



**ORACLE®**

## **Vergessene Schätze**

**Datenbankfeatures, die nicht jeder nutzt**

Carsten Czarski

ORACLE Deutschland B.V. & Co KG

# Vergessene Schätze

Datenbankfeatures, die nicht jeder nutzt

**SQL MODEL CLAUSE**

**UTL\_HTTP DBMS\_ERRLOG**

**DBMS\_APPLICATION\_INFO**

**MERGE EXTRACTVALUE INSERT ALL SQL**

**V\$SESSION\_LONGOPS, XMLTYPE**

**SUBTYPE PL/SQL**

**NLS\_SORT NLS\_COMP**

# MERGE-Befehl: Übersicht

- Zwei Operationen in einem Statement
  - UPDATE, wenn Zeile in Zieltabelle existiert
  - INSERT, wenn nicht
- Vereinfachte Entwicklung
  - SQL-Standard
  - Keine komplexe PL/SQL Logik
- Performance
  - Sehr gut geeignet zum Bewegen von Daten (ETL)
  - Einmaliges "Scannen" der Quelldaten
  - Parallele Ausführung möglich

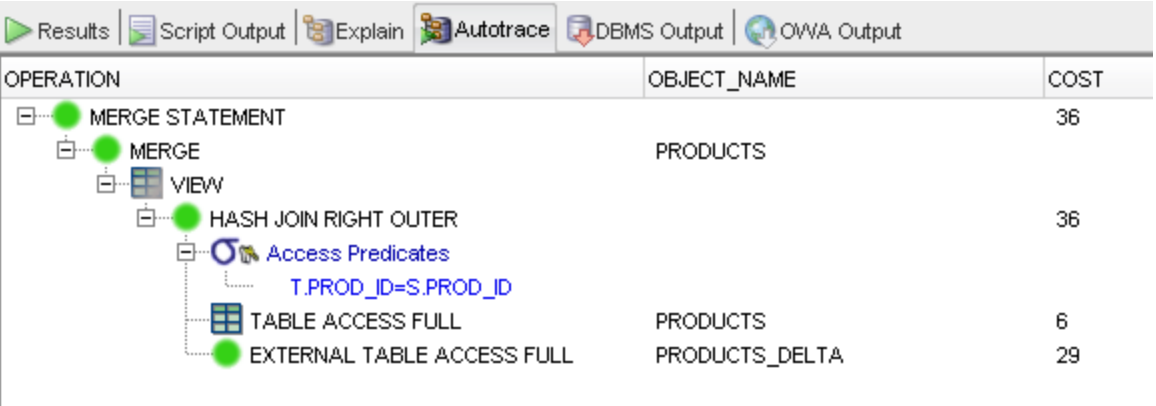
# MERGE-Erweiterungen

- Konditionales INSERT und UPDATE
  - DML Operationen nur für veränderte Daten
  - Flexibilität für individuelle Daten-Filterung
- Optionale DELETE Klausel für UPDATE
- UPDATE „only“ oder INSERT „only“ Klausel möglich

# MERGE-Kommando

## Einfaches Beispiel

```
MERGE INTO products t
  USING products_delta s
  ON (t.prod_id = s.prod_id)
WHEN MATCHED THEN
  UPDATE SET
    t.prod_list_price=s.prod_list_price,
    t.prod_min_price=s.prod_min_price
WHEN NOT MATCHED THEN
  INSERT (prod_id, ..,prod_min_price)
  VALUES (s.prod_id, .., s.prod_min_price)
```



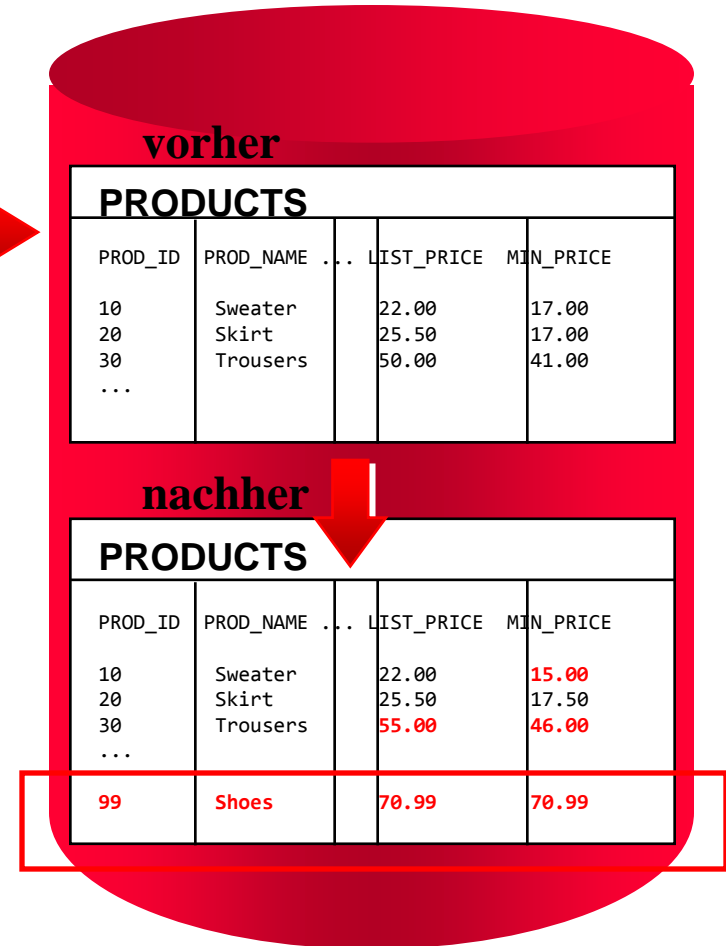
OPERATION	OBJECT_NAME	COST
MERGE STATEMENT		36
MERGE	PRODUCTS	36
VIEW		
HASH JOIN RIGHT OUTER		36
Access Predicates		
T.PROD_ID=S.PROD_ID		
TABLE ACCESS FULL	PRODUCTS	6
EXTERNAL TABLE ACCESS FULL	PRODUCTS_DELTA	29

# MERGE: Konditionales Insert

```
10, Sweater, ..., 22.00, 15.00, 'on stock', ...  
20, Skirt, ..., 25.50, 17.00, 'on stock', ...  
30, Trousers, ..., 55.00, 46.00, 'available', ..  
40, ...  
98, shoes, ..., 12.50, 10.00, 'obsolete', ...  
99, shoes, ..., 70.99, 70.99, 'available', ...
```

**MERGE**

```
MERGE INTO products t  
  USING products_delta s  
  ON (t.prod_id = s.prod_id)  
  WHEN MATCHED THEN  
  UPDATE SET  
  t.prod_list_price=s.prod_list_price,  
  t.prod_min_price=s.prod_min_price  
  WHEN NOT MATCHED THEN  
  INSERT (prod_id, ..,prod_min_price)  
  VALUES (s.prod_id,..,s.prod_min_price)  
  WHERE s.prod_status <> 'obsolete'
```



# MERGE: DELETE-Erweiterung

## Zeilen in Zieltabelle bei Bedarf löschen

```
MERGE INTO bonuses d
USING (SELECT employee_id, salary, department_id FROM
       oe.employees WHERE department_id = 80) s
ON (d.employee_id = s.employee_id)
WHEN MATCHED THEN UPDATE
      SET d.bonus = d.bonus + s.salary*.01
      DELETE WHERE (s.salary > 8000)
WHEN NOT MATCHED THEN INSERT (d.employee_id, d.bonus)
      VALUES (s.employee_id, s.salary*.01)
      WHERE (s.salary <= 8000);
```

# Multi Table INSERT

- Ein Statement
  - fügt Daten gleichzeitig in mehrere Ziel-Tabellen ein
  - Erspart mehrfaches Scannen der Originaldaten
- Ausprägungen:
  - unconditional INSERT
  - pivoting INSERT
  - conditional ALL INSERT |FIRST INSERT
- Performance-Optionen
  - PARALLEL für Parallele Ausführung
  - APPEND für Direct Path INSERT
- Wird intern für Refresh von MatViews verwendet



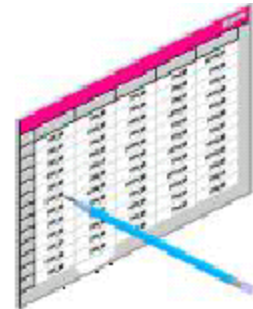
# Multi Table Insert: Ein Beispiel

## Conditional ALL INSERT

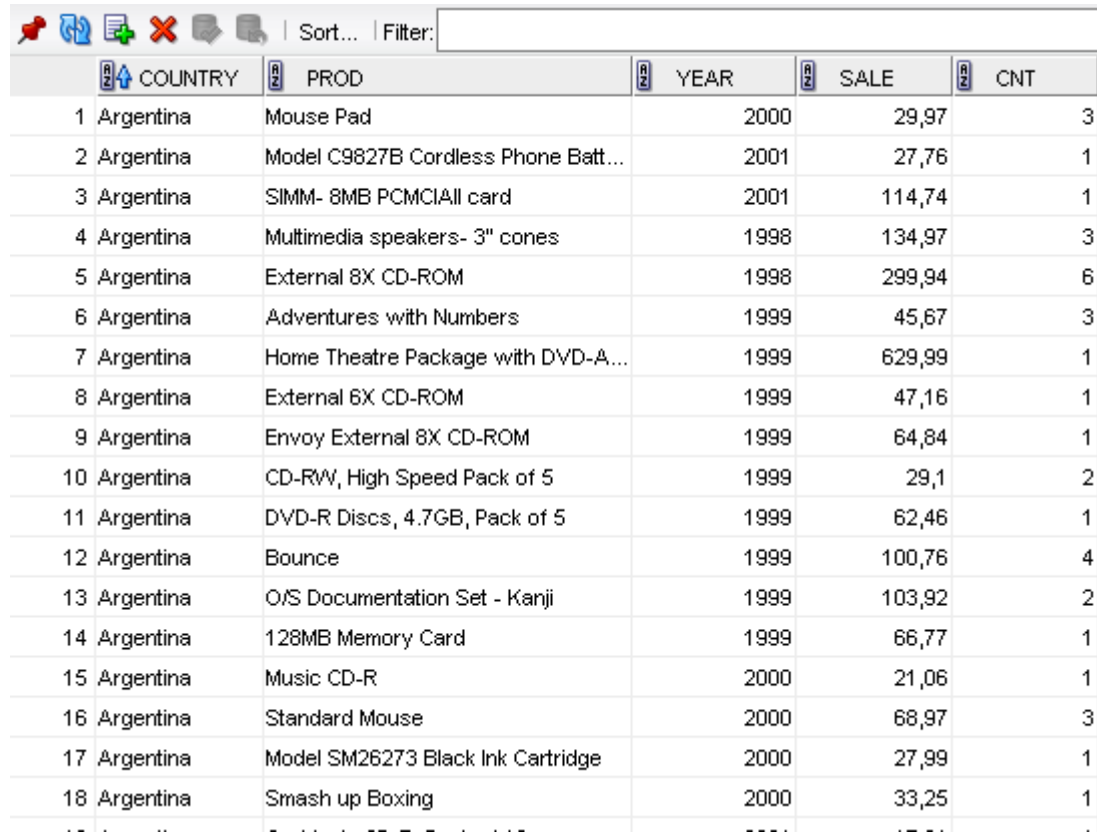
```
INSERT ALL
  WHEN product_id IN (
    SELECT product_id
    FROM promotional_items
  ) THEN INTO promotional_sales
    VALUES(product_id,list_price)
  WHEN order_mode = 'online'
  THEN INTO web_orders
    VALUES(product_id, order_total)
SELECT product_id, list_price,
       order_total, order_mode
FROM orders;
```

# SQL MODEL Klausel

- Nutzbar für
  - Komplexe Berechnungen
  - in SQL und PL/SQL
- Ersetzt komplizierte ...
  - ... JOIN- und UNION-Operationen
  - ... Berechnungen in Tabellenkalk.-Programmen
- Beispiele
  - Marktanteile,
  - Budgetberechnungen
  - Umsatzprognosen



# MODEL-Berechnungen: Ein Beispiel



The screenshot shows a window with a toolbar at the top containing icons for refresh, undo, redo, delete, and save. Below the toolbar is a search bar with 'Sort...' and 'Filter:' labels. The main area displays a table with the following columns: COUNTRY, PROD, YEAR, SALE, and CNT. The table contains 18 rows of data, all for Argentina, sorted by SALE in descending order.

	COUNTRY	PROD	YEAR	SALE	CNT
1	Argentina	Mouse Pad	2000	29,97	3
2	Argentina	Model C9827B Cordless Phone Batt...	2001	27,76	1
3	Argentina	SIMM- 8MB PCMCIAII card	2001	114,74	1
4	Argentina	Multimedia speakers- 3" cones	1998	134,97	3
5	Argentina	External 8X CD-ROM	1998	299,94	6
6	Argentina	Adventures with Numbers	1999	45,67	3
7	Argentina	Home Theatre Package with DVD-A...	1999	629,99	1
8	Argentina	External 6X CD-ROM	1999	47,16	1
9	Argentina	Envoy External 8X CD-ROM	1999	64,84	1
10	Argentina	CD-RW, High Speed Pack of 5	1999	29,1	2
11	Argentina	DVD-R Discs, 4.7GB, Pack of 5	1999	62,46	1
12	Argentina	Bounce	1999	100,76	4
13	Argentina	O/S Documentation Set - Kanji	1999	103,92	2
14	Argentina	128MB Memory Card	1999	66,77	1
15	Argentina	Music CD-R	2000	21,06	1
16	Argentina	Standard Mouse	2000	68,97	3
17	Argentina	Model SM26273 Black Ink Cartridge	2000	27,99	1
18	Argentina	Smash up Boxing	2000	33,25	1
..	..	..	..	..	..

# MODEL-Berechnungen: Beispiel

- P für **Partition**
- D für **Dimension**
- M für **Measure**

P	D	D	M	
COUNTRY	PROD	YEAR	SALE	CNT
1 Argentina	Mouse Pad	2000	29,97	3
2 Argentina	Model C9827B Cordless Phone Batt...	2001	27,76	1
3 Argentina	SIMM- 8MB PCMCIAII card	2001	114,74	1
4 Argentina	Multimedia speakers- 3" cones	1998	134,97	3
5 Argentina	External 8X CD-ROM	1998	299,94	6
6 Argentina	Adventures with Numbers	1999	45,67	3
7 Argentina	Home Theatre Package with DVD-A...	1999	629,99	1
8 Argentina	External 6X CD-ROM	1999	47,16	1
9 Argentina	Envoy External 8X CD-ROM	1999	64,84	1
10 Argentina	CD-RW, High Speed Pack of 5	1999	29,1	2
11 Argentina	DVD-R Discs, 4.7GB, Pack of 5	1999	62,46	1
12 Argentina	Bounce	1999	100,76	4
13 Argentina	O/S Documentation Set - Kanji	1999	103,92	2
14 Argentina	128MB Memory Card	1999	66,77	1
15 Argentina	Music CD-R	2000	21,06	1
16 Argentina	Standard Mouse	2000	68,97	3
17 Argentina	Model SM26273 Black Ink Cartridge	2000	27,99	1
18 Argentina	Smash up Boxing	2000	33,25	1
19 Argentina	OraMusic CD-R, Pack of 10	2001	17,21	1
20 Argentina	Music CD-R	2001	18,67	1
21 Argentina	O/S Documentation Set - English	1998	134,97	3
22 Argentina	Model K8822S Cordless Phone Batt...	1999	215,66	6
23 Argentina	Mouse Pad	1999	21,37	2

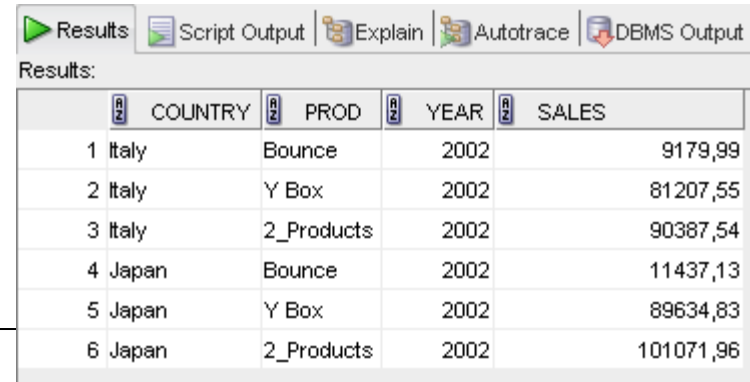
# MODEL-Berechnungen: Syntax

- Spalten einer Tabelle werden in 3 Gruppen aufgeteilt:
  - **PARTITION**: logischer Bereich der Resultatmenge
    - Beispiel: COUNTRY
  - **DIMENSION**: Identifiziert Zelle
    - Beispiel: PROD, YEAR
  - **MEASURE**: Werte, die berechnet werden
    - Beispiel: SALE
- Regeln werden folgendermaßen angegeben

```
sale['Bottle',2004] =  
    sale[prod='Bottle',year=2001] + sale[prod='Bottle',year=2000]  
sale['Hair Dryer',2004] =  
    sale[prod='Hair Dryer',year=2000]
```

# MODEL-Berechnungen

## Ein Beispiel



The screenshot shows the Oracle SQL Developer interface. At the top, there are menu items: Results, Script Output, Explain, Autotrace, and DBMS Output. Below these, the word "Results:" is displayed. The main area contains a table with 6 rows and 5 columns. The columns are labeled COUNTRY, PROD, YEAR, and SALES. The rows contain data for Italy and Japan in the year 2002, with sales values ranging from 91,799.99 to 1,010,719.96.

	COUNTRY	PROD	YEAR	SALES
1	Italy	Bounce	2002	9179,99
2	Italy	Y Box	2002	81207,55
3	Italy	2_Products	2002	90387,54
4	Japan	Bounce	2002	11437,13
5	Japan	Y Box	2002	89634,83
6	Japan	2_Products	2002	101071,96

```
SELECT country, prod, year, sales
FROM sales_view
WHERE country IN ('Italy','Japan')
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales)
RULES (
sales['Bounce',2002]      = sales['Bounce',2001]+sales['Bounce',2000],
sales['Y Box',2002]      = sales['Y Box',2001],
sales['2_Products',2002] = sales['Bounce',2002]+sales['Y Box',2002]
)
ORDER BY country, prod, year;
```

# Model Berechnungen

## Weitere Beispiele für Regeln

- Regel für ganze Bereiche ...

```
sales[FOR prod in('Mouse Pad', 'Bounce', 'Y Box'), 2005] =  
1.3 * sales[cv(prod), 2001]
```

R	COUNTRY	R	PROD	R	YEAR	R	SALES
1	Italy		Bounce		2005		6300,19
2	Italy		Mouse Pad		2005		6172,27
3	Italy		Y Box		2005		105569,815

- Loop-Konstrukte

```
sales['Mouse Pad', FOR year FROM 2005 TO 2012 INCREMENT 2] =  
1.2 * sales[cv(prod), 2001]
```

R	COUNTRY	R	PROD	R	YEAR	R	SALES
1	Italy		Mouse Pad		2005		5697,48
2	Italy		Mouse Pad		2007		5697,48
3	Italy		Mouse Pad		2009		5697,48
4	Italy		Mouse Pad		2011		5697,48

# Linguistische Unterstützung in SQL

- Berücksichtigung spezieller Zeichen bei ...
  - Sortierung (ORDER BY)
  - Filter (WHERE)
  - Index-Lookup
- Gesteuert durch Session-Parameter
  - NLS\_COMP = BINARY | LINGUISTIC
  - NLS\_SORT stellt die konkrete Sprache ein  
Suffix beachten



# Linguistische Unterstützung in SQL

- Suffix für NLS\_SORT
  - \_CI: Ignorieren von Groß-/Kleinschreibung
  - \_AI: Ignorieren von diakritischen Zeichen (è, á, ü)
  - \_M: Multilinguale Sortierung
- Beispiele

```
dbms_session.set_nls('NLS_SORT', 'XFRENCH_AI');
```

```
ALTER SESSION SET NLS_SORT=XFRENCH_AI;
```

```
select * from sortierung  
order by nlssort(text, 'NLS_SORT=XFRENCH_AI');
```

# Linguistische SQL Filter

- Case- und Akzentinsensitive Suche

```
SQL> alter session set NLS_COMP=BINARY;
SQL> alter session set NLS_SORT=XGERMAN_AI;

SQL> select * from product where product like 'Ächzen';
```

```
PRODUCT
```

```
-----
```

```
Ächzen
```

```
SQL> alter session set NLS_COMP=Linguistic;
```

```
SQL> select * from product where product like 'Achzen';
```

```
PRODUCT
```

```
-----
```

```
ächzen
```

```
Ächzen
```

# PL/SQL: Datentypen ...



- Datentyp-Constraints nicht hart codieren
  - Subtypen in eigenem Package: Bessere Wartbarkeit
- %TYPE-Deklarationen nutzen
  - Automatische "Bindung" an Tabelle oder View
  - Weniger Datentyp-Konvertierungen
- Neue Datentypen in Betracht ziehen
  - BINARY FLOAT, BINARY DOUBLE, SIMPLE\_INTEGER
- An spezialisierte Datentypen denken ...
  - XMLTYPE
  - SDO\_GEOMETRY
  - ORDIMAGE
  - ...

# PL/SQL: Subtypen



- Deklaration in einem Package

```
create or replace package my_types_package is
  subtype text_line_t is varchar2(80);
  subtype percent_t   is number(5,2);
end my_types_package;
/

declare
  v_percent my_types_package.percent_t;
begin
  :
```

- Und bitte nicht ...

```
create or replace package my_types_package is
  subtype vc2_80_t is varchar2(80);
  subtype num_5_2_t is number(5,2);
end my_types_package;
/
```

# PL/SQL *Named Parameter*



- Übergabe mit Parameter-Namen
- Reihenfolge hat keine Bedeutung mehr
- Ab Oracle 11g R2 auch in SQL nutzbar
- Vorteile ...
  - Klarheit, Übersichtlichkeit
  - Stabilität bei Änderungen der API

```
my_log_pack.log(  
  module    => 'FETCH_CUSTOMERS'  
  ,errcode  => sqlcode  
  ,severity => my_log_pack.ERROR  
);
```

# Mehr Information für den DBA

- DBMS\_APPLICATION\_INFO
  - SET\_ACTION, SET\_CLIENT\_INFO, SET\_MODULE
  - SET\_SESSION\_LONGOPS
- Aufrufbar auch von PHP, JDBC, .NET u.a.

```
SQL> select sid, serial#, module, client_info  
2 from v$session where username is not null;
```

SID	SERIAL#	MODULE	CLIENT_INFO
133	7	SQL*Plus	
159	92	Meine Applikation	APPUSER: Meier

# Mehr Information für den DBA

## V\$SESSION\_LONGOPS

- Mehr Informationen für den DBA ...
  - Beim Session-Monitoring
  - Zum Monitoring langlaufender Operationen
- Oracle nutzt es out-of-the-box
- Auch für eigene Tasks nutzbar

```
SQL> select sid, serial#, sofar / totalwork prozent  
2 from v$session_longops where sofar < totalwork
```

SID	SERIAL#	PROZENT
150	285	50,0

# DML ERROR Logging

- SQL\*Loader-Verhalten für SQL-Kommandos
- INSERT, UPDATE, DELETE und MERGE
- Logging-Tabelle mit DBMS\_ERRLOG erzeugen
- REJECT-Option unterstützt das Zurückrollen des Statements bei Fehleranzahl-Überschreitung



# DML Error Logging

## Beispiel

```
exec DBMS_ERRLOG.CREATE_ERROR_LOG('EMP_T', 'FEHLER')
```

```
INSERT INTO emp_t SELECT empno,hiredate,sal FROM emp  
LOG ERRORS INTO fehler REJECT LIMIT UNLIMITED;
```

```
SELECT id, hiredate, wert, ora_err_mesg$ FROM fehler;
```

```
ID      HIREDATE  WERT  
ORA_ERR_MESG$
```

```
-----  
7566    02-APR-81 2975  
ORA-02290: check constraint (SCOTT.CK) violated
```

# Zugriff auf HTTP-Datenquellen

## Beispiel: RSS-Feed OTN Software Downloads

- 1. Schritt: RSS-Feed als XML abrufen

```
select
  HTTPURITYPE(
    'http://www.oracle.com/technology/syndication/' ||
    'rss_otn_soft.xml'
  ).getxml() as XML_STREAM
from dual

XML_STREAM
-----
<rss version="2.0">
  <channel>
    <title>Oracle New Downloads</title>
    <link>http://www.oracle.com/technology/...
    :
```

# Zugriff auf HTTP-Datenquellen

## Beispiel: RSS-Feed OTN Software Downloads

- 2. Schritt: Elemente abrufen I

```
select
  extractvalue(
    HTTPURITYPE(
      'http://www.oracle.com/technology/...'
    ).getxml(),
    '/rss/channel/title/text()')
  ) rss_titel
from dual

RSS_TITEL
-----
Oracle New Downloads
```

# Zugriff auf HTTP-Datenquellen

## Beispiel: RSS-Feed OTN Software Downloads

- 2. Schritt: Elemente abrufen II

```
select
  extractvalue(value(i), '/item/title') item_title
from table(
  xmlsequence(
    extract(
      HTTPURITYPE(
        'http://www.oracle.com/technology/...'
      ).getxml(),
      '/rss/channel/item'
    )
  )
) i
```

# Zugriff auf HTTP-Datenquellen

## Beispiel: RSS-Feed OTN Software Downloads

- 2. Ergebnis

```
ITEM_TITLE
```

```
-----
```

```
Oracle SOA Suite 10g Release 3 (10.1.3.1.0)
```

```
Oracle Containers for J2EE (10.1.3.1.0)
```

```
Oracle JDeveloper (10.1.3.1.0)
```

```
Oracle Warehouse Builder (10.2.0.1) on IBM zSeri...
```

```
Oracle Database 10g Release 2 (10.2.0.2) for Sol...
```

```
OpenVMS Itanium
```

```
Oracle Content Database (v10.2.0.0.0) for Micros...
```

```
:
```

```
:
```

# Zugriff auf HTTP-Datenquellen

## Beispiel: RSS-Feed OTN Software Downloads

- Anwendungsbeispiele ...
  - Link-Liste in einer Application Express-Anwendung
  - Daten in relationale Tabellen übernehmen

Feed auswählen

Feed auswählen

Inhalte ...

<a href="#">Oracle SOA Suite 10g Release 3 (10.1.3.1.0)</a>
<a href="#">Oracle Containers for J2EE (10.1.3.1.0)</a>
<a href="#">Oracle JDeveloper (10.1.3.1.0)</a>
<a href="#">Oracle Warehouse Builder (10.2.0.1) on IBM zSeries Linux</a>
<a href="#">Oracle Database 10g Release 2 (10.2.0.2) for Solaris Operating System (x86) and OpenVMS Itanium</a>
<a href="#">Oracle Content Database (v10.2.0.0.0) for Microsoft Windows, Solaris SPARC, Linux Itanium, HP, and AIX5L</a>
<a href="#">Oracle JDBC Drivers (v10.2.0.2, 10.1.0.5, and 9.2.08)</a>
<a href="#">Oracle Application Server 10g Release 2 (10.1.2.0.2) for z/Linux, IBM Power Linux</a>
<a href="#">Oracle Enterprise Manager 10g Grid Control Management Agent (v10.2.0.2) for multiple platforms</a>
<a href="#">Oracle Identity Management 10g (10.1.4.0.1) for Microsoft Windows, Linux, Solaris SPARC, and others</a>
<a href="#">Oracle Instant Client (v10.2.0.2) Multiple platforms</a>
<a href="#">Oracle Secure Backup (v10.1.0.1) for Linux x86-64, Linux Itanium</a>
<a href="#">Oracle Secure Backup (v10.1.0.2) for Microsoft Windows (32-bit), Linux x86, Solaris SPARC (64-bit)</a>
<a href="#">Oracle Database 10g Release 2 (10.2.0.2) for HP Tru64 UNIX</a>
<a href="#">Oracle Identity Management 10g (10.1.4.0.1)</a>

1 - 15

# Datenbank-JVM: Grenzen überschreiten

- Bestandteil der Datenbank
- Vollständig J2SE-Konform: 100% Java
- Seit Oracle8i
  - Oracle 9.2: Java 1.3
  - Oracle 10.2: Java 1.4
  - Oracle 11.1: Java 1.5
  - Oracle 11.2: Java 1.5
  - Oracle 12.1: ?

# Java Stored Procedures

- Java-Code in der Datenbank
- Quellcode oder Bytecode ...
  - Compilation außerhalb; Bytecode innerhalb der Datenbank
  - Compilation und Bytecode innerhalb der Datenbank
- Java in die Datenbank laden
  - SQL: CREATE JAVA SOURCE
  - Kommandozeile: loadjava, dropjava





# Java Stored Procedures

## Ein einfaches Beispiel



- Java-Klasse einspielen: HelloWorld

```
create or replace java source named "HelloWorldTest" as
```

```
public class HelloWorld {  
    public static void sayHello() {  
        System.out.println("Hello World");  
    }  
  
    public static String getHello(String sName) {  
        return "Hello World " + sName;  
    }  
}  
/  

```

```
alter java source "HelloWorldTest" compile  
/  

```

# Weitere Anwendungsbeispiele

- Verzeichnisinhalt als Tabelle zurückgeben  
<http://plsqlxecoscomm.sourceforge.net/>  
<http://sql-plsql-de.blogspot.com/2007/07/verzeichnisinhalte-anzeigen-mit-sql.html>
- Verzeichnisinhalt in eine Tabelle laden  
<http://sql-plsql-de.blogspot.com/2008/09/ein-einfacher-ansatz-dateien-eines.html>
- IMAP-Postfach auslesen  
<http://sql-plsql-de.blogspot.com/2007/12/emails-aus-einem-postfach-abrufen.html>
- "tkprof" mit SQL ausführen  
<http://sql-plsql-de.blogspot.com/2008/10/aktuelle-session-tracedatei-ansehen-mit.html>
- ZIP-Archiv auspacken  
<http://www.oracle.com/global/de/community/tipps/zip/index.html>
- Web Services aus der Datenbank heraus aufrufen  
<http://www.oracle.com/global/de/community/tipps/webservices-3/index.html>

# Paket für Betriebssystem-Interaktion

## File system interaction with SQL and PL/SQL

### Packages

[FILE\\_PKG](#)  
[FILE\\_TYPE](#)  
[LOB\\_WRITER\\_PLSQL](#)  
[OS\\_COMMAND](#)

**Overview** [Deprecated](#) [Index](#) [Help](#)

*File system interaction with SQL and PL/SQL*

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [FIELD](#) | [METHOD](#)

DETAIL: [FIELD](#) | [METHOD](#)

## File system interaction with SQL and PL/SQL

This project is about file system interaction from the Oracle database using SQL and PL/SQL. Four PL/SQL objects are provided for this purpose:

- **FILE\_TYPE** represents an operating system file
- **FILE\_PKG** is a helper package to obtain one or multiple file handles
- **OS\_COMMAND** is a package to execute shell commands
- **LOB\_WRITER\_PLSQL** is a helper package to write LOBs (CLOB, BLOB) to operating system files with pure PL/SQL and "traditional" directory objects

### Usage examples:

1. executing a shell command returning text output (ls -la)

```
SQL> select os_command.exec_clob('/bin/ls -la /home/oracle') COMMAND
2* from dual
```

COMMAND

```
-----
insgesamt 121920
drwx----- 20 oracle oracle      4096 18. Jan 09:16 .
drwxr-xr-x   3 root  root      4096 24. Apr 2007 ..
-rw-----   1 oracle oracle         0 24. Apr 2007 .autorun.lck
-rw-----   1 oracle oracle    11067 17. Jan 12:17 .bash_history
-rw-r--r--   1 oracle oracle      24 24. Apr 2007 .bash_logout
-rw-r--r--   1 oracle oracle    1342 13. Nov 11:47 .bash_profile
-rw-r--r--   1 oracle oracle     124 24. Apr 2007 .bashrc
-rwx-----   1 oracle oinstall    1583 14. Jan 09:33 calc.sql
drwx-----   3 oracle oracle      4096 24. Apr 2007 .config
-rwx-----   1 oracle oinstall     161  6. Nov 14:49 csv.txt
drwxr-xr-x   4 oracle oinstall    4096  6. Dez 10:49 dbws
:
```

2. creating a directory



[Carsten.Czarski@oracle.com](mailto:Carsten.Czarski@oracle.com)

<http://tinyurl.com/apexcommunity>

<http://sql-plsql-de.blogspot.com>

<http://oracle-text-de.blogspot.com>

<http://oracle-spatial.blogspot.com>

<http://plsqlxecoscomm.sourceforge.net>

<http://plsqlmailclient.sourceforge.net>

Twitter: @cczarski @oraclebudb