

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Доц, к.т.н.

должность, уч. степень, звание

подпись, дата

Турнецкая Е.Л

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

ТЕСТИРОВАНИЕ БАЗЫ ДАННЫХ

по курсу: Инженер по тестированию. Основы тестирования программного обеспечения

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

2346

подпись, дата

А. М. Воронцова

Е. Н. Костынюк

С. А. Себелев

М. Н. Федорук

инициалы, фамилия

Санкт-Петербург 2025

1. Цель работы

Приобретение практических навыков составления тестовых запросов к реляционной БД.

2. Описание предметной области

MySQL — это реляционная система управления базами данных (СУБД) с открытым исходным кодом, разработанная компанией Oracle. Она является одной из самых популярных и широко используемых СУБД в мире благодаря своей надежности, производительности и простоте использования.

3. Ход работы

В нашем варианте выделено 3 сущности: Пациент(Patient), Врач(Doctor), Обращение к врачу(Contacting). Для удобства составления инфологической модели (рис. 1) были составлены таблицы 1.1, 1.2, 1.3.

В таблицах приняты следующие сокращения:

NN (Not Null) – поле с обязательным заполнением,

AI (Avto Increment) – поле типа счетчик,

ПК – первичный ключ,

ВК – внешний ключ.

Таблица 1.1

Определение атрибутов сущности Пациент (Patient)

Атрибуты	Тип данных	Размер(бит)	Комментарий
patientId	INT	32	Код пациента, NN, AI, ПК
patientSurname	VARCHAR	45	Фамилия, NN
patientName	VARCHAR	45	Имя, NN
patientSex	VARCHAR	45	Пол
patientBirthDate	DATE	24	Год рождения

Таблица 1.2

Определение атрибутов сущности Врач (Doctor)

Атрибуты	Тип данных	Размер(бит)	Комментарий
docId	INT	32	Код врача, NN, AI, ПК
docSurname	VARCHAR	45	Фамилия, NN
docSpecialization	VARCHAR	45	Специализация, NN
docCategory	VARCHAR	45	Категория, NN

Таблица 1.3

Определение атрибутов сущности Обращение к врачу (Contacting)

Атрибуты	Тип данных	Размер(бит)	Комментарий
----------	------------	-------------	-------------

contactId	INT	32	Код обращения, NN, AI
patientId	INT	32	Код пациента, NN, BK
docId	INT	32	Код врача, NN, BK
contactDate	DATETIME	48	Дата обращения, NN
contactPrice	DECIMAL	10, 2	Оплата за прием, NN

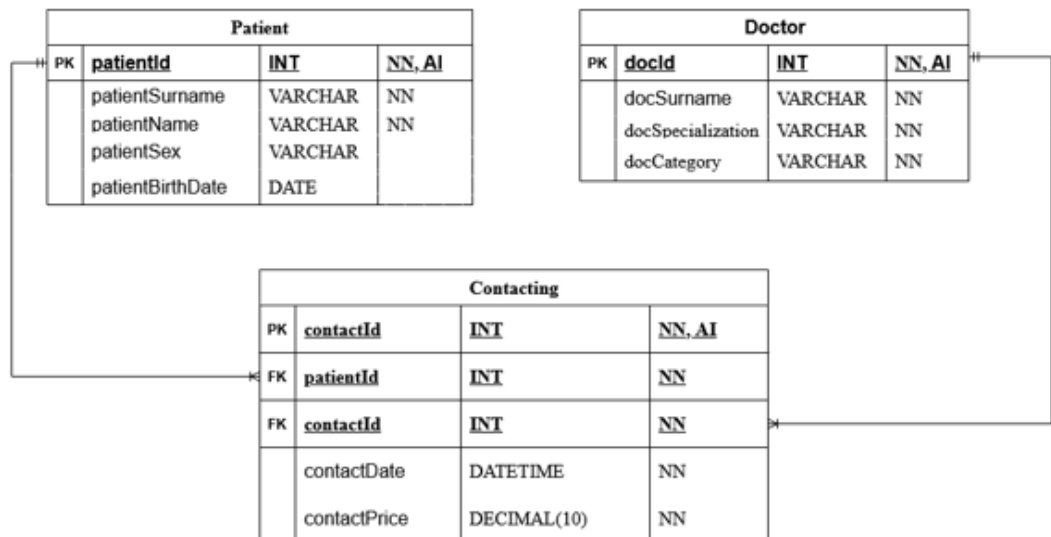


Рис. 1 - инфологическая модель

В качестве среды разработки была выбран IDE VScode с расширением SQLtools, потому что оно позволяет произвести бесконечное число запросов в отличие от онлайн платформа <https://sqliteonline.com/>.

Ниже код для создания БД

```

CREATE TABLE Patient(
    patientId INT AUTO_INCREMENT PRIMARY KEY,
    patientName VARCHAR(45) Not Null,
    patientSurname VARCHAR(45) Not Null,
    patientSex VARCHAR(45),
    patientBirthDate DATE
);

CREATE TABLE Doctor(
    docId INT AUTO_INCREMENT PRIMARY KEY,
    docSurname VARCHAR(45) Not Null,
    docSpecializationn VARCHAR(45) Not Null,
    docCategory VARCHAR(45) Not Null
);

CREATE TABLE Contacting(
    contactId INT AUTO_INCREMENT PRIMARY KEY,
    docId INT Not Null,
    patientId INT Not Null,
    contactPrice DECIMAL(10) Not Null,

```

```

        contactDate DATETIME Not Null,
        FOREIGN KEY (patientId) REFERENCES Patient(patientId) ON
DELETE CASCADE,
        FOREIGN KEY (docId) REFERENCES Doctor(docId) ON DELETE
CASCADE
    );

```

Увидеть созданные таблицы можно в MySQL Workbench рис. 2.

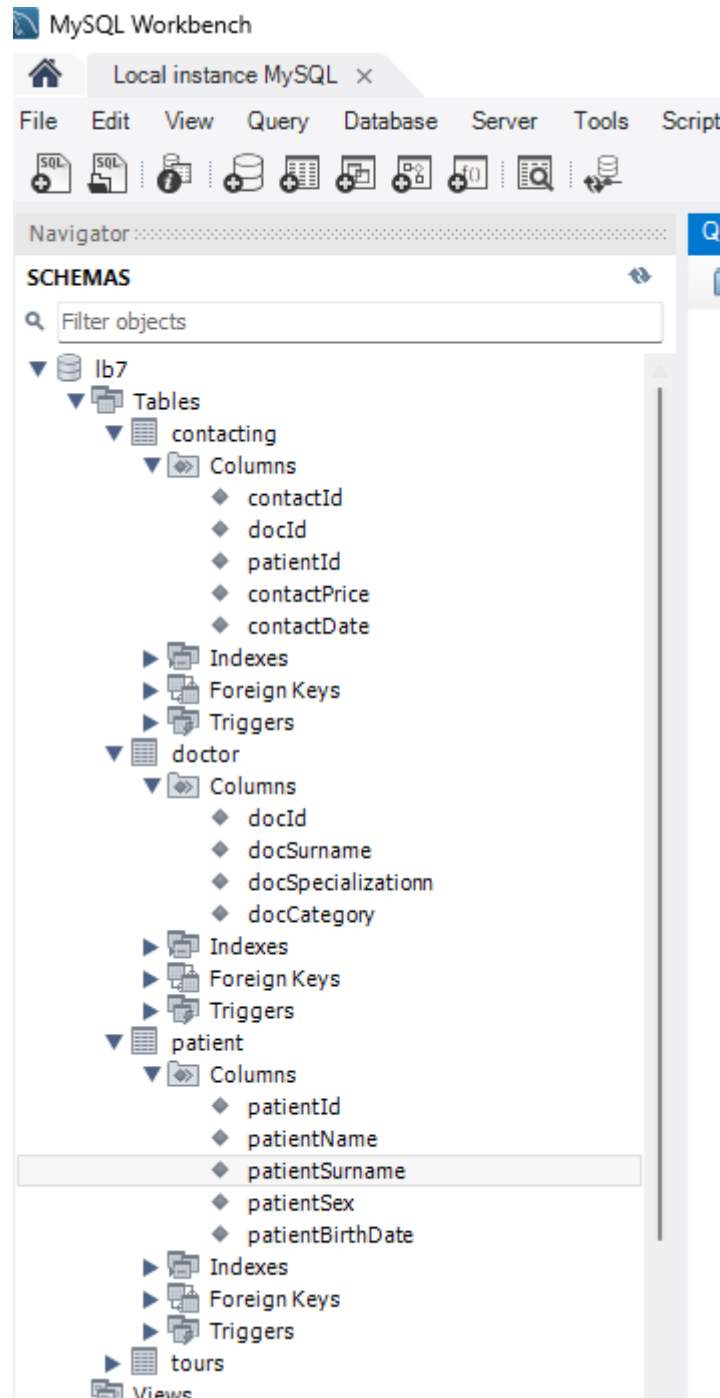


Рис. 2 – создание таблиц

Таблицы надо заполнить. Код для заполнения:

```

INSERT INTO Patient (patientId, patientName, patientSurname,
patientSex, patientBirthDate)
VALUES

```

```
(1, 'Денис', 'Амитов', 'Мужской', '2003-02-01'),
(2, 'Ксения', 'Васенина', 'Женский', '2012-01-10'),
(3, 'София', 'Воронцова', 'Женский', '1999-03-20'),
(4, 'Анна', 'Воронцова', 'Женский', '2004-06-17'),
(5, 'Георгий', 'Герасимов', 'Мужской', '1980-06-25'),
(6, 'Карина', 'Голант', 'Женский', '2000-11-25'),
(7, 'Дарья', 'Ильина', '', '2005-06-01');
```

```
INSERT INTO Doctor (docId, docSurname, docSpecializationn,
docCategory)
```

```
VALUES
```

```
(1, 'Исаева', 'Психиатр', 'Вторая'),
(2, 'Козлова', 'Кардиолог', 'Первая'),
(3, 'Костынюк', 'Спортивный врач', 'Высшая'),
(4, 'Кроян', 'Дерматолог', 'Высшая'),
(5, 'Лобанова', 'Невролог', 'Первая'),
(6, 'Лысак', 'Невролог', 'Вторая'),
(7, 'Матюшенко', 'Анестезиолог', 'Вторая');
```

```
INSERT INTO Contacting (contactId, docId, patientId, contactPrice,
contactDate)
```

```
VALUES
```

```
(1, 3, 4, 300, '2025-11-23 12:00:00'),
(2, 3, 5, 900, '2025-11-23 12:30:00'),
(3, 6, 3, 500, '2025-11-23 12:30:00'),
(4, 5, 1, 999, '2025-11-23 10:15:00'),
(5, 4, 2, 891, '2025-11-23 17:45:00'),
(6, 1, 7, 333, '2025-11-23 14:15:00'),
(7, 2, 6, 888, '2025-11-23 16:20:00');
```

Для первичного осмотра можно вызвать таблицы из VScode (рис. 3) или просмотреть в MySQL Workbench (рис. 4).

The screenshot shows a VSCode editor with a SQL script on the left and its results in a table on the right. The script consists of three lines: `SELECT * from contacting;`, `SELECT * from doctor;`, and `select * from patient;`. The results are displayed in three separate tables, each with a filter icon.

contactId	docId	patientId	contactPrice	contactDate
1	3	4	300	1763892000000
2	3	5	900	1763901000000
3	6	3	500	1763901000000
4	5	1	999	1763892900000
5	4	2	891	1763919900000
6	1	7	333	1763907300000
7	2	6	888	1763914800000

docId	docSurname	docSpecializationn	docCategory
1	Исаева	Психиатр	Вторая
2	Козлова	Кардиолог	Первая
3	Костынюк	Спортивный врач	Высшая
4	Кроян	Дерматолог	Высшая
5	Лобанова	Невролог	Первая
6	Лысак	Невролог	Вторая
7	Матюшенко	Анестезиолог	Вторая

Рис. 3 – таблица в VScode

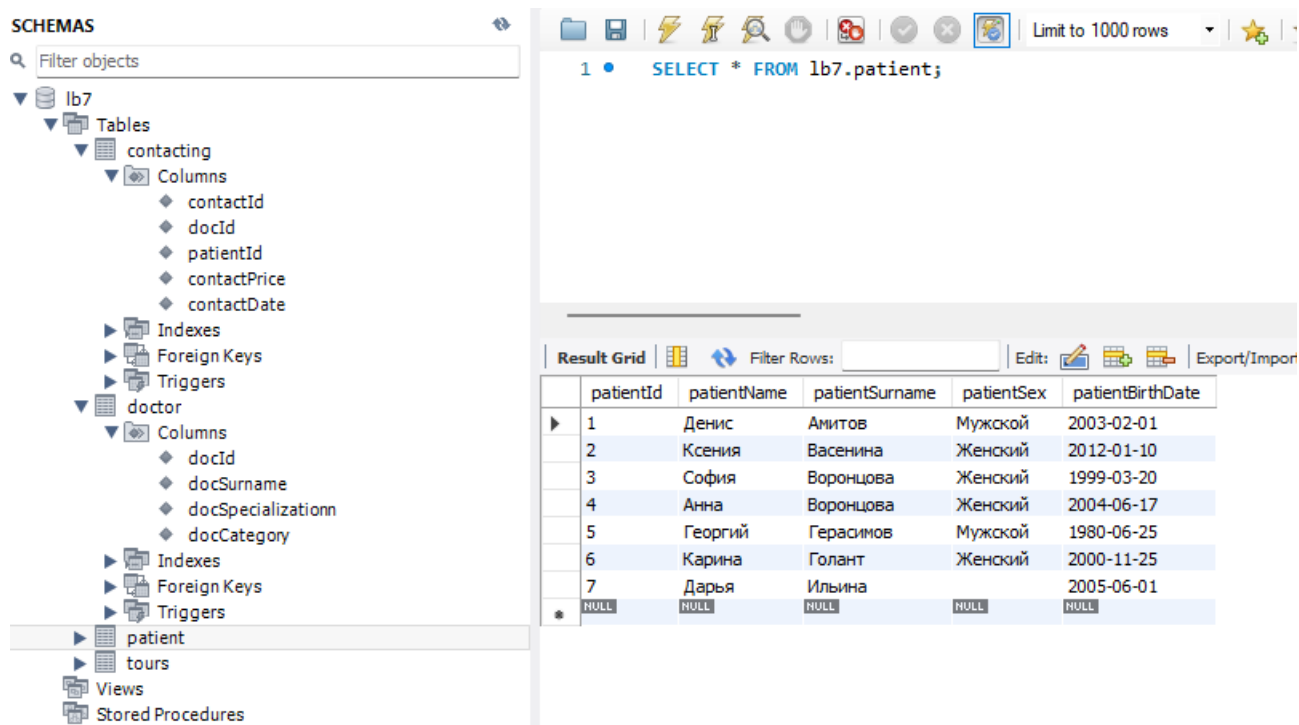


Рис. 4 – таблица в MySQL Workbench

Теперь для тестирования БД создадим 5 запросов. Результаты их исполнения видны на рис. 4-8. Код приведён ниже:

```
-- Подсчитать общее количество приемов врачей
SELECT
    count(contactId) AS 'Количество приемов врачей'
FROM
    contacting;

-- нахождение пациентов Воронцовых
SELECT
    patientSurname,
    patientName
FROM
    patient
WHERE
    patientSurname like 'Воронцова';

-- Вывести услуги врачей стоимостью большее 400
SELECT
    d.docSurname AS 'Врач',
    d.docSpecializationn AS 'Специализация',
    p.patientSurname AS 'Пациент',
    c.contactPrice AS 'Стоимость приёма',
    TIME(c.contactDate) AS 'Время'
FROM
    contacting AS c
    INNER JOIN doctor as d ON c.docId = d.docId
    INNER JOIN patient as p ON c.patientId = p.patientId
GROUP BY
    d.docSurname,
    d.docSpecializationn,
    p.patientSurname,
```

```

        c.contactPrice,
        c.contactDate
HAVING
        c.contactPrice > 400;

-- Вывести услуги врачей стоимостью большее 400 и принимают позже 12:30
SELECT
    d.docSurname AS 'Врач',
    d.docSpecializationn AS 'Специализация',
    p.patientSurname AS 'Пациент',
    c.contactPrice AS 'Стоимость приёма',
    TIME(c.contactDate) AS 'Время'
FROM
    contacting AS c
    INNER JOIN doctor as d ON c.docId = d.docId
    INNER JOIN patient as p ON c.patientId = p.patientId
WHERE
    c.contactPrice > 400
    AND TIME(c.contactDate) > '12:30:00';

-- Врачи с максимальным количеством приемов
SELECT
    d.docId,
    d.docSurname AS 'Фамилия врача',
    d.docSpecializationn AS 'Специализация',
    COUNT(c.contactId) AS 'Количество приёмов'
FROM
    contacting AS c
    INNER JOIN doctor AS d ON c.docId = d.docId
GROUP BY
    d.docId,
    d.docSurname,
    d.docSpecializationn
HAVING
    COUNT(c.contactId) = (
        SELECT
            MAX(appointment_count)
        FROM
            (
                SELECT
                    COUNT(contactId) as appointment_count
                FROM
                    contacting
                GROUP BY
                    docId
            ) AS tmp
    );

```


-- Подсчитать общее количество приемов врачей	Количество пр...
SELECT	abc Filter...
count(contactId) AS 'Количество приемов врачей'	
FROM	7
contacting;	

Рис. 4 – Первый запрос: подсчитать общее количество приемов врачей

```
-- нахождение пациентов Воронцовых
SELECT
  patientSurname,
  patientName
FROM
  patient
WHERE
  patientSurname like 'Воронцова';
```

patientSurname	patientName
Воронцова	София
Воронцова	Анна

Рис. 5 – Второй запрос: нахождение пациентов Воронцовых

sql_scripts >  zapros.sql

-- Вывести услуги врачей стоимостью больше 400

SELECT

d.docSurname AS 'Врач',

d.docSpecializationn AS 'Специализация',

p.patientSurname AS 'Пациент',

c.contactPrice AS 'Стоимость приёма',

TIME(c.contactDate) AS 'Время'

FROM

contacting AS c

INNER JOIN doctor as d ON c.docId = d.docId

INNER JOIN patient as p ON c.patientId = p.patientId

GROUP BY

d.docSurname,

d.docSpecializationn,

p.patientSurname,

c.contactPrice,

c.contactDate

HAVING

c.contactPrice > 400;

-- Подсчитать общее количество прием...

-- нахождение пациентов Воронцовых ...

-- Вывести услуги врачей стоимостью б...







Врач	Специализация	Пациент	Стоимость при...	Время
 Filter...	 Filter...	 Filter...	 Filter...	 Filter...
Костынюк	Спортивный врач	Герасимов	900	+12:30:00.000000
Лысак	Невролог	Воронцова	500	+12:30:00.000000
Лобанова	Невролог	Амитов	999	+10:15:00.000000
Кроян	Дерматолог	Васенина	891	+17:45:00.000000
Козлова	Кардиолог	Голант	888	+16:20:00.000000

Рис. 6 – Третий запрос: вывести услуги врачей стоимостью больше 400

sql_scripts >  zapros.sql

```
35 -- Вывести услуги врачей стоимостью больше 400 и принимают позже 12:30
36 SELECT
37   d.docSurname AS 'Врач',
38   d.docSpecializationn AS 'Специализация',
39   p.patientSurname AS 'Пациент',
40   c.contactPrice AS 'Стоимость приёма',
41   TIME(c.contactDate) AS 'Время'
42 FROM
43   contacting AS c
44   INNER JOIN doctor as d ON c.docId = d.docId
45   INNER JOIN patient as p ON c.patientId = p.patientId
46 WHERE
47   c.contactPrice > 400
48 AND TIME(c.contactDate) > '12:30:00';
```

-- Подсчитать общее количество прием...-- нахождение пациентов Воронцовых...-- Вывести услуги врачей стоимостью 6...-- Вывести услуги врачей стоимостью 6...






Врач	Специализация	Пациент	Стоимость при...	Время
 Filter...	 Filter...	 Filter...	 Filter...	 Filter...
Кроян	Дерматолог	Васенина	891	+17:45:00.000000
Козлова	Кардиолог	Голант	888	+16:20:00.000000

Рис. 7 – Четвёртый запрос: Вывести услуги врачей стоимостью больше 400 и принимают позже 12:30

47

МНЕКЕ

51

-- Врачи с максимальным количеством приемов

52

SELECT

53

d.docId,

54

d.docSurname AS 'Фамилия врача',

55

d.docSpecializationn AS 'Специализация',

56

COUNT(c.contactId) AS 'Количество приёмов'

57

FROM

58

contacting AS c

59

INNER JOIN doctor AS d ON c.docId = d.docId

60

GROUP BY

61

d.docId,

62

d.docSurname,

63

d.docSpecializationn

64

HAVING

65

COUNT(c.contactId) = (

66

SELECT

67

MAX(appointment_count)

68

FROM

69

(

70

SELECT

71

COUNT(contactId) as appointment_count

72

FROM

73

contacting

74

GROUP BY

75

docId

76

) AS tmp

77

);

78

-- Подсчитать общее количество прием...

-- нахождение пациентов воронцовых ...

-- вывести





docId	Фамилия врача	Специализация	Количество приёмов
 Filter...	 Filter...	 Filter...	 Filter...
3	Костынюк	Спортивный врач	2

Рис. 8 – Пятый запрос: врачи с максимальным количеством приемов

4. Вывод

В ходе выполнения практической работы по тестированию базы данных были успешно приобретены и закреплены практические навыки работы с реляционными базами

данных на всех этапах их жизненного цикла.

Полученные и закреплённые навыки:

- Проектирование БД с помощью таблиц со сущностями и инфологической модели.
- Реализация БД MySQL на персональном компьютере посредством IDE VsCode
- Наполнение БД
- Тестирование запросами

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. DeepSeek – URL: <https://chat.deepseek.com>
2. “MySQL. Сборник рецептов” - Поль Дюбуа
3. Учим Базы Данных за 1 час! #От Профессионала - Хауди Хо™ - Просто о мире IT!
– URL: <https://youtu.be/ZKU7-ktaa2o?si=tGcuWa9-hWp8kBIZ>