Содержание:

Введение

Трудно найти в компьютерном мире человека, который хотя бы на интуитивном уровне

не понимал, что такое базы данных и зачем они нужны. В отличие от традиционных

реляционных СУБД, концепция OLAP [1]н

так широко известна, хотя загадочный термин "кубы OLAP" слышали, наверное,

почти все. Что же такое Online Analytical Processing?

OLAP - это не отдельно взятый программный продукт, не язык программирования и

даже не конкретная технология. Если постараться охватить OLAP во всех его

проявлениях, то это совокупность концепций, принципов и требований, лежащих в

основе программных продуктов, облегчающих аналитикам доступ к данным.

Для начала мы выясним, зачем аналитикам надо как-то специально облегчать

доступ к данным. Дело в том, что аналитики - это особые потребители

корпоративной информации. Задача аналитика - находить закономерности в

больших массивах данных. Поэтому аналитик не будет обращать внимания на

отдельно взятый факт, что в четверг четвертого числа контрагенту Чернову

была продана партия черных чернил - ему нужна информация о сотнях и тысячах

подобных событий. Одиночные факты в базе данных могут заинтересовать, к

примеру, бухгалтера или начальника отдела продаж, в компетенции которого

находится сделка. Аналитику одной записи мало - ему, к примеру, могут понадобиться все сделки данного филиала или представительства за месяц, год. Заодно аналитик отбрасывает ненужные ему подробности вроде ИНН покупателя, его точного адреса и номера телефона, индекса контракта и тому подобного. В то же время данные, которые требуются аналитику для работы, обязательно содержат числовые значения - это обусловлено самой сущностью его деятельности.

Централизация и удобное структурирование - это далеко не все, что нужно аналитику. Ему ведь еще требуется инструмент для просмотра, визуализации информации. Традиционные отчеты, даже построенные на основе единого хранилища, лишены одного - гибкости. Их нельзя “покрутить”, “развернуть” или “свернуть”, чтобы получить желаемое представление данных. Конечно, можно вызвать программиста, и он сделает новый отчет достаточно быстро - скажем, в течение часа. Получается, что аналитик может проверить за день не более двух идей. А ему (если он хороший аналитик) таких идей может приходить в голову по нескольку в час. И чем больше “срезов” и “разрезов” данных аналитик видит, тем больше у него идей, которые, в свою очередь, для проверки требуют все новых и новых “срезов”. Вот бы ему такой инструмент, который позволил бы разворачивать и сворачивать данные просто и удобно! В качестве такого инструмента и выступает OLAP.

Хотя OLAP и не представляет собой необходимый атрибут хранилища данных, он все чаще и чаще применяется для анализа накопленных в этом хранилище сведений.

Компоненты, входящие в типичное хранилище, представлены на рис. 1.

Реферат: OLAP технологии

Рисунок 1. Структура хранилища данных

Оперативные данные собираются из различных источников, очищаются, интегрируются

и складываются в реляционное хранилище[2].

При этом они уже доступны для анализа при помощи различных средств построения

отчетов. Затем данные (полностью или частично) подготавливаются для

OLAP-анализа. Они могут быть загружены в специальную БД OLAP или оставлены в

реляционном хранилище. Важнейшим его элементом являются метаданные, т. е.

информация о структуре, размещении и трансформации данных. Благодаря ним

обеспечивается эффективное взаимодействие различных компонентов хранилища.

Подытоживая, можно определить OLAP как совокупность средств многомерного

анализа данных, накопленных в хранилище. Теоретически средства OLAP можно

применять и непосредственно к оперативным данным или их точным копиям (чтобы

не мешать оперативным пользователям). Но мы тем самым рискуем наступить на

уже описанные выше грабли, то есть начать анализировать оперативные данные,

которые напрямую для анализа непригодны.

2 Хранилища данных

2.1 Что такое хранилище данных?

Информационные системы масштаба предприятия, как правило, содержат приложения, предназначенные для комплексного многомерного анализа данных, их динамики, тенденций и т.п. Такой анализ в конечном итоге призван содействовать принятию решений. Нередко эти системы так и называются — системы поддержки принятия решений.

Принять любое управленческое решение невозможно не обладая необходимой для этого информацией, обычно количественной. Для этого необходимо создание хранилищ данных (Data warehouses), то есть процесс сбора, отсеивания и предварительной обработки данных с целью предоставления результирующей информации пользователям для статистического анализа (а нередко и создания аналитических отчетов).

Ральф Кимбалл (Ralph Kimball), один из авторов концепции хранилищ данных, описывал хранилище данных как "место, где люди могут получить доступ к своим данным" (см., например, Ralph Kimball, "The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses", John Wiley & Sons, 1996 и "The Data Web house Toolkit: Building the Web-Enabled Data Warehouse", John Wiley & Sons, 2000). Он же сформулировал и основные требования к хранилищам данных:

· поддержка высокой скорости получения данных из хранилища;

· поддержка внутренней непротиворечивости данных;

· возможность получения и сравнения так называемых срезов данных (slice and dice);

· наличие удобных утилит просмотра данных в хранилище;

· полнота и достоверность хранимых данных;

· поддержка качественного процесса пополнения данных.

Удовлетворять всем перечисленным требованиям в рамках одного и того же продукта зачастую не удается. Поэтому для реализации хранилищ данных обычно используется несколько продуктов, одни их которых представляют собой собственно средства хранения данных, другие — средства их извлечения и просмотра, третьи — средства их пополнения и т.д.

Типичное хранилище данных, как правило, отличается от обычной реляционной базы данных. Во-первых, обычные базы данных предназначены для того, чтобы помочь пользователям выполнять повседневную работу, тогда как хранилища данных предназначены для принятия решений. Например, продажа товара и выписка счета производятся с использованием базы данных, предназначенной для обработки транзакций, а анализ динамики продаж за несколько лет, позволяющий спланировать работу с поставщиками, — с помощью хранилища данных.

Во-вторых, обычные базы данных подвержены постоянным изменениям в процессе работы пользователей, а хранилище данных относительно стабильно: данные в нем обычно обновляются согласно расписанию (например, еженедельно, ежедневно или ежечасно — в зависимости от потребностей). В идеале процесс пополнения представляет собой просто добавление новых данных за определенный период времени без изменения прежней информации, уже находящейся в хранилище.

И, в-третьих, обычные базы данных чаще всего являются источником данных, попадающих в хранилище. Кроме того, хранилище может пополняться за счет внешних источников, например статистических отчетов.

2.2 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. 2.

Реферат: OLAP технологии

Рисунок 2. Структура базы данных Northwind

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

2.2.1 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 рис. 3.

Реферат: OLAP технологии

Реферат: OLAP технологии

Рисунок 3. 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.

2.2.2 Таблицы измерений

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. Таблицы измерений

также содержат как минимум одно описательное поле (обычно с именем члена измерения) и, как правило, целочисленное ключевое поле (обычно это суррогатный ключ) для однозначной идентификации члена измерения. Если будущее
измерение, основанное на данной таблице измерений, содержит иерархию, то таблица измерений также может содержать поля, указывающие на «родителя» данного члена в этой иерархии. Нередко (но не всегда) таблица измерений может содержать и поля, указывающие на «прародителей», и иных «предков» в данной иерархии (это обычно характерно для сбалансированных иерархий), а также дополнительные атрибуты членов измерений, содержавшиеся в исходной оперативной базе данных (например, адреса и телефоны клиентов).

Каждая таблица измерений должна находиться в отношении «один ко многим» с таблицей фактов.

Отметим, что скорость роста таблиц измерений должна быть незначительной по сравнению со скоростью роста таблицы фактов; например, добавление новой записи в таблицу измерений, характеризующую товары, производится только при появлении нового товара, не продававшегося ранее.

Пример таблицы измерений приведен на рис. 4.

Реферат: OLAP технологии

Реферат: OLAP технологии

Рисунок 4. Пример таблицы измерений

Одно измерение куба может содержаться как в одной таблице (в том числе и при наличии нескольких уровней иерархии), так и в нескольких связанных таблицах,

соответствующих различным уровням иерархии в измерении. Если каждое измерение

содержится в одной таблице, такая схема хранилища данных носит название

«звезда» (star schema). Пример такой схемы приведен на рис. 5.

Реферат: OLAP технологии

Рисунок 5. Пример схемы "звезда"

Если же хотя бы одно измерение содержится в нескольких связанных таблицах,

такая схема хранилища данных носит название «снежинка» (snowflake schema).

Дополнительные таблицы измерений в такой схеме, обычно соответствующие

верхним уровням иерархии измерения и находящиеся в соотношении «один ко

многим» в главной таблице измерений, соответствующей нижнему уровню иерархии,

иногда называют консольными таблицами (outrigger table). Пример схемы

«снежинка» приведен на рис. 6.

Реферат: OLAP технологии

Рисунок 6. Пример схемы "снежинка"

Отметим, что даже при наличии иерархических измерений с целью повышения

скорости выполнения запросов к хранилищу данных нередко предпочтение отдается

схеме «звезда».

Однако не все хранилища данных проектируются по двум приведенным выше схемам.

Так, довольно часто вместо ключевого поля для измерения, содержащего данные

типа «дата», и соответствующей таблицы измерений сама таблица фактов может

содержать ключевое поле типа «дата». В этом случае соответствующая таблица измерений просто отсутствует.

В случае несбалансированной иерархии (например, такой, которая может быть основана на таблице Employees базы данных Northwind, имеющей поле Employee ID, которое одновременно является и первичным, и внешним ключом и отражает подчиненность одних сотрудников другим (см. рис. 6) в схему «снежинка» также следует вносить коррективы. В этом случае обычно в таблице измерений присутствует связь, аналогичная соответствующей связи в оперативной базе данных.

Еще один пример отступления от правил — наличие нескольких разных иерархий для одного и того же измерения. Типичные примеры таких иерархий — иерархии для календарного и финансового года (при условии, что финансовый год начинается не с 1 января), или с различными способами группировки членов измерения (например, группировать товары можно по категориям, а можно и по компаниям-поставщикам). В этом случае таблица измерений содержит поля для всех возможных иерархий с одними и теми же членами нижнего уровня, но с разными членами верхних уровней (пример такой таблицы приведен на рис. 3).

Как мы уже отмечали выше, таблица измерений может содержать поля, не имеющие

отношения к иерархиям и представляющие собой просто дополнительные атрибуты

членов измерений (member properties). Иногда такие атрибуты могут быть использованы при анализе данных.

Следует сказать, что для создания реляционных хранилищ данных нередко

применяются специализированные СУБД, хранение данных в которых оптимизировано

с точки зрения скорости выполнения запросов. Примером такого продукта является Sybase Adaptive Server IQ, реализующий нетрадиционный способ хранения данных в таблицах (не по строкам, а по столбцам). Однако создавать хранилища можно и в обычных реляционных СУБД.

2.3 OLAP на клиенте и на сервере

Многомерный анализ данных может быть произведен с помощью различных средств,

которые условно можно разделить на клиентские и серверные OLAP-средства.

Клиентские OLAP-средства представляют собой приложения, осуществляющие вычисление агрегатных данных (сумм, средних величин, максимальных или минимальных значений) и их отображение, при этом сами агрегатные данные содержатся в кэше внутри адресного пространства такого OLAP-средства.

Если исходные данные содержатся в настольной СУБД, вычисление агрегатных данных производится самим OLAP-средством. Если же источник исходных данных —

серверная СУБД, многие из клиентских OLAP-средств посылают на сервер SQL-запросы, содержащие оператор GROUP BY, и в результате получают агрегатные данные, вычисленные на сервере.

Как правило, OLAP-функциональность реализована в средствах статистической обработки данных (из продуктов этого класса на российском рынке широко распространены продукты компаний Stat Soft и SPSS) и в некоторых электронных

таблицах. В частности, неплохими средствами многомерного анализа обладает Microsoft Excel 2000. С помощью этого продукта можно создать и сохранить в виде файла небольшой локальный многомерный OLAP-куб и отобразить его двух- или трехмерные сечения.

Многие средства разработки содержат библиотеки классов или компонентов, позволяющие создавать приложения, реализующие простейшую OLAP-функциональность (такие, например, как компоненты Decision Cube в Borland Delphi и Borland C++Builder). Помимо этого многие компании предлагают элементы управления ActiveX и другие библиотеки, реализующие подобную функциональность.

Отметим, что клиентские OLAP-средства применяются, как правило, при малом числе измерений (обычно рекомендуется не более шести) и небольшом разнообразии значений этих параметров, — ведь полученные агрегатные данные должны умещаться в адресном пространстве подобного средства, а их количество растет экспоненциально при увеличении числа измерений. Поэтому даже самые примитивные клиентские OLAP-средства, как правило, позволяют произвести предварительный подсчет объема требуемой оперативной памяти для создания в ней многомерного куба.

Многие (но не все!) клиентские OLAP-средства позволяют сохранить содержимое кэша с агрегатными данными в виде файла, что, в свою очередь, позволяет не производить их повторное вычисление. Отметим, что нередко такая возможность используется для отчуждения агрегатных данных с целью передачи их другим организациям или для публикации. Типичным примером таких отчуждаемых

агрегатных данных является статистика заболеваемости в разных регионах и в различных возрастных группах, которая является открытой информацией, публикуемой министерствами здравоохранения различных стран и Всемирной организацией здравоохранения. При этом собственно исходные данные, представляющие собой сведения о конкретных случаях заболеваний, являются конфиденциальными данными медицинских учреждений, которые ни в коем случае не должны попадать в руки страховых компаний и тем более становиться достоянием гласности.

Идея сохранения кэша с агрегатными данными в файле получила свое дальнейшее развитие в серверных OLAP-средствах, в которых сохранение и изменение агрегатных данных, а также поддержка содержащего их хранилища осуществляются отдельным приложением или процессом, называемым OLAP-сервером. Клиентские приложения могут запрашивать подобное многомерное хранилище и в ответ получать те или иные данные. Некоторые клиентские приложения могут также создавать такие хранилища или обновлять их в соответствии с изменившимися исходными данными.

Преимущества применения серверных OLAP-средств по сравнению с клиентскими OLAP-средствами сходны с преимуществами применения серверных СУБД по сравнению с настольными: в случае применения серверных средств вычисление и хранение агрегатных данных происходят на сервере, а клиентское приложение получает лишь результаты запросов к ним, что позволяет в общем случае снизить сетевой трафик, время выполнения запросов и требования к ресурсам,

потребляемым клиентским приложением. Отметим, что средства анализа и обработки данных масштаба предприятия, как правило, базируются именно на серверных OLAP-средствах, например, таких как Oracle Express Server, Microsoft SQL Server 2000 Analysis Services, Hyperion Essbase, продуктах компаний Crystal Decisions, Business Objects, Cognos, SAS Institute.

Поскольку все ведущие производители серверных СУБД производят (либо лицензировали у других компаний) те или иные серверные OLAP-средства, выбор их достаточно широк и почти во всех случаях можно приобрести OLAP-сервер того же производителя, что и у самого сервера баз данных.

Отметим, что многие клиентские OLAP-средства (в частности, Microsoft Excel 2000, Seagate Analysis и др.) позволяют обращаться к серверным OLAP-хранилищам, выступая в этом случае в роли клиентских приложений, выполняющих подобные запросы. Помимо этого имеется немало продуктов, представляющих собой клиентские приложения к OLAP-средствам различных производителей.

2.4 Технические аспекты многомерного хранения данных

В многомерных хранилищах данных содержатся агрегатные данные различной степени подробности, например, объемы продаж по дням, месяцам, годам, по категориям товаров и т.п. Цель хранения агрегатных данных — сократить время выполнения запросов, поскольку в большинстве случаев для анализа и прогнозов интересны не детальные, а суммарные данные. Поэтому при создании многомерной

базы данных всегда вычисляются и сохраняются некоторые агрегатные данные. Отметим, что сохранение всех агрегатных данных не всегда оправданно. Дело в том, что при добавлении новых измерений объем данных, составляющих куб, растет экспоненциально (иногда говорят о «взрывном росте» объема данных). Если говорить более точно, степень роста объема агрегатных данных зависит от количества измерений куба и членов измерений на различных уровнях иерархий этих измерений. Для решения проблемы «взрывного роста» применяются разнообразные схемы, позволяющие при вычислении далеко не всех возможных агрегатных данных достичь приемлемой скорости выполнения запросов.

Как исходные, так и агрегатные данные могут храниться либо в реляционных, либо в многомерных структурах. Поэтому в настоящее время применяются три способа хранения данных:

· MOLAP (Multidimensional OLAP) –— исходные и агрегатные данные хранятся в многомерной базе данных. Хранение данных в многомерных структурах позволяет манипулировать данными как многомерным массивом, благодаря чему скорость вычисления агрегатных значений одинакова для любого из измерений. Однако в этом случае многомерная база данных оказывается избыточной, так как многомерные данные полностью содержат исходные реляционные данные.

· ROLAP (Relational OLAP) — исходные данные остаются в той же реляционной базе данных, где они изначально и находились. Агрегатные же данные помещают в специально созданные для их хранения служебные таблицы в той же базе данных.

· HOLAP (Hybrid OLAP) — исходные данные остаются в той же реляционной

базе данных, где они изначально находились, а агрегатные данные хранятся в многомерной базе данных.

Некоторые OLAP-средства поддерживают хранение данных только в реляционных структурах, некоторые — только в многомерных. Однако большинство современных серверных OLAP-средств поддерживают все три способа хранения данных. Выбор способа хранения зависит от объема и структуры исходных данных, требований к скорости выполнения запросов и частоты обновления OLAP-кубов.

Отметим также, что подавляющее большинство современных OLAP-средств не хранит «пустых» значений (примером «пустого» значения может быть отсутствие продаж сезонного товара вне сезона).

3 Основные понятия OLAP

3.1 Тест FAMSI

Технология комплексного многомерного анализа данных получила название OLAP (On-Line Analytical Processing). OLAP — это ключевой компонент организации хранилищ данных. Концепция OLAP была описана в 1993 году Эдгаром Коддом, известным исследователем баз данных и автором реляционной модели данных (см. E.F. Codd, S.B. Codd, and C.T.Salley, Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Technical report, 1993). В 1995 году на основе требований, изложенных Коддом, был сформулирован так называемый тест FASMI (Fast Analysis of Shared Multidimensional Information —

быстрый анализ разделяемой многомерной информации), включающий следующие требования к приложениям для многомерного анализа:

· Fast (Быстрый) - предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;

· Analysis (Анализ) - возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;

· Shared (Разделяемой) - многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;

· Multidimensional (Многомерной) - многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это — ключевое требование OLAP);

· Information (Информации) - приложение должно иметь возможность обращаться к любой нужной информации, независимо от ее объема и места хранения.

Следует отметить, что OLAP-функциональность может быть реализована различными способами, начиная с простейших средств анализа данных в офисных приложениях и заканчивая распределенными аналитическими системами, основанными на серверных продуктах.

3.2 Многомерное представление информации

3.3 Кубы

OLAP предоставляет удобные быстродействующие средства доступа, просмотра и анализа деловой информации. Пользователь получает естественную, интуитивно понятную модель данных, организуя их в виде многомерных кубов (Cubes). Осями многомерной системы координат служат основные атрибуты анализируемого бизнес-

процесса. Например, для продаж это могут быть товар, регион, тип покупателя. В качестве одного из измерений используется время. На пересечениях осей - измерений (Dimensions) - находятся данные, количественно характеризующие процесс - меры (Measures). Это могут быть объемы продаж в штуках или в денежном выражении, остатки на складе, издержки и т. п. Пользователь, анализирующий информацию, может “разрезать” куб по разным направлениям, получать сводные (например, по годам) или, наоборот, детальные (по неделям) сведения и осуществлять прочие манипуляции, которые ему придут в голову в процессе анализа.

В качестве мер в трехмерном кубе, изображенном на рис. 2, использованы суммы продаж, а в качестве измерений - время, товар и магазин. Измерения представлены на определенных уровнях группировки: товары группируются по категориям, магазины - по странам, а данные о времени совершения операций - по месяцам. Чуть позже мы рассмотрим уровни группировки (иерархии) подробнее.

Рисунок 7. Пример куба

3.3.1 “Разрезание” куба

Даже трехмерный куб сложно отобразить на экране компьютера так, чтобы были видны значения интересующих мер. Что уж говорить о кубах с количеством измерений, большим трех? Для визуализации данных, хранящихся в кубе, применяются, как правило, привычные двумерные, т. е. табличные, представления, имеющие сложные иерархические заголовки строк и столбцов. Двумерное представление куба можно получить, “разрезав” его поперек одной или нескольких осей (измерений): мы фиксируем значения всех измерений, кроме двух, - и получаем обычную двумерную таблицу. В горизонтальной оси таблицы (заголовки столбцов) представлено одно измерение, в вертикальной (заголовки строк) - другое, а в ячейках таблицы - значения мер. При этом набор мер фактически рассматривается как одно из измерений - мы либо выбираем для показа одну меру (и тогда можем разместить в заголовках строк и столбцов два измерения), либо показываем несколько мер (и тогда одну из осей таблицы займут названия мер, а другую - значения единственного “неразрезанного” измерения).

Взгляните на рис. 8 - здесь изображен двумерный срез куба для одной меры - Unit Sales (продано штук) и двух “неразрезанных” измерений - Store (Магазин) и Время (Time).

Реферат: OLAP технологии

Рисунок 8. Двумерный срез куба для одной меры

На рис. 9 представлено лишь одно “неразрезанное” измерение - Store, но зато здесь отображаются значения нескольких мер - Unit Sales (продано штук), Store Sales (сумма продажи) и Store Cost (расходы магазина).

Реферат: OLAP технологии

Рисунок 9. Двумерный срез куба для нескольких мер

Двумерное представление куба возможно и тогда, когда “неразрезанными” остаются и более двух измерений. При этом на осях среза (строках и столбцах) будут размещены два или более измерений “разрезаемого” куба - см. рис. 10.

Реферат: OLAP технологии

Рисунок 10. Двумерный срез куба с несколькими измерениями на одной оси

3.3.2 Метки

Значения, “откладываемые” вдоль измерений, называются членами или метками (members). Метки используются как для “разрезания” куба, так и для ограничения (фильтрации) выбираемых данных - когда в измерении, остающемся “неразрезанным”, нас интересуют не все значения, а их подмножество, например три города из нескольких десятков. Значения меток отображаются в двумерном представлении куба как заголовки строк и столбцов.

3.3.3 Иерархии и уровни

Метки могут объединяться в иерархии, состоящие из одного или нескольких уровней (levels). Например, метки измерения “Магазин” (Store) естественно объединяются в иерархию с уровнями:

All (Мир)

Country (Страна)

State (Штат)

City (Город)

Store (Магазин).

В соответствии с уровнями иерархии вычисляются агрегатные значения, например объем продаж для USA (уровень “Country”) или для штата California (уровень “State”). В одном измерении можно реализовать более одной иерархии - скажем, для времени: {Год, Квартал, Месяц, День} и {Год, Неделя, День}.

Отметим, что иерархии могут быть сбалансированными (balanced), как, например, иерархия, представленная на рис. 11, а также иерархии, основанные на данных типа "дата—время", и несбалансированными (unbalanced). Типичный пример несбалансированной иерархии — иерархия типа "начальник—подчиненный" (ее можно

построить, например, используя значения поля Salesperson исходного набора данных из рассмотренного выше примера), представлен на рис. 12

Реферат: OLAP технологии

Рисунок 11. Иерархия в измерении, связанном с географическим положением клиентов

Реферат: OLAP технологии

Рисунок 12. Несбалансированная иерархия

Иногда для таких иерархий используется термин Parent-child hierarchy.

Существуют также иерархии, занимающие промежуточное положение между

сбалансированными и несбалансированными (они обозначаются термином ragged —

"неровный"). Обычно они содержат такие члены, логические "родители" которых

находятся не на непосредственно вышестоящем уровне (например, в

географической иерархии есть уровни Country, City и State, но при этом в

наборе данных имеются страны, не имеющие штатов или регионов между уровнями

Country и City; (рис. 13).

Реферат: OLAP технологии

Рисунок 13. "Неровная" иерархия

Отметим, что несбалансированные и "неровные" иерархии поддерживаются далеко не всеми OLAP-средствами. Например, в Microsoft Analysis Services 2000 поддерживаются оба типа иерархии, а в Microsoft OLAP Services 7.0 — только сбалансированные. Различным в разных OLAP-средствах может быть и число уровней иерархии, и максимально допустимое число членов одного уровня, и максимально возможное число самих измерений.

3.3.4 Архитектура OLAP приложений

Все, что говорилось выше про OLAP, по сути, относилось к многомерному представлению данных. То, как данные хранятся, грубо говоря, не волнует ни конечного пользователя, ни разработчиков инструмента, которым клиент пользуется.

Многомерность в OLAP-приложениях может быть разделена на три уровня:

Ø Многомерное представление данных - средства конечного

пользователя, обеспечивающие многомерную визуализацию и манипулирование данными; слой многомерного представления абстрагирован от физической структуры данных и воспринимает данные как многомерные.

Ø Многомерная обработка - средство (язык) формулирования многомерных запросов (традиционный реляционный язык SQL здесь оказывается непригодным) и процессор, умеющий обработать и выполнить такой запрос.

Ø Многомерное хранение - средства физической организации данных, обеспечивающие эффективное выполнение многомерных запросов.

Первые два уровня в обязательном порядке присутствуют во всех OLAP-средствах. Третий уровень, хотя и является широко распространенным, не обязателен, так как данные для многомерного представления могут извлекаться и из обычных реляционных структур; процессор многомерных запросов в этом случае транслирует многомерные запросы в SQL-запросы, которые выполняются реляционной СУБД.

Конкретные OLAP-продукты, как правило, представляют собой либо средство многомерного представления данных, OLAP-клиент (например, Pivot Tables в Excel 2000 фирмы Microsoft или ProClarity фирмы Knosys), либо многомерную серверную СУБД, OLAP-сервер (например, Oracle Express Server или Microsoft OLAP Services).

Слой многомерной обработки обычно бывает встроен в OLAP-клиент и/или в OLAP-сервер, но может быть выделен в чистом виде, как, например, компонент Pivot Table Service фирмы Microsoft.

Заключение

В данной работе мы ознакомились с основами OLAP. Мы узнали следующее:

Назначение хранилищ данных — предоставление пользователям информации для статистического анализа и принятия управленческих решений.

Хранилища данных должны обеспечивать высокую скорость получения данных, возможность получения и сравнения, так называемых срезов данных, а также непротиворечивость, полноту и достоверность данных.

OLAP

(On-Line Analytical Processing) является ключевым компонентом построения и применения хранилищ данных. Эта технология основана на построении многомерных наборов данных — OLAP-кубов, оси которого содержат параметры, а ячейки — зависящие от них агрегатные данные.

· Приложения с OLAP-функциональностью должны предоставлять пользователю результаты анализа за приемлемое время, осуществлять логический и статистический анализ, поддерживать многопользовательский доступ к данным, осуществлять многомерное концептуальное представление данных и иметь возможность обращаться к любой нужной информации.

Кроме того, мы рассмотрели основные принципы логической организации OLAP-кубов, а также узнали основные термины и понятия, применяемые при многомерном

анализе. И наконец, мы выяснили, что представляют собой различные типы иерархий в измерениях OLAP-кубов.

Список использованной литературы

Материалы сайта http://www.softkey.info

Материалы сайта

http://www.iemag.ru

Материалы сайта http://www.compress.ru

Материалы сайта www.olap.ru

Дюк В.А. Обработка данных на ПК в

примерах. — СПб: Питер, 1997.

6 Список иллюстраций

[1] Online Analytical Processing.

[2] Хранилище данных — это интегрированный

накопитель информации, собранной из других систем, на основе которого строятся

процессы принятия решений и анализа данных