

Нормализация

Это процесс, направленный на
снижение избыточности
информации в реляционных
базах данных.





Цель нормализации

- получение такого проекта БД, в котором каждый факт появляется лишь в одном месте (*т.е.исключена избыточность данных*).

Метод нормализации впервые был предложен Э.Ф. Коддом в 1972г.

Процесс нормализации



- Включает последовательность тестов для проверки соответствия (*или несоответствия*) отношения требованиям заданной нормальной формы.
- Сначала были предложены: первая (1НФ), вторая (2НФ) и третья (3НФ).
- Затем Р. Бойсом и Э. Ф. Коддом (1974) было сформулировано более строгое определение 3НФ - нормальная форма Бойса-Кодда (НФБК).
- Затем появились определения четвертой (4НФ) и пятой (5НФ) нормальных форм (1977, 1979).



В основу нормализации положена концепция функциональных зависимостей.

Нормализация отношений модели предметной области позволяет создать логическую модель реляционной БД.



С математической точки зрения задача построения модели предметной области (и затем логической модели РБД) является результатом решения комбинаторных задач:

- группировка атрибутов в отношении предметной области;
- распределение атрибутов по отношениям базы данных.

Выбор наиболее рационального варианта обусловлен соблюдением различных соглашений и требований.

Наиболее важные требования



1. Первичные ключи отношений должны быть минимальными.
2. Число отношений БД должно по возможности давать наименьшую избыточность данных.
3. Число отношений БД не должно приводить к потере производительности системы.
4. Данные не должны быть противоречивыми, т.е. при выполнении операций вставки, удаления и обновления данных их потенциальная противоречивость должна быть минимальна.
5. Схема отношений БД должна быть устойчивой, способной адаптироваться к изменениям при ее расширении дополнительными атрибутами.
6. Разброс времени реакции на различные запросы к БД не должен быть большим.
7. Данные должны правильно отражать состояние предметной области БД в каждый конкретный момент времени.

Однако...



Создание системы, одновременно удовлетворяющей всем требованиям, представляет собой сложную оптимизационную задачу.

Многие из требований находятся в противоречии друг к другу, например:

- требование производительности находится в противоречии к требованию гибкости;
- требование минимизации числа отношений в БД находится в противоречии к требованию надежности данных.



- При хранении данных в РБД преследуют две основные цели: 1) понизить избыточность данных; 2) повысить их достоверность.
- Для этого необходимо накладывать ограничения на хранимые данные, на взаимосвязи и взаимозависимости между ними.
- Обычно эти ограничения вытекают из анализа предметной области.
- Одним из средств формализации результатов такого анализа являются зависимости между данными.



Функциональные зависимости

- Пусть $r(A_1, A_2, \dots, A_n)$ - схема отношения R , а X и Y - подмножества r .
- Говорят, что X функционально определяет Y , если каждому значению атрибутов кортежа из X соответствует не более одного значения атрибутов того же кортежа из Y :

$$F: X \rightarrow Y.$$

- Функциональная зависимость инвариантна к изменению состояний БД во времени.

Продemonстрируем понятие ФЗ на примере графика полетов аэропорта.



ГРАФИК_ПОЛЕТОВ (Пилот, Рейс, Дата_вылета, Время_вылета)

Ивано В	100	8.07	10:20
Ивано В	102	9.07	13:30
Исаев	90	7.07	6:00
Исаев	100	11.07	10:20
Исаев	103	10.07	19:30
Петро В	100	12.07	10:20
Петро В	102	11.07	13:30
Фроло В	90	8.07	6:00

Известно, что:

- 1) каждому рейсу соответствует определенное время вылета;
- 2) для каждого пилота, даты и времени вылета возможен только один рейс;
- 3) на определенный день и рейс назначается определенный пилот.



Продолжение примера

ГРАФИК_ПОЛЕТОВ (Пилот, Рейс, Дата_вылета, Время_вылета)

- "Время_вылета" функционально зависит от "Рейс":

Рейс → Время_вылета;

- "Рейс" функционально зависит от ("Пилот", "Дата_вылета", "Время_вылета"):

("Пилот", "Дата_вылета", "Время_вылета") → "Рейс";

- "Пилот" функционально зависит от ("Рейс", "Дата_вылета"):

("Рейс", "Дата_вылета") → "Пилот".



F-зависимость

Пусть имеется отношение R со схемой r , X и Y - два подмножества R .

ФЗ $F: X \rightarrow Y$ имеет место на R , если множество

$$\pi_Y (\sigma_{X=x} (R))$$

имеет не более одного кортежа для каждого значения x .

Такая ФЗ называется F-зависимостью.



T-зависимости

Пусть X, Y, Z - атрибуты отношения R .

- Если имеются ФЗ $F: X \rightarrow Y$ и $F: Y \rightarrow Z$,
- но отсутствуют ФЗ $F: Z \rightarrow Y$ и $F: Y \rightarrow X$,

то Z транзитивно зависит от X .

Такие ФЗ называются транзитивными (**T-зависимостями**).

T-зависимость возникает в случае, если неключевой атрибут зависит от другого неключевого атрибута.

Пример T-зависимости



СОТРУДНИКИ (Табельный_номер, ФИО, Должность, Оклад)

Табельный номер сотрудника определяет его должность, т.е. имеет место ФЗ

Табельный_номер → Должность.

Согласно тарификации каждой должности назначается определенный оклад, т.е. имеем ФЗ

Должность → Оклад.

T-зависимость: оклад сотрудника определяется через его должность.

При эксплуатации РБД



- происходит модификация данных - добавляются или удаляются кортежи, изменяются значения некоторых атрибутов кортежей и т.п.
- данные в БД становятся несогласованными и противоречивыми:
 - значения некоторых атрибутов могут дублироваться (избыточность данных),
 - значения некоторых атрибутов могут удаляться, хотя потребность в этих значениях остается,
 - некоторые значения атрибутов не могут быть добавлены, поскольку их значения неизвестны.
- использование ненормализованных таблиц может привести к нарушению целостности данных в БД.



Потенциальная противоречивость данных - это аномалии обновления, вставки, удаления.

Нормализация позволяет предотвратить возможность возникновения этих аномалий модификации данных.

Аномалии могут быть устранены путем разбиения исходного отношения на два или более новых отношения.

Пример аномалии обновления



Пример отношения:

СОТРУДНИКИ (Код, ФИО, Должность, ДомТелефон, Адрес)

- Пусть один и тот же сотрудник занимает две должности
- Пусть у него изменился номер телефона
- Изменения нужно вносить в две записи. Иначе возникнет несоответствие информации (аномалия обновления).
- Избыточность проявляется в том, что в двух записях таблицы БД повторяется одна и та же информация.

Пример аномалии вставки



Возникает при добавлении в таблицу новых записей.

Пример:

СОТРУДНИКИ (Код, ФИО, Должность, Отдел,
Руководитель, Тел_отдела)

При вставке сведений о сотрудниках нового отдела необходимо указать и сведения об этом отделе.

Нельзя записать в таблицу некоторый факт об одной сущности, не указав дополнительно некоторый факт о другой сущности.



Пример аномалии удаления

Удаляя информацию об одной сущности, мы теряем факты, относящиеся к другой сущности.

Пример:

СОТРУДНИКИ (Код, ФИО, Должность, Отдел, Руководитель, Тел_отдела)

При удалении строки с информацией о последнем сотруднике некоторого отдела сведения об этом отделе будут полностью удалены из БД.

При удалении одной строки происходит удаление информации о двух сущностях.

Суть нормализации



- Каждое нормализованное отношение должно иметь одну-единственную тему.
- Любое отношение, затрагивающее две или более темы, следует разбить на два или более отношения, у каждого из которых будет только одна тема.



Классы отношений и способы предотвращения аномалий называются нормальными формами.

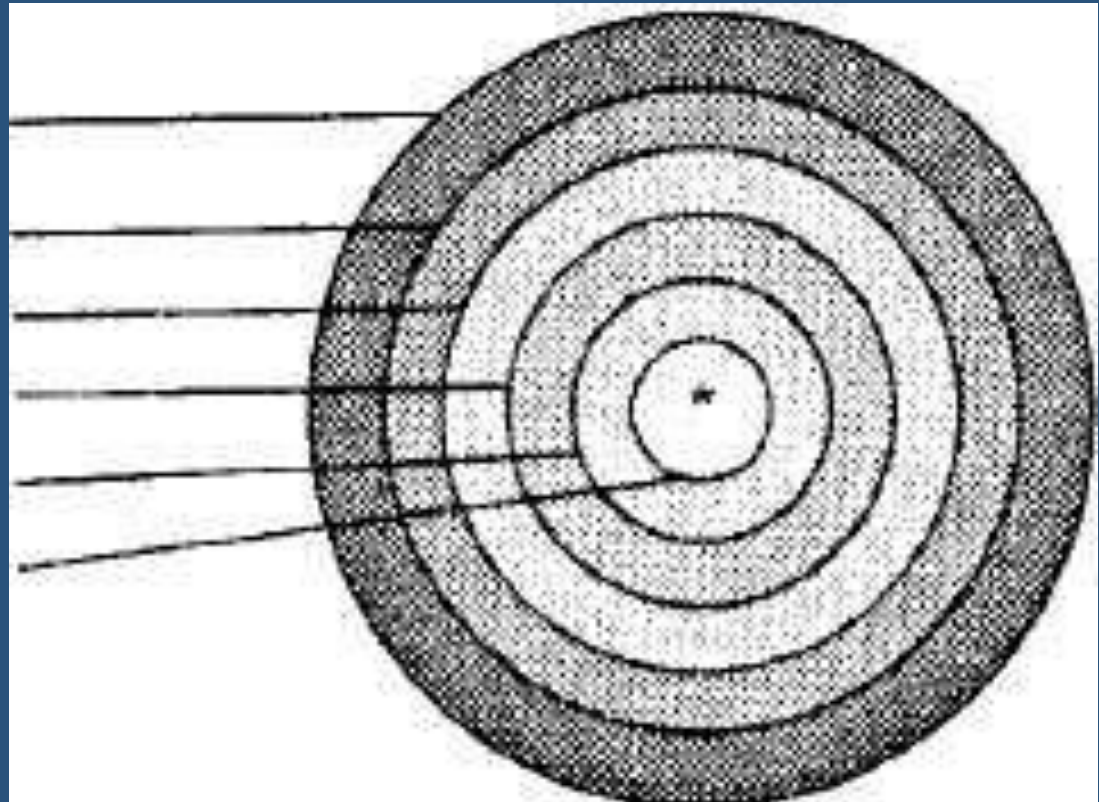
В зависимости от своей структуры, отношение может быть в 1НФ, 2НФ или другой НФ.

Взаимосвязь нормальных форм



Нормальные формы как бы вложены друг в друга.

- 1НФ
- 2НФ
- 3НФ
- НФБК
- 4НФ
- 5НФ



1НФ



- Любая таблица данных, удовлетворяющая определению отношения, находится в 1НФ:
 - На пересечении каждой строки и каждого столбца содержится только одно значение.
 - Имеется первичный ключ, значения которого однозначно идентифицируют каждую строку.
- Решение: Приведение отношения к 1НФ - это вынесение неатомарных атрибутов в отдельное подчиненное отношение, т.е. получение двух проекций. Назначение первичного ключа.

2НФ



- 2НФ - это отношение, которое находится в 1НФ, а каждый неключевой атрибут функционально полно зависит от первичного ключа.
- Если отношение имеет в качестве ключа единственный атрибут, то оно автоматически находится в 2НФ.
- Решение: Если между атрибутами существует частичная зависимость, то функционально-зависимые атрибуты удаляются из отношения и помещаются в новое отношение вместе с копией их детерминанта.

Пример приведения таблицы к 2НФ



Пусть первичный ключ **Сотрудник + Должность**:

Сотрудник	Должность	Зарплата	Наличие ПК
Гришин	Кладовщик	20000	Нет
Васильев	Программист	40000	Да
Васильев	Кладовщик	25000	Нет

Наличие ПК у сотрудника зависит только от должности, т.е. зависимость от первичного ключа неполная.

В результате приведения к 2НФ получаются две таблицы:

Сотрудник	Должность	Зарплата
Гришин	Кладовщик	20000
Васильев	Программист	40000
Васильев	Кладовщик	25000

Должность	Наличие ПК
Кладовщик	Нет
Программист	Да

3НФ



- Таблица находится в 3НФ, когда выполняются следующие условия:
 - Отношение R (таблица) находится в 2НФ;
 - Каждый неключевой атрибут R находится в нетранзитивной (*т.е. прямой*) зависимости от ключа.

3NF



- Нормализация 2NF-отношений с получением 3NF-отношений включает устранение транзитивных зависимостей.
- Решение: При наличии Т-зависимостей следует удалить транзитивно зависимые атрибуты из отношения и поместить в новое отношение вместе с копией их детерминанта.

Пример приведения к 3НФ



Исходная таблица

Фамилия	Отдел	Телефон
Гришин	1	11-22-33
Васильев	1	11-22-33
Петров	2	44-55-66

В результате приведения к 3НФ получаются две таблицы

Фамилия	Отдел
Гришин	1
Васильев	1
Петров	2

Отдел	Телефон
1	11-22-33
2	44-55-66

Нормальная форма Бойса—Кодда



Учитывает ФЗ, в которых участвуют все потенциальные ключи отношения, а не только его первичный ключ.

Для отношения выполняются условия:

- 1) имеется два (или более) потенциальных ключа;
- 2) два потенциальных ключа являются составными;
- 3) они перекрываются (т.е. имеют, по крайней мере, один общий атрибут).

Нормальная форма Бойса—Кодда



- Отношение находится в НФБК тогда и только тогда, когда каждый его детерминант является потенциальным ключом.

Пример



СОБЕСЕДОВАНИЕ (Клиент, Дата, Время, Сотрудник, Комната)

Для проведения собеседования в распоряжение сотрудника предоставляется особая комната. В течение дня эта комната может использоваться несколькими разными сотрудниками. С клиентом проводится только одно собеседование в день, но он может участвовать в нескольких собеседованиях в разные дни.

Отношение **СОБЕСЕДОВАНИЕ** имеет 3 составных потенциальных ключа, которые перекрываются (совместно используют общий атрибут Дата):

(Клиент, Дата), (Сотрудник, Дата, Время), (Комната, Дата, Время).

Продолжение примера



Отношение

СОБЕСЕДОВАНИЕ (Клиент, Дата, Время, Сотрудник,
Комната)

Находится в 1НФ. Почему?

Находится в 2НФ. Почему?

Находится в 3НФ. Почему?

Отношение **СОБЕСЕДОВАНИЕ** обладает аномалиями обновления данных. Почему?

ФЗ: (Дата, Сотрудник) → Комната

Однако, детерминант не является потенциальным ключом.

Приведение к НФБК



Отношение

**СОБЕСЕДОВАНИЕ (Клиент, Дата, Время,
Сотрудник, Комната)**

разбивается на 2 отношения:

1) **ИНТЕРВЬЮ (Клиент, Дата, Время,
Сотрудник)**

2) **КОМНАТА (Сотрудник, Дата, Комната)**

Для проверки принадлежности отношения к НФБК . . .



**необходимо найти все его детерминанты
и убедиться в том, что они являются
потенциальными ключами.**



Нарушения требований НФБК происходят крайне редко, поскольку это может случиться только тогда, когда:

- имеются два (или более) составных потенциальных ключа;
- эти потенциальные ключи перекрываются, т.е. ими совместно используется, по крайней мере, один общий атрибут.



Спасибо за внимание!