Принципы организации хранилища

- *Проблемно-предметная ориентация*. Данные объединяются в категории и хранятся в соответствии с областями, которые они описывают, а не с приложениями, которые они используют.
- *Интегрированность*. Данные объединены так, чтобы они удовлетворяли всем требованиям предприятия в целом, а не единственной функции бизнеса.
- *Некорректируемость*. Данные в хранилище данных не создаются: то есть поступают из внешних источников, не корректируются и не удаляются.
- *Зависимость от времени*. Данные в хранилище точны и корректны только в том случае, когда они привязаны к некоторому промежутку или моменту времени.

Существуют два основных архитектурных направления — нормализованные хранилища данных и хранилища с измерениями.

В нормализованных хранилищах, данные находятся в предметно ориентированных таблицах третьей нормальной формы. Нормализованные хранилища характеризуются как простые в создании и управлении, недостатки нормализованных хранилищ — большое количество таблиц как следствие нормализации, из-за чего для получения какой-либо информации нужно делать выборку из многих таблиц одновременно, что приводит к ухудшению производительности системы. Для решения этой проблемы используются денормализованные таблицы — витрины данных, на основе которых уже выводятся отчетные формы. При громадных объемах данных могут использовать несколько уровней «витрин»/«хранилищ».

Хранилища с измерениями используют схему «звезда» или схему «снежинка». При этом в центре «звезды» находятся данные (таблица фактов), а измерения образуют лучи звезды. Различные таблицы фактов совместно используют таблицы измерений, что значительно облегчает операции объединения данных из нескольких предметных таблиц фактов (пример – факты продаж и поставок товара). Таблицы данных и соответствующие измерения образуют архитектуру «шина». Измерения часто создаются в третьей нормальной форме, в том числе, для протоколирования изменения в измерениях.

Основным недостатком является более сложные процедуры подготовки и загрузки данных, а также управление и изменение измерений данных.

При достаточно большом объеме данных схемы «звезда» и «снежинка» также дают снижение производительности при соединениях с измерениями.

https://studfiles.net/preview/5906694/page:18/

Типичная структура хранилищ данных

Как мы уже знаем, конечной целью использования OLAP является анализ данных и представление результатов этого анализа в виде, удобном для восприятия и принятия решений. Основная идея OLAP заключается в построении многомерных кубов, которые будут доступны для пользовательских запросов. Однако исходные данные для построения OLAP-кубов обычно хранятся в реляционных базах данных. Нередко это специализированные реляционные базы данных, называемые также хранилищами данных (Data Warehouse). В отличие от так называемых оперативных баз данных, с которыми работают приложения, модифицирующие данные, хранилища данных предназначены исключительно для обработки и анализа информации, поэтому проектируются они таким образом, чтобы время выполнения запросов к ним было минимальным. Обычно данные копируются в хранилище из оперативных баз данных согласно определенному расписанию.

Типичная структура хранилища данных существенно отличается от структуры обычной реляционной СУБД. Как правило, эта структура денормализована (это позволяет повысить скорость выполнения запросов), поэтому может допускать избыточность данных.

Для дальнейших примеров мы снова воспользуемся базой данных Northwind, входящей в комплекты поставки Microsoft SQL Server и Microsoft Access. Ее структура данных приведена на рис. 13.

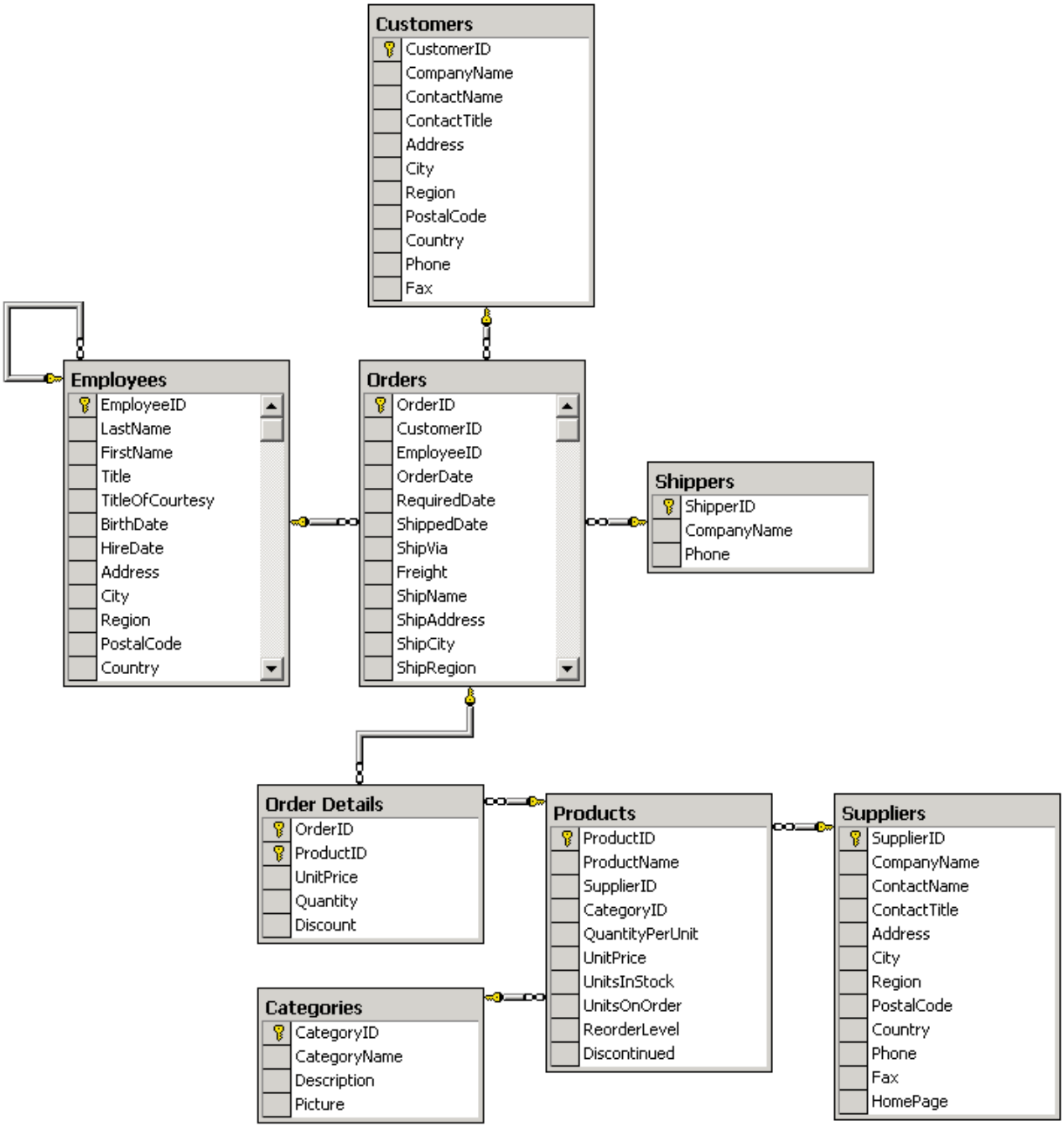


Рис. 13. Структура базы данных Northwind

Основными составляющими структуры хранилищ данных являются таблица фактов (fact table) и таблицы измерений (dimension tables).

Таблица фактов

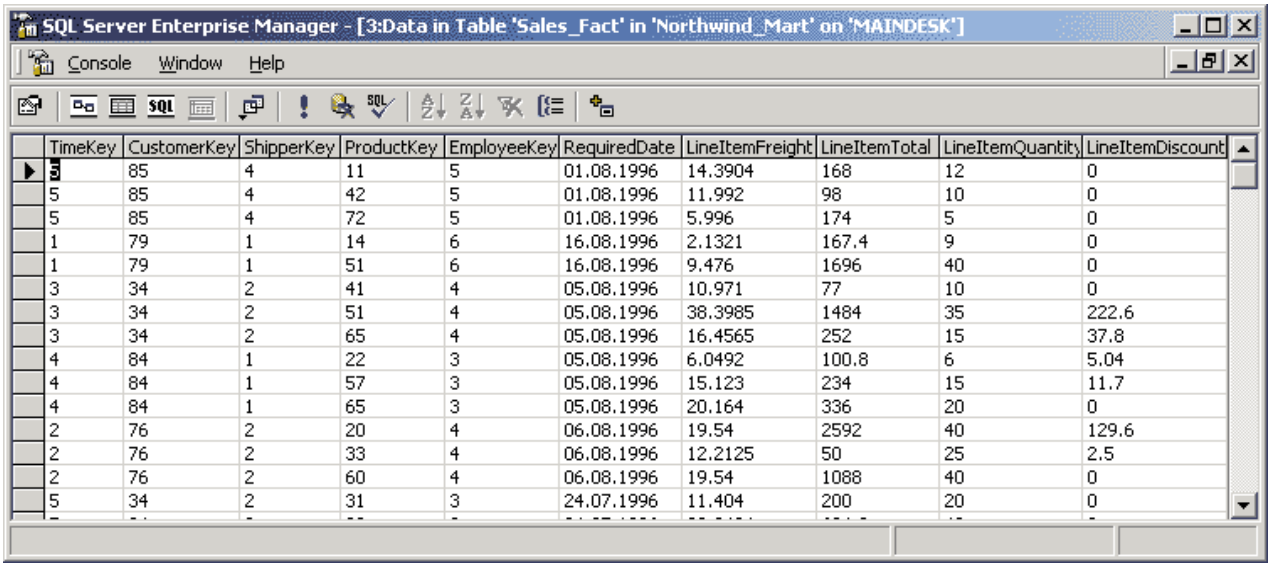
Таблица фактов является основной таблицей хранилища данных. Как правило, она содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться. Обычно говорят о четырех наиболее часто встречающихся **типах фактов**. К ним относятся:

- 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).

Для примера рассмотрим факты, связанные с элементами документа (в данном случае счета, выставленного за товар).

Таблица фактов, как правило, содержит уникальный составной ключ, объединяющий первичные ключи таблиц измерений. Чаще всего это целочисленные значения либо значения типа «дата/время» — ведь таблица фактов может содержать сотни тысяч или даже миллионы записей, и хранить в ней повторяющиеся текстовые описания, как правило, невыгодно — лучше поместить их в меньшие по объему таблицы измерений. При этом как ключевые, так и некоторые неключевые поля должны соответствовать будущим измерениям OLAP-куба. Помимо этого таблица фактов содержит одно или несколько числовых полей, на основании которых в дальнейшем будут получены агрегатные данные.

Пример таблицы фактов, которая может быть построена на основе базы данных Northwind, приведен на 4.



TimeKey	CustomerKey	ShipperKey	ProductKey	EmployeeKey	RequiredDate	LineItemFreight	LineItemTotal	LineItemQuantity	LineItemDiscount
5	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

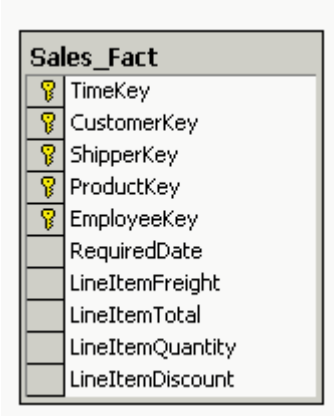


Рис. 14. Пример таблицы фактов

В данном примере измерениям будущего куба соответствуют первые шесть полей, а агрегатным данным — последние четыре.

Отметим, что для многомерного анализа пригодны таблицы фактов, содержащие как можно более подробные данные (то есть соответствующие членам нижних уровней иерархии соответствующих измерений). В данном случае предпочтительнее взять за основу факты продажи товаров отдельным заказчикам, а не суммы продаж для разных стран — последние все равно будут вычислены OLAP-средством. Исключение можно сделать, пожалуй, только для клиентских OLAP-средств (о них мы поговорим чуть позже), поскольку в силу ряда ограничений они не могут манипулировать большими объемами данных.

Отметим, что в таблице фактов нет никаких сведений о том, как группировать записи при вычислении агрегатных данных. Например, в ней есть идентификаторы продуктов или клиентов, но отсутствует информация о том, к какой категории относится данный продукт или в каком городе находится данный клиент. Эти сведения, в дальнейшем используемые для построения иерархий в измерениях куба, содержатся в таблицах измерений.

Организация Хранилища Данных

Данные в ХД делятся *на три основных категории:*

- *Детальные данные* (переносимые из источников),
- *Агрегированные данные* (обобщение путем суммирования фактических данных по определен. измерениям),
- *Метаданные* (информация о содержащихся в Хранилище данных).
  - Описание объектов.
  - Описание пользователей.
  - Описание места хранения.
  - Описание действий над данными.
  - Причины, повлекшие выполнения над данными тех или иных операций.

Применимость систем хранения разных типов

Блочные хранилища

Блочные хранилища обладают набором инструментов, которые обеспечивают повышенную производительность: хост-адаптер шины разгружает процессор и освобождает его ресурсы для выполнения других задач. Поэтому блочные системы хранения часто используются для виртуализации. Также хорошо подходят для работы с базами данных.

позволяют использовать хранилище как жесткий диск, с которым могут производиться операции, как с любым логическим диском: форматирование, установка ОС, использование в сетевых файловых системах и т.п. Файловый уровень может применяться только для удаленного доступа к данным в хранилище с целью хранения и извлечения файлов (как общая папка).

Протоколы **блочного уровня** предоставляют большую скорость доступа в связи с меньшим количеством промежуточных «слоев» при доступе к NAS, но для их настройка достаточно трудоемка. Создать хранилище **на файловом уровне гораздо проще**, однако скорость его работы будет значительно ниже.

Недостатками блочного хранилища являются высокая стоимость и сложность в управлении. Еще один минус блочных хранилищ (который относится и к файловым, о которых далее) — ограниченный объем метаданных. Любую дополнительную информацию приходится обрабатывать на уровне приложений и баз данных.

Файловые хранилища

Среди плюсов файловых хранилищ выделяют простоту. Файлу присваивается имя, он получает метаданные, а затем «находит» себе место в каталогах и подкаталогах. Файловые хранилища обычно дешевле по сравнению с блочными системами, а иерархическая топология удобна при обработке небольших объемов данных. Поэтому с их помощью организуются системы совместного использования файлов и системы локального архивирования.

Пожалуй, основной недостаток файлового хранилища — его «ограниченность». Трудности возникают по мере накопления большого количества данных — находить нужную информацию в куче папок и вложений становится трудно. По этой причине файловые системы не используются в дата-центрах, где важна скорость.

Объектные хранилища

Что касается объектных хранилищ, то они хорошо масштабируются, поэтому способны работать с петабайтами информации. По статистике, объем неструктурированных данных во всем мире достигнет 44 зеттабайт к 2020 году — это в 10 раз больше, чем было в 2013. Объектные хранилища, благодаря своей возможности работать с растущими объемами данных, стали стандартом для большинства из самых популярных сервисов в облаке: от Facebook до DropBox.

Такие хранилища, как Haystack Facebook, ежедневно пополняются 350 млн фотографий и хранят 240 млрд медиафайлов. Общий объем этих данных оценивается в 357 петабайт.

Хранение копий данных — это другая функция, с которой хорошо справляются объектные хранилища. По данным исследований, 70% информации лежит в архиве и редко изменяется. Например, такой информацией могут выступать резервные копии системы, необходимые для аварийного восстановления.

Но недостаточно просто хранить неструктурированные данные, иногда их нужно интерпретировать и организовывать. Файловые системы имеют ограничения в этом плане: управление метаданными, иерархией, резервным копированием — все это становится препятствием. Объектные хранилища оснащены внутренними механизмами для проверки корректности файлов и другими функциями, обеспечивающими доступность данных.

Плоское адресное пространство также выступает преимуществом объектных хранилищ — данные, расположенные на локальном или облачном сервере, извлекаются одинаково просто. Поэтому такие хранилища часто применяются для работы с Big Data и медиа. Например, их используют Netflix и Spotify. Кстати, возможности объектного хранилища сейчас доступны и в сервисе 1cloud.

Схема «звезда»

Главная таблица – таблица фактов, с ней связаны таблицы размерностей.

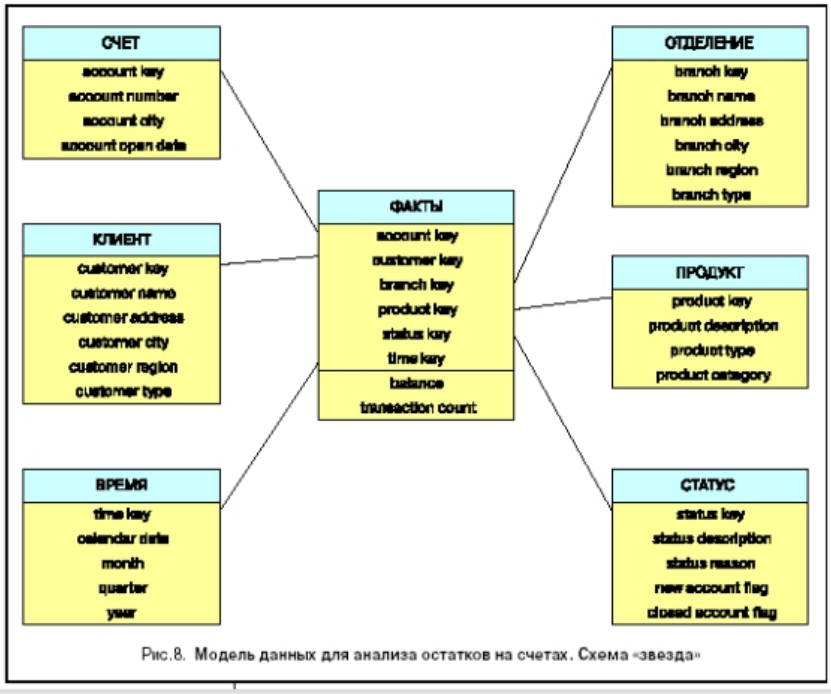


Рис.8. Модель данных для анализа остатков на счетах. Схема «звезда»

Пример использования OLAP-технологии

В компании, занимающейся продажей различных товаров, может быть получен и проанализирован с помощью OLAP следующий многомерный куб: