

S.Karimov, I.Bektenaliev, B.T.Kalimbetov

Uniform approximation of singularly perturbed solution of differential equations in a special critical case

In this paper the approximate solution of singularly perturbed differential system in particularly critical case. The existence and uniqueness solutions of the original problem is constructed by successive approximations. The proper variable matrices have zeros and change independent variable does not affect order. We present convergence successive approximations in class of continuous functions. Prove convergence approximate solutions to exact one with any degree of accuracy.

References

- 1 Karimov S. *Asymptotic behavior solutions certain classes differential equations with a small parameter in case change stability in plane rest point of «fast motion»*: Dissertation for obtaining of scientific degree doctor of physical-mathematical sciences by speciality: 01.01.02. — Osh, 1980, 260 p.
- 2 Kalimbetov B.T., Mamatqulova M. *Asymptotic behavior solutions singularly perturbed differential equations in case change stability* // Bulletin Karaganda University. Ser. Mathematics, 2012, № 4, 55–60 p.
- 3 Alybaev K.S. *The method line-level research singularly perturbed equations when the condition of stability*. Dissertation for obtaining of scientific degree doctor of physical-mathematical sciences by speciality: 01.01.02. — Jalal-Abad, 2001, 376 p.

УДК 004.65

Р.М.Мендыбаев, А.М.Омаров

Карагандинский государственный университет им. Е.А.Букетова (E-mail: lorasugralina@mail.ru)

Основные проблемы миграции СУБД с MySQL на Oracle DBMS

В статье рассмотрены аспекты сложности изменения СУБД на конкретном приложении. Затронуты особенности СУБД Oracle DBMS, показано сравнение методик подхода указанных СУБД к работе с временными таблицами и отмечена уникальность данных методик. Предложены различные решения задач, которые были получены в результате исследования исходного кода приложения, как с использованием функций процедурных языков, так и методов объектно-ориентированного языка программирования Java. Отражены различия в синтаксисе языка описания данных СУБД.

Ключевые слова: миграция, СУБД, MySQL, Oracle DBMS, Java, работа с временными таблицами, последовательности, процедуры PL\SQL.

Стремительное развитие информационных технологий диктует необходимость своевременно обновлять и изменять аппаратное обеспечение информационных систем и приложений, операционные системы, а также версии и даже СУБД целиком. Цель данной работы — показать с какими проблемами и трудностями предстоит встретиться и решить их при замене СУБД. Исследование проводилось в рамках поставленной задачи на проекте по рефакторингу приложения генерирования отчетов для компаний, осуществляющих логистические услуги.

Изначальное приложение использовало в качестве СУБД продукт MySQL. Была поставлена задача — адаптировать приложение для работы с Oracle DBMS. Первая проблема, которая была выявлена, это различный подход для генерации уникальных значений первичных ключей. В MySQL для этого используется автоинкрементация поля, которая при добавлении каждой новой записи, инкрементирует значение первичного ключа на заданное количество. В Oracle DBMS такой возможности не предусмотрено, а для получения уникального значения используются последовательности (*sequence*). Это бывает очень удобно, и снимает с разработчика необходимость реализовывать алгоритм для создания значений первичного ключа. Каждое новое значение в последовательности созда-

ется на основании правил, указанных при создании [1]. Общий синтаксис создания и пример показаны на рисунке 1.

```
CREATE SEQUENCE [schema].sequence_name
[INCREMENT BY number]
[START WITH number]
[MAXVALUE number | NOMAXVALUE]
[MINVALUE number | NOMINVALUE]
[CYCLE | NOCYCLE]
[CACHE number | NOCACHE]
[ORDER | NOORDER] ;
```

Рисунок 1. Создание последовательности для Oracle DBMS

Следующая проблема, с которой пришлось столкнуться, — это длинные имена названий таблиц, индексов, ключей. Oracle DBMS накладывает численное ограничение на длину названия в 30 символов. А так как процесс генерации отчетов в некоторых местах использовал сложение нескольких имен или прибавление нескольких символов для придания уникальности, то выходом послужило написание вспомогательных методов, показанных на рисунке 2.

```
public static String getShortenedFieldByLength(String field, Integer length) {
    return field.length() > length ? field.substring(0, length) : field;
}

public static String getShortenedFieldWithAlias(String field) {
    String[] values = field.split("\\.");
    return values[0] + "." + getShortenedFieldByLength(values[1], 30);
}

public static String removeFirstUnderScore(String str) {
    return str.indexOf("_") == 0 ? str.replaceFirst("_", "") : str;
}
```

Рисунок 2. Вспомогательные методы, написанные на языке Java

Рассмотрим, что делают данные методы: *getShortenedFieldByLength()* вернет подстроку строки, обрезанной на определенное количество, переданной в качестве параметра; *getShortenedFieldWithAlias()* работает по аналогии, только разбивает строку, в которой присутствует псевдоним; *removeFirstUnderScore()* удаляет символ — подчеркивание, если он находится в начале строки, так как Oracle DBMS вызывает исключение при обнаружении таких имен.

Одна из основных и довольно трудоемких проблем, которые встретились, заключалась в различном подходе двух СУБД к работе с временными таблицами. В приложении они выполняли функцию буфера, т.е. на основании данных, полученных с пользовательского интерфейса, создавались временные таблицы, из которых, как следствие, выгружались пользовательские отчеты. И так, как вели себя MySQL и Oracle, будет рассмотрено далее. Начнем с MySQL, так как это была изначальная платформа для хранения данных. В данной СУБД временные таблицы, по умолчанию, создавались только с областью видимости сессии подключения и были не доступны извне. То есть была точная уверенность, что даже если временные таблицы будут иметь одинаковые имена, целостность данных сохранится. Пример создания временной таблицы в MySQL показан на рисунке 3.

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition, ...)
    [table_options]
    [partition_options]
```

Рисунок 3. Общая схема синтаксиса создания таблиц в СУБД MySQL

Исходя из такого подхода, было реализовано функционирование приложения. Однако подход СУБД Oracle DBMS к работе с временными таблицами в корне отличается. Первое отличие состоит в том, что пользователю при создании предлагается выбрать как таблице вести себя с данными, либо выбрать и указать опцию ON COMMIT DELETE ROWS. Эта опция используется во временных таблицах, данные которой существуют в пределах одной транзакции. Oracle удаляет все строки — все данные из временной таблицы после завершения транзакции, после выполнения команды COMMIT. Данная инструкция подразумевается по умолчанию, т.е. при создании временной таблицы — она всегда будет ON COMMIT DELETE ROWS. Другая опция, ON COMMIT RESERVE ROWS, используется во временных таблицах, данные которой существуют в пределах одной сессии. СУБД Oracle удаляет все строки из временной таблицы — очищает таблицу после завершения сессии [2]. Также следует отметить, что в Oracle все временные таблицы глобальные, что означает их доступность любому открытому соединению к экземпляру базы данных. При более тщательном исследовании было выявлено, что при генерации скриптов для создания отчетов были использованы функции как DDL (*Data Definition Language* — язык описания данных), так и DML (*Data Manipulation Language* — язык управления (манипулирования) данными). Чем же отличаются эти две группы операторов? В то время, как операторы DML достаточно однотипны для различных реализаций SQL (что дает возможность каждому поставщику программной продукции вводить свои расширения), DDL имеет существенные различия для разных продуктов. Каждый поставщик системы управления базой данных на физическом уровне различным образом реализует реляционную модель и каждый поставщик DDL неизбежно отражает эти различия. Большинство поставщиков предоставляют графические инструменты для определения данных и многие, включая и Microsoft, не ограничивают вас использованием только SQL DDL. Например, Microsoft предоставляет поддержку двух стандартов определения данных: ADO и DAO [3]. Обзор основных команд языков DDL, DML, DCL показан на рисунке 4.

Language	Statement
Data Manipulation Language (DML)	SELECT
	INSERT*
	UPDATE*
	DELETE*
Data Definition Language (DDL)	ALTER
	CREATE
	DROP
Data Control Language (DCL)	GRANT
	REVOKE

Рисунок 4. Основные команды

Причем при вызове какой-либо из функций DDL Oracle совершает неявный вызов команды COMMIT, что приводит к тому, что использовать опцию ON COMMIT DELETE ROWS не имеет смысла. При использовании же опции ON COMMIT RESERVE ROWS была выявлена проблема, что поскольку временные таблицы в Oracle являются глобальными, то сессия подключения первого же пользователя вызовет блокировку данной таблицы для манипулирования данными.

Проанализировав ситуацию, нами было предложено два решения данной проблемы. Первое заключалось в написании отдельного механизма с применением парадигмы «один пользователь — одно соединение», своего рода специального пула соединений для обхода блокировки временных таб-

лиц. Второе решение подразумевало собой создание уникальных таблиц, причем для каждого пользователя создавались свои уникальные имена для таблиц, которые могли повториться с интервалом приблизительно через три дня. В конечном итоге второе решение оказалось стабильным, достаточно производительным, и требовало меньше трудозатрат для настройки самой базы данных.

Еще одним из наиболее значимым отличием Oracle DBMS от MySQL является то, что в Oracle отсутствует функция, аналогичная в MySQL, IF EXIST. Данная функция при вызове любой функции языка DDL проверяет, существуют ли такие данные на самом деле. В Oracle DBMS ничего подобного нет, а при вызове указанных функций при существовании данных с такими же именами генерируется исключение. Для наиболее быстрого и менее трудозатратного решения была написана процедура на языке PL\SQL, которая бы при удалении таблицы проверяла наличие таблицы с данным именем. PL/SQL (Procedure Language) — расширяет стандартный SQL и добавляет возможность работать с переменными, константами, процедурами, функциями, модулями, условными операторами, циклами, обрабатывать исключения, создавать пакеты и триггеры и т.д. Язык PL/SQL разработан корпорацией Oracle для своих баз данных. Приложения, написанные на этом языке, хранятся и выполняются внутри базы данных [4]. Синтаксис процедуры показан на рисунке 5.

```

1 CREATE OR REPLACE PROCEDURE DeleteTempTableIfExists (table_name_in IN VARCHAR2)
2 IS
3     v_count NUMBER := 0;
4 BEGIN
5     SELECT COUNT(*) INTO v_count FROM all_tables WHERE table_name = table_name_in;
6     IF v_count = 1 THEN
7         EXECUTE IMMEDIATE 'DROP TABLE ' || table_name_in;
8     END IF;
9     NULL;
10 END;
11

```

Рисунок 5. Создание процедуры DeleteTempTableIfExists

Подводя итоги, можно отметить, что миграция СУБД довольно трудоемкая задача, которая требует от исполнителя, помимо быстро вхождения в исходный код проекта, хорошего знания как СУБД, с которой будут мигрировать, так и той, на которую назначена миграция.

Список литературы

- 1 <http://all-oracle.ru/content/view/?part=1&id=111> Объекты Oracle: Последовательности
- 2 <http://orasource.ru/vremennyye-tablitsyi-v-oracle.html> Временные таблицы в Oracle
- 3 <http://www.intuit.ru/department/database/prmssql2000/22/> Урок: Язык определения данных
- 4 <http://plssql.ru/index.xhtml?page=intro> PL\SQL, Введение

Р.М.Мендібаев, А.М.Омаров

MySQL-ден Oracle DBMS-ке ДҚБЖ миграциясының негізгі мәселелері

Мақалада нақты қосымшадағы ДҚБЖ өзгерту күрделілігінің аспектілері қарастырылған. Oracle DBMS ДҚБЖ ерекшеліктері айқындалып, уақытша кестелер мен жұмысқа ДҚБЖ-мен көрсетілген тәсілдемелердің әдістемелерін салыстырылған және әдістемелер деректерінің бірегейлігі белгіленген. Процедуралық тілдер функцияларын, сондай-ақ Java объекті-бағыттау бағдарламалау тілінің әдістемелерін қолданумен қосымшаның бастапқы кодын зерттеу нәтижесінде алынған тапсырмалардың әр түрлі шешімдері ұсынылған. ДҚБЖ деректерін сипаттау тілінің синтаксисіндегі айырмашылықтар туралы айтылған.

Main problems of migration from MySQL to Oracle DBMS

The article describes the changing aspects of the DBMS in a particular application. Touches upon particularities of the Oracle DBMS, shows a comparison of these methods approach of the DBMS to work with temporary tables and marked the uniqueness of these techniques. Offered a variety solution of tasks that have been obtained from the study of the application source code, based both on the usage the functions of procedural languages, and methods of the object-oriented programming language Java. Reflect differences in the syntax data definition language of the database.

References

- 1 <http://all-oracle.ru/content/view/?part=1&id=111> Oracle's Objects: Sequences
- 2 <http://orasource.ru/vremennyye-tablitsyi-v-oracle.html> Temporary tables in oracle
- 3 <http://www.intuit.ru/department/database/prmssql2000/22/> Lesson: Data definition language
- 4 <http://plsqli.ru/index.xhtml?page=intro> PL\SQL, Introduction

УДК 517.946

К.Н.Оспанов, Р.Д.Ахметкалиева

Евразийский национальный университет им. Л.Н.Гумилева, Астана (E-mail: ospanov_k@mail.ru)

О разделимости вырожденного дифференциального оператора в гильбертовом пространстве

Статья посвящена исследованию разделимости и аппроксимативных свойств дифференциального оператора $ly := -y'' + r(x)y' + q(x)y$, $x \in R$, в гильбертовом пространстве $L_2 := L_2(R)$, $R = (-\infty, +\infty)$. Установлена коэрцитивная оценка решения соответствующего дифференциального уравнения второго порядка, продемонстрировано его применение к спектральным вопросам для дифференциального оператора l . Получены достаточные условия существования решения одного класса нелинейных вырожденных дифференциальных уравнений второго порядка на числовой оси.

Ключевые слова: дифференциальное уравнение, вырожденный дифференциальный оператор, гильбертово пространство, разделимость оператора, коэрцитивная оценка, аппроксимативная свойства, замыкание оператора, вполне непрерывная резольвента, k -поперечники по Колмогорову, ограниченная обратимость, обратный оператор, сопряженный оператор.

1 Введение и основные результаты

Понятие разделимости введено в фундаментальной работе [1]. Согласно определению, оператор Штурма-Лиувилля

$$Jy = -y'' + q(x)y, \quad x \in (a, +\infty),$$

называется разделимым в пространстве $L_2(a, +\infty)$, если соотношения $y, -y'' + qy \in L_2(a, +\infty)$ влекут включения $-y'', qy \in L_2(a, +\infty)$. Разделимость оператора J эквивалентна наличию оценки

$$\|y''\|_{L_2(a, +\infty)} + \|qy\|_{L_2(a, +\infty)} \leq c \left(\|Jy\|_{L_2(a, +\infty)} + \|y\|_{L_2(a, +\infty)} \right), \quad y \in D(J).$$

Авторы [1] получили (см. [2, 3]) ряд признаков разделимости J , в зависимости от поведения q и его производных, а также указали примеры негладких функций q , при которых J не является разделимым. Признаки разделимости J , когда q не обязательно дифференцируема, получены в [4, 5]. В [6,