

Тема 2

Интерактивная аналитическая обработка данных (OLAP)

4. OLAP-системы

Интерактивная аналитическая обработка данных

Основной вопрос при обработке информации заключается в том, как обрабатывать всё более и более крупные БД, которые содержат информацию с постоянной усложняющейся структурой, и как создать систему, отвечающую на запросы за приемлемое время.

OLAP занимается динамическим синтезом, анализом и обобщением больших объемов многомерных данных. Эта технология должна поддерживать сложные аналитические системы.

2.1 Правила для OLAP систем

12 правил Б.Кодда:

1. Многомерное концептуальное представление данных
2. Прозрачность
3. Доступность
4. Неизменно высокая производительность подготовки отчетов
5. Архитектура клиент-сервер
6. Универсальность измерений
7. Динамическое управление разреживанием матриц
8. Многопользовательская поддержка
9. Неограниченные перекрестные операции между размерностями
10. Поддержка удобных средств манипулирования данными
11. Гибкость средств формирования отчетов
12. Неограниченное число измерений и уровней агрегирования.

Рассмотрим более детально каждое из правил.

1. Многомерное концептуальное представление данных

Инструменты OLAP должны предоставлять пользователям многомерную модель, отвечающую представлениям пользователей о

деятельности организации. Модель должна быть понятной и простой в использовании.

2. Прозрачность

Технология OLAP, используемые для нее БД и архитектура, а также неоднородные источники входных данных должны быть прозрачны для пользователей.

3. Доступность

Инструмент OLAP должен обеспечивать доступ к требуемым для анализа данным, сохраняемым в различных неоднородных корпоративных источниках данных, включая, реляционные, нереляционные и прочие существующие системы.

4. Неизменно высокая производительность подготовки отчетов

Пользователи не должны ощущать заметного снижения производительности по мере возрастания количества измерений, уровней агрегирования данных и самого размера БД. При этом не должны изменяться способы вычисления важных значений.

5. Архитектура клиент-сервер

Система OLAP должна быть способна эффективно функционировать в среде клиент-сервер. Использование этой архитектуры должно обеспечивать оптимальную производительность, гибкость, адаптивность, масштабируемость и способность к взаимодействию.

6. Универсальность измерений

Все измерения данных должны быть эквивалентны по структуре и функциональным возможностям. То есть основная структура, формулы и средства создания отчетов не должны быть привязаны к конкретной размерности

7. Динамическое управление разреживанием матриц

Система OLAP должна быть способна адаптировать свою физическую схему к конкретной аналитической модели, что подразумевает

динамическую оптимизацию разреженности матриц, выполняемую с целью достижения и поддержки требуемого уровня производительности системы.

8. Многопользовательская поддержка

Система OLAP должна быть в состоянии поддерживать параллельную работу группы пользователей с одной или несколькими моделями корпоративных данных.

9. Неограниченные перекрестные операции между размерностями

Система OLAP должна уметь распознавать иерархии размерностей и автоматически выполнять перекрестные агрегирующие вычисления внутри одной размерности и между несколькими размерностями.

10. Поддержка удобных средств манипулирования данными

Разбиение и поворот (создание сводных таблиц), нисходящий анализ, консолидация (суммирование), а также любые другие манипуляции с данными должны выполняться с помощью простейших действий по принципу «указать и щелкнуть» или «перетащить и опустить», выполняемых по отношению к ячейкам куба

11. Гибкость средств формирования отчетов

Необходимо иметь инструменты упорядочивания строк, столбцов и ячеек, которые позволяют упростить анализ данных за счет понятного визуального представления аналитических отчетов. Пользователи должны иметь возможность получить любое желаемое представление необходимых им данных.

12. Неограниченное число измерений и уровней агрегирования

В зависимости от конкретных деловых требований аналитическая модель может иметь самое разное количество размерностей, обладающих собственной иерархической структурой. Система OLAP не должна накладывать никаких ограничений на количество измерений или уровней агрегирования данных.

2.2 Многомерная OLAP-технология

Рассмотрим, как лучше всего представить запрос типа следующего:

Запрос к БД: «Каким был общий доход от продаж объектов недвижимости в каждом городе и в каждом квартале 2016 г?»

Можно предложить 4 варианта представления результата запроса. В OLAP-технологиях используется МНОГОМЕРНОЕ представление данных (вариант [г])

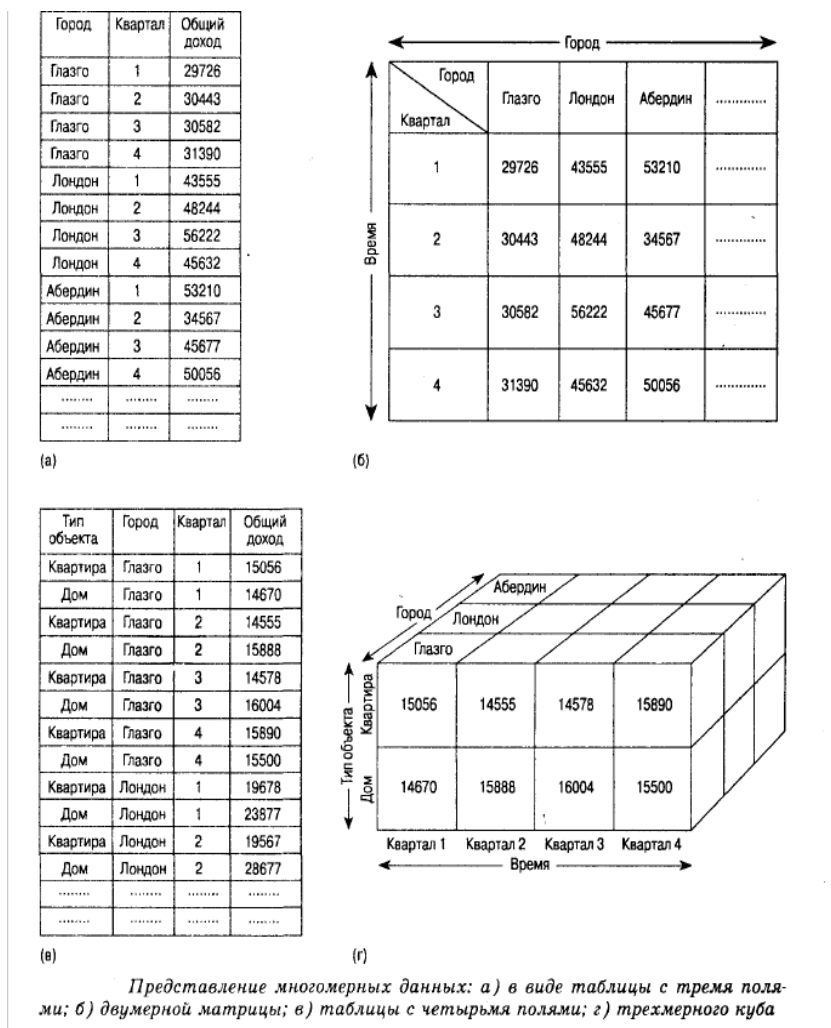


Рис.2.1

Данные можно представить в таблице с тремя столбцами, представленной в таблице следующей далее.

Город	Квартал	Общий доход
-------	---------	-------------

Глазго	1	29726
Глазго	2	30443
Глазго	3	31582
Глазго	4	31390
Лондон	1	43555
Лондон	2	48244
Лондон	3	56222
Лондон	4	45632

Можно представить в виде двухмерной таблицы:

город Квартал	Глазго	Лондон	...
1	29726	43555	
2	30443	48244	
3	31582	56222	
4	31390	45632	

Если мы расширим запрос добавив информацию по типу недвижимости, то данные могут быть размещены в таблице с четырьмя полями.

Тип объекта	Город	Квартал	Общий доход
Квартира	Глазго	1	15056

Дом	Глазго	1	14670
.	.	.	.
Квартира	Лондон	1	19678
Дом	Лондон	1	23877

Но более естественно было бы разместить результаты запроса в трехмерном кубе.

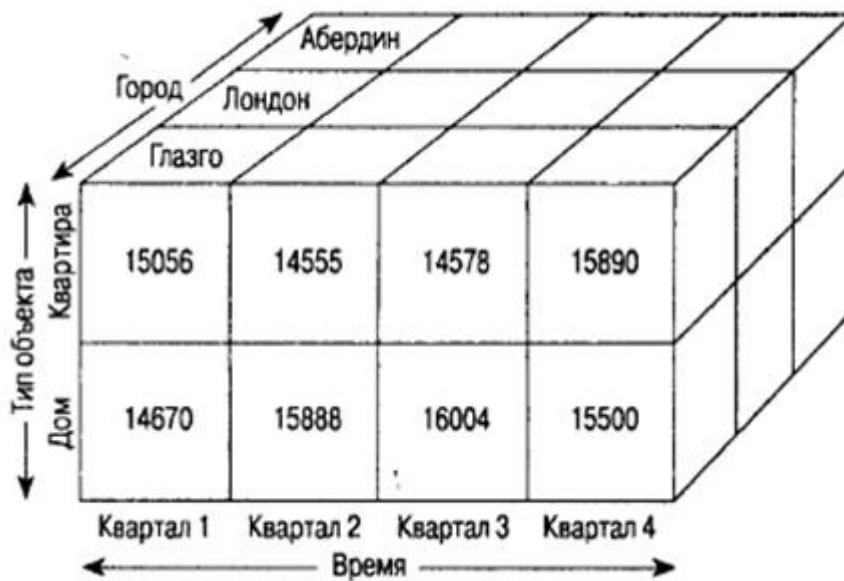


Рисунок 2.2 – Результат запроса в графическом представлении

5. Многомерная модель данных

За последние несколько лет анализ большого объема данных для принятия стратегических решений и совершенствования процессов повседневного принятия решений вышел на первый план. Возникла потребность в системах, призванных обеспечить надёжный и быстрый доступ к большому объёму накопленной информации для её анализа.

OLAP системы имеют высокую скорость поиска информации в отличие от реляционных СУБД. Строится многомерная база данных на основе многомерной модели данных.

Многомерная модель данных определяет представление данных на трёх уровнях:

- Концептуальной модели;
- Физической модели;
- Прикладной модели данных.

Концептуальная модель данных описывает представление данных в системе и методы их описания. В терминах этой модели описываются данные предприятия, их структура, методы расчёта.

Физическая модель данных определяет, как характеризуются данные на физических носителях, в каких типах файла, как хранятся в сети, как осуществляется доступ к ним, как данные кэшируются.

Прикладная модель данных определяет формат данных, в котором они передаются аналитическому приложению.

2.4. Многомерное пространство

При работе с реляционными базами данных используется двумерное пространство-таблица с записями и столбцами. При работе с многомерными базами данных используется термин куб для описания многомерного пространства, хотя это не в геометрическом смысле слова. Геометрический куб имеет только три измерения. Многомерное пространство данных может иметь любое количество измерений, и эти измерения не обязаны быть одинакового или похожего размера.

Одним из самых важных отличий между геометрическим пространством и многомерным пространством данных является то, что геометрическое пространство состоит из бесконечного числа точек, а многомерное пространство дискретно и содержит дискретное количество значений на каждом измерении.

Терминология и обозначения в гиперкубе:

1. **Измерения (dimension -> dim)** – оси гиперкуба;

Например: **Время, Страны, Фирмы-перевозчики**;

2. **Элемент** – точка на измерении.

Обозначение - [];

Например: [**2016**];

3. **Кортеж** – координаты измерений, определяемые значениями элементов измерения.

Обозначение - ();

Например: ([**2016**],[**Наименование фирмы-перевозчика-Ф1**]);

4. **Массив**- набор кортежей составляют массив.

Обозначение - { };

Например:

{([**2014**],[**FedShip**]),([**2015**],[**FedShip**]), ([**2016**],[**FedShip**])}

5. **Ячейка**-точка пространства многомерного куба

6. **Мера**- значение в ячейке (**Measure**) или значение меры.

Например: В **2016** году компания **FedShip** перевезла

во **Францию** общее количество грузов на сумму **570 000 \$**.

Значение **меры** равно **570 000**.

2.5 Классификация OLAP –технологий.

Оперативные базы данных - OLTP решают задачи динамического взаимодействия с пользователями. Исходные данные для OLAP-кубов обычно хранятся в реляционных базах данных.

В хранилище данных (Data Warehouse) данные OLTP загружаются в многомерную базу для анализа.

Существует три основных способа реализации многомерной модели- MOLAP, ROLAP, HOLAP.

- многомерная OLAP (Multidimensional OLAP — MOLAP);
- реляционная OLAP (Relational OLAP — ROLAP);
- гибридная OLAP (Hybrid OLAP — HOLAP).

MOLAP — это классическая форма OLAP, так что её часто называют просто OLAP. Она использует суммирующую БД, специальный вариант процессора пространственных БД и создаёт требуемую пространственную схему данных с сохранением как базовых данных, так и агрегатов.

ROLAP работает напрямую с реляционным хранилищем, факты и таблицы с измерениями хранятся в реляционных таблицах, и для хранения агрегатов создаются дополнительные реляционные таблицы.

HOLAP использует реляционные таблицы для хранения базовых данных и многомерные таблицы для агрегатов.

2.5.1 Описание МПД

- 2) Измерение (dimension) – описывает элемент данных, по которому производится анализ.
- 3) Элемент – одна точка на измерении.
- 4) Значение элемента – уникальная характеристика элемента.
- 5) Атрибут – полная коллекция элементов одного типа.
- 6) Размер (кардинальное число) – кардинальность измерения – количество элементов измерения.
- 7) Кортеж – координата в многомерном пространстве.
- 8) Срез – секция или сечение многомерного пространства, которая может быть определена кортежем.

Многомерное пространство определяется набором координат (кортежем).

2.5.2 Атрибуты измерений

Если в запросе проходит время по месяцам, то по координатам могут быть выделены наиболее крупные элементы, например, квартал.

Данные на оси могут иметь разную степень детализации => **иерархия измерений**.

Три типа иерархии измерений:

- 1) Сбалансированная
- 2) Несбалансированная
- 3) Неровная

Ячейки. Если меняем шкалу измерения, то появляются новые позиции, соответствующие новым элементам атрибутов. Эти элементы атрибутов содержат множество новых точек в пространстве, но для них нет данных в исходных таблицах. Значения их могут быть вычислены из значений, заданных фактическими данными -> появляется новое пространство – логическое пространство данных.

Полный набор точек пространства, объединяющий фактические и логические пространства, называется многомерным кубом или моделью.

Мера – значение данных в ячейке.

Функция агрегирования данных – функция, позволяющая вычислять значения ячеек в логическом пространстве из значения ячеек фактического пространства.

Подкуб – часть полного пространства куба в виде некоторой многомерной фигуры в виде куба; срезы – это подкубы. Подкубы могут иметь нормальную и произвольную форму.

2.5.3. Классификация OLAP-технологий

Способы организации OLAP-систем:

1. MOLAP – классическая форма OLAP, использующая многомерную БД. Данные представляют собой упорядоченные многомерные массивы, разделенные на гиперкубы, атрибуты которых имеют одинаковую размерность, и поликубы, в которых каждый атрибут хранится с собственным набором измерений.

Преимущества MOLAP:

- Легко встраиваются различные функции;
- Поиск и выборка выполняются быстрее, чем в реляционных БД.

Недостатки:

- В многомерной БД резко уменьшается объем детализированных данных, что сокращает диапазон анализа;
- Увеличивается объем информации;
- Многомерная БД чувствительна к изменениям в многомерной модели.

Условия эффективности использования многомерных БД:

- объем исходных данных составляет порядка нескольких гигабайт;
- набор измерений стабилен;
- требуется широкое использование встроенных функций для выполнения вычислений над ячейками гиперкуба, в том числе написание пользовательских функций.

2. ROLAP работает напрямую с реляционным хранилищем, факты и таблицы с измерениями хранятся в реляционных таблицах, и для хранения агрегатов создаются дополнительные реляционные таблицы. При этом размер хранилища не критичен, реляционная БД обеспечивает значительно больший уровень защиты данных и дает хорошие возможности разграничения прав доступа.

Основной недостаток ROLAP – невысокая производительность, однако при тщательной настройке ROLAP может быть приближена по скорости работы к MOLAP.

3. HOLAP (hybrid OLAP) использует реляционные таблицы для хранения базовых данных и многомерные таблицы для хранения агрегированных данных.

В лабораторных работах по данному учебному курсу используется ROLAP.

2.5.4. Структура системы, используемой в лабораторных работах.

Разработан и реализован вариант системы с использованием ROLAP хранилища данных, позволяющего производить оперативно-аналитический анализ данных, а также с использованием программного обеспечения для интеллектуального анализа данных.

Поскольку большинство корпоративных систем используют реляционную модель хранения данных, то для хранения данных в данной конфигурации выбрана система управления реляционными базами данных MySQL (версия 5.5). Данная система выбрана с учетом простоты ее настройки и поддержки. Тем не менее, она может быть легко заменена на более новые ее аналоги, например, MariaDB или PostgreSQL.

Рассмотрим структуру хранилища данных.

В первую очередь, необходимо отметить, что для OLAP-хранилища данных необходим специальный OLAP-сервер. Есть несколько вариантов свободно распространяемый OLAP-серверов, но в данном конкретном случае используется Mondrian OLAP-Server, от компании Pentaho. Выбор именно этого сервера обусловлен моделью хранения данных: поскольку используется реляционная база данных, необходим сервер, поддерживающий ROLAP модель хранения данных. Mondrian данную модель поддерживает.

Для запуска Mondrian, необходим веб-сервер java-приложений. В качестве такого сервера выбран свободно распространяемый проект Apache Tomcat (ранее Catalina).

Для удобства построения запросов в хранилище данных необходим OLAP-клиент. В данной конфигурации используется плагин Saiku (Community Edition). Он имеет удобный графический интерфейс пользователя с возможностью визуального формирования запроса. Это позволяет любому желающему с легкостью составить любой запрос к хранилищу данных.

В дополнение к оперативно-аналитическому анализу данных, планируется интеллектуальный (статистический) анализ данных. Для этого в данной конфигурации решено использовать язык R.

Для удобства работы с этим языком используется среда разработки R-Studio. Для быстрого доступа к интерфейсу среды через веб-браузер используется R-Studio Server. Он устанавливается на основной сервер системы анализа данных.

Но также возможно использование клиентского приложения R-Studio Desktop. Оно устанавливается на клиентскую машину и предоставляет свой интерфейс для работы.

Общая структура рассмотренной системы анализа данных представлена на рисунке 1.

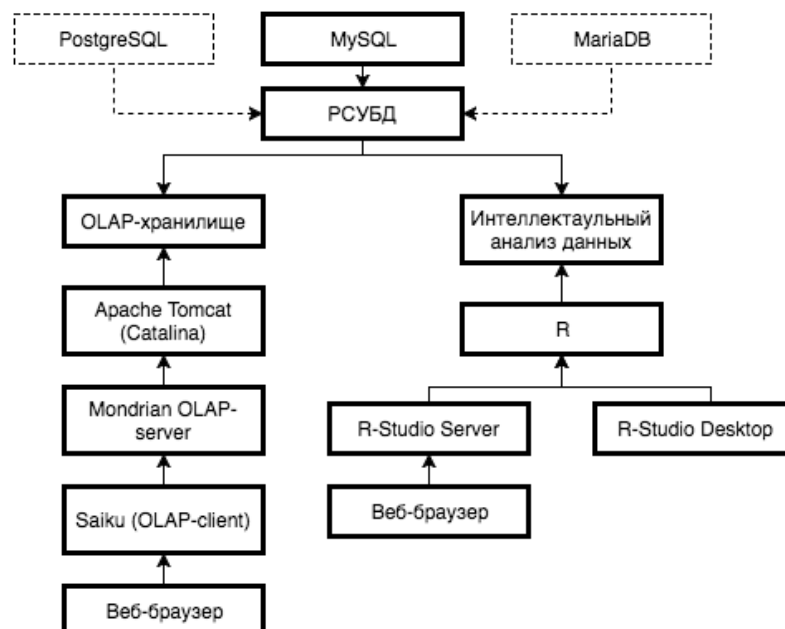


Рис 2.3. Структура хранилища данных

2.5.5 Схемы для реализации многомерного представления данных

Существуют 2 схемы для реализации многомерного представления данных с помощью таблиц:

- Звезда – каждое измерение хранится в одной таблице;

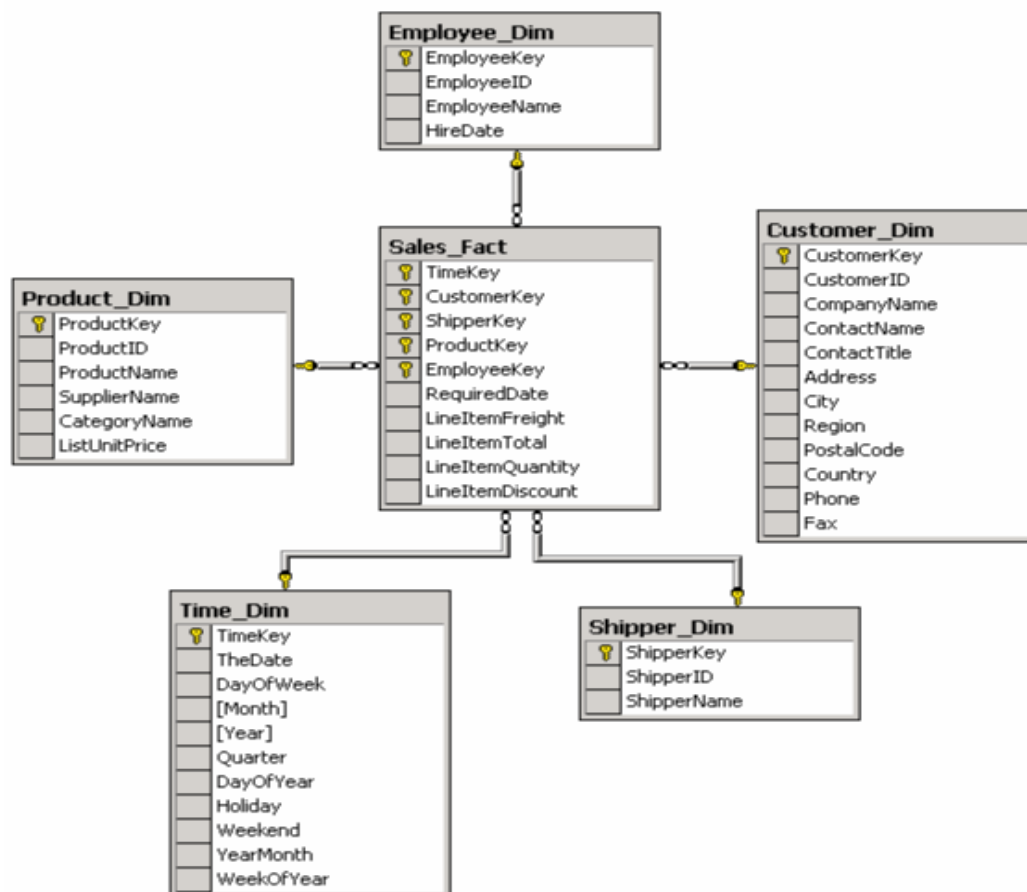


Рис 2.4 Пример схемы «Звезда»

- Снежинка – одно измерение может храниться в нескольких таблицах.

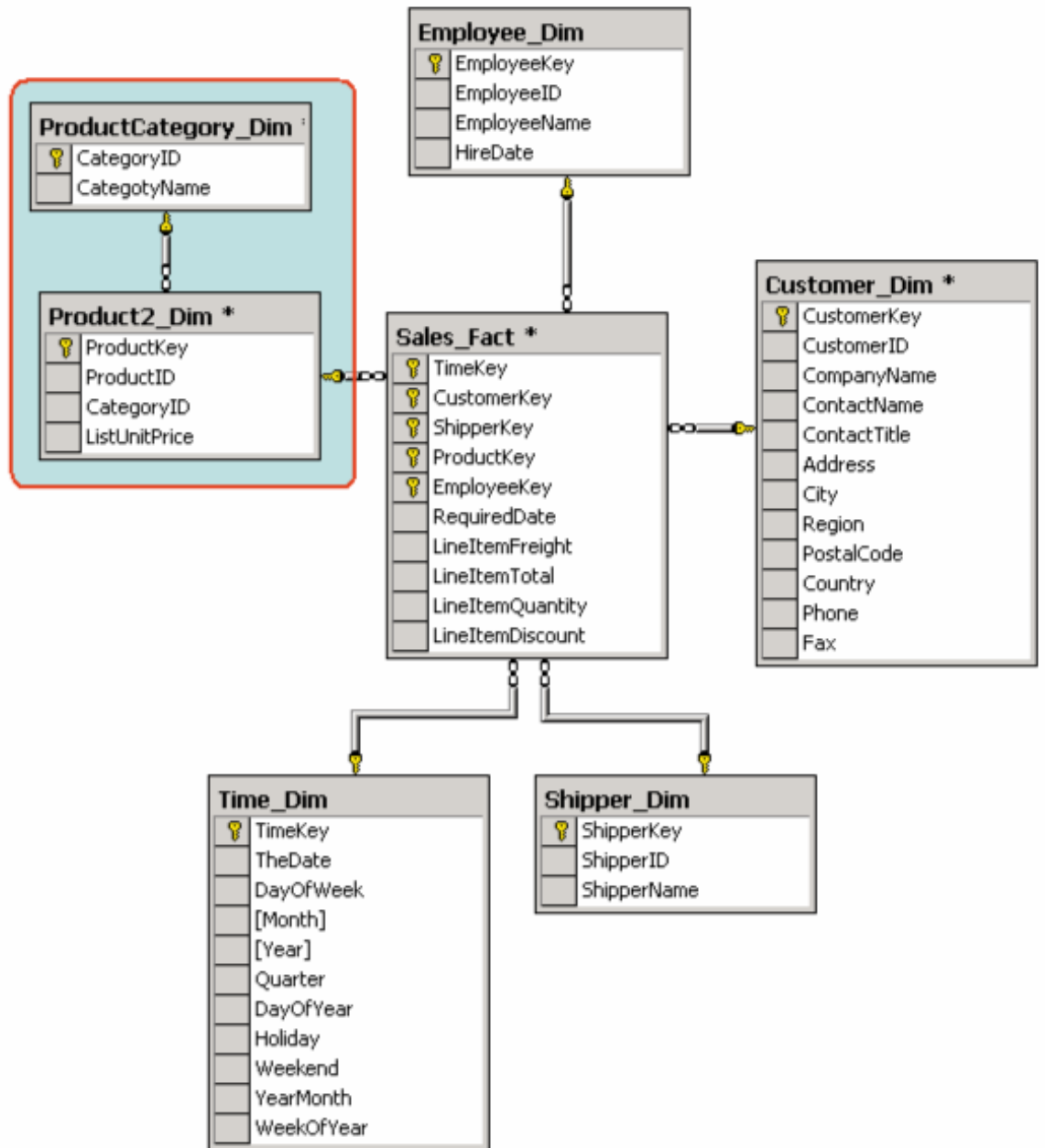



Рис.2.5. Пример схемы «Снежинка»

Каждая схема в центре имеет таблицу фактов, которая имеет связи с таблицами измерений типа «один»-«ко»-«многим».

Основой для хранилища данных является денормализованная модель, составляющими которой являются таблица фактов (fact table) и таблицы измерений (dimension tables).

SQL Server Enterprise Manager - [3:Data in Table 'Sales_Fact' in 'Northwind_Mart' on 'MAINDESK']

Console Window Help



	TimeKey	CustomerKey	ShipperKey	ProductKey	EmployeeKey	RequiredDate	LineItemFreight	LineItemTotal	LineItemQuantity	LineItemDiscount
3	85	4	11	5	01.08.1996	14.3904	168	12	0	
5	85	4	42	5	01.08.1996	11.992	98	10	0	
5	85	4	72	5	01.08.1996	5.996	174	5	0	
1	79	1	14	6	16.08.1996	2.1321	167.4	9	0	
1	79	1	51	6	16.08.1996	9.476	1696	40	0	
3	34	2	41	4	05.08.1996	10.971	77	10	0	
3	34	2	51	4	05.08.1996	38.3985	1484	35	222.6	
3	34	2	65	4	05.08.1996	16.4565	252	15	37.8	
4	84	1	22	3	05.08.1996	6.0492	100.8	6	5.04	
4	84	1	57	3	05.08.1996	15.123	234	15	11.7	
4	84	1	65	3	05.08.1996	20.164	336	20	0	
2	76	2	20	4	06.08.1996	19.54	2592	40	129.6	
2	76	2	33	4	06.08.1996	12.2125	50	25	2.5	
2	76	2	60	4	06.08.1996	19.54	1088	40	0	
5	34	2	31	3	24.07.1996	11.404	200	20	0	
-	-	-	-	-	-	-	-	-	-	

Sales_Fact	
🔑	TimeKey
🔑	CustomerKey
🔑	ShipperKey
🔑	ProductKey
🔑	EmployeeKey
	RequiredDate
	LineItemFreight
	LineItemTotal
	LineItemQuantity
	LineItemDiscount

Рис.2.6.Пример таблицы фактов

Таблица фактов является основной таблицей хранилища данных и , как правило, содержит уникальный составной ключ, объединяющий первичные ключи таблиц измерений. Как правило, она содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться.

Обычно говорят о четырех наиболее часто встречающихся типах фактов. К ним относятся:

- факты, связанные с транзакциями (Transaction facts). Они основаны на отдельных событиях, типичными примерами которых являются телефонный звонок или снятие денег со счета с помощью банкомата;

- факты, связанные с «моментальными снимками» (Snapshot facts). Основаны на состоянии объекта (например, банковского счета) в определенные моменты времени, например на конец дня или месяца. Типичными примерами таких фактов являются объем продаж за день или дневная выручка;

- факты, связанные с элементами документа (Line-item facts). Основаны на том или ином документе (например, счете за товар или услуги) и содержат подробную информацию об элементах этого документа (например, количестве, цене, проценте скидки);

- факты, связанные с событиями или состоянием объекта (Event or state facts). Представляют возникновение события без подробностей о нем (например, просто факт продажи или факт отсутствия таковой без иных подробностей).

Таблицы измерений содержат неизменяемые либо редко изменяемые данные. В подавляющем большинстве случаев эти данные представляют собой по одной записи для каждого члена нижнего уровня иерархии в измерении.

SQL Server Enterprise Manager - [4:Data in Table 'Product_Dim' in 'Northwind_Mart' on 'MAINDESK']

Console Window Help

SQL

	ProductKey	ProductID	ProductName	SupplierName	CategoryName	ListUnitPrice
▶	1	1	Chai	Exotic Liquids	Beverages	18
	2	2	Chang	Exotic Liquids	Beverages	19
	3	3	Aniseed Syrup	Exotic Liquids	Condiments	10
	4	4	Chef Anton's Cajun Seasoning	New Orleans Cajun Delights	Condiments	22
	5	5	Chef Anton's Gumbo Mix	New Orleans Cajun Delights	Condiments	21.35
	6	6	Grandma's Boysenberry Spread	Grandma Kelly's Homestead	Condiments	25
	7	7	Uncle Bob's Organic Dried Pears	Grandma Kelly's Homestead	Produce	30
	8	8	Northwoods Cranberry Sauce	Grandma Kelly's Homestead	Condiments	40
	9	9	Mishi Kobe Niku	Tokyo Traders	Meat/Poultry	97
	10	10	Ikura	Tokyo Traders	Seafood	31
	11	11	Queso Cabrales	Cooperativa de Quesos 'Las Cabras'	Dairy Products	21
	12	12	Queso Manchego La Pastora	Cooperativa de Quesos 'Las Cabras'	Dairy Products	38
	13	13	Konbu	Mayumi's	Seafood	6
	14	14	Tofu	Mayumi's	Produce	23.25
	15	15	Genen Shouyu	Mayumi's	Condiments	15.5
	16	16	Pavlova	Pavlova, Ltd.	Confections	17.45
	17	17	Alice Mutton	Pavlova, Ltd.	Meat/Poultry	39
	18	18	Carnarvon Tigers	Pavlova, Ltd.	Seafood	62.5
	19	19	Teatime Chocolate Biscuits	Specialty Biscuits, Ltd.	Confections	9.2

Product_Dim

🔑	ProductKey
	ProductID
	ProductName
	SupplierName
	CategoryName
	ListUnitPrice

Рис.2.7. Пример таблицы измерений

2.5.6 Структура хранилища NorthWind.

Рассмотрим структуру хранилища NorthWind.

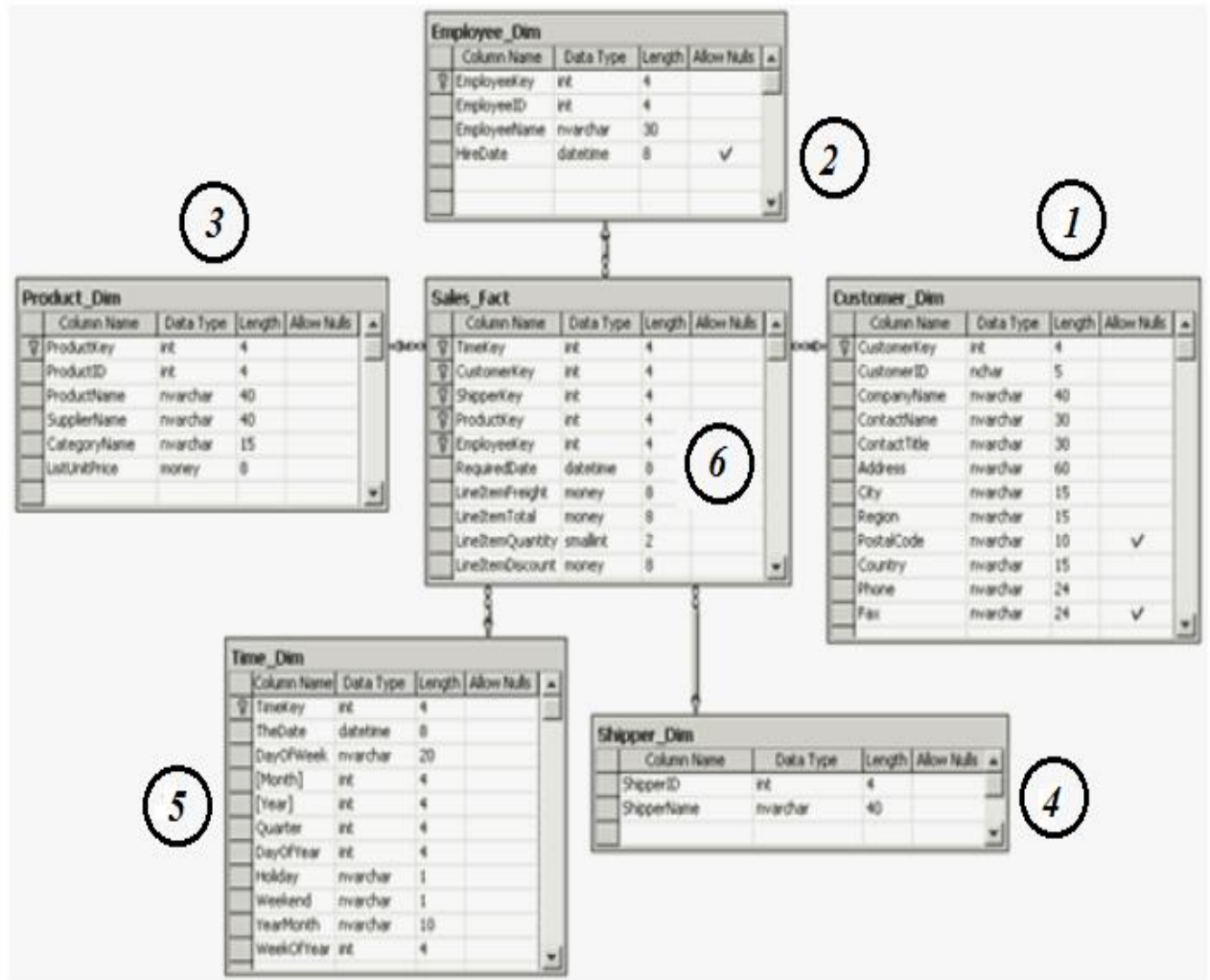


Рис. 2.8. Структура хранилища данных

Nornhwind

В центре хранилища – таблица фактов (sales facts), а вокруг нее располагаются таблицы измерений, соединенные с таблицей фактов связями «один-ко-многим».

В создаваемом хранилище присутствуют пять измерений:

- Перевозчик: ключ – id Перевозчика – имя Перевозчика

- Продукт: ключ – id продукта – наименование – имя поставщика – наименование категории – цена единицы продукта
- Покупатель: ключ – id – имя компании – контактное лицо – должность – адрес – город – регион – почтовый индекс – страна – телефон - факс
- Работник: ключ – id – имя работника – дата приема на работу
- Время

Таблица фактов содержит следующие поля:

- 5 полей – ключи таблиц измерений;
- Предполагаемая дата заказа;
- Плата за перевозку грузов морем;
- Суммарная стоимость заказа;
- Количество перевозимого товара;
- Скидка.

Особое внимание следует уделить измерению «Время». Члены измерений или осей могут быть объединены одной или несколькими иерархиями. Зачем нужно несколько иерархий? Например, по оси с датой заказа можно группировать точки (т.е. дни доставки заказов) по иерархии Год-Месяц-День и др.

В нашем случае для измерения «Время» используется следующая иерархия:

Год-квартал-месяц-номер месяца-номер недели-день недели-день года-праздник-выходной.



	1998	1997	1996
	Federal Shipping	Speedy Express	United Package
Argentina	11806.28	9190.48	1263.9
Austria			4039.5
Belgium	1745.42	1207.28	14924.12
Brazil			5208.28
Canada	2952.4		
Denmark	1739.76	1376	
Finland	5470.98	3538.92	2328.46
France	11927.48	9823.43	11052.28
Germany	2208.62	1739.6	4681.16
Ireland		330.9	608
Italy	2139.1		1357.6
Mexico			786
Norway	459		
Poland	1268.3	716.72	285.12
Portugal	236.5	220.3	2235.8
Spain	3021.23	2380	1488.8
Sweden	2490.5		1628.32
Switzerland	5094.88	1520.8	901.2
UK	11192.65	6347.52	14091.93
USA	3925.58	3171.92	
Venezuela	11806.28	9190.48	1263.9

Рис. 2.9. Трехмерный набор данных ХД «Nornhwind»