

4. РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

4.1. Основные понятия реляционной модели данных

Принципы реляционной модели данных были впервые предложены Э.Ф. Коддом в 1970 г. Кодд, математик по образованию, внес недостоверную точность и строгость в область управления базами данных. Эти идеи стали общепринятыми и оказали существенное влияние на все аспекты технологии баз данных и другие смежные области.

Реляционная модель данных основана на математическом понятии **отношения**, физическим представлением которого является таблица. Рассмотрим сначала каждый термин реляционной модели неформально, а затем представим формальные определения. На Рис.9 представлено отношение «**Поставщик**», отражающее информацию о поставщиках и их характеристике.

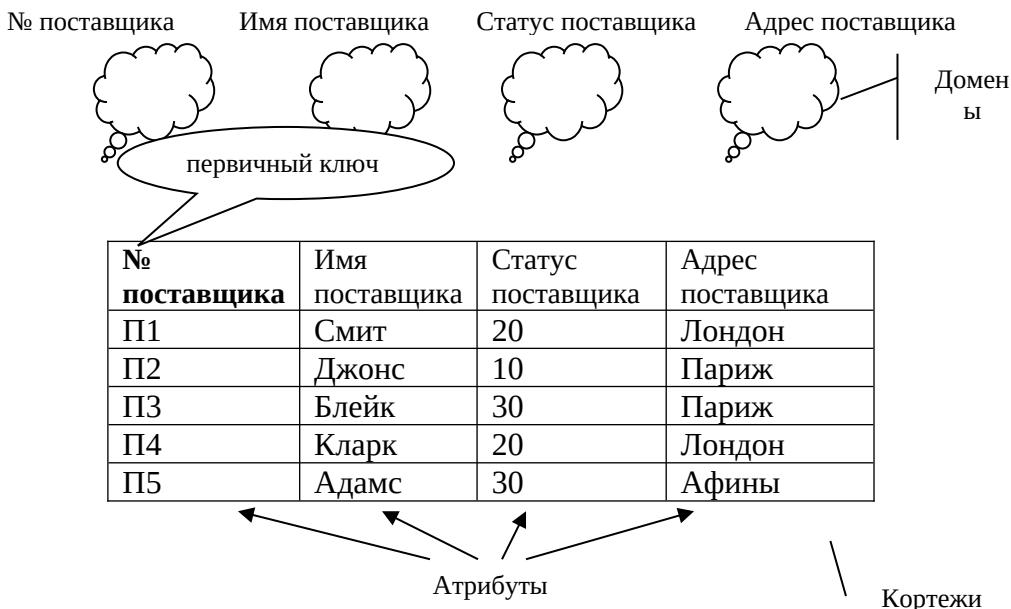


Рис. 9. Отношение «Поставщик»

Итак, **отношение** «Поставщик» представлено в виде таблицы. **Кортеж** соответствует строке этой таблицы, а **атрибут** – столбцу. Количество кортежей называется **кардинальным числом**, а количество атрибутов – **степенью** отношения. **Первичный ключ** – уникальный идентификатор этой таблицы, т. е. такой столбец, что в любой момент времени не существует двух строк, содержащих одинаковое значение в этом столбце. **Домен** – общая совокупность значений, из которых берутся настоящие значения для определенных атрибутов отношения. **Реляционная база данных** – набор нормализованных отношений.

Домен – это не что иное как *тип данных*, определяемый пользователем или системой, подобно типам INTEGER (целый) или CHAR (символьный). С понятием типа данных связаны во-первых, множество допустимых для типа принимаемых значений, во-вторых, набор допустимых операторов, корректно применяемых для этого типа. Домен определяется как именованное множество скалярных (атомарных, неделимых) значений одного типа. Роль доменов в реляционной модели

заключается в том, что домены ограничивают сравнения, т.е. если значения атрибутов взяты из одного и того же домена, тогда сравнения, соединения и другие операции, допустимые для этого типа будут иметь смысл, т.к. сравнивается подобный атрибут с подобным. И наоборот, если значения двух атрибутов взяты из разных доменов, тогда сравнения и другие операции скорее всего не будут иметь смысла (например, бессмысленно сравнивать количество деталей и их вес). Полноценная поддержка механизма доменов характерна далеко не для всех СУБД, хотя это не уменьшает их роли в теории реляционных моделей данных. Дадим теперь более строгое, математическое определение отношения.

Отношение R, определенное на множестве n доменов D₁, D₂, ...D_n (не обязательно различных), содержит две части: заголовок и тело.

1. Заголовок содержит фиксированное множество атрибутов вида A_i:D_i, где A_i – имена атрибутов отношения, а D_i – соответствующие имена доменов (i=1,...n).

2. Тело содержит множество кортежей, каждый из которых представляет собой множество пар A_i:v_{ij}, (i=1,...,n; j=1,...m), где A_i – имя атрибута, а v_{ij} – соответствующее значение этого атрибута в j-том кортеже.

Значения n и m называют соответственно степенью и кардинальным числом отношения.

Из определения следует, что отношение и таблица – не одно и то же. Отношение – в соответствии с определением, абстрактный вид объекта, а таблица – приемлемое, удобное изображение этого объекта. Следует также отметить, что под отношением фактически понимается переменная- отношение, изменяющаяся со временем (со временем в отношение «Поставщик» будут добавляться строки о новых поставщиках, или удаляться, изменяться данные о старых). Однако все возможные значения отношения будут иметь одинаковые заголовки (неизменную структуру).

У отношений есть определенные свойства, имеющие важное значение.

- ◆ Отсутствие одинаковых кортежей. Это следует из того факта, что тело отношения – математическое множество кортежей, а множество, по определению, не может содержать одинаковых элементов.
- ◆ Отсутствие упорядочения кортежей. Это свойство следует из того факта, что тело отношения – множество, а простые множества не упорядочены в математике.
- ◆ Отсутствие упорядочения атрибутов. Данное свойство следует из того факта, что заголовок отношения также определен как множество атрибутов.
- ◆ Каждый кортеж содержит ровно одно значение для каждого атрибута. Это свойство является следствием того, что все лежащие в его основе домены содержат только атомарные, скалярные, неделимые значения.

Заголовок отношения можно считать *предикатом*, а любой кортеж – истинным высказыванием, полученным из предиката в результате подстановки значения соответствующего типа вместо параметров в этом предикате. *Допущение замкнутости мира* гласит: если какой-то другой

допустимый кортеж не содержится в теле отношения, то можно предположить, что соответствующее ему утверждение ложно. Т. е. тело отношение содержит все такие и только такие кортежи, которым соответствует истинное утверждения.

4.2. Целостность реляционной модели

Большинство баз данных подчиняется многим *правилам целостности*, например, для отношения «Поставщик» справедливы правила:

- ◆ номера поставщиков должны быть заданы в виде Ппппп, где пппп может принимать значение до четырех значащих цифр;
- ◆ значение статуса поставщика должно быть в диапазоне от 1 до 100;
- ◆ если город поставщика – Лондон, то статус поставщика равен 20 и т.д.

Любое данное правило является *специфическим* для базы данных в том смысле, что оно применяется к одной конкретной базе данных. Однако в дополнение к специфическим правилам имеются *общие* особые правила целостности. Эти особые правила применяются к любой базе данных, а не только к некоторым определенным. Эти два особых правила относятся к первичным (потенциальным) ключам и к внешним ключам.

Первичный ключ отношения – это просто уникальный идентификатор для некоторого отношения. Однако первичный ключ на самом деле является частным случаем более общего понятия – *потенциального* ключа.

Пусть R - некоторое отношение. Тогда **потенциальный ключ** K для R – это подмножество множества атрибутов R, обладающее следующими свойствами:

- 1) свойством уникальности. Нет двух различных кортежей в отношении R с одинаковым значением K.
- 2) Свойством неизбыточности. Никакое из подмножеств K не обладает свойством уникальности.

Очевидно, *каждое* отношение имеет, по крайней мере, один потенциальный ключ, т.к. не содержит одинаковых кортежей, а поскольку кортежи уникальны, то, по крайней мере, комбинация всех атрибутов обладает свойством уникальности.

Потенциальный ключ, состоящий более чем из одного атрибута, называется **составным**, а состоящий из одного атрибута – **простым**. Базовое отношение может иметь больше одного потенциального ключа. В таком случае традиционно один из потенциальных ключей должен быть выбран в качестве **первичного** ключа, а остальные потенциальные ключи будут называться **альтернативными** ключами. Каждое базовое отношение должно всегда иметь **первичный** ключ.

Причина важности потенциальных ключей заключается в том, что они обеспечивают основной **механизм адресации на уровне кортежей** в реляционной системе. Следовательно, единственный гарантированный системой способ точно указать на какой-нибудь кортеж – это указать значение некоторого потенциального ключа.

Внешний ключ – это атрибут или множество атрибутов внутри отношения, которое соответствует потенциальному ключу некоторого отношения. Более точное определение внешнего ключа.

Пусть R2 – базовое отношение. Тогда **внешний** ключ, например, FK в отношении R2 – это подмножество множества атрибутов R2, такое что:

- ◆ существует базовое отношение R1 (R1 и R2 не обязательно различны) с потенциальным ключом СК;
- ◆ каждое значение FK в текущем значении R2 всегда совпадает со значением СК некоторого кортежа в текущем значении R1.

Значение внешнего ключа представлено **ссылкой** к кортежу, содержащему соответствующее значение потенциального ключа (**ссыпочный кортеж** или целевой кортеж). Проблема обеспечения того, что база данных не включает никаких неверных значений внешних ключей известна как проблема **ссыпочной целостности**.

Ограничение, по которому значения данного внешнего ключа должны быть адекватны значениям соответствующих потенциальных ключей называют **ссыпочным ограничением**. Отношение, которое содержит внешний ключ, называется **ссылающимся отношением**, а отношение, которое содержит соответствующий потенциальный ключ – **ссыпочным отношением**. Существующие в базе данных ссыпочные ограничения можно представить в виде **ссыпочной диаграммы**:



Каждая стрелка обозначает внешний ключ в отношении, из которого эта стрелка выходит; этот ключ специфически ссылается на первичный ключ отношения, на который указывает стрелка.

Вместе с понятием внешнего ключа реляционная модель включает следующее правило **ссыпочной целостности**: база данных не должна содержать несогласованных значений внешних ключей, т.е. таких значений, для которых не существует отвечающего значения соответствующего потенциального ключа в целевом отношении.

Для каждого внешнего ключа необходимо ответить на два вопроса:

1. Что должно случиться при попытке удалить объект ссылки внешнего ключа? В общем случае существует по крайней мере две возможности:

- ◆ ограничить – «ограничить» операцию удаления до момента, когда не будет существовать объекта ссылки;
- ◆ каскадировать – «каскадировать», распространить операцию удаления, удаляя все объекты ссылки.

2. Что должно случиться при попытке обновить потенциальный ключ, на который ссылается внешний ключ? Как и для удаления, существует по меньшей мере две возможности:

- ◆ ограничить – «ограничить» операцию обновления до момента, когда не будет существовать объекта ссылки;
- ◆ каскадировать – «каскадировать», распространить операцию обновления, обновляя все объекты ссылки.

Описанные **правила внешних ключей**, хотя и не являются частью реляционной модели как таковой, но необходимы на практике в качестве

основы механизма для поддержания ссылочной целостности. Для каждого из этих правил следует установить опцию «ограничить» или «каскадировать».

При представлении хранимых данных сталкиваются с проблемой представления отсутствующей информацией, реально встречающейся в жизни (например, информация по какой-то причине не известна, временно отсутствует). Коддом было предложено использовать специальные маркеры, называемые *null-значениями*. Null-значения – это не то же самое, что и пробелы или числовые нули. В общем, для каждого атрибута может быть разрешено или не разрешено содержать null-значения, это зависит от определения данного атрибута. **Правило целостности объектов** включается в реляционную модель и гласит: ни один элемент *первичного ключа* базового отношения *не может быть null-значением*. Правило **ссылочной целостности** расширяется до того факта, что ни один элемент *внешнего ключа* отношения *не может быть null-значением*.

4.3. Правила формирования реляционной модели данных по ER-диаграмме

Путем использования ER-диаграммы возможно осуществить формирование набора предварительных отношений с указанием предполагаемого первичного ключа для каждого отношения. Правила формирования отношений следующие.

1) Каждая сущность преобразуется в отношение. Именем отношения является имя сущности, а набор элементов данных сущности становится набором атрибутов отношения. Идентифицирующие атрибуты сущности становятся первичным ключом отношения.

2) Особенность реляционной модели данных состоит в том, что связь между сущностями формирует также отношение. При этом именуется отношение тем же именем, что и связь ER-диаграммы. Атрибутный состав отношения определяется, как правило, набором идентифицирующих атрибутов каждой участвующей в связи сущности, а также собственными атрибутами связи. Ключ нового отношения может быть составным (набор ключей участвующих в связи сущностей), либо новым.

3) Связи между сильными и слабыми сущностями, а также связи подтип-супертип моделируются отношением, в котором атрибутный состав определяется набором ключевых атрибутов сильной сущности и всеми атрибутами слабой сущности. Именуется отношение также, как именуется слабая сущность и может иметь ключ – комбинацию ключевых атрибутов обеих сущностей.

4.4. Пример построения реляционной модели данных «Факультет»

Построим по ER-диаграмме, приведенной на Рис.8, реляционную модель данных системы «Факультет» в соответствии с описанной в п.4.3. методикой.

Получим следующую систему отношений (ключевые атрибуты выделены):

- 1) **СТУДЕНТ** (Код студента, ФИО студента, Фото студента, Дата рождения студента, Родители, Домашний адрес студента, Дата поступления, Примечания студента, № зачетной книжки, Дата анкеты студента)
- 2) **ФАКУЛЬТЕТ** (Код факультета, ВУЗ, Название факультета)
- 3) **КАФЕДРА** (Код кафедры, Название кафедры, Специализация кафедры, Учебный год)
- 4) **ГРУППА** (Код группы, Код факультета, Учебный год, Семестр, Группа, Курс, Шифр специальности)
- 5) **ПРЕПОДАВАТЕЛЬ** (Код преподавателя, ФИО преподавателя, Адрес преподавателя, Дата рождения преподавателя, Ученая степень и звание, Дата анкеты преподавателя, Примечания преподавателя)
- 6) **ПРЕДМЕТ** (Код предмета, Наименование предмета, Вид предмета, Описание предмета, Объем в часах по виду)
- 7) **СОСТАВ ГРУППЫ** (Код группы, Код студента)
- 8) **СОСТАВ ФАКУЛЬТЕТА** (Код факультета, Код кафедры)
- 9) **СОСТАВ КАФЕДРЫ** (Код кафедры, Код преподавателя, Должность)
- 10) **НАГРУЗКА ПРЕПОДАВАТЕЛЯ** (Код преподавателя, Код предмета, Код факультета, Учебный год, Семестр, Объем в часах преподавателя)
- 11) **РАСПИСАНИЕ** (Код группы, Код предмета, Код преподавателя, День недели, Номер пары, Номер аудитории)
- 12) **ЭКЗАМЕН** (Код студента, Код предмета, Код преподавателя, Дата экзамена, Оценка)

Данная схема отношений является первоначальной, не окончательной и в нуждается в процедуре нормализации, которая и будет произведена позже.

4.5. Реляционная алгебра. Синтаксис основных операторов реляционной алгебры

Перечень операций:

- Проекция
- Естественное соединение
- ϑ - соединение
- Декартово произведение
- Селекция
- Булевы операции
- Частное
- Переименование атрибутов

Проекция:

Проекция это набор унарных операций выбора подмножества X столбцов отношений $\text{proj}_x(r)$, где R схема отношения r и $X \subseteq R$ – набор столбцов.

Пример:

$r :$

C_1	C_2	C_3
a_1	1	21.03.2019

$\text{proj}_{[C_1, C_2]}(r)$

C_1	C_2
a_1	1

a_2	2	07.08.2018
a_3	3	10.02.2020
a_2	4	30.12.2020
a_3	3	13.09.2021

a_2	2
a_3	3
a_2	4

Свойство: если $Y \subseteq X \subseteq R$, то $\text{proj}_y(\text{proj}_x(r)) = \text{proj}_y(r)$

Из определения операции проекции и общих положений реляционной алгебры следует, что картежи в результирующем отношении **не будут** повторяться.

Естественное соединение:

Пусть отношения r_1 и r_2 имеют схемы

$$R_1(A_1, \dots, A_k, B_1, \dots, B_n)$$

и

$$R_2(A_1, \dots, A_k, C_1, \dots, C_m).$$

Тогда **естественное соединение** (*join*) отношений r_1 и r_2 есть отношение r_3 со схемой:

$$R_3(A_1, \dots, A_k, B_1, \dots, B_n, C_1, \dots, C_m)$$

в котором каждая запись (экземпляр) получена конкатенацией каждой записи из r_1 с теми записями из r_2 , у которых совпадают значения в общих атрибутах A_1, \dots, A_k .

Пример:

r_1 :	r_2	$\text{join}_{\{=A\}}(r_1, r_2)$																																	
<table border="1"> <tr><td>B</td><td>A</td></tr> <tr><td>b_1</td><td>3</td></tr> <tr><td>b_2</td><td>7</td></tr> <tr><td>b_3</td><td>4</td></tr> <tr><td>b_4</td><td>4</td></tr> </table>	B	A	b_1	3	b_2	7	b_3	4	b_4	4	<table border="1"> <tr><td>A</td><td>C</td></tr> <tr><td>3</td><td>c_1</td></tr> <tr><td>3</td><td>c_2</td></tr> <tr><td>4</td><td>c_3</td></tr> </table>	A	C	3	c_1	3	c_2	4	c_3	<table border="1"> <tr><td>A</td><td>B</td><td>C</td></tr> <tr><td>3</td><td>b_1</td><td>c_1</td></tr> <tr><td>3</td><td>b_1</td><td>c_2</td></tr> <tr><td>4</td><td>b_3</td><td>c_3</td></tr> <tr><td>4</td><td>b_4</td><td>c_3</td></tr> </table>	A	B	C	3	b_1	c_1	3	b_1	c_2	4	b_3	c_3	4	b_4	c_3
B	A																																		
b_1	3																																		
b_2	7																																		
b_3	4																																		
b_4	4																																		
A	C																																		
3	c_1																																		
3	c_2																																		
4	c_3																																		
A	B	C																																	
3	b_1	c_1																																	
3	b_1	c_2																																	
4	b_3	c_3																																	
4	b_4	c_3																																	

Обозначения: $\text{join}(r_1, r_2)$ или $\text{join}_{\{=A\}}(r_1, r_2)$ или $r_1 \text{join} r_2$

ϑ – соединение:

Определение: Пусть даны отношения r_1, r_2 со схемами $R_1(A_1, \dots, A_k, B_1, \dots, B_l), R_2(C_1, \dots, C_m, D_1, \dots, D_n)$, соответственно, и ϑ – оператор сравнения на группах атрибутов A и C . Тогда ϑ – **соединение** отношений r_1 и r_2 есть отношение r_3 со схемой $R_3(A_1, \dots, A_k, B_1, \dots, B_l, C_1, \dots, C_m, D_1, \dots, D_n)$, полученной объединением атрибутов схем R_1 и R_2 **без повторения**. Записи r_3 получаются конкатенацией тех записей из r_1 и r_2 , у которых значения группы столбцов A в r_1 и группы столбцов C в r_2 находятся в отношении ϑ (удовлетворяют ϑ).

Обозначение: $\text{join}_{A\vartheta C}(r_1, r_2)$

Замечание: Очевидно, если ϑ есть равенство “=” и $A \equiv C$ получим естественное соединение со схемой $R_3(A_1, \dots, A_k, B_1, \dots, B_l, D_1, \dots, D_n)$.

Пример:

Settlement:

SetName	Countr	Population
---------	--------	------------

SettlementType:

NameType	LowPop	HighPop
----------	--------	---------

	у	
Краснодар	РФ	1000000
Москва	РФ	12000000

Деревня	0	50000
Малый город	50000	250000
Средний город	250000	500000
Большой город	500000	2000000
Мегаполис	2000000	1000000000

Зададим условие:

$$\vartheta = \text{Population} < \text{HighPop} \wedge \text{Population} \geq \text{LowPop}.$$

В итоге получим:

$$\text{join}_{\text{Population} < \text{HighPop} \wedge \text{Population} \geq \text{LowPop}}(\text{Settlement}, \text{SettlementType})$$

SettName	Country	Population	NameType	LowPop	HighPop
Краснодар	РФ	1000000	Большой город	500000	2000000
Москва	РФ	12000000	Мегаполис	2000000	1000000000

Декартово произведение:

Определение: Декартовым произведением отношений r и s ностей k_r и k_s , с непересекающимися множествами атрибутов, соответственно, R и S , называется отношение $t = r \times s$ арности $k_r + k_s$, состоящее из кортежей, первые k_r компонентов которых есть кортежи из r , а последние k_s компонентов выбираются из s . Иначе говоря, кортежи t образованы конкатенацией каждого кортежа из r с каждым кортежем из s . Поэтому, если в текущем состоянии r и s имеют n_r и n_s кортежей, то в t их $n_r \times n_s$.

Замечание: В одном отношении недопустим повтор имен. Поэтому, в частности, не существует декартов квадрат. При соединении отношений с одноименными атрибутами некоторые из них могут быть переименованы исходя из семантики данных и соединения.

Пример:

r :

A	B
1	3
2	7
3	4

s :

C	D	E
c_1	09.09.2019	Предмет
c_1	27.03.2017	Студент

$r \times s$

A	B	C	D	E
1	3	c_1	09.09.2019	Предмет
1	3	c_1	27.03.2017	Студент
2	7	c_1	09.09.2019	Предмет
2	7	c_1	27.03.2017	Студент

3	4	c_1	09.09.2019	Предмет
3	4	c_1	27.03.2017	Студент

Селекция:

Определение: Пусть F – формула, образованная:

- операндами в виде констант или имен столбцов (номеров столбцов);
- операторами сравнения $\textcolor{red}{i}, =, >, \leq, \geq, \neq;$;
- логическими операторами $\bigwedge(\textcolor{red}{i}), \bigvee(\textcolor{red}{i}), \neg(\text{not}())$.

Тогда результат селекции $\text{sel}_F(r)$ есть множество кортежей t из r , для которых формула F истинна.

Пример: $r:$

A	B
1	3
2	7
3	4

 $s:$

C	D	E
c_1	09.09.2019	Предмет
c_1	27.03.2017	Студент

 $\text{sel}_{D > 1.01.2018 \wedge \text{not}(E = \text{Студент})}(r \times s)$

A	B	C	D	E
1	3	c_1	09.09.2019	Предмет
2	7	c_1	09.09.2019	Предмет
3	4	c_1	09.09.2019	Предмет

Булевые операции:

Два отношения r_1 и r_2 с одной и той же схемой R могут рассматриваться как подмножества множества всех возможных кортежей в схеме R . Поэтому к ним применимы булевые операции: $\cup, \cap, -$.

Замечание: Булевые операции могут применяться к совместимым отношениям, у которых атрибуты попарно имеют совместимые типы и общую семантику.

Это расширение понятия односортных отношений.

Примеры: $r_1:$

A	B	C
01.01.2017	Отлично	Диф.Ур.
05.06.2018	Отлично	БД
20.11.2019	Хорошо	Диф.Ур.

 $r_2:$

A	B	C
01.01.2017	Отлично	Диф.Ур.
05.06.2018	Отлично	Диф.Ур.
20.11.2019	Хорошо	Диф.Ур.

 $r_1 \cup r_2:$

A	B	C
01.01.2017	Отлично	Диф.Ур.

 $r_1 \cap r_2:$

A	B	C
01.01.2017	Отлично	Диф.Ур.

05.06.201 8	Отлично	БД
20.11.201 9	Хорошо	Диф.Ур.
05.06.201 8	Отлично	Диф.Ур.

20.11.2019	Хорошо	Диф.Ур.
------------	--------	---------

$r_1 - r_2$		
A	B	C
05.06.2018	Отлично	БД

Частное:

Определение: Пусть даны:

- отношение r с арностью k_r и схемой R

и

- отношение s с арностью $k_s < k_r$ и схемой S , которая не пуста, то есть $S \neq \emptyset$, и является собственным подмножеством схемы R , то есть $S \subset R$.

Тогда **частным** называется отношение $r \div s$ арности $k_r - k_s$, которое:

- содержит столбцы отношения r отсутствующие в s ;
- часть записи r включается в $r \div s$ если в r она сцеплена с каждой записью из s .

Пример:

r :

A	B	C	D
01.01.201 7	5	Новиков	БД
01.01.201 7	5	Авакимов	UML
05.06.201 8	4	Авакимов	UML
20.11.201 9	3	Новиков	БД
20.11.201 9	3	Авакимов	UML

s :

C	D
Новиков	БД
Авакимов	UML

$r \div s$

A	B
01.01.2017	5
20.11.2019	3

Переименование атрибутов:

Теоретико-множественные операторы объединение, пересечение и разность требуют, чтобы отношения – операнды были совместимы, то есть относились к элементам одного сорта. Это означает, что отношения отличаются только именами и состояниями. Сигнатуры у них одинаковы, то есть количества атрибутов совпадают и атрибуты попарно совпадают по типам, а в простейшем случае, по именам.

Если же имена отношений и/или атрибутов не совпадают, необходимо установить соответствие между именами отношений или изменить некоторые из имён атрибутов.

В операциях соединений и декартовом произведении может появиться повтор одинаковых атрибутов, что делает невозможным выполнение операции. И здесь переименование может позволить выполнение операции.

Итак, некоторые несовместимые отношения могут стать совместимыми после переименования атрибутов. Поэтому необходимо ввести операцию переименования атрибутов.

Замечание: При более детальном анализе выясняется, что для осмысленного переименования атрибутов необходимо, чтобы парный атрибут имел тот же или родственный смысл.

Синтаксис:

[имя_отношения] **RENAME** список_старых_атрибутов **AS**
список_новых_атрибутов

Пример:

Исходное отношение:

Торговый_агент (№_т_а, имя_т_а, №_рук, офис, процент_ком)

Операция:

Торговый_агент **RENAME** №_т_а, имя_т_а, №_рук, офис, процент_ком
AS ID_ta, name_ta, ID_ta_boss, of_adr, sal_per

Выход:

ТОРГОВЫЙ_АГЕНТ(ID_ta, name_ta, ID_ta_boss, of_adr, sal_per)

4.6. Задания по формированию запросов на языке реляционной алгебры для аудиторной и самостоятельной работы

Для данной схемы отношений написать запросы на языке реляционной алгебры.

Задача 1.

Заданы отношения:

Рабочий

Код_рабочего	Имя_рабочего	Специальность	Почасовая_ставка	Код_руководителя
1235	М. Фарадей	Электрик	12,50	1311
1412	К Немо	Штукатур	13,75	1520
2920	Р. Гаррет	Кровельщик	10,00	2920
3231	П. Мэйсон	Плотник	17,40	3231
1311	Х. Колумб	Электрик	15,50	1311
3001	Д. Барристер	Плотник	8,20	3231
1520	Г. Риковер	Штукатур	11,75	1520

Назначение

Код_рабочего	Код_здания	Дата_начала_работы	Число_дней
1235	312	10.10	5
1412	312	01.10	10
1235	515	17.10	22
2920	460	05.10	18
1412	460	08.12	18
2920	435	28.10	10
2920	210	10.11	15
1520	111	10.10	8
1412	435	15.10	15

Здание

Код_здания	Адрес	Тип	Кол-во этажей	Статус
312	ул.Вязов, 123	Офис	2	2
435	Кленовая ул. 456	Магазин	1	1
515	Дубовая ул. 789	Жилой дом	3	1
210	Березовая ул. 1011	Офис	3	1
111	Осиновая ул. 1213	Офис	4	1
460	Буковая ул. 1415	Склад	3	3

Написать запросы на языке реляционной алгебры:

- 1) найти имена и почасовую оплату рабочих, строящих здание, расположенное на Кленовой ул.456, и работающих не более 10 дней;
- 2) найти здания, на строительстве которых работают Электрики и не работают Штукатуры;
- 3) среди зданий, на строительстве которых работают **все** рабочие, найти имеющие статус более 2;
- 4) найти здания, на строительстве которых работают Штукатуры и не работают Электрики;

- 5) определить общие объемы выполняемой работы всеми рабочими, которыми руководит Х. Колумб, либо теми из них, которые работают над строительством Офисов.
- 6) найти рабочих, строящих более одного здания, одно из которых является Складом;
- 7) получить полную информацию о рабочих, участвующих в строительстве Магазина и Склада. Каковы имена руководителей этих рабочих?
- 8) найти здания с количеством этажей больше 2, на строительстве которых работают Электрики и Штукатуры.

Задача 2.

Дана схема отношений:

ПАЛАТА (**№_ПАЛ**, **ОТДЕЛЕНИЕ**, **ЭТАЖ**)
ПАЦИЕНТ (**№_ПАЦ**, **ИМЯ_П**, **НОМЕР_КОЙКИ**, **ПОЛ**, **ДИАГНОЗ**,
ДАТА_ПОСТУПЛЕНИЯ, **ДАТА_ВЫПИСКИ**)

ВРАЧ (**№_ВРАЧА**, **ИМЯ_В**, **ТЕЛЕФОН**)
ЛЕКАРСТВО (**ЛЕК_НАЗВАНИЕ**, **ПОСТАВЩИК**, **СРОК_ГОДНОСТИ**)

ПАЦИЕНТ_В_ПАЛАТЕ (**№_ПАЛ**, **№_ПАЦ**)

ЛЕЧЕНИЕ (**№_ПАЦ**, **№_ВРАЧА**)

НАЗНАЧЕНИЕ (**№_ПАЦ**, **ЛЕК_НАЗВАНИЕ**)

Написать запросы на языке реляционной алгебры:

- 1) получить список пациентов отделения больницы "кардиология" в период с 16 июня 2001 по 8 августа 2001;
- 2) найти поставщиков лекарств, назначенных пациентам в палатах отделений "кардиология" и "интенсивная терапия" одновременно;
- 3) найти номера палат, в которых не было ни одного пациента с диагнозом "рак" в первом квартале 2001 г.;
- 4) найти палаты, в которых лежат пациенты, лечащиеся у врача Иванова И.И. и не лечащиеся у врача Сидорова С.С.;
- 5) найти имена и телефоны врачей, закрепленных за **всеми** пациентами палат второго этажа;
- 6) получить информацию обо всех женщинах-пациентках больницы (палата, ее местонахождение, лечащий врач);
- 7) найти пациентов, принимающих *аспирин* и *цефазолин* одновременно по назначению врача Иванова И.И.;
- 8) получить полную информацию о пациентах, лежавших в больнице два и более раз.

Задача 3.

Дана схема отношений:

ФАКУЛЬТЕТ (НАЗВАНИЕ, ДЕКАН)

**СТУДЕНТ (№_СТУДЕНТА, ФИО_СТУДЕНТА, ДАТА_РОЖДЕНИЯ,
ПОЛ, ДОМАШНИЙ АДРЕС, ДАТА_ПОСТУПЛЕНИЯ,
ДАТА_ЗАПОЛНЕНИЯ)**

**ГРУППА (НАЗВАНИЕ_ФАКУЛЬТЕТА, КУРС, №_ГРУППЫ,
УЧЕБНЫЙ ГОД, СТАРОСТА)**

СПИСОК_ГРУППЫ (№_ГРУППЫ, № СТУДЕНТА)

ПРЕДМЕТ (№_ПРЕДМЕТА, НАЗВАНИЕ_ПРЕДМЕТА)

**УСПЕВАЕМОСТЬ (№_СТУДЕНТА, №_ПРЕДМЕТА,
ДАТА_ЭКЗАМЕНА, ФИО_ПРЕПОДАВАТЕЛЯ, ОЦЕНКА)**

Написать запросы на языке реляционной алгебры:

- 1) получить список группы факультета прикладной математики, **все** студенты которой сдали экзамены на "хорошо" и "отлично" в 2000/01 уч. году;
- 2) найти всех студентов факультета прикладной математики, которые сдали в 2000 году на "отлично" Информатику и Математический анализ;
- 3) найти всех студентов факультета прикладной математики, которые сдали в 2001 году Информатику и не сдали Алгебру;
- 4) получить информацию о студентах, не сдавших экзамен 21.06.2001 г. Нагорному С.В. Какие еще экзамены не дали эти студенты?
- 5) найти всех "отличников", обучающихся на 1 курсе факультета прикладной математики;
- 6) найти всех неуспевающих юношей, поступивших в вуз в 1997 г. на экономический факультет;
- 7) найти всех студентов факультета прикладной математики, проживавших в период с 1.09.2000 г. по 1.09.2001 г. в г. Новороссийске. В каких группах обучаются эти студенты?
- 8) получить сведения об успеваемости по Математическому анализу в 2000/01 уч. году на факультете прикладной математики в группах, которых преподавал Малыхин К.А.