

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Пермская государственная сельскохозяйственная академия  
имени академика Д.Н. Прянишникова»

**ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**  
направление 230700 «Прикладная информатика»

***ЛАБОРАТОРНОЕ ЗАНЯТИЕ № 6***

**Тема: ФИЗИЧЕСКАЯ МОДЕЛЬ ДАННЫХ В МЕТОДОЛОГИИ IDEF1X**

**Учебные вопросы:**

1. Уровни физической модели данных.
2. Индексы.
3. Правила валидации и значения по умолчанию.
4. Денормализация отношений.
5. Построение физической модели данных.

**Литература, техническое и программное обеспечение:**

1. Методическая разработка по теме занятия.
2. Класс ПЭВМ.
3. AllFusion Erwin Data Modeler 7.

## Вопрос 1. Уровни физической модели

Различают два уровня физической модели:

- трансформационная модель (Transformation Model);
- модель СУБД (DBMS Model).

**Трансформационная модель** содержит информацию для реализации отдельного проекта, который может быть частью общей ИС и описывать подмножество предметной области. ERwin поддерживает ведение отдельных проектов, позволяя проектировщику выделять подмножество модели в виде предметных областей (Subject Area). Трансформационная модель позволяет проектировщикам и администраторам БД лучше представлять, какие объекты БД хранятся в словаре данных, и проверить, насколько физическая модель данных удовлетворяет требованиям к ИС. Трансформационная модель должна помочь разработчикам выбрать структуру хранения данных и реализовать систему доступа к ним.

**Модель СУБД** автоматически генерируется из трансформационной модели и является точным отображением системного каталога СУБД. ERwin непосредственно поддерживает эту модель путем генерации системного каталога. При составлении схемы БД в качестве индексов могут использоваться как ключевой атрибут, так и остальные поля БД.

По умолчанию ERwin генерирует имена таблиц и индексов по шаблону на основе имен соответствующих сущностей и ключей логической модели, которые в дальнейшем могут быть откорректированы вручную. Имена таблиц и колонок будут сгенерированы по умолчанию на основе имен сущностей и атрибутов логической модели.

## Вопрос 2. Индексы

В БД данные обычно хранятся в том порядке, в котором их ввели в таблицу. Многие реляционные СУБД имеют страничную организацию, при которой таблица может храниться фрагментарно в разных областях диска, причем строки таблицы располагаются на страницах неупорядоченно. Такой способ позволяет быстро вводить новые данные, но затрудняет поиск данных. Чтобы решить проблему поиска, СУБД используют объекты, называемые *индексами*.

**Индекс** содержит отсортированную по колонке или нескольким колонкам информацию и указывает на строки, в которых хранится конкретное значение колонки.

Поскольку значения в индексе хранятся в определенном порядке, при поиске просматривать нужно значительно меньший объем данных, что существенно уменьшает время выполнения запроса. Индекс рекомендуется создавать для тех колонок, по которым часто производится поиск.

При генерации схемы физической БД ERwin автоматически создает индекс на основе первичного ключа каждой таблицы, а также на основе всех альтернативных ключей и внешних ключей, поскольку эти колонки наиболее часто используются для поиска данных. Можно отказаться от генерации индексов по умолчанию и создать собственные индексы. Для увеличения эффективности поиска администратор БД должен анализировать часто выполняемые запросы и на основе анализа создавать собственные индексы.

### Вопрос 3. Правила валидации и значения по умолчанию

ERwin поддерживает правила валидации для колонок, а также значение, присваиваемое колонкам по умолчанию.

**Правило валидации** задает список допустимых значений для конкретной колонки и/или правила проверки допустимых значений. В список допустимых значений можно вносить новые значения. ERwin позволяет сгенерировать правила валидации соответственно синтаксису выбранной СУБД с учетом границ диапазона или списка значений.

**Значение по умолчанию** – значение, которое нужно ввести в колонку, если никакое другое значение не задано явным образом во время ввода данных. С каждой колонкой или доменом можно связать значение по умолчанию. Список значений можно редактировать.

После создания правила валидации и значения по умолчанию их можно присвоить одной или нескольким колонкам или доменами.

Создание правил валидации и значений по умолчанию будет рассмотрено ниже при построении физической модели данных.

### Вопрос 4. Денормализация отношений

После нормализации все взаимосвязи данных становятся определены, исключая ошибки при оперировании данными. Но нормализация данных снижает быстродействие БД. Для более эффективной работы с данными, используя возможности конкретного сервера БД, приходится производить процесс, обратный нормализации, – денормализацию.

Для процесса денормализации не существует стандартного алгоритма, поэтому в каждом конкретном случае приходится искать свое решение (рис. 4.1). Денормализация обычно проводится на физическом уровне модели.

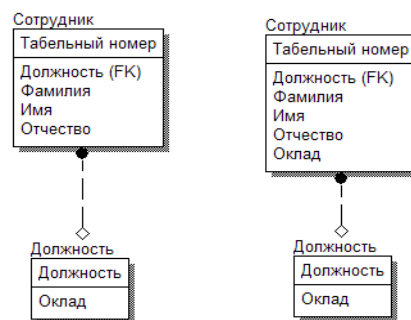


Рисунок 4.1 – Пример денормализации

Слева данные находятся в 3НФ. Но для получения из БД информации о сотруднике, включая его оклад, приходится обращаться к таблицам Должность и Сотрудник. Если в таблицу Сотрудник добавить колонку Оклад (справа на рисунке), то тогда при выборке информации о сотруднике достаточно обратиться к таблице Сотрудник (исключается объединение). При этом нарушается 3НФ и возникают аномалии, в том числе аномалии при обновлении (например, информации об окладе). Для решения этой проблемы можно делать выборку только из таблицы Сотрудник, а обновлять значение оклада только в таблице Должность.

ERwin имеет следующие возможности по поддержке процесса денормализации:

- Сущности, атрибуты, группы ключей и домены можно создавать только на логическом уровне модели. В ERwin существует возможность выделения элементов логической модели таким образом, чтобы они не появлялись на физическом уровне.

- Таблицы, столбцы, индексы и домены можно создавать только на физическом уровне. В ERwin существует возможность выделения элементов модели таким образом, чтобы они не появлялись на логическом уровне. Эта возможность напрямую поддерживает денормализацию физической модели, так как позволяет проектировщику включать таблицы, столбцы и индексы в физическую модель, ориентированную на конкретную СУБД.
- Разрешение связей «многие-ко-многим». При разрешении этих связей в логической модели ERwin добавляет ассоциированные сущности и позволяет добавить в них атрибуты. При разрешении связей в логической модели автоматически разрешаются связи и в физической модели.

## Вопрос 5. Построение физической модели данных

Создать физическую модель можно несколькими способами.

**Во-первых**, в самом начале при создании нашей логической модели данных в диалоговом окне Create Model – Select Template мы указали тип модели Logical / Physical (рис. 5.1). Для перехода на физический уровень достаточно на панели инструментов логической модели выбрать значение индикатора Model Type Indicator – Physical. При этом логическая модель трансформируется в физическую для СУБД Access 2000.

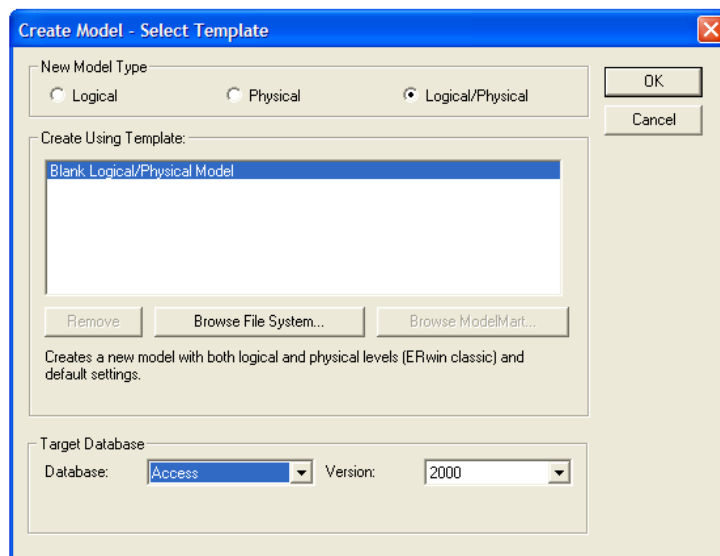



Рисунок 5.1 – Диалоговое окно Create Model – Select Template

**Во-вторых**, воспользовавшись кнопкой Create Model  на панели инструментов, можно вызвать знакомое нам диалоговое окно Create Model – Select Template и с помощью него создать физическую модель.

**В-третьих**, на панели меню можно выбрать меню Tools – Derive New Model. В появившемся диалоговом окне Derive New Model (рис. 5.2) необходимо указать тип модели – Physical и выбрать СУБД (например, Access 2000), после чего нажать кнопку Далее.

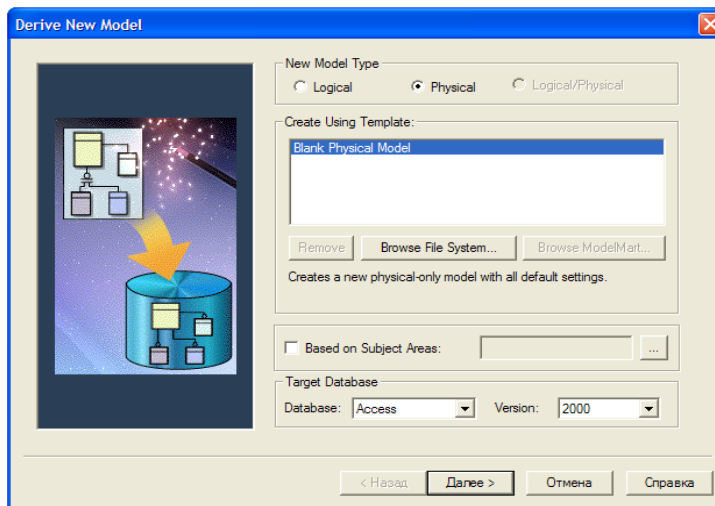


Рисунок 5.2 – Диалоговое окно Derive New Model

В появившемся окне Device Model – Select Source Model Object (рис. 5.3) указать необходимые объекты и нажать кнопку Далее. В следующем окне Device Model – Select Source Model Object после настройки параметров нажать кнопку Готово.

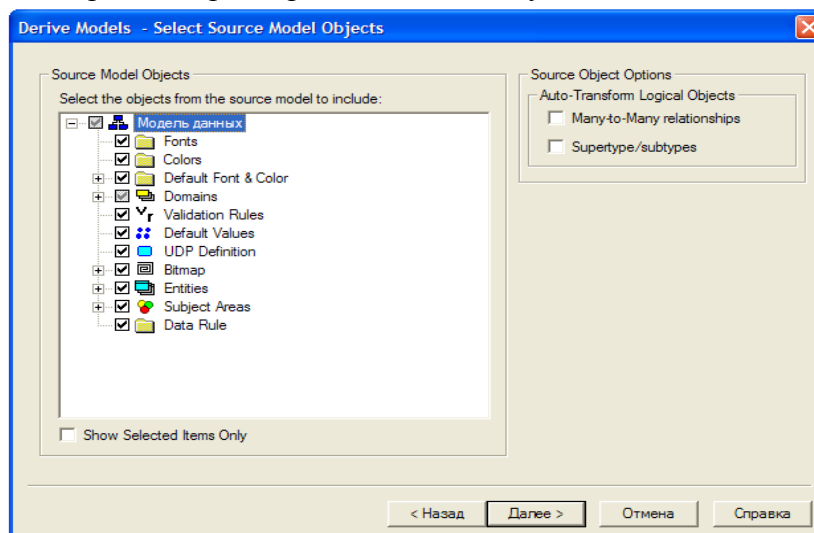


Рисунок 5.3 – Диалоговое окно Device Model – Select Source Model Object

Трансформация логической модели в физическую выполняется по следующему принципу: сущности становятся таблицами, атрибуты становятся столбцами, а ключи становятся индексами (табл. 5.1).

Таблица 5.1 – Сопоставление компонентов логической и физической модели

Логическая модель	Физическая модель
Сущность	Таблица
Атрибут	Столбец
Логический тип (текст, число, дата, blob)	Физический тип (корректный тип, зависящий от выбранной СУБД)
Первичный ключ	Первичный ключ, индекс РК
Внешний ключ	Внешний ключ, индекс FK
Альтернативный ключ	АК-индекс – уникальный, непервичный индекс
Правило бизнес-логики	Триггер или сохраненная процедура
Взаимосвязи	Взаимосвязи, определяемые использованием FK-атрибутов

Окно диаграммы для физической модели выглядит следующим образом (рис. 5.4):

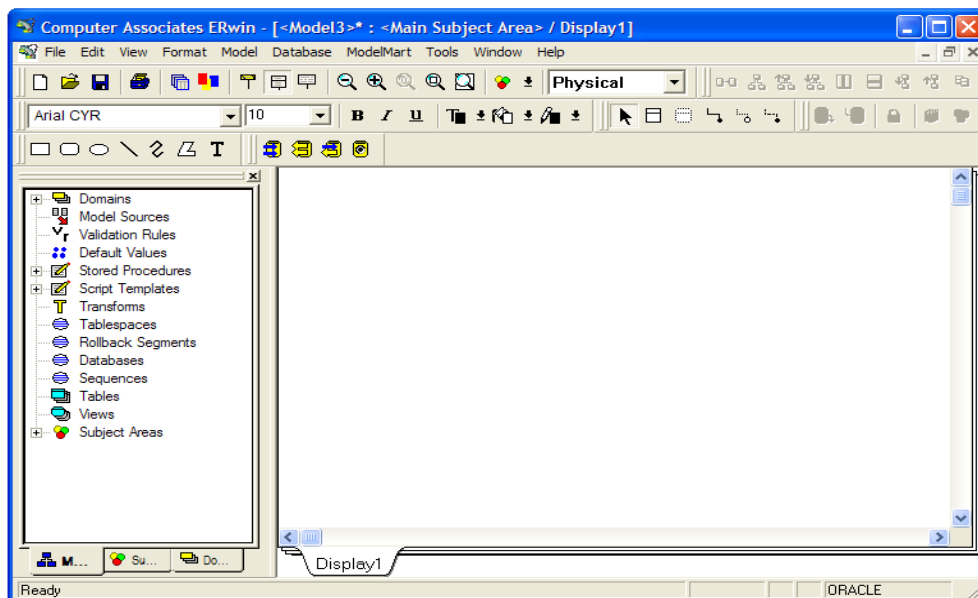








Рисунок 5.4 – Окно диаграммы физического уровня представления данных

На физическом уровне палитра инструментов имеет следующие кнопки:

-  – указатель (режим мыши) – в этом режиме можно установить фокус на каком-либо объекте модели;
-  – таблица;
-  – представление;
-  – идентифицирующая связь;
-  – связь между представлением и временной таблицей;
-  – неидентифицирующая связь.

**Представления (временные или производные таблицы)** – это объекты БД, данные в которых не хранятся постоянно, как в таблице, а формируются динамически при обращении к представлению.

Представление не может существовать само по себе, а определяться в терминах одной или нескольких таблиц. Его применение позволяет разработчику БД обеспечить каждому пользователю свой взгляд на данные, что решает проблемы простоты использования и безопасности данных.

После трансформации логической модели данных в физическую последняя приняла следующий вид (рис. 5.5):

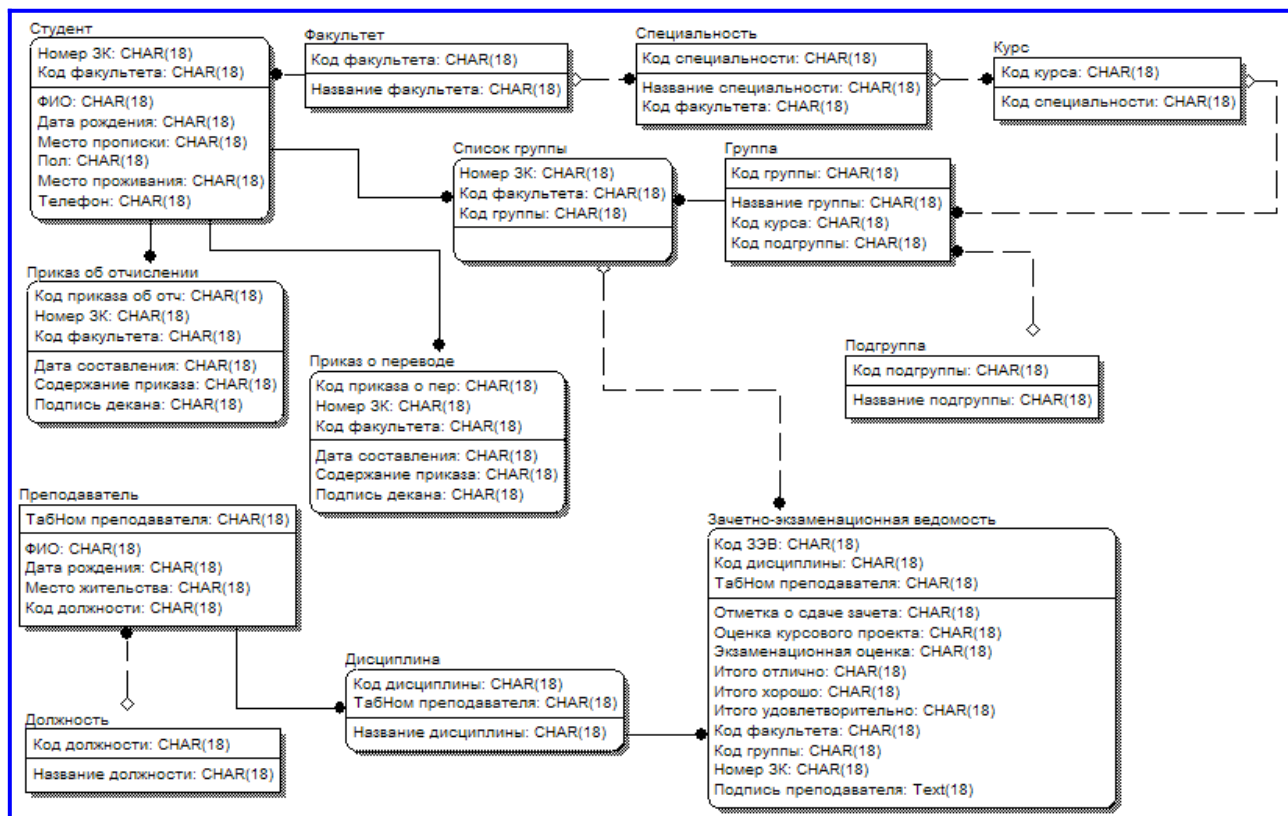


Рисунок 5.5 – Физическая модель после автоматического преобразования логической модели

В полученной модели скорректируем названия некоторых колонок, а также типы и размеры полей. Кроме того, на этапе создания физической модели данных введем правила валидации колонок, определяющие списки допустимых значений и значения по умолчанию.

Для корректировки физической модели данных необходимо выполнить следующие действия.

1. Поскольку на физическом уровне объекты БД могут называться так, как того требуют ограничения СУБД, т.е. короткими словами или «техническими» наименованиями, понятными порой только программистам, переименуем некоторые колонки таблиц. Кроме того, скорректируем типы и размеры полей.

Для того чтобы изменить таблицу, нужно щелкнуть по ней мышью два раза. Появится диалоговое окно Columns (рис. 5.6).

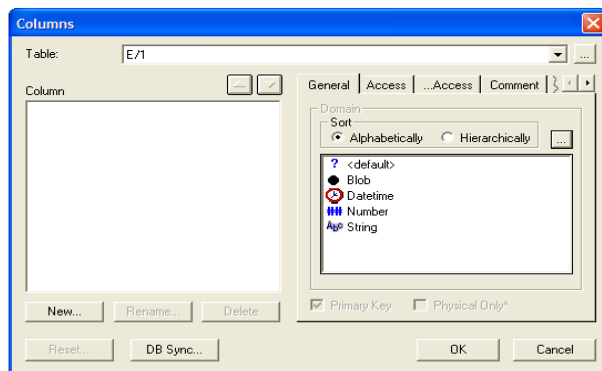


Рисунок 5.6 – Диалоговое окно Columns

В нем для работы с колонками используются следующие вкладки:

- **General** – позволяет поставить в соответствие колонке определенный домен, создать колонку только на физическом уровне и включить ее в состав первичного ключа.
- **Закладка, соответствующая выбранной СУБД** – имя закладки устанавливается автоматически соответствующей выбранной СУБД. Позволяет задать тип данных, опцию

NULL, правила валидации и значение по умолчанию. Для СУБД Access, AS/400, PROGRESS и Teradata создаются дополнительные закладки для задания свойств.

- **Comment** – служит для внесения комментария к каждой колонке.
- **Data Source** – доступна только при моделировании хранилищ данных.
- **UDP** – задает свойства, определяемые пользователем.
- **Index** – служит для включения колонки в состав индексов.
- **History** – отображает историю создания и изменения свойств атрибутов.

2. С помощью кнопки Rename и вкладки Access этого диалога скорректируем имена первичных ключей, типы и размеры полей соответственно согласно рис. 5.7.

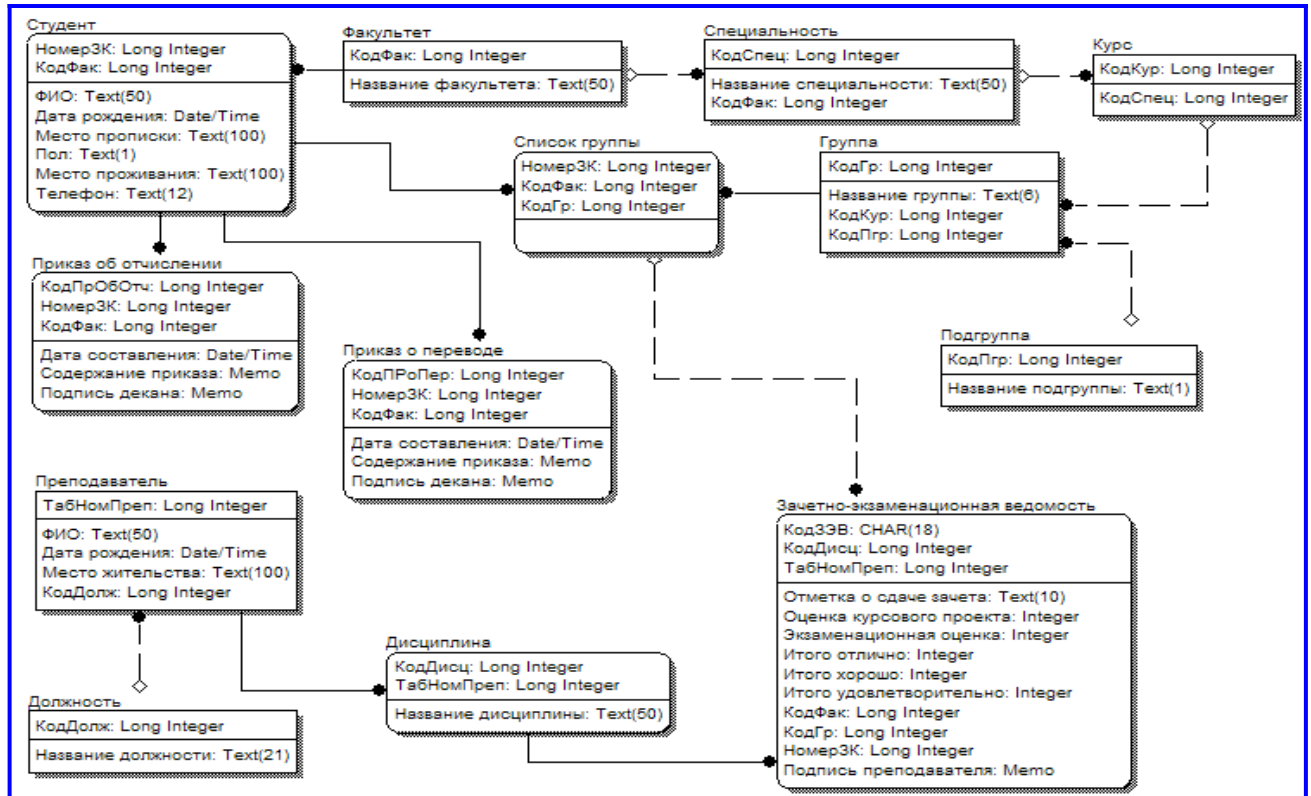


Рисунок 5.7 – Скорректированная физическая модель

3. Введем в нашу модель валидации колонок, определяющие списки допустимых значений и значения по умолчанию согласно таблице 5.2. Это можно сделать, например, для сущности «Зачетно-экзаменационная ведомость», вызвав диалоговое окно Columns и открыв вкладку Access (рис. 5.8).

Таблица 5.2 – Правила валидации

Колонка	Правило валидации
Пол	М или Ж
Название подгруппы	А или Б
Оценка курсового проекта	$\geq 2$ и $\leq 5$
Экзаменационная оценка	$\geq 2$ и $\leq 5$



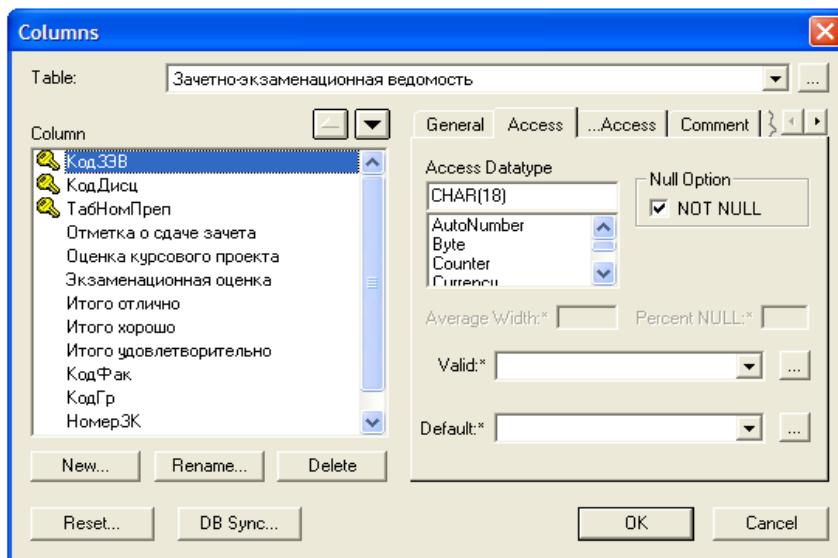



Рисунок 5.8 – Вкладка Access диалогового окна Columns

4. Нажмите кнопку  после поля Valid:\*. Появится диалоговое окно Validation Rules (рис. 5.9).

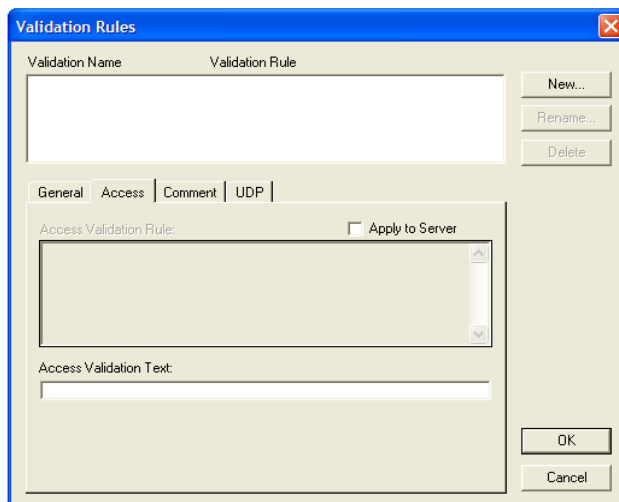


Рисунок 5.9 – Диалоговое окно Validation Rules

5. С помощью кнопки New создайте новое правило валидации в диалог New Validation Rule (рис. 5.10).

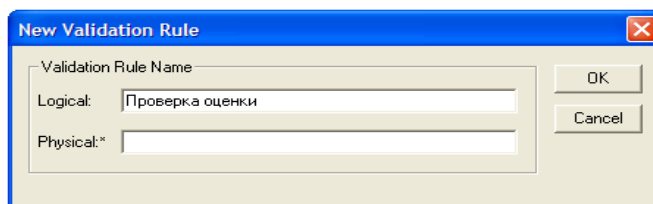


Рисунок 5.10 – Диалоговое окно New Validation Rule

6. Само правило необходимо записать в окне Validation Rules в поле Access Validation Rule согласно рис. 5.11.

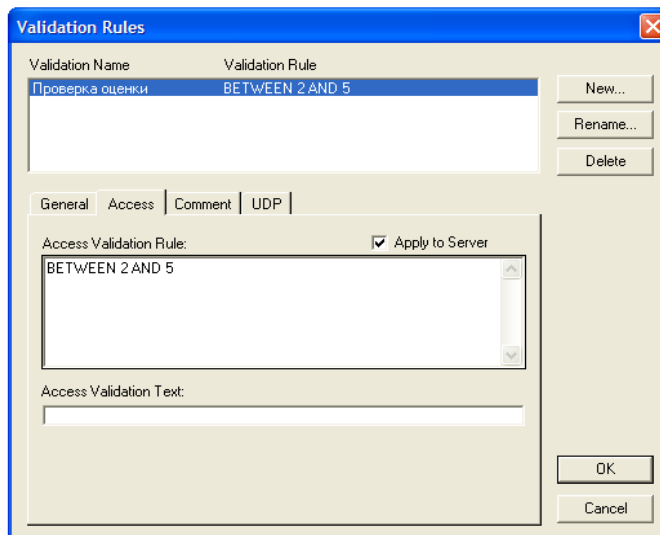


Рисунок 5.11 – Диалоговое окно Validation Rules с созданным правилом валидации «Проверка оценки»

Создайте таким же образом остальные правила согласно рис. 5.12.

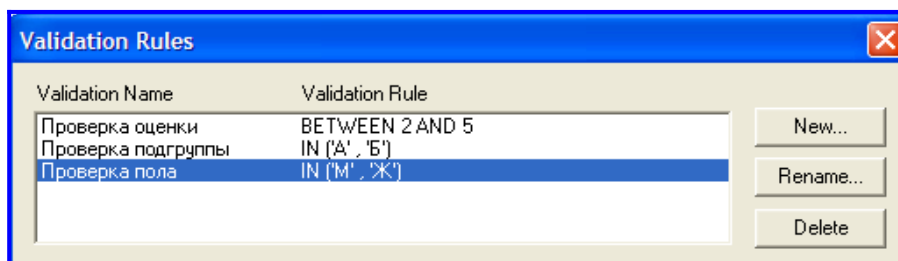


Рисунок 5.12 – Правила валидации

7. После создания правил валидации в диалоговом окне Column необходимо присвоить соответствующим колонкам таблиц установленные для них правила. Для этого встав на соответствующие колонки в поле Columns диалогового окна Columns таблиц «Студент», «Подгруппа» и «Зачетно-экзаменационная ведомость» выберите установленные для них правила.

Таким образом, проделав все вышеописанные действия, мы получили модель БД, готовую для помещения в СУБД.

8. Для генерации кода создания БД необходимо на панели меню выбрать меню Tools, а в нем – Forward Engineer / Schema Generation. При этом откроется диалоговое окно Access Schema Generation (рис. 5.13), в котором можно сделать соответствующие настройки.

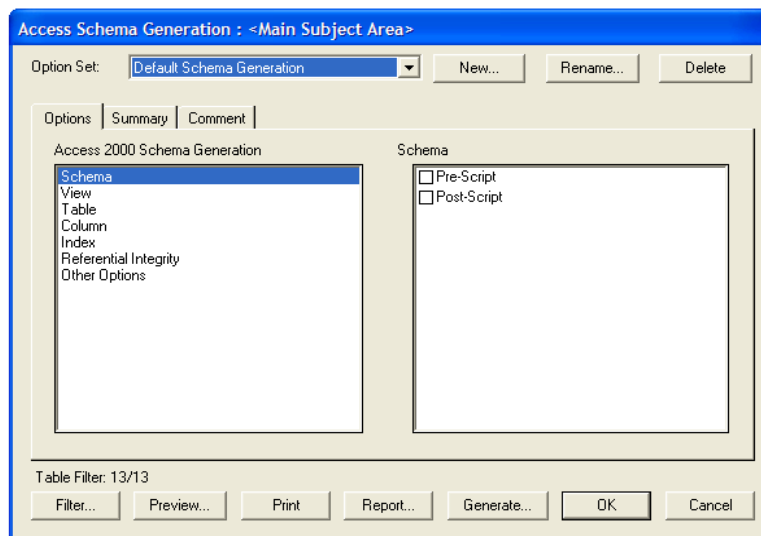


Рисунок 5.13 – Диалоговое окно Access Schema Generation

Диалог Access Schema Generation имеет 3 вкладки:

- **Options** – служит для задания опций генерации объектов БД – таблиц, представлений, колонок, индексов и т.д. Для задания опций генерации какого-либо объекта следует выбрать объект в левом списке закладки, после чего включить соответствующую опцию в правом списке.
- **Summary** – отображает все опции, заданные во вкладке Options.
- **Comment** – позволяет внести комментарии для каждого набора опций.

9. Просмотрите SQL-скрипт, создаваемый ERwin для генерации системного каталога СУБД. Для этого кнопкой Preview вызовите диалог Access Schema Generation Preview (рис. 5.14).

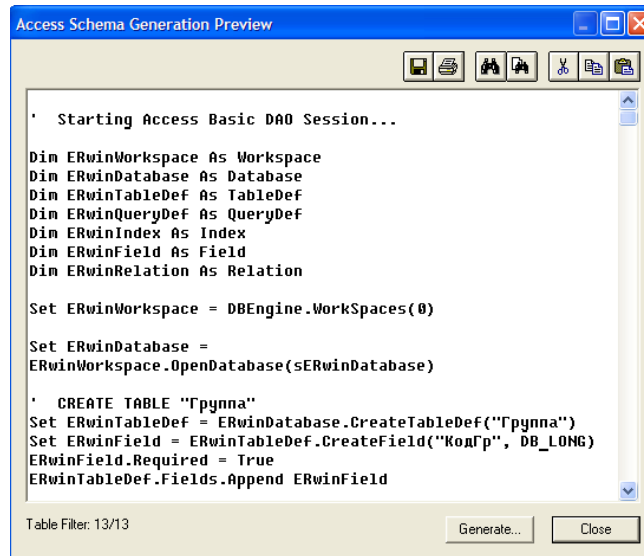


Рисунок 5.14 – Диалоговое окно Access Schema Generation Preview

10. Запустите процесс генерации схемы с помощью кнопки Generate. В процессе генерации ERwin связывается с БД, выполняя SQL-скрипт. Если в процессе генерации возникают какие-либо ошибки, то она прекращается, открывается окно с сообщениями об ошибках.

## Контрольные вопросы

1. Уровни физических моделей данных.
2. Понятие денормализации отношений.

## Задание на СРС

1. Построить физическую модель данных в 3 нормальной форме для рассматриваемого процесса.