

# Построение семейства согласованных моделей данных в распределенных базах данных, основанных на семантике

**А.Т. Касымалиева**

доцент кафедры информационных систем в экономике  
Кыргызский государственный технический университет им. И.Раззакова  
Адрес: Кыргызская Республика, 720044, г. Бишкек, пр. Мира, 66  
E-mail: aisu@rambler.ru

## Аннотация

При разработке проектов интегрированных корпоративных систем с ориентацией на дальнейшее расширение функций необходимо помнить о преемственности создаваемых моделей и своевременности их уточнений. Эти уточнения соотносятся с перспективой развития организаций или их подразделений, занимающихся разработкой информационных систем (ИС) или других программных продуктов (ПП), расширением функциональности интегрированных корпоративных информационных систем (ИКИС), а также развитием сред проектирования и программирования. В связи с этим автором предлагается применение расширенной схемы уровней моделирования данных и баз данных.

При изучении функций каждого подразделения и построении их модельного описания в подсистемах (частных моделях) можно выделить одни и те же объекты для обеспечения функциональности. Согласованность является одним из преимуществ создаваемой модели, обеспечивая типизацию и стандартизацию процессов создания ИС.

В работе используются механизмы распределения данных, которые на сегодняшний день являются весьма актуальными. Предложенное решение на основе семантического словаря, отражающего основные термины и понятия функциональных задач бизнес-среды моделируемого предприятия, позволяет унифицировать разработку приложений и дополняет стратегию распределения данных по узлам предприятия.

В статье изложены принципы формирования семейства согласованных моделей данных, приведены их формальные описания, разработаны алгоритмы и возможные практики формирования ядра. Рассматриваются преимущества использования и подходы к возможному применению.

**Ключевые слова:** распределенные базы данных, информационные системы, моделирование данных, ядро модели данных.

**Цитирование:** Kasymalieva A.T. Construction of a set of harmonized data models in distributed databases based on semantics // Business Informatics. 2016. No. 2 (36). P. 48–56. DOI: 10.17323/1998-0663.2016.2.48.56.

## Введение

Вопросам распределенных баз данных (распределения данных по узлам) посвящено довольно много разработок и математических моделей (например, [1–3]). При этом в большинстве случаев рассматриваются возможности оптимизации распределенных запросов уже существующих систем.

Все имеющиеся разработки в области моделирования распределенных баз данных можно подразделить на две наиболее часто рассматриваемые группы.

К первой, самой многочисленной, группе относятся алгоритмы динамического перераспределения данных по узлам сети (они описаны в работах таких авторов, как Д.В. Павлов [4], П.П.-Ч. Чен и

Ж. Аюба [5], Х.И. Абдалла [6]). Недостатками этой модели являются сложность алгоритмов перераспределения, организация дополнительных вычислений уже на этапе функционирования системы, блокировка баз данных относительно пользовательских запросов на время проведения перестройки ее структуры, а также оторванность моделей от концептуальной модели предметной области.

Ко второй группе относятся алгоритмы синтеза физических структур модели информационной системы, описанные в работах М.Т. Оззу и П. Валдуриз [7], А.В. Силина [8], В.В. Бескорвайного [9], В.В. Кульбы [2]. Эти алгоритмы основываются на привязке к пользователям системы, а не к функциям, что сказывается на масштабируемости системы и согласованности данных в разных узлах.

В то же время ни одна из рассмотренных групп не использует понятие согласованности моделей на этапах проектирования больших систем или понятие согласованной интеграции уже существующих систем на основе бизнес-функций.

Данная тема не потеряла своей актуальности и на сегодняшний день.

Информационные системы, моделирование которых является затруднительным вследствие их многомерности, принято называть большими. Существует два способа перевода таких систем в разряд относительно малых. В первом случае предполагается использовать более мощные вычислительные средства, с развитой системой набора информационных объектов (баз данных), приступить к их непосредственной разработке и постоянно наращивать. Во втором случае можно изначально, на уровне моделирования, осуществлять декомпозицию многомерной системы на совокупность подсистем меньшей размерности, отслеживая при этом информационные связи, обеспечивающие целостность системы (в этом случае изначально осуществляется разработка распределенной архитектуры системы).

Автором предлагается модель, которая на этапе проектирования интегрированной корпоративной информационной системы (КИС) на основе частных моделей данных (*entity-relationship diagrams, ERD*), отражающих различные аспекты деятельности подсистем большой информационной системы, оптимизирует возможности распределения этих данных по функциональным узлам, с выделением особого узла, определяемого как ядро модели. Методология моделирования распределенных баз данных использует семантический словарь, основанный на принципах онтологии. В состав такого

словаря входят информационные модели систем, соотносимые с каждым объектом или бизнес-процессом модели организации – бизнес-среды. При этом частные модели в момент их слияния в единую систему для унификации использования словаря данных формируют глобальную модель данных (*global entity-relationship diagram, GERD*).

## 1. Требования к модели

Для того, чтобы обеспечить интеграцию частных диаграмм в общую диаграмму «сущность – связь» необходимо выполнение следующих требований:

1. Имена сущностей, представляющих семантически однородные объекты во всех частных диаграммах, должны быть согласованы;

2. Базовый словарь, на основе которого формируются имена сущностей в частных диаграммах *ERD*, должен формироваться на основе единых классификаторов, содержащих сходные сущности предметных областей и/или их акронимов;

3. Каждая сущность, входящая в семейство частных моделей *ERD*, должна иметь строгое вербальное описание;

4. Совокупность сущностей, входящих в глобальную диаграмму «сущность-связь» *GERD*, должна представляться в виде множества имен, без дублирования имен и их псевдонимов, т.е. удовлетворять требованиям первой формы теоретико-множественного представления на уровне перечисления элементов;

5. Связи между сущностями в глобальной модели «сущность – связь» *GERD* образуют семейство пересекающихся множеств сущностей, каждое из которых обеспечивает поддержку основных и сопровождающих бизнес-процессов, функций управления и функций связи с внешней средой;

6. Отбор сущностей в глобальную диаграмму *GERD* должен подчиняться рассмотренному ниже алгоритму построения семейства согласованных моделей.

## 2. Ядро модели данных

**Ядро модели данных (*data model core, DMC*)** – это модель данных, состоящая из сущностей семейства частных моделей, носящих базовые признаки подсистем и их функций в информационной системе управления, выделенных с целью уменьшения противоречивости распределенных данных и обеспечивающих их координирование.

Ядро модели данных формируется путем выявления общих сущностей, описанных в семействе

частных моделей *ERD*. При соблюдении единых правил к формированию имен сущностей и атрибутов процессу создания *DMC* можно придать формализованный характер, путем использования теоретико-множественных операций над семейством сущностей, входящих в частные модели *ERD*. Алгоритм формирования *DMC* приведен ниже. Что касается унифицированных имен атрибутов, то их составы в сущностях моделей *ERD* и *DMC* согласуются на уровне проекций (в смысле операций реляционной алгебры), а полнота атрибутов в каждой из сущностей моделей *ERD* уточняется на уровне разработки частных полностью атрибутизованных моделей проектного уровня (рисунки 1).

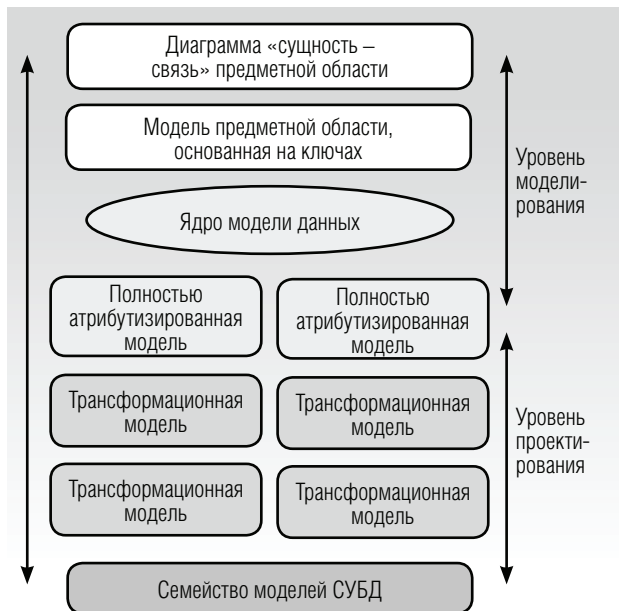


Рис. 1. Уровни моделирования баз данных

### 3. Формализованное описание семейства согласованных моделей данных

Покажем, что существует формальный алгоритм, позволяющий решить как прямую задачу формирования согласованных моделей на основе ядра модели данных и частных диаграмм данных, составляющих основу отдельных программных подсистем, так и обратную – на основе частных диаграмм составить модель данных ядра некоторой глобальной системы, представляющей предметную область. Для доказательства этого факта воспользуемся рядом положений реляционной алгебры, предлагаемых в нотации Д. Мейера [10].

Пусть задано семейство множеств *ERD*, составленное из наборов имен сущностей  $E_i$  частных диаграмм моделей данных  $ERD_i$ .

$ERD = \{ERD_1, ERD_2, \dots, ERD_i, \dots, ERD_m\}$ , где  $ERD_i \in ERD, 1 \leq i \leq m$ ;

$$E_i = \{e_{i1}, e_{i2}, \dots, e_{ij}, \dots, e_{in}\}, 1 \leq i \leq m, 1 \leq j \leq n,$$

где  $m$  – число диаграмм «сущность – связь», входящих в множество *ERD*;

$n$  – число сущностей, входящих в каждую диаграмму  $ERD_i$ .

Кроме того, определим некоторое глобальное множество *GERD*, составленное из всех имен  $e_{ij}$  сущностей, входящих в диаграммы  $ERD_1, ERD_2, \dots, ERD_i, \dots, ERD_m$  такое, что

$$E_1 \in ERD_1, E_2 \in ERD_2, \dots, E_m \in ERD_m;$$

$$E_i = \{e_{i1}, e_{i2}, \dots, e_{ij}, \dots, e_{in}\}, 1 \leq i \leq m, 1 \leq j \leq n,$$

т.е. множество  $GERD = E_1 \cup E_2 \cup \dots \cup E_m =$

$$\{\{e_{11}, e_{12}, \dots, e_{1j}, \dots, e_{1n}\}, \dots, \{e_{i1}, e_{i2}, \dots, e_{ij}, \dots, e_{in}\}, \dots, \{e_{m1}, e_{m2}, \dots, e_{mj}, \dots, e_{mn}\}\}$$

Поставим в соответствие каждому из имен сущностей  $e_{ij}$  некоторую интерпретирующую функцию  $\varphi$ , такую, что

$$\varphi_{ij} : e_{ij} \in u_{ij}, e_{ij} \in GERD, u_{ij} \in SI,$$

где  $u_{ij}$  – некоторое вербальное описание смысла (семантики) имени сущности  $e_{ij}$ ;

*SI* – некоторое множество семантической информации (*semantic information*), составляющее пространство интерпретации элементов глобального множества *GERD*.

Определим некоторое функциональное отображение *F*, такое, что *F*:

$$GERD \times SI \rightarrow DTN,$$

где *DTN* – некоторое множество двоек, такое, что каждому имени сущности  $e_{ij}$  сопоставляется ее смысловое значение (семантика)  $u_{ij}$ . Тогда отображение *DTN* может интерпретироваться как словарь имен сущностей, где каждое имя сущности наделено соответствующей предметно-ориентированной интерпретацией (описанием).

Определимся с тем, как будут соотноситься между собой сущности модели ядра данных *DMC* и сущности частных моделей данных *ERD* на этапе разработки ключевых моделей. Напомним, что каждая из рассмотренных сущностей есть отношение. Поэтому в дальнейшем эти сущности будем рассматривать как отношения *R*, определенные как подмножества некоторого декартова произведения семейства множеств  $A_i$ .

Введем следующие определения.

**Определение 1.** Схемой отношения  $R$  будем называть конечное множество имен атрибутов  $\{A_1, A_2, \dots, A_n\}$ , где каждому атрибуту поставлено в соответствие множество  $D_i$ , называемое доменом атрибута  $A_i$ ,  $1 \leq i \leq n$ .

Домен атрибута принято обозначать  $dom(A_i)$ . При этом домены являются произвольными непустыми конечными (счетными) множествами.

Пусть  $D = D_1 \cup D_2 \cup \dots \cup D_n$ . Тогда можно ввести следующее определение.

**Определение 2.** Отношение  $r$  со схемой  $R$  – это конечное множество отображений  $\{t_1, t_2, \dots, t_p\}$  из  $R$  в  $D$ , где каждое отображение  $t \in r$  должно удовлетворять следующему ограничению:  $t(A_i)$  принадлежит  $D_i$ ,  $1 \leq i \leq n$ . Эти отображения называются кортежами отношения  $r$  со схемой  $R$ . При этом под кортежем принято понимать множество значений, взятых по одному для каждого имени атрибута из схемы отношения  $R$ .

Если интерпретировать  $t$  как строку таблицы, то  $A$ -значение  $t(A)$  кортежа  $t$  является содержанием (значением) кортежа  $t$  в столбце  $A$ . Таким образом, отношение  $r$  можно рассматривать как некоторую таблицу с множеством кортежей  $t$ , удовлетворяющих схеме отношения  $R$ .

Основываясь на приведенных определениях, будем полагать, что каждая сущность  $e$ , принадлежащая множеству  $ERD$ , построена на основе схемы отношения  $R$  с набором кортежей  $t$ , а каждая конкретизированная сущность  $e_{ij}$  представлена схемой  $R_{ij}$ .

Пусть некоторым образом определен набор сущностей  $e_k^0, e_k^0, 1 \leq k \leq K$ , входящих в ядро модели данных  $DMC$ , и для каждой сущности определена схема отношения  $R_k^0$ .

**Определение 3.** Будем считать, что сущность  $e_{ij}$  полностью совместима с сущностью  $e_k^0$  (*full compatible data*), если соответствующие им схемы отношений  $R_{ij}$  и  $R_k^0$  удовлетворяют условию  $R_{ij} = R_k^0$ .

**Определение 4.** Будем считать, что сущность  $e_{ij}$  частично совместима (*partial compatible data*) с сущностью  $e_k^0$ , если соответствующие им схемы отношений  $R_{ij}$  и  $R_k^0$  удовлетворяют условию  $R_{ij} = S_k^0$ , где  $S_k^0 \subseteq R_k^0$ .

При формировании ключевых или полностью атрибутизированных моделей ядра модели данных  $DMC(KB^0)$  на основе частных ключевых моделей  $ERD(KB_i)$  (или в обратном направлении) следует придерживаться следующих вариантов совместимости:

♦ схемы отношений сущностей (подсистем)  $R_{ij}$ , входящих в некоторую  $ERD_i$ , и схемы отношений

одноименных аналогов  $R_k^0$  в  $DMC$  должны обладать свойством полной совместимости. Иначе имена и число атрибутов, порядок их следования и домены, на которых они определены, должны быть одинаковыми;

♦ схемы отношений сущностей (подсистем)  $R_{ij}$ , входящих в некоторую  $ERD_i$ , и схемы отношений одноименных аналогов  $R_k^0$  в  $DMC$  могут обладать свойством частичной совместимости. Иначе в частично совместимых схемах отношений схема данных  $R_{ij}$  является подмножеством схемы данных  $R_k^0$ , т.е.  $R_{ij} \subseteq R_k^0$ , т.е.  $R_{ij} = R_{ij}'$ .

♦ схемы отношений сущностей (подсистем)  $R_{ij}$ , входящих в некоторую  $DMC$ , и схемы отношений одноименных аналогов  $R_k^0$  в  $ERD_i$  могут обладать свойством частичной совместимости. В этом случае в частично совместимых схемах отношений схема данных  $R_k^0$  является подмножеством одноименной схемы данных  $R_{ij}$ , т.е.  $R_k^0 \subseteq R_{ij}$ .

Естественно, ключевые поля в совместимых схемах данных из  $DMC$  и  $ERD_i$  должны быть одинаковыми.

При расположении атрибутов в ключевых моделях желательно придерживаться следующей рекомендации. Одноименные атрибуты в обеих схемах должны иметь один и тот же порядок следования. Это значительно упростит процедуру передачи данных между таблицами ядра данных и частными моделями данных, когда они будут обрабатываться в некоторой  $СУБД$  в формате полностью атрибутизированных или трансформационных моделей.

Последнее требование не претендует на исключительность. При некотором усложнении процедуры копирования данных одинаковый порядок следования атрибутов (столбцов) в обрабатываемых моделях не обязателен. Более того, возможна ситуация, когда в сопоставляемых моделях соответствующие имена атрибутов соотносятся как синонимы, определенные на одних и тех же доменах. Такое переименование атрибутов теоретически допустимо и предусмотрено соответствующими теоремами реляционной теории. Заметим, что на этой основе построены практически все модели (службы  $DTS$ ) трансформации данных современных  $СУБД$ .

**Передача данных в  $СУБД$**  между частично совместимыми сущностями  $e_k^0$  ядра модели данных  $DMC$  и частными моделями данных  $e_{ij}$  из множества  $ERD$  выполняется путем применения операций выбора  $\sigma$  или проекции  $\pi$  к отношениям  $r_k^0$  (таблицам) со схемой отношений  $R_k^0$  из множества сущностей  $DMC$  и обеспечивается механизмами репликации современных  $СУБД$  [11, 12].



**Операция выбора.** Результатом применения операции выбора  $\sigma$  к отношению  $r$  является другое отношение, которое представляет собой подмножество кортежей отношения  $r$  с определенным значением в выделенном атрибуте. Пусть  $r$  – отношение со схемой  $R$ ,  $A$  – атрибут в  $R$  и  $a$  – элемент из  $dom(A)$ . Тогда  $\sigma_{A=a}(r)$  – обозначение операции выбора («выбрать из  $r$  кортежи, в которых значение  $A$  равно  $a$ »). Рассматривая кортежи как отображения, можно записать:  $r'(R) = \{t \in r \mid t(A) = a\}$ . Операция выбора  $\sigma_{A=a, B=b}(r)$  по нескольким атрибутам возможна в силу того, что она обладает свойством коммутативности.

Операция проекции – это операция, которая позволяет обмениваться данными между частными моделями данных  $e_{ij}$  из множества  $ERD$  и  $r_k^0$  из множества сущностей модели  $DMC$ , путем вырезки части атрибутов из схемы  $R_k^0$  и формирования нового отношения  $r'_{ij}$ .

Пусть  $r$  – отношение со схемой  $R$ , а  $X$  – подмножество  $R$ . Проекция  $r$  на  $X$ , записанная как  $\pi_X(R)$ , есть отношение  $r'(X)$ , полученное вычеркиванием столбцов, соответствующих атрибутам в  $R - X$  (операция разности множеств), и исключением из оставшихся столбцов повторяющихся строк. Рассматривая кортежи как отображения,  $\pi_X(R)$  можно записать в виде  $r'(X) = \{t(X) \mid t \in r\}$ .

#### 4. Алгоритм формирования ядра модели данных

Алгоритм формирования ядра модели данных выглядит следующим образом.

1. Составить множество  $GERD$  имен сущностей  $e_i$ , где  $e_i \in R$ , путем приписывания к составу сущностей первой модели  $ERD_1$  всех сущностей остальных моделей  $ERD_2, \dots, ERD_n$ ;

2. Отобрать из набора  $GERD$  повторяющиеся имена сущностей, зафиксировав их имена и число вхождений в множество  $GERD$ , и занести их в множество двоек  $G$ , где каждая двойка составлена из имени сущности и числа вхождений  $x$  этой сущности в множество  $GERD$ ;

3. Выполнить ранжирование элементов множества  $G$ , упорядочивая сущности по убыванию их числа вхождений  $x$  в множество  $G$ ;

4. Отобрать сущности, имеющие  $x$  вхождений в множество  $G$  по условию  $x \geq k$ , где  $k$  – пороговое значение числа имен сущностей, отобранное экспертами для включения их имени в модель ядра  $DMC$ .

Из приведенного (возможно, не полного) перечня требований следует, что удовлетворить их возможно только при условии существования развернутой бизнес-модели организации – семейств моделей, описывающих ее организационную и процессную структуру, бизнес-функционал, модели связи организации с внешней средой и модели распределения функций ответственности за исполнение элементов функционала.

Практика формирования ядра модели  $DMC$  показала, что в это ядро чаще всего оказывается включенными сущности, представляющие следующие данные:

- ◆ данные, описывающие организационную структуру предприятия или организации;
- ◆ данные, специфицирующие бизнес-направление организации;
- ◆ данные, специфицирующие продуктовый портфель или состав рыночных услуг организации;
- ◆ данные детализированного описания товаров (продуктов) или услуг;
- ◆ данные, специфицирующие ресурсную составляющую организации (кадры, знания, материальные и информационные ресурсы);
- ◆ данные, поддерживающие управление основными и сопровождающими процессами – учетную компоненту вектора характеристик управления;
- ◆ данные, определяющие классификатор базовых документов организации и документов, составляющих ее документооборот;
- ◆ данные, описывающие внешнюю среду, включая поставщиков, клиентов, потребителей и т.п.;
- ◆ данные, поддерживающие программное управление ядром  $DMC$ ;
- ◆ прочие данные, характеризующие отраслевую принадлежность организации (производственную, социальную, административную, консультативную и др.).

Естественно, что состав сущностей, включаемых в модель  $DMC$ , зависит от профиля организации, поэтому совсем не обязательно включать в модель ядра все вышеперечисленные группы сущностей.

Возможен и другой, формализованный путь отбора сущностей в ядро  $DMC$ , основанный на алгоритмическом подходе.

#### 5. Преимущества предлагаемого подхода

С использованием ядра преимущества выглядят следующим образом.

**Локальная независимость.** Получая данные ядра,

каждая подсистема содержит актуальные данные, необходимые для функционирования в соответствующей операционной среде, в то же время оставаясь независимой от данных ядра.

Надежность такой системы повышается за счет независимости подсистем как друг от друга, так и от ядра. При этом повышается доступность каждой из подсистем: отсутствие работоспособности одной из них никак не скажется на доступности других.

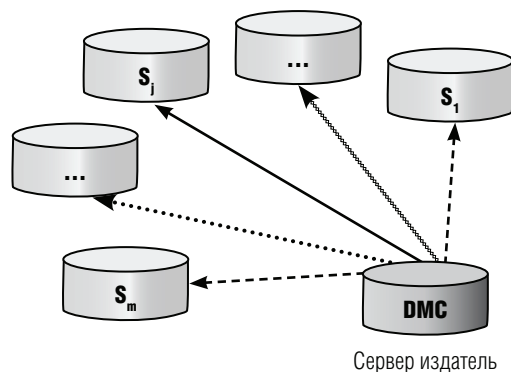
**Независимость от расположения.** Запрос получает необходимое значение узла из ядра и перенаправляет действия системы к заданному фрагменту данных. Подобное моделирование позволяет разработчикам унифицировать написание кода, оптимизируя его за счет использования согласованных схем частных моделей.

**Независимость от фрагментации.** Поскольку каждая подсистема является локально независимой и имеет актуальные данные, при восстановлении ядра из фрагментов достигается минимизация потерь информации.

**Минимизация использования сетей.** Поскольку каждая подсистема локально независима, можно минимизировать использование распределенных запросов. Что же касается тактических и стратегических задач, то их, наоборот, эффективнее решать через ядро.

## 6. Подход к выбору критерия $k$

Как ни парадоксально, чем чаще информация фигурирует в процессах различных подсистем, тем меньше она подвержена различным изменениям, при этом обращение к ней происходит гораздо реже. Ценность такой информации выше, чем стоимость ее хранения. Что касается информации, ха-



- Подписка без фрагментации
- ⋯ Подписка с вертикальной фрагментацией
- - - Подписка с использованием фильтров - горизонтальная фрагментация
- ⋯ - - - Подписка смешанной фрагментации

Рис 2. Возможные варианты подписки на реплицируемые данные

рактерной для задач оперативного уровня, то она, наоборот, чаще подвержена модификации. Следовательно, возрастает количество обращений к такой информации (такие задачи решаются гораздо чаще тактических или стратегических).

Таким образом, при решении оперативных задач в подсистемах число обращений к ядру будет возрастать, а объем такой информации увеличиваться. Все это может снизить эффективность использования распределенных данных. Такую информацию целесообразно хранить в распределенной среде, доступ к которой будет происходить только при решении данной задачи именно данным подразделением.

Поэтому с целью минимизации использования сетей (задача любой распределенной базы данных) значение  $k$  должно быть таким, чтобы стоимость хранения информации сущности мощностью  $|H|$ , обеспечивающей решение задач, была ниже стоимости необходимых обращений к ней (*communication costs*,  $CC$ ) и затрат на обновления (*updating costs*,  $UC$ ). Эти операции требуют многочисленных пересылок больших объемов данных с одного узла на другой (или другие), с целью необходимой актуализации информации и ожидания задержки обновления.

Предложенная в работе модель предполагает, что есть распределенная база данных, состоящая из  $m$  сайтов

$$S = \{DMC, S_1, S_2, \dots, S_m\}$$

построенных на принципах согласованности моделей на основе ядра. Связь между  $DMC$  и  $S_j$  имеет положительное целое число коммуникационных затрат (*communication costs* –  $CC_j$ ), представляющих стоимость трансфера блоков данных с  $DMC$  на  $S_j$ , и положительное целое число затрат на обновление (*updating costs* –  $UC_j$ ), которые возникают с задержкой  $l$ .

Пусть на решение определенной задачи приходится множество запросов  $Q = \{Q_1, Q_2, \dots, Q_l\}$ , которые являются наиболее частыми запросами, и на которые приходится основная нагрузка от обработки, суммарная стоимость которых определена как  $QC_j$ .

Пусть  $t$  – частота решения задачи поиска,  $v$  – частота обновления данных,  $k$  – количество сайтов в которых распределена сущность.

Если мы отобрали сущность в состав ядра, то полная стоимость составляет:

$$TC_c = t(CC_j + QC_j) + 2vUC_j.$$

Также для сущности, которая не войдет в ядро, справедливо:

$$TC_s = tk(CC_j + QC_j) + vUC_j.$$

При обновлении данных необходимо обновить две подсистемы – ядро и непосредственно ту подсистему, для которой данные актуальны (репликация вертикальной фрагментации).

Обозначив через  $a = t(CC_j + QC_j)$ ,  $b = vUC_j$ , можно увидеть две функции, рост которых можно оценить в зависимости от  $k$ :

$$TC_s = ak + b;$$

$$TC_c = a + 2b.$$

При  $k = 1$  более предпочтителен вариант размещения сущности в одном сайте.

При  $k = 2$  необходимо оценить величину, равную  $a - b$ , а также во временных расчетах дополнительно учесть величину  $l$  (задержка на репликацию данных из ядра).

Исключив такую сущность из ядра мы также уменьшим мощность ( $|r|$ ) данной сущности и не увеличим затраты  $QC_j$ .

Если важна производительность системы, то проблема оптимальности, согласно предложенной модели перераспределения, может быть определена как минимизация затрат на связь в процессе перераспределения данного отношения из ДМС на необходимые фрагменты из  $S$ .

При  $k > 2$  возрастает величина  $TC_s$ , то есть затраты на запросы, связанные с поиском по разным фрагментам и передачей данных по сети, будут в значительной степени превышать эффективность от их хранения в одном месте – ядре БД.

## 7. Результаты

Результаты предложенного подхода апробированы на тестовых данных АСУ вуза по двум факультетам двух первых курсов Кыргызского государственного технического университета им. И.Раззакова.

При построении модели были выделены частные модели и создан унифицированный словарь атрибутов. При рассмотрении учебного процесса был выявлен ряд подмоделей, а на следующем шаге было создано ядро модели.

Все частные модели актуализируются за счет данных ядра. Поэтому сведения о направлениях и планах с их содержимым, а также об учебных подразделениях, участвующих в учебном процессе, будут получать одинаковые наборы данных. Сущность эксперимента заключается в измерении времени, которое потребуется каждой системе управления данными на выполнения запроса для удовлетворения информационной потребности сотрудника АСУ вуза.

Для решения задач на оперативном уровне системы было предложено два варианта запросов: к централизованной базе данных и распределенной. На приведенных ниже рисунках результаты показывают временную разницу при одинаковых состояниях аппаратного обеспечения с использованием Microsoft SQL Server 2008 на одной и тоже вычислительной машине. Подписки распределили по разным базам данных. Цель запроса – поиск студента по фамилии и номеру зачетной книжки, вывод перечня дисциплин и оценок на данный момент.

Таблица 1.

Параметры замеров запросов для централизованной базы данных

No	Параметры					
	Общая память (МБ)	Занятая память (МБ)	Свободная память (МБ)	Температура процессора (°C)	Напряжение процессора (В)	Сила тока (А)
1	3071	1559	1512	33	1.072	9.11
2	3071	1569	1502	33	1.072	9.11
3	3071	1569	1502	33	1.072	9.11
4	3071	1572	1499	33	1.072	9.11
5	3071	1618	1453	33	1.072	9.11
6	3071	1571	1500	33	1.064	9.11
7	3071	1563	1508	33	1.128	9.11
8	3071	1578	1493	33	1.072	9.11
9	3071	1570	1495	33	1.072	9.11
10	3071	1583	1488	33	1.072	9.11
среднее	3071	1575	1496	33	1.077	9.11

Таблица 2.

Параметры замеров запросов для распределенной базы данных

No	Parameters					
	Общая память (МБ)	Занятая память (МБ)	Свободная память (МБ)	Температура процессора (°C)	Напряжение процессора (В)	Сила тока (А)
1	3071	1630	1441	33	1.072	9.11
2	3071	1581	1490	34	1.072	9.11
3	3071	1585	1586	33	1.072	9.11
4	3071	1559	1512	34	1.072	9.11
5	3071	1578	1493	33	1.072	9.11
6	3071	1579	1492	33	1.072	9.11
7	3071	1621	1450	33	1.072	9.11
8	3071	1581	1490	33	1.072	9.11
9	3071	1578	1493	33	1.072	9.11
10	3071	1575	1496	33	1.072	9.11
среднее	3071	1586	1494	33	1.072	9.11

Результаты постановки эксперимента приведены на графике (рисунок 3), запрос выполнялся по 10 раз через каждые 30 минут.

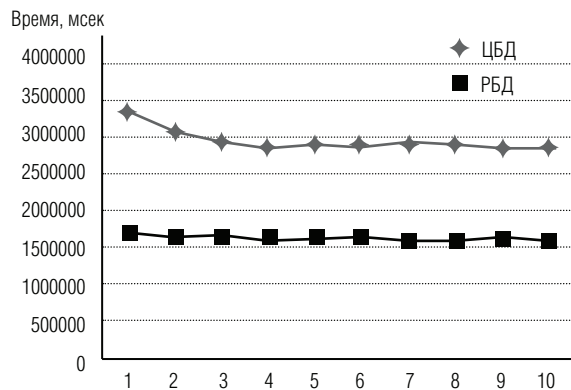


Рис. 3. Результаты эксперимента

**Заключение**

Физически архитектура может быть представлена как семейство серверов, поддерживающих частные подсхемы моделей данных, по одному на каждую категорию функциональных задач. Из них выделяется один сервер, содержащий схему ядра модели данных. Возможна реализация одного из следующих решений:

- ◆ схема ядра модели данных является эталонной моделью, на основе которой формируются частные модели серверов;
- ◆ схема ядра модели данных является моделью

такой базы данных, которая собирает все вводимые и редактируемые данные (там, где они появляются) в одном месте, а затем тиражирует их по частным моделям. Иначе говоря, эта база данных берет на себя функции диспетчеризации, которые поддерживают целостность и избыточность данных в частных СУБД. Вся функциональность этой СУБД направлена только на поддержание функций диспетчеризации (реплицирование, поддержание целостности и избыточности).

◆ схема ядра модели данных является базовой моделью СУБД, которая поддерживает главные бизнес-функции и функции управления организации, реализуемые в рамках ИКИС. Все остальные СУБД обеспечивают обеспечивающие (сопровождающие) бизнес-функции и функции управления организации, реализованные в виде данных различного функционального назначения.

Возможны и другие архитектурные решения, являющиеся комбинацией перечисленных структур.

Выбор той или иной структурной организации пространства данных может быть выполнен либо на основе экспертной оценки, либо путем математического решения задачи оптимального выбора.

Преимущества предложенного подхода, прежде всего, состоят в согласованности распределенных данных, масштабируемости системы, ее быстрой адаптации к меняющейся среде, а также в привязке системы не к конкретным пользователям, а к бизнес-функциям. ■



### Литература

1. Yu C.T., Chang C.C. On the design of a query processing strategy in a distributed database environment // Proceedings of SIGMOD'83, Annual Meeting, San Jose, California, May 23–26, 1983. New York: ACM Press, 1983. P. 30–39.
2. Теоретические основы проектирования оптимальных структур распределенных баз данных / В.В. Кульба и [др.]. М.: Синтег, 1999. 660 с.
3. Лаздынь С.В., Землянская С.Ю. Оптимизация распределенных корпоративных информационных сетей с использованием генетических алгоритмов и объектного моделирования // Наукові праці ДонНТУ. 2009. № 147. С. 83–95.
4. Павлов Д.В. Реляционная распределенная система управления базами данных с автоматической масштабируемостью // Вестник УГАТУ. 2012. Том 16, № 3 (48). С. 138–142.
5. Chen P.P.-S., Akoka J. Optimal design of distributed information systems // IEEE Transactions on Computers. 1980. Vol. c-29. No. 12. P. 1068–1080.
6. Abdalla H.I. A new data re-allocation model for distributed database systems // International Journal of Database Theory and Application. 2012. Vol. 5. No. 2. P. 45–60.
7. Tamer Ozsu M., Valduriez P. Principles of distributed database systems. Upper Saddle River, NJ: Prentice-Hall, 1991. P. 125–128.
8. Силин А.В., Воробьев В.И., Ревунков Г.И. Методы и модели проектирования структур территориально-распределенных баз данных // Деп. рук. ВИНТИ № 3282-00В. 2005. С. 21–25.
9. Бескоровайный В.В., Ульянова О.С. Методы синтеза физических структур распределенных баз данных // Открытые информационные и компьютерные интегрированные технологии. 2010. № 47. С. 136–146.
10. Maier D. The theory of relational databases. Computer Science Press, 1983. 656 p.
11. Bell D.A., Grimson J.B. Distributed database systems. Wokingham, England: Addison-Wesley, 1992. 426 p.
12. Бабак В., Касымалиева А., Акматбеков Р. Поддержка распределенных данных в системах автоматизированного управления учебным процессом вуза // Известия КГТУ. 2008. № 13. С. 333–337.