

Московская финансово-юридическая академия филиал московского
финансово-юридического института МФЮА г.Ярославля

**Практическая работа №2 по дисциплине “Основы
проектирования баз данных”**

Выполнил:

Студент 2 курса группы 31ИПо8381

Корпачёв Дмитрий Денисович

Руководитель:

Фурмавнин Станислав Анатольевич

Ярославль

2025

Тема работы: Простейшие SQL-запросы. DDL.

Цель работы: Научиться создавать базу данных в СУБД MS SQL Management Studio/PostgreSQL/MySQL согласно схеме, созданной в проекте. Научиться строить SQL-запросы CREATE, ALTER и TABLE.

Основная часть:

Теоретические сведения:

Используемые в ходе работы команды:

CREATE – Создать объект БД (таблицу, БД, и т.д.)

TABLE – Объект - таблица, используется с CREATE, ALTER, DROP

DATABASE – Объект - база данных, используется с CREATE, DROP

USE – Выбрать текущую базу данных.

BEGIN, END, GO – Текущий блок кода

GO – Разделитель пакетов (MSSQL)

IF NOT EXIST – Выполнить, если объект не существует

WHERE – “Где” искать объект

CHECK – Ограничения на допустимые значения в столбце

DEFAULT – Значение по умолчанию для столбца

SELECT – Выборка данных/ Выбрать данные

FROM – Указать источник данных (таблицу)

ALTER – Изменить объект

ADD CONSTRAINT – Добавить ограничения/е к таблице

PRIMARY KEY – Уникальный идентификатор записи

FOREIGN KEY – Ссылка на первичный ключ в другой таблице.

REFERENCES – Указать, на какую таблицу/столбец ссылается внешний ключ.

DROP – Удалить объект БД

Структура Базы Данных:

1. Каждый файл начинается с:

```
IF NOT EXISTS (SELECT * FROM sys.databases WHERE [name] = 'Hospital')
BEGIN
    CREATE DATABASE Hospital
END
GO
USE Hospital;
GO
```

Это обусловлено тем, что если не найдена База Данных - она создаётся.

2. Первый файл к первому заданию:

I. Определим Сущности:

1. Пациенты
2. Специальности
3. Категории
4. Диагнозы
5. Доктора
6. Посещения
7. Медицинские Карты
8. Записи в Медицинские Карты

II. Определим связи между сущностями

1. 1:1 Пациент и медкарта (К 1 пациенту относится 1 медкарта)
2. N:N Медкарта и Диагноз (Множество медкарт относится к множеству диагнозов)
3. N:N Прием и Пациент (Множество Приемов ко Множеству Пациентов)
4. 1:N Врач и диагноз (1 Врач ко множеству диагнозов)
5. 1:1 Врач и Категория (1 Врач к 1 Категории)
6. 1:1 Врач к Специальности (1 Врач к 1 Специальности)
7. N:N Врач к Приему (Множество Врачей ко Множеству Приемов)

III. Определим атрибуты и типы данных для каждой сущности:

1) Атрибуты Пациенты:

- Id Пациента INT

- Фιο пациента VARCHAR(255)
- Дата Рождения Пациента DATE
- Пол Пациента VARCHAR(1)
- Скидка DECIMAL (5,2)
- Адрес Пациента VARCHAR(255)

2) Атрибуты Специальность:

- Id Специальности INT
- Специальность VARCHAR(255)

3) Атрибуты Категории:

- Номер Категории INT
- Название Категории VARCHAR(255)
- Цена Визита DECIMAL(10,2)

4) Атрибуты Диагноза:

- Код Диагноза INT
- Название Диагноза VARCHAR(255)

5) Атрибуты Доктора:

- Id доктора INT
- Фιο VARCHAR(255)
- Id категории INT
- Номер Специальности INT

6) Атрибуты Приема:

- Id Приема INT
- Id Доктора INT
- Id Пациента INT
- Причина Обращения VARCHAR(255)
- Дата Обращения DATE
- Номер Купона INT
- Стоимость Визита DECIMAL(10,2)
- Категория Доктора INT

7) Атрибуты Медицинской карты:

- Id Мед Карты INT
- Id Пациента INT
- Дата Посещения DATE
- Дата открытия Мед Карты DATE
- Код Диагноза INT
- Id Диагноза INT

8) Записи в медкарту:

- Номер Записи INT
- Id Медкарты INT
- Id Пациента INT
- Код Диагноза INT
- Фιο Врача VARCHAR(255)

Можно сказать, что это полностью описывает структуру второго файла(создание таблиц без ограничений). Но задача первого задания - создать таблицы, на которые наложены некоторые ограничения.

1. Во всех случаях, где стоит INT мы добавляем ограничение:

```
CHECK("column name" >=0 AND " column name " <= 9999999)
```

Что позволяет нам сказать, что это замена “NOT NULL”

2. Так же мы можем добавить инкремент, для того, что бы в будущем, при добавлении РК упростит нам процесс внесения данных в БД. Авто-Инкремент можно добавить с помощью:

```
<TYPE DATE> ("точка отсчёта", "шаг") ...
```

!Поправка! - невозможно добавить вместе с нашим “NOT NULL” ещё одно ограничение, но, его можно будет написать в файле №3

3. Добавим ограничение на столбец “Пол Пациента” с помощью:

```
CHECK ("column name" IN('M', 'F')) DEFAULT 'M'
```

Это ограничение похоже на ENUM из MYSQL, он удобен, но в MSSQL его нет

На этом описание структуры и синтаксиса первого файла завершена. Перейдём к третьему, так как второе задание подразумевает под собой создание таблиц без ограничений целостности. Одна пометка - при написании скрипта для второго скрипта - ко всем именам была добавлена приписка "NoConstraints" во избежание проблем с целостностью таблиц

Файл №3.

Задача для третьего задания - Написать скрипт добавления связей между таблицами и ограничений целостности в них с использованием ALTER.

1. Мы имеем файл №1 как основа, мы можем очень просто добавить нужные РК к каждому из сущностей. Структура запроса выглядит так:

```
ALTER TABLE "table name" ADD CONSTRAINT "constraint name"  
PRIMARY KEY ("column name");
```

2. Далее определим внешние ключи для таблиц. Это позволит нам "связать" таблицы. Ограничения добавляются таким образом:

```
ALTER TABLE "table name"  
  
ADD CONSTRAINT "constraint name" FOREIGN KEY ("column name")  
REFERENCES "table name"("column name");
```

3. И, стоит вспомнить примечание из первого файла. Нам стоит добавить авто-инкремент. Мы можем добавить его вместе с РК, что немного облегчит нам написание кода. Структура написания запроса:

```
ALTER TABLE "table name" ADD "column name" INT IDENTITY(1, 1)  
PRIMARY KEY;
```

3.1 Я не смог разобраться с identity и, кажется, я не знаю, как добавить автоинкремент вместе с ограничением РК, поэтому, identity не было добавлено в код.

Файл №4

В этом файле все таблицы, которые мы создавали в файлах №1,2 удаляются с помощью DROP TABLE.

Вывод: Мы научились создавать базу данных в СУБД MS SQL Management Studio/PostgreSQL/MySQL согласно схеме, созданной в проекте. Научились строить SQL-запросы CREATE, ALTER и TABLE. А так же, мы немного увеличили уровень практики, и попробовали писать и структурировать запросы вместе с ограничениями.