



Реляционная модель

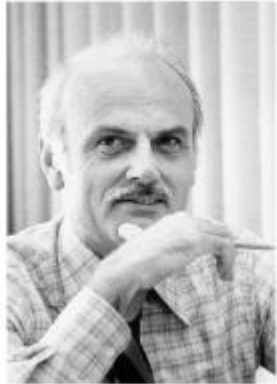
Теоретические основы



Реляционная модель

- Реляционная модель представляет данные на концептуальном уровне
- Реляционная модель является логической, то есть элементы модели являются логическими (абстрактными), а не физическими (храняемыми) структурами
- Основу реляционной модели составляет теория множеств

Реляционная модель



Edgar F. Codd



Chris Date



Ronald Fagin

Определение

$\{D_i\} \quad i \in [1, N]$

домены

$a_i^j \in D_i$

атрибуты

A_i

имена атрибутов

$R \subset D_1 \times D_2 \times \dots \times D_N$

$(a_1^j, a_2^j, \dots, a_N^j) \in R \quad j \in [1, m]$ кортеж

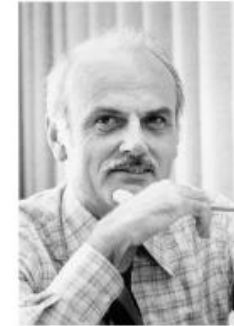
Отношение :

заголовок : $\{A_i : D_i\} \quad i \in [1, N]$

тело : $\{t_j\} \quad j \in [1, m]$, где $t_j = \{A_i : a_i^j\} \quad i \in [1, N]$

N – степень отношения

m – кардинальное число отношения



Edgar F. Codd

Свойства отношений

- Атомарность атрибутов в кортеже
- У отношения должен быть хотя бы один атрибут
- Неупорядоченность кортежей
- Неупорядоченность атрибутов
- Уникальность кортежей
- Набор атрибутов, достаточный для однозначного определения кортежа - возможный (потенциальный) ключ
- Отношение может иметь нулевое количество кортежей – пустое отношение

Основные требования

- Требования к реляционным базам данных для возможности их применения на практике сформулированы и развиты Кристофером Дж. Дейтом (IBM), который объединил их в группы, называемые аспектами
 - Структурный аспект
 - Аспект целостности
 - Аспект обработки



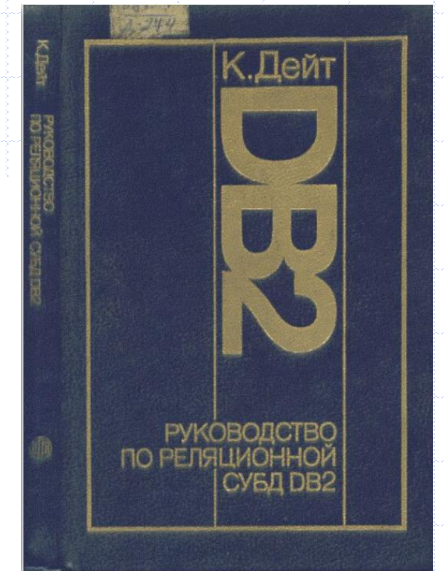
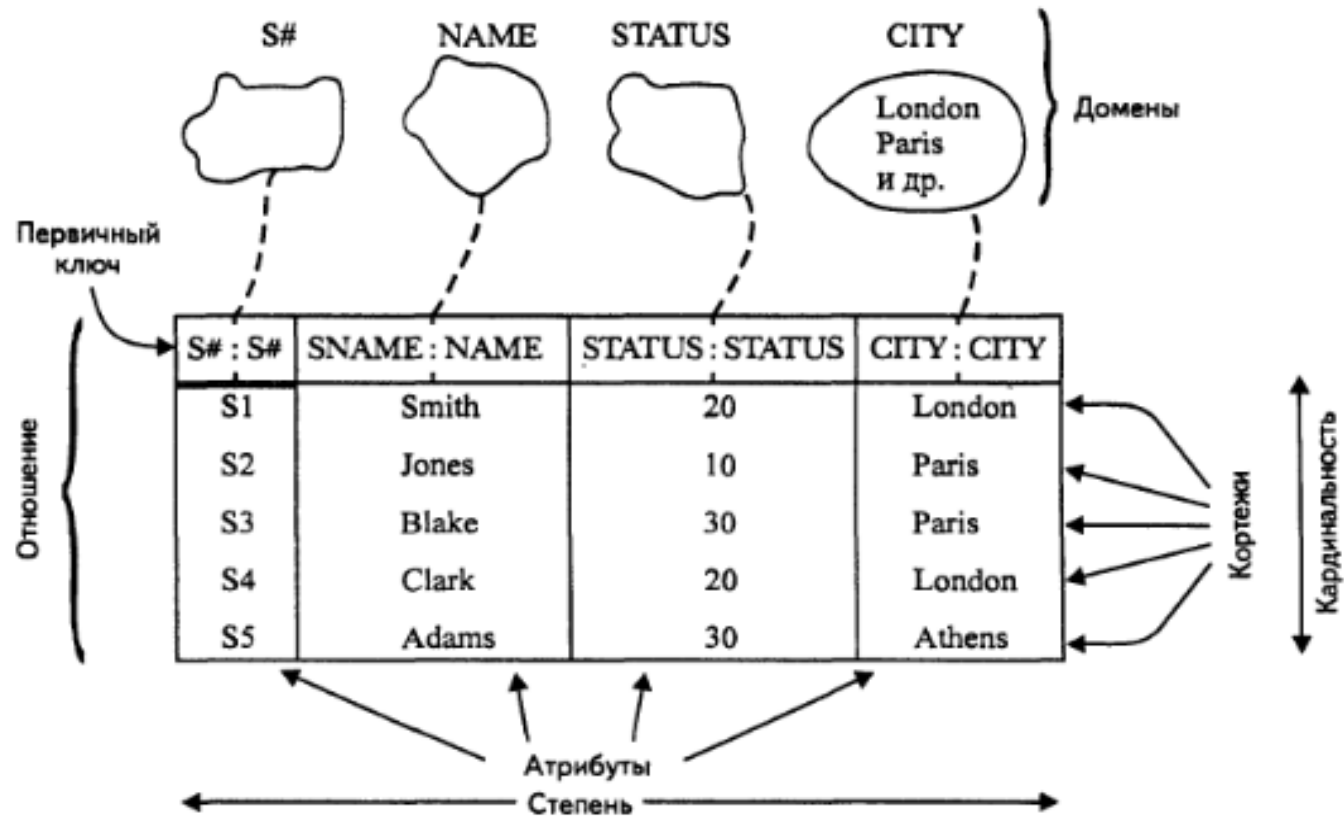
Chris Date

Структурный аспект

▶ данные в системе воспринимаются пользователем как таблицы

Формальный термин	Неформальный эквивалент
<ul style="list-style-type: none">● Отношение● Кортеж● Кардинальное число● Атрибут● Домен● Имя атрибута● Степень отношения● Заголовок отношения● Возможный (первичный) ключ	<ul style="list-style-type: none">● Таблица● Строка таблицы (запись)● Количество строк● Столбец (поле)● Множество допустимых значений (тип)● Имя столбца● Количество столбцов● Схема таблицы● Уникальный идентификатор строки

Структурный аспект



Аспект целостности

- ▶ **Целостность базы данных (database integrity)** — соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам
- ▶ Каждое правило, налагающее некоторое ограничение на возможное состояние базы данных, называется ограничением целостности (integrity constraint)
- ▶ Целостность БД не гарантирует достоверности содержащейся в ней информации, но обеспечивает по крайней мере правдоподобность этой информации, отвергая заведомо невероятные, невозможные значения

Аспект целостности

- Целостность сущностей (отношений)
 - Чтобы все кортежи отношения были различными, отношение должно иметь хотя бы один возможный (потенциальный) ключ
- Целостность по ссылкам
 - Если сущность реализуется несколькими отношениями, должны быть атрибуты, отвечающие за связь кортежей из разных отношений (внешние ключи)
- Целостность транзакций
 - Любая транзакция отражается в БД целиком или не отражается вообще

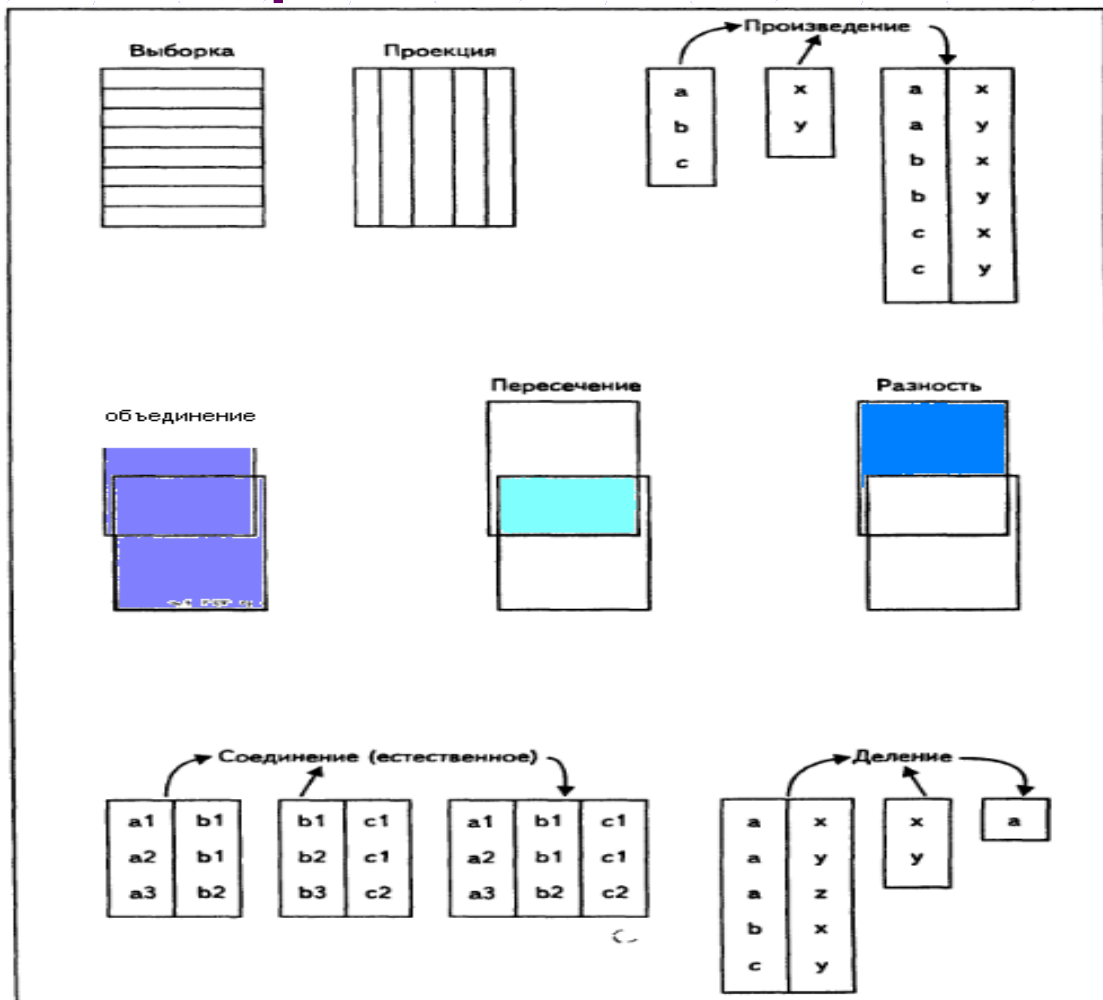
Аспект обработки

- Пользователь должен иметь операторы манипулирования данными, которые порождают *новые таблицы* на основе уже имеющихся.
- Среди операторов должны быть, по крайней мере, операторы селекции (выборки), проекции и соединения
- Операторы должны соответствовать реляционной алгебре или реляционному исчислению
- Под пользователем здесь понимаются разработчик ИС и администратор БД

Реляционная алгебра

- **Реляционная алгебра** — замкнутая система операций над отношениями в реляционной модели данных
- Первоначально операции были введены Коддом

Операции реляционной алгебры (Кодд)



Операции реляционной алгебры (Кодд)

$R(A_1: a_1, A_2: a_2, \dots, A_n: a_n)$

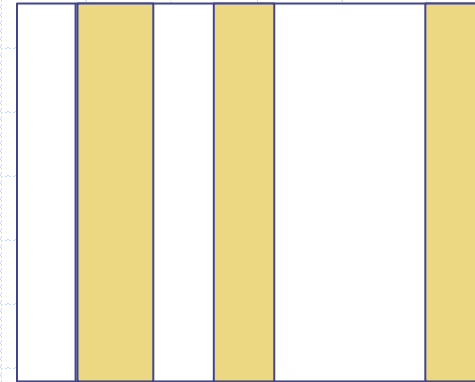
$S(B_1: b_1, B_2: b_2, \dots, B_m: b_m)$

Проекция

$\pi_{A_i, A_j, A_k}(R)$

$R[A_i, A_j, A_k]$

PROJECT $R\{A_i, A_j, A_k\}$

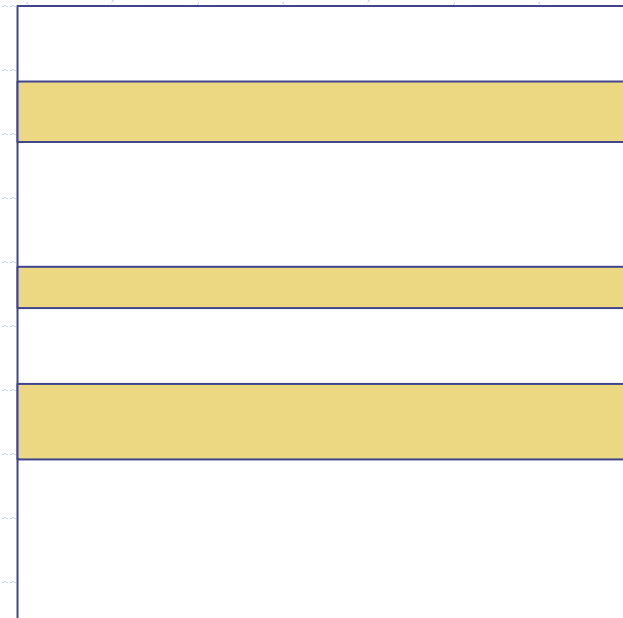


Операции реляционной алгебры (Кодд)

Селекция

$\sigma_{\theta} (R)$

R WHERE θ

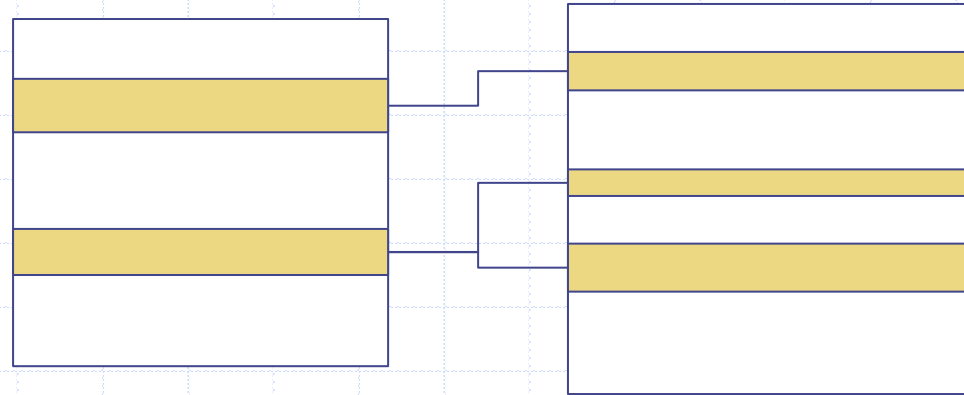


Операции реляционной алгебры (Кодд)

Соединение по условию

$$R \bowtie_{\theta} S$$

R JOIN S ON θ



Соединение по эквивалентности
Естественное соединение

Операции реляционной алгебры (Кодд)

Объединение

$R \cup S$

R UNION S

Пересечение

$R \cap S$

R INTERSECT S

Разность

$R - S$

R MINUS S

Произведение

$R \times S$

R TIMES S

Деление

$R \div S$

R DIVIDE S

Реляционная алгебра

- Все операции выполняются над отношениями
- Отношение в операции участвует целиком
- Результатом любого набора операций над отношениями в реляционной алгебре является отношение (замкнутость множества операций)
- Результат любой реляционной операции может быть использован в другой реляционной операции (суперпозиция)
- Можно считать, что неявно определена операция присваивания (новое отношение – результат операции)
- Множество операций является избыточным, минимально необходимый набор – селекция, проекция, соединение

Дополнительные операции

- Расширение (extend) для скалярных вычислений
- Переименование (rename), чтобы избежать коллизии имен
- Группировка (group)
- Обобщение (summarize)
 - Обобщающие функции (sum, count, max, min)
- Операции обновления
 - вставка кортежа
 - редактирование кортежа
 - удаление кортежа

Реляционное исчисление

- Наряду с **реляционной алгеброй** является способом получения результирующего отношения в **реляционной модели данных**
- Реляционное исчисление основано на исчислении предикатов

Интерпретация отношений

- Заголовок отношения можно считать *предикатом*, а любой кортеж – истинным высказыванием, полученным из этого предиката путем подстановки значений соответствующих типов вместо *параметров* этого предиката

Экзамен (студент:ФИО, предмет:НАЗВАНИЕ, балл:ОЦЕНКА)

Экзамен (студент:'Анна', предмет:'БД', балл:5)

Экзамен (предмет:'алгебра', студент:'Анна', балл:5)

Экзамен (предмет:'алгебра', балл:4, студент:'Иван')

Реляционное исчисление

- В основе исчисления лежит понятие переменной с определенной для нее областью допустимых значений и понятие *правильно построенной формулы*, опирающейся на переменные, **предикаты** и **кванторы**
- В зависимости от того, что является областью определения переменной, различают
 - Исчисление кортежей
 - Исчисление доменов

Исчисление кортежей

Определение переменной

range of T is *отношение*

Выражение

результатирующий список

where *правильно построенная формула*

Результатирующий список и формула содержат ссылки на атрибуты переменной $T.a_k$

Исчисление кортежей

➤ Формула

- задает предикат, проверяющий истинность кортежной переменной
- используют простые операции сравнения, логические операции и кванторы FORALL и EXISTS

ranges T is СТУДЕНТ
ranges P is СПЕЦИАЛЬНОСТЬ

N=T.фио, T.группа, P.факультет
where EXISTS

P (P.код_спец=T.код_спец
and (P.наим_спец = 'ФИИТ'
and T.курс=3)
or P.наим_спец = 'ПМИ'
and T.курс=4))

Исчисление доменов

- Переменные принимают значения в пределах домена
- При построении выражений используются предикаты, задающие условия членства
- Каждый предикат имеет вид

$R(t1:v1, t2:v2, \dots)$

или

$R(t1=v1, t2=v2, \dots)$

Исчисление доменов

(F, G, S) where

СТУДЕНТ (фио : F, группа : G, курс =3,
код_спец : X)

and

СПЕЦИАЛЬНОСТЬ (код_спец :Y,
наим_спец = 'ФИИТ', факультет : S)

and (X=Y)

Or ...

- Реляционная алгебра и реляционное исчисление представляют два альтернативных подхода к реализации операций в БД
 - Реляционная алгебра – описывает порядок выполнения операций, позволяющих из исходных выражений получить результат
 - Реляционное исчисление – описывает результат в терминах исходных отношений

Реляционная полнота

- Реляционная алгебра и реляционное исчисление эквивалентны
- Язык принято называть реляционно полным, если он по своим возможностям не уступает реляционной алгебре
- Восемь операторов Кодда являются мерой выразительной силы любого языка БД

Важные моменты

- Операции реляционного исчисления позволяют реализовать декларативное программирование и декларативное описание ограничений целостности, в дополнение к процедурному программированию и процедурной проверке условий, реализуемому средствами реляционной алгебры
- Для реляционных баз данных верен информационный принцип: всё информационное наполнение базы данных представлено одним и только одним способом, а именно — явным заданием значений атрибутов в кортежах отношений; в частности, нет никаких указателей (адресов), связывающих одно значение с другим

12 правил Кодда (1985 г)

В силу строгости правил современные реляционные СУБД не соответствуют многим из них

➤ 0. Основное правило (Foundation Rule)

Система, которая рекламируется или позиционируется как реляционная система управления базами данных, должна быть способной управлять базами данных, используя исключительно свои реляционные возможности

➤ 1. Информационное правило (The Information Rule)

Вся информация в реляционной базе данных на логическом уровне должна быть явно представлена единственным способом – значениями в таблицах

➤ 2. Гарантированный доступ к данным (Guaranteed Access Rule)

каждое отдельное (атомарное) значение данных должно быть логически доступно с помощью комбинации имени таблицы, значения первичного ключа и имени столбца

12 правил Кодда (1985 г)

➤ 3. Систематическая поддержка отсутствующих значений

Неизвестные, или отсутствующие значения, отличные от любого известного значения, должны поддерживаться для всех типов данных при выполнении любых операций единообразно

➤ 4. Доступ к метаданным (словарю) в терминах реляционной модели

Словарь данных должен сохраняться в форме реляционных таблиц, и СУБД должна поддерживать доступ к нему при помощи стандартных языковых средств, тех же самых, которые используются для работы с реляционными таблицами, содержащими пользовательские данные

12 правил Кодда (1985 г)

➤ 5. Полнота множества языка

Система управления реляционными базами данных должна поддерживать хотя бы один реляционный язык, который

(а) имеет линейный синтаксис,

(б) может использоваться как интерактивно, так и в прикладных программах,

(в) поддерживает операции определения данных, определения представлений, манипулирования данными (интерактивные и программные), ограничители целостности, управления доступом и операции управления транзакциями (begin, commit и rollback).

12 правил Кодда (1985 г)

➤ 6. Возможность изменения представлений

Каждое представление должно поддерживать все операции манипулирования данными, которые поддерживают реляционные таблицы: операции выборки, вставки, изменения и удаления данных

➤ 7. Высокоуровневые операции управления данными

Операции вставки, изменения и удаления данных должны поддерживаться не только по отношению к одной строке реляционной таблицы, но и по отношению к любому множеству строк.

➤ 8. Физическая независимость данных

Приложения не должны зависеть от используемых способов хранения данных на носителях

12 правил Кодда (1985 г)

➤ 9. Логическая независимость данных

Представление данных в приложении не должно зависеть от структуры реляционных таблиц. Если в процессе нормализации одна реляционная таблица разделяется на две, представление должно обеспечить объединение этих данных, чтобы изменение структуры реляционных таблиц не сказывалось на работе приложений.

➤ 10. Независимость контроля целостности

Вся информация, необходимая для поддержания целостности, должна находиться в словаре данных. Язык для работы с данными должен выполнять проверку входных данных и автоматически поддерживать целостность данных.

➤ 11. Независимость от расположения

База данных может быть распределённой, может находиться на нескольких компьютерах, и это не должно оказывать влияния на приложения. Перенос базы данных на другой компьютер не должен оказывать влияния на приложения.

12 правил Кодда (1985 г)

➤ 12. Согласование языковых уровней

Если используется низкоуровневый язык доступа к данным, он не должен игнорировать правила безопасности и правила целостности, которые поддерживаются языком более высокого уровня