

Практическая работа 52

Создание, настройка и заполнение таблиц в новой базе данных

Цель работы: Получить практический опыт создания, настройки и заполнения таблиц в новой базе данных.

Перечень оборудования и программного обеспечения

- Персональный компьютер
- Microsoft Office (Word, Visio, Access)
- MSSQLServer 2008 R2

Краткие теоретические сведения

Создание баз данных в среде MS SQL Server

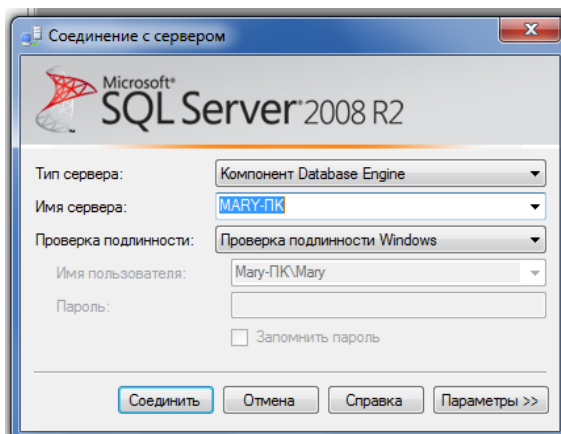
Базы данных составляют основу для построения информационных систем любого масштаба и предназначения. В теории баз данных одними из основных являются вопросы, связанные с анализом предметной области и моделированием структуры данных, управлением данными и их анализом.

Основой любой базы данных является реализованная в ней модель данных, представляющая собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и существующие между ними связи.

В реляционной базе данных данные представлены в виде собрания таблиц. Таблица состоит из определенного числа столбцов (полей) и произвольного числа строк (записей).

Начало работы в Microsoft SQL Server Management Studio

Для создания баз данных используем среду Microsoft SQL Server Management Studio.



На запрос соединения с сервером выбираем:

Тип сервера: **Компонент Database Engine**

Имя сервера: _____.

Под таким именем доступна машина, на которой установлены серверные компоненты MS SQL Server 2008. Можно попробовать выбрать сервер из выпадающего списка серверов. Можно также обратиться к этой машине по IP-адресу 192.168._____ из локальной сети.

Проверка подлинности: **Проверка подлинности Windows.**

Такая настройка позволяет создавать пользователей данного экземпляра SQL Server независимо от компьютера, с которого производится вход.

Нажимаем кнопку «Соединить».

Примечание. В случае, когда права доступа пользователя не ограничены (как в рассматриваемом случае), вкладку Параметры можно не открывать. Если же пользователь имеет доступ только к определенным базам данных, при подключении к серверу нужно одну из этих баз указывать.

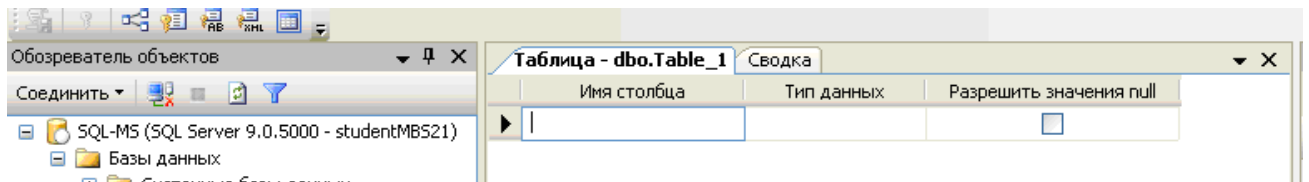
Создание таблиц базы данных в среде Microsoft SQL Server Management Studio

Начнем с создания таблицы Speciality. Структура таблицы приведена ниже:

Имя поля (столбца)	Содержание	Тип данных	Возможность содержать NULL
Num	Первичный ключ	int	нет
Name	Название специальности	varchar(60)	нет

В реляционных базах данных первичный ключ используется как уникальный идентификатор записи. Это поле является обязательным, оно используется для связи таблиц по внешним ключам (примеры такого связывания будут рассмотрены далее). Первичный ключ должен иметь целочисленный тип (в данном случае – *int*). Во втором поле будет храниться название специальности – некоторая строка, поэтому мы выбираем для этого поля тип *varchar(60)*. Число в скобках означает максимальное число символов в строке. Детальную информацию об этих типах можно посмотреть в справке.

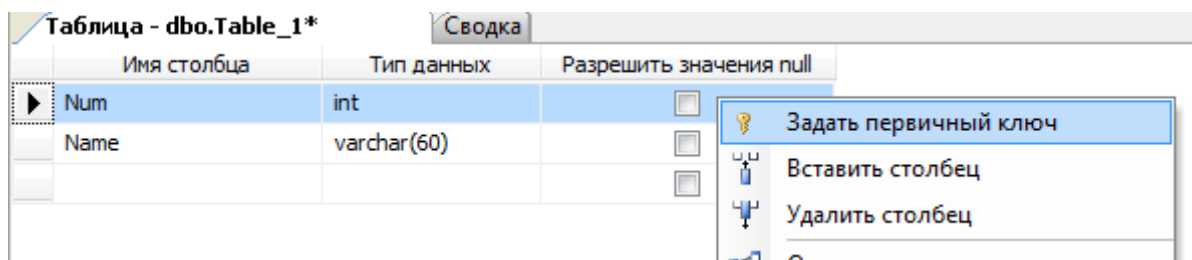
Простейшим образом можно создавать таблицы средствами MS SQL Server Management Studio (правая кнопка мыши на заголовке «Таблицы» > Создать таблицу.). Получаем следующее:



Вводим имя первого столбца Num (первичный ключ – в том столбце хранится номер записи), выбираем из выпадающего списка тип данных int. Первичный ключ не может быть пустым, поэтому и оставляем неотмеченным поле «Разрешить значения null». Затем аналогичным образом вводим имя второго столбца, задаем тип, запрещаем полю иметь значение null. Таблица принимает следующий вид:

Имя столбца	Тип данных	Разрешить значения null
Num	int	<input type="checkbox"/>
Name	varchar(60)	<input type="checkbox"/>

Теперь необходимо указать, что поле Num будет являться первичным ключом. Правой кнопкой мыши щелкаем по этому полю и выбираем «Задать первичный ключ»:

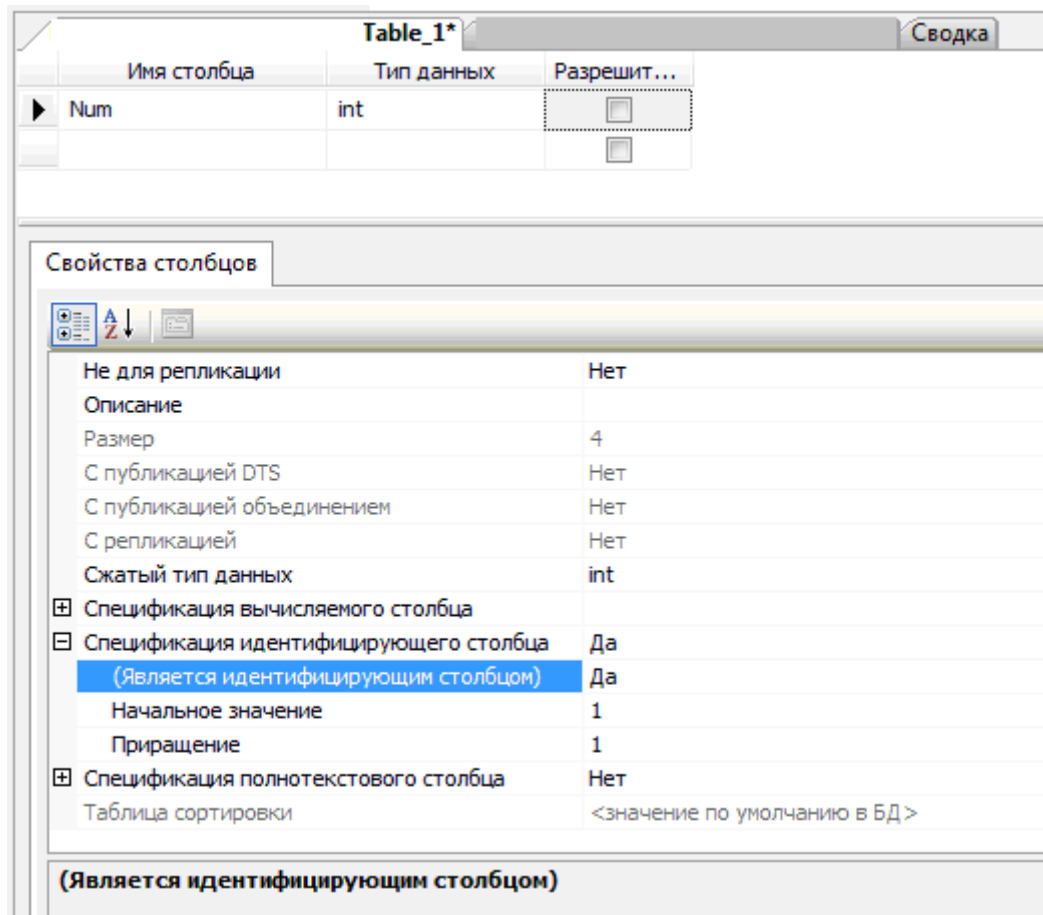


Сохраняем таблицу под именем Speciality (после этого таблица должна появиться в обозревателе объектов). Теперь можно перейти к заполнению этой таблицы (для этого нужно в обозревателе объектов выбрать эту таблицу и в контекстном меню нажать «Открыть таблицу»):

Num	Name
1	Математика
2	Информатика
3	Физика
NULL	NULL

При заполнении вы обнаружите, что каждый раз приходится вводить не только полезную информацию (название специальности), но и номер записи. Чтобы вводить номер записи автоматически, нужно задать спецификацию идентифицирующего столбца. Для этого необходимо в

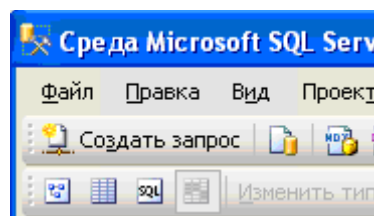
свойствах столбца указать, что данный столбец является идентифицирующим:



4. Создание таблиц базы данных с помощью SQL-запроса

Создание таблиц в графическом режиме, безусловно, удобно, однако не универсально. При использовании других средств разработки баз данных (например, IBM DB2) придется привыкать к новым приемам работы. Использование конструкций языка SQL позволяет работать с базами данных, исходя из единого подхода, в любой среде управления базами данных.

Выберите на панели инструментов «Создать запрос»:



Создадим новую базу данных запросом. Напишем `CREATEDATABASE имя_базы` и нажмем F5. В обозревателе объектов должна появиться новая база (если сразу не появилась, то надо выделить мышью раздел «Базы данных» и в контекстном меню выбрать «Обновить»).

Теперь создадим таблицу Speciality. Упрощенный синтаксис создания таблиц следующий:

```
CREATE TABLE<имя таблицы> (  
<имя столбца 1><тип данных> [NOT NULL] [DEFAULT <значение по  
умолчанию>],  
<имя столбца 2><тип данных> [NOT NULL] [DEFAULT <значение по  
умолчанию>],  
...  
)
```

Задания

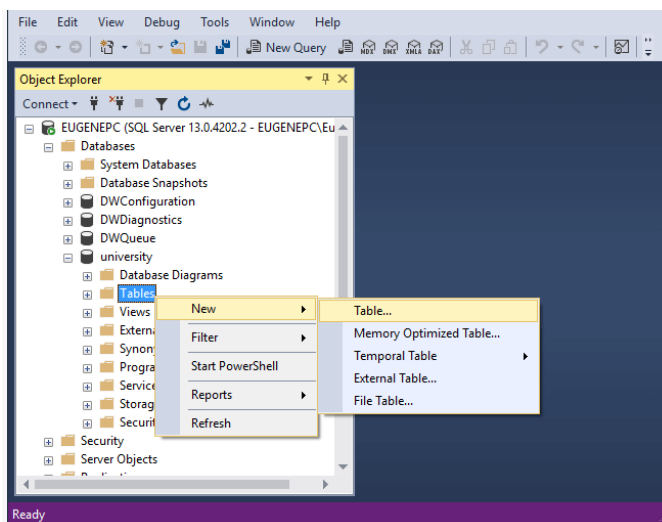
- 1 Изучить теоретические сведения.
- 2 В соответствии с вариантом задания выполнить операции созданию, настройке и заполнению таблиц:
 - одну из таблиц создать графическим способом;
 - остальные с помощью скрипта на языке SQL.
- 3 Создать диаграмму базы данных.

Порядок выполнения работы

Аналогично базам данных для создания таблицы в SQL Server Management Studio можно применить скрипт на языке SQL, либо воспользоваться графическим дизайнером.

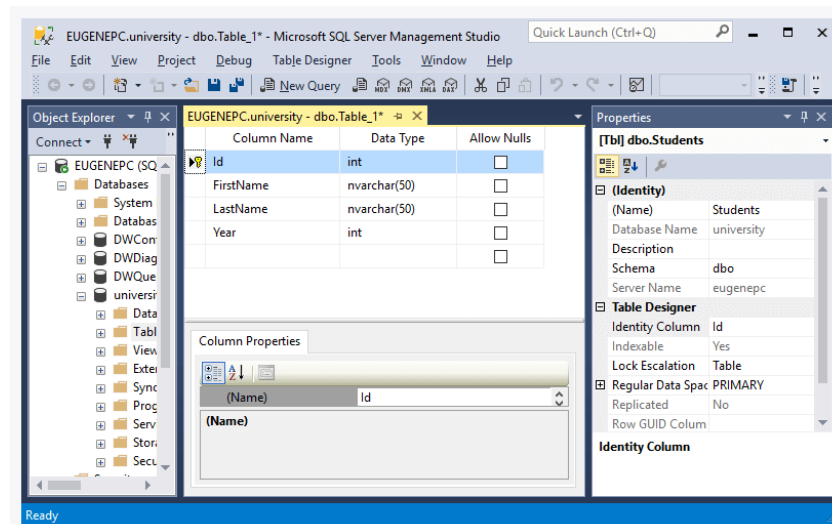
Начнем с графического способа.

Для этого раскроем узел созданной базы данных в SQL Server Management Studio, нажмем на его подузел Tables правой кнопкой мыши и далее в контекстном меню выберем New -> Table...:



После этого нам откроется дизайнер таблицы. В центральной части в таблице необходимо ввести данные о столбцах таблицы. Дизайнер содержит три поля:

- Column Name: имя столбца
- Data Type: тип данных столбца. Тип данных определяет, какие данные могут храниться в этом столбце. Например, если столбец представляет числовой тип, то он может хранить только числа.
- Allow Nulls: может ли отсутствовать значение у столбца, то есть может ли он быть пустым



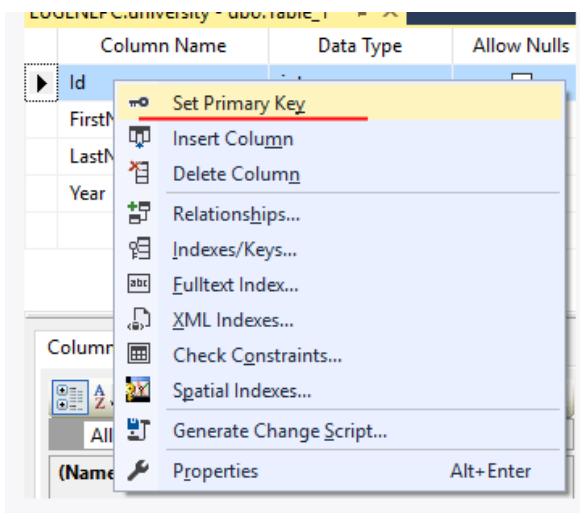
Допустим, нам надо создать таблицу с данными учащихся в учебном заведении. Для этого в дизайнера таблицы четыре столбца: Id, FirstName, LastName и Year, которые будут представлять соответственно уникальный идентификатор пользователя, его имя, фамилию и год рождения. У первого и четвертого столбца надо указать тип int (то есть целочисленный), а у столбцов FirstName и LastName - тип nvarchar(50) (строковый).

Затем в окне Properties, которая содержит свойства таблицы, в поле Name надо ввести имя таблицы - Students, а в поле Identity ввести Id, то есть тем самым указывая, что столбец Id будет идентификатором.

Имя таблицы должно быть уникальным в рамках базы данных. Как правило, название таблицы отражает название сущности, которая в ней хранится. Например, мы хотим сохранить студентов, поэтому таблица называется Students (слово студент во множественном числе на английском языке). Существуют разные мнения по поводу того, стоит использовать название сущности в единственном или множественном числе (Student или Students). В данном случае вопрос наименования таблицы всецело ложится на разработчика базы данных.

И в конце нам надо отметить, что столбец Id будет выполнять роль первичного ключа (primary key). Первичный ключ уникально идентифицирует каждую строку. В роли первичного ключа может выступать один столбец, а может и несколько.

Для установки первичного ключа нажмем на столбец Id правой кнопкой мыши и в появившемся меню выберем пункт Set Primary Key.

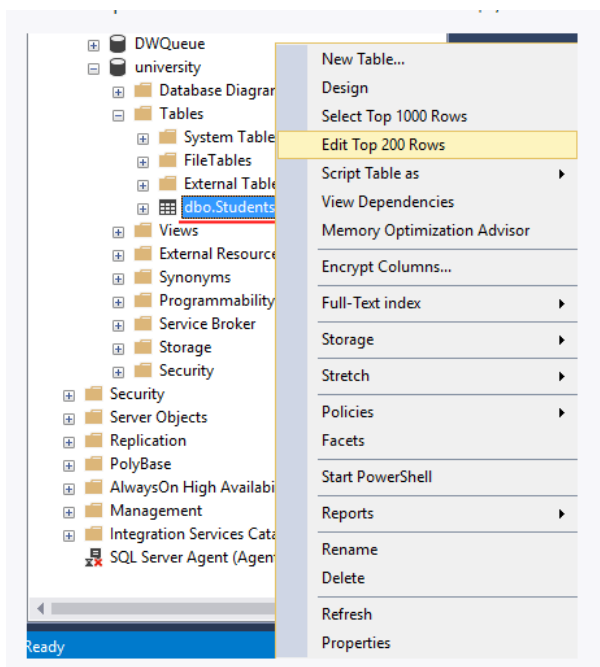


После этого напротив поля Id должен появиться золотой ключик. Этот ключик будет указывать, что столбец Id будет выполнять роль первичного ключа.

И после сохранения в базе данных university появится таблица Students:

Мы можем заметить, что название таблицы на самом деле начинается с префикса dbo. Этот префикс представляет схему. Схема определяет контейнер, который хранит объекты. То есть схема логически разграничивает базы данных. Если схема явным образом не указывается при создании объекта, то объект принадлежит схеме по умолчанию - схеме dbo.

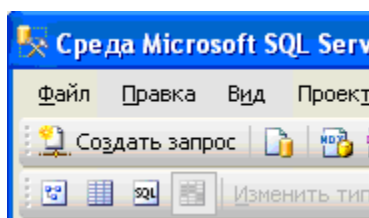
Нажмем правой кнопкой мыши на название таблицы, и нам отобразится контекстное меню с опциями:



С помощью этих опций можно управлять таблицей. Так, опция Delete позволяет удалить таблицу. Опция Design откроет окно дизайнера таблицы, где мы можем при необходимости внести изменения в ее структуру.

Для добавления начальных данных можно выбрать опцию Edit Top 200 Rows. Она открывает в виде таблицы 200 первых строк и позволяет их изменить. Но так как у нас таблица только создана, то естественно в ней будет никаких данных.

Применим для создания таблицы в SQL Server Management Studio скрипт на языке SQL. Для этого выберите на панели инструментов «Создать запрос»:



Введем новый запрос:

USE testing;

создавать таблицы:

-- определяем базу данных, в которой будем

```
CREATE TABLE groups
(id_gr INT NOT NULL,
name_gr CHAR(10) NOT NULL,
struct INT,
years CHAR(4),
department CHAR(20) NOT NULL,
PRIMARY KEY (id_gr)
);
```

Для того, чтобы поле первичного ключа id_gr автоматически нумеровалось, добавим слово **IDENTITY(1,1)**, тогда поле первичного ключа будет автоматически нумероваться, начиная с единицы, последовательными числами.

```
CREATE TABLE groups
(id_gr INT IDENTITY(1,1) NOT NULL,
name_gr CHAR(10) NOT NULL,
struct INT,
years CHAR(4),
department CHAR(20) NOT NULL,
PRIMARY KEY (id_gr)
);
```


В обозревателе объектов можно видеть, что таблица действительно создана. Файл с SQL-запросом сохраняем в папке проекта.

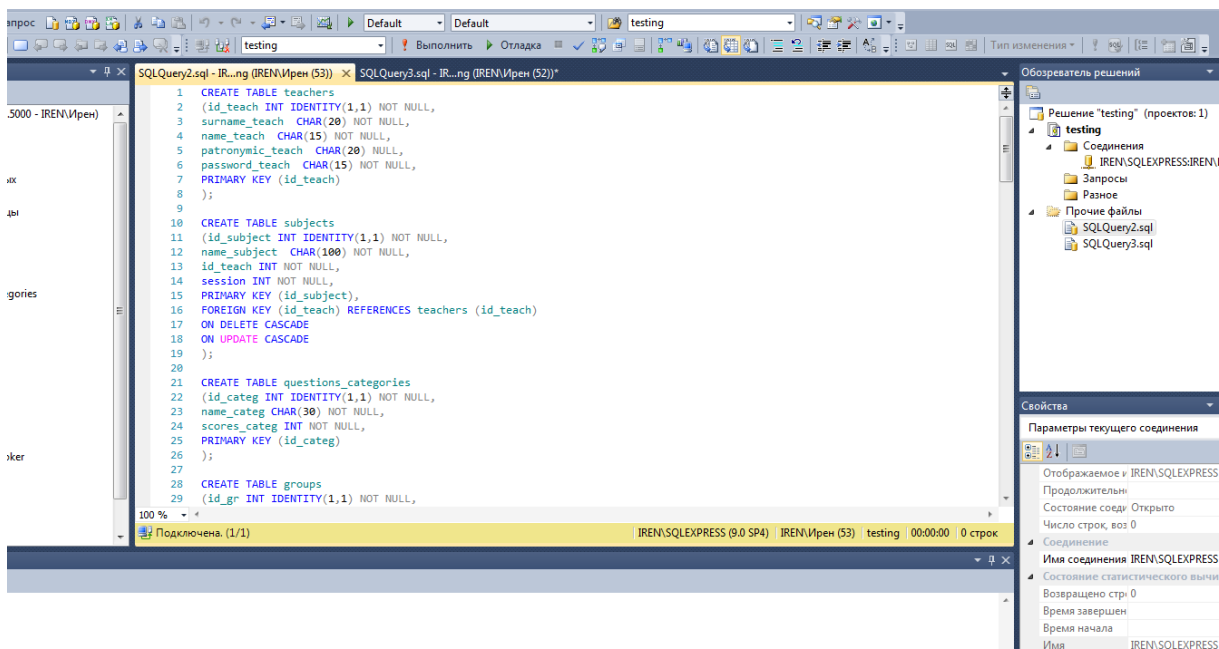
Таким же образом необходимо добавить в запрос остальные таблицы. Сначала записываем в запрос таблицы, в которых имеется только первичный ключ, а затем переходим к созданию таблиц с внешними ключами, для организации связей в базе данных в параметры создаваемой таблицы добавим строки

```
FOREIGN KEY (<имя внешнего ключа>) REFERENCES <имя связываемой
таблицы> (<имя внешнего ключа>)
ON DELETE CASCADE
ON UPDATE CASCADE
```

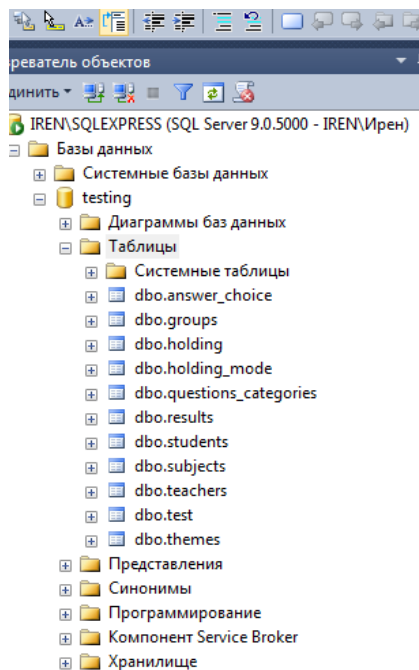
Например, таблица предметов, ссылающаяся на таблицу преподавателей через код преподавателя:

```
CREATE TABLE subjects
(id_subject INT IDENTITY(1,1) NOT NULL,
name_subject CHAR(100) NOT NULL,
id_teach INT NOT NULL,
session INT NOT NULL,
PRIMARY KEY (id_subject),
FOREIGN KEY (id_teach) REFERENCES teachers (id_teach)
ON DELETE CASCADE
ON UPDATE CASCADE
);
```

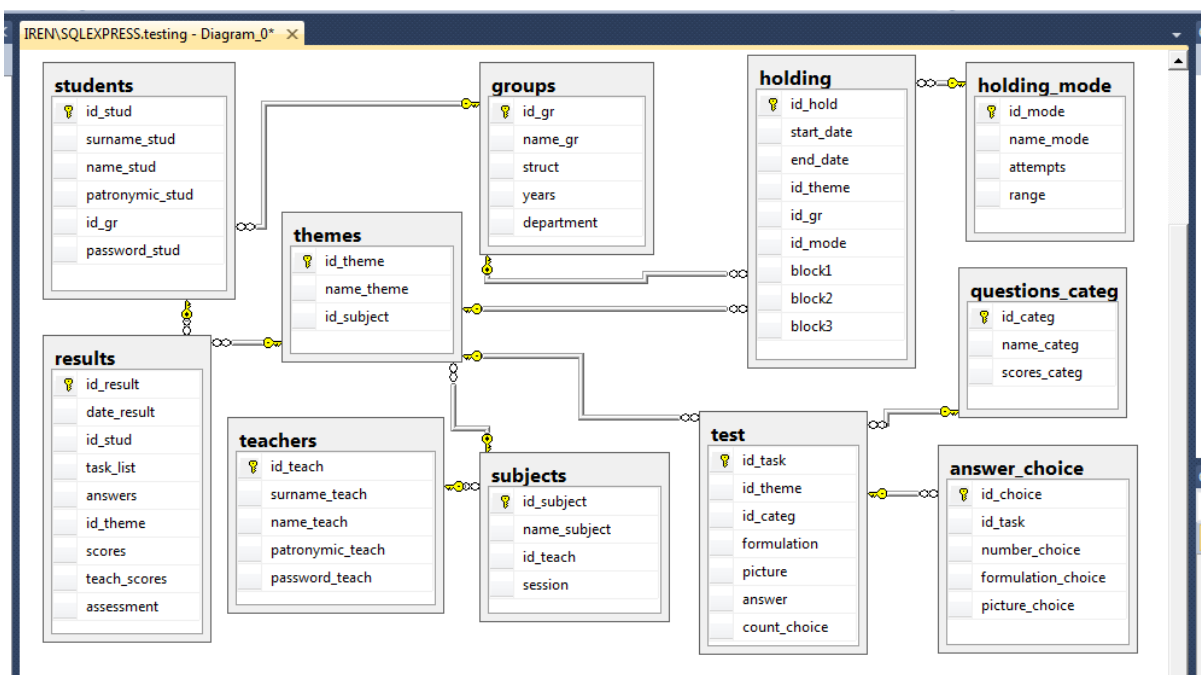
Запрос на создание таблиц базы данных:



После выполнения запроса необходимо обновить таблицы базы данных в обозревателе объектов, тогда можно увидеть созданные таблицы.



Для отображения связей создадим диаграмму созданной базы данных, для этого в обозревателе объектов вызовем контекстное меню раздела диаграммы баз данных и выберем Создать диаграмму базы данных. Затем в появившемся окне добавим таблицы и разместим их:



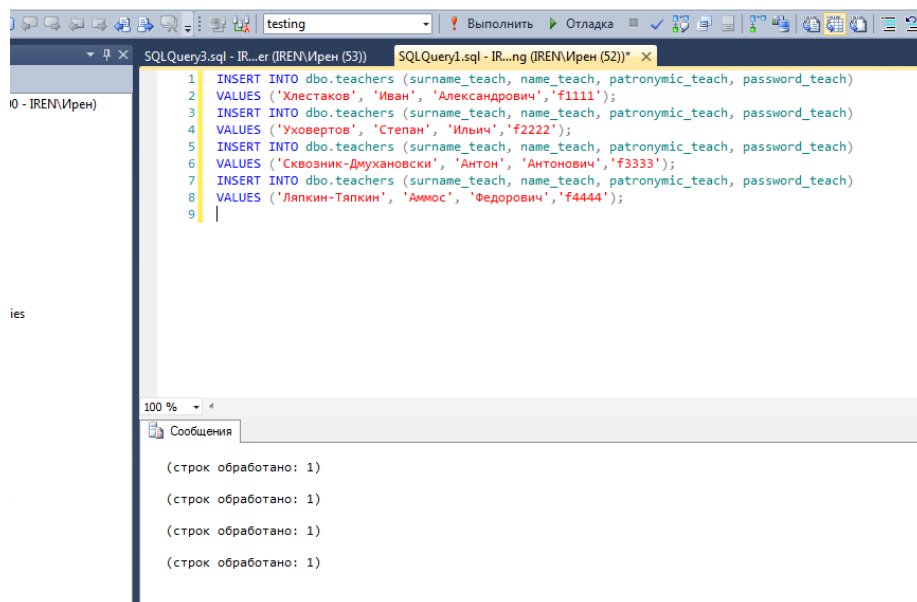
Осталось только наполнить таблицы данными.

Для ввода данных в таблицу воспользуемся командой INSERT INTO непосредственного ввода данных. Создадим запрос и запишем, например,

такие данные в таблицу Преподаватели. Все строковые значения, а также даты и время, необходимо обрамлять кавычками:

```
INSERT INTO dbo.teachers (surname_teach, name_teach, patronymic_teach, password_teach)
VALUES ('Хлестаков', 'Иван', 'Александрович', 'f1111');
INSERT INTO dbo.teachers (surname_teach, name_teach, patronymic_teach, password_teach)
VALUES ('Уховертов', 'Степан', 'Ильич', 'f2222');
INSERT INTO dbo.teachers (surname_teach, name_teach, patronymic_teach, password_teach)
VALUES ('Сквозник-Дмухановски', 'Антон', 'Антонович', 'f3333');
INSERT INTO dbo.teachers (surname_teach, name_teach, patronymic_teach, password_teach)
VALUES ('Ляпкин-Тяпкин', 'Аммос', 'Федорович', 'f4444');
```

Можно заметить, что поле `id_teach` в команде не присутствует, но заполняется автоматически:

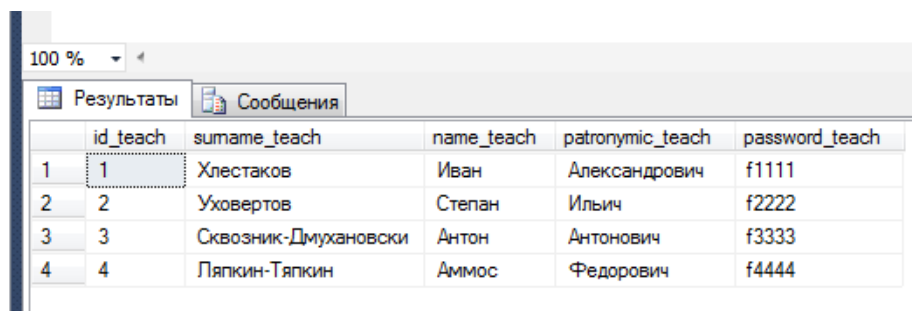


```
1 INSERT INTO dbo.teachers (surname_teach, name_teach, patronymic_teach, password_teach)
2 VALUES ('Хлестаков', 'Иван', 'Александрович', 'f1111');
3 INSERT INTO dbo.teachers (surname_teach, name_teach, patronymic_teach, password_teach)
4 VALUES ('Уховертов', 'Степан', 'Ильич', 'f2222');
5 INSERT INTO dbo.teachers (surname_teach, name_teach, patronymic_teach, password_teach)
6 VALUES ('Сквозник-Дмухановски', 'Антон', 'Антонович', 'f3333');
7 INSERT INTO dbo.teachers (surname_teach, name_teach, patronymic_teach, password_teach)
8 VALUES ('Ляпкин-Тяпкин', 'Аммос', 'Федорович', 'f4444');
```

Сообщения

(строк обработано: 1)
(строк обработано: 1)
(строк обработано: 1)
(строк обработано: 1)

В результате:



	id_teach	surname_teach	name_teach	patronymic_teach	password_teach
1	1	Хлестаков	Иван	Александрович	f1111
2	2	Уховертов	Степан	Ильич	f2222
3	3	Сквозник-Дмухановски	Антон	Антонович	f3333
4	4	Ляпкин-Тяпкин	Аммос	Федорович	f4444

Содержание отчета

- 1 Название работы
- 2 Цель работы
- 3 Перечень технических средств обучения
- 4 Порядок выполнения работы
- 5 Ответы на контрольные вопросы
- 6 Вывод

Варианты заданий

Варианты заданий представлены в практической работе № 39

Контрольные вопросы:

1. Каким образом можно получить доступ к базе данных MS SQL Server?
2. С помощью каких средств можно создать таблицу для MS SQL Server?
3. Как в SQL Server Management Studio задать первичный ключ?
4. Каким образом можно создать автоматическую нумерацию строк таблицы?
5. Что означают Not Null?
6. Как в SQL Server Management Studio осуществить ввод данных в таблицу?
7. Как в SQL Server Management Studio создать диаграмму базы данных?

Используемая литература

- Г.Н.Федорова Основы проектирования баз данных. М.: Академия, 2020
- Г.Н.Федорова Разработка, администрирование и защита баз данных. М.: Академия, 2018
- http://wiki.vspu.ru/users/kontchenkov/my_project/indexhttps://metanit.com/sql/sqlserver/3.1.php