

АНУФРИЕВ Владимир Натанович

**ИНТЕГРАЦИЯ РЕШЕНИЙ НА
ПЛАТФОРМЕ “1С” В ГЕТЕРОГЕННУЮ
КОРПОРАТИВНУЮ СРЕДУ.**

**РАСКОДИРОВАНИЕ МЕТАДААННЫХ.
ПРЯМЫЕ ОБРАЩЕНИЯ К БАЗЕ ДАННЫХ
ПОСРЕДСТВОМ ЯЗЫКА SQL**

Москва
2010

Vladimir Anufriev

1C Platform in Complex Corporate Environment

Comments may be addressed to: avn@avnsite.com

When you send information to AVN, you grant AVN a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright by Vladimir Anufriev 2010. All rights reserved.

Введение

Платформа «1С» является безусловным шедевром отечественного опыта проектирования и создания массовых программных продуктов. В классе средств автоматизации бухгалтерского учета и отчетности для малых и средних предприятий в мире пока не создано ничего подобного. Это подтверждают наши коллеги, ныне вынужденные сопровождать аналогичные системы в Северной Америке, где конкуренция среди качественных программных решений наиболее высока.

Основной технической особенностью платформы является возможность совмещать эксплуатацию системы с ее модификацией и доработками с помощью удобной, бизнес-ориентированной интегрированной среды с максимально унифицированным графическим интерфейсом пользователя и разработчика. То есть система изначально проектировалась как предельно гибкая и относительно легко настраиваемая на любой сектор промышленности и услуг, требующий автоматизации операций, связанных с учетом и отчетностью.

Наряду с бесспорными достоинствами у системы есть органические недостатки, затрудняющие ее интеграцию в гетерогенную корпоративную среду. Некоторые из этих недостатков и возможные пути их обхода рассмотрены в настоящей статье.

Особенности архитектуры платформы

Фирмой «1С» было реализовано решение, в котором структура базы данных конкретного приложения, формы, отчеты и алгоритмы обработки хранятся в базе данных вместе с самими данными¹. В случае модификации метаданных (конфигурации) автоматически модифицируется (реструктурируется) и база данных.

Как это часто бывает, стремление достичь максимума в одном или сразу нескольких направлениях, приводит к затруднениям в других. И у платформы-лидера рынка отечественных бухгалтерских систем есть некоторые недостатки, среди которых для целей настоящей статьи хочется выделить следующие²:

- 1) система тяжело масштабируется, несмотря на специально реализованную для решения этой проблемы в версии 8.1 трехуровневую архитектуру (клиент - сервер приложений - СУБД) и появившуюся в версии 8.2 четырехуровневую архитектуру (см. рис. 1); в связи с этим, применение системы на крупных предприятиях представляется затруднительным, хотя четырехуровневая архитектура в значительной степени решает проблему поддержки распределенных рабочих мест³;
- 2) система предъявляет повышенные требования к пропускной способности сети, что неявно признается разработчиком, который, как нам представляется,

1 Автор не является сотрудником Фирмы «1С», никогда в ней не работал и не пользуется внутренней неопубликованной информацией этой компании. В данной работе используется информация, полученная из открытых источников и в результате собственных исследований системы в ходе ее боевой эксплуатации в нескольких организациях и с различными конфигурациями как типовыми так и разработанными под заказ.

2 Здесь и далее без специальных оговорок идет речь о версии платформы 8.1 с хранением данных в реляционной СУБД. В настоящее время текущей версией платформы является 8.2.9. У автора не было возможности проверить все излагаемое для последней версии, но при переходе с версии 8.0 на 8.1 в рассматриваемом аспекте система существенных изменений не претерпела.

3 При наличии у системы жесткой структуры данных в большинстве случаев достаточно было бы двухуровневой архитектуры (клиент - СУБД), как у основной массы эксплуатируемых в настоящее время автоматизированных банковских систем (АБС)

- пытается решить эту проблему выпуском тонкого клиента⁴ и веб-клиента⁵;
- 3) система не имеет API (Application Programming Interface) и единственным документированным средством для экспорта и импорта информации для нее



Рис. 1. Четырехуровневая архитектура платформы 1С версии 8.2 (схема с сайта фирмы «1С»)

- являются текстовые XML файлы⁶; для визуализации данных и при определенных усилиях для экспорта можно использовать веб-сервисы;
- 4) экспорт и импорт данных на сервере в пакетном режиме через XML файлы настраивается непросто даже при передаче информации между системами на базе одной платформы, несмотря на наличие механизма так называемых «планов обмена»;
- 5) синхронизация справочников как между системами на базе той же платформы 1С, а тем более со сторонними системами представляет из себя трудноразрешимую проблему⁷;
- 6) доступ к данным привычными средствами SQL запросов недокументирован и по причинам, описываемым ниже, без специальных ухищрений может быть опасен для целостности данных; в то же время обмен данными с помощью

4 http://v8.1c.ru/overview/Term_000000124.htm#1

5 http://v8.1c.ru/overview/Term_000000125.htm#1

6 На самом деле файлы обмена только внешне выглядят как XML, но строго спецификации не соответствуют, описатели DTD также отсутствуют

7 Во избежание рассинхронизации справочников приходится внедрять жесткие регламенты и постоянно отслеживать изменения. Импорт данных из внешних систем, в частности «клиент-банка» и чаще «1С:Зарплата и Управление Персоналом», все равно в некоторых случаях может приводить к дублированию сущностей. Согласование «разъехавшихся» справочников представляет собой нетривиальный ручной процесс.

- документированных средств при росте объема хранимых данных быстро становится неприемлемым по временным параметрам;
- 7) обращение к данным невозможно оптимизировать средствами СУБД, так как структура хранения данных недокументирована и изменяется вместе с метаданными при обновлении конфигурации;
 - 8) система не использует средства обеспечения целостности и встроенные механизмы оптимизации СУБД, главным образом в связи с отсутствием жесткой структуры данных, некоторого отступления от реляционной модели, а также дополнительно ради поддержки целого ряда СУБД⁸.

Далее рассмотрим подход, позволяющий в определенных случаях решить проблемы, отмеченные выше в пп. 4-7. В частности, предлагаемая технология позволяет за минимальное время извлекать данные для последующей обработки в хранилище данных, а также автоматически заполнять справочники, объекты которых не содержат или содержат минимальное количество логических ссылок на другие объекты базы данных.⁹

Идея метода

В условиях отсутствия у системы API и затруднениях при обмене через XML не остается другого способа, как попытаться с соблюдением необходимых предосторожностей раскрыть структуру данных активной конфигурации и обращаться к данным с помощью стандартного языка запросов SQL.

Беглого взгляда на структуру данных любой рабочей конфигурации достаточно, чтобы признать, что без полного разбора метаданных невозможно средствами СУБД понять, в каких объектах базы данных (полях, таблицах) и в каком формате хранятся какие данные. Большинство объектов базы данных имеют абстрактные имена, а метаданные вообще выглядят зашифрованными.

Однако, после внимательного изучения данных и нахождения определенных закономерностей можно написать программу, которая будет создавать структуру представлений (view) с читаемыми и привычными для программистов «1С» именами и понятным составом полей, которые будут служить расшифровками соответствующих рабочих таблиц, для которых мы добавим в качестве комментариев их имена и имена их полей, в остальном оставив в неприкосновенности. Основной сложностью будет то, что разработчики конфигураций не задумываясь используют очень длинные русские имена объектов, а все СУБД так или иначе ограничивают длину идентификаторов в языке SQL. Кроме того в «1С» объекты разных классов могут иметь одинаковые имена, поэтому приходится использовать префиксы, чтобы добиться уникальности имен представлений.

Платформа «1С» версии 8.1 предусматривает 12 классов прикладных объектов (внутри системы они именуются латиницей):

```
accounts
account registers
accumulation registers
documents
document journals
enumerations
calculation kinds
```

⁸ Фирма «1С» в версии платформы 8.2 заявляет о поддержке четырех распространенных СУБД: Oracle, MS SQL, PostgreSQL и IBM UDB

⁹ Примерами таких справочников могут быть котировки валют или ценных бумаг

```
chrc
information registers
nodes
references
sequences
```

Поскольку как исходные данные так и результаты работы программы располагаются в одной стандартной базе данных, мы можем, теоретически использовать любой язык программирования, позволяющий манипулировать данными в реляционной СУБД.

Для реализации разбора метаданных мы выбрали язык Java, как платформенно-независимый. Для упрощения процедур запуска мы вызываем эту программу из скриптов на языке Shell.

Запустив нашу программу, указав ей имя базы данных с нужной конфигурацией, получим в случае успеха приблизительно следующие сообщения¹⁰:

```
./create_metadata.sh

Database: gl
URL: jdbc:postgresql:gl
Username: postgres
Password:
/usr/bin/java -jar md.jar org.postgresql.Driver jdbc:postgresql:gl postgres
password
CREATE FUNCTION
ALTER FUNCTION
  droplcvIEWS
-----

(1 record)

DROP TABLE
DROP TABLE
DROP TABLE
Driver loaded: org.postgresql.Driver@82701e
Connected to jdbc:postgresql:gl as user postgres
Create db objects table... 6305 objects to read estimated
done
Processing accounts... done
Processing account registers... done
Processing accumulation registers... done
Processing documents... done
Processing document journals... done
Processing enumerations... done
Processing calculation kinds... done
Processing chrc... done
Processing information registers... done
Processing nodes... done
Processing references... done
Processing sequences... done
Create db schema table and scripts... 865 tables were found in configuration...
done
Create comments... done
Build views... done
Congratulations!
```

¹⁰ Здесь приведен реальный вывод при обработке одной из версий конфигурации «1С:Бухгалтерия»

```
CREATE TABLE _reference60
(
  _idrrref bytea NOT NULL,
  _version integer NOT NULL,
  _marked boolean NOT NULL,
  _ismetadata boolean NOT NULL,
  _parentidrrref bytea NOT NULL,
  _folder boolean NOT NULL,
  _code mchar(9) NOT NULL,
  _description mvvarchar(96) NOT NULL,
  _fld684 mvvarchar NOT NULL, -- Описание
  _fld685 mvvarchar, -- Формула содержания
  _fld686 mvvarchar, -- Формула суммы операции
  _fld687 mvvarchar, -- Алгоритм перед вводом
  _fld688 mvvarchar, -- Алгоритм при вводе
  _fld689 mvvarchar, -- Алгоритм после ввода
  CONSTRAINT _reference60_pkey PRIMARY KEY (_idrrref)
)
WITH (OIDS=FALSE);
ALTER TABLE _reference60 OWNER TO v8;
COMMENT ON TABLE _reference60 IS 'Типовые операции';
COMMENT ON COLUMN _reference60._fld684 IS 'Описание';
COMMENT ON COLUMN _reference60._fld685 IS 'Формула содержания';
COMMENT ON COLUMN _reference60._fld686 IS 'Формула суммы операции';
COMMENT ON COLUMN _reference60._fld687 IS 'Алгоритм перед вводом';
COMMENT ON COLUMN _reference60._fld688 IS 'Алгоритм при вводе';
COMMENT ON COLUMN _reference60._fld689 IS 'Алгоритм после ввода';
```

Рис. 2. Описание рабочей таблицы "Типовые операции"

Открыв теперь в административном интерфейсе произвольную рабочую таблицу мы увидим, что программа прокомментировала большинство ее полей. На рисунке 2 представлен вид таблицы, техническое название которой `_reference60`. На самом деле эта таблица содержит «Типовые операции». Часть полей имеет predeterminedенные имена, их назначение для специалистов по «IC» более-менее понятно из названий. Однако, использовать такую даже прокомментированную таблицу напрямую невозможно, так как при следующем изменении конфигурации как она сама так и ее поля могут поменять свои имена.

Из описания таблицы видно, что платформа использует первичные ключи. Внешние же ключи использовать невозможно, так как при отсутствии объекта, на который есть ссылка, эта ссылка не устанавливается в значение NULL, а содержит predeterminedенное значение GUID¹¹. Здесь, похоже, проявляется дефект первоначального проекта, рассчитанного на поддержание собственной файловой СУБД. Кроме того, модель структуры данных отличается от реляционной, так как поле может ссылаться на разные объекты.

А вот связанное с рабочей таблицей представление (view) уже можно смело использовать для извлечения данных. Посмотрим на пример такого представления для другой таблицы, представленный на рисунке 3.

В данном представлении семь первых полей являются служебными, а остальные, очевидно, созданы с помощью встроенного в платформу интерактивного средства разработки и модификации приложений, которое называется Конфигуратор.

После изменения конфигурации нужно еще раз запустить нашу программу и

11 <http://ru.wikipedia.org/wiki/GUID> – Globally Unique Identidier

представление будет перепривязано к рабочей таблице. Таким образом, мы отстраиваемся от изменений технических названий таблиц и полей и работаем с постоянными именами представлений и их полей. Разумеется они тоже могут меняться при изменении функциональности соответствующих объектов, но это уже обычное дело для любых приложений.

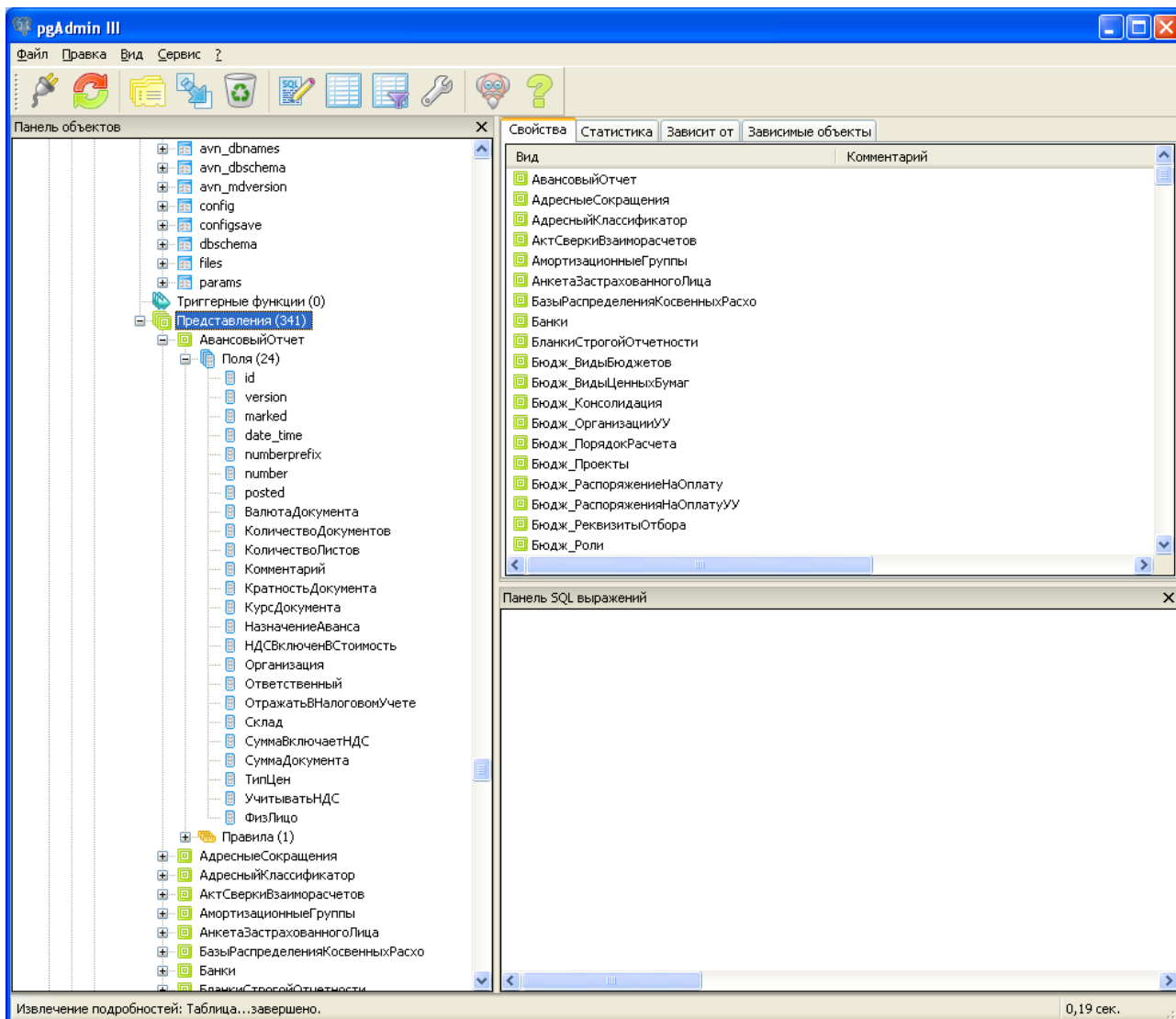


Рис. 3. Представление "Авансовый отчет"

Выборка данных после получения понятных человеку представлений становится тривиальной задачей. Запись одновременно в одну таблицу также тривиальна для всех СУБД, кроме PostgreSQL, которая не поддерживает редактируемых представлений¹². Для этой СУБД можно в процессе разбора метаданных дополнительно создавать правила RULE, а можно искать нужную таблицу динамически, как показано в следующем примере:

¹² Все разработанные программы проверялись в среде Linux OpenSuse 10.x и СУБД PostgreSQL 8.2.5.


```

CREATE OR REPLACE FUNCTION avn.load_micex_quotes( in quote varchar )
  RETURNS void AS
$BODY$
DECLARE
  inforeg TEXT;
  statement TEXT;
BEGIN
  -- Здесь мы определяем имя таблицы 1С, связанной с view 'иКотировки'
  -- Внутренние имена объектов конфигурации 1С могут изменяться при каждом
  -- изменении этой конфигурации
  SELECT table_name INTO STRICT inforeg FROM information_schema.view_table_usage
    WHERE view_schema = 'avn' AND view_name = 'иКотировки';

  statement := 'insert into '||inforeg||
  ' SELECT a.date::timestamp as period,
    b.id as "ЦеннаяБумага",
    c.id as "Биржа",
    COALESCE(a.waprice, ''0'')::numeric as Average,
    0 as average_ch,
    COALESCE(a.low, ''0'')::numeric as min_price,
    COALESCE(a.high, ''0'')::numeric as max_price,
    COALESCE(a.open, ''0'')::numeric as open_bid,
    COALESCE(a.open, ''0'')::numeric as open_ask,
    0 as bid,
    0 as ask,
    COALESCE(a.vol, ''0'')::numeric as volume,
    0 as trades_num,
    0 as usd_volume,
    COALESCE(a.close, ''0'')::numeric as last_price,
    0 as Nkd,
    COALESCE(a.waprice, ''0'')::numeric as "ПризнаваемаяКотировка",
    COALESCE(a.waprice, ''0'')::numeric as "РыночнаяЦена",
    COALESCE(a.waprice, ''0'')::numeric as "Оценка",
    COALESCE(a.waprice, ''0'')::numeric as "РыночнаяЦена2",
    d.id as "ВалютаКотировки"
  FROM avn.micex_quotes a, avn."сЦенныеБумаги" b, avn."сБиржи" c, avn."сВалюты" d
  WHERE mvarchar_in(textout(a.ticker)) = b."ДопКод1"
    AND c.code = 'MICEX'
    AND d.code = '810'
    AND a.waprice IS NOT NULL
    AND a.waprice <> 0
    AND a.vol IS NOT NULL
    AND a.ticker = '||$1||'
    AND a.date > (SELECT COALESCE( MAX( e.period ), '2003-12-31' )::date
      FROM avn."иКотировки" e, avn."сБиржи" f, avn."сЦенныеБумаги" g
      WHERE f.code = 'MICEX' AND e."Биржа" = f.id AND g.id=e."ЦеннаяБумага" AND
g."ДопКод1" = '||$1||')';
  EXECUTE statement;
END;
$BODY$
LANGUAGE 'plpgsql' VOLATILE;

```

В вышеприведенном примере происходит заполнение справочника (информационного регистра) «Котировки» данными, выбираемыми из дополнительной таблицы, в которую они заливаются специальным агентом, стартующим по расписанию. Структура справочника взята из одной из версий конфигурации «1С-Рарус:Паевые инвестиционные фонды, редакция 2». В имени представления появляется префикс «и», показывающий, что речь идет об информационном регистре.

Выводы и рекомендации

Рассмотренный выше подход был реализован и прошел успешную эксплуатацию на нескольких типовых и заказных конфигурациях на платформах «1С» версий 8.0 и 8.1. Он позволяет подступиться к решению некоторых проблем, описанных в разделе «Особенности архитектуры платформы».

Экспорт и импорт данных (проблема N 4) после создания набора представлений можно осуществлять в пакетном режиме любыми программными средствами. Правда, при необходимости сложной обработки невозможно задействовать алгоритмы, хранящиеся в метаданных, так как исполнительная система платформы «1С» из внешних средств пока недоступна. Всю дополнительную алгоритмическую нагрузку, если она необходима, придется перепрограммировать на удобном разработчику импортно-экспортных процедур языке.

Синхронизацию справочников (проблема N 5) также можно выполнять с помощью SQL, однако при этом нужно учитывать все возможные внешние ссылки, которые система поддерживает только на прикладном уровне.

Привычная работа с базой данных посредством SQL (проблема N 6) становится доступной, но серьезную обработку написать довольно сложно из-за множества внутренних недокументированных соглашений, которые нужно учитывать.

Становится возможна некоторая оптимизация доступа к данным (проблема N 7), несмотря на сохраняющуюся невозможность использования внешних ключей. Получение сложных отчетов можно полностью переписать, хотя более правильным решением является экспорт оперативных данных в хранилище и получение отчетности уже в нем.

В ходе эксплуатации необходимо помнить, что платформа «1С» не рассчитана на какие-либо изменения или дополнения в стандартной схеме public. Мы специально вынесли все дополнительные объекты в отдельную схему. При обновлении конфигурации любая ошибка при выполнении операции в БД, связанная, например, с наличием зависимых объектов, приводит к автоматической инвалидации всей БД и требованию восстановить БД с резервной копии. Поэтому, перед любыми манипуляциями в Конфигураторе необходимо удалить все дополнительные объекты. После завершения работы в Конфигураторе и сохранения новой конфигурации объекты нужно создать заново тем же способом, что и в предыдущий раз.

Таким образом, предлагаемая технология позволяет ускорить ввод и вывод данных из «1С» в сотни раз и решить часть проблем масштабирования, снизить количество ручного ввода данных, автоматизировать перенос данных из других систем и естественным образом встроить базы данных «1С» в гетерогенную корпоративную среду.

Владимир Ануфриев
2010-03-11

