinic add foreign key(id_doctor) references doctor(id_doctor);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Рисунок 1.11 – Создание связей между таблицами

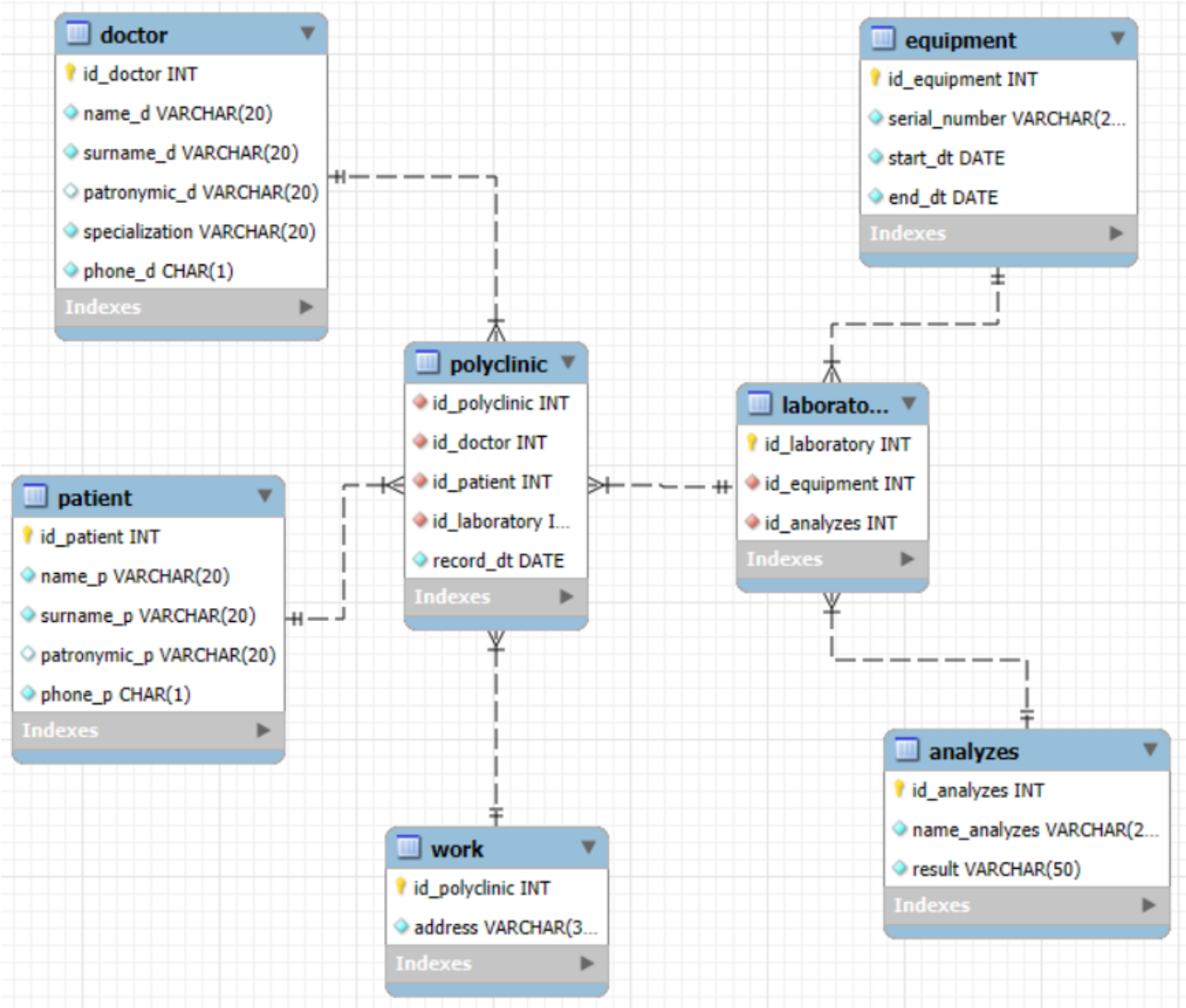

Рисунок 1.12 – EER диаграмма базы данных MySQL по теме “Организация работы поликлиники” в MySQL Workbench

Что бы добавить данные в таблицы используется команды insert into ‘название таблицы’ (‘название столбца’,…) values (‘данные’).

```
mysql> insert into patient(name_p, surname_p, patronymic_p, phone_p) values ('Vladimir', 'Semenov', 'Ilyich', '79269997003'), (
'Ilya', 'Petrov', 'Arturovich', '79269036543'), ('Arina', 'Smertina', 'Vladimirovna', '79267881020'), ('Julia', 'Grin', 'Andree
wna', '79261002222'), ('Vera', 'Vasileva', 'Petrovna', '79261055837'), ('Naimi', 'Farid', null, '79241119045'), ('Micle', 'Batk
in', null, '79139974637'), ('Nikita', 'Grigoriev', null, '79131110003'), ('Anna', 'Sharova', null, '79269910667'), ('Vasya', 'N
aymov', null, '79267780963');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Рисунок 1.13 – Заполнение таблицы patient

```
mysql> insert into doctor(name_d, surname_d, patronymic_d, specialization, phone_d) values ('Georgy', 'Stepanov', 'Ilyich', 'th
erapist', '79269621042'), ('Marina', 'Dmitrova', 'Petrovna', 'therapist', '79031338066'), ('Vladimir', 'Andreew', 'Ilyich', 'ps
ychologist', '79991001313'), ('Vasya', 'Nikitin', 'Vladimirovich', 'narcologist', '78031131567'), ('Petya', 'Kostin', 'Alexeyev
ich', 'therapist', '79261118004'), ('Michael', 'Pechin', 'Dmitrievich', 'surgeon', '79813041233'), ('Alexander', 'Reminen', nul
l, 'venerologist', '78991024435'); ('Michael', 'Stechin', null, 'dermatologist', '79041024489'), ('Arkady', 'Pilonov', null, 's
urgeon', '79001003447'), ('Kostya', 'Sidorov', null, 'narcologist', '79811055670');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Рисунок 1.14 – Заполнение таблицы doctor

```
mysql> insert into work (address) values ('Yablochkova street, 3A'), ('Glider street, 8'), ('Kravchenko street, 14');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Рисунок 1.15 – Заполнение таблицы work

```
mysql> insert into equipment(serial_number, start_dt, end_dt) values ('AA/1542TW14', '2021-03-01', '2026-03-01'), ('AA/1532SU71', '2024-05-13', '2025-11-13'
), ('AA/1567TT04', '2024-03-16', '2024-12-16'), ('AA/1442DR14', '2023-09-01', '2027-09-01'), ('AA/1940SE10', '2024-08-15', '2025-02-14'), ('AA/1940DE39', '2
023-09-27', '2028-09-27'), ('AA/1940VV14', '2022-11-14', '2024-11-14'), ('AA/1355DL14', '2024-09-22', '2025-05-17'), ('AA/1009RT13', '2023-07-03', '2025-07-
03'), ('AA/1335DC61', '2024-05-12', '2024-11-12');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Рисунок 1.16 – Заполнение таблицы equipment

```
mysql> insert into analyzes(name_analyzes, result) values('blood test', 'Hemoglobin and platelets are normal'), ('fluorography', 'The che
st is normal'), ('ECG', 'The heart rate is calm'), ('sugar analysis', 'The sugar level is normal'), ('urine analysis', 'The color is norm
al, there is no smell'), ('sugar analysis', 'The sugar level is above normal'), ('blood test', 'Hemoglobin and platelets are NO normal'),
 ('sugar analysis', 'High sugar levels'), ('ECG', 'Heart rate is above normal'), ('urine analysis', 'Kidney problems');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Рисунок 1.17 – Заполнение таблицы analyzes

```
mysql> insert into laboratory(id_equipment, id_analyzes) values('1', '1'), ('3', '2'), ('4', '3'), ('2', '4'), ('5', '5'), ('2', '6'), ('
6', '7'), ('7', '8'), ('4', '9'), ('9', '10');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Рисунок 1.18 – Заполнение таблицы laboratory

```
mysql> insert into polyclinic(id_polyclinic, id_doctor, id_patient, id_laboratory, record_dt) values('1', '1', '1', '1',
'2024-09-03'), ('1', '2', '2', '2', '2024-09-03'), ('2', '3', '3', '3', '2024-09-04'), ('2', '3', '4', '4', '2024-09-05')
, ('1', '4', '5', '5', '2024-09-06'), ('2', '6', '6', '6', '2024-09-07'), ('2', '6', '7', '7', '2024-09-07'), ('2', '8',
'8', '8', '2024-09-07'), ('3', '9', '9', '9', '10', '2024-09-08'), ('3', '10', '10', '9', '2024-09-08');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Рисунок 1.19 – Заполнение таблицы polyclinic

С помощью команды select * from ‘название таблицы’, проверим наши заполненные данные.

```
mysql> select * from patient;
+------------+----------+-----------+---------------+-------------+
| id_patient | name_p   | surname_p | patronymic_p  | phone_p     |
+------------+----------+-----------+---------------+-------------+
|          1 | Vladimir | Semenov   | Ilyich        | 79269997003 |
|          2 | Ilya     | Petrov    | Arturovich    | 79269036543 |
|          3 | Arina    | Smertina  | Vladimirovna  | 79267881020 |
|          4 | Julia    | Grin      | Andreewna     | 79261002222 |
|          5 | Vera     | Vasileva  | Petrovna      | 79261055837 |
|          6 | Naimi    | Farid     | NULL          | 79241119045 |
|          7 | Micle    | Batkin    | NULL          | 79139974637 |
|          8 | Nikita   | Grigoriev | NULL          | 79131110003 |
|          9 | Anna     | Sharova   | NULL          | 79269910667 |
|         10 | Vasya    | Naymov    | NULL          | 79267780963 |
+------------+----------+-----------+---------------+-------------+
10 rows in set (0.00 sec)
```

Рисунок 1.20 – Данные таблицы patient

```
mysql> select * from doctor
    -> ;
+-----------+-----------+-----------+---------------+----------------+-------------+
| id_doctor | name_d    | surname_d | patronymic_d  | specialization | phone_d     |
+-----------+-----------+-----------+---------------+----------------+-------------+
|         1 | Georgy    | Stepanov  | Ilyich        | therapist      | 79269621042 |
|         2 | Marina    | Dmitrova  | Petrovna      | therapist      | 79031338066 |
|         3 | Vladimir  | Andreew   | Ilyich        | psychologist   | 79991001313 |
|         4 | Vasya     | Nikitin   | Vladimirovich | narcologist    | 78031131567 |
|         5 | Petya     | Kostin    | Alexeyevich   | therapist      | 79261118004 |
|         6 | Michael   | Pechin    | Dmitrievich   | surgeon        | 79813041233 |
|         7 | Alexander | Reminen   | NULL          | venerologist   | 78991024435 |
|         8 | Michael   | Stechin   | NULL          | dermatologist  | 79041024489 |
|         9 | Arkady    | Pilonov   | NULL          | surgeon        | 79001003447 |
|        10 | Kostya    | Sidorov   | NULL          | narcologist    | 79811055670 |
+-----------+-----------+-----------+---------------+----------------+-------------+
10 rows in set (0.00 sec)
```

Рисунок 1.21 – Данные таблицы doctor

```
mysql> select * from work;
+---------------+------------------------+
| id_polyclinic | address                |
+---------------+------------------------+
|             1 | Yablochkova street, 3A |
|             2 | Glider street, 8       |
|             3 | Kravchenko street, 14  |
+---------------+------------------------+
3 rows in set (0.00 sec)
```

Рисунок 1.22 – Данные таблицы work

```
mysql> select * from equipment;
+--------------+---------------+------------+------------+
| id_equipment | serial_number | start_dt   | end_dt     |
+--------------+---------------+------------+------------+
|            1 | AA/1542TW14   | 2021-03-01 | 2026-03-01 |
|            2 | AA/1532SU71   | 2024-05-13 | 2025-11-13 |
|            3 | AA/1567TT04   | 2024-03-16 | 2024-12-16 |
|            4 | AA/1442DR14   | 2023-09-01 | 2027-09-01 |
|            5 | AA/1940SE10   | 2024-08-15 | 2025-02-14 |
|            6 | AA/1940DE39   | 2023-09-27 | 2028-09-27 |
|            7 | AA/1940VV14   | 2022-11-14 | 2024-11-14 |
|            8 | AA/1355DL14   | 2024-09-22 | 2025-05-17 |
|            9 | AA/1009RT13   | 2023-07-03 | 2025-07-03 |
|           10 | AA/1335DC61   | 2024-05-12 | 2024-11-12 |
+--------------+---------------+------------+------------+
10 rows in set (0.00 sec)
```

Рисунок 1.23 – Данные таблицы equipment

```
mysql> select * from analyzes;
+-------------+---------------+---------------------------------------+
| id_analyzes | name_analyzes | result                                |
+-------------+---------------+---------------------------------------+
|           1 | blood test    | Hemoglobin and platelets are normal   |
|           2 | fluorography  | The chest is normal                   |
|           3 | ECG           | The heart rate is calm                |
|           4 | sugar analysis | The sugar level is normal            |
|           5 | urine analysis | The color is normal, there is no smell |
|           6 | sugar analysis | The sugar level is above normal      |
|           7 | blood test    | Hemoglobin and platelets are NO normal |
|           8 | sugar analysis | High sugar levels                    |
|           9 | ECG           | Heart rate is above normal            |
|          10 | urine analysis | Kidney problems                      |
+-------------+---------------+---------------------------------------+
10 rows in set (0.00 sec)
```

Рисунок 1.24 – Данные таблицы analyzes

```
mysql> select * from laboratory;
+---------------+--------------+-------------+
| id_laboratory | id_equipment | id_analyzes |
+---------------+--------------+-------------+
|             1 |            1 |           1 |
|             2 |            3 |           2 |
|             3 |            4 |           3 |
|             4 |            2 |           4 |
|             5 |            5 |           5 |
|             6 |            2 |           6 |
|             7 |            6 |           7 |
|             8 |            7 |           8 |
|             9 |            4 |           9 |
|            10 |            9 |          10 |
+---------------+--------------+-------------+
10 rows in set (0.00 sec)
```

Рисунок 1.25 – Данные таблицы laboratory

```
mysql> select * from polyclinic;
+---------------+-----------+------------+---------------+------------+
| id_polyclinic | id_doctor | id_patient | id_laboratory | record_dt  |
+---------------+-----------+------------+---------------+------------+
|             1 |         1 |          1 |             1 | 2024-09-03 |
|             1 |         2 |          2 |             2 | 2024-09-03 |
|             2 |         3 |          3 |             3 | 2024-09-04 |
|             2 |         3 |          4 |             4 | 2024-09-05 |
|             1 |         4 |          5 |             5 | 2024-09-06 |
|             2 |         6 |          6 |             6 | 2024-09-07 |
|             2 |         6 |          7 |             7 | 2024-09-07 |
|             2 |         8 |          8 |             8 | 2024-09-07 |
|             3 |         9 |          9 |            10 | 2024-09-08 |
|             3 |        10 |         10 |             9 | 2024-09-08 |
+---------------+-----------+------------+---------------+------------+
10 rows in set (0.00 sec)
```

Рисунок 1.26 – Данные таблицы polyclinic

Вывод: в ходе практической работы научились создавать базу данных, создавать таблицы и заполнять их. Также в ходе практической были изучены INSERT INTO, SELECT * FROM и т.д.

Практическая работа № 2

Постановка задачи: изучить и создать выборку и сортировку данных. Изучить и применить операторы для изменения данных в таблицах.

Чтобы выбрать данные из таблицы используется команда select ‘что требуется’ from ‘таблицы’.

```
mysql> select * from equipment;
+--------------+---------------+------------+------------+
| id_equipment | serial_number | start_dt   | end_dt     |
+--------------+---------------+------------+------------+
|            1 | AA/1542TW14   | 2021-03-01 | 2026-03-01 |
|            2 | AA/1532SU71   | 2024-05-13 | 2025-11-13 |
|            3 | AA/1567TT04   | 2024-03-16 | 2024-12-16 |
|            4 | AA/1442DR14   | 2023-09-01 | 2027-09-01 |
|            5 | AA/1940SE10   | 2024-08-15 | 2025-02-14 |
|            6 | AA/1940DE39   | 2023-09-27 | 2028-09-27 |
|            7 | AA/1940VV14   | 2022-11-14 | 2024-11-14 |
|            8 | AA/1355DL14   | 2024-09-22 | 2025-05-17 |
|            9 | AA/1009RT13   | 2023-07-03 | 2025-07-03 |
|           10 | AA/1335DC61   | 2024-05-12 | 2024-11-12 |
+--------------+---------------+------------+------------+
10 rows in set (0.01 sec)
```

Рисунок 2.1 – Выбор всех элементов из таблицы

```
mysql> select start_dt from equipment;
+------------+
| start_dt   |
+------------+
| 2021-03-01 |
| 2024-05-13 |
| 2024-03-16 |
| 2023-09-01 |
| 2024-08-15 |
| 2023-09-27 |
| 2022-11-14 |
| 2024-09-22 |
| 2023-07-03 |
| 2024-05-12 |
+------------+
10 rows in set (0.00 sec)
```

Рисунок 2.2 – Вывод элементов одного столбца

```
mysql> select name_p, phone_p from patient;
+----------+-------------+
| name_p   | phone_p     |
+----------+-------------+
| Vladimir | 79269997003 |
| Ilya     | 79269036543 |
| Arina    | 79267881020 |
| Julia    | 79261002222 |
| Vera     | 79261055837 |
| Naimi    | 79241119045 |
| Micle    | 79139974637 |
| Nikita   | 79131110003 |
| Anna     | 79269910667 |
| Vasya    | 79267780963 |
+----------+-------------+
10 rows in set (0.00 sec)
```

Рисунок 2.3 – Вывод элементов двух столбцов

Чтобы сортировать выборку команда дополняется order by ‘название столбца’ (desc).

```
mysql> select * from analyzes order by name_analyzes;
+-------------+----------------+----------------------------------------+
| id_analyzes | name_analyzes  | result                                 |
+-------------+----------------+----------------------------------------+
|           1 | blood test     | Hemoglobin and platelets are normal    |
|           7 | blood test     | Hemoglobin and platelets are NO normal |
|           3 | ECG            | The heart rate is calm                 |
|           9 | ECG            | Heart rate is above normal             |
|           2 | fluorography   | The chest is normal                    |
|           4 | sugar analysis | The sugar level is normal              |
|           6 | sugar analysis | The sugar level is above normal        |
|           8 | sugar analysis | High sugar levels                      |
|           5 | urine analysis | The color is normal, there is no smell |
|          10 | urine analysis | Kidney problems                        |
+-------------+----------------+----------------------------------------+
10 rows in set (0.01 sec)
```

Рисунок 2.4 – Выбор значений с сортировкой по выбранному столбцу

```
mysql> select * from analyzes order by name_analyzes desc;
+-------------+----------------+----------------------------------------+
| id_analyzes | name_analyzes  | result                                 |
+-------------+----------------+----------------------------------------+
|           5 | urine analysis | The color is normal, there is no smell |
|          10 | urine analysis | Kidney problems                        |
|           4 | sugar analysis | The sugar level is normal              |
|           6 | sugar analysis | The sugar level is above normal        |
|           8 | sugar analysis | High sugar levels                      |
|           2 | fluorography   | The chest is normal                    |
|           3 | ECG            | The heart rate is calm                 |
|           9 | ECG            | Heart rate is above normal             |
|           1 | blood test     | Hemoglobin and platelets are normal    |
|           7 | blood test     | Hemoglobin and platelets are NO normal |
+-------------+----------------+----------------------------------------+
10 rows in set (0.00 sec)
```

Рисунок 2.5 – Выбор значений с сортировкой по убыванию

```
mysql> select * from analyzes order by name_analyzes desc, id_analyzes desc;
+-------------+---------------+--------------------------------------+
| id_analyzes | name_analyzes | result                               |
+-------------+---------------+--------------------------------------+
|          10 | urine analysis | Kidney problems                      |
|           5 | urine analysis | The color is normal, there is no smell |
|           8 | sugar analysis | High sugar levels                    |
|           6 | sugar analysis | The sugar level is above normal      |
|           4 | sugar analysis | The sugar level is normal            |
|           2 | fluorography   | The chest is normal                  |
|           9 | ECG            | Heart rate is above normal           |
|           3 | ECG            | The heart rate is calm               |
|           7 | blood test     | Hemoglobin and platelets are NO normal |
|           1 | blood test     | Hemoglobin and platelets are normal  |
+-------------+---------------+--------------------------------------+
10 rows in set (0.00 sec)
```

Рисунок 2.6 – Выбор значений с сортировкой по убыванию по двум столбцам

Также можно добавить условии для выбора с помощью where

```
mysql> select * from patient where id_patient=3;
+------------+--------+-----------+--------------+-------------+
| id_patient | name_p | surname_p | patronymic_p | phone_p     |
+------------+--------+-----------+--------------+-------------+
|          3 | Arina  | Smertina  | Vladimirovna | 79267881020 |
+------------+--------+-----------+--------------+-------------+
1 row in set (0.00 sec)
```

Рисунок 2.7 – Вывод элементов с id равным 3

```
mysql> select * from patient where name_p='Anna';
+------------+--------+-----------+--------------+-------------+
| id_patient | name_p | surname_p | patronymic_p | phone_p     |
+------------+--------+-----------+--------------+-------------+
|          9 | Anna   | Sharova   | NULL         | 79269910667 |
+------------+--------+-----------+--------------+-------------+
1 row in set (0.00 sec)
```

Рисунок 2.8 – Вывод элементов с именем «Anna»

```
mysql> select * from patient where id_patient>5;
+------------+--------+-----------+--------------+-------------+
| id_patient | name_p | surname_p | patronymic_p | phone_p     |
+------------+--------+-----------+--------------+-------------+
|          6 | Naimi  | Farid     | NULL         | 79241119045 |
|          7 | Micle  | Batkin    | NULL         | 79139974637 |
|          8 | Nikita | Grigoriev | NULL         | 79131110003 |
|          9 | Anna   | Sharova   | NULL         | 79269910667 |
|         10 | Vasya  | Naymov    | NULL         | 79267780963 |
+------------+--------+-----------+--------------+-------------+
5 rows in set (0.01 sec)
```

Рисунок 2.9 – Вывод элементов с id больше 5

```
mysql> select * from patient where id_patient<5;
+------------+----------+-----------+--------------+-------------+
| id_patient | name_p   | surname_p | patronymic_p | phone_p     |
+------------+----------+-----------+--------------+-------------+
|          1 | Vladimir | Semenov   | Ilyich       | 79269997003 |
|          2 | Ilya     | Petrov    | Arturovich   | 79269036543 |
|          3 | Arina    | Smertina  | Vladimirovna | 79267881020 |
|          4 | Julia    | Grin      | Andreewna    | 79261002222 |
+------------+----------+-----------+--------------+-------------+
4 rows in set (0.00 sec)
```

Рисунок 2.10 – Вывод элементов с id меньше 5

```
mysql> select * from patient where id_patient>=6;
+------------+--------+-----------+--------------+-------------+
| id_patient | name_p | surname_p | patronymic_p | phone_p     |
+------------+--------+-----------+--------------+-------------+
|          6 | Naimi  | Farid     | NULL         | 79241119045 |
|          7 | Micle  | Batkin    | NULL         | 79139974637 |
|          8 | Nikita | Grigoriev | NULL         | 79131110003 |
|          9 | Anna   | Sharova   | NULL         | 79269910667 |
|         10 | Vasya  | Naymov    | NULL         | 79267780963 |
+------------+--------+-----------+--------------+-------------+
5 rows in set (0.00 sec)
```

Рисунок 2.11 – Вывод элементов с id больше или равно 6

```
mysql> select * from patient where id_patient<=6;
+------------+----------+-----------+--------------+-------------+
| id_patient | name_p   | surname_p | patronymic_p | phone_p     |
+------------+----------+-----------+--------------+-------------+
|          1 | Vladimir | Semenov   | Ilyich       | 79269997003 |
|          2 | Ilya     | Petrov    | Arturovich   | 79269036543 |
|          3 | Arina    | Smertina  | Vladimirovna | 79267881020 |
|          4 | Julia    | Grin      | Andreewna    | 79261002222 |
|          5 | Vera     | Vasileva  | Petrovna     | 79261055837 |
|          6 | Naimi    | Farid     | NULL         | 79241119045 |
+------------+----------+-----------+--------------+-------------+
6 rows in set (0.00 sec)
```

Рисунок 2.12 – Вывод элементов с id меньше или равно 6

```
mysql> select * from analyzes where name_analyzes!='sugar analysis';
+-------------+----------------+-----------------------------------------+
| id_analyzes | name_analyzes  | result                                  |
+-------------+----------------+-----------------------------------------+
|           1 | blood test     | Hemoglobin and platelets are normal     |
|           2 | fluorography   | The chest is normal                     |
|           3 | ECG            | The heart rate is calm                  |
|           5 | urine analysis | The color is normal, there is no smell  |
|           7 | blood test     | Hemoglobin and platelets are NO normal  |
|           9 | ECG            | Heart rate is above normal              |
|          10 | urine analysis | Kidney problems                         |
+-------------+----------------+-----------------------------------------+
7 rows in set (0.00 sec)
```

Рисунок 2.13 – Вывод элементов без названия «sugar analysis»

```
mysql> select * from doctor where patronymic_d is not null
    -> ;
+-----------+---------+-----------+---------------+----------------+-------------+
| id_doctor | name_d  | surname_d | patronymic_d  | specialization | phone_d     |
+-----------+---------+-----------+---------------+----------------+-------------+
|         1 | Georgy  | Stepanov  | Ilyich        | therapist      | 79269621042 |
|         2 | Marina  | Dmitrova  | Petrovna      | therapist      | 79031338066 |
|         3 | Vladimir| Andreew   | Ilyich        | psychologist   | 79991001313 |
|         4 | Vasya   | Nikitin   | Vladimirovich | narcologist    | 78031131567 |
|         5 | Petya   | Kostin    | Alexeyevich   | therapist      | 79261118004 |
|         6 | Michael | Pechin    | Dmitrievich   | surgeon        | 79813041233 |
+-----------+---------+-----------+---------------+----------------+-------------+
6 rows in set (0.00 sec)
```

Рисунок 2.14 – Вывод элементов с not null

```
mysql> select * from doctor where patronymic_d is null;
+-----------+-----------+-----------+--------------+----------------+-------------+
| id_doctor | name_d    | surname_d | patronymic_d | specialization | phone_d     |
+-----------+-----------+-----------+--------------+----------------+-------------+
|         7 | Alexander | Reminen   | NULL         | venerologist   | 78991024435 |
|         8 | Michael   | Stechin   | NULL         | dermatologist  | 79041024489 |
|         9 | Arkady    | Pilonov   | NULL         | surgeon        | 79001003447 |
|        10 | Kostya    | Sidorov   | NULL         | narcologist    | 79811055670 |
+-----------+-----------+-----------+--------------+----------------+-------------+
4 rows in set (0.00 sec)
```

Рисунок 2.15 – Вывод элементов с null

```
mysql> select * from doctor where id_doctor between 2 and 6;
+-----------+---------+-----------+---------------+----------------+-------------+
| id_doctor | name_d  | surname_d | patronymic_d  | specialization | phone_d     |
+-----------+---------+-----------+---------------+----------------+-------------+
|         2 | Marina  | Dmitrova  | Petrovna      | therapist      | 79031338066 |
|         3 | Vladimir| Andreew   | Ilyich        | psychologist   | 79991001313 |
|         4 | Vasya   | Nikitin   | Vladimirovich | narcologist    | 78031131567 |
|         5 | Petya   | Kostin    | Alexeyevich   | therapist      | 79261118004 |
|         6 | Michael | Pechin    | Dmitrievich   | surgeon        | 79813041233 |
+-----------+---------+-----------+---------------+----------------+-------------+
5 rows in set (0.00 sec)
```

Рисунок 2.16 – Вывод элементов с помощью BETWEEN

```
mysql> select * from doctor where id_doctor in (1, 3, 6);
+-----------+---------+-----------+---------------+----------------+-------------+
| id_doctor | name_d  | surname_d | patronymic_d  | specialization | phone_d     |
+-----------+---------+-----------+---------------+----------------+-------------+
|         1 | Georgy  | Stepanov  | Ilyich        | therapist      | 79269621042 |
|         3 | Vladimir| Andreew   | Ilyich        | psychologist   | 79991001313 |
|         6 | Michael | Pechin    | Dmitrievich   | surgeon        | 79813041233 |
+-----------+---------+-----------+---------------+----------------+-------------+
3 rows in set (0.00 sec)
```

Рисунок 2.17 – Вывод элементов с помощью in

```
mysql> select * from doctor where id_doctor not in (1, 3, 6);
+-----------+-----------+-----------+---------------+----------------+-------------+
| id_doctor | name_d    | surname_d | patronymic_d  | specialization | phone_d     |
+-----------+-----------+-----------+---------------+----------------+-------------+
|         2 | Marina    | Dmitrova  | Petrovna      | therapist      | 79031338066 |
|         4 | Vasya     | Nikitin   | Vladimirovich | narcologist    | 78031131567 |
|         5 | Petya     | Kostin    | Alexeyevich   | therapist      | 79261118004 |
|         7 | Alexander | Reminen   | NULL          | venerologist   | 78991024435 |
|         8 | Michael   | Stechin   | NULL          | dermatologist  | 79041024489 |
|         9 | Arkady    | Pilonov   | NULL          | surgeon        | 79001003447 |
|        10 | Kostya    | Sidorov   | NULL          | narcologist    | 79811055670 |
+-----------+-----------+-----------+---------------+----------------+-------------+
7 rows in set (0.00 sec)
```

Рисунок 2.18 – Вывод элементов с помощью not in

В качестве условия можно использовать ключевое слово like позволяющий делать выбор на основе шаблона.

```
mysql> select * from analyzes where name_analyzes like '%lysis';
+-------------+----------------+-----------------------------------------+
| id_analyzes | name_analyzes  | result                                  |
+-------------+----------------+-----------------------------------------+
|           4 | sugar analysis | The sugar level is normal               |
|           5 | urine analysis | The color is normal, there is no smell  |
|           6 | sugar analysis | The sugar level is above normal         |
|           8 | sugar analysis | High sugar levels                       |
|          10 | urine analysis | Kidney problems                         |
+-------------+----------------+-----------------------------------------+
5 rows in set (0.00 sec)
```

Рисунок 2.19 – Выбор по условию «like»

```
mysql> select * from analyzes where name_analyzes not like 'blo%';
+-------------+----------------+-----------------------------------------+
| id_analyzes | name_analyzes  | result                                  |
+-------------+----------------+-----------------------------------------+
|           2 | fluorography   | The chest is normal                     |
|           3 | ECG            | The heart rate is calm                  |
|           4 | sugar analysis | The sugar level is normal               |
|           5 | urine analysis | The color is normal, there is no smell  |
|           6 | sugar analysis | The sugar level is above normal         |
|           8 | sugar analysis | High sugar levels                       |
|           9 | ECG            | Heart rate is above normal              |
|          10 | urine analysis | Kidney problems                         |
+-------------+----------------+-----------------------------------------+
8 rows in set (0.00 sec)
```

Рисунок 2.20 – Выбор по условию не равному «like»

С помощью alter table ‘название таблицы’ ‘ключевое слово’ можно изменять существующею таблицу. Add column ‘название столбца’ ‘тип данных’ позволяет создавать столбец в таблице.

```
mysql> alter table work add column metro_station varchar(20);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show columns from work;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| id_polyclinic | int         | NO   | PRI | NULL    | auto_increment |
| address       | varchar(30) | NO   |     | NULL    |                |
| metro_station | varchar(20) | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

Рисунок 2.21 – Добавление столбца в таблицу

Drop column ‘название столбца’ позволяет удалить столбец

```
mysql> alter table work drop column metro_station;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show columns from work;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| id_polyclinic | int         | NO   | PRI | NULL    | auto_increment |
| address       | varchar(30) | NO   |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

Рисунок 2.22 – Удаление столбца из таблицы

Update ‘название таблицы’ set ‘название столбца’= ‘данные’ меняет данные у всего столбца. Так же можно добавить условие устанавливать только требуемые строки.

```
mysql> select * from doctor where id_doctor=9;
+-----------+--------+-----------+--------------+----------------+-------------+
| id_doctor | name_d | surname_d | patronymic_d | specialization | phone_d     |
+-----------+--------+-----------+--------------+----------------+-------------+
|         9 | Arkady | Pilonov   | NULL         | surgeon        | 79001003447 |
+-----------+--------+-----------+--------------+----------------+-------------+
1 row in set (0.00 sec)

mysql> update doctor set patronymic_d='Alexeyevich' where id_doctor=9;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from doctor where id_doctor=9;
+-----------+--------+-----------+--------------+----------------+-------------+
| id_doctor | name_d | surname_d | patronymic_d | specialization | phone_d     |
+-----------+--------+-----------+--------------+----------------+-------------+
|         9 | Arkady | Pilonov   | Alexeyevich  | surgeon        | 79001003447 |
+-----------+--------+-----------+--------------+----------------+-------------+
1 row in set (0.00 sec)
```

Рисунок 2.23 – Изменение данных таблицы по условию

Change ‘название старого столбца’ ‘название нового столбца’ ‘тип данных’ позволяет изменить уже существующий столбец

```
mysql> show columns from doctor;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| id_doctor      | int         | NO   | PRI | NULL    | auto_increment |
| name_d         | varchar(20) | NO   |     | NULL    |                |
| surname_d      | varchar(20) | NO   |     | NULL    |                |
| patronymic_d   | varchar(20) | YES  |     | NULL    |                |
| specialization | varchar(20) | NO   |     | NULL    |                |
| phone_d        | varchar(11) | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

mysql> alter table doctor change specialization profession varchar(20) not null;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show columns from doctor;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| id_doctor    | int         | NO   | PRI | NULL    | auto_increment |
| name_d       | varchar(20) | NO   |     | NULL    |                |
| surname_d    | varchar(20) | NO   |     | NULL    |                |
| patronymic_d | varchar(20) | YES  |     | NULL    |                |
| profession   | varchar(20) | NO   |     | NULL    |                |
| phone_d      | varchar(11) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Рисунок 2.24 – Изменение названия столбца

Delete from ‘название таблицы’ позволяет удалить строки из таблицы. Так же можно добавить условие удалять только требуемые строки.

```
mysql> insert into equipment(serial_number, start_dt, end_dt) values('AA/1004RT19', '2023-01-19', '2024-01-19');
Query OK, 1 row affected (0.01 sec)

mysql> select * from equipment;
+--------------+---------------+------------+------------+
| id_equipment | serial_number | start_dt   | end_dt     |
+--------------+---------------+------------+------------+
|            1 | AA/1542TW14   | 2021-03-01 | 2026-03-01 |
|            2 | AA/1532SU71   | 2024-05-13 | 2025-11-13 |
|            3 | AA/1567TT04   | 2024-03-16 | 2024-12-16 |
|            4 | AA/1442DR14   | 2023-09-01 | 2027-09-01 |
|            5 | AA/1940SE10   | 2024-08-15 | 2025-02-14 |
|            6 | AA/1940DE39   | 2023-09-27 | 2028-09-27 |
|            7 | AA/1940VV14   | 2022-11-14 | 2024-11-14 |
|            8 | AA/1355DL14   | 2024-09-22 | 2025-05-17 |
|            9 | AA/1009RT13   | 2023-07-03 | 2025-07-03 |
|           10 | AA/1335DC61   | 2024-05-12 | 2024-11-12 |
|           11 | AA/1004RT19   | 2023-01-19 | 2024-01-19 |
+--------------+---------------+------------+------------+
11 rows in set (0.00 sec)

mysql> delete from equipment where id_equipment=11;
Query OK, 1 row affected (0.01 sec)

mysql> select * from equipment;
+--------------+---------------+------------+------------+
| id_equipment | serial_number | start_dt   | end_dt     |
+--------------+---------------+------------+------------+
|            1 | AA/1542TW14   | 2021-03-01 | 2026-03-01 |
|            2 | AA/1532SU71   | 2024-05-13 | 2025-11-13 |
|            3 | AA/1567TT04   | 2024-03-16 | 2024-12-16 |
|            4 | AA/1442DR14   | 2023-09-01 | 2027-09-01 |
|            5 | AA/1940SE10   | 2024-08-15 | 2025-02-14 |
|            6 | AA/1940DE39   | 2023-09-27 | 2028-09-27 |
|            7 | AA/1940VV14   | 2022-11-14 | 2024-11-14 |
|            8 | AA/1355DL14   | 2024-09-22 | 2025-05-17 |
|            9 | AA/1009RT13   | 2023-07-03 | 2025-07-03 |
|           10 | AA/1335DC61   | 2024-05-12 | 2024-11-12 |
+--------------+---------------+------------+------------+
10 rows in set (0.00 sec)
```

Рисунок 2.25 – Удаления строк по условию

```
mysql> delete from analyzes where id_analyzes=5;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`polyclinic`.`laboratory`, CONSTRAINT `laboratory_ibfk_1` FOREIGN KEY (`id_analyzes`) REFERENCES `analyzes` (`id_analyzes`))
```

Рисунок 2.26 – Попытка удаления данных из таблицы с вязанных с другой

Вывод: в ходе практической работы были изучены функции удаления данных из таблицы, обновление данных, изменение названия столбца в таблице, добавление новых столбцов и выборка по условию.

Практическая работа № 3

Постановка задачи:

1. Научится формировать модель БД с помощью инструментов СУБД (MySQL Workbench, dbForge Studio, PostgreSQL – по выбору студента) по своей теме.
2. Научится осуществлять перенос своей БД на другой сервер.
3. Изучить команды модификации данных (DML)
4. Осуществить выборку данных по своей теме с помощью различных операторов.
5. Изучить и применить к своей БД хранимые процедуры, функции и триггеры.

Ход работы:

В Workbench была создана физическая модель базы данных.

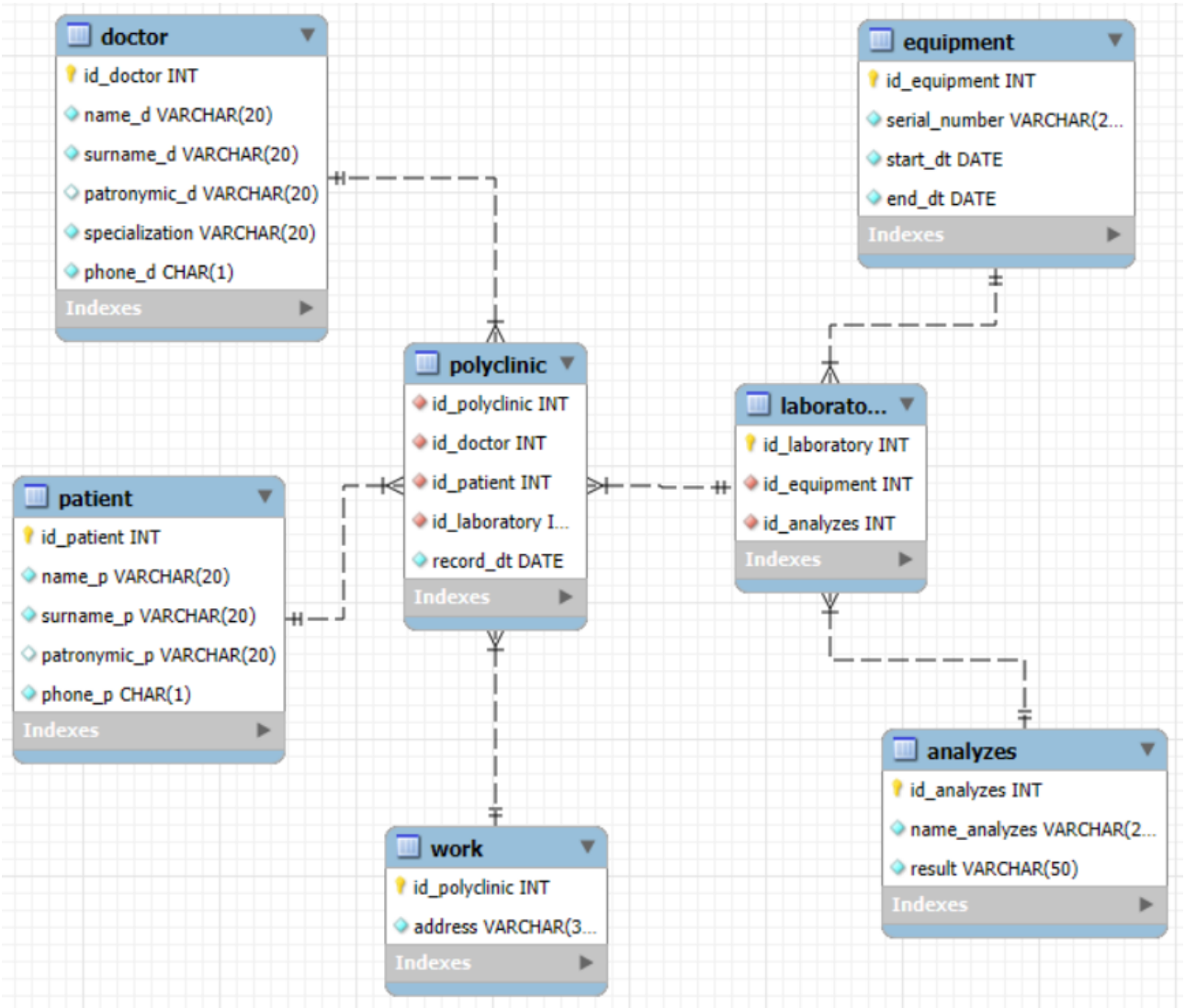

Рисунок. Физическая модель базы данных

Чтобы выбрать данные из 2 таблиц используется команда join ‘название таблицы’ on ‘по каким столбцам объединение’.

Вывод данных с двух таблиц при помощи join и с условием where.

```
mysql> select * from analyzes join laboratory on analyzes.id_analyzes=laboratory.id_analyzes where analyzes.id_analyzes>=5;
+-------------+----------------+---------------------------------------+---------------+--------------+-------------+
| id_analyzes | name_analyzes  | result                                | id_laboratory | id_equipment | id_analyzes |
+-------------+----------------+---------------------------------------+---------------+--------------+-------------+
|           5 | urine analysis | The color is normal, there is no smell |             5 |            5 |           5 |
|           6 | sugar analysis | The sugar level is above normal       |             6 |            2 |           6 |
|           7 | blood test     | Hemoglobin and platelets are NO normal |             7 |            6 |           7 |
|           8 | sugar analysis | High sugar levels                     |             8 |            7 |           8 |
|           9 | ECG            | Heart rate is above normal            |             9 |            4 |           9 |
|          10 | urine analysis | Kidney problems                       |            10 |            9 |          10 |
+-------------+----------------+---------------------------------------+---------------+--------------+-------------+
6 rows in set (0.00 sec)
```

Рисунок 3.1 – Вывод данных при помощи join и с условием

На рисунке 3.2 представлен вывод данных при помощи декартова произведения

```
mysql> select * from patient, polyclinic where patient.id_patient<4 and patient.id_patient=polyclinic.id_patient;
+------------+----------+-----------+--------------+-------------+---------------+-----------+------------+---------------+------------+
| id_patient | name_p   | surname_p | patronymic_p | phone_p     | id_polyclinic | id_doctor | id_patient | id_laboratory | record_dt  |
+------------+----------+-----------+--------------+-------------+---------------+-----------+------------+---------------+------------+
|          1 | Vladimir | Semenov   | Ilyich       | 79269997003 |             1 |         1 |          1 |             1 | 2024-09-03 |
|          2 | Ilya     | Petrov    | Arturovich   | 79269036543 |             1 |         2 |          2 |             2 | 2024-09-03 |
|          3 | Arina    | Smertina  | Vladimirovna | 79267881020 |             2 |         3 |          3 |             3 | 2024-09-04 |
+------------+----------+-----------+--------------+-------------+---------------+-----------+------------+---------------+------------+
3 rows in set (0.00 sec)
```

Рисунок 3.2 – Вывод данных при помощи декартового произведения

На рисунке 3.3 представлен вывод данных с логическим оператором или

```
mysql> select * from doctor join polyclinic on doctor.id_doctor=polyclinic.id_doctor where doctor.id_doctor<=2 or doctor.id_doctor>7;
+-----------+----------+-----------+--------------+--------------+-------------+---------------+-----------+------------+---------------+------------+
| id_doctor | name_d   | surname_d | patronymic_d | profession   | phone_d     | id_polyclinic | id_doctor | id_patient | id_laboratory | record_dt  |
+-----------+----------+-----------+--------------+--------------+-------------+---------------+-----------+------------+---------------+------------+
|         1 | Georgy   | Stepanov  | Ilyich       | therapist    | 79269621042 |             1 |         1 |          1 |             1 | 2024-09-03 |
|         2 | Marina   | Dmitrova  | Petrovna     | therapist    | 79031338066 |             1 |         2 |          2 |             2 | 2024-09-03 |
|         8 | Michael  | Stechin   | NULL         | dermatologist| 79041024489 |             2 |         8 |          8 |             8 | 2024-09-07 |
|         9 | Arkady   | Pilonov   | Alexeyevich  | surgeon      | 79001003447 |             3 |         9 |          9 |            10 | 2024-09-08 |
|        10 | Kostya   | Sidorov   | NULL         | narcologist  | 79811055670 |             3 |        10 |         10 |             9 | 2024-09-08 |
+-----------+----------+-----------+--------------+--------------+-------------+---------------+-----------+------------+---------------+------------+
5 rows in set (0.00 sec)
```

Рисунок 3.3 - Вывод данных с логическим оператором или

На рисунке 3.4 представлен вывод данных с использованием union

```
mysql> (select * from analyzes inner join laboratory on analyzes.id_analyzes=laboratory.id_analyzes where analyzes.id_analyzes=1) union (select * from analyzes inner
join laboratory on analyzes.id_analyzes=laboratory.id_analyzes where analyzes.id_analyzes=5);
+-------------+----------------+---------------------------------------+---------------+--------------+-------------+
| id_analyzes | name_analyzes  | result                                | id_laboratory | id_equipment | id_analyzes |
+-------------+----------------+---------------------------------------+---------------+--------------+-------------+
|           1 | blood test     | Hemoglobin and platelets are normal   |             1 |            1 |           1 |
|           5 | urine analysis | The color is normal, there is no smell |             5 |            5 |           5 |
+-------------+----------------+---------------------------------------+---------------+--------------+-------------+
2 rows in set (0.00 sec)
```

Рисунок 3.4 - Вывод данных с использованием union

На рисунке 3.5 представлен вывод данных при помощи exist

```
mysql> select * from polyclinic join work on polyclinic.id_polyclinic=work.id_polyclinic where work.id_polyclinic=1 and polyclinic.id_laboratory=2 and exist
s(select * from polyclinic as q where q.id_polyclinic=polyclinic.id_polyclinic and q.id_polyclinic=1 and q.id_doctor=1);
+---------------+-----------+------------+---------------+------------+---------------+-------------------------+
| id_polyclinic | id_doctor | id_patient | id_laboratory | record_dt  | id_polyclinic | address                 |
+---------------+-----------+------------+---------------+------------+---------------+-------------------------+
|             1 |         2 |          2 |             2 | 2024-09-03 |             1 | Yablochkova street, 3A  |
+---------------+-----------+------------+---------------+------------+---------------+-------------------------+
1 row in set (0.00 sec)
```

Рисунок 3.5 - Вывод данных при помощи exist

На рисунке 3.6 представлен вывод при помощи not exist

```
mysql> select * from polyclinic join doctor on polyclinic.id_doctor=doctor.id_doctor where polyclinic.id_polyclinic=1 and polyclinic.id_laboratory>2 and not exists(select * from polyclinic
as q where q.id_polyclinic=polyclinic.id_polyclinic and q.id_polyclinic=1 and q.id_doctor=3);
+---------------+-----------+------------+---------------+------------+-----------+---------+-----------+--------------+-------------+
| id_polyclinic | id_doctor | id_patient | id_laboratory | record_dt  | id_doctor | name_d  | surname_d | patronymic_d | profession  | phone_d     |
+---------------+-----------+------------+---------------+------------+-----------+---------+-----------+--------------+-------------+
|             1 |         4 |          5 |             5 | 2024-09-06 |         4 | Vasya   | Nikitin   | Vladimirovich| narcologist | 78031131567 |
+---------------+-----------+------------+---------------+------------+-----------+---------+-----------+--------------+-------------+
1 row in set (0.00 sec)
```

Рисунок 3.6 - Вывод при помощи not exist

На рисунке 3.7 и 3.8 представлены выводы данных при помощи count

```
mysql> select count(*) from doctor where id_doctor between 3 and 9;
+----------+
| count(*) |
+----------+
|        7 |
+----------+
1 row in set (0.00 sec)
```

Рисунок 3.7 - Выводы данных при помощи count

```
mysql> select name_p, count(*) as count from patient where id_patient=3 group by name_p;
+--------+-------+
| name_p | count |
+--------+-------+
| Arina  |     1 |
+--------+-------+
1 row in set (0.00 sec)
```

Рисунок 3.8 - Выводы данных при помощи count

На рисунке 3.9 представлен вывод с использованием сортировки

```
mysql> select polyclinic.id_polyclinic, count(*) as kolvo from polyclinic inner join work on work.id_polyclinic=polyclin
ic.id_polyclinic where work.id_polyclinic>=1 group by work.id_polyclinic order by kolvo;
+---------------+-------+
| id_polyclinic | kolvo |
+---------------+-------+
|             3 |     2 |
|             1 |     3 |
|             2 |     5 |
+---------------+-------+
3 rows in set (0.00 sec)
```

Рисунок 3.9 - Вывод с использованием сортировки

Хранимые процедуры

На рисунке 3.10 представлена первая хранимая процедура обновление поля patronymic_d для таблицы doctor на null.

Написание и применение хранимой процедуры по обновлению элемента.

```
1  CREATE DEFINER=`root`@`localhost` PROCEDURE `null_patronymicd`(id int)
2  BEGIN
3      update doctor
4      set patronymic_d = null
5      where id_doctor = id;
6  END
```

Рисунок 3.10 – Хранимая процедура обновления данных в поле patronymic_d

На рисунке 3.11 представлен запуск хранимой процедуры null_patronymicd.

```
mysql> select * from doctor;
+-----------+-----------+-----------+--------------+--------------+-------------+
| id_doctor | name_d    | surname_d | patronymic_d | profession   | phone_d     |
+-----------+-----------+-----------+--------------+--------------+-------------+
|         1 | Georgy    | Stepanov  | Ilyich       | therapist    | 79269621042 |
|         2 | Marina    | Dmitrova  | Petrovna     | therapist    | 79031338066 |
|         3 | Vladimir  | Andreew   | Ilyich       | psychologist | 79991001313 |
|         4 | Vasya     | Nikitin   | Vladimirovich| narcologist  | 78031131567 |
|         5 | Petya     | Kostin    | Alexeyevich  | therapist    | 79261118004 |
|         6 | Michael   | Pechin    | Dmitrievich  | surgeon      | 79813041233 |
|         7 | Alexander | Reminen   | NULL         | venerologist | 78991024435 |
|         8 | Michael   | Stechin   | NULL         | dermatologist| 79041024489 |
|         9 | Arkady    | Pilonov   | Alexeyevich  | surgeon      | 79001003447 |
|        10 | Kostya    | Sidorov   | NULL         | narcologist  | 79811055670 |
+-----------+-----------+-----------+--------------+--------------+-------------+
10 rows in set (0.00 sec)

mysql> call null_patronymicd(2);
Query OK, 1 row affected (0.01 sec)

mysql> select * from doctor;
+-----------+-----------+-----------+--------------+--------------+-------------+
| id_doctor | name_d    | surname_d | patronymic_d | profession   | phone_d     |
+-----------+-----------+-----------+--------------+--------------+-------------+
|         1 | Georgy    | Stepanov  | Ilyich       | therapist    | 79269621042 |
|         2 | Marina    | Dmitrova  | NULL         | therapist    | 79031338066 |
|         3 | Vladimir  | Andreew   | Ilyich       | psychologist | 79991001313 |
|         4 | Vasya     | Nikitin   | Vladimirovich| narcologist  | 78031131567 |
|         5 | Petya     | Kostin    | Alexeyevich  | therapist    | 79261118004 |
|         6 | Michael   | Pechin    | Dmitrievich  | surgeon      | 79813041233 |
|         7 | Alexander | Reminen   | NULL         | venerologist | 78991024435 |
|         8 | Michael   | Stechin   | NULL         | dermatologist| 79041024489 |
|         9 | Arkady    | Pilonov   | Alexeyevich  | surgeon      | 79001003447 |
|        10 | Kostya    | Sidorov   | NULL         | narcologist  | 79811055670 |
+-----------+-----------+-----------+--------------+--------------+-------------+
10 rows in set (0.00 sec)
```

Рисунок 3.11 – Работа хранимой процедуры null_patronymicd

На рисунке 3.12 представлена хранимая процедура обновления анализа.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `rename_analyzes`(id_a int, name_a varchar(20), res varchar(50))
BEGIN
if exists(select * from analyzes where id_analyzes=id_a) then
update analyzes
set name_analyzes = name_a, result = res
where id_analyzes=id_a;
else
insert into analyzes(name_analyzes, result) values(name_a, res);
END if;
END
```

Рисунок 3.12 – Хранимая процедура обновления анализа

На рисунке 3.13 представлен запуск хранимой процедуры rename_analyzes.

```
mysql> select * from analyzes;
+-------------+----------------+----------------------------------------+
| id_analyzes | name_analyzes  | result                                 |
+-------------+----------------+----------------------------------------+
|           1 | blood test     | Hemoglobin and platelets are normal    |
|           2 | fluorography   | The chest is normal                    |
|           3 | ECG            | The heart rate is calm                 |
|           4 | sugar analysis | The sugar level is normal              |
|           5 | urine analysis | The color is normal, there is no smell |
|           6 | sugar analysis | The sugar level is above normal        |
|           7 | blood test     | Hemoglobin and platelets are NO normal |
|           8 | sugar analysis | High sugar levels                      |
|           9 | ECG            | Heart rate is above normal             |
|          10 | urine analysis | Kidney problems                        |
+-------------+----------------+----------------------------------------+
10 rows in set (0.00 sec)

mysql> call rename_analyzes(4, 'ECG', 'normal');
Query OK, 1 row affected (0.00 sec)

mysql> select * from analyzes;
+-------------+----------------+----------------------------------------+
| id_analyzes | name_analyzes  | result                                 |
+-------------+----------------+----------------------------------------+
|           1 | blood test     | Hemoglobin and platelets are normal    |
|           2 | fluorography   | The chest is normal                    |
|           3 | ECG            | The heart rate is calm                 |
|           4 | ECG            | normal                                 |
|           5 | urine analysis | The color is normal, there is no smell |
|           6 | sugar analysis | The sugar level is above normal        |
|           7 | blood test     | Hemoglobin and platelets are NO normal |
|           8 | sugar analysis | High sugar levels                      |
|           9 | ECG            | Heart rate is above normal             |
|          10 | urine analysis | Kidney problems                        |
+-------------+----------------+----------------------------------------+
10 rows in set (0.00 sec)
```

Рисунок 3.13 – Работа хранимой процедуры rename_analyzes

На рисунке 3.14 представлена хранимая процедура по обновлению номера телефона пациента.

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `update_phone`(name varchar(20), surname varchar(20), phone varchar(11))
BEGIN
if exists(select * from patient where name_p=name and surname_p=surname) then
update patient
set phone_p = phone
where name_p=name and surname_p = surname;
else
insert into patient(name_p, surname_p, patronymic_p, phone_p) values(name, surname, null, phone);
END if;
END
```

Рисунок 3.14 – Хранимая процедура обновления номера телефона пациента

На рисунке 3.15 представлен запуск хранимой процедуры.

```
mysql> select * from patient;
+------------+----------+-----------+--------------+-------------+
| id_patient | name_p   | surname_p | patronymic_p | phone_p     |
+------------+----------+-----------+--------------+-------------+
|          1 | Vladimir | Semenov   | Ilyich       | 79269997003 |
|          2 | Ilya     | Petrov    | Arturovich   | 79269036543 |
|          3 | Arina    | Smertina  | Vladimirovna | 79267881020 |
|          4 | Julia    | Grin      | Andreewna    | 79261002222 |
|          5 | Vera     | Vasileva  | Petrovna     | 79261055837 |
|          6 | Naimi    | Farid     | NULL         | 79241119045 |
|          7 | Micle    | Batkin    | NULL         | 79139974637 |
|          8 | Nikita   | Grigoriev | NULL         | 79131110003 |
|          9 | Anna     | Sharova   | NULL         | 79269910667 |
|         10 | Vasya    | Naymov    | NULL         | 79267780963 |
+------------+----------+-----------+--------------+-------------+
10 rows in set (0.00 sec)

mysql> call update_phone('Arina', 'Smertina', '79990000000');
Query OK, 1 row affected (0.01 sec)

mysql> select * from patient;
+------------+----------+-----------+--------------+-------------+
| id_patient | name_p   | surname_p | patronymic_p | phone_p     |
+------------+----------+-----------+--------------+-------------+
|          1 | Vladimir | Semenov   | Ilyich       | 79269997003 |
|          2 | Ilya     | Petrov    | Arturovich   | 79269036543 |
|          3 | Arina    | Smertina  | Vladimirovna | 79990000000 |
|          4 | Julia    | Grin      | Andreewna    | 79261002222 |
|          5 | Vera     | Vasileva  | Petrovna     | 79261055837 |
|          6 | Naimi    | Farid     | NULL         | 79241119045 |
|          7 | Micle    | Batkin    | NULL         | 79139974637 |
|          8 | Nikita   | Grigoriev | NULL         | 79131110003 |
|          9 | Anna     | Sharova   | NULL         | 79269910667 |
|         10 | Vasya    | Naymov    | NULL         | 79267780963 |
+------------+----------+-----------+--------------+-------------+
10 rows in set (0.00 sec)
```

Рисунок 3.15 – Работа хранимой процедуры

Функции:

На рисунке 3.16 представлена функция подсчета количества докторов с заданным именем.

```
CREATE DEFINER=`root`@`localhost` FUNCTION `named_count`(name varchar(20)) RETURNS int
    DETERMINISTIC
BEGIN
declare d_count int;
select count(*) into d_count from doctor where name_d=name;
RETURN d_count;
END
```

Рисунок 3.16 – Функция подсчета количества докторов с заданным именем

На рисунке 3.17 представлена работа функции named_count.

```
mysql> select * from doctor;
+-----------+-----------+-----------+---------------+---------------+-------------+
| id_doctor | name_d    | surname_d | patronymic_d  | profession    | phone_d     |
+-----------+-----------+-----------+---------------+---------------+-------------+
|         1 | Georgy    | Stepanov  | Ilyich        | therapist     | 79269621042 |
|         2 | Marina    | Dmitrova  | NULL          | therapist     | 79031338066 |
|         3 | Vladimir  | Andreew   | Ilyich        | psychologist  | 79991001313 |
|         4 | Vasya     | Nikitin   | Vladimirovich | narcologist   | 78031131567 |
|         5 | Petya     | Kostin    | Alexeyevich   | therapist     | 79261118004 |
|         6 | Michael   | Pechin    | Dmitrievich   | surgeon       | 79813041233 |
|         7 | Alexander | Reminen   | NULL          | venerologist  | 78991024435 |
|         8 | Michael   | Stechin   | NULL          | dermatologist | 79041024489 |
|         9 | Arkady    | Pilonov   | Alexeyevich   | surgeon       | 79001003447 |
|        10 | Kostya    | Sidorov   | NULL          | narcologist   | 79811055670 |
+-----------+-----------+-----------+---------------+---------------+-------------+
10 rows in set (0.00 sec)

mysql> select named_count('Michael') as count_Michael;
+---------------+
| count_Michael |
+---------------+
|             2 |
+---------------+
1 row in set (0.00 sec)
```

Рисунок 3.17 – Работа функции named_count

На рисунке 3.18 представлена функция вывода списка анализов

```
CREATE DEFINER=`root`@`localhost` FUNCTION `analyzes_list`() RETURNS text
    DETERMINISTIC
BEGIN
declare list_a text;
select group_concat(distinct name_analyzes separator ',') into list_a from analyzes where name_analyzes is not null;
RETURN list_a;
END
```

Рисунок 3.18 – Функция вывода списка анализов

На рисунке 3.19 представлен запуск функции analyzes_list.

```
mysql> select * from analyzes;
+-------------+----------------+---------------------------------------+
| id_analyzes | name_analyzes  | result                                |
+-------------+----------------+---------------------------------------+
|           1 | blood test     | Hemoglobin and platelets are normal   |
|           2 | fluorography   | The chest is normal                   |
|           3 | ECG            | The heart rate is calm                |
|           4 | ECG            | normal                                |
|           5 | urine analysis | The color is normal, there is no smell |
|           6 | sugar analysis | The sugar level is above normal       |
|           7 | blood test     | Hemoglobin and platelets are NO normal |
|           8 | sugar analysis | High sugar levels                     |
|           9 | ECG            | Heart rate is above normal            |
|          10 | urine analysis | Kidney problems                       |
+-------------+----------------+---------------------------------------+
10 rows in set (0.00 sec)

mysql> select analyzes_list() as list_a;
+-----------------------------------------------------------+
| list_a                                                    |
+-----------------------------------------------------------+
| blood test,ECG,fluorography,sugar analysis,urine analysis |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

Рисунок 3.19 – Работа функции analyzes_list

На рисунке 3.20 представлена функция для обновления даты записи в конкретной поликлинике.

```
1  CREATE DEFINER=`root`@`localhost` FUNCTION `update_date`(id int, new_dt date) RETURNS int
2      DETERMINISTIC
3  BEGIN
4  declare rename_rec int;
5  update polyclinic join work on work.id_polyclinic=polyclinic.id_polyclinic
6  set polyclinic.record_dt = new_dt
7  where work.id_polyclinic=id;
8  set rename_rec = row_count();
9  return rename_rec;
10 END
```

Рисунок 3.20 – Функция обновления даты записи в конкретной поликлинике

На рисунке 3.21 представлен запуск функции update_date.

```
mysql> select * from work join polyclinic on work.id_polyclinic=polyclinic.id_polyclinic where work.id_polyclinic=2;
+---------------+-----------------+---------------+-----------+------------+---------------+------------+
| id_polyclinic | address         | id_polyclinic | id_doctor | id_patient | id_laboratory | record_dt  |
+---------------+-----------------+---------------+-----------+------------+---------------+------------+
|             2 | Glider street, 8 |             2 |         3 |          3 |             3 | 2024-09-04 |
|             2 | Glider street, 8 |             2 |         3 |          4 |             4 | 2024-09-05 |
|             2 | Glider street, 8 |             2 |         6 |          6 |             6 | 2024-09-07 |
|             2 | Glider street, 8 |             2 |         6 |          7 |             7 | 2024-09-07 |
|             2 | Glider street, 8 |             2 |         8 |          8 |             8 | 2024-09-07 |
+---------------+-----------------+---------------+-----------+------------+---------------+------------+
5 rows in set (0.00 sec)

mysql> select update_date(2, '2024-11-11') as count;
+-------+
| count |
+-------+
|     5 |
+-------+
1 row in set (0.01 sec)

mysql> select * from work join polyclinic on work.id_polyclinic=polyclinic.id_polyclinic where work.id_polyclinic=2;
+---------------+-----------------+---------------+-----------+------------+---------------+------------+
| id_polyclinic | address         | id_polyclinic | id_doctor | id_patient | id_laboratory | record_dt  |
+---------------+-----------------+---------------+-----------+------------+---------------+------------+
|             2 | Glider street, 8 |             2 |         3 |          3 |             3 | 2024-11-11 |
|             2 | Glider street, 8 |             2 |         3 |          4 |             4 | 2024-11-11 |
|             2 | Glider street, 8 |             2 |         6 |          6 |             6 | 2024-11-11 |
|             2 | Glider street, 8 |             2 |         6 |          7 |             7 | 2024-11-11 |
|             2 | Glider street, 8 |             2 |         8 |          8 |             8 | 2024-11-11 |
+---------------+-----------------+---------------+-----------+------------+---------------+------------+
5 rows in set (0.00 sec)
```

Рисунок 3.21 – Работа функции update_date

Триггеры:

На рисунке 3.22 представлен триггер, который добавляет данные в другую таблицу.

```
CREATE DEFINER=`root`@`localhost` TRIGGER `equipment_AFTER_INSERT` AFTER INSERT ON `equipment` FOR EACH ROW BEGIN
insert into equipment_log(log_id, serial_number_log, start_dt_log, end_dt_log, message_log)
values(new.id_equipment, new.serial_number, new.start_dt, new.end_dt, 'New equipment add');
END
```

Рисунок 3.22 – Триггер, который добавляет данные в другую таблицу

На рисунке 3.23 представлен запуск триггера.

```
mysql> insert into equipment(serial_number, start_dt, end_dt) values('AA/1000CN58', '2024-11-11', '2025-11-11');
Query OK, 1 row affected (0.01 sec)

mysql> select * from equipment_log;
+--------+-------------------+--------------+------------+-------------------+
| log_id | serial_number_log | start_dt_log | end_dt_log | message_log       |
+--------+-------------------+--------------+------------+-------------------+
|     14 | AA/1000CN58       | 2024-11-11   | 2025-11-11 | New equipment add |
+--------+-------------------+--------------+------------+-------------------+
1 row in set (0.00 sec)
```

Рисунок 3.23 – Работа триггера

На рисунке 3.24 представлен триггер, который выводит ошибку при попытке изменения имени или фамилии доктора на «OOOOO».

```
CREATE DEFINER=`root`@`localhost` TRIGGER `doctor_BEFORE_UPDATE` BEFORE UPDATE ON `doctor` FOR EACH ROW BEGIN
if new.surname_d='OOOOO' or new.name_d='OOOOO' then
signal sqlstate '45000'
set message_text='NO! (*_*)';
END if;
end
```

Рисунок 3.24 – Триггер, который выводит ошибку при попытке изменения имени или фамилии доктора на «OOOOO»

На рисунке 3.25 представлен запуск триггера.

```
mysql> update doctor set surname_d='OOOOO' where id_doctor=9;
ERROR 1644 (45000): NO! (*_*)
```

Рисунок 3.25 – Работа триггера

На рисунке 3.26 представлен триггер, который выводит ошибку при попытке удаления ЭКГ.

```
CREATE DEFINER=`root`@`localhost` TRIGGER `analyzes_BEFORE_DELETE` BEFORE DELETE ON `analyzes` FOR EACH ROW BEGIN
if old.name_analyzes='ECG' then
signal sqlstate '45000'
set message_text='Sorry, you cannot delete these analyzes';
END if;
end
```

Рисунок 3.26 – Триггер, который выводит ошибку при попытке удаления ЭКГ

На рисунке 3.27 представлен запуск триггера.

```
mysql> delete from analyzes where name_analyzes='ECG';
ERROR 1644 (45000): Sorry, you cannot delete these analyzes
```

Рисунок 3.27 – Работа триггера

Экспорт и импорт:

На рисунках 3.28 представлен импорт базы данных.

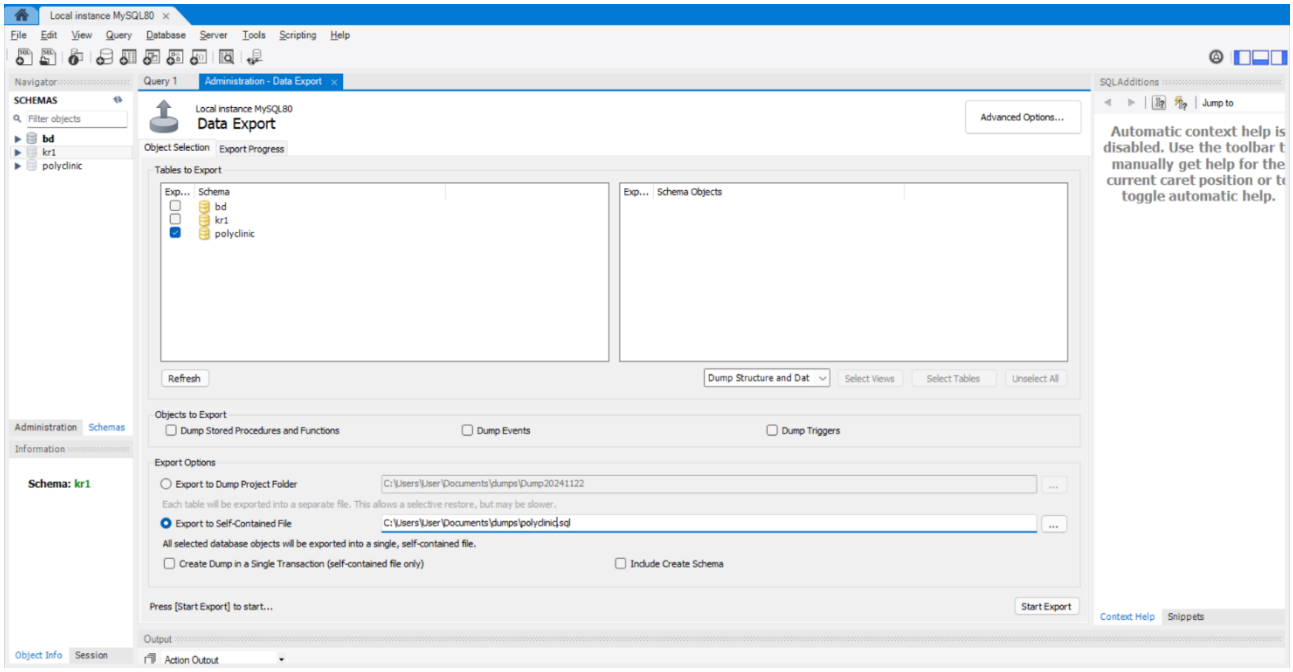

Рисунок 3.28 – Импорт базы данных

На рисунке 3.29 представлен запуск импорта базы данных.

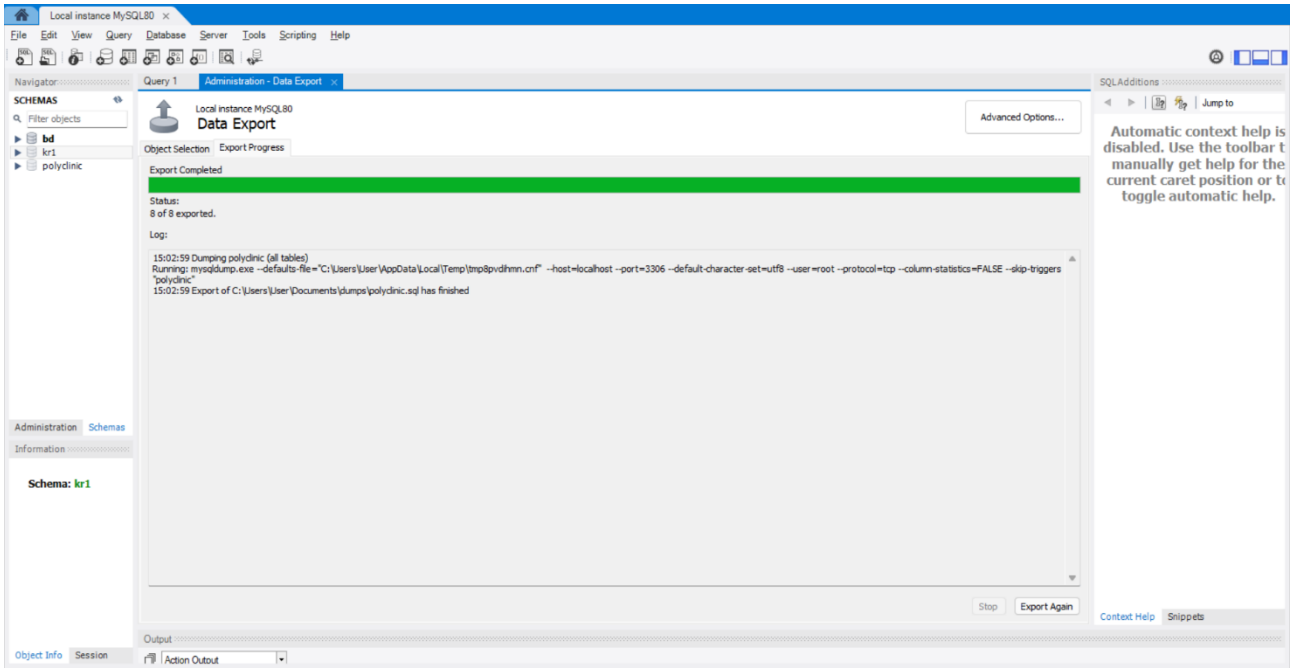

Рисунок 3.29 - Запуск импорта базы данных

На рисунке 3.30 создание новой базы данных для экспорта в нее до этого импортированной бд.

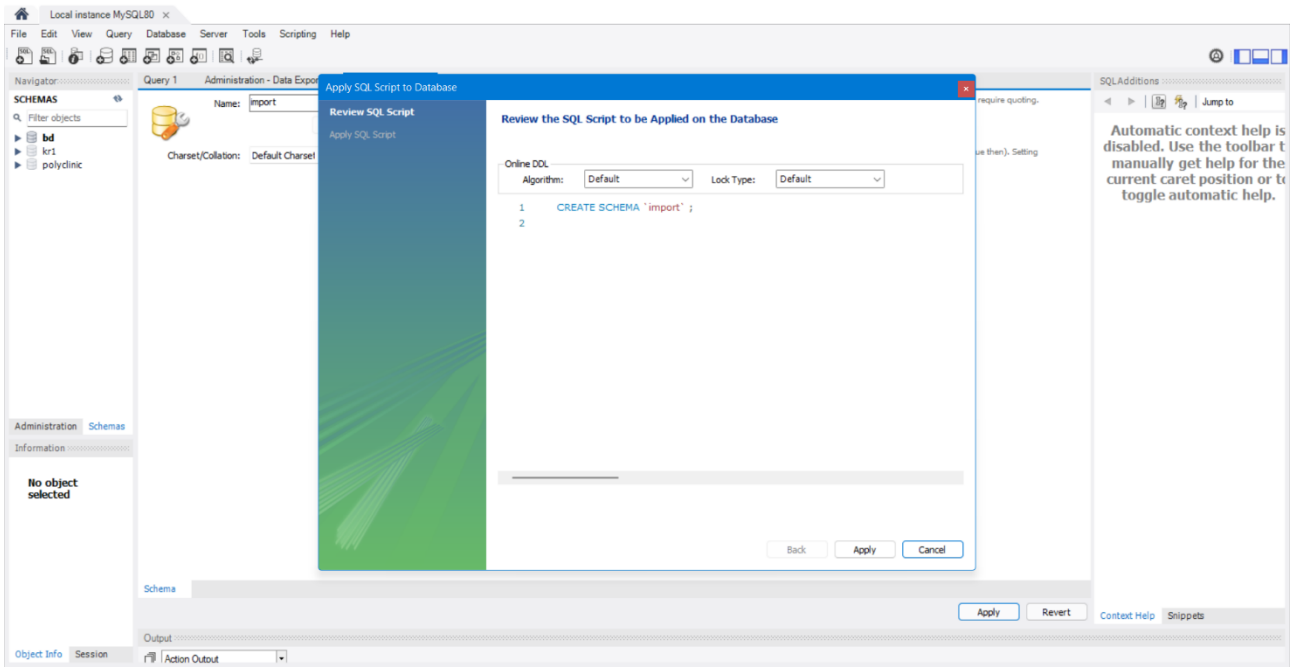

Рисунок 3.30 – Создание новой базы данных

На рисунке 3.31 показана созданная новая база данных.

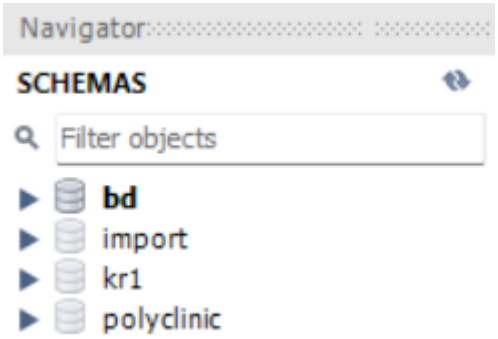

Рисунок 3.31 – Новая база данных

На рисунке 3.32 представлен экспорт базы данных.

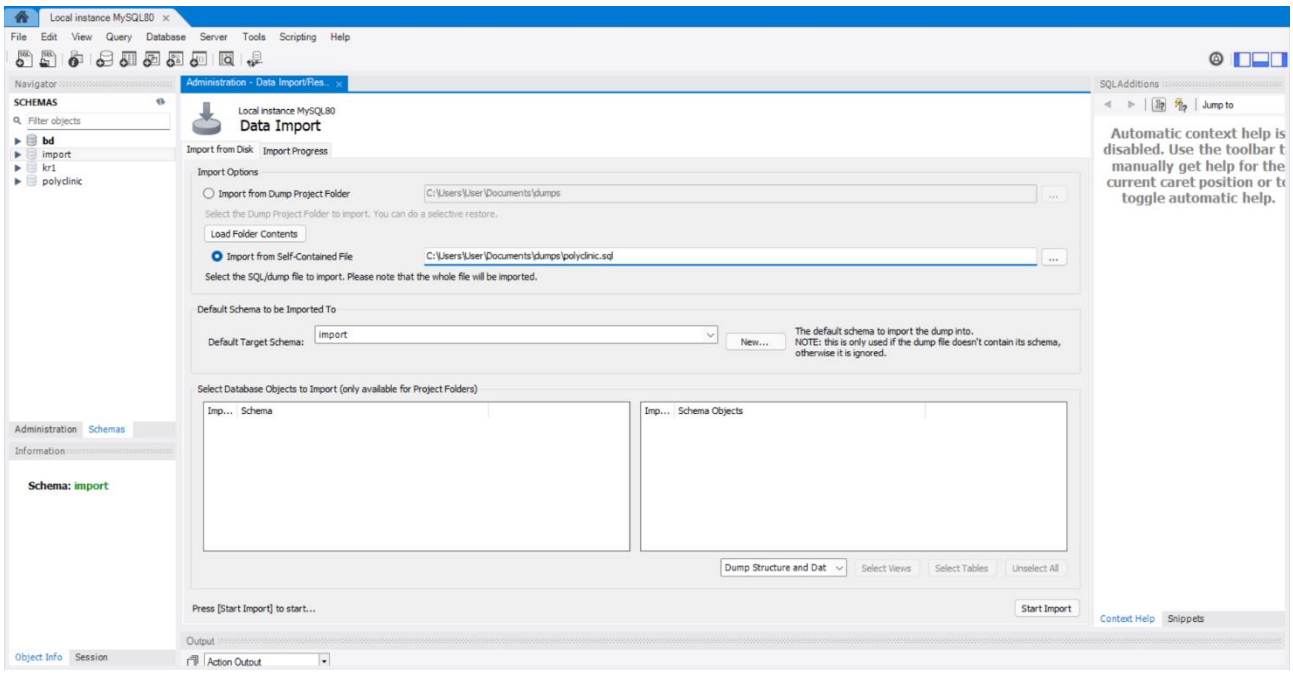

Рисунок 3.32 – Экспорт базы данных

На рисунке 3.33 представлен запуск экспорта базы данных.

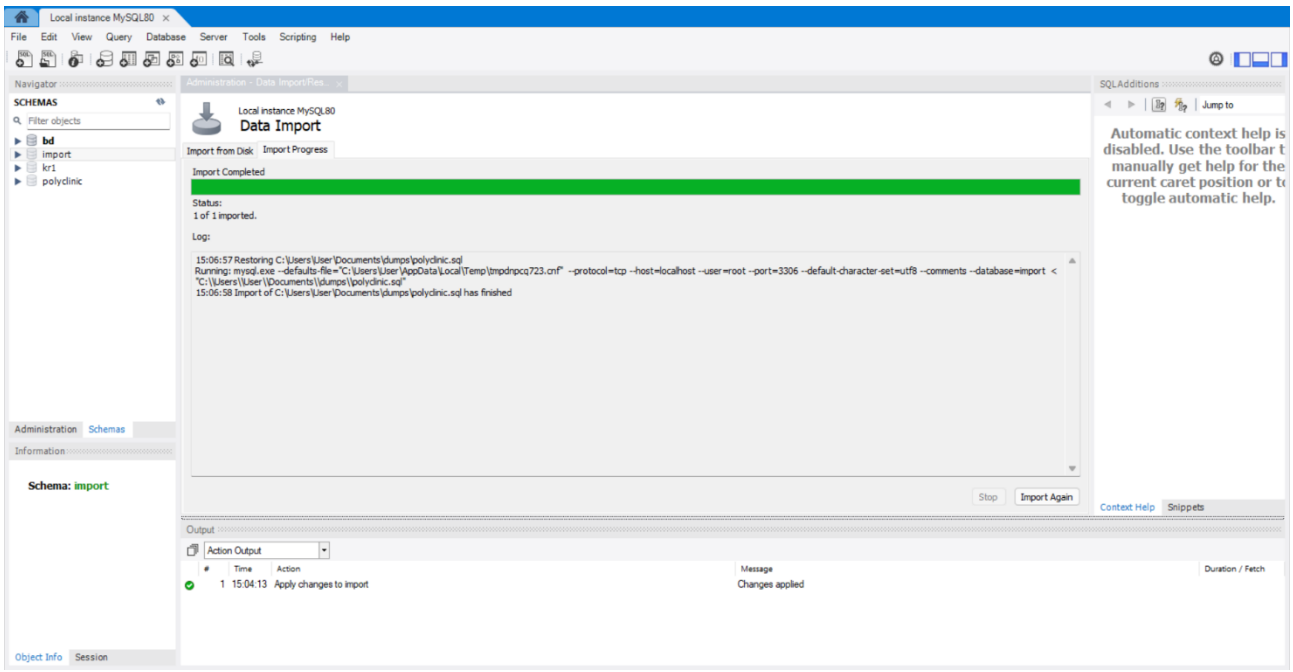

Рисунок 3.33 – Запуск экспорта базы данных

Вывод: в ходе практической работы были изучены и написаны функции, триггеры и хранимые процедуры также были изучены и написаны более сложные запросы к базе данных и научились импортировать и экспортировать базу данных.

Практическая работа 4

Постановка задачи: изучить оконные функции такие как агрегатные, ранжирующие и смещения.

Агрегатные функции:

На рисунке 4.1 представлена оконная функция count, которая выводит количество докторов с одинаковой профессией.

```
mysql> select polyclinic.id_polyclinic, address, surname_d, profession, count(*) over (partition by profession) as count
_prof from doctor join polyclinic on doctor.id_doctor=polyclinic.id_doctor join work on polyclinic.id_polyclinic=work.id
_polyclinic;
+---------------+------------------------+-----------+---------------+------------+
| id_polyclinic | address                | surname_d | profession    | count_prof |
+---------------+------------------------+-----------+---------------+------------+
|             2 | Glider street, 8       | Stechin   | dermatologist |          1 |
|             1 | Yablochkova street, 3A | Nikitin   | narcologist   |          2 |
|             3 | Kravchenko street, 14  | Sidorov   | narcologist   |          2 |
|             2 | Glider street, 8       | Andreew   | psychologist  |          2 |
|             2 | Glider street, 8       | Andreew   | psychologist  |          2 |
|             2 | Glider street, 8       | Pechin    | surgeon       |          3 |
|             2 | Glider street, 8       | Pechin    | surgeon       |          3 |
|             3 | Kravchenko street, 14  | Pilonov   | surgeon       |          3 |
|             1 | Yablochkova street, 3A | Dmitrova  | therapist     |          2 |
|             1 | Yablochkova street, 3A | Stepanov  | therapist     |          2 |
+---------------+------------------------+-----------+---------------+------------+
10 rows in set (0.00 sec)
```

Рисунок 4.1 – Оконная функция count

На рисунке 4.2 представлена оконная функция sum, которая выводит сумму за сдачу анализов.

```
mysql> select surname_p, name_analyzes, price, sum(price) over(partition by name_analyzes) as sum_price from analyzes join
patient where patient.id_patient=analyzes.id_analyzes;
+-----------+---------------+-------+-----------+
| surname_p | name_analyzes | price | sum_price |
+-----------+---------------+-------+-----------+
| Semenov   | blood test    |   280 |       480 |
| Batkin    | blood test    |   200 |       480 |
| Smertina  | ECG           |   600 |      1580 |
| Grin      | ECG           |   570 |      1580 |
| Sharova   | ECG           |   410 |      1580 |
| Petrov    | fluorography  |  1500 |      1500 |
| Farid     | sugar analysis|   410 |       710 |
| Grigoriev | sugar analysis|   300 |       710 |
| Vasileva  | urine analysis|   300 |       540 |
| Naymov    | urine analysis|   240 |       540 |
+-----------+---------------+-------+-----------+
10 rows in set (0.00 sec)
```

Рисунок 4.2 – Оконная функция sum

На рисунке 4.3 представлена оконная функция avg которая выводит среднюю цену за анализ.

```
mysql> select surname_p, name_analyzes, price, avg(price) over(partition by name_analyzes) as sum_price from analyzes join
patient where patient.id_patient=analyzes.id_analyzes;
+-----------+---------------+-------+-----------+
| surname_p | name_analyzes | price | sum_price |
+-----------+---------------+-------+-----------+
| Semenov   | blood test    |   280 |  240.0000 |
| Batkin    | blood test    |   200 |  240.0000 |
| Smertina  | ECG           |   600 |  526.6667 |
| Grin      | ECG           |   570 |  526.6667 |
| Sharova   | ECG           |   410 |  526.6667 |
| Petrov    | fluorography  |  1500 | 1500.0000 |
| Farid     | sugar analysis|   410 |  355.0000 |
| Grigoriev | sugar analysis|   300 |  355.0000 |
| Vasileva  | urine analysis|   300 |  270.0000 |
| Naymov    | urine analysis|   240 |  270.0000 |
+-----------+---------------+-------+-----------+
10 rows in set (0.00 sec)
```

Рисунок 4.3 – Оконная функция avg

На рисунке 4.4 представлена оконная функция min, которая выводит минимальную цену за анализ.

```
mysql> select surname_p, name_analyzes, price, min(price) over(partition by name_analyzes) as sum_price from analyzes join
patient where patient.id_patient=analyzes.id_analyzes;
+-----------+---------------+-------+-----------+
| surname_p | name_analyzes | price | sum_price |
+-----------+---------------+-------+-----------+
| Semenov   | blood test    |   280 |       200 |
| Batkin    | blood test    |   200 |       200 |
| Smertina  | ECG           |   600 |       410 |
| Grin      | ECG           |   570 |       410 |
| Sharova   | ECG           |   410 |       410 |
| Petrov    | fluorography  |  1500 |      1500 |
| Farid     | sugar analysis|   410 |       300 |
| Grigoriev | sugar analysis|   300 |       300 |
| Vasileva  | urine analysis|   300 |       240 |
| Naymov    | urine analysis|   240 |       240 |
+-----------+---------------+-------+-----------+
10 rows in set (0.00 sec)
```

Рисунок 4.4 – Оконная функция min

На рисунке 4.5 представлена оконная функция max, которая выводит максимальную цену за анализ.

```
mysql> select surname_p, name_analyzes, price, max(price) over(partition by name_analyzes) as sum_price from analyzes join
patient where patient.id_patient=analyzes.id_analyzes;
+-----------+---------------+-------+-----------+
| surname_p | name_analyzes | price | sum_price |
+-----------+---------------+-------+-----------+
| Semenov   | blood test    |   280 |       280 |
| Batkin    | blood test    |   200 |       280 |
| Smertina  | ECG           |   600 |       600 |
| Grin      | ECG           |   570 |       600 |
| Sharova   | ECG           |   410 |       600 |
| Petrov    | fluorography  |  1500 |      1500 |
| Farid     | sugar analysis|   410 |       410 |
| Grigoriev | sugar analysis|   300 |       410 |
| Vasileva  | urine analysis|   300 |       300 |
| Naymov    | urine analysis|   240 |       300 |
+-----------+---------------+-------+-----------+
10 rows in set (0.00 sec)
```

Рисунок 4.5 – Оконная функция max

Ранжирующие функции:

На рисунке 4.6 представлена оконная функция row_number, которая выводит индекс по списку, в котором они идут.

```
mysql> select surname_p, name_analyzes, price, row_number() over(partition by name_analyzes) as rownumber from analyzes join
patient where patient.id_patient=analyzes.id_analyzes;
+-----------+----------------+-------+-----------+
| surname_p | name_analyzes  | price | rownumber |
+-----------+----------------+-------+-----------+
| Semenov   | blood test     |   280 |         1 |
| Batkin    | blood test     |   200 |         2 |
| Smertina  | ECG            |   600 |         1 |
| Grin      | ECG            |   570 |         2 |
| Sharova   | ECG            |   410 |         3 |
| Petrov    | fluorography   |  1500 |         1 |
| Farid     | sugar analysis |   410 |         1 |
| Grigoriev | sugar analysis |   300 |         2 |
| Vasileva  | urine analysis |   300 |         1 |
| Naymov    | urine analysis |   240 |         2 |
+-----------+----------------+-------+-----------+
10 rows in set (0.00 sec)
```

Рисунок 4.6 – Оконная функция row_number

На рисунке 4.7 представлена оконная функция rank, которая выводит индекс по списку, в котором они идут. Если есть совпадения по индексу – то выводит одинаковый индекс с пропуском.

```
mysql> select surname_p, name_analyzes, price, rank() over(partition by name_analyzes order by price desc) as rang from
analyzes join patient where patient.id_patient=analyzes.id_analyzes;
+-----------+----------------+-------+------+
| surname_p | name_analyzes  | price | rang |
+-----------+----------------+-------+------+
| Semenov   | blood test     |   280 |    1 |
| Batkin    | blood test     |   200 |    2 |
| Grin      | ECG            |   570 |    1 |
| Sharova   | ECG            |   570 |    1 |
| Smertina  | ECG            |   540 |    3 |
| Petrov    | fluorography   |  1500 |    1 |
| Farid     | sugar analysis |   410 |    1 |
| Grigoriev | sugar analysis |   300 |    2 |
| Vasileva  | urine analysis |   300 |    1 |
| Naymov    | urine analysis |   240 |    2 |
+-----------+----------------+-------+------+
10 rows in set (0.00 sec)
```

Рисунок 4.7 – Оконная функция rank

На рисунке 4.8 представлена оконная функция dense_rank, которая выводит индекс по списку, в котором они идут. Если есть совпадения по индексу – то выводит одинаковый индекс без пропуска.

```
mysql> select surname_p, name_analyzes, price, dense_rank() over(partition by name_analyzes order by price desc) as derang
from analyzes join patient where patient.id_patient=analyzes.id_analyzes;
+-----------+----------------+-------+--------+
| surname_p | name_analyzes  | price | derang |
+-----------+----------------+-------+--------+
| Semenov   | blood test     |   280 |      1 |
| Batkin    | blood test     |   200 |      2 |
| Grin      | ECG            |   570 |      1 |
| Sharova   | ECG            |   570 |      1 |
| Smertina  | ECG            |   540 |      2 |
| Petrov    | fluorography   |  1500 |      1 |
| Farid     | sugar analysis |   410 |      1 |
| Grigoriev | sugar analysis |   300 |      2 |
| Vasileva  | urine analysis |   300 |      1 |
| Naymov    | urine analysis |   240 |      2 |
+-----------+----------------+-------+--------+
10 rows in set (0.00 sec)
```

Рисунок 4.8 – Оконная функция dense_rank

На рисунке 4.9 представлена оконная функция ntile, которая распределяет строки в группы.

```
mysql> select surname_p, name_analyzes, price, ntile(3) over(partition by name_analyzes) as ntl from analyzes join patient
where patient.id_patient=analyzes.id_analyzes;
+-----------+----------------+-------+-----+
| surname_p | name_analyzes  | price | ntl |
+-----------+----------------+-------+-----+
| Semenov   | blood test     |   280 |   1 |
| Batkin    | blood test     |   200 |   2 |
| Smertina  | ECG            |   540 |   1 |
| Grin      | ECG            |   570 |   2 |
| Sharova   | ECG            |   570 |   3 |
| Petrov    | fluorography   |  1500 |   1 |
| Farid     | sugar analysis |   410 |   1 |
| Grigoriev | sugar analysis |   300 |   2 |
| Vasileva  | urine analysis |   300 |   1 |
| Naymov    | urine analysis |   240 |   2 |
+-----------+----------------+-------+-----+
10 rows in set (0.00 sec)
```

Рисунок 4.9 – Оконная функция ntile

Функции смещения:

На рисунке 4.10 представлена оконная функция lag, которая выводит данные из предыдущей строчки.

```
mysql> select surname_p, name_analyzes, price, lag(price) over(partition by name_analyzes order by price desc) as lagg
from analyzes join patient where patient.id_patient=analyzes.id_analyzes;
+-----------+----------------+-------+------+
| surname_p | name_analyzes  | price | lagg |
+-----------+----------------+-------+------+
| Semenov   | blood test     |   280 | NULL |
| Batkin    | blood test     |   200 |  280 |
| Grin      | ECG            |   570 | NULL |
| Sharova   | ECG            |   570 |  570 |
| Smertina  | ECG            |   540 |  570 |
| Petrov    | fluorography   |  1500 | NULL |
| Farid     | sugar analysis |   410 | NULL |
| Grigoriev | sugar analysis |   300 |  410 |
| Vasileva  | urine analysis |   300 | NULL |
| Naymov    | urine analysis |   240 |  300 |
+-----------+----------------+-------+------+
10 rows in set (0.00 sec)
```

Рисунок 4.10 – Оконная функция lag

На рисунке 4.11 представлена оконная функция lead которая выводит данные из следующей строчки.

```
mysql> select surname_p, name_analyzes, price, lead(price) over(partition by name_analyzes order by price desc) as leadd
from analyzes join patient where patient.id_patient=analyzes.id_analyzes;
+-----------+----------------+-------+-------+
| surname_p | name_analyzes  | price | leadd |
+-----------+----------------+-------+-------+
| Semenov   | blood test     |   280 |   200 |
| Batkin    | blood test     |   200 |  NULL |
| Grin      | ECG            |   570 |   570 |
| Sharova   | ECG            |   570 |   540 |
| Smertina  | ECG            |   540 |  NULL |
| Petrov    | fluorography   |  1500 |  NULL |
| Farid     | sugar analysis |   410 |   300 |
| Grigoriev | sugar analysis |   300 |  NULL |
| Vasileva  | urine analysis |   300 |   240 |
| Naymov    | urine analysis |   240 |  NULL |
+-----------+----------------+-------+-------+
10 rows in set (0.00 sec)
```

Рисунок 4.11 – Оконная функция lead

На рисунке 4.12 представлена оконная функция first_value которая выводит первые значения в столбце.

```
mysql> select surname_p, name_analyzes, price, first_value(price) over(partition by name_analyzes order by price desc) as
firstvalue from analyzes join patient where patient.id_patient=analyzes.id_analyzes;
+-----------+----------------+-------+------------+
| surname_p | name_analyzes  | price | firstvalue |
+-----------+----------------+-------+------------+
| Semenov   | blood test     |   280 |        280 |
| Batkin    | blood test     |   200 |        280 |
| Grin      | ECG            |   570 |        570 |
| Sharova   | ECG            |   570 |        570 |
| Smertina  | ECG            |   540 |        570 |
| Petrov    | fluorography   |  1500 |       1500 |
| Farid     | sugar analysis |   410 |        410 |
| Grigoriev | sugar analysis |   300 |        410 |
| Vasileva  | urine analysis |   300 |        300 |
| Naymov    | urine analysis |   240 |        300 |
+-----------+----------------+-------+------------+
10 rows in set (0.00 sec)
```

Рисунок 4.12 – Оконная функция first_value

На рисунке 4.13 представлена оконная функция last_value, которая выводит последние значения в столбце.

```
mysql> select surname_p, name_analyzes, price, last_value(price) over(partition by name_analyzes) as lastvalue from
analyzes join patient where patient.id_patient=analyzes.id_analyzes;
+-----------+----------------+-------+-----------+
| surname_p | name_analyzes  | price | lastvalue |
+-----------+----------------+-------+-----------+
| Semenov   | blood test     |   280 |       200 |
| Batkin    | blood test     |   200 |       200 |
| Smertina  | ECG            |   540 |       570 |
| Grin      | ECG            |   570 |       570 |
| Sharova   | ECG            |   570 |       570 |
| Petrov    | fluorography   |  1500 |      1500 |
| Farid     | sugar analysis |   410 |       300 |
| Grigoriev | sugar analysis |   300 |       300 |
| Vasileva  | urine analysis |   300 |       240 |
| Naymov    | urine analysis |   240 |       240 |
+-----------+----------------+-------+-----------+
10 rows in set (0.00 sec)
```

Рисунок 4.13 – Оконная функция last_value

Вывод: в ходе практической работы изучили виды оконных функций и попробовали ими по пользоваться.