**6. Моделирование в ERwin**

Место ERwin в информационном моделировании  
Процесс построения информационной модели состоит из следующих шагов:

* определение сущностей;
* определение зависимостей между сущностями;
* задание первичных и альтернативных ключей;
* определение атрибутов сущностей;
* приведение модели к требуемому уровню нормальной формы;
* переход к физическому описанию модели: назначение соответствий имя сущности - имя таблицы, атрибут сущности - атрибут таблицы; задание триггеров, процедур и ограничений;
* генерация базы данных.

ERwin создает визуальное представление (модель данных) для решаемой задачи. Это представление может использоваться для детального анализа, уточнения и распространения как части документации, необходимой в цикле разработки. Однако ERwin далеко не только инструмент для рисования. ERwin автоматически создает базу данных (таблицы, индексы, хранимые процедуры, триггеры для обеспечения ссылочной целостности и другие объекты, необходимые для управления данными).

**Отображение логического и физического уровня модели данных в ERwin**

В ERwin существуют два уровня представления и моделирования - логический и физический . Логический уровень означает прямое отображение фактов из реальной жизни. Например, люди, столы, отделы, собаки и компьютеры являются реальными объектами. Они именуются на естественном языке, с любыми разделителями слов (пробелы, запятые и т.д.). На логическом уровне не рассматривается использование конкретной СУБД, не определяются типы данных (например, целое или вещественное число) и не определяются индексы для таблиц.  
Целевая СУБД, имена объектов и типы данных, индексы составляют второй (физический) уровень модели ERwin.  
ERwin предоставляет возможности создавать и управлять этими двумя различными уровнями представления одной диаграммы (модели), равно как и иметь много вариантов отображения на каждом уровне.

**Компоненты диаграммы ERwin и основные виды представлений диаграммы**

Диаграмма ERwin строится из трех основных блоков - сущностей, атрибутов и связей. Если рассматривать диаграмму как графическое представление правил предметной области, то сущности являются существительными, а связи - глаголами.  
Выбор между логическим и физическим уровнем отображения осуществляется через линейку инструментов или меню. Внутри каждого из этих уровней есть следующие режимы отображения:

* Режим "сущности" - внутри прямоугольников отображается имя сущности (для логической модели) или имя таблицы (для физического представления модели); служит для удобства обзора большой диаграммы или размещения прямоугольников сущностей на диаграмме.
* Режим "определение сущности" служит для презентации диаграммы другим людям.
* Режим "атрибуты". При переходе от предметной области к модели требуется вводить информацию о том, что составляет сущность. Эта информация вводится путем задания атрибутов (на физическом уровне - колонок таблиц). В этом режиме прямоугольник-сущность делится линией на две части - в верхней части отображаются атрибуты (колонки), составляющие первичный ключ, а в нижней - остальные атрибуты. Этот режим является основным при проектировании на логическом и физическом уровнях.
* Режим "первичные ключи" - внутри прямоугольников - сущностей показываются только атрибуты/колонки, составляющие первичный ключ.
* Режим "пиктограммы". Для презентационных целей каждой таблице может быть поставлена в соответствие пиктограмма (bitmap).
* Режим "показ глагольной фразы". На дугах связей показываются глагольные фразы, связывающие сущности (для логического уровня) или имена внешних ключей (для физического уровня).

Диаграмма может занимать более чем один экран и более чем один лист при печати. Для обзора модели предусмотрены, кроме прокруток экрана, режимы уменьшения/увеличения изображения, отображение всей модели, отображение выделенной части модели.

**Инструменты для создания модели в ERwin**

Основные инструменты создания модели доступны как из меню, так и через окно инструментов. С их помощью создаются независимые и зависимые сущности, идентифицирующие и неидентифицирующие связи, полные и неполные категории, неспецифические связи и текстовые элементы.  
Нажатием мыши над сущностью производится вход в один из многочисленных редакторов ERwin:

* редакторы, связанные с сущностью в целом (определение сущности, дополнительная информация, триггеры, индексы, характеристики таблицы, хранимые процедуры, связанные с таблицей);
* редакторы атрибутов (определение атрибутов, колонки таблицы в физическом представлении модели, репозитарий средства 4GL, например, расширенные атрибуты в PowerBuilder ).

**Идентификация сущностей. Сущности в ERwin**

На диаграмме сущность изображается прямоугольником. В зависимости от режима представления диаграммы прямоугольник может содержать имя сущности, ее описание, список ее атрибутов и другие сведения.  
Горизонтальная линия прямоугольника разделяет атрибуты сущности на два набора - атрибуты, составляющие первичный ключ в верхней части и прочие (не входящие в первичных ключ) в нижней части.  
Сущность представляет собой множество реальных или абстрактных объектов, например, люди, места, события, факты, которые имеют общие характеристики. Сущность - это логическое понятие. Сущности соответствует таблица в реальной СУБД. В ERwin сущность визуально представляет три основных вида информации:

* атрибуты, составляющие первичный ключ;
* неключевые атрибуты;
* тип сущности (независимая/зависимая).

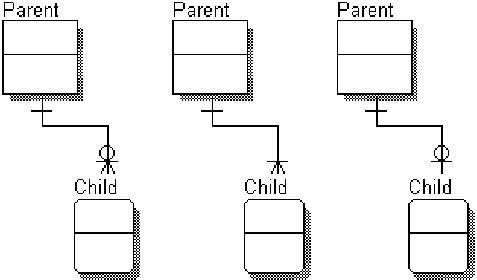
Первичный ключ - это атрибут или набор атрибутов, уникально идентифицирующий экземпляр сущности. Если несколько наборов атрибутов могут уникально идентифицировать сущность, то выбор одного из них осуществляется разработчиком на основании анализа предметной области.  
Для каждого первичного ключа ERwin создает при генерации структуры БД уникальный индекс.  
Экземпляры независимой сущности могут быть уникально идентифицированы без определения ее связей с другими сущностями; зависимая сущность, наоборот, не может быть уникально идентифицирована без определения ее связей с другими сущностями. Зависимая сущность отображается в ERwin прямоугольником с закругленными углами.

**Связи (relationships) в ERwin**

Связь - это функциональная зависимость между двумя сущностями (в частности, возможна связь сущности с самой собой). Например, важно знать фамилию сотрудника, и не менее важно знать, в каком отделе он работает. Таким образом, между сущностями "отдел" и "сотрудник" существует связь "состоит из" (отдел состоит из сотрудников). Связь - это понятие логического уровня, которому соответствует внешний ключ на физическом уровне. В ERwin связи представлены пятью основными элементами информации:

* тип связи (идентифицирующая, неидентифицирующая, полная/неполная категория, неспецифическая связь);
* родительская сущность;
* дочерняя (зависимая) сущность;
* мощность связи (cardinality);
* допустимость пустых (null) значений.

Связь называется идентифицирующей, если экземпляр дочерней сущности идентифицируется через ее связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в первичный ключ дочерней сущности. Дочерняя сущность при идентифицирующей связи всегда является зависимой.  
Связь называется неидентифицирующей, если экземпляр дочерней сущности идентифицируется иначе, чем через связь с родительской сущностью. Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в состав неключевых атрибутов дочерней сущности.  
Для определения связей ERwin выбирается тип связи, затем мышью указывается родительская и дочерняя сущность. Идентифицирующая связь изображается сплошной линией; неидентифицирующая - пунктирной линией. Линии заканчиваются точкой со стороны дочерней сущности.  
При определении связи происходит миграция атрибутов первичного ключа родительской сущности в соответствующую область атрибутов дочерней сущности. Поэтому такие атрибуты не вводятся вручную.  
Атрибуты первичного ключа родительской сущности по умолчанию мигрируют со своими именами. ERwin позволяет ввести для них роли, т.е. новые имена, под которыми мигрирующие атрибуты будут представлены в дочерней сущности. В случае неоднократной миграции атрибута такое переименование необходимо. Например, сущность "посредническая сделка" имеет атрибут "код предприятия-продавца" и "код предприятия-покупателя". В данном случае первичный ключ сущности "предприятие" ("код предприятия") имеет две роли в дочерней сущности.  
На физическом уровне имя роли - это имя колонки внешнего ключа в дочерней таблице.  
Мощность связи представляет собой отношение количества экземпляров родительской сущности к соответствующему количеству экземпляров дочерней сущности. Для любой связи, кроме неспецифической, эта связь записывается как 1:n.  
ERwin в соответствии с методологией IDEF1X предоставляет 4 варианта для n, которые изображаются дополнительным символом у дочерней сущности: ноль, один или больше (по умолчанию); ноль или один; ровно N, где N - конкретное число.  
Допустимость пустых (NULL) значений в неидентифицирующих связей ERwin изображает пустым ромбиком на дуге связи со стороны родительской сущности.  
Обозначения мощности соответственно ноль, один или больше, один или больше, ноль или один в нотации IE приведены на рис. 1.



***Рис.1. Обозначения мощности связи в нотации IE***

Имя связи на логическом уровне представляет собой "глагол", связывающий сущности. Физическое имя связи (которое может отличаться от логического) для ERwin означает имя ограничения (constraint) или индекса.

**Графическое редактирование модели**

Все объекты модели ERwin могут редактироваться средствами, принятыми в Windows - группировка, копирование, удаление, перемещение, использование системного буфера. Установка цветов и шрифтов осуществляется в удобных диалогах.  
Компоненты модели, представленные текстом (имена сущностей, атрибутов, текстовые элементы) могут редактироваться непосредственно на экране.

**Альтернативные ключи**

Альтернативный ключ - это атрибут (или группа атрибутов), несовпадающий с первичным ключом и уникально идентифицирующий экземпляр сущности. Например, для сущности служащий (идентификатор служащего, фамилия. имя, отчество) группа атрибутов "фамилия", "имя", "отчество" может являться альтернативным ключом (в предположении, что на предприятии не работают полные тезки).  
Для альтернативного ключа, как и для первичного, ERwin автоматически создает индексы при генерации БД.

**Инвертированные индексы**

Атрибуты, составляющие альтернативный ключ, однозначно (уникально) идентифицируют экземпляры сущности. В ERwin можно также составлять группы атрибутов, которые не идентифицируют уникально экземпляры сущности, но часто используются для доступа к данным. Для каждой такой группы атрибутов ERwin создает неуникальные индексы.  
Одни и те же атрибуты сущности могут входить в несколько различных групп ключей.

**Унификация атрибутов**

Зависимая сущность может наследовать один и тот же внешний ключ от более чем одной родительской сущности, или от одной и той же родительской сущности через несколько связей. Если не введены различные роли для такого множественного наследования, ERwin считает, что в зависимой сущности атрибуты внешнего ключа появляются только один раз.  
Унификация - это объединение двух или более групп атрибутов внешних ключей в один внешний ключ (группу атрибутов), в предположении, что значения одноименных атрибутов в дочерней сущности всегда одинаковы.  
Рассмотрим пример: сущность "сотрудник" имеет первичный ключ "код сотрудника" и связан идентифицирующей связью с сущностями "супруга" и "дети". При этом происходит миграция первичного ключа в зависимые сущности. В свою очередь, сущность "супруга" связана неидентифицирующей связью с сущностью "дети". Имеются два пути миграции ключа, однако в сущности "дети" атрибут "код сотрудника" появляется один раз в качестве элемента первичного ключа.  
Существуют случаи, когда унификация атрибутов дает неверный с точки зрения предметной области результат. Для отмены унификации для атрибутов вводятся имена ролей.

**Связи категоризации**

Некоторые сущности определяют целую категорию объектов одного типа. В ERwin в таком случае создается сущность для определения категории и для каждого элемента категории, а затем вводится для них связь категоризации. Родительская сущность категории называется супертипом, а дочерние - подтипом.  
Например, сущность "сотрудник" может содержать данные как о штатных работниках, так и о временно нанятых. Первые и вторые имеют различные, частично пересекающиеся наборы атрибутов (минимальное пересечение подтипов составляет первичный ключ). Общая часть этих атрибутов, включая первичный ключ, помещается в сущность-супертип "сотрудник".  
Различная часть (например, данные почасовой оплаты для временных работников и данные о зарплате и отпуске для штатных работников) помещается в сущности-подтипы.  
В сущности-супертипе вводится атрибут-дискриминатор, позволяющий различать конкретные экземпляры сущности - подтипа.  
В зависимости от того, все ли возможные сущности-подтипы включены в модель, категорийная связь является полной или неполной. Продолжая пример, если супертип может содержать данные об уволенных сотрудниках, то эта связь - неполной категоризации, так как для него не существует записи в сущностях - подтипах.  
В ERwin полная категория изображается окружностью с двумя подчеркиваниями, а неполная - окружностью с одним подчеркиванием.

**Реализация ссылочной целостности с помощью ERwin**

Ссылочная целостность - это обеспечение требования, чтобы значения внешнего ключа экземпляра дочерней сущности соответствовали значениям первичного ключа в родительской сущности. Ссылочная целостность может контролироваться при всех операциях, изменяющих данные (INSERT/UPDATE/DELETE). Средства контроля ссылочной целостности в ERwin включают автоматическую генерацию триггеров и использование механизмов декларативной ссылочной целостности (для тех СУБД, которые поддерживают данные механизмы).  
Для каждой связи на логическом уровне могут быть заданы требования по обработке операций INSERT/UPDATE/DELETE для родительской и дочерней сущности. ERwin представляет следующие варианты обработки этих событий:

* отсутствие проверки;
* проверка допустимости;
* запрет операции;
* каскадное выполнение операции (DELETE/UPDATE);
* установка пустого (NULL-значения) или заданного значения по умолчанию.

В соответствии с выбранным вариантом ERwin автоматически создает необходимые триггеры на диалекте SQL целевой СУБД. При этом ERwin пользуется библиотекой шаблонов триггеров, которые можно модифицировать.  
При генерации структуры базы данных триггеры, обеспечивающие ссылочную целостность могут быть переопределены на трех уровнях:

1. Могут быть переопределены триггеры, обеспечивающие правила для всей модели.
2. Могут быть переопределены триггеры, указанные для конкретной связи.
3. Могут быть переопределены триггеры, указанные для конкретной таблицы.

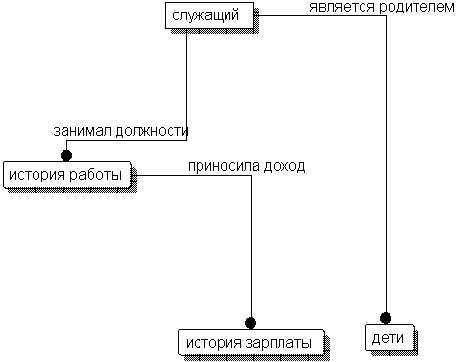
Тип переопределения указывается разработчиком при генерации схемы базы данных (рис. 6- соответственно RI Type Override, Relationship Override, Entity Override).

**Хранение информации в модели ERwin**

Обычно модели ERwin сохраняются на диск в виде файла. Имеется возможность хранить модель в целевой СУБД. Для этого с помощью самого ERwin в целевой СУБД создается метабаза ERwin. В этой базе данных сохраняется информация модели. В частном случае базой данных могут быть и dBase-файлы, с которыми ERwin работает через ODBC.

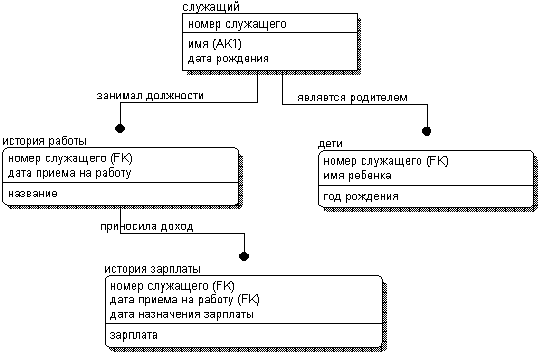
**Пример разработки модели в ERwin**

Рассмотрим цикл разработки на примере, приведенном в статье Кодда [1].  
Коротко напомним содержательную сторону задачи. Ведется учет служащих. Для каждого служащего хранится информация о детях и о списке занимавшихся этим служащим должностей. Для должностей хранится информация по установленным должностным окладам.  
Сначала создадим логический уровень модели. Для этого зададим режим отображения сущностей (Display/Entity Level). Создадим при помощи линейки инструментов сущности "служащий", "дети", "история работы", "история зарплаты". Будем именовать сущности на русском языке.  
Выбрав каждую сущность, зададим для нее подробное описание на русском языке в редакторе "Entity Definition". Это описание появится в отчетах ERwin и может быть отображено на диаграмме.  
Укажем связи между сущностями. Например, "служащий" связан идентифицирующей связью "является родителем" с сущностью "дети". Описание связи вводится в редакторе "Editor/Relationship".  
Результат работы отображен на диаграмме ERwin (рис. 2).



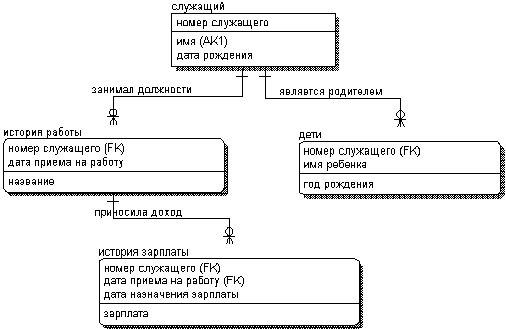
***Рис. 2. Диаграмма уровня сущности***

Теперь перейдем в режим задания атрибутов (Display/Atribute Level). В редакторе "Entity/Attribute" зададим на русском языке имена ключевых и неключевых атрибутов. Заметим, что для дочерней сущности "дети" ключевой атрибут "номер служащего" не указывается вручную. ERwin обеспечивает его миграцию из родительской сущности. То же происходит с другими дочерними сущностями.  
Для атрибута "имя" сущности "служащий" укажем, что он является альтернативным ключом (будем считать, что у всех служащих уникальные имена/фамилии). Для этого после имени атрибута поместим указатель AK1 в скобках.  
Результат работы отображен на диаграмме ERwin (рис. 3) в нотации IDEF1X.



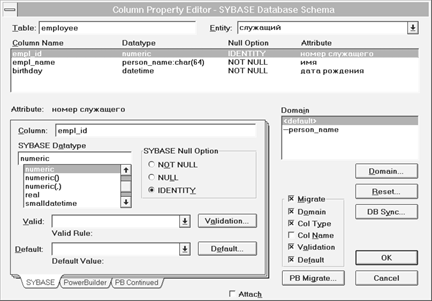
***Рис. 3. Диаграмма уровня атрибутов в нотации IDEF1X***

Вид той же диаграммы в нотации IE (Information Engineering) показан на рис.4.



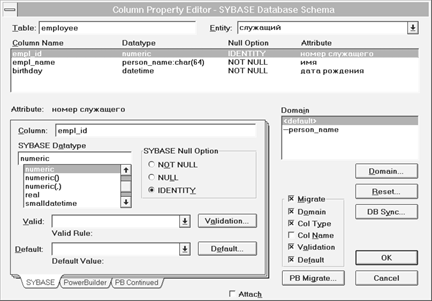
***Рис. 4. Диаграмма уровня атрибутов в нотации IE***

Так как имена атрибутов и сущностей задавались нами на русском языке, для перехода к физическому уровню модели следует поставить им в соответствие идентификаторы таблиц, колонок и ограничений, удовлетворяющие правилам целевой СУБД (обычно это означает использование латинских букв, цифр и некоторых специальных символов).  
В редакторе "Database Schema" указываем для каждой сущности соответствующее имя таблицы. Затем в редакторе "Attribute Definition" задаем имена колонок таблиц, соответствующие атрибутам сущностей. ERwin и здесь обеспечивает миграцию имен колонок в подчиненные таблицы.  
На этом этапе можно воспользоваться и редактором "Extended Attributes" для определения расширенных атрибутов PowerBuilder (формата отображения, маски редактирования, правила контроля, выравнивания, заголовков и комментариев).  
В редакторе "Relationship Definitions" указывается физическое имя связи, которое соответствует имени ограничения (constraint), создаваемого ERwin в базе данных.  
Теперь все готово к созданию БД и нужно выбрать целевую СУБД (если этого не было сделано раньше). Выберем, например, Sybase System 10.  
В редакторе SYBASE Database Schema задаем типы данных для колонок таблиц.  
Диалог, в котором происходит выбор типа данных, приведен на рис.5.



***Рис. 5. Определение физической модели***

Теперь можно перейти к созданию базы данных. Для этого выполняется команда "Sybase schema generation". ERwin построит пакет SQL-предложений генерации базы данных. На рис.6 показан диалог выбора параметров генерации пакета для генерации БД. На рисунке видно, что может быть задан фильтр (генерация не всех таблиц), пакет SQL-предложений можно просмотреть (preview), распечатать, сохранить в файл (report), выполнить генерацию (generate).



***Рис. 6. Выбор параметров генерации базы данных***

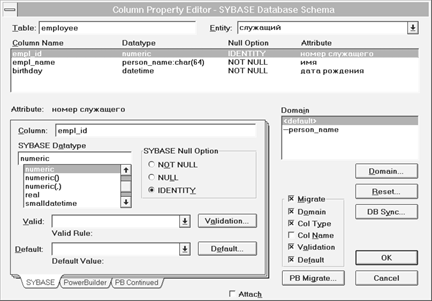
**7. Расширенные функции ERwin**

**Обратное проектирование (Reverse engineering)**

Обратное проектирование, то есть восстановление информационной модели по существующей базе данных, используется при выборе оптимальной платформы (rightsizing) для существующей настольной (desktop) базы данных или базы данных на mainframe, а также при расширении (или модификации) существующей структуры, которая была построена без необходимой сопроводительной документации. После завершения процесса восстановления модели ERwin автоматически "раскладывает" таблицы на диаграмме. Теперь можно выполнять модификации уже с использованием логической схемы - добавлять сущности, атрибуты, комментарии, связи и т.д. По завершении изменений одна команда - синхронизировать модель с базой данных - актуализирует все проведенные изменения.  
Построение модели может быть выполнено как на основании данных каталога базы данных, так и на основании пакета операторов SQL, с помощью которого была создана база данных.

**Синхронизация с базой данных**

В процессе разработки информационной системы может возникнуть ситуация, когда структура базы данных и информационная модель не соответствуют друг другу. ERwin предоставляет возможность привести их в соответствие.  
Для этого предусмотрена функция синхронизации с базой данных. После подключения к СУБД предлагается список несоответствий между существующей структурой данных и моделью. Например, если в базе данных создана новая таблица, то ERwin предложит провести включение ее в модель. Если в модель добавлена новая таблица, ERwin предложит создать ее в реальной базе данных. Аналогично, при добавлении колонок в базе данных или в модели ERwin предлагает провести соответствующие операции по синхронизации. Процедура выбора синхронизируемых таблиц показана на рис.7.

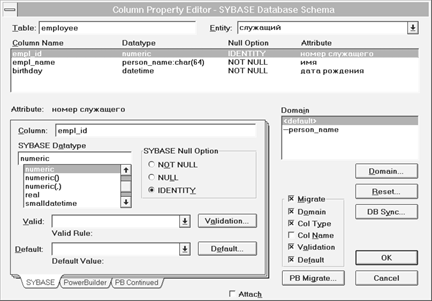


***Рис. 7. Выбор синхронизируемых таблиц***

ERwin "знает" о таких особенностях хранения данных в отдельных СУБД, как сегменты (в Sybase) и табличное пространство (в Oracle). Информация о физическом размещении может быть включена в модель и использована при прямом и обратном проектировании.

**Интерфейсы к СУБД**

ERwin поддерживает прямой интерфейс с основными СУБД: DB2 версии 2 и 3, Informix версий 5.1, 6.0, 7.1, Ingres, NetWare SQL, ORACLE версий 6 и 7, Progress, Rdb версий 4 и 6, SQL/400 версий 2 и 3, SQLBase версий 5 и 6, SQL Server версий 4 и 6, Sybase версии 4.2, Sybase System 10 и 11, Watcom SQL. Отметим, что поддерживаются как самые современные, так и предыдущие версии основных СУБД (рис.8).

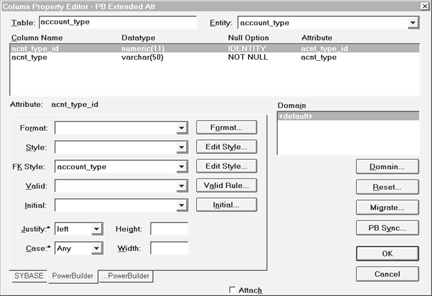


***Рис. 8. Выбор СУБД для создания модели***

ERwin поддерживает также настольные (desktop) СУБД: Microsoft Access, FoxPro, Clipper, dBASE III, dBASE IV и Paradox.  
Проектирование на физическом уровне выполняется в терминах той базы данных, которую предполагается использовать в системе. Важно, что ERwin "известны" соответствия между возможностями СУБД различных производителей, вследствие чего возможно преобразование физической схемы, спроектированной для одной СУБД, в другую.   
Для создания физической структуры БД может быть запрошена генерация DDL-скрипта (data definition language). При этом используется диалект SQL для выбранного типа и версии сервера. Хотя сгенерированный код не нуждается в модификации, имеется возможность его сохранить в файл или распечатать.

**Поддержка средств 4GL**

ERwin выпускается в нескольких различных редакциях, ориентированных на наиболее распространенные средства разработки 4GL. В числе поддерживаемых средств - PowerBuidler фирмы Powersoft, SQL Windows фирмы Gupta, Visual Basic фирмы Microsoft, Oracle\*CASE фирмы Oracle.  
Средства двунаправленного взаимодействия ERwin с базой данных обеспечивают управление информацией, ориентированной как на серверную, так и на клиентскую часть. Например, для PowerBuilder можно просматривать/редактировать расширенные атрибуты в редакторах ERwin.  
Ориентация ERwin на средства 4GL позволяет задать для будущих приложений большинство параметров, непосредственно связанных с базой данных, уже на стадии проектирования информационной модели.  
Покажем принципы организации такого взаимодействия на примере PowerBuilder.  
PowerBuilder создает в базе данных несколько внутренних таблиц для хранения своего репозитария (расширенных атрибутов для datawindow). Использование расширенных атрибутов гарантирует сохранение стиля отображения одних и тех же колонок базы данных для всех приложений, создаваемых рабочей группой. В расширенных атрибутах задаются такие параметры, как формат отображения, стиль редактирования, выражение проверки на корректность, начальное значение, выравнивание, ширина и высота элемента отображения, метка для формы редактирования, заголовок для табличного отображения.  
Для расширенных атрибутов допустимы те же операции синхронизации, что и для всей модели, то есть описания могут быть загружены в базу данных и, наоборот, созданные из среды PowerBuilder описания расширенных атрибутов могут быть загружены из базы данных в ERwin для модификации.  
Пример определения расширенных атрибутов показан на рис.9.



***Рис. 9. Задание расширенных атрибутов PowerBuilder***

Функция ERwin по генерации DataWindow позволяет сгенерировать прототипы окон данных будущего приложения уже на стадии создания информационной модели. Для создания Data Windows предлагается Wizard, с помощью которого указывается стиль окна и выбранные колонки таблиц.