

Лабораторная работа № 1

Проведение системного анализа предметной области.

Разработка логической и физической моделей БД.

Цель работы:

Закрепление и более глубокое усвоение знаний по проектированию информационных моделей.

Методика выполнения работы:

Структурное проектирование с использованием IDEF позволяет построить так называемую модель требований (логическую модель) системы, состоящую из множества взаимоувязанных диаграмм, текстов и словаря данных. Эта модель описывает что должна делать проектируемая система без ссылок на то, как это достигается.

В процессе проектирования системы разрабатываются последовательно ER (сущность-связь), КВ (ключевой уровень) и физическая модели.

Эти модели обеспечивают:

- разработку документации базы данных;
- разработку ссылочной целостности БД;
- разработку логической модели БД независимой от конкретного типа СУБД;

- разработку документирования физического проектирования БД в соответствии с бизнес требованиями.

Физическая модель позволяет:

- обеспечить администратору БД достаточность информации, чтобы создать эффективную БД;
- создать контекст для процессов определения и записи в словари данных;
- ассистировать группам приложений в выборе физической структуры программы, которая будет запрашивать данные.

При переходе от логической модели Erwin каждой ее опции ставит соответствующую опцию физической модели.

1 Пример описания проекта.

Задание: разработать информационную модель реализации настольных компьютеров по заказам клиентов.

Для описания объектов предметной области по реализации настольных компьютеров по заказам клиентов выделены следующие сущности: **Тип компьютера, Компьютер, Клиент, Заказ, Продажа, Менеджер, Отдел** (рис. 1)

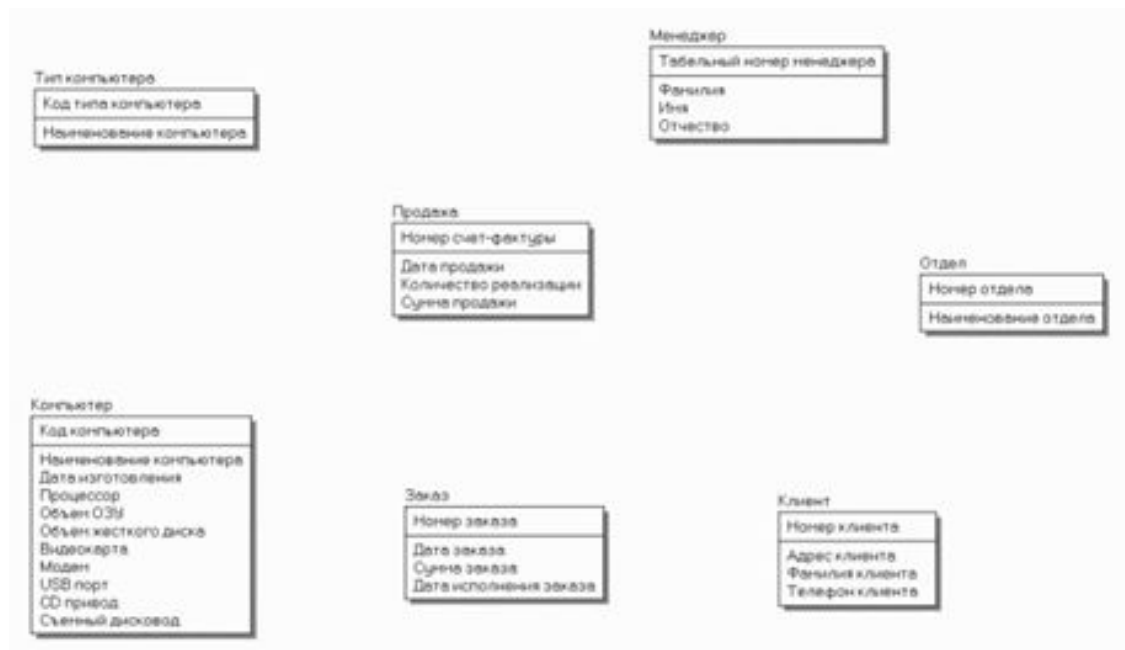


Рисунок 1. Сущности, выделенные для описания объектов предметной области по реализации настольных компьютеров по заказам клиентов

Логическим соотношением между сущностями является *связь*. Каждому виду связи соответствует определенная кнопка, расположенная на палитре инструментов. Имя связи выражает некоторое ограничение или бизнес-правило и облегчает чтение диаграммы. Каждая связь должна именоваться глаголом или глагольной фразой. Например, связь между сущностями **Компьютер**, **Продажа** и **Менеджер** показывает (рис. 2), какой компьютер продан и какой менеджер оформил сделку продажи компьютера: – каждый **Компьютер** < продается > **Продажа**; – каждая **Продажа** < оформляет > **Менеджер**.



Рисунок 2. Связь между сущностями Компьютер, Продажа и Менеджер

В нотации IDEF1X различают *зависимые* и *независимые* сущности. Тип сущности определяется ее связью с другими сущностями. *Идентифицирующая связь* устанавливается между независимой (родительский конец связи) и зависимой (дочерний конец связи) сущностями и показывается на диаграмме сплошной линией с жирной точкой на дочернем конце связи. При установлении идентифицирующей связи ERwin автоматически преобразует дочернюю сущность в зависимую сущность. Зависимая сущность на диаграмме изображается прямоугольником со скругленными углами, например, сущность **Продажа** (см. рис. 2). Экземпляр зависимой сущности определяется только через отношение к родительской сущности. Например, информация о продаже не может быть внесена и не имеет смысла без информации о проданном компьютере. При установлении *идентифицирующей связи* атрибуты первичного ключа родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности.

Операция дополнения атрибутов дочерней сущности при создании связи называется *миграцией атрибутов*. В дочерней сущности новые (мигрированные) атрибуты помечаются как внешний ключ (FK). В случае идентифицирующей связи при генерации схемы базы данных атрибутам внешнего ключа присваивается признак NOT NULL, что означает невозможность внесения записи в таблицу, соответствующей дочерней сущности, без идентификационной информации из таблицы, соответствующей родительской сущности. Например, невозможность внесения записи в таблицу продаж без информации об идентификационном номере проданного компьютера, определенного в таблице описания имеющихся в наличии компьютеров.

Неидентифицирующая связь показывается на диаграмме пунктирной линией с жирной точкой и служит для установления связи между независимыми сущностями. При установлении неидентифицирующей связи дочерняя сущность остается независимой, а атрибуты первичного ключа мигрируют в состав неключевых атрибутов родительской сущности (рис. 3).



Рисунок 3. Пример неидентифицирующей связи между независимыми сущностями
Менеджер и Отдел

Для создания новой связи необходимо:

- установить курсор на кнопке, расположенной на палитре инструментов и соответствующей требуемому виду связи, и нажать левую кнопку мыши;
- щелкнуть левой кнопкой мыши сначала по родительской, а затем по дочерней сущности.

Изменить форму линии связи и ее местоположение между связанными сущностями можно путем захвата мышью выделенной линии связи и переноса ее с места на место, пока линия не примет необходимые местоположение и форму.

Редактирование свойств связи осуществляется в диалоговом окне **Relationship**, которое открывается через пункт **Relationship Properties** контекстного меню, активизируемого посредством нажатия правой кнопки мыши на выделенной связи.

Вкладка **General** диалогового окна **Relationship** позволяет задать мощность, имя и тип связи.

Мощность связи (Cardinality) служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней сущности. Различают 4 типа мощности связи:

- *общий случай* – не помечается каким-либо символом и соответствует ситуации, когда одному экземпляру родительской сущности соответствуют 0, 1 или много экземпляров дочерней сущности;
- *символом P* помечается случай, когда одному экземпляру родительской сущности соответствуют 1 или много экземпляров дочерней сущности, т.е. исключено нулевое значение;
- *символом Z* помечается случай, когда одному экземпляру родительской сущности соответствуют 0 или 1 экземпляр дочерней сущности, т.е. исключены множественные значения;
- *цифрой* помечается случай точного соответствия, когда одному экземпляру родительской сущности соответствует заранее заданное число экземпляров дочерней сущности.

Результат разработки логической модели данных системы "Реализация средств вычислительной техники", предназначенной для учета продаж настольных компьютеров по заказам клиентов приведен на рис. 4.

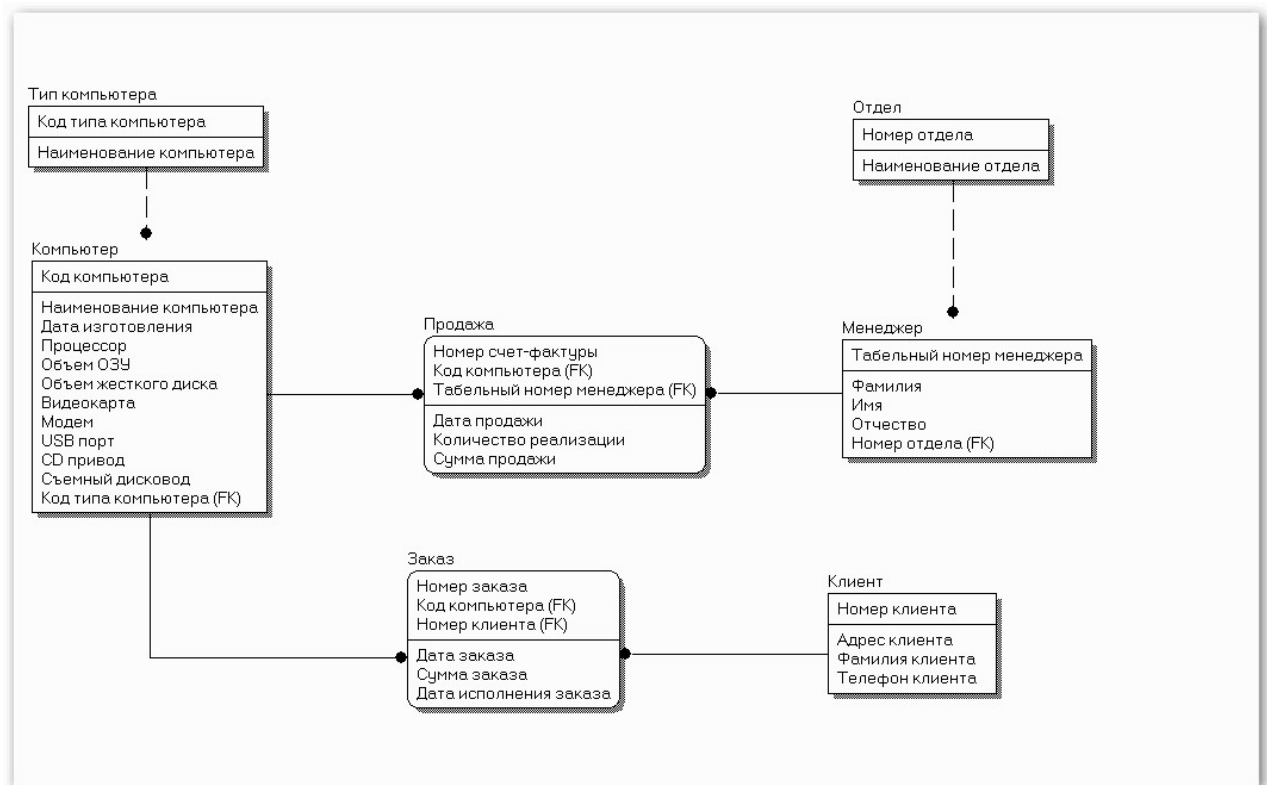


Рисунок 4. Логическая модель данных системы «Реализация средств вычислительной техники»

Созданная логическая модель данных системы является основанием для создания физической модели данных под выбранную СУБД.

Физический уровень представления модели зависит от конкретной реализации СУБД, поэтому необходимо предварительно осуществить ее выбор. CA ERwin Data Modeler 7.3 поддерживает 17 наиболее распространенных СУБД. Выбор СУБД осуществляется в окне **Target Database** диалога **Create Model**. Для выбора СУБД необходимо открыть выпадающий список с перечнем поддерживаемых СУБД и щелкнуть по соответствующему имени. При этом в поле **Version** отобразится версия выбранной СУБД.

В случае автоматического перехода к физической модели ERwin генерирует имена таблиц и колонок по умолчанию на основе имен соответствующих сущностей и атрибутов логической модели, учитывая максимальную длину имени и другие синтаксические ограничения, накладываемые выбранной СУБД. При этом правила ссылочной целостности, принятые на логическом уровне модели данных системы, также принимаются по умолчанию, т.е. сохраняются.

Построение физической модели данных для системы "Реализация средств вычислительной техники" осуществлено путем автоматического перехода от логической модели к физической модели, так как при создании логической модели данных системы был выбран логико – физический тип модели.

Физическая модель данных системы "Реализация средств вычислительной техники" приведена на рис. 5.

Для генерации кода создания базы данных можно использовать пункт главного меню Tools/Forward Engineer.

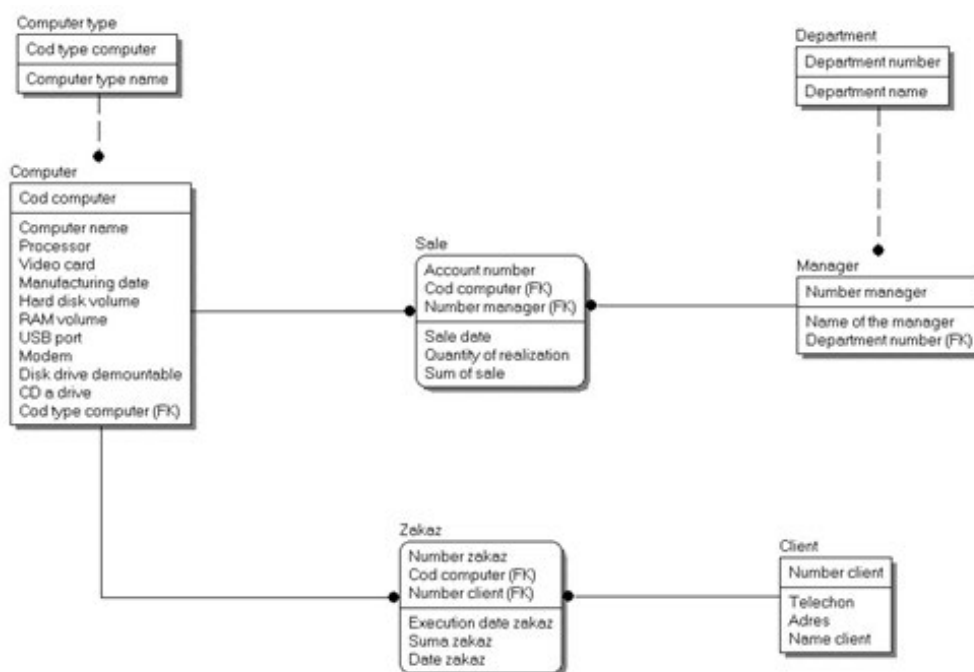


Рисунок 5. Физическая модель данных системы "Реализация средств вычислительной техники"

Порядок выполнения работы.

Настройка работы.

1. Запускается CASE - система Erwin.
2. Осуществляется настройка работы для работы с русским языком и конкретным СУБД.
3. Осуществляется настройка Ссылочной Целостности (Referential Integrity). Для этого из скролинг- меню (Identifying) выбираются опции для отношений IDENTIFYING (Ch.Del- Par.Upd) соответственно (N, Restr., Restr., Restr., None, Restr.). (Здесь опции Cascad указывает, что все атрибуты, указанные в сущности родителя для удаления, одновременно удаляются и у наследуемых сущностей).

Логическое проектирование.

Задача 1. Проектирование ER диаграммы.

Цель: построение диаграммы уровня “сущность- связь” и глоссария к ней.

Последовательность действий:

- выделить сущности и присвоить им уникальные имена;
- занести в глоссарий модели формальное определение имен сущностей.

Требования к диаграмме и глоссарию:

- сущности должны быть представлены на диаграмме только именами;
- допускаются зависимые и независимые сущности;
- глоссарий должен содержать краткие формальные определения сущностей.

Порядок разработки ER диаграммы.

- 1 Выбирается из ErWin Toolbox тип сущности зависимая (если она потомок) или независимая (если она родитель).
- 2 Щелкнуть по месту расположения сущности на рабочем поле.
- 3 Выбрать в Toolbox стрелку Select. Дважды щелкнуть по полю сущности. В открывшемся редакторе Entity- Attribute Editor в окне Entity записать имя сущности. ОК.

Задача 2. Проектирование отношений.

Цель: проектирование и осмысление характера взаимосвязей между сущностями.

Последовательность действий:

1. определить тип отношения;
2. указать имя отношения.

Требования к диаграмме:

- имена связей должны выбираться в глагольной форме, так, чтобы диаграмма читалась осмысленными фразами русского языка;
- при разработке допускаются определенные, неопределенные, связи типа “many-many” и категоризационные связи.

Порядок проектирования диаграммы.

- 1 Устанавливаются в поле экрана сущности Родителя и Потомка. Щелкнуть по полю тип отношений ErWin Toolbox (определенное, неопределенное, неспецифическое).
- 2 Щелкнуть сначала по полю сущности Родителя, а затем по полю сущности Потомка.
- 3 Дважды щелкнуть по отношению. В открывшемся редакторе отношений в окне Verb Phrase вводят глагольную форму, связывающую обе сущности в смысловое предложение. Триггерами Кардинальности указывается мощность отношения (Р, Z или число).

Обычно ставится максимальная. В дисплейном окне Rolename (можно изменить ролевое назначение ключа). В окне Foreign Key указывается (название ключа наследования).

4 Чтобы наблюдать на рабочем экране имя отношения необходимо выбрать в панели меню DISPLAY/ Verb Phrase.

5 В случае необходимости, чтобы добавить в схему отношение типа категории между Родителем и двумя Потомками необходимо:

- выбрать тип категории (категория с одной линией показывает, что имеются другие сущности категории, которые не включены в схему, категория с двумя линиями указывает, что все категории учтены);
- щелкнуть последовательно сначала по сущности Родителя, а затем по сущности одного из Потомков;
- щелкнуть по месту установки знака отношения категории, а затем щелкнуть по оставшейся сущности.

Задача 3. Проектирование КВ диаграммы.

Цель: проектирование идентификаторов сущностей и уяснение логики взаимосвязей на уровне идентификаторов.

Последовательность действий:

- преобразовать все неспецифические отношения в специфические;
- определить возможные ключи независимых сущностей и выделить первичные ключи;
- ввести формальные определения имен ключевых атрибутов в глоссарий;
- показать первичные и все возможные ключи на диаграмме;

Требования к диаграмме:

- на диаграмме допускаются только специфические и категоризационные связи;
- глоссарий должен содержать формальные определения ключей.

Физическое проектирование.

Задача 4. Физическое проектирование.

Цель: разработка структуры реляционной базы данных для выбранной СУБД.

Последовательность действий:

1. поставить в соответствие именам сущностей и атрибутов логической модели имена таблиц и полей БД (физические имена);
2. разработать сопроводительную документацию.

Требования к диаграмме:

1. Проектируются типы данных в соответствующем СУБД. Для этого включается в верхней линейке пиктограмма PHYSICAL VIEW,
2. Выбирается указателем мышки проектируемая сущность, нажатой правой выбирают Вариант СУБД DATABASE, в редакторе выбирают тип данных,
3. размер записи, ОК.

4. Описываются свойства связей между таблицами, в частности, реакцию сервера на попытки нарушения ссылочной целостности со стороны клиентского приложения. Для этого выделив связь, щелкнуть правой кнопкой и выбрать в редакторе отношений опцию RELATIONSHIP INTEGRITY, при этом можно отредактировать ее VERB PHRASE, установить триггеры запрещения действий клиентскому приложению по удалению записей CHILD DELETE- NONE, PARENT DELETE- NONE, а все остальное - RESTRICT.

После этого, можно создавать на сервере схему БД. Для этого отредактировать тип данных (войти в какую нибудь сущность и правой кнопкой щелкнуть, а затем выбрать Attribute Editors, ввести редактируемую сущность в окне Entity), введя типы для атрибутов и ключей. Ввести определения атрибутов и имена владельца данных. Вызвать Domain Editor (Column Property Editor/ Domain). Ввести имена доменов и присвоить им типы (длина) и определения.

Задание

1. Провести системный анализ предметной области согласно вашему варианту.
2. Разработать концептуальную модель данных.
3. Разработать в логическую модель данных.
4. Разработать физическую модель данных.

При разработке можно использовать CASE-систему ErWin Data Modeler (или аналогичную).

Полученный набор отношений должен находиться в третьей или выше нормальной форме и содержать не менее 5-ти сущностей.

Контрольные вопросы

1. В чем разница между концептуальной, логической и физической моделями данных?
2. Объясните смысл терминов (реляционная модель данных, мощность отношения, тип отношения).
3. Дайте определения нормальных форм.
4. Объясните применение ограничительных условий, поддерживающих целостность данных.

Содержание и оформление отчета

Отчет должен содержать: титульный лист, название и цель работы; вариант задания; скриншоты результатов работы; выводы по работе.