

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

УДК 004.042

МОДЕЛЬ ДАННЫХ НА ОСНОВЕ РАСШИРЯЕМОЙ СТРУКТУРЫ ДЛЯ ИНФОРМАЦИОННЫХ СИСТЕМ, В УСЛОВИЯХ ДИНАМИЧЕСКОГО ВНЕШНЕГО ОКРУЖЕНИЯ

В. И. Орешков, к.т.н., доцент кафедры САПР ВС РГРТУ, Рязань, Россия;
orcid.org/0000-0003-0316-4927, e-mail: vyacheslav.oreshkov@yandex.ru

И. А. Цепулин, программист кафедры САПР ВС РГРТУ, Рязань, Россия;
orcid.org/0000-0002-6417-630X, e-mail: tsepylin@mailru

Рассматривается расширяемая модель данных для структур хранения информационных систем, функционирующих в условиях динамичной внешней среды. Целью работы является разработка модели данных, которая способна отражать частые и значительные изменения в предметной области с помощью незначительных и простых изменений в системах хранения, а также обеспечивать скорость выполнения сложных аналитических запросов в масштабе времени, близком к реальному. В основе предлагаемой модели лежит идея декомпозиции сложной схемы данных на множество более простых компонентов, которыми проще манипулировать. В частности, её эволюция происходит не через изменение существующих структур данных, а через создание новых (расширение). Гибкость и модульность структур данных достигаются за счёт размещения бизнес-ключей в таблицах атрибутов.

Ключевые слова: информационная система, аналитический запрос, хранилище данных, базы данных, модель данных, предметная область, бизнес-ключ.

DOI: 10.21667/1995-4565-2020-74-56-64

Введение

Одной из основных функций информационных систем (ИС) предприятия в настоящее время является поддержка процессов принятия управленческих решений. Как правило, эта функция связана с выполнением сложных аналитических запросов к системам хранения данных предприятия (базам, хранилищам и витринам данных), которые должны производиться в масштабе времени, близком к реальному [2, 3].

ИС, работающие в динамичном бизнес-окружении, подвержены изменениям, которые могут происходить быстрее или медленнее, но происходят постоянно. Также меняются и цели использования данных в таких системах – отчётность, анализ данных, поддержка принятия решений и т.д. Могут меняться применяемые алгоритмы и методы обработки данных, их модели и структуры. Поэтому модели данных в системах хранения ИС должны быть модульными, гибкими и отслеживать изменения в обрабатываемых данных, связанные с изменениями во внешнем окружении организации.

Особенностью современных ИС является необходимость оперативной доставки данных потребителю в виде отчётов, сформированных по результатам сложных аналитических запросов, а также обеспечение накопления и хранения больших и сверхбольших объёмов данных. Основная проблема при решении данных задач заключается в том, что требования к объёму хранения данных и скорости выполнения запросов противоречивы. Реляционные модели обеспечивают ком-

пактное хранение, но из-за необходимости денормализации данных в процессе выполнения запросов не обеспечивают предоставление информации в реальном масштабе времени. Многомерные модели, не использующие нормализацию и позволяющие быстро выполнять сложные запросы, в общем случае не оптимальны с точки зрения использования памяти [4, 8].

Еще одной проблемой является необходимость «встраивания» в модель новых структур данных, отражающих изменения в предметной области. Для классических моделей типа «звезда» и «снежинка» процесс модификации структуры является весьма длительным и трудоёмким.

Поэтому разработчики ИС, в которых активно используется аналитическая обработка данных для целей поддержки принятия решений, находятся в постоянном поиске структур и моделей систем хранения данных, которые обеспечивали компромисс между эффективностью использования памяти, скоростью обслуживания и простотой модификации структур.

Одним из направлений такого поиска является выявление особенностей самих данных, регламента и технологий их сбора и накопления, а также целей использования. Действительно, некоторые данные меняются медленно или не изменяются совсем, а другие подвержены постоянным и быстрым изменениям. Сами изменения данных могут быть структурными, когда изменяется сама структура данных (добавляются или исключаются измерения, атрибуты и сущности), или контентными (когда модифицируется содержание данных без изменения структуры). Кроме этого, данные могут носить «оперативный» характер, отражая незначительные, краткосрочные изменения во внешнем бизнес-окружении, и «долговременный» – представляя длительные, медленно изменяющиеся тенденции и закономерности [7].

Таким образом, при построении структур хранения данных кроме инфологической модели, учитывающей в основном особенности предметной области, необходимо использовать «техническую» модель, учитывающую особенности их накопления, хранения и изменения. В данной статье рассматривается один из подходов к оптимизации структуры хранения данных в системе поддержки принятия решений (СППР) на основе «анкерной» модели.

Теоретическая часть

В основе модели, предлагаемой в данной статье, лежит предположение, что изменения в данных требуют только расширения, а не модификации существующих структур хранения. Это предположение соответствует принципу неизменчивости, реализуемому в хранилищах данных, – в любой момент времени на один и тот же запрос должен быть получен одинаковый отклик. Следовательно, все существующие версии схемы данных являются подмножествами текущей версии.

В этой связи можно разделить жизненный цикл системы на две фазы: формирование и эволюцию. На первой фазе создаётся модель данных, максимально учитывающая все сущности и зависимости предметной области в её текущем состоянии. В процессе эволюции создаются новые схемы, присоединяемые к начальной и учитывающие изменения в бизнес-окружении.

Эволюция модели данных через расширения, а не через модификацию, приводит к модульности, что позволяет разбивать модели данных на небольшие, стабильные и управляемые элементы [7, 8]. Если изменения в данных часто повторяются, то ранее выполненные расширения могут использоваться повторно. Таким образом даже значительные изменения в бизнес-окружении могут быть отражены сравнительно небольшими изменениями в структурах хранения данных.

Основными компонентами предлагаемой модели данных являются «анкер» (по аналогии с крепёжным изделием, фиксирующим конструкцию на неподвижном носителе), узел, атрибут и связь, представленные на рисунке 1.

Анкер представляет собой множество сущностей, представляющих бизнес-объекты или события во внешнем окружении. Узлы используются для представления статичных, обычно небольших наборов сущностей, которые не изменяются с течением времени. В то время как

анкеры используются для представления произвольных сущностей, узлы используются для управления свойствами, которые разделяемы по многим экземплярам некоторого анкера.

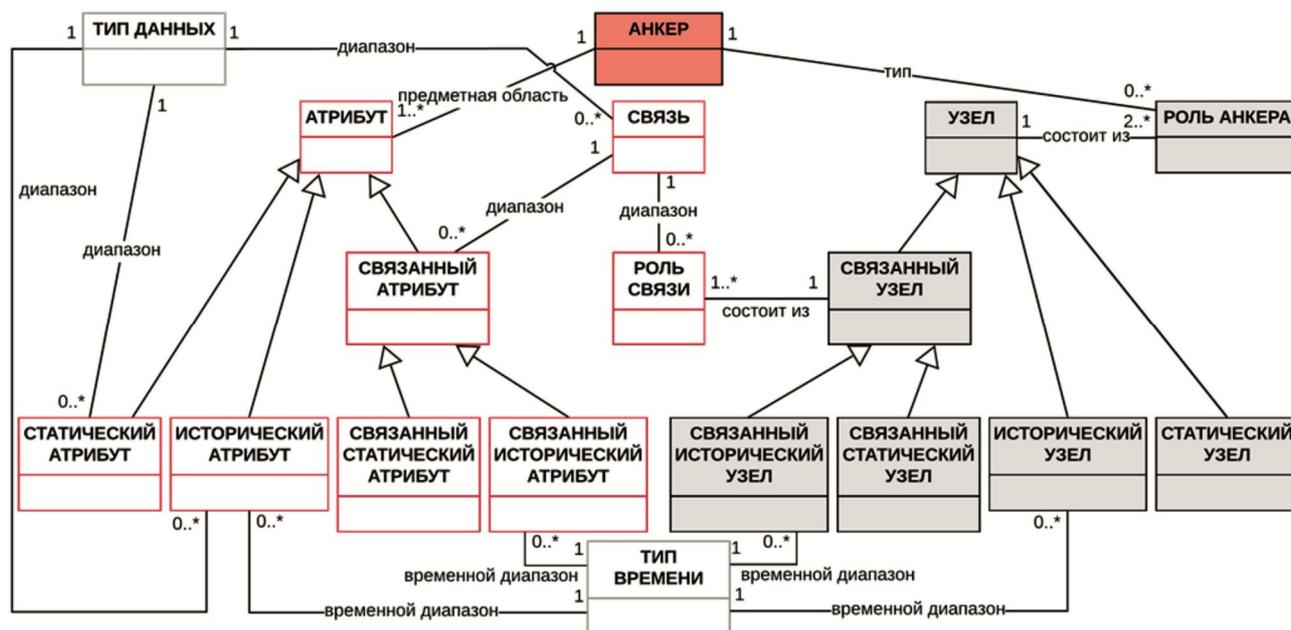


Рисунок 1 – Основные компоненты метода анкерной модели данных
Figure 1 – Main components of anchor modeling

Например, свойство «Пол» может быть представлено узлом GEN_Gender, который включает два значения: «Male» и «Female». Это свойство разделяется многими экземплярами анкера AC_Employee, поэтому использование узла сводит к минимуму избыточность: вместо повторения строк оказывается достаточно одного значения на один экземпляр.

Атрибуты используются для представления свойств анкеров. В модели выделяется четыре типа атрибутов: статические (static), исторические (historized), связанные статические (knotted static) и связанные исторические (knotted historized). Статические атрибуты используются для представления свойств сущностей, когда не требуется сохранять историю изменений значений атрибута. Примером статического атрибута может быть дата рождения.

Историческими являются атрибуты, изменения значений которых должны сохраняться (например, возраст, рост, цена и так далее). Связанные статические атрибуты используются для представления связи между анкерами и узлами, то есть для связи со свойствами, которые могут принимать только фиксированные, обычно небольшие количества значений. И, наконец, связанные исторические атрибуты используются, когда связь со значением узла слабая, но может изменяться во времени.

Связи представляют отношения между двумя или более сущностями в анкерах и необязательными сущностями узла. Аналогично атрибутам связи могут быть четырех типов – статические, исторические, связанные статические и связанные исторические. Поскольку один и тот же объект может появляться в связи более одного раза, вхождения должны быть идентифицированы с помощью понятия роли – идентификатора, имеющего тот же тип, что и анкер или узел, к которому она относится.

Узлы графа анкерной модели представлены на рисунке 2.

Что касается имен элементов схемы данных, то никаких формальных ограничений на имена объектов в схеме данных анкерной модели нет. Но наличие некоторого неформального соглашения об именах с описанием того, каким требованиям они должны удовлетворять, делает схему более понятной и упрощает работу с моделью данных. Хорошее соглашение об именах должно соответствовать ряду критериев, некоторые из которых могут оказаться противоречивыми. С одной стороны, имена должны быть короткими, но при этом хорошо ин-

терпретируемыми. Они должны быть уникальными и недвусмысленными, но иметь общие черты с соответствующими объектами.

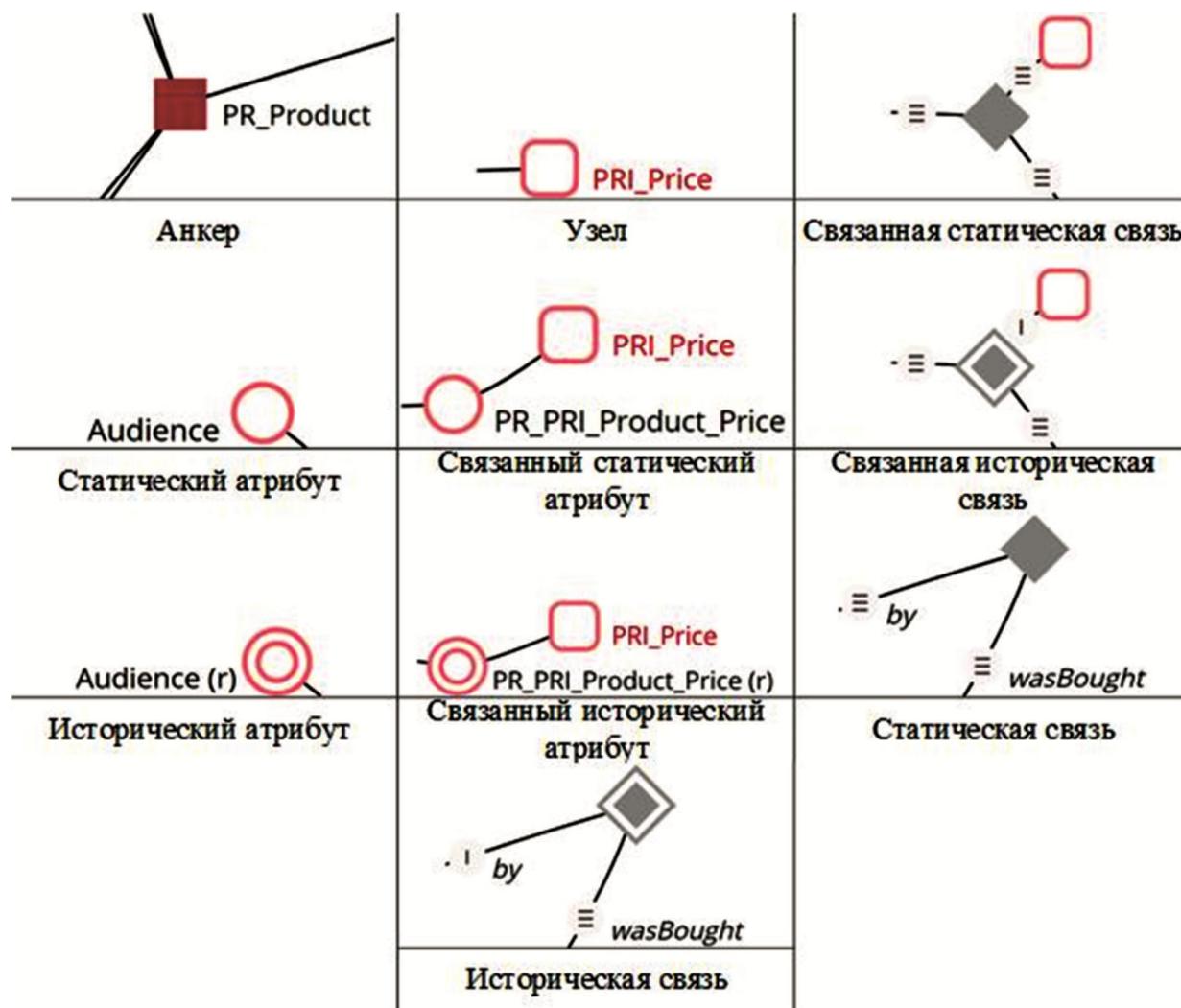


Рисунок 2 – Узлы графа анкерной модели
Figure 2 – The graph nodes of anchor model

Возможный пример соглашения об именах представлен в таблице 1.

Таблица 1 – Соглашение об именах

Table 1 – Naming agreement

Сущность	Мнемоника/шаблон	Дескриптор/шаблон	Имя
Анкер	$A_m [A-Z]\{2\}$	$A_d ([A-Z][a-z]^*)+$	$A_m_A_d$
Узел	$K_m [A-Z]\{3\}$	$K_d ([A-Z][a-z]^*)+$	$K_m_K_d$
Атрибут	$B_m [A-Z]\{3\}$	$B_d ([A-Z][a-z]^*)+$	$A_m_B_m_A_d_B_d$
Роль		$r ([a-z][A-Z]^*)+$	r
Связь			$A_m_r_..._(K_m_r)$

Соглашение предполагает использование только латинских букв, цифр и символа подчеркивания. Это гарантирует, что одни и те же имена могут использоваться во множестве представлений, например в реляционной базе данных. В предлагаемом соглашении об именах используется синтаксис с несколькими семантическими правилами, позволяющий связывать сущность с ее именем. Синтаксис использует три примитива: мнемоник, дескриптор и роль. С использованием комбинаций этих примитивов создаются имена элементов схемы.

Мнемоники анкеров и узлов должны быть уникальны в модели, в то время как мнемоники атрибутов должны быть уникальными в наборе атрибутов, ссылающихся на один и тот же анкер. Названия связей построены из мнемоники смежных анкеров и узлов вместе с ролями, которые они играют в отношениях. Мнемоники анкеров содержат две заглавных буквы, мнемоники остальных атрибутов и узлов – 3 буквы. В таблице представлены следующие правила (все элементы имени разделяются подчёркиваниями):

– имя анкера строится из мнемоники и дескриптора анкер, например, PR_Product (PR – мнемоника, Product – дескриптор);

– имя узла строится из мнемоники и дескриптора узла, например PRI_Price (PRI – мнемоника, Price – дескриптор, «Цена» – свойство бизнес-сущности «Товар»);

– имя атрибута строится в виде последовательности мнемоник и дескрипторов анкера и узла, например, PR_PRI_Product_Price (интерпретировать такую запись можно как «анкер Продукт имеет атрибут Цена»);

– дескриптор роли строится на основе действия, которое выполняется над анкером или атрибутом, указывается курсивом, например, wasPay – был куплен;

– дескриптор связи формируется как список анкеров, между которыми эта связь устанавливает отношения. После каждого анкера указывается его роль.

Таким образом, в основе предлагаемого метода лежит идея декомпозиции сложной схемы данных на множество более простых компонентов, которыми проще манипулировать при необходимости изменить структуру данных или сами данные. При этом анкер включает наиболее важные с точки зрения предметной области сущности. Если традиционные реляционные таблицы хранят как суррогатный (технический), так и естественный (бизнес-ключ) ключи, анкер содержит только суррогатный ключ, а бизнес-ключи содержатся в таблице атрибутов.

Данный подход обеспечивает дополнительную гибкость, особенно когда бизнес-ключ извлекается из системы оперативной обработки данных. Такие ключи, как правило, имеют короткий срок жизни по сравнению с бизнес-ключами, взятыми из корпоративной среды. Сохранения бизнес-ключа за пределами анкера позволяет разработчику не задумываться о том, как будет изменяться состояние ключа в будущем. Добавление нового бизнес-ключа в анкерной модели приводит всего лишь к появлению новой таблицы атрибутов.

Атрибут в анкерной модели содержит контекстные (описательные) данные, которые по своей природе зависят от времени. Однако, чтобы сохранить временную целостность в шестой нормальной форме, в анкерной модели каждый атрибут помещается в отдельную таблицу. Анкерная модель поддерживает либо временной (исторический) атрибут, либо статический, для которого не определяется период актуальности.

Связь в анкерной модели представляет отношения между анкерами.

Экспериментальные исследования

Рассмотрим пример практического построения анкерной модели. Выделим три бизнес-объекта: Товар, Категория и Клиент. При этом свойства товара будут Цена и Количество (исторические атрибуты), свойство клиента – Имя клиента (статический атрибут), свойство категории – Имя категории. На основе бизнес объектов, в соответствии с неформальным соглашением об именах в таблице 1, формируем анкеры PR_Product, CA_Category и CU_Customer.

Наличие таких свойств, как Цена, Количество, Имя товара и Имя клиента и категории, отражается узлами:

- цена товара – PRI_Price;
- количество товара – AMT_Amount;
- имя товара – PNM_Pname;
- имя клиента – NAM_Name;
- имя категории – CAT_Name.

Тогда атрибуты узлов будут:

- PR_PRI_Product_Price – цена товара;
- PR_AMT_Product_Amount – количество товара;
- PR_PNM_Product_Pname – имя товара;
- CU_NAM_Customer_Name – имя клиента;
- CA_CAT_Category_Name – имя категории.

Теперь рассмотрим отношения между анкерами, которые представляются с помощью связей (tie). Объект Товар связан с обоими другими анкерами по следующим правилам:

- товар относится к категории PR_in_CA_refersTo;
- товар приобретается клиентом PR_by_CU_wasBought;
- клиент приобретает товар CU_to_PR_wasBought.

Таким образом, основные объекты предметной области должны быть представлены в виде анкеров. Свойства объектов моделируются как атрибуты при соответствующем анкере. Отношения между объектами моделируются как связи. Свойства отношения представляются как узлы или как атрибуты при анкерах, связанных с помощью связей.

Построенный граф анкерной модели представлен на рисунке 3.

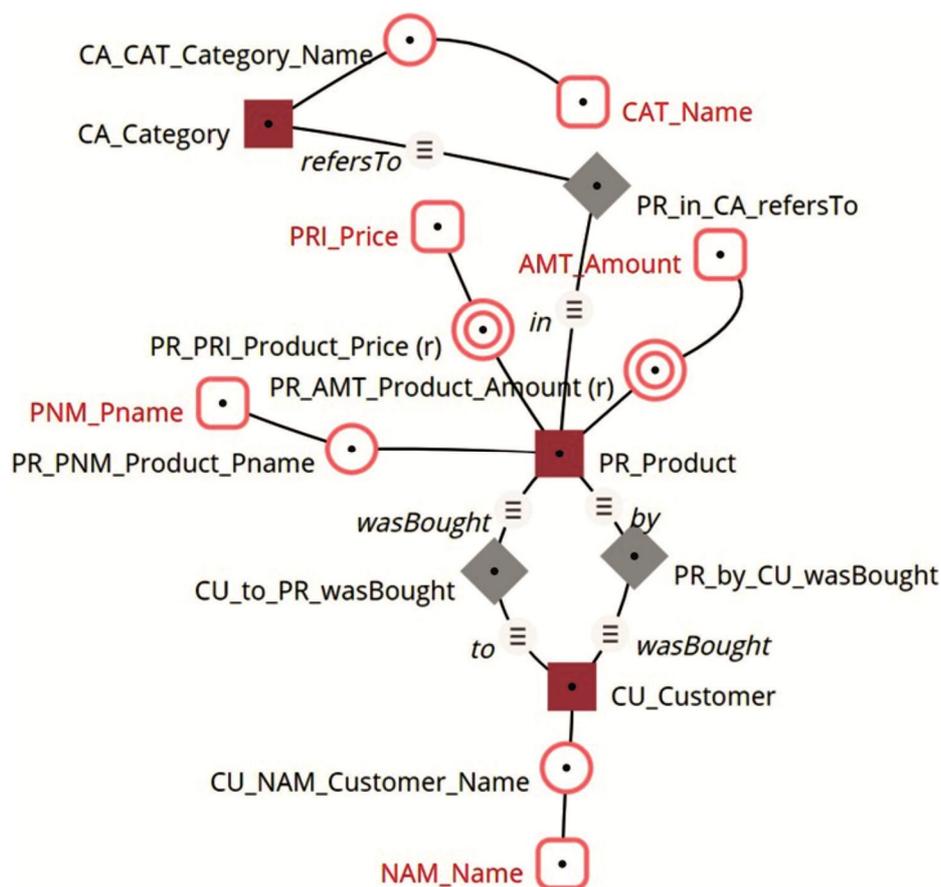


Рисунок 3 – Общая схема анкерной модели
Figure 3 – General scheme of anchor model

Формализованная нотация анкерной схемы позволяет автоматически преобразовывать её в схему реляционной базы данных с помощью некоторого набора преобразующих правил. Отношения в полученной схеме будут находиться в шестой нормальной форме.

Каждый анкер преобразуется в таблицу, содержащую единственный столбец – первичный ключ. Имя таблицы формируется из имени анкера и имени ключевого поля (с префиксом имени анкера и суффиксом ID). Например, для анкера клиента PR_Product имя таблицы будет (PR_ID). Узел трансформируется в таблицу из двух столбцов. Один столбец содержит значения узла, а второй столбец является первичным и внешним ключом.

Атрибут преобразуется в таблицу, содержащую ключевое поле и поле значений. Если атрибут исторический, то в таблицу добавляется поле интервала актуальности значения.

Первичный ключ в таблицах атрибутов ссылается на ключевое поле в таблице анкера, к которому относится атрибут. Если статический атрибут связан с узлом (knotted attribute), то второе поле в нём будет внешним ключом, ссылающимся на первичный ключ узла.

И, наконец, связи преобразуются в таблицы с первичными и внешними ключами анкеров, на которые ссылается связь. Число столбцов в таблицах связей соответствует числу связываемых с помощью данной связи анкеров и их ролей.

Полные результаты преобразования схемы анкерной модели в схему базы данных представлены в таблице 2.

Таблица 2 – Схема базы данных
Table 2 – Database schema

Схема анкеров	Поля таблиц
PR_Product	(PR_ID*)
CA_Category	(CA_ID*)
CU_Customer	(CU_ID*)
Схема узлов	Поля таблиц
PRI_Price	(PRI_ID*, PRI_Price)
PNM_Pname	(PNM_ID*, PNM_Pname)
AMT_Amount	(AMT_ID*, AMT_Amount)
CAT_Name	(CAT_ID*, CAT_Name)
Схема атрибутов	Поля таблиц
PR_PRI_Product_Price	(PR_ID*, PRI_ID,Product_Price)
PR_PNM_Product_Name	(PR_ID*, PNM_ID,Product_Name)
CU_NAM_Customer_Name	(CU_ID*, CU_NAM_Name_Customer)
CA_CNM_Category_Name	(CA_ID*, CA_CNM_Name_Category)
PR_AMT_Amount_Product	(PR_ID*, PR_AMT_Amount_Product,PR_AMT_ValidForm*)
Схема связей	Поля таблиц
PR_in_CA_RefersTo	(PR_ID_in*, CA_ID_RefersTo*)
CU_to_PR_wasBought	(CU_ID_to*, PR_ID_wasBought*)
PR_by_CU_wasBought	(PR_ID_in*, CU_ID_wasBought*)

В процессе экспериментальных исследований оценивалось время выполнения однотипных аналитических запросов к кроссплатформенному хранилищу данных Interbase/Fairbird. Было установлено, что время выполнения одинаковых запросов на одной и той же аппаратной платформе при использовании анкерной модели в среднем на 18 % меньше, чем для модели «звезда» и на 22,5 %, чем для модели «снежинка». Кроме этого, время встраивания в схему данных нового измерения в анкерную модель происходит на 7,5 % быстрее, чем в модель на основе схемы «звезда» и на 9,2 %, чем в схему снежинка. Таким образом, применение анкерной модели позволит значительно сократить время выполнения запросов и повысить эффективность работы персонала.

Заключение

Таким образом, в статье рассмотрена модель данных, предназначенная для применения в системе хранения ИС, использующей сложные аналитические запросы в масштабе времени, близком к реальному. В основе модели лежит комбинирование инфологической и технической составляющих. Первая учитывает семантику предметной области, а вторая – технические свойства данных. Использование формальных соглашений об именах позволяет сделать модель более понятной и интерпретируемой. Использование бизнес-ключей в таблицах атрибутов обеспечивает гибкость при отражении изменений в предметной области. Применение рассматриваемой модели данных позволяет примерно на 20 % сократить время выпол-

нения аналитических запросов относительно классических схем типа «звезда» и «снежинка» и на 7-9 % ускорить встраивание новых структур в схему данных.

Библиографический список

1. **Архипенков С. Я., Голубев Д., Максименко О.** Хранилища данных. От концепции до внедрения. М.: Диалог-МИФИ, 2002. 528 с. ISBN 5-86404-167-х.
2. **Ноженкова Л. Ф., Шайдунов В. В.** OLAP-технологии оперативной информационно-аналитической поддержки организационного управления // Информационные технологии и вычислительные системы. 2010. № 2. С. 15-27.
3. **Паклин Н. Б., Орешков В. И.** Бизнес-аналитика: от данных к знаниям (+ CD): учеб. пособие. 2-е изд., испр. СПб.: Питер, 2013. 704 с.: ил. ISBN 978-5-459-00717-6.
4. **Орешков В. И.** Хранилища данных и OLAP-технологии: учеб. пособие / Рязан. гос. радиотехн. ун-т. Рязань, 2017. 64 с.
5. **Спирли Э.** Корпоративные хранилища данных. Планирование, разработка, реализация. М.: Издательский дом «Вильямс». Том I. 2001. С. 400. ISBN 5-8459-0191-X.
6. **Adamson, Christopher.** Star Schema The Complete Reference. McGraw-Hill Education, 2010. P. 510. – ISBN 9780071744331.
7. **Rönnbäck L.** Anchor Modeling. Agile Information Modeling in Evolving Data Environments / L. Rönnbäck, O. Regardt, M. Bergholtz, P. Johannesson, P. Wohed – URL: <http://www.anchormodeling.com/wp-content/uploads/2011/05/Anchor-Modeling.pdf> (Preprint submitted to DKE 05.10.2010).
8. **Thomsen Erik.** OLAP Solutions: Building Multidimensional Information Systems. 2nd ed. Wiley Publishing, 2002. P. 696.
9. **Wrembel Robert.** Data Warehouses and OLAP: Concepts, Architectures and Solutions. IGI Global, 2006. P. 332.

UDC 004.042

DATA MODEL BASED ON EXTENSIBLE STRUCTURE FOR INFORMATION SYSTEMS OPERATING IN DYNAMIC EXTERNAL ENVIRONMENT

V. I. Oreshkov, Ph.D. (Tech.), associate professor, CAD department, RSREU, Ryazan, Russia; orcid.org/0000-0003-0316-4927, e-mail: vyacheslav.oreshkov@yandex.ru

I. A. Cepulin, software developer, CAD department, RSREU, Ryazan, Russia; orcid.org/0000-0002-6417-630X, e-mail: tsepylin@mail.ru

An extensible data model for information system storage structures operating in dynamic environment is considered. The aim is to develop a data model that can reflect frequent and significant changes in the subject area using minor and simple changes in storage systems, as well as to provide the speed of performing complex analytical queries on a time scale close to real. The proposed model is based on the idea of decomposing complex data schema into many simpler components that are easier to manipulate. In particular, its evolution occurs not in changing existing data structures, but in creating new ones (expanding). Flexibility and modularity of data structure is achieved by placing business keys in attribute tables.

Key words: information system, analytical query, data warehouse, databases, data model, subject area, business key.

DOI: 10.21667/1995-4565-2020-74-56-64

References

1. **Arhipenkov S. Ja., Golubev D., Maksimenko O.** Hranilishha dannyh. Ot koncepcii do vnedrenija. (Data warehouse. From concept to implementation) Moscow: Dialog-MIFI, 2002. 528 p. ISBN 5-86404-167-x (in Russian).
2. **Nozhenkova L. F., Shajdurov V. V.** OLAP-tehnologii operativnoj informacionno-analiticheskoy podderzhki organizacionnogo upravlenija (OLAP-technology of operative information-analytical support of

organizational management). *Informacionnye tehnologii i vychislitel'nye sistemy*. 2010, no. 2, pp. 15-27 (in Russian).

3. **Paklin N. B., Oreshkov V. I.** *Biznes-analitika: ot dannyh k znanijam (+ CD)*. (Business intelligence: from data to knowledge (+ CD)) 2-d edition, corr. SPb.: Piter. 2013, 704 p.: il. ISBN 978-5-459-00717-6 (in Russian).

4. **Oreshkov V. I.** *Hranilishha dannyh i OLAP-tehnologii (Data warehouses and OLAP technologies)*. Ryazan State Radio Engineering University, Ryazan. 2017, 64 p (in Russian).

5. **Eric Sperley.** *Korporativnye hranilishha dannyh. Planirovanie, razrabotka, realizacija. (The enterprise data warehouse. Planning, building, and implementation)*. M.: Izdatel'skij dom «Vil'jams». 2001, vol. I., 400 p. ISBN 5-8459-0191-X (in Russian).

6. **Adamson, Christopher.** *Star Schema The Complete Reference*. McGraw-Hill Education. 2010, 510 p. ISBN 9780071744331 (in English).

7. **Rönnbäck L.** Anchor Modeling. Agile Information Modeling in Evolving Data Environments / L. Rönnbäck, O. Regardt, M. Bergholtz, P. Johannesson, P. Wohed – URL: <http://www.anchormodeling.com/wp-content/uploads/2011/05/Anchor-Modeling.pdf> (Preprint submitted to DKE 05.10.2010).

8. **Thomsen, Erik.** *OLAP Solutions: Building Multidimensional Information Systems*. 2nd ed. Wiley Publishing. 2002, 696 p. (in English).

9. **Wrembel, Robert.** *Data Warehouses and OLAP: Concepts, Architectures and Solutions*. IGI Global. 2006, 332 p. (in English).