

УДК 004.021:004.045:004.67:004.622

**МОДЕЛИРОВАНИЕ РАБОТЫ ПСЕВДО-СЕТЕВОЙ ВЕРСИИ
БАЗЫ ДАННЫХ В MS ACCESS НА ПРИМЕРЕ ОТДЕЛА ПРОДАЖ
МАЛОГО ПРЕДПРИЯТИЯ**

С.П. Макаров, Т.А. Бороненко, А.Г. Коробейников
(публикуется в порядке дискуссии)

Рассмотрена проблематика и особенности сетевых баз в Access в соответствии со спецификой предметной области. Предлагается архитектура для построения базы данных. Смоделирована миграция данных по всем узлам построенной модели псевдо-сетевой версии базы данных. Проанализирована динамика модели. На примере отдела продаж малого предприятия показана эффективность разработанной модели.

Ключевые слова: моделирование, сетевые базы данных, сетевая архитектура, MS Access, архитектура файл-сервер, AllFusion Process Modeller, нотация IDEF0.

Введение

Специфика экономических процессов в наше время заставляет малые предприятия искать более простые и удобные решения обработки, анализа и хранения финансово-экономических данных. Это ниша программного обеспечения, управляющего базами данных (БД). В данный момент наиболее доступным и простым решением является использование СУБД Microsoft® Access 2003 (в дальнейшем MS Access). Один из основных плюсов разработки на MS Access – плотная связь с Microsoft Office. Для работы созданного приложения достаточно установить Microsoft Office (в дальнейшем – «Офис»). При этом устанавливаются все необходимые для работы MS Access библиотеки, драйверы работы с базами данных (ODBC, OLEDB, MS ADO и т.д.). Кроме того, MS Access во многом совместим с СУБД Microsoft SQL Server и другими форматами баз данных [1].

Проблематика и особенности сетевых баз в Access

К сожалению, MS Access имеет существенный минус – в нем отсутствует возможность реализации полноценной сетевой версии СУБД «клиент–сервер». MS Access не имеет собственной серверной компоненты, но в этом продукте реализована возможность применить технологию ADO, которая имеет поддержку локальных СУБД различных типов [1]. Таким образом, проблему отсутствия возможности реализации полноценной сетевой версии БД «клиент–сервер» можно решить путем создания псевдосетевой версии БД.

Один из наиболее распространенных методов реализации сетевой БД в MS Access базируется на использовании монопольного доступа. В данной конфигурации единый MDB-файл располагается только на сетевом сервере. Основные положения метода заключаются в следующем.

- Пользователи получают доступ к базе данных при обращении к единому MDB-файлу на сервере.
- При совместной работе нескольких пользователей с БД MS Access сохранение изменений возможно только в случае открытия БД в монопольном режиме. В этом случае пользователи в режиме поиска и считывания практически не зависят от пользователей, изменяющих информацию. Пользователь, монопольно изменяющий информацию, подлежит индивидуальной регулировке доступа к информационным структурам.
- При попытке внести изменения в структуру объекта БД (кроме таблиц и запросов) или элемента в режиме общего доступа MS Access временно предоставляет пользователю (частично) монопольный доступ к БД MS Access [2]. После сохранения всех изменений структуры и закрытия всех окон режима конструктора (или завершения процесса, в контексте которого были произведены изменения данных) MS Access возвращает БД MS Access в режим общего доступа. До этого времени другие пользователи не имеют возможности вносить изменения в БД. Если встроенный локальный редактор программных модулей MS Access (Visual Basic Editor – VBE), VisualStudio.NET или прочие подобные продукты запущены в данный момент и частично блокируют фрагменты БД, необходимо сохранить все открытые модули и закрыть приложение, чтобы вернуть БД в режим общего доступа.
- При попытке внесения серьезных изменений (например, изменения формы ввода или отображения данных) в БД MS Access, открытую несколькими пользователями в режиме общего доступа, MS Access выводит предупреждение о том, что изменения могут быть не сохранены.

Данный метод подразумевает, что все формы, отчеты, модули, запросы, EXE-файлы Access, а также все библиотеки DLL и т.п., должны передаваться по сети на рабочую станцию, сетевой трафик возрастает, а производительность значительно снижается.

К тому же данный метод не обеспечивает возможности работы с БД при отсутствии сетевого подключения или нарушении работы локальной сети, что понижает мобильность рабочих станций (например, при использовании в качестве рабочих станций ноутбуков).

В условиях отдела продаж малого предприятия, в котором ноутбуки используются в качестве пользовательских рабочих станций и периодически не имеют подключения к локальной сети, возникает потребность использования иного метода реализации БД.

Таким образом, ставится задача предложить и смоделировать работу более эффективного метода реализации БД, который позволяет работать с БД одновременно нескольким пользователям, а также дает возможность запускать БД и открывать данные для чтения в условиях отсутствия подключения к локальной сети. В работе предлагается решение с применением связующего звена. Это решение назовем псевдо-сетевой версией БД, поскольку для реализации этого решения используется несетевая СУБД MS Access. Основные положения следующие.

- Существует файл-сервер, на котором хранятся все MDB-файлы, используемые БД, с внесенными в таблицы наборами сведений.
- Пользовательские станции хранят на своих жестких дисках MDB-файлы, загруженные с файл-сервера.
- Администратор экспортирует актуальные данные на сервер, импортирует данные из внешних источников к себе на жесткий диск. Также данные можно вносить посредством пользовательского интерфейса на рабочей станции администратора.
- Обновление данных осуществляется как при каждом запуске БД на пользовательской рабочей станции, так и периодически в установленный пользователем момент.

Архитектура и моделирование работы базы данных

Архитектура модели псевдо-сетевой БД состоит из трех основных модулей (рис. 1).

- Административный модуль – БД, реализованная посредством MS Access, в которой доступен просмотр, внесение и изменение данных, а также экспорт их на серверный модуль.
- Серверный модуль (в дальнейшем сервер) – пространство на общедоступном сетевом диске, где хранятся файлы БД с наборами сведений предметной области.
- Пользовательский модуль – БД, реализованная посредством MS Access, по сути, являющаяся частичной копией административного модуля, в которой доступны только просмотр и импорт данных с сервера.

В административный модуль вносятся данные (набор сведений). Это может происходить двумя способами:

- пользователь, оперирующий административным модулем (далее – администратор) вносит изменения или корректирует имеющиеся данные;
- администратор импортирует данные (набор сведений) из внешних источников (бухгалтерской или/и складской программы, представленных в виде таблиц Microsoft (с) Excel 2003).

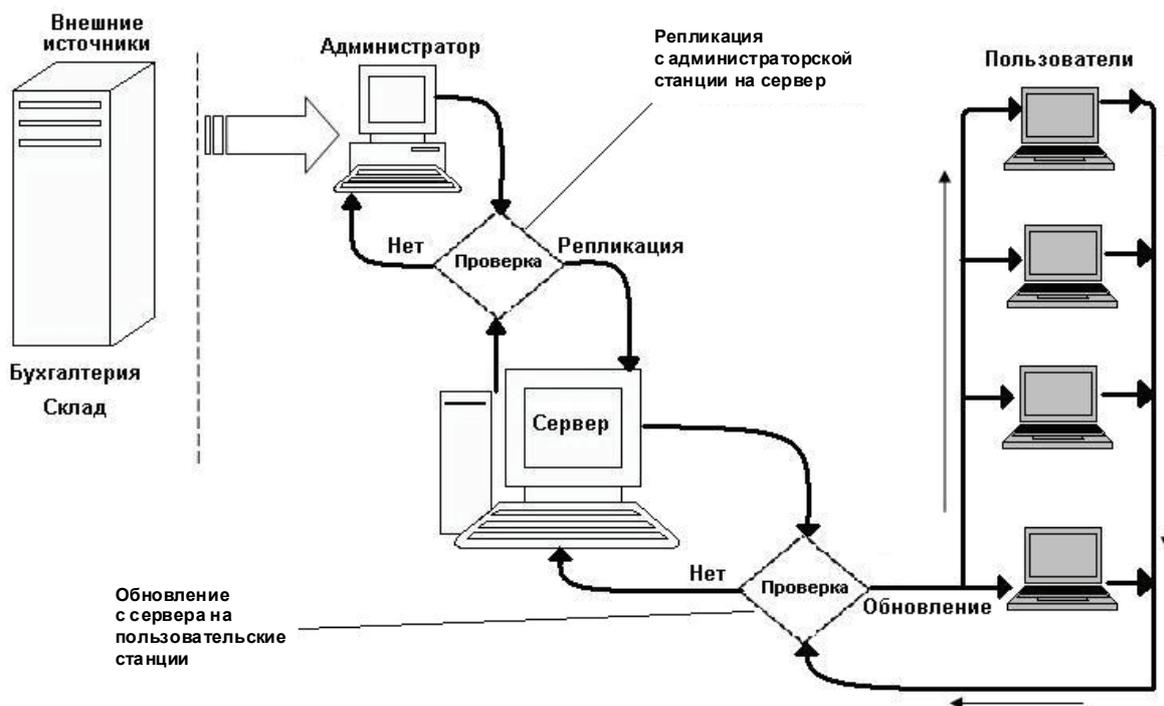


Рис. 1. Архитектура псевдо-сетевой базы данных

Администратор реплицирует их со своей рабочей станции на сервер с целью обновить имеющиеся данные и поддержать их актуальность.

Пользователь, оперирующий пользовательским модулем (в дальнейшем – пользователь), при запуске реплицирует данные с сервера на свою рабочую станцию с целью поддержания актуальности данных, либо использует уже имеющиеся у себя на диске при отсутствии подключения к локальной сети.

Миграция данных осуществляется по цепочке: внешний источник, администратор, сервер, пользователь.

Здесь следует обратить внимание на два момента (рис. 1):

1. *Момент репликации администратором данных на сервер.* В момент репликации данных администратором на сервер осуществляется проверка на их актуальность. С помощью клиентского программного обеспечения TortoiseSVN системы контроля версий Subversion осуществляется контроль внесения изменений в файлы БД. С помощью данной системы из файлов БД, хранящихся на сервере, формируется специальное абстрактное хранилище, в которое включаются все файлы БД, административный и пользовательские модули, а также ведется история изменений файлов. Администратор при помощи TortoiseSVN посылает запрос на проверку актуальности данных на своей рабочей станции и на сервере. Система Subversion синхронизирует файлы и анализирует историю изменений файлов. Если данные на сервере актуальны, то обновление не происходит. Это позволяет снизить загруженность сети и файлов на сервере. Если же данные на сервере не актуальны, то происходит внесение всех изменений в хранилище. Аналогичный процесс протекает и в обратном направлении: если на рабочей станции данные не актуальны, то в рабочую копию администратора вносятся изменения из хранилища. При сохранении новых версий пользовательских модулей используется дельта-компрессия: система находит отличия новой версии от предыдущей и записывает только их, избегая дублирования данных. Соответствующая история по любым изменениям, внесенным в файлы, сохраняется в специальном журнале на сервере.
2. *Момент обновления данных на пользовательской рабочей станции с сервера.* При запуске программы пользователем автоматически осуществляется проверка на необходимость обновления данных с сервера в пользовательский модуль. Если данные у пользователя идентичны данным на сервере, то они не обновляются, чтобы также не замедлять работу приложения, сети и сервера. Репликацию с сервера пользователь также может осуществить принудительно во время работы программы при помощи пользовательского интерфейса TortoiseSVN, так как при длительной работе с данными в пользовательском архиве, данные на сервере могут поменяться. В процессе этого вида репликации пользовательский модуль освобождает все активные файлы, и осуществляется процесс проверки данных на актуальность, идентичный процессу проверки при репликации данных с административного модуля на сервер.

Также при наличии локальной сети администратор и пользователь могут вносить изменения непосредственно в некоторые таблицы (что определяется реализацией интерфейса пользовательского и адми-

нистративного модуля) на сервере, при этом доступ к этим таблицам реализуется с построчной блокировкой записей при помощи технологии ADO. MS Access блокирует изменяемую в данный момент запись, не позволяя изменять ее другим пользователям. Заблокированными могут оказаться другие записи, расположенные рядом на диске. Если другой пользователь пытается изменить заблокированную запись, в его объекте в режиме таблицы появляется специальный маркер, который указывает, что вносятся изменения и запись заблокирована. При попытке ввести данные в заблокированную запись раздается звуковой сигнал. При попытке одновременного внесения изменений в запись пользователем и администратором приоритет отдается администратору. При попытке одновременного внесения изменений в запись несколькими пользователями, приоритет отдается пользователю с наиболее высоким значением в списке приоритета (реализованном с помощью встроенного локального редактора программных модулей MS Access Visual Basic Editor – VBE). С помощью интерфейса пользовательского модуля определяется набор таблиц, в которые пользователь может вносить изменения. Реализуется это посредством меню с наличием или отсутствием пунктов, обеспечивающих доступ к просмотру форм, использующих определенные таблицы. Эти наборы таблиц достаточно индивидуальны, и пересечение таблиц в них встречается достаточно редко, что делает конкуренцию доступа к данным незначительной (например, пользователь, работающий только с продажами; пользователь, работающий с остатками на складах; пользователь, работающий с бонусами, премиями, акциями). Эта стратегия не является основной, а используется как дополнительный способ поддержания целостности данных. Она доступна только при работе в локальной сети.

Процесс восстановления данных при сбое рабочей станции или сервера осуществляется путем восполнения или замены поврежденных данных из специальных резервных копий, данных на сервере и на оптическом носителе. Для этого применяются:

- RAID 1, обеспечивающий восстановление самой свежей информации. Файлы, расположенные на сервере с RAID, более защищены от поломок, чем хранящиеся на локальной машине;
- ручное или автоматическое копирование на оптический носитель, с помощью которого используется специализированная программа резервного копирования (System Center Data Protection Manager).

При дефектах файлов БД на сервере, возникших в ходе репликации с пользовательских станций или при частичном физическом выходе из строя серверного оборудования, MS Access предупреждает с помощью системного сообщения о невозможности открытия файла. Администратор вручную восполняет поврежденные данные за счет синхронизации с заранее созданной резервной копией БД на сервере с RAID 1. Резервная копия ежедневно создается автоматически при помощи System Center Data Protection Manager, формируя список контрольных точек для восстановления БД. При повреждении данных в БД на администраторской или пользовательской рабочей станции, восстановление происходит путем принудительной репликации данных с сервера на станции. При повреждении файлов модулей администратора или пользователя происходит процесс восстановления, аналогичный процессу восстановления администратором БД на сервере, при котором вручную осуществляется запуск cmd-файла на рабочей станции, файлы модулей синхронизируются с резервной копией и восполняются. При необходимости администратор может создавать резервную копию вручную на сервере, внешнем жестком диске или DVD. При использовании такого метода восстановления БД максимальная потеря данных составляет объем изменений, внесенных в БД за один рабочий день, что является не критичным, и может быть восполнена повторным внесением данных в БД вручную.

Специфика и динамика движения данных в рамках разных условий требует более наглядного моделирования процесса работы сетевой БД. Для визуализации процессов обработки данных можно использовать программное средство AllFusion Process Modeller: BPwin 4.1 от студии Computer Associates [3]. Моделирование обработки данных осуществляется с помощью нотации IDEF0©. Вся модель разбита на диаграммы разного уровня. Ее архитектура выглядит следующим образом: существует диаграмма верхнего уровня, которая декомпозируется на диаграммы нижних уровней, отражающие все происходящие процессы в родительской диаграмме [4].

Верхним уровнем является диаграмма «Работа базы данных ОП», она отражает процесс работы базы данных под воздействием на нее входящей информации и данных из других отделов предприятия и получившийся результат.

Эта диаграмма декомпозируется на три дочерние диаграммы: «Администрирование» (Модуль Администратора), «Хранение на файловом сервере» (Файловый сервер), «Обработка пользователем» (Модуль Пользователя). Здесь отображены все процессы, связанные тремя основными составляющими псевдо-сетевой базы данных (рис. 2).

1. Администратор получает финансовые данные (данные по закупкам, продажам, остаткам, бюджетам и т.д.), адаптирует их и реплицирует на сервер, на сервере сортирует, каталогизирует и архивирует их.
2. Сервер хранит полученные с администраторской станции данные, к которым обращаются пользователи для копирования на свои рабочие станции и изменения. Администратор также обращается к данным, хранящимся на сервере, для их восстановления или изменения.
3. Пользователь получает данные на свою рабочую станцию, просматривает их, а также копирует измененные файлы с данными с сервера к себе на жесткий диск.

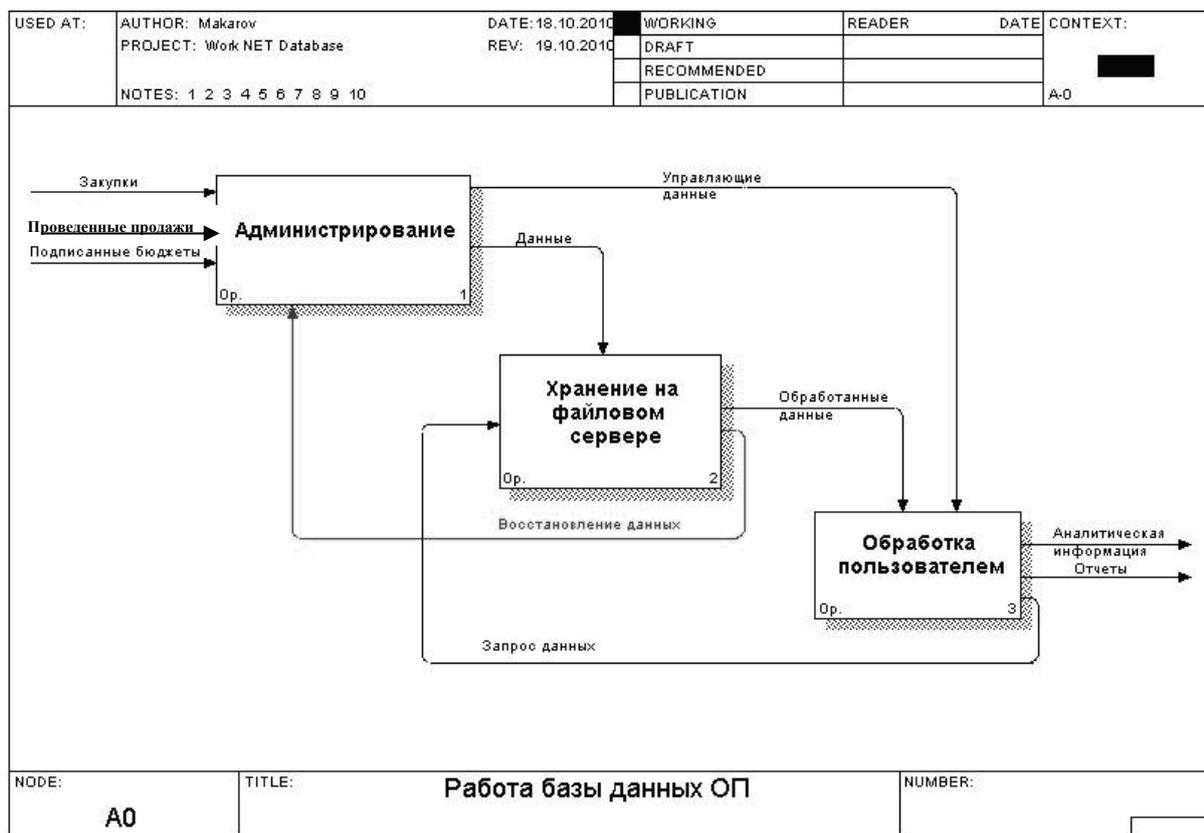


Рис. 2. Диаграмма верхнего уровня, отображающая миграцию данных

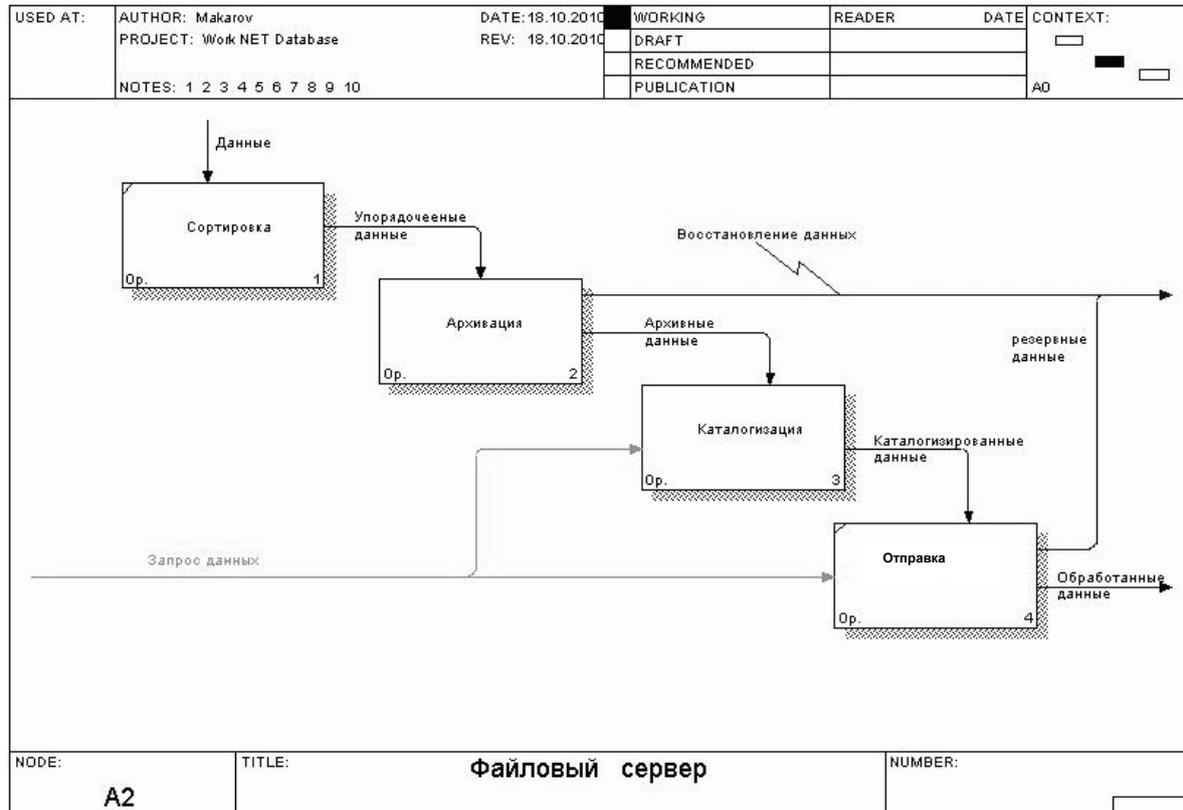


Рис. 3. Диаграмма процессов, происходящих на уровне сервера

Дальнейшая декомпозиция диаграмм «Администрирование», «Хранение на файловом сервере», «Обработка пользователем» (рис. 2) позволяет более полно смоделировать процессы, происходящие на конкретных узлах этой структуры БД. Рассмотрим схему диаграмм, получившуюся при декомпозиции диаграммы «Хранение на файловом сервере» (Файловый сервер) (рис. 3).

В этот узел данные мигрируют из административного модуля и хранятся на нем. Здесь осуществляется консистенция всех проблемных данных. Администратор может манипулировать данными на сервере – сортировать, архивировать и каталогизировать их. Также данные могут изменяться администратором и восстанавливаться путем репликации из административного модуля либо подготавливаться и отправляться на рабочую станцию пользователя по его запросу. Данные на этом узле системы могут использоваться параллельно пользователями в режиме чтения. Администратор обрабатывает данные в монопольном режиме. Доступ к этому узлу системы реализован при помощи групповых политик серверного программного обеспечения на основе Active Directory и распространяется только на участников группы отдела продаж и администратора.

За счет использования данного метода реализации БД, осуществляется довольно равномерное распределение плотности данных в этой конфигурации. Концентрация проблемных данных осуществляется на сервере, а управляющие элементы БД распределены между рабочими станциями пользователей и администратора, что позволяет эффективнее использовать ресурсы системы и сети, в которой она находится.

Данную модель работы псевдо-сетевой БД можно представить схематично в виде иерархии [5] (рис. 4).

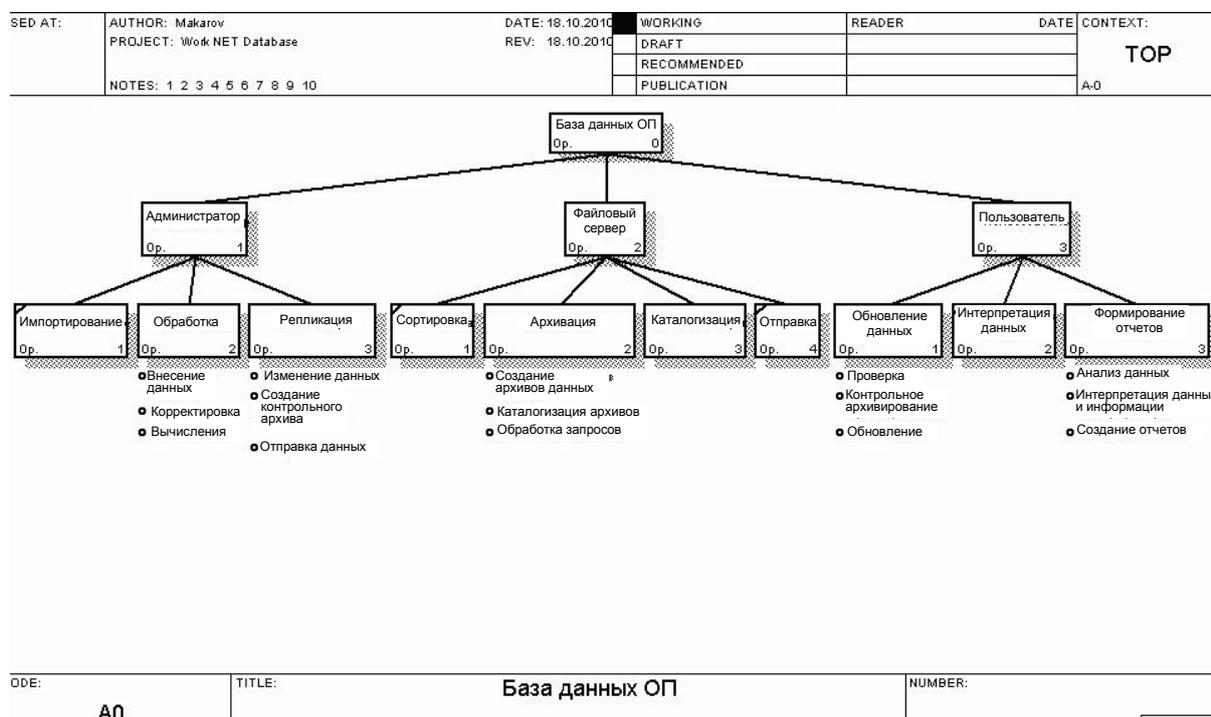


Рис. 4. Иерархия модели работы псевдосетевой базы данных

Результат применения

Смоделированный метод реализации псевдо-сетевой БД был применен в рамках отдела продаж малого предприятия ООО «РусХенк» – филиала, находящегося в Ленинградской области. В отделе используются 9 ПК: 8 пользовательских и администраторский компьютер, а также компьютер-сервер. Такая конфигурация сети позволяет оптимально воспользоваться этой моделью. Результатом ее применения стало следующее.

- Снижена нагрузка на физические файлы, содержащие данные и плотности распределения данных, что позволило увеличить скорость обработки информации.
- Снижена нагрузка на все рабочие станции сети отдела продаж за счет распределения нагрузки сохранения и обработки данных.
- Пользователь получает возможность изменять формы, отчеты и другие объекты в собственной клиентской базе данных, не влияя при этом на других пользователей, за счет использования пользовательского модуля, а также параллельно изменять данные в таблицах (определенных конкретным интерфейсом пользовательского модуля).

- Обеспечена мобильность данных за счет возможности работы пользователя с БД при отсутствии подключения к локальной сети (при использовании ноутбуков в качестве рабочих станций и работы вне отдела).
- Создан дополнительный уровень безопасности хранения данных. Теперь при выходе из строя любой из рабочих станций отдела данные остаются в полных резервных копиях на сервере и съемных оптических носителях.

Процесс моделирования позволил эффективно применить описанное решение в условиях отдела продаж, а использование продукта из наиболее распространенного и известного пакета программ – упростить освоение программы персоналом и ускорить адаптацию.

Заключение

Результатом исследования явилась модель распределенной БД и модель ее работы, которая отражает динамику данных и функционал БД. Это позволило проследить миграцию данных от входа в систему до передачи их конечному элементу – пользователю.

Суть подхода заключается в том, каждый пользователь работает со своей локальной частичной копией проблемных данных. При запуске пользовательского модуля идет проверка на необходимость синхронизации локально частичной копии данных с сервером. Пользователь также может принудительно инициировать проверку на необходимость синхронизации данных посредством пользовательского модуля, после чего данные синхронизируются с центральным хранилищем через процесс серверного типа, который монополюбно обрабатывает данные.

Полученная модель БД смоделирована с учетом специфики миграции данных в отделе продаж компании, использующей небольшую пользовательскую нагрузку на данные, но может применяться и для реализации других решений при определенной адаптации.

Литература

1. Кен Блюттман. Анализ данных в Access. Сборник рецептов. – СПб: Питер, 2008. – 352 с.
2. Microsoft Office // Справка и инструкции по Access [Электронный ресурс]. – Режим доступа: <http://office.microsoft.com/ru-ru/access-help/>, своб.
3. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. – М: Диалог-МИФИ, 2007. – 400 с.
4. Дубейковский В.И. Практика функционального моделирования с ALLFusion Process Modeler 4.1. Где? Зачем? Как? – М: Диалог-МИФИ, 2004. – 464 с.
5. SDTEAM.COM // Сравнительный анализ нотаций [Электронный ресурс]. – Режим доступа: <http://www.sdteam.com/?tid=6485>, своб.

- | | |
|--|---|
| Макаров Сергей Петрович | – Ленинградский государственный университет им. А.С. Пушкина, аспирант, immoral-idiom@rambler.ru |
| Бороненко Татьяна Алексеевна | – Ленинградский государственный университет им. А.С. Пушкина, доктор педагогических наук, профессор, зав. кафедрой, tataleks@mail.ru |
| Коробейников Анатолий Григорьевич | – Санкт-Петербургский филиал учреждения Российской академии наук «Институт Земного магнетизма, ионосферы и распространения радиоволн им. Н.В.Пушкова РАН», доктор технических наук, профессор, зам. директора, Korobeynikov_A_G@mail.ru |