УДК 629.12.004.5

Хнюнин С.Г.
ОНМА

ОРГАНИЗАЦИЯ ЭКСПОРТА-ИМПОРТА ДАННЫХ В ИНФОРМАЦИОННОЙ СИСТЕМЕ ТЕХНИЧЕСКОГО МЕНЕДЖМЕНТА СУДОВ

Для капиталоемкого предприятия важнейшей задачей является управление техническим обслуживанием и ремонтом производственных фондов. На судоходных предприятиях эта задача наиболее актуальна из-за территориальной удаленности производственных фондов. Издержки, связанные с техническим обслуживанием и ремонтом, составляют 20 – 30 % [1], занимая второе место после затрат на топливо. В этой связи на судоходных предприятиях уделяется большое значение снижению издержек на данные работы.

Для эффективного менеджмента в таких условиях руководству и специалистам компании для управления необходима информационная система. То есть нужен инструмент, который делал бы реально выполнимым сбор и анализ информации, обеспечивал оперативность и достоверность данных, предоставлял поддержку менеджмента при принятии решений, позволял оценивать эффективность этих решений и на основе их оценки вырабатывать корректирующие (предупреждающие) воздействия на бизнес-процессы. Если основной массив информации судов компании сведен в единую базу данных, то в любой момент времени руководители разного уровня могут использовать эту информацию, определить сбои в работе подразделений, связанных с техническим менеджментом судов вплоть до конкретного исполнителя, конкретной операции (работы). При этом исключено дублирование при вводе информации – информацию вносит один исполнитель, в обязанности которого входит именно эта операция.

Предприятия данного типа имеют распределенную систему. В распределенной базе данных оптимальным является использование различных СУБД на сервере и филиалах. Транспорт пакетов между узлами осуществляется по существующим каналам связи. При этом сокращается время отправки торговых документов (заявок, заказов), т.к. отправка ведется непосредственно из системы без перевода информации в промежуточные файлы. Полученные ответы автоматически попадают в систему.

В зависимости от технического оснащения объектов и территориального местонахождения связь сервера с узлами может осуществляться не только с помощью выделенного канала связи, но и другими

различными способами – от глобальной сети Internet до использования внешних носителей информации с применением обычной почты. В таком случае для обеспечения работоспособности таких распределенных систем необходимо произвести преобразование информации необходимой для передачи к стандартному электронному виду в виде отдельного выделенного файла. Ввод-вывод, хранение и передача, представленные на рис. 1, должны осуществляться в стандартных форматах между всеми участниками жизненного цикла.

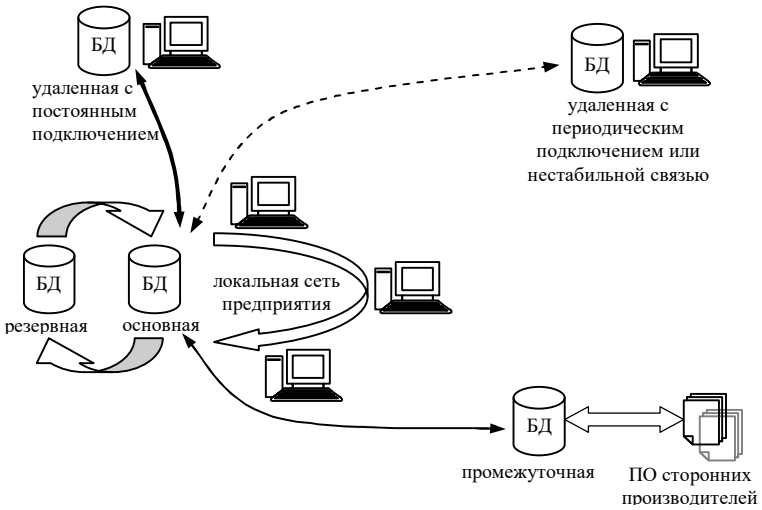

Рис. 1. Организация передачи информации в информационной системе

Помимо переноса данных из одной базы в другую в задачу модуля экспорта-импорта входит синхронизация баз – через определенные промежутки времени (то есть по расписанию) должны импортироваться все изменения в БД. Синхронизацию можно производить как по расписанию, так и по запросу пользователя. При этом импортировать можно как данные отдельных компонентов базы, так и целых разделов. Эта простая задача многократно усложняется в случае распределенной системы, не имеющей постоянной связи в режиме реального времени между территориально удаленными объектами [2]. Случай добавления или удаления информации довольно прост в обработке и синхронизации. Основной проблемой является отслеживание достоверности при модификации имеющихся данных произошедших между двумя сеансами связи. Причем именно в случае одновременного изменения в обеих БД. Кажущаяся на первый взгляд простой при-

вязка позволяющаяся задать приоритет той операции, которая была проведена первой по времени, невозможна по нескольким причинам. Первая, это невозможность провести синхронизацию часов между компьютерами. Точной настройке часов компьютера могут препятствовать как сбои оборудования (нередки ситуации сброса часов на материнских платах), так и недостаточная квалификация персонала работающего с программой (редкий пользователь ежедневно проводит проверку системного времени, а некоторые и не знают основ настройки операционной системы). Все это приводит к путанице при временном отражении записей. Другая причина состоит в том, что оба изменения записи могут быть достоверными, но существование их одновременно в одной базе невозможно. Данная проблема сложности синхронизации пропорционально возрастает между тремя и более копиями данных.

Решать данную задачу приходится комбинированием многих способов, каждый из которых решает свою часть данной проблемы.

1. При одновременном изменении одной и той же записи на разных узлах после синхронизации на всех узлах останется вариант более поздней модификации записи (определяется датой/временем модификации).

2. Поток данных всегда идет в направлении от головного офиса, а с филиалов идут только результаты работы. Тем самым нет необходимости производить анализ на идентичность данных.

3. В каждую оперативную таблицу добавляется код удаленного филиала. Таким образом, с каждого рабочего места можно читать любые данные, а писать (добавлять) только в свои.

4. Если при синхронизации в удаленном филиале не сделали вовремя обратную синхронизацию, то в следующую синхронизацию отсутствие изменения будет воспринято как новое обратное изменение. Требуется обязательное подтверждение прихода каждого отосланного изменения и в случае утери повторная пересылка заблаговременно сохраненного.

5. Каждая создаваемая транзакция нумеруется уникально в пределах одной копии. При синхронизации используется комбинированный индекс номер копии и номер транзакции. При синхронизации просматриваются все записи с номером, больше последнего имеющегося в своей базе данных чужого номера копии программы.

6. Физическое удаление и изменение записей подменяется логическим и окончательно редактируется лишь после разрешения всех конфликтов.

Также преобразования необходимо совершать для согласования и экспорта (импорта) данной информационной системы с системами

финансового и бухгалтерского учета. В этом случае желательно использовать промежуточную базу данных (например, SQL), через которую будет передаваться информация. Для использования промежуточной базы есть одна существенная причина – далеко не всегда другие системы разрешают прямое вмешательство в свои базы данных. В этом случае достаточно скомпоновать промежуточную базу данных, которую затем можно импортировать в другую систему посредством стандартных средств, предусмотренных в программном обеспечении.

В процессе экспорта-импорта данных важным моментом является слежение за целостностью данных. Целостность базы данных – соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам. При проектировании требуется наиболее полно выявить все имеющиеся ограничения целостности и задать их в базе данных. Целостность БД не гарантирует достоверности содержащейся в ней информации, но обеспечивает, по крайней мере, правдоподобность этой информации, отвергая заведомо невероятные, невозможные значения. Обеспечение целостности данных подразумевает наличие средств, позволяющих удостовериться, что информация в базе данных всегда остается корректной и полной. Целостность данных должна обеспечиваться независимо от того, каким образом данные заносятся в память (в интерактивном режиме, посредством импорта или с помощью специальной программы). Используемые в настоящее время СУБД обладают средствами обеспечения целостности данных и надежной безопасности. Поэтому СУБД может (и должна) контролировать целостность БД, но принципиально не в состоянии контролировать достоверность БД. Контроль достоверности БД может быть возложен только на человека, да и то в ограниченных масштабах, поскольку в ряде случаев люди тоже не обладают полнотой знаний о происходящих процессах.

Частным случаем применения модуля экспорта-импорта можно назвать архивирование, резервное копирование и последующее восстановление информации. Резервное копирование - один из самых надежных способов сохранить и предохранить свои данные от потери или порчи. При этом сохраняется текущая информация и структура таблиц базы данных в резервном источнике (файл с расширением .sql). В случае экстренных ситуаций (взломы, удаление информации, ошибки администратора и т.д.) из копии всегда можно восстановить базу данных в её прежнем состоянии. Поэтому резервное копирование базы данных рекомендуется делать регулярно.

Рассмотрим работу резервного копирования на примере программы Sypex Dumper [3], созданной специалистами украинской компании "Биноватор" (версия Lite распространяется по лицензии GNU GPL,

т.е. является абсолютно бесплатной). При создании резервной копии требуется знать и задать основные параметры работы, представленные на интерфейсе рис. 2.

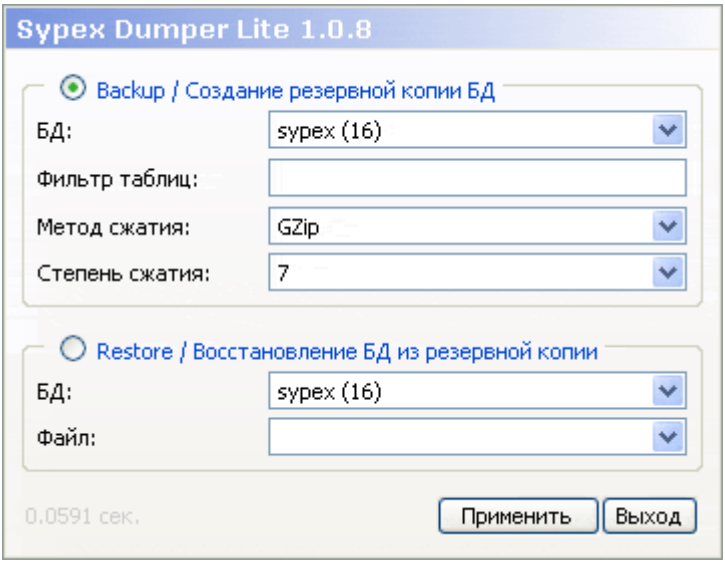

Рис. 2. Главная форма программы Sypex Dumper Lite
при создании резервной копии БД

Данная форма дает возможность делать тонкие настройки для управления процессом создания резервных копий баз данных или отдельных таблиц. В процессе работы необходимо задать имя базы, для которой будет производиться резервное копирование. Следующим этапом будет выбор одной или нескольких таблиц для работы. Если требуется сделать копию всей базы данных, то этот пункт оставляется пустым. Однако нужно понимать, что в данном случае время работы программы может занять значительное время. Наиболее оптимальным будет проводить резервное копирование только тех таблиц, которые подверглись изменениям в недавнем прошлом.

Следующим этапом предстоит выбор метода архивирования и степени сжатия. Выбор стандартного метода архивирования (например, zip ставшим де-факто в настоящее время) позволит избежать "недоразумений" при обратных действиях в случаях, если процесс будет осуществляться другим программным обеспечением. В случае, если такие действия не планируются, то лучше применить один из методов

архивирования, разработанных позднее и имеющих лучший алгоритм архивации, но не получивших столь высокое распространение в мире (lharc, lha, ice, ain, rar, pak, zoo, expand). Высокую степень сжатия имеет смысл устанавливать в случае, если планируется пересылка копии на значительное расстояние – время создания получается максимальным. В случае резервного копирования с последующим локальным хранением результата целесообразно выбирать наименьшее сжатие, при котором действие занимает наименьшее время и простой БД минимизируется. Результат работы программы представлен на рис. 3.

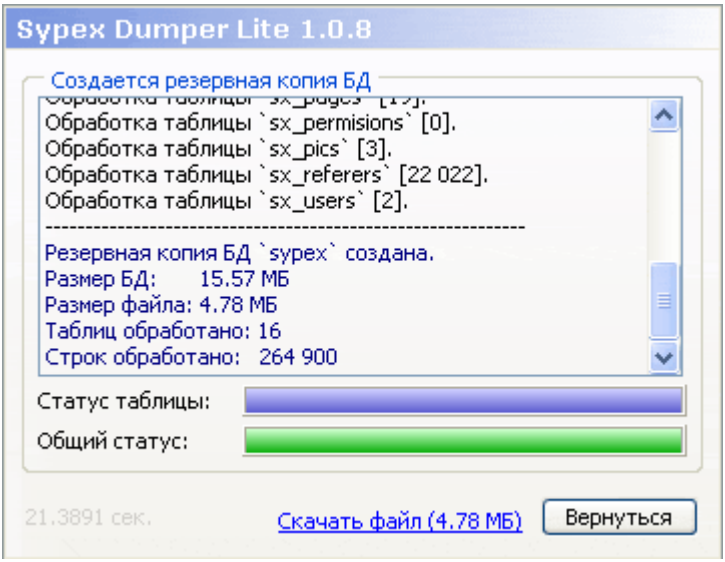

Рис. 3. Вывод результата процесса создания резервной копии БД

В данном случае на экран последовательно выводится список обрабатываемых таблиц и количество записей в каждой. В конце отчета приводится общая информация об обработанных объектах и занимаемый объем оригинальной БД и архивной сжатой копии. В обратном случае, если необходимо восстановить БД из резервной копии, требуется лишь задать имя архивного файла и наименование восстанавливаемой БД (см. рис. 2). Процесс восстановления аналогичен показанному на рис. 3 – на экран выводится список восстанавливаемых таблиц.

Процесс резервного копирования также необходимо делать в профилактических целях, для увеличения производительности базы дан-

ных - это достигается за счет того, что в момент копирования происходит считывание последних версий всех записей, старые же версии в копию никогда не попадают. Здесь важно заметить, что недостаточно одного лишь резервного копирования, нужно периодически проверять восстанавливаемость базы данных из резервной копии, потому что бывают случаи, что база данных работает в режиме 24 часа в сутки и 7 дней в неделю, процесс резервирования базы данных может происходит нормально, но в силу определенных причин база данных не восстанавливается, последствия могут быть плачевными для всех данных. Для профилактики следует восстанавливать базу данных в тестовую, и лишь при успешном завершении процесса восстановления, делать запись в текущую базу. В случае возникновения ситуации с поврежденным файлом следует найти в базе данных несоответствия и исправить их.

Таким образом проанализирована организация экспорта-импорта данных в случае распределенной системы, не имеющей постоянной связи в режиме реального времени между территориально удаленными объектами. Предложена технология реализации данного процесса для синхронизации удаленных баз данных, обеспечения целостности данных, резервного копирования и архивирования.

СПИСОК ЛИТЕРАТУРЫ

1. Антоненко И.Н. Автоматизация технического менеджмента судоходной компании "Волга-флот". // Судоходство. – 2005. - №1-2. – С. 48 – 49.

2. Коннолли Томас, Бегг Каролин. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание. : Пер. с англ. – М.: Издательский дом "Вильямс". – 2003. – 1440 с.

3. Sypex Dumper – бекап и восстановление БД MySQL [Электронный ресурс] – http://sypex.net/.