

1. Введение

Настоящие методические указания предназначены для проведения лабораторных работ по курсу "Организация баз данных" для студентов специальности 22.02 (Автоматизированные системы обработки информации и управления) и посвящены изучению фундаментальных свойств реляционных баз данных.

В настоящее время реляционный подход является наиболее распространенным как в области настольных, так и среди промышленных СУБД. К числу важных достоинств реляционного подхода можно отнести:

- наличие небольшого набора абстракций, позволяющих сравнительно просто моделировать большую часть распространенных предметных областей и допускающих точные формальные определения, оставаясь интуитивно понятными;
- наличие простого и в то же время мощного математического аппарата, опирающегося главным образом на теорию множеств и математическую логику и обеспечивающего теоретический базис реляционного подхода к организации баз данных.

2. Реляционная модель данных

Реляционные модели широко используются при построении баз данных.

Реляционная модель данных была разработана Э.Коддом в 1969 - 1979 годах во время его работы в IBM (E.F.Codd. "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM, 13, № 6 (June 1970))* . На формирование современного представления о реляционной модели большое влияние оказали результаты практической реализации реляционных систем управления базами данных, включающих средства представления данных на физическом уровне, средства логического представления данных и поддержки реляционной модели данных, а также средства доступа к данным и манипулирования данными на пользовательском и административном уровнях. В процессе эволюции были введены ограничения на структурирование и тип хранимых данных, разработаны способы реализации полной реляционной модели и создан инструментарий для манипулирования данными, не зависящими от их физического представления.

2.1. Понятие модели данных

Модель данных можно определить как средство описания логического представления физических данных. Модель данных задает некоторый базовый набор понятий и свойств, которым должны удовлетворять все базы данных этой модели.

Согласно [4], модель данных состоит из трех основных компонент.

1. Структура данных.
2. Допустимые операции, выполняемые на структуре данных и составляющие основу языка данных рассматриваемой модели.
3. Ограничения для контроля целостности.

Практически аналогично Дейт [3] определил реляционную модель данных, как состоящую из трех частей: структурной части, манипуляционной части и целостной части.

Структурная часть модели определяет, что единственной структурой данных является нормализованное n -арное отношение.

Манипуляционная часть модели определяет два фундаментальных механизма манипулирования данными - реляционная алгебра и реляционное исчисление.

Целостная часть модели определяет требования целостности сущностей и целостности ссылок.

В следующих разделах достаточно подробно рассматриваются три составные части реляционной модели данных.

* Существует русский перевод данной статьи: Кодд Е.Ф. Реляционная модель данных для больших совместно используемых банков данных. СУБД N1, 1995.

2.2. Реляционная структура данных

Единственной структурой данных реляционной модели является нормализованное n -арное отношение. Термин “отношение” используется здесь в общепринятом математическом смысле, как подмножество декартова произведения.

Пусть D_1, D_2, \dots, D_n - произвольные (не обязательно различные) конечные множества.

Декартовым произведением этих множеств $D_1 \times D_2 \times \dots \times D_n$ называется множество элементов вида $d = \langle d_1, d_2, \dots, d_n \rangle$, где $d_i \in D_i, i = 1, 2, \dots, n$.

Элементы декартова произведения называются *кортежами*. Число компонент кортежа n называется *степенью* или *арностью* кортежа.

Рассмотрим пример. Пусть первое множество D_1 состоит из двух элементов, $D_1 = \{s_1, s_2\}$, второе множество D_2 состоит из трех элементов $D_2 = \{p_1, p_2, p_3\}$. Тогда декартово произведение этих множеств есть множество $D_1 \times D_2 = \{\langle s_1 p_1 \rangle, \langle s_1 p_2 \rangle, \langle s_1 p_3 \rangle, \langle s_2 p_1 \rangle, \langle s_2 p_2 \rangle, \langle s_2 p_3 \rangle\}$, состоящее из $6 = 2 \cdot 3$ элементов.

Данный пример можно проиллюстрировать рисунком:

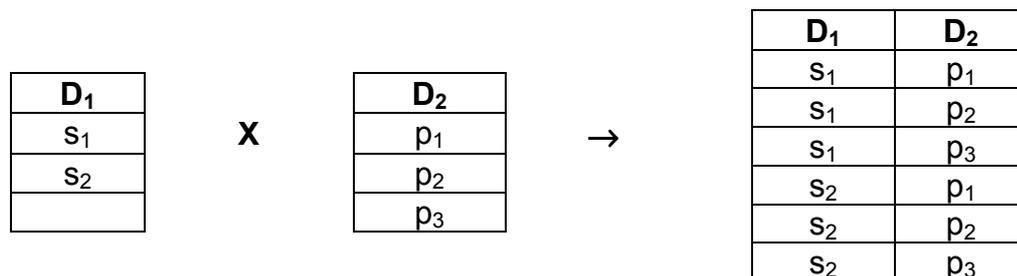


Рис.2.1. Пример декартова произведения

Декартово произведение позволяет получить все возможные комбинации элементов исходных множеств.

Отношением R , определенным на множествах D_1, D_2, \dots, D_n , называется подмножество декартова произведения $D_1 \times D_2 \times \dots \times D_n$. Множества D_1, D_2, \dots, D_n называются *доменами* отношения R .

Степень или арность кортежей (в рассматриваемом случае n) определяет *степень* или *арность* отношения. Отношения арности 1 - унарные, арности 2 - бинарные, арности 3 - тернарные, арности k - k -арные.

Число кортежей в отношении R называется *кардинальным числом* или *мощностью* отношения. Кардинальное число отношения может изменяться, в отличие от его степени.

Порядок кортежей в отношении несуществен. Напротив, домены должны быть упорядочены внутри отношения (отношение есть множество кортежей, где j -й элемент каждого кортежа взят из их j -го домена).

Отношения удобно представлять в форме таблиц, где каждая строка есть кортеж, а каждый столбец - *атрибут*, определенный на некотором домене.

Данный неформальный подход к понятию отношения дает более привычную для разработчиков и пользователей форму представления, где реляционная база данных представляет собой конечный набор таблиц.

Наименьшей единицей данных в реляционной модели является отдельное значение данных. Такие значения рассматриваются как атомарные, т.е. более не детализируемые в рамках данной модели. При описании реляционной модели базы данных в качестве доменов будем рассматривать множества таких значений одного и того же типа. Таким образом, домены представляют собой *пулы значений*, из которых берутся фактические значения, появляющиеся в атрибутах (столбцах) таблицы. В этом смысле домен можно рассматривать, как синоним термина “область определения” для столбца таблицы. Если атрибут определен на некотором домене, то значения этого атрибута должны удовлетворять условиям, наложенным на данный домен.

Несколько атрибутов отношения могут быть определены на одном и том же домене. Если значения двух атрибутов (столбцов) берутся из одного и того же домена, то можно определить отношение сравнения двух этих атрибутов. Если, с другой стороны, два атрибута берутся из разных доменов, то сравнения, с использованием этих атрибутов, вероятно, лишены смысла.

Понятие домена является для реляционной системы концептуальным, поскольку сама система ориентирована на определенный уровень поддержки этого понятия, начиная от способа описания домена (фиксированные типы, перечисляемые, именование и локализация описаний) и связывания с атрибутами и вплоть до функционального использования в реляционной системе для контроля значений данных на уровне операций (внесение данных в базу, контроль полей экранных форм). Условия, которые мы накладываем на домен, могут явным образом храниться в базе и использоваться при присваивании значения столбцу, и, в этом смысле, понятие домена поддерживается ядром реляционной СУБД.

2.3. Реляционная алгебра

Реляционная алгебра (от английского слова *relation* - отношение) базируется на классической теории множеств и в ее основе лежит совокупность операций над отношениями. Кодд [1] определил множество таких операций и показал,

что это множество обладает реляционной полнотой, т.е. обеспечивает такую же селективную мощность как и реляционное исчисление.

Реляционная алгебра содержит две группы операций.

Первая группа операций. Поскольку отношения являются множествами, то к ним применимы следующие обычные теоретико-множественные операции:

- объединение отношений;
- пересечение отношений;
- разность (вычитание) отношений;
- декартово произведение отношений.

Вторая группа операций содержит следующие специальные реляционные операции:

- селекция (ограничение) отношения;
- проекция отношений;
- соединение отношений;
- деление отношений.

Заметим, что данный набор операций несколько расширен относительно варианта первоначально предложенного Коддом [1].

Отметим также, что реляционная алгебра является замкнутой относительно понятия отношения, т.е. исходными операндами и результатами операций являются отношения.

Рассмотрим операции реляционной алгебры.

2.3.1. Теоретико-множественные операции

Для всех теоретико-множественных операций, кроме декартова произведения, отношения-операнды должны быть совместимы по объединению, т.е. должны быть одной и той же степени (арности). Более того, в заголовках обоих отношений должен содержаться один и тот же набор атрибутов, и одноименные атрибуты должны быть определены на одних и тех же доменах.

1. *Объединение отношений* $R = S_1 \cup S_2$

Результатом выполнения операции объединения двух отношений S_1 и S_2 является отношение R , включающее все кортежи, входящие хотя бы в одно из отношений-операндов.

Пример.

к	л	м
у	ф	х
а	б	в

ж	з	и
а	б	в
о	п	р
э	ю	я

к	л	м
у	ф	х
а	б	в
ж	з	и
о	п	р
э	ю	я

Рис.2.2. Пример операции объединения отношений

2. Пересечение отношений $R = S_1 \cap S_2$

Результатом выполнения операции пересечения двух отношений S_1 и S_2 является отношение R , включающее все кортежи, входящие в оба отношения-операнды.

Пример.

к	л	м
у	ф	х
а	б	в

ж	з	и
а	б	в
о	п	р
э	ю	я

а	б	в
---	---	---

Рис.2.3. Пример операции пересечения отношений

3. Разность (вычитание) отношений $R = S_1 - S_2$

Результатом выполнения операции разности (вычитания) двух отношений S_1 и S_2 является отношение R , включающее все кортежи, входящие в отношение S_1 -первый операнд, и не входящие в отношение S_2 -второй операнд.

Пример.

к	л	м
у	ф	х
а	б	в

ж	з	и
а	б	в
о	п	р
э	ю	я

к	л	м
у	ф	х

Рис.2.4. Пример операции вычитания отношений

4. Декартово произведение отношений $R = S_1 \times S_2$

Отличительной особенностью данной операции от предыдущих является то, что операнды могут представлять из себя отношения, построенные по разным схемам (разной степени). Если отношение S_1 имеет арность k_1 , а отношение S_2 имеет арность k_2 , то декартовым произведением отношений S_1 и S_2 является множество кортежей арности $(k_1 + k_2)$. Причем первые k_1 элементов образуют кортеж их отношений S_1 , а последние k_2 элементов - из отношения S_2 .

Пример.

S_1			S_2		$R = S_1 \times S_2$				
к	л	м	о	п	к	л	м	о	п
у	ф	х	э	ю	к	л	м	э	ю
а	б	в			у	ф	х	о	п
					у	ф	х	э	ю
					а	б	в	о	п
					а	б	в	э	ю

Рис.2.5. Пример операции декартова произведения отношений

2.3.2. Специальные реляционные операции

1. Селекция (ограничение) отношения

Селекция (ограничение) R отношения S по формуле F есть подмножество всех кортежей для которых истинна формула F :

$$R = \sigma_F(S),$$

где F - формула, образованная:

1. операндами, являющимися номерами столбцов;
2. логическими операторами \wedge (И), \vee (ИЛИ), \neg (НЕ);
3. арифметическими бинарными отношениями сравнения: $<$, $=$, $>$, \leq , \neq , \geq

Операция селекции имеет одно отношение на входе и одно отношение на выходе. Результирующее отношение состоит из подмножества кортежей исходного отношения.

Пример.

к	л	м
у	ф	х
а	б	в

S

к	л	м
а	б	в

$R_1 = \sigma_{1=k \vee 1=a}(S)$

у	ф	х
---	---	---

$R_2 = \sigma_{3=x}(S)$

Рис.2.6. Примеры операции селекции отношения

2. Проекция отношений

Операция взятия проекции позволяет получить такое подмножество исходного отношения, которое получается выбором заданных атрибутов с последующим исключением, избыточных кортежей-дубликатов.

Операция требует наличия двух операндов - проецируемого отношения S и списка имен атрибутов, входящих в заголовок отношения. Таким образом, проекция позволяет получать вертикальное подмножество заданного отношения, путем выборки заданных столбцов и компоновки их в указанном порядке:

$$R = \pi_{i_1, i_2, \dots, i_n}(S)$$

где i_1, i_2, \dots, i_n - номера столбцов отношения S.

Пример.

к	л	м
у	ф	х
а	б	в

S

к	л
у	ф
а	б

$R_1 = \pi_{1,2}(S)$

м	к
х	у
в	а

$R_2 = \pi_{3,1}(S)$

Рис.2.7. Пример операции проекции отношений

3. Соединение отношений

Пусть θ - арифметический оператор сравнения; n - арность отношения S_1 ; m - арность отношения S_2 , i, j - номера (имена) столбцов в отношениях S_1 и S_2 соответственно.

Соединением R отношений S_1 и S_2 называется множество всех кортежей r таких, что r является конкатенацией (сцеплением) какого-либо кортежа s_1 из S_1 и какого-либо кортежа s_2 из S_2 с условием, что выражение $i \theta j$ истинно:

$$R = S_1 \underset{i\theta j}{\bowtie} S_2 = \sigma_{i\theta (n+j)}(S_1 \times S_2)$$

Как видно из определения, операция соединения имеет сходство с декартовым произведением. Однако при соединении в результирующее отношение включаются только кортежи, удовлетворяющие определенному соотношению между атрибутами соединения соответствующих отношений.

Пример.

S_1			S_2	
А	Б	В	Г	Д
к	л	м	л	а
у	ф	х	о	п
а	б	в		

$$R = S_1 \underset{B=\Gamma}{\bowtie} S_2$$

к	л	м	л	а
---	---	---	---	---

Рис.2.8. Пример операции соединения отношений

В работе [4] отмечены следующие варианты операции соединения:

- *Эквисоединение.* Если θ - арифметический оператор равенства, то операцию соединения называют эквисоединением. Из определения следует, что результат эквисоединения должен содержать хотя бы два идентичных атрибута.
- *Естественное соединение.* Если присутствуют хотя бы два равных атрибута и один из них исключается (что можно сделать с помощью проекции) результат называется *естественным соединением*. Операция естественного соединения используется для восстановления сложных отношений, декомпозированных по требованиям нормализации.

- *Композиция.* Это соединение отличается от естественного тем, что из результирующего отношения удаляются оба атрибута соединения. Поэтому степень результирующего отношения на две единицы меньше суммы степеней отношений.

4. Деление отношений

В простейшей форме операция деления делит отношение степени два (делимое) на отношение степени один (делитель) и продуцирует отношение степени один (частное). Пусть делимое S_1 имеет атрибуты А и Б, а делитель S_2 - атрибут А. Результатом деления S_1 на S_2 является отношение R с единственным атрибутом Б таким, что каждое \bar{b} этого атрибута $S.\bar{b}$ появляется как значение $S_1.Б$ и пара значений (a, \bar{b}) входит в S_1 для всех значений a , входящих в S_2 .

Пример.

S_1				S_2	
П	Л	О	В	я	м
Э	Ф	я	М	О	В
В	К	О	В		
П	Л	я	М		
В	К	я	М		
П	Л	К	В		
				$R = S_1 / S_2$	
				П	Л
				В	К

Рис.2.9. Пример операции деления отношений

2.4. Ограничения целостности

Неотъемлемой частью реляционной модели является требования целостности базы данных. Важной функцией любой СУБД является обеспечение поддержки базы данных в согласованном состоянии. Каждое обновление информации должно удовлетворять структурным и семантическим ограничениям схемы базы данных. Обеспечение согласованности данных является особенно важным при многопользовательском режиме доступа к базе данных.

Рассмотрим два базовых требования целостности реляционных баз данных:

- требование целостности сущностей;
- требование целостности по ссылкам.

Требование целостности сущностей. В реляционной модели каждому типу объектов предметной области соответствует одно или несколько отношений. Экземплярам объектов соответствуют кортежи отношений. Каждый экземпляр

объекта должен быть отличим от любого другого. Следовательно каждый кортеж отношения должен быть уникальным, т.е. в любом отношении должен существовать атрибут (или совокупность атрибутов), значения которого для каждого кортежа различны. Это требование выполняется автоматически при выполнении базовых свойств отношений. Поскольку отношение - это множество, а множества по определению не содержат совпадающих элементов, то никакие два кортежа в произвольный момент времени не могут совпадать. Таким образом полный набор значений всех атрибутов однозначно идентифицирует запись в таблице. Минимальный набор атрибутов отношения, обладающих тем же свойством называется *первичным ключом*. Все значения атрибутов, участвующих в первичном ключе должны быть полностью определены.

Требование целостности по ссылкам. Экземпляры сложных сущностей реального мира могут представляться в реляционной модели базы данных несколькими кортежами нескольких отношений. Если отношение R_2 связано с отношением R_1 (базовое отношение, для которого определен первичный ключ), то отношение R_2 обязано иметь *внешний ключ*, представляющий собой атрибут (или комбинацию атрибутов), значения которых обязательно должны совпадать со значениями первичного ключа R_1 , причем внешний ключ и соответствующий ему первичный ключ должны быть определены на одних и тех же доменах.

Правило целостности по ссылкам гласит: если базовое отношение R_2 имеет внешний ключ F , соответствующий первичному ключу K отношения R_1 , то каждое значение F должно быть:

- либо равным какому-либо значению K ;
- либо быть полностью неопределенным.

Значения атрибутов внешнего ключа либо совпадают с соответствующим первичным ключом, либо полностью не определены, причем возможность последнего определяется конкретной предметной областью, оперирующей данными сущностями.

При проектировании базы в зависимости от предметной задачи необходимо определить:

- Может ли внешний ключ принимать неопределенные значения.
- Что должно происходить при попытке удаления кортежа отношения, на который есть ссылки другого отношения (операция отвергается, обе сущности удаляются).
- Как обновляются внешние ключи (внешний ключ принимает неопределенное значение).

Если ядро СУБД не поддерживает целостность реляционной модели, эта поддержка осуществляется на прикладном уровне программированием

соответствующих процедур, реализуемых либо непосредственно в программном коде, либо через механизм хранимых процедур и триггеров.

2.5. Нормализация отношений

В реляционных базах данных схема содержит как структурную, так и семантическую информацию. Структурная информация связана с объявлением отношений. Семантическая информация выражается множеством функциональных зависимостей между атрибутами отношений.

Некоторые функциональные зависимости могут приводить к возникновению трудностей (аномалий) при модификации баз данных. Процедура устранения нежелательных функциональных зависимостей (а, следовательно, и связанных с ними аномалий) составляет суть процесса нормализации.

Нормализация - это пошаговый обратимый процесс замены данной схемы (или совокупности отношений) базы данных другой схемой, в которой отношения имеют более простую и регулярную структуру [4].

В теории нормальных форм определены различные нормальные формы, которые ограничивают типы допустимых функциональных зависимостей отношения. Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет заданному набору ограничений.

Коддом первоначально [1] выделено три нормальных формы - первая (1НФ), вторая (2НФ) и третья (3НФ) и описан механизм, позволяющий любое отношение преобразовать к 3НФ.

В работе [2] Коддом предложена усиленная третья нормальная форма, или нормальная форма Бойса-Кодда (БКНФ). Четвертая нормальная форма (4НФ) определена Фейджином в работе [7] и в работах [8], [9] определены другие нормальные формы.

Рассмотрим три нормальные формы, предложенные Коддом. Эти нормальные формы ограничивают зависимость непервичных атрибутов от ключей.

Уровень нормализации отношения зависит от его семантики и не может быть однозначно определен из данных, содержащихся в текущий момент в базе данных. Семантика базы данных должна быть задана с помощью функциональных зависимостей. Понятие *функциональной зависимости* является фундаментальным в теории реляционных баз данных. Поэтому, прежде чем рассмотреть три первые нормальные формы, дадим несколько определений.

Функциональная зависимость. В отношении R атрибут Y функционально зависит от атрибута X тогда и только тогда, когда в любой момент времени каждому значению X соответствует в точности одно значение Y.

$$R.X \rightarrow R.Y$$

Пример.

Товар	Имя_Поставщика	Адр_Поставщика	Кол_товара	Цена_товара
Компьютер	“Электрон”	Нахимова, 47	200	6200000
Ризограф	“Протон”	Гагарина, 12	12	90000000
Принтер	“Электрон”	Нахимова, 47	100	1500000
Ксерокс	“Сигма”	Садовая, 128	50	5000000

Рис.2.10. Пример функциональной зависимости

В примере атрибут Адр_Поставщика функционально зависит от атрибута Имя_Поставщика.

$$\text{Имя_Поставщика} \rightarrow \text{Адр_Поставщика}$$

Полная функциональная зависимость. Функциональная зависимость $R.X \rightarrow R.Y$ называется полной, если атрибут Y зависит от всей группы атрибутов X и не зависит функционально от любой ее части (подмножества).

Транзитивная функциональная зависимость. Функциональная зависимость $R.X \rightarrow R.Y$ называется транзитивной, если существует такой атрибут Z, что имеются функциональные зависимости:

$$R.X \rightarrow R.Z \text{ и } R.Z \rightarrow R.Y$$

Первая нормальная форма (1НФ). Отношение R называется нормализованным или приведенным к *первой нормальной форме*, если значения всех его атрибутов простые (атомарные), т.е. значение атрибута не должно быть множеством или повторяющейся группой.

Пример.

Отношение S₁

Номер_сотрудника	ФИО_Сотрудника	Аудитория	Телефон	Дети	
				Имя	Возраст

31	Петров Т.Н.	326	217	Таня Саша	13 6
44	Смирнов Ю.А.	412	258	Игорь	9
51	Павлов И.С.	412	258	Олег Таня	2 6

Рис.2.11. Пример ненормализованного отношения

В отношении S_1 первичным ключом является атрибут **Номер_сотрудника**. Отношение S_1 содержит один сложный атрибут – **Дети**. Для приведения отношения S_1 к 1НФ необходимо разбить сложный атрибут **Дети** на два простых атрибута - **Имя** и **Возраст** и введем атрибут **Имя** в состав первичного ключа. В итоге получим нормализованное отношение S_2 , представленное на рисунке 2.12.

Отношение S_2

Номер_сотрудника	Имя	Возраст	ФИО_сотрудника	Аудитория	Телефон
31	Таня	13	Петров Т.Н.	326	217
31	Саша	6	Петров Т.Н.	326	217
44	Игорь	9	Смирнов Ю.А.	412	258
51	Олег	2	Павлов И.С.	412	258
51	Таня	6	Павлов И.С.	412	258

Рис.2.12. Пример нормализованного отношения (1НФ)

В отношении S_2 первичным ключом являются атрибуты **Номер_сотрудника** и **Имя**.

Отношение, приведенное к 1НФ, характеризуется как минимум двумя аномалиями – аномалией включения и аномалией модификации. Рассмотрим эти аномалии на примере отношения S_2 .

Аномалия включения. Невозможно вставить в отношение кортеж, описывающий сотрудника, не имеющего детей, т.к. первичный ключ не может содержать неопределенное значение.

Аномалия модификации. В общем случае одни и те же значения атрибутов **ФИО_сотрудника**, **Аудитория**, **Телефон** неоднократно дублируются в отношении S_2 . Эта избыточность вызывает проблемы модификации обновления. Так, если сотрудник сменит кафедру, то необходимо будет просмотреть отношение S_2 для нахождения всех кортежей, содержащих информацию о данном сотруднике. Т.о., изменение значения атрибута одного объекта влечет необходимость изменений в нескольких кортежах одновременно: в противном случае база данных окажется несогласованной.

Вторая нормальная форма (2НФ). Отношение R находится во *второй нормальной форме*, если оно находится в 1НФ, и каждый неключевой атрибут функционально полно зависит от первичного ключа.

Отношение, которое находится в 1НФ и не находится во 2НФ, всегда может быть преобразовано в эквивалентную совокупность отношений, находящихся во 2НФ [3]. Преобразование заключается в замене исходного отношения соответствующими проекциями. Совокупность полученных отношений эквивалентна первоначальному отношению в том смысле, что исходное отношение может быть восстановлено путем соответствующей операции соединения полученных отношений. Т.о. в процессе преобразования отсутствуют потери информации.

Пример. Преобразуем отношение S₂ (представленное на рис.2.12.) ко 2НФ путем декомпозиции этого отношения по атрибуту **Номер_сотрудника**. Получим два отношения S₃ и S₄, отвечающих требованиям 2НФ (рис.2.13.).

Отношение S₃

Номер_сотрудника	Имя	Возраст
31	Таня	13
31	Саша	6
44	Игорь	9
51	Олег	2
51	Таня	6

Отношение S₄

Номер_сотрудника	ФИО_сотрудника	Аудитория	Телефон
31	Петров Т.Н.	326	217
44	Смирнов Ю.А.	412	258
51	Павлов И.С.	412	258

Рис.2.13. Пример отношений во 2НФ

Рассмотрим отношение S₄. Его атрибуты **Номер_сотрудника**, **Аудитория** и **Телефон** находятся в транзитивной зависимости.

Номер_сотрудника → **Аудитория** → **Телефон**

Хранение в отношении атрибутов, находящихся в транзитивной зависимости от ключа, порождает ряд трудностей. Рассмотрим их на примере атрибута **Телефон** [6]. Номер телефона характеризует аудиторию, что приводит к многократному дублированию сведений о телефоне для всех сотрудников, работающих в данной аудитории. Это приведет к аномалии модификации при смене номера телефона.

Третья нормальная форма (3НФ). Отношение R находится в *третьей нормальной форме*, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Для приведения отношения S_4 к 3НФ проведем декомпозицию схемы этого отношения.

Декомпозицией схемы отношения $R = \{A_1, A_2, \dots, A_n\}$ называется замена схемы совокупностью схем $\rho = \{R_1, R_2, \dots, R_k\}$ подмножеств, таких, что $R_1 \cup R_2 \cup \dots \cup R_k = R = \{A_1, A_2, \dots, A_n\}$ [11]. При этом не требуется, чтобы схемы R_i были непересекающимися.

Получим два отношения S_5 и S_6 (рис.2.14), находящихся в 3НФ и свободных от аномалий транзитивной зависимости.

Отношение S_5

Номер_сотрудника	ФИО_сотрудника	Аудитория
31	Петров Т.Н.	326
44	Смирнов Ю.А.	412
51	Павлов И.С.	412

Отношение S_6

Аудитория	Телефон
326	217
412	258

Рис.2.14. Пример отношений в 3НФ

На практике приведение отношений к 3НФ часто оказывается достаточным и процесс нормализации, как правило, заканчивается [5]. В тех случаях, когда процесс нормализации необходимо продолжить можно использовать источники [3-5, 7-9, 11] для знакомства с механизмами нормализации более высоких форм.

Как видно из примеров, в процессе нормализации число отношений в схеме БД увеличивается. При проектировании промышленных баз данных число нормализованных отношений в базе может достигать нескольких сотен и даже тысяч. Устранение избыточности информации достигается дорогой ценой, т.к. нормализованные базы данных, представляющие совокупности большого количества отношений, намного сложнее в построении и понимании, чем ненормализованные. Процитируем Д.Васкевича [10]: «Интеллект людей останавливается перед крепостью из 20-30 табличных соединений, а результат создания этих больших соединений остается под большим вопросом, несмотря на значительный рост быстродействия компьютеров». Большое количество нормализованных отношений порождает проблемы поддержания целостности базы данных. Кроме

того, процесс нормализации затрудняет описание семантики предметной области для представления смысла данных. Поэтому степень нормализации базы данных определяется проектировщиком в каждом случае отдельно, путем нахождения компромисса между достоинствами и трудностями работы с нормализованными базами данных.

3. Описание лабораторной работы

Порядок проведения работы

1. Изучение теоретической части.
2. Ответ на контрольные вопросы.
3. Получение у преподавателя варианта задания.
4. Выполнение лабораторной работы.
5. Оформление и защита отчета.

Контрольные вопросы

1. Дайте определение степени и кардинального числа отношения. Приведите примеры отношений различной степени, но с одинаковым кардинальным числом.
2. Какие операции над множествами относятся к специальным реляционным операциям? Дайте определения этих операций и приведите примеры.
3. Какие требования предъявляются к операндам следующих теоретико-множественных операций: объединение, пересечение, разность?
4. Сформулируйте два базовых требования ограничения целостности реляционных баз данных. Приведите пример нарушения требования целостности по ссылкам.
5. В чем заключается суть процесса нормализации реляционных отношений?
6. Дайте определения 1НФ, 2НФ, 3НФ.

Пример варианта индивидуального задания

1. Заданы следующие отношения:

Отношение S_1

Отношение S_2

Отношение S_3

о	у	р	с	т
а	м	и	я	э
л	н	о	с	т
г	ж	е	д	о
с	т	у	ф	х

ц	о	п	р	е
а	л	о	ю	р
о	у	р	с	т
г	ж	е	д	о

ч	у	р
г	о	л

Задание: Используя исходные отношения S_1 , S_2 и S_3 построить следующие отношения R :

- a) $R = S_1 \cup S_2$;
- b) $R = S_1 \times S_3$;
- c) $R = \pi_3(S_1)$;
- d) $R = S_1 \underset{5=1}{\succ\prec} S_2$.

2. Задано отношение:

Номер_рейса	Пункт_отправления	Пункт_назначения	Расписание	
			День	Время
512	Благовещенск	Иркутск	Понед.	11.15
			Вторник	11.15
			Пятница	14.35
46	Благовещенск	Москва	Понед.	9.20
			Среда	14.15
			Суббота	8.50

Задание: Нормализовать данное отношение.

- 3. Задано отношение: **Хранение (Фирма, Склад, Объем)**, которое хранит информацию о фирмах, получающих товары со складов, и объемы этих складов. В отношении имеются функциональные зависимости:

Фирма \rightarrow Склад и Склад \rightarrow Объем

Задание: Привести исходное отношение к 3НФ.

Л и т е р а т у р а

1. Кодд Е.Ф. Реляционная модель данных для больших совместно используемых банков данных. СУБД N1, 1995.
2. Codd E.F. Recent Investigation in Relational Data Base Systems. – In: Information Processing-74, North-Holland, 1974, p. 1017-1021.
3. Дейт К. Введение в системы баз данных. - М.: Наука, 1980. - 464 с.
4. Озкарахан Э. Машины баз данных и управление базами данными. - М.: Мир, 1989. - 696 с.
5. Кузнецов С.Д. Введение в СУБД. СУБД 1995, №№ 1-4, 1996 №№ 1-
6. Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. - М.: Фининсы и статистика, 1989. - 351 с.
7. Fagin R. Multivalued Dependencies and a New Normal Forms for Relational Databases. - IBM Research Report RJ 1812 (July 1976).
8. Fagin R. A Normal Form for Relational Databases That is Based of Domains and Keys. – ACM Trans. On Database Systems. 1981, vol. 6, № 3, p. 387-415.
9. Delobel C. An Overview of the Relational Data Theory. – In: Information Processing-80, North-Holland, 1980, p. 413-426.
10. Васкевич Д. Стратегии Клиент/Сервер. – Киев. Диалектика, 1996. – 384 с.
11. Вейнеров О.М., Самохвалов Э.Н. Проектирование баз данных САПР. – М.: Высшая школа, 1990. – 114 с.
12. Четвериков В.Н., Ревунков Г.И., Самохвалов Э.Н. Базы и банки данных. – М.: Высшая школа, 1987. – 248 с.

С о д е р ж а н и е

1. Введение	3
2. Реляционная модель данных	4
2.1. Понятие модели данных	4
2.2. Реляционная структура данных	5
2.3. Реляционная алгебра	6
2.3.1. Теоретико-множественные операции	7
2.3.2. Специальные реляционные операции	9
2.4. Ограничения целостности	12
2.5. Нормализация отношений	14
3. Описание лабораторной работы	19
Литература	21

Реляционная модель данных

**Методическое руководство к лабораторным работам
для студентов специальности
«Автоматизированные системы обработки
информации и управления» – 22.02**

ЛР от

Составитель: А.Д.Плутенко

**Рецензенты: кафедра математического анализа и алгебры АмГУ
(зав. кафедрой А.В.Миронов)**

Редактор:

Сдано в печать

Формат 60x84 1/16. Бум. типогр. 2.

Печать офсетная. Печ. л. 1,5. Тираж 75 экз.

Зак. . Цена свободная.