

Лабораторная работа №7. Физическое проектирование баз данных. Представления БД

ЦЕЛЬ РАБОТЫ: Ознакомиться с подробностями физического проектирования баз данных в ERwin, разработкой индексов и объектов физической памяти, познакомиться с понятием представления базы данных и процессом его проектирования категории, и применить полученные знания для разработки модели базы данных и реализации ее для СУБД SQL Server.

Под физическим проектированием будем понимать процесс развития физической модели данных, полученной из логической модели, с целью получения модели, реализующей задачи, поставленные перед базой данных, и при этом наиболее эффективно. Для ERwin процесс физического проектирования выражается в выполнении следующих операций:

- индексирование таблиц и настройка индексов;
- задание и настройка объектов физической памяти;
- создание и настройка представлений;
- создание триггеров и скриптов обработки данных;
- подготовка документации по схеме базы данных.

В данной работе будут рассмотрены только два аспекта физического проектирования: индексирование таблиц и работа с представлениями базы данных.

Индексирование таблиц базы данных

ERwin располагает мощным редактором индексов, который позволяет изменять индекс, с тем чтобы улучшить работу системы по поиску данных и использовать те специальные возможности индексирования(например, кластеризация), которые доступны в конкретной СУБД.

Понятие индекса

В таблице реляционной базы данных данные обычно хранятся в том же порядке, в котором их ввели в таблицу. Хотя такой способ хранения и позволяет Вам быстро вводить новые данные, но поиск нужной строки в таблице представляется трудным, так как в общем случае необходимо просмотреть *все* записи таблицы.

Чтобы решить эту проблему поиска данных, СУБД использует особый тип файла, называемый *индексом*. Применение индексирования к конкретной колонке таблицы позволяет определить именно те строки в которых хранится конкретное значение колонки и тем самым значительно ускорить поиск информации.

Когда Вы генерируете схему физической базы данных, ERwin автоматически создает отдельный индекс на основе первичного ключа каждой таблицы, а также на основе всех альтернативных ключей, внешних ключей и Inversion Entry, поскольку эти колонки наиболее часто используются для поиска данных. Используя любой из автоматически сгенерированных индексов в качестве точки отсчета, ERwin позволяет легко создавать новые индексы по мере изменения потребностей. При создании нового индекса, можно присвоить ему разные колонки и изменить порядок, в котором отсортированы значения в индексе, так чтобы были удовлетворены требования поиска для каждого конкретного приложения, работающего с базой данных.

Создание и модифицирование индексов в ERwin

Когда Вы генерируете физическую модель на основе модели данных, ERwin автоматически создает индекс для первичного ключа (PK) и отдельный индекс для каждого альтернативного ключа (AK), внешнего ключа (FK) и инверсного входа (IE). Если у сущности не было назначено альтернативных ключей и Inversion Entry, то ERwin создает индексы только для первичного ключа и внешних ключей.

После того как индекс создан, Вы можете изменить его характеристики в редакторе индексов (см. рис. 1), например изменить его имя, изменить его определение так, чтобы он принимал уникальные или дублирующиеся значения, или изменить порядок сортировки данных.

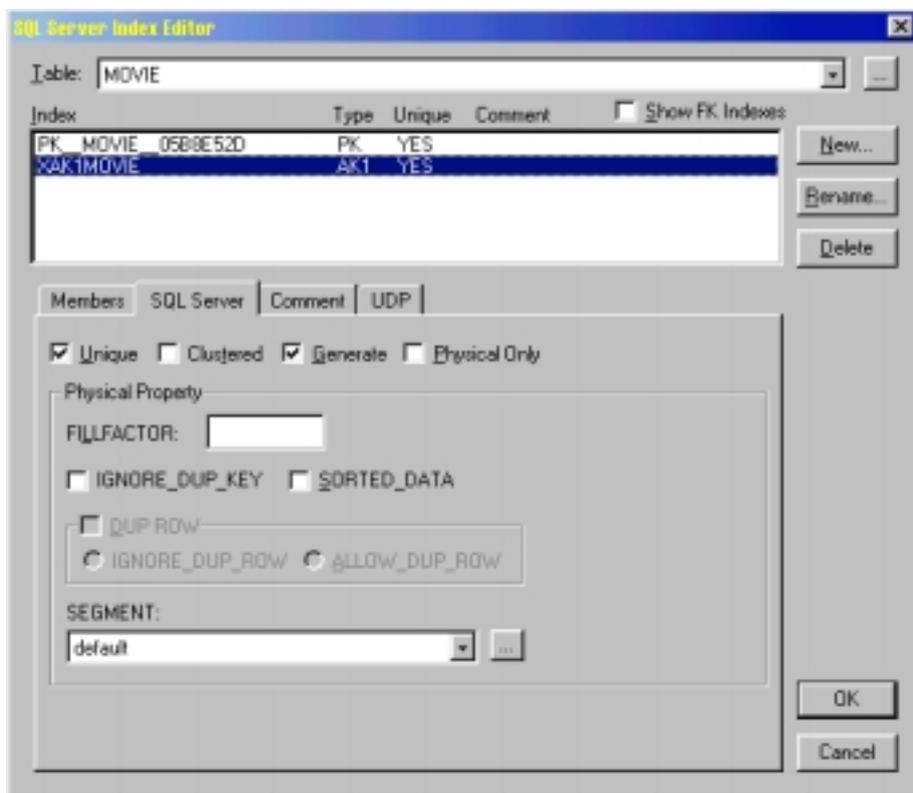


Рис. 1. Окно редактора индексов для таблицы СУБД SQL Server.

Большинство организаций используют специальные соглашения при присвоении имен объектам модели данных. Вы можете переименовать индекс, созданный ERwin, так чтобы его имя соответствовало соглашениям, в редакторе индексов. ERwin формирует имя индекса следующим образом: «X»+»PK» (или «AK», «IF», «IE» + «n») + Имя таблицы сущности, где n - целое число ≥ 1 , используемое для различения нескольких индексов одного типа, обычно внутри одной таблицы.

Вы можете увеличить скорость выполнения многих запросов, изменив порядок расположения колонок. Вы можете использовать технику drag&drop для изменения порядка расположения колонок в редакторе индексов.

Редактор индексов выводит на экран информацию об индексах каждой сущности в модели данных. ERwin вставляет имя выбранной СУБД в качестве префикса перед заголовком редактора и настраивает режимы редактора в соответствии с СУБД. Хотя режимы редактора адаптируются для каждой конкретной СУБД, остается один и тот же основной метод для создания и изменения информации об индексах, независимо от конкретной СУБД.

Чтобы создать новый индекс по альтернативному ключу или инверсному входу, выберите имеющийся индекс, который послужит источником, и нажмите кнопку *New*. Если установить флажок *Unique*, то ERwin создает индекс по альтернативному ключу. В противном случае ERwin создает индекс по Inversion Entry. Когда ERwin создает индекс, он присваивает очередной порядковый номер имени индекса, а также присваивает значения по умолчанию всем характеристикам нового индекса, таким как порядок сортировки, кластеризация и т.д. По умолчанию в индекс не включаются колонки таблицы.

Изменение характеристик индекса

Интерфейс редактора индексов достаточно прост (см. рис. 1). Вверху окна находится список сущностей модели и кнопка перехода в редактор таблицы. Ниже приводится перечень имеющихся индексов с указанием их характеристик. Внизу окна находится блокнот с параметрами текущего индекса. Кнопки *Rename*, *Delete*, *New* управляют списком индексов.

Вкладка *Members* управляет списком колонок индекса (*Index members*), их порядком (стрелки «вверх» и «вниз») и направлением сортировки колонок (флажок *DESC*). Последний доступен только если СУБД поддерживает такую возможность и установлен если сортировка идет по убыванию (по умолчанию считается по возрастанию). Вкладка *Comment* позволяет задать комментарий для текущего индекса.

Также редактор индексов имеет одну или более вкладок, хранящих параметры индекса, присущие конкретной СУБД (на рис. 1 это вкладка *SQL Server*).

Когда ERwin создает индекс, он автоматически присваивает ему имя и значения по умолчанию для режимов, которые определяют, будет ли индекс:

- содержать повторяющиеся значения.
- отсортирован по возрастанию или по убыванию.
- кластеризованным или некластеризованным.

Когда ERwin создает индекс, он автоматически по умолчанию запрещает повторяющиеся значения в индексах по первичным и альтернативным ключам и разрешает повторяющиеся значения в индексах по Inversion Entry. По умолчанию ERwin автоматически сохраняет значения в индексе в порядке возрастания (значения сортируются по алфавиту от A до Z, а числа от 0 до 9). Если Вы хотите изменить порядок сортировки для колонки и СУБД поддерживает режим сортировки по убыванию (Z-A, 9-0), выберите колонку и затем поставьте флажок *Descending*.

Кластеризованный индекс - это специальная техника индексирования, при которой данные в таблице физически располагаются в индексированном порядке. Использование кластеризованного индекса значительно ускоряет выполнение запросов, которые производят выборку строк с одинаковыми значениями в индексированной колонке (колонках). Поскольку данные физически расположены в индексированном порядке, для каждой таблицы может существовать только один кластеризованный индекс. Если Ваша СУБД поддерживает использование кластеризованного индекса, то ERwin автоматически создает индекс первичного ключа кластеризованным. Если Вы хотите произвести кластеризацию индекса не по первичному ключу, а каким-то другим способом, то ERwin автоматически снимает кластеризацию с индекса по первичному ключу.

Редактор индексов позволяет также удалять индекс, если он уже не нужен, хотя индекс по первичному ключу Вы не можете удалить. Если Вы удаляете индекс по альтернативному ключу или по инверсному входу, то атрибуты, составляющие ключ, не удаляются - убирается только знак альтернативного ключа или инверсного входа.

Если Вы генерируете физическую схему, Вы можете включить как часть схемы любые индексы, определенные Вами в ERwin. ERwin автоматически преобразует определения индексов в выражения схемы и вставляет информацию о заданных параметрах с соблюдением синтаксиса.

Групповое окно «Index Option» в редакторе Schema Generation Report позволяет Вам выбирать индексы, которые ERwin будет включать в схему при генерации. Редактор Schema Generation Report содержит все режимы индексирования, поддерживаемые производителем СУБД.

Представления базы данных

До сих пор рассматривались только таблицы, которые реально хранятся в базе данных. Это, так называемые, базовые таблицы (base tables). Однако, существует другой вид таблиц, получивший название "представления" (иногда их называют "представляемые таблицы"). *Представление (view)* - это таблица, содержимое которой берется из других таблиц посредством запроса. При этом новые копии данных не создаются. Когда содержимое базовых таблиц меняется, СУБД автоматически перевыполняет запросы, создающие view, что приводит к соответствующим изменениям в представлениях.

Чаще всего представления включаются в состав физической модели для выражения определенных бизнес-правил модели. Так на рис. 1. представлена модель данных, описывающая торговлю иностранными автомобилями. На рис. 2. приведена физическая версия той же модели, в которую включено представление «поиск_автомобиля», которое объясняет механизм поиска конкретного автомобиля в базе данных по известным параметрам.



Рис. 1. Пример логической модели данных предметной области.

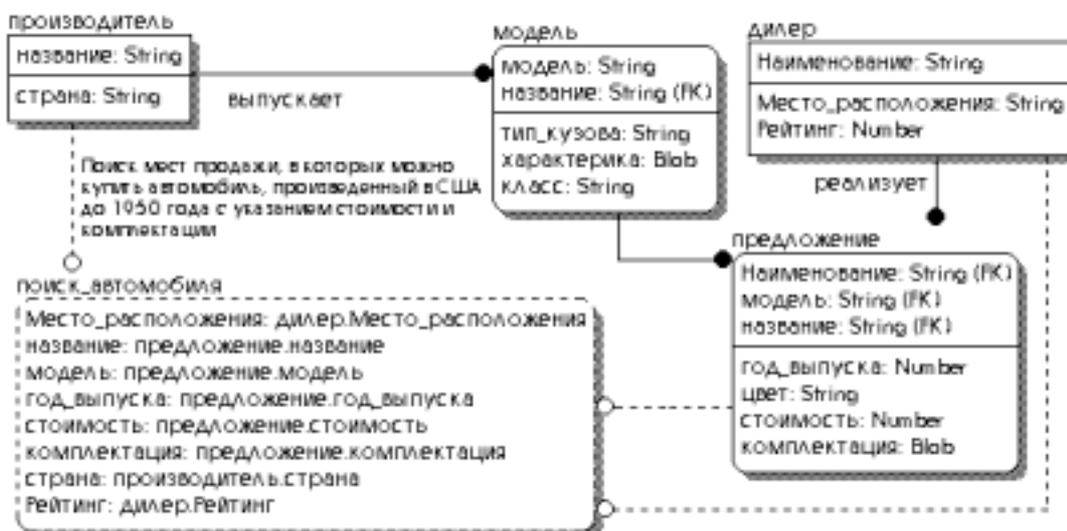


Рис. 2. Пример физической модели базы данных с представлением.

Представления – объекты физической модели, они не представимы на логическом уровне. Для вставки в модель представлений используется специальный инструмент из Toolbox. Так как по своей сути представление является запросом, то для его использования необходимо указать таблицы так или иначе участвующие в нем. Это делается также с помощью специального инструмента – связи представлений (*view relationship*). Другие типы связей не допустимы и автоматически преобразуются к связи представлений.

Для настройки параметров представлений используется несколько редакторов – редактор представлений, редактор колонок представлений и редактор связей представлений.

Редактор представлений (рис. 3) используется для настройки характеристик представлений физической модели данных. Его интерфейс схож с интерфейсом редактора таблиц и отличается наличием вкладок *Select*, *From*, *Where*, *SQL*. Эти вкладки управляют структурой предложения запроса SQL, который служит основой представления. Флажок *Generate* определяет какое представление будет сгенерировано при создании базы данных, а какое – нет. Вкладка *Select* определяет состав представления (*View Columns*), порядок его колонок, параметры каждой колонки (*Name*, *Expr*). Флажок *Distinct* определяет повторяемость результатов запроса, а кнопка *New Expression* позволяет задать колонку, основанную на формуле.

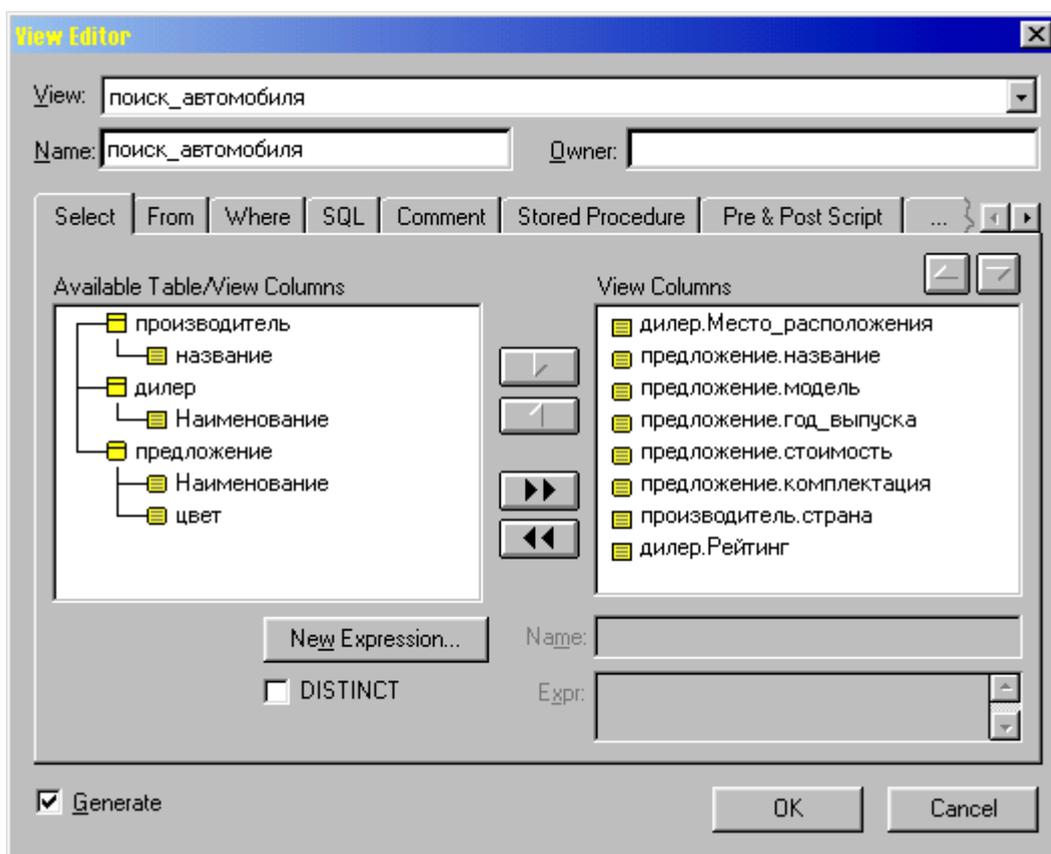


Рис. 3. Окно редактора представлений базы данных.

Вкладка *From* содержит перечень таблиц представления и позволяет менять их состав и порядок. Кроме того, для каждой таблицы может быть задан псевдоним (*alias*), в результате чего одна и та же таблица может несколько раз участвовать в запросе. Вкладка *Where* определяет условия отбора записей в представлении. Сюда входят условие отбора (*where*), условия группировки данных (*group by*) и отбора групп (*having*) и прочие параметры. Значения каждого из параметров полностью определяются языком SQL. Вкладка содержит полный текст запроса представления на языке SQL. Включив флажок *User-defined SQL* можно записать произвольное предложение SQL в соответствии с потребностями не прибегая к услугам этого редактора.

Редактор колонок представления (рис. 4) используется для настройки параметров каждой колонки представления и является модифицированным редактором колонок таблицы. Вкладка *General* свойств колонки позволяет переопределить тип данных колонки запроса по сравнению с исходной колонкой таблицы, а вкладка *Select* – параметры присущие конкретной СУБД. В остальном этот редактор совпадает с редактором колонок таблицы.

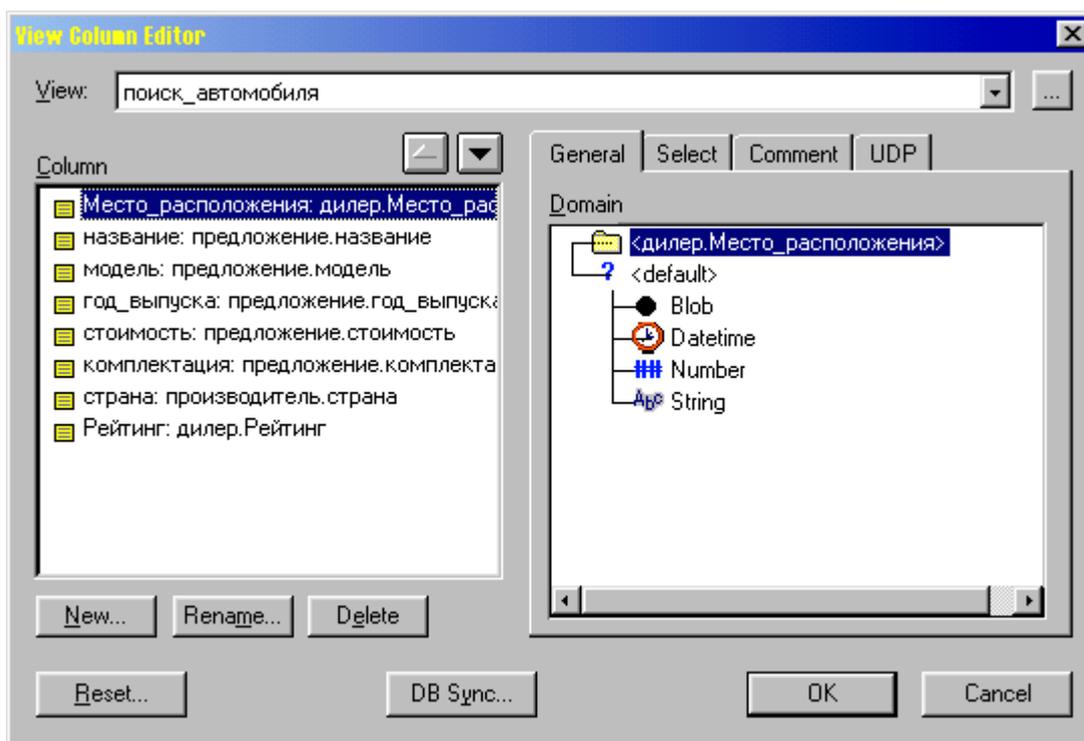


Рис. 4. Окно редактора колонок представления.

Редактор связи представления (рис. 5) используется для настройки отдельной связи представления и таблицы. С его помощью можно изменить псевдоним таблицы этой связи (*alias*) и указать порядок таблицы в представлении (*Sequence*). В остальном этот редактор соответствует редактору связей физического уровня.

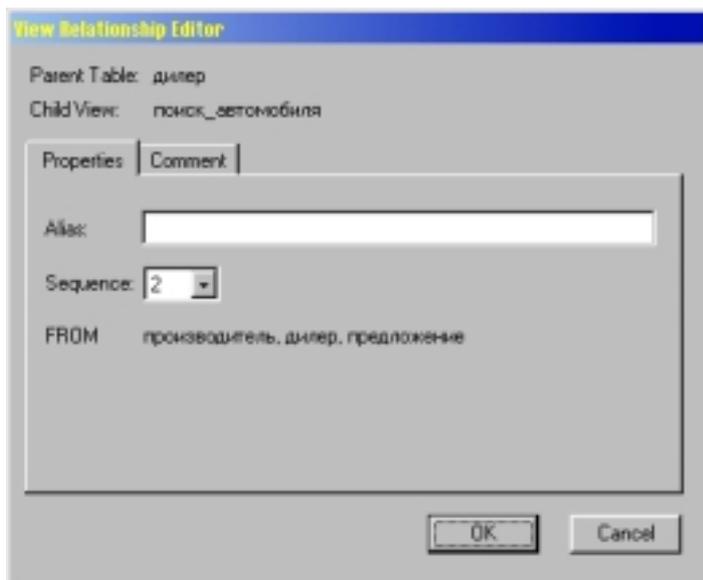


Рис. 5. Окно редактора связи представления.

Задание на лабораторную работу

1. Внимательно изучить материалы введения и изучить соответствующие редакторы *ERwin*.
2. Разработать модель данных, позволяющую решить следующую задачу. Пусть имеется видеотека, занимающаяся прокатом видеокассет. В видеотеке предлагаются к прокату произвольное количество наименований видеокассет, каждое наименование представлено в нескольких экземпляров. Для целей удобства поиска интересующей кассеты, каждое наименование кассеты имеет собственную карточку, в которой указываются: полное название фильма, его жанр, длительность, год, выпускающая киностудия, параметры (цветное/черно-белое, рейтинг), режиссер и состав актеров с указанием их ролей. При разработке системы автоматизированного управления видеотекой требуется разработать базу для хранения всех этих, а также и других параметров: для каждого актера необходимо хранить страну происхождения, год рождения, фотография, биография; для киностудии необходимо указание страны расположения. Для каждого экземпляра фильма должно быть известно место ее расположения на складе, или, если она выдана, то когда и кому.
3. При построении логической модели необходимо задокументировать все ее элементы и обосновать выбор доменов для каждого атрибута.
4. Выбрать в качестве СУБД назначения SQL Server (версия 7.0), привести в соответствие домены и типы данных.
5. На физическом уровне необходимо спроектировать по одному дополнительному индексу (по выбору) для каждой таблицы и настроить их параметры.

6. Добавить в модель представление «арендованные кассеты с французскими комедиями». Связать ее с необходимыми таблицами с тем расчетом, чтобы можно было получить код экземпляра кассеты, название фильма, имя арендовавшего его человека.
7. Сгенерировать БД. При подключении к серверу SQL необходимо указать сервер ЛВС как сервер БД.
8. Сохранить полученную модель в свою папку под именем **task7-1.er1**.

Контрольные вопросы

1. Какие операции входят в понятие «физическое проектирование»?
2. Что такое индекс и какие разновидности индексов существуют?
3. В чем различие между ключом и индексом ? В чем различие между представлением и таблицей?
4. Какими конструкциями языка SQL создаются таблицы и представления?
5. В чем отличие отношения представления от других типов отношений? Имеет ли отношение представления направление?