

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ**

КАФЕДРА «ЭВМ и СИСТЕМЫ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по дисциплине

«БАЗЫ И БАНКИ ДАННЫХ»

Часть 3. ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

для студентов специальности

«Автоматизированные системы обработки информации»

Брест 2001

Данные указания являются частью методических разработок авторов в области проектирования структур баз данных. Рассмотрена структура процесса проектирования реализации баз данных, определены основные понятия теории реляционных моделей данных, приведена методика перехода от концептуальной модели предметной области к реляционной структуре базы данных и ее нормализация. Рассмотрены примеры проектирования реляционной структуры базы данных, даны методические указания для выполнения лабораторных работ по данной тематике.

Методические указания предназначены для использования студентами специальностей Т.10.01.00 и Т.10.03.00 в ходе выполнения лабораторных и самостоятельных работ по дисциплинам «Базы и банки данных», «Проектирование систем обработки информации», «Базы данных, знаний и экспертные системы», «Технология разработки программных систем» и при выполнении курсовых и дипломных работ.

Ил. 4, прил. 2, список лит. 7 назв.

Составители: В.Г. Брич, доцент, к.т.н.
В.И. Хвещук, доцент, к.т.н.

Рецензенты: А.А. Прихожий, профессор, д.т.н., БГПА
О.И. Бугай, доцент, к.т.н., БГПА
И.А. Бухвалова, ст.преподаватель, БГПА

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТИРОВАНИИ БАЗ ДАННЫХ	4
2. ПОСТРОЕНИЕ РЕЛЯЦИОННОЙ СТРУКТУРЫ БАЗЫ ДАННЫХ	6
2.1. ОРГАНИЗАЦИЯ ДАННЫХ	6
2.2. МОДЕЛИ ДАННЫХ	7
2.3. ТИПЫ ОТНОШЕНИЙ	10
2.4. КЛЮЧИ	11
2.5. ТИПЫ ФУНКЦИОНАЛЬНЫХ ЗАВИСИМОСТЕЙ	13
2.6. МЕТОДИКА ПОСТРОЕНИЕ РЕЛЯЦИОННОЙ СТРУКТУРЫ БД	15
3. НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ В БАЗЕ ДАННЫХ	16
3.1. ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА	17
3.2. ВТОРАЯ НОРМАЛЬНАЯ ФОРМА	18
3.3. ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА	19
3.4. УСИЛЕННАЯ НОРМАЛЬНАЯ ФОРМА БОЙСА - КОДДА	20
3.5. МНОГОЗНАЧНЫЕ ЗАВИСИМОСТИ	21
3.6. ПЯТАЯ НОРМАЛЬНАЯ ФОРМА	24
3.7. ДРУГИЕ НОРМАЛЬНЫЕ ФОРМЫ	26
ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ	27
СПИСОК ЛИТЕРАТУРЫ	27
ПРИЛОЖЕНИЕ №1. ПОСТРОЕНИЕ РЕЛЯЦИОННОЙ СТРУКТУРЫ БАЗЫ ДАННЫХ	28
ПРИЛОЖЕНИЕ №2. НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ В БАЗЕ ДАННЫХ	30

1. ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТИРОВАНИИ БАЗ ДАННЫХ

В методических указаниях рассмотрены вопросы проектирования реализации структуры базы данных (БД) с ориентацией на реляционную модель данных. Детально представлены два аспекта этого процесса - преобразование концептуальной модели (КМ) предметной области (ПрО) в реляционную структуру БД и нормализация этой структуры БД.

В основу методических указаний положены теоретические и практические материалы изложенные в [1-3].

Процесс проектирования систем БД включает следующие основные компоненты:

ИСХОДНЫЕ ДАННЫЕ

РЕЗУЛЬТАТЫ

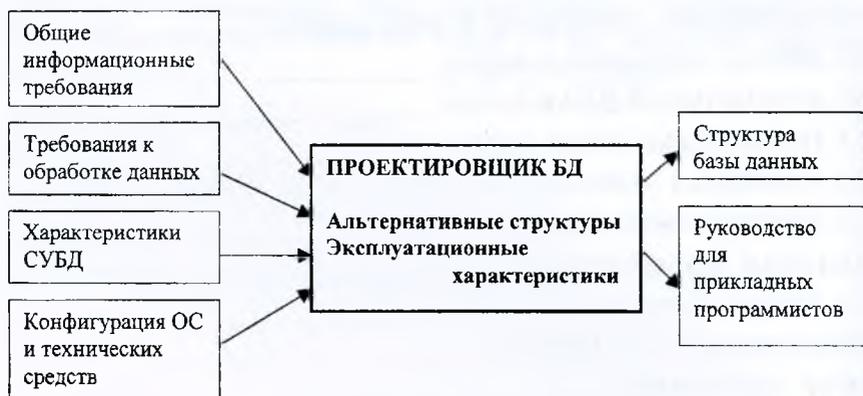


Рис.1.1. Структура проектирования базы данных.

К основным исходным данным процесса проектирования БД относятся:

- **общие информационные требования**, включающие формулирование цели на создание системы обработки данных (СОД), определение совокупности данных, включаемых в БД, описание использования данных конкретными пользователями, подразделениями и т.д. Эти требования не связаны с каким-либо конкретным приложением;
- **требования к обработке данных** включают - специфические элементы данных, требуемые для каждого приложения, объем данных и частота обработки данных для каждого приложения;
- **характеристики и конфигурация** системы управления базами данных (СУБД), операционной системы (ОС) и технических средств.

Основными результатами проектирования БД являются следующие:

- **полная структура БД** (логические и физические компоненты);
- **руководства для прикладных программистов**, основанные на структуре БД и требованиях к обработке данных;
- **эксплуатационных характеристик и ограничений системы.**

Проектирование баз данных представляет собой трудоемкий, длительный, итеративный и во многих случаях не формализуемый процесс. С точки зрения проектировщика и пользователя БД жизненный цикл систем БД можно представить в виде двух фаз, которые в свою очередь включают следующие этапы:

ФАЗА АНАЛИЗА И ПРОЕКТИРОВАНИЯ:

1. Формулирование и анализ требований.
2. Концептуальное проектирование.
3. Проектирование реализации.
4. Физическое проектирование.

ФАЗА РЕАЛИЗАЦИИ И СОПРОВОЖДЕНИЯ:

1. Реализация базы данных.
2. Анализ функционирования и поддержка.
3. Модификация и адаптация.

Структура процесса проектирования реализации БД (этап 3 фазы анализа и проектирования) представлена на рис.1.2.

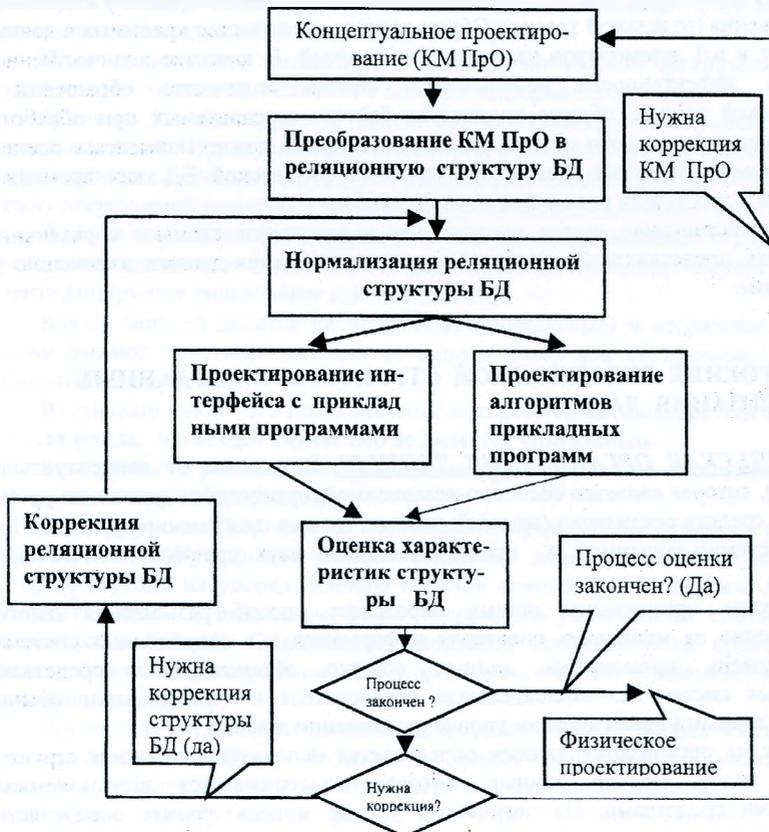


Рис.1.2. Структура процесса проектирования реализации базы данных

Исходной информацией для проектирования реализации БД являются результаты концептуального проектирования. Рассматриваемый в данных методических указаниях этап «Проектирование реализации» включает следующие действия:

- преобразование концептуальной модели предметной области (результат этапа концептуального проектирования) в реляционную базу данных;
- нормализацию реляционной структуры базы данных;
- проектирование интерфейса конечных пользователей с данными (со своей подсхемой, которая является подмножеством структуры БД) в процессе решения их задач. Отдельная задача реализуется отдельной прикладной программой (ПП), которая входит в состав программного обеспечения (ПО) СОД;
- проектирование ПП заключается в разработке схемы или структуры программ, обеспечивающих реализацию пользовательских задач. Определяются функции доступа к данным и данные для оценки схемы БД;
- оценка схемы заключается в количественной оценке логической структуры БД на основе эффективности функционирования СОД. Определяется объем обработки как сочетание двух параметров- частоты обработки и объем данных. Частота обработки - частота, с которой нужно проводить обработку отдельного приложения (отдельной задачи). Объем данных - количество хранимых в данный момент в БД экземпляров каждого типа записей. В качестве количественных оценок эффективности используются оценки количества обращений к логической записи, общего количества байтов, передаваемых при обработке приложения, и полного объема БД. С их помощью можно попытаться оценить усредненные значения таких характеристик физической БД, как временные затраты и требуемый объем физической памяти;
- усовершенствование схемы заключается в адаптации схемы к различным способам представления данных. Содержание и состав данных изменению не подлежит.

2. ПОСТРОЕНИЕ РЕЛЯЦИОННОЙ СТРУКТУРЫ БАЗЫ ДАННЫХ

2.1. ОРГАНИЗАЦИЯ ДАННЫХ

ЛОГИЧЕСКАЯ ОРГАНИЗАЦИЯ ДАННЫХ. В отличие от концептуальной модели ПрО, которая является системно-независимой, организация данных на уровне конкретных средств реализации (моделей данных, языков программирования, СУБД) является системно-зависимой и представляется на двух уровнях: логическом и физическом.

Физическая организация данных определяет способ размещения данных непосредственно на машинных носителях информации и в современных системах данный уровень организации данных обычно обеспечивается средствами операционных систем без вмешательства пользователя и пользователь обычно оперирует с данными на логическом уровне организации данных.

Логическая организация данных определяется используемым типом структур данных и видом модели данных, которая поддерживается используемыми программными средствами. На логическом уровне модель данных определяется тремя компонентами:

1. Допустимой структурой данных, разнообразием и количеством типов объектов, которые можно описать с помощью этой модели.
2. Множеством допустимых операций над данными.
3. Ограничениями для контроля целостности.

Эти логические ограничения модели, накладываемые на данные ПрО, необходимы для сохранения непротиворечивости данных и обеспечения адекватного (достоверного) отображения ПрО в БД. Вид модели данных и используемые в ней типы структур данных отражает концепцию организации и обработки данных, используемую в СУБД, поддерживающей модель, или в языке программирования, на котором создается прикладная программа обработки данных.

2.2. МОДЕЛИ ДАННЫХ

МОДЕЛЬ ДАННЫХ - это формальный аппарат для описания совокупности взаимосвязанных структур ПрО и операций над ними. К наиболее распространенным моделям относятся иерархическая, сетевая и реляционная.

ФАЙЛОВАЯ МОДЕЛЬ ДАННЫХ. В файловых системах в основном реализуется модель данных типа **ПЛОСКИЙ ФАЙЛ**. Основными типами структур данных данной модели являются поле, запись, файл.

Поле представляет элементарную единицу логической организации данных и соответствует отдельной, неделимой единице информации.

Запись - это совокупность полей, соответствующих совокупности логически связанных полей. Структура записи линейная и в ней отсутствуют групповые данные. Запись является основной структурой данных для обработки и для обмена данными между оперативной памятью и внешними накопителями данными.

Файл - это множество экземпляров записей с одинаковыми структурами и со значениями данных в отдельных полях. Каждый экземпляр записи однозначно идентифицируется уникальным ключом записи.

Ключи записей делятся на первичные (уникальные) и вторичные. Первичные ключи бывают простыми (состоят из одного поля) или составными (состоят из нескольких полей).

Вторичный ключ - это поле, значение которого может повторяться в нескольких записях файла, и в общем случае оно не является уникальным.

База данных в рамках файловых систем представляется в виде совокупности не связанных между собой файлов.

Для эффективного доступа к записям файлов по ключам используется индексирование. При индексировании создается дополнительный индексный файл, который состоит из упорядоченного перечня ключей файла данных. Для каждого значения ключа в индексном файле содержится указатель на соответствующую запись файла данных. С помощью указателей осуществляется прямой доступ к записям файла. Индексирование может проводиться как по первичному ключу, так и по вторичному ключу.

Логическая организация данных для файловой модели данных следующая. БД - это совокупность файлов. Отдельному файлу присваивается уникальное имя. Описание структуры записи включает перечень входящих в нее полей. Для каждого поля задается идентификатор, формат поля (тип хранимого данного, длина поля, точность и другие). Для ключевых полей указывается признак ключа.

СТРУКТУРЫ ДАННЫХ В МОДЕЛЯХ ДАННЫХ.

К типовым структурам данных при определении моделей данных можно отнести следующие: элемент данных, агрегат данных, запись, БД и другие.

ЭЛЕМЕНТ ДАННЫХ - это минимальная именованная структура данных.

АГРЕГАТ ДАННЫХ - это именованное множество элементов данных или других агрегатов данных или других агрегатов внутри записи. В агрегатах допускается множественный элемент, который содержит несколько значений элемента в одном экземпляре агрегата.

ЗАПИСЬ в общем случае является составным агрегатом, который не входит в состав других агрегатов. Она характеризуется структурной взаимосвязей ее элементов и агрегатов. Структура записи может иметь иерархический характер. Все множество экземпляров записи одинаковой структуры образует тип записи. Запись конкретного типа является объектом в модели данных.

СВЯЗИ МЕЖДУ ОБЪЕКТАМИ. Модель данных обычно включает несколько типов объектов (записей). Между объектами устанавливаются связи. Совокупность взаимосвязанных конкретных объектов модели ПрО образуют базу данных.

Связи между объектами модели (типами записей) определяются групповыми отношениями между их экземплярами.

Групповое отношение (набор) - это строго иерархическое отношение между объектами двух типов: главным объектом (главная запись набора) и подчиненным объектом (подчиненными записями набора).

Наибольшее распространение среди моделей данных, поддерживаемых СУБД, являются сетевые, иерархические и реляционные. Соответственно различают сетевые, иерархические и реляционные СУБД. Сетевые и иерархические СУБД получили наибольшее распространение на больших и мини ЭВМ. Основной вклад в разработку теории сетевых и, как частный случай, иерархических моделей данных, языков описания и манипулирования данными СУБД внесла КОДАСИЛ (CODASYL) - Ассоциация по языкам и системам обработки данных [2].

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ ориентирована на представление модели ПрО в виде древовидной (иерархической) структуры. В иерархических моделях любой объект может подчиняться только одному объекту. Непосредственный доступ к объекту по ключу возможен только к объекту самого высокого уровня, который не подчинен другим объектам БД. К другим объектам доступ осуществляется по связям от объекта на вершине модели. Структура объекта в иерархических моделях может быть иерархической или плоской.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ использует модель представления данных в виде произвольного графа. В сетевых моделях любой объект может быть подчинен нескольким объектам. Непосредственный доступ по ключу может обеспечиваться к любому объекту независимо от уровня, на котором он находится в модели. Возможен также доступ по связям от любой точки доступа. Структура объекта в сетевых моделях чаще бывает линейной и реже имеет иерархическую структуру. Сетевые модели являются более универсальными (большинство ПрО имеет сетевой характер), что ограничивает использование иерархических моделей. Сетевые модели позволяют отображать иерархические взаимосвязи между объектами. Достоинством сетевых моделей является отсутствие дублирования данных в различных элементах модели. Технология работы с сетевыми моделями является удобной для пользователя, так как

доступ к данным не имеет ограничений и возможен непосредственно к объекту любого уровня. Допустимы всевозможные запросы к БД.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ. Концепция реляционной модели данных (РМД) была предложена Е.Ф. Коддом в 1970 г. Основная цель РМД - обеспечить независимость представления и описания данных от прикладных программ. В основе РМД лежит понятие отношения (англ. relation). Отношение удобно представляется в виде двумерной таблицы. Таблица понятна, обозрима и привычна для человека. Набор отношений (таблиц) может быть использован для хранения данных об объектах реального мира и моделирования связей между ними. Например, для хранения информации об объекте (сущности) "студент" можно использовать отношение СТУДЕНТ, в котором свойства объекта располагаются в столбцах таблицы (см.рис.2.1).

Таблица - это основной тип структуры данных РМД. Структура таблицы определяется совокупностью столбцов. В каждой строке таблицы содержится по одному значению в соответствующем столбце. В таблице не может быть двух одинаковых строк. Общее число строк не ограничено.

Столбцы отношения называют атрибутами и им присваиваются имена. Список имен атрибутов отношения называется схемой отношения. Схема нашего отношения СТУДЕНТ можно представить следующим образом:

СТУДЕНТ (Фамилия, Курс, Специальность)

Реляционная база данных (РБД) - это набор взаимосвязанных отношений. Отдельное отношение представляется в виде отдельного файла или таблицы.

Основные понятия РБД следующие:

- **АТРИБУТ** - это столбец таблицы, имеющий свое уникальное имя.
- **КОРТЕЖ** - это строка таблицы.
- **КАРДИНАЛЬНОЕ ЧИСЛО** - это количество кортежей в таблице,
- **СТЕПЕНЬ ТАБЛИЦЫ** - это количество атрибутов в таблице.
- **ПЕРВИЧНЫЙ КЛЮЧ** - это уникальный идентификатор таблицы, т.е. это столбец или такая совокупность столбцов, что в любой момент времени не существует в таблице двух строк, содержащих одинаковое значение в этом столбце или в комбинации столбцов.



Рис.2.1. Основные понятия реляционной модели данных

Предложенный подход Кодда позволяет применять к отношениям систему операций и обеспечивает вывод (или вычисление подобно арифметическим операциям) одних отношений из других. Это дает возможность делить информацию на хранимую и не хранимую (вычисляемую) части, и тем самым экономить память. При необходимости не хранимая часть информации вычисляется из хранимой информации. Совокупность нормализованных отношений, логически взаимосвязанных и отражающих некоторую ПрО, образует **РЕЛЯЦИОННУЮ БАЗУ ДАННЫХ**.

Для работы с данными в рамках РБД используются операции над строками (кортежами) - включение, удаление, обновление - и над отношениями (отношение рассматривается как единый объект). К основным операциями над отношениями относятся:

- операции над множествами - объединение, пересечение, разность (вычитание), декартово произведение, деление;
- специальные реляционные операции - проекция, соединение и выбор (селекция, ограничение).

Эффективность конкретной СУБД определяется наличием и удобством использования языковых средств выполнения этих операций. Языки для выполнения операций над отношениями делятся на:

- языки реляционной алгебры;
- языки реляционного исчисления.

ЯЗЫКИ РЕЛЯЦИОННОЙ АЛГЕБРЫ основаны на реляционной алгебре (алгебра Кодда, d-алгебра). Записывая последовательности операций над отношениями в соответствующем порядке, можно получить желаемый результат. Языки реляционной алгебры в основном являются процедурными языками программирования.

ЯЗЫКИ РЕЛЯЦИОННОГО ИСЧИСЛЕНИЯ основаны на классическом исчислении предикатов. Они предоставляют пользователю набор правил для записи "запросов" к БД, в которых содержится лишь информация о желаемом результате. На основании информации запроса СУБД формирует совокупность необходимых новых отношений, которые обеспечивают желаемый результат. Языки реляционного исчисления являются непроцедурными.

Практически большинство современных СУБД для персональных компьютеров поддерживают реляционную модель данных.

2.3. ТИПЫ ОТНОШЕНИЙ

Отношения реляционной БД в зависимости от содержания подразделяют на два класса: объектные и связанные отношения.

ОБЪЕКТНОЕ ОТНОШЕНИЕ хранит данные об объектах (экземплярах сущности). Приведенное выше отношение СТУДЕНТ является объектным отношением. В объектном отношении один из атрибутов (или несколько атрибутов) однозначно идентифицирует отдельный объект. Такой атрибут называют ключом отношения или первичным атрибутом. В отношении СТУДЕНТ на роль ключа претендует атрибут <Фамилия>. Для удобства ключ записывают в первом столбце таблицы. Остальные атрибуты функционально зависят от этого ключа.

Ключ может включать несколько атрибутов (составной ключ) или быть частью значения атрибута (частичный ключ). В объектном отношении не должно быть строк с одинаковыми ключами, т.е. не должно быть дублирования объектов. Это основное ограничение РМД для обеспечения целостности данных.

СВЯЗНОЕ ОТНОШЕНИЕ хранит ключи двух или более объектных отношений, т.е. по ключам устанавливаются связи между объектами отношений. Пусть в БД имеются объектные отношения СТУДЕНТ (Фамилия, Курс, Дисциплина) и ПРЕДМЕТ (Название, ЧислоСеместров) со следующими значениями атрибутов:

СТУДЕНТ

Фамилия	Курс	Дисциплина
Иванов	2	Программирование
Петров	3	Иностранный язык
Сидоров	1	Физика

ПРЕДМЕТ

Дисциплина	ЧислоСеместров
Программирование	2
Иностранный язык	6
Физика	4

В качестве примера рассмотрим связное отношение ИЗУЧАЕТ (Фамилия, Предмет), означающее, что студент изучает определенный предмет. Связное отношение ИЗУЧАЕТ имеет следующий вид:

ИЗУЧАЕТ

Фамилия	Дисциплина
Иванов	Программирование
Петров	Иностранный язык
Сидоров	Физика

Связное отношение кроме связываемых ключей может иметь и другие атрибуты, которые функционально зависят от этой связи. Примером может быть связное отношение РЕЗУЛЬТАТ (Фамилия, Дисциплина, Оценка):

РЕЗУЛЬТАТ

Фамилия	Дисциплина	Оценка
Иванов	Программирование	4
Петров	Иностранный язык	5
Сидоров	Физика	3

2.4. КЛЮЧИ

Ключевые атрибуты должны однозначно определять экземпляры сущностей. Ключи бывают следующих типов: первичные (простые, полностью составные, полусоставные); внешние. Ключи в связных отношениях называются внешними (или посторонними) ключами, поскольку являются первичными ключами других отношений.

Ключ называется простым, если он состоит из одного атрибута. Последний является атомарным, а его значение - уникальным.

СОСТАВНОЙ КЛЮЧ содержит два и более атрибутов, каждый из которых необходим для однозначной идентификации экземпляров сущности. Полезно различать два варианта составного ключа.

ПОЛНОСТЬЮ СОСТАВНОЙ КЛЮЧ содержит несколько атрибутов, между которыми существует связь типа М:N, одиночная связь типа М или связь отсутствует. Атрибуты, составляющие такой ключ, не зависят друг от друга. Ни один из этих атрибутов не является дополнительным признаком другого атрибута.

ПОЛУСОСТАВНОЙ КЛЮЧ содержит несколько атрибутов и построен с использованием связей типа М:1.

Реляционная модель накладывает на внешние ключи ограничение для обеспечения целостности данных, называемое ссылочной целостностью. Это означает, что каждому внешнему ключу должна соответствовать строка какого-либо объектного отношения. Без такого ограничения может случиться, что внешний ключ ссылается на объект, о котором ничего не известно. В реляционной БД на каждое отношение накладывается и другое ограничение - они должны быть нормализованы. Это означает, что каждый атрибут должен быть простым - содержать атомарные, неделимые значения. К примеру, приведенное ниже отношение СТУДЕНТ не нормализовано, поскольку содержит сложный атрибут <Спорт>:

СТУДЕНТ

Фамилия	Курс	Дисциплина	Спорт
Иванов	2	Программирование	Плавание, МС
Петров	3	Иностранный язык	Футбол, КМС
Сидоров	1	Физика	Шахматы, 1Р

Здесь ключом является атрибут <Фамилия>. Приведем это отношение к нормализованному виду, т.е. избавимся от сложного атрибута <Спорт>. В полученном отношении СТУДЕНТ ключ является составным, состоящим из атрибутов <Фамилия> и <ВидСпорта>.

СТУДЕНТ

Фамилия	ВидСпорта	Курс	Дисциплина	СпортивныйРазряд
Иванов	Плавание	2	Программирование	МС
Петров	Футбол	3	Иностранный язык	КМС
Сидоров	Шахматы	1	Физика	1Р

УСЛОВИЯ и ОГРАНИЧЕНИЯ, накладываемые на отношения реляционной моделью данных, которые позволяют таблицы считать отношениями:

1. Не может быть одинаковых первичных ключей, т.е. все строки (записи) таблицы должны быть уникальны.
2. Все строки таблицы должны иметь одну и ту же структуру, т.е. одно и то же количество атрибутов с соответственно совпадающими именами.
3. Имена столбцов таблицы должны быть различны, а значения столбцов должны быть однородными (однотипными).
4. Значения атрибутов должны быть атомарными, следовательно, отношения не могут иметь в качестве компонент другие отношения.
5. Должна соблюдаться ссылочная целостность для внешних ключей.
6. Порядок следования строк в таблице несуществен, так как влияет лишь на скорость доступа к строке.

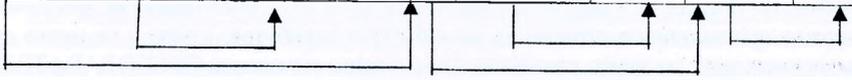
При выполнении операций обновления над отношениями БД могут возникать побочные эффекты, когда между атрибутами отношений существуют нежелательные функциональные зависимости. Устранение этих эффектов достигается нормализацией отношений БД, т.е. рациональной группировкой атрибутов в отношениях.

2.5. ТИПЫ ФУНКЦИОНАЛЬНЫХ ЗАВИСИМОСТЕЙ

Концепция функциональных зависимостей лежит в основе проектирования структур реляционных БД. Функциональная зависимость обладает набором формальных свойств, позволяющих формально и строго решить многие проблемы. В качестве примера рассмотрим отношение с составным ключом, состоящим из двух атрибутов <ЛичныйНомер> и <Дисциплина>:

ПРЕПОДАВАТЕЛЬ*ПРЕДМЕТ

Личный Номер	Дисцип- лина	Количес- твоЧасов	Фами- лия	Должность	Оклад	Кафед- ра	Теле- фон
201	Мини ЭВМ	36	Фролов	Доцент	380	ЭВМ	4-89
301	СУБД ПК	48	Бойко	Профессор	520	АСУ	4-89
401	Физика	52	Глазов	Ассистент	270	ТЭФ	4-12
401	Оптика	30	Глазов	Ассистент	270	ТЭФ	4-12



ПОНЯТИЕ ФУНКЦИОНАЛЬНОЙ ЗАВИСИМОСТИ. Пусть имеются два атрибута отношения R: A и B. Если в любой момент времени каждому значению A соответствует не более чем одно значение атрибута B, то B функционально зависит от A. Функциональная зависимость обозначается: $A \rightarrow B$. Например: *Должность \rightarrow Оклад, Кафедра \leftarrow Фамилия, Телефон \leftarrow Кафедра.*

Более общим является следующее определение функциональной зависимости. Пусть имеется отношение R и произвольное подмножество атрибутов X и Y данного отношения. Y функционально зависит от X тогда и только тогда, когда каждое значение множества X связано в точности с одним значением множества Y. Из выше приведенного примера следует:

(ЛичныйНомер, Дисциплина) \rightarrow (КоличествоЧасов, Фамилия, Должность, Оклад, Кафедра, Телефон).

Степень зависимости описательных (неключевых) атрибутов отношения от ключа может быть различной.

ЧАСТИЧНАЯ ЗАВИСИМОСТЬ. Если описательный атрибут зависит только от части ключа, то это частичная зависимость. В нашем примере описательный атрибут <КоличествоЧасов> зависит от части ключа, т.е. только от атрибута <Дисциплина>.

ПОЛНАЯ ФУНКЦИОНАЛЬНАЯ ЗАВИСИМОСТЬ. Если описательный атрибут зависит от всего составного ключа и не находится в частичной зависимости от его частей, то это полная функциональная зависимость от составного ключа. В приведенном примере нет атрибутов, находящихся в полной функциональной зависимости от составного ключа.

ТРАНЗИТИВНАЯ ЗАВИСИМОСТЬ. Если для атрибутов А, В, С выполняются условия $A \rightarrow B$ и $B \rightarrow C$, то С транзитивно зависит от А. В рассматриваемом примере транзитивная зависимость следующая:

Фамилия \rightarrow Кафедра \rightarrow Телефон

Множество всех функциональных зависимостей, которые задаются данным множеством функциональных зависимостей S, называется замыканием S и обозначим S^+ . Для вычисления S^+ на основе S используются правила вывода функциональных зависимостей на основе заданных (эти правила также называются аксиомами Армстронга). Пусть в перечисленных ниже правилах X, Y, Z — произвольные подмножества множества атрибутов заданного отношения R, а символическая запись XY означает объединение X и Y.

1. РЕФЛЕКСИВНОСТЬ: если Y является подмножеством X, то $X \rightarrow Y$.

2. ДОПОЛНЕНИЕ: если $X \rightarrow Y$, то $XZ \rightarrow YZ$.

3. ТРАНЗИТИВНОСТЬ: если $X \rightarrow Y$ и $Y \rightarrow Z$, то $X \rightarrow Z$.

Каждое из этих правил может быть непосредственно доказано на основе определения функциональной зависимости. В отношениях между атрибутами может существовать многозначная зависимость.

МНОГОЗНАЧНАЯ ЗАВИСИМОСТЬ. В отношении R атрибут В многозначно зависит от А ($A \twoheadrightarrow B$), если каждому значению А соответствует множество значений В, никак не связанных с другими атрибутами из R. Многозначная зависимость возможна при наличии в отношении хотя бы трех атрибутов: ключа и не менее двух независимых друг от друга атрибутов. Рассмотрим отношение ПРЕПОДАВАТЕЛЬ с ключом <Фамилия>:

ПРЕПОДАВАТЕЛЬ

Фамилия	Группа	Дисциплина
Серов	АС31	СУБД
Серов	АС31	ОС
Серов	Э42	СУБД
Серов	Э42	ОС
Иванов	АС51	СУБД

Между преподавателем и группами студентов существует связь типа "один-ко-многим" (1:M), поскольку преподаватель может читать лекции в одной и более группах, однако каждой группе соответствует один преподаватель:

1 : M

Фамилия \rightarrow Группа

Между преподавателем и дисциплинами существует связь типа "многие-ко-многим" (M:N), поскольку преподаватель может вести одну и более дисциплин, и наоборот, одну дисциплину могут вести несколько преподавателей:

M : N

Фамилия \leftrightarrow Дисциплина

В рассматриваемом отношении существуют независимые многозначные зависимости Фамилия- \twoheadrightarrow Группа и Фамилия- \twoheadrightarrow Дисциплина, так как значения многозначных атрибутов <Группа> и <Дисциплина> никак не связаны между собой и возможно изменение их значений в любой строке отношения.

На практике могут встречаться самые необычные случаи функциональных зависимостей, поиск или определение которых может оказаться очень сложным.

Кроме того, некоторые функциональные зависимости могут изменяться со временем. Например, в рассматриваемом отношении ПРЕПОДАВАТЕЛЬ преподаватель закреплен за одной кафедрой. Но со временем он может начать работать параллельно и на другой кафедре. Тогда ему уже нельзя будет однозначно определить атрибут <Кафедра>.

2.6.МЕТОДИКА ПОСТРОЕНИЕ РЕЛЯЦИОННОЙ СТРУКТУРЫ БД

Методика построения реляционной структуры БД на основе КМ ПрО представляет собой совокупность следующих действий:

1. Определение типов связей между атрибутами отдельной сущности (1:1, 1:M, N:M).
2. Уточнение набора ключевых атрибутов для каждой сущности.
3. Дополнение сущностей описаниями связей между сущностями.
4. Упрощение ключевых атрибутов сущности.
5. Представление сущности в виде реляционной таблицы.

Перечисленные выше действия необходимо выполнить для всех сущностей концептуальной модели ПрО.

ОПРЕДЕЛЕНИЕ ТИПОВ СВЯЗЕЙ МЕЖДУ АТТРИБУТАМИ СУЩНОСТИ.

Определение связей между атрибутами в рамках отдельной сущности реализуется аналогично установлению связей между отдельными сущностями. Данное действие является очень важным, так как от его результатов зависит успех процесса нормализации отношений. От точности определения типов связей зависит процесс уточнения ключей сущностей и результаты процесса построения нормальных форм.

УТОЧНЕНИЕ НАБОРА КЛЮЧЕВЫХ АТТРИБУТОВ СУЩНОСТИ.

Существуют следующие правила определения ключевых атрибутов сущности на основе анализа типов связей между ее атрибутами:

- в качестве ключевых атрибутов выбираются те атрибуты от которых направлена (выходит) к другим атрибутам сущности хотя бы одна единичная связь (типа 1).
- Каждый такой атрибут определяется как ключ.
- если между двумя атрибутами сущности нет единичной связи, а задана связь типа M:N или связь типа 1:M, то из указанных двух атрибутов строится составной ключ.

ДОПОЛНЕНИЕ СУЩНОСТЕЙ ОПИСАНИЯМИ СВЯЗЕЙ МЕЖДУ СУЩНОСТЯМИ.

Связи между сущностями в РМД представляются с помощью ключей. Эти ключи необходимо добавить в те таблицы, для которых выполняются следующие условия:

- если от j-ой сущности выходит единичная связь к i-ой сущности, то в j-ю сущность добавляются ключи i-ой сущности;
- если между j-ой и i-ой сущностями существует связь типа M:N, то эти сущности либо объединяются в единую сущность, либо дополняются ключами соседней сущности (к j-ой сущности добавляются ключи i-ой сущности, а к i-ой сущности ключи j-ой сущности).

УПРОЩЕНИЕ КЛЮЧЕВЫХ АТТРИБУТОВ СУЩНОСТИ. Упрощение ключевых атрибутов сущностей рекомендуется применять в тех случаях, когда ключевые атрибуты используются для описания связей (т.е. они дополняются к ключевым атрибутам других сущностей) и занимают много места. В этих случаях, в сущность включается еще один дополнительный ключ. Этот дополнительный ключ представляется в виде числовой переменной и является ключом к тому ключу, который необходимо упростить.

ПРЕДСТАВЛЕНИЕ КМ В ВИДЕ НАБОРА РЕЛЯЦИОННЫХ ТАБЛИЦ. Результаты, полученные после выполнения перечисленных выше действий, должны быть представлены в виде реляционных таблиц. Каждая таблица представляет собой описание (одной или нескольких) сущности КМ Про и ее связей с другими сущностями. Описание этих таблицы должно содержать следующую информацию:

- название таблицы, определяющее сущность;
 - идентификаторы ключевых и описательных атрибутов сущности;
 - изображение связей между атрибутами внутри отдельной сущности
 - изображений связей между отдельными сущностями (таблицами).
- Первичный вариант покрытия КМ Про реляционными таблицами можно сделать, представив каждую сущность КМ в виде отдельной реляционной таблицы и дополнив ее описанием связей с другими сущностями.

3. НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ В БАЗЕ ДАННЫХ

Отношения реляционной БД содержат как структурную, так и семантическую (смысловую) информацию. Структурная информация задается схемой отношения, а семантическая выражается функциональными связями между атрибутами, известными и учитываемыми в схеме отношения.

Состав атрибутов отдельного отношения БД должен удовлетворять двум основным требованиям:

- между атрибутами отношения не должно быть нежелательных функциональных зависимостей;
- группировка атрибутов должна обеспечивать минимальное дублирование данных, обеспечивать их обработку и обновление без различного рода аномалий.

Удовлетворение этих требований достигается нормализацией отношений БД.

НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ - это пошаговый обратимый процесс декомпозиции (разложения) исходных отношений БД на другие, более мелкие и простые отношения. При этом устанавливаются (выясняются) все возможные функциональные зависимости.

Аппарат нормализации отношений был разработан Е.Ф.Коддом. Этот аппарат определяет различные нормальные формы (НФ). Каждая НФ ограничивает конкретный тип функциональных зависимостей отношения. Кодд выделил четыре НФ (сокращенные названия: 1НФ, 2НФ, 3НФ, НФБК - нормальная форма Бойса-Кодда). При наличии многозначных зависимостей между атрибутами используются 4НФ и 5НФ. Структура процесса нормализации представляется в виде последовательного построения НФ разного типа следующим образом:

1НФ -> 2НФ -> 3НФ -> НФБК -> 4НФ -> 5НФ

3.1. ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА

Отношение называется нормализованным или в 1НФ, если каждый из его атрибутов является атомарным. Это значит, что на пересечении любого столбца и строки отношения может быть только одно значение, а существование групп значений не допускается. Преобразование КМ Про в БД, находящуюся в 1НФ реализуется следующим образом:

1. Если все атрибуты всех таблиц атомарны, то КМ сразу находится в 1НФ и никаких действий в этом случае предпринимать не надо.
2. Если в таблицах существуют не атомарные атрибуты, то необходимо:
 - либо вернуться на этап концептуального проектирования структуры БД и изменить фрагмент КМ (например, изменить КМ так, чтобы неатомарные атрибуты были выделены в отдельную сущность);
 - либо преобразовать не атомарные атрибуты таблицы в атомарные. Полученные новые атрибуты остаются в таблице либо выделяются в отдельную таблицу.

В приведенном ниже примере отношение не находится в 1НФ, так как атрибут «СпортивныйРазряд» не атомарен. Он содержит два атрибута.

СТУДЕНТ*СПОРТ

Фамилия	Курс	СпортивныйРазряд
Иванов	2	Футбол, мастер спорта
Петров	3	Бег, кандидат в мастера спорта
Сидоров	1	Прыжки, первый разряд

Преобразование таких таблиц можно вести путем разделения не атомарных атрибутов на совокупность атомарных в рамках этой таблицы. В данном примере не атомарный атрибут <СпортивныйРазряд> можно преобразовать в совокупность атомарных атрибутов – <ВидСпорта> и <Разряд>. Преобразованная таблица будет иметь следующий вид:

СТУДЕНТ*СПОРТ

Фамилия	Курс	ВидСпорта	Разряд
Иванов	2	Футбол	Мастер спорта
Петров	3	Бег	Кандидат в мастера спорта
Сидоров	1	Прыжки	Первый разряд

Второй вариант преобразования таких таблиц – это деление таблицы на две. Первая таблица получается путем удаления из исходной таблицы неатомарного атрибута, а вторая - включает неатомарный атрибут (его в дальнейшем необходимо преобразовать в совокупность атомарных атрибутов) и ключ исходной таблицы, из которой этот атрибут выделен.

Каждая тип НФ ограничивает определенный тип функциональной зависимости и устраняет соответствующие аномалии при выполнении операций над отношениями БД. Для иллюстрации используем отношение ПРЕПОДАВАТЕЛЬ*ПРЕДМЕТ. Это отношение находится в 1НФ. В этом отношении можно отметить частичную функциональную зависимость:

- атрибута <Фамилия> от части составного ключа - <ИнициалыНомер>;
- атрибута <КоличествоЧасов> от части составного ключа - <Дисциплина>.

Такая частичная зависимость приводит к следующим аномалиям:

1. Имеет место дублирование данных о преподавателе, поскольку преподаватель может читать несколько дисциплин.
2. Существует проблема контроля избыточности данных, так как изменение оклада влечет за собой необходимость поиска и изменения значений окладов во всех кортежах с данным преподавателем.
3. Возникает проблема с преподавателями, которые в данное время не ведут дисциплины (например, находятся на повышении квалификации), и с лаборантами, которые вообще не ведут предметов. Преподавателя без дисциплины невозможно включить в отношение, и наоборот, если преподаватель увольняется и удаляется из отношения, то удаляется и дисциплина, хотя дисциплина должна проводиться (возможно другим преподавателем).

Таким образом, отношение в 1НФ требует дальнейших преобразований.

3.2. ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

Отношение находится во 2НФ, если оно находится в 1НФ, и каждый описательный атрибут функционально полно зависит от составного ключа. Если ключ отношения содержит один атрибут, то это отношение задано во 2НФ. Отношение не задано во 2НФ, если существуют описательные атрибуты, не имеющие полной функциональной зависимости от ключа. Чтобы устранить неполную функциональную зависимость описательных атрибутов от ключа, необходимо разбить таблицу на части. Каждая из полученных частей должна представлять собой отдельную таблицу, в которой содержатся только функционально зависимые атрибуты. Этот процесс продолжается до тех пор, пока не будут получены таблицы состоящие из функционально зависимых описательных атрибутов от ключа.

Устранение частичной функциональной зависимости реализуется путем разложения отношения на другие отношения следующим образом:

- построить проекцию отношения без атрибутов, которые находятся в частичной функциональной зависимости от составного ключа;
- построить проекцию на часть составного ключа и атрибутов, зависящих от этой части.

В нижеприведенном отношении присутствуют частично функциональные зависимости:

ПРЕПОДАВАТЕЛЬ*ПРЕДМЕТ

Личный Номер	Дисциплина	Количество Часов	Фамилия	Должность	Оклад	Кафедра	Телефон
201	Мини ЭВМ	36	Фролов	Доцент	380	ЭВМ	4-89
301	СУБД ПК	48	Бойко	Профессор	520	АСУ	4-89
401	Физика	52	Глазов	Ассистент	270	ТЭФ	4-12
401	Оптика	30	Глазов	Ассистент	270	ТЭФ	4-12



- атрибута <Фамилия> от части составного ключа - <ЛичныйНомер>;
- атрибута <КоличествоЧасов> от части составного ключа - <Дисциплина>.

В итоге преобразования получим два отношения ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ, находящиеся во 2НФ:

ПРЕДМЕТ

ЛичныйНомер	Дисциплина	КоличествоЧасов
201	Мини ЭВМ	36
301	СУБД ПК	48
401	Физика	52
401	Оптика	30

ПРЕПОДАВАТЕЛЬ

ЛичныйНомер	Фамилия	Должность	Оклад	Кафедра	Телефон
201	Фролов	Доцент	380	ЭВМ	4-89
301	Бойко	Профессор	520	АСУ	4-89
401	Глазов	Ассистент	270	ТЭФ	4-12
401	Глазов	Ассистент	270	ТЭФ	4-12

В полученном отношении ПРЕПОДАВАТЕЛЬ имеются следующие транзитивные функциональные зависимости, например:

ЛичныйНомер → Кафедра → Телефон

ЛичныйНомер → Должность → Оклад

Наличие транзитивных зависимостей порождает неудобства и аномалии следующего характера (на примере атрибута <Телефон>):

1. Имеет место дублирование информации о телефоне для преподавателей одной кафедры.
2. Существует проблема контроля избыточности, поскольку изменение номера телефона кафедры влечет за собой необходимость поиска и изменения номеров телефонов у всех преподавателей этой кафедры.
3. Нельзя включить данные о новой кафедре (название кафедры и номер телефона), если на данный момент еще отсутствуют преподаватели. И наоборот, при увольнении всех преподавателей с кафедры данные о ней нельзя сохранить.

Таким образом, для отношения во 2НФ могут потребоваться дальнейшие преобразования.

3.3. ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА

Отношение задано в 3НФ, если оно задано во 2НФ и каждый атрибут из этого отношения, не являющийся ключом, нетранзитивно зависит от каждого возможного ключа отношения. Для этого необходимо, чтобы отсутствовала функциональная транзитивная зависимость между описательными атрибутами отношения.

Для преобразования отношения из 2НФ в 3НФ необходимо ликвидировать транзитивную зависимость. Она реализуется путем деления такого отношения на составляющие отношения. В нашем примере отношение ПРЕПОДАВАТЕЛЬ (из

предыдущего параграфа) необходимо разделить на три отношения: ПРЕПОДАВАТЕЛЬ, ДОЛЖНОСТЬ и КАФЕДРА.

ПРЕПОДАВАТЕЛЬ

ЛичныйНомер	Фамилия	Должность	Кафедра
201	Фролов	Доцент	ЭВМ
202	Костин	Доцент	ЭВМ
301	Бойко	Профессор	АСУ
401	Глазов	Ассистент	ТЭФ

КАФЕДРА

Кафедра	Телефон
ЭВМ	4-89
АСУ	4-77
ТЭФ	4-22

ДОЛЖНОСТЬ

Должность	Оклад
Доцент	3 380
Профессор	3 920
Ассистент	2 270

ЗНФ освобождает от избыточности и аномалий выполнения операций включения, удаления и обновления, если отношение имеет один ключ и другие зависимости, в том числе многозначные, в нем отсутствуют. Если имеются другие зависимости, кроме зависимости от ключа, то ЗНФ не обеспечивает отсутствия аномалий операций. В этом случае применяют усиленную ЗНФ.

3.4. УСИЛЕННАЯ НОРМАЛЬНАЯ ФОРМА БОЙСА - КОДДА

Пусть имеется отношение КУРС*ПРОЕКТ (Преподаватель, Дисциплина, Студент), отражающее руководство преподавателей курсовыми проектами, которые выполняются студентами:

КУРС*ПРОЕКТ

Преподаватель	Дисциплина	Студент
ПР1	П1	СТ1
ПР2	П1	СТ2
ПР3	П1	СТ3
ПР1	П2	СТ4
ПР2	П2	СТ5
ПР2	П3	СТ6
ПР3	П3	СТ7

Курсовыми проектами могут руководить несколько преподавателей, и каждый студент закреплен за одним из них. Отдельный студент выполняет только один проект, а один и тот же проект могут выполнять несколько студентов, но у разных преподавателей.

На роль ключа в этом отношении претендуют два атрибута - <Преподаватель> и <Дисциплина>. Объединим эти атрибуты в составной ключ. Между атрибутами отношения существуют функциональные зависимости:

Преподаватель, Дисциплина → Студент (зависимость от ключа)

Студент → Дисциплина

Отношение находится в ЗНФ, поскольку в нем отсутствуют частичные и транзитивные функциональные зависимости описательных атрибутов от ключа. Однако в отношении присутствуют два факта (о руководстве преподавателем проектом и о выполнении студентом проекта) и наблюдается зависимость части <Дисциплина> составного ключа от описательных атрибутов <Студент>. Такая зависимость приводит к следующим аномалиям:

1. Существует проблема контроля непротиворечивости данных, так как замена студента требует просмотра всего отношения с целью поиска и изменения кортежей, содержащих данные о преподавателе этого студента.
2. Данные о студенте и его проекте не могут быть занесены в таблицу до тех пор, пока не назначен руководитель проекта. И наоборот, если необходимо удалить преподавателя (например, в связи с увольнением), то будут удалены данные и о руководимом им студенте.

Устранение этих аномалий достигается путем устранения функциональной зависимости части составного ключа от описательных атрибутов, т.е. зависимости <Студент> → <Дисциплина>. Для этого отношение КУРС_ПРОЕКТ разлагается на два отношения, которые будут находиться в НФБК.

Отношение находится в НФБК, если оно находится в ЗНФ и в нем отсутствуют зависимости ключей от описательных атрибутов:

РУКОВОДСТВО

Преподаватель	Дисциплина
ПР1	П1
ПР2	П1
ПР3	П1
ПР1	П2
ПР2	П2
ПР2	П3
ПР3	П3

ВЫПОЛНЕНИЕ

Студент	Дисциплина
СТ1	П1
СТ2	П1
СТ3	П1
СТ4	П2
СТ5	П2
СТ6	П3
СТ7	П3

Следует отметить, что соединение полученных отношений по атрибуту <Дисциплина> дает исходное отношение КУРС*ПРОЕКТ.

3.5. МНОГОЗНАЧНЫЕ ЗАВИСИМОСТИ

Пусть X , Y и Z являются произвольными подмножествами множества атрибутов отношения R . Тогда Y многозначно зависит от X , что символически выражается записью

$$X \twoheadrightarrow Y \text{ ("X многозначно определяет Y")},$$

тогда и только тогда, когда множество значений Y , соответствующее заданной паре (значение X , значение Z) отношения R , зависит только от X , но не зависит от Z . Так же, как и в случае с функциональными зависимостями, в данном случае интерес представляют не те многозначные зависимости, которые выполняются в некоторый момент времени, а те, которые выполняются всегда.

Отношение необходимо привести к 4НФ, если в нем присутствуют многозначные функциональные зависимости. Для иллюстрации рассмотрим отно-

шение ПРЕПОДАВАТЕЛЬ с независимыми многозначными зависимостями
Фамилия $\rightarrow\rightarrow$ Группа и Фамилия $\rightarrow\rightarrow$ Дисциплина:

ПРЕПОДАВАТЕЛЬ

Фамилия	Дисциплина	КоличествоЧасов	Группа
Фролов	ПО ЭВМ	36	Э17
Фролов	ПЭВМ	72	Э17
Фролов	СУБД ПК	48	АС1
Костин	Экономика	64	Э17
Фролов	СУБД ПК	48	Э17

Аномалия выполнения операций над данным отношением может быть следующая. При появлении новой группы у преподавателя в отношении необходимо добавить не один кортеж, а столько, сколько дисциплин преподает в этой группе преподаватель. Например, если преподаватель Фролов должен проводить занятия в группе АС2. Аналогичная ситуация возникает с включением в отношении новой дисциплины. Устранение аномалий достигается разложением исходного отношения на несколько отношений с многозначными зависимостями от одного и того же ключа (в данном примере таким ключом является атрибут <Фамилия>):

ПРЕПОДАВАТЕЛЬ*ГРУППЫ

Фамилия	Группа
Фролов	Э17
Фролов	АС1
Костин	Э17

ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНЫ

Фамилия	Дисциплина	КоличествоЧасов
Фролов	ПО ЭВМ	36
Фролов	ПЭВМ	72
Фролов	СУБД ПК	48
Костин	Экономика	64

Отношение находится в 4НФ, если оно находится в НФБК и в нем отсутствуют независимые многозначные зависимости, т.е. все независимые многозначные зависимости выделены (разнесены) в отдельные отношения с одним и тем же ключом.

Необходимо подчеркнуть важность описания формальной теории для нормализации некоторых ненормализованных отношений. Основным при этом правилом является разбиение ненормализованного отношения на независимые. Нетрудно показать, что для данного отношения $R\{X,Y,Z\}$ многозначная зависимость $X \rightarrow\rightarrow Y$ выполняется тогда и только тогда, когда также выполняется многозначная зависимость $X \rightarrow\rightarrow Z$. Таким образом, многозначные зависимости всегда образуют связанные пары, и потому их представляюг вместе в символическом в виде: $X \rightarrow\rightarrow Y|Z$. Для рассматриваемого примера такая запись будет иметь следующий вид:

Фамилия $\rightarrow\rightarrow$ Группа | Дисциплина

Многозначные зависимости являются обобщением функциональных зависимостей в том смысле, что всякая функциональная зависимость является многозначной.

В рассматриваемом примере описанная проблема возникает из-за того, что отношение ПРЕПОДАВАТЕЛЬ содержит многозначные зависимости, которые не являются функциональными. Следует отметить, что именно наличие таких функциональных зависимостей в данном примере требует вставлять два кортежа (для каждого отношения), когда необходимо добавить данные о новом преподавателе.

Проекция отношения ПРЕПОДАВАТЕЛЬ на два отношения ПРЕПОДАВАТЕЛЬ*ГРУППЫ и ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНЫ, которые не содержат многозначных зависимостей, представляют собой некоторое усовершенствование исходного отношения. Поэтому исходное отношение ПРЕПОДАВАТЕЛЬ целесообразно заменить предложенными двумя его проекциями.

ЧЕТВЕРТАЯ НОРМАЛЬНАЯ ФОРМА. Отношение R находится в четвертой нормальной форме (4НФ) тогда и только тогда, когда существуют такие подмножества X и Y атрибутов отношения R, что выполняется (нетривиальная) многозначная зависимость $X \twoheadrightarrow Y$. Тогда все атрибуты отношения R также функционально зависят от атрибута A.

В отношении R могут находиться только нетривиальные зависимости (функциональные или многозначные) вида $A \twoheadrightarrow B$ (т.е. некоторый атрибут B функционально зависит от потенциального ключа A).

Это можно также сформулировать в следующей эквивалентной форме: отношение R находится в 4НФ, если оно находится в НФБК и все многозначные зависимости отношения R фактически являются функциональными зависимостями "от потенциальных ключей". Функциональная зависимость является тривиальной тогда и только тогда, когда правая часть символической записи является подмножеством (не обязательно собственным подмножеством) левой части. Очевидно, что сократить размер множества функциональных зависимостей можно с помощью исключения тривиальных зависимостей, т.е. таких, которые не могут выполняться. Приведем следующий пример тривиальной функциональной зависимости (см.п.2):

(*Фамилия, Дисциплины*) \rightarrow *Фамилия*

Отношение ПРЕПОДАВАТЕЛЬ не находится в 4НФ, поскольку содержит многозначную зависимость, которая не является функциональной зависимостью "от потенциального ключа". Однако обе проекции, ПРЕПОДАВАТЕЛЬ*ГРУППЫ и ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНЫ, находятся в 4НФ, которая по сравнению с НФБК позволяет создать улучшенную структуру и заменить ею прежнюю.

Кроме того, 4НФ всегда может быть получена, т.е. любое отношение может быть подвергнуто декомпозиции без потерь на эквивалентный набор отношений в 4НФ. Однако (как показано в параграфе 3.6 на примере отношения PGS), такая декомпозиция не всегда оказывается полезной и нужной.

Нормализация отношений выполняется декомпозицией (разложением) их схем. Декомпозиция должна гарантировать обратимость, т.е. обеспечивать получение исходных отношений путем выполнения операций соединения над их проекциями.

Обратимость предполагает, что:

- отсутствуют потери кортежей;
- не появляются ранее отсутствовавшие кортежи;
- сохраняются функциональные зависимости.

Однако не всегда декомпозиции гарантируют обратимость. Отношение может быть восстановлено без потерь соединением его проекций, если оно удовлетворяет зависимости по соединению. При наличии в отношении более трех многозначных зависимостей требуются специальные меры для обеспечения зависимости по соединению его проекций. В отношениях со многими многозначными зависимостями 4НФ может не устранять избыточности, а вместе с тем и аномалий обновления. В этом случае необходимо применяют 5НФ.

3.6. ПЯТАЯ НОРМАЛЬНАЯ ФОРМА

Иногда в процессе нормализации выполнить без потерь декомпозицию отношения на две проекции нельзя. В этом случае такое отношение может быть подвергнуто декомпозиции на три или более проекции. Такое отношение называется "*n*-декомпозируемое отношение" для $n > 2$. Рассмотрим в качестве примера отношение PGS (см.рис.3.1), содержащее информацию о преподавателях (*p*), о группах (*g*) и об изучаемых дисциплинах (*s*). Один преподаватель может проводить занятия в нескольких группах по нескольким дисциплинам. Например, преподаватель p_1 преподает дисциплину s_1 в группах g_1 и g_2 . Для данного примера можно сделать заключение, что отношение PGS находится в 4НФ - отношение PSG не содержит нетривиальных функциональных и многозначных зависимостей и все атрибуты отношения являются ключевыми. Однако в отношении в 4НФ могут быть аномалии, которые связаны с возможностью повторения значений атрибутов в нескольких кортежах. Например, преподаватель может преподавать в нескольких группах и при его увольнении требуется поиск всех записей о конкретном преподавателе и их замены или удаления.

Введем необходимые определения для зависимости соединения.

ЗАВИСИМОСТЬ СОЕДИНЕНИЯ. Пусть *R* - это отношение с произвольными подмножествами X, Y, \dots, Z множества атрибутов. Отношение *R* удовлетворяет зависимости соединения $*(X, Y, \dots, Z)$ тогда и только тогда, когда оно равносильно соединению своих проекций с подмножествами атрибутов X, Y, \dots, Z .

ТЕОРЕМА ФЕЙГИНА. Пусть *X*, *Y* и *Z* являются множествами атрибутов отношения $R\{X, Y, Z\}$. Отношение *R* будет равно соединению его проекций $\{X, Y\}$ и $\{X, Z\}$ тогда и только тогда, когда для отношения *R* выполняется многозначная зависимость $X \twoheadrightarrow YZ$.

Исходя из теоремы и определения, можно заключить, что зависимость соединения является обобщением многозначной и функциональной зависимостей.

ПЯТАЯ НОРМАЛЬНАЯ ФОРМА. Отношение *R* находится в пятой нормальной форме (5НФ) или в проекционно-соединительной нормальной форме, тогда и только тогда, когда каждая зависимость соединения в отношении *R* подразумевается потенциальными ключами отношения *R*.

На рис.3.1. представлены три бинарные проекции PG, GS и SP отношения PGS, результат соединения проекций PG и GS по атрибуту *g*, а также этот результат, соединенный вместе с проекцией SG по комбинации атрибутов (*s, g*). В результате первого соединения получено исходное отношение PGS с одним дополнительным (и

лишним) кортежем, а в результате второго соединения этот лишний кортеж исключается. Отсюда ясно, что отношение PGS с зависимостью соединения $*(PG,GS,SP)$ может быть 3-декомпозируемым. При этом промежуточные результаты будут разными в каждом случае. Независимо от того, какая пара проекций будет выбрана для первого соединения, в итоге будет получен одинаковый результат.

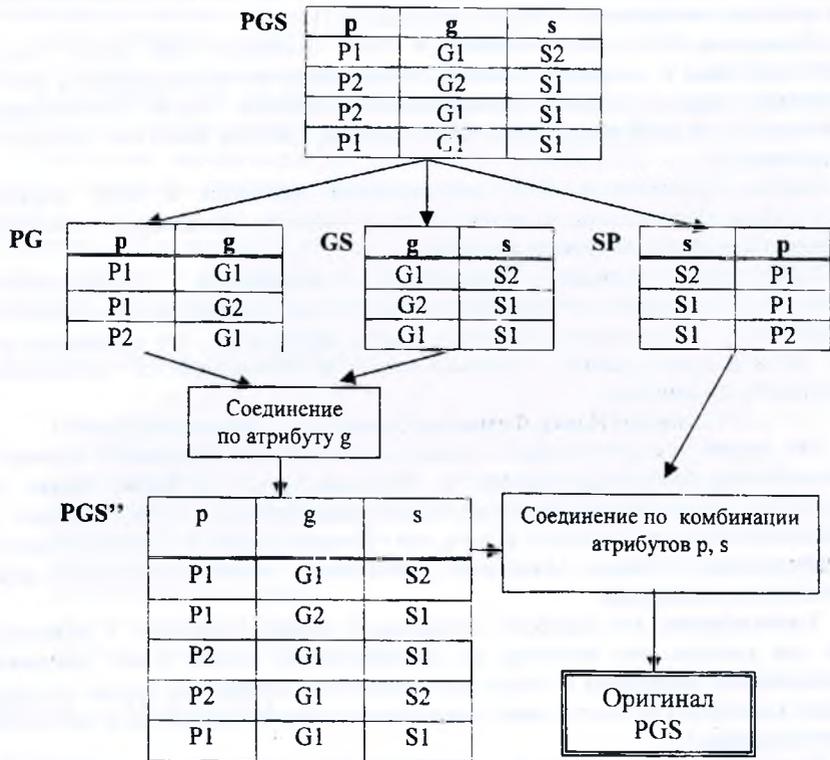


Рис.3.1.

Приведенный пример конкретного набора данных соответствует некоторому моменту времени. Для того, чтобы отношение не зависело от времени рассмотрим утверждение о том, что пара $(p1,g1)$ присутствует в отношении PG тогда и только тогда, когда тройка $(p1,g1,s2)$ присутствует в отношении PGS для некоторого $s2$. Это аналогично и для $(g1,s1)$ и $(p1,s1)$. Тогда приведенное утверждение можно переписать в виде ограничения, накладываемого на отношение PGS:

Если $(p1,g1,s2)$, $(p2,g1,s1)$ и $(p1,g2,s1)$ находятся в отношении PGS, тогда $(p1,g1,s1)$ также находится в отношении PGS.

Данное ограничение имеет циклическую структуру –

"если значение $p1$ связано с $g1$ и $g1$ связано с $s1$, а $s1$ связано с $p1$, то $p1, g1, s1$ должны находиться в одном кортеже".

Для иллюстрации циклического ограничения рассмотрим пример, в котором под этими ограничениями подразумевается, что если в отношении PGS:

1. Преподаватель Фролов (P1) преподает дисциплину СУБД (S1).
2. Дисциплина СУБД (S1) изучается в группе AC1 (G1).
3. Преподаватель Фролов (P1) является преподавателем в группе AC1 (G1).
4. Преподаватель Фролов (P1) ведет дисциплину СУБД (S1) в группе AC1 (G1).

Из взятых вместе условий (1), (2) и (3) не следует условие (4). Но в реальности отношение PGS содержит кортеж (P1,G1,S1), а это значит, что преподаватель Фролов (P1) преподает дисциплину СУБД (S1) в группе AC1 (G1).

Отношение PGS не находится в 5НФ, поскольку оно может быть 3-декомпозировано и возможность такой декомпозиции не подразумевается тем, что комбинация {p,g,s} является потенциальным ключом. После 3-декомпозиции проекции PG, GS и SP находятся в 5НФ, поскольку для них вовсе нет зависимостей соединения.

Любое отношение в 5НФ автоматически находится в 4НФ, поскольку многозначная зависимость является частным случаем зависимости соединения, подразумеваемой потенциальными ключами.

Рассмотрим понятие "зависимости соединения, подразумеваемой потенциальными ключами" на примере, в котором дано отношение преподавателей P с атрибутами <ЛичныйНомер, Фамилия, Адрес, Должность>. Это отношение имеет два потенциальных ключа <ЛичныйНомер> и <Фамилия> и удовлетворяет зависимости соединения -

*(<ЛичныйНомер,Фамилия,Должность>, <Фамилия,Адрес>).

Это значит, что отношение P равно соединению его проекций с атрибутами <ЛичныйНомер,Фамилия,Должность> и <Фамилия,Адрес>, а потому может быть подвергнуто декомпозиции без потерь на указанные проекции. Существование этой зависимости соединения следует из того, что <ЛичныйНомер> и <Фамилия> является потенциальными ключами. Аналогично, отношение P может удовлетворять другим зависимостям соединения.

Таким образом, для заданного отношения R можно утверждать: R находится в 5НФ при условии, что известны все потенциальные ключи и все зависимости соединения для отношения R. Найти все зависимости соединения совсем не просто. Однако, как следует из опыта, такие отношения достаточно экзотичны и чрезвычайно редки на практике.

Исходя из определения 5НФ, она является окончательной нормальной формой по отношению к операциям проекции и соединению. Таким образом, гарантируется, что отношение в 5НФ не содержит аномалий, которые могут быть исключены разбиением на проекции.

Конечно, это не значит, что данное отношение свободно от всех возможных аномалий. Это всего лишь означает, что оно свободно от всех аномалий, которые могут быть исключены с помощью разбиения на проекции.

3.7. ДРУГИЕ НОРМАЛЬНЫЕ ФОРМЫ

Теория нормализации (теория зависимостей) и связанные с ней вопросы развилась в значительную область знаний с несколькими различными (хотя и не пересекающимися) направлениями. В качестве примера приведено описание другой нормальной формы.

ДОМЕННО-КЛЮЧЕВАЯ НОРМАЛЬНАЯ ФОРМА - ДКНФ. Она была предложена Фейгиным. В отличие от рассмотренных выше нормальных форм, она не определяется на основе функциональных, многозначных зависимостей или зависимостей соединения. Вместо этого утверждается, что отношение R находится в ДКНФ тогда и только тогда, когда каждое ограничение, наложенное на отношение R, является логическим следствием ограничений доменов и ограничений ключей, наложенных на отношение R.

ОГРАНИЧЕНИЕ ДОМЕНА - это ограничение на использование значений для данного атрибута только из некоторого предписанного домена.

ОГРАНИЧЕНИЕ КЛЮЧА - это ограничение на то, что некоторый атрибут (некоторая комбинация атрибутов) представляют собой потенциальный ключ.

Наложение ограничений на отношение, находящееся в ДКНФ, является концептуально очень простым, поскольку для этого достаточно задать ограничения домена и ключа, а все остальные ограничения будут приведены в действие автоматически. Под выражением "все остальные ограничения" подразумевается нечто большее, чем просто функциональные и многозначные зависимости или зависимости соединения. Фактически это обозначает предикат данного отношения.

ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ

БД	- база данных
КМ	- концептуальная модель
НФ	- нормальная форма
ПО	- программное обеспечение
ПП	- прикладная программа
ПрО	- предметная область
РБД	- реляционная база данных
СОД	- система обработки данных
СУБД	- система управления базами данных

СПИСОК ЛИТЕРАТУРЫ

1. Т.А.Куправа. Создание и программирование баз данных средствами СУБД.- М.: Мир, 1991.-110с.
2. Тиори Г., Фрай Д. Проектирование структур баз данных: Пер.с англ.- М.: Мир, 1985. В 2-х кн.-287с.,-320с.
3. Дейт К.Дж. Введение в системы баз данных: Пер. с англ. – 6-е изд.- К.:Диалектика, 1998.-784с.
4. Мартин Дж. Организация баз данных в вычислительных системах: Пер. с англ. - М.: Мир, 1980.-662с.
5. Атре Ш. Структурный подход к организации баз данных: Пер. с англ. -М.: Финансы и статистика, 1983.
6. Змитрович А.И. Базы данных: учеб. пособие для вузов.-Мн.: Университетское, 1991.
7. Хаббард Д. Автоматизированное проектирование баз данных: Пер. с англ. М.: Мир, 1984.-296с.

ПРИЛОЖЕНИЕ №1.

ПОСТРОЕНИЕ РЕЛЯЦИОННОЙ СТРУКТУРЫ БАЗЫ ДАННЫХ

1. ПОСТАНОВКА ЗАДАЧИ НА ЛАБОРАТОРНУЮ РАБОТУ

ДАНО:

Концептуальная модель предметной области.

ЦЕЛЬ РАБОТЫ:

1. Ознакомиться с основными компонентами процесса проектирования реализации структуры БД.
2. Ознакомиться с теоретическими основами построения реляционных структур БД, получить практические навыки построения реляционных структур баз данных.
3. На основе концептуальной модели предметной области создать реляционную структуру БД.

РЕЗУЛЬТАТ:

Отчет по лабораторной работе должен содержать: цель работы, описание созданной реляционной структуры БД (описание таблиц БД, графическое представление общей структуры БД), выводы по работе.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:

1. Ознакомление с теоретической частью работы.
2. Выполнение работы.
3. Оформление работы.
4. Защита работы.

2. ПОРЯДОК ВЫПОЛНЕНИЯ И ОФОРМЛЕНИЯ РАБОТЫ

Построение реляционной структуры БД на основе КМ ПрО реализуется в соответствии с методикой приведенной в п.2.6. Она представляет собой последовательное выполнение следующих действий:

1. Определение типов функциональных связей между атрибутами отдельной сущности.
2. Уточнение набора ключевых атрибутов для каждой сущности.
3. Дополнение сущностей описаниями связей между сущностями.
4. Упрощение ключевых атрибутов сущностей.
5. Представление отдельной сущности в виде реляционной таблицы.

Перечисленные выше действия выполняются для всех сущностей исходной концептуальной модели.

Созданная реляционная структура БД представляется в виде:

- описания структуры таблиц (отношений) БД;
- графического представления структуры БД.

Каждое отношение БД представляется в виде отдельной таблицы, которая именуется (имя отношения) и состоит из описания атрибутов отношения. Для каждого атрибута определяется следующая информация:

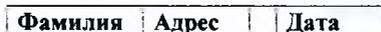
- тип атрибута (например, К - ключевой, О - описательный);
- имя атрибута;
- назначение атрибута;
- формат атрибута (предполагаемый).

Например, описание структуры отношения КЛИЕНТ можно представить следующим образом:

КЛИЕНТ

Тип атрибута	Имя атрибута	Назначение атрибута	Форма представления
К	Фамилия	Фамилия клиента	Символьный
О	Дата	Дата рождения	Символьный
К	Адрес	Место рождения	Символьный

Графическое представление структуры БД изображается в виде графа. Отдельная вершина графа представляет собой строку таблицы, которая содержит только имена ключевых и описательных атрибутов этой таблицы. Ключевые атрибуты располагаются в левой части строки, а описательные – в правой. Эти группы атрибутов разделяются двумя вертикальными линиями. Например:



Функциональные связи между вершинами графа изображаются следующим образом:

- - функциональная связь типа «1:1»;
- - функциональная связь типа «1:N»;
- ←→ - функциональная связь типа «N:M».

Эти связи соединяют между собой ключи разных таблиц, т.е. вершин.

3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Основные фазы и этапы жизненного цикла БД, их назначение.
2. Основные компоненты процесса проектирования реализации БД.
3. Что такое модель данных? Основные типы моделей данных.
4. Основные понятия реляционной модели данных.
5. Типы отношений и их назначение.
6. Определение ключа отношения, основные типы ключей.
7. Типы функциональных зависимостей, их определение.
8. Методика преобразования концептуальной модели в реляционную БД.

ПРИЛОЖЕНИЕ №2.

НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ В БАЗЕ ДАННЫХ

1. ПОСТАНОВКА ЗАДАЧИ НА ЛАБОРАТОРНУЮ РАБОТУ

ДАНО:

Реляционная структура БД (результат работы из приложения 1).

ЦЕЛЬ РАБОТЫ:

1. Ознакомиться с теоретическими основами теории нормализации отношений для реляционных структур БД, получить практические навыки нормализации отношений.
2. Провести нормализацию отношений для заданной реляционной структуры БД.

РЕЗУЛЬТАТ:

Отчет по работе должен содержать: цель работы, описание нормализованной реляционной структуры БД (описание нормализованных таблиц БД, графическое представление нормализованной структуры БД), выводы по работе.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:

1. Ознакомление с теоретической частью работы.
2. Выполнение работы.
3. Оформление работы.
4. Защита работы.

2. ПОРЯДОК ВЫПОЛНЕНИЯ И ОФОРМЛЕНИЯ РАБОТЫ

Исходной информацией для нормализации структуры БД являются результаты преобразования концептуальной модели ПрО в реляционную структуру БД (приложение 1). Они представляются в виде описания структуры таблиц и схемы взаимосвязей между таблицами БД.

Методика построения нормализованной структуры БД реляционного типа включает последовательное выполнение следующих действий:

- построение первой нормальной формы;
- построение второй нормальной формы (2НФ);
- построение третьей нормальной формы (3НФ);
- построение усиленной третьей нормальной формы (НФБК);
- построение четвертой нормальной формы (4НФ);
- построение пятой нормальной формы (5НФ).

Описание созданной реляционной структуры БД представляется в виде: описания таблиц (отношений) БД и графического представления структуры БД (схемы взаимосвязей между таблицами).

Описание отдельного отношения нормализованной структуры БД представляется в виде отдельной таблицы, структура которой приведена в

приложении I. В рамках данной работы описываются только новые таблицы, которые появились в БД в процессе нормализации или модифицированные.

Графическое представление структуры БД выполняется только в том случае, если по отношению к структуре БД полученной в предыдущей лабораторной работе появились новые таблицы.

3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое нормализация отношений?
2. Основные положения теории нормализации отношений в БД.
3. Методика построения нормализованной реляционной БД.
4. Функциональная зависимость, полная и неполная функциональная зависимость.
5. Транзитивная и многозначная зависимость.
6. Понятие зависимости соединения.
7. Построение 1НФ. Аномалии отношений находящихся в 1НФ.
8. Построение 2НФ. Аномалии отношений находящихся во 2НФ.
9. Построение 3НФ. Аномалии отношений находящихся в 3НФ.
10. Построение НФБК. Аномалии отношений находящихся в НФБК.
11. Построение 4НФ. Аномалии отношений находящихся в 4НФ.
12. Теорема Фейгина.
13. Построение 5НФ.

УЧЕБНОЕ ИЗДАНИЕ

Составители: Виктор Григорьевич Брич,
Владимир Иванович Хвещук

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по дисциплине
«БАЗЫ И БАНКИ ДАННЫХ»
Часть 3. Проектирование реляционных баз данных
для студентов специальности
«Автоматизированные системы обработки информации»

Ответственный за выпуск: Хвещук В.И.

Редактор: Строкач Т.В.

Подписано к печати 26.02.2001 г. Офсетная печать. Бумага писч. Формат 60x84 1/16. Заказ № 204. Усл. печ. л. 1.86. Уч. изд. л. 2.0. Тираж 150 экз. Отпечатано на ризографе Брестского государственного технического университета. 224017, Брест, Московская, 267.