Структуры базы данных. Нормализация

Наумов Д.А., доц. каф. КТ

Основы компьютерных наук (3 часть), 2019

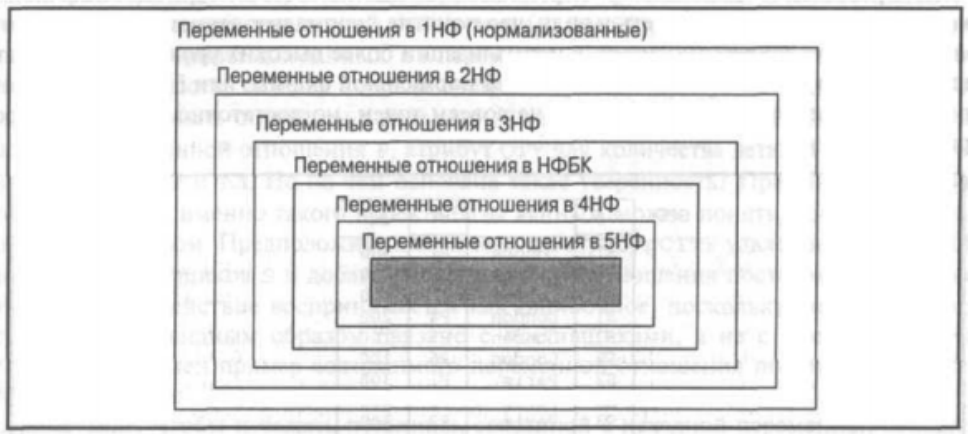
Содержание лекции

1. Нормализация
2. Типы приложений: транзакционная и аналитическая обработка
3. Денормализация

- Нормализация схемы реляционной БД оказывает существенное влияние буквально на все аспекты взаимодействия с БД: от затрат на модификацию структур и данных до производительности запросов приложений и хранимых объёмов информации.
- В ряде случаев структуры могут быть сознательно денормализованы.
- Нормализация — не догма, но чтобы её нарушать, нужны основания
- На практике проектирования схем баз данных достижение третьей нормальной формы (3НФ) считается достаточным условием для большинства случаев.

Основные свойства нормальных форм

- каждая следующая нормальная форма в некотором смысле улучшает свойства предыдущей;
- при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются;



Схемы БД называются эквивалентными

если содержание исходной БД может быть получено путем эквивалентного соединения отношений, входящих в результирующую схему, и при этом не появляется новых кортежей.

1НФ - первая нормальная форма

выполняется, если все значения атрибутов (колонок таблицы) атомарны, то есть неделимы.

- собственные типы данных СУБД считаются атомарными, исключение могут составлять массивы, в том числе символьные (текстовые) и байтовые.
- атомарность может быть относительна выбранного взгляда со стороны предметной области и контекста.

Пример 1: телефонный номер (в базе данных маркетинга, у телефонных операторов), колонки для хранения комментариев, целая и дробная части действительного числа, дата-время.
Пример 2: фамилия, имя, отчество в одной колонке.

Пример

Преподаватель	День недели	Номер пары	Название дисциплины	Тип занятий	Группа
Петров В. И.	Понед.	1	Теор.выч.проц.	Лекция	4906
	Вторник	1	Комп.графика	Лаб.раб.	4907
	Вторник	2	Комп.графика	Лаб.раб.	4906
Киров В. А.	Понед.	2	Теор.информ.	Лекция	4906
	Вторник	3	Пр-е на C++	Лаб.раб.	4907
	Вторник	4	Пр-е на C++	Лаб.раб.	4906
Серов А. А.	Понед.	3	Защита информ.	Лекция	4944
	Среда	3	Пр-е на VB	Лаб.раб.	4942
	Четверг	4	Пр-е на VB	Лаб.раб.	4922

Пример

Преподаватель	**День недели**	**Номер пары**	**Название дисциплины**	**Тип занятий**	**Группа**
Петров В. И.	Понед.	1	Теор.выч.проц.	Лекция	4906
Петров В. И.	Вторник	1	Комп.графика	Лаб.раб.	4907
Петров В. И.	Вторник	2	Комп.графика	Лаб.раб.	4906
Киров В. А.	Понед.	2	Теор.информ.	Лекция	4906
Киров В. А.	Вторник	3	Пр-е на C++	Лаб.раб.	4907
Киров В. А.	Вторник	4	Пр-е на C++	Лаб.раб.	4906
Серов А. А.	Понед.	3	Защита информ.	Лекция	4944
Серов А. А.	Среда	3	Пр-е на VB	Лаб.раб.	4942
Серов А. А.	Четверг	4	Пр-е на VB	Лаб.раб.	4922

2НФ - вторая нормальная форма

означает, что выполнены требования 1НФ, при этом все атрибуты целиком зависят от составного ключа и не зависят ни от какой его части.

- в определении говорится о ключах вообще, а не только о первичных.
- в отношении может быть несколько ключей, и некоторые из них могут являться составными

Пример 1: Описание продаж товара.
Пример 2: Сдача сессии (ФИО, №Зачетки, Группа, Дисциплина, Оценка).
Пример 3: Ключ - атрибут не выделенной еще сущности.

3НФ - вторая нормальная форма

означает, что выполнены требования 2НФ, при этом в между атрибутами отношения нет транзитивных зависимостей.

Пример: продажа каждой товарной позиции имеет своим основанием документ (заказ, счёт и т.д.), а её стоимость характеризуется ценой, количеством и валютой.

Транзитивные зависимости:

- Идентификатор продажи $\rightarrow$ Номер документа
- Идентификатор продажи $\rightarrow$ Код валюты
- Номер документа $\rightarrow$ Код валюты

Результатом нарушения 3НФ является избыточность хранения и необходимость обновления данных в связанной таблице.

Пример: (ФИО, Номер зачетки, Группа, Факультет, Специальность, Выпускающая кафедра)

Пример

Номер зачетки - > ФИО
Номер зачетки - > Группа
Номер зачетки - > Факультет
Номер зачетки - > Специальность
Номер зачетки -> Выпускающая кафедра
Группа - > Факультет
Группа -> Специальность
Группа - > Выпускающая кафедра
Выпускающая кафедра - > Факультет

OLTP (On-Line Transaction Processing) — интерактивная транзакционная обработка.

- Обработка идёт в режиме реального или приближенного к реальному времени.
- Запросы представляют собой интенсивный поток коротких операций по вставке, изменению и удалению небольшого числа записей в БД.

OLAP (On-Line Analytical Processing) - интерактивная аналитическая обработка.

- Данные находятся **в режиме чтения**, за исключением моментов их обновления.
- Выборки представляют собой **одиночные тяжёлые запросы**: поиски и расчёты по множеству произвольных критериев.
- **Время отклика системы не регламентировано**.
- Размеры базы данных на порядок и больше транзакционных.

Цели нормализации:

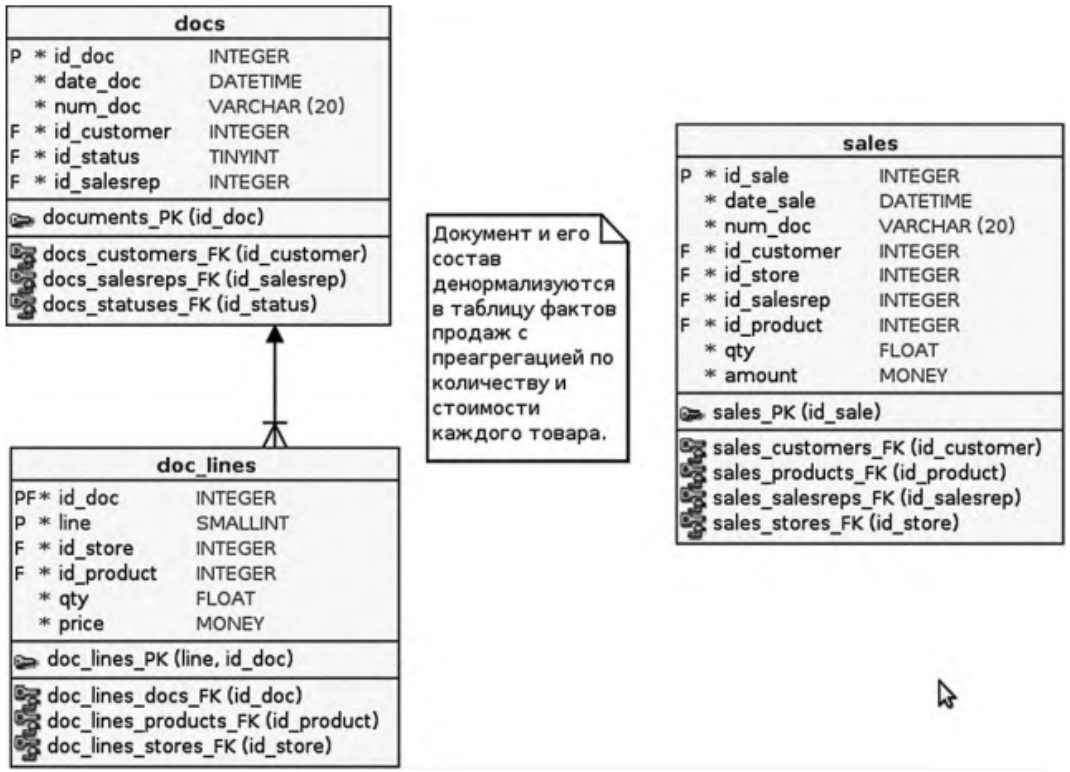
- устранение избыточности при хранении данных, приводящей к увеличению размера БД.
- исключение необходимости модификации данных в связных таблицах для минимизации времени и операций, проводящихся в одной транзакции.

В приложениях интерактивной аналитической обработки приоритет меняется: на первый план выходит время отклика системы, в ущерб которому данные могут быть избыточны.

Допустимые схемы для OLAP: "снежинка "звезда":

- центральным элементом являются таблицы фактов, содержащие события, транзакции, документы и др.
- в таблице фактов одному документу (каждой его строке), соответствует одна запись.

Денормализация документов в таблицу фактов



Пример SQL-запроса в транзакционной СУБД

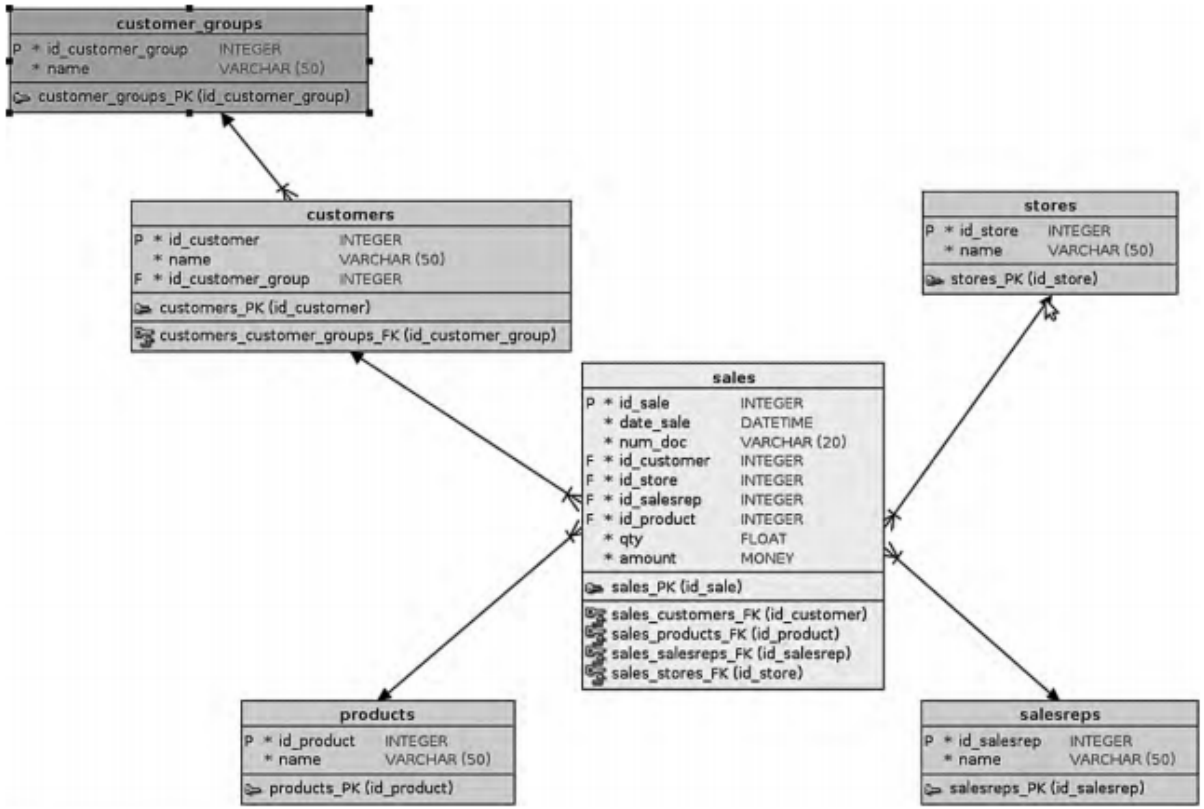
```
SELECT
  SUM(dl.qty) AS total qty, SUM(dl.price) AS total amount,
c.name
FROM
  docs d
  INNER JOIN doc_lines dl ON d.id_doc = dl.id_doc
  INNER JOIN customers c ON d.id_customer = c.id_customer
  INNER JOIN products p ON dl.id_product = p.id_product
WHERE
  c.name IN ('ООО Пирожки', 'ЗАО Ватрушки') AND
  p.name = 'Мука' AND
  d.date BETWEEN '2014-01-01' AND '2014-02-01'
GROUP BY c.name
```

Пример SQL-запроса в OLAP-СУБД

```
SELECT
  SUM(s.qty) AS total_qty, SUM(s.amount) AS total_amount,
c.name
FROM
  sales s
  INNER JOIN customers c ON d.id_customer = c.id_customer
  INNER JOIN products p ON dl.id_product = p.id_product
WHERE
  c.name IN ('ООО Пирожки', 'ЗАО Ватрушки') AND
  p.name = 'Мука' AND
  s.date BETWEEN '2014-01-01' AND '2014-02-01'
GROUP BY c.name
```

Если таблица фактов ссылается на таблицы-измерения, имеющие ссылки на другие измерения, то такая схема называется "снежинка".

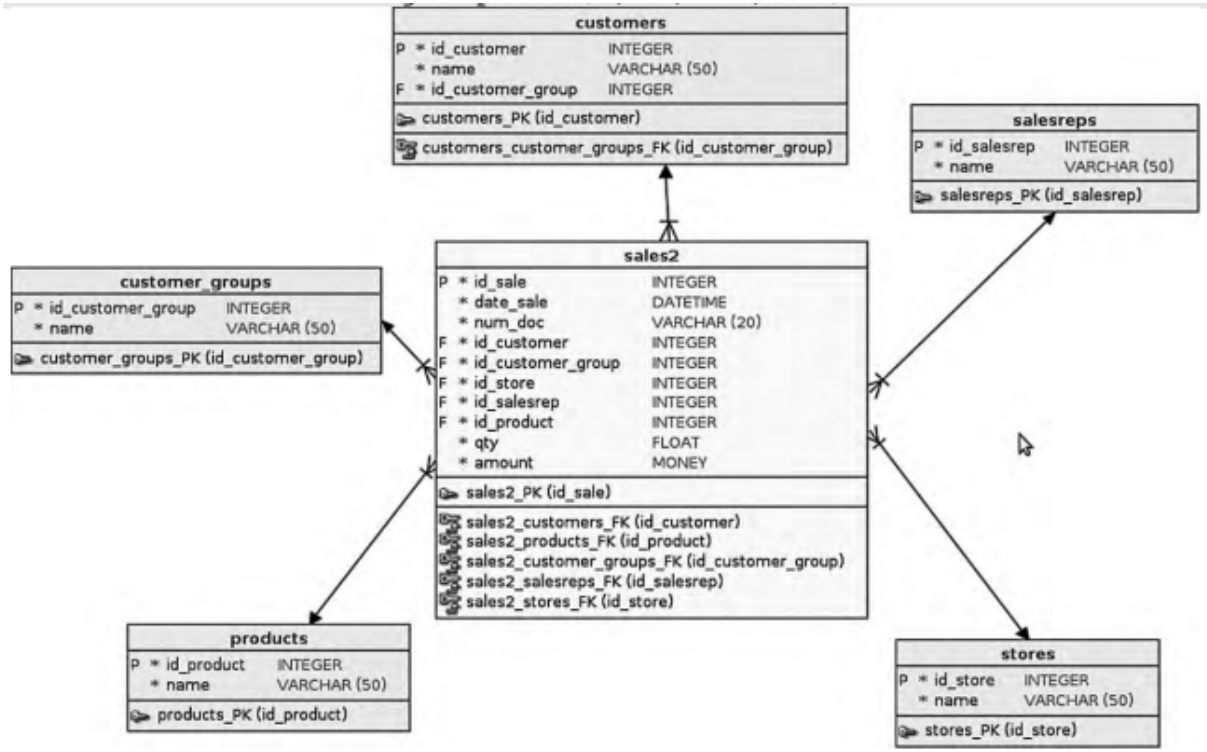
Таблица фактов в схеме "снежинка"



Для запросов, включающих фильтрацию по группам клиентов, приходится делать дополнительное соединение.

Схема "Звезда"полностью исключает иерархию измерений и необходимость соединения соответствующих таблиц в одном запросе.

Таблица фактов в схеме "звезда"



Обратной стороной денормализации всегда является избыточность, являющаяся причиной увеличения размера БД как в случае транзакционных, так и аналитических приложений.

Пример SQL-запроса в схеме "снежинка"

```
SELECT sum(amount)
FROM sales s
WHERE s.id customer group IN (1, 2, 10, 55)
```

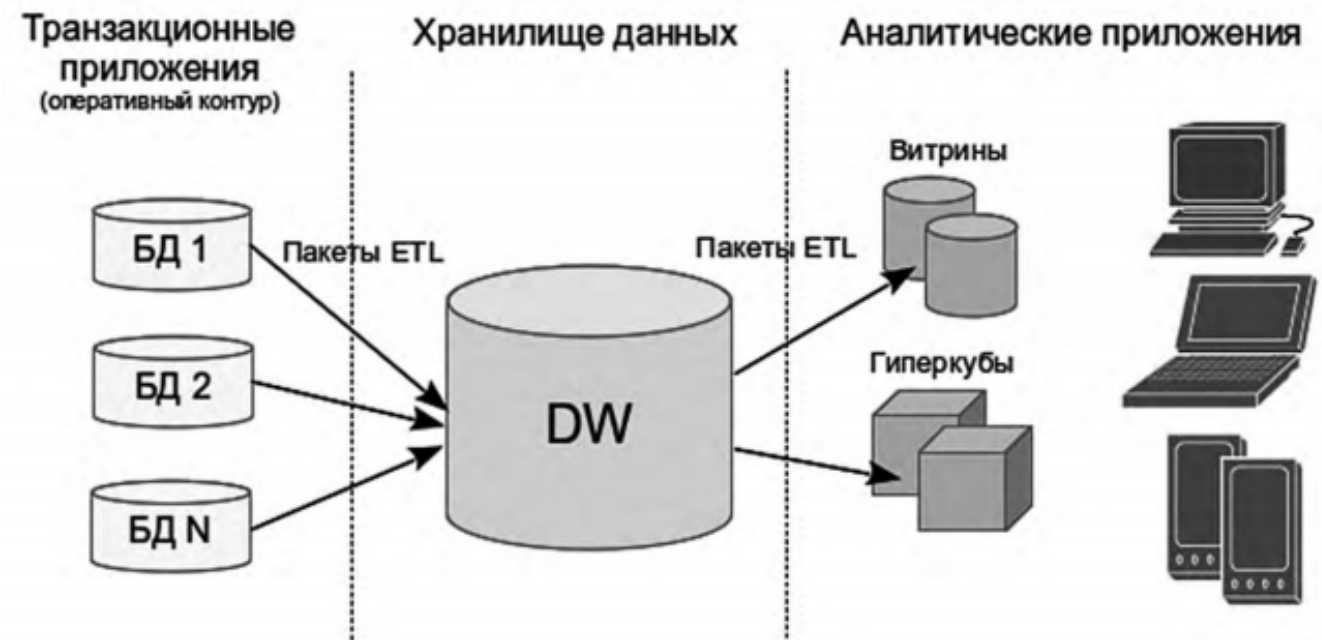
Пример SQL-запроса в схеме "звезда"

```
SELECT sum(amount)
FROM sales s
     INNER JOIN customers c ON s.id_customer = c.id_customer
WHERE c.id_customer_group IN (1, 2, 10, 55)
```

Задача

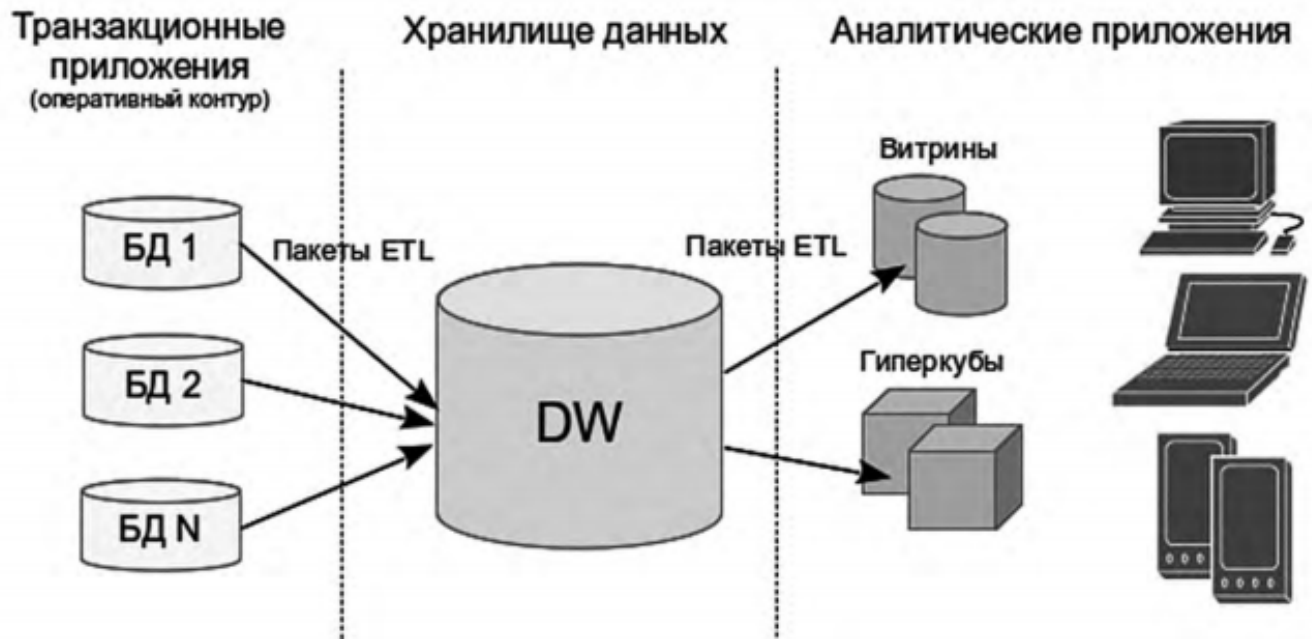
Посчитаем, примерную дельту на приведённом выше примере преобразования "снежинки"в "звезду если таблица продаж не использует компрессию данных и содержит около 500 миллионов строк, а количество групп покупателей порядка 1000.

Пример типовой архитектуры OLAP



- тяжёлые запросы по произвольным критериям выборки.
- разделение собственно хранилища (data warehouse), где данные представлены в минимально агрегированном виде, и конечных БД, адаптированных под нужды пользователей разных профилей и предметных областей.

Пример типовой архитектуры OLAP



- в роли баз данных конечных пользователей могут выступать как вполне реляционные витрины данных (datamart), так и другие типы СУБД, прежде всего, основанные на многомерных моделях.
- витрина данных представляет собой реляционную БД или даже одну таблицу фактов, содержащую более агрегированную и отфильтрованную информацию, извлечённую непосредственно из хранилища данных.