Лабораторная работа № 2

Проектирование и создание базы данных на сервере Microsoft SQL Server

***Цель работы*:** получить навыки проектирования и создания баз данных с использованием утилиты SQL Server Management Studio (SSMS).

Продолжительность работы - 4 ч.

Теоретические сведения

Проектирование базы данных

Проектирование базы данных (БД) было рассмотрено на лекциях на примере учебного задания, по условию которого нужно спроектировать БД для учета жителей и их доходов.

В БД должна храниться информация:

1) о жителях;
2) квартирах, занимаемых жителями;
3) телефонах, установленных в квартирах;
4) источниках и размерах доходов жителей.

В учебном задании объектами, представляющими интерес, являются ЖИТЕЛЬ, КВАРТИРА, ТЕЛЕФОН, ДОХОД. Описание каждого из объектов показано на рис.1. Связи между объектами отражаются на диаграмме ER-типа (рис.2).

В соответствии с методом проектирования БД на основе инфологической модели, рассмотренным на лекционных занятиях, объекты и связи между ними представляются в учебной БД пятью таблицами:

PERSON (Nom, FIO, Rdate, Pol, SumD, Adr)
FLAT (Adr, Skv, Nrooms, KCategory)
TPHONE (Ntel, TCategory, Adr)
PROFIT (Id, Source, Moneys)
HAVE_D(Nom, Id)

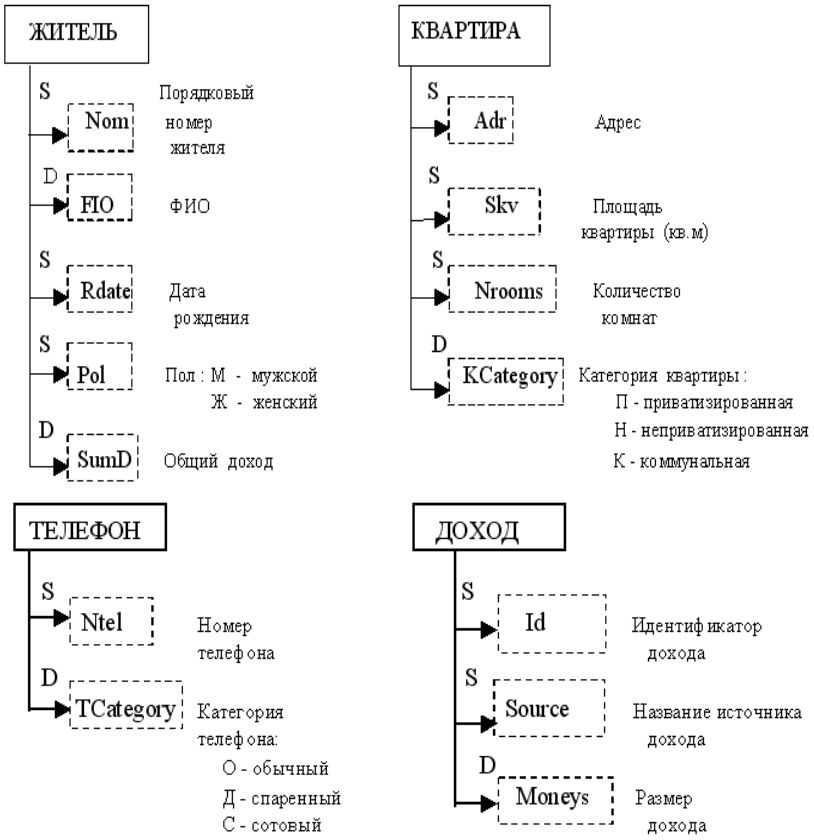

Рис.1. Описание объектов

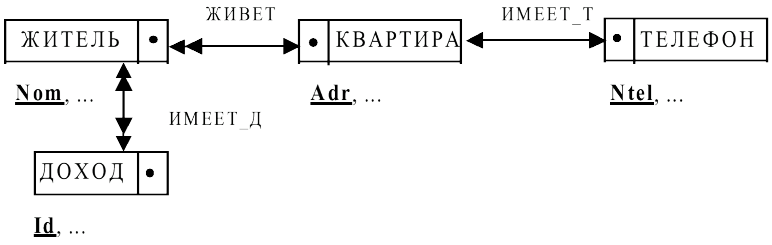

Рис.2. Диаграмма ER-типа

Проектирование БД завершается оформлением схемы БД, в которой перечисляются входящие в БД таблицы, атрибуты (столбцы) таблиц и характеристики атрибутов (тип, длина, индексация (ключ) и др.). Схема БД составляется с учетом особенностей системы управления базами данных (СУБД), которая будет выбрана для реализации БД.

Для учебного задания выберем СУБД Microsoft SQL Server 2008. В этой СУБД имеется ряд типов данных, которые хорошо подходят в качестве типов атрибутов (столбцов) таблиц и позволяют адекватно представить специфику атрибутов (табл.1).

Таблица 1

Некоторые типы данных СУБД Microsoft SQL Server 2008

Тип	Примечание
nchar (n)	Строка символов фиксированной длины n
nvarchar (n)	Строка символов переменной длины (от 0 до n символов)
decimal (p,s)	Числа с фиксированной запятой
float (n)	Числа с плавающей запятой
money	Денежный тип (аналог типа decimal, сформатированный для отображения денежных сумм)
tinyint	Однобайтовое целое
smallint	Двухбайтовое целое
int	Четырехбайтовое целое
date	Дата
time	Время
datetime	Комбинация даты и времени

В таблице PERSON атрибут Nom обозначает порядковый номер жителя и используется для однозначной идентификации жителя. Чтобы порядковые номера жителей различались, номер жителя, вновь включаемого в таблицу PERSON, должен быть на 1 больше номера предыдущего жителя. Такая установка номера будет осуществляться автоматически, если атрибут Nom будет целого типа и при создании таблицы для атрибута будут заданы равные 1 свойства Identity Seed и Identity Increment, обозначающие начальное значение атрибута и шаг его изменения. Атрибуты FIO, Pol, Adr содержат символьную информацию и могут быть типа nchar или nvarchar. Атрибут Rdate должен быть типа date. Атрибут SumD целесообразно отнести к типу money.

В таблице FLAT атрибут Adr должен иметь тот же тип, что и одноименный атрибут в таблице PERSON. Атрибут Skv обозначает площадь квартиры, задаваемую числом с дробной частью, которое относится к

типу decimal. Атрибут Nrooms может быть типа tinyint. Атрибут KCategory будет типа nchar.

В таблице TPHONE атрибуты Ntel и TCategory относятся к типу nchar, а атрибут Adr должен иметь тот же тип, что и одноименные атрибуты в таблицах PERSON и FLAT.

В таблице PROFIT атрибут Id используется для идентификации различных видов доходов (вид дохода - сочетание названия источника и размера дохода). Двух одинаковых видов доходов в таблице PROFIT быть не должно, и для автоматического обеспечения уникальности значения атрибута Id его целесообразно отнести к целому типу и задать для атрибута свойства Identity Seed и Identity Increment. Атрибуты Source и Moneys относятся к типам nvarchar и money соответственно.

В таблице HAVE_D атрибуты Nom и Id должны быть целого типа, поскольку их значения являются копиями значений одноименных атрибутов в таблицах PERSON и PROFIT соответственно.

Схема БД приведена в табл.2. Обратите внимание, что в схеме БД список атрибутов таблицы HAVE_D дополнен атрибутом Comment типа nvarchar, чтобы при необходимости иметь возможность записать дополнительную информацию (комментарий) о виде дохода Id, который имеется у жителя с номером Nom.

Таблица 2

Схема БД для учебного задания

Таблица БД	Атрибут	Тип	Размер	Допустимые значения	Значение по умолчанию
PERSON	Nom	int			
	FIO	nvarchar	(30)		
	Rdate	date			
	Pol	nchar	(1)	М,Ж	
	SumD	money			0
	Adr	nvarchar	(30)		

Окончание

Таблица БД	Атрибут	Тип	Размер	Допустимые значения	Значение по умолчанию

Таблица БД	Атрибут	Тип	Размер	Допустимые значения	Значение по умолчанию
FLAT	Adr	nvarchar	(30)		
	Skv	decimal	(3, 1)	>=0	0
	Nrooms	tinyint		0, 1, 2, 3, 4	0
	KCategory	nchar	(1)	П,Н,К	Н
TPHONE	Ntel	nchar	(8)	_ _ _ _ _ _ _ _	
	TCategory	nchar	(1)	О, Д, С	О
	Adr	nvarchar	(30)		
PROFIT	Id	int			
	Source	nvarchar	(20)		
	Moneys	money		>=0	0
HAVE_D	Nom	int		>0	
	Id	int		>0	
	Comment	nvarchar	(30)		

На основе схемы БД создается сама БД. При этом стремятся, чтобы обеспечивались целостность БД и высокая скорость доступа к информации, хранящейся в БД.

Обеспечение целостности БД

Целостность БД выражается в том, что в любой момент времени БД содержит правильные данные. Целостность БД обеспечивается заданием и проверкой определенных условий, которым должны удовлетворять значения атрибутов, связи атрибутов разных таблиц и другие информационные элементы БД. Такие условия называются ограничениями целостности (ОЦ). Рассмотрим некоторые ОЦ для значений атрибутов и для связей таблиц.

Чтобы память, предназначенная для хранения таблиц, использовалась рационально, в таблицах должно быть сведено к минимуму количество пустых полей, соответствующих неопределенным значениям атрибутов.

В проектируемой БД все атрибуты, за исключением некоторых значений атрибута Comment в таблице HAVE_D, должны иметь определенные значения. Такое ОЦ может быть установлено при создании БД. В СУБД Microsoft SQL Server это делается отменой признака Allow Nulls (Разрешить неопределенные значения) во время перечисления атрибутов таблицы. Для атрибутов, входящих в первичный ключ, этот признак отменяется автоматически, поскольку их значения всегда определены.

Следующее ОЦ относится к допустимым значениям некоторых атрибутов (см. табл.2). Оно также может быть установлено средствами СУБД Microsoft SQL Server при перечислении атрибутов таблицы как одно из свойств атрибута или таблицы.

Важным ОЦ является требование отсутствия одинаковых строк в таблице. Это ОЦ обеспечивается заданием первичного ключа таблицы, представляющего собой атрибут (или набор атрибутов), который однозначно идентифицирует конкретную строку таблицы. Для проектируемой БД первичные ключи перечислены в табл.3.

Таблица 3

Первичные ключи для таблиц проектируемой БД

Таблица	Первичный ключ	Таблица	Первичный ключ
PERSON	Nom	PROFIT	Id
FLAT	Adr	HAVE_D	Nom, Id
TPHONE	Ntel		

Связи между объектами, изображенные на рис.2, представлены в проектируемой БД связями таблиц, образующих БД (рис.3).

Связь двух таблиц, изображенная на рис.3 линией со стрелками, указывающими на имена атрибутов таблиц, означает, что в первой таблице имеется одна строка (одиночная стрелка) или не менее одной строки (двойная стрелка), в которой значение отмеченного стрелкой атрибута совпадает со значением соответствующего атрибута во второй таблице.

Например, связь таблицы FLAT с таблицей PERSON по атрибуту Adr выражается в том, что в таблице FLAT имеется только одна строка со значением атрибута Adr = x, которой соответствует одна или более строк в таблице PERSON, содержащих значение атрибута Adr = x; связь таблицы FLAT с таблицей TPHONE по атрибуту Adr выражается в том, что если в квартире установлен телефон, то в таблице FLAT имеется только одна строка со значением атрибута Adr = x, которой соответствует ровно одна строка в таблице TPHONE, содержащая значение атрибута Adr = x.

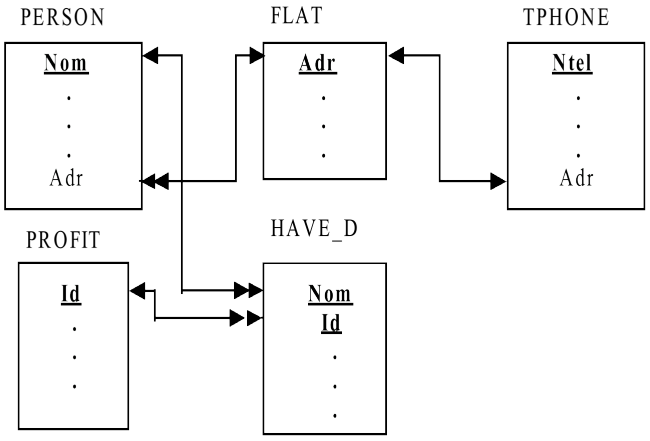

Рис.3. Связи таблиц в проектируемой БД
(первичные ключи подчеркнуты)

Из двух связанных таблиц одна называется главной (master, parent), а другая - подчиненной (detailed, child). Атрибут (или набор атрибутов), по которому связаны две таблицы, в главной таблице является первичным ключом, а в подчиненной таблице - внешним ключом (табл.4).

Таблица 4

Главные и подчиненные таблицы и их ключи

Главная таблица	Первичный ключ	Подчиненная таблица	Внешний ключ
PERSON	Nom	HAVE_D	Nom
FLAT	Adr	PERSON	Adr
FLAT	Adr	TPHONE	Adr
PROFIT	Id	HAVE_D	Id

Множество значений внешнего ключа должно содержаться во множестве значений соответствующего первичного ключа. Это ОЦ для связей таблиц должно гарантировать отсутствие в подчиненной таблице строк, не связанных через значение внешнего ключа со строкой в главной таблице, а также должно предотвратить удаление из главной таблицы строки, связанной через значение первичного ключа со строками в подчиненной таблице.

ОЦ для связей таблиц, называемое ограничением ссылочной целостности, обеспечивается установкой соответствия значений внешних и первичных ключей для подчиненных и главных таблиц.

Обеспечение быстрого доступа к информации

Для ускорения доступа к информации, хранящейся в БД, таблицы индексируются. В результате создается индекс, упорядоченный по значениям индексного ключа и содержащий ссылки на строки таблицы.

В качестве индексного ключа используется атрибут (или набор атрибутов) индексируемой таблицы, который часто применяется для поиска в таблице. Если индексным ключом является первичный ключ, то созданный индекс называется первичным (Primary Index). Если индексный ключ отличается от первичного, то созданный индекс называется вторичным (Secondary Index). Так, для быстрого поиска жителя по его фамилии в таблице PERSON целесообразно создать вторичный индекс, указав в качестве индексного ключа атрибут FIO. Другим примером индексного ключа служит внешний ключ подчиненной таблицы, в которой созданный вторичный индекс позволяет быстро определить наличие строк, содержащих значения внешних ключей, равные значению первичного ключа главной таблицы.

В СУБД Microsoft SQL Server первичный индекс формируется автоматически при создании БД, когда атрибут таблицы отмечается как принадлежащий первичному ключу. Так же автоматически формируется вторичный индекс по внешнему ключу, когда устанавливается связь между таблицами. Другие вторичные индексы, необходимые для работы с таблицей, формируются с помощью специальной команды Indexes/Keys утилиты SSMS.

Создание базы данных

Для создания базы данных пользователю должны быть предоставлены соответствующие права на уровне всего сервера, например, посредством назначения пользователю так называемой серверной роли DBCREATOR.

При использовании утилиты SSMS база данных создается в окне обозревателя объектов выбором из контекстного меню узла Databases команды New Database. После активизации одноименного окна в поле Database Name необходимо набрать имя базы данных и затем нажать кнопку OK.

Описание структуры таблиц и связей между ними. Схема проектируемой БД, представленная в табл.2, задается описанием структуры отдельных таблиц. Наличие связей между таблицами (см. рис.3) опреде-

ляет порядок, в котором описываются структуры таблиц: в первую очередь описываются структуры наименее зависимых таблиц FLAT и PROFIT, а затем структуры таблиц TPHONE, PERSON и HAVE_D.

Описание структуры таблиц, входящих в создаваемую базу данных, начинается с выбора в окне обозревателя объектов узла Tables, соответствующего этой базе, и вызова контекстного меню выбранного узла. Команда New Table, содержащаяся в контекстном меню, активизирует окно конструктора таблиц (рис.4), в котором описывается структура таблицы.

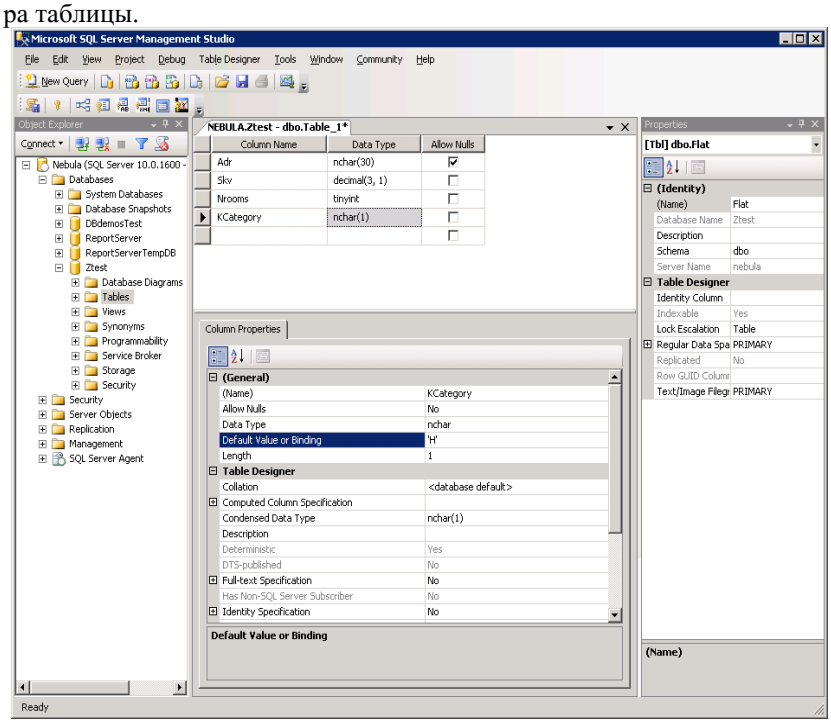

Рис.4. Окно конструктора таблиц

Для каждого столбца таблицы указывается его имя (графа Column Name), тип данных (графа Data Type) и разрешение или запрет неопределенных значений (графа Allow Nulls).

В области Column Properties задается значение по умолчанию, а также можно задать возможность получения автоинкрементальных значений для столбца, указав в разделе Identity Specification начальное значение Identity Seed и шаг Identity Increment.

Принадлежность столбца (атрибута) первичному ключу задается указанием столбца и выбором команды Set Primary Key из контекстного меню или нажатием одноименной кнопки (с пиктограммой в виде ключа) на панели инструментов конструктора таблиц. Если первичный ключ состоит из нескольких столбцов, то перед выбором команды Set Primary Key все столбцы, входящие в первичный ключ, должны быть указаны мышью при нажатой клавише Ctrl.

Для исключения столбца из первичного ключа служит команда контекстного меню Remove Primary Key.

Для указания допустимых значений, хранящихся в столбце, предназначена команда Check Constraints (Проверочные ограничения) контекстного меню конструктора таблиц, активизирующая одноименное окно (рис.5), в котором задаются условия, проверяемые при добавлении или изменении данных. Данные, для которых условие истинно, сохраняются в таблице.

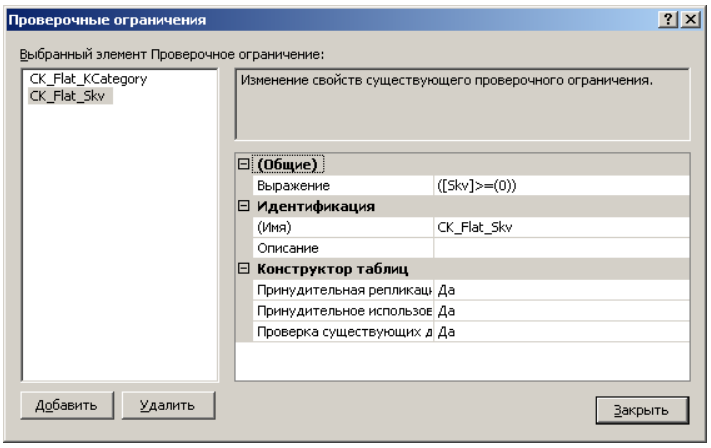

Рис.5. Окно, содержащее список ограничений целостности

Новое условие добавляется нажатием кнопки Add (Добавить) и задается в поле Expression (Выражение) раздела General (Общие), например: Skv>=0. Условиям присваиваются имена, задаваемые в поле Name (Имя) раздела Identity (Идентификация), а в поле Description (Описание) записываются словесные формулировки условий. Поскольку при вводе данных, не соответствующих условию, СУБД выводит в сообщении об ошибке имя нарушенного условия, то желательно, чтобы имя условия (например, CK_Flat_Skv) отражало связь условия с таблицей и столбцом (столбцами).

Имя создаваемой таблицы задается в поле Name раздела Identity, расположенного в окне свойств (см. рис.4).

Структура создаваемой таблицы сохраняется либо командой главного меню утилиты File | Save..., либо командой Save… контекстного меню, либо нажатием соответствующей кнопки на панели инструментов утилиты.

Для изменения структуры сохраненной таблицы следует в окне обозревателя объектов выбрать команду Design из контекстного меню этой таблицы, чтобы активизировать окно конструктора таблиц. Добавление или удаление столбцов таблицы осуществляется командой Insert Column или Delete Column контекстного меню конструктора таблиц.

Если при сохранении измененной структуры таблицы появляется сообщение о невозможности выполнить сохранение и предлагается удалить таблицу и повторно ее создать, то следует задать режим работы утилиты SSMS, разрешающий сохранение изменений, выбрав в главном меню утилиты команду Tools | Options, а в дереве параметров - узел Designers | Table and DB Designers и сняв флаг для параметра Prevent Saving Changes.

Для описания структуры таблицы FLAT нужно задать соответствующий список атрибутов (см. табл.2) и отметить атрибут Adr как первичный ключ. Для всех атрибутов следует запретить неопределенные значения, а для атрибутов Skv, Nrooms, KCategory указать допустимые значения и значения по умолчанию.

Структура таблицы PROFIT описывается в соответствии со схемой БД, приведенной в табл.2. Для всех атрибутов, кроме Id, запрещаются неопределенные значения, а атрибут Id отмечается как первичный ключ, и запрет неопределенных значений будет установлен автоматически. Кроме того, в области Column Properties для атрибута Id в разделе Identity Specification свойство Is Identified следует задать равным Yes, чтобы установить начальное значение Identity Seed и приращение Identity Increment для атрибута.

Структура таблицы TPHONE описывается в соответствии со схемой БД (см. табл.2). Для всех атрибутов устанавливается запрет неопределенных значений, а атрибут Ntel отмечается как первичный ключ. С учетом наличия в этой подчиненной таблице внешнего ключа (см. табл.4) она должна быть связана с главной таблицей FLAT.

Для описания связи подчиненной таблицы с главной таблицей следует в окне конструктора таблиц (рис.6), в котором отображается структура подчиненной таблицы, командой контекстного меню

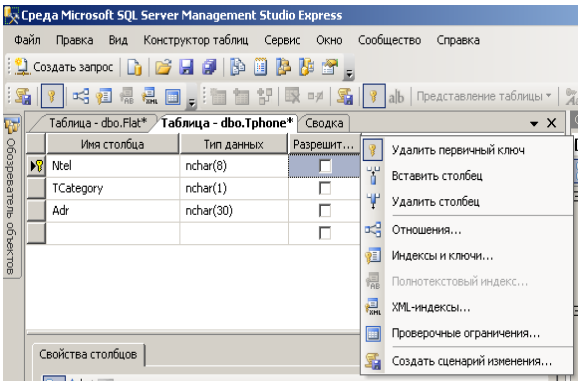

Рис.6. Окно конструктора таблиц со структурой подчиненной таблицы и командами контекстного меню

Relationships (Отношения) активизировать окно со списком внешних ключей (рис.7) и приступить к созданию внешнего ключа, нажав последовательно кнопку Add (Добавить) и кнопку с многоточием в поле Tables And Columns Specification (Спецификация таблиц и столбцов), расположенном в разделе General (Общие).

В результате появляется окно, в котором задается соответствие между первичным ключом главной таблицы и внешним ключом подчиненной таблицы (рис.8). С правой стороны окна расположен список главных таблиц и список их столбцов. Следует выбрать таблицу FLAT, связанную с подчиненной таблицей TPHONE, и столбец Adr, являющийся первичным ключом главной таблицы FLAT. С левой стороны окна расположен список столбцов подчиненной таблицы TPHONE. Из этого списка нужно выбрать атрибут Adr, являющийся внешним ключом.

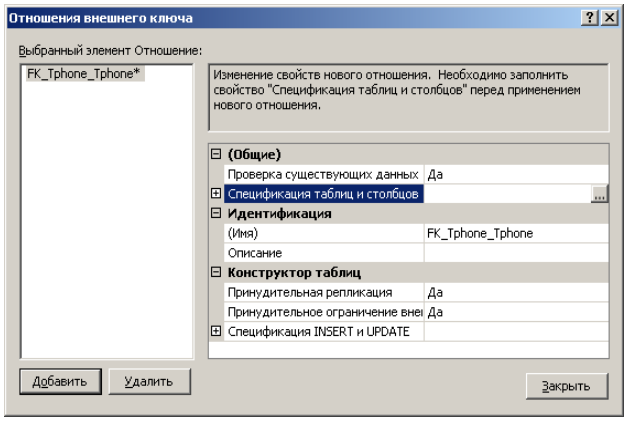

Рис.7. Окно со списком внешних ключей подчиненной таблицы

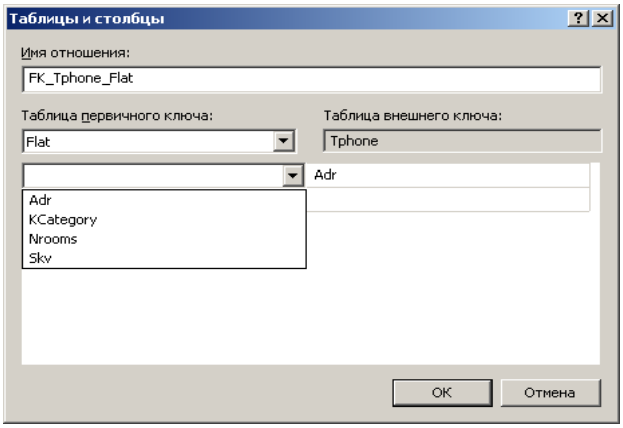

Рис.8. Описание связи главной и подчиненной таблиц

После выполнения нужных установок следует в поле Name раздела Identity задать имя ограничения ссылочной целостности и нажать кнопку OK. Имя ограничения ссылочной целостности FK_Tphone_Flat_Adr означает, что таблицы связаны по адресу квартиры.

Структура таблицы PERSON описывается в соответствии со схемой БД (см. табл.2) и с учетом наличия в этой таблице внешнего ключа (см. табл.4). Для всех атрибутов, кроме Nom, следует запретить

неопределенные значения, а атрибут Nom отметить как первичный ключ и задать для него свойства Identity Seed и Identity Increment.

Внешний ключ описывается тем же способом, что и для таблицы TPHONE. В качестве имени ограничения ссылочной целостности для таблицы PERSON следует задать FK_Person_Flat_Adr.

Чтобы создать вторичный индекс по атрибуту FIO для таблицы PERSON, нужно в контекстном меню конструктора таблиц (см. рис.6) выбрать команду Indexes/Keys (Индексы и ключи) и после появления одноименного окна со списком индексов и ключей таблицы PERSON нажать кнопку Add или ДОБАВИТЬ (рис.9).

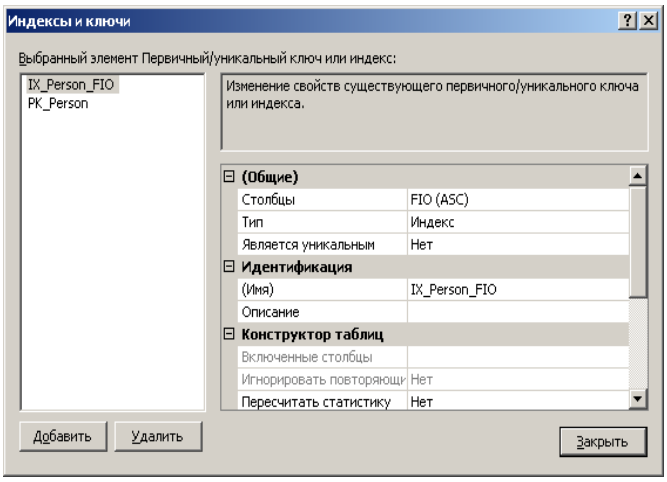

Рис.9. Окно со списком индексов и ключей таблицы

Затем в поле Name раздела Identity следует ввести имя вторичного индекса IX_Person_FIO (индекс принадлежит таблице PERSON, индексным ключом является атрибут FIO) и в поле Columns раздела General нажать кнопку с многоточием, чтобы в открывшемся окне задать индексный ключ и порядок сортировки его значений в индексе.

Структура таблицы HAVE_D описывается в соответствии со схемой БД (см. табл.2) и с учетом наличия в этой таблице двух внешних ключей (см. табл.4). Для атрибутов Nom и Id устанавливается запрет неопределенных значений, и они отмечаются как принадлежащие первичному ключу. Атрибут Nom описывается как внешний ключ, связанный с одноименным первичным ключом таблицы PERSON. Соответствующее ограничение ссылочной

целостности обозначается именем FK_Have_Person_Nom. Атрибут Id описывается как внешний ключ, связанный с одноименным первичным ключом таблицы PROFIT. Соответствующее ограничение ссылочной целостности обозначается именем FK_Have_Profit_Id.

После описания структуры таблиц можно проверить правильность установленных связей главных таблиц с подчиненными таблицами (см. рис.3), построив диаграмму созданной базы данных. Для этого нужно в окне обозревателя объектов из контекстного меню узла Database Diagram командой New Database Diagram вызвать окно со списком таблиц базы данных, кнопкой Add добавить все таблицы в диаграмму и нажать кнопку Close. (Если при выборе команды New Database Diagram утилита предложит установить поддержку диаграмм для базы данных, то нужно принять это предложение.)

Для каждой таблицы в диаграмме можно задать желаемое представление (рис.10), указав его с помощью команды Table View (Представление таблицы) контекстного меню выбранной таблицы: стандартное со списком имен столбцов, их типов и признаков разрешения/запрета неопределенных значений (см. таблицу Profit на рис.10), имена столбцов (см. таблицы Flat и Have_D), ключи (см. таблицу Tphone), только имя таблицы (см. таблицу Person).

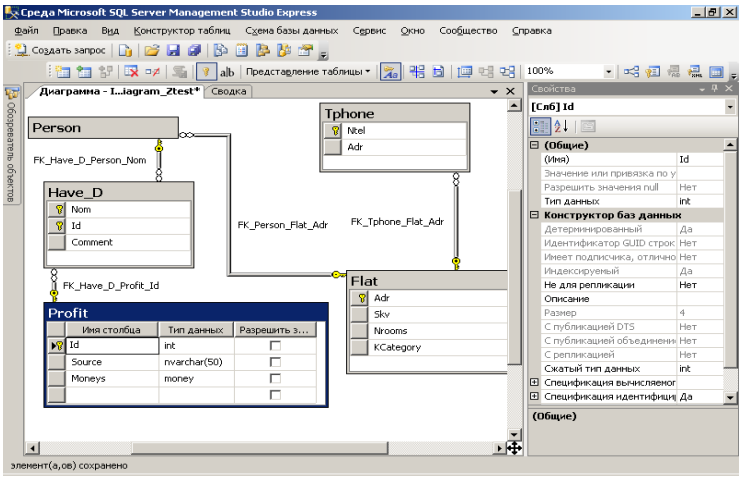

Рис.10. Окно с диаграммой базы данных

Выделяя на диаграмме таблицу, столбец или связь таблиц, можно в окне свойств проверить или задать свойства выделенного элемента диаграммы.

После проверки связей таблиц окно закрывается нажатием кнопки с крестиком. На запрос о сохранении изменений следует ответить утвердительно и затем задать имя диаграмме.

Описание структуры таблиц в окне Database Diagram. Окно с диаграммой базы данных можно использовать не только для просмотра связей таблиц и свойств объектов базы данных, заданных с помощью конструктора таблиц, но и для создания новых таблиц и связей между ними.

Новая таблица создается командой главного меню утилиты Database Diagram | New Table или командой New Table контекстного меню, появляющегося при щелчке мышью в свободной области диаграммы. При создании таблицы запрашивается ее имя, а стандартное представление создаваемой таблицы позволяет описать структуру таблицы так же, как в конструкторе таблиц. Свойства столбца таблицы задаются в окне свойств (см. рис.10).

Другие действия по описанию структуры таблиц и связей между ними выполняются с использованием тех же команд контекстного меню таблиц или кнопок панели инструментов, которые предусмотрены в конструкторе таблиц. Однако проще связь таблиц задать, выделив мышью в подчиненной таблице столбец (столбцы) внешнего ключа и переместив (при нажатой левой кнопке мыши) курсор, помеченный плюсом, в область главной таблицы. При этом откроется окно, аналогичное показанному на рис.8, для записи имени связи и проверки соответствия первичного и внешнего ключей. После нажатия кнопки ОК активизируется окно, аналогичное показанному на рис.7, со списком внешних ключей.

Все изменения, сделанные в окне с диаграммой (добавление или удаление таблиц и столбцов, изменение свойств, установка связей, первичных ключей, индексов и т.д.), при сохранении диаграммы будут запоминаться в базе данных.

Ввод информации в таблицы базы данных

После создания таблиц базы данных в них заносится информация, для чего можно воспользоваться командой Edit... из контекстного меню таблицы, в которую будет вводиться информация. При этом открывается окно конструктора запросов с областью результатов (рис.11).

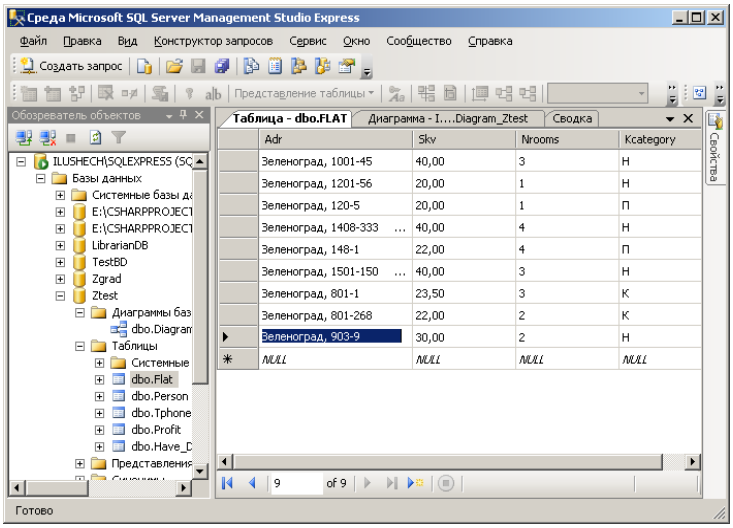

Рис.11. Окно ввода информации в таблицу

Новые данные набираются в строке, помеченной звездочкой. Для перемещения по строкам таблицы предназначены кнопки навигатора, расположенные в нижней части окна. Ранее введенные данные доступны для изменений, а изменения сохраняются при переходе в другую строку таблицы.

При первоначальном вводе данных переход к вводу следующей строки осуществляется нажатием клавиши со стрелкой вниз.

Генерация описания таблиц на языке SQL

Утилита SSMS позволяет автоматически генерировать схему созданной БД в виде сценария на языке SQL. Для этого в окне обозревателя объектов в контекстном меню базы данных или конкретной таблицы следует выбрать команду Script Database as | CREATE to или Script Table as | CREATE to.

Генерируемый сценарий по выбору пользователя может быть помещен либо в буфер обмена (Clipboard), либо в файл (File), либо открыт в новом окне редактора запросов для просмотра или изменения (New Query Editor Window).

Лабораторное задание

1. Ознакомиться с рассмотренной на лекциях методикой проектирования БД на основе инфологической модели (ИЛМ).
2. Используя рассмотренную на лекциях методику, при домашней подготовке разработать ИЛМ и спроектировать БД для своего варианта задания (номер варианта совпадает с номером компьютера).
3. Ознакомиться с описанным в лабораторной работе процессом создания учебной БД средствами утилиты SSMS и ответить на контрольные вопросы.
4. Создать БД для СУБД SQL Server согласно своему варианту задания.
5. Оформить отчет и защитить работу.

Порядок выполнения работы

1. При домашней подготовке спроектировать БД на основе ИЛМ, разработанной для своего варианта задания, изучить описание лабораторной работы и письменно ответить на контрольные вопросы.
2. Запустить утилиту SSMS через Терминал 4100 (skylab.sipc.miet.ru) командой Start | Microsoft SQL Server 2008 | SQL Server Management Studio.
3. Создать базу данных с именем U<№СТУДБИЛЕТА>.
4. В конструкторе таблиц описать в соответствии со схемой базы данных структуру двух таблиц и связать их как главную и подчиненную.
5. Создать диаграмму базы данных и продолжить описывать структуру и связи остальных таблиц в окне диаграммы.
6. Создать необходимые индексы для таблиц.
7. Ввести данные в созданную базу, добавив в каждую таблицу не менее 5 строк.
8. Сгенерировать описания всех таблиц базы данных на языке SQL и сохранить их в одном файле, имеющем расширение .sql.
9. Защитить работу, показав результаты и ответив на заданные преподавателем вопросы.

Требования к отчету

Отчет должен содержать:
1) название и цель работы;

2) ответы на контрольные вопросы;

3) соответствующую варианту задания инфологическую модель, включающую все необходимые компоненты;

4) схему БД, связи таблиц в проектируемой БД, ограничения целостности БД, перечень первичных и внешних ключей, изображение связей таблиц в проектируемой БД, список необходимых вторичных индексов;

5) файл сценария с описанием таблиц на языке SQL.

Контрольные вопросы

1. Из каких компонентов состоит инфологическая модель предметной области?
2. Что представляет собой целостность базы данных и как она обеспечивается?
3. Какие виды ограничений целостности существуют?
4. Как обеспечить быстрый доступ к данным в проектируемой базе данных?
5. Что представляет собой индексный ключ?
6. Какие виды индексов существуют?
7. Структура каких таблиц описывается в первую очередь при создании базы данных?

Варианты заданий

1. Спроектировать и создать БД для хранения сведений о студентах, обучающихся на факультетах института, с учетом изучаемых дисциплин.
2. Спроектировать и создать БД для хранения сведений о распределении библиотечных книг между студентами, обучающимися на факультетах института.
3. Спроектировать и создать БД для хранения сведений о преподавателях, ведущих занятия по различным дисциплинам в студенческих группах.
4. Спроектировать и создать БД для хранения сведений о сотрудниках, работающих на кафедрах института, с учетом совмещения должностей.
5. Спроектировать и создать БД для учета студентов-дипломников, выпускаемых кафедрами института, и их руководителей.

6. Спроектировать и создать БД для хранения сведений о доходах жителей и жилой площади, принадлежащей жителям Зеленограда, предусмотрев возможность владения несколькими квартирами.

7. Спроектировать и создать БД для учета продажи туристических путевок конкретным клиентам различными туроператорами.

8. Спроектировать и создать БД для учета товаров, поступающих в магазин от определенных поставщиков и продаваемых конкретным покупателям.

9. Спроектировать и создать БД для учета размещения журналов и книг в личной библиотеке.

10. Спроектировать и создать БД для хранения сведений об абитуриентах, поступающих на факультеты института, и о результатах сдачи ими вступительных экзаменов.

11. Спроектировать и создать БД для хранения сведений о процессе ремонта телевизоров, поступающих от заказчиков, мастерами телеателье.

12. Спроектировать и создать БД для учета передачи книг из библиотечного коллектора в фонды различных библиотек города.

13. Спроектировать и создать БД для хранения сведений о студентах, обучающихся на факультетах института, с учетом мест прохождения практики.

14. Спроектировать и создать БД для учета библиотечных книг, выданных студентам института.

15. Спроектировать и создать БД для учета занятий, проводимых кафедрами со студентами в аудиториях института.

16. Спроектировать и создать БД для хранения сведений о кадровом составе кафедр института с учетом данных о детях сотрудников.

17. Спроектировать и создать БД для регистрации граждан, находящихся в санатории, с учетом распределения их по комнатам и назначения им лечебных процедур.

18. Спроектировать и создать БД для регистрации доставки определенных товаров на конкретные оптовые базы транспортной организацией с указанных предприятий и с учетом транспортных расходов, а также сроков доставки.

19. Спроектировать и создать БД для учета рейсов, организованных разными авиакомпаниями на арендуемых самолетах.

20. Спроектировать и создать БД для учета антикварных книг, сдаваемых в магазин конкретными гражданами и оформляемых различными приемщиками.

21. Спроектировать и создать БД для учета автомобилей, продаваемых гражданам и организациям.

22. Спроектировать и создать БД для учета заказов на использование грузового и погрузочного автотранспорта по заявкам граждан или организаций.

23. Спроектировать и создать БД для учета распределения автобусов по маршрутам, предусмотрев возможность использования одного автобуса на разных маршрутах в различные периоды времени.

24. Спроектировать и создать БД для учета продажи железнодорожных билетов пассажирам.

25. Спроектировать и создать БД для учета заявок, поступающих от слушателей, с просьбой передать музыкальные произведения в радиоэфире.

26. Спроектировать и создать БД для учета использования аудиторий для занятий по различным дисциплинам в студенческих группах.

27. Спроектировать и создать БД для учета жилой площади и родственных связей между жителями Зеленограда.

28. Спроектировать и создать БД для учета оплаты дополнительных занятий, проводимых преподавателями кафедр института.

29. Спроектировать и создать БД для учета занятости взлетно-посадочных полос самолетами на аэродроме.

30. Спроектировать и создать БД для учета сотрудников и проектов, в разработке которых они участвуют, предусмотрев хранение сведений о стоимости проекта, размере заработной платы сотрудника и об оплате работ, выполненных отдельным сотрудником по проекту.