

ИТОГОВАЯ РАБОТА

БАЗА ДАННЫХ: ПОЛИКЛИНИКА

ИВАНОВ ВЛАДИМИР ВЛАДИМИРОВИЧ

ГР. БД-131

22.06.2024

ВВЕДЕНИЕ

- **Цель работы:**

Продemonстрировать навыки проектирования и работы с базами данных.

Применить полученные знания на практике.

- **Важность SQL:**

SQL (Structured Query Language) — это стандартный язык для работы с реляционными базами данных.

Широко используется в разработке приложений и анализе данных.

О П И С А Н И Е П Р О Е К Т А

- **Задача:**

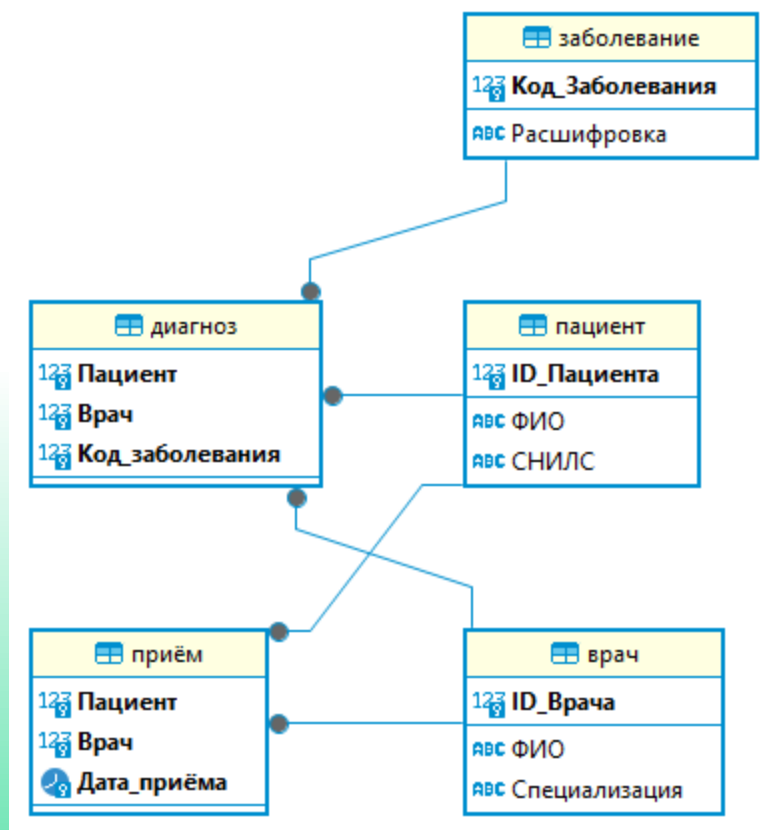
Создание базы данных для управления клиникой.

- **Основные требования:**

Хранение информации о пациентах, врачах, заболеваниях, приёмах и диагнозах.

Обеспечение целостности и надёжности данных.

СХЕМА БАЗЫ ДАННЫХ:



РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ

- **Создание базы данных и таблиц:**

```
CREATE DATABASE clinic;
```

```
USE clinic;
```

```
CREATE TABLE ПАЦИЕНТ (
```

```
    ID_Пациента INT NOT NULL AUTO_INCREMENT,
```

```
    ФИО VARCHAR(100) NOT NULL,
```

```
    СНИЛС VARCHAR(50) NOT NULL,
```

```
    PRIMARY KEY (ID_Пациента)
```

```
);
```

```
CREATE TABLE ВРАЧ (  
    ID_Врача INT NOT NULL AUTO_INCREMENT,  
    ФИО VARCHAR(100) NOT NULL,  
    Специализация VARCHAR(50) NOT NULL,  
    PRIMARY KEY (ID_Врача)  
);
```

```
CREATE TABLE ЗАБОЛЕВАНИЕ (  
    Код_Заболевания INT NOT NULL AUTO_INCREMENT,  
    Расшифровка VARCHAR(255) NOT NULL,  
    PRIMARY KEY (Код_Заболевания)  
);
```

```
CREATE TABLE ПРИЁМ (  
    Пациент INT NOT NULL,  
    Врач INT NOT NULL,  
    Дата_приёма DATETIME NOT NULL,  
    FOREIGN KEY (Пациент) REFERENCES ПАЦИЕНТ(ID_Пациента),  
    FOREIGN KEY (Врач) REFERENCES ВРАЧ(ID_Врача),  
    PRIMARY KEY (Пациент, Врач, Дата_приёма)  
);  
  
CREATE TABLE ДИАГНОЗ (  
    Пациент INT NOT NULL,  
    Врач INT NOT NULL,  
    Код_заболевания INT NOT NULL,  
    FOREIGN KEY (Пациент) REFERENCES ПАЦИЕНТ(ID_Пациента),  
    FOREIGN KEY (Врач) REFERENCES ВРАЧ(ID_Врача),  
    FOREIGN KEY (Код_заболевания) REFERENCES ЗАБОЛЕВАНИЕ(Код_Заболевания),  
    PRIMARY KEY (Пациент, Врач, Код_заболевания)  
);
```

ПРИМЕРЫ ЗАПРОСОВ

- **Запрос с использованием WHERE, AND, OR**

SELECT *

FROM ПРИЁМ

WHERE (Дата_приёма BETWEEN '2024-06-18' AND '2024-06-22')

AND (Врач = 1 OR Врач = 3);

Этот запрос выбирает все записи из таблицы ПРИЁМ, где дата приема находится в интервале между 18 и 22 июня 2024 года, и врач имеет идентификатор 1 или 3.

- **Запрос с использованием BETWEEN, ORDER BY**

SELECT *

FROM ПАЦИЕНТ

WHERE СНИЛС BETWEEN '234-567-890 12' AND '567-890-123 45'

ORDER BY ФИО;

Этот запрос выбирает все записи из таблицы ПРИЁМ, где дата приема находится в интервале между 18 и 22 июня 2024 года, и врач имеет идентификатор 1 или 3.

- **Запрос с использованием GROUP BY и агрегатной функции**

```
SELECT Врач, COUNT(*) AS Количество_приёмов
```

```
FROM ПРИЁМ
```

```
GROUP BY Врач;
```

Этот запрос подсчитывает количество приемов для каждого врача. Результаты группируются по полю Врач, и для каждой группы выводится количество приемов.

- **Запрос с подзапросом и ANY или ALL**

```
SELECT *
```

```
FROM ПАЦИЕНТ
```

```
WHERE ID_Пациента IN (
```

```
    SELECT Пациент
```

```
    FROM ДИАГНОЗ
```

```
    WHERE Код_заболевания > ALL (
```

```
        SELECT Код_Заболевания
```

```
        FROM ЗАБОЛЕВАНИЕ
```

```
        WHERE Код_Заболевания >= 5
```

```
    )
```

```
);
```

Этот запрос выбирает всех пациентов, которые имеют диагноз с кодом заболевания больше, чем все коды заболеваний, начинающиеся с 5 и выше. Это сложный запрос, использующий подзапросы для фильтрации данных.

- Объединенные запросы с использованием UNION/UNION ALL

```
SELECT ID_Пациента AS Идентификатор, ФИО, 'Пациент' AS Тип  
FROM ПАЦИЕНТ  
  
UNION  
  
SELECT ID_Врача, ФИО, 'Врач'  
FROM ВРАЧ;
```

Этот запрос объединяет результаты двух запросов: один из таблицы ПАЦИЕНТ и другой из таблицы ВРАЧ. В результате выводится общий список, включающий идентификатор, ФИО и тип (Пациент или Врач).

- Запрос с использованием LIKE

```
SELECT *  
FROM ЗАБОЛЕВАНИЕ  
WHERE Расшифровка LIKE 'Г%';
```

Этот запрос выбирает все записи из таблицы ЗАБОЛЕВАНИЕ, где расшифровка заболевания начинается с буквы "Г".

- **Запрос с использованием CASE**

```
SELECT *,  
  
    CASE  
  
        WHEN Дата_приёма > NOW() THEN 'Предстоящий прием'  
  
        ELSE 'Прошедший прием'  
  
    END AS Статус_приёма  
  
FROM ПРИЁМ;
```

Этот запрос выбирает все записи из таблицы ПРИЁМ и добавляет столбец Статус_приёма, который указывает, является ли прием предстоящим или прошедшим в зависимости от текущей даты и времени.

ПРИМЕРЫ ОПЕРАЦИЙ

- Пример INSERT

INSERT INTO ПАЦИЕНТ (ФИО, СНИЛС)

VALUES ('Зайцева Екатерина Владимировна', '123-123-123 45');

- Пример UPDATE

UPDATE ВРАЧ

SET Специализация = 'Отоларинголог'

WHERE ID_Врача = 2;

- Пример DELETE

DELETE FROM ПРИЁМ

WHERE Дата_приёма < NOW();

ЗАКЛЮЧЕНИЕ

- **Итоги работы:**

Создана и заполнена база данных для клиники.

Выполнены различные SQL-запросы для анализа и управления данными.

- **Перспективы:**

Возможность расширения функционала базы данных.

Применение полученных знаний в реальных проектах.

ВОПРОСЫ И ОТВЕТЫ

СПАСИБО ЗА ВНИМАНИЕ!