

Частное учреждение образования
«Колледж бизнеса и права»

УТВЕРЖДАЮ
Заведующий
методическим кабинетом
_____ Е.В. Фалей
«___» _____ 2017

Специальность: «Программное обеспечение информационных технологий»	Дисциплина: «Базы данных и системы управления базами данных»
Составлена на основании учебной программы, утвержденной директором Колледжа бизнеса и права 30.12.2016	

Лабораторная работа № 2
Инструкционно-технологическая карта

Тема: Построение структуры реляционной БД

Цель работы: Научиться строить структуру реляционной БД

Время выполнения: 2 часа

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить раздел лекционного курса по теме «Общие понятия реляционного подхода к организации баз данных (БД)», а также теоритическую часть настоящей инструкционно-технологической карты.
2. Рассмотреть методику построения структуры реляционной БД, описанную в разделе «Пример выполнения работы» настоящей инструкционно-технологической карты.
3. Получить у преподавателя индивидуальное задание и выполнить процедуру построения реляционной модели данных из диаграммы «Сущность-связь», построив необходимый набор отношений. Определить состав атрибутов отношений. Определить первичные и внешние ключи отношений.
4. Ответить на контрольные вопросы.

1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

При проектировании логической структуры БД осуществляется преобразование исходной концептуальной модели в модель данных, поддерживаемую конкретной СУБД, и проверка адекватности полученной логической модели отображаемой предметной области.

Для любой предметной области существует множество вариантов проектных решений ее отображения в логической модели. Методика проектирования должна обеспечивать выбор наиболее подходящего проектного решения.

Этап создания логической модели называется *логическим проектированием*. При переходе от концептуальной (инфологической) модели к логической (дatalogической) следует иметь в виду, что концептуальная модель должна включать в себя всю информацию о предметной области, необходимую для проектирования базы данных. Преобразование концептуальной модели в логическую осуществляется по формальным правилам. Логическая модель базы данных строится в терминах информационных единиц, допустимых в той конкретной СУБД, в среде которой необходимо спроектировать базу данных.

Описание логической структуры базы данных на языке СУБД называется *схемой данных*.

Спроектировать логическую структуру базы данных означает определить все информационные единицы и связи между ними, задать их имена; если для информационных единиц возможно использование разных типов, то определить их тип. Следует также задать некоторые количественные характеристики, например, длину поля.

Любая система управления данными оперирует с допустимыми для нее логическими единицами данных. Кроме того, многие СУБД накладывают количественные и иные ограничения на структуру базы данных. Поэтому прежде чем приступить к построению логической модели, необходимо детально изучить особенности СУБД, возможности и целесообразности ее использования, ознакомиться с существующими методиками проектирования, а также проанализировать имеющиеся средства автоматизации проектирования.

Хотя логическое проектирование является проектированием логической структуры базы данных, на него оказывают влияние возможности физической организации данных, предоставляемые конкретной СУБД. Поэтому знание особенностей физической организации данных является полезным при проектировании логической структуры.

В базе данных отражается определенная предметная область. Поэтому процесс проектирования БД предусматривает предварительную классификацию объектов предметной области, систематизированное представление информации об объектах и связях между ними. На проектные решения оказывают влияние особенности требуемой обработки данных. Поэтому соответствующая информация должна быть определенным образом представлена и проанализирована на начальных этапах проектирования БД.

В концептуальной модели должна быть отображена вся информация, циркулирующая в информационной системе. Но это не означает, что вся она должна храниться в базе данных и все сущности, зафиксированные в концептуальной модели должны в явном виде отражаться в логической модели. Прежде чем строить логическую модель, необходимо решить, какая информация будет храниться в базе данных. Например, в концептуальной модели должны быть отражены вычисляемые показатели, но совсем не обязательно, что они должны храниться в базе данных. В связи с этим одним из первых шагов проектирования является определение состава БД, т. е. перечня тех

показателей, которые целесообразно хранить в базе данных.

Кроме того, не все виды связей, существующие в предметной области, могут быть непосредственно отображены в конкретной логической модели. Так, многие СУБД не поддерживают непосредственно отношение «многие ко многим» между элементами. В этом случае в логическую модель вводится дополнительный вспомогательный элемент, отображающий эту связь (таким образом, отношение «многие ко многим» как бы разбивается на два отношения «один ко многим» между этим вновь введенным элементом и исходными элементами).

Ниже описана типовая пошаговая процедура преобразования диаграммы «Сущность –связь» в реляционную схему базы данных.

1. Каждая сущность превращается в таблицу. Имя сущности становится именем таблицы. Экземплярам типа сущности соответствуют строки соответствующей таблицы.

2. Каждый атрибут становится столбцом таблицы с тем же именем; может выбираться более точный формат представления данных. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам – не могут.

3. Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы. Если имеется несколько возможных уникальных идентификаторов, для первичного ключа выбирается наиболее характерный. Если в состав уникального идентификатора входят связи, к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на другом конце связи. Для именования этих столбцов используются имена концов связей и/или имена парных сущностей.

4. Связи «один к одному» становятся внешними ключами. Необязательные связи соответствуют столбцам внешнего ключа, допускающим наличие неопределенных значений; обязательные связи — столбцам, не допускающим неопределенных значений. Если между двумя сущностями А и В имеется связь «один к одному», то соответствующий внешний ключ по желанию проектировщика может быть объявлен как в таблице А, так и в таблице В.

5. Для поддержки связи «многие ко многим» между типами сущности А и В создается дополнительная таблица С с двумя обязательными столбцами, один из которых содержит уникальные идентификаторы экземпляров сущности А, а другой — уникальные идентификаторы экземпляров сущности В.

6. Связи типа «один ко многим» между типами сущности А и В реализуется через внешний ключ. Ключ вводится для того отношения, к которому осуществляется множественная связь.

Для привязки логической модели к среде хранения используется модель данных физического уровня, для краткости часто называемая физической моделью. Эта модель определяет способы физической организации дан-

ных в среде хранения. Модель физического уровня строится с учетом возможностей, предоставляемых СУБД. Описание физической структуры базы данных называется *схемой хранения*. Соответствующий этап проектирования БД называется *физическим проектированием*. СУБД обладают разными возможностями по физической организации данных. В связи с этим различаются для конкретных систем сложность и трудоемкость физического проектирования и набор выполняемых шагов.

Физическая модель данных описывает данные средствами конкретной СУБД. На этапе физического проектирования учитывается специфика конкретной модели данных и специфика конкретной СУБД. Отношения, разработанные на стадии формирования логической модели данных, преобразуются в таблицы, атрибуты становятся столбцами таблиц, для ключевых атрибутов создаются уникальные индексы, домены преобразуются в типы данных, принятые в используемой СУБД. Ограничения, имеющиеся в логической модели данных, реализуются различными средствами СУБД, например, при помощи индексов, ограничений целостности, триггеров, хранимых процедур. При этом решения, принятые на уровне логического моделирования, определяют некоторые границы, в пределах которых можно развивать физическую модель данных. Точно так же в пределах этих границ можно принимать различные решения. Например, отношения, содержащиеся в логической модели данных, должны быть преобразованы в таблицы, но для каждой таблицы можно дополнительно объявить различные индексы, повышающие скорость обращения к данным. Многое тут зависит от конкретной СУБД.

Физическое проектирование БД предполагает также выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения.

На этапе физического моделирования производится оценка требований к вычислительным ресурсам, необходимым для функционирования системы, выбор типа и конфигурации ЭВМ, типа и версии операционной системы. Выбор зависит от таких показателей, как:

- примерный объем данных в БД;
- динамика роста объема данных;
- характер запросов к данным (извлечение и обновление отдельных записей) обработка групп записей, обработка отдельных отношений или соединение отношений);
- интенсивность запросов к данным по типам запросов
- требования к времени отклика системы по типам запросов.

Выбор СУБД и инструментальных программных средств является одним из важнейших моментов в разработке проекта базы данных, так как он принципиальным образом влияет на весь процесс проектирования БД и реализации информационной системы. Теоретически при осуществлении этого выбора нужно принимать во внимание десятки факторов. Но на практике разработчики руководствуются лишь собственной интуицией и несколькими наиболее важными критериями, к которым, в частности, относятся:

- тип модели данных, которую поддерживает данная СУБД, аде-

кватность модели данных структуре рассматриваемой ПО;

- характеристики производительности СУБД;
- запас функциональных возможностей для дальнейшего развития информационной системы;
- степень оснащенности СУБД инструментарием для персонала администрирования данными;
- удобство и надежность СУБД в эксплуатации;
- стоимость СУБД и дополнительного программного обеспечения.

Проектирование является важнейшей стадией при создании базы, так как именно на этом этапе принимаются очень важные стратегические решения, влияющие на весь процесс создания эффективной информационной системы.

2. ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

Постановка задачи. Необходимо разработать систему учета проведения занятий в школе в течение одного учебного года в соответствии со следующим описанием. Занятия в школе организуются для классов. Все ученики распределены по классам (по возрастному признаку), причем каждый ученик может быть включен только в один класс, и численность класса ограничена. Каждый класс изучает определенные предметы. Перечень предметов для класса определен и ограничен. Предметы преподаются учителями школы. В старшие классы назначается классный руководитель.

Диаграмма «Сущность-связь» (далее – ER-диаграмма) рассматриваемой задачи представлена на рисунке 1.

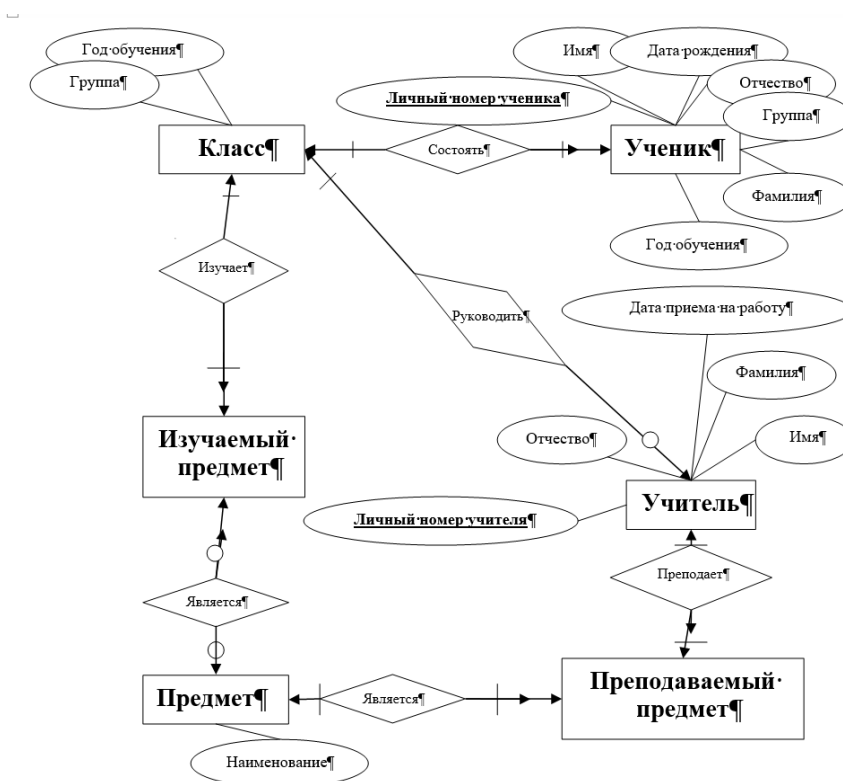


Рисунок 1

Первые шаги преобразования состоят в превращении каждой сущности в отношение (таблицу). Каждое свойство становится атрибутом – столбцом соответствующей таблицы.

Результат реализации первых шагов для ER-диаграмма, представленной на рисунке 1, описан ниже:

Класс (Год_обучения, Группа)

Предмет (Наименование)

Ученик (Личный_номер_ученика, Фамилия, Имя, Отчество, Дата_рождения, Год_обучения)

Учитель (Личный_номер_учителя, Фамилия, Имя, Отчество, Дата_приема_на_работу)

Изучаемый предмет

Преподаваемый предмет

Далее необходимо преобразовать связи во внешние ключи. При этом, если для однозначной идентификации экземпляров сущности не достаточно атрибутов, достаточно добавить в таблицу искусственный столбец. Например, для однозначной идентификации сущности Класс достаточно добавить столбец ID_класса.

Результат преобразования связи во внешние ключи представлен ниже:

Класс (**ID_класса (PK)**, Год_обучения, Группа, **Личный_номер_учителя (FK)**)

Предмет (**ID_предмета (PK)**, Наименование)

Ученик (**Личный_номер_ученика (PK)**, Фамилия, Имя, Отчество, Дата_рождения, Год_обучения, **ID_класса (FK)**)

Учитель (**Личный_номер_учителя (PK)**, Фамилия, Имя, Отчество, Дата_приема_на_работу)

Изучаемый предмет (**ID_класс_предмет (PK)**, **ID_класса (FK)**, **ID_предмета (FK)**)

Преподаваемый предмет (**ID_предмет_учитель (PK)**, **ID_предмета (FK)**, **Личный_номер_учителя (FK)**)

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите, как называется этап создания логической модели данных.
2. Дайте определение понятию «схема данных».
3. Назовите, что включает в себя понятие «спроектировать логическую структуру базы данных».
4. Опишите типовую пошаговую процедуру преобразования диаграммы «Сущность – связь» в реляционную схему базы данных.
5. Дайте определение понятию «схема хранения».
6. Поясните этап физического проектирования базы данных.

ДОМАШНЕЕ ЗАДАНИЕ

[1], страницы 116-137, 495-569

[2], страницы 196-201

ЛИТЕРАТУРА

1. Коннолли, Т. Базы данных: проектирование, реализация, сопровождение. Теория и практика: пер. с английского / Т. Коннолли, К. Бегг. – М.: Изд. дом Вильямс, 2003.
2. Федорова, Г. Разработка и администрирование баз данных: учеб. для студ. учреждений сред. проф. образования / Г.Н. Федорова. – М. : Издательский центр «Академия», 2015 – 320 с.
3. Сеть разработчиков Microsoft [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library>
4. Федорова, Г. Основы проектирования баз данных / Г.Н. Федорова. - М.: Академия, 2014. – 219 с.

Преподаватель

С.В. Банцевич

Рассмотрено на заседании цикловой
комиссии программного обеспечения
информационных технологий №10
Протокол № от « » _____ 2017
Председатель ЦК С.В. Банцевич