

FHDW-Fachbericht

Band 2/2005

**Bernd Soldan
Ulrich Reus (Hrsg.)**

**Automatisiertes Testen von PL/SQL-Prozeduren
im Data-Warehouse-Umfeld**

Shaker Verlag
Aachen 2006

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Copyright Shaker Verlag 2006

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 3-8322-4731-9

ISSN 1861-3292

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • eMail: info@shaker.de

Vorwort des Herausgebers

Der Versuch, fehlerfreie Software zu produzieren, ähnelt in einigen Aspekten dem Streben nach absoluten Werten wie z.B. Gerechtigkeit oder der Suche nach dem heiligen Gral. Auch Idealisten, die sich bemühen dieses Ziel zu erreichen, müssen irgendwann erkennen, dass eine solche Aufgabe von Menschen nicht zu bewältigen ist.

Aber auch wenn es nicht möglich ist, von Fehlern freie Programme zu schreiben, dann sollte es trotzdem ein Ziel jedes Softwareentwicklers sein, die Zahl von Fehlfunktionen in Programmen zu minimieren bzw. die Anzahl der Fälle zu maximieren, in denen ein Programm wie erwartet funktioniert. Eine solche Steigerung bzw. Sicherung der Softwarequalität erfordert natürlich Ressourcen, die Kosten und Entwicklungszeit für die Software in die Höhe treiben. Auf der anderen Seite verursacht fehlerhafte Software bei den Anwendern höchst unerwünschte Aufwände. Prominente Beispiele, wie der Absturz der Ariane 5 beim ihrem Erstflug im Juni 1996 oder die Unfälle mit dem medizinischen Bestrahlungsgerät Therac-25 in den achtziger Jahren, stellen nur die Spitze des Eisbergs dar. Um Softwarequalität möglichst effizient und kostengünstig sicherzustellen wurden daher in den vergangenen Jahren unterschiedliche Methoden zum systematischen Testen von Software und Softwarewerkzeuge zur automatisierten Anwendung von Testmethoden entwickelt.

Gerade mit der Verfügbarkeit von einfach einsetzbaren Testwerkzeugen wie z.B. JUnit und Softwareentwicklungsmethoden wie XP (eXtreme Programming) hat die Bedeutung systematischer Softwaretests in den vergangenen Jahren deutlich zugenommen. Auf der anderen Seite gibt es immer noch Bereiche, in denen Tests nicht oder nur schwer automati-

siert und nur mit entsprechend hohem manuellen Aufwand durchgeführt werden können. Zu diesen Bereichen zählen unter anderem Benutzeroberflächen und Datenbanken.

In der vorliegenden Veröffentlichung wird ein Verfahren vorgestellt, das es ermöglicht, Tests von PL/SQL-Programmen in großen ORACLE-Datenbanken automatisiert durchzuführen. Hierbei wird eine funktionierende Lösung für das Problem der Bereitstellung konsistenter Testdaten für jede Testdurchführung vorgestellt.

Paderborn, Oktober 2005

Ulrich Reus

Inhaltsverzeichnis

Tabellenverzeichnis	VIII
Abkürzungsverzeichnis	X
Glossar	XIV
1. Einleitung	1
1.1. Einführung in die Thematik	1
1.2. Zielsetzung	3
1.3. Aufbau der Arbeit	3
2. Data Warehouse-Grundlagen	5
2.1. Schichten eines Data Warehouse	5
2.1.1. Operative Quelldatenschicht	5
2.1.2. ETL-Schicht und Data Warehouse Schicht	5
2.1.3. Applikationsschicht	6
2.1.4. Präsentationsschicht	7
2.2. Der ETL-Prozess	7
2.2.1. Definitionsphase	7
2.2.2. Ausführungsphase	8
2.3. Klassifizierung der Daten	10
2.4. Anwender des Data Warehouse	11
2.5. Online Analytical Processing	12
2.5.1. Definition	12
2.5.2. OLAP-Funktionen	14
2.5.3. Verschiedene OLAP-Architekturen	15
2.5.3.1. Relationales OLAP	16

2.5.3.2.	Multidimensionales OLAP	16
2.5.3.3.	Hybrid OLAP	17
3.	Software-Qualitätssicherung	18
3.1.	Software-Qualität	18
3.2.	Testdurchlauf	19
3.3.	Testphasen	21
3.4.	Testverfahren	23
3.5.	Teststrategien	25
3.6.	Testautomatisierung	27
3.6.1.	Manuelles Unit-Testing	27
3.6.2.	JUnit	29
3.6.3.	utPLSQL	32
3.6.4.	JUnit und utPLSQL - ein Vergleich	34
3.6.5.	Vorteile des automatisierten Unit-Testing	38
4.	Testen eines Data Warehouse	41
4.1.	Manuelles Testen	41
4.2.	Vorteile der Testautomatisierung	42
4.3.	Probleme beim automatisierten Testen	43
5.	Konzeption der Testumgebung	47
5.1.	Testdaten	47
5.2.	Table-Copy-Approach	52
5.2.1.	Testdurchlauf	52
5.2.2.	Prozedurschachtelung	55
5.2.3.	Nachteile	57
5.2.3.1.	Kopiervorgang	57
5.2.3.2.	Multi-User-Betrieb	58
5.2.3.3.	Änderung der Datenbank-Objekte	59
5.2.3.4.	Referentielle Integrität	59
5.3.	Table-Rename-Approach	60
5.3.1.	Testdurchlauf	60

5.3.2.	Vorteile gegenüber dem Table-Copy-Approach . .	64
5.3.3.	Nachteile	64
5.3.4.	Auflösung der Kollisionen	67
5.4.	Syn-Layer-Approach	68
5.4.1.	Definition	68
5.4.2.	Testdurchlauf	70
5.4.3.	Vorteile	71
5.4.4.	Kollisionen	71
5.5.	Extended-Syn-Layer-Approach	73
5.5.1.	Definition	73
5.5.2.	Vorteile	74
5.6.	Zeitstempel in den Tabellen	75
5.7.	Berücksichtigung der referentiellen Integrität	75
5.8.	Anforderungen	77
5.8.1.	Musskriterien	77
5.8.2.	Wunschkriterien	78
6.	Realisierung	80
6.1.	Verwendung des Frameworks	80
6.1.1.	Voraussetzungen	80
6.1.2.	Generierung eines Tests	81
6.1.3.	Ausführung eines Tests	83
6.1.4.	Definition der Exclusion List	91
6.1.5.	DWHUT-Dictionary	94
6.1.6.	Logging	94
6.1.7.	Definition von Testsuiten	95
6.1.8.	Ausführung von Testhierarchien	95
6.2.	Architektur	95
6.2.1.	Generierung eines Tests	95
6.2.2.	Integration in utPLSQL	98
6.2.3.	Kollisionen	103
6.2.4.	Exclusion List	104

6.2.5.	Berechtigungen	106
6.2.5.1.	Hauptschemata	106
6.2.5.2.	Testschema	108
6.2.6.	Anpassungen utPLSQL	112
6.3.	Grenzen des Frameworks	114
6.3.1.	Nicht realisierte Features	114
6.3.2.	Bekannte Probleme	115
6.4.	Bewertung utPLSQL_DWH	121
6.5.	Ausblick utPLSQL_DWH	122
7.	Resümee	125
A.	Anhang	128
A.1.	OLAP-Funktion Drill-Down	128
A.2.	OLAP-Funktion Roll-Up	129
A.3.	OLAP-Funktion Pivot	130
A.4.	OLAP-Funktion Slicing	131
A.5.	OLAP-Funktion Dicing	132
A.6.	Erfolgreiche JUnit-Testausführung	133
A.7.	JUnit-Testausführung im Falle eines Fehlers	134
A.8.	JUnit-Test	135
A.9.	JUnit-Testsuite	135
A.10.	JUnit-Architektur	135
A.11.	Erfolgreicher utPLSQL-Test	135
A.12.	Fehlgeschlagener utPLSQL-Test	135
A.13.	utPLSQL-Test	135
A.14.	Fehlgeschlagener utPLSQL_DWH-Test	135
A.15.	Erfolgreicher utPLSQL_DWH-Test	135
A.16.	utPLSQL-Testsuite	135
A.17.	Pflichtenheft	135
A.18.	Aufwandsschätzung Implementierung	135
A.19.	utPLSQL_DWH-Fehlercodes	135
A.20.	Installation des Frameworks utPLSQL_DWH	135

A.21.Deinstallation von utPLSQL_DWH	136
A.22.Code eines utPLSQL_DWH-Testes	136
A.23.Zugesicherte Grants für das Testschema	136
A.24.Zugesicherte Grants für das Hauptschema	136
A.25.Das Schema Scott	136
A.26.Selektion des zuletzt generierten Tests	136
A.27.Namenskonventionen	136
A.28.Identifikation der QA-Tables eines Tests	136
A.29.Generierung eines utPLSQL_DWH-Testes	136
A.30.DWHUT-Dictionary	137
A.31.Fehlerbehebung im utGen-Package	141
A.32.Ausführung von Testhierarchien in utPLSQL	141
A.33.utPLSQL_DWH-Logging-Level	141
A.34.Code utPLSQL_DWH-Testausführung	141

Literaturverzeichnis	142
-----------------------------	------------

Abbildungsverzeichnis

1.	Architektur eines Data Warehouse	6
2.	Multidimensionaler Datenwürfel	14
3.	Testplanungsmodell	21
4.	Testphasen	24
5.	Architektur des Frameworks utPLSQL	32
6.	Unit-Test einer Klasse	43
7.	Testen einer Prozedur in einem Data Warehouse	44
8.	Testdurchlauf mit dem Table-Copy-Approach	52
9.	Ausführung eines Unit-Tests mit dem Aufruf weiterer Pro- zeduren	56
10.	Testdurchlauf mit dem Table-Rename-Approach	61
11.	Testdurchlauf mit dem Syn-Layer-Approach	68
12.	Testdurchlauf mit dem Extended-Syn-Layer-Approach . .	73
13.	Generierung eines utPLSQL_DWH-Testes	81
14.	Ausführung eines utPLSQL_DWH-Testes	85
15.	Identifikation der QA-Tabellen eines Tests	90
16.	Kennzeichnung des Änderungsdatums eines Datensatzes .	92
17.	Identifikation der Target-Extract-Tables eines Tests	93
18.	Definition der Exclusion für einen Test	93
19.	Generierung eines utPLSQL_DWH-Testes	94
20.	Integration in utPLSQL	100
21.	Generierung eines utPLSQL-Testes	113
22.	Ausgabe utPLSQL Fehler im utGen-Package	113
23.	Test überladener Prozeduren innerhalb eines Packages . .	118

Tabellenverzeichnis

1.	Vergleich zwischen utPLSQL und JUnit	38
----	--	----

Abkürzungsverzeichnis

API	Application Program Interface; eine Programmierschnittstelle
CRM	Customer Relationship Management, Kundenbindungs- management
Cube	multidimensionaler Datenwürfel
DB	Datenbank
DBA	Datenbank-Administrator
DDL	Data Definition Language; Bestandteil des SQL-Standards; mit DDL-Anweisungen können Datenbank-Objekte definiert werden.
DML	Data Manipulation Language Das Absetzen eines Insert , Delete - oder Update - Statements auf einer Tabelle einer Datenbank.
DWH	Data Warehouse

ETL	Extraction, Transformation and Loading Dieser Vorgang innerhalb eines Data Warehouses extrahiert Daten aus operativen Quellsystemen, transformiert diese und lädt sie in die Datenbank des Data Warehouse.
ERM	Ein Enterprise Resource Management System koordiniert verschiedene Organisationen oder Standorte eines Unternehmens. Ein mögliches Ziel ist die Einsparung finanzieller oder personeller Ressourcen. Im Gegensatz zu ERP-Systemen wird lediglich eine operative Unterstützung geboten.
ERP	Ein Enterprise Resource Planning System ist eine stabile Plattform für unternehmensübergreifende Prozesse wie beispielsweise E-Business oder Supply Chain Management. Das System enthält Informationen über Finanzen, personelle Ressourcen, Produktion und Vertrieb. Ein ERP-System vereinigt die Daten unabhängiger Teilsysteme in einem gemeinsamen Informationspool. Dadurch werden Insellösungen, d.h. viele kleine Systemlösungen verdrängt.
GB	Gigabyte
GE	Geldeinheiten
GIF	Gesellschaft Informationen Factory
GPL	GNU Public License ¹

¹weitere Informationen bzgl. der GPL s. [WWW31]

GUI	Graphische Benutzeroberfläche
IDE	Integrated Development Environment; eine Entwicklungsumgebung.
JSP	Java Server Pages. Eine HTML-Seite mit Java-Code, der serverseitig ausgeführt wird. Beim Aufrufen einer JSP wird dynamisch zur Laufzeit HTML-Code generiert, der zum anfragenden Client gesendet wird.
KB	Kilobyte
HOLAP	Hybrid OLAP
MB	Megabyte
MDDB	Multidimensionale Datenbank
MOLAP	Multidimensionales OLAP
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
OO	Objektorientiert
RDBMS	Relationales Database Managment System
ROLAP	Relationales OLAP

S&N	Finanzdienstleister aus Paderborn, welcher e:Business und Banking-Lösungen entwickelt ² .
SQL	Structured Query Language; Eine strukturierte Abfragesprache zum Selektieren der Daten aus einer Datenbank.
Std.	Stunden
Z.	Zeile

²Weitere Informationen bzgl. der S&N AG s. [WWW29].

Glossar

Aggregate

Daten werden verdichtet und in einer Datenbank abgelegt. Bei einer Anfrage erfolgt keine Verdichtung zur Laufzeit, sondern es wird auf bereits gespeicherte Daten zugegriffen. Aggregate ermöglichen einen Performance-Vorteil bei einer Anfrage an ein Data Warehouse-System.

Child-Table

Ein Foreign Key Constraint legt fest, dass die Datenwerte einer Tabelle auch in einer anderen Tabelle vorhanden sein müssen. Die Tabelle, auf die das Foreign Key Constraint definiert ist, wird als Child-Table bezeichnet (Vgl. Constraints [WWW10].).

Droppen

Löschung eines Datenbank-Objektes.

Extrakt-Tabelle

Tabelle, die während einer Testausführung kopiert wurde. Eine Extrakt-Tabelle enthält in der Regel dieselben Daten wie die zugeordnete Original-Tabelle.

Hauptschema	Schema in einer Datenbank, in dem sich Objekte befinden, die durch utPLSQL_DWH getestet werden.
Java-Bean	Eine Java-Klasse, die innerhalb einer JSP-Seite verwendet wird.
Legacy-Systeme	Software-Systeme, die schlecht überschaubar sind und nur unter hohen Kosten erweitert werden können.
Original-Tabelle	Eine Tabelle, auf die ein Testobjekt bei einer Test-Ausführung zugreift. Bei dem Extended-Syn-Layer-Approach befinden sich die Original-Tabellen in einem Hauptschema.
Parent-Table	Ein Foreign Key Constraint legt fest, dass die Datenwerte einer Tabelle auch in einer anderen Tabelle vorhanden sein müssen. Die Tabelle, auf die ein Child-Table referenziert, wird als Parent-Table bezeichnet (Vgl. Constraints [WWW10] und Child-Table).
PL/SQL	Prozedurale Spracherweiterungen für SQL. PL/SQL wird innerhalb einer Oracle-Datenbank ausgeführt.

QA-Tabelle	Eine Tabelle, die die erwarteten Daten eines Tests enthält. Wenn ein Target-Extract-Table (tatsächliche Daten) und eine QA-Tabelle nach einer Testausführung identische Daten enthalten, ist ein Test erfolgreich. Wenn die Daten beider Tabellen nicht übereinstimmen, so ist der entsprechende Test fehlgeschlagen.
Refactoring	Das Umstrukturieren des Programmcodes.
Release	Die Freigabe einer neuen Programmversion aufgrund von Designentscheidungen.
Scope	Der Block einer Prozedur, Funktion oder Methode (Vgl. Feuerstein et al (2003), S. 64 ff.).
Source-Table	Tabelle, auf die eine Prozedur, ein Package oder eine Funktion lesend zugreift, d.h. es wird ein SELECT -Statement auf die entsprechende Tabelle abgesetzt.

Target-Table	Tabelle, auf die eine Prozedur, ein Package oder eine Funktion schreibend zugreift, d.h. es wird ein INSERT-, UPDATE- oder DELETE-Statement auf diese Tabelle abgesetzt. Ein zusätzlicher lesender Zugriff auf diese Tabelle kann des Weiteren auch erfolgen.
Target-Table erster Instanz	Target-Table, auf die ein Testobjekt während eines Tests schreibend zugreift.
Target-Table zweiter Instanz	Beim Testschritt setup werden die Daten dieser Tabelle in die entsprechende Target-Table erster Instanz kopiert. Durch den Zugriff auf die Target-Table erster Instanz greift ein Testobjekt während eines Tests immer wieder auf definierte Testdaten zu. Ein Testobjekt greift auf die Target-Table zweiter Instanz während eines Tests nicht zu.
Target-Extract- Table	Extrakt-Tabelle, auf die eine Prozedur, ein Package oder eine Funktion während eines Tests schreibend zugreift.
Testobjekt	Objekt (Prozedur, Package oder Funktion), welches getestet wird.

Testpackage	utPLSQL oder utPLSQL_DWH-Test.
Testprozedur	Prozedur innerhalb eines utPLSQL oder utPLSQL_DWH-Tests. Jede Testprozedur kann ein oder mehrere Assertions enthalten und ruft eine Prozedur des Testobjektes oder die zu testende Prozedur auf.
Testschema	Schema, in dem sich unter anderem das Framework utPLSQL_DWH, die Extrakt-Tabellen sowie die Testpackages befinden. Ein Testschema liegt lediglich beim Extended-Syn-Layer-Approach vor.