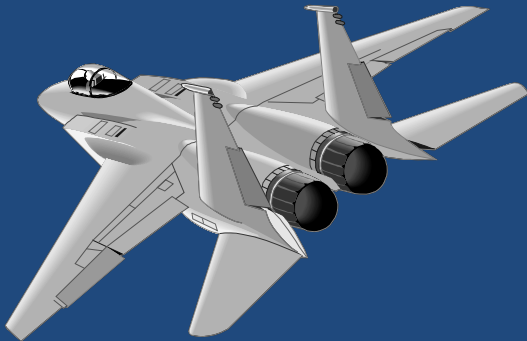


РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

*Предложение есть модель действительности,
как мы ее себе мыслим.*

Л. Витгенштейн



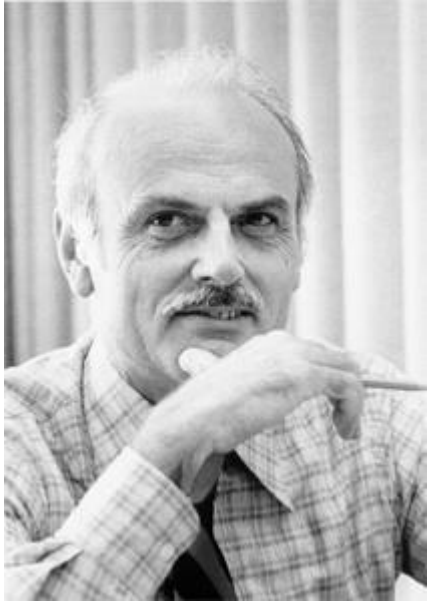
Содержание

2

- Историческая справка
- Особенности и состав модели
- Основные понятия реляционной модели данных
- Целостность реляционных данных

Историческая справка

3



Эдгар Франк («Тед»)
Кодд
1923-2003

- *Реляционная модель данных (РМД)* изобретена Т. Коддом в 1970 г.
 - Codd E.F. The Relation Model for Large Shared Data Banks // Communications of the ACM. 1970. Vol. 13, No. 6. P. 377-387.
 - За разработку реляционной модели данных Кодд был удостоен звания "IBM Fellow" (1976) и премии Тьюринга (1981).
- В настоящее время 99,99% коммерческих СУБД основаны на РМД.

Особенности РМД

4

- Логический характер объектов модели
 - Отношения – логические, а не физические структуры.
- Информационный принцип построения объектов
 - Информация в базе данных представлена одним и только одним способом – явным заданием значений атрибутов в кортежах отношений; нет никаких указателей - физических адресов для связи значений.
- Поддержка декларативного и императивного программирования
 - Реляционная алгебра – декларативное программирование и декларативное описание ограничений целостности.
 - Процедурный язык манипулирования данными.

Состав РМД

5

- Структурный аспект
 - Данные в базе данных представляют собой набор отношений.
- Аспект целостности
 - Отношения отвечают определенным условиям целостности.
 - РМД поддерживает декларативные ограничения целостности уровня домена (типа данных), уровня отношения и уровня базы данных.
- Аспект манипулирования
 - РМД поддерживает операторы манипулирования отношениями (реляционная алгебра, реляционное исчисление).
- Аспект нормализации
 - Ограничения на структуру отношений базы данных, улучшающие эффективность работы с базой данных.

Основные термины РМД

6

Термин РМД	Англ. термин	Неформальный термин
Отношение	Relation	Таблица
Кортеж	Tuple	Запись таблицы
Атрибут	Attribute	Столбец таблицы
Домен	Domain	Тип данных у значений в столбце таблицы
Первичный ключ	Primary key	Поле - уникальный идентификатор записи

Основные термины РМД

7

Атрибут
(на соответствующем домене)

Отношение

Первичный ключ

Заголовок отношения

Код_П	Имя	Город	Рейтинг
S1	Бендер	Одесса	10
S2	Воробьянинов	Старгород	14
S5	Деточкин	Энск	5
S3	Горбунков	Черноморск	8

Кортеж

Тело отношения

Домен

8

- *Домен* – именованное множество скалярных значений одного типа.
 - Скалярное (атомарное) значение не имеет внутренней структуры.
 - ФИО: строка[30] – скаляр
 - ФИО: { строка[10], строка[10], строка[10] } – не скаляр
 - Создание /удаление домена
 - `create domain Код_Д char(5)`
 - `drop domain Код_Д`

Для чего нужны домены?

9

- Домены ограничивают сравнения
 - ▣ `select Имя_Д
from P, SP
where P.Код_Д=SP.Код_Д`
 - ▣ `select Имя_Д
from P, SP
where P.Вес=SP.Количество`
- Домены допускают запросы к словарю базы данных
 - ▣ `select Имя_Отношения, Имя_Атрибута
from СловарьБД_Отношения
where Имя_Домена='Код_П'`

Отношение

10

- *Отношение R* , определенное на множестве доменов $D = \{D_1, \dots, D_k\}$ состоит из двух частей: заголовок и тело.
- *Заголовок отношения* – множество пар $\langle \text{имя-атрибута} : \text{имя-домена} \rangle$, т.е. множество $\{ \langle A_1 : D_1 \rangle, \dots, \langle A_n : D_n \rangle \}$, где имена атрибутов A_j различны, домены атрибутов $D_j \in D$ не обязательно различны.
- *Тело отношения* – множество кортежей.
- *Кортеж* – множество пар $\langle \text{имя-атрибута} : \text{значение-атрибута} \rangle$, т.е. множество $\{ \langle A_1 : v_{i1} \rangle, \dots, \langle A_n : v_{in} \rangle \}$, где $i \in \{1, \dots, m\}$.
- Значение m – *кардинальное число отношения*.
- Значение n – *степень (арность) отношения*.

Отношение

11

Код_П	Имя	Город	Рейтинг
S1	Бендер	Одесса	10
S2	Воробьянинов	Старгород	14
S5	Деточкин	Энск	5
S3	Горбунков	Черноморск	8

- { Код_П : Код_П, Имя : Имя, Город : Город, Рейтинг : Рейтинг }
- {
 - {<Код_П : S1>, <Имя : Бендер>, <Город : Одесса>, <Рейтинг : 10>},
 - { ... }
 - ...
- }

Отношения vs таблицы

12

Отношение	Таблица
Не может содержать одинаковых кортежей (тело отношения – множество)	Может содержать одинаковые строки
Кортежи не упорядочены (тело отношения – множество)	Строки таблицы могут быть упорядочены
Атрибуты не упорядочены (заголовок отношения – множество)	Столбцы таблицы могут быть упорядочены
Значения атрибутов кортежей атомарные (или отношение <i>нормализовано</i>)	Ячейка таблицы может содержать другую таблицу

Отношения vs таблицы

13

- Нет одинаковых кортежей
 - `select Имя from S where Рейтинг >= 15`

- SQL

Имя
Бендер
Бендер

- РМД

Имя
Бендер

Код_П	Имя	Город	Рейтинг
S1	Бендер	Одесса	15
S2	Воробьянинов	Старгород	14
S5	Деточкин	Энск	5
S3	Горбунков	Черноморск	8
S10	Бендер	Москва	20

Отношения vs таблицы

14

□ Кортежи не упорядочены (сверху вниз)

□ SQL

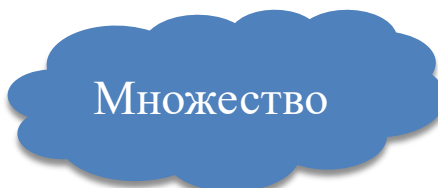
```
select Код_П, Имя,  
Рейтинг  
from S  
order by Рейтинг desc
```

□ РМД

```
select Код_П, Имя,  
Рейтинг  
from S
```

Код_П	Имя	Город	Рейтинг
S1	Бендер	Одесса	15
S2	Воробьянинов	Старгород	14
S5	Деточкин	Энск	5
S3	Горбунков	Черноморск	8
S10	Бендер	Москва	20

Код_П	Имя	Рейтинг
S10	Бендер	20
S1	Бендер	15
S2	Воробьянинов	14
S3	Горбунков	8
S5	Деточкин	5



Код_П	Имя	Рейтинг
S1	Бендер	15
S2	Воробьянинов	14
S5	Деточкин	5
S3	Горбунков	8
S10	Бендер	20

Отношения vs таблицы

15

□ Атрибуты не упорядочены (слева направо)

□ SQL

```
select Имя, Код_П,  
Рейтинг  
from S  
order by Рейтинг desc
```

□ РМД

```
select Код_П, Имя,  
Рейтинг  
from S
```

Код_П	Имя	Город	Рейтинг
S1	Бендер	Одесса	15
S2	Воробьянинов	Старгород	14
S5	Деточкин	Энск	5
S3	Горбунков	Черноморск	8
S10	Бендер	Москва	20

Имя	Код_П	Рейтинг
Бендер	S10	20
Бендер	S1	15
Воробьянинов	S2	14
Горбунков	S3	8
Деточкин	S5	5

Код_П	Имя	Рейтинг
S1	Бендер	15
S2	Воробьянинов	14
S5	Деточкин	5
S3	Горбунков	8
S10	Бендер	20



Отношения vs таблицы

16

- Все значения атрибутов атомарны

Код_П	Код_Д_Количество	
S1	Код_Д	Количество
	P1	200
	P2	300
	P3	100
S2	Код_Д	Количество
	P1	200
	P4	400
S3	Код_Д	Количество
	P5	500
	P1	800



Код_П	Код_Д	Количество
S1	P1	200
S1	P2	300
S1	P3	100
S2	P1	200
S2	P4	400
S3	P5	500
S3	P1	800

Виды отношений

17

- *Базовое отношение* – именованное отношение (переменная отношения), которое постоянно хранится в базе данных.
- *Производное отношение* – отношение, определенное через базовые посредством реляционного выражения.
- *Представление* – виртуальное (не хранящееся постоянно в базе данных) именованное производное отношение. Применяется для реализации внешнего уровня ANSI/SPARC архитектуры систем баз данных.
- *Снимок* – именованное производное отношение, которое постоянно хранится в базе данных. Применяется для реконструкции базы данных после сбоев.
- *Результат запроса* – неименованное производное отношение.
- *Хранимое отношение* – отношение, которое поддерживается в физической памяти.

Виды отношений

18

- Базовое отношение
 - create **base relation** S (
Код_П domain (Код_П),
Имя domain (Имя),
Город domain (Город),
primary key (Код_П))
- Представление
 - create **view** Надежные_П as
select Код_П, Имя
from S
where Рейтинг>=15
- Снимок
 - create **snapshot** Поставляемые_Детали as
select Код_Д, Имя
from P, SP
where P.Код_Д=SP.Код_Д
refresh every day
- Промежуточный результат (выражение реляционной алгебры)
 - ((S JOIN SP) WHERE Код_Д='P1')[Код_П, Имя_П]

Реляционная БД

19

- *Реляционная база данных* – база данных, воспринимаемая пользователем (на внешнем уровне ANSI/SPARC архитектуры систем баз данных) как набор нормализованных отношений различной степени.

Целостность реляционных данных

20

- Ключи
 - ▣ Потенциальные, первичные и альтернативные
 - ▣ Простые и составные
 - ▣ Внешние
- Ссылочная целостность
- NULL-значения

Потенциальный ключ

21

- Подмножество K атрибутов отношения R является *потенциальным ключом*, если K обладает следующими свойствами:
 1. Уникальность
 - Никакие два кортежа в R не могут иметь одинаковое значение в K .
 2. Неизбыточность
 - Никакое подмножество K не обладает свойством уникальности.

Потенциальный ключ

22

- Каждое отношение имеет минимум один потенциальный ключ (все атрибуты отношения)
- Потенциальных ключей может быть несколько
 - Таблица химических элементов Д.И. Менделеева.
- Неизбыточность потенциального ключа важна! В противном случае
 - нельзя обеспечить должным образом целостность
 - Если взять потенциальный ключ {Код_П, Город} вместо {Код_П}, то код поставщика будет уникальным только в пределах одного города, а не всех городов.
 - внешний ключ, на который ссылается избыточный потенциальный ключ, будет также избыточным.
- Потенциальный ключ \neq индекс по атрибуту
 - Не обязательно существует индекс (физический файл для ускорения доступа к данным отношения) по потенциальному ключу.

Выбор потенциальных ключей

23

- ❑ На основе определения и семантики предметной области.
- ❑ Пример: выбрать потенциальные ключи отношения Персона
 - ❑ Фамилия
 - ❑ Имя
 - ❑ Отчество
 - ❑ Пол
 - ❑ Дата рождения
 - ❑ ИНН
 - ❑ СНИЛС
 - ❑ Серия паспорта
 - ❑ Номер паспорта
 - ❑ Место и дата выдачи паспорта
 - ❑ Индекс
 - ❑ Город
 - ❑ Адрес

Назначение потенциальных ключей

24

- Потенциальный ключ позволяет адресовать кортежи отношения.

- ▣ `select Имя
from S
where Код_П='S1'`

Имя
Бендер

Код_П*	Имя	Город	Рейтинг
S1	Бендер	Одесса	15
S2	Воробьянинов	Старгород	14
S5	Деточкин	Энск	5
S3	Горбунков	Черноморск	8
S10	Бендер	Москва	20

Потенциальный ключ

Виды потенциальных ключей

25

- Если потенциальный ключ состоит из более чем одного атрибута, он называется *составным*, иначе — *простым*.
- Один из потенциальных ключей должен быть выбран в качестве *первичного ключа*. Остальные потенциальные ключи называются *альтернативными ключами*.

Внешний ключ

26

- *Внешний ключ* FK в отношении $R2$ – это подмножество атрибутов $R2$ такое, что
 - существует отношение $R1$ (не обязательно отличное от $R2$) с потенциальным ключом $СК$
 - для каждого значения FK существует кортеж в $R1$ с совпадающим значением $СК$.

Внешние ключи

27

Поставщики

Код_П*	Имя_П
S1	Бендер
S2	Воробьянинов
S5	Деточкин
S3	Горбунков
S10	Бендер

Внешний ключ

Внешний ключ

Поставки

Код_П	Код_Д	К-во
S1	P1	300
S1	P5	500
S3	P2	900
S3	P1	100
S10	P2	400

Детали

Код_Д*	Имя_Д
P1	Гайка
P2	Болт
P5	Шуруп
P7	Дюбель

Внешние ключи

28

- Внешний ключ должен определяться на тех же доменах, что и соответствующий потенциальный ключ.
- Внешний ключ не обязательно должен входить в первичный (потенциальный) ключ, но это желательно.

Внешние ключи

29

- Потенциальный ключ может содержать значение, не являющееся значением внешнего ключа.

Поставщики

Код_П*	Имя_П
S1	Бендер
S2	Воробьянинов
S5	Деточкин
S3	Горбунков
S10	Бендер

Поставки

Код_П*	Код_Д*	К-во
S1	P1	300
S1	P5	500
S3	P2	900
S3	P1	100
S10	P2	400

Детали

Код_Д*	Имя_Д
P1	Гайка
P2	Болт
P5	Шуруп
P7	Дюбель

Внешние ключи

30

- Внешний ключ составной (простой), если соответствующий потенциальный ключ составной (простой).

Внешние ключи

31

- Внешний ключ и соответствующий потенциальный ключ могут принадлежать одному и тому же отношению (ссылающееся и целевое отношение совпадают).

Код_Сотр*	ФИО	Начальник
001	Бендер О.И.	NULL
002	Воробьянинов И.М.	001
005	Деточкин Ю.И.	002
007	Бонд Д.	001

Внешние ключи

32

- create base relation SP (
 - Код_П domain (Код_П),
 - Код_Д domain (Код_Д),
 - Количество domain (Количество),
 - primary key (Код_П, Код_Д),
 - foreign key (Код_П) references S (Код_П),**
 - foreign key (Код_Д) references P (Код_Д)**

Правила внешних ключей

33

- Что делать в случае попытки удалить (обновить) потенциальный ключ, на который ссылается внешний ключ?
- ▣ Удаление поставщика, имеющего хотя бы одну поставку.
- ▣ Изменение кода детали, входящей хотя бы в одну поставку.

Поставщики

Код_П*	Имя_П
S1	Бендер
S2	Воробьянинов
S5	Деточкин
S3	Горбунков
S10	Бендер

Поставки

Код_П*	Код_Д*	К-во
S1	P1	300
S1	P5	500
S3	P2	900
S3	P1	100
S10	P2	400

Детали

Код_Д*	Имя_Д
P1	Гайка
P2	Болт
P5	Шуруп
P7	Дюбель

Правила внешних ключей

34

- Нетипичные правила
 - ▣ Запретить
 - ▣ Архивировать
- Типичные правила
 - ▣ Ограничить
 - ▣ Каскадировать
 - ▣ Установить в NULL
 - ▣ Установить по умолчанию

Правила внешних ключей

35

- *Ограничить* – запретить удаление (обновление) кортежей ссылочного отношения до момента, когда в ссылающемся отношении не будут отсутствовать кортежи с соответствующим значением внешнего ключа.

Правила внешних ключей: ограничить

36

НЕ могут быть обновлены (удалены)

Поставщики

Код_П*	Имя_П
S1	Бендер
S2	Воробьянинов
S5	Деточкин
S3	Горбунков
S10	Бендер

Поставки

Код_П*	Код_Д*	К-во
S1	P1	300
S1	P5	500
S3	P2	900
S3	P1	100
S10	P2	400

Детали

Код_Д*	Имя_Д
P1	Гайка
P2	Болт
P5	Шуруп
P7	Дюбель

Могут быть обновлены (удалены)

Правила внешних ключей

37

- *Каскадировать* – удалить (обновить) кортежи ссылочного отношения с соответствующим значением внешнего ключа.

Правила внешних ключей: каскадировать

38

Поставщики

Код_П*	Имя_П
007	Бендер
S2	Воробьянинов
S5	Деточкин
S3	Горбунков
S10	Бендер

Поставки

Код_П*	Код_Д*	К-во
007	P1	300
007	P5	500
УДАЛЕН		
S3	P1	100
УДАЛЕН		

Детали

Код_Д*	Имя_Д
P1	Гайка
УДАЛЕН	
P5	Шуруп
P7	Дюбель

Ограничение и каскад

39

- create base relation SP (
 - Код_П domain (Код_П),
 - Код_Д domain (Код_Д),
 - Количество domain (Количество),
 - primary key (Код_П, Код_Д),
 - foreign key (Код_П) references S (Код_П)**
 - on delete restrict**
 - on update cascade,**
 - foreign key (Код_Д) references P (Код_Д)**
 - on delete restrict**
 - on update cascade)**

Правила внешних ключей

40

- *Установить в NULL* – удалить (обновить) кортежи ссылочного отношения и в ссылающемся отношении установить у соответствующих кортежей неопределенное значение внешнего ключа (NULL).
- *Установить по умолчанию* – удалить (обновить) кортежи ссылочного отношения и в ссылающемся отношении установить у соответствующих кортежей значение внешнего ключа по умолчанию (в этой базе данных).

Установка в умолчание и NULL

41

- create base relation EMP (
Код_Сотр domain (Код_Сотр),
ФИО domain (ФИО),
Начальник domain (Код_Сотр),
primary key (Код_Сотр),
foreign key (Начальник) references EMP (Код_Сотр)
on delete set NULL
on update set default '????')

NULL-значения

42

- *NULL-значение* – специальное значение, показывающее отсутствие информации.
 - $NULL \neq \{ "", ' ', ' ', 0 \}$
 - $NULL \approx \emptyset$
 - create base relation R (
Attr domain (Domain) nulls [not] allowed)
- NULL-значения порождают трехзначную логику.

AND	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	NULL
FALSE	FALSE	FALSE	FALSE
NULL	NULL	FALSE	NULL

OR	TRUE	FALSE	NULL
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	NULL
NULL	TRUE	NULL	NULL

NOT	
TRUE	FALSE
FALSE	TRUE
NULL	NULL

NULL-значения и потенциальные ключи

43

- Ни один элемент первичного ключа базового отношения не может принимать NULL-значение.
 - ▣ Записываемые в отношение кортежи должны быть идентифицируемы!
- Результирующее отношение может иметь NULL-значение в первичном ключе.
 - ▣ `select Цвет from P`
- Для альтернативных ключей NULL значения могут быть разрешены или запрещены.
 - ▣ Если альтернативный ключ может принимать NULL-значения, то он не может быть выбран в качестве первичного.

NULL-значения и внешние ключи

44

- *Внешний ключ FK* в отношении $R2$ – это подмножество атрибутов $R2$ такое, что
 - ▣ существует отношение $R1$ (не обязательно отличное от $R2$) с потенциальным ключом CK
 - ▣ **FK может принимать неопределенное значение NULL**, а для каждого отличного от NULL значения FK в $R1$ существует кортеж с совпадающим значением CK .
- create base relation $R2$ (
foreign key (FK) references $R1$ (CK)
on (delete | update) restrict | cascade | set null
nulls [not] allowed)

Правила внешних ключей: установить в NULL

45

Код_П*	Имя_П
007	Бендер
S2	Воробьянинов
S5	Деточкин
S3	Горбунков
S10	Бендер

Код_ПД*	Код_П	Код_Д	К-во
SP1	NULL	P1	300
SP2	NULL	P5	500
SP5	S3	NULL	900
SP9	S3	P1	100
SP4	S10	NULL	400
SP5	NULL	NULL	700

Код_Д*	Имя_Д
P1	Гайка
УДАЛЕН	
P5	Шуруп
P7	Дюбель

Заключение

46

- Реляционная модель данных – основа современных технологий баз данных.
- Базовые понятия РМД – домены, атрибуты, кортежи, реляционные отношения.
- Целостность реляционных данных
 - Потенциальные и первичные ключи
 - Внешние ключи и ссылочная целостность
 - NULL-значения