

Министерство цифрового развития, связи и
массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего
образования «Сибирский государственный университет телекоммуникаций и
информатики» (СибГУТИ)

Отчет
по расчетно-графической работе
по дисциплине «**Базы данных**»

Выполнил:

студент гр. ИС-142

«__» декабря 2024 г.

/Григорьев Ю.В./

Проверил:

преподаватель

«__» декабря 2024 г.

/Мейкшан В.И./

Оценка « _____ »

Новосибирск 2024

ЗАДАНИЕ

Индивидуальный вариант: Медицинская клиника

В базе данных медицинской клиники хранятся данные о пациентах, врачах, услугах и приемах. У каждого пациента сохраняются его ФИО, дата рождения, контактные данные (телефон, email), а также адрес проживания. В базе данных фиксируются медицинские услуги, предоставляемые клиникой (например, анализы, обследования, консультации), с указанием их стоимости и специализации врача. Врачи клиники имеют свою специализацию (терапевт, кардиолог, стоматолог и т.д.), стаж работы и контактные данные. Каждый врач может вести несколько приёмов, к которым могут быть привязаны несколько пациентов. Прием включает дату и время встречи, врача и пациента, а также медицинские назначения (процедуры или медикаменты). Все назначения связаны с услугами, предоставляемыми врачом. В системе фиксируются не только текущие приёмы, но и история посещений пациентов, что позволяет вести медицинские карты с полным перечнем всех назначенных процедур и рекомендаций.

ВЫПОЛНЕНИЕ РАБОТЫ

Для выполнения данной работы был использован инструментарий **MySQL Workbench + MySQL Shell + MySQL Server**, находящийся в открытом доступе на ресурсах корпорации Oracle.

1. Текстовое описание предметной области

Проектирование базы данных для **Медицинской клиники** включает управление данными о пациентах, врачах, услугах, приемах и назначениях. Врачи с разными специальностями ведут приемы для пациентов, на которых они назначают услуги и процедуры. База данных должна хранить историю посещений пациентов, включая даты приёмов, диагнозы, назначенные услуги, а также данные о врачах, проводивших приёмы.

2. Описание выделенных сущностей

- **Пациенты (Patients):** Хранит информацию о пациентах, включая их ФИО, дату рождения, контактные данные и адрес проживания.
- **Врачи (Doctors):** Хранит данные о врачах, включая их ФИО, специализацию, контактные данные и стаж работы.
- **Услуги (Services):** Хранит информацию о медицинских услугах, которые предоставляются клиникой, включая название услуги, её стоимость и специализацию врача.

- **Приёмы (Appointments):** Хранит данные о приемах пациентов, включая дату, время приёма, врача и пациента.
- **Медицинские назначения (Prescriptions):** Хранит информацию о назначенных услугах или лекарствах в ходе приема пациента.
- **Кабинеты (Rooms):** Хранит информацию о кабинетах, в которых проводятся приемы.

3. Описание атрибутов и их ограничений

- **Patients:**
 - **Patient_ID** (INT): Идентификатор пациента. Первичный ключ.
 - **First_Name** (VARCHAR(50)): Имя пациента. Не допускает NULL.
 - **Last_Name** (VARCHAR(50)): Фамилия пациента. Не допускает NULL.
 - **Date_of_Birth** (DATE): Дата рождения пациента. Не допускает NULL.
 - **Phone** (VARCHAR(20)): Контактный телефон пациента.
 - **Email** (VARCHAR(100)): Электронная почта пациента.
 - **Address** (VARCHAR(255)): Адрес пациента.
- **Doctors:**
 - **Doctor_ID** (INT): Идентификатор врача. Первичный ключ.
 - **First_Name** (VARCHAR(50)): Имя врача. Не допускает NULL.
 - **Last_Name** (VARCHAR(50)): Фамилия врача. Не допускает NULL.
 - **Specialty** (VARCHAR(100)): Специализация врача.
 - **Experience** (INT): Стаж работы врача.
 - **Phone** (VARCHAR(20)): Контактный телефон врача.
 - **Email** (VARCHAR(100)): Электронная почта врача.
- **Services:**
 - **Service_ID** (INT): Идентификатор услуги. Первичный ключ.
 - **Service_Name** (VARCHAR(100)): Название услуги. Не допускает NULL.
 - **Cost** (DECIMAL(10, 2)): Стоимость услуги. Не допускает NULL.
 - **Specialization** (VARCHAR(100)): Специализация, для которой предназначена услуга.
- **Appointments:**
 - **Appointment_ID** (INT): Идентификатор приёма. Первичный ключ.
 - **Patient_ID** (INT): Внешний ключ на пациента.
 - **Doctor_ID** (INT): Внешний ключ на врача.
 - **Room_ID** (INT): Внешний ключ на комнату.
 - **Appointment_Date** (DATE): Дата приёма. Не допускает NULL.

- **Appointment_Time** (TIME): Время приёма. Не допускает NULL.
- **Prescriptions:**
 - **Prescription_ID** (INT): Идентификатор назначения. Первичный ключ.
 - **Appointment_ID** (INT): Внешний ключ на приём.
 - **Service_ID** (INT): Внешний ключ на услугу.
 - **Prescription_Details** (TEXT): Описание назначения.
- **Rooms:**
 - **Room_ID** (INT): Идентификатор кабинета. Первичный ключ.
 - **Room_Number** (VARCHAR(10)): Фактический номер кабинета. Не допускает NULL.
 - **Room_Description** (VARCHAR(100)): Описание кабинета (например, процедурный/операционный/реанимация).

4. Описание выбранных первичных ключей

Простые первичные ключи: В каждой таблице используется уникальный атрибут для идентификации записей, такие как **Patient_ID**, **Doctor_ID**, **Service_ID**, **Appointment_ID**, **Prescription_ID** и **Room_ID**.

5. Описание типов связей между сущностями

- **Пациенты и Приемы:** Связь один-ко-многим. Один пациент может иметь много приёмов.
- **Врачи и Приемы:** Связь один-ко-многим. Один врач может принимать нескольких пациентов.
- **Приёмы и Медицинские назначения:** Связь один-ко-многим. Один прием может иметь несколько назначений.
- **Услуги и Медицинские назначения:** Связь один-ко-многим. Одна услуга может быть назначена в разных приемах.
- **Кабинеты и Приемы:** Связь один-ко-многим. Одна комната может использоваться для разных приемов.

6. Диаграмма "сущность-связь", структура и схема связей таблиц БД

Диаграмма связей таблиц (ER-диаграмма) отражает концептуальную модель. На ней показаны сущности, их атрибуты, первичные ключи и связи между сущностями. Структура таблиц была реализована в MySQL, где каждая сущность имеет свои поля с соответствующими типами данных, ключами и ограничениями. Схема связей между таблицами включает один-ко-многим связи между сущностями. Внешние ключи были определены для поддержания ссылочной целостности между таблицами.

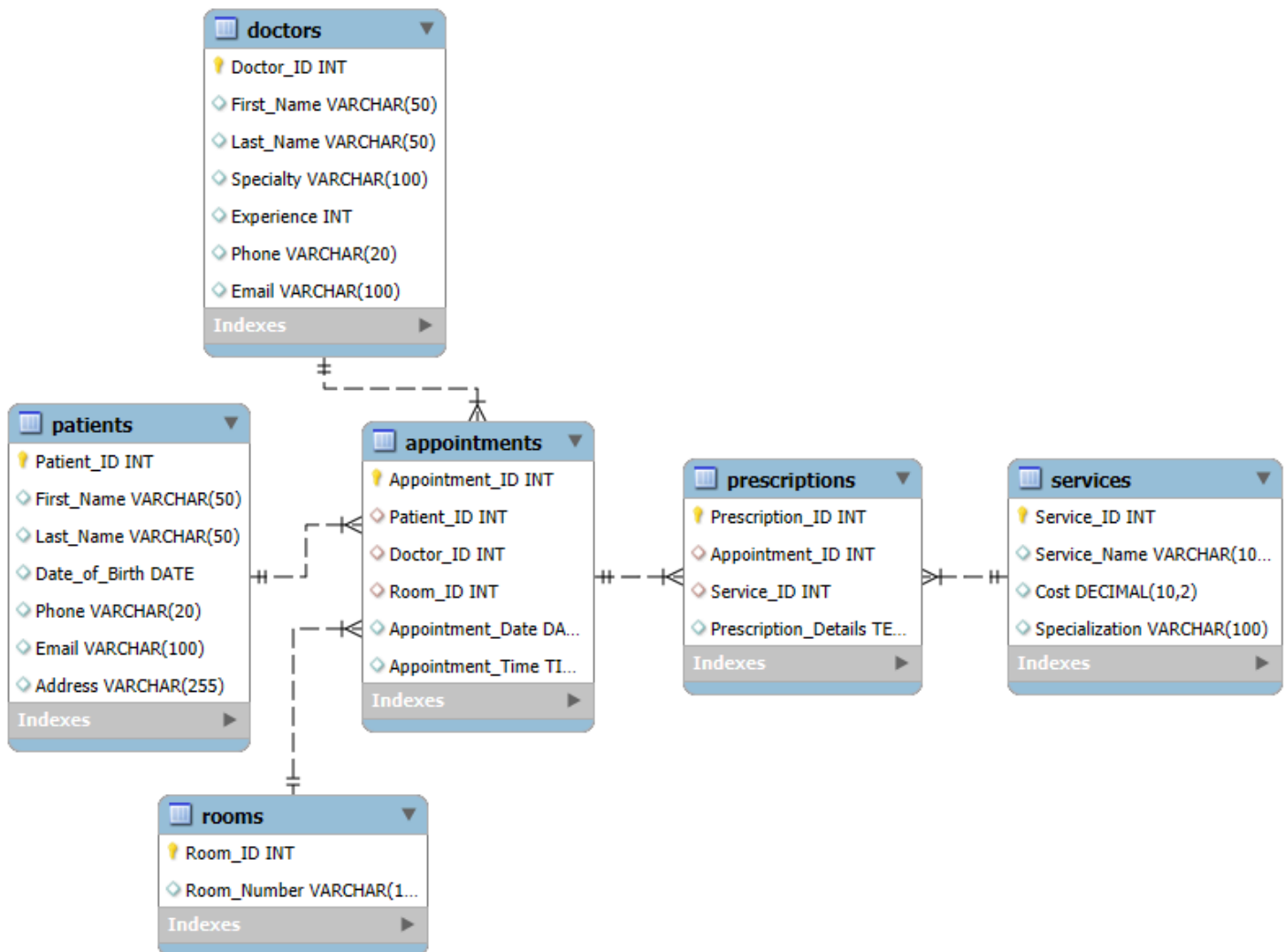


Рисунок 1. Диаграмма связей таблиц БД

7. Описание процесса преобразования концептуальной модели в логическую

Концептуальная модель была преобразована в логическую модель реляционной базы данных путем создания таблиц для каждой сущности с атрибутами. Для

установления связей между сущностями использовались внешние ключи, что обеспечило целостность данных и управление зависимостями между таблицами.

Создание таблиц:

```
CREATE TABLE Patients (  
    Patient_ID INT PRIMARY KEY AUTO_INCREMENT,  
    First_Name VARCHAR(50) NOT NULL,  
    Last_Name VARCHAR(50) NOT NULL,  
    Date_of_Birth DATE NOT NULL,  
    Phone VARCHAR(20),  
    Email VARCHAR(100),  
    Address VARCHAR(255)  
);  
  
CREATE TABLE Doctors (  
    Doctor_ID INT PRIMARY KEY AUTO_INCREMENT,  
    First_Name VARCHAR(50) NOT NULL,  
    Last_Name VARCHAR(50) NOT NULL,  
    Specialty VARCHAR(100) NOT NULL,  
    Experience INT,  
    Phone VARCHAR(20),  
    Email VARCHAR(100)  
);  
  
CREATE TABLE Services (  
    Service_ID INT PRIMARY KEY AUTO_INCREMENT,  
    Service_Name VARCHAR(100) NOT NULL,  
    Cost DECIMAL(10, 2) NOT NULL,  
    Specialization VARCHAR(100)  
);  
  
CREATE TABLE Rooms (  
    Room_ID INT PRIMARY KEY AUTO_INCREMENT,  
    Room_Number VARCHAR(10) NOT NULL,  
    Room_Description VARCHAR(100)  
);  
  
CREATE TABLE Appointments (  
    Appointment_ID INT PRIMARY KEY AUTO_INCREMENT,  
    Patient_ID INT,  
    Doctor_ID INT,  
    Room_ID INT,  
    Appointment_Date DATE NOT NULL,  
    Appointment_Time TIME NOT NULL,  
    FOREIGN KEY (Patient_ID) REFERENCES Patients(Patient_ID) ON DELETE CASCADE,  
    FOREIGN KEY (Doctor_ID) REFERENCES Doctors(Doctor_ID) ON DELETE CASCADE,  
    FOREIGN KEY (Room_ID) REFERENCES Rooms(Room_ID) ON DELETE CASCADE  
);  
  
CREATE TABLE Prescriptions (  
    Prescription_ID INT PRIMARY KEY AUTO_INCREMENT,  
    Appointment_ID INT,  
    Service_ID INT,  
    Prescription_Details TEXT,  
    FOREIGN KEY (Appointment_ID) REFERENCES Appointments(Appointment_ID) ON DELETE CASCADE,  
    FOREIGN KEY (Service_ID) REFERENCES Services(Service_ID) ON DELETE CASCADE  
);
```

8. Описание всех установленных ограничений целостности данных

- Первичные ключи (PK) обеспечивают уникальность записей.
- Внешние ключи (FK) в таблицах **Appointments** и **Prescriptions** обеспечивают целостность ссылок на пациентов, врачей, комнаты и услуги.
- Установлены ограничения на NULL для обязательных полей (например, **First_Name**, **Appointment_Date**).
- Установлено каскадное удаление записей в таблицах **Appointments** и **Prescriptions** в случае удаления из БД записей о конкретном враче, пациенте или приемном кабинете.

9. Содержание таблиц БД

Таблицы были заполнены тестовыми данными, чтобы продемонстрировать работу базы данных:

```
INSERT INTO Patients (First_Name, Last_Name, Date_of_Birth, Phone, Email, Address)
VALUES
('Александр', 'Кузнецов', '1985-03-12', '89067891234', 'kuznetsov@gmail.com', 'ул. Советская, 10'),
('Екатерина', 'Фёдорова', '1992-11-22', '89098765432', 'fedorova@mail.ru', 'ул. Октябрьская, 8'),
('Михаил', 'Петров', '1978-06-03', '89065432198', 'petrov@gmail.com', 'ул. Центральная, 15'),
('Оксана', 'Иванова', '1995-02-25', '89056781234', 'ivanova@gmail.com', 'ул. Ленина, 12'),
('Артем', 'Гришин', '1990-05-18', '89034569876', 'grishin@gmail.com', 'ул. Пушкина, 5'),
('Иван', 'Иванов', '1980-05-14', '89012345678', 'ivanov@gmail.com', 'ул. Ленина, 5'),
('Мария', 'Сидорова', '1990-09-21', '89023456789', 'sidorova@gmail.com', 'ул. Гагарина, 3');
```

```
INSERT INTO Doctors (First_Name, Last_Name, Specialty, Experience, Phone, Email)
VALUES
('Алексей', 'Петров', 'Терапевт', 15, '89034567890', 'petrov@clinic.com'),
('Ольга', 'Смирнова', 'Кардиолог', 12, '89045678901', 'smirnova@clinic.com'),
('Дмитрий', 'Смирнов', 'Офтальмолог', 20, '89056785432', 'smirnov@clinic.com'),
('Анна', 'Васильева', 'Дерматолог', 10, '89067894321', 'vasilieva@clinic.com'),
('Игорь', 'Козлов', 'Хирург', 18, '89065476854', 'kozlov@clinic.com'),
('Наталья', 'Морозова', 'Невролог', 15, '89064567623', 'morozova@clinic.com'),
('Сергей', 'Соколов', 'Уролог', 12, '89087654329', 'sokolov@clinic.com');
```

```
INSERT INTO Rooms (Room_Number)
VALUES
('101'),
('102'),
('103'),
('104'),
('105'),
('106'),
('107'),
('108');
```

```
INSERT INTO Services (Service_Name, Cost, Specialization)
VALUES
('Общий анализ крови', 500.00, 'Терапевт'),
('ЭКГ', 1500.00, 'Кардиолог'),
('Массаж спины', 2000.00, 'Невролог'),
('УЗИ брюшной полости', 1800.00, 'Уролог'),
('Консультация хирурга', 1500.00, 'Хирург'),
('Приём дерматолога', 1200.00, 'Дерматолог'),
('Диагностика зрения', 800.00, 'Офтальмолог');
```

```

INSERT INTO Appointments (Patient_ID, Doctor_ID, Room_ID, Appointment_Date, Appointment_Time)
VALUES
(1, 1, 1, '2024-10-15', '09:00:00'),
(2, 2, 2, '2024-10-16', '11:00:00'),
(3, 3, 3, '2024-10-17', '12:00:00'),
(4, 4, 4, '2024-10-17', '09:30:00'),
(5, 5, 5, '2024-10-17', '14:00:00'),
(6, 1, 1, '2024-10-18', '13:00:00'),
(7, 2, 2, '2024-10-18', '10:00:00'),
(1, 3, 3, '2024-10-18', '11:00:00'),
(2, 4, 4, '2024-10-18', '15:00:00'),
(3, 5, 5, '2024-10-19', '16:00:00');

```

```

INSERT INTO Prescriptions (Appointment_ID, Service_ID, Prescription_Details)
VALUES
(1, 1, 'Массаж спины для лечения хронической боли.'),
(2, 2, 'УЗИ брюшной полости для диагностики.'),
(3, 3, 'Консультация хирурга перед операцией.'),
(4, 4, 'Приём дерматолога для лечения кожных заболеваний.'),
(5, 5, 'Диагностика зрения для определения очков.'),
(6, 1, 'Повторный массаж для закрепления результата.'),
(7, 2, 'Контрольное УЗИ после лечения.'),
(8, 3, 'Повторная консультация хирурга.'),
(9, 4, 'Дерматологическое обследование.'),
(10, 5, 'Проверка зрения для подбора линз.');
```

В режиме отображения данных:

Doctors

Doctor_ID	First_Name	Last_Name	Specialty	Experience	Phone	Email
1	Алексей	Петров	Терапевт	15	89034567890	petrov@clinic.com
2	Ольга	Смирнова	Кардиолог	12	89045678901	smirnova@clinic.com
3	Дмитрий	Смирнов	Офтальмолог	20	89056785432	smirnov@clinic.com
4	Анна	Васильева	Дерматолог	10	89067894321	vasilieva@clinic.com
5	Игорь	Козлов	Хирург	18	89065476854	kozlov@clinic.com
6	Наталья	Морозова	Невролог	15	89064567623	morozova@clinic.com
7	Сергей	Соколов	Уролог	12	89087654329	sokolov@clinic.com

Patients

Patient_ID	First_Name	Last_Name	Date_of_Birth	Phone	Email	Address
1	Александр	Кузнецов	1985-03-12	89067891234	kuznetsov@gmail.com	ул. Советская, 10
2	Екатерина	Фёдорова	1992-11-22	89098765432	fedorova@mail.ru	ул. Октябрьская, 8
3	Михаил	Петров	1978-06-03	89065432198	petrov@gmail.com	ул. Центральная, 15
4	Оксана	Иванова	1995-02-25	89056781234	ivanova@gmail.com	ул. Ленина, 12
5	Артем	Гришин	1990-05-18	89034569876	grishin@gmail.com	ул. Пушкина, 5
6	Иван	Иванов	1980-05-14	89012345678	ivanov@gmail.com	ул. Ленина, 5
7	Мария	Сидорова	1990-09-21	89023456789	sidorova@gmail.com	ул. Гагарина, 3

Rooms

Room_ID	Room_Number	Room_Description
1	101	NULL
2	102	NULL
3	103	NULL
4	104	NULL
5	105	NULL
6	106	NULL
7	107	NULL
8	108	NULL

Services

Service_ID	Service_Name	Cost	Specialization
1	Общий анализ крови	500.00	Терапевт
2	ЭКГ	1500.00	Кардиолог
3	Массаж спины	2000.00	Невролог
4	УЗИ брюшной полости	1800.00	Уролог
5	Консультация хирурга	1500.00	Хирург
6	Приём дерматолога	1200.00	Дерматолог
7	Диагностика зрения	800.00	Офтальмолог

Appointments

Appointment_ID	Patient_ID	Doctor_ID	Room_ID	Appointment_Date	Appointment_Time
1	1	1	1	2024-10-15	09:00:00
2	2	2	2	2024-10-16	11:00:00
3	3	3	3	2024-10-17	12:00:00
4	4	4	4	2024-10-17	09:30:00
5	5	5	5	2024-10-17	14:00:00
6	6	1	1	2024-10-18	13:00:00
7	7	2	2	2024-10-18	10:00:00
8	1	3	3	2024-10-18	11:00:00
9	2	4	4	2024-10-18	15:00:00
10	3	5	5	2024-10-19	16:00:00

Prescriptions

Prescription_ID	Appointment_ID	Service_ID	Prescription_Details
1	1	1	Массаж спины для лечения хронической боли.
2	2	2	УЗИ брюшной полости для диагностики.
3	3	3	Консультация хирурга перед операцией.
4	4	4	Приём дерматолога для лечения кожных заболеваний.
5	5	5	Диагностика зрения для определения очков.
6	6	1	Повторный массаж для закрепления результата.
7	7	2	Контрольное УЗИ после лечения.
8	8	3	Повторная консультация хирурга.
9	9	4	Дерматологическое обследование.
10	10	5	Проверка зрения для подбора линз.

10. Работа с полученной базой данных

Для демонстрации были разработаны тестовые запросы на языке SQL, чтобы показать работоспособность БД.

Например:

Получить список всех врачей, у которых есть приемы в кабинете 104, с датами приема

```
SELECT D.First_Name, D.Last_Name, R.Room_Number, A.Appointment_Date
FROM Appointments A
JOIN Doctors D ON A.Doctor_ID = D.Doctor_ID
JOIN Rooms R ON A.Room_ID = R.Room_ID
WHERE R.Room_Number = '104';
```

First_Name	Last_Name	Room_Number	Appointment_Date
Анна	Васильева	104	2024-10-17
Анна	Васильева	104	2024-10-18

Список всех назначенных услуг с указанием врачей и пациентов

```
SELECT P.Prescription_Details, S.Service_Name, D.First_Name AS Doctor_First_Name, D.Last_Name AS Doctor_Last_Name,
Pt.First_Name AS Patient_First_Name, Pt.Last_Name AS Patient_Last_Name
FROM Prescriptions P
JOIN Services S ON P.Service_ID = S.Service_ID
JOIN Appointments A ON P.Appointment_ID = A.Appointment_ID
JOIN Doctors D ON A.Doctor_ID = D.Doctor_ID
JOIN Patients Pt ON A.Patient_ID = Pt.Patient_ID;
```

Prescription_Details	Service_Name	Doctor_First_Name	Doctor_Last_Name	Patient_First_Name	Patient_Last_Name
Массаж спины для лечения хронической боли.	Общий анализ крови	Алексей	Петров	Александр	Кузнецов
Повторный массаж для закрепления результата.	Общий анализ крови	Алексей	Петров	Иван	Иванов
УЗИ брюшной полости для диагностики.	ЭКГ	Ольга	Смирнова	Екатерина	Фёдорова
Контрольное УЗИ после лечения.	ЭКГ	Ольга	Смирнова	Мария	Сидорова
Консультация хирурга перед операцией.	Массаж спины	Дмитрий	Смирнов	Михаил	Петров
Повторная консультация хирурга.	Массаж спины	Дмитрий	Смирнов	Александр	Кузнецов
Приём дерматолога для лечения кожных заболеваний.	УЗИ брюшной полости	Анна	Васильева	Оксана	Иванова
Дерматологическое обследование.	УЗИ брюшной полости	Анна	Васильева	Екатерина	Фёдорова
Диагностика зрения для определения очков.	Консультация хирурга	Игорь	Козлов	Артем	Гришин
Проверка зрения для подбора линз.	Консультация хирурга	Игорь	Козлов	Михаил	Петров

Список всех приемов конкретного пациента (история болезни)

```
SELECT
    A.Appointment_Date,
    A.Appointment_Time,
    D.First_Name AS Doctor_First_Name,
    D.Last_Name AS Doctor_Last_Name,
    Ptn.First_Name AS Patient_First_Name,
    Ptn.Last_Name AS Patient_Last_Name,
    S.Service_Name
FROM Appointments A
JOIN Doctors D ON A.Doctor_ID = D.Doctor_ID
JOIN Prescriptions P ON A.Appointment_ID = P.Appointment_ID
JOIN Services S ON P.Service_ID = S.Service_ID
JOIN Patients Ptn ON A.Patient_ID = Ptn.Patient_ID
WHERE A.Patient_ID = 2;
```

Appointment_Date	Appointment_Time	Doctor_First_Name	Doctor_Last_Name	Patient_First_Name	Patient_Last_Name	Service_Name
2024-10-16	11:00:00	Ольга	Смирнова	Екатерина	Фёдорова	ЭКГ
2024-10-18	15:00:00	Анна	Васильева	Екатерина	Фёдорова	УЗИ брюшной полости

Список всех назначений конкретного врача (например, с Doctor_ID = 2)

```
SELECT
    P.Prescription_Details,
    S.Service_Name,
    A.Appointment_Date,
    Ptn.First_Name AS Patient_First_Name,
    Ptn.Last_Name AS Patient_Last_Name,
    D.First_Name AS Doctor_First_Name,
    D.Last_Name AS Doctor_Last_Name
FROM Prescriptions P
JOIN Services S ON P.Service_ID = S.Service_ID
JOIN Appointments A ON P.Appointment_ID = A.Appointment_ID
JOIN Patients Ptn ON A.Patient_ID = Ptn.Patient_ID
JOIN Doctors D ON A.Doctor_ID = D.Doctor_ID
WHERE A.Doctor_ID = 2;
```

Prescription_Details	Service_Name	Appointment_Date	Patient_First_Name	Patient_Last_Name	Doctor_First_Name	Doctor_Last_Name
УЗИ брюшной полости для диагностики.	ЭКГ	2024-10-16	Екатерина	Фёдорова	Ольга	Смирнова
Контрольное УЗИ после лечения.	ЭКГ	2024-10-18	Мария	Сидорова	Ольга	Смирнова

Проверка целостности данных: удаление из БД конкретного пациента (например, с Patient_ID = 2) и проверка связанных таблиц на наличие записей о нем (приемы, назначения)

```
-- Удаление пациента с Patient_ID = 2
DELETE FROM Patients
WHERE Patient_ID = 2;

-- Проверяем каскадное удаление записей в связанных таблицах
SELECT * FROM Appointments
WHERE Patient_ID = 2;

SELECT * FROM Prescriptions P
JOIN Appointments A ON P.Appointment_ID = A.Appointment_ID
WHERE A.Patient_ID = 2;
```

Appointment_ID	Patient_ID	Doctor_ID	Room_ID	Appointment_Date	Appointment_Time
NULL	NULL	NULL	NULL	NULL	NULL

Prescription_ID	Appointment_ID	Service_ID	Prescription_Details	Appointment_ID	Patient_ID	Doctor_ID	Room_ID	Appointment_Date	Appointr
-----------------	----------------	------------	----------------------	----------------	------------	-----------	---------	------------------	----------

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была успешно спроектирована и реализована база данных для медицинской клиники, включающей таблицы для управления данными о пациентах, врачах, услугах, приемах и медицинских назначениях. На всех этапах разработки была соблюдена целостность данных с использованием первичных и внешних ключей.

Основные этапы выполнения работы включали:

1. **Концептуальное проектирование:** Выделены основные сущности (пациенты, врачи, услуги, приёмы, назначения, комнаты) и связи между ними, что позволило построить логическую модель базы данных.
2. **Логическое проектирование:** Модель данных была преобразована в реляционную структуру с созданием таблиц, связей и атрибутов. Были определены первичные ключи для каждой сущности, а также внешние ключи для обеспечения связей между таблицами.
3. **Реализация в СУБД:** В MySQL созданы таблицы с полным описанием атрибутов, типов данных, а также ограничениями на значения полей (например, ограничения NOT NULL). Для обеспечения целостности данных добавлены каскадные удаления, которые позволяют автоматически удалять связанные записи при удалении основной записи.
4. **Заполнение базы данных:** Таблицы базы данных были заполнены тестовыми данными, включая записи о пациентах, врачах, услугах, приемах и назначениях.
5. **Диаграмма связей:** Была построена ER-диаграмма, отображающая сущности и связи между ними.

Дополнительное внимание было уделено целостности данных за счёт использования каскадного удаления. Это позволило избежать проблем с "висячими" данными при удалении записей из связанных таблиц. Все связи типа "один ко многим" были установлены через внешние ключи, что гарантирует корректное управление данными в базе данных.

Результатом работы является полнофункциональная база данных, обеспечивающая эффективное управление медицинскими записями в клинике, которая готова к использованию в реальных условиях или может быть дополнительно расширена для удовлетворения более сложных потребностей клиники.