

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра интеллектуальных информационных технологий

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по дисциплине
«ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ»
ПРОЕКТИРОВАНИЕ ЛОГИЧЕСКОЙ
МОДЕЛИ БАЗЫ ДАННЫХ

для студентов специальности 53 01 02 01
«Автоматизированные системы обработки информации»

Брест 2007

В лабораторном практикуме рассмотрен процесс построения логической модели базы данных для заданного объекта автоматизации. Приведены основные положения методики проектирования структур баз данных, дан обзор основных этапов процедуры проектирования логической модели базы данных. Определены основные положения и методика проектирования логической модели базы данных. Рассмотрены примеры преобразования концептуальной модели базы данных в логическую модель, приведены основные положения теории нормализации отношений базы данных, рассмотрены вопросы проверки выполнимости задач пользователей для спроектированной базы данных, а также требования поддержки целостности данных. Даны методические указания для выполнения лабораторных работ по данной тематике.

Лабораторный практикум предназначен для использования студентами специальности «Автоматизированные системы обработки информации» в ходе выполнения лабораторных и самостоятельных работ по дисциплине «Проектирование баз данных». Лабораторный практикум можно использовать при выполнении курсовых проектов (работ) по данной тематике.

Рис. 31, табл. 1, список лит. 9 назв.

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

ОГЛАВЛЕНИЕ

1. ПОСТАНОВКА ЗАДАЧ НА ЛАБОРАТОРНУЮ РАБОТУ «ПОСТРОЕНИЕ ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ РЕЛЯЦИОННОГО ТИПА».....	4
2. ОБЩИЕ ПОЛОЖЕНИЯ О РЕЛЯЦИОННОЙ МОДЕЛИ БАЗЫ ДАННЫХ	5
2.1. ТИПЫ ОТНОШЕНИЙ.....	6
2.2. ЦЕЛОСТНОСТЬ БАЗЫ ДАННЫХ.....	7
2.3. ТИПЫ ФУНКЦИОНАЛЬНЫХ ЗАВИСИМОСТЕЙ.....	8
2.4. ОБЩИЕ СВЕДЕНИЯ О НОРМАЛИЗАЦИИ ОТНОШЕНИЙ В РБД.....	9
2.5. МЕТОДИКА ПОСТРОЕНИЯ БД РЕЛЯЦИОННОГО ТИПА.....	10
3. ПОСТРОЕНИЕ ЛОГИЧЕСКОЙ МОДЕЛИ БД РЕЛЯЦИОННОГО ТИПА.....	11
3.1. УДАЛЕНИЕ ИЗ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ НЕЖЕЛАТЕЛЬНЫХ ЭЛЕМЕНТОВ.....	11
3.2. ОПРЕДЕЛЕНИЕ НАБОРА ОТНОШЕНИЙ ДЛЯ БД.....	15
3.3. ОПРЕДЕЛЕНИЕ СВЯЗЕЙ МЕЖДУ ОТНОШЕНИЯМИ БД.....	17
3.4. ДОКУМЕНТИРОВАНИЕ РЕЗУЛЬТАТОВ ПОСТРОЕНИЯ ЛОГИЧЕСКОЙ МОДЕЛИ БД.....	20
4. ПОСТАНОВКА ЗАДАЧ НА ЛАБОРАТОРНУЮ РАБОТУ «ПРОВЕРКА СТРУКТУРЫ БД».....	20
5. ПРОВЕРКА СТРУКТУРЫ ЛОГИЧЕСКОЙ МОДЕЛИ БД.....	21
5.1. ПРОВЕРКА ЛОГИЧЕСКОЙ МОДЕЛИ С ПОМОЩЬЮ ПРАВИЛ НОРМАЛИЗАЦИИ.....	21
5.1.1. ПОСТРОЕНИЕ ПЕРВОЙ НОРМАЛЬНОЙ ФОРМЫ.....	22
5.1.2. ПОСТРОЕНИЕ ВТОРОЙ НОРМАЛЬНОЙ ФОРМЫ.....	24
5.1.3. ПОСТРОЕНИЕ ТРЕТЬЕЙ НОРМАЛЬНОЙ ФОРМЫ.....	25
5.1.4. ПОСТРОЕНИЕ УСИЛЕННОЙ НОРМАЛЬНОЙ ФОРМЫ БОЙСА – КОДДА.....	26
5.2. ПРОВЕРКА ВЫПОЛНИМОСТИ ЗАДАЧ ПОЛЬЗОВАТЕЛЕЙ.....	27
5.3. ПРОВЕРКА ТРЕБОВАНИЙ ПОДДЕРЖКИ ЦЕЛОСТНОСТИ ДАННЫХ.....	28
5.4. УТОЧНЕНИЕ ДОКУМЕНТАЦИИ НА ПРОЕКТ БД.....	29
КОНТРОЛЬНЫЕ ВОПРОСЫ.....	30
ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ.....	30
СПИСОК ЛИТЕРАТУРЫ.....	31

1. ПОСТАНОВКА ЗАДАЧ НА ЛАБОРАТОРНУЮ РАБОТУ «ПОСТРОЕНИЕ ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ РЕЛЯЦИОННОГО ТИПА»

ТЕМА

Построение логической модели базы данных реляционного типа

ЦЕЛЬ:

Сформировать знания и умения по созданию логической модели (ЛМ) БД реляционного типа на основе концептуальной модели (КМ) предметной области (ПрО).

ЗАДАЧИ:

1. Удалить из концептуальной модели БД нежелательные элементы (см.п.3.1):
 - Удалить связи типа N:M и связи с атрибутами из КМ БД;
 - Удалить сложные, рекурсивные, множественные и избыточные связи из КМ БД.
2. Определить набор отношений для ЛМ БД на основе КМ БД (см.п.3.2).
3. Определить связи между отношениями ЛМ БД (см.п.3.3).
4. Документировать результаты проектирования ЛМ БД и уточнить документацию на КМ БД.

ИСХОДНЫЕ ДАННЫЕ:

Результаты выполнения лабораторных работ по созданию КМ БД [8] - перечень документации на КМ БД и графическое представление общей КМ ПрО в виде диаграммы «сущность-связь».

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:

1. Изучить теоретическую часть лабораторной работы (см.п.2).
2. Ответить на контрольные вопросы.
3. Изучить правила и примеры удаления из КМ нежелательных элементов (см.п.3.1), определить отношения (см.п.3.2) и связи между отношениями (см. п.3.3) для ЛМ БД.
4. Решить перечисленные выше задачи.
5. Оформить результаты выполнения лабораторной работы.
6. Защитить у преподавателя результаты выполнения лабораторной работы.

РЕЗУЛЬТАТ:

Отчет по работе должен содержать документацию на логическую модель БД в виде:

1. Документация на ЛМ БД в составе:
 - Описание структуры отношений ЛМ БД в табличном виде (см.п.3.4);
 - Графическое представление схемы связей между отношениями ЛМ БД;
2. Результаты уточнения документации на КМ БД:
 - Общая КМ ПрО в виде схемы диаграммы «сущность-связь»;
 - Описание содержания таблиц сущностей, атрибутов, ключей и доменов КМ БД.

Примечание: для автоматизации представления схемы связей между отношениями ЛМ БД, а также для документирования структуры и содержания отношений можно использовать возможности систем управления базами данных (СУБД).

2. ОБЩИЕ ПОЛОЖЕНИЯ О РЕЛЯЦИОННОЙ МОДЕЛИ БАЗЫ 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 используется для представления сущностей, а также для представления связей между сущностями. Отдельное отношение - это двумерная таблица. Таблица понятна, обозрима и привычна для человека. Например, для хранения информации о сущности "студент" можно использовать отношение **СТУДЕНТ**, которое представлено в табличном виде на рис.2.1.

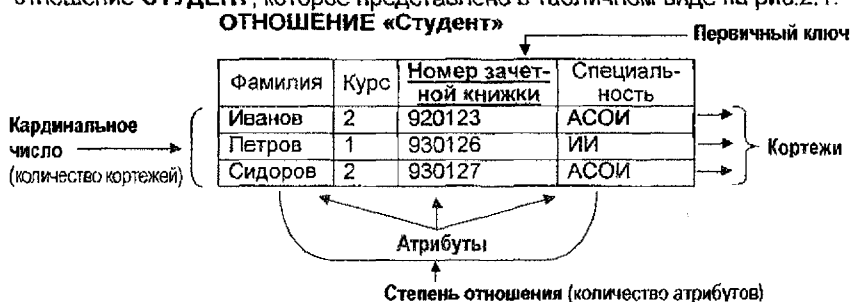


Рис.2.1. Пример отношения **СТУДЕНТ**

Общее число строк в отношении не ограничено. Столбцы отношения называют атрибутами и им присваиваются имена. Список имен атрибутов отношения называется схемой отношения. Схему отношения для отношения **СТУДЕНТ** в текстовой форме можно представить следующим образом:

СТУДЕНТ (Номер зачетной книжки, Фамилия, Курс, Специальность)

Основные понятия и определения РМД следующие:

Реляционная база данных (РБД) - это набор нормализованных отношений.

Атрибут - это поименованный столбец отношения, имеющий свое уникальное имя. Например, атрибуты **Фамилия**, **Курс**, **Номер зачетной книжки** и **Специальность** (см. рис.2.1).

Домен - это набор допустимых значений одного или нескольких атрибутов. Например, для атрибута **Курс** это могут быть значения 1, 2, 3, 4 и 5.

Кортеж - это отдельная строка отношения.

Кардинальное число - это количество кортежей в отношении. Для отношения на рис.2.1. кардинальное число равно 3.

Степень таблицы - это количество атрибутов в отношении. Для отношения на рис.2.1. степень отношения равно 4.

Ключ - это столбец (атрибут) или такая совокупность столбцов (атрибутов), которые однозначно и единственным образом могут идентифицировать строку (кортеж) отношения. Эти атрибуты называются ключевыми и должны однозначно определять экземпляры сущностей, определяемых отношением.

Потенциальный ключ - это множество атрибутов отношения является потенциальным ключом отношения тогда и только тогда, когда удовлетворяются два независимых от времени условия:

Уникальность - в произвольный заданный момент времени никакие два различных кортежа отношения не имеют одного и того же значения ключа;

Минимальность - ни один из атрибутов, входящих в ключ, не может быть исключен из ключа без нарушения уникальности.

Альтернативный ключ – это потенциальный ключ, который не выбран в качестве первичного ключа.

Ключи бывают следующих типов: **первичные** (простые, составные); **внешние**.

Первичный ключ (суперключ) – это потенциальный ключ, который выбран для уникальной идентификации кортежей внутри отношения.

Простой ключ состоит из одного атрибута.

Составной ключ содержит два и более атрибутов, которые в совокупности однозначно идентифицируют экземпляры отношения.

Внешний ключ – это атрибут или множество атрибутов одного отношения, являющиеся потенциальным ключом другого отношения.

Отношение в РБД обладает следующими свойствами:

- Отношение имеет имя, которое отличается от имен всех других отношений;
- Каждая ячейка отношения содержит только атомарное (неделимое) значение;
- Каждый атрибут имеет уникальное имя;
- Значения атрибута берутся из одного и того же домена;
- Порядок следования атрибутов не имеет никакого значения;
- Каждый кортеж является уникальным, т.е. дубликатов кортежей не может быть;
- Порядок следования кортежей в отношении не имеет никакого значения.

2.1. ТИПЫ ОТНОШЕНИЙ

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

Для демонстрации типов отношений рассмотрим фрагмент КМ ПрО, представленный на рис.2.2. На этом рисунке изображены сущности – **ИЗУЧАЕТ**, **СТУДЕНТ** и **ДИСЦИПЛИНА**, связи между этими сущностями – **Кто изучает** и **Что изучает**. Для каждой сущности в виде сносок изображены атрибуты сущностей.

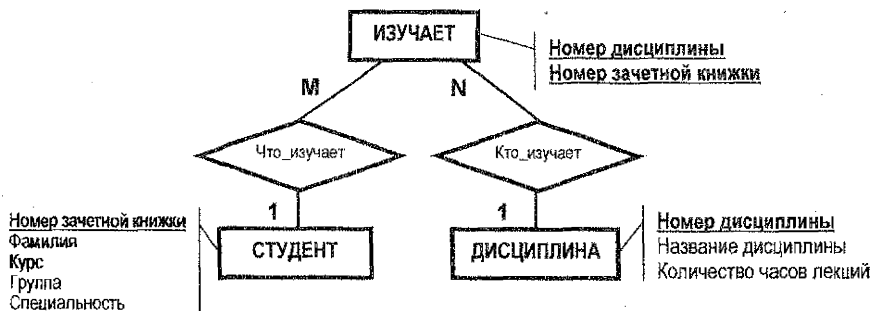


Рис.2.2. Фрагмент ПрО для представления объектных и связных отношений

Объектное отношение предназначено для хранения данных об объектах (экземплярах сущности) из ПрО. Пусть в БД имеются две сущности **СТУДЕНТ** с атрибутами - **Номер зачетной книжки**, **Фамилия**, **Курс**, **Группа**, **Специальность** и **ДИСЦИПЛИНА** с атрибутами - **Номер дисциплины**, **Название дисциплины**, **Количество часов лекций** (см. рис.2.2). На основе этих сущностей и атрибутов создадим аналогичные отношения (см. рис.2.3). Эти отношения являются объектными отношениями. 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 **Номер зачетной книжки**. Для отношения **ДИСЦИПЛИНА** в качестве ключа можно ис-

пользовать атрибут Номер дисциплины. Для удобства ключ записывают в первом столбце отношения. 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.

СТУДЕНТ

Номер зачетной книжки	Фамилия	Курс	Группа	Специальность
20030111	Иванов	2	АС-21	АСОИ
20030112	Петров	2	АС-22	АСОИ
20030113	Сидоров	3	Э-31	ЭВМиС

ДИСЦИПЛИНА

Номер дисциплины	Название дисциплины	Количество часов лекций
2001	Программирование	68
2002	Базы и банки данных	48
2003	Системотехника	32

Рис. 2.3. Структура отношений СТУДЕНТ и ДИСЦИПЛИНА

Связное отношение предназначено для хранения ключей двух или более объектных отношений, т.е. по ключам устанавливаются связи между объектами отношений (см. рис.2.4). В качестве примера рассмотрим отношение ИЗУЧАЕТ (Номер зачетной книжки, Номер дисциплины), которое является связным и определяет, что студент изучает определенную дисциплину (см. рис.2.4). Отношение ИЗУЧАЕТ состоит из ключей Номер зачетной книжки и Номер дисциплины, которые определяют связи с отношениями СТУДЕНТ и ДИСЦИПЛИНА.

ИЗУЧАЕТ

Номер зачетной книжки	Номер дисциплины
20030111	2001
20030112	2002
20030113	2003

Рис. 2.4. Структура отношения ИЗУЧАЕТ

Связное отношение, кроме связываемых ключей, может иметь и другие атрибуты, которые функционально зависят от этой связи. Примером может быть другое связное отношение РЕЗУЛЬТАТ (Номер зачетной книжки, Номер дисциплины, Оценка), которое изображено на рис.2.5.:

РЕЗУЛЬТАТ

Номер зачетной книжки	Номер дисциплины	Оценка
20030111	2001	4
20030112	2002	5
20030113	2003	3

Рис. 2.5. Структура отношения РЕЗУЛЬТАТ

2.2. ЦЕЛОСТНОСТЬ БАЗЫ ДАННЫХ

Целостность БД это поддержание данных БД в целостном непротиворечивом состоянии. Именно такое состояние гарантирует корректность данных. Поддержка целостности БД реализуется посредством ограничений, которые накладываются на данные.

Целостность данных. К данному типу ограничений относятся:

- **Ограничения домена.** Поскольку каждый атрибут связан с некоторым доменом, то для множества допустимых значений каждого атрибута отношения определяются ограничения домена. Или наборот домен атрибута задает множество значений, которое может принимать атрибут. Если значение атрибута в настоящий момент неизвестно или неприемлемо для данного кортежа, то в этом случае рекомендуется использовать значение NULL – это обозначение того, что отсутствует значение атрибута. Например, преподаватель принят на работу, но состав его нагрузки еще не определен. Следует отметить, что не каждая СУБД поддерживает реализацию NULL значения.
- **Категорная целостность (целостность сущностей)** ограничивает набор значений первичных ключей базовых отношений. Кортеж не может записываться в БД до тех пор, пока значения его ключевых атрибутов не будут полностью определены. Базовое отношение – это отношение, являющееся частью БД.
- **Ссылочная целостность** – если в отношении существует внешний ключ, то значение внешнего ключа должно соответствовать значению потенциального ключа некоторого кортежа в его базовом отношении. 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.
- **Компаративные ограничения целостности** – дополнительные правила поддержки целостности данных, определяются пользователем или администратором БД. Например, номер сотрудника должен находиться в диапазоне 1 – 245, студентам нельзя планировать занятия продолжительностью более восьми часов в день.

2.3. ТИПЫ ФУНКЦИОНАЛЬНЫХ ЗАВИСИМОСТЕЙ

Концепция функциональных зависимостей лежит в основе проектирования структур реляционных БД. Функциональная зависимость обладает набором формальных свойств, позволяющих формально решить многие проблемы. В качестве примера рассмотрим отношение **ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНА** с составным ключом, состоящим из двух атрибутов **ЛичныйНомер** и **НомерДисциплины** и неключевых атрибутов **КоличествоЧасов**, **Фамилия**, **Должность**, **Оклад**, **Кафедра**, **Телефон** (см.рис.2.6).

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 → B**. Для примера из рис.2.6 можно привести следующие функциональные зависимости:

Должность → Оклад, Фамилия → Кафедра, Кафедра → Телефон.

ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНА

Личный Номер пре- подавателя	Номер дисци- плины	Количес- тво часов лнкций	Фамилия	Должность	Оклад	Кафедра	Теле- фон
201	2004	36	Фролов	Доцент	380	ЭВМ	4-89
301	2005	48	Бойко	Профессор	520	АСУ	4-89
401	2005	52	Глазов	Ассистент	270	ТЭФ	4-12
401	2006	30	Глазов	Ассистент	270	ТЭФ	4-12

Рис. 2.6. Структура отношения ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНА

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 вышеприведенного примера (см. рис.2.6.) следует:

(Личный номер преподавателя, Номер дисциплины) →
(Количество Часов, 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 **Количество Часов** зависит от части ключа, т.е. только от атрибута **Номер дисциплины**.

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 A , B и C выполняются условия $A \rightarrow B$ и $B \rightarrow C$, то C транзитивно зависит от A . В рассматриваемом примере транзитивная зависимость следующая:

Фамилия → Кафедра → 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 .

Многозначная зависимость. В отношении R атрибут B многозначно зависит от A ($A \twoheadrightarrow B$), если каждому значению A соответствует множество значений B , 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 между преподавателем и группами студентов существует связь типа "один-ко-многим" (1:M), поскольку преподаватель может читать лекции в одной и более группах, а каждая группа посещает лекции у одного преподавателя.

1 : M

Личный номер преподавателя → **Номер группы**

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

Личный номер преподавателя ↔ **Номер дисциплины**

2.4. ОБЩИЕ СВЕДЕНИЯ О НОРМАЛИЗАЦИИ ОТНОШЕНИЙ В РБД

Основные положения теории нормализации и применение правил нормализации к отношениям БД приведены в [1-6].

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

атрибутами, учитываемыми в схеме отношения. Состав атрибутов отдельного отношения БД должен удовлетворять двум основным требованиям:

- атрибуты должны быть сгруппированы так, чтобы обеспечивалось минимальное дублирование данных;
- между атрибутами отношения не должно быть нежелательных функциональных зависимостей, которые ведут к аномалиям при вставке, удалении или обновлении данных.

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 между атрибутами отношения. Аппарат нормализации отношений определяет различные нормальные формы (НФ), каждая из которых ограничивает конкретный тип функциональных зависимостей отношения. Выделены следующие НФ - 1НФ, 2НФ, 3НФ, НФБК - нормальная форма Бойса-Кодда, 4НФ, 5НФ. Общая структура процесса нормализации представляется в виде последовательного построения НФ разного типа:

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

Процесс нормализации включает пять следующих основных этапов:

- 1) приведение отношений БД к первой нормальной форме (1НФ), позволяющее удалить из отношений повторяющиеся группы атрибутов;
- 2) приведение отношений БД ко второй нормальной форме (2НФ), позволяющее устранить частичную зависимость атрибутов от первичного ключа;
- 3) приведение отношений БД к третьей нормальной форме (3НФ), позволяющее устранить транзитивную зависимость атрибутов от первичного ключа;
- 4) приведение отношений БД к нормальной форме Бойса-Кодда (НФБК), позволяющее удалить из функциональных зависимостей оставшиеся аномалии;
- 5) приведение отношений БД к четвертой и пятой нормальным формам.

2.5. МЕТОДИКА ПОСТРОЕНИЯ БД РЕЛЯЦИОННОГО ТИПА

Проектирование логической модели БД реляционного типа – это одна из фаз процесса проектирования структуры БД, которая заключается в преобразовании КМ БД в ЛМ БД ПрО (с учетом возможностей выбранной СУБД реляционного типа), в проверке созданной ЛМ (нормализация отношений, выполнимость задач пользователей и поддержка целостности данных) и обсуждении полученных результатов с будущими пользователями БД.

Общая схема процесса создания ЛМ БД реляционного типа представлена на рис.2.7.

Процесс построения логической модели БД состоит из четырех этапов (рис.2.7.):

- **Этап 1.** Построение ЛМ БД реляционного типа.
- **Этап 2.** Проверка ЛМ БД с помощью правил нормализации.
- **Этап 3.** Проверка выполнимости задач и требований целостности данных.
- **Этап 4.** Обсуждение ЛМ БД с пользователями.

Учебный процесс по созданию ЛМ БД организован в виде двух лабораторных работ («Построение ЛМ БД реляционного типа» и «Проверка структуры БД»), которые являются продолжением предыдущих лабораторных работ по дисциплине, и предполагает индивидуальную реализацию каждым студентом БД для заданной ПрО. Некоторые аспекты процесса создания ЛМ БД упрощены и адаптированы под индивидуальную работу студентов.

ПРОЦЕСС СОЗДАНИЯ ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ

ЭТАПЫ

ДЕЙСТВИЯ

Этап 1. Построение логической модели БД реляционного типа

- Удаление из КМ нежелательных элементов
- Построение отношений для ЛМ БД
- Построение связей между отношениями

Этап 2. Проверка ЛМ БД с помощью правил нормализации

- Построение 1НФ, 2НФ, 3НФ и других НФ

Этап 3. Проверка выполнимости задач и требований целостности данных

- Проверка выполнимости задач пользователя
- Проверка поддержки целостности данных

Этап 4. Обсуждение логической модели БД с пользователями

- Проверка возможности расширения модели в будущем
- Обновление документации на ЛМ БД
- Обсуждение ЛМ БД с пользователями

Лабораторная работа. Построение логической модели БД реляционного типа

Лабораторная работа. Проверка структуры реляционной БД

Рис.2.7. Общая схема методики логического проектирования структуры БД

3. ПОСТРОЕНИЕ ЛОГИЧЕСКОЙ МОДЕЛИ БД РЕЛЯЦИОННОГО ТИПА

Процесс построения ЛМ БД реляционного типа включает последовательное выполнение следующих четырех действий:

1. Удаление из КМ БД нежелательных элементов (см.п.3.1).
2. Построение набора отношений для ЛМ БД на основе КМ БД (см.п.3.2).
3. Определение связей между отношениями (см.п.3.3).
4. Документирование результатов построения ЛМ БД (см.п.3.4)

В качестве исходной информации для построения ЛМ БД используются результаты выполнения лабораторных работ по тематике «Проектирование концептуальной модели базы данных» [8].

3.1. УДАЛЕНИЕ ИЗ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ НЕЖЕЛАТЕЛЬНЫХ ЭЛЕМЕНТОВ

Основная цель данного действия - это подготовка КМ БД для преобразования в ЛМ БД путем удаления из нее нежелательных элементов, реализация которых в выбранной СУБД будет затруднена.

Для удаления нежелательных элементов рекомендуется выполнить следующие действия [1]:

1. Удаление связей типа $M:N$.
2. Удаление сложных связей.
3. Удаление рекурсивных связей.
4. Удаление связей с атрибутами.
5. Удаление избыточных связей.
6. Удаление множественных атрибутов.
7. Перепроверка связей типа $1:1$.

1. Удаление связей типа $M:N$. Если в КМ присутствуют связи типа $M:N$, то их следует устранить путем определения некоторой промежуточной сущности. Связь типа $M:N$ заменяется двумя связями типа $1:M$, устанавливаемыми со

вновь созданной сущностью. В качестве примера рассмотрим связь типа **M:N** (**ПРЕПОДАВАТЕЛЬ читает ДИСЦИПЛИНА** (рис.3.1), когда один и тот же преподаватель может читать несколько дисциплин, и одна и та же дисциплина может читаться несколькими преподавателями). С целью устранения связи **читает** необходимо определить промежуточную сущность **ЧТЕНИЕ** и создать две новые связи типа **1:M** (**читает** и **читается**). В результате связь **читает** типа **M:N** будет заменена двумя связями — связью **читает** между сущностями **ПРЕПОДАВАТЕЛЬ** и **ЧТЕНИЕ** и связью **читается** между сущностями **ДИСЦИПЛИНА** и **ЧТЕНИЕ** (см. рис.3.2).

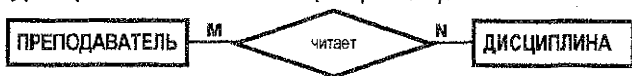


Рис.3.1. Пример представления связи типа **M:N**



Рис.3.2. Результат преобразования связи типа **M:N** в совокупность бинарных связей

2. Удаление сложных связей. Сложной называется связь, существующая между тремя и больше сущностями.

Если в КМ присутствует сложная связь, ее следует устранить с помощью промежуточной сущности. Она заменяется необходимым количеством бинарных связей типа **1:M**, устанавливаемых со вновь созданной сущностью.

Например, тройная связь (**ведет**), отражает отношения, существующие между преподавателями, занятиями и лабораториями (см. рис.3.3.). Эту сложную связь можно упростить путем введения новой сущности и определения бинарных связей между новой сущностью и каждой из исходных сущностей сложной связи. Например, связь **ведет** можно устранить посредством введения новой сущности с именем **ЛАБ_ЗАНЯТИЯ**. Эта сущность будет связана с исходными сущностями тремя новыми бинарными связями: связь **ведет** между сущностями **ПРЕПОДАВАТЕЛЬ** и **ЛАБ_ЗАНЯТИЯ**, связь **ведутся** между сущностями **ЗАНЯТИЯ** и **ЛАБ_ЗАНЯТИЯ**, связь **в** между сущностями **ЛАБОРАТОРИЯ** и **ЛАБ_ЗАНЯТИЯ** (см. рис.3.4).

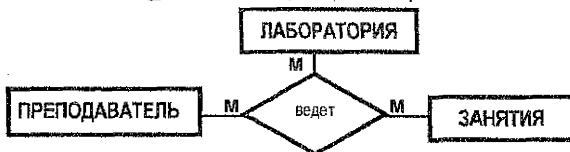


Рис.3.3. Представление на диаграмме «сущность-связь» тернарного типа связи

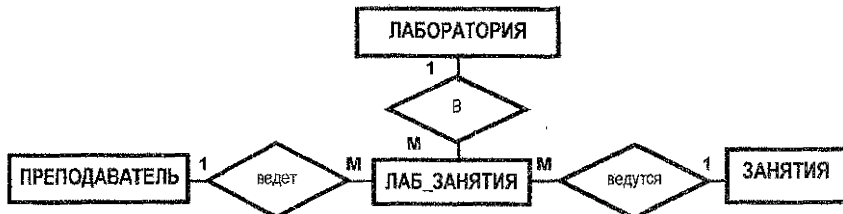


Рис.3.4. Результаты преобразования тернарной связи в совокупность бинарных связей

3. Удаление рекурсивных связей. Рекурсивными называются такие связи, в которых некоторая сущность взаимодействует сама с собой.

Если КМ БД содержит рекурсивные связи, то они должны быть устранены посредством определения некоторой промежуточной сущности. Например, для отображения ситуации, когда один из сотрудников руководит группой других сотрудников, может быть установлена рекурсивная связь типа 1:M **СОТРУДНИК управляет СОТРУДНИКАМИ** (см.рис.3.5).

Для упрощения данной рекурсивной связи типа 1:M заменим ее вновь созданной сущностью **ПОДЧИНЕННЫЙ СОТРУДНИК** и дополнительной связью типа 1:1 с именем **управляется** (см. рис.3.5.). Удаление рекурсивной связи типа M:N выполняется так же, как и бинарных связей типа M:N, описанных выше.

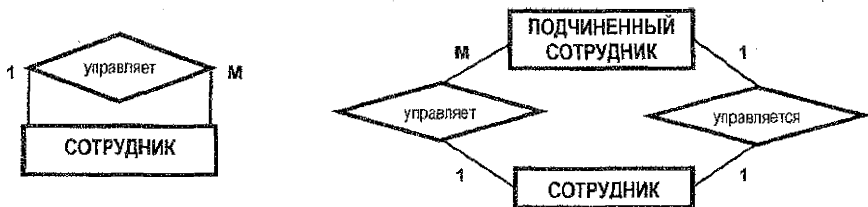


Рис.3.5. Результаты преобразования рекурсивной связи в совокупность бинарных связей

4. Удаление связей с атрибутами. Если в КМ БД присутствуют связи, имеющие собственные атрибуты, они должны быть преобразованы путем создания новой сущности.

Например, рассмотрим ситуацию, когда требуется фиксировать количество учебных часов, которые отведено для лекций по конкретной дисциплине (рис.3.6). Связь **ПРЕПОДАВАТЕЛЬ читает ДИСЦИПЛИНА** имеет атрибут с именем **Время**. Преобразуем связь **читает** в сущность с именем **ЛЕКЦИЯ**, которому назначим атрибут **Время**, после чего создадим две новые связи типа 1:M (см. рис.3.7).

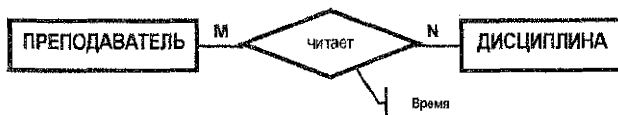


Рис.3.6. Пример связи с атрибутами



Рис.3.7. Результаты преобразования связи с атрибутами в совокупность бинарных связей

5. Удаление множественных атрибутов. Множественными называют атрибуты, которые могут иметь одновременно несколько значений для одного и того же экземпляра сущности.

Если в КМ БД присутствует множественный атрибут, его следует преобразовать путем определения новой сущности. Например, для отображения ситуации, когда одна и та же кафедра имеет несколько телефонных номеров, в КМ БД был определен множественный атрибут **НомерТелефона**, относящийся

к сущности **КАФЕДРА**. Этот множественный атрибут следует удалить, определив новую сущность, например **ТЕЛЕФОН**, имеющую единственный простой атрибут **НомерТелефона**, и создав новую связь типа **1:М** с именем **имеет**.

Данное действие не является обязательным. Оно может быть выполнено при проверке отношений с помощью правил нормализации (см. этап 2 на рис.2.7).

8. Перепроверка связей типа 1:1. В процессе определения сущностей могли быть созданы две различные сущности, которые на самом деле представляют один и тот же объект в ПрО. В подобных случаях следует объединить эти две сущности в одну.

Процедура объединения сущностей аналогична рассмотренной процедуре в [8]. Если первичные ключи объединяемых сущностей различны, выберите один из них в качестве первичного ключа, а другой укажите как альтернативный ключ.

7. Удаление избыточных связей. Связь является избыточной, если одна и та же информация может быть получена не только через нее, но и с помощью другой связи.

Если избыточная связь не является очевидно необходимой, ее следует удалять. Установить, что между двумя сущностями имеется больше одной связи, довольно просто. Однако из этого еще не следует, что одна из двух связей обязательно является избыточной, поскольку обе они могут представлять различные объединения, реально существующие в ПрО.

Например, рассмотрим ситуацию, когда необходимо смоделировать связи между сущностями **МУЖЧИНА**, **ЖЕНЩИНА** и **РЕБЕНОК** (см. рис.3.8). Очевидно, что между сущностями **МУЖЧИНА** и **РЕБЕНОК** имеется два пути доступа; один – через непосредственную связь **является отцом** и другой – через связь **женат на** и **является матерью**. На первый взгляд, кажется, что связь **является отцом** является избыточной. Однако это утверждение может оказаться ошибочным по двум причинам. Во-первых, отец может иметь детей от предыдущего брака, а мы моделируем только текущий брак отца (через связь **1:1**). Во вторых, отец и мать могут быть вообще неженаты, или отец может быть женат на женщине, которая не является матерью данного ребенка (или же мать может быть замужем за мужчиной, который не является отцом ребенка). Поэтому все существующие взаимоотношения не могут быть смоделированы без использования связи типа **является отцом**.



Рис.3.8. Пример множественных связей, не являющихся избыточными

Результатом выполнения действия «удаление из концептуальной модели нежелательных элементов» является удаление из КМ БД всех тех элементов, реализация которых затруднительна в среде выбранной реляционной СУБД. Следует отметить, что перечисленные выше действия выполняются только при необходимости.

Завершающим действием этого процесса является внесение соответствующих изменений в документацию по КМ БД.

3.2. ОПРЕДЕЛЕНИЕ НАБОРА ОТНОШЕНИЙ ДЛЯ БД

Основная цель данного действия – это построение набора отношений для логической модели БД. Отношения создаются на основе информации о сущностях и их атрибутах, которые определены в документации на КМ БД [8] и уточнены в рамках п. 3.1.

Обобщенную структуру любого отношения в ЛМ можно представить в виде совокупности полей, которые по функциональному назначению делятся на четыре группы (см. рис.3.9):

1. Внешние ключи отношения – определяют связи между отношениями БД;
2. Первичный ключ отношения;
3. Альтернативные ключи отношения;
4. Список простых (неключевых) атрибутов отношения.

ИмяОтношения			
Внешние ключи отношения	Первичный ключ отношения	Альтернативные ключи отношения	Список атрибутов отношения

Рис.3.9. Обобщенная структура отношения для ЛМ БД

Рекомендуется располагать атрибуты отношения БД в порядке, который предложен выше.

Отдельная сущность КМ БД в ЛМ БД представляется в виде отдельного отношения. Таким образом, ЛМ БД представляется в виде совокупности отношений, равных количеству сущностей в КМ БД.

Определение отдельного отношения в ЛМ реализуется путем выполнения следующих действий для отдельной сущности:

1. **Определение идентификатора отношения.** В качестве имени отношения рекомендуется использовать имя сущности;
2. **Уточнение первичного и альтернативных ключей для отношения.** Эти ключи определяются на основе описания ключей сущности (см. документация на КМ БД [8]);
3. **Уточнение не ключевых атрибутов отношения.** Аналогично ключам отношения не ключевые атрибуты отношения определяются из таблицы атрибутов сущности (см. документация на КМ БД [8]).

Аналогичным образом для остальных сущностей, которые входят в состав КМ БД создаются отношения для ЛМ. Следует отметить, что в рамках данного этапа в этих отношениях (таблицах) пока отсутствуют внешние ключи, так как эта информация формулируется в рамках следующего этапа. В качестве примера для построения отношений для ЛМ будем использовать фрагмент КМ, который изображен на рис.3.10.

Для документирования отношений в ЛМ используется табличный способ представления. В отличие от документации на КМ каждое отношение представляется в виде отдельной таблицы (см. табл.3.1), которая объединяет описание отдельной сущности, ее атрибутов и ключей. Пример табличного представления для отношения **СТУДЕНТ** приведен в табл.3.1.

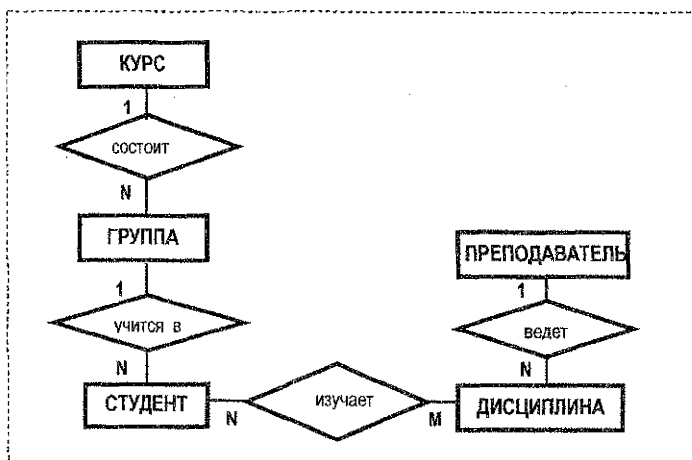


Рис.3.10. Пример фрагмента диаграммы «сущность-связь»

Таблица 3.1

Описание отношения СТУДЕНТ

№ п/п	Имя атрибута	Тип атрибута	Описание	Тип данных, длина	Ограничения	Значение по умолчанию	Псевдоним	Допустимость NULL	Производный
1	Номер зачетной книжки	Перв. ключ	Уник. идентиф. отделения	Числовой,3	Первичный ключ	нет	нет	нет	нет
2	ФИО студента	Альт. ключ	Уник. идентиф. отделения	Строковый,60		нет	нет	нет	нет
3	Номер паспорта	Альт. ключ	Уник. идентиф. отделения	Числовой,13		нет	нет	нет	нет
4	Улица	Простой	Назв. улицы	Строковый,25		нет	нет	нет	нет
5	Город	Простой	Назв.города	Строковый,15		нет	нет	нет	нет
6	Номер дома	Простой		Числовой,3		>0 и < 400	нет	нет	нет
7	Номер квартиры	Простой		Числовой,3		>0 и < 432	нет	нет	нет
8	Почтовый индекс	Простой		Числовой,6		>100000 и < 999999	нет	нет	нет

Для графического представления отдельного отношения можно использовать различные способы. Например, на рисунке 3.11 отношение СТУДЕНТ представлено как в горизонтальном, так и вертикальном виде. Возможны и другие способы представления. Способ представления определяется разработчиком.

Результатом выполнения действия «определение набора отношений для БД» является набор стношений для логической модели БД реляционного типа, который построен на основе документации на КМ (результаты построения концептуальной модели БД [8]) и задокументирован в табличном виде (см.табл.3.1).

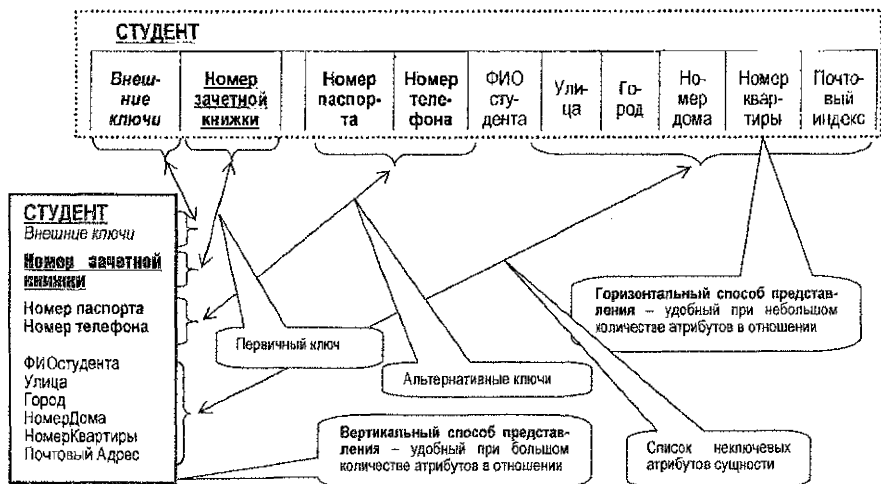


Рис.3.11. Пример графического представления отдельного отношения ЛМ БД

3.3. ОПРЕДЕЛЕНИЕ СВЯЗЕЙ МЕЖДУ ОТНОШЕНИЯМИ БД

Основная цель данного действия – это определение связей между отношениями и добавление их в отношения ЛМ БД. Эти связи определяются на основе связей, которые были определены между сущностями в КМ БД и представлены в таблице связей [8].

Связи между сущностями описываются с помощью механизма первичных и внешних ключей. Для принятия решения о том, откуда взять и куда поместить значения атрибута(ов) внешнего ключа, предварительно следует установить, какая из участвующих в связи сущностей является родительской, а какая – дочерней. Родительской считается сущность, которая передает копию набора значений своего первичного ключа в отношение, представляющее дочернюю сущность, где эти значения будут играть роль внешнего ключа.

Бинарные связи типа «один к одному» (1:1). Для каждой присутствующей в ЛМ бинарной связи типа 1:1, установленной между сущностями **E1** и **E2**, необходимо переслать атрибуты первичного ключа сущности **E1** в отношение, представляющее сущность **E2**. Эти атрибуты будут использоваться в нем в качестве внешнего ключа. Определение родительской и дочерней сущностей зависит от ограничений участия, наложенных на члены отношения **E1** и **E2**.

Сущность, которая частично участвует в связи, определяется как родительская, а та сущность, которая участвует в связи полностью (тотально), определяется как дочерняя. Копия набора значений первичного ключа родительской сущности помещается в отношение, представляющее дочернюю сущность.

Отметим, что в том случае, когда оба вида сущностей участвуют в связи типа 1:1 либо тотально, либо частично, выбор родительской и дочерней сущностей может выполняться произвольно. Более того, если обе сущности уча-

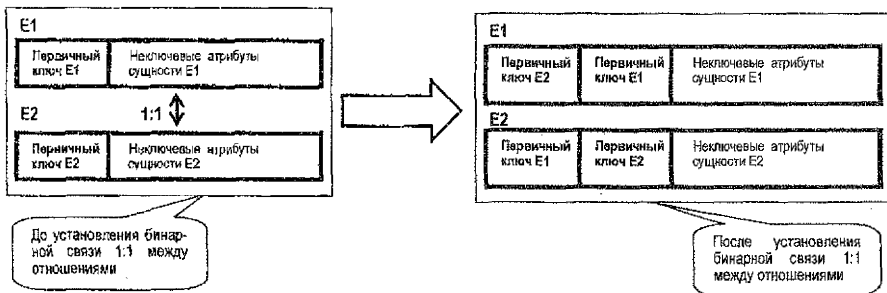


Рис.3.12. Пример установления бинарных связей типа 1:1 между отношениями E1 и E2

ствуют в связи тотально, можно (на выбор) либо представить эту связь с помощью пары первичного и внешнего ключей (как описывалось выше), либо слить атрибуты обеих сущностей в единое отношение. Слияние в единое отношение предпочтительнее в том случае, если данные сущности не принимают участия в других связях. Пример объединения отношений приведен на рис.3.13.

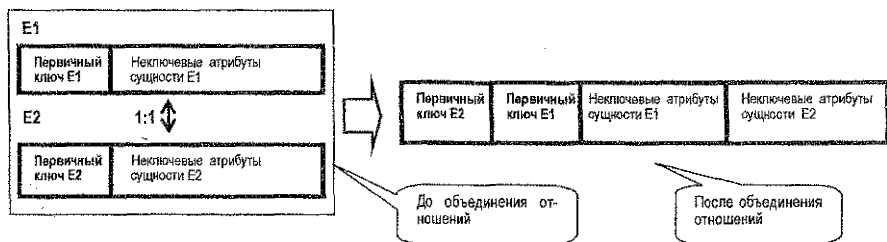


Рис.3.13. Пример объединения отношений E1 и E2

Бинарные связи типа «один ко многим» (1:M). Для каждой бинарной связи типа 1:M, установленной в ЛМ между сущностями E1 и E2, необходимо переслать копию атрибутов первичного ключа сущности E1 в отношение, представляющее сущность E2, где они будут играть роль внешнего ключа (см. рис.3.14). Сущность, представляющая "единичную" сторону связи определяется как родительская сущность, а сущность, представляющая "множественную" сторону как дочерняя сущность.

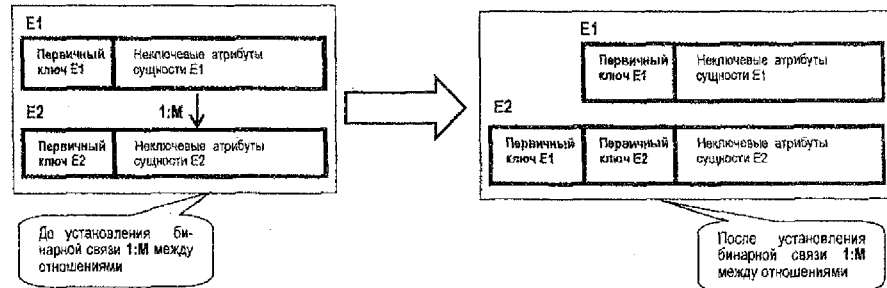


Рис.3.14. Пример установления бинарных связей типа 1:M между отношениями E1 и E2

Для представления данной связи необходимо скопировать первичный ключ родительской сущности в отношение, представляющее дочернюю сущность, где этот ключ должен быть описан как внешний.

Пример построения отношений и связей типа 1:M изображен на рис.3.15. В качестве исходных приведены сущности КУРС, ГРУППА и СТУДЕНТ, а также связи КУРС состоит из ГРУПП (тип 1:M), в ГРУППе учатся СТУДЕНТЫ (тип 1:M).

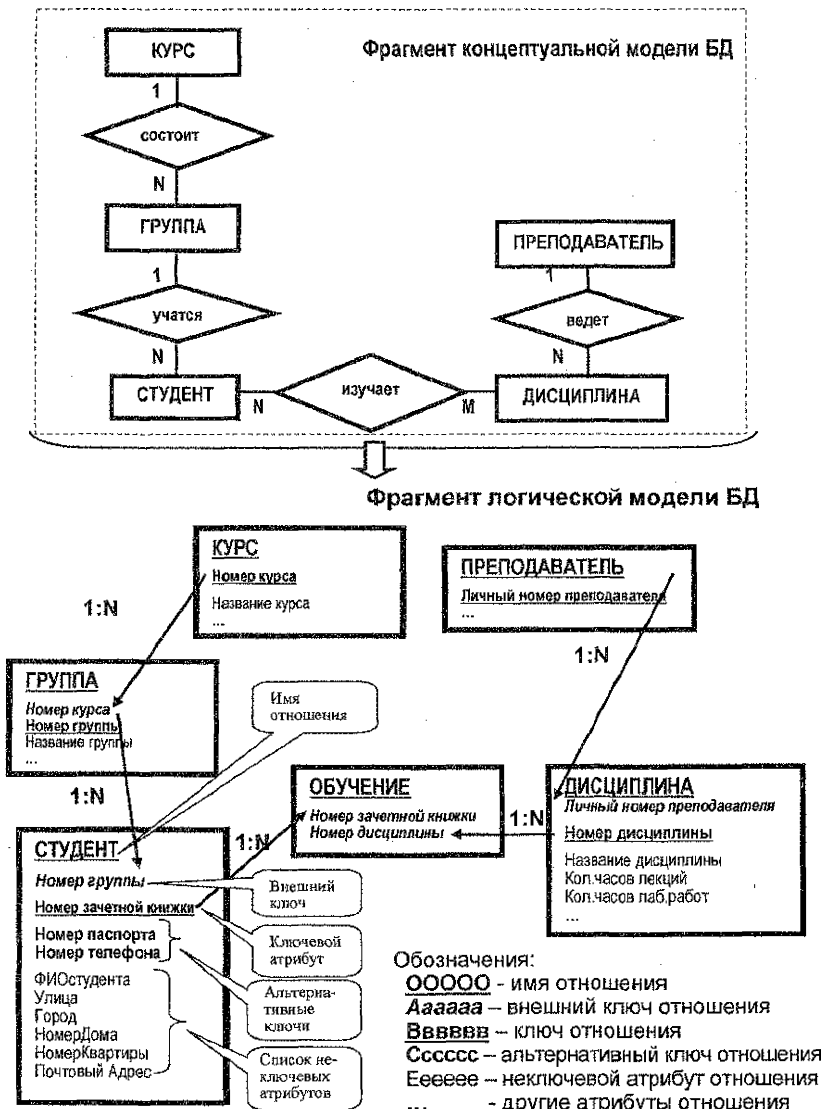


Рис.3.15. Пример построения отношений и схемы связей для фрагмента ЛМ БД

Кроме этого, на КМ приведена связь между сущностями **СТУДЕНТ изучает ДИСЦИПЛИНЫ** имеет тип **N:M**, поскольку каждый студент изучает много дисциплин, а отдельную дисциплину изучает много студентов. Удаление этой нежелательной связи реализуется путем введения дополнительной сущности. Этот процесс представлен на рис.3.15 путем введения сущности **ОБУЧЕНИЕ** и связей между этой сущностью и сущностями **СТУДЕНТ** и **ДИСЦИПЛИНА**. В результате окончательный вид отношений и схемы взаимосвязей между отношениями логической модели БД представлен на рис.3.15.

Результатом выполнения действия «определение связей между отношениями БД» является выполнения построение схемы связей между отношениями ЛМ БД, которые фиксируются в отношениях в виде внешних ключей и документируются в графическом виде (см. рис.3.15).

3.4. ДОКУМЕНТИРОВАНИЕ РЕЗУЛЬТАТОВ ПОСТРОЕНИЯ ЛОГИЧЕСКОЙ МОДЕЛИ БД

Основная цель данного действия – это представление результатов проектирования логической модели БД в виде документации на ЛМ БД и уточнение документации на КМ БД [8].

Документация на ЛМ БД представляется в виде совокупности следующих компонентов:

1. **Определение списка отношений** и назначение каждого отношения для ЛМ БД - это текстовый документ.
2. **Описание отношений** ЛМ БД. Каждое отдельное отношение представляется в виде отдельного описания, которое содержит структуру отношения и описание всех атрибутов и их характеристик (см. табл.3.1).
3. **Схема связей между отношениями** ЛМ БД. Рекомендуется графическая форма представления этой схемы (см. рис.3.15). Обязательными компонентами этой схемы являются отношения, ключевые атрибуты и связи между отношениями.

При документировании результатов проектирования ЛМ БД можно использовать различные доступные средства автоматизации проектирования и документирования ЛМ БД. Например, возможности пакета ERWIN [6].

4. ПОСТАНОВКА ЗАДАЧ НА ЛАБОРАТОРНУЮ РАБОТУ «ПРОВЕРКА СТРУКТУРЫ БД»

ТЕМА:

Проверка структуры реляционной базы данных

ЦЕЛЬ:

Сформировать знания и практические умения по проверке:

- логической модели БД с помощью правил нормализации;
- выполнимости задач пользователей с использованием БД;
- требований целостности данных БД.

ЗАДАЧИ:

1. Проверить ЛМ БД с помощью правил нормализации путем решения следующих задач:
 - Построить первую нормальную форму (1НФ) для отношений ЛМ БД;
 - Построить вторую нормальную форму (2НФ) для отношений ЛМ БД;
 - Построить третью нормальную форму (3НФ) для отношений ЛМ БД;
 - Построить усиленную нормальную форму Бойса-Кодда (НФБК);
 - Обновить документацию на ЛМ БД при необходимости.

2. Проверить ЛМ БД в отношении выполнимости задач пользователей и при необходимости внести необходимые изменения в проект ЛМ БД.
3. Определить требования поддержки целостности данных и внести необходимые изменения и уточнения в документацию на ЛМ БД.
4. Создать окончательный вариант диаграммы «сущность-связь» для КМ БД.
5. Обновить документацию на БД (при необходимости).

ИСХОДНЫЕ ДАННЫЕ:

Результаты построения логической модели БД

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:

1. Изучить теоретическую часть лабораторной работы (см. п.3 и п.5).
2. Ответить на контрольные вопросы.
3. Изучить примеры нормализации отношений БД.
4. Изучить примеры проверки выполнимости задач пользователя и поддержки целостности данных.
5. Решить перечисленные выше задачи для заданной реляционной БД.
6. Оформить результаты выполнения лабораторной работы.
7. Защитить у преподавателя результаты выполнения лабораторной работы.

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ РАБОТЫ:

Отчет по работе должен содержать обновленную документацию на БД:

1. Структура нормализованных отношений (таблиц) БД;
2. Схема взаимосвязей между отношениями БД.
3. Перечень задач, которые не могут быть реализованы в рамках созданной структуры БД, и предполагаемые изменения проекта структуры БД;
4. Список требований поддержки целостности данных;
5. Окончательный вариант диаграммы «сущность-связь» для КМ БД;
6. Уточненная документация на БД (при необходимости).

5. ПРОВЕРКА СТРУКТУРЫ ЛОГИЧЕСКОЙ МОДЕЛИ БД

Проверка структуры ЛМ БД включает три следующих проверки:

1. Проверка ЛМ БД с помощью правил нормализации отношений.
2. Проверка выполнимости задач пользователей в рамках ЛМ БД.
3. Проверка целостности данных для БД.

5.1. ПРОВЕРКА ЛОГИЧЕСКОЙ МОДЕЛИ С ПОМОЩЬЮ ПРАВИЛ НОРМАЛИЗАЦИИ

Основная цель данного действия – это проверка созданной структуры отношений для ЛМ БД (результат предыдущей лабораторной работы) с помощью правил нормализации. Отношения должны находиться, как минимум в 3 нормальной форме.

В рамках данного лабораторного практикума рассмотрены только первые три этапа нормализации:

1. Приведение отношений БД к 1 НФ, позволяющее удалить из отношений повторяющиеся группы атрибутов;
2. Приведение отношений БД ко 2НФ, позволяющее устранить частичную зависимость атрибутов от первичного ключа;
3. Приведение отношений БД к 3НФ, позволяющее устранить транзитивную зависимость атрибутов от первичного ключа;

5.1.1. ПОСТРОЕНИЕ ПЕРВОЙ НОРМАЛЬНОЙ ФОРМЫ

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 отношения логической модели БД, находящаяся в 1НФ, реализуется следующим образом:

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

Определить понятие атомарности трудно. Значение, атомарное в одном приложении, может быть не атомарным в другом. Можно руководствоваться общим принципом, что значение не атомарно, если в приложении оно используется по частям.

Рассмотрим фрагмент отношения **СТУДЕНТ**, представленный на рис.5.1. Если значение атрибута **Дата рождения** предполагается использовать целиком, то в этом случае данное отношение находится в 1НФ. Если бы потребовалось выделить и отдельно использовать, скажем, год, число, месяц, то это отношение не находилось бы в 1НФ, так как требуемые данные являются только частями значения атрибута **Дата рождения**. Для того чтобы привести данное отношение к 1НФ, атрибут должен быть разбит на три части – атрибут **День рождения**, атрибут – **Месяц рождения**, атрибут – **Год рождения** (см.рис.5.2).

СТУДЕНТ

Номер зачетной книжки	Курс	Дата рождения	...
940132	2	10 января 1980	...
930013	3	15 марта 1981	...
950104	1	20 апреля 1981	...
940133	2	1 октября 1980	...
950105	1	10 февраля 1980	...

Рис. 5.1. Структура отношения **СТУДЕНТ**

СТУДЕНТ

Номер зачетной книжки	Курс	День рождения	Месяц рождения	Год рождения
940132	2	10	Январь	1980
930013	3	15	Март	1981
950104	1	20	Апрель	1981
940133	2	1	Октябрь	1980
950105	1	10	Февраль	1980

Рис. 5.2. Структура отношения **СТУДЕНТ** после приведения к 1НФ

Второй вариант преобразования таких отношений – это деление отношения на два. Первое отношение получается путем удаления из исходного отношения неатомарного атрибута, а второе - включает неатомарный атрибут (его в дальнейшем необходимо преобразовать в совокупность атомарных атрибутов) и ключ исходного отношения, из которого этот атрибут выделен.

В качестве второго примера для иллюстрации построения 1НФ используем отношение **ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНА** (см. рис.5.3).

ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНА

Личный номер преподавателя	Номер дисциплины	Количество часов лекций	Фамилия	Должность-Оклад	Кафедра - Телефон
201	2004	36	Фролов	Доцент, 380	ЭВМ, тел.4-89
301	2005	48	Бойко	Профессор, 520	АСУ, тел. 4-89
401	2005	52	Глазов	Ассистент, 270	ТЭФ, тел. 4-12
401	2006	30	Глазов	Ассистент, 270	ТЭФ, тел. 4-12

Рис. 5.3. Структура отношения ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНА

В этом отношении два атрибута являются претендентами на преобразование: **Должность-Оклад** и **Кафедра-Телефон**. В результате приведения этих атрибутов к атомарному виду получим отношение в 1НФ (см.рис.5.4).

ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНА

Личный номер преподавателя	Номер дисциплины	Количество часов лекций	Фамилия	Должность	Оклад	Кафедра	Телефон
201	2004	36	Фролов	Доцент	380	ЭВМ	4-89
301	2005	48	Бойко	Профессор	520	АСУ	4-89
401	2005	52	Глазов	Ассистент	270	ТЭФ	4-12
401	2006	30	Глазов	Ассистент	270	ТЭФ	4-12

Рис. 5.4. Структура отношения ПРЕПОДАВАТЕЛЬ*ДИСЦИПЛИНА

Каждая тип НФ ограничивает определенный тип функциональной зависимости и устраняет соответствующие аномалии при выполнении операций над отношениями БД. Приведенное на рис.5.4. отношение находится в 1НФ, но в этом отношении можно отметить частичную функциональную зависимость:

- атрибута **Фамилия** от части составного ключа – **Личный номер преподавателя**;
- атрибута **Количество часов лекций** от части составного ключа – **Номер дисциплины**.

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.

5.1.2. ПОСТРОЕНИЕ ВТОРОЙ НОРМАЛЬНОЙ ФОРМЫ

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 отношение на части. Каждая из полученных частей должна представлять собой отдельное отношение, в котором содержатся только функционально зависимые атрибуты. Этот процесс продолжается до тех пор, пока не будут получены отношения, 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 (см. рис.5.4) присутствуют частично функциональные зависимости:

- атрибута **Фамилия** от части составного ключа – **Личный номер преподавателя**;
- атрибута **Количество часов лекций** от части составного ключа – **Номер дисциплины**.

В итоге преобразования получим два отношения **ПРЕДМЕТ** и **ПРЕПОДАВАТЕЛЬ** (см. рис.5.5), находящиеся во 2НФ.

ДИСЦИПЛИНА

Личный номер преподавателя	Номер дисциплины	Количество часов лекций
201	2004	36
301	2005	48
401	2005	52
401	2006	30

ПРЕПОДАВАТЕЛЬ

Личный номер преподавателя	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

Рис.5.5. Структура отношений ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ

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

Личный номер преподавателя → Кафедра → Телефон
Личный номер преподавателя → 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.

5.1.3. ПОСТРОЕНИЕ ТРЕТЬЕЙ НОРМАЛЬНОЙ ФОРМЫ

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** (см.рис.5.5) необходимо разделить на три отношения: **ПРЕПОДАВАТЕЛЬ**, **ДОЛЖНОСТЬ** и **КАФЕДРА** (см.рис.5.6).

ПРЕПОДАВАТЕЛЬ

<u>Личный номер преподавателя</u>	<u>Кафедра</u>	<u>Должность</u>	<u>Фамилия</u>
201	ЭВМ	Доцент	Фролов
202	АСУ	Профессор	Костин
301	ТЭФ	Ассистент	Бойко
401	АСУ	Доцент	Глазов

КАФЕДРА

<u>Кафедра</u>	<u>Телефон</u>
ЭВМ	4-89
АСУ	4-77
ТЭФ	4-22

ДОЛЖНОСТЬ

<u>Должность</u>	<u>Оклад</u>
Доцент	3 380
Профессор	3 920
Ассистент	2 270

Рис.5.6. Структура отношений **ПРЕПОДАВАТЕЛЬ**, **КАФЕДРА** и **ДОЛЖНОСТЬ**

3НФ 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 (форма БОЙСА-КОДДА).

5.1.4. ПОСТРОЕНИЕ УСИЛЕННОЙ НОРМАЛЬНОЙ ФОРМЫ БОЙСА – КОДДА

Пусть имеется отношение **КУРС*ПРОЕКТ** (**Преподаватель**, **Дисциплина**, **Студент**), отражающее руководство преподавателей курсовыми проектами, которые выполняются студентами (см. рис.5.7).

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.

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

Рис.5.7. Структура отношения КУРС*ПРОЕКТ

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 атрибута **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 атрибута, т.е. зависимости **Студент** → **Дисциплина**. Для этого отношение **КУРС_ПРОЕКТ** разлагается на два отношения, которые будут находиться в НФБК.

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

Рис.5.8. Структура Отношений РУКОВОДСТВО и ВЫПОЛНЕНИЕ

Отношение находится в НФБК, если оно находится в ЗНФ и в нем отсутствуют зависимости ключей от описательных атрибутов. Следует отметить, что соединение полученных отношений по атрибуту **Дисциплина** дает исходное отношение **КУРС*ПРОЕКТ**.

Результатом выполнения действия «проверки логической модели путем применения к ней правил нормализации» является набор отношений ЛМ БД, который либо:

1. **Остается неизменным (см.п.3.2)**, так как все отношения ЛМ сразу находятся в третьей нормальной форме.
2. **Его необходимо изменить** в соответствии с результатами применения правил нормализации к отношениям ЛМ. В этом случае необходимо внести соответствующие изменения в структуру первоначальных отношений и связей ЛМ, а также внести изменения в документацию на БД, как для КМ (изменить набор сущностей и структуру диаграммы «сущность-связь»), так и для ЛМ.

5.2. ПРОВЕРКА ВЫПОЛНИМОСТИ ЗАДАЧ ПОЛЬЗОВАТЕЛЕЙ

Основная цель данного действия – это проверка реализуемости задач пользователей в рамках созданной ЛМ БД.

Реализацию проверки реализуемости задач можно выполнять следующим образом. Для каждой задачи необходимо вручную нанести непосредственно на схему связей ЛМ всех путей, которые потребуются для выполнения задач. Этот подход позволяет визуально выделить те области модели, которые не используются для выполнения задач, а также те области, которые наиболее существенны с точки зрения выполнения задач. В нашем распоряжении оказывается удобное средство прямого анализа поддержки, которую ЛМ обеспечивает для выполнения необходимых задач пользователя.

Если на схеме связей **имеются области, которые не используются** ни в одной из задач, возникает вопрос о целесообразности представления этой информации в ЛМ. В этом случае необходимо удалить лишние компоненты из модели и повторно провести все виды проверок для ЛМ. При положительном завершении этих проверок необходимо модифицировать компоненты БД и соответствующую документацию на БД.

Если в модели **отсутствуют компоненты, которые необходимы** для решения некоторой задачи, то потребуется провести анализ того, какие из

обязательных для выполнения задачи отношения, связи или атрибуты пропущены при создании модели. В этом случае необходимо модернизировать созданную модель, путем возврата на предыдущие этапы логического, а возможно, и концептуального проектирования БД, и добавить недостающие компоненты. Затем процесс проверки продолжить. При необходимости корректируется документация на БД.

Результатом «проверки реализуемости задач пользователя» является доказательство того, что задачи можно реализовать на основе созданной структуры БД для ПрО пользователя. При невозможности реализации задач структуру БД необходимо изменить, т.е. проанализировать выявленные недостатки и повторить необходимые операции проектирования БД.

5.3. ПРОВЕРКА ТРЕБОВАНИЙ ПОДДЕРЖКИ ЦЕЛОСТНОСТИ ДАННЫХ

Основная цель данного действия – это проверка требований поддержки целостности данных в рамках созданной ЛМ БД.

Ограничения целостности данных представляют собой такие ограничения, которые вводятся с целью предотвратить помещение в БД противоречивых данных. Следует отметить, что в конкретных СУБД функции контроля целостности могут как поддерживаться, так и не поддерживаться. Можно выделить следующие типы ограничений целостности данных:

1. *Обязательные данные;*
2. *Ограничения для доменов атрибутов;*
3. *Целостность сущностей;*
4. *Ссылочная целостность;*
5. *Требования данной ПрО (предприятия).*

1. Обязательные данные. Уточняется перечень атрибутов, которые всегда должны содержать одно из допустимых значений. Эти ограничения должны фиксироваться при занесении сведений об атрибуте в словарь данных [9].

2. Ограничения для доменов атрибутов. Каждый атрибут имеет домен, представляющий собой набор его допустимых значений. Например, атрибут "пол" может содержать одно из двух допустимых значений – "М" или "Ж", поэтому его домен состоит из двух символьных строк длиной в один символ, содержащих указанные значения. Данные ограничения устанавливаются при определении доменов атрибутов, присутствующих в модели данных на этапе проектирования КМ БД [8].

3. Целостность сущностей. Первичный ключ любой сущности не может содержать пустого значения. Например, каждая строка отношения **СОТРУДНИК** должна содержать уникальное значение атрибута первичного ключа; в данном случае это – атрибут **НомерСотрудника**. Эти ограничения определяются при описании первичных ключей для сущностей [8].

4. Ссылочная целостность. Внешний ключ связывает каждую строку дочернего отношения с той строкой родительского отношения, которая содержит это же значение соответствующего потенциального ключа. Понятие ссылочной целостности означает, что если внешний ключ содержит некоторое значение, то оно обязательно должно присутствовать в потенциальном ключе одной из строк родительского отношения. Например, атрибут **НомерОтделения** отношения **Сотрудник** связывает данные о каждом из работников со строкой

в отношении **ОТДЕЛЕНИЕ**, соответствующей тому отделению предприятия, в котором он работает. Если поле **НомерОтделения** не пусто, оно должно содержать допустимое значение, присутствующее в атрибуте **НомерОтделения** одной из строк отношения **ОБЪЕКТ**. В противном случае окажется, что работник трудится в несуществующем отделении.

5. Требования данной ПрО. Данная группа ограничений представляет ограничения конкретной ПрО (предприятия) или ограничения, вызванные бизнес-правилами. Например, обновление отношений может регламентироваться принятыми на предприятии правилами, описывающими методы выполнения задач, связанных с подобными обновлениями. Например, может быть принято правило, запрещающее одному сотруднику одновременно заниматься более чем десятью объектами недвижимости.

Результаты выполнения действия «проверка требований целостности данных для БД» фиксируются в виде отдельного документа для ЛМ БД – «Требования поддержки целостности данных», состоящего из следующих разделов:

1. Обязательные данные;
2. Ограничения для доменов атрибутов;
3. Целостность сущностей;
4. Ссылочная целостность;
5. Требования данного предприятия (приложения).

Форма представления этих требований определяется разработчиком.

5.4. УТОЧНЕНИЕ ДОКУМЕНТАЦИИ НА ПРОЕКТ БД

Основная цель данного действия – это обсуждение с пользователями точности, полноты и понятности представления ПрО в моделях БД, а также состава, структуры и содержания созданной документации для разработанной версии структуры БД.

После разработки ЛМ БД набор документации для БД, с учетом выполнения предыдущих лабораторных работ [8,9] включает следующие компоненты:

Требования к БД АСОИ:

1. Описание объекта автоматизации.
2. Каталог задач для автоматизации.
3. Словарь данных для хранения в БД.

Концептуальная модель БД:

1. Диаграммы «Сущность-связь» - обобщенная и набор локальных (для каждой задачи своя диаграмма).
2. Таблицы – сущностей, атрибутов, связей, доменов, ключей.

Логическая модель БД:

1. Схема связей между отношениями.
2. Таблицы – отношений БД.
3. Требования поддержки целостности данных.

КОНТРОЛЬНЫЕ ВОПРОСЫ

Построение логической модели БД

1. Основные этапы проектирования логической модели БД и их назначение.
2. Основные действия этапа построения и проверки логической модели БД.
3. Структура процесса удаления из концептуальной модели нежелательных элементов.
4. Необходимость удаления связей типа N:M и связей с атрибутами.
5. Как выполняются операции по удалению сложных, рекурсивных, множественных и избыточных связей в концептуальной модели.
6. Как определяются отношения для логической модели БД.
7. Как определяются связи между отношениями в логической модели БД.
8. Перечислите состав и назначение документов для фиксации результатов проектирования логической модели БД.

Нормализация отношений в БД

9. Определите основные положения реляционной модели данных.
10. Определите понятия: домен, кортеж, кардинальное число, степень таблицы, потенциальный ключ, первичный ключ, простой и составной ключ, внешний ключ, объектные и связанные отношения.
11. Определите основные типы ограничений для целостности данных для БД.
12. Приведите основные типы и примеры функциональных зависимостей (полная и неполная, транзитивная и многозначная).
13. Основные положения теории нормализации отношений в БД.
14. Методика построения нормализованной реляционной БД.
15. Определите 1НФ. Приведите пример аномалии отношений, находящихся в 1НФ.
16. Определите 2НФ. Приведите пример аномалии отношений, находящихся в 2НФ.
17. Определите 3НФ. Приведите пример аномалии отношений, находящихся в 3НФ.

Определение выполнимости задач и требований поддержки целостности данных

18. Назначение проверки возможности реализации задачи в рамках созданной структуры БД.
19. Приведите классификацию типов ограничений целостности данных.
20. Приведите пример обязательных данных и ограничений для доменов атрибутов.
21. Приведите пример нарушения целостности сущностей (отношений).
22. Назначение и особенности ссылочной целостности.
23. Ограничения предприятия или бизнес-правил.

ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ

- АСОИ - автоматизированная система обработки информации
БД - база данных
КМ - концептуальная модель
ЛМ - логическая модель
НФ - нормальная форма
НФБК - нормальная форма Бойса-Кодда
ПрО - предметная область
РБД - реляционная база данных
РМД - реляционная модель данных
СУБД - система управления базами данных

СПИСОК ЛИТЕРАТУРЫ

1. Малыгина М.Л. Базы данных: основы, проектирование, использование. – СПб.: БХВ-Петербург, 2004.-512 с.: Пер. с англ. – М.: Мир, 1984. – 296 с.
2. Конопли Т., Бегг К., Страчан А. Теория и практика, 2-е изд.: Пер. с англ. : Учебное пособие. – М.: Издательский дом «Вильямс», 2000. – 1120 с.
3. Змитрович А.И. Базы данных: учеб. пособие для вузов. – Мн.: Университетское, 1991.
4. Дейт К.Дж. Введение в системы баз данных: Пер. с англ. – 6-е изд.- К.: Диалектика, 1998. – 784 с.
5. Гиори Т., Фрай Дж. Проектирование структур баз данных: В 2-х кн. Кн.1. Пер. с англ. – М.: Мир, 1985. – 287 с.
6. Маклаков С.В. VPwin и ERwin. CASE-средства разработки информационных систем. – М.: ДИАЛОГ-МИФИ, 1999.
7. Мартин Дж. Организация баз данных в вычислительных системах: Пер. с англ. Под ред. А.А. Стогния и А.Л. Щерса. – М.: Мир, 1980. – 662 с.
8. Муравьев Г.Л., Хвещук В.И. Лабораторный практикум по дисциплине «Проектирование баз данных». Проектирование концептуальной модели базы данных. – Брест: БГТУ, 2007. – 40 с.
9. Хвещук В.И. Лабораторный практикум по дисциплине «Проектирование баз данных». Формирование требований к базе данных. – Брест: БГТУ, 2006.

УЧЕБНОЕ ИЗДАНИЕ

Составители: **Хвещук Владимир Иванович**
Муравьев Геннадий Леонидович

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по дисциплине
«ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ»

ПРОЕКТИРОВАНИЕ ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ

для студентов специальности 53 01 02 01
«Автоматизированные системы обработки информации»

Ответственный за выпуск: **Хвещук В.И.**
Редактор: **Строкач Т.В.**
Корректор: **Никитчик Е.В.**
Компьютерная верстка: **Кармаш Е.Л.**

Подписано к печати 29.11.2007 г. Формат 60x84¹/₁₆.
Усл. печ.л. 1,86. Уч.изд.л. 2. Зак. № **1270**. Тираж 100 экз.
Отпечатано на ризографе Учреждения образования
«Брестский государственный технический университет».
224017, Брест, ул. Московская, 267.