

Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Институт информатики и вычислительной техники

09.03.01 "Информатика и вычислительная техника"
профиль "Программное обеспечение средств вычислительной
техники и автоматизированных систем"

Расчетно-графическая работа

по дисциплине «Базы данных»

Выполнил: студент 4 курса
ИВТ, гр. ИП-113
Ефремов К.В.

Работу проверил: ассистент кафедры ПМиК
Дьячкова Марина Сергеевна

Новосибирск 2024

1. Задание

Тема базы данной: Больница

В ходе работы необходимо:

1. Спроектировать произвольную базу данных, соответствующую предметной области (согласно варианту) и позволяющую корректно хранить информацию об объектах этой предметной области.
2. Обеспечить пользовательский веб-интерфейс для добавления записей и просмотра содержимого БД.
3. Сформулировать и выполнить запросы к БД, имеющие смысл в контексте предметной области (согласно варианту). Обеспечить веб-интерфейс для запуска этих запросов.
4. Составить отчет, содержащий описание выполненной работы, в том числе некоторые этапы проектирования БД (см. ниже).

2. Описание базы данных

2.1

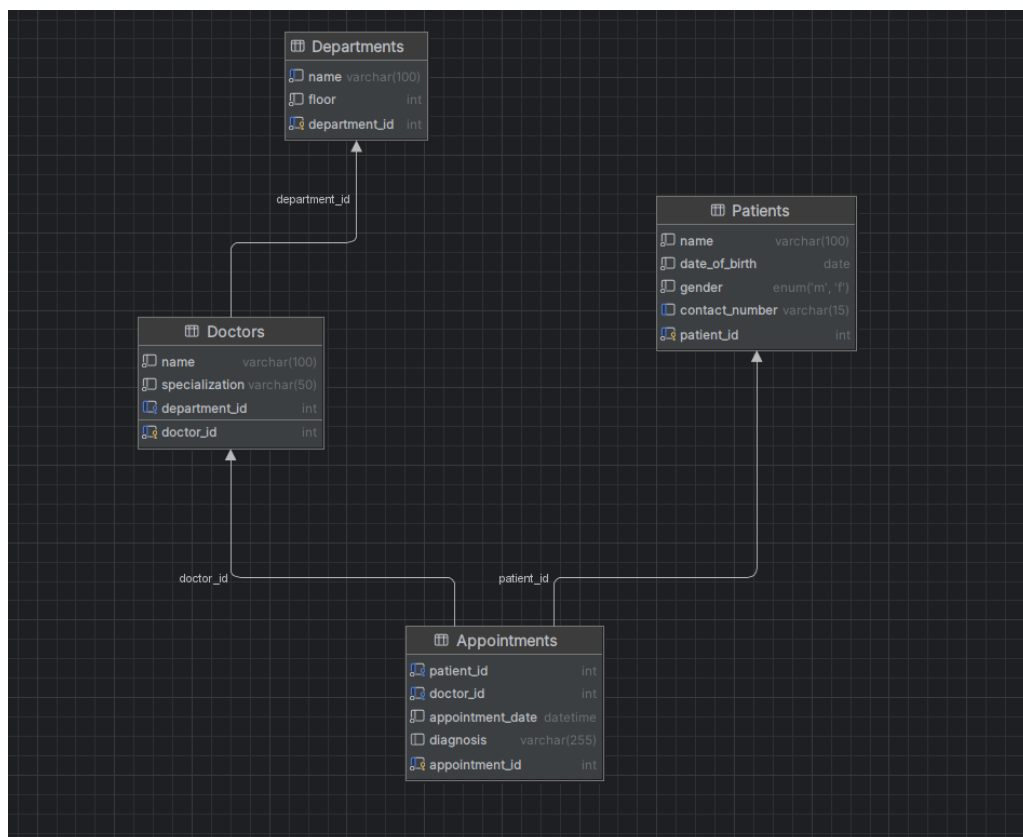


Рисунок 1. ER-диаграмма БД

Таблица 1. Сущности и связи

Сущность 1	Сущность 2	Тип связи (1:1, 1:M, M:N)	Описание связи
Пациенты	Запись о приеме	1:M	Один пациент может иметь множество записей на прием.
Врачи	Запись о приеме	1:M	Один врач может принимать множество пациентов (разные записи на прием).
Отделения	Врачи	1:M	Один отдел (например, кардиология) может включать множество врачей.

Таблица 2. Атрибуты сущностей

Сущность, название таблицы в БД	Название атрибута	Название атрибута в БД (имя поля)	Тип данных	Not Null	Primary key	Foreign key	Дополнительные ограничения целостности
Пациента Patients	номер пациента	<i>patient_id</i>	int	1	P		Auto increment
	ФИО пациента	<i>name</i>	varchar (100)	1			
	Дата рождения пациента	<i>date_of_birth</i>	Date	1			
	Пол пациента	<i>gender</i>	ENUM(M,F)	1			
	Номер телефона пациента	<i>contact_number</i>	varchar (15)	1			UNIQUE
Отделения Departaments	номер отделения	<i>department_id</i>	int	1	P		Auto increment UNIQUE
	Название отделения	<i>name</i>	varchar (100)	1			
	Этаж, на котором базируется отделение	<i>floor</i>	int	1 1			
Врачи Doctors	номер врача	<i>doctor_id</i>	int	1	P	F	Auto increment
	ФИО врача	<i>name</i>	varchar (100)	1			
	Специальность врача	<i>specialization</i>	varchar (50)	1			
	Номер отделения, к которому относится врач	<i>department_id</i>	int	1			
Записи о приеме Appointments	номер записи о приеме	<i>appointment_id</i>	int	1	P	F F	Auto increment
	номер пациента	<i>patient_id</i>	int	1			
	номер доктора	<i>doctor_id</i>	int	1			
	дата и время приема	<i>appointment_date</i>	datetime	1			
	диагноз выявленный на приеме	<i>diagnosis</i>	varchar(255)				

2.2 Данные БД (содержимое таблиц)

Таблица **Patients**

patient_id	name	date_of_birth	gender	contact_number
1	Иванов Иван Иванович	1980-01-01	M	8912345678
2	Петрова Мария Николаевна	1990-02-02	F	8999887766
3	Сидоров Петр Петрович	1975-03-03	M	8911223344
4	Васильева Ольга Ивановна	1985-04-04	F	8955500011
5	Николаев Дмитрий Николаевич	1995-05-05	M	8900112233

Таблица **Departaments**

department_id	name	floor
1	Кардиология	1
2	Хирургия	2
3	Терапия	3
4	Неврология	4
5	Офтальмология	5

Таблица **Doctors**

doctor_id	name	specialization	department_id
1	Смирнов Иван Петрович	Кардиолог	1
2	Иванова Лидия Петровна	Хирург	2
3	Петров Петр Петрович	Терапевт	3
4	Сидорова Дарья Дмитриевна	Офтальмолог	4
5	Васильев Василий Васильевич	Педиатр	5

Таблица **Appointments**

appointment_id	patient_id	doctor_id	appointment_date	diagnosis
1	1	1	2024-11-10 09:00:00	Гипертония
2	2	2	2024-11-10 10:00:00	Острый аппендицит
3	3	3	2024-11-10 11:00:00	ОРВИ
4	4	4	2024-11-10 12:00:00	Мигрень
5	5	5	2024-11-10 13:00:00	Дальнозоркость

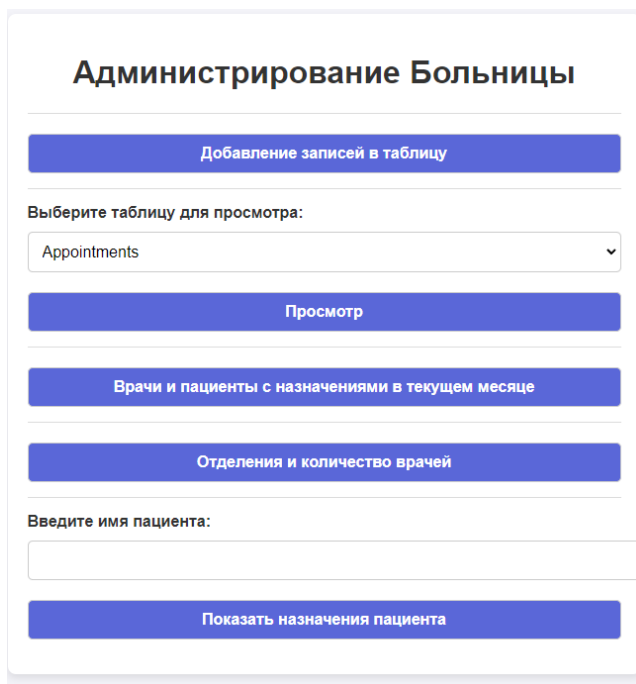
3. Описание веб-интерфейса

Веб-интерфейс состоит из нескольких страниц, каждая из которых предоставляет доступ к определенному функционалу для управления данными в базе.

3.1 Главная страница

Функционал:

- Содержит кнопку с ссылкой для перехода на страницу с добавлением пациентов, отделений, врачей, записей о приеме.
- Содержит кнопку с ссылкой на страницу для просмотра таблиц
- Содержит кнопки с ссылками на страницы запросов



3.2 Просмотр данных таблиц

Кнопка “Просмотр” на главной странице предоставляет пользователю возможность просматривать данные из выбранной таблицы базы данных (таких как Appointments, Departaments, Doctors, Patients).

Интерфейс выполнен с динамическим выбором и загрузкой содержимого таблицы на основе параметра, передаваемого через форму.

Функционал:

- Выбор таблицы для отображения: пользователь может выбрать таблицу, данные из которой он хочет просмотреть, с помощью выпадающего списка.
- Просмотр содержимого выбранной таблицы: при выборе таблицы и нажатии на кнопку «Просмотр» загружается содержимое указанной таблицы в виде HTML-таблицы.

Элементы интерфейса:

- Форма выбора таблицы:
 - Поле «Выберите таблицу»: Выпадающий список с названиями таблиц.
 - Кнопка «Просмотр»: при нажатии отправляет запрос, чтобы загрузить данные из выбранной таблицы.

- Таблица данных:
 - Если выбранная таблица существует, данные выводятся в таблице с колонками, соответствующими структуре таблицы.
- Строки данных: Данные из таблицы базы отображаются в строках. Каждая строка соответствует записи в таблице базы данных.

Проверки и ограничения:

1. Проверка существования таблицы:

- Если таблица с указанным именем пуста в базе данных, выводится сообщение об ошибке: «Таблица пуста.»

patient_id	name	date_of_birth	gender	contact_number
1	Иванов Иван Иванович	1980-01-01	M	8912345678
2	Петрова Мария Николаевна	1990-02-02	F	8999887766
3	Сидоров Петр Петрович	1975-03-03	M	8911223344
4	Васильева Ольга Ивановна	1985-04-04	F	8955500011
5	Николаев Дмитрий Николаевич	1995-05-05	M	8900112233

3.3 Добавление пациента

Эта страница позволяет пользователю добавлять новую запись в таблицу Patients, в которой сохраняются данные о пациентах, включая их ФИО, дату рождения, гендер, номер телефона.

Функционал страницы:

- Форма добавления пациента: Пользователь вводит ФИО пациента, дату рождения, гендер, номер телефона.
- Валидация данных: проводится несколько проверок данных, чтобы гарантировать их корректность.
- Запись данных в базу: при успешной проверке данных новая запись добавляется в таблицу Patients.

Элементы интерфейса:

- Поле «Имя»:
 - Текстовое поле для ввода имени пациента. Поле обязательно для заполнения.
- Поле «Дата рождения»:
 - Поле типа date, позволяет вводить дату рождения в формате ДД-ММ-ГГГГ. Обязательно для заполнения.
- Поле «Пол»:
 - Выпадающий список с выбором гендера (мужской, женский). Обязательно для заполнения.
- Поле «Номер телефона»:
 - На поле наложено регулярное выражение, которое следит за тем, чтобы пользователь ввел данные в правильном формате (первые два числа “89” и общая длина данных – 11 символов) . Обязательно для заполнения.
 -
- Кнопка «Добавить»:
 - Кнопка отправки формы для добавления пациента в базу данных.

Проверки и ограничения:

1. Проверка имени:
 - Имя должно содержать три слова и может состоять из кириллических символов, включая букву ё, каждое слово должно начинаться с заглавной буквы.
2. Проверка даты рождения:
 - Дата рождения проверяется на соответствие формату ДД-ММ-ГГГГ.
3. Проверка пола:
 - Список не позволяет выбрать другие варианты.
4. Проверка номера телефона:
 - Первые два числа “89” и общая длина данных – 11 символов.

Действия при отправке формы:

1. Если данные успешно прошли все проверки, запись добавляется в базу данных.
2. При успешной записи выводится сообщение «Студент успешно добавлен!».
3. Если добавление не удалось, на экран выводится сообщение с описанием ошибки.

Добавить нового пациента

Имя:
Иван Иванович

Дата рождения:
25.03.2004

Пол:
Мужской

Номер телефона:
89130997324

Добавить

3.4 Добавление отделения

Эта страница предназначена для добавления нового отделения в таблицу Departments. Пользователь вводит название отделения, выбирает этаж, на котором базируется отделение. При корректном заполнении форма добавляет запись в базу данных.

Функционал страницы

- Форма добавления отделения: Пользователь вводит название отделения, выбирает этаж.
- Валидация данных: Форма проводит проверки введенных данных на корректность.
- Запись данных в базу: при успешной проверке новая запись добавляется в таблицу Departments.

Элементы интерфейса

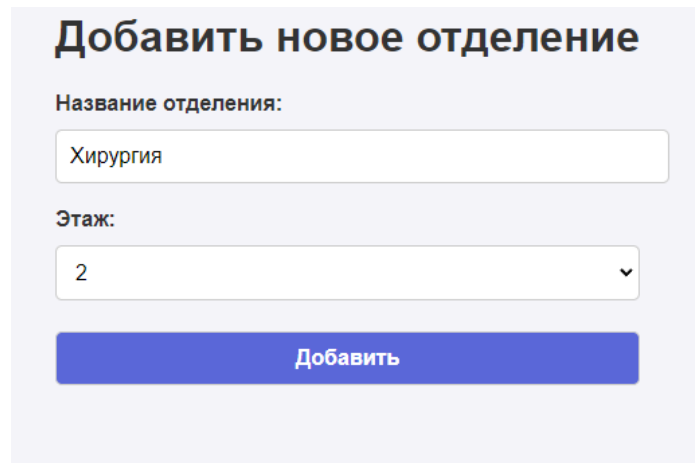
- Поле «Название отделения»:
 - Текстовое поле для ввода названия отделения (обязательно для заполнения).
- Поле «Этаж»:
 - Выпадающий список с номером этажа от 1 до 10 (обязательно для заполнения).
- Кнопка «Добавить»:
 - Кнопка отправки формы для добавления отделения в базу данных

Проверки и ограничения

1. Проверка имени:
 - Название отделения должно начинаться с большой буквы и содержать только буквы.
2. Проверка поля этажа:
 - Список не позволяет выбрать неправильное значение
3. Проверка специализации:
 - Поле обязательно для заполнения, и пользователь должен выбрать одно из значений в выпадающем списке.

Действия при отправке формы

1. При успешном заполнении всех полей данные записываются в таблицу Departments.
2. Если запись была добавлена успешно, отображается сообщение «Запись добавлена».
3. Если происходит ошибка, на экран выводится сообщение с описанием ошибки.



Добавить новое отделение

Название отделения:

Хирургия

Этаж:

2

Добавить

3.4 Добавление врача

Эта страница позволяет пользователю добавить нового врача в таблицу Doctors. Пользователь вводит ФИО врача, его специальность, в списке выбирает к какому отделению относится врач.

Функционал страницы:

1. Заполнение данных о враче.
2. Добавление врача: при корректном заполнении формы новое занятие записывается в таблицу Doctors.

Элементы интерфейса:

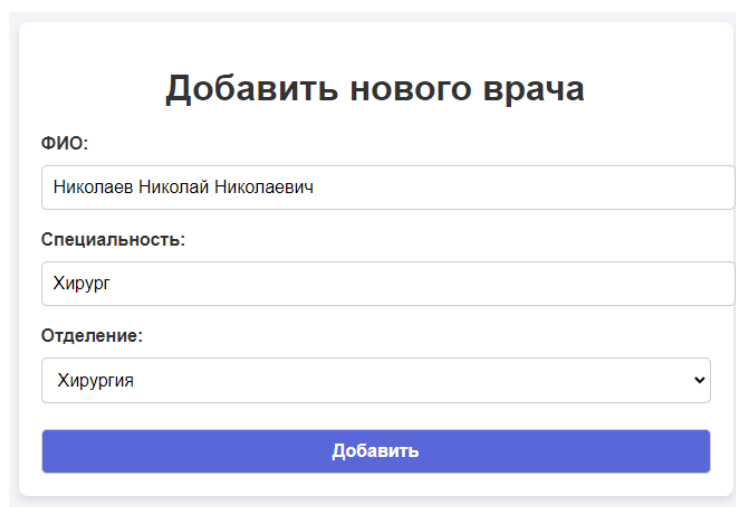
- Поле «ФИО»:
 - Текстовое поле для ввода имени врача. Поле обязательно для заполнения.
- Поле «Специальность»:
 - Текстовое поле для ввода специальности врача. Поле обязательно для заполнения:
- Поле «Отделение»:
 - Список отделения из таблицы Departments.
- Кнопка «Добавить»:
 - Добавляет запись в таблицу.

Проверки и ограничения:

1. Проверка заполнения ФИО:
 - Имя должно содержать три слова и может состоять из кириллических символов, включая букву ё, каждое слово должно начинаться с заглавной буквы.
2. Проверка заполнения Специальности:
 - Название специальности должно начинаться с большой буквы и содержать только буквы.
3. Проверка заполнения Отделения:
 - Регулируется списком

Действия при отправке формы:

1. Если все поля заполнены корректно, данные записываются в таблицу **Doctors**.
2. При успешном добавлении записи на странице выводится сообщение «Доктор успешно добавлен!».
3. В случае, если какое-либо поле не заполнено, появляется сообщение с просьбой заполнить все поля.



Добавить нового врача

ФИО:

Специальность:

Отделение:

Добавить

3.5 Добавление приема пациента

Эта страница позволяет пользователю добавить новую запись о приеме пациента в таблицу Appointments. Пользователь выбирает пациента из списка, врача из списка, Дата приема и заполняет диагноз. После успешного заполнения формы данные сохраняются в базеданныхх.

Функционал страницы:

1. Заполнение информации о приеме пациента: Выбор пациента, врача, даты приема и заполнение диагноза.
2. Добавление группы: при корректном заполнении всех полей новая группа добавляется в таблицу Appointments.

Элементы интерфейса:

- Поле «Пациент»:
 - Выпадающий список с пациентами из таблицы Patients.
- Поле «Доктор»:
 - Выпадающий список с врачами из таблицы Doctors.
- Поле «Дата и время приема»:
 - Поле типа datetime для заполнения даты и времени приема
- Поле «Диагноз»:
 - Текстовое поле, в которое указывается диагноз пациента.
- Кнопка «Добавить»:
 - Отправляет форму для добавления группы в базу данных.

Проверки и ограничения:

1. Проверка заполнения полей:
 - Если какое-либо из них не заполнено, появится сообщение с просьбой заполнить все поля.
2. Проверка выбора пациента и доктора:
 - Выпадающие списки не позволяют, выбрать неправильный вариант

Действия при отправке формы:

1. Если все поля заполнены корректно и выбран преподаватель, данные сохраняются в таблице Appointments.
2. При успешном добавлении записи выводится сообщение «Группа успешно добавлена!».
3. В случае, если какое-либо поле не заполнено или преподаватель не выбран, выводится соответствующее сообщение об ошибке.

Добавить новый прием

Пациент:
Иванов Николай Иванович

Доктор:
Николаев Николай Николаевич

Дата и время приема:
18.11.2024 22:50

Диагноз:
Перелом

Добавить

3.6 Запросы

На главной странице пользователь может выполнять три основных запроса к базе данных: просмотр студентов, записанных в определенную группу, подсчет количества студентов в каждой группе и просмотр занятий, проводимых конкретным преподавателем.

Функционал страницы

1. Запрос 1: Врачи и пациенты с назначениями в текущем месяце — выводит список врачей и пациентов, которые были на приеме у этого врача в текущем месяце.
2. Запрос 2: Отделения и количество врачей — отображает название отделений и количество врачей, которые работают в этих отделениях
3. Запрос 3: Показать назначения пациента — показывает имя пациента, его лечащего врача, дату и время приема, диагноз.

Элементы интерфейса

1. Три кнопки для выбора конкретного запроса.
2. Текстовое поле для ввода пользователем имени пациента

Обработка запросов и вывод данных

Запрос 1: если пользователь нажал кнопку, то срабатывает SQL-запрос, который:

- Извлекает имена врачей и пациентов для приемов, назначенных в текущем месяце. Он объединяет таблицы **Appointments**, **Doctors**, и **Patients** по соответствующим ключам, чтобы получить данные о врачах и пациентах, связанных с приемами.
- Условие отбора проверяет, что месяц и год даты приема совпадают с текущими
- Результат включает имя врача и пациента для каждого актуального приема.

Запрос 2: если пользователь нажал кнопку, то срабатывает SQL-запрос, который:

- Выводит названия отделений и количество врачей, работающих в каждом из них. Он объединяет таблицы **Departments** и **Doctors** с помощью левого соединения, чтобы учесть все отделения, даже те, где нет врачей.
- Сгруппировав данные по названиям отделений, запрос подсчитывает количество врачей в каждом отделении и возвращает результат.

Запрос 3: если пользователь выбрал пациента, выполняется запрос, который:

- Запрос отображает информацию о назначениях для конкретного пациента. Он возвращает имя пациента, имя врача, дату приема и диагноз. Таблицы **Appointments**, **Patients**, и **Doctors** объединяются, чтобы связать данные о пациентах и врачах с их назначениями.

Проверки и ограничения

- Третий запрос требует выбора заполнения значения (ФИО пациента).
- Если поле не заполнено, запрос не выполняется, и данные не выводятся.

2. SQL-запросы

Запрос 1: Врачи и пациенты с назначениями в текущем месяце

Формулировка: выводит список врачей и пациентов, которые были на приеме у этого врача в текущем месяце.

```
$sql = "SELECT Doctors.name AS doctor_name, Patients.name AS patient_name
FROM Appointments
JOIN Doctors ON Appointments.doctor_id = Doctors.doctor_id
JOIN Patients ON Appointments.patient_id = Patients.patient_id
WHERE MONTH(Appointments.appointment_date) = MONTH(CURRENT_DATE())
AND YEAR(Appointments.appointment_date) = YEAR(CURRENT_DATE());"
```

Врач: Смирнов Иван Петрович - **Пациент:** Иванов Иван Иванович
Врач: Иванова Лидия Петровна - **Пациент:** Петрова Мария Николоевна
Врач: Петров Петр Петрович - **Пациент:** Сидоров Петр Петрович
Врач: Сидорова Дарья Дмитриевна - **Пациент:** Васильева Ольга Ивановна
Врач: Васильев Василий Васильевич - **Пациент:** Николаев Дмитрий Николаевич

Запрос 2: Отделения и количество врачей

Формулировка: отображает название отделений и количество врачей, которые работают в этих отделениях.

```
$sql = "SELECT Departments.name AS department_name, COUNT(Doctors.doctor_id) AS doctor_count
FROM Departments
LEFT JOIN Doctors ON Departments.department_id = Doctors.department_id
GROUP BY Departments.name";
```

Отделение: Кардиология - **Количество врачей:** 1
Отделение: Неврология - **Количество врачей:** 1
Отделение: Офтальмология - **Количество врачей:** 1
Отделение: Терапия - **Количество врачей:** 1
Отделение: Хирургия - **Количество врачей:** 1

Запрос 3: Показать назначения пациента

Формулировка: показывает имя пациента, его лечащего врача, дату и время приема, диагноз.

```
"SELECT Patients.name AS patient_name, Doctors.name AS doctor_name,  
|      | Appointments.appointment_date, Appointments.diagnosis  
FROM Appointments  
JOIN Patients ON Appointments.patient_id = Patients.patient_id  
JOIN Doctors ON Appointments.doctor_id = Doctors.doctor_id  
WHERE Patients.name = ?");
```

Пациент:Иванов Иван Иванович, **Врач:** Смирнов Иван Петрович, **Дата назначения:** 2024-11-10 09:00:00, **Диагноз:** Гипертония

3. Листинг

5.1 SQL-скрипты для создания таблиц, вставки данных

Создание таблиц

--Таблица Пациенты

```
CREATE TABLE IF NOT
EXISTS Patients (
    patient_id INT
PRIMARY KEY
AUTO_INCREMENT,
    name VARCHAR(100)
NOT NULL,
    date_of_birth DATE
NOT NULL,
    gender ENUM('M', 'F')
NOT NULL,
    contact_number
VARCHAR(15) UNIQUE
);
```

--Таблица Отделения

```
CREATE TABLE IF NOT
EXISTS Departments (
    department_id INT
PRIMARY KEY
AUTO_INCREMENT,
    name VARCHAR(100)
NOT NULL UNIQUE,
    floor INT NOT NULL
);
```

--Таблица Врачи

```
CREATE TABLE IF NOT EXISTS Doctors (
    doctor_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    specialization VARCHAR(50) NOT NULL,
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES Departments(department_id)
);
```

--Таблица Записи

```
о приёме
CREATE TABLE IF NOT
EXISTS Appointments (
    appointment_id INT
PRIMARY KEY
AUTO_INCREMENT,
    patient_id INT NOT
NULL,
    doctor_id INT NOT
NULL,
    appointment_date
DATETIME NOT NULL,
    diagnosis
VARCHAR(255),
    FOREIGN KEY
(patient_id) REFERENCES
Patients(patient_id),
    FOREIGN KEY
(doctor_id) REFERENCES
Doctors(doctor_id)
);
```

5.2 PHP-скрипты

config.php

```
<?php
$servername = "mysql"; // Имя сервера MySQL (контейнер)
$username = "root";    // Пользователь
$password = "root";    // Пароль
$dbname = "maindb";    // Имя базы данных

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Ошибка подключения: " . $conn->connect_error);
}
?>
```

index.php

```
<?php
include 'functions.php';
$tables = getTables($conn);
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Администрирование Больницы</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Администрирование Больницы</h1>

        <div class="section">
            <form method="get" action="index2.php">
                <button type="submit">Добавление записей в таблицу</button>
            </form>
        </div>

        <div class="section">
            <form method="get" action="view_table.php">
                <div class="form-group">
                    <label for="table_view">Выберите таблицу для просмотра:</label>
                    <select name="table" id="table_view">
                        <?php foreach ($tables as $table): ?>
                            <option value="<?= $table ?>"><?= $table ?></option>
                        <?php endforeach; ?>
                    </select>
                </div>
            </form>
        </div>
    </div>
</body>
</html>
```



```

        </div>
        <button type="submit">Просмотр</button>
    </form>
</div>

<div class="section">
    <form method="get" action="query1.php">
        <button type="submit">Врачи и пациенты с назначениями в текущем
месяце</button>
    </form>
</div>

<div class="section">
    <form method="get" action="query2.php">
        <button type="submit">Отделения и количество врачей</button>
    </form>
</div>

<div class="section">
    <form method="get" action="query3.php">
        <div class="form-group">
            <label for="patient_name">Введите имя пациента:</label>
            <input type="text" name="patient_name" id="patient_name" required>
        </div>
        <button type="submit">Показать назначения пациента</button>
    </form>
</div>
</div>
</body>
</html>

```

view_table.php

```

<?php
include 'functions.php';

$table = $_GET['table'];

$rows = getTableContent($conn, $table);
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Просмотр данных из <?= $table ?></title>
</head>
<body>
    <div class="container">
        <h1>Содержимое таблицы: <?= $table ?></h1>
    </div>

```

```

<?php if (empty($rows)): ?>
    <p>Таблица пуста.</p>
<?php else: ?>
    <table border="1">
        <thead>
            <tr>
                <?php foreach ($rows[0] as $column => $value): ?>
                    <th><?= $column ?></th>
                <?php endforeach; ?>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($rows as $row): ?>
                <tr>
                    <?php foreach ($row as $value): ?>
                        <td><?= $value ?></td>
                    <?php endforeach; ?>
                </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
<?php endif; ?>
</div>
</body>
</html>

```

add_patients.php

```

<?php
include 'config.php';
include 'functions.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $name = $_POST['name'] ?? '';
    $date_of_birth = $_POST['date_of_birth'] ?? '';
    $gender = $_POST['gender'] ?? '';
    $contact_number = $_POST['contact_number'] ?? '';

    $errors = [];
    if (empty($name)) {
        $errors[] = "Поле 'Имя' обязательно для заполнения.";
    } else {
        if (!preg_match('/^[A-ЯЁ][a-яё]+ [A-ЯЁ][a-яё]+ [A-ЯЁ][a-яё]+$/u', $name)) {
            $errors[] = "Поле 'Имя' должно содержать Фамилию, Имя и Отчество, начинающиеся с заглавной буквы.";
        }
    }

    if (empty($date_of_birth)) $errors[] = "Поле 'Дата рождения' обязательно для заполнения.";
    if (empty($gender)) $errors[] = "Поле 'Пол' обязательно для выбора.";
    if (empty($contact_number)) {

```

```

        $errors[] = "Поле 'Номер телефона' обязательно для заполнения.";
    } else {
        if (!preg_match('/^89\d{9}$/', $contact_number)) {
            $errors[] = "Поле 'Номер телефона' должно начинаться с '89' и содержать ровно
10 цифр.";
        }
    }

    if (empty($errors)) {
        $sql = "INSERT INTO Patients (name, date_of_birth, gender, contact_number) VALUES
(?, ?, ?, ?)";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("ssss", $name, $date_of_birth, $gender, $contact_number);

        if ($stmt->execute()) {
            echo "<p>Пациент успешно добавлен!</p>";
        } else {
            echo "<p style='color:red;'>Ошибка: " . $stmt->error . "</p>";
        }
        $stmt->close();
    } else {
        foreach ($errors as $error) {
            echo "<p style='color:red;'>$error</p>";
        }
    }
}
?>

```

```

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Добавить пациента</title>
</head>
<body>
<div class="container">
    <h1>Добавить нового пациента</h1>
    <form method="post">
        <label for="name">Имя:</label>
        <input type="text" id="name" name="name" required
            pattern="[А-ЯЁ][а-яё]+ [А-ЯЁ][а-яё]+ [А-ЯЁ][а-яё]+"
            title="ФИО должно содержать три слова с заглавной буквы (Фамилия Имя
Отчество)"><br><br>

        <label for="date_of_birth">Дата рождения:</label>
        <input type="date" id="date_of_birth" name="date_of_birth" required><br><br>

        <label for="gender">Пол:</label>
        <select id="gender" name="gender" required>
            <option value="">Выберите пол</option>
            <option value="М">Мужской</option>
            <option value="Ж">Женский</option>

```

```

        </select><br><br>

        <label for="contact_number">Номер телефона:</label>
        <input type="text" id="contact_number" name="contact_number" required
            pattern="89\d{9}" title="Номер должен начинаться с 89 и содержать 10
цифр"><br><br>

        <button type="submit">Добавить</button>
    </form>
    <div class="container">
</body>
</html>

```

add_departments.php

```

<?php
include 'config.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $name = $_POST['name'] ?? '';
    $floor = $_POST['floor'] ?? '';

    $errors = [];

    if (empty($name)) {
        $errors[] = "Поле 'Название отделения' обязательно для заполнения.";
    } else {
        if (!preg_match('/^[А-ЯЁ][а-яё]+$/u', $name)) {
            $errors[] = "Поле 'Название отделения' должно начинаться с заглавной буквы и
содержать только буквы.";
        }
    }

    if (empty($floor)) {
        $errors[] = "Поле 'Этаж' обязательно для заполнения.";
    } elseif ($floor < 1 || $floor > 10) {
        $errors[] = "Поле 'Этаж' должно быть в диапазоне от 1 до 10.";
    }

    if (empty($errors)) {
        $sql = "INSERT INTO Departments (name, floor) VALUES (?, ?)";
        $stmt = $conn->prepare($sql);

        if ($stmt === false) {
            die('Ошибка при подготовке запроса: ' . $conn->error);
        }

        $stmt->bind_param("si", $name, $floor);

        if ($stmt->execute()) {
            echo "<p>Отдел успешно добавлен!</p>";
        }
    }
}

```

```

    } else {
        echo "<p style='color:red;'>Ошибка: " . $stmt->error . "</p>";
    }

    $stmt->close();
} else {
    foreach ($errors as $error) {
        echo "<p style='color:red;'>$error</p>";
    }
}
}
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Добавить отделение</title>
</head>
<body>
    <div class=container>
    <h1>Добавить новое отделение</h1>
    <form method="post">
        <label for="name">Название отделения:</label>
        <input type="text" id="name" name="name" required
            pattern="[А-ЯЁ][а-яё]+"
            title="Поле 'Название отделения' должно начинаться с заглавной буквы и
содержать только буквы."><br><br>

        <label for="floor">Этаж:</label>
        <select id="floor" name="floor" required>
            <?php
                for ($i = 1; $i <= 10; $i++) {
                    echo "<option value='$i'>$i</option>";
                }
            ?>
        </select><br><br>

        <button type="submit">Добавить</button>
    </form>
    </div>
</body>
</html>

```

add_doctors.php

```

<?php
include 'config.php';

$sql_departments = "SELECT department_id, name FROM Departments";
$result_departments = $conn->query($sql_departments);
$departments = [];

```

```

if ($result_departments->num_rows > 0) {
    while ($row = $result_departments->fetch_assoc()) {
        $departments[] = $row;
    }
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $name = $_POST['name'] ?? '';
    $specialization = $_POST['specialization'] ?? '';
    $department_id = $_POST['department_id'] ?? '';

    $errors = [];

    if (empty($name)) {
        $errors[] = "Поле 'Имя' обязательно для заполнения.";
    } else {
        if (!preg_match('/^[А-ЯЁ][а-яё]+ [А-ЯЁ][а-яё]+ [А-ЯЁ][а-яё]+$/u', $name)) {
            $errors[] = "Поле 'Имя' должно содержать Фамилию, Имя и Отчество, начинающиеся с заглавной буквы.";
        }
    }

    if (empty($specialization)) {
        $errors[] = "Поле 'Специальность' обязательно для заполнения.";
    } else {
        if (!preg_match('/^[А-ЯЁ][а-яё]+$/u', $specialization)) {
            $errors[] = "Поле 'Специальность' должно начинаться с заглавной буквы и содержать только буквы.";
        }
    }

    if (empty($department_id)) {
        $errors[] = "Поле 'Отделение' обязательно для выбора.";
    }

    if (empty($errors)) {
        $sql = "INSERT INTO Doctors (name, specialization, department_id) VALUES (?, ?, ?)";
        $stmt = $conn->prepare($sql);

        if ($stmt === false) {
            die('Ошибка при подготовке запроса: ' . $conn->error);
        }

        $stmt->bind_param("ssi", $name, $specialization, $department_id);

        if ($stmt->execute()) {
            echo "<p>Доктор успешно добавлен!</p>";
        } else {
            echo "<p style='color:red;'>Ошибка: " . $stmt->error . "</p>";
        }
    }
}

```

```

        $stmt->close();
    } else {
        foreach ($errors as $error) {
            echo "<p style='color:red;'>$error</p>";
        }
    }
}
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Добавить врача</title>
</head>
<body>
<div class="container">
    <h1>Добавить нового врача</h1>
    <form method="post">
        <label for="name">ФИО:</label>
        <input type="text" id="name" name="name" required
            pattern="[А-ЯЁ][а-яё]+ [А-ЯЁ][а-яё]+ [А-ЯЁ][а-яё]+"
            title="ФИО должно содержать три слова с заглавной буквы (Фамилия Имя
Отчество)"><br><br>

        <label for="specialization">Специальность:</label>
        <input type="text" id="specialization" name="specialization" required
            pattern="[А-ЯЁ][а-яё]+"
            title="Поле 'Специальность' должно начинаться с заглавной буквы и содержать
только буквы."><br><br>

        <label for="department_id">Отделение:</label>
        <select id="department_id" name="department_id" required>
            <option value="">Выберите отделение</option>
            <?php foreach ($departments as $department): ?>
                <option value="<?= $department['department_id'] ?>"><?= $department['name']
?></option>
            <?php endforeach; ?>
        </select><br><br>

        <button type="submit">Добавить</button>
    </form>
</div class="container">
</body>
</html>

```

add_appointments.php

```
<?php
include 'config.php';

$sql_patients = "SELECT patient_id, name FROM Patients";
$result_patients = $conn->query($sql_patients);
$patients = [];

if ($result_patients->num_rows > 0) {
    while ($row = $result_patients->fetch_assoc()) {
        $patients[] = $row;
    }
}

$sql_doctors = "SELECT doctor_id, name FROM Doctors";
$result_doctors = $conn->query($sql_doctors);
$doctors = [];

if ($result_doctors->num_rows > 0) {
    while ($row = $result_doctors->fetch_assoc()) {
        $doctors[] = $row;
    }
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $patient_id = $_POST['patient_id'] ?? '';
    $doctor_id = $_POST['doctor_id'] ?? '';
    $appointment_date = $_POST['appointment_date'] ?? '';
    $diagnosis = $_POST['diagnosis'] ?? '';

    $errors = [];

    if (empty($patient_id)) {
        $errors[] = "Поле 'Пациент' обязательно для выбора.";
    }

    if (empty($doctor_id)) {
        $errors[] = "Поле 'Доктор' обязательно для выбора.";
    }

    if (empty($appointment_date)) {
        $errors[] = "Поле 'Дата и время' обязательно для заполнения.";
    } elseif (!strtotime($appointment_date)) {
        $errors[] = "Некорректный формат даты и времени.";
    }

    if (empty($diagnosis)) {
        $errors[] = "Поле 'Диагноз' обязательно для заполнения.";
    }

    if (empty($errors)) {
```



```

        $sql = "INSERT INTO Appointments (patient_id, doctor_id, appointment_date,
diagnosis) VALUES (?, ?, ?, ?)";
        $stmt = $conn->prepare($sql);
        if ($stmt === false) {
            die('Ошибка при подготовке запроса: ' . $conn->error);
        }

        $stmt->bind_param("iiss", $patient_id, $doctor_id, $appointment_date, $diagnosis);

        if ($stmt->execute()) {
            echo "<p>Запись успешно добавлена!</p>";
        } else {
            echo "<p style='color:red;'>Ошибка: " . $stmt->error . "</p>";
        }

        $stmt->close();
    } else {
        foreach ($errors as $error) {
            echo "<p style='color:red;'>$error</p>";
        }
    }
}
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Добавить прием</title>
</head>
<body>
    <div class=container>
    <h1>Добавить новый прием</h1>
    <form method="post">
        <label for="patient_id">Пациент:</label>
        <select id="patient_id" name="patient_id" required>
            <option value="">Выберите пациента</option>
            <?php foreach ($patients as $patient): ?>
                <option value="<?= $patient['patient_id'] ?>"><?= $patient['name']
?></option>
            <?php endforeach; ?>
        </select><br><br>

        <label for="doctor_id">Доктор:</label>
        <select id="doctor_id" name="doctor_id" required>
            <option value="">Выберите доктора</option>
            <?php foreach ($doctors as $doctor): ?>
                <option value="<?= $doctor['doctor_id'] ?>"><?= $doctor['name'] ?></option>
            <?php endforeach; ?>
        </select><br><br>

        <label for="appointment_date">Дата и время приема:</label>

```

```

        <input type="datetime-local" id="appointment_date" name="appointment_date"
required><br><br>

        <label for="diagnosis">Диагноз:</label>
        <textarea id="diagnosis" name="diagnosis" required></textarea><br><br>

        <button type="submit">Добавить</button>
    </form>
</div>
</body>
</html>

```

index2.php

```

<?php
include 'functions.php';

$tables = getTables($conn);
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Выбор таблицы для добавления новой записи</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Выберите таблицу для работы</h1>

        <form method="get" action="add_patients.php">
            <button type="submit">Добавить нового пациента</button>
        </form>
        <br>

        <form method="get" action="add_departaments.php">
            <button type="submit">Добавить новое отделение</button>
        </form>
        <br>

        <form method="get" action="add_doctors.php">
            <button type="submit">Добавить нового врача</button>
        </form>
        <br>

        <form method="get" action="add_appoitments.php">
            <button type="submit">Добавить прием пациента</button>
        </form>
        <br>
    </div>

```

```

    <form method="get" action="index.php">
        <button type="submit">Вернуться на главную страницу</button>
    </form>
    <br>
</div>

</body>
</html>

```

query1.php

```

<?php
include 'config.php';

$sql = "SELECT Doctors.name AS doctor_name, Patients.name AS patient_name
        FROM Appointments
        JOIN Doctors ON Appointments.doctor_id = Doctors.doctor_id
        JOIN Patients ON Appointments.patient_id = Patients.patient_id
        WHERE MONTH(Appointments.appointment_date) = MONTH(CURRENT_DATE())
              AND YEAR(Appointments.appointment_date) = YEAR(CURRENT_DATE())";

$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "<b>Врач:</b> " . $row['doctor_name'] . " - <b>Пациент:</b> " .
$row['patient_name'] . "<br>";
    }
} else {
    echo "Назначений за текущий месяц не найдено.";
}

$conn->close();
?>

```

query2.php

```

<?php
include 'config.php';

$sql = "SELECT Departments.name AS department_name, COUNT(Doctors.doctor_id) AS
doctor_count
        FROM Departments
        LEFT JOIN Doctors ON Departments.department_id = Doctors.department_id
        GROUP BY Departments.name";

$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {

```

```

        echo "<b>Отделение: </b>" . $row['department_name'] . " - <b>Количество врачей:</b>"
    " . $row['doctor_count'] . "<br>";
    }
} else {
    echo "Отделения и врачи не найдены.";
}

$conn->close();
?>

```

query3.php

```

<?php
include 'config.php';

$patient_name = trim($_GET['patient_name'] ?? '');

if (empty($patient_name)) {
    echo "Введите имя пациента.";
} else {
    $stmt = $conn->prepare("SELECT Patients.name AS patient_name, Doctors.name AS
doctor_name,
                        Appointments.appointment_date, Appointments.diagnosis
                        FROM Appointments
                        JOIN Patients ON Appointments.patient_id = Patients.patient_id
                        JOIN Doctors ON Appointments.doctor_id = Doctors.doctor_id
                        WHERE Patients.name = ?");
    $stmt->bind_param("s", $patient_name);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            echo "<b>Пациент:</b>" . $row['patient_name'] . ", <b>Врач:</b> " .
$row['doctor_name'] .
            ", <b>Дата назначения:</b> " . $row['appointment_date'] .
            ", <b>Диагноз:</b> " . $row['diagnosis'] . "<br>";
        }
    } else {
        echo "Назначения для пациента с именем '$patient_name' не найдены.";
    }

    $stmt->close();
}

$conn->close();
?>

```

