WELCOME

Checking Compliance with Custom Guidelines for PL/SQL Code

Philipp Salvisberg

22nd September 2011

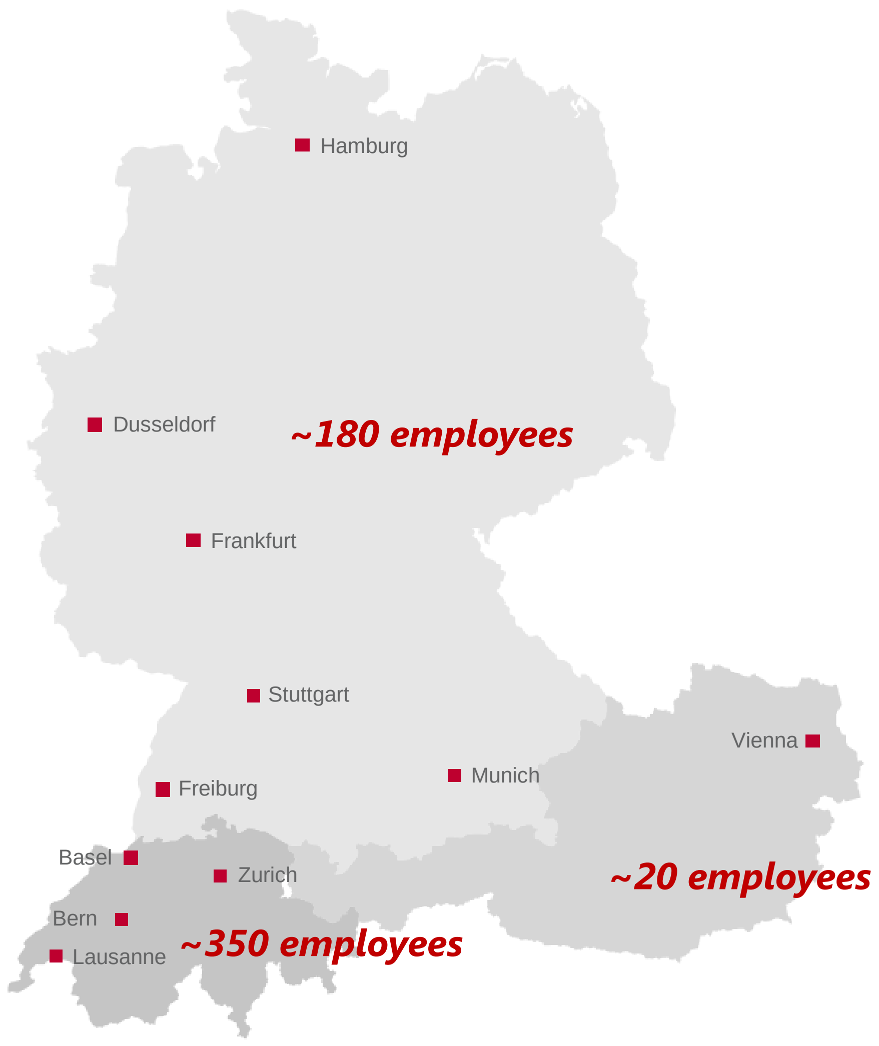
BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

trivadis
makes IT easier.

About Me

- A Trivadian since April 2000
  - Principle Consultant, Partner
  - Member of the Board of Directors
  - Bachelor of Science in Business Administration
  - philipp.salvisberg@trivadis.com
  - www.trivadis.com
- Member of the trivadis performanceteam
- Main focus on database centric development with Oracle DB
  - Application Performance Management
  - Application Development
  - Business Intelligence
- Over 20 years experience in using Oracle products

Trivadis facts & figures



11 Trivadis locations with more than 550 employees

Financially independent and sustainably profitable

Key figures 2010

- Revenue CHF 101 / EUR 73 Mio.
- Services for more than 700 clients in over 1,800 projects
- Over 170 Service Level Agreements
- More than 5,000 training participants
- Research and development budget: CHF 5.0 / EUR 3.6 Mio.

trivadis
makes IT easier.

AGENDA

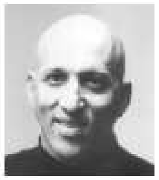
1. Introduction
2. Xtext Live – Parsing & Validating
3. Finalizing Grammar, Checks and Tooling
4. Continuous Integration
5. Challenges
6. Conclusion

trivadis
makes IT easier.

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

See http://www.trivadis.com/technologie/oracle/oracle-application-development/oracle-sql-und-plsql.html

Trivadis PL/SQL & SQL Guideline #25

PL/SQL
& SQL

CODING GUIDELINES
VERSION 1.3.1

25. 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.

Example:

```
-- 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 );
```

PL/SQL Assessment

- Code Analysis based on Trivadis SQL & PL/SQL Guidelines
- Cookbook using e.g.
  - Quest CodeXpert
  - SQL Scripts using PLScope
  - SQL Scripts
  - Manual checks
  - Interviews
- Final Report
  - Results
  - Recommendations
- Fixed Price Offering

SWISS *IT* UP!

Trivadis AG
Europa-Strasse 5
CH-8152 Zürich (Glattbrugg)
www.trivadis.com

ORACLE Platinum Partner

EXPERIENCE *IT*

GARANTIERTE SCHWEIZER QUALITÄT – EIN GUTES GEFÜHL!

Trivadis ist ein erfolgreiches Schweizer Unternehmen für IT-Lösungen und -Dienstleistungen mit über 15 Jahren Erfahrung und mehr als 550 Mitarbeitern in der Schweiz, Deutschland und Österreich!

Auszug aus unserer Kundenliste:
ASTRA – Bundesamt für Strassen, Bixi Systems, BMW AG, Boehringer Ingelheim, Deutsche Lufthansa, ETH Zürich, FIFA Zürich, Groupe Mutuel, Novartis Pharma, PEARL Agency GmbH, PostFinance AG, Raiffeisen, SWISS International Air Lines AG, Syngenta, UBS AG

trivadis makes IT easier.

TRIVADIS PL/SQL ASSESSMENT

SWISS *IT* UP!

USE *IT*

UNSERE GUIDELINES – JETZT DOWNLOADEN.

Unsere «PL/SQL und SQL Coding Guidelines» ermöglichen Ihnen eine optimale und standardisierte Codierung von PL/SQL Anwendungen.

«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, PL/SQL Evangelist

Kostenloser Download unter:
www.trivadis.com/plsql

LEARN *IT*

VON DEN BESTEN LERNEN.

Profitieren Sie von unseren Best-Practice-Kursen!

Einführung in PL/SQL:
www.trivadis.com/o-pl/sql

PL/SQL für Fortgeschrittene:
www.trivadis.com/o-pl/sql-adv

Alle Kursbewertungen unserer Teilnehmer finden Sie direkt bei den Kursbeschreibungen.

KNOW *IT*

DAS GEBALLTE WISSEN UNSERER PL/SQL CRACKS.

1 Roger Troller, Senior Consultant im Oracle Umfeld, seit über 20 Jahren im IT-Business, Autor der «Trivadis PL/SQL und SQL Guidelines»

2 Perry Pakull, Technology Manager für Oracle Based Development, seit über 20 Jahren im IT-Business

3 Daniel Liebhart, Solution Manager Application Development, fokussiert auf den Bereich «Service Oriented Architecture (SOA)», seit über 25 Jahren im IT-Business

CHECK *IT* – PL/SQL

DAS ASSESSMENT FÜR IHRE PL/SQL ANWENDUNG!

Lassen Sie Ihre PL/SQL Anwendung auf Qualität, Wartbarkeit und Optimierungspotenzial checken – zum Fixpreis von CHF 5000.– / EUR 3000.–.

Unser umfassendes Assessment beruht auf unseren «PL/SQL und SQL Guidelines».

Mehr Infos unter:
www.trivadis.com/plsql

GET *IT*

PROFITIEREN SIE VON UNSEREM PL/SQL ASSESSMENT!

Ihre Vorteile:
- Klare Aussagen zur Qualität des Source Codes nach den «PL/SQL und SQL Guidelines» von Trivadis
- Klare Empfehlungen hinsichtlich qualitätssichernder Massnahmen
- Abschlussbericht mit Ergebnissen und Empfehlungen
- Deutlich einfachere Wartungs- und Weiterentwicklungsmöglichkeiten
- Steigerung der Code-Flexibilität
- Bessere Produktivität zur Bereitstellung datennaher Funktionen
- Mehr Transparenz in zentralen Applikationen

Fixpreis: CHF 5000.–/EUR 3000.–
(Für maximal 10000 Lines of Code oder 3 Arbeitstage)

DO *IT*

LOS GEHT'S: NEHMEN SIE JETZT KONTAKT AUF!

www.trivadis.com/plsql

Direkter Kontakt:
Trivadis AG
Perry Pakull
assessment@trivadis.com
Tel. 0800 87 48 23 47
(Kostenfreie Rufnummer)

See http://www.trivadis.com/technologie/swiss-it-up/plsql-assessment.html

Shortcoming of PL/SQL Assessment

- Some guidelines check scripts need manual post-processing
- Some guidelines checks are not automated at all
- One snapshot – Assessment of a defined release
- Repetitive execution is time-consuming, expensive, not feasible
- Not part of an automated, continuous integration strategy

Goal

- Fully automated code checking
- Considering the Trivadis PL/SQL & SQL Guidelines
- Extendable and adaptable to suit customer needs
- Part of an automated build process

Approach & Considerations

- Requirements
  - Parser to process SQL*Plus files
  - Code checking framework
- Options
  - SQL & PL/SQL grammar as part of Oracle JDeveloper Extensions
    - http://www.oracle.com/technetwork/developer-tools/jdev/index-099997.html, see class oracle.javatools.parser.plsql.PlsqlParser
    - Required libraries (javatools-nodeps.jar) are part of SQL Developer
  - ANTLR
    - Several SQL & PL/SQL grammars on http://www.antlr.org/grammar/list
  - Eclipse Xtext
    - Framework for development of textual domain specific languages (DSL)
    - Used successfully to generate database access layer for bitemporal tables
    - Uses ANTLR behind the scenes

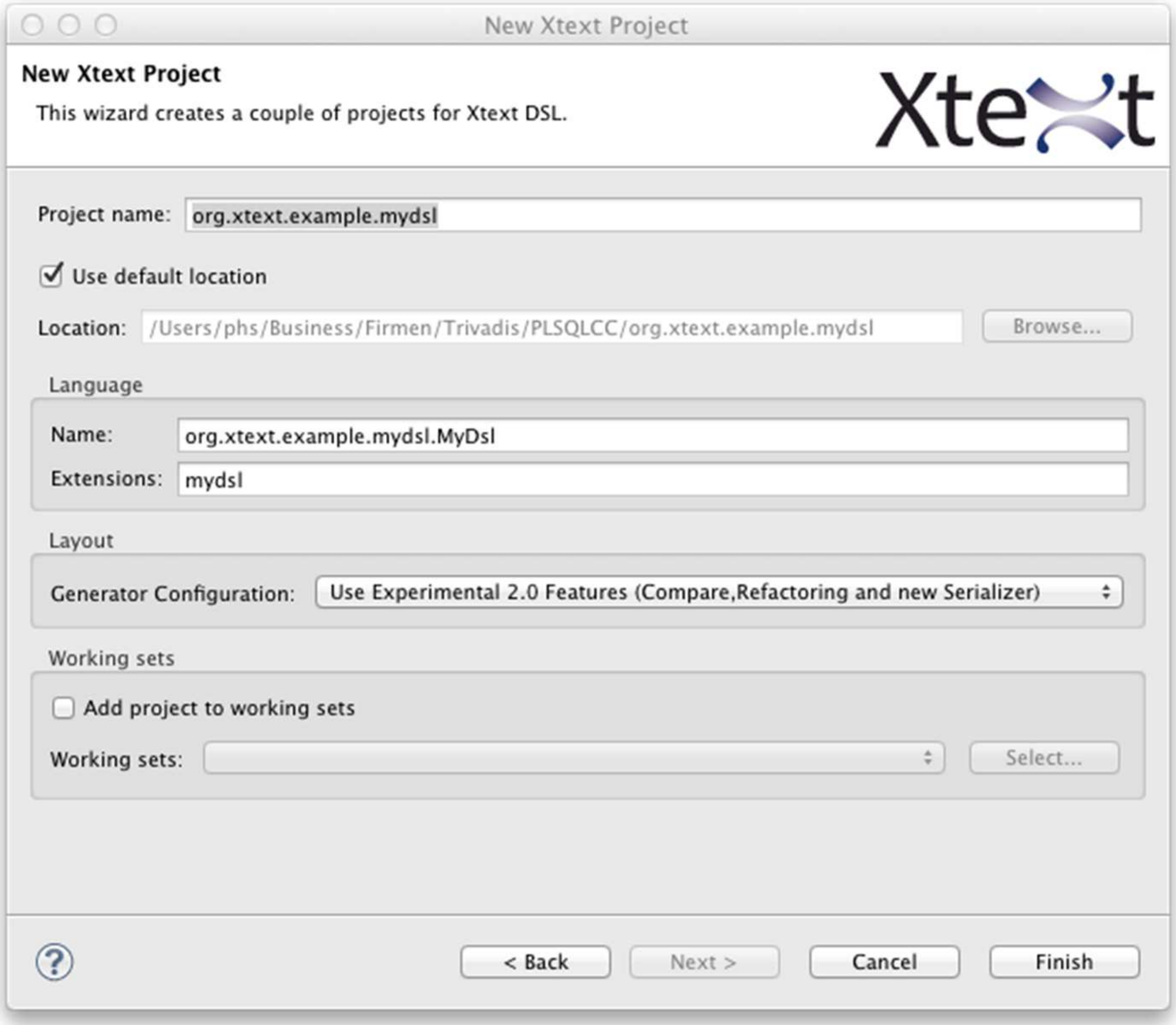
Xtext Features

- Eclipse-based Editors
  - Validation and Quick Fixes
  - Syntax Coloring
  - Code Completion
  - Outline View
  - Code Formatting
  - Bracket Matching
- Integration
  - Eclipse Modeling Framework (e.g. for graphical editors)
  - Eclipse Workbench (e.g. for problems)
  - Export into self-executing JAR (e.g. to build a command-line utility)

AGENDA

Default Xtext Project

DEMO



New Xtext Project

New Xtext Project

This wizard creates a couple of projects for Xtext DSL.

Project name: org.xtext.example.mydsl

☑ Use default location

Location: /Users/phs/Business/Firmen/Trivadis/PLSQLCC/org.xtext.example.mydsl

Language

Name: org.xtext.example.mydsl.MyDsl

Extensions: mydsl

Layout

Generator Configuration: Use Experimental 2.0 Features (Compare,Refactoring and new Serializer)

Working sets

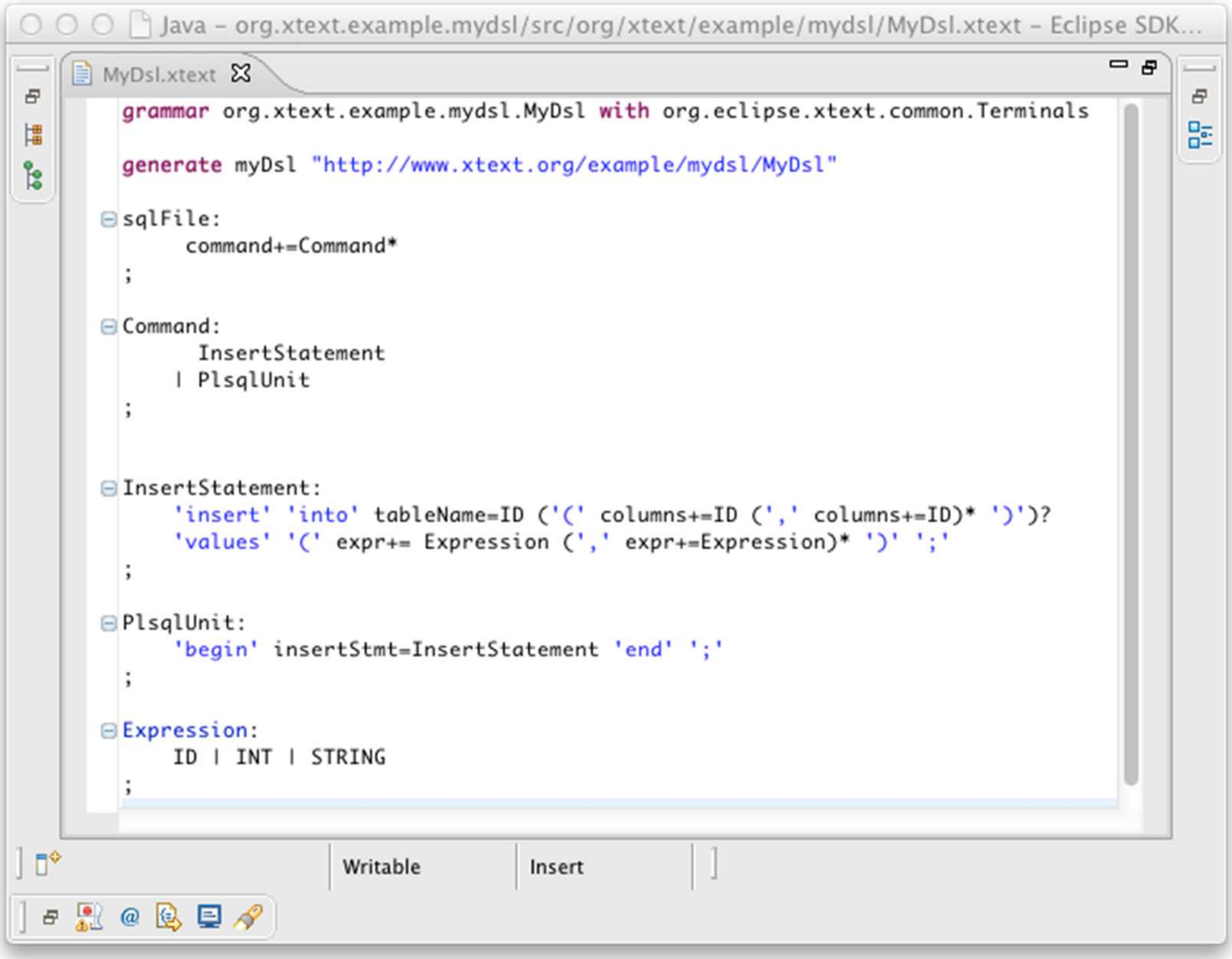
☐ Add project to working sets

Working sets:

trivadis makes IT easier.

Simplified Grammar

DEMO

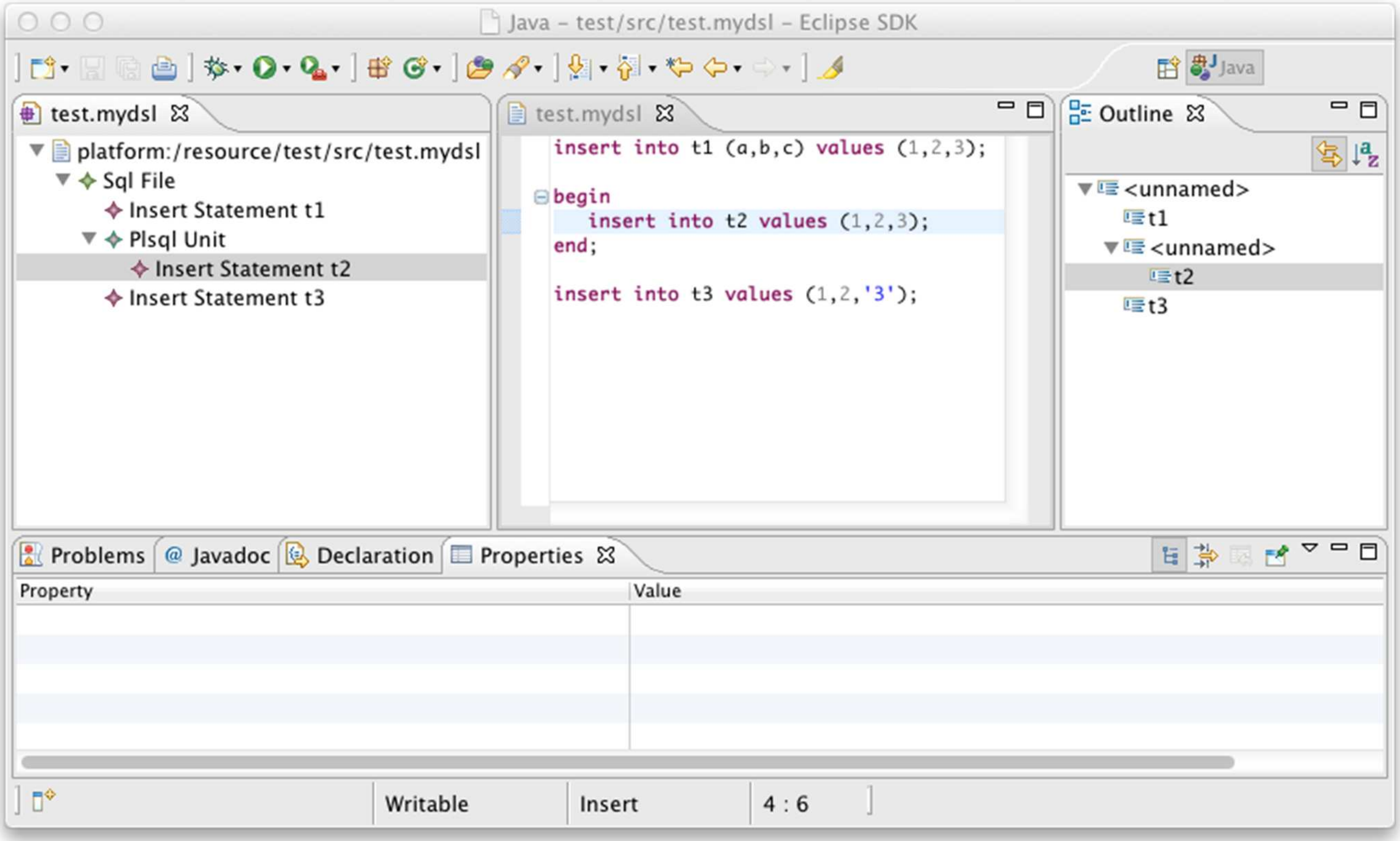

Java – org.xtext.example.mydsl/src/org/xtext/example/mydsl/MyDsl.xtext – Eclipse SDK...
MyDsl.xtext
Writable
Insert

```
grammar org.xtext.example.mydsl.MyDsl with org.eclipse.xtext.common.Terminals

generate myDsl "http://www.xtext.org/example/mydsl/MyDsl"

sqlFile:
      command+=Command*
;

Command:
       InsertStatement
     | PlsqlUnit
;


InsertStatement:
     'insert' 'into' tableName=ID ('(' columns+=ID (',' columns+=ID)* ')')?
     'values' '(' expr+= Expression (',' expr+=Expression)* ')' ';'
;

PlsqlUnit:
     'begin' insertStmt=InsertStatement 'end' ';'
;

Expression:
     ID | INT | STRING
;
```

trivadis makes IT easier.

Eclipse Editors

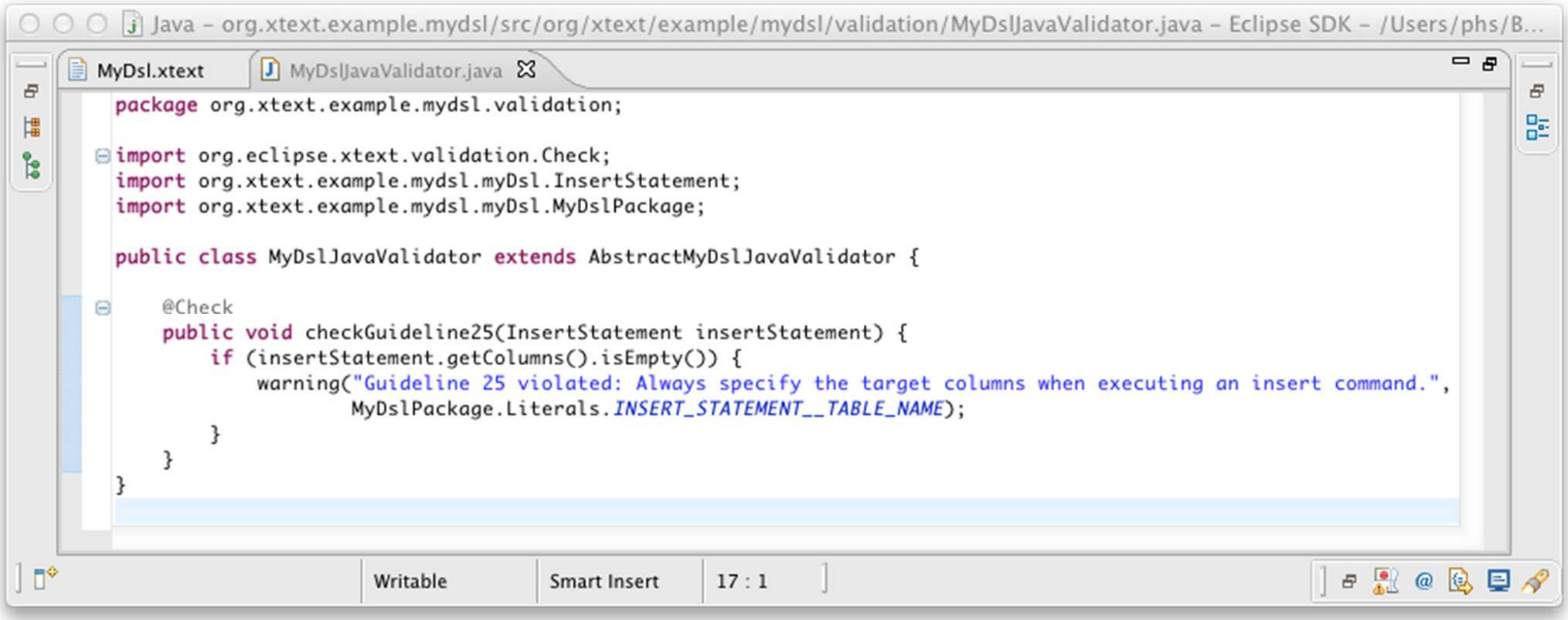
DEMO



```
insert into t1 (a,b,c) values (1,2,3);

begin
   insert into t2 values (1,2,3);
end;

insert into t3 values (1,2,'3');
```

Validator for Guideline #25

DEMO

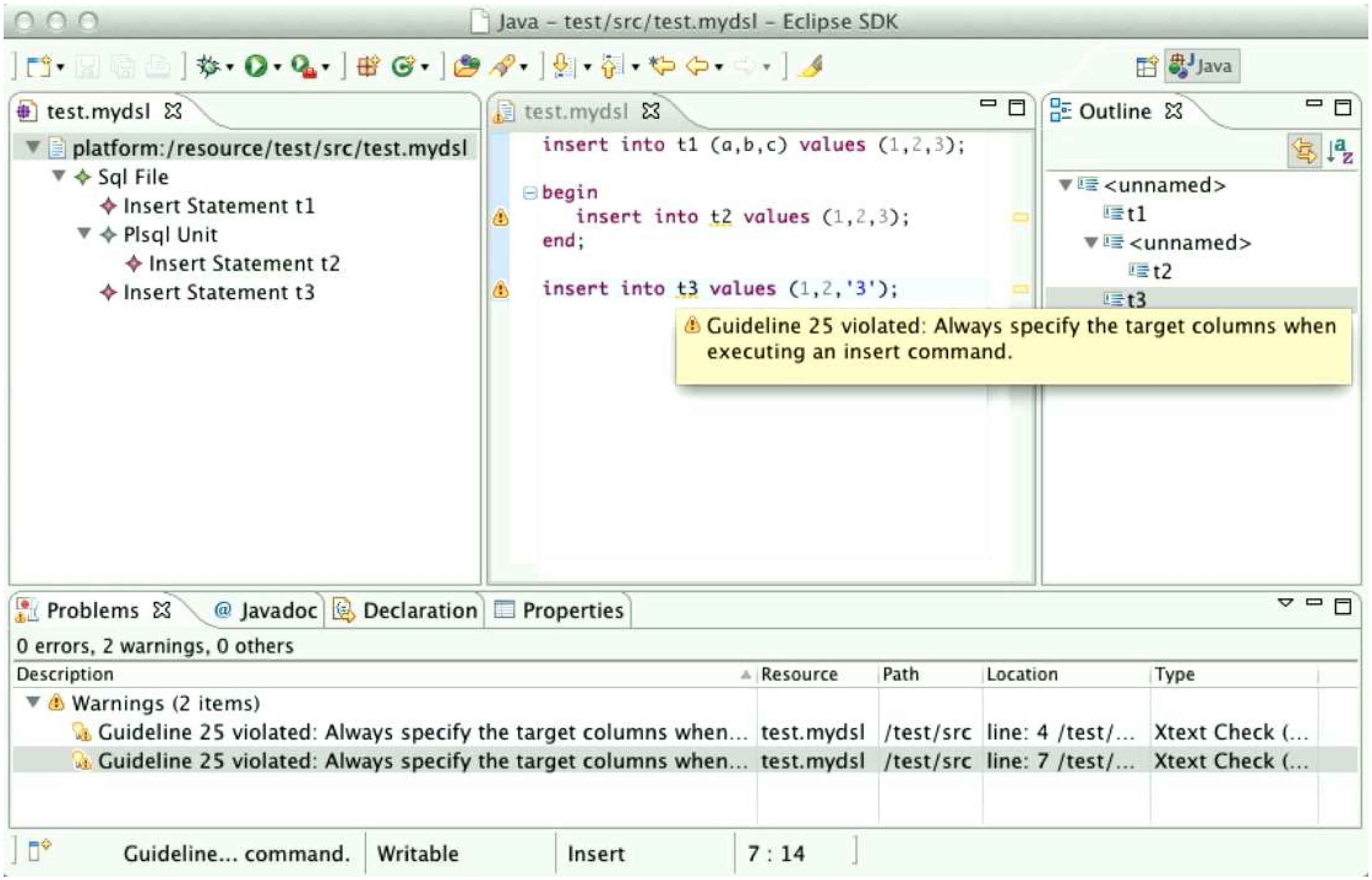

Java – org.xtext.example.mydsl/src/org/xtext/example/mydsl/validation/MyDslJavaValidator.java – Eclipse SDK – /Users/phs/B...
MyDsl.xtext
MyDslJavaValidator.java
Writable
Smart Insert
17 : 1

```java
package org.xtext.example.mydsl.validation;

import org.eclipse.xtext.validation.Check;
import org.xtext.example.mydsl.myDsl.InsertStatement;
import org.xtext.example.mydsl.myDsl.MyDslPackage;

public class MyDslJavaValidator extends AbstractMyDslJavaValidator {

    @Check
    public void checkGuideline25(InsertStatement insertStatement) {
        if (insertStatement.getColumns().isEmpty()) {
            warning("Guideline 25 violated: Always specify the target columns when executing an insert command.",
                    MyDslPackage.Literals.INSERT_STATEMENT__TABLE_NAME);
        }
    }
}
```

trivadis
makes IT easier.

Validator in Action

DEMO



```
insert into t1 (a,b,c) values (1,2,3);

begin
   insert into t2 values (1,2,3);
end;

insert into t3 values (1,2,'3');
```

Guideline 25 violated: Always specify the target columns when executing an insert command.

0 errors, 2 warnings, 0 others

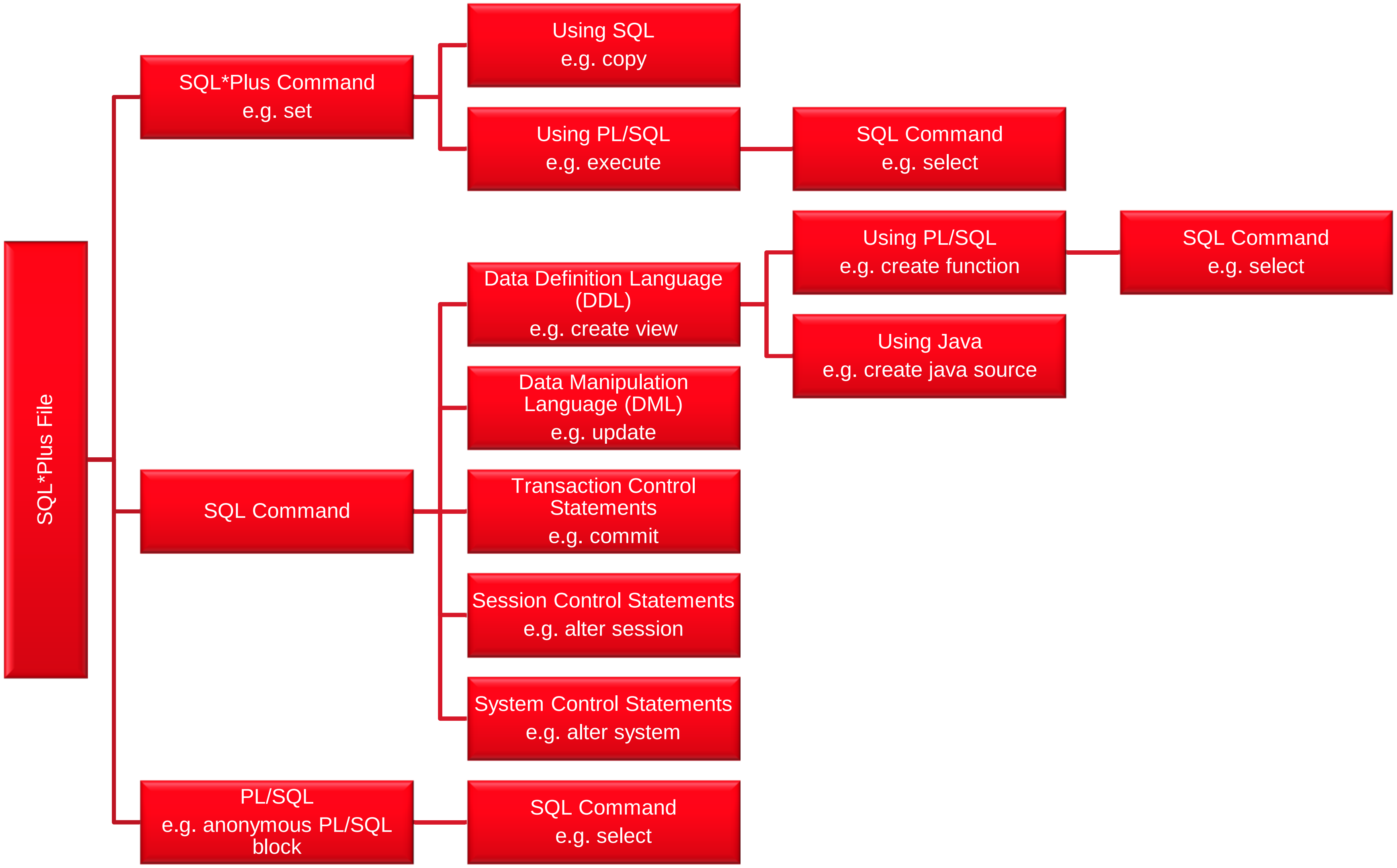
Description	Resource	Path	Location	Type
Warnings (2 items)				
Guideline 25 violated: Always specify the target columns when...	test.mydsl	/test/src	line: 4 /test/...	Xtext Check (...
Guideline 25 violated: Always specify the target columns when...	test.mydsl	/test/src	line: 7 /test/...	Xtext Check (...

Guideline... command. | Writable | Insert | 7 : 14

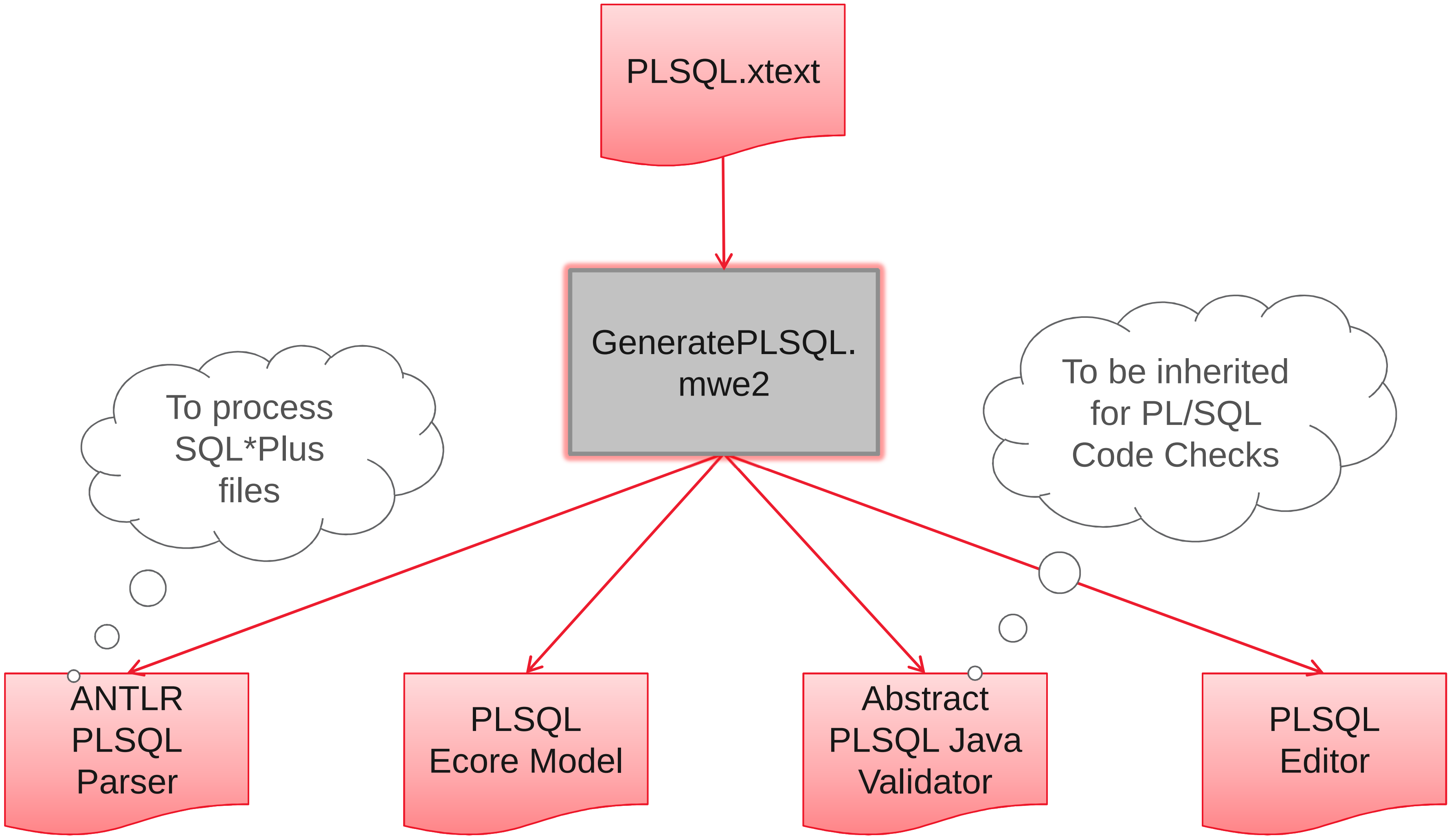
trivadis makes IT easier.

AGENDA

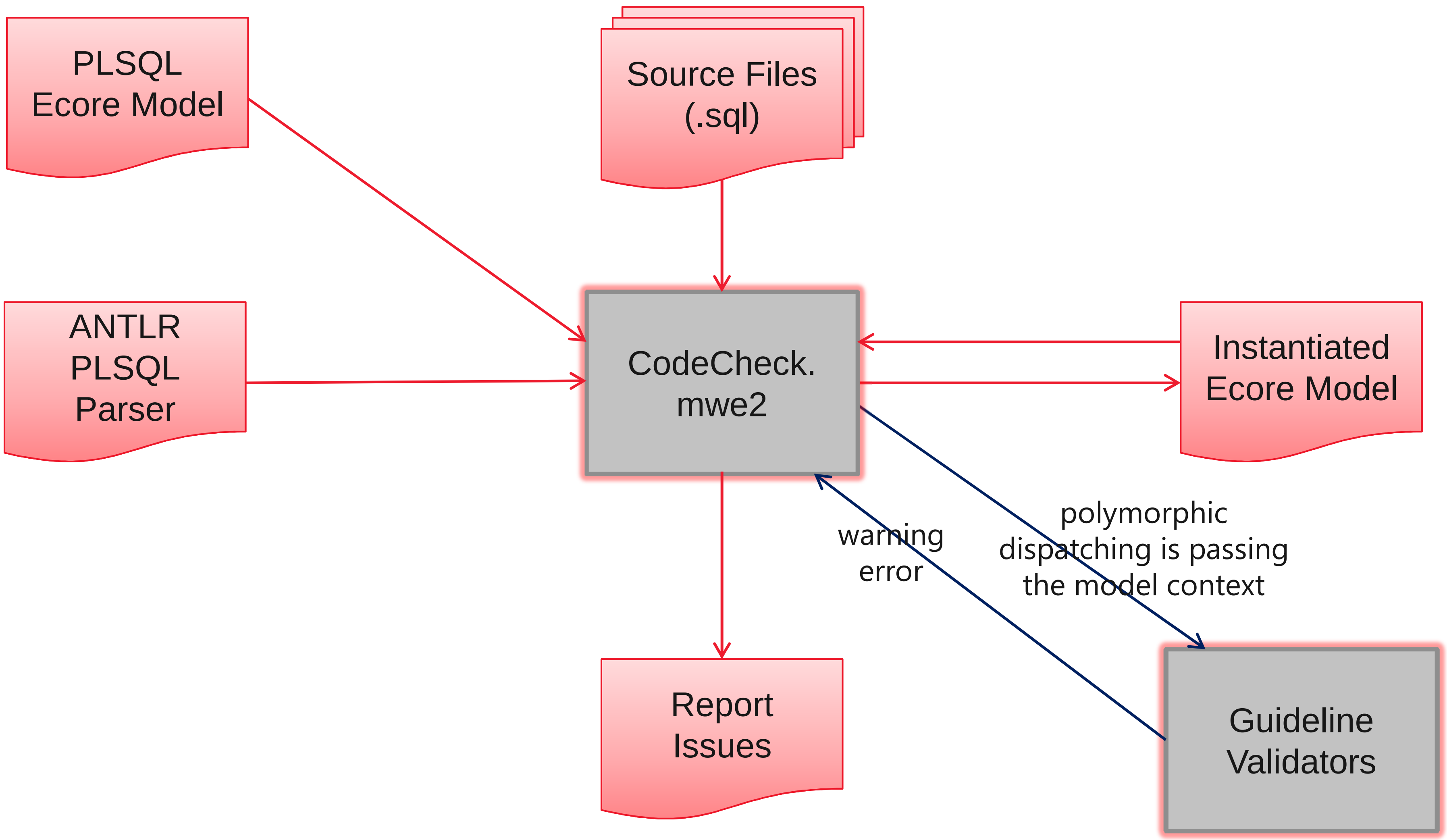
Content of a SQL*Plus File


SQL*Plus File
SQL*Plus Command e.g. set
Using SQL e.g. copy
Using PL/SQL e.g. execute
SQL Command e.g. select
SQL Command
Data Definition Language (DDL) e.g. create view
Using PL/SQL e.g. create function
SQL Command e.g. select
Using Java e.g. create java source
Data Manipulation Language (DML) e.g. update
Transaction Control Statements e.g. commit
Session Control Statements e.g. alter session
System Control Statements e.g. alter system
PL/SQL e.g. anonymous PL/SQL block
SQL Command e.g. select

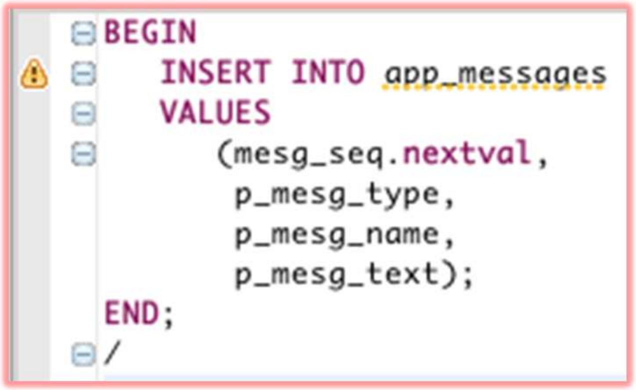
Generate PL/SQL Grammar via Xtext


PLSQL.xtext
GeneratePLSQL.mwe2
To process SQL*Plus files
To be inherited for PL/SQL Code Checks
ANTLR PLSQL Parser
PLSQL Ecore Model
Abstract PLSQL Java Validator
PLSQL Editor

Apply Code Checks (via Command Line)

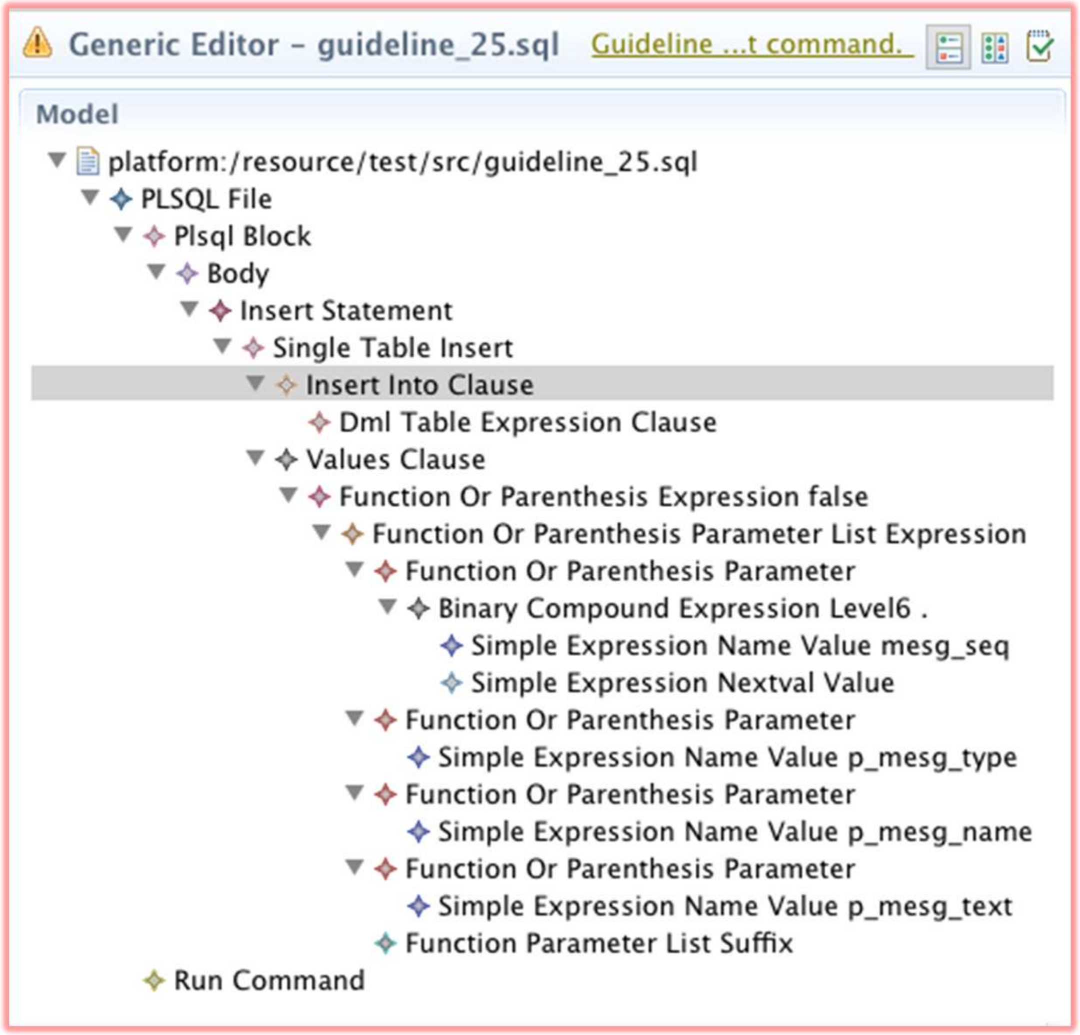


Source, Model & Warning for Guideline #25



```
BEGIN
    INSERT INTO app_messages
    VALUES
       (mesg_seq.nextval,
        p_mesg_type,
        p_mesg_name,
        p_mesg_text);
END;
/
```

line 2 - Guideline 25 violated: Always specify the target columns when executing an insert command.



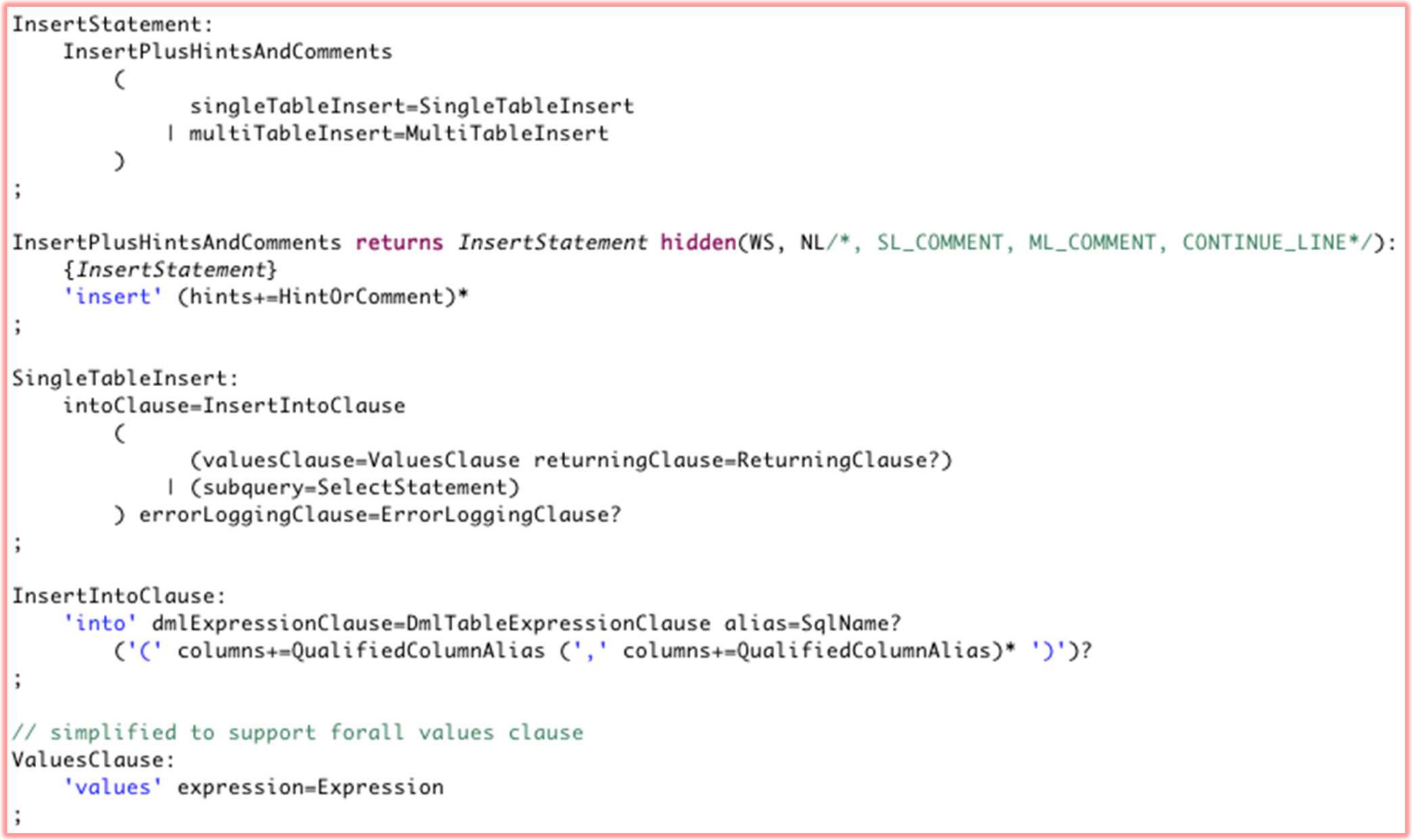
Generic Editor – guideline_25.sql Guideline ...t command.

Model

- platform:/resource/test/src/guideline_25.sql
  - PLSQL File
    - Plsql Block
      - Body
        - Insert Statement
          - Single Table Insert
            - Insert Into Clause
              - Dml Table Expression Clause
            - Values Clause
              - Function Or Parenthesis Expression false
                - Function Or Parenthesis Parameter List Expression
                  - Function Or Parenthesis Parameter
                    - Binary Compound Expression Level6 .
                      - Simple Expression Name Value mesg_seq
                      - Simple Expression Nextval Value
                  - Function Or Parenthesis Parameter
                    - Simple Expression Name Value p_mesg_type
                  - Function Or Parenthesis Parameter
                    - Simple Expression Name Value p_mesg_name
                  - Function Or Parenthesis Parameter
                    - Simple Expression Name Value p_mesg_text
                  - Function Parameter List Suffix
    - Run Command

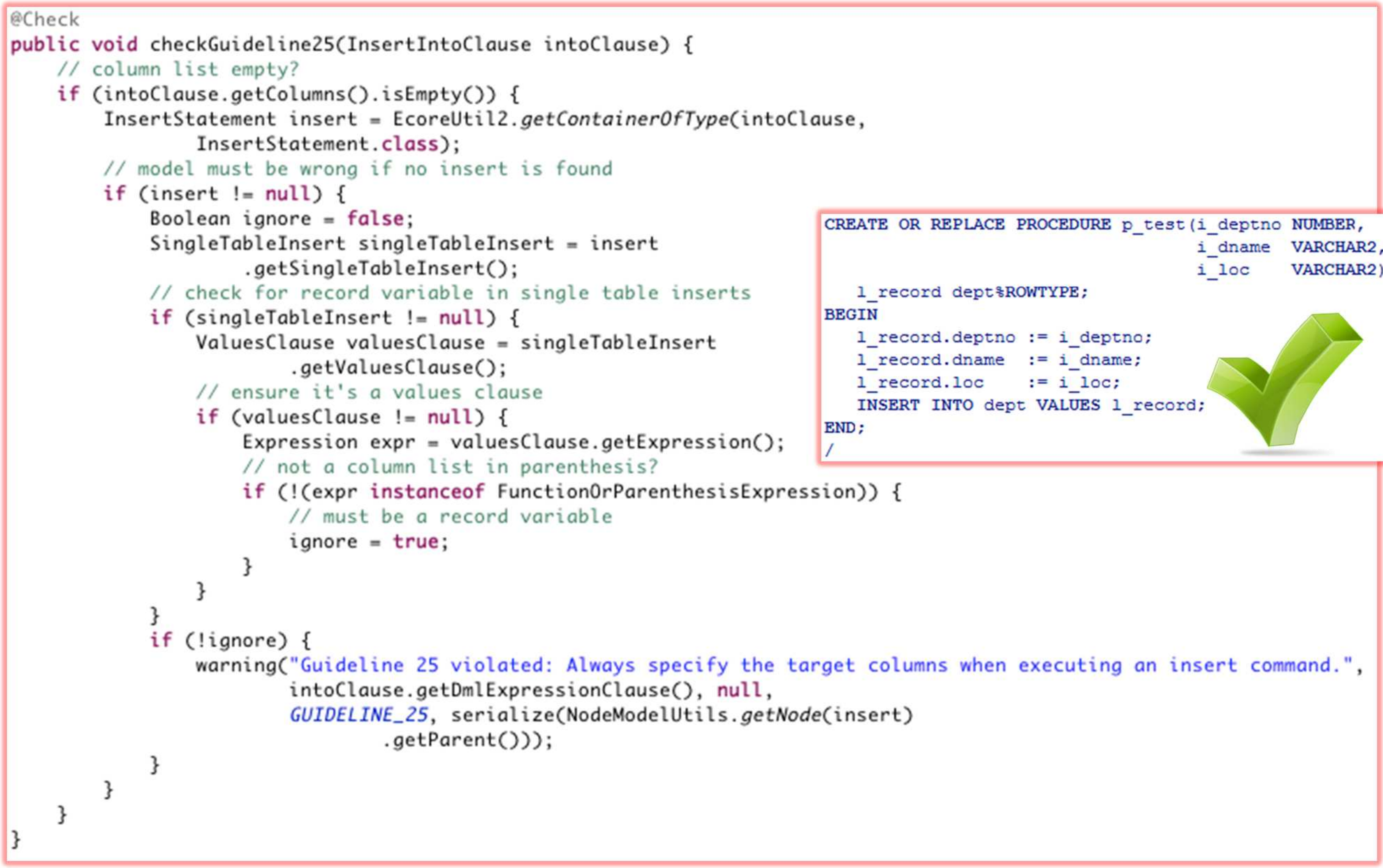
trivadis makes IT easier.

Excerpt of Grammar for Insert Statement

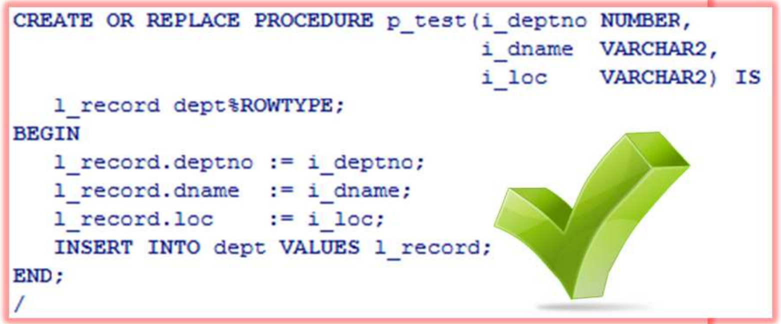


```
InsertStatement:
    InsertPlusHintsAndComments
        (
              singleTableInsert=SingleTableInsert
            | multiTableInsert=MultiTableInsert
        )
;

InsertPlusHintsAndComments returns InsertStatement hidden(WS, NL/*, SL_COMMENT, ML_COMMENT, CONTINUE_LINE*/):
    {InsertStatement}
    'insert' (hints+=HintOrComment)*
;

SingleTableInsert:
    intoClause=InsertIntoClause
        (
              (valuesClause=ValuesClause returningClause=ReturningClause?)
            | (subquery=SelectStatement)
        ) errorLoggingClause=ErrorLoggingClause?
;

InsertIntoClause:
    'into' dmlExpressionClause=DmlTableExpressionClause alias=SqlName?
        ('(' columns+=QualifiedColumnAlias (',' columns+=QualifiedColumnAlias)* ')')?
;

// simplified to support forall values clause
ValuesClause:
    'values' expression=Expression
;
```

Validator for Guideline #25



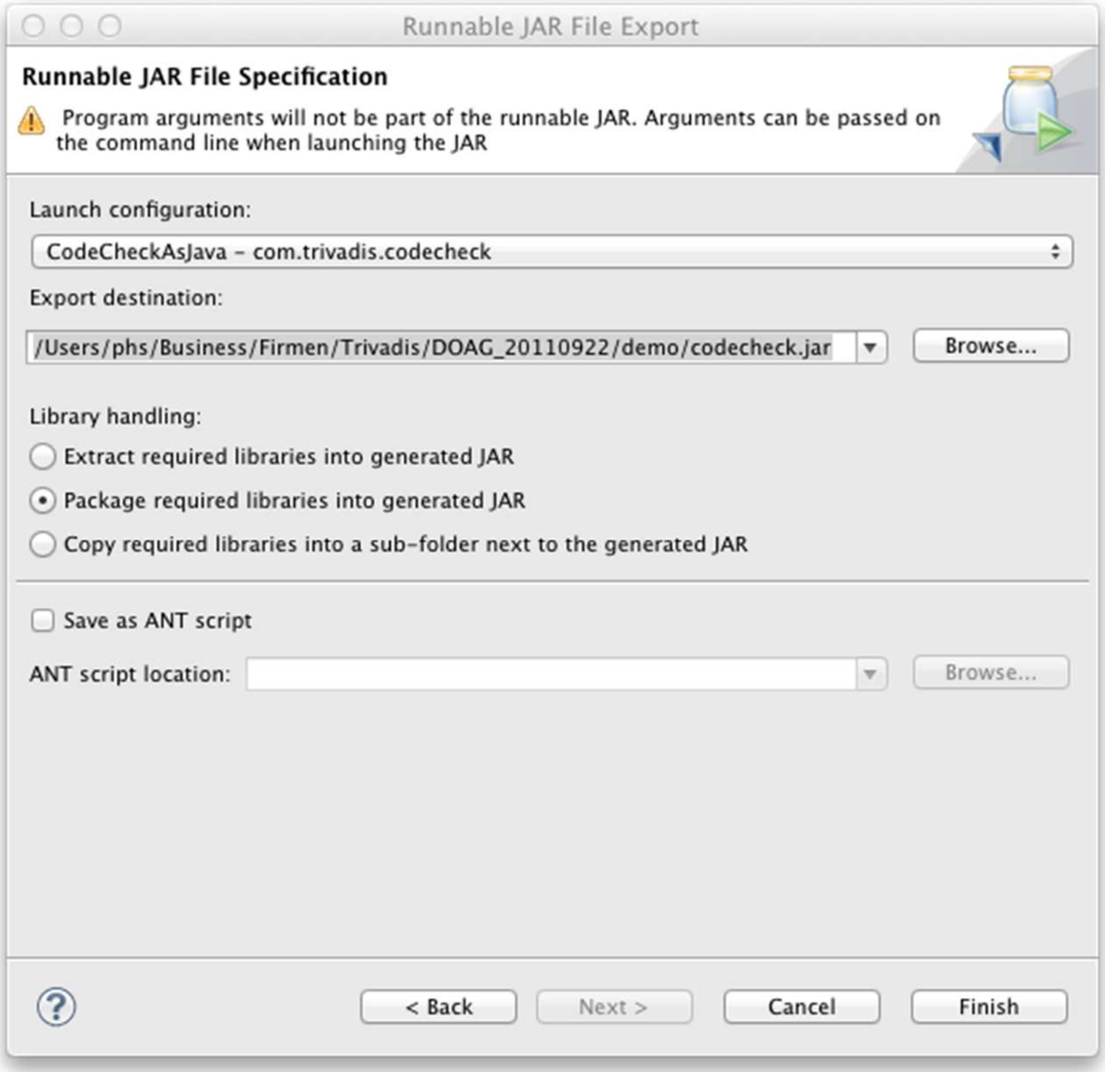
```
@Check
public void checkGuideline25(InsertIntoClause intoClause) {
    // column list empty?
    if (intoClause.getColumns().isEmpty()) {
        InsertStatement insert = EcoreUtil2.getContainerOfType(intoClause,
                InsertStatement.class);
        // model must be wrong if no insert is found
        if (insert != null) {
            Boolean ignore = false;
            SingleTableInsert singleTableInsert = insert
                    .getSingleTableInsert();
            // check for record variable in single table inserts
            if (singleTableInsert != null) {
                ValuesClause valuesClause = singleTableInsert
                        .getValuesClause();
                // ensure it's a values clause
                if (valuesClause != null) {
                    Expression expr = valuesClause.getExpression();
                    // not a column list in parenthesis?
                    if (!(expr instanceof FunctionOrParenthesisExpression)) {
                        // must be a record variable
                        ignore = true;
                    }
                }
            }
            if (!ignore) {
                warning("Guideline 25 violated: Always specify the target columns when executing an insert command.",
                        intoClause.getDmlExpressionClause(), null,
                        GUIDELINE_25, serialize(NodeModelUtils.getNode(insert)
                                .getParent()));
            }
        }
    }
}
```



```
CREATE OR REPLACE PROCEDURE p_test(i_deptno NUMBER,
                                   i_dname  VARCHAR2,
                                   i_loc    VARCHAR2) IS
   l_record dept%ROWTYPE;
BEGIN
   l_record.deptno := i_deptno;
   l_record.dname  := i_dname;
   l_record.loc    := i_loc;
   INSERT INTO dept VALUES l_record;
END;
/
```

Build Runnable JAR

DEMO


Runnable JAR File Export

Runnable JAR File Specification

⚠ Program arguments will not be part of the runnable JAR. Arguments can be passed on the command line when launching the JAR

Launch configuration: CodeCheckAsJava – com.trivadis.codecheck

Export destination: /Users/phs/Business/Firmen/Trivadis/DOAG_20110922/demo/codecheck.jar

Library handling:

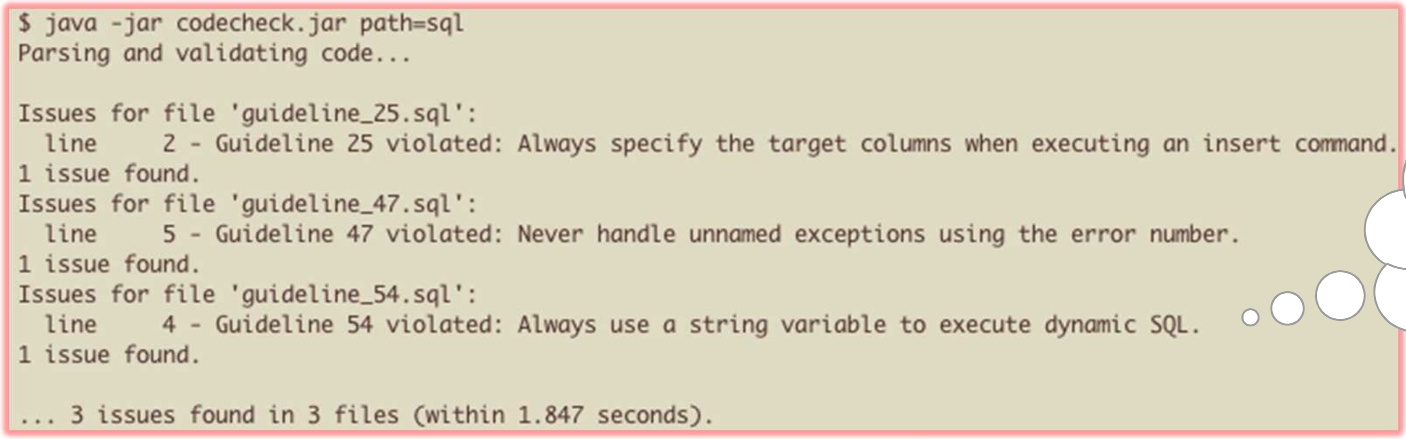
- ( ) Extract required libraries into generated JAR
- (●) Package required libraries into generated JAR
- ( ) Copy required libraries into a sub-folder next to the generated JAR

☐ Save as ANT script

ANT script location:

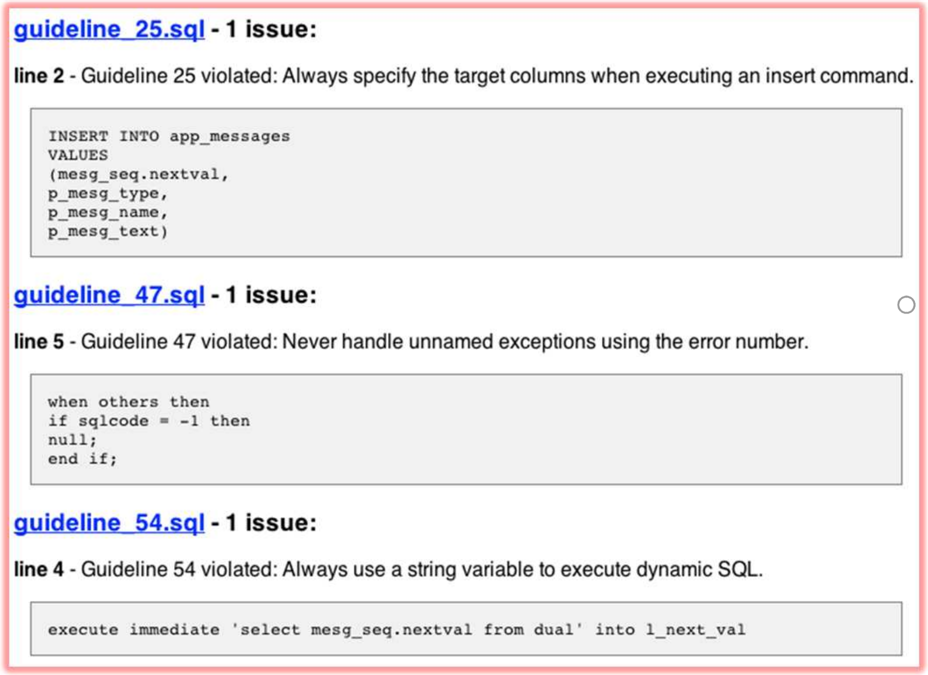
Command Line Interface

DEMO



```
$ java -jar codecheck.jar path=sql
Parsing and validating code...

Issues for file 'guideline_25.sql':
  line     2 - Guideline 25 violated: Always specify the target columns when executing an insert command.
1 issue found.
Issues for file 'guideline_47.sql':
  line     5 - Guideline 47 violated: Never handle unnamed exceptions using the error number.
1 issue found.
Issues for file 'guideline_54.sql':
  line     4 - Guideline 54 violated: Always use a string variable to execute dynamic SQL.
1 issue found.

... 3 issues found in 3 files (within 1.847 seconds).
```

Console Strategy



guideline_25.sql - 1 issue:

line 2 - Guideline 25 violated: Always specify the target columns when executing an insert command.

```
INSERT INTO app_messages
VALUES
(mesg_seq.nextval,
p_mesg_type,
p_mesg_name,
p_mesg_text)
```

guideline_47.sql - 1 issue:

line 5 - Guideline 47 violated: Never handle unnamed exceptions using the error number.

```
when others then
if sqlcode = -1 then
null;
end if;
```

guideline_54.sql - 1 issue:

line 4 - Guideline 54 violated: Always use a string variable to execute dynamic SQL.

```
execute immediate 'select mesg_seq.nextval from dual' into l_next_val
```

HTML Strategy

trivadis makes IT easier.

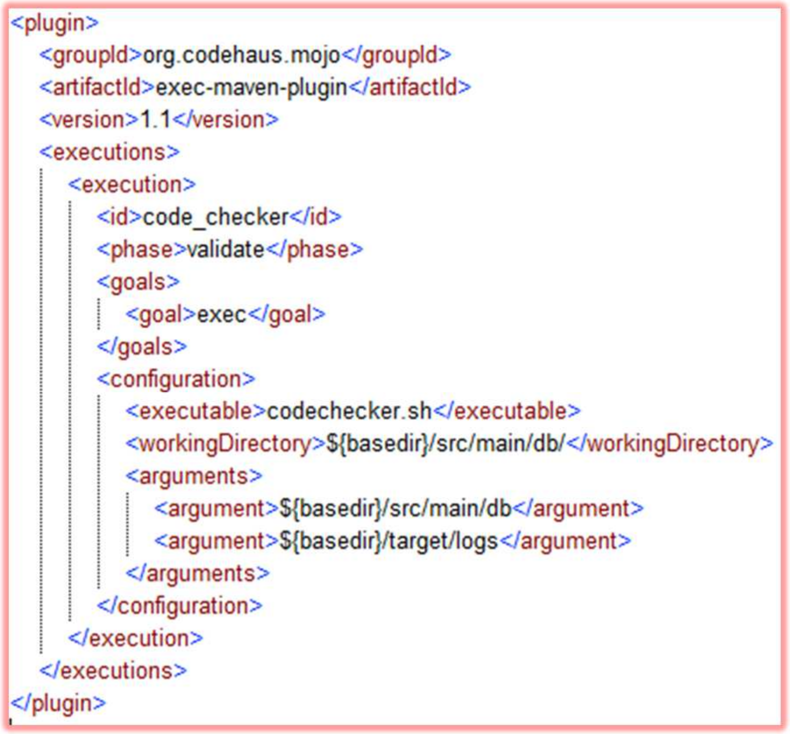
AGENDA

Initial Thoughts

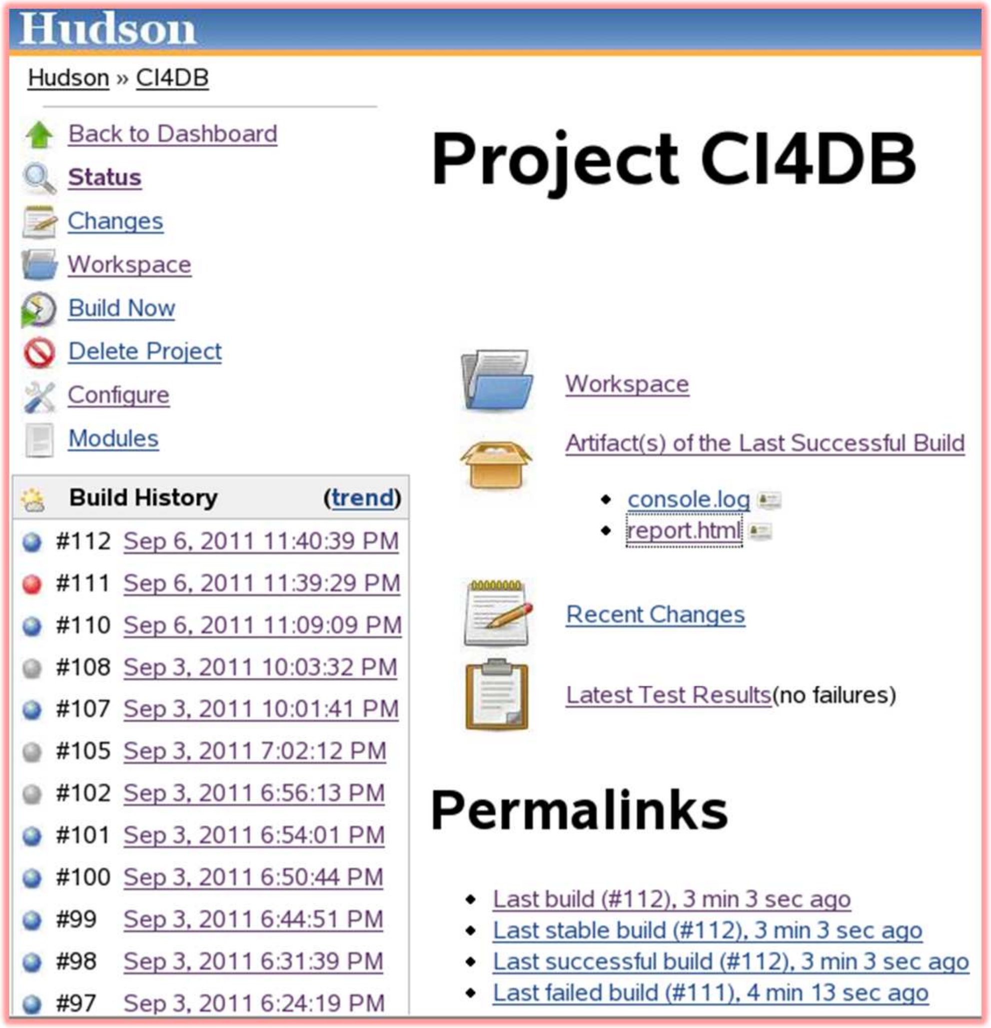
- The initial setup for a continuous integration environment supporting your database code is probably the most challenging part
- PL/SQL CodeChecker is designed to support composite output strategies
- Multiple options to the PL/SQL CodeChecker into a continuous integration environment
  - Hudson / Maven
  - Sonar

Hudson / Maven Integration

- PL/SQL CodeChecker is a Command Line Tool
- Use exec-maven-plugin



```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>exec-maven-plugin</artifactId>
  <version>1.1</version>
  <executions>
    <execution>
      <id>code_checker</id>
      <phase>validate</phase>
      <goals>
        <goal>exec</goal>
      </goals>
      <configuration>
        <executable>codechecker.sh</executable>
        <workingDirectory>${basedir}/src/main/db/</workingDirectory>
        <arguments>
          <argument>${basedir}/src/main/db</argument>
          <argument>${basedir}/target/logs</argument>
        </arguments>
      </configuration>
    </execution>
  </executions>
</plugin>
```



Hudson

Hudson » CI4DB

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure
- Modules

Build History (trend)

```
#112 Sep 6, 2011 11:40:39 PM
#111 Sep 6, 2011 11:39:29 PM
#110 Sep 6, 2011 11:09:09 PM
#108 Sep 3, 2011 10:03:32 PM
#107 Sep 3, 2011 10:01:41 PM
#105 Sep 3, 2011 7:02:12 PM
#102 Sep 3, 2011 6:56:13 PM
#101 Sep 3, 2011 6:54:01 PM
#100 Sep 3, 2011 6:50:44 PM
#99  Sep 3, 2011 6:44:51 PM
#98  Sep 3, 2011 6:31:39 PM
#97  Sep 3, 2011 6:24:19 PM
```

Project CI4DB

Workspace

Artifact(s) of the Last Successful Build

- console.log
- report.html

Recent Changes

Latest Test Results(no failures)

Permalinks

- Last build (#112), 3 min 3 sec ago
- Last stable build (#112), 3 min 3 sec ago
- Last successful build (#112), 3 min 3 sec ago
- Last failed build (#111), 4 min 13 sec ago

AGENDA

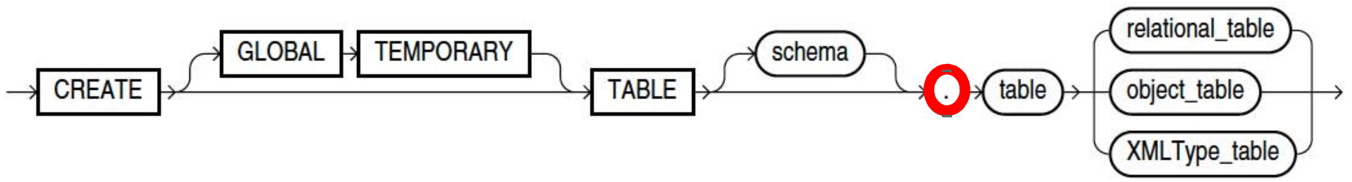
trivadis
makes IT easier.

Xtext

- One grammar, one Parser
  - The workflow GeneratePLSQL.mwe2 needs 4 minutes to complete
  - Bug 256403 - Multiple Grammar Mixin / Grammars as Library
- Maximum size of 64 KB for Java classes and methods
  - Use Xtext 2.0.1 and later to address "... is exceeding 65535 bytes ..." errors
- Output of underlying parser generator is passed 1:1 to the user
  - Fundamental knowledge of ANTLR is mandatory
  - Ability to distinguish between ANTLR and Xtext artifacts is necessary
- Convention over configuration
  - The first DSL incl. editors are created very fast using Xtext
  - Typically it's working but you easily do not know why and how
  - Usually things may be amended very elegantly and with just a few lines of code (e.g. outline, validators, formatter)
  - However, to find out what to do could take a serious time for an inexperienced fellow

Grammar

- Unquoted Identifiers may conflict with keywords of other grammars
  - "describe" is a keyword, but not a reserved word in SQL (valid for table etc.)
  - Abbreviatory notation of SQL*Plus, e.g. run command (r | ru | run)
- Undocumented, old or incorrect grammar may break the parser
  - "timestamp" clause for packages, procedures and functions
  - Use of "id" or "oid" instead of "identifier" for object views
- Documentation bugs may lead to wrong grammar



- User defined operators lead to ambiguous grammar
  - Probably solvable by refactoring the Expression and Condition parser rules
  - The workaround is, to simply add the customer's operators when needed
- Reduced grammar in the area of less interesting statements
  - AlterTable: 'alter' 'table' text=GenericText SqlCmdEnd ;

Some minor SQL*Plus Limitations

- The block terminator character '.' is not supported (nor configurable)
- The command separator character ';' is not supported (nor configurable)
- The SQLTerminator is not configurable, the default ';' is supported
- The line continuation character '-' does not support tailing whitespaces
- REMARK and PROMPT must not contain unterminated single/double quotes, single line or multi line comments (these commands cannot be defined as terminals because of conflicts with other parser rules – mainly identifiers)

AGENDA

Conclusion

PL/SQL & SQL

Tooling

- The grammar to parse SQL*Plus files is huge
  - a solution to reduce/separate the grammars is necessary to make the development process feasible
  - since Xtext 2.0.1 the size restrictions ceased to apply
- Xtext is a complete DSL framework
  - More than just a parser generator
  - Separation of parser and validators
  - Promising for further applications like code fixing, presenting graphical models, calculating complexity, etc.
- Even if a significant subset of the SQL*Plus, SQL, PL/SQL grammar needs to be maintained continuously, Xtext is a good choice to implement the future PL/SQL CodeChecker and Dependency Analysis requirements
- The PL/SQL CodeChecker will be part of the Trivadis Continuous Integration environment

THANK YOU.

Trivadis AG

Philipp Salvisberg

Europastrasse 5
8152 Glattbrugg (Zürich)

Tel. +41-44-808 70 20
Fax +41-44-808 70 21

philipp.salvisberg@trivadis.com
www.trivadis.com

BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

trivadis
makes IT easier.