

ным уровнем квалификации достоверности бухгалтерской отчетности. Если стороны, подписавшие аудиторское заключение, намеренно искажают истину, находятся в родстве и пользуются влиянием данных связей, руководствуются лишь соображениями «не потерять клиента» или же «не испортить хорошие отношения», то соответственные итоговые материалы проведения аудита могут потерять какую-либо ценность для пользователей отчетности. В таком случае проведение аудита можно считать пустой тратой времени, сил и средств.

В успешной аудиторской фирме должна иметься система мер, которая обеспечивала бы контроль за объективностью и честностью сотрудников. В первую очередь сюда нужно относить документирование аудита и разнообразные внутренние проверки, одной из целей которых является заверение в надежности работы аудитора и отсутствии нежелательного влияния на его деятельность.

Отдельно стоит отметить требование независимости. Рекомендацией в данном случае выступает разработка специального бланка с вопросами, раскрывающего информацию об отсутствии у аудиторов, участвующих в проверке, родственных связей с руководителями аудируемого лица, финансового интереса в проверяемой организации, фактов, подтверждающих участие в подготовке отчетности, которую необходимо проверить в ходе аудита и т.д.

Литература:

1. Аудит: Учебник для бакалавров / А.Е. Суглобов, Б.Т. Жарылгасова, Е.Е. Коба, И.П. Драчёна; под ред. д. э. н., проф. А.Е. Суглобова.- 2-е изд. – М.: Издательско-торговая корпорация «Дашков и К°», 2017.
2. Овсяичук В.Я., Кондратьев К.М., Драчёна И.П. Внутренний аудит Учебное пособие / Москва, 2004.
3. Суглобов А.Е., Драчена И.П., Музалев С.В. Методологические и концептуальные основы стандартизации аудиторской деятельности в России Аудитор. 2014. № 12 (238). С. 3-14.
4. Касьянова С.А. Аудит: Учебное пособие – М.: Вузовский учебник, НИЦ ИНФРА-М, 2016.

МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СТРУКТУРЫ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ НА БАЗЕ OLAP-СИСТЕМЫ MODELING OF THE INFORMATION STRUCTURE OF THE MANAGEMENT OF THE ENTERPRISE ON THE BASIS OF THE OLAP SYSTEM

В.Г. Дрозд, канд. экон. наук, доцент,

Б.Ж. Спанова, канд. экон. наук, доцент,

Карагандинский экономический университет Казпотребсоюза

Аннотация. Авторами статьи рассматривается информационная инфраструктура предприятия, которая представлена в виде иерархических уровней – «аналитической пирамиды». Среди классов информационных систем, расположенных на

разных уровнях пирамиды, рассматриваются транзакционные системы, хранилища данных и алгоритм реализации аналитические приложения OLAP-системы.

Abstract. *The authors of the article consider the information infrastructure of the enterprise, which is represented in the form of hierarchical levels – the «analytical pyramid». Among classes of information systems located at different levels of the pyramid, transactional systems, data warehousing and implementation algorithms are considered analytical applications of the OLAP system.*

Ключевые слова / keywords: *управляющее решение, многомерные данные, OLAP системы, структура информационной системы, аналитическая пирамида, хранилище данных, база данных / control solution, multidimensional data, OLAP system, information system structure, analytical pyramid, data storage, database.*

Современные условия организации и ведения бизнеса предъявляют высокие требования к действующим системам управления. В условиях ужесточения конкуренции и активно меняющейся внешней среды, все более значительную роль приобретают информационно-аналитические методы и модели, дающие возможность оперативно реагировать на проблемы и возможные перспективы. Задачи бизнес-анализа имеют большие сложности и здесь на помощь руководителю предприятия приходят имеющиеся современные управленческие концепции и технологии.

За долгий период своего развития теория и практика управления создали множество подходов, а также методов и моделей, которые нацелены на повышение эффективности. В свою очередь, эти методы и модели, способствовали появлению и развитию многих аналитических информационных систем. Подобные управленческие системы занимают в информационной инфраструктуре предприятия определенное положение и играют важную роль в управленческих процессах.

Информационную инфраструктуру предприятия возможно представить в виде ряда иерархических уровней, где каждый уровень будет характеризоваться своей ролью в процессе управления и степенью агрегированности информации.

Реализуя процесс построения многомерной модели, следует прежде отметить следующее:

- витрины данных и многомерная модель ориентированы не предприятия в целом, а на отдельные подразделения этого предприятия;
- невозможно интегрировать витрины данных и многомерную модель в целом.
- витрины данных и многомерная модель предназначены только для хранения агрегированных данных;

Сам же процесс проектирования многомерной модели состоит из четырех основных этапов:

1. Выстраивание для модели вектора бизнес-процесса $\{M_1, M_2, \dots, M_p\}$, где элементами выборки выступают количественные атрибуты транзакции сущностей или их агрегированные данные.

2. Уточнение составляющих для каждого бизнеса-процесса, где элементами выборки являются остальные атрибуты транзакции сущностей, которые будут являться высшими уровнями иерархии класса.

3. В соответствии с потребностями аналитиков дается определение всех классов иерархии.

4. Проводится определение соответствующих атрибутов фактам.

В качестве примера схематического представления информационной инфраструктуры можно привести так называемую аналитическую пирамиду (analytical stack), разработанную компанией Gartner. В этой иерархии прослеживаются несколько уровней:

- систем бизнес-интеллекта, включая хранилища данных, витрины данных и OLAP-системы;

- транзакционных систем;

- аналитических приложений.

В основании построенной аналитической пирамиды лежат транзакционные системы. При продвижении к вершине пирамиды от ее основания, идет медленное преобразование в агрегированную информацию имеющихся детальных операционных данных. Эта информация предназначена для поддержки принятия управленческих решений[1].

Соотнести тот или иной программный продукт к какому-либо одному конкретному классу не всегда будет возможно, в связи с тем, что многие системы дают возможность выполнить решение аналитических задач нескольких категорий.

Следует отметить, что целью применения OLAP является проведение анализа данных и представление результатов этого анализа в удобном для восприятия и принятия решений виде. Определяющая цель OLAP заключается в создании многомерных кубов, данные в которых будут доступны для многопользовательских запросов. Исходные данные, необходимые для построения OLAP-кубов обычно хранятся в хранилищах данных.

Реализуемая OLAP-функциональность может быть получена различными способами, начиная с простейших инструментальных средств анализа данных имеющихся в офисных приложениях и заканчивая сложными и специализированными распределенными аналитическими системами, основанными на серверных продуктах [2].

Механизм OLAP становится на сегодня одним из популярнейших методов для проведения анализа данных. Существует два основных подхода для решения данной задачи. Первый подход называется Multidimensional OLAP (MOLAP), т.е. реализация механизма при помощи многомерной базы данных на стороне сервера, а второй подход – Relational OLAP (ROLAP) – построение кубов «на лету», т.е. на основе SQL запросов к реляционной СУБД. Вполне понятно, что каждый из этих подходов имеет свои плюсы и минусы.

Рассмотрим реализацию ядра настольного ROLAP модуля.

Общую схему работы настольной OLAP системы можно представить следующим образом (рис. 1).

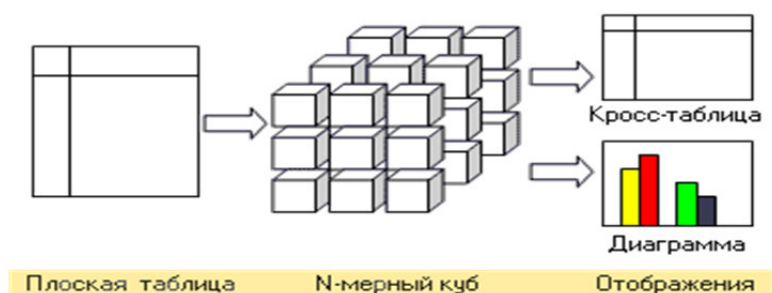


Рисунок 1 – Схема работы настольного ROLAP модуля

Алгоритм работы будет следующим:

- Проводится сбор данных в виде плоской таблицы или результата выполнения SQL запроса.
- Проводится кэширование данных и ведется их преобразование к многомерному кубу.
- Проводится отображение построенного куба при помощи кросс-таблицы или диаграммы и т.п. В общем случае получаем, что к одному кубу может быть подключено произвольное количество отображений.

Чаще всего многомерные модели отображаются в виде куба с многими измерениями [3]. Такое хранилище должно обладать следующими свойствами:

- продвижение снизу-вверх и возможность эволюции;
- ориентация на бизнес – процессы;
- интеграция, которая достигается посредством соответствия измерений;
- преимущество многомерной модели, а не реляционной;
- применение схемы «звезда» (или «снежинка»), что обеспечивает семантику запросов.

Основным компонентом для реляционной модели данных выступает сущность, и при этом первым шагом при подготовке многомерной модели выступает необходимость проведения классификации сущностей по категориям:

1. Сущность транзакции – это та сущность, где проводится сбор информация о конкретных событиях происходящих в бизнесе. Например, заказ, продажа и т.д. Определяющими характеристиками данной сущности являются:

Здесь описываются события происходящие в конкретный, определенный момент времени.

Данная сущность содержит количества или значения, которые обобщаются, т.е. например количество товаров, количество продаж и т.д. Эти сущности являются наиболее определяющими объектами находясь в хранилище данных, а также служат базовой основой при построении таблиц фактов в схеме звезда. Но не все эти сущности будут представлять интерес

при принятии решений. Только пользователи хранилищ данных будут определять, какие операции являются необходимыми.

2. Составные сущности – это те сущности, которые напрямую связаны с сущностями транзакций по средствам отношения «один–ко–многим». Эти сущности характеризуют компоненты или детали каждой транзакции.

К примеру, сущность продажи может быть определена рядом компонентов: Товар, Клиент, Время, Место.

3. Сущности «классификация» – это те сущности, имеющие отношение к составным сущностям по типу «один–ко–многим». Здесь присутствует их функциональная зависимость от составных сущностей (транзитивно или непосредственно). Т.е. присутствует иерархия, встроенная в модель данных, которые могут быть свернуты в составные компоненты в форме таблиц измерений в схеме «звезда». К примеру, сущность типа товара, типа клиента.

Рассмотрим как данного вида система может быть устроена внутри. Начнем с той стороны, которую можно посмотреть, т.е. с отображений.

Отображения, которые применяются в OLAP системах, чаще всего делятся на два вида – диаграммы и кросс-таблицы. Проведем рассмотрение вида – кросс-таблица. Этот вид является основным и наиболее распространенным способом для отображения куба.

На изображенном ниже рисунке, отображена кросс-таблица, где строки и столбцы отображены желтым цветом. Они содержат агрегированные результаты. Ячейки, в которые попадают факты, отмечены светло-серым цветом и ячейки, содержащие данные размерностей, выделены темно-серым цветом (рис. 2).

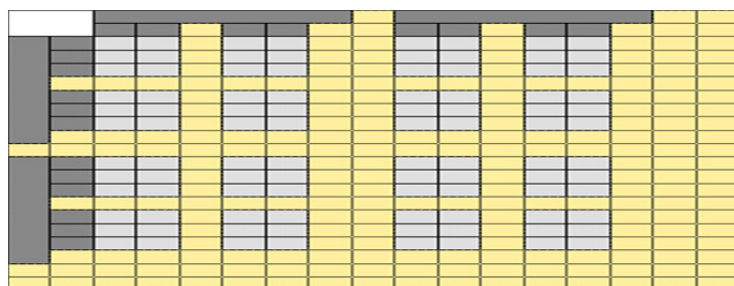
A grid representing a cross-table. The grid is 10 columns wide and 10 rows high. The first row and the first column are highlighted in yellow. The remaining cells in the grid are shaded in light grey. This visualizes the structure of a cross-table where the dimensions are represented by the yellow rows and columns, and the data points are in the grey cells.

Рисунок 2 – Кросс таблица

Таким образом, таблицу возможно поделить на следующие элементы (рис. 3).

Делая заполнение матрицы с фактами, следует придерживаться следующего правила:

- На базе имеющихся данных об измерениях определяются координаты добавляемого элемента в матрице.

- Определяются координаты для столбцов и строк итогов, на которые будет влиять добавляемый элемент.

- Добавляется элемент в матрицу и соответствующие столбцы и строки итогов.

При этом необходимо отметить, что полученная матрица будет довольно сильно разрежена, поэтому ее представление в виде двумерного массива (вариант, лежащий на поверхности) невозможна в связи с достаточно большой размерностью этой матрицы, что говорит о том, что для хранения такой матрицы не хватит никакого объема оперативной памяти [4].

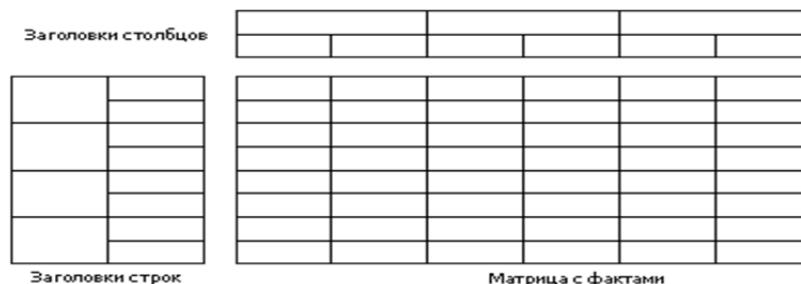


Рисунок 3 – Таблица с разделенными элементами

Такие данные, находящиеся в таблице требуется преобразовать для их использования. Поэтому, с целью повышения вычислительной производительности при построении гиперкуба, рекомендуется подбирать уникальные элементы, которые хранятся в столбцах, являющихся измерениями куба. Кроме этого, следует выполнять предварительное агрегирование фактов для записей, обладающих одинаковыми значениями размерностей. Как уже было отмечено ранее, важное значение имеют уникальные значения, находящиеся в полях измерений. Тогда для их хранения возможно создать следующую структуру (рис. 4):

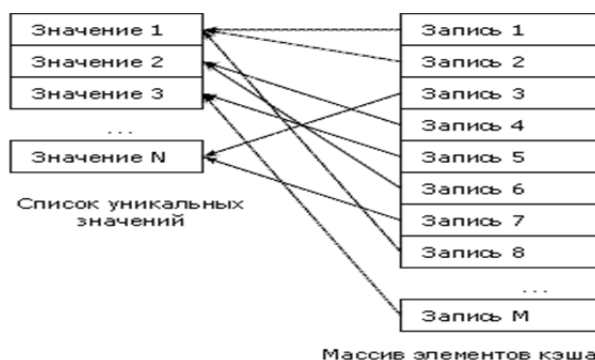


Рисунок 4 – Структура базы данных

При применении подобной структуры значительно снижается потребность в памяти. Это весьма актуально, т.к. для увеличения оперативности вычислительной работы следует хранить данные в оперативной памяти. Кроме этого, хранить можно только массив элементов, а их значения выгружать на диск, так как они будут требоваться только при отображении кросс-таблицы.

Рассмотрим библиотеку компонентов CubeBase и следуя вышеизложенному создаем библиотеки компонентов CubeBase.

Компонент TCubeSource осуществляет преобразование и кэширование данных во внутренний формат, а также предварительное агрегирование данных. Компонент TCubeEngine проводит вычисление гиперкуба и операции с ним. Т.е. фактически, он является OLAP-машиной, выполняющей преобразование в многомерный набор данных имеющейся плоской таблицы. Компонент TCubeGrid осуществляет вывод на экран крест-таблицы и управление отображением гиперкуба. TCubeChart позволяет увидеть гиперкуб в виде графиков, а компонент TCubePivote осуществляет управление работой ядра куба.

Следует выполнить загрузку данных в гиперкуб. Для этого, на первом этапе работы системы будет проведена загрузка данных и преобразование их во внутренний формат.

С целью оптимизации работы гиперкуба следует определить то, какие именно задачи необходимо решать в первоочередном порядке, и по каким критериям требуется добиваться повышения качества работы. Важным является повышение скорости работы программы, и при этом желательно, чтобы не требовался большой объем оперативной памяти. Увеличение быстродействия возможно за счет введения дополнительных механизмов доступа к данным, например, введение индексирования. Однако, это повышает накладные расходы оперативной памяти.

Литература:

1. Миронов А.А., Мордвинов В.А., Скуратов А.К. Семантико-энтропийное управление OLAP и модели интеграции xOLAP в SemanticNET (ONTONET). Информатизация образования и науки №2, 2009. С. 21-30.
2. Ларри Урман. MySQL. Перевод с английского Слинкина А.А. ДМК Пресс СПб Питер 2004 г.
3. Дэн Тоу Настройка SQL. Для профессионалов. СПб. Питер 2004 г.
4. Кудрявцев Ю.А. OLAP технологии: обзор решаемых задач и исследований // Бизнес-информатика. – 2008. №1. – С. 66-70.
5. Аткинсон, Леон. MySQL. Библиотека профессионала.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002.

АВТОМАТИЗАЦИИ БУХГАЛТЕРСКОГО УЧЕТА С ИСПОЛЬЗОВАНИЕМ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ AUTOMATION OF ACCOUNTING WITH THE USE OF CLOUD CALCULATIONS

В.Г. Дрозд, канд. экон. наук, доцент,

Б.Ж. Спанова, канд. экон. наук, доцент,

Карагандинский экономический университет Казпотребсоюза

Аннотация. *Проводится рассмотрение технологии распределённой обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользовате-*