Дисциплина
«Базы и банки данных»

Маркова Ирина Васильевна,
начальник управления информатизации
markova@miit.ru

Модель данных

Литература:

1. Е.Ф. Кодд. Реляционная модель данных для больших совместно используемых банков данных (перевод М.Р. Когаловского). http://citforum.ru/database/classics/codd/ .
2. К. Дейт. Введение в системы баз данных.: Пер. с англ. - 8-е издание. - М.: Издательский дом «Вильямс», - 2008. - 1328 с. с ил.
3. К. Рубинсон. NULL, трехзначная логика и неопределенность в SQL: критика критики Дейта, 2007, (пересказ С.Кузнецова). http://citforum.ru/database/articles/nulls/ .

Модель данных – это формальная теория представления и обработки данных, позволяющая моделировать их структуру, поведение и обеспечивающая доступ к данным.

Классические модели данных:

- иерархическая (данные организованы в виде дерева графа);
- сетевая (данные организованы в виде графа общего вида);
- реляционная (данные организованы в виде двумерных таблиц).

Реляционная модель данных

Автор – Эдгар Франк Кодд (E. F. Codd) , 1969—1970 г., Альмаденский Исследовательский Центр IBM (Калифорния).

В 2002 журнал Forbes поместил реляционную модель данных в список важнейших инноваций последних 85 лет.

Основные понятия:

Скаляр – наименьшая неделимая смысловая (семантическая) единица данных (скаляр атомарен).

Домен – поименованное множество скаляров одного типа. Содержимое домена не изменяется со временем (домен статичен) и, кроме того, он содержит все возможные значения подходящего типа.

$$D(Город) = \{Москва, Киев, Рига, ...\}.$$

Атрибут – поименованное подмножество домена, скаляры в котором несут одинаковую смысловую нагрузку.

$$Место\ рождения,\ Место\ жительства,\ Пункт\ назначения \subset D(Город).$$

Отношение – поименованное множество атрибутов, взятых из одного и более доменов. Значения атрибутов, составляющих отношение, могут изменяться со временем (отношения динамичны).

Схема отношения – множество имен атрибутов в отношении.

Расписание (Номер поезда, Станция отправления, Станция назначения, Время отправления, Время прибытия).

Основные понятия (продолжение)

Кортеж – элемент отношения как множества.

{16, Москва, Мурманск, 22.30, 7.00}

Кардинальное число (мощность) отношения – количество кортежей в нем. Одинаковых кортежей в отношении быть не может и они не упорядочены.

Степень (-арность) отношения – количество атрибутов в нем. Атрибуты в отношении также, как и кортежи, не упорядочены.

Ключ – неизбыточное подмножество атрибутов в схеме отношения, которые однозначно идентифицируют его кортежи.
Ключ может быть простым (из одного атрибута) или составным (из нескольких атрибутов).

В отношении может быть более одного ключа. Каждый из этого множества ключей называют потенциальным, а наиболее предпочтительный из них - первичным.
В общем случае ключом может оказаться вся схема отношения.

Соответствие реляционных и табличных терминов

Реляционный термин	Табличный термин
Отношение	Таблица
Схема отношения	Набор столбцов
Тело отношения	Содержимое таблицы
Атрибут	Наименование столбца
Кортеж отношения	Строка
Степень отношения	Количество столбцов
Кардинальное число отношения	Количество строк
Скаляр	Ячейка таблицы
Первичный ключ	Уникальный столбец

Структура реляционной модели

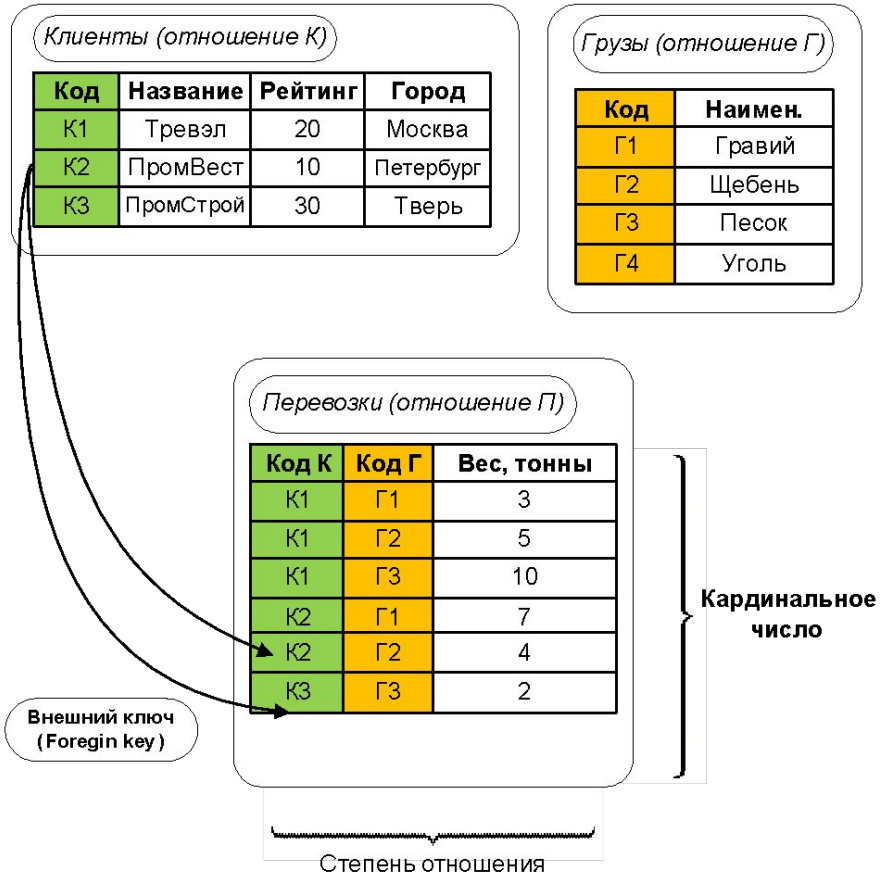
Отношение – единственное средство описания структуры данных в реляционной модели.

Базовое отношение – это поименованное отношение, которое является непосредственной частью БД. Базовые отношения реально существуют и действительно хранятся в БД.

Производное отношение – это отношение, определенное посредством реляционного выражения через другие отношения (чаще всего базовые). Такие отношения реально не существуют, а просто предоставляют различные способы просмотра данных.

Хранимое отношение – это отношение, которое поддерживается непосредственно в физической памяти. Хранимое отношение не всегда совпадает с базовым, однако набор хранимых отношений должен быть таким, чтобы из него можно было произвести все базовые отношения.

Реляционная модель «Перевозка сыпучих грузов»



Целостность данных

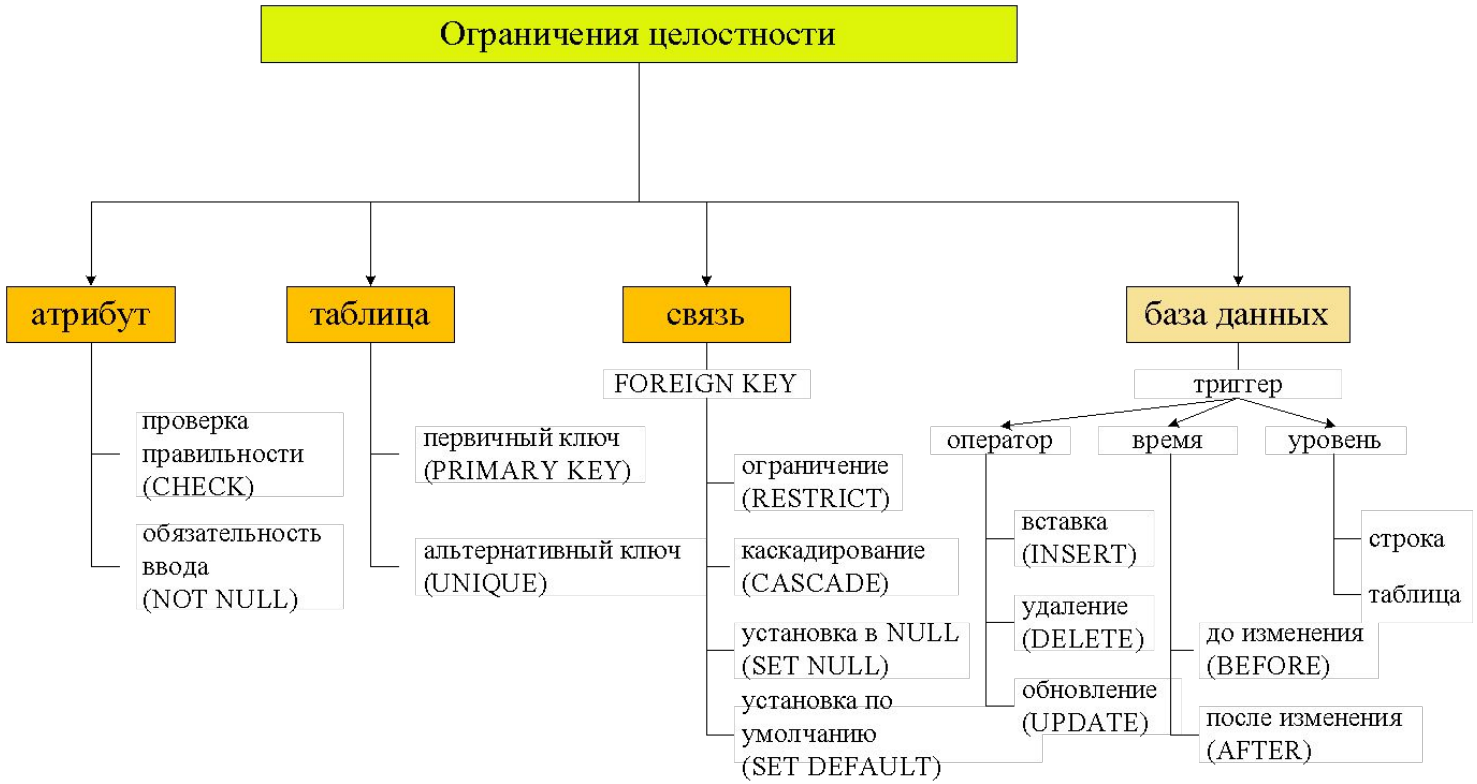
Целостность – это свойство данных, характеризующее их достоверность и непротиворечивость в любом состоянии БД.

Целостность данных в базе может нарушиться в следующих случаях:

- внесение неправильных данных;
- присвоение некорректных значений;
- потеря данных;
- внесение несогласованных данных и т.д.

Ограничения целостности (integrity constraint) обеспечиваются набором логических операции и процедур, которые реализуются средствами конкретной СУБД.

Классификация ограничений целостности



По способу реализации ограничения могут быть:

- декларативными (для атрибута, таблицы и связи);
- процедурными (для базы данных).

Декларативные ограничения целостности

Ограничениями целостности для атрибута являются проверка правильности (CHECK) и его обязательность (NOT NULL).

Целостность сущности (таблицы) предполагает, что имеется первичный ключ, который предотвращает ввод повторяющихся значений и обеспечивает однозначную идентификацию каждой строки.

Ссылочная целостность обеспечивает контроль связей между таблицами и опирается на понятие внешнего ключа (foreign key).

Внешний ключ – это множество атрибутов (не обязательно ключевых) в одном базовом отношении, являющееся первичным ключом другого базового отношения. Упомянутая ссылочная целостность предполагает отсутствие в отношениях несогласованных значений внешних ключей, т. е. значения внешнего ключа должны быть подмножеством значений соответствующего первичного ключа.

Стратегии ссылочной целостности

RESTRICT – запрещает выполнение операций, приводящих к нарушению ссылочной целостности.

CASCADE – разрешает выполнение требуемой операции, но требует внести при этом необходимые поправки в другие отношения так, чтобы не допустить нарушения ссылочной целостности и сохранить все имеющиеся связи. Изменение начинается в родительском отношении и каскадно выполняется в дочернем отношении.

SET NULL – разрешает выполнение операции, но все возникающие некорректные значения внешних ключей устанавливает неопределёнными.

SET DEFAULT – разрешает выполнение требуемой операции, но все возникающие некорректные значения внешних ключей устанавливает в некоторое значение, принятое по умолчанию.

Процедурные ограничения целостности

Ограничения целостности базы данных – условия, накладываемые на значения двух или более связанных между собой отношений.

К моменту проверки ограничений БД должны быть выполнены условия целостности отношений. Ограничения целостности для БД задаются с помощью **триггеров** – специальных программ, которые выполняются, когда происходит определённое событие (вставка, удаление или изменение данных).

Операторы **INSERT**, **UPDATE** или **DELETE** определяют, какая из соответствующих операций DML вызывает активизацию триггера.

Ключевые слова **BEFORE** или **AFTER** обуславливают момент активизации триггера (до или после выполнения указанного оператора).

Действие триггера может распространяться на строку или на таблицу в целом. В первом случае он активизируется для каждой из строк, на которые воздействует оператор, а во втором случае – один раз (до или после оператора).

По времени контроля ограничение целостности БД может быть **проверяемым немедленно** или **отложенными**. Первые проверяются сразу после совершения изменения, а вторые осуществляются по окончании операций, предназначенных для перевода БД из одного непротиворечивого состояния в другое через некоторые противоречивые состояния.

Трёхзначная логика и NULL-значения

NULL – специальный символ, показывающего, что значение атрибута пусто или не определено в данный момент.

Проблема отсутствующей информации и подход к её решению базируются на трёхзначной логике (three-valued logic, 3VL): результатом любых операций сравнения, в которых один из операндов содержит Null-значение, служит третье логическое значение – неопределён (F – False, T–True и U– Unknow).

Таблицы истинности для трехзначной логики:

AND	F	T	U
F	F	F	F
T	F	T	U
U	F	U	U

OR	F	T	U
F	F	T	U
T	T	T	T
U	U	T	U

NOT	
F	T
T	F
U	U

Парадоксы трёхзначной логики

Примеры парадоксов, связанных с трехзначной логикой:

1. Выражение $Null = Null$ дает значение НЕИЗВЕСТНО, а не ИСТИНА, т. е. выражение не обязательно ИСТИНА.

2. Выражение $Null \neq Null$ принимает значение не ИСТИНА, а НЕИЗВЕСТНО, т. е. выражение $x \neq x$ тоже не обязательно ЛОЖЬ.

3. Выражение a OR NOT(a) не обязательно ИСТИНА. Значит, в трехзначной логике не работает принцип исключенного третьего (любое высказывание либо истинно, либо ложно).

4. и др.

Обработка Null-значений

1. Ограничиться использованием обычных типов данных и не использовать Null-значения, а вместо неизвестных данных вводить либо нулевые значения, либо значения специального вида, которые затем обрабатывать специальным образом.

2. Использовать Null-значения вместо неизвестных данных, принимая во внимание все неоднозначности их дальнейшей обработки.

Реляционная алгебра

Реляционная алгебра – замкнутая система операций над отношениями в реляционной модели данных.

Первоначально Э.Ф. Кодд предложил 8 операторов:

- проекция
- выборка
- объединение
- вычитание
- пересечение
- декартово произведение
- соединение (естественное)
- деление

самостоятельно см. лекцию «Реляционная алгебра».

12 правил Э.Кодда (1985 г)

Правило 0: Основное правило (Foundation Rule):
Реляционная СУБД должна быть способна полностью управлять базой данных, используя связи между данными:
Чтобы быть реляционной системой управления базами данных (СУБД), система должна использовать исключительно свои реляционные возможности для управления базой данных.

Правило 1: Явное представление данных (The Information Rule):
Информация должна быть представлена в виде данных, хранящихся в ячейках. Данные, хранящиеся в ячейках, должны быть атомарны. Порядок строк в реляционной таблице не должен влиять на смысл данных.

Правило 2: Гарантированный доступ к данным (Guaranteed Access Rule):
Доступ к данным должен быть свободен от двусмысленности. К каждому элементу данных должен быть гарантирован доступ с помощью комбинации имени таблицы, первичного ключа строки и имени столбца.

12 правил Э.Кодда (продолжение)

Правило 3: Полная обработка неизвестных значений (Systematic Treatment of Null Values):
Неизвестные значения NULL, отличные от любого известного значения, должны поддерживаться для всех типов данных при выполнении любых операций. Например, для числовых данных неизвестные значения не должны рассматриваться как нули, а для символьных данных — как пустые строки.

Правило 4: Доступ к словарю данных в терминах реляционной модели (Active On-Line Catalog Based on the Relational Model):
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.

Правило 5: Полнота подмножества языка (Comprehensive Data Sublanguage Rule):
Система управления реляционными базами данных должна поддерживать хотя бы один реляционный язык, который

a) имеет линейный синтаксис,
b) может использоваться как интерактивно, так и в прикладных программах,
c) 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 Э.Кодда (продолжение)

Правило 6: Возможность модификации представлений (View Updating Rule):
Каждое представление должно поддерживать все операции манипулирования данными, которые поддерживают реляционные таблицы: операции выборки, вставки, модификации и удаления данных.

Правило 7: Наличие высокоуровневых операций управления данными (High-Level Insert, Update, and Delete):
Операции вставки, модификации и удаления данных должны поддерживаться не только по отношению к одной строке реляционной таблицы, но по отношению к любому множеству строк.

Правило 8: Физическая независимость данных (Physical Data Independence):
Приложения не должны зависеть от используемых способов хранения данных на носителях, от аппаратного обеспечения компьютеров, на которых находится реляционная база данных.

12 правил Э.Кодда (продолжение)

Правило 9: Логическая независимость данных (Logical Data Independence):
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.

Правило 10: Независимость контроля целостности (Integrity Independence):
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.

Правило 11: Дистрибутивная независимость (Distribution Independence):
База данных может быть распределённой, может находиться на нескольких компьютерах, и это не должно оказывать влияние на приложения. Перенос базы данных на другой компьютер не должен оказывать влияния на приложения.

Правило 12: Согласование языковых уровней (The Nonsubversion Rule):
Если используется низкоуровневый язык доступа к данным, он не должен игнорировать правила безопасности и правила целостности, которые поддерживаются языком более высокого уровня.

Связь ER-модели и реляционной модели

- сущности представляются отношениями явно;
- связи представляются неявно:
  - посредством связующих атрибутов (связь типа 1:1 и 1:m);
  - отдельным отношением (связь типа m:n).

1. Идентифицирующая:

Степень связи и класс принадлежности:

1:0,1,m
1:1,m
1:0,1
1:1

Конструкция ER-модели:

Таблицы реляционной модели:

R1 (**k1**,a2i)
R2 (**k2**,**k1**,a2i)

Связь ER-модели и реляционной модели (продолжение)

2. Неидентифицирующая:

Степень связи и класс принадлежности:

1:0,1,m
1:1,m
1:0,1
1:1

Конструкция ER-модели:

Таблицы реляционной модели:

R1 (**k1**,a2i)
R2 (**k2**,k1,a2i)

Связь ER-модели и реляционной модели (продолжение)

3. Многие ко многим:

Степень связи и класс принадлежности:

n:m

Конструкция ER-модели:

Таблицы реляционной модели:

R1 (**k1**,a2i)
R2 (**k2**,a2i)
R2 (**k2**,**k1**,a3i)

Связь ER-модели и реляционной модели (продолжение)

3. Многие ко многим:

Степень связи и класс принадлежности:

n:m

Конструкция ER-модели:

Таблицы реляционной модели:

R1 (**k1**,a2i)
R2 (**k2**,a2i)
R2 (**k2**,**k1**,a3i)

Связь ER-модели и реляционной модели (продолжение)

4. Категоризация:

Конструкция ER-модели:

Таблицы реляционной модели:

R1 (**k1**,a1i)
R2 (**k1**,a2i)
R3 (**k1**,a3i)
R4 (**k1**,a4i)

Преимущества и недостатки реляционной модели

Преимущества:

1. Универсальность построения модели и простота её восприятия.
2. Высокая степень формализации в описании модели с опорой на существующий математический аппарат (теория множеств и предикативная логика).
3. Компактное формулирование запросов в терминах более крупных агрегатов данных.

Недостатки:

Трудности совмещения языка запросов реляционной СУБД с традиционными языками программирования, ориентированными на «позаписную» обработку данных (при обеспечении интерфейса).