

Практическая работа 53

Проектирование базы данных.

Управление базой данных

Цель работы: Получить практический опыт проектирования базы данных.

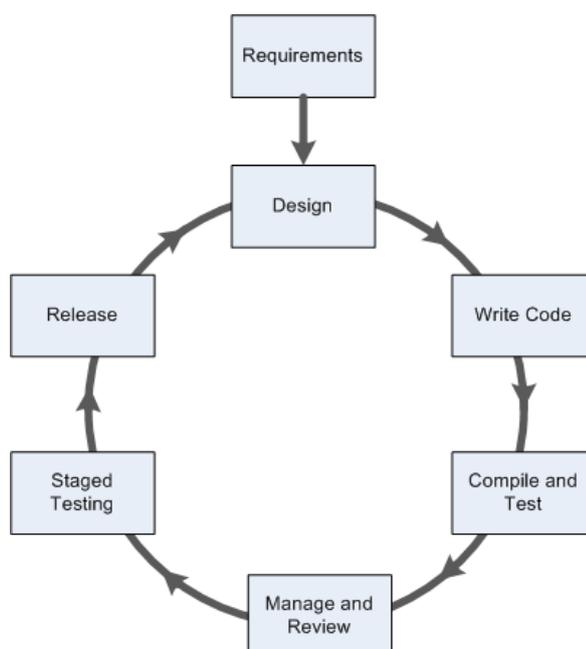
Перечень оборудования и программного обеспечения

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

Краткие теоретические сведения

Являясь одной из самых больших компаний по производству программного обеспечения в мире, корпорация Майкрософт сосредоточенно помогает группам разработки успешно работать.

На рисунке показан цикл разработки программного обеспечения, от первоначального сбора требований клиента до выпуска программного обеспечения и его постоянной итеративной разработки, обслуживания и изменения после развертывания.

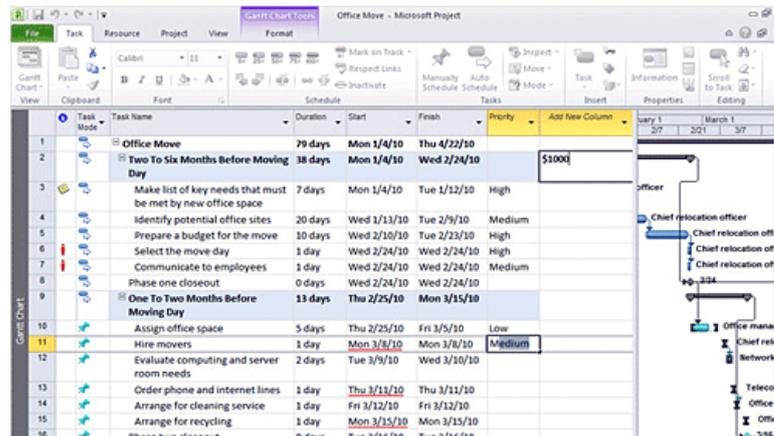


Планирование и анализ требований

Хотя обычно эта задача относится к разработке архитектуры, группы разработчиков будут ответственны за преобразование требований клиента в практическую архитектуру, которая может быть реализована с учетом

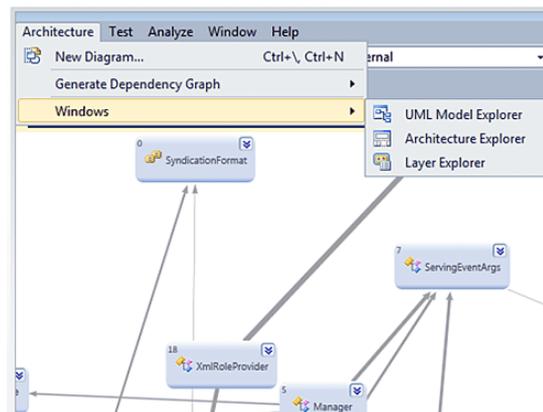
ограничений денежных и временных затрат и бюджет прекрасно работать с доступным оборудованием и инфраструктурой.

Сбор и анализ требований — это первый этап, на котором обычно используются такие инструменты и приложения, как программы MicrosoftOffice: Word, Excel, Visio, Project и другие. Требования могут предоставляться как документы, диаграммы, схемы, модели и описания среды выполнения. MicrosoftProject и MicrosoftExcel — это идеальные инструменты для планирования сроков и оценки затрат проектов разработки на данном начальном этапе. На рисунке показан пример планирования проекта с помощью MicrosoftProject.

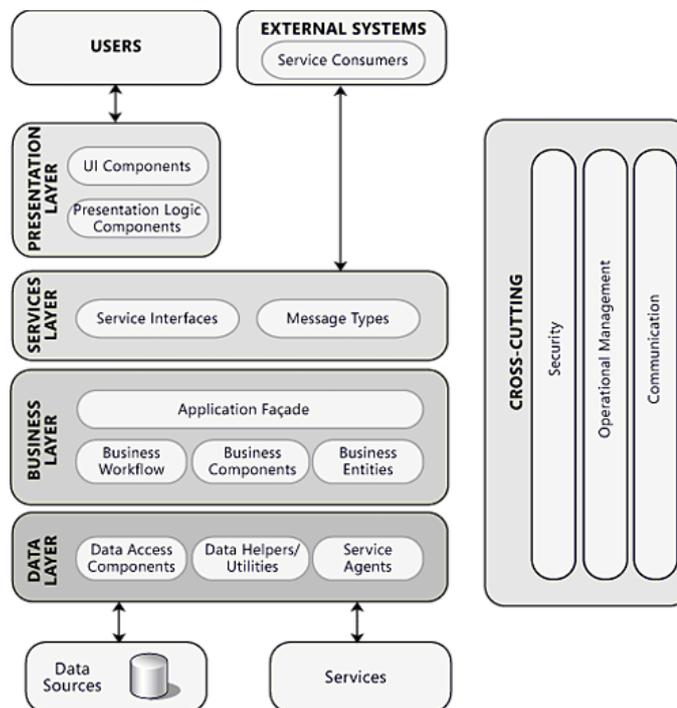


Архитектура, проектирование и моделирование

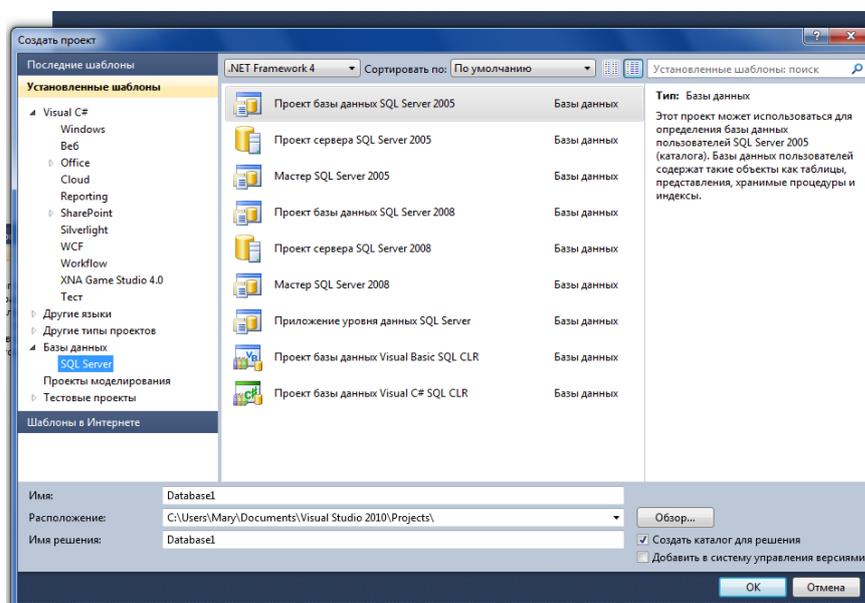
Что касается более подробной архитектуры, в состав MicrosoftVisualStudio входит ряд инструментов, упрощающих проектирование архитектуры приложения и создание зависимостей модели и развертывания. Они могут использоваться для моделирования общего проекта и распространения компонентов, а также последующего наложения на модель аппаратной инфраструктуры для подтверждения практичности архитектуры. Эти инструменты также можно использовать для создания диаграмм взаимодействия и зависимости компонентов, а также моделирования схем баз данных как часть подробного плана. На рисунке показан конструктор моделей в VisualStudio 2010.



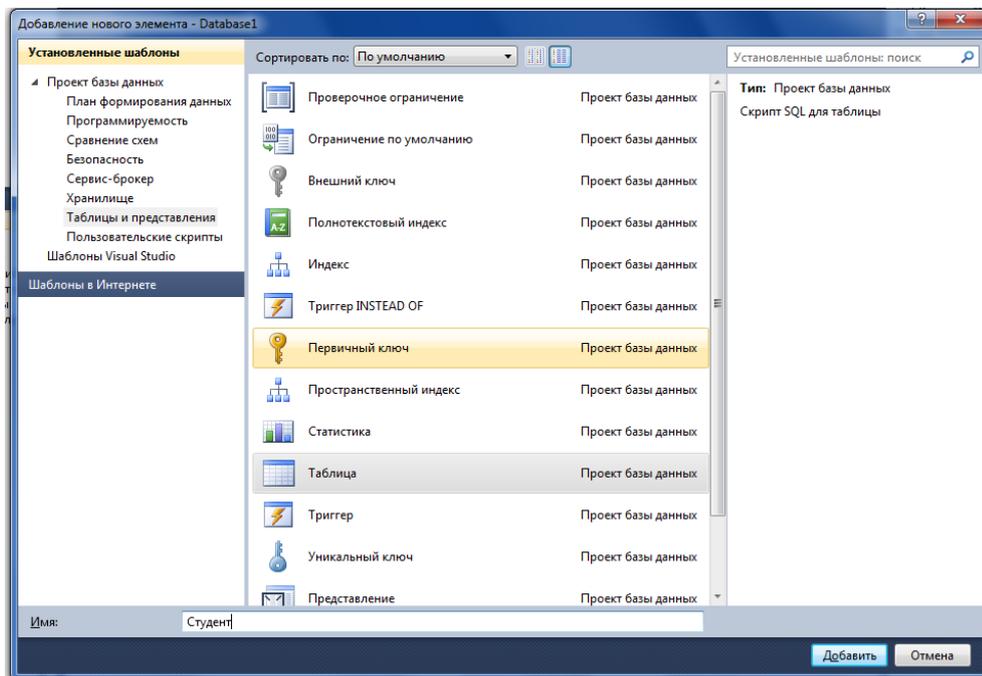
Майкрософт также предоставляет инструкции, помогающие на этапе проектирования. Они включают в себя общие архитектурные рекомендации, информацию о типичных проектах для различных типов приложений и справку по оценке компромиссов, необходимых для обеспечения соответствия программного обеспечения требованиям. Например, на рисунке показана типичная современная архитектура для приложений.



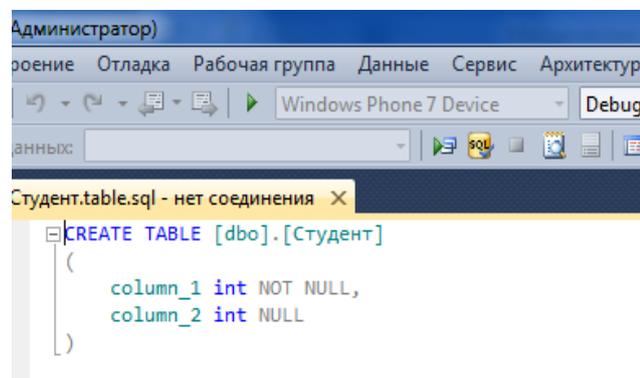
В VisualStudio для проектирования БД необходимо выбрать шаблон Базы данных – SQLServer.



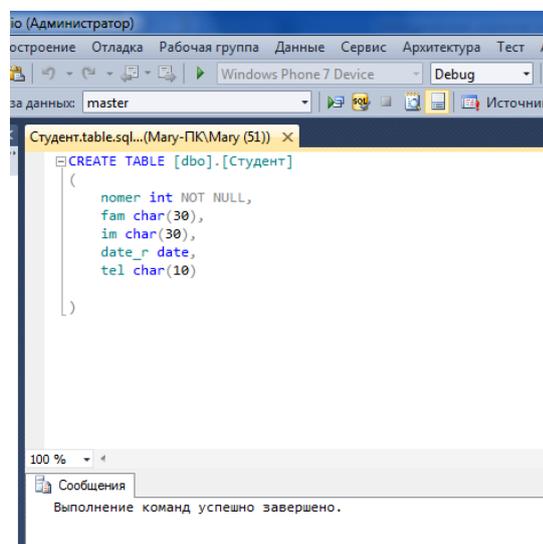
Далее выберите Проект базы данных SQLServer 2008. Нажмите ОК. Далее нажатием правой кнопки выберите Добавить новый элемент.



В открывшемся окне можно создать SQL-запрос для создания таблицы БД



После заполнения запроса выполните его и убедитесь в его выполнении.



В процессе эксплуатации созданной базы данных возникает необходимость изменить либо физические параметры, либо логическую структуру этой базы. К управлению базой данных на физическом уровне относится вся работа по изменению имен, размера, количества, положения файлов базы данных, усечению базы данных и журнала транзакций, созданию групп файлов, изменению группы файлов по умолчанию, изменению имени и владельца базы данных. Большинство действий по изменению конфигурации базы данных выполняется с помощью команды ALTER DATABASE. Для уменьшения размера базы данных можно также использовать команды DBCC SHRINKDATABASE, DBCC SHRINKFILE и системная хранимая процедура sp_dboption. На логическом уровне изменяются такие параметры, как выполнение автоматического усечения журнала транзакций, автоматическое создание и обновление статистики, возможность выполнения вложенных триггеров и т.п. – всего 22 параметра. Изменять эти параметры можно командой ALTER DATABASE с параметром SET и процедурой sp_dboption.

Сервер позволяет отсоединять (sp_detach) и присоединять (sp_attach_db) до 32767 баз данных, изменять владельца (sp_changedbowner), просматривать свойства (sp_dboption и DATABASEPROPERTY), получать справки (sp_helpdb) и т.д.

В Microsoft SQL Server предусмотрены параметры, которые влияют на результат и производительность инструкций SQL. Эти параметры можно задать следующими способами.

Чтобы задать параметры конфигурации для всего экземпляра, воспользуйтесь хранимой процедурой sp_configure.

Чтобы задать параметры на уровне базы данных, используйте инструкцию ALTER DATABASE SET Options.

Чтобы задать уровень совместимости базы данных, используйте инструкцию ALTER DATABASE CompatibilityLevel.

Чтобы указать параметры на уровне пакета (параметры инструкции SET), используйте такие инструкции SET, как SET ANSI_PADDING и SET ANSI_NULLS.

Параметры уровня пакета в более ранних версиях SQL Server, а также в соединениях без использования результирующего набора MARS называются параметрами уровня соединения.

Чтобы указать параметры на уровне инструкций (такие, как подсказки в запросе, таблицах и соединении), используйте отдельные инструкции языка Transact-SQL. Приложения ODBC могут задавать особые параметры соединения, которые управляют некоторыми параметрами ANSI SET. Поставщик OLE DB для собственного клиента SQL Server и ODBC-драйвер собственного клиента SQL Server задают некоторые параметры SET по умолчанию.

По возможности лучше не изменять параметры SET или задавать их с помощью инструкций SET. Вместо этого рекомендуется задавать параметры SET на уровне соединения через свойства соединения ODBC или OLE DB.

Параметр SET можно также изменить с помощью хранимой процедуры `sp_configure`.

Процедура `sp_configure` имеет параметр `user options`. Это позволяет изменять значения по умолчанию некоторых параметров SET. Хотя `useroptions` кажется параметром экземпляра, `useroptions` — параметр SET.

Иерархия параметров

Если параметр поддерживается на нескольких уровнях, используется следующая иерархия.

Параметр базы данных имеет приоритет над параметром экземпляра.

Параметр SET имеет приоритет над параметром базы данных.

Подсказка имеет приоритет над параметром SET.

Параметры SET, заданные внутри динамического пакета SQL, действуют только в области этого пакета. Такие параметры SET, как `QUOTED_IDENTIFIER` и `ANSI_NULLS`, сохраняются вместе с определениями хранимых процедур и, следовательно, имеют приоритет над различными значениями, явно заданными для них.

Задания

1. Изучить теоретические сведения.
2. В соответствии с вариантом задания выполнить операции по управлению данными:
 - добавление поля;
 - удаление поля;
 - изменение свойств поля;
 - создание и удаление первичного ключа;
 - создание и удаление внешнего ключа;
 - объединение таблиц;
 - разделение таблиц

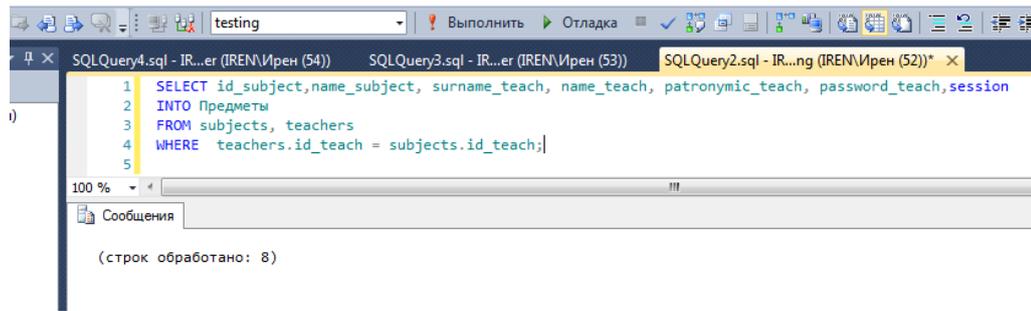
Порядок выполнения работы

Объединение таблиц в одну.

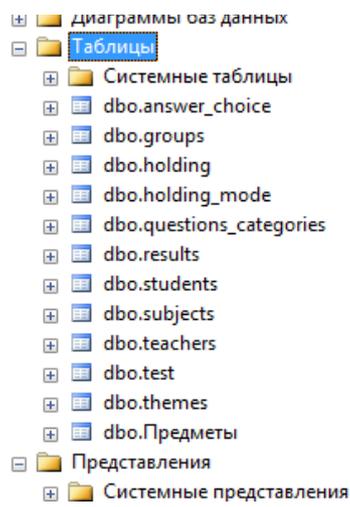
Для получения результата:

- а) скопировать поля из исходных таблиц в новую с помощью команды `SELECT`;
- б) в полученной таблице добавить первичный ключ

Из таблицы `subjects` переносим в созданную таблицу `Предметы` все поля, кроме кода преподавателя, а из таблицы `teachers` поля, кроме кода преподавателя с помощью команды:



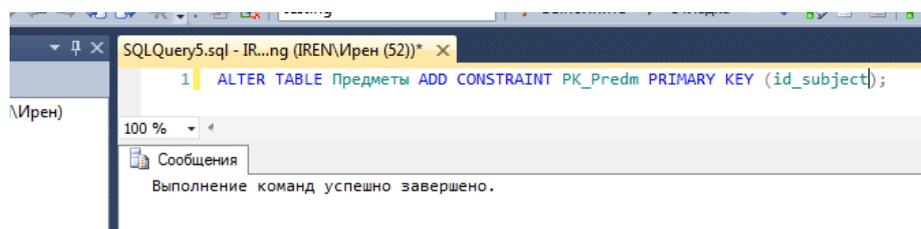
В результате в списке появляется таблица Предметы:



И данные в ней

id_subject	name_subject	surname_teach	name_teach	patronymic_te...	password_teach	session
1	МДК 02.01 Про...	Хлестаков	Иван	Александрови...	f1111	5
2	МДК 02.02 Раз...	Уховертов	Степан	Ильич	f2222	5
4	Информацион...	Сквозник-Дму...	Антон	Антонович ...	f3333	3
5	ОП.13 Инфор...	Сквозник-Дму...	Антон	Антонович ...	f3333	4
6	ОП.05 Инфор...	Сквозник-Дму...	Антон	Антонович ...	f3333	4
7	МДК 01.01 Раз...	Уховертов	Степан	Ильич	f2222	6
8	МДК 01.02 Раз...	Уховертов	Степан	Ильич	f2222	6
9	ОП.12 Компью...	Уховертов	Степан	Ильич	f2222	3
*	NULL	NULL	NULL	NULL	NULL	NULL

Добавим в новой таблице первичный ключ:



И результат:

Имя столбца	Тип данных	Разрешить ...
id_subject	int	<input type="checkbox"/>
name_subject	char(100)	<input type="checkbox"/>
surname_teach	char(20)	<input type="checkbox"/>
name_teach	char(15)	<input type="checkbox"/>
patronymic_teach	char(20)	<input checked="" type="checkbox"/>
password_teach	char(15)	<input type="checkbox"/>
session	int	<input type="checkbox"/>

Разделение таблицы на несколько новых таблиц.

Вспомним действия, необходимые для получения результата:

- в) скопировать поля из исходной таблицы во вспомогательную с помощью команды SELECT;
- г) во вспомогательной таблице добавить ключевое поле код данных с помощью ALTER TABLE ... ADD..., причем определить это поле таким образом, чтобы записи пронумеровались автоматически;
- д) в случае необходимости, можно изменить тип первичного ключа на целый с помощью команды ALTER TABLE ... ALTER COLUMN;
- е) создать основную таблицу из исходной, но вместо полей, вынесенных во вспомогательную, перенести из вспомогательной ключевое поле, а чтобы в основную таблицу попали нужные значения, к команде добавить условия (SELECT ...INTO ...FROM ...WHERE);
- ж) после того, как обе таблицы созданы, останется определить связи, т.е. добавить внешние ключи с помощью команды ALTER TABLE ... ADD CONSTRAINT.

Вновь созданную таблицу Предметы разделим на Преподаватели и Предметы1. Сначала создадим таблицу Преподаватели с помощью команды:

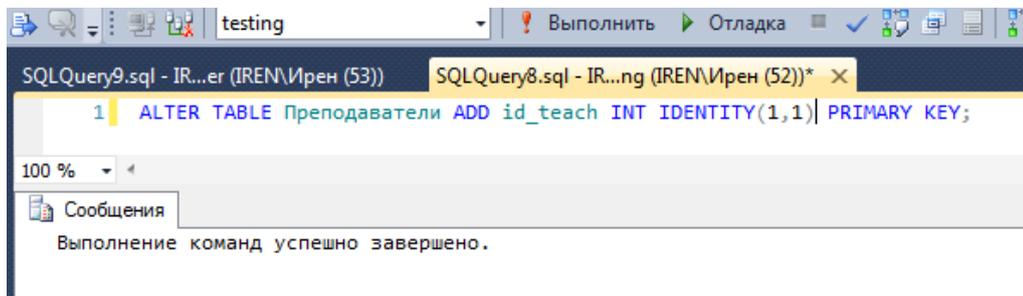
```

1 | SELECT DISTINCT| surname_teach,name_teach,patronymic_teach,password_teach INTO Преподаватели FROM Предметы;

```

(строк обработано: 3)

Вместе с созданием таблицы, в ней сформированы три строки, т.к. именно столько неповторяющихся данных было в исходной таблице. Затем добавим поле id_teach, определим его как первичный ключ.



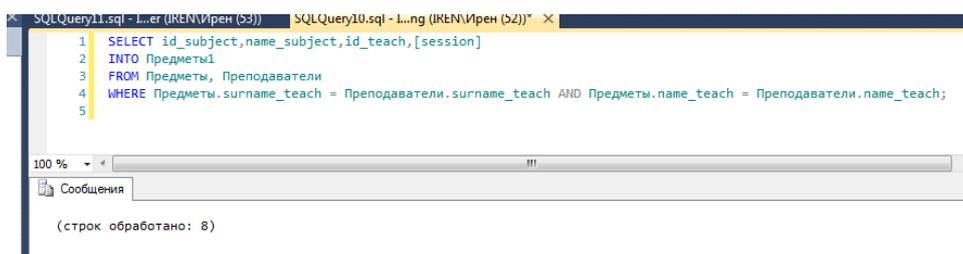
В результате получим таблицу:

Имя столбца	Тип данных	Разрешить ...
surname_teach	char(20)	<input type="checkbox"/>
name_teach	char(15)	<input type="checkbox"/>
patronymic_teach	char(20)	<input checked="" type="checkbox"/>
password_teach	char(15)	<input type="checkbox"/>
id_teach	int	<input type="checkbox"/>

И данные в ней:

	surname_teach	name_teach	patronymic_te...	password_teach	id_teach
Ирен)	Сквозник-Дмухановски	Антон	Антонович ...	f3333	1
	Уховертов	Степан	Ильич	f2222	2
	Хлестаков	Иван	Александрови...	f1111	3
*	NULL	NULL	NULL	NULL	NULL

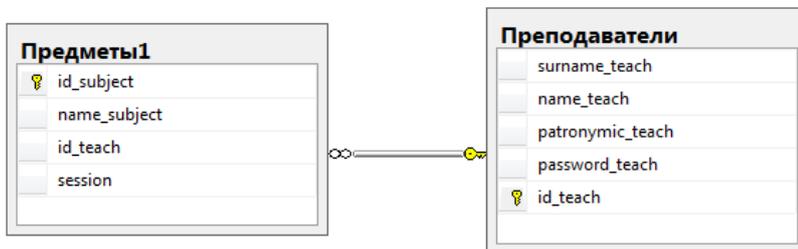
Эти команды можно выполнить в одном запросе. Следующий этап – создание таблицы Предметы1, в которой из таблицы Предметы использованы все поля, кроме данных о преподавателях, а из таблицы Преподаватели id_teach. Для того, чтобы нужный номер был занесен в запись, используется условие.



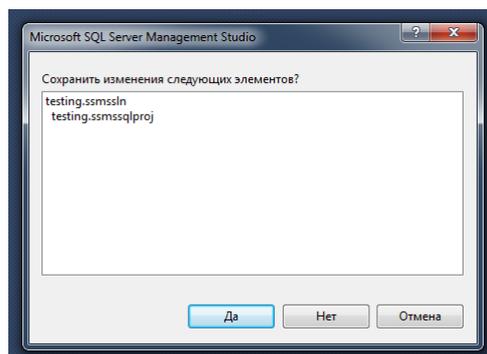
В результате получим таблицу:

Добавим ключи:

Результат на схеме данных:



После выполнения преобразований в базе данных SQL Server предлагает сохранить изменения в файлах базы данных и проекта:



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. Какой командой осуществляются действия по изменению конфигурации базы данных?
2. Какие действия относятся к управлению базой данных на физическом уровне?
3. С помощью какой команды выполняется большинство действий по изменению конфигурации базы данных?
4. Какая команда выполняет действия по изменению структуры таблиц базы данных?
5. Какие изменения в структуре таблиц базы данных возможны?
6. Синтаксис команды на добавление полей таблицы.
7. Как записывается команда на изменение характеристик поля?
8. Синтаксис команды на создание в заданном поле внешнего ключа.
9. Как строится команда на удаление поля таблицы?

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