

УДК 61:004.652

 10.25209/2079-3316-2022-13-4-77-91

Миграция с Oracle на PostgreSQL сильно связанной МИС Интерин

Владимир Леонидович **Малых**^{1✉}, Алексей Николаевич **Калинин**²

^{1,2} Институт программных систем им. А. К. Айламазяна РАН, Веськово, Россия

^{1✉} mvl@interin.ru

(подробнее об авторах на с. 90)

Аннотация. Во многих сферах прикладного программного обеспечения наблюдается тенденция к более широкому использованию свободного программного обеспечения (СПО). Эта тенденция связана с переходом к более экономичной модели создания и эксплуатации ПО, с повышением безопасности использования ПО, исключением контроля проприетарного ПО за пользователями и их данными, с преодолением различных санкций в области проприетарного ПО. Для России становится актуальным отказ от проприетарного ПО Oracle и переход к СПО PostgreSQL. В статье предлагается подход к решению проблемы миграции с Oracle на PostgreSQL сильно связанного прикладного ПО. Подход отрабатывался на примере миграции МИС Интерин. Результаты могут быть использованы разработчиками ПО, решающими задачи миграции на СПО. (*see abstract in English on p. 91*)

Ключевые слова и фразы: медицинская информатика, медицинские информационные системы, миграция, Oracle, PostgreSQL

Для цитирования: Малых В.Л., Калинин А.Н. *Миграция с Oracle на PostgreSQL сильно связанной МИС Интерин* // Программные системы: теория и приложения. 2022. Т. 13. № 4(55). С. 77–91. http://psta.pstiras.ru/read/psta2022_4_77-91.pdf

Введение

Переход на свободное программное обеспечение (СПО) актуален для всего мира. Причины такого движения хорошо сформулированы Ричардом Столлманом (*Свободное программное обеспечение и стандартная общественная лицензия GNU*)^[URL]. Если для Столлмана главное – это «убежать» из Оруэлловского мира тотального контроля, то для государственных органов России главное – это безопасность, независимость и отсутствие контроля со стороны наших потенциальных противников. Использование качественного западного проприетарного ПО позволяет создавать качественные прикладные продукты, но при этом возникают и значительные риски контроля за поведением и данными пользователей, а также имеются риски полного отказа проприетарного ПО.

В связи с санкционной политикой Запада по отношению к России, в том числе и в сфере ПО, становится актуальным использование свободного ПО с открытым кодом. В особенности это касается ПО, которое пока сложно заменить на отечественные разработки (реляционные СУБД и др.).

Государство поддерживает такое движение (*Свободное программное обеспечение в госорганах*)^[URL]. Комиссия Госдумы по развитию стратегических информационных систем, действующая при Комитете Госдумы по науке и наукоемким технологиям, на своем заседании в сентябре 2015 года сообщила о том, что приступила к разработке законопроекта, который предоставит преимущества СПО перед проприетарными продуктами на закупках по 223-ФЗ и 44-ФЗ.

Как отмечает CNews (*После ухода из России Microsoft, Oracle и SAP бешено вырос спрос на отечественное ПО*)^[URL]: «После того, как свою деятельность на российском рынке приостановили зарубежные компании Microsoft, Oracle, SAP и др., спрос на отечественные решения вырос на 300%. Больше всего заказов приходится на офисные пакеты, ОС и системы управления базами данных. К российским компаниям, которые производят такие классы ПО, относятся <...> и PostgreSQL Professional.»

Переход на СПО является общемировой тенденцией и позволяет сокращать расходы на проприетарное закрытое ПО. В частности, для такой категории ПО как СУБД является актуальным переход с Oracle на PostgreSQL. Выбор именно PostgreSQL как альтернативы Oracle обусловлен рейтингом этой СУБД, относящейся к СПО (*Самые популярные базы данных – 2006–2021 гг*)^[URL]. В рейтинге (*TOPDB Top Database index*)^[URL], анализирующем относительную частоту присутствия

имени БД в запросах Google, PostgreSQL находится на 5 месте. В рейтинге (*DB-Engines Ranking - May 2021*)^{[URL](#)} PostgreSQL находится на 4 месте. Более высокие места в указанных рейтингах занимают проприетарные СУБД.

Медицинская информационная система (МИС) Интерин PROMIS (разработчик ООО «Интерин технологии») имеет уже 30-летний возраст жизни (*Подробная информация о медицинской информационной системе Интерин представлена на сайте с указанием внедренных проектов*)^{[URL](#)}. В своем технологическом развитии МИС прошла два этапа согласно представлению об эволюции IT архитектур из [1]. Технологически в настоящее время архитектура МИС относится к 5-му поколению (сервер хранения данных, сервер приложений, сервер интерфейса пользователя, браузер). МИС в настоящее время разворачивается на новой платформе Интерин IPS [2]. Исторически МИС была Oracle-центрической системой, ее ядром является СУБД Oracle. Существенной особенностью начальной архитектуры системы, относящейся к 4 поколению (клиент-сервер), был большой объем серверного кода на языке PL/SQL, реализующего значительную часть бизнес-логики системы. При переходе на архитектуру следующего этапа весь этот код остался в СУБД, так как его перенос на сервер приложений виделся весьма трудоемким и рискованным. В настоящее время государственная политика России в области ПО и санкционная политика Запада и вынуждают разработчиков МИС Интерин PROMIS отказаться от проприетарной СУБД Oracle и перенести МИС на PostgreSQL.

Основной проблемой миграции МИС Интерин с Oracle на PostgreSQL является значительный объем мигрирующего серверного кода. Ниже будут приведены данные по числу переносимых серверных объектов для одной из подсистем МИС Интерин PROMIS. Процесс миграции оказался невозможно выстроить как простой акт пересоздания структур данных в СУБД PostgreSQL и переноса данных. Процесс переноса системы сильно растянут во времени и скорее напоминает процесс сборки новой системы из новых компонентов. Можно привести аналогию с материальным миром, разборка самолета Sukhoi SuperJet 100, замена всех импортных компонентов на отечественные и повторная сборка. Что-то подобное требуется сделать для ПО МИС Интерин PROMIS.

Для решения проблемы перехода с Oracle на PostgreSQL сложных сильно связанных систем ПО необходимо разработать и предложить IT сообществу программное обеспечение для поддержки данного перехода. Далее в статье предлагается подход к решению указанной проблемы.

1. Основная идея построения процесса миграции

Процесс миграции рассматривается как перенос различных серверных объектов из Oracle в PostgreSQL. Объекты в ходе переноса преобразуются (конвертируются) и оптимизируются. Объекты сильно связаны друг с другом (зависят друг от друга), поэтому подготовка объектов и их миграция отвечает некоторой стратегии миграции и ведется в определенном порядке.

Строится сеть взаимосвязанных объектов. В первую очередь в сеть включаются объекты СУБД Oracle. Объекты получают рейтинги, характеризующие прямую и косвенную зависимость объектов друг от друга. Для построения сети используется лексический анализ DDL (data definition language) кода объектов, также используется словарь БД Oracle. В сеть могут быть также добавлены объекты, реализующие прикладной пользовательский интерфейс (ресурсы, формы). Язык реализации пользовательского интерфейса может быть любым высокоуровневым языком программирования (PL/SQL, Java, JavaScript и другие). Благодаря рассчитываемому по сетевым связям рейтингу объекты могут быть упорядочены по степени важности, что позволяет сформировать порядок реализации и миграции объектов. ПО реализующее указанную идею далее будем называть Интерин O2P.

Наиболее близким к Интерин O2P является продукт Ora2Pg – Oracle to PostgreSQL database schema converter (*Конвертор разработан Gilles Darold, как свободное ПО, стоимость разработки авторы на момент подготовки публикации оценивали в \$ 729,557*)^{URL}.

Продукт Ora2Pg имеет достаточно длительную историю развития – 20 лет. Он схож по своему функционалу с Интерин O2P. Продукт может выполнить миграцию тех же объектов, что и Интерин O2P за исключением объектов, связанных с интерфейсными модулями. Трансляция или конвертация функционального кода в Ora2Pg также ограничена – «Provide some basic automatic conversion of PLSQL code to PLPGSQL». При этом Ora2Pg является свободно распространяемым продуктом.

Ключевым преимуществом Интерин O2P перед Ora2Pg является более гибкий и управляемый процесс миграции. Ora2Pg рассчитан на автоматическую миграцию объектов целыми схемами. Интерин O2P строит модель миграции, предполагает ревизию и оптимизацию мигрирующих объектов, ранжирует объекты и рекомендует последовательность их реализации, позволяет включать в объекты миграции интерфейсные модули, позволяет распределять объекты по ответственным лицам (синонимы: пользователи, программисты),

контролирует протекание процесса миграции и текущие статусы объектов.

Интерин O2P лучше подходит для миграции сильно связанного ПО с большим количеством серверного функционального кода на PL/SQL. В этом случае процесс миграции становится достаточно длительным и возникает необходимость в его организации, в определении последовательности переноса и реализации объектов миграции, в привязке объектов миграции к ответственным исполнителям, в контроле текущего состояния процесса миграции.

Интерин O2P и Ora2Pg могут применяться совместно синергетически усиливая друг друга, см. таблицу 1.

Таблица 1. Сравнение Интерин O2P с наиболее близким аналогом по характерным ключевым параметрам

№	Основные характеристики продукта	Ora2Pg	Интерин O2P
1	Экспорт объектов схемы БД Oracle (таблицы, представления, последовательности, индексы, ограничения)	+	+
2	Выборочный экспорт таблиц	+	+
3	Экспорт функций, триггеров, процедур и пакетов	+	+
4	Экспорт полных табличных данных или выбранных с условием WHERE	+	+
5	Базовая автоматическая конвертация PL/SQL кода в PLPGSQL	+	+
6	Построение модели миграции и поддержка процесса миграции	—	+
7	Поддержка конвертации внешних модулей в части их работы с БД	—	+

2. Программное решение

Идея Интерин O2P основана на формировании сети взаимосвязанных мигрирующих объектов, сеть также называется моделью миграции. Объекты типизированы. Объектами становятся мигрирующие объекты СУБД Oracle: таблицы, представления, функции, процедуры, пакеты, последовательности, связанные с таблицами индексы, ограничения и триггера. Другие типы объектов добавляются в модель миграции для учета связей объектов БД с интерфейсными модулями, реализованными на различных высокоуровневых языках программирования. Как видим, объект в модели миграции – это достаточно широкое абстрактное понятие, основанное на объектах СУБД Oracle. Каждый

объект имеет имя, тип и владельца (схему). Эта тройка определяет идентичность объекта в модели. Объект может быть реализован (создан) в различных средах, в СУБД Oracle и в СУБД PostgreSQL. Каждой реализации объекта ставится в соответствие DDL скрипт на создание или описание объекта. Процесс миграции из одной СУБД в другую сводится к формированию новых реализаций объектов для целевой СУБД и созданию этих объектов. Ниже приведены две реализации (два DDL скрипта на создание) одного объекта – функции N_GET_DEP_LIST_NAME на PL/SQL и PLPGSQL.

-- Тип объекта FUNCTION. Объект N_GET_DEP_LIST_NAME.

```
CREATE OR REPLACE FUNCTION "APT"."N_GET_DEP_LIST_NAME"
(DEP_ID NUMBER) RETURN VARCHAR2 IS
RES    VARCHAR2(255);
BEGIN
    SELECT DEPOSIT_NAME INTO RES FROM N_DEPOSIT_LISTS WHERE ID = DEP_ID;
    RETURN RES;
EXCEPTION
    WHEN OTHERS THEN RETURN NULL;
END;
;
```

-- Реализация объекта N_GET_DEP_LIST_NAME на языке PL/SQL в среде Oracle.

-- Тип объекта FUNCTION. Объект N_GET_DEP_LIST_NAME.

```
CREATE OR REPLACE FUNCTION APT.N_GET_DEP_LIST_NAME (DEP_ID NUMERIC)
RETURNS VARCHAR AS
$$
DECLARE
    RES    VARCHAR(255);
BEGIN
    SELECT DEPOSIT_NAME INTO RES FROM APT.N_DEPOSIT_LISTS
    WHERE ID = DEP_ID;
    RETURN RES;
EXCEPTION
    WHEN OTHERS THEN
    RETURN NULL;
END;
$$ LANGUAGE 'plpgsql';
```

--Реализация объекта N_GET_DEP_LIST_NAME на языке PLPGSQL в среде PostgreSQL.

Как видим реализации объекта в разных СУБД отличаются, но при этом они относятся к одному и тому же объекту модели миграции. Текстовое представление объекта позволяет использовать лексический анализ для поиска связей между объектами. В приведенном выше примере функция `N_GET_DEP_LIST_NAME` обращается с запросом к таблице `N_DEPOSIT_LISTS` и, следовательно, зависит от этой таблицы. Объекты и связи между объектами можно представить в виде направленного графа, в котором направленная дуга выходит из зависимого объекта. В нашем примере этот граф будет таким: `N_DEPOSIT_LISTS ← N_GET_DEP_LIST_NAME`.

Каждому объекту из сети ставится в соответствие несколько DDL скриптов на создание или описание объекта на языках реализации объекта (SQL, PL/SQL, PLPGSQL, JavaScript, и другие). DDL каждого объекта анализируется лексически для выделения токенов. Словарь токенов формируется динамически на основе словаря объектов СУБД Oracle, а также включает в себя лексические конструкции языков реализации объектов. Словарь токенов должен быть открыт и может пополняться пользователями. Для построения сети используются данные словаря СУБД Oracle. Для связывания табличных объектов используются также данные СУБД Oracle о внешних ключах.

Следует отметить определенные тонкости в проведении лексического анализа DDL скриптов. Предполагается, что все имена выделяемых при лексическом анализе объектов загружены в словарь. Но использование при лексическом анализе только имени без учета контекста использования этого имени может привести к ошибкам выделения связанных объектов. Например, идентификатор переменной может иметь имя, совпадающее с именем объекта БД. На уровне словаря объектов БД такие конфликты невозможны, а хорошо реализованный программный код также должен избегать таких конфликтов, применяя определенные правила присвоения имен. Авторы считают, что на практике можно отказаться от анализа контекстов использования имен и выделять объекты прямо по имени из словаря. Возможные немногочисленные ошибки можно будет потом найти и устранить вручную.

Сеть является направленным графом и содержит объекты миграции и их связи. Сеть формируется итерациями. В модель вносится некоторое начальное множество объектов. При этом указывается владелец объекта (схема), тип объекта и имя объекта. Далее запускается процедура загрузки не загруженных объектов. В ходе загрузки для объектов

формируются их DDL описания, которые анализируются с целью выделения связанных объектов. Выделенные и отсутствующие в модели миграции объекты вносятся в модель, одновременно в модель вносятся новые выделенные связи объектов. Далее итерация повторяется для вновь выделенных и еще не загруженных объектов. Формирование сети объектов может продолжаться до полного исчерпания попадающих в сеть объектов. В любой момент сеть может быть пополнена новыми объектами, и процедура формирования сети может быть продолжена.

В ходе формирования сети автоматически поддерживается расчет прямого рейтинга каждого объекта в сети. Прямой рейтинг равен числу объектов, напрямую зависящих от данного (число направленных дуг графа сети, входящих в данный объект). Отдельной процедурой рассчитывается полный рейтинг объекта, который учитывает не только прямую зависимость объектов между собой, но и косвенную (число объектов сети, имеющих путь к данному объекту по направленным дугам графа сети).

Проиллюстрируем сказанное на примере. Пусть сеть состоит из четырех объектов и ее граф имеет вид: $A \leftarrow B \leftarrow C \leftarrow D$. Объект B зависит от объекта A , объект C зависит от объекта B , объект D зависит от объекта C . Прямые рейтинги объектов (A, B, C, D) соответственно равны $(1, 1, 1, 0)$, а полные рейтинги равны $(3, 2, 1, 0)$.

Алгоритм расчета полного рейтинга объекта основан на реализации в Oracle SQL операции «minus». Во временную таблицу записывается идентификатор объекта, для которого вычисляется полный рейтинг. Затем в цикле следует итерация. Найти по связям идентификаторы множества объектов, которые зависят от объектов, уже имеющихся в таблице, и вычесть из этого множества уже внесенные в таблицу идентификаторы. Цикл завершается, когда разность двух множеств становится пустой, что неизбежно в силу конечности числа объектов и их связей. Число записей в таблице по завершении цикла равно полному рейтингу объекта. Реализация этого алгоритма в виде функции имеет 28 строк кода на PL/SQL. Расчет рейтингов десятка тысяч объектов потребовал всего нескольких минут.

Чем выше рейтинги объекта, тем больше число объектов, зависящих от него прямо или косвенно. Поскольку миграция является итерационным последовательным процессом реализации мигрирующих объектов в другой СУБД, то естественной разумной стратегией миграции является реализация в первую очередь высокорейтинговых объектов,

без создания которых невозможно создать другие зависящие от них объекты. Рейтинги позволяют сформировать стратегию миграции и определить последовательность реализации объектов. Стратегия реализации по рейтингам не является догмой, можно использовать и другие стратегии, например сначала реализовать все таблицы вместе с данными и представлениями, а уже затем переходить к реализации функциональных объектов согласно их рейтингу. Можно строить стратегию реализации от интерфейсных модулей. В таком случае включение интерфейсных модулей в сеть позволит определить все связанные объекты необходимые для реализации модуля. Выбор стратегии остается за пользователем и определяется приоритетами пользователя, при этом он обладает полной информацией о связности модели миграции. Если критерием является полная миграция системы в наиболее короткий срок, то стратегия миграции объектов в порядке их полного рейтинга представляется нам оптимальной.

Следующей задачей модели миграции является конвертация объектов модели в объекты реализации. Процедура конвертации переводит DDL исходного объекта в DDL объекта реализации в среде PostgreSQL. Важно отметить, что не всегда возможна конвертация объекта «один в один». Например, пакеты на языке PL/SQL СУБД Oracle не могут быть перенесены непосредственно в СУБД PostgreSQL как пакеты. Поэтому пакеты разбираются на отдельные функциональные единицы, включая не только объявленные в интерфейсе пакетов процедуры и функции, но и процедуры и функции, инкапсулированные в теле пакетов. Табличные триггера в СУБД PostgreSQL в отличие от Oracle реализуются не одним, а двумя объектами – триггерными функциями и собственно триггерами. Процедура «разборки» объектов Oracle (пакетов и триггеров) может быть реализована автоматически и не потребует участия пользователей в этом процессе. Особенности реализации учитываются при загрузке объектов в сеть. Там, где необходимо, объекты автоматически «разбираются» на отдельные части, подлежащие отдельной реализации в СУБД PostgreSQL. При этом автоматически формируются имена объектов реализации.

Для мигрирующих табличных объектов может быть выполнена оптимизация. Анализируются табличные данные переносимых табличных объектов, выделяются пустые колонки, оценивается реальная точность данных в не пустых колонках. При формировании DDL реализации табличного объекта могут быть исключены некоторые (можно все пустые) колонки, точность колонок может автоматически приводиться

к реальной с некоторым увеличением в расчете на ожидаемую точность данных. Объекты, попавшие в сеть, но исключаемые из реализации помечаются особым статусом и могут быть оставлены в модели миграции или удалены из нее. Оптимизация может выполняться автоматически, но оптимизация под контролем пользователя будет более качественной поскольку только пользователь может иметь достаточно полное представление о вносимых в таблицы данных.

Интерин О2Р должно поддерживать перенос табличных данных. Для этого табличные данные выгружаются в большие объекты (LOB) СУБД Oracle в формате для загрузки в СУБД PostgreSQL командой Copy From. Так переносятся только данные, не относящиеся к типам LOB и Long Raw (данные этих типов переносятся отдельно).

Написание полного транслятора функциональных объектов с одного языка на другой в общем случае является достаточно долговременной и трудоемкой задачей. На данном этапе для трансляции функционального кода предполагается использовать ограниченный конвертор, который ставит своей задачей решение наиболее часто встречаемых проблем трансляции исходных функциональных DDL. Это приведение типов транслируемых данных к типам PostgreSQL (за исключением объектных типов Oracle), автоматическое формирование имен транслируемых функциональных единиц (функций и процедур) и их вызовов, в ограниченной степени синтаксическое приведение кода к синтаксису PostgreSQL, в ограниченной степени замена внутренних функций и процедур языка PL/SQL, не поддерживаемых в PostgreSQL, на соответствующие встроенные процедуры и функции PostgreSQL. Конвертор функциональных объектов не будет обеспечивать 100% синтаксическую и семантическую точность перевода, но он обеспечит массовое рутинное преобразование кода, оставив тонкие детали перевода на программиста. На данном этапе трудно количественно оценить экономию труда программиста, достигаемую при автоматической конвертации объектов по отношению к полностью ручной трансляции объектов. Для такой оценки надо накопить статистику по внесенным в код конвертором и программистом изменениям, но сомнений в повышении производительности труда программиста при конвертации нет.

3. Апробация решения

Техническая реализуемость и обоснованность расчётных характеристик Интерин О2Р подтверждена разработкой и применением

прототипа Интерин О2Р для миграции подсистемы «Аптека» МИС Интерин PROMIS с СУБД Oracle на СУБД PostgreSQL. Была сформирована модель миграции с показанными в таблице 2 статистическими показателями.

Таблица 2. Частичное представление модели миграции для подсистемы «Аптека»

Схема	Тип объекта	Количество объектов	Статус объекта
ALPHA	ALPHA_FORM	328	Загружен
ALPHA	ALPHA_FORM	2	Создан
ALPHA	ALPHA_RESOURCE	563	Загружен
ALPHA	ALPHA_RESOURCE	16	Создан
APT	FK	7	Не загружен
APT	FK	84	Загружен
APT	FUNCTION	508	Загружен
APT	FUNCTION	8	Создан
APT	FUNCTION	2	Готов
APT	INDEX	167	Загружен
APT	INDEX	1	Создан
APT	MATERIALIZED VIEW	1	Транслирован
APT	PROCEDURE	845	Загружен
APT	PROCEDURE	1	Выведен
APT	TABLE	15	Загружен
APT	TABLE	121	Транслирован
APT	TABLE	55	Создан
APT	TFUNCTION	67	Загружен
APT	TRIGGER	67	Загружен
APT	VIEW	23	Транслирован

Модель миграции содержала 8400 объектов в различных схемах и 15302 связи. Для миграции подсистемы «Аптека» в основном загружались серверные объекты только одной схемы БД Oracle, а также связанные с подсистемой интерфейсные модули и ресурсы платформы Интерин IPS. Построение модели уложилось в одни сутки.

Ниже в таблице 3 представлены высокорейтинговые объекты модели миграции, отсортированные по прямому рейтингу. Представляет интерес вопрос о корреляции между полным и прямым рейтингом, насколько они соответствуют друг другу. Расчет коэффициента корреляции по данным таблицы показал средний уровень корреляционной связи 0,58. Можно утверждать, что рейтинги не являются взаимозаменяемыми, и рекомендовать пользоваться полным рейтингом при построении стратегии миграции.

Таблица 3. Объекты модели миграции с высоким рейтингом

Ранг	Полный ранг	Схема	Тип	Объект
578	1124	APT	TABLE	N_REQ_HEADS
514	1066	APT	TABLE	N_DEPOSITS
489	958	APT	TABLE	N_REQUESTS
478	921	APT	TABLE	N_REQUESTS_OUT
419	1276	APT	TABLE	N_MEDIC_REGS
398	1124	APT	TABLE	N_BILL_HEADS
388	653	APT	TABLE	N_DEPOSITS_BILL
316	1590	APT	TABLE	NA_TREE_DICTS
303	1556	APT	TABLE	N_DEPOSIT_LISTS
289	575	APT	TABLE	N_MED_DEPOSITS
281	1306	APT	TABLE	N_MEASURES
240	470	APT	FUNCTION	N_GET_DEP_LIST_NAME
196	419	APT	TABLE	N_MED_DEPOSITS_BILL
148	1214	APT	TABLE	N_BID_HEADS
144	998	APT	TABLE	N_BID_RECS
143	419	APT	FUNCTION	N_GET_PREP_PROPS
129	358	APT	TABLE	N_DEP_OWNERS
122	335	APT	FUNCTION	N_GET_REG_NAME
122	1259	APT	TABLE	N_SUPPLIERS
118	358	APT	FUNCTION	N_IF_QT_LESS_1
103	304	APT	FUNCTION	N_GET_MEASURE_NAME
99	949	APT	TABLE	N_STORES
95	307	APT	FUNCTION	N_GET_PAYMENT
85	164	APT	TABLE	N_REAL_DEPOSITS
84	963	APT	TABLE	N_DOC_EXP_HEADS
79	947	APT	TABLE	N_DOC_EXP_CONTS
78	198	APT	TABLE	N_STORES_BILL
74	218	APT	FUNCTION	N_GET_STORE_GR
67	308	APT	FUNCTION	N_GET_SUPP_NAME
65	109	APT	PROCEDURE	AU_FILL_TABLE

Переносились табличные данные в объеме более 10 млн записей в отдельных таблицах. Размеры больших файлов (LOB) с данными достигали 10^9 символов. Время формирования файлов с данными и загрузки данных являлось незначительным (минуты на формирование

и загрузку 10 млн записей).

Вычислительные эксперименты полностью подтвердили принятую концепцию решения и хорошие показатели скорости работы Интерин O2P.

Заключение

В статье предлагается программное решение поддержки миграции, которое в рамках импортозамещения и ухода с проприетарного лицензионного ПО на свободное ПО с открытым кодом может предоставить отечественным разработчикам ПО средство для осуществления миграции серверных объектов СУБД Oracle, включая табличные данные, в СУБД PostgreSQL. Интерин O2P должно обеспечивать выгрузку объектов из СУБД Oracle, формирование модели миграции в виде сети связанных выгруженных объектов, трансляцию (конвертацию) объектов Oracle в объекты PostgreSQL, включая перенос табличных данных. Интерин O2P поддерживает командный процесс миграции, определяя последовательность операций миграции на основе рассчитываемых в модели рейтингов переносимых объектов, отслеживает состояние (статус) каждого переносимого объекта, позволяет назначить ответственных за миграцию каждого объекта, контролирует взаимозависимость объектов и предлагает последовательность реализации объектов. Интерин O2P позволяет выполнить оптимизацию структуры данных при миграции (вывод из использования и отказ от реализации объектов, исключение не заполняемых атрибутов из таблиц, приведение задаваемой точности атрибутов к реальной). Предполагается, что Интерин O2P будет доступно с открытым документированным кодом, что позволит любому разработчику вносить в него и в процесс миграции изменения. Для проверки заложенных в Интерин O2P концепций в ИПС РАН был разработан прототип Интерин O2P. В ходе проведения в 2022 году экспериментов по миграции одной из подсистем МИС Интерин PROMIS с СУБД Oracle на СУБД PostgreSQL прототип подтвердил эффективность заложенных в него программных решений.

Концепция решения создавалась в интересах разработчиков ПО, переводящих свое ПО с СУБД Oracle на СУБД PostgreSQL. Сфера применения решения любая, где используется СУБД Oracle и есть необходимость в миграции на СУБД PostgreSQL. Актуальна реализация решения в виде отечественного СПО.

Благодарности. Авторы искренне благодарны С. А. Амелюкину за интерес к работе и конструктивную критику, позволившую улучшить статью.

Список литературы

- [1] Прозоров А., Шнырев Р., Волков Д. *Архитектура цифровых платформ будущего* // Открытые системы. СУБД.– 2021.– № 02.  ↑79
- [2] Гулиев Я. И., Бельшев Д. В., Кочуров Е. В. *Медицинская информационная система «Интерин PROMIS Alpha» — новые горизонты* // Врачи и информационные технологии.– 2016.– № 6.– с. 6–15.   ↑79

Поступила в редакцию 20.04.2022;
одобрена после рецензирования 25.04.2022;
принята к публикации 01.11.2022.

Рекомендовал к публикации

к.т.н. Я. И. Гулиев

Информация об авторах:



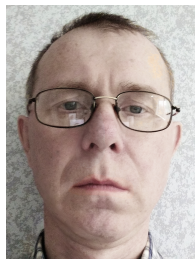
Владимир Леонидович Малых

Зав. лабораторией Института программных систем им. А. К. Айламазяна, 30 лет работы в сфере медицинской информатики, архитектор семейства МИС Интерин.



0000-0002-0072-0724

e-mail: mvl@interin.ru



Алексей Николаевич Калинин

Ведущий инженер-программист ООО «Интерин технологии», ведущий разработчик семейства МИС Интерин.



0000-0002-3607-7400

e-mail: ank@interin.ru

*Все авторы сделали эквивалентный вклад в подготовку публикации.
Авторы заявляют об отсутствии конфликта интересов.*

Migration from Oracle to PostgreSQL

Vladimir Leonidovich **Malykh**¹, Aleksey Nikolayevich **Kalinin**²

^{1,2} Ailamazyan Program Systems Institute of RAS, Ves'kovo, Russia

¹ mvl@interin.ru

(learn more about the authors in Russian on p. 90)

Abstract. There is a trend towards a wider use of free software in IT. This trend is associated with the transition to a more economical model of software creation and operation, with an increase in the security of software use, the exclusion of proprietary software control over users and users' data, with the overcoming of various sanctions in the field of proprietary software. For Russia, the rejection of proprietary Oracle software and the transition to PostgreSQL is becoming relevant. The article suggests an approach to solving the problem of migration from Oracle to PostgreSQL of complex application software. The approach was worked out on the example of the migration of the MIS Interin. The results can be used by software developers solving migration tasks. (*In Russian*).

Key words and phrases: medical Informatics, medical information systems, software migration, Oracle, PostgreSQL

2020 *Mathematics Subject Classification:* 68P15; 68P20, 92C50

For citation: Vladimir L. Malykh, Aleksey N. Kalinin. *Migration from Oracle to PostgreSQL* // Program Systems: Theory and Applications, 2022, **13**:4(55), pp. 77–91. (*In Russian*). http://psta.psiras.ru/read/psta2022_4_77-91.pdf

References

- [1] A. Prozorov, R. Shnyrev, D. Volkov. “Architecture of digital platforms of the future”, *Otkrytyye sistemy. SUBD*, 2021, no. 02 (in Russian). ↑⁷⁹
- [2] Ya. I. Guliyev, D. V. Belyshev, Ye. V. Kochurov. “Medical information system Interin PROMIS Alpha – new horizons”, *Vrach i informatsionnyye tekhnologii*, 2016, no. 6, pp. 6–15 (in Russian). ↑⁷⁹