Тема 3. Основы теории реляционных баз данных

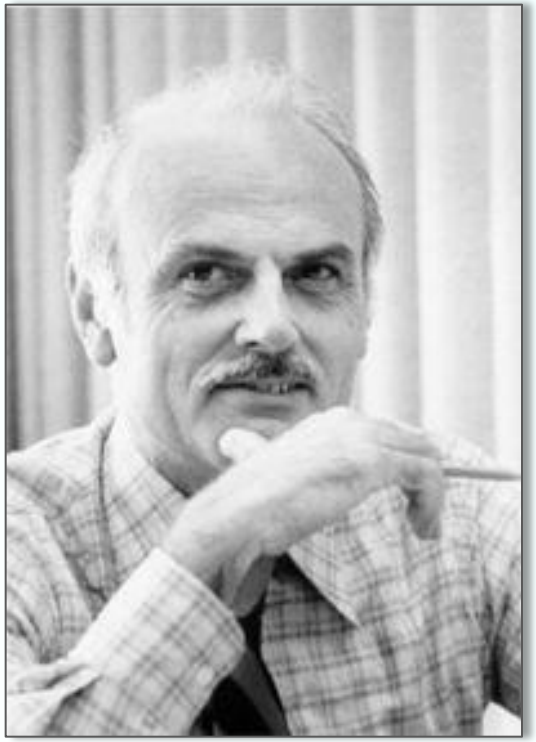
Лекция 3

Вопросы лекции:

1. Основные понятия реляционной модели данных
2. Операции реляционной алгебры
3. Проектирование реляционных БД на основе принципов нормализации

1. Основные понятия реляционной модели данных

История появления реляционной модели данных



Эдгар Франк Кодд (1923 -2003) — британский учёный, работы которого заложили основы теории реляционных баз данных.

В 1970 издал работу «A Relational Model of Data for Large Shared Data Banks», которая считается первой работой по реляционной модели данных.

В начале 80-х опубликовал **«12 правил Кодда»**, описывающие, что должна содержать реляционная СУБД.

Реляционная модель данных

В реляционной модели рассматриваются три принципиальных аспекта данных:

- ✔ структура данных (объекты данных)
- ✔ манипулирование данными (операторы)
- ✔ поддержание целостности данных.

Реляционная модель данных

В реляционной модели рассматриваются три принципиальных аспекта данных:

- **Структурный аспект** — данные в базе данных представляют собой набор отношений.
- **Аспект целостности** — отношения (таблицы) отвечают определенным условиям целостности. РМД поддерживает декларативные ограничения целостности уровня домена (типа данных), уровня отношения и уровня базы данных.
- **Аспект обработки (манипулирования)** — РМД поддерживает операторы манипулирования отношениями (реляционная алгебра, реляционное исчисление).

Реляционные БД

Реляционная база данных — база данных, основанная на реляционной модели данных.

Реляционные СУБД – системы для работы с реляционными БД

Определение (неформальное)

Реляционная база данных (от англ. Relation – отношение) – набор таблиц, связанных между собой по значениям определенных столбцов.

Схема БД – графическое описание логической структуры БД.

При создании информационной системы совокупность отношений позволяет хранить данные об объектах предметной области и моделировать связи между ними.

Реляционные БД

- **Отношение** – двухмерная таблица не содержащая строк-дубликатов
- **Сущность** есть объект любой природы, данные о котором хранятся в базе данных. Данные о сущности хранятся в отношении
- **Запись** – строка (ряд, запись, row, кортеж) таблицы
- **Отношение** – множество кортежей
- **Атрибут** (столбец). Атрибуты представляют собой свойства, характеризующие сущность. В структуре таблицы каждый атрибут именуется и ему соответствует заголовок некоторого столбца таблицы
- **Домен** – множество значений атрибута

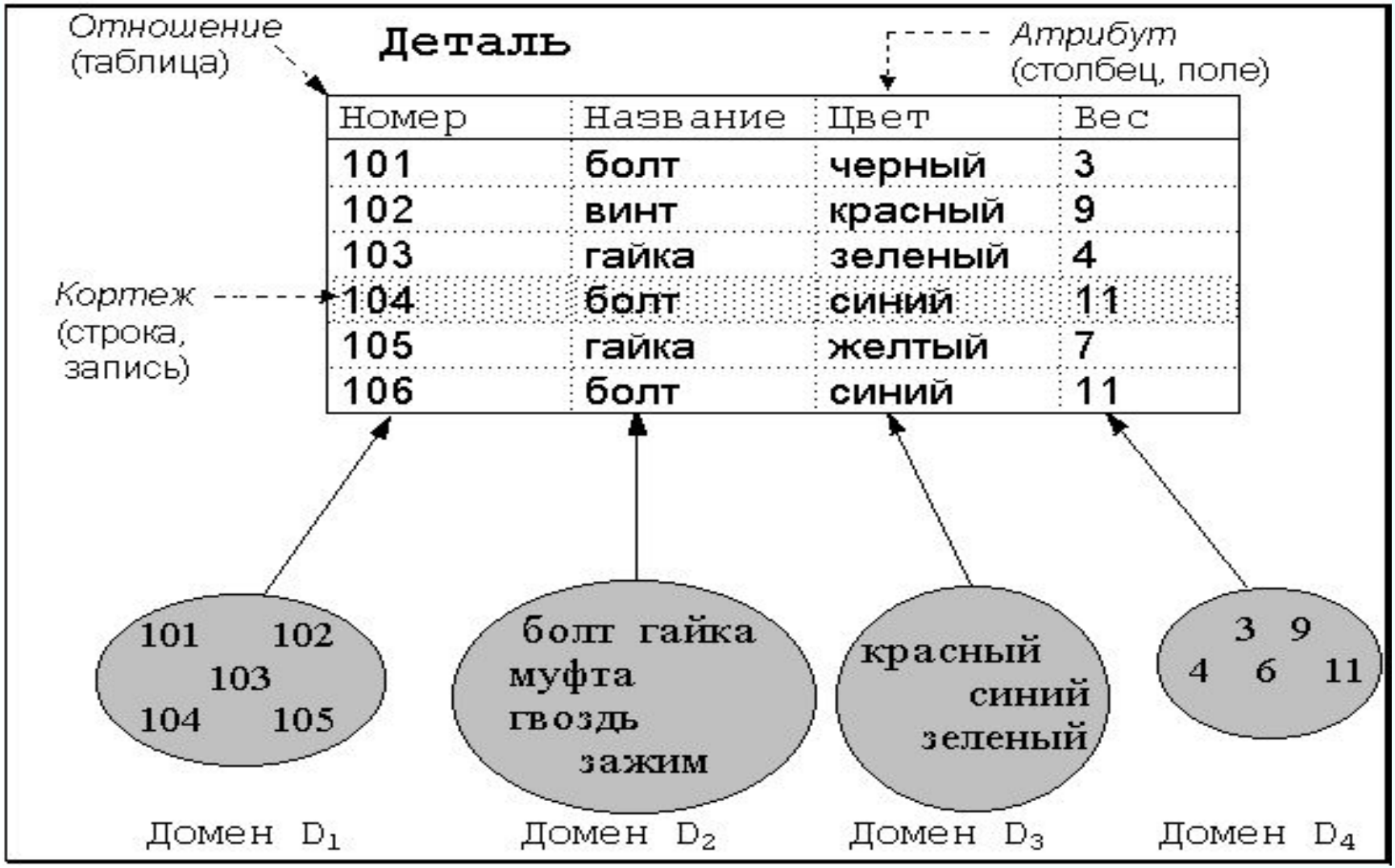
Элементы реляционной модели

Элемент реляционной модели	Форма представления
Отношение	Таблица
Схема отношения	Строка заголовков столбцов таблицы
Кортеж (запись)	Строка таблицы
Сущность	Объект
Атрибут	Заголовок столбца таблицы
Домен	Множество допустимых значений атрибута
Значение атрибута (реквизит)	Значение поля в записи
Первичный ключ	Один или несколько атрибутов
Тип данных	Тип значений элементов таблицы

Пример реляционной модели данных

Представление отношения СОТРУДНИК

Пример реляционной модель данных



Представление отношения ДЕТАЛЬ

Свойства отношений

Отсутствие кортежей-дубликатов

Данное свойство следует из определения отношения как множества кортежей. В классической теории множеств по определению каждое множество состоит из различных элементов.

Свойства отношений

Отсутствие упорядоченности кортежей

Свойство отсутствия упорядоченности кортежей отношения также является следствием определения отношения-экземпляра как множества кортежей. Отсутствие требования к поддержанию порядка на множестве кортежей отношения дает дополнительную гибкость СУБД при хранении баз данных во внешней памяти и при выполнении запросов к базе данных. Это не противоречит тому, что при формулировании запроса к БД, например, на языке SQL можно потребовать сортировки результирующей таблицы в соответствии со значениями некоторых столбцов. Такой результат, вообще говоря, не отношение, а некоторый упорядоченный список кортежей.

Свойства отношений

Атомарность значений атрибутов

Значения всех атрибутов являются атомарными. Это следует из определения домена как потенциального множества значений простого типа данных, т.е. среди значений домена не могут содержаться множества значений (отношения). Принято говорить, что в реляционных базах данных допускаются только нормализованные отношения или отношения, представленные в первой нормальной форме.

Понятие ключа

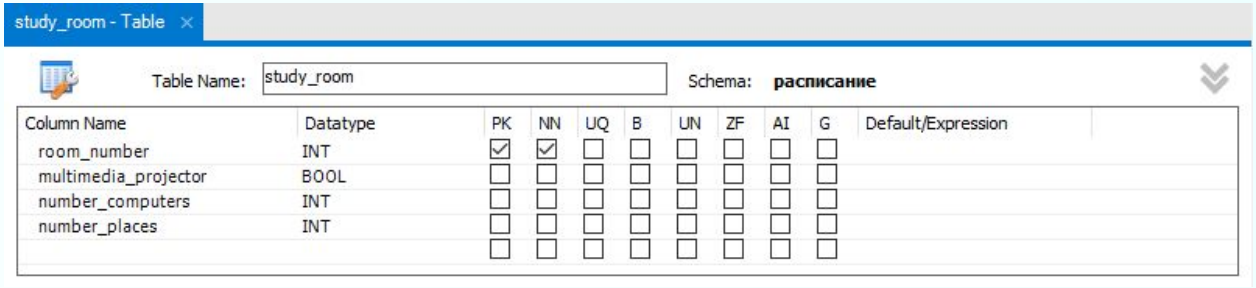
- **Первичный ключ** – атрибут или совокупность атрибутов однозначно идентифицирующих строку отношения.

- Каждая таблица может содержать **только один первичный ключ**.
- Ключ, состоящий из одного атрибута, называется **простым**.

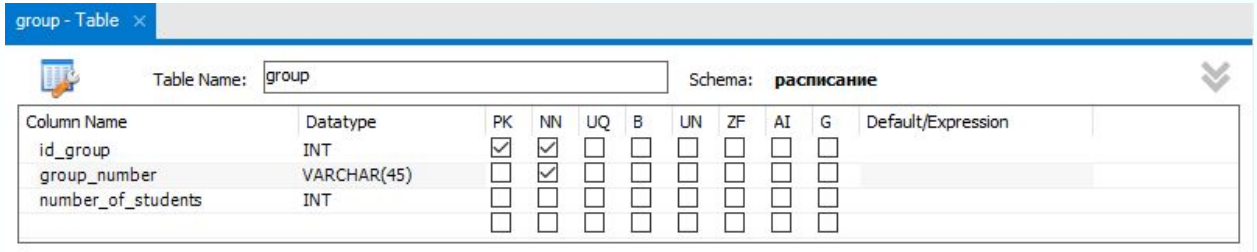
- Ключ, состоящий из нескольких атрибутов, называется **составным**.

Понятие ключа

Естественный ключ – первичный ключ, состоящий из информационных полей таблицы (то есть полей, содержащих полезную информацию об описываемых объектах).



Суррогатный ключ – автоматически сгенерированное поле, никак не связанное с информационным содержанием записи. Обычно в роли СК выступает автоинкрементное поле типа INTEGER.



Понятие ключа

- **Потенциальный ключ** — подмножество атрибутов отношения, удовлетворяющее требованиям **уникальности** и **минимальности**.
- Свойство **уникальности** определяется по всем возможным значениям, то есть следует из внешнего знания о природе и закономерностях данных, которые могут находиться в переменной отношения.
- **Минимальность** (несократимость) означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, удовлетворяющее условию уникальности. Иными словами, если из потенциального ключа убрать любой атрибут, он утратит свойство уникальности.
- В отношении может быть одновременно несколько потенциальных ключей. Один из них может быть выбран в качестве **первичного ключа** отношения, тогда другие потенциальные ключи называют **альтернативными ключами**.

Свойства ключа:

- ✔ Уникальность
- ✔ Неизбыточность
- ✔ Не может содержать пустых значений

Ключи обычно используют для достижения следующих целей:

- ✔ исключения дублирования значений в ключевых атрибутах (остальные атрибуты в расчет не принимаются)
- ✔ упорядочения кортежей
- ✔ ускорения работы с кортежами отношения;
- ✔ организации связывания таблиц.

Понятие ключа

Каждое отношение обязательно имеет комбинацию атрибутов, которая может служить ключом. Ее существование гарантируется тем, что отношение – это множество, которое не содержит одинаковых элементов – кортежей.

Информационный объект

Студент

Ключевое поле

Номер	Фамилия	Имя	Отчество	Дата
16493	Сергеев	Петр	Михайлович	01.01.76
16593	Петрова	Анна	Владимировна	15.03.75
16693	Анохин	Андрей	Борисович	14.04.76

Свойства реляционной таблицы:

- ✔ Все строки таблицы должны быть уникальны, т. е. не может быть строк с одинаковыми первичными ключами.
- ✔ Имена столбцов таблицы должны быть различны, а значения их простыми, т. е. недопустима группа значений в одном столбце одной строки.
- ✔ Все строки одной таблицы должны иметь одну структуру, соответствующую именам и типам столбцов.
- ✔ Порядок размещения строк в таблице может быть произвольным.

Связывание таблиц

- При проектировании БД информацию обычно размещают в нескольких таблицах. Таблицы при этом связаны семантикой информации. В реляционных СУБД для указания связей таблиц производят операцию их связывания.
- Многие СУБД при связывании таблиц автоматически выполняют контроль ссылочной целостности вводимых в базу данных в соответствии с установленными связями, что способствует повышению достоверности хранимой в БД информации.
- Установление связи между таблицами облегчает доступ к данным. Связывание таблиц при выполнении таких операций, как поиск, просмотр, редактирование, выборка и подготовка отчетов, обеспечивает возможность обращения к произвольным полям связанных записей.

Основные виды связи таблиц

- При связывании двух таблиц выделяют основную (родительскую) и дополнительную (подчиненную, дочернюю) таблицы. Логическое связывание таблиц производится с помощью ключа связи.
- Ключ связи, по аналогии с обычным ключом таблицы, состоит из одного или нескольких полей, которые называют полями связи.
- Суть связывания состоит в установлении соответствия полей связи основной и дополнительной таблиц.
- В зависимости от того, как определены поля связи основной и дополнительной таблиц (как соотносятся ключевые поля с полями связи), между двумя таблицами могут устанавливаться следующие четыре основных вида связи: один — один (1:1); один — много (1:М); много — много (М:М или М:N).

Связь

Связь - это логическая ассоциация, устанавливаемая между таблицами БД.

Связь определяет **количество записей данной таблицы, которые могут быть связаны с одной записью другой таблицы.**

Связи бывают следующих типов:

- **один к одному**
- **один ко многим**

- Пример связи один к одному:
  «Страны» - «Столицы»
- Пример связи один ко многим:
  «Страны» - «Города»
- Пример связи многие ко многим:
  «Международные Союзы» - «Страны»

Характеристика видов связей таблиц

Характеристика полей связи по видам	1:1	1:М	М:М
Поля связи основной таблицы	являются ключом	являются ключом	не являются ключом
Поля связи дополнительной таблицы	являются ключом	не являются ключом	не являются ключом

Характеристика связей:

Связь вида 1:1

Связь вида 1:1 образуется в случае, когда все поля связи основной и дополнительной таблиц являются ключевыми. Обеспечивается взаимно-однозначное соответствие записей связываемых таблиц.

На практике связи вида 1:1 используются сравнительно редко, так как хранимую в двух таблицах информацию легко объединить в одну таблицу, которая занимает гораздо меньше места в памяти ЭВМ.

Связь вида 1:M

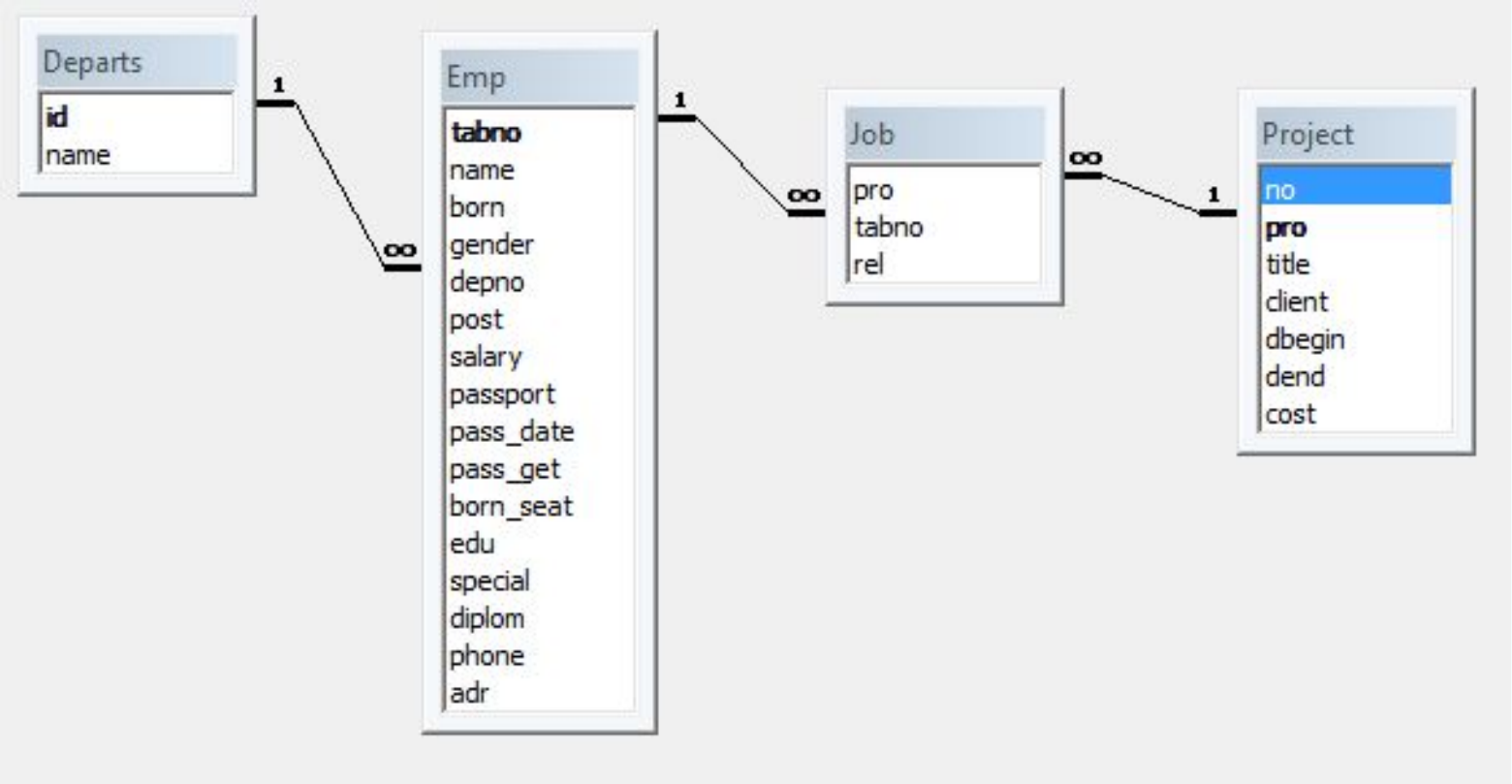
Связь 1:M имеет место в случае, когда одной записи родительской таблицы ответствует несколько записей дочерней таблицы.

Характеристика связей:

Связь вида M:M

Самый общий вид связи M:M возникает в случаях, когда нескольким записям основной таблицы соответствует несколько записей дополнительной таблицы.

Пример БД: «Проектная организация»



Departs – отделы, **Project – проекты,**

Emp – сотрудники, **Job – участие в проектах.**

Связь «один-ко-многим»: Отделы - Сотрудники

Таблица «Сотрудники» (Emp)

Табельный номер	*ФИО сотрудника*	*Отдел*
023	Волкова Елена Павловна	2
113	Белов Сергей Юрьевич	1
101	Рогов Сергей Михайлович	2
056	Панина Анна Алексеевна	1
…	…	…
098	Фролов Юрий Вадимович	9

Таблица «Отделы» (Departs)

Номер отдела	*Название отдела*
1	Информационный отдел
2	Администрация
3	Отдел кадров
…	…
9	Проектный отдел

«Номер отдела» – первичный ключ в таблице «Отделы»

«Отдел» – внешний ключ в таблице «Сотрудники» к таблице «Отделы»

Связь «один-ко-многим»: Отделы - Сотрудники

Таблица «Сотрудники» (Emp)

Табельный номер	*ФИО сотрудника*	*Отдел*
023	Волкова Елена Павловна	2
113	Белов Сергей Юрьевич	1
101	Рогов Сергей Михайлович	2
056	Панина Анна Алексеевна	1
...	...	...
098	Фролов Юрий Вадимович	9

Таблица «Отделы» (Departs)

Номер отдела	*Название отдела*
1	Информационный отдел
2	Администрация
3	Отдел кадров
...	...
9	Проектный отдел

«Номер отдела» – первичный ключ в таблице «Отделы»

«Отдел» – внешний ключ в таблице «Сотрудники» к таблице «Отделы»

В отношении **Сотрудники** имеется не ключевой атрибут «**Отдел**», значения которого являются значениями **первичного ключа** «**Номер отдела**» отношения **Отделы**. Тогда говорят, что атрибут «**Отдел**» отношения **Сотрудники** есть внешний ключ.

Связь «многие-со-многими»: Сотрудники- Проекты

- Связь "**многие-со-многими**" в реляционной модели данных реализуется через «**развязочные таблицы**». При этом связь "**многие-со-многими**" можно заменить двумя связями **«один ко многим»**.
- Для организации связей в развязочную таблицу добавляются внешние ключи (FK).
- Развязочная таблица является дочерней по отношению к связываемым.
- Например: «участие в проектах» (таблица из двух полей: код сотрудника (FK), код проекта (FK)).

Связь «многие-со-многими»: Сотрудники- Проекты

Таблица «Сотрудники»

ФИО	*Номер*
Волкова Е.П.	023
Белов С.Ю.	113
Рогов С.М.	101
Панина А.А.	056
Фролов Ю.В.	098
...	...

Таблица «Участие в проектах»

Участник	*Роль*	*Проект*
113	исполнитель	23/Н
101	руководитель	18-К
056	исполнитель	18-К
101	консультант	09/Р
098	руководитель	23/Н
...	...	...

Таблица «Проекты»

Шифр	*Название проекта*
23/Н	АИС "Налог"-2
18-К	ИПС "Жители"
09/Р	ГИС "Город"

В таблице «Участие»:

«Участник» – внешний ключ к таблице «Сотрудники»

«Проект» – внешний ключ к таблице «Проекты»

Контроль целостности связей

- В РМД наиболее широко используется связь вида 1:M (связь вида 1:1 можно считать ее частным случаем).
- При образовании связи вида 1:M одна запись главной таблицы (главная, родительская запись) оказывается связанной с несколькими записями дополнительной (дополнительные, подчиненные записи).

Контроль целостности связей

- Контроль целостности связей обычно означает анализ содержимого двух таблиц на соблюдение следующих правил:

✔ каждой записи основной таблицы соответствует ноль или более записей дополнительной таблицы;

✔ в дополнительной таблице нет записей, которые не имеют родительских записей в основной таблице;

✔ каждая запись дополнительной таблицы имеет только одну родительскую запись основной таблицы.

Контроль целостности связей

Опишем действие контроля целостности при манипулировании данными в таблицах. Рассмотрим три основные операции над данными двух таблиц:

- ✔ ввод новых записей
- ✔ модификацию записей
- ✔ удаление записей

Контроль целостности связей

При вводе новых записей данные сначала вводятся в основную таблицу, а потом — в дополнительную. Очередность ввода может быть установлена на уровне целых таблиц или отдельных записей (случай одновременного ввода в несколько открытых таблиц).

В процессе заполнения основной таблицы контроль значений полей связи ведется как контроль обычного ключа (на совпадение со значениями тех же полей других записей). Заполнение полей связи дополнительной таблицы контролируется на предмет совпадения со значениями полей связи основой таблицы. Если вновь вводимое значение в поле связи дополнительной таблицы не совпадет ни с одним соответствующим значением в записях основной таблицы, то ввод такого значения должен блокироваться.

Контроль целостности связей

Модификация записей. Изменение содержимого полей связанных записей не относящихся к полям связи происходит обычным образом. Рассмотрим механизм изменения полей связи.

При редактировании полей связи дополнительной таблицы очевидным требованием является то, чтобы новое значение поля связи совпадало с соответствующим значением какой-либо записи основной таблицы. Т. е. дополнительная запись может сменить родителя, но остаться без него не должна.

Редактирование поля связи основной таблицы разумно подчинить одному из следующих правил:

- редактировать записи, у которых нет подчиненных записей. Если есть подчиненные записи, то блокировать модификацию полей связи;
- изменения в полях связи основной записи мгновенно передавать во все поля связи всех записей дополнительной таблицы (каскадное обновление).

Контроль целостности связей

В операциях удаления записей связанных таблиц большую свободу имеют записи дополнительной таблицы. Удаление их должно происходить практически бесконтрольно.

Удаление записей основной таблицы логично подчинить одному из следующих правил:

- удалять можно запись, которая не имеет подчиненных записей;
- запретить (блокировать) удаление записи при наличии подчиненных записей, либо удалять ее вместе со всеми подчиненными записями (каскадное удаление).

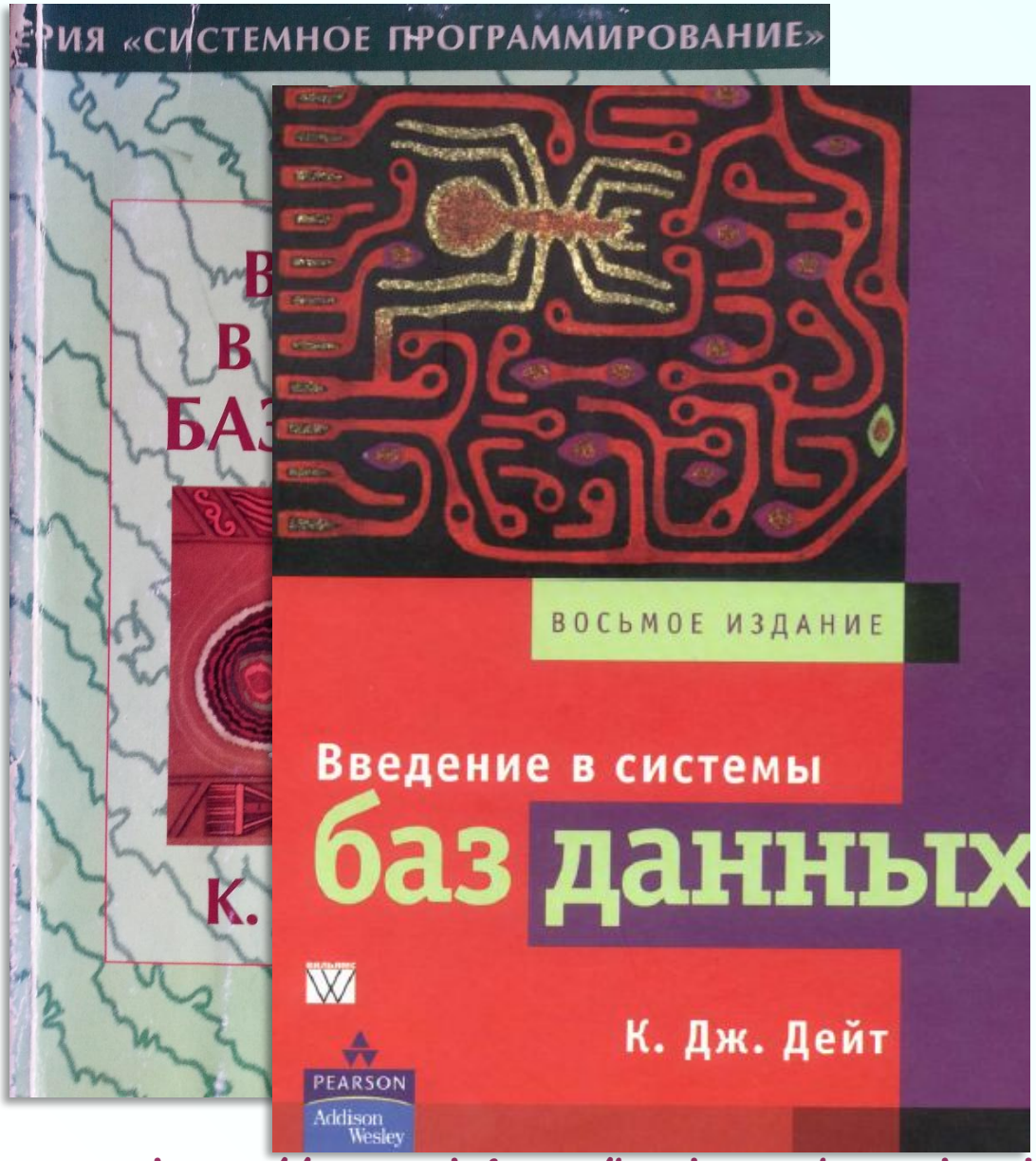
2. Операции реляционной алгебры

Реляционная алгебра

Реляционная алгебра — замкнутая система операций над отношениями в реляционной модели данных. Операции реляционной алгебры также называют реляционными операциями.

Реляционная алгебра — это коллекция операций, которые принимают отношения в качестве операндов и возвращают отношение в качестве результата, т.е. удовлетворяют условию: «отношение на входе – отношение на выходе» (свойство замкнутости).

Реляционная алгебра



Дейт, К. Дж.

Введение в системы баз данных

8-е издание.: Пер. с англ. — М.: Издательский дом "Вильяме", 2005. — 1328 с.

http://www.ph4s.ru/book_pc_bazy.html

Зачем нужна реляционная алгебра

Основная цель алгебры — обеспечить запись реляционных выражений.

Некоторые из возможных применений подобных выражений:

- ✔ **Определение области выборки**, т.е. тех данных, которые должны быть доставлены в результате выполнения операции выборки.
- ✔ **Определение области обновления**, т.е. данных, которые должны быть вставлены, изменены или удалены в результате выполнения операции обновления.
- ✔ **Определение правил поддержки целостности данных**, т.е. некоторых особых требований, которым должна удовлетворять база данных.
- ✔ В целом, выражения реляционной алгебры **служат для символического высокоуровневого представления намерений пользователя**

  Например, рассмотрим следующее выражение («Получить имена поставщиков детали с номером 'P2'»)

  ((SP WHERE P# = P# ('P2')) JOIN S) { SNAME

Реляционная алгебра

Первая версия **реляционной алгебры** алгебры была определена Э. Коддом.

Эта "оригинальная" алгебра включала восемь операций, которые подразделялись на две группы с четырьмя операциями каждая:

- ✔ Традиционные операции с множествами — **объединение, пересечение, разность и декартово произведение** (все они были немного модифицированы с учетом того факта, что их операндами являются именно отношения, а не произвольные множества).
- ✔ Специальные реляционные операции, такие как **сокращение** (известное также под названием выборки), **проекция, соединение и деление**

Описание базы данных, используемой в примерах

Для иллюстрации излагаемого материала будем использовать **базу данных поставщиков и деталей**, которая называется базой данных поставщиков, деталей и проектов. Поставщики (S), детали (P) и проекты (J) однозначно определяются номером поставщика (S#), номером детали (P#) и номером проекта (J#) соответственно. Значение строки SPJ (поставки) следующее: определенный поставщик поставляет определенную деталь для определенного проекта в определенном количестве (причем комбинация значений столбцов S#-P#-J# уникальна для

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Black	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P2	Bolt	Green	17.0	Paris
P3	Screw	Blue	17.0	Rome
P4	Screw	Red	14.0	London
P5	Cam	Blue	12.0	Paris
P6	Cog	Red	19.0	London

J

J#	JNAME	CITY
J1	Sorter	Paris
J2	Display	Rome
J3	OCR	Athens
J4	Console	Athens
J5	RAID	London
J6	EDS	Oslo
J7	Tape	London

SPJ

S#	P#	J#	QTY
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800
S2	P5	J2	100
S3	P3	J1	200
S3	P4	J2	500
S4	P6	J3	300
S4	P6	J7	300
S5	P2	J2	200
S5	P2	J4	100
S5	P5	J5	500
S5	P5	J7	100
S5	P6	J2	200
S5	P1	J4	100
S5	P3	J4	200
S5	P4	J4	800
S5	P5	J4	400
S5	P6	J4	500

Примеры отношений

Поставщики S { S#, SNAME,
STATUS, CITY } PRIMARY KEY { S# }

Поставки P { P#, PNAME, COLOR,
WEIGHT, CITY } PRIMARY KEY { P# }

Поставлено SPJ { S#, P#, J# QTY }
PRIMARY KEY { S#, P#, J# }
FOREIGN KEY { S# } REFERENCES S
FOREIGN KEY { P# } REFERENCES P
FOREIGN KEY { J# } REFERENCES J

Операции реляционной алгебры

Существует пять основных операций РА:

- ✔ селекция
- ✔ проекция
- ✔ декартово произведение
- ✔ объединение
- ✔ разность

и три вспомогательных операции РА, которые могут быть выражены через основные:

- ✔ пересечение

Операции реляционной алгебры

- На операциях реляционной алгебры (РА) основан язык **SQL**.
- Операции РА применяются к **отношениям** и в результате применения операций РА получаются **отношения** (таблицы).
- Различают **унарные** и **бинарные** операции РА: унарные применяются к **одному отношению** (таблице), бинарные – **к двум**.

Унарные операции реляционной алгебры

Проекция (project)

Это унарная операция служит для выбора подмножества атрибутов из отношения R. Она уменьшает *арность* (число атрибутов) отношения и может уменьшить *мощность* (число кортежей) отношения за счёт исключения одинаковых кортежей.

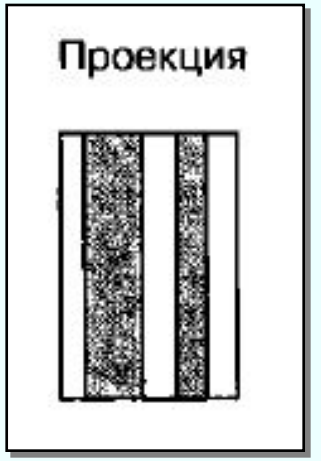
Отношение R

A	B	C
a	b	c
c	a	d
c	b	d

Проекция $\pi_{A,C}(R)$

A	C
a	c
c	d

Операция проекции



Предположим, что отношение A имеет атрибуты X, Y, . . ., Z (и, возможно, другие атрибуты). В таком случае проекция отношения A по атрибутам X, Y, . . . , Z, которая определяется с помощью следующего выражения: A { X, Y, . . . , Z }
является отношением, соответствующим следующим требованиям:

- ✔ Его заголовок формируется из заголовка отношения A путем удаления всех атрибутов, не указанных в множестве { X, Y, . . . , Z }.
- ✔ Тело состоит из всех кортежей {x, y, ..., z}, таких что в отношении A присутствует кортеж со значением x атрибута X, y атрибута Y... и z атрибута Z.

Операция проекции

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Black	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

S {CITY}

CITY
London
Paris
Athens

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P2	Bolt	Green	17.0	Paris
P3	Screw	Blue	17.0	Rome
P4	Screw	Red	14.0	London
P5	Cam	Blue	12.0	Paris
P6	Cog	Red	19.0	London

P {COLOR, CITY}

COLOR	CITY
Red	London
Green	Paris
Blue	Rome
Blue	Paris

В первом примере, несмотря на то что исходное отношение S имеет пять кортежей (и, следовательно, пять значений городов), в результирующем отношении присутствует только три города, поскольку дублирующиеся кортежи исключены.

Операция проекции

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Black	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

(S WHERE CITY = PARIS') {S#}

S#
S2
S3

Унарные операции реляционной алгебры

Селекция (select)

Это унарная операция, результатом которой является подмножество кортежей исходного отношения, соответствующих условиям, которые накладываются на значения определённых атрибутов.

Отношение R

A	B	C
a	b	c
c	a	d
c	b	d

Селекция

$\sigma_{C=d}(R)$

A	B	C
c	a	d
c	b	d

Оператор выборки (сокращения)

- **Выборка (R WHERE f)** отношения **R** по формуле **f** представляет собой новое отношение с таким же заголовком и телом, состоящим из таких кортежей отношения **R**, которые удовлетворяют истинности логического выражения, заданного формулой **f**.
- Для записи формулы используются операнды — **имена атрибутов** (или номера столбцов), **константы, логические операции** (AND — И, OR — ИЛИ, NOT — НЕ), **операции сравнения и скобки**.
- Оператор сокращения по сути позволяет получить **"горизонтальное" подмножество заданного отношения**, т.е. подмножество кортежей заданного отношения, для которых удовлетворяется некоторое указанное условие.

Оператор селекции (сокращения) Примеры

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Black	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

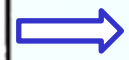
S WHERE CITY ='London'

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P2	Bolt	Green	17.0	Paris
P3	Screw	Blue	17.0	Rome
P4	Screw	Red	14.0	London
P5	Cam	Blue	12.0	Paris
P6	Cog	Red	19.0	London

P WHERE WEIGHT<WEIGHT(14,0)



P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P5	Cam	Blue	12.0	Paris

SPJ

S#	P#	J#	QTY
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800

SP WHERE S#=S#('S6') OR P#=P#('P7')

S#	P#	QTY

Бинарные операции реляционной алгебры

Бинарные операции РА:

- **разносхемные** – применяются к любым двум отношениям.
- **односхемные** – применяются к односхемным отношениям.

  Исходные отношения должны иметь одинаковое количество столбцов одинаковых (или сравнимых) типов.

  Сравнимыми считаются типы, относящиеся к одному и тому же семейству данных (числовые, символьные и пр.).

Разносхемная основная операция РА

Декартово произведение

Это бинарная операция над **разносхемными** отношениями, соответствующая определению декартова произведения для РМД: в результате получается отношение, схема которого включает все атрибуты исходных отношений. Результирующее отношение содержит все возможные комбинации кортежей исходных отношений.

Отношение R

A	B
1	4
2	5
3	6

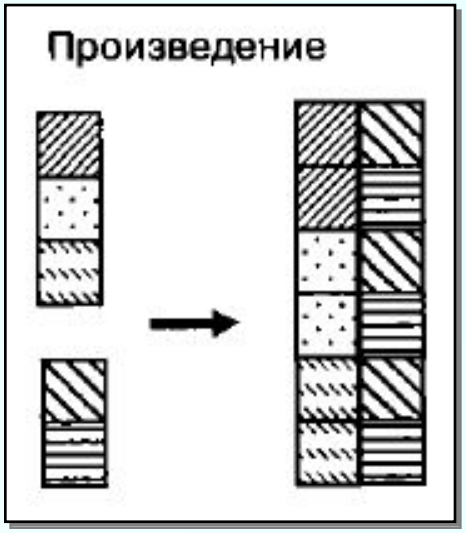
Отношение S

C	D	E
g	h	a
a	b	c

Декартово произведение R×S

A	B	C	D	E
1	4	g	h	a
1	4	a	b	c
2	5	g	h	a
2	5	a	b	c
3	6	g	h	a
3	6	a	b	c

Декартово произведение



- В математике декартовым произведением (или сокращенно произведением) двух множеств является множество всех таких упорядоченных пар, что **в каждой паре первый элемент берется из первого множества, а второй — из второго множества.**
- Поэтому декартово произведение двух отношений, неформально выражаясь, представляет собой множество упорядоченных пар кортежей, но для сохранения **свойства замкнутости** необходимо, чтобы результат содержал кортежи как таковые, а не упорядоченные пары кортежей.

Произведение отношений

Поэтому реляционной версией декартова произведения служит расширенная форма этой операции, в которой каждая упорядоченная пара кортежей заменяется одним кортежем, являющимся объединением двух рассматриваемых кортежей.

Это означает, что если даны следующие кортежи:

{A1, A2, . . . , Am}

и

{B1, B2, ..., Bn}

то теоретико-множественное объединение этих двух кортежей представляет собой приведенный ниже единственный кортеж:

{A1, A2, ..., Am, B1, B2, ..., Bn}

Произведение отношений

- Декартово произведение A TIMES B отношений A и B, не имеющих общих атрибутов, - это отношение, заголовок которого представляет собой (теоретико-множественное) объединение заголовков отношений A и B, а тело состоит из всех кортежей t, таких, что t является (теоретико-множественным) объединением кортежа, принадлежащего к отношению A, и кортежа, принадлежащего к отношению B.

Пример операции декартового произведения

A

S#
S1
S2
S3
S4
S5

B

P#
P1
P2
P3
P4
P5
P6

Декартово произведение (A TIMES B)

S#	P#
S1	P1
S1	P2
S1	P3
S1	P4
S1	P5
S1	P6
..	..

..	..
S2	P1
S2	P2
S2	P3
S2	P4
S2	P5
S2	P6
..	..

..	..
S3	P1
S3	P2
S3	P3
S3	P4
S3	P5
S3	P6
..	..

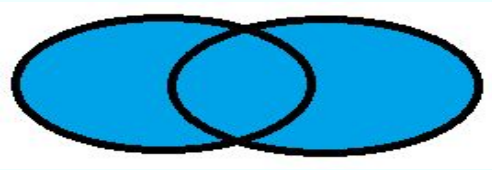
..	..
S4	P1
S4	P2
S4	P3
S4	P4
S4	P5
S4	P6
..	..

..	..
S5	P1
S5	P2
S5	P3
S5	P4
S5	P5
S5	P6

Бинарные односхемные операции РА

Объединение (union).

Объединением двух односхемных отношений R и S называется отношение T = R U S, которое включает в себя все кортежи исходных отношений без повторов.



Отношение R

A	B	C
a	b	c
c	a	d
c	h	c

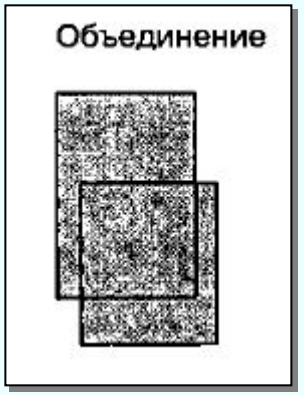
Отношение S

A	B	C
g	h	a
a	b	c
h	d	d

Объединение R U S

A	B	C
a	b	c
c	a	d
c	h	c
g	h	a
h	d	d

Объединение



В математике объединение двух множеств представляет собой множество всех элементов, принадлежащих либо к одному из них, либо к обоим заданным множествам.

Объединение в реляционной алгебре не полностью соответствует общему определению объединения в математике; скорее, оно является объединением особого рода, в котором два входных отношения должны принадлежать к одному типу.

Объединение отношений

Если даны отношения **A** и **B** одного и того же типа, то объединение этих отношений **A UNION B** является отношением того же типа с телом, которое состоит из всех кортежей **t**, **присутствующих в A или B или в обоих отношениях**.

Объединение отношений. Пример

Предположим, что отношения A и B имеют вид, показанный на рис. (оба они получены из текущего значения отношения поставщиков S)

A

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

B

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris

а) Объединение (A UNION B)

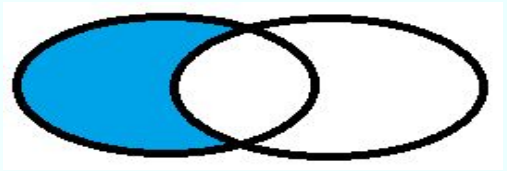
S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London
S2	Jones	10	Paris

В «A» приведены данные о поставщиках из Лондона, а в «B» — данные о поставщиках, которые поставляют деталь P1). В таком случае в объединение A UNION B (см. рис. а) входят поставщики, которые либо находятся в Лондоне, либо поставляют деталь P1, либо соответствуют обоим этим условиям.

Бинарные односхемные операции РА

Разность (except)

Разностью односхемных отношений R и S называется множество кортежей R, не входящих в S.



Отношение R

A	B	C
a	b	c
c	a	d
c	h	c

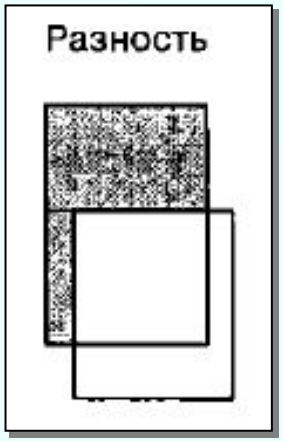
Отношение S

A	B	C
g	h	a
a	b	c
h	d	d

Разность R-S

A	B	C
c	a	d
c	h	c

Разность отношений



- Как и для объединения, для реляционной операции **разности** требуется, чтобы ее операнды **принадлежали к одному и тому же типу**.

- Если даны отношения **A** и **B** **одного** и того же **типа**, то разностью этих отношений **A MINUS B** (в указанном порядке), является отношение того же типа с телом, состоящим из всех кортежей **t**, таких, что **t** присутствует в **A**, но не в **B**.

Разность отношений. Пример

Отношения A и B показаны на рис.

A

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

B

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris

Результат операции разности **A MINUS B** (рис. в) включает поставщиков, которые находятся в Лондоне и не поставляют деталь **P1**.

Результат операции разности **B MINUS A** (рис. г) включает поставщиков, которые поставляют деталь P1 и не находятся в Лондоне.

в) Вычитание (A MINUS B)

S#	SNAME	STATUS	CITY
S4	Clark	20	London

г) Вычитание (B MINUS A)

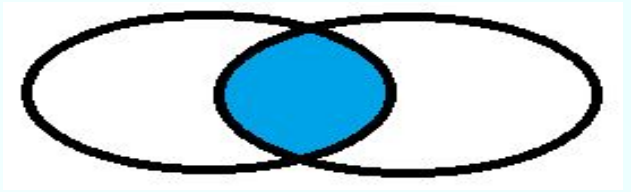
S#	SNAME	STATUS	CITY
S2	Jones	10	Paris

Оператор **MINUS** характеризуется направленностью (некоммутативностью), так же, как вычитание в обычной арифметике (например, "5 - 2" и "2 - 5" не являются одним и тем же).

Бинарные односхемные операции РА

Пересечение (intersect)

Пересечение двух односхемных отношений R и S есть подмножество кортежей, принадлежащих обоим отношениям. Это можно выразить через разность:



$R \cap S = R - (R - S)$.

Отношение R

A	B	C
a	b	c
c	a	d
c	h	c
d	r	t

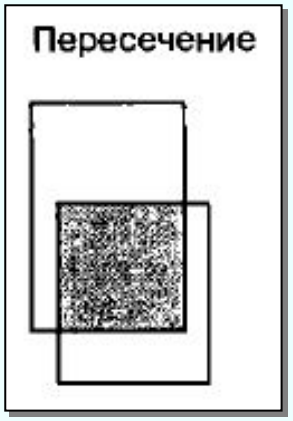
Отношение S

A	B	C
g	h	a
a	b	c
c	a	d
g	u	v

Пересечение R-S

A	B	C
a	b	c
c	a	d

Пересечение отношений



Как и для объединения, и фактически по той же причине, для **реляционной операции пересечения требуется, чтобы ее операнды принадлежали к одному и тому же типу.**

Если даны отношения **A** и **B** одного и того же типа, то пересечением этих отношений **A INTERSECT B** является отношение того же типа с телом, состоящим из всех кортежей **t**, таких, что **t** присутствует одновременно в **A** и **B**.

Пересечение отношений. Пример.

Предположим, что отношения A и B показаны на рис. Тогда пересечение A INTERSECT B (рис. б) включает всех поставщиков, которые находятся в Лондоне и поставляют деталь P1.

A

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

B

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris

б) Пересечение (A INTERSECT B)

S#	SNAME	STATUS	CITY
S1	Smith	20	London

Разносхемные операции РА

Соединение (join)

Эта операция определяет подмножество декартова произведения двух **разносхемных отношений**. Кортеж декартова произведения входит в результирующее отношение, если для атрибутов разных исходных отношений выполняется некоторое условие F. Соединение может быть выражено так:

$$R \underset{F}{\bowtie} S = \sigma_F (R \times S)$$

Если условием является равенство значений двух атрибутов исходных отношений, такая операция называется **эквисоединением**. **Естественным называется эквисоединение по одинаковым атрибутам исходных отношений.**

Разносхемные операции РА

Соединение

Отношение R

A	B	C
a	b	c
c	a	d
c	h	c
g	b	d

Отношение S

A	D	E
g	h	a
c	b	c
h	d	d

Соединение R⋈S

A	B	C	D	E
c	a	d	b	c
c	h	c	b	c
g	b	d	h	a

Операция соединения

Предположим, что отношения A и B, соответственно, имеют следующие атрибуты:

X1, X2, . . . , Xm, Y1, Y2, . . . , Yn (A)

Y1, Y2, . . . , Yn, Z1, Z2, . . . , Zp (B)

Это означает, что два рассматриваемых отношения имеют общее множество атрибутов Y, состоящее из атрибутов Y1, Y2 , . . . , Yn (и только из этих атрибутов),

Другие атрибуты отношения A образуют множество X, состоящее из атрибутов X1, X2,…, Xm.

Другие атрибуты отношения B образуют множество Z, состоящее из атрибутов Z1, Z2, . . , Zp.

Операция соединения

Теперь множества {X1, X2, . . . , Xm }, {Y1, Y2, . . . , Yn} и {Z1, Z2,. . ., Zp} могут рассматриваться, соответственно, как три составных атрибута X, Y и Z.

В таком случае (естественное) соединение A и B выражается следующим образом: A JOIN B

Оно представляет собой отношение с заголовком {X, Y, Z} и телом, состоящим из всех таких кортежей {X:x, Y:y, Z:z}, что любой из этих кортежей присутствует и в отношении A, со значением x атрибута X и значением y атрибута Y, и в отношении B, со значением y атрибута Y и значением z атрибута Z.

Пример естественного соединения (естественное соединение S JOIN P по общему атрибуту CITY).

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Black	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P2	Bolt	Green	17.0	Paris
P3	Screw	Blue	17.0	Rome
P4	Screw	Red	14.0	London
P5	Cam	Blue	12.0	Paris
P6	Cog	Red	19.0	London

S JOIN P

S#	SNAME	STATUS	CITY	P#	PNAME	COLOR	WEIGHT
S1	Smith	20	London	P1	Nut	Red	12.0
S1	Smith	20	London	P4	Screw	Red	14.0
S1	Smith	20	London	P6	Cog	Red	19.0
S2	Jones	10	Paris	P2	Bolt	Green	17.0
S2	Jones	10	Paris	P5	Cam	Blue	12.0
S3	Blake	30	Paris	P2	Bolt	Green	17.0
S3	Blake	30	Paris	P5	Cam	Blue	12.0
S4	Clark	20	London	P1	Nut	Red	12.0
S4	Clark	20	London	P4	Screw	Red	14.0
S4	Clark	20	London	P6	Cog	Red	19.0

Операция деления (division)

Пусть отношение A содержит атрибуты $\{r_1, r_2, \ldots, r_k\}$, а отношение B – атрибуты $\{r_{k+1}, \ldots, r_n\}$. Тогда результирующее отношение содержит атрибуты $\{r_1, r_2, \ldots, r_k\}$. Кортеж отношения A включается в результирующее отношение, если его декартово произведение с отношением B входит в C.

Отношение A(делимое)

A	B
a	b
g	h
c	f

Отношение C (посредник)

A	B	C	D
a	b	c	b
a	b	g	h
c	f	g	h
c	f	c	b
a	v	c	b
c	v	g	h

Отношение B (делитель)

C	D
c	b
g	h

Частное A/B

A	B
a	b
c	f

Операция деления

Предположим, что отношения A и B, соответственно, имеют следующие атрибуты:

X1, X2, . . . ,Xm (A); Y1, Y2, ..., Yn (B)

Здесь ни один из атрибутов Xi (i = 1, 2, . . ., m) не имеет одинакового имени с любым из атрибутов Yj (j = 1, 2, . . ., n).

Пусть отношение C имеет следующие атрибуты:

X1 , X2 , ... , Xm, Y1 , Y2 , ... , Yn (C)

ЭТО означает, что C имеет заголовок, представляющий собой (теоретико-множественное) объединение заголовков A и B.

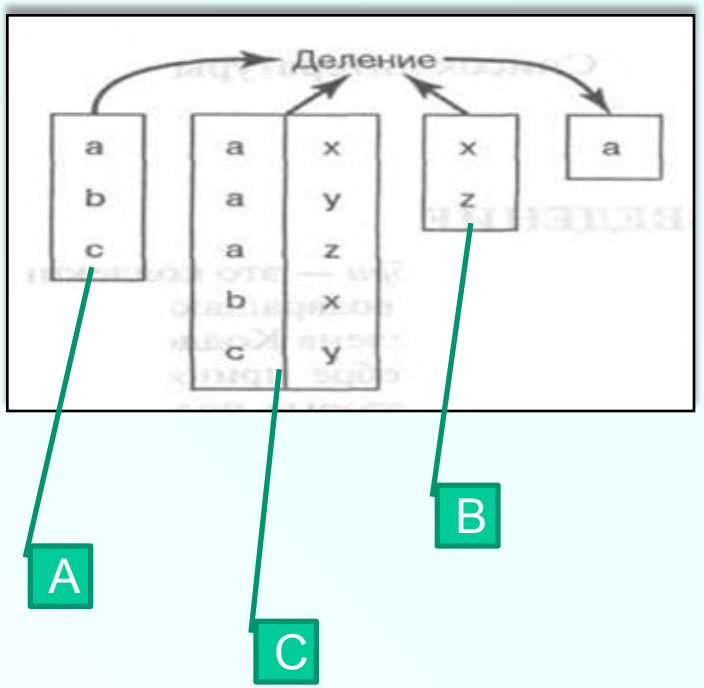
Будем рассматривать множества {X1, X2, ..., Xm} { Y1, Y2, . . ., Yn }, соответственно, как составные атрибуты X и Y. В таком случае операция деления A на B по C (где A — делимое, B — делитель, а C — посредник) может быть представлена с помощью следующего выражения:

A DIVIDEBY B PER C

Операция деления

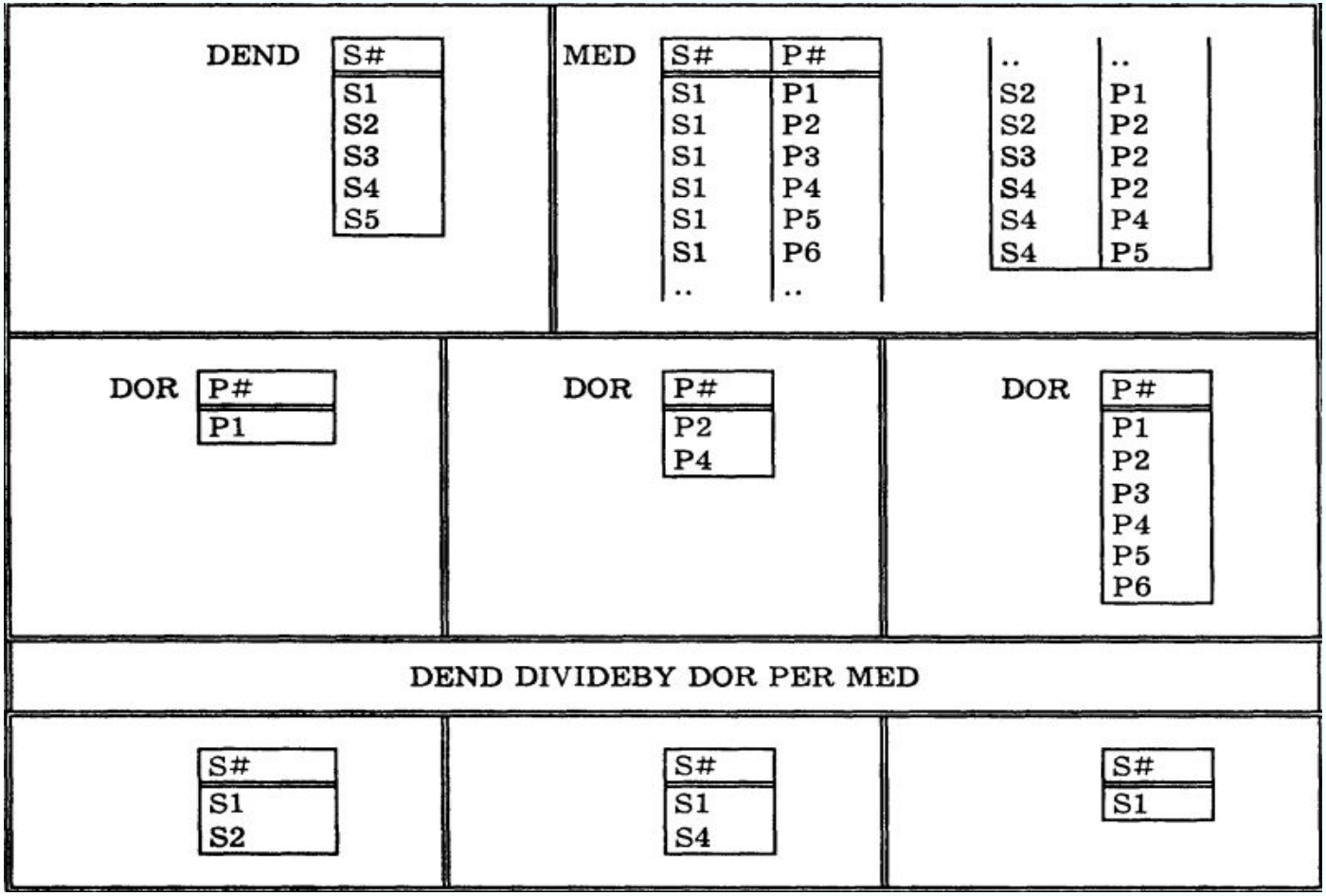
Деление представляет собой отношение с заголовком {X} и телом, состоящим из всех кортежей {X:x}, присутствующих в A, причем таких, что кортеж {X:x, Y:y} присутствует в C для всех кортежей {Y:y}, присутствующих в B.

Иными словами, данный результат состоит из тех значений x, присутствующих в A, для которых соответствующие значения y в C включают все значения y из B.



Пример деления

На рис. приведены некоторые примеры деления. В каждом случае делимое (DEND) представляет собой проекцию текущего значения отношения S по атрибуту S#; посредник (MED) в каждом случае является проекцией текущего значения отношения SP по атрибутам S# и P#; а три делителя (DOR) являются такими, как указано на этом рисунке.



3. Проектирование реляционных БД на основе принципов нормализации

Метод нормальных форм

- Атрибут **B** функционально зависит от атрибута **A**, если каждому значению **A** соответствует в точности одно значение **B**. Математически функциональная зависимость **B** от **A** обозначается записью **A→B.**

  Должн→Оклад

- Атрибут **C** зависит от атрибута **A** **транзитивно** (существует транзитивная зависимость), если для атрибутов **A, B, C** выполняются условия **A→B** и **B→C**, но обратная зависимость отсутствует, например, транзитивной зависимостью связаны атрибуты:

  ФИО→Должн→Оклад

Формирование исходного отношения

- Проектирование БД начинается с определения всех объектов, сведения о которых будут включены в базу, и определения их атрибутов. Затем атрибуты сводятся в одну таблицу – **исходное отношение**.
- Предположим, что для учебной части факультета создается БД о преподавателях.

Формирование исходного отношения

Установим атрибуты, которые должны содержаться в отношениях БД, и связи между ними:

- **ФИО** - фамилия и инициалы преподавателя. Исключаем возможность совпадения фамилии и инициалов у преподавателей.
- **Должн** - должность, занимаемая преподавателем.
- **Оклад** - оклад преподавателя.
- **Стаж** - преподавательский стаж.
- **Д_Стаж** - надбавка за стаж.
- **Каф** - номер кафедры, на которой числится преподаватель.
- **Предм** - название предмета (дисциплины), читаемого преподавателем.
- **Группа** - номер группы, в которой преподаватель проводит занятия.
- **ВидЗан** - вид занятий, проводимых преподавателем в учебной группе.

Формирование исходного отношения

Предполагается, что один преподаватель в одной группе может проводить один вид занятий (лекции или практические занятия).

ФИО	Должн	Оклад	Стаж	Д_Стаж	Каф	Предм	Группа	ВидЗан
Иванов И.М.	преп	500	5	100	25	СУБД	256	Практ
Иванов И.М.	преп	500	5	100	25	ПЛ/1	123	Практ
Петров М.И.	ст.преп	800	7	100	25	СУБД	256	Лекция
Петров М.И.	ст. преп	800	7	100	25	Паскаль	256	Практ
Сидоров Н. Г.	преп	500	10	150	25	ПЛ/1	123	Лекция
Сидоров Н. Г.	преп	500	10	150	25	Паскаль	256	Лекция
Егоров В.В.	преп	500	5	100	24	ПЭВМ	244	Лекция

Избыточное дублирование данных

Исходное отношение ПРЕПОДАВАТЕЛЬ содержит избыточное дублирование данных.

Явная избыточность заключается в том, что в отношении ПРЕПОДАВАТЕЛЬ строки с данными о преподавателях, проводящих занятия в нескольких группах, повторяются соответствующее число раз.

Неявная избыточность в отношении ПРЕПОДАВАТЕЛЬ проявляется в одинаковых окладах у всех преподавателей и в одинаковых добавках к окладу за одинаковый стаж. Поэтому, если при изменении окладов за должность с 500 на 510 это значение изменят у всех преподавателей, кроме, например, Иванова, то база станет противоречивой. Это пример ***аномалии редактирования для варианта с неявной избыточностью.***

Первая нормальная форма

- Средством исключения избыточности в отношениях и, как следствие, аномалий является **нормализация отношений**.
- **Первая нормальная форма**. Отношение находится в **1НФ**, если все его атрибуты являются простыми (имеют единственное значение).
- Исходное отношение **ПРЕПОДАВАТЕЛЬ**, используемое для иллюстрации метода, имеет составной ключ **ФИО, Предм, Группа** и находится в **1НФ**, поскольку все его атрибуты простые.

Вторая нормальная форма

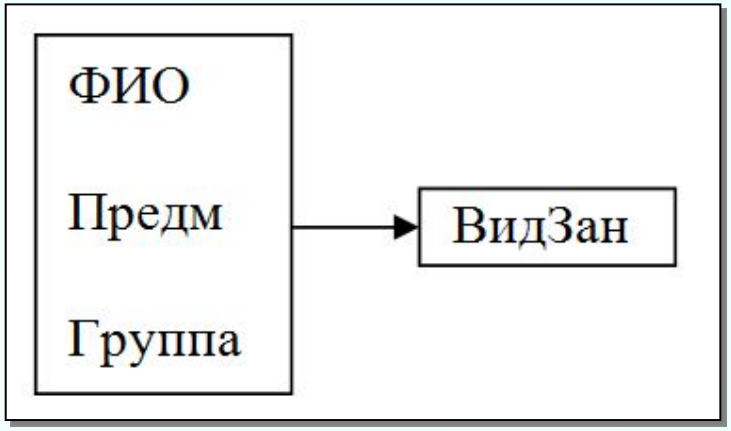
- Отношение находится в **2НФ**, если оно находится в **1НФ** и каждый неключевой атрибут функционально полно зависит от первичного ключа (составного).
- Для устранения частичной зависимости и перевода отношения в **2НФ** необходимо, используя операцию проекции, разложить его на несколько отношений следующим образом:
  - ✓ построить **проекцию** без атрибутов, находящихся в частичной функциональной зависимости от первичного ключа;
  - ✓ построить **проекции** на части составного первичного ключа и атрибуты, зависящие от этих частей.

Вторая нормальная форма

В результате получим два отношения R1 и R2 в 2НФ:

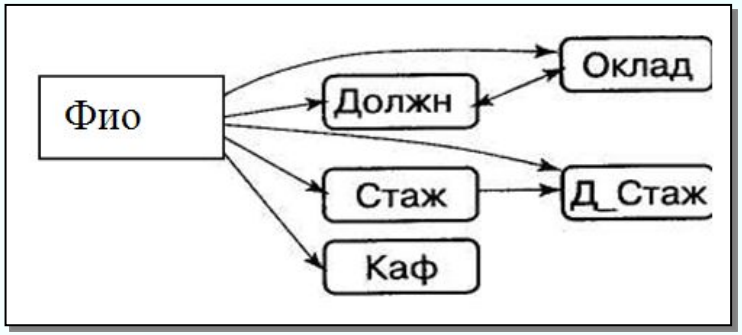
R1

ФИО	Предм	Группа	ВидЗан
Иванов И.М.	СУБД	256	Практ
Иванов И.М.	ПЛ/1	123	Практ
Петров М.И.	СУБД	256	Лекция
Петров М.И.	Паскаль	256	Практ
Сидоров Н.Г.	ПЛ/1	123	Лекция
Сидоров Н.Г.	Паскаль	256	Лекция
Егоров В.В.	ПЭВМ	244	Лекция



R2

ФИО	Должн	Оклад	Стаж	Д_Стаж	Каф
Иванов И.М.	преп	500	5	100	25
Петров М.И.	ст.преп	800	7	100	25
Сидоров Н.Г.	преп	500	10	150	25
Егоров В.В.	преп	500	5	100	24



В отношении R1 первичный ключ является составным и состоит из атрибутов ФИО, Предм, Группа. В отношении R2 ключ ФИО.
Переход к 2НФ позволил исключить явную избыточность данных в таблице R2 - повторение строк со сведениями о преподавателях.
В R2 по-прежнему имеет место неявное дублирование данных.

Третья нормальная форма

Определение 1. Отношение находится в 3НФ, если оно находится в 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Альтернативное определение

Определение 2. Отношение находится в 3НФ в том и только в том случае, если все неключевые атрибуты отношения взаимно независимы и полностью зависят от первичного ключа.

Третья нормальная форма

Если в отношении R1 транзитивные зависимости отсутствуют, то в отношении R2 они есть:

ФИО→Должн→Оклад,

ФИО→Оклад→Должн,

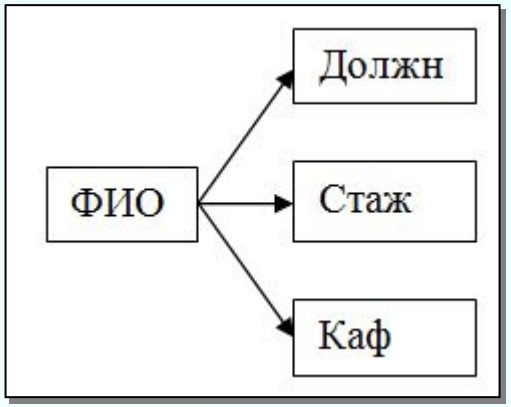
ФИО→Стаж→Д_Стаж.

Используя операцию проекции на атрибуты, являющиеся причиной транзитивных зависимостей, преобразуем отношение R2, получив при этом отношения R3, R4 и R5, каждое из которых находится в 3НФ.

Третья нормальная форма

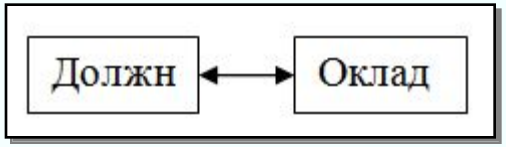
R3

ФИО	Должн	Стаж	Каф
Иванов И.М.	преп	5	25
Петров М.И.	ст.преп	7	25
Сидоров Н.Г.	преп	10	25
Егоров В.В.	преп	5	24



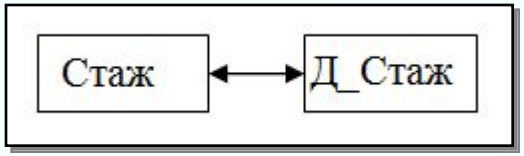
R4

Должн	Оклад
преп	500
ст.преп	800



R5

Стаж	Д_Стаж
5	100
7	100
10	150



Результатом проектирования является БД, состоящая из следующих таблиц: R1, R3, R4, R5.

Третья нормальная форма

R1

ФИО	Предм	Группа	ВидЗан
Иванов И.М.	СУБД	256	Практ
Иванов И.М.	ПЛ/1	123	Практ
Петров М.И.	СУБД	256	Лекция
Петров М.И.	Паскаль	256	Практ
Сидоров Н.Г.	ПЛ/1	123	Лекция
Сидоров Н.Г.	Паскаль	256	Лекция
Егоров В.В.	ПЭВМ	244	Лекция

R4

Должн	Оклад
преп	500
ст.преп	800

R3

ФИО	Должн	Стаж	Каф
Иванов И.М.	преп	5	25
Петров М.И.	ст.преп	7	25
Сидоров Н.Г.	преп	10	25
Егоров В.В.	преп	5	24

R5

Стаж	Д_Стаж
5	100
7	100
10	150

Результатом проектирования является БД, состоящая из следующих таблиц: R1, R3, R4, R5.

Нормальная форма Бойса - Кодда

- На практике построение **3НФ** схем отношений в большинстве случаев является **достаточным** и приведением к ним процесс проектирования реляционной БД **заканчивается**.
- Если в отношении имеется **зависимость атрибутов составного ключа от неключевых атрибутов**, то необходимо перейти к **усиленной 3НФ.**

Нормальная форма Бойса – Кодда (усиленная 3НФ)

Усиленная 3НФ или нормальная форма Бойса - Кодда (БКНФ). Отношение находится в БКНФ, если оно находится в 3НФ и в нем отсутствуют зависимости ключей (атрибутов составного ключа) от неключевых атрибутов.

Спасибо за внимание!