ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

Основные понятия теории нормализации

В чем суть теории нормализации

Теория нормализации схем отношений реляционной модели данных – это теория, устанавливающая:

- ✓ **каким образом** исходная схема отношений может быть преобразована в другую результирующую схему, которая
- ✓ **эквивалентна** в некотором смысле исходной и
- ✓ **в каком-то смысле лучше** ее.

Таким образом эта теория должна ответить на следующие вопросы:

- ✓ какие существуют **критерии эквивалентности** схем отношений;
- ✓ какие существуют **критерии оценки качества** схем отношений;
- ✓ какие существуют **механизмы эквивалентных преобразований** схем отношений, дающие более качественные схемы.

Основные понятия теории нормализации

Нормализация отношений - это формализованный пошаговый процесс построения **оптимальной структуры таблиц и связей** в реляционной БД (процесс уменьшения избыточности информации или процесс **декомпозиции** исходных отношений БД на более простые).

Нормальная форма (НФ) - это теоретические правила, которым отвечает **схема отношения**.

Основные свойства нормальных форм:

1. каждая следующая нормальная форма (НФ) в некотором смысле лучше предыдущей;
2. при переходе к следующей НФ свойства предыдущих сохраняются.

В теории РБД принято выделять следующую последовательность нормальных форм:

- *первая нормальная форма (**1NF**);*
- *вторая нормальная форма (**2NF**);*
- *третья нормальная форма (**3NF**);*
- *нормальная форма Бойса-Кодда (**BCNF**);*
- *четвертая нормальная форма (**4NF**);*
- *пятая нормальная форма (**5NF** или **PJ/NF**);*

Первая нормальная форма (1НФ)

Первая нормальная форма требует, чтобы **домены** всех атрибутов базы данных содержали **только простые неделимые значения**, а **значением** атрибута в кортеже должно быть **одно значение из его домена**.

Или:

Каждый атрибут отношения должен содержать атомарные значения.

Первая нормальная форма (1НФ)

Пример1 . Дано отношение ***КНИГИ***:

ID – первичный ключ

ID	шифр рубрики	название рубрики	Автор	название книги	редактор	тип издания	год издания	кол-во страниц
2	1.3	ВТ	A1 A2	К1	Р1	учебник	2008	384
1	1.3	ВТ	А3	К2		учебник	2009	552
3	1.5	МО	А4	К3	Р2	учебник	2011	544
6	07	ИИ	А5	К4	Р3 Р4	учебное пособие	2003	176
4	2.9	ВТ		К5	Р5	справочник	2010	208
5	1.8	ПО	А6 А7	К6	Р6 Р7	учебное пособие	2009	304

ВТ – вычислительная техника;

ПО – программное обеспечение;

Основные понятия теории нормализации

Следствия несовершенной структуры отношения:

- **Аномалии вставки.** Нельзя вставить одни данные в связи с тем, что не определены другие данные.
- **Аномалии обновления**. Может возникнуть потеря целостности. в связи с избыточностью данных и их частичным обновлением.
- **Аномалии удаления.** При удалении одних данных можно потерять другие данные.

Причина! Такая ситуация возможна, когда в одном отношении хранится информация о нескольких сущностях

Отношение находится в 1НФ (1NF), если все его атрибуты построены на атомарных (простых) доменах, и, следовательно, значения в ячейках таблицы являются простыми (неделимыми), и ни одно из ключевых полей не пусто.

Первая нормальная форма (1НФ)

Отношение ***КНИГИ в 1НФ***:

ID	шифр рубрики	название рубрики	автор	название книги	редактор	тип издания	год издания	кол-во страниц
2	1.3	ВТ	А1	К1	Р1	учебник	2008	384
2	1.3	ВТ	А2	К1	Р1	учебник	2008	384
1	1.3	ВТ	А3	К2		учебник	2009	552
3	1.5	МО	А4	К3	Р2	учебник	2011	544
6	07	ИИ	А5	К4	Р3	учебное пособие	2003	176
6	07	ИИ	А5	К4	Р4	учебное пособие	2003	176
4	2.9	ВТ		К5	Р5	справочник	2010	208
5	1.8	ПО	А6	К6	Р6	учебное пособие	2009	304
5	1.8	ПО	А7	К6	Р7	учебное пособие	2009	304

Итак, 1НФ требует, чтобы на любом пересечении строки и столбца находилось **единственное значение**, которое должно быть **атомарным**, **и в таблице не должно быть повторяющихся строк**.

Прежде чем переходить к рассмотрению 2НФ, необходимо рассмотреть понятия **функциональных зависимостей**, которые есть в теории нормализации.

Определение. ***Неключевой атрибут***

- любой атрибут отношения, **не входящий** в состав никакого ключа (в частности, первичного).

Основные понятия теории нормализации

Определение 1. *Функциональная зависимость (FD)* (зависимость между атрибутами отношения)

Пусть задано отношение R, которое содержит наборы атрибутов A и B. В отношении R B ***функционально зависит*** от A и A ***функционально определяет*** B, тогда и только тогда, когда каждое значение проекции R[A] в любой момент времени связано точно с одним значением проекции R[B]. Эта ФЗ обозначается следующим образом: R.A → R.B. (или .A → B). Набор атрибутов A называют ***детерминантом*** для набора атрибутов B.

Наличие функциональной зависимости является свойством схемы, а не того или иного экземпляра отношения, и отражает семантику моделируемой предметной области.

или

Пусть ***R* (*A*1, *A*2,…, *A*n)** – схема отношения, а ***X*** и ***Y*** – произвольные **подмножества множества атрибутов** {*A*1, *A*2,…, *A*n}.

Тогда, в отношении **R** атрибут ***Y*** **функционально зависит от** атрибута ***X*** (или ***X*** **функционально определяет *Y***), *т.е.* $X \rightarrow Y$

тогда и только тогда, когда **каждое** значение множества ***X*** **связано** в точности **с одним значением** множества ***Y***.

Вывод. При наличии **X→Y** любые две записи, содержащие одинаковые значения **X**, должны включать и совпадающие значения **Y**. Это **ограничение** распространяется не только на уже имеющиеся записи, но и на те, которые могут быть добавлены в рассматриваемое отношение.

Определение 2. ***Ключи***

Набор атрибутов К называется ***суперключом*** отношения R, если все атрибуты R ***функционально зависят от К***.

Набор атрибутов К называется ***потенциальным (возможным)*** ключом отношения R, **если верно**, что:

1. ***Все атрибуты отношения R функционально зависят от К;***
2. ***Ни один атрибут из набора К не может быть удален без нарушения 1 свойства.***

Определение 3. ***Взаимно независимые атрибуты***

Атрибуты называются **взаимно независимыми**, если ни один из них **не является** функционально зависимым от другого.

Определение 4. ***Полная функциональная зависимость***

Функциональная зависимость **X → Y** называется **полной**, если атрибут **Y не зависит функционально** ни от какого **подмножества X**, иначе зависимость будет называться **частичной**.

Определение 5. ***Взаимно однозначная зависимость***

Определение 6.

Транзитивная функциональная зависимость

Функциональная зависимость **X → Y** называется **транзитивной**, если существует такой атрибут **Z**, что имеются функциональные зависимости

$$X \to Z \quad \text{и} \quad Z \to Y$$

и отсутствует функциональная зависимость $Z \to X$

Вторая нормальная форма (2НФ)

Отношение R (A_1, A_2 ..., A_n) находится **во 2НФ**, если оно находится в 1НФ, и *нет* ***неключевых атрибутов****, зависящих от части составного ключа (*каждый **неключевой** атрибут **зависит от всего** первичного ключа)

или

если оно находится в 1НФ и каждый **неключевой атрибут функционально полно** зависит от всего **составного ключа**.

Для перевода отношения во 2НФ необходимо, используя операцию **проекции**, разложить его на несколько отношений следующим образом:

1) построить проекцию отношения **без** атрибутов, находящихся в **частичной** ФЗ от первичного ключа;

Третья нормальная форма (3НФ)

Отношение находится **в 3НФ**, если оно находится во 2НФ, и каждый **неключевой** атрибут **нетранзитивно** зависит от первичного ключа.

или

Отношение находится **в 3НФ** в том и только том случае, если **все неключевые** атрибуты отношения **взаимно независимы** и полностью **зависят от всего первичного ключа**.

Нормальная форма Бойса-Кодда (BCNF)

Определение.

Детерминант ФЗ - минимальная **группа атрибутов**, от которой **зависит** некоторый другой атрибут или группа атрибутов, причем эта зависимость - **нетривиальная**.

Отношение находится в НФБК, если каждый его детерминант является потенциальным ключом

Нормальная форма Бойса-Кодда (BCNF)

Приведение к НФБК.

Если имеются отношения, содержащие **несколько потенциальных ключей**, то необходимо проверить, имеются ли **функциональные зависимости**, **детерминанты** которых **не** являются **потенциальными ключами**.

Если такие функциональные зависимости имеются, то необходимо провести дальнейшую **декомпозицию** отношений: **атрибуты, которые зависят от детерминантов, не являющихся потенциаль-ными ключами, выносятся в отдельное отношение вместе с детерминантами**.

Многозначные зависимости

Многозначная зависимость (**MDV**) подразумевает, что **два атрибута** (или два множества атрибутов) **независимы друг от друга**.

Многозначная зависимость возникает между атрибутами кортежей отношения в ситуации, когда отношение пытается представить **более одной связи типа —многие ко многим.**

Многозначные зависимости

Многозначность присутствует в тех отношениях, где моделируются связи типа 1:М.

Определение.

В отношении **R(X,Y,Z)** существует **многозначная зависимость X→→Y** в том и только в том случае, если множество значений **Y**, соответствующее паре значений **X и Z**, *зависит только от* **X** *и не зависит от* **Z**.

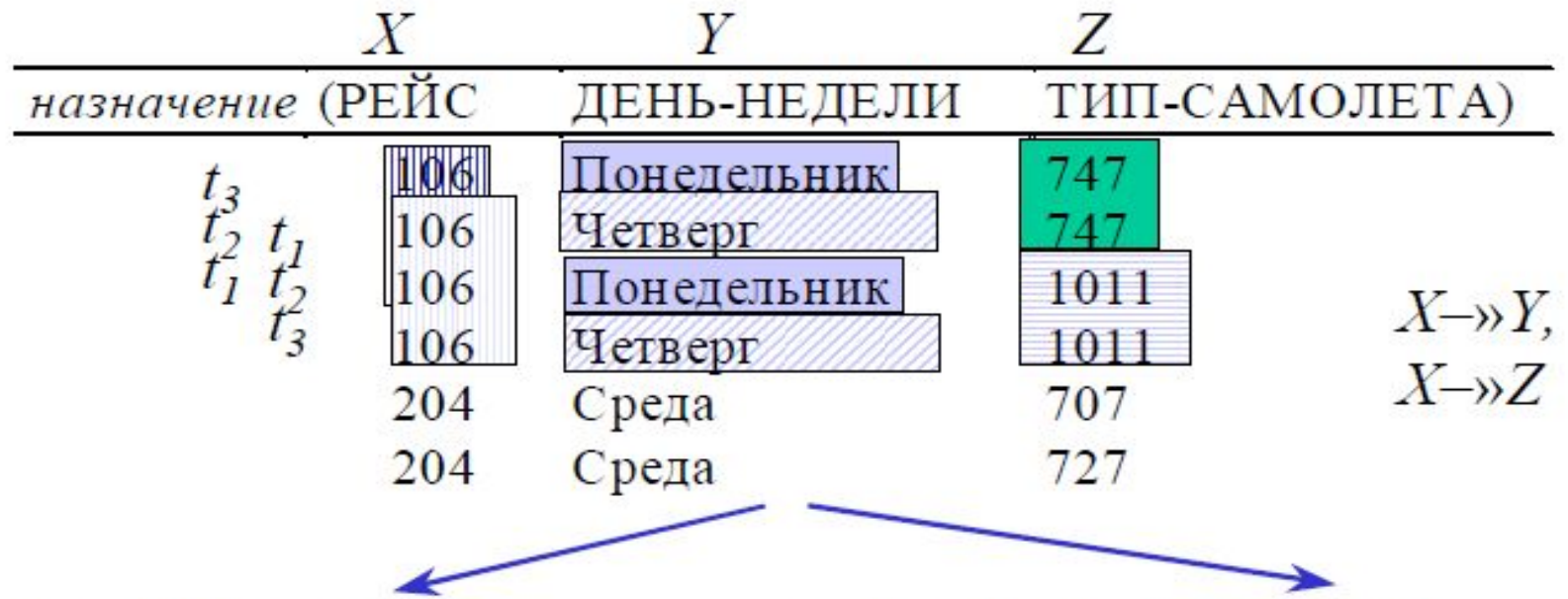
Многозначная зависимость X→→Y называется ***нетривиальной***, если ***не***

Многозначные зависимости

Следствие.

Наличие многозначной зависимости ***X→→Y*** *означает, что если 2 кортежа* ***совпадают*** *в части* ***X****, то можно* ***обменять*** *значения компонентов из* ***Y*** *между собой (при этом оставив нетронутыми оставшиеся атрибуты* ***Z****) и* ***получить*** *кортежи из того же отношения.*

Многозначные зависимости. Пример



	X	Y	Z	
назначение	(РЕЙС	ДЕНЬ-НЕДЕЛИ	ТИП-САМОЛЕТА)	
t_3	106	Понедельник	747	
t_2 t_1	106	Четверг	747	
t_1 t_2	106	Понедельник	1011	$X \rightarrow\!\!\rightarrow Y$,
t_3	106	Четверг	1011	$X \rightarrow\!\!\rightarrow Z$
	204	Среда	707	
	204	Среда	727	

день назначения (РЕЙС	ДЕНЬ-НЕДЕЛИ)
106	Понедельник
106	Четверг
204	Среда

тип самолета назначения (РЕЙС	ТИП-САМОЛЕТА)
106	747
106	1011
204	707
204	727

Если в отношении *назначение* существуют кортежи $<f\ d\ p>$ и $<f\ d'\ p'>$, то должен быть кортеж $<f\ d'\ p>$ и $<f\ d\ p'>$

Четвертая нормальная форма (4NF)

Многозначные зависимости. Пример

	X	*Y*	*Z*
назначение	(РЕЙС	ДЕНЬ-НЕДЕЛИ	ТИП-САМОЛЕТА)
t_3	106	Понедельник	747
t_2	106	Четверг	747
t_1	106	Понедельник	1011
	204	Среда	707
	204	Среда	727

~~$X \rightarrow\!\!\!\rightarrow Y$~~

$\neq$ *назначение*

день назначения	(РЕЙС	ДЕНЬ-НЕДЕЛИ)
	106	Понедельник
	106	Четверг
	204	Среда

тип самолета назначения	(РЕЙС	ТИП-САМОЛЕТА)
	106	747
	106	1011
	204	707
	204	727

Многозначные зависимости

Дальнейшая **нормализация** отношений основывается на **теореме Фейджина:**

Отношение R (X, Y, Z) можно спроецировать *без потерь* в отношения R1 (X, Y) и R2 (X, Z) в том и только в том случае, когда существует $X \rightarrow\rightarrow Y \mid Z$.

Под ***проецированием без потерь*** понимается такой способ декомпозиции отношения, при котором исходное отношение **полностью и без избыточности** восстанавливается путем **естественного соединения** полученных

Четвертая нормальная форма (4NF)

Определение. Отношение находится в 4НФ, если оно находится в 3НФ, и в нём <u>отсутствуют нетривиальные многo-значные зависимости</u>.

Для того чтобы привести отношение к 4НФ, нужно построить две или более проекции исходного отношения, каждая из которых содержит ключ и одну из

Соотношение между НФБК и многозначной зависимостью:

- **Всякая функциональная зависимость есть частный случай многозначной зависимости;**
- **Поэтому, если отношение в 4НФ, то оно и в НФБК;**
- **Но отношение может быть в НФБК, но не быть в 4НФ.**

Если отношение не в 4НФ, то его можно декомпозировать, пользуясь теми же приемами, что и для НФБК.

Полное множество функциональных зависимостей

Заданное множество ФЗ для отношения ***R*** обозначается ***F***.

Полное множество функциональных зависимостей, которые можно **логически получить из *F***, называется **замыканием *F*** и обозначается ***F+***.

Если множество функциональных зависимостей **F совпадает** с

АКСИОМЫ АРМСТРОНГА

1. **Аксиома рефлексивности**.

Если ***Y*** входит в ***X***, а ***X*** входит в ***U*** (***Y*** ⊆ ***X*** ⊆ ***U***), то ФЗ ***X*** → ***Y*** логически следует из *F*.

2. **Аксиома пополнения**.

Если ***X*** → ***Y*** и ***Z*** есть подмножество ***U***, то ***XZ*** → ***YZ***.

3. **Аксиома транзитивности**.

Если ***X*** → ***Y*** **и** ***Y*** → ***Z***, то ***X*** → ***Z***.

ПРАВИЛА ВЫВОДА ФЗ

- **Правило самоопределения.** $X \rightarrow X$
- **Правило объединения.**

  Если $X \rightarrow Y$ и $X \rightarrow Z$, то $X \rightarrow YZ$.
- **Правило псевдотранзитивности.**

  Если $X \rightarrow Y$ и $WUY \rightarrow Z$, то $XUW \rightarrow Z$.
- **Правило композиции.**

  Если $X \rightarrow Y$ и $Z \rightarrow W$, то $XW \rightarrow YW$.
- **Правило декомпозиции.**

  Если $X \rightarrow YZ$, то $X \rightarrow Y$ *и* $X \rightarrow Z$.

Пятая нормальная форма (5NF)

Зависимость соединения ***(X,Y,…,Z)** называется ***тривиальной***, если выполняется одно из условий:

- Либо ***все*** *множества атрибутов* **(X,Y,…,Z)** *содержат* ***потенциальный ключ*** отношения **R**.
- Либо **одно** из множеств атрибутов *совпадает со множеством всех атрибутов* отношения **R**.

Теорема Фейджина. Отношение **R(X,Y,Z)** $X \twoheadrightarrow Y \mid Z$ удовлетворяет **зависимости соединения**

Пятая нормальная форма (5NF)

Зависимость соединения является обобщением как **многозначной**, так и **функциональной** зависимости.

Отношение находится в 5НФ тогда и только тогда, когда любая имеющаяся зависимость соединения является тривиальной.

ВЫВОДЫ:

Декомпозиция – это разделение отношения на две или более таблицы с целью устранения аномалий:

1. ***Аномалия обновления*** – противоречивость данных, связанная с избыточностью и частичным обновлением.
2. ***Аномалии удаления*** – непреднамеренная потеря данных в связи с удалением других данных.
3. ***Аномалии ввода*** – это невозможность ввести данные в таблицу ввиду отсутствия других данных.

Выводы: **Нормализация устраняет** следующие **типы функциональных зависимостей:**

- **2НФ** — **частичные** зависимости **неключевых** атрибутов **от ключевых**;
- **3НФ** — **транзитивные** зависимости **неключевых** атрибутов **от ключевых**;
- **Усиленная 3НФ (НФБК)** — зависимости **ключей от неключевых** атрибутов;
- **4НФ** — **многозначные** зависимости в

Основные правила процедуры нормализации, применяемые к данным в информационной системе

1. **Отношение в 1НФ** следует разбить на проекции для исключения **частичных функциональных зависимостей**. В результате должен быть получен набор отношений **во 2НФ**.

2. **Отношения во 2НФ** следует разбить на проекции для исключения **транзитивных зависимостей между неключевыми атрибутами**. В результате должен быть получен набор отношений **в 3НФ**.

3. **Отношения в 3НФ** следует разбить на проекции для исключения любых оставшихся **функциональных зависимостей**, в которых

Основные правила процедуры нормализации, применяемые к данным в информационной системе

4. **Отношения в НФБК** следует разбить на проекции для исключения **любых многозначных зависимостей**, которые **не являются функциональными**. В результате должен быть получен набор отношений **в 4НФ.**

(На практике многозначные зависимости, воспринимающиеся как **повторяющиеся группы**, исключаются из исходного отношения до выполнения этапов 1 - 3).

5. **Отношения в 4НФ** следует разбить на проекции для исключения любых **зависимостей соединения**, если их удается распознать в отношении. В результате должен быть получен

Проектирование баз данных на основе нормальных форм

Пример.

Имеется схема отношения:

СТУДЕНТ (№_зачетки, Фамилия, Группа, Факультет, Семестр, Предмет, Преподаватель, Вид_Работы, Оценка)

Нормализовать отношение.

№ЗК	Фамилия	Группа	Факультет	Семестр	Предмет	Препод	Вид_работы	Оценка
01	Панов	Г1	Ф1	1	Химия	Сомов	Экз	Отл
01	Панов	Г1	Ф1	1	Физика	Петров	Экз	Отл
01	Панов	Г1	Ф1	1	История	Львов	Экз	Отл
02	Туров	Г2	Ф1	1	Химия	Сомов	Экз	Хор

На данный промежуток времени справедливы следующие функциональные зависимости:

F_1 = **№_зачетки** → Фамилия, Группа, Факультет

F_2 = **№_зачетки, Семестр, Предмет** → Преподаватель, Вид_Работы, Оценка

F_3 = **№_зачетки, Семестр, Предмет** → Фамилия, Группа, Факультет

F_4 = **№_зачетки, Семестр, Предмет** → Оценка

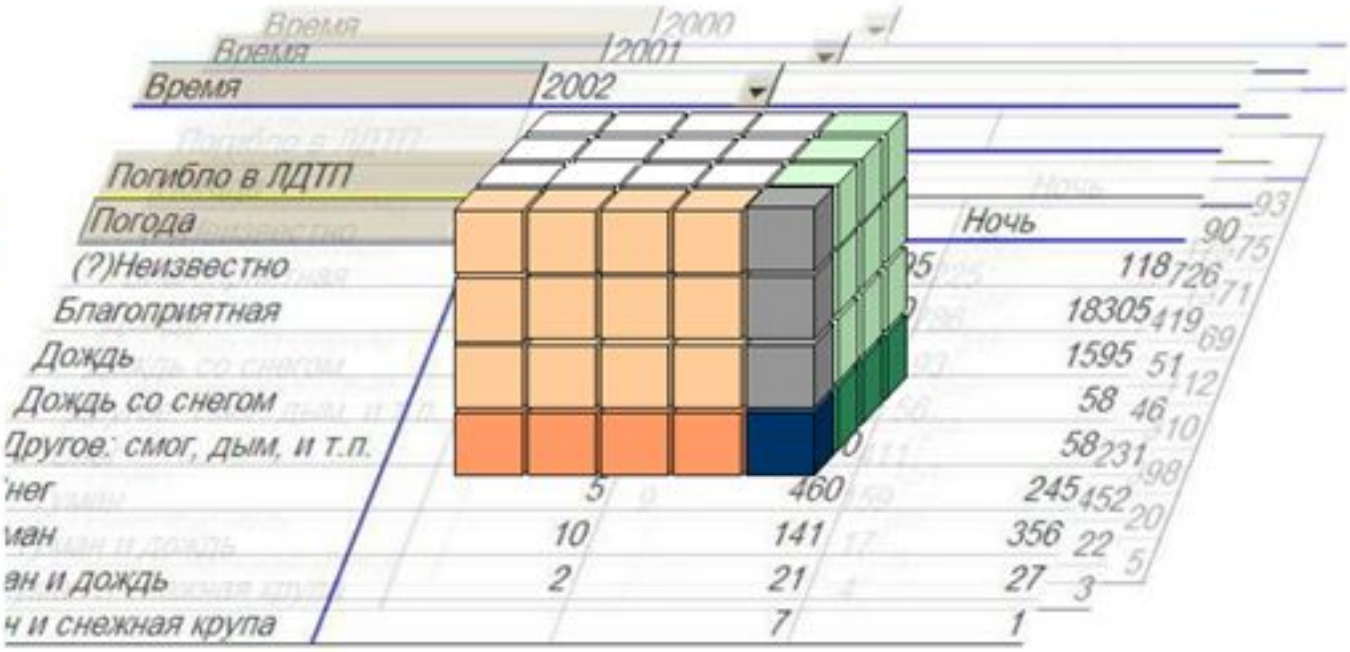
F_5 = **Предмет** → Преподаватель

F_6 = **Семестр, Предмет** → Вид_Работы

F_7 = **Группа** → Факультет

Многомерные базы данных

OLAP и Информационные Хранилища



Основные концепции

OLAP: Основные понятия

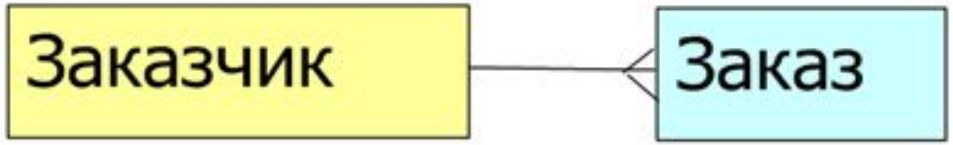
Термин **OLAP** – (**On-Line Analytical Processing -** обработка данных в реальном времени) был введен в употребление в 1993 году Эдгаром Коддом. В своей работе “Providing OLAP to User-Analysts: An IT Mandate” он сформулировал **12 правил OLAP**, которым должна удовлетворять **OLAP система**.

Для решения большинства задач анализа оказываются полезными принципы многомерной модели данных и соответствующие им многомерные базы данных.

OLAP: Основные понятия

Реляционная модель

- **Модель данных через двухмерные таблицы**



ID	Фамилия	...
001	Таранов	...
002	Фомин	...
...	...	...

Номер	Дата	Id_customer	...
01	16 ноября 2006	002	...
02	17 ноября 2006	002	...
...	...	...	...

OLAP: Основные понятия

12 правил Кодда для РСУБД

- Система должна быть и *реляционной, и базой данных, и системой управления.*
- Явное представление данных.
- Гарантированный доступ к данным.
- Полная обработка неизвестных значений.
- Доступ к словарю данных в терминах реляционной модели.
- Полнота подмножества языка.
- Возможность модификации представлений.
- Наличие высокоуровневых операций управления данными.
- Физическая независимость данных.
- Логическая независимость данных.
- Независимость контроля целостности.
- Дистрибутивная независимость.
- Согласование языковых уровней.

Проблемы **анализа** накопленной информации:

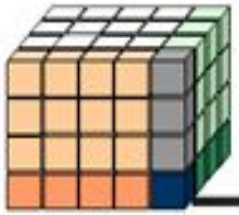
- В большой системе могут быть сотни связанных между собой сложными цепочками отношений таблиц. И зачастую нужно долго **писать и отлаживать запрос SQL**, объединяющий десятки таблиц. Запросов у пользователей может быть много и разных, под каждый такой SQL не напишешь.
- Часто результатом запроса является не детальная, а **агрегированная выборка**. Следствие: запросы, которые суммируют миллионы записей, сильно нагружают сервер и могут мешать накоплению транзакционных данных.
- Вечная мечта разработчиков аналитических систем – сделать так, чтобы бизнес-пользователи, сами, с минимальным привлечением IT специалистов могли получать интересующую их информацию. Очевидно, что обычный пользователь не напишет SQL на 10 листов и не сможет сам оптимизировать запросы.

Многомерные базы. Хранилища данных.

- OLTP (on-line transaction processing)
  - Основное назначение реляционных СУБД
  - Ежедневные операции: покупки, заказы, производство, регистрация и т.п..
- OLAP (on-line analytical processing)
  - Основное назначение хранилищ данных;
  - Анализ данных и поддержка принятия рациональных решения.

Сравнительные характеристики OLTP и OLAP - технологий

	OLTP	OLAP
Пользователи	Клерки и IT-шники сопровождения	Эксперт-аналитик (предметник)
Режим работы	Ежедневные операции	При поиске оптимального решения
Архитектура	Ориентировано на приложение	Предметно-ориентированная
Данные	Текущие, актуальные, детализированные, реляционные, нормализованные (безизбыточные).	Исторические, агрегированные, многомерные, консолидированные, денормализованные.
Использование	Однородное, повторяющееся	Априори неизвестное (ad-hoc)
доступ	Чтение/запись, доступ по к отдельным записям по индексам.	Массовые операции над большими обьемами.
Элемент доступа	Простые короткие транзакции	Сложные запросы
# строк доступа	десятки	миллионы
# пользователей	тысячи	сотни
Размер базы	<GB	100GB-TB
Мера производительности	Транзакций в секунду	Скорость выполнения аналитических запросов



OLAP-тезисы Кодда (1993)

(Теперь входят в критерии FASMI)

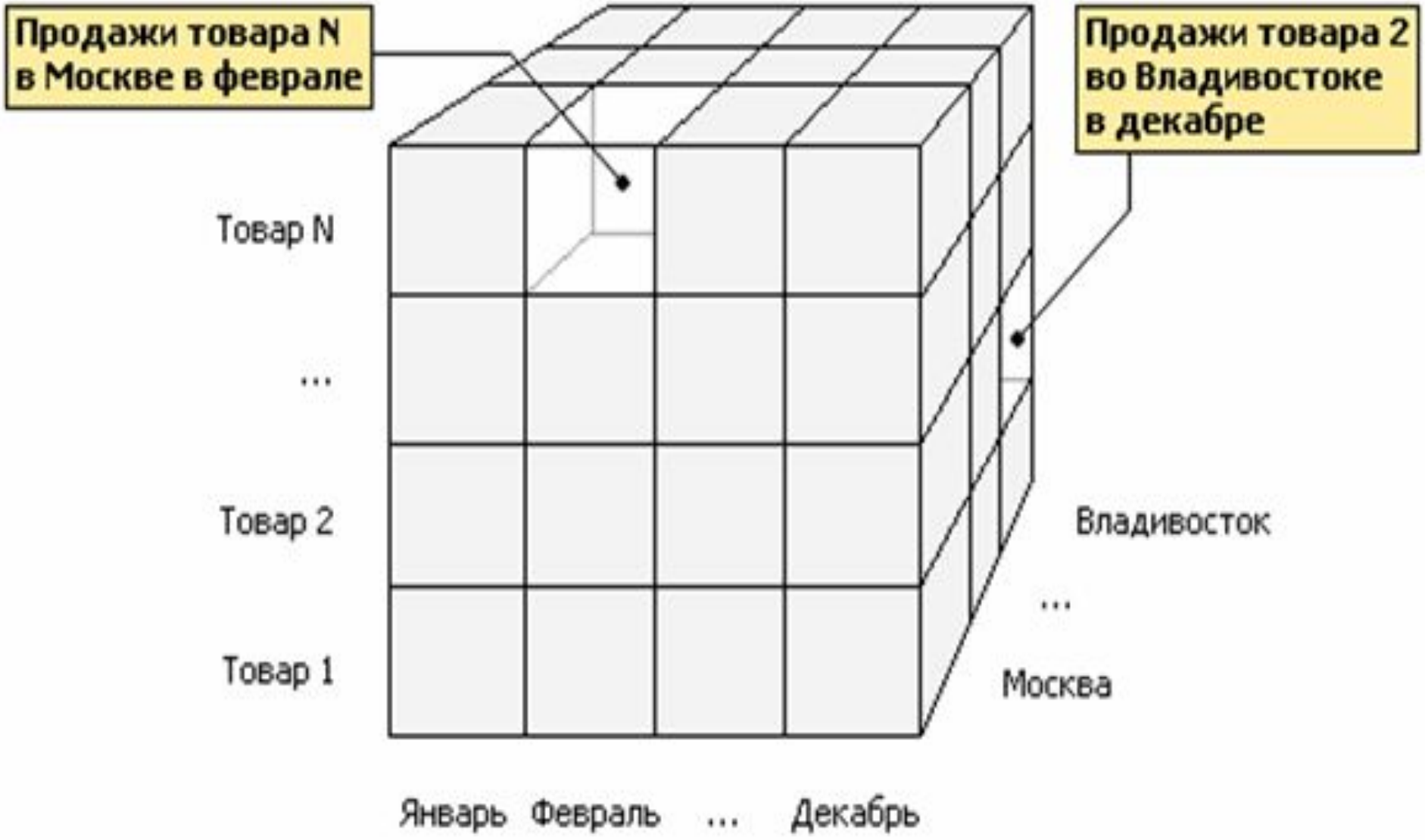
1. Многомерность (*Multi-Dimensional Conceptual View*)
2. Прозрачность сервера (*Transparency*);
3. Доступность (*Accessibility*);
4. стабильные доступ и работа (*Consistent Reporting Performance*);
5. архитектура "клиент-сервер" (*Client-Server Architecture*);
6. видовая размерность;
7. управление разреженностью данных (*Dynamic Sparse Matrix Handling*);
8. многопользовательский режим (*Multi-User Support*);
9. операции с измерениями (*Unrestricted Cross-dimensional Operations*);
10. интуитивное манипулирование данными (*Intuitive Data Manipulation*);
11. гибкая запись и редактирование (*Flexible Reporting*);
12. Неограниченная размерность и число уровней агрегации (*Unlimited Dimensions and Aggregation Levels*)

Многомерная модель данных

Основной идеей является представление информации в виде **многомерных кубов**, где **оси** представляют собой **измерения** (н-р, время, товары, клиенты), а в ячейках помещаются **показатели** (н-р, сумма продаж, средняя цена закупок и пр.).

Пользователь манипулирует измерениями и получает информацию в нужном разрезе.

Многомерный куб данных



Многомерная модель данных. OLAP – куб. Куб – это хранилище фактов.

Из **OLAP-куба** может быть составлен обычный **плоский отчёт**.

По столбцам и строкам отчёта будут **бизнес-категории** (грани куба), а в ячейках - **показатели**.

		Январь	Февраль	Март	
Москва	Яблоки	20 000	21 000	22 000	63 000
	Груши	25 000	27 000	29 000	81 000
	Киви	30 000	33 000	36 000	99 000
	Итого	75 000	81 000	87 000	243 000
Пермь	Яблоки	18 000	19 000	20 000	57 000
	Груши	20 000	22 000	24 000	66 000
	Киви	14 000	17 000	20 000	51 000
	Итого	52 000	58 000	64 000	174 000
Уфа	Яблоки	12 000	13 000	14 000	39 000
	Груши	18 000	20 000	22 000	60 000
	Киви	22 000	25 000	28 000	75 000
	Итого	52 000	58 000	64 000	174 000
Итого		179 000	197 000	215 000	591 000

OLAP-кубы содержат **бизнес-показатели**, используемые для анализа и принятия управленческих решений, например: прибыль, рентабельность продукции, совокупные средства (активы), собственные средства и т.д.

Бизнес-показатели хранятся в кубах **не в виде простых таблиц**, как в обычных системах учета или бухгалтерских программах, **а в разрезах**, представляющих собой основные бизнес-категории деятельности организации: товары, магазины, клиенты, время продаж и т. д.

Благодаря детальному **структурированию** информации **OLAP-кубы** позволяют **оперативно осуществлять анализ данных** и формировать отчёты в различных разрезах и с произвольной глубиной детализации.

Представление многомерных данных (витрины)

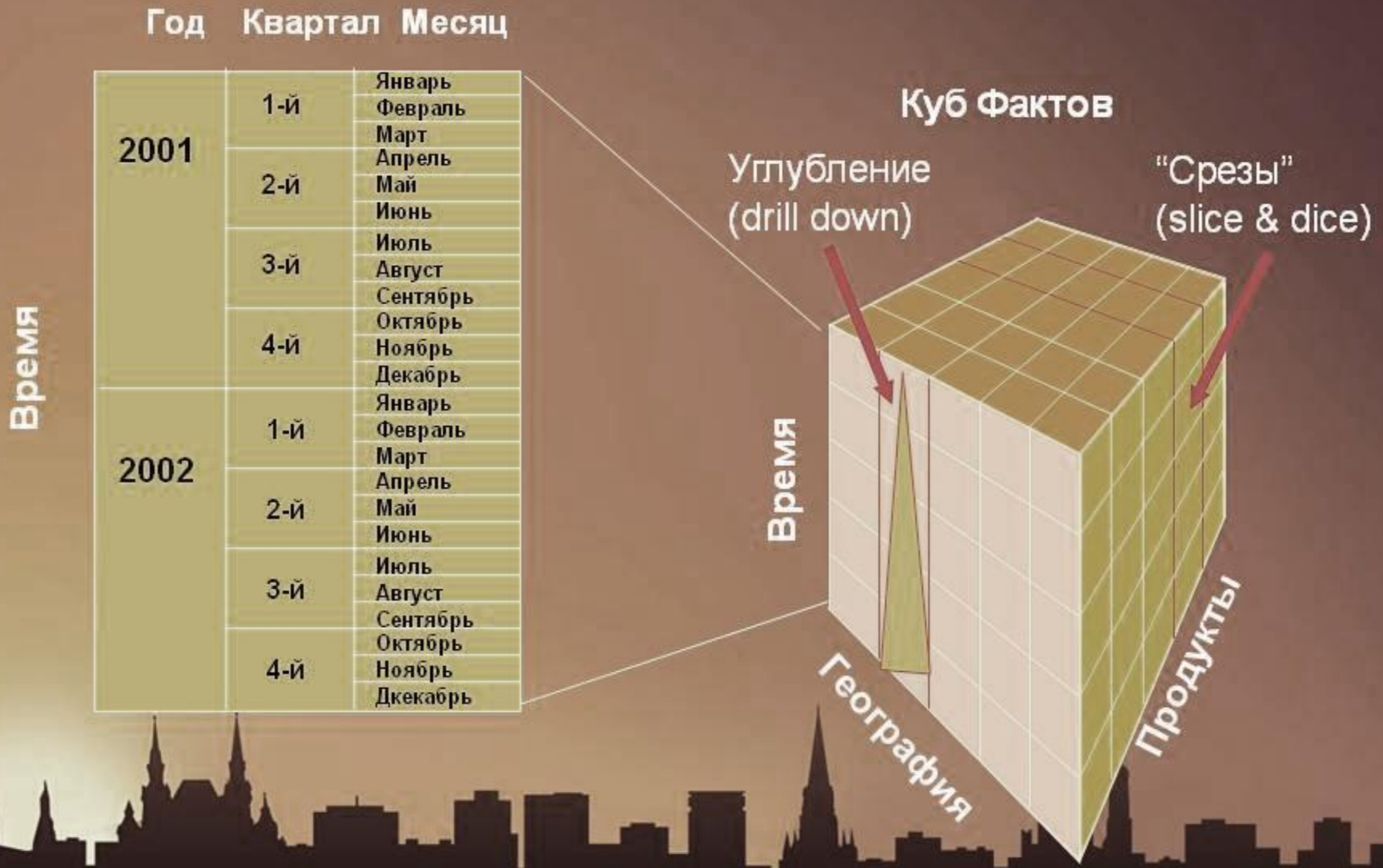
Кубы:

- Меры (Продажи, Затраты)
- Измерения (Продукт, Регион, Время)

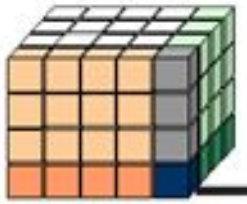
Иерархии измерений:

Продукт	Регион	Время
Группа	Страна	Год
Категория	Регион	Квартал
Товар	Город	Месяц Неделя
	Филиал	День

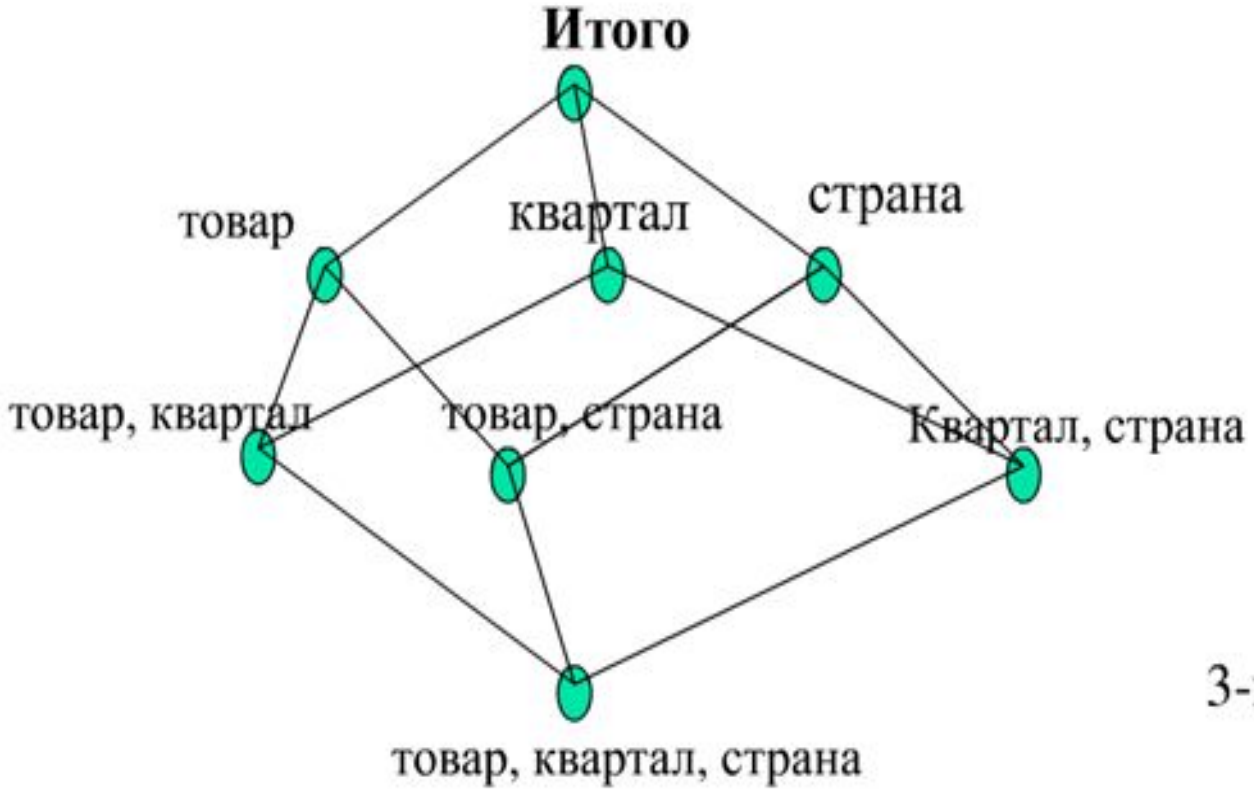
Базовые операции с кубом



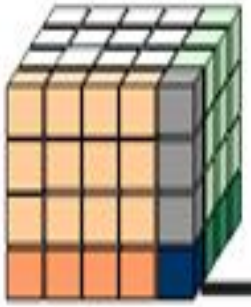
OLAP: Основные понятия



Кубоиды в кубе

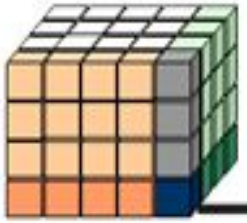


Кубоиды выделяют данные на разных уровнях агрегации.



Основые OLAP-операции

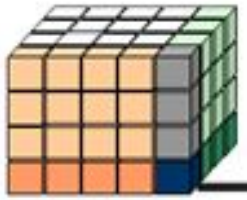
- Roll up: агрерация данных: по иерархии(-ям) до полного исключения измерения.
- Drill down: детализация: от обощенных данных к более детальным, от верхних уровней измерений – к нижним, детализация данных по дополнительным измерениям.
- Slice and dice: проекции и выборки – выборка нужных "ломтей" кубика
- Pivot (rotate): вращение куба, визуализация, выборка и ориентация одно-, двух-, трехмерных срезов для визуального анализа



Реляционные модели хранилища

- Схема "Звезда" ("*Star*") : Таблица фактов "в середине" соединяется с набором "сателлитов"-таблиц измерений. Все уровни агрегации для каждой координаты являются атрибутами соответствующей записи из таблицы измерений.
- Схема "Снежинка" ("*Snowflake*"): Базовый кубоид также представляется в схеме "Звезда", но уровни агрегации реляционно нормализованы, и каждый уровень хранится в своей собственной таблице.

Пример “Звезды”



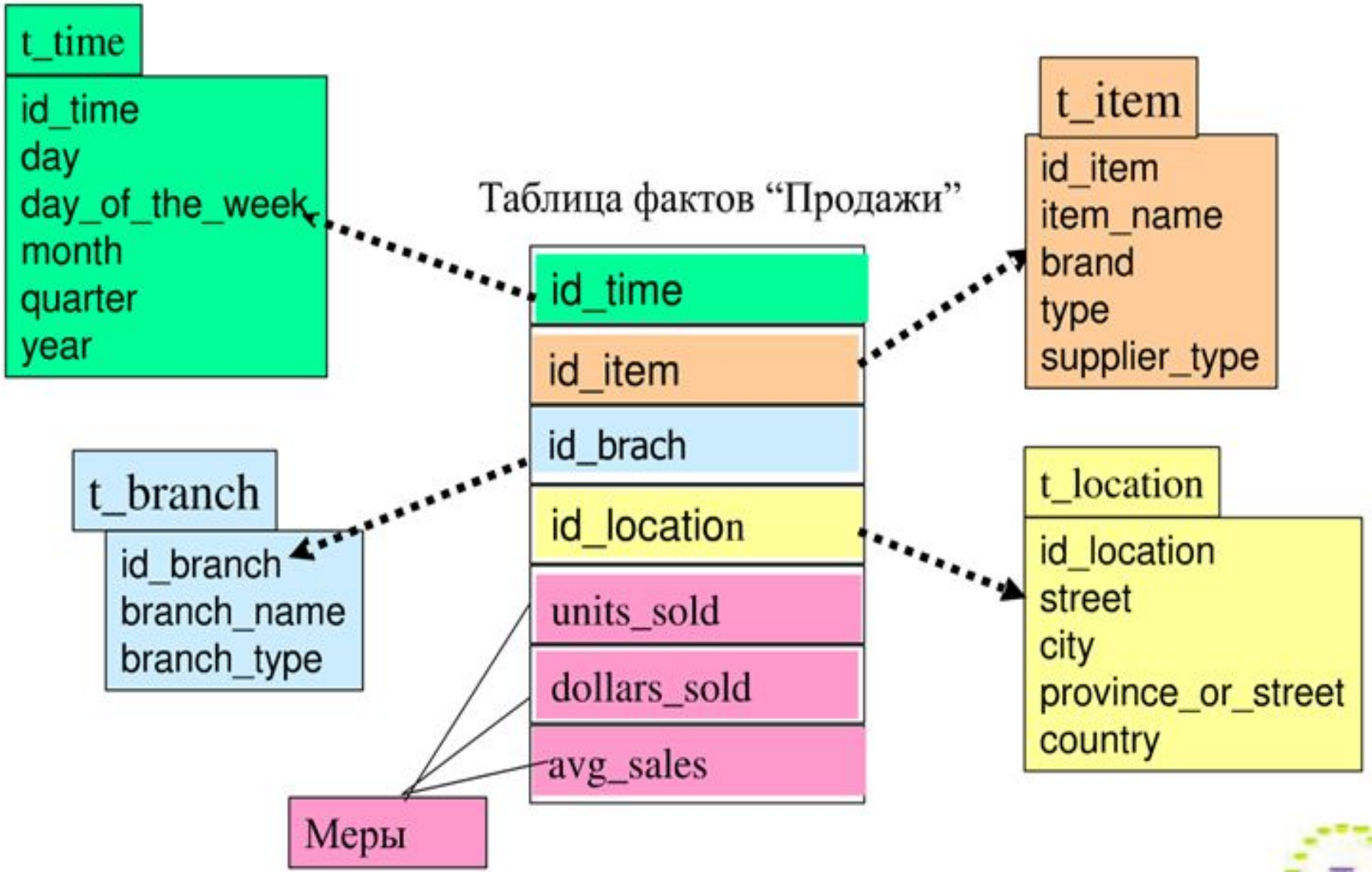
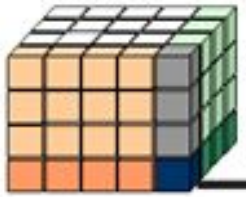


Схема "Снежинка"



Пример. Трехмерное представление данных о продажах

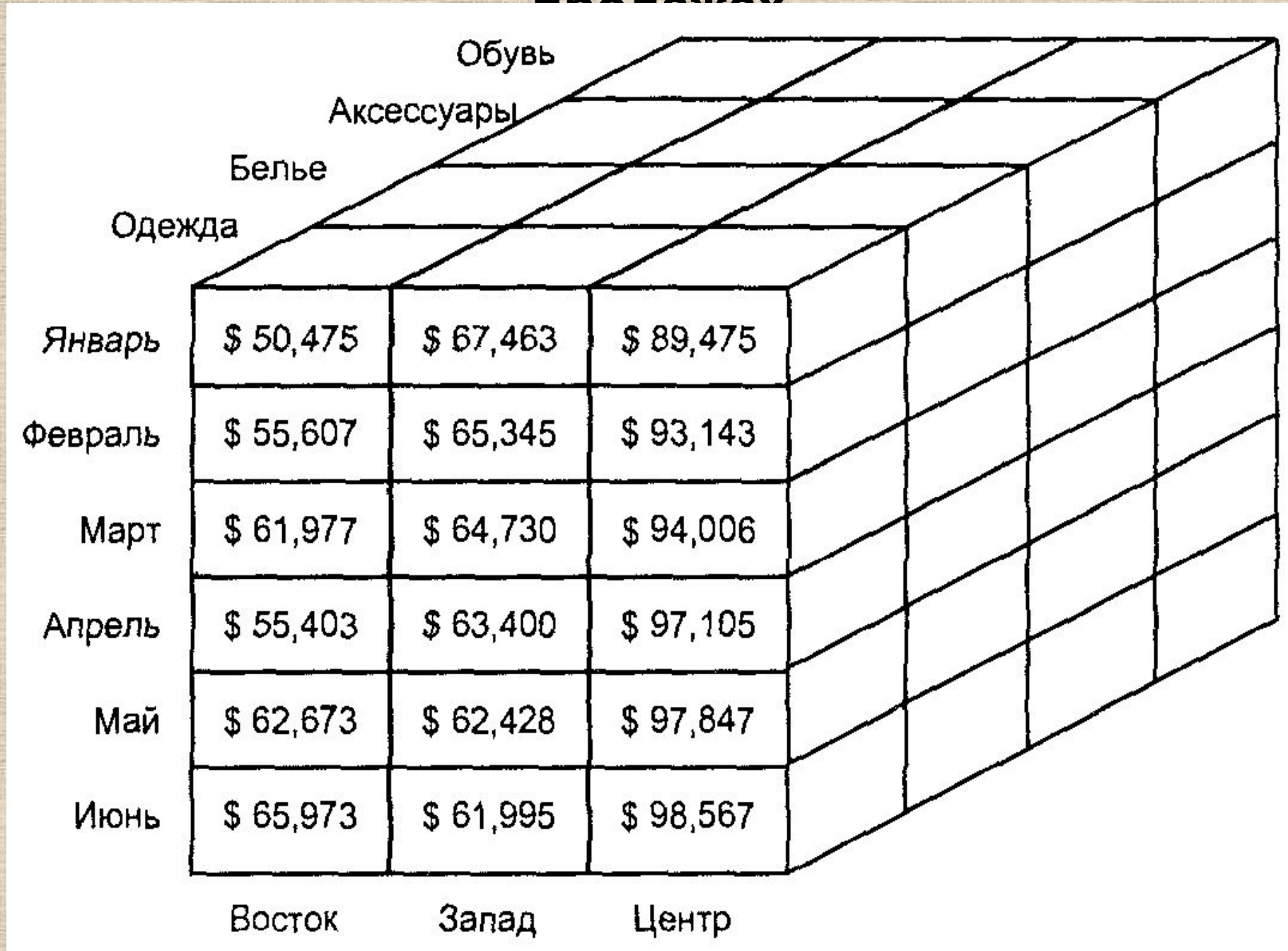
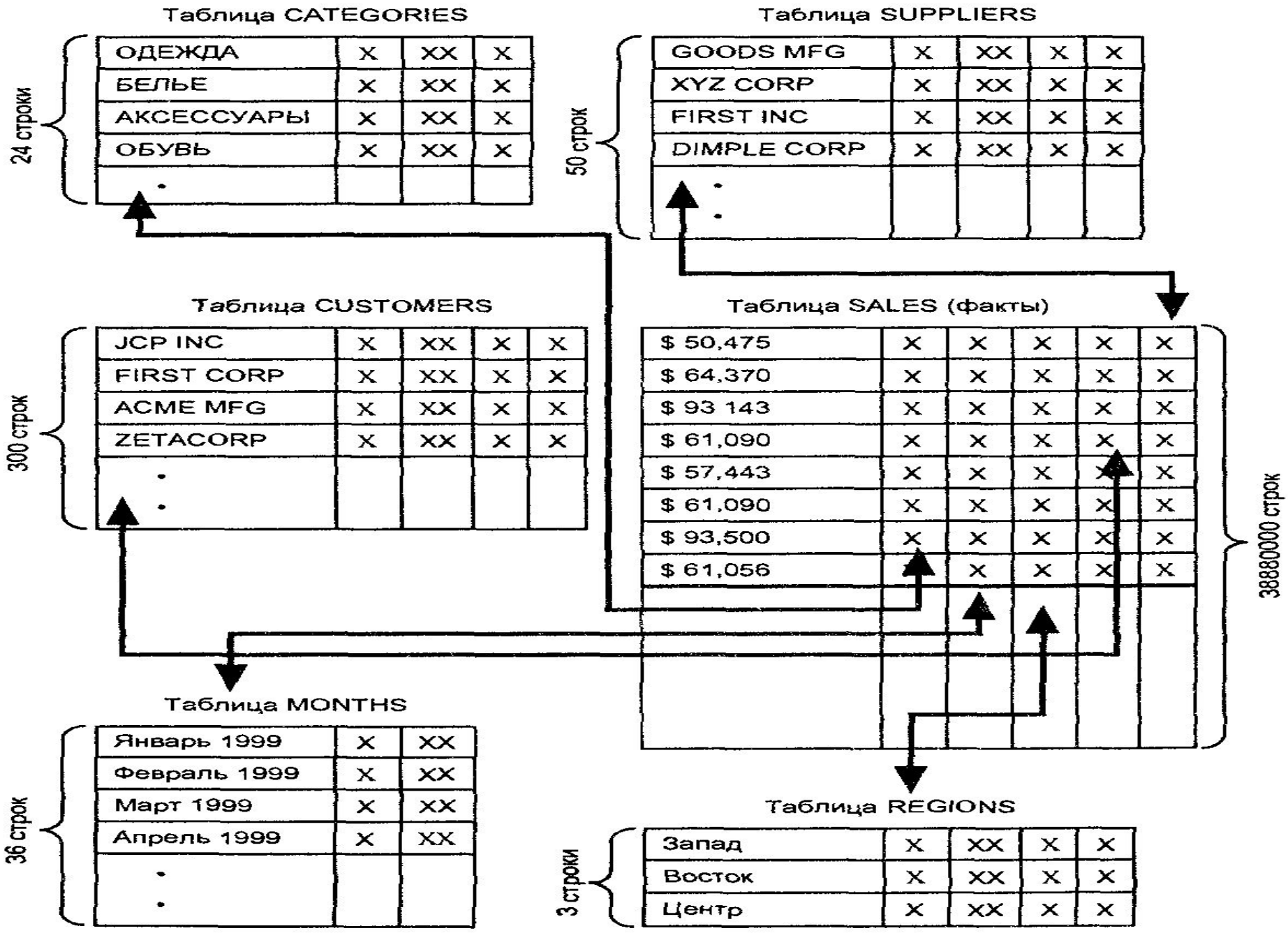
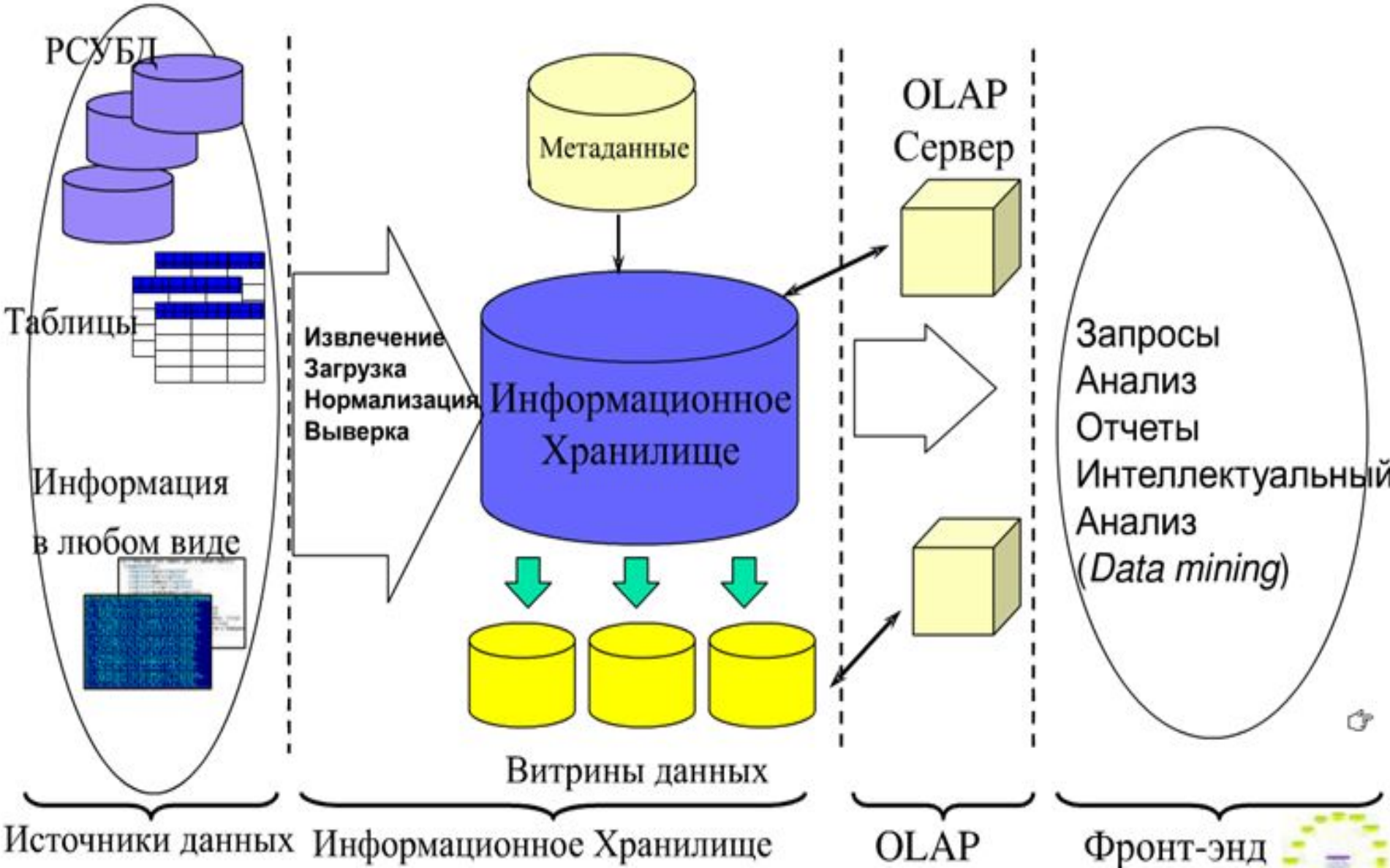
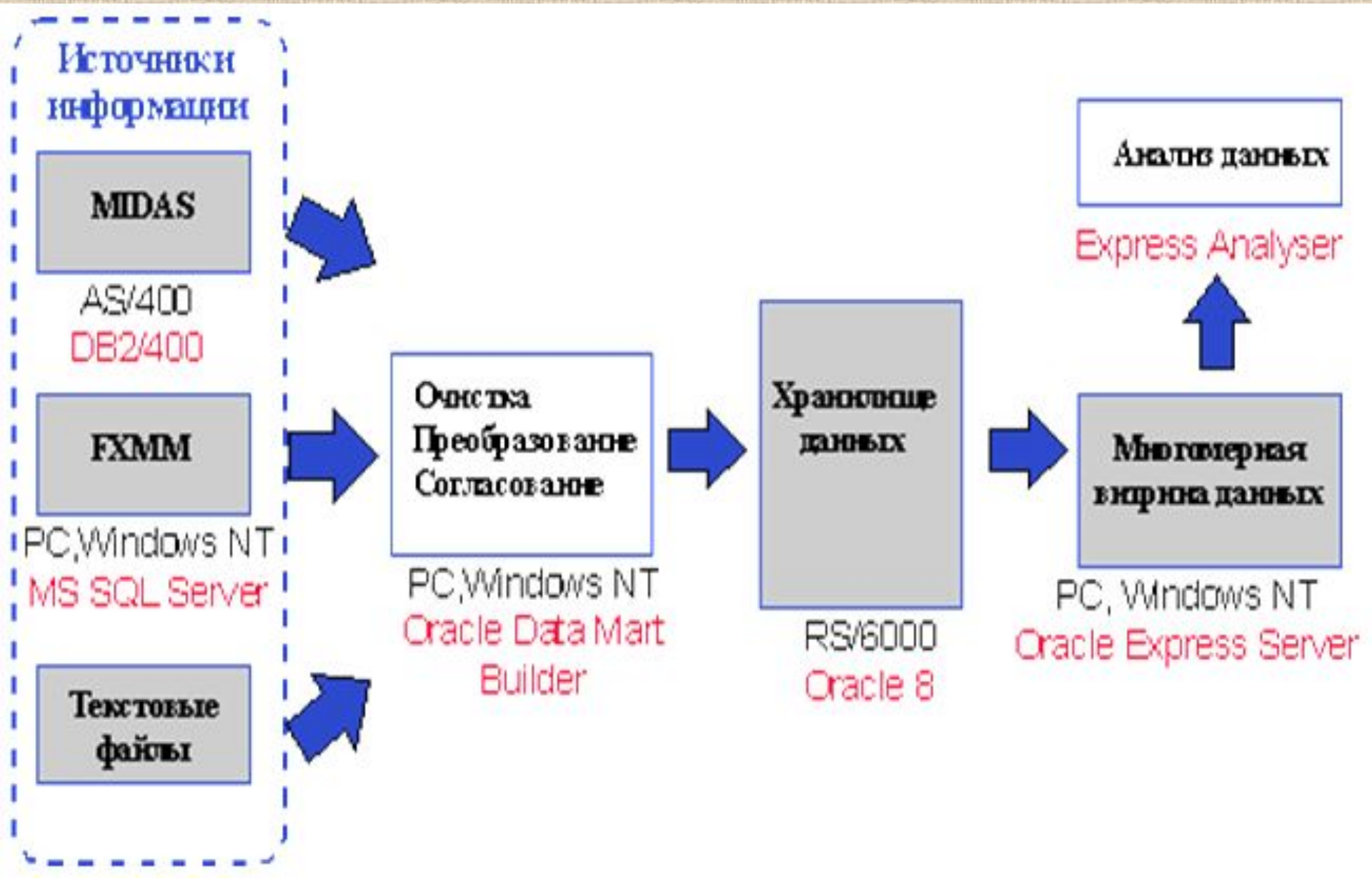


Схема «звезда» для хранилища данных



Архитектура многоуровнего Хранилища



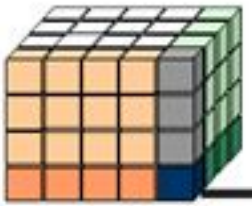

Источники информации
MIDAS
AS/400
DB2/400
FXMM
PC,Windows NT
MS SQL Server
Текстовые файлы
Очистка
Преобразование
Согласование
PC,Windows NT
Oracle Data Mart Builder
Хранилище данных
RS/6000
Oracle 8
Многомерная витрина данных
PC, Windows NT
Oracle Express Server
Анализ данных
Express Analyser

Витрины данных - это небольшие хранилища с упрощенной архитектурой, предназначенные для хранения небольшого подмножества данных и снятия нагрузки с основного информационного хранилища предприятия.

Многомерные базы данных реализуют в двух основных формах:

1. **Системы многомерной оперативной аналитической обработки (MOLAP)** хранят данные в специализированных многомерных структурах.

2. **Реляционные системы OLAP (ROLAP)** для хранения данных используют реляционные базы данных, а также применяют специализированные индексные структуры, такие как битовые карты, чтобы добиться высокой скорости выполнения запросов.



OLAP - Архитектуры

- Реляционный OLAP (ROLAP)
  - Используется РСУБД для хранения ИХ.
  - Оптимизируется агрегационные возможности РСУБД
  - (+) Масштабируемость
- Многомерный OLAP (MOLAP)
  - Механизм хранения многомерных массивов (как плотных так и разряженных)
  - (+) Очень быстрый доступ к любым срезам, с произвольной агрегацией
- Гибридный OLAP (HOLAP)
  - HOLAP = ROLAP + MOLAP (масштабируемость+скорость)
  - Нижние уровни (факты) – в реляционной БД, верхние, агрегированные уровни – в кубах.