# ER-диаграмма

ER-диаграмма является очень удачным решением моделирования. В ней сочетаются функциональный и информационный подходы, что позволяет представлять как совокупность выполняемых функций, так и отношения между элементами системы, задаваемые структурами данных.

Сущности. Каждый тип сущности в ER-диаграммах представляется в виде прямоугольника, содержащего имя сущности. В качестве имени обычно используются существительные (или обороты существительного) в единственном числе. Для отражения сущностей слабых типов используются прямоугольники, стороны которых рисуются двойными линиями. Например, в рассматриваемой далее ER-диаграмме, приведенной на рис. 5.4, ПОДЧИНЕННЫЙ — сущность слабого типа.

Свойства. Свойства служат для уточнения, идентификации, характеристики или выражения состояния сущности или связи. Свойства отображаются в виде эллипсов, содержащих имя свойства. Эллипс соединяется с соответствующей сущностью или связью линией.
Имена ключевых свойств подчеркиваются, например, свойство «Табельный номер» сущности СОТРУДНИК.

Контур эллипса рисуется двойной линией, если свойство многозначное, например, свойство «Специальность» сущности СОТРУДНИК.

Контур эллипса рисуется штриховой линией, если свойство производное, например, свойство «Кол-во» сущности ПОСТАВЩИК.

Эллипс соединяется пунктирной линией, если свойство условное, например, свойство «Иностранный язык» сущности СОТРУДНИК.

Если свойство составное, то составляющие его свойства отобра жаются другими эллипсами, соединенными с эллипсом составного, например, свойство «Адрес» сущности СОТРУДНИК состоит из простых свойств «Город», «Улица», «Дом».

Связи. Связь — это графически изображаемая ассоциация, уста навливаемая между сущностями. Каждый тип связи на ER-диаграмме отображается в виде ромба с именем связи внутри. В качестве имени обычно используются отглагольные существительные.

Стороны ромба рисуют двойными линиями, если это связь сущности слабого типа с сущностью, от которой она зависит. Например, связь «Подчинение», связывающая сущность слабого типа ПОДЧИ НЕННЫЙ с сущностью СОТРУДНИК, от которой она зависит.

Участники связи соединены со связью линиями. Двойная линия обозначает полное участие сущности в связи с данной стороны. Напри мер, связь «Подчинение» со стороны сущности ПОДЧИНЕННЫЙ.

Связь может быть модифицирована указанием роли. Например, для рекурсивной связи «Состав» указаны роли: «Деталь состоит из ...» и «Деталь входит в состав ...».

Тип связи указывается индексами «1» или «М» над соответствующей линией. Например, связь «Руководство» имеет тип «один ко многим»: один сотрудник может руководить многими проектами; связь «Участие» имеет тип «многие ко многим»: один сотрудник может участвовать во многих проектах, и в проекте могут участвовать многие сотрудники.

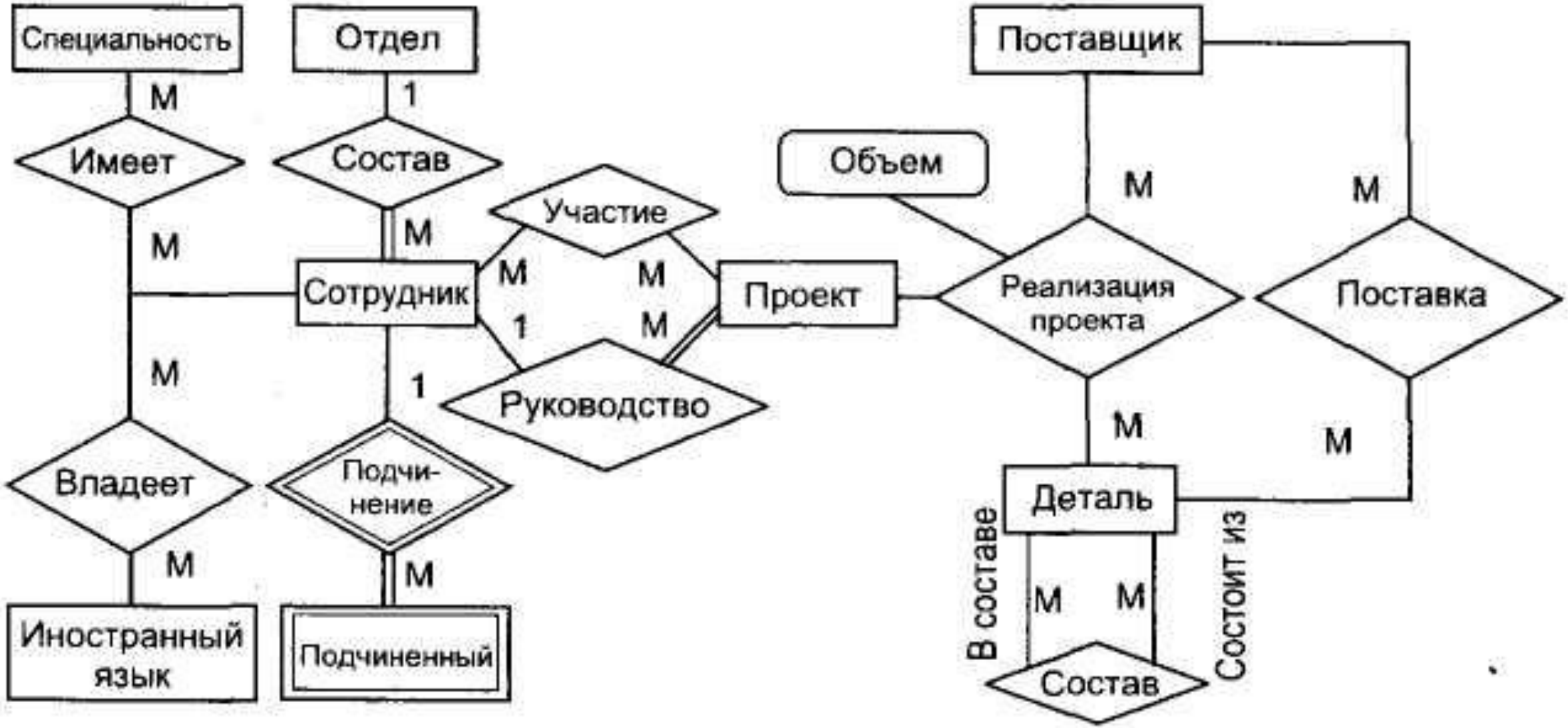
# Нормальные формы ER-диаграмм

В первой нормальной форме ER-диаграммы устраняются повторяющиеся атрибуты или группы атрибутов, т. е. производится выявление неявных сущностей, «замаскированных» под атрибуты.

Во второй нормальной форме устраняются атрибуты, зависящие только от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

В третьей нормальной форме устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности.

На рис. 12 представлена ER-диаграмма рис. 10 в третьей нормальной форме.


Специальность
Отдел
Поставщик
Объем
Имеет
Состав
Участие
Сотрудник
Проект
Реализация проекта
Поставка
Руководство
Владеет
Подчи-нение
Деталь
Иностранный язык
Подчиненный
Состав
В составе
Состоит из
M
1

# Даталогические модели

Задачей следующей стадии проектирования системы базы данных является выбор подходящей СУБД и отображение в ее среду (структуру данных) спецификаций инфологической модели предметной области. Результатом даталогического проектирования базы данных является концептуальная схема базы данных, включающая определение всех информационных элементов (единиц) и связей, в том числе задание типов, характеристик и имен.

Даталогическое проектирование оперирует логическими понятиями, связанными со структурой базы данных, но особенности представления данных, правила и языки агрегирования и манипулирования данными имеют определяющее влияние. Не все виды связей, например, «многие ко многим», могут быть непосредственно отображены в логической модели.

Может быть много вариантов отображения инфологической модели предметной области в даталогическую модель базы. Следует учитывать влияние двух факторов.

Во-первых, связи предметной области могут отображаться двумя путями: как декларативным — в логической схеме, так и процедурным — отработкой связей через программные модули, обрабатывающие (связывающие) соответствующие хранимые данные.
Во-вторых, существенным фактором может оказаться характер обработки информации. Например, частые обращения к совместно обрабатываемым данным, очевидно, предполагают их совместное хранение, а данные (особенно большого объема), к которым обращаются редко, целесообразно хранить отдельно от часто используемых.
Рассмотрим по шагам общий подход к построению реляционной базы данных на основе инфологической модели, представленной ER-диаграммой.

## Получение реляционной схемы из ER-диаграммы

1.Каждая простая сущность превращается в таблицу (отношение). Имя сущности становится именем таблицы.

2.Каждый атрибут становится возможным столбцом с тем же именем. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, — не могут. Если атрибут является множественным, то для него строится отдельное отношение.

3.Компоненты уникального идентификатора сущности превращаются в первичный ключ. Если имеется несколько возможных уникальных идентификаторов, выбирается наиболее используемый. Если в состав уникального идентификатора входят связи, то к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (этот процесс может продолжаться рекурсивно). Для именования этих столбцов используются имена концов связей и/или имена сущностей.

4.Связи «многие к одному» и «один к одному» становятся внешними ключами. Т.е. создается копия уникального идентификатора с конца связи «один», и соответствующие столбцы составляют внешний ключ.

5. Индексы создаются для первичного ключа (уникальный индекс), а также внешних ключей и тех атрибутов, которые будут часто использоваться в запросах.

6. Если в концептуальной схеме присутствуют подтипы, то возможны два варианта.
*Все подтипы хранятся в одной таблице, которая создается для самого внешнего супертипа, а для подтипов создаются представления. В таблицу добавляется по крайней мере один столбец, содержащий код ТИПА, и он становится частью первичного ключа.*

*Во втором случае для каждого подтипа создается отдельная таблица (для более нижних — представления) и для каждого подтипа первого уровня супертип воссоздается следующим образом: из всех таблиц подтипов выбираются общие столбцы — столбцы супертипа.*

7. Если остающиеся внешние ключи все принадлежат одному домену, т. е. имеют общий формат, то создаются два столбца: идентификатор связи и идентификатор сущности. Столбец идентификатора связи используется для различения связей. Столбец идентификатора сущности используется для хранения значений уникального иденти фикатора сущности на дальнем конце соответствующей связи.

*Если результирующие внешние ключи не относятся к одному домену, то для каждой связи, покрываемой дугой исключения, создаются явные столбцы внешних ключей.*

# Физические модели

Стадия физического проектирования базы данных в общем случае включает:

•выбор способа организации базы данных;

•разработку спецификации внутренней схемы средствами модели данных ее внутреннего уровня;

•описание отображения концептуальной схемы во внутреннюю.

Важно заметить, что в отличие от ранних СУБД, многие современные системы не предоставляют разработчику какого-либо выбора на этой стадии. Реально к вопросам проектирования физической модели можно отнести выбор схемы размещения данных.

Способ хранения базы данных определяется механизмами СУБД автоматически «по умолчанию» на основе спецификаций концептуальной схемы базы данных, и внутренняя схема в явном виде в таких системах не используется.

Внешние схемы базы данных обычно конструируются на стадии разработки приложений.

# Проектирование реляционной базы данных

Задача проектирования БД для предметной области состоит в том, чтобы обеспечить поддержку не только любых ныне используемых, но и будущих приложений. Таким образом, БД создают основу для обработки неформализованных, изменяющихся и неизвестных запросов и приложений, для которых невозможно заранее определить требования к данным. Это позволяет в дальнейшем строить на основе предметных БД достаточно стабильные информационные системы, т. е. системы, в которых большинство изменений можно осуществить без переписывания старых приложений.

Задача проектирования БД — это сокращение избыточности хранимых данных, а следовательно, экономия объема используемой памяти, уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте. Такой проект БД можно создать, используя методологию нормализации отношений.

# Универсальное отношение

Рассмотрим задачу проектирования БД на базе сводной таблицы, пример которой приведен на рис. 6.1. Предложенная таблица отражает результаты сдачи сессии (шкала оценок: 0 — незачет; 1 — зачет; 2, 3, 4, 5 — экзаменационная оценка).

| ФИО студента | Семестр | Дисциплина | Форма отчетности | Оценка | Количество часов | ФИО преподават |
|---|---|---|---|---|---|---|
| Иванов В. П. | 1 | Английский язык | зачет | 1 | 60 | Цветкова А.Ю |
| | | Математический анализ | зачет | 1 | 28 | Рыбин К. К. |
| | | Математический анализ | экзамен | 5 | 32 | Раков И.И. |
| | | Программирование | зачет | 1 | 36 | Незабудкина З.П. |
| | | Программирование | экзамен | 5 | 32 | Зайчиков А. А |
| | | Линейная алгебра | зачет | 1 | 24 | Волков Г. И. |
| | | Линейная алгебра | экзамен | А | 28 | Волков Г.И. |
| | | История Отечества | экзамен | 5 | 24 | Москвин А. П |
| Петрова А Л. | 1 | Английский язык | зачет | 1 | 60 | Цветкова А.Ю |
| | | Математический анализ | зачет | 1 | 28 | Рыбин К. К. |
| | | Математический анализ | экзамен | 3 | 32 | Раков И.И. |
| | | Программирование | зачет | 1 | 36 | Незабудкина З.П. |
| | | Программирование | экзамен | 4 | 32 | Зайчиков А. А |
| | | Линейная алгебра | зачет | 1 | 24 | Волков Г.И. |

Такое преобразование приводит к возникновению большого объема избыточных данных.

Таблица на рис. 14 представляет собой корректное отношение. Такое отношение называют универсальным отношением проектируемой БД. В одно универсальное отношение включаются все представляющие интерес атрибуты, и оно может содержать все данные, которые предполагается размещать в БД в будущем. При проекти ровании некоторых БД универсальное отношение может использоваться в качестве отправной точки.

Однако при использовании универсального отношения возникают, по крайней мере, две проблемы.

1. Избыточность данных. Значения столбцов таблицы многократно повторяются. Повторяются также и некоторые наборы значений столбцов, например, данные о дисциплине.

2. Потенциальная противоречивость. Если при вводе данных, например, количества часов для дисциплины «Английский язык», была допущена ошибка, то для ее исправления необходимо найти все строки, содержащие сведения об этой дисциплине, и во всех этих строках произвести изменения. Более того, при заполнении такой таблицы могут быть использованы различные формы записи одного и того же значения, например: «Англ. язык» и «Английский язык», «Мат. анализ» и «Математический анализ».

Решение этих проблем состоит в разделении данных и связей, т. е. в выделении в отдельные таблицы сведений о студентах, преподавателях, дисциплинах и результатах сдачи экзаменов (рис. 15).

**Студенты Преподаватели Дисциплины**

| № | **ФИО студента** |
|---|---|
| 1 | Иванов В. П. |
| 2 | Петрова А.П. |
| 3 | Сидоров **К.** К. |

| № | ФИО преподавателя |
|---|---|
| 1 | Волков Г. И. |
| 2 | Зайчиков А. А. |
| 3 | Карпов К. Ю. |
| 4 | Лабиринтов Е. Н. |
| 5 | Москвин А. П. |
| 6 | Незабудкина З. П. |
| 7 | Пиков И. И. |
| 8 | Рыбин К. К. |
| 9 | Соболев И. Г. |
| 10 | Цветкова А. Ю. |

| № | Дисциплина |
|---|---|
| 1 | Алгоритмы и структуры данных |
| 2 | Английский язык |
| 3 | История Отечества |
| 4 | Линейная алгебра |
| 5 | Математический анализ |
| 6 | Операционные системы, среды и оболочки |
| 7 | Программирование |
| 8 | Теория вероятностей и математическая статистика |
| 9 | Экономическая теория |

***Рис. 15 Разделение данных и связей***

Заменим в таблицах «Результаты сессии» и «Учебный план» конкретные значения на их номера в других таблицах и получим, помимо значительного упрощения процедуры модификации текстовых значений, дополнительные возможности по включению строк в таблицы «Студенты», «Преподаватели», «Дисциплины», что значительно расширяет возможности БД.

**Учебный план**

| № | Дисциплина | Семестр | Кол-во часов | Форма отчетности | Преподаватель |
|---|---|---|---|---|---|
| 1 | 2 | 1 | 60 | зачет | 10 |
| 2 | 3 | 1 | 24 | экзамен | 5 |
| 3 | 4 | 1 | 24 | зачет | 1 |
| 4 | 4 | 1 | 28 | экзамен | 1 |
| 5 | 5 | 1 | 28 | зачет | 8 |
| 6 | 5 | 1 | 32 | экзамен | 7 |
| 7 | 7 | 1 | 36 | зачет | 6 |
| 8 | 7 | 1 | 32 | экзамен | 2 |
| 9 | 2 | 3 | 60 | зачет | 10 |
| 10 | 5 | 3 | 20 | зачет | 3 |
| 11 | 5 | 3 | 28 | экзамен | 7 |
| 12 | 1 | 3 | 32 | экзамен | 2 |
| 13 | 8 | 3 | 32 | экзамен | 9 |
| 14 | 6 | 3 | 36 | зачет | 6 |
| 15 | 6 | 3 | 32 | экзамен | 6 |
| 16 | 9 | 3 | 24 | зачет | 4 |

**Результаты сессии**

| Студент | Учебный план | Оценка |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 2 | 5 |
| 1 | 3 | 1 |
| 1 | 4 | 4 |
| 1 | 5 | 1 |
| 1 | 6 | 5 |
| 1 | 7 | 1 |
| 1 | 8 | 5 |
| 2 | 1 | 1 |
| 2 | 2 | 5 |
| 2 | 3 | 1 |
| 2 | 4 | 4 |
| 2 | 5 | 1 |
| 2 | 6 | 3 |

Теперь при изменении названия «Математический анализ» на «Мат. анализ» исправляется единственное значение в таблице «Дисциплины». И даже если оно вводится с ошибкой, то это не может повлиять на связь между дисциплиной, преподавателем и студентом (в связующей таблице «Результаты сессии» используются номера дисциплин учебного плана, а не их названия).

## Функциональная и многозначная зависимости

Функциональная зависимость, по сути, является связью типа «многие к одному» между множествами атрибутов (столбцов) рассматриваемого отношения.

Например, в таблице «Учебный план» столбцы Дисциплина, Семестр и Форма отчетности функционально зависят от ключа № (порядковый номер) в учебном плане, а в таблице «Результаты сессии» столбец Оценка функционально зависит от составного ключа (Студент, Учебный план).

Многозначная зависимость. Говорят, что один атрибут таблицы многозначно определяет другой атрибут той же таблицы, если для каждого значения первого атрибута существует хорошо определенное множество соответствующих значений второго атрибута.

# Нормальные формы

Таблица находится в первой нормальной форме (1НФ) тогда и только тогда, когда в любом допустимом значении этой таблицы каждая ее строка содержит только одно значение для каждого атрибута (столбца).

Из таблиц, рассмотренных ранее, не удовлетворяет этим требованиям (т. е. не находится в 1НФ) только таблица на рис.13.

Таблица находится во второй нормальной форме (2НФ), если она удовлетворяет определению 1НФ и все ее атрибуты (столбцы), не входящие в первичный ключ, связаны полной функциональной за висимостью с первичным ключом.

Не удовлетворяют этим требованиям таблицы, представленные на рис. 13 и на рис. 14. Таблица 14 имеет составной первичный ключ (ФИО студента, Семестр, Дисциплина, Форма отчетности) и содержит множество неключевых атрибутов (Оценка, Количество часов, ФИО преподавателя), зависящих лишь от той или иной части первичного ключа. Так, атрибуты Количество часов и ФИО преподавателя зависят только от атрибутов Семестр, Дисциплина, Форма отчетности. Следовательно, эти атрибуты не связаны с первичным ключом полной функциональной зависимостью.

Ко второй нормальной форме приведены все таблицы рис. 15.

Таблица находится в третьей нормальной форме (3НФ), если она удовлетворяет определению 2НФ и ни один из ее неключевых атрибутов не связан функциональной зависимостью с любым другим неключевым атрибутом.

Таблица «Учебный план» (рис. 15), очевидно, не находилась бы в третьей нормальной форме, если включала бы в себя столбец Должность преподавателя. В этом случае необходимо было бы провести декомпозицию таблицы «Учебный план» и в результате получить дополнительную таблицу «Кадровый состав» с атрибутами: №, ФИО преподавателя, Должность преподавателя.

Следует отметить, что в таблице «Учебный план» на самом деле существует функциональная зависимость между атрибутами Количество часов и ФИО преподавателя, с одной стороны, и совокупностью атрибутов Семестр, Дисциплина и Форма отчетности — с другой. Однако тройка атрибутов {Семестр, Дисциплина и Форма отчетности) в свою очередь может выступать в качестве первичного ключа, который представлен в таблице атрибутом Порядковый номер. Чтобы избегать в процессе нормализации подобных противоречий, Кодд и Бойс обосновали и предложили более строгое определение для ЗНФ, которое учитывает, что в таблице может быть несколько первичных ключей.