

TVDCC & TVDCA – An Introduction

Philipp Salvisberg
Senior Principal Consultant



BASEL BERN BRUGG LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MUNICH STUTTGART VIENNA

■ AGENDA

1. TVDCC – Trivadis PL/SQL & SQL CodeChecker
2. TVDCA – Trivadis PL/SQL & SQL CodeAnalyzer
3. SQL Developer Extension
4. Further Information

■ PL/SQL & SQL Coding Guidelines



Coding Guidelines are a crucial part of software development. It is a matter of fact, that code is more often read than written – therefore we should take efforts to ease the work of the reader, which is not necessarily the author.

I am convinced that this standard may be a good starting point for your own guidelines.

Roger Troller
Senior Consultant Trivadis



"Roger and his team have done an excellent job of providing a comprehensive set of clear standards that will undoubtedly improve the quality of your code. If you do not yet have standards in place, you should give strong consideration to using these as a starting point."

Steven Feuerstein

Steven Feuerstein
PL/SQL Evangelist

- Openly available since August 2009
- Download for free from www.trivadis.com



■ Coding Guideline #26

26. Always specify the target columns when executing an insert command.

■ Reason

- Data structures often change.
- Having the target columns in your insert statements will lead to change-resistant code.

■ Examples

```
-- Bad
INSERT INTO messages
    VALUES (l_mess_no, l_mess_typ, l_mess_text);
```

```
-- Good
INSERT INTO messages (mess_no, mess_typ, mess_text)
    VALUES (l_mess_no, l_mess_typ, l_mess_text);
```

■ TVDCC – Command Line Interface

```
processing file 'ESC_USER_ADMIN.pkb'... 346 issues found.
processing file 'ESC_USER_ADMIN.pks'... 129 issues found.
processing file 'ESC_USER_ADMIN_DEFAULT.pkb'... 30 issues found.
processing file 'ESC_USER_ADMIN_DEFAULT.pks'... 6 issues found.
processing file 'ESC_UTIL.pkb'... 193 issues found.
processing file 'ESC_UTIL.pks'... no issues found.
```

Summary:

- Total files: 42
- Total bytes: 1806872
- Total lines: 32062
- Total commands: 5322
- Total statements (PL/SQL): 5800
- Max. cyclomatic complexity: 140
- Total issues: 4668
- Total warnings: 4667
- Total errors: 1
- Total processing time in seconds: 23.496

```
transforming tvdcc_report.xml into tvdcc_report.h
transforming tvdcc_report.xml into tvdcc_report.x
cleanup completed.
```

XML, HTML, Excel
Output

Issue Overview

16.0% **Guideline 62 violated:** Always use parameters or pull in definitions rather than referencing external variables in a local program unit.

7.9% **Guideline 53 violated:** Avoid use of WHEN OTHERS clause in an exception section without any other specific handlers.

7.3% **Guideline 72 violated:** Try to use no more than one RETURN statement within a function.

6.8% **Guideline 27 violated:** Always use table aliases when your SQL statement involves more than one source.

6.7% **Guideline 10 violated:** Try to use subtypes for constructs used often in your application.

5.5% **Guideline 18 violated:** Avoid declaring NUMBER variables or subtypes with no precision.

4.9% **Guideline 01 violated:** Try to label your sub blocks.

4.4% **Guideline 68 violated:** Avoid using a IN OUT parameters as IN / OUT only.

4.3% **Guideline 23 violated:** Always define your VARCHAR2 variables using CHAR SEMANTIC.

3.8% **Guideline 76 violated:** Always prefix ORACLE supplied packages with owner schema name.

3.7% **Guideline 30 violated:** Use BULK OPERATIONS (BULK COLLECT, FORALL) whenever you have to.

3.1% **Guideline 28 violated:** Try to use ANSI-join syntax, if supported by your ORACLE version.

2.7% **Guideline 40 violated:** Always label your loops.

2.4% **Guideline 60 violated:** Try to use named notation when calling program units.

2.1% **Guideline 66 violated:** Always use forward declaration for private functions and procedures.

1.6% **Guideline 58 violated:** Always use a string variable to execute dynamic SQL.

1.2% **Guideline 36 violated:** Try to use CASE rather than DECODE.

0.8% **Guideline 50 violated:** Avoid hard-coded upper or lower bound values with FOR loops.

0.8% **Guideline 67 violated:** Avoid declaring global variables public.

0.8% **Guideline 11 violated:** Never initialize variables with NULL.

0.8% **Guideline 79 violated:** Always use synonyms when accessing objects of another application schema.

0.6% **Guideline 49 violated:** Avoid use of unreferenced FOR loop indexes.

0.5% **Guideline 07 violated:** Avoid nesting comment blocks.

0.5% **Guideline 16 violated:** Avoid using overly short names for declared or implicitly declared identifiers.

0.5% **Guideline 26 violated:** Always specify the target columns when executing an insert command.

0.5% **Guideline 56 violated:** Avoid unhandled exceptions.

0.4% **Guideline 61 violated:** Always add the name of the program unit to its end keyword.

0.3% **Guideline 22 violated:** Never use zero-length strings to substitute NULL.

0.2% **Guideline 70 violated:** Avoid using RETURN statements in a PROCEDURE.

0.2% **Guideline 13 violated:** Avoid initializing variables using functions in the declaration section.

Console

■ TVDCC – Trivadis PL/SQL & SQL CodeChecker

- Command line utility to check Oracle SQL*Plus files for compliance violations of the Trivadis PL/SQL & SQL Coding Guidelines Version 2.0, which are downloadable free of charge from www.trivadis.com.
- Furthermore McCabe's cyclomatic complexity and some other software metrics are calculated for each PL/SQL unit and aggregated on file level.
- The code checking results are stored in XML, HTML and Excel files for further processing.

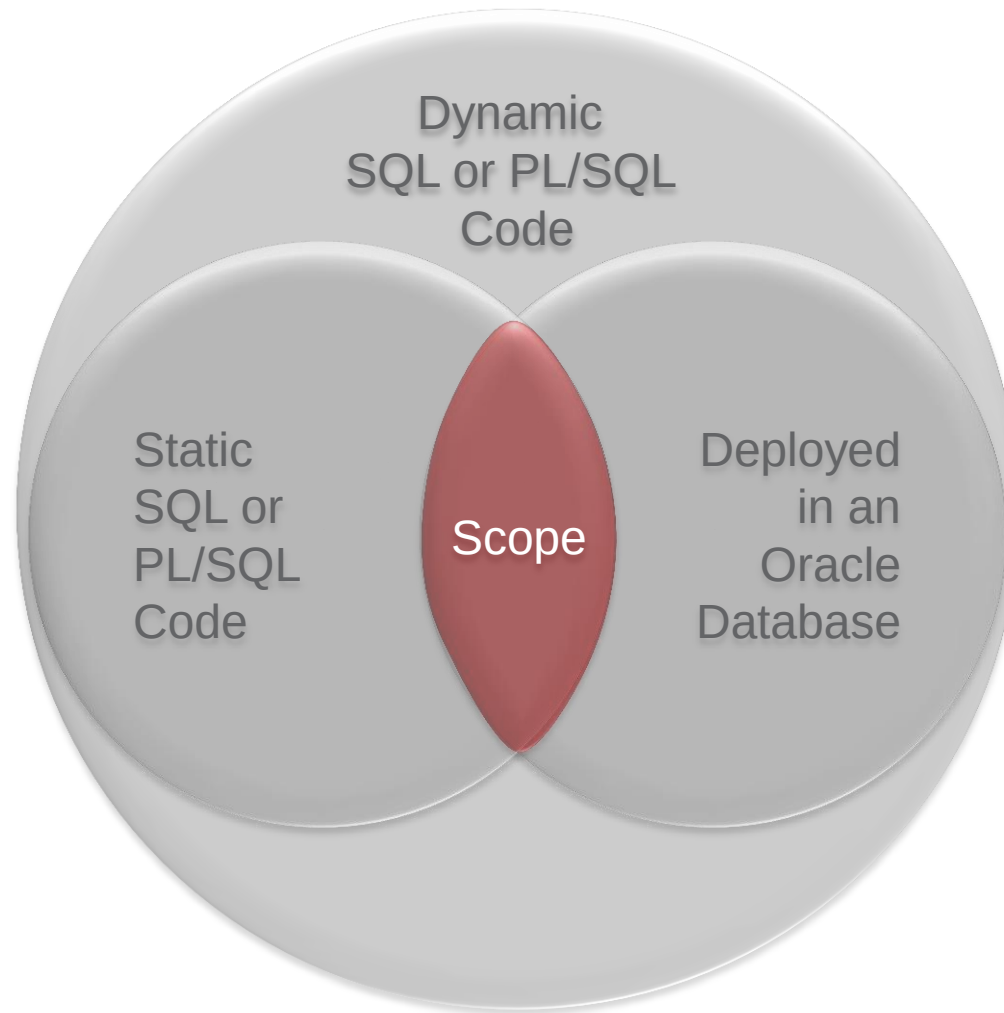
■ AGENDA

1. TVDCC – Trivadis PL/SQL & SQL CodeChecker
2. TVDCA – Trivadis PL/SQL & SQL CodeAnalyzer
3. SQL Developer Extension
4. Further Information

■ Motivation

- Find Object/Subobject Usage
 - Table/view statements (e.g. Select, Insert, Update, Delete, Merge)
 - Package procedures/functions
 - Type methods
 - Table/view columns
- Manage Accessibility of Sensitive Columns
 - Client identifying data
 - Turnover, cost per order/customer, gross margin, discount
 - Nested views, named queries, subqueries – column name changes
- Estimate Impact of Software Design Changes
 - Aspect of design decisions
 - Identify impacted modules (e.g. for testing purposes)

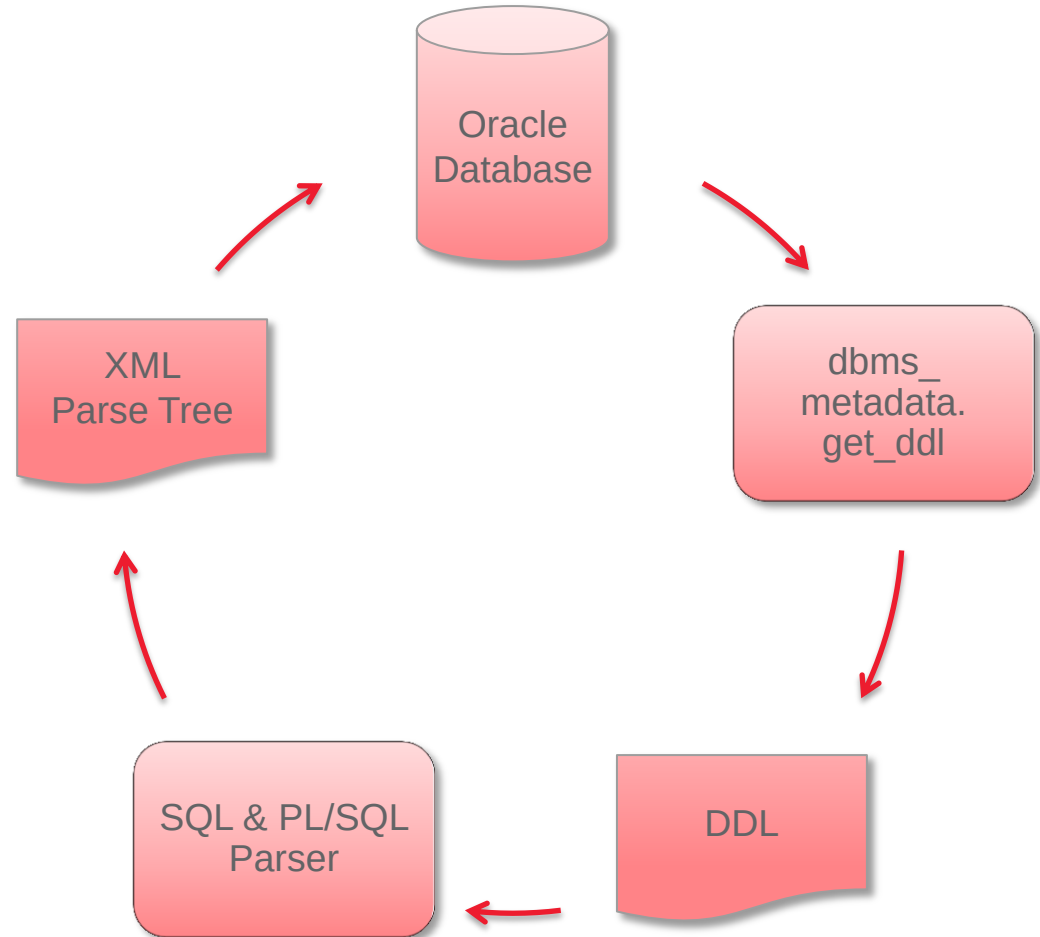
■ Scope of Database Dependency Analysis



■ Extend the Oracle Data Dictionary

```
SQL> desc tvd_parsed_objects_t
```

Name	Type
OBJECT_ID	NUMBER
OWNER	VARCHAR2 (30)
OBJECT_NAME	VARCHAR2 (128)
OBJECT_TYPE	VARCHAR2 (30)
LAST_DDL_TIME	DATE
DDL_SOURCE	CLOB
PARSE_TREE	XMLTYPE



■ TVDCA – Tables, Views used in DML Statements

- Usage within Function, Procedure, Trigger, Package Spec/Body, Type Body
 - Consider Insert, Update, Delete, Merge statements
 - Procedure_name semantics according DBA_PROCEDURES
- Example

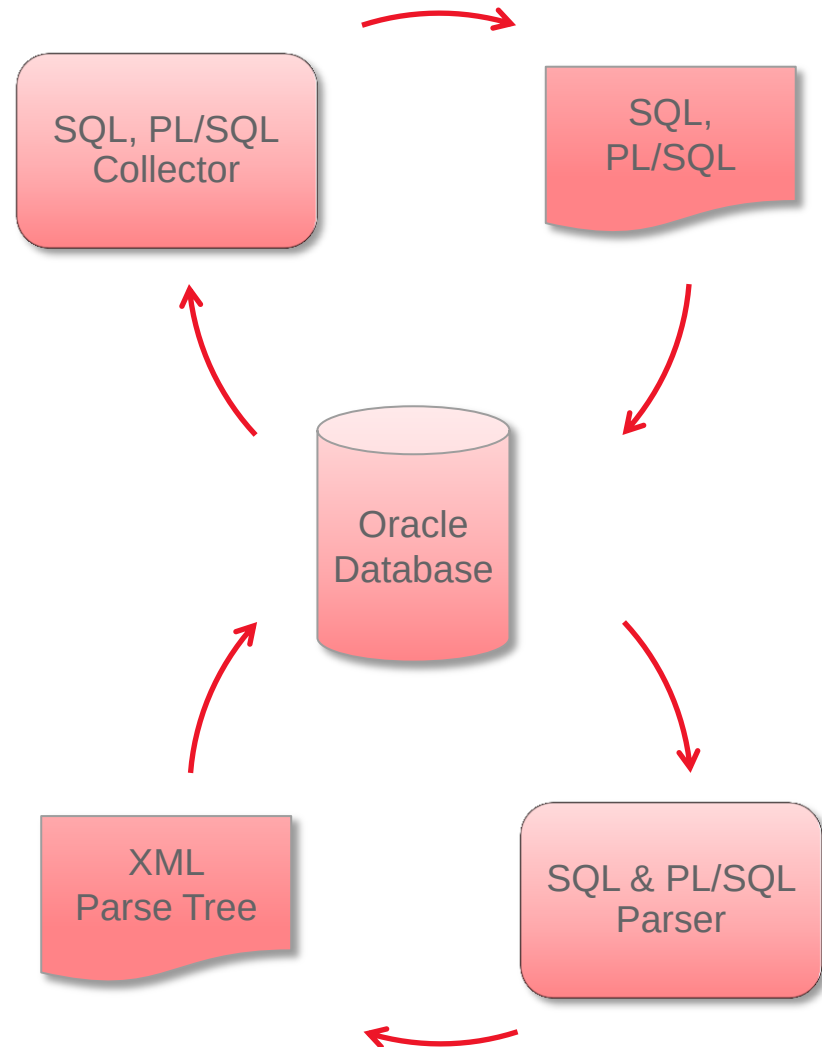
```
SQL> SELECT object_type, object_name, operation AS op, procedure_name,  
2          table_owner AS t_own, table_name  
3          FROM tvd_object_dml_usage_v t  
4          WHERE owner = 'TVDCC';
```

OBJECT_TYPE	OBJECT_NAME	OP	PROCEDURE_NAME	T_OW	TABLE_NAME
PACKAGE BODY	TVD_PARSED_OBJECTS_PKG	INSERT	REFRESH	TVDCC	TVD_PARSED_OBJECTS_T
FUNCTION	TVD_SAMPLE_FUNCTION	INSERT	INNER_PROCEDURE	TVDCC	TVD_PARSED_OBJECTS_T
FUNCTION	TVD_SAMPLE_FUNCTION	INSERT	INNER_FUNCTION	TVDCC	TVD_PARSED_OBJECTS_T
FUNCTION	TVD_SAMPLE_FUNCTION	INSERT		TVDCC	TVD_PARSED_OBJECTS_T
PACKAGE BODY	TVD_SAMPLE_PACKAGE	INSERT	MOST_INNER_PROCEDURE	TVDCC	TVD_PARSED_OBJECTS_T

■ Extend the Scope of Dependency Analysis

```
SQL> desc tvd_captured_sql_t
```

Name	Type
CAP_ID	NUMBER
CAP_SOURCE	CLOB
SQL_ID	VARCHAR2 (13)
USER_NAME	VARCHAR2 (30)
SCHEMA_NAME	VARCHAR2 (30)
MODULE	VARCHAR2 (64)
ACTION	VARCHAR2 (64)
LAST_LOAD_TIME	DATE
...	
PARSE_TREE	XMLTYPE



■ TVDCA – Trivadis PL/SQL & SQL CodeAnalyzer

- Command line utility to parse PL/SQL and SQL source code within an Oracle database and store the resulting XML parse-trees in dedicated relational tables – as an extension to the Oracle Data Dictionary – for further analysis.
- Analysis is supported for static and dynamic code.
- Sample analyses are provided as views and pipelined table functions, e.g.:
 - TVD_OBJECT_HINT_USAGE_V: Used hints in static code.
 - TVD_OBJECT_USAGE_V: Used tables/views in static code.
 - TVD_OBJECT_COL_USAGE_V: Used columns in static code.
 - TVD_COLDEP_PKG: View column dependencies.
 - TVD_SQL_HINT_USAGE_V: Used hints in captured SQL statements.
 - TVD_SQL_USAGE_V: Used tables/views in captured SQL statements.

■ AGENDA

1. TVDCC – Trivadis PL/SQL & SQL CodeChecker
2. TVDCA – Trivadis PL/SQL & SQL CodeAnalyzer
3. SQL Developer Extension
4. Further Information

■ What is the SQL Developer?

- Oracle's free integrated development environment
 - Runs on Windows, Linux, MacOS, etc.
- More and more users are using it
 - As a secondary tool, e.g.
 - DBA tasks
 - Data modeling
 - Data migration (from Microsoft, MySQL, etc.)
 - Working with MySQL, MS database through JDBC
 - etc.
 - As the primary development environment

■ What is the SQL Developer Extension?

- TVDCC integrated into the SQL Developer
- For the currently open editor
 - Check code for guideline violations
 - Navigate quickly through violations
 - Calculate metrics
 - Produce a HTML report based on the editor content
- Free
- Underlines our expertise
- Teaser for our continuous integration solution

Preview

The screenshot displays the Oracle SQL Developer interface. The main window shows a PL/SQL procedure named `limit_repo_size` in the `tvdcapman_etl.sql` file. The procedure includes a loop to calculate repository size and a series of `DELETE` statements for session and database snapshots. On the right, the **Trivadis PL/SQL & SQL Code Checker** pane is open, showing a list of violations. The most prominent error is at line 264: **Guideline 70 violated: Avoid using RETURN statements in a PROCEDURE.** Other violations include Guideline 62 (parameter usage), Guideline 72 (RETURN statements), and Guideline 68 (IN OUT parameters).

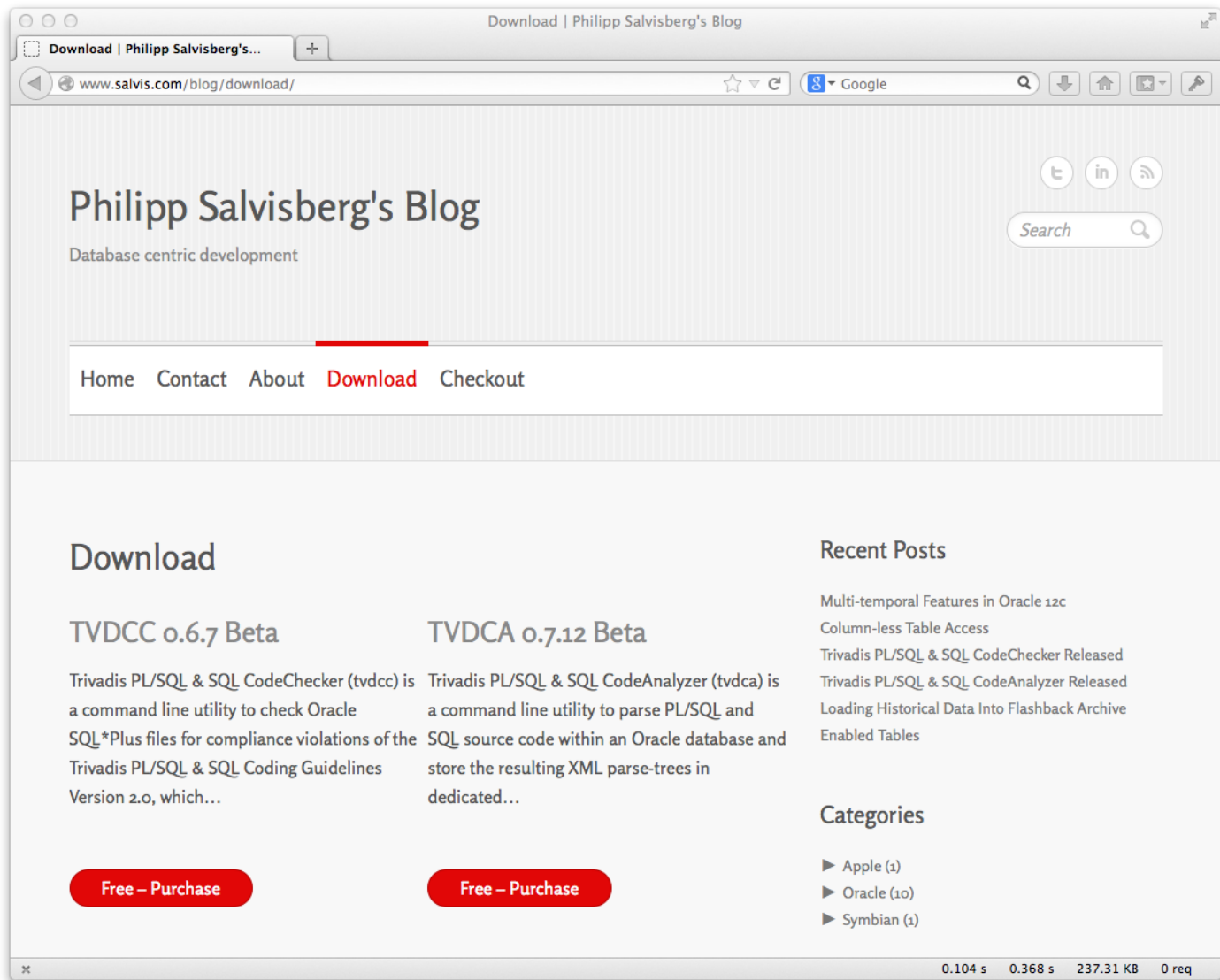
```
251 LOOP
252   RETURN NVL(r.value,1);
253 END LOOP;
254 RETURN 1;
255 END get_cpu_count;
256
257 PROCEDURE limit_repo_size IS
258   n_current_size_mb NUMBER;
259   n_deleted_rows NUMBER;
260   n_limit_size_mb NUMBER;
261 BEGIN
262   SELECT repo_size_th_mb INTO n_limit_size_mb FROM maintenanc
263   IF n_limit_size_mb IS NULL OR n_limit_size_mb=0 THEN
264     RETURN;
265   END IF;
266   SELECT SUM(bytes)/1024/1024 INTO n_current_size_mb FROM use
267   IF n_current_size_mb>n_limit_size_mb THEN
268     logger('INFO: repo size limit of '||n_limit_size_mb||'MB
269     logger('INFO: current repo size is '||n_current_size_mb||
270     DELETE session_stat WHERE snap_id<=(SELECT MAX(snap_id) F
271     n_deleted_rows:=sql%rowcount;
272     DELETE session_snap WHERE snap_date<(SELECT TRUNC(MIN(snc
273     n_deleted_rows:=sql%rowcount;
274     DELETE database_stat WHERE snap_id<=(SELECT MAX(snap_id)
275     n_deleted_rows:=sql%rowcount;
276     DELETE database_snap WHERE snap_date<(SELECT TRUNC(MIN(sr
277     n_deleted_rows:=sql%rowcount;
278     DELETE server_stat WHERE snap_id<=(SELECT MAX(snap_id) FR
279     n_deleted_rows:=sql%rowcount;
```

Line number	Message
250	Guideline 62 violated: Always use parameters or pull in definitions rather t...
250	Guideline 62 violated: Always use parameters or pull in definitions rather t...
250	Guideline 62 violated: Always use parameters or pull in definitions rather t...
250	Guideline 62 violated: Always use parameters or pull in definitions rather t...
250	Guideline 62 violated: Always use parameters or pull in definitions rather t...
250	Guideline 62 violated: Always use parameters or pull in definitions rather t...
250	Guideline 62 violated: Always use parameters or pull in definitions rather t...
252	Guideline 72 violated: Try to use no more than one RETURN statement withi...
254	Guideline 72 violated: Try to use no more than one RETURN statement withi...
257	Guideline 66 violated: Always use forward declaration for private functions ...
258	Guideline 18 violated: Avoid declaring NUMBER variables or subtypes with ...
259	Guideline 18 violated: Avoid declaring NUMBER variables or subtypes with ...
260	Guideline 18 violated: Avoid declaring NUMBER variables or subtypes with ...
262	Guideline 56 violated: Avoid unhandled exceptions.
264	Guideline 70 violated: Avoid using RETURN statements in a PROCEDURE.
285	Guideline 53 violated: Avoid use of WHEN OTHERS clause in an exception s...
289	Guideline 66 violated: Always use forward declaration for private functions ...
289	Guideline 68 violated: Avoid using a IN OUT parameters as IN / OUT only.
291	Guideline 18 violated: Avoid declaring NUMBER variables or subtypes with ...
293	Guideline 56 violated: Avoid unhandled exceptions.
294	Guideline 40 violated: Always label your loops.
297	Guideline 72 violated: Try to use no more than one RETURN statement withi...
300	Guideline 72 violated: Try to use no more than one RETURN statement withi...
301	Guideline 53 violated: Avoid use of WHEN OTHERS clause in an exception s...
303	Guideline 72 violated: Try to use no more than one RETURN statement withi...
306	Guideline 66 violated: Always use forward declaration for private functions ...
306	Guideline 68 violated: Avoid using a IN OUT parameters as IN / OUT only.
307	Guideline 68 violated: Avoid using a IN OUT parameters as IN / OUT only.
308	Guideline 68 violated: Avoid using a IN OUT parameters as IN / OUT only.
309	Guideline 68 violated: Avoid using a IN OUT parameters as IN / OUT only.
310	Guideline 68 violated: Avoid using a IN OUT parameters as IN / OUT only.
311	Guideline 68 violated: Avoid using a IN OUT parameters as IN / OUT only.

■ AGENDA

1. TVDCC – Trivadis PL/SQL & SQL CodeChecker
2. TVDCA – Trivadis PL/SQL & SQL CodeAnalyzer
3. SQL Developer Extension
4. Further Information

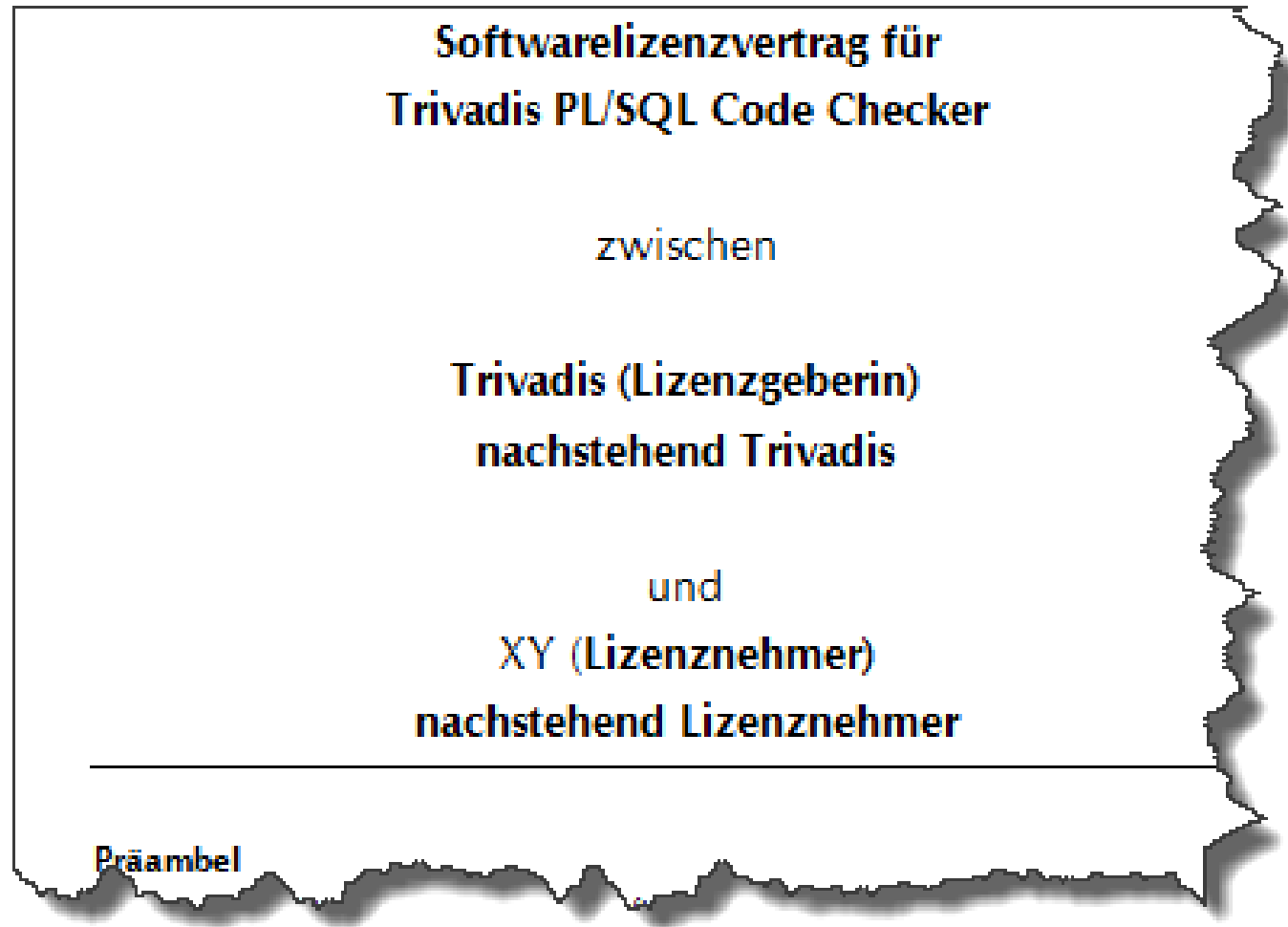
■ Download Preview/Trial Version



■ Price List TVDCC

Mietmodell - Miete pro Jahr					
# User	Einzelpreis	Berechneter Preis	Listenpreis	Listenpreis pro User	
1	€ 1,200.00	€ 1,200.00	€ 1,200.00	€	1,200.00
10	€ 840.00	€ 8,400.00	€ 9,000.00	€	900.00
20	€ 588.00	€ 11,760.00	€ 12,500.00	€	625.00
40	€ 411.60	€ 16,464.00	€ 17,500.00	€	437.50
80	€ 288.12	€ 23,049.60	€ 25,000.00	€	312.50
Lizenz					
# User	Einzelpreis	Berechneter Preis	Listenpreis	Listenpreis pro User	
1	€ 3,400.00	€ 3,400.00	€ 3,400.00	€	3,400.00
10	€ 2,108.00	€ 21,080.00	€ 22,000.00	€	2,200.00
20	€ 1,306.96	€ 26,139.20	€ 27,500.00	€	1,375.00
40	€ 810.32	€ 32,412.61	€ 33,500.00	€	837.50
80	€ 502.40	€ 40,191.63	€ 43,000.00	€	537.50

■ License Template



Questions and answers ...

Philipp Salvisberg
Senior Principal Consultant

philipp.salvisberg@trivadis.com



BASEL BERN BRUGES LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MUNICH STUTTGART VIENNA