

## Лабораторная работа № 3

### Проектирование базы данных для СУБД Microsoft SQL Server и оценка ее размера средствами программы ERwin

**Цель работы:** спроектировать и создать базу данных для СУБД Microsoft SQL Server с помощью программы ERwin, реализующей методологию IDEF1X.

**Продолжительность работы** - 4 ч.

#### Теоретические сведения

Проектирование базы данных (БД) начинается с разработки инфологической модели (ИЛИМ) предметной области. ИЛИМ содержит, в частности, описание объектов и связей между ними, которые могут задаваться диаграммой “сущность - связь” (ER-диаграммой). Результатом проектирования БД является даталогическая модель (ДЛИМ) базы данных, содержащая описание таблиц, образующих проектируемую БД, на языке выбранной СУБД.

На использовании ER-диаграмм основана методология IDEF1X, реализованная в программе ERwin, предназначенная для автоматизации процесса проектирования и создания БД.

В методологии IDEF1X сущности делятся на независимые и зависимые. Независимая сущность - это сущность, каждый экземпляр которой может быть идентифицирован без учета его подчиненности другим сущностям. Примерами независимых сущностей являются ОТДЕЛ и СЛУЖАЩИЙ, если всем экземплярам каждой из этих сущностей присвоить уникальные номера, которые будут значениями атрибутов DepId (номер отдела) и EmpId (табельный номер служащего). В этом случае, несмотря на то что каждый служащий “подчинен” отделу, он идентифицируется своим номером независимо от отдела, в котором работает. Аналогичным образом каждый отдел идентифицируется независимо от служащих, которые в нем работают. Независимая сущность изображается в виде прямоугольного блока, внутри которого указан список атрибу-

тов. Атрибуты, входящие в ключ сущности, размещаются в начале списка и отделяются от других атрибутов горизонтальной чертой (рис.1,*а*).

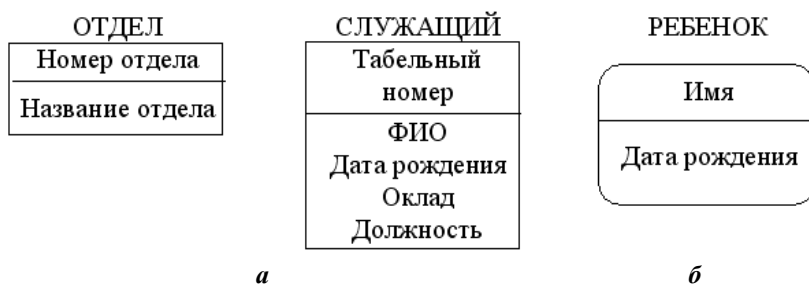


Рис.1. Графическое обозначение независимых (*а*) и зависимой (*б*) сущностей

Зависимая сущность - это сущность, однозначная идентификация экземпляра которой зависит от его подчиненности другой сущности. Примером зависимой сущности служит сущность РЕБЕНОК, если для обозначения каждого экземпляра использовать имя ребенка, родителем которого является служащий. Поскольку у разных служащих могут быть дети с одинаковыми именами, то однозначная идентификация ребенка зависит от его “подчиненности” своему родителю-служащему. Зависимая сущность изображается в виде блока с закругленными углами (рис.1,*б*).

При рассмотрении связи двух сущностей подчиненная сущность называется сущностью-потомком (СП), а подчиняющая сущность - сущностью-родителем (СР).

Связь сущностей характеризуется идентификацией и степенью.

Идентифицирующая связь, обозначаемая сплошной линией, соединяет сущность-родителя с зависимой сущностью-потомком (рис.2,*а*) и представляет на диаграмме степень связи 1:N (или 1:1).

Неидентифицирующая связь, обозначаемая штриховой линией, соединяет сущность-родителя с независимой сущностью-потомком и представляет степень связи 1:N или 1:1 (рис.2,*б*).

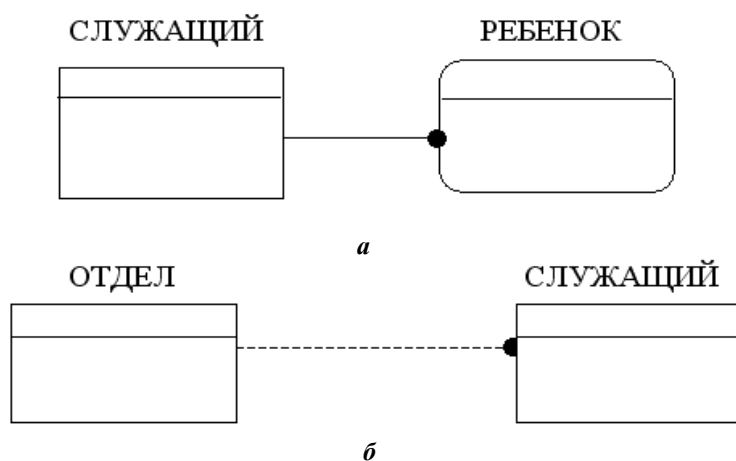


Рис.2. Графическое обозначение идентифицирующей (а) и неидентифицирующей (б) связей между сущностями

Методология IDEF1X позволяет представить изображаемые на ER-диаграммах классы принадлежности и степени связи (1:1, 1:N, N:1, N:M) с помощью идентифицирующей и неидентифицирующей связей и связи “многие ко многим” (рис.3).

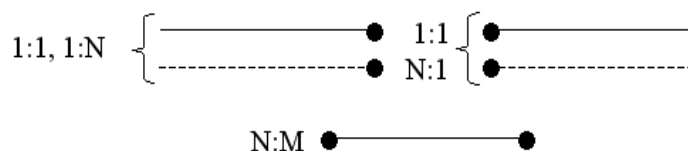


Рис.3. Представление степеней связи

Степень связи 1:N со стороны “многие” характеризуется мощностью (cardinality), которая обозначает количество экземпляров сущности-потомка, существующих для каждого экземпляра сущности-родителя. Мощность позволяет задать класс принадлежности (обязательный - О и необязательный - Н) для сущности-потомка и может принимать значения, указанные в табл.1.

Для сущности-родителя класс принадлежности графически не обозначается и не является существенным для определения соответствующей ДЛМ.

Таблица 1

Представление степеней связи и классов принадлежности

Мощность	Графическое обозначение связи (IDEF1X)	Степень связи	Класс принадлежности для СП и ER-диаграмма
N		1:N	
P		1:N	
Z		1:1	
n		1:1 1:n	

### ***Назначение, возможности и особенности программы ERwin***

Программа ERwin предназначена для построения ИЛМ с использованием методологии IDEF1X и автоматической генерации соответствующей ДЛМ с учетом особенностей выбранной СУБД. Результатом генерации ДЛМ является схема базы данных, представленная на языке SQL, и созданные таблицы, входящие в базу данных. ERwin может генерировать ДЛМ для различных СУБД.

Для обозначения моделей данных ERwin использует терминологию, отличную от рассмотренной в лекциях: ИЛМ именуется логической (Logical) моделью, а ДЛМ - физической (Physical) моделью.

Создание БД с помощью ERwin начинается с построения логической модели. После описания логической модели проектировщик выбирает необходимую СУБД, а ERwin автоматически создает соответствующую физическую модель. На основе физической модели ERwin генерирует схему БД на языке SQL и может сформировать таблицы, образующие БД, если успешно выполнено подключение к выбранной СУБД. Этот процесс называется прямым проектированием (Forward Engineering) и обеспечивает масштабируемость: создав одну логическую модель, можно сгенерировать физические модели для любой СУБД, поддерживаемой программой ERwin.

Кроме того, программа ERwin способна для существующей БД воссоздать физическую и логическую модели, т.е. обеспечить обратное

проектирование (Reverse Engineering). На основе полученной логической модели можно сгенерировать физическую модель для другой СУБД и затем сформировать новую БД. Следовательно, ERwin позволяет решить задачу по переносу структуры БД с одной СУБД на другую.

После запуска программы ERwin появляется окно программы (рис.4) со строкой главного меню, панелью инструментов, рабочей областью и палитрой инструментов с кнопками.

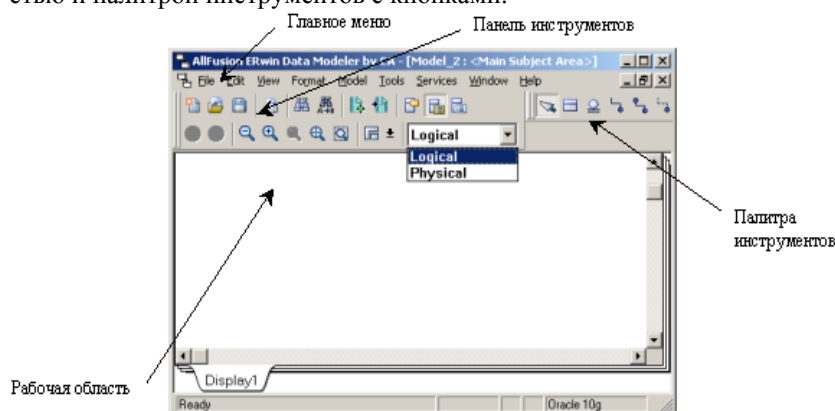



Рис.4. Окно программы ERwin


Вид палитры инструментов (Toolbox) зависит от выбора логической или физической модели, который осуществляется с помощью списка Logical-Physical, расположенного в правой части панели инструментов (см. рис.4). Для логической модели палитра инструментов имеет кнопки, назначение которых указано в табл.2.





Таблица 2

#### Кнопки палитры инструментов

Кноп-ка	Назначение
	Указатель элемента модели. Элементами модели являются сущности и связи

Окончание

Кноп-ка	Назначение
	Добавление сущности

	Добавление категориальной связи (используется для описания обобщенных объектов (сущностей))
	Добавление идентифицирующей связи “один ко многим”
	Добавление связи “многие ко многим”
	Добавление неидентифицирующей связи “один ко многим”

### **Использование программы ERwin для проектирования и создания базы данных**

Рассмотрим в качестве предметной области предприятие, в структуре которого имеются отделы, и спроектируем БД для хранения сведений о служащих, работающих в отделах, и их детях. Описание сущностей и связей между ними представлено на рис.5.

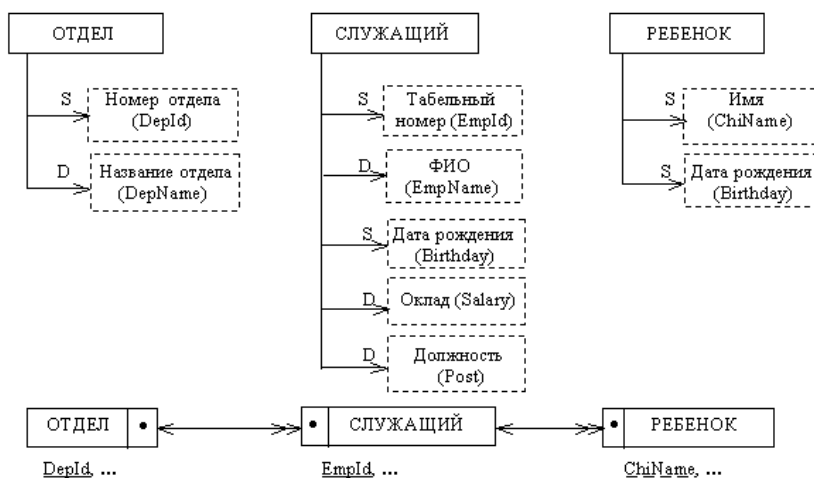


Рис.5. Описание объектов и связей между ними

На ER-диаграмме атрибут ChiName (Имя ребенка) подчеркнут штриховой линией, чтобы указать на то, что по имени можно идентифицировать ребенка, только “подчинив” его служащему-родителю, т.е. ребенок является зависимой сущностью по отношению к служащему.

(Сделать сущность РЕБЕНОК независимой можно, если пронумеровать всех детей и использовать их номера для однозначной идентификации.)

Для использования программы ERwin связь объектов, показанную на ER-диаграмме, необходимо представить в соответствии с методологией IDEF1X (см. табл.1), как показано на рис.6.

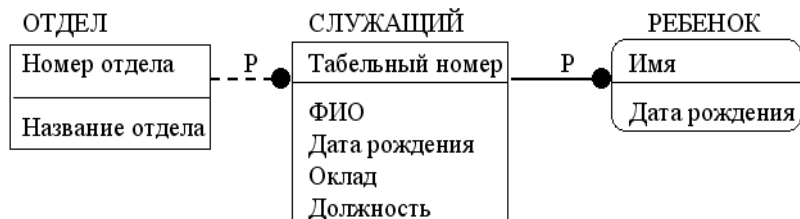


Рис.6. Логическая модель

Полученная диаграмма описывается средствами ERwin и помещается в файл с расширением erwin. После выбора СУБД программа ERwin формирует физическую модель БД как совокупность взаимосвязанных таблиц. Для последующего использования БД удобнее, чтобы имена таблиц и атрибутов записывались латинскими буквами, поскольку не все СУБД допускают работу с кириллицей. В табл.3 приведено соответствие между именами в логической и физической моделях и указаны типы атрибутов.

Таблица 3

**Соответствие между именами логической и физической моделей**

Имя		Имя		Тип	
сущности	таблицы	атрибута	столбца	атрибута	столбца
ОТДЕЛ	Department	Номер отдела	DepId	Числовой	Number
		Название отдела	Dep-Name	Строковый	String

*Окончание*


Имя		Имя		Тип	
сущности	таблицы	атрибута	столбца	атрибута	столбца
СЛУЖАЩИЙ	Employee	Табельный номер	EmpId	Числовой	Number
		ФИО	Emp-Name	Строковый	String

Имя		Имя		Тип	
сущности	таблицы	атрибута	столбца	атрибу- та	столбца
		Дата ро- ждения	Birthday	Дата	Datetime
		Оклад	Salary	Число- вой	Number
		Долж- ность	Post	Стро- ковый	String
РЕБЕ- НОК	Children	Имя	ChiName	Стро- ковый	String

Чтобы описать сущности, входящие в логическую модель, показанную на рис.6, нужно выполнить следующие действия.

1. На панели инструментов задать режим создания логической модели (Logical).

2. Командой File | New создать новую модель; в появившемся окне Create Model - Select Template указать тип модели Logical/Physical, выбрать шаблон Blank Logical/Physical Diagram и нажать кнопку ОК.

3. На палитре инструментов нажать кнопку , соответствующую сущности.

4. В рабочем поле диаграммы щелчком мыши разместить три сущности.

5. Курсором мыши указать на первую сущность в рабочем поле диаграммы, щелкнуть правой кнопкой мыши и выбрать из локального меню команду Entity Properties.

6. В области Name диалогового окна Entities набрать имя сущности Department, которое будет использоваться в модели вместо имени ОТДЕЛ.

7. На закладке Definition в одноименной области набрать определение сущности (Отдел предприятия) и нажать кнопку ОК. Таким способом на закладке Definition для сущности фиксируется информация, содержащаяся в лингвистических отношениях инфологической модели предметной области.

8. Щелкнуть правой кнопкой на сущности Department и выбрать из локального меню команду Attributes для описания атрибутов сущности.

9. В одноименном диалоговом окне (рис.7) нажать кнопку New и в появившемся диалоговом окне New Attribute (рис.8) указать имя атрибута, имя соответствующего столбца в таблице БД и тип данных, хранящихся в столбце (домен): в области Attribute Name набрать Номер отдела, в области Column Name - DepId, в области Domain щелчком мыши выбрать числовой тип Number.



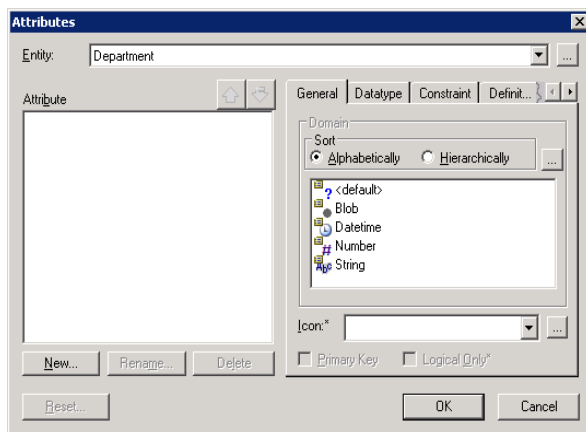


Рис. 7. Окно редактора атрибутов

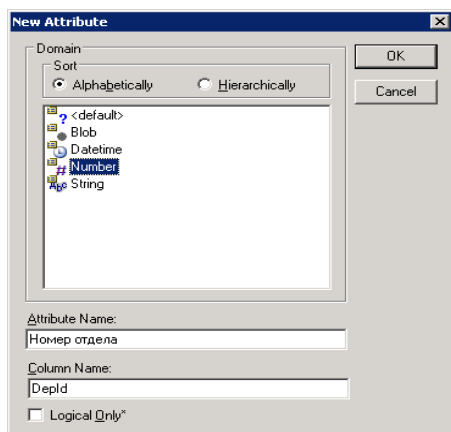


Рис. 8. Окно описания нового атрибута

10. Нажать кнопку ОК для возврата в диалоговое окно Attributes и на закладке General щелчком мыши установить признак Primary Key, чтобы отметить описанный атрибут как входящий в ключ сущности.

11. На закладке Definition для описываемого атрибута набрать определение атрибута: Номер отдела задается уникальным целым числом и используется для идентификации отдела. Таким способом на закладке Definition для атрибута фиксируется информация, содержащаяся в лингвистических отношениях инфологической модели предметной области.

12. Описать атрибут Название отдела согласно табл.3, повторив пп. 9 - 11, но без включения атрибута в состав ключа сущности.

13. Нажать кнопку ОК в диалоговом окне Attributes.

14. Описать сущности СЛУЖАЩИЙ и РЕБЕНОК согласно табл.3, выполнив действия, указанные в пп. 5 - 13, применительно к этим сущностям, и набрать определения сущностей и атрибутов на закладке Definition.

Чтобы задать связи между сущностями (см. рис.6), нужно выполнить следующие действия:

1. Соединить сущность ОТДЕЛ и СЛУЖАЩИЙ неидентифицирующей связью: на палитре инструментов нажать кнопку со штриховой линией, щелкнуть мышью по сущности ОТДЕЛ, а затем по сущности СЛУЖАЩИЙ. Таким образом будет установлена связь “один ко многим”. При этом ключ сущности-родителя появится среди атрибутов сущности-потомка и будет помечен как внешний ключ (FK).

2. Щелкнуть правой кнопкой мыши по связи и в локальном меню выбрать команду Relationship Properties, чтобы задать такие характеристики связи, как имя связи (Verb Phrase) и мощность (Cardinality).

3. В диалоговом окне Relationships (рис.9) в области Parent-to-Child задать имя связи со стороны сущности-родителя (*состоит из*), а в области Child-to-Parent - со стороны сущности-потомка (*работает в*).

4. Задать мощность связи, щелкнув мышью по радиокнопке One or More (P) в области Cardinality.

5. Задать обязательный класс принадлежности для сущности-потомка СЛУЖАЩИЙ, щелкнув в области Relationship Type по радиокнопке No Nulls (служащий обязательно работает в каком-либо отделе).

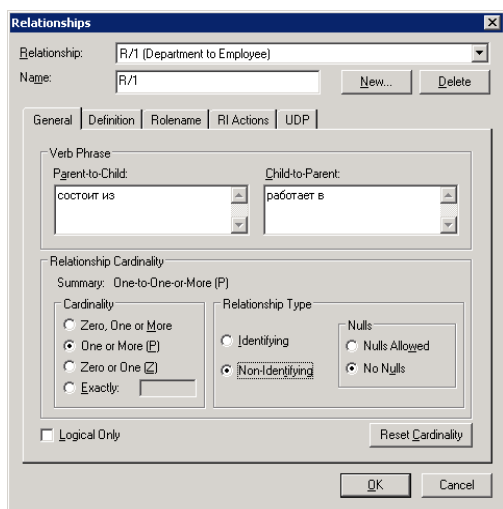


Рис. 9. Окно редактора связей

*Примечание.* Расположенная в области Relationship Type радио-кнопка Nulls Allowed соответствует необязательному классу принадлежности сущности-потомка и разрешает использование пустых значений NULL в качестве значений внешнего ключа, задающего связь служащего с отделом. На диаграмме такая “необязательная” неидентифицирующая связь помечается белым ромбом со стороны сущности-родителя. В физической модели, формируемой программой ERwin, такой связи двух сущностей соответствуют две таблицы, в одной из которых в столбце внешнего ключа будут пустые значения. Если наличие пустых значений недопустимо, то связь 1:N для сущности-потомка с необязательным классом принадлежности лучше заменить связью N:M, чтобы в физической модели информация о связях конкретных экземпляров сущности-родителя и сущности-потомка хранилась в отдельной таблице.

6. Нажать кнопку ОК в диалоговом окне Relationships и проверить, появились ли на диаграмме имя связи и обозначение мощности связи. Если эти характеристики отсутствуют, то для их отображения следует щелкнуть правой кнопкой мыши по любому свободному месту в рабочей области, в локальном меню выбрать команду Relationship Display и щелчком мыши отметить пункт Verb Phrase. Затем таким же способом отметить в локальном меню пункт Cardinality.

7. Соединить сущности СЛУЖАЩИЙ и РЕБЕНОК идентифицирующей связью: на палитре инструментов нажать кнопку со сплошной ли-

нией, щелкнуть мышью по сущности СЛУЖАЩИЙ, а затем по сущности РЕБЕНОК. Таким образом будет установлена связь “один ко многим”, сущность РЕБЕНОК будет изображена как зависимая, и ключ сущности-родителя появится среди атрибутов сущности-потомка, образующих ее ключ, и будет помечен как внешний ключ (FK).

8. Задать имя (*имеет/принадлежит*) и мощность (P) связи так, как указано в пп. 2 - 4.

Вид диаграммы в рабочей области для созданной логической модели показан на рис.10.

Для полученной логической модели можно легко сформировать со-

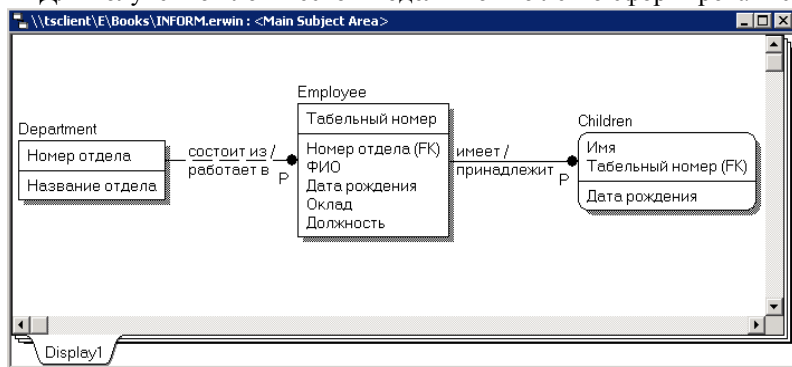


Рис.10. Диаграмма для созданной логической модели

ответствующую физическую модель, раскрыв список (см. рис.4), расположенный справа на панели инструментов, и выбрав из него тип модели Physical.

*Примечание.* Имеющиеся в логических моделях связи “многие ко многим” в физических моделях должны быть преобразованы. Преобразование связи заключается в создании новой таблицы и двух новых связей “один ко многим” от таблиц, соответствующих связанным сущностям в логической модели, к новой таблице. По умолчанию ERwin автоматически такое преобразование при переходе от логической модели к физической модели не выполняет. Режим автоматического преобразования связи “многие ко многим” задается командой главного меню Model | Model Properties, активизирующей окно, в котором на закладке General следует поставить галочки в переключателях Show source objects in logical, target objects in physical и Auto apply Many-To-Many transform. Принудительное преобразование связи “многие ко многим” задается щелчком по связи правой кнопкой мыши и выбором в локальном

меню команды Create Association Table, которая активизирует Мастер преобразования связи.

Чтобы адаптировать спроектированную базу данных для СУБД Microsoft SQL Server, нужно выполнить следующие действия.

1. В главном меню выбрать команду Database | Choose Database, в появившемся диалоговом окне (рис.11) ознакомиться с перечнем СУБД (Target SQL DBMS), поддерживаемых программой ERwin, затем щелчком мыши выбрать радиокнопку SQL Server и нажать кнопку OK.

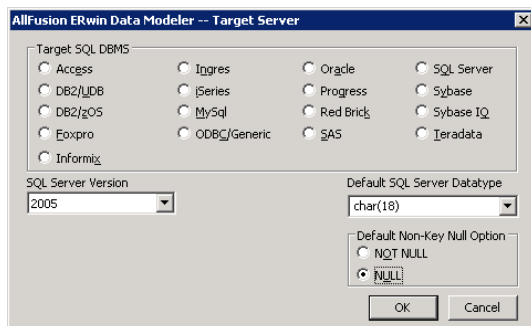


Рис.11. Окно выбора СУБД

2. В ответ на запрос о преобразовании типов данных нажать кнопку Yes.

3. Щелкнуть правой кнопкой мыши по таблице Department и выбрать в локальном меню команду Columns для уточнения свойств столбцов таблицы.

4. В диалоговом окне Columns (рис.12) щелкнуть мышью по закладке SQL Server, чтобы учесть особенности СУБД SQL Server.

5. В области Column выделить столбец DepId и на закладке SQL Server задать тип numeric (6,0), а затем выделить столбец DepName и задать тип nchar (20). На этой же закладке задать признак запрета (NOT NULL) или разрешения (NULL) неопределенных значений в столбце и, если необходимо, - автоинкрементальное изменение значений (IDENTITY).

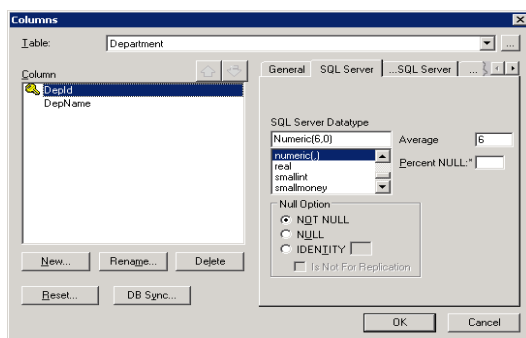


Рис.12. Окно редактора столбцов таблицы

6. Допустимые значения и значения по умолчанию задать на закладке Constraint в областях Validation Constraint и Default соответственно (рис.13). Чтобы задать условие для проверки допустимости значений и значение по умолчанию, следует нажать кнопку ..., расположенную справа от полей Valid и Default, для активизации окна, в котором создается новое имя для условия или значения по умолчанию, а также записывается условие на языке SQL (Validation Rule, например, DepId > 0) или вводится необходимая константа (Default Value, например, 1).

7. Нажать кнопку OK для сохранения изменений.

8. Выполнить пп. 3 - 7 для таблиц Employee и Children, выбрав

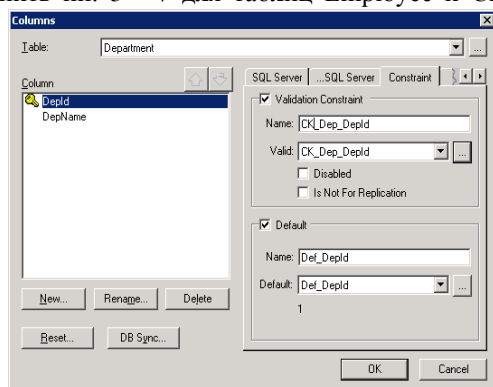


Рис.13. Закладка Constraint

самостоятельно размер и разрядность строковых и числовых столбцов.

Чтобы для полученной физической модели типы данных, заданные для столбцов, отображались на диаграмме, следует щелкнуть правой

14

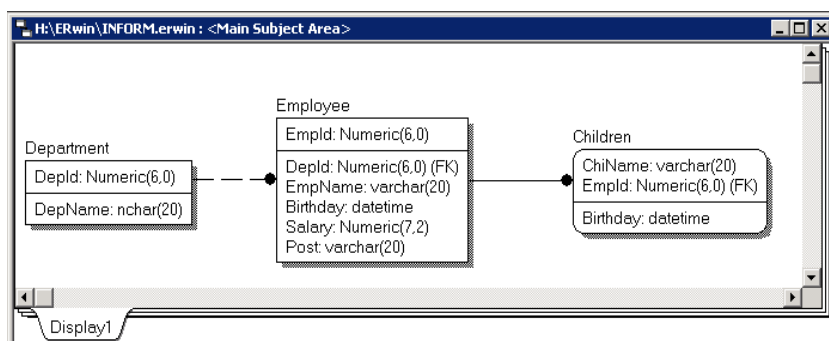


Рис.14. Диаграмма для физической модели

кнопкой мыши по любому свободному месту в рабочей области, в локальном меню выбрать команду Table Display и щелчком мыши отметить пункт Column Datatype. После этого диаграмма приобретет вид, показанный на рис.14.

После этого создать БД для выбранной СУБД можно командой главного меню Tools | Forward Engineering | Schema Generation. Эта команда активизирует диалоговое окно (рис.15), в котором представлены режимы генерации схемы БД, включающей описания таблиц, столбцов, индексов и т.д. Нажав кнопку Preview, можно вызвать окно, в котором отображаются операторы языка SQL, формируемые программой ERwin для создания объектов БД и задающие схему БД (рис.16).

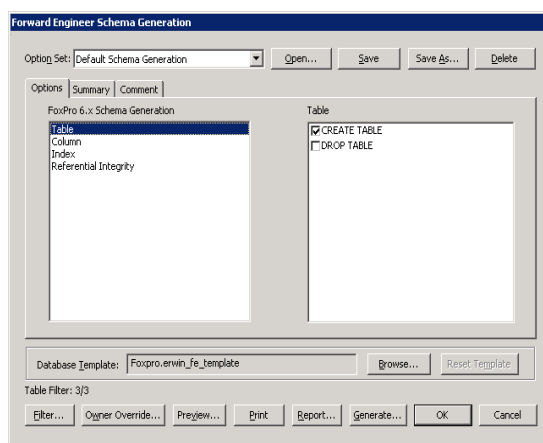


Рис.15. Окно генерации схемы БД

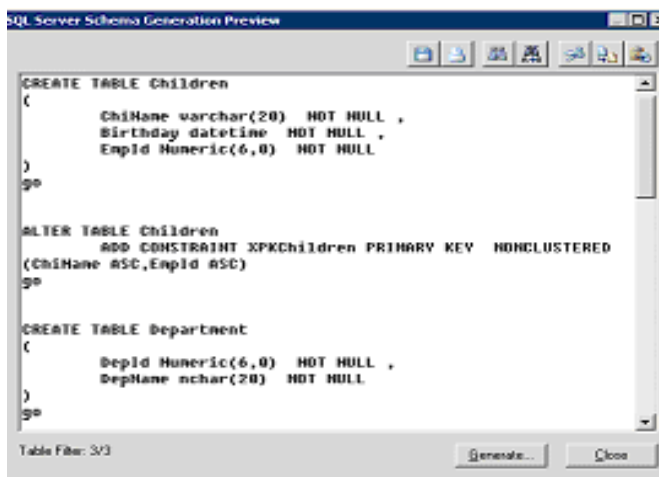


Рис.16. Окно просмотра схемы БД

Создание БД задается нажатием кнопки Generate. При этом появляется диалоговое окно, предназначенное для подключения к СУБД, которая будет создавать объекты БД (рис.17). После настройки параметров подключения Server и Database нажатием кнопки Connect

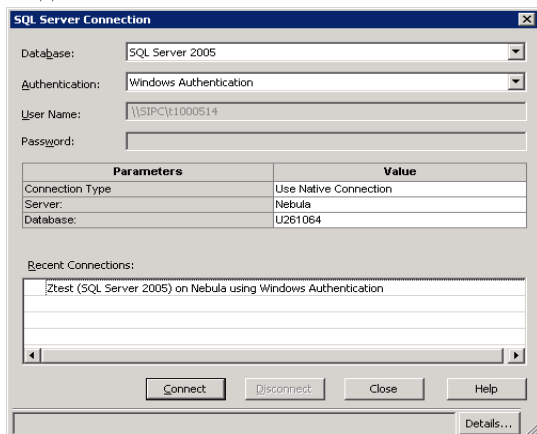


Рис.17. Окно параметров подключения к СУБД

запускается процесс создания объектов БД (таблиц, индексов и др.). Протокол создания объектов БД отображается в окне Generate Database Schema (рис.18).



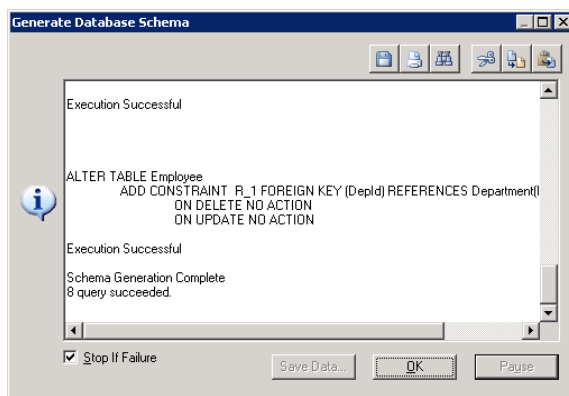


Рис.18. Протокол создания объектов БД

После создания БД нужно последовательно нажать кнопки ОК, Close, ОК в активных диалоговых окнах.

### ***Вычисление размера базы данных***

ERwin позволяет рассчитать приблизительный размер базы данных в целом, а также таблиц, индексов и других объектов через определенный период времени после начала эксплуатации БД. Для расчета размера объектов БД предназначена команда главного меню Tools | Volumetrics, которая активизирует окно Volumetrics (рис.19), содержащее три закладки Settings, Report и Parameters.

Закладка Settings служит для задания основных параметров, на основе которых вычисляется размер базы данных. Для таблицы, выбранной в левом списке Table, в группе Table Row Counts задается начальное количество строк (Initial), максимальное количество строк (Max) и прирост строк в месяц (Grow By). Если параметры Max и Grow By используются одновременно, то рост размера таблицы прекращается при достижении максимального размера. После задания параметров Initial, Max и Grow By в группе Sizing Estimates, расположенной в левом нижнем углу окна Volumetrics, отображается средний размер строки, начальный размер таблицы и индексов.

Табличная форма Column Properties позволяет задать свойства столбцов (полей) таблицы, выбранной в левом списке Table. Имена столбцов, их тип и размер не редактируются. Можно изменять среднюю ширину поля ( для тех типов, для которых это допускается) и параметр Pct NULL (ожидаемый средний процент строк таблицы, в которых поле

принимает значение NULL). ERwin в зависимости от выбранной СУБД автоматически определяет, какие ячейки табличной формы Column Properties доступны для ввода.

Группа Include Indexes позволяет учесть или игнорировать при вычислении размера БД индексы, создаваемые для внешних (FK), первичных (PK) и альтернативных (AK) ключей.

Группа Storage позволяет задать объект физической памяти, в котором будет храниться выбранная таблица. Если объект не описан, то его можно определить, воспользовавшись кнопкой [...]

Закладка Report (рис.20) содержит результаты вычисления размера

**Volumetrics: <Main Subject Area>**

Settings | Report | Parameters

Table:  
CHILDREN  
DEPARTMENT  
EMPLOYEE

Sizing Input  
Table Row Counts  
Initial: 100 Max: 200 Grow By: 5 per Month

Column Properties:

Column	Datatype	Alloc	Avg Width	Pct NULL
BIRTHDAY	DATE	7		
CHINAME	VARCHAR2(20)	20	20	
EMPID	NUMBER(6)	3	6	

Sizing Estimates  
Average Row Size: 30  
Initial Table Size: 2 K  
Initial size of Index(es): 2 K

Include Indexes  
☒ PK ☒ AK  
☒ FK ☒ IE

Storage  
Physical Object: ...  
USERS

OK Cancel

Рис.19. Окно Volumetrics с закладкой Settings

**Volumetrics: <Main Subject Area>**

Settings | Report | Parameters

Options  
Report  
☐ Physical Objects  
☒ Database Objects  
☐ Tables

Time  
☐ Initial  
☒ Projections  
Months: 12

Send to Data Browser...

Physical Objects:

Physical Object	Table/Index	Type	Size
USERS		Total	0 K
<default>	CHILDREN	Table	3 K
<default>	XPKCHILDREN	Index	2 K
<default>	DEPARTMENT	Table	0 K
<default>	XPKDEPARTMENT	Index	0 K
<default>	EMPLOYEE	Table	0 K
<default>	XPKEMPLOYEE	Index	0 K
<default>	Total		5 K
	Grand Total		5 K

OK Cancel

Рис.20. Окно Volumetrics с закладкой Report

БД. Группа Options позволяет выбрать тип объектов, по которым проводятся вычисления, а группа Time – момент для оценки размера БД (Ini-

tial - начальное состояние; Projections – определенное время после начала эксплуатации БД). Полученные результаты можно отправить в генератор отчетов Data Browser, нажав кнопку Send to Data Browser.

Закладка **Parameters** (рис.21) служит для задания дополнительных параметров, используемых для оценки размера:

- **TableFactor** показывает накладные расходы на хранение таблицы в БД. Например, значение 2 увеличит размер таблицы вдвое;
- **IndexFactor** показывает накладные расходы на хранение индекса в БД. Например, значение 1.5 увеличит размер индекса с 1 до 1,5 Мбайт;
- **RowOverhead** используется для дополнительного пересчета количества байт в каждой строке. Например, значение 10 увеличит размер каждой строки таблицы на 10 байт;
- **BlobFactor** и **BlobBlockFactor** используются для пересчета размера столбцов Blob-типа, хранящихся физически вне базы данных;
- **BytesPerChar** используется для задания количества байт, необходимых для хранения одного символа строкового типа. Например, для кода ASCII – это 1 байт, для Unicode – 2 байта;
- **LogPercent** используется для вычисления размеров log-файлов базы данных. Например, значение 100 увеличивает размер вдвое.

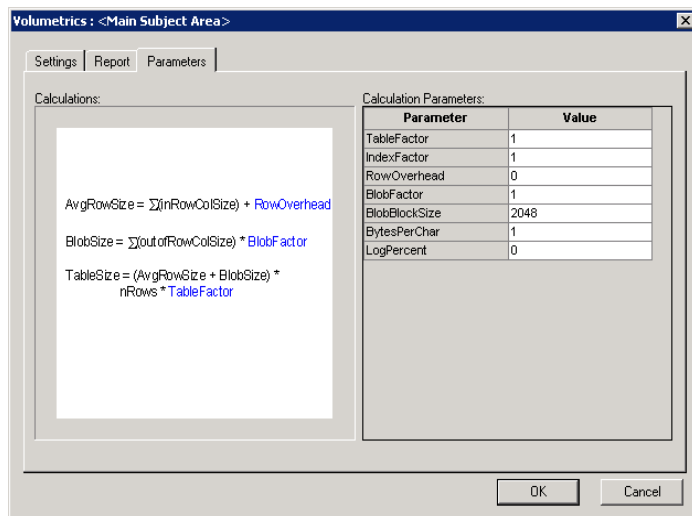


Рис.21. Окно Volumetrics с закладкой Parameters

## Документирование моделей

Для документирования моделей в программе ERwin имеется простое в использовании средство - Data Browser, которое позволяет формировать стандартные отчеты (сгруппированные по типам), создавать собственные отчеты, печатать и экспортировать их в распространенные текстовые форматы. Data Browser вызывается командой главного меню Tools | Data Browser.

Окно генератора отчетов Data Browser (рис.22) имеет собственное меню и панель инструментов. В левой области окна отображается дерево отчетов. Каждый отчет может включать несколько результирующих наборов данных, сгенерированных при формировании отчета.

Для формирования отчета достаточно дважды щелкнуть по нему в дереве отчетов или нажать кнопку с зеленым треугольником на панели инструментов. Результат будет отображен в правой области окна

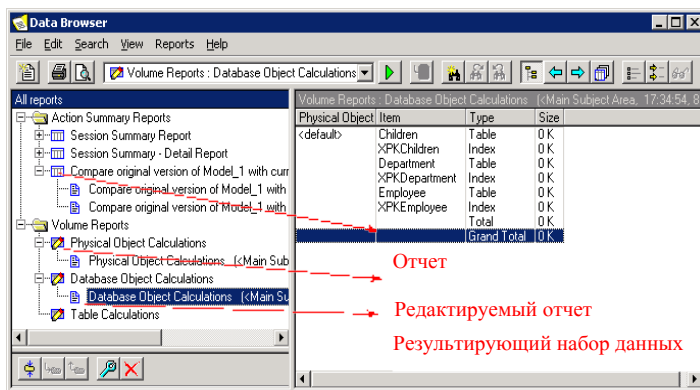


Рис.22. Окно генератора отчетов Data Browser

генератора отчетов, а иконка результирующего набора данных будет добавлена в дерево отчетов. Под областью с деревом отчетов расположены кнопки для управления этим деревом. В частности, для удаления отчета служит кнопка с красным крестиком.

Создание нового отчета задается командой File | New Report, которая активизирует окно Reports, где указываются параметры отчета (рис.23). Переключатели Logical/Physical задают модель, для которой создается отчет, в поле Name набирается имя отчета, в поле Category

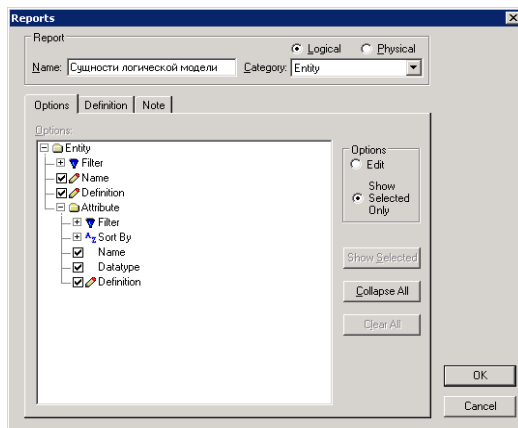


Рис.23. Окно Reports с параметрами отчета

указывается тип элементов модели, по которым будет создаваться отчет (атрибуты, сущности, домены, связи и т.д.). Закладки Definition и Note служат для внесения определения и комментария к отчету.

Закладка Options содержит представляемую в отчете информацию о модели. В левой части закладки находится иерархический список характеристик того типа элементов модели, который указан в поле Category. Для включения характеристики в отчет следует щелкнуть мышкой по квадратику, расположенному слева от наименования характеристики, чтобы в квадратике появилась галочка. Параметры, указанные на рис.19, относятся к отчету, содержащему такие сведения о сущностях логической модели, как имена сущностей и их определения, а также имена, типы и определения атрибутов каждой сущности. После нажатия на кнопку ОК новый отчет добавляется в дерево отчетов (рис.24). Формирование созданного отчета запускается двойным щелчком по нему или нажатием кнопки с зеленым треугольником на панели инструментов, а результат отображается справа от дерева отчетов (см. рис.24).

Полученный после формирования отчета результат

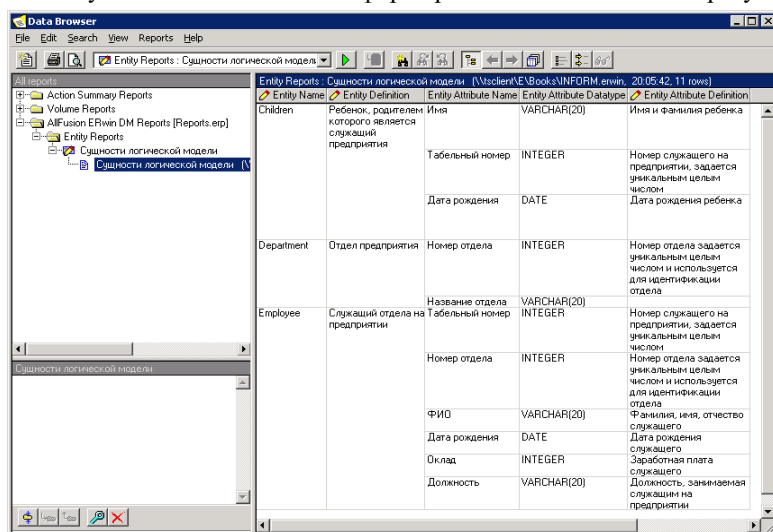


Рис.24. Окно генератора отчетов с созданным отчетом

(результатирующий набор данных) можно отредактировать, распечатать, экспортировать или сохранить. Для экспорта можно использовать команду File | Export или вызвать локальное меню для экспортируемого

результата и выбрать в нем команду Export result set, активизирующую окно, в котором задается формат экспорта (рис.25):

HTML - гипертекстовый файл;

DDE - экспорт в Word или Excel;

CSV - текстовый файл;

RPTwin - экспорт в специализированный генератор отчетов;

BRS file - файл формата Browser Result Set (для последующего импорта в Data Browser командой File | Import).

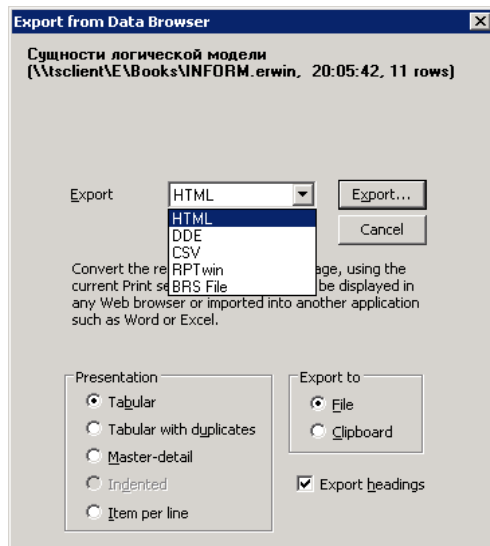


Рис.25. Окно выбора формата экспорта отчета

После выбора формата процесс экспорта запускается нажатием кнопки Export.

### Лабораторное задание

Спроектировать БД для конкретного варианта задания (см. лабораторную работу № 2), выбрав СУБД Microsoft SQL Server.

## Порядок выполнения работы

1. Подключиться к Терминалу 4100 и запустить программу ERwin командой Start | All Programs | Computer Associates | AllFusion | ERwin Data Modeler | ERwin Data Modeler.

2. Описать логическую модель, показанную на рис.6, следуя инструкциям, приведенным в разделе “Использование программы ERwin для проектирования и создания базы данных”. При описании модели для сущностей и атрибутов на закладке Definition указать их назначение.

3. Сохранить логическую модель в файле INFORM.erwin на устройстве H командой File | Save As.

4. Ознакомиться с уровнями отображения диаграммы, вызвав локальное меню щелчком правой кнопки мыши; выбрать команду Display Level и уровень отображения: сущности (Entity), атрибуты (Attribute), первичный ключ (Primary Key), определение (Definition), иконки (Icon). Записать в отчет, как представлена на диаграмме сущность СЛУЖАЩИЙ при выборе каждого уровня отображения.

5. Сформировать физическую модель, выбрав СУБД Microsoft SQL Server, и записать в отчет сведения о структуре спроектированной БД, указав таблицы и их связи.

6. Сохранить схему спроектированной БД на языке SQL в файле INFORM.sql, воспользовавшись кнопкой Save в окне просмотра схемы БД (см. рис.16), и создать представленные в схеме объекты в БД с именем U<№СТУДБИЛЕТА> на сервере Microsoft SQL Server 2008.

7. Проверить, появились ли таблицы Department, Employee, Children в базе данных, и ввести в каждую из них по две строки, чтобы убедиться в действии ограничений целостности и установке начальных значений, заданных при проектировании базы данных.

8. Закрыть модель командой главного меню File | Close.

9. Для варианта задания, выполненного в лабораторной работе № 2, разработать логическую модель, описав сущности и их атрибуты и представив связи сущностей в соответствии с методологией IDEF1X. Для сущностей и атрибутов на закладке Definition указать их назначение и особенности. Записать в отчет полученную логическую модель и сохранить ее в файле MYMODEL.erwin на устройстве H:.

10. Использовать программу ERwin для проектирования физической модели БД, соответствующей варианту задания, выбрав СУБД Microsoft SQL Server. Записать в отчет структуру сформированной физической модели БД и сохранить схему БД на языке SQL в файле MYMODEL.sql. **Объекты, представленные в схеме БД, на сервере Microsoft SQL Server 2008 не создавать.**

11. Сравнить сохраненную схему БД с описанием таблиц, сохраненным в файле сценария при выполнении лабораторной работы № 2. Результаты сравнения записать в отчет.

12. С помощью программы ERwin определить размер базы данных перед началом ее эксплуатации и оценить изменение размера базы данных в течение годовичного срока эксплуатации. Результаты записать в отчет.

13. Средствами документирования моделей сформировать показанный на рис.24 отчет для логической модели, сохраненной в файле INFORM.erwin. Экспортировать результат в гипертекстовый файл.

14. Документировать логическую модель, сохраненную в файле MYMODEL.erwin, создав отчет со списком сущностей и их атрибутов. Экспортировать результат в гипертекстовый файл.

15. Документировать физическую модель, создав отчет со списком таблиц и их столбцов с указанием типов. Экспортировать результат в гипертекстовый файл.

16. Оформить отчет по лабораторной работе и защитить работу.

### **Требования к отчету**

Отчет должен содержать:

- 1) название лабораторной работы;
- 2) результаты выполнения заданий, указанных в разделе “Порядок выполнения работы”.