# Testing with utPLSQL

## That Doesn't Work Anyway! Nonsense!

Philipp

@phsalvisberg https://www.salvis.com/blog

18.11.2019

# Philipp

- Database centric development
- Model Driven Software Development
- Author of free SQL Developer Extensions PL/SQL Unwrapper, PL/SQL Cop, utPLSQL, plscope-utils, oddgen and Bitemp Remodeler

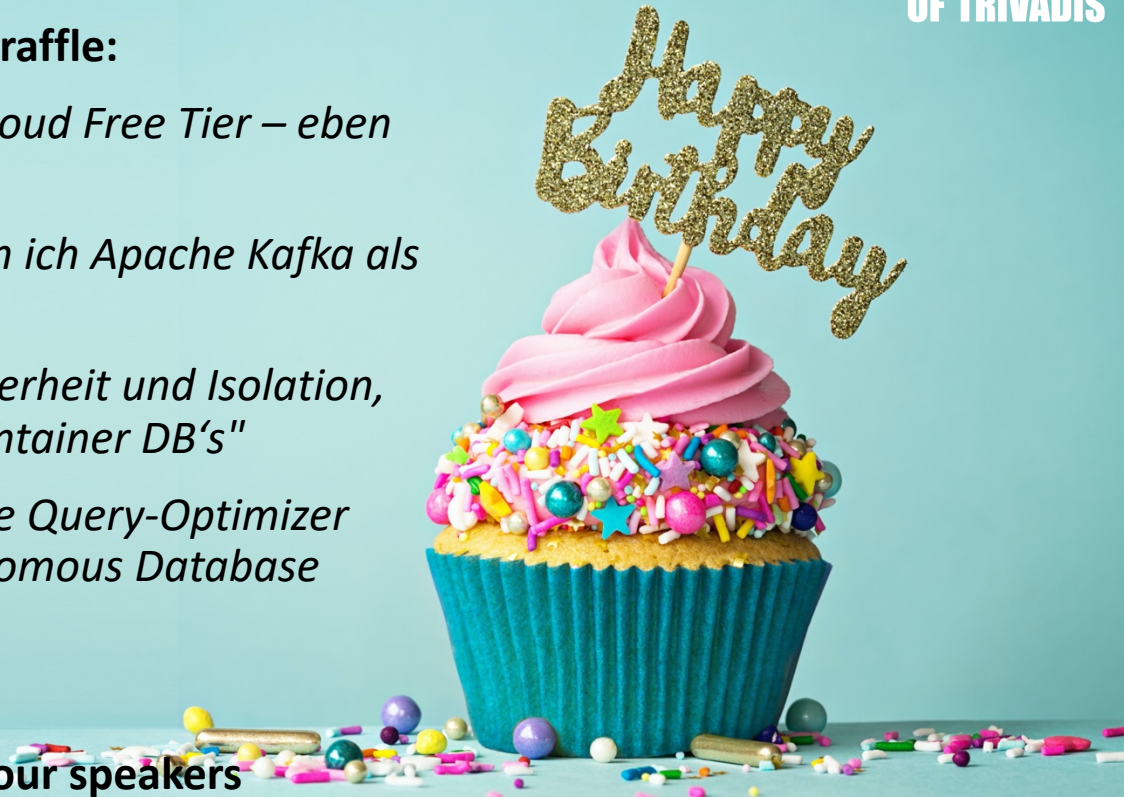
@phsalvisberg

https://www.salvis.com/blog

# Visit Us at Our Booth on Level 3

25 YEARS OF TRIVADIS

- **Trivadis barista (good coffee from morning to night)**
- **Birthday cake (daily from 14:00)**
- **Speed-Sessions at the booth with raffle:**
  - **Di 14:45:** Martin Berger *"Oracle Cloud Free Tier – eben mal kurz ausprobiert…"*
  - **Mi 10:45:** Guido Schmutz *"Warum ich Apache Kafka als IT-Spezialist kennen sollte!"*
  - **Mi 14:45:** Stefan Oehrli *"PDB Sicherheit und Isolation, Herausforderungen von Oracle Container DB‘s"*
  - **Do 10:45:** Chris Antognini *"Welche Query-Optimizer Funktionalitäten sind nur in Autonomous Database verfügbar? “*
- **Participation in our DOAG raffle**
- **Networking and discussions with our speakers**



# Agenda

trivadis

1. Introduction
2. The Problem with Database Development
3. Solution Approaches
4. Core Messages

trivadis

# Introducation

# Test Automation

trivadis

"The use of software
separate from the software being tested
to control the execution of tests
and the comparison of
actual outcomes with predicted outcomes."

Source: https://en.wikipedia.org/wiki/Test_automation

# Reasons to Automate Tests

trivadis

1. Reduces Time

2. Reduces Effort

3. Increases Productivity

4. Increases Predictability

5. Increases Reliability


Source: https://blog.codeship.com/5-reasons-for-automated-testing/

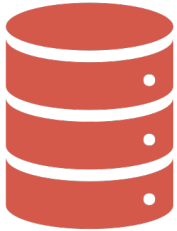
# You Need Tests to Automate Them

trivadis

# Testing Scope in Database Development

trivadis

Every component of an application
that is deployed in the database.



For example:
Types, packages, procedures, functions,
views, triggers, constraints.

utPLSQL

trivadis

## Core Testing Framework

- Schema in the database
- No repository
- Annotation based tests

ORACLE DATABASE

## Development

- SQL Developer Extension
- Various Reporters


## Test Automation

- Command Line Client
- Maven Plugin Maven
- Various Reporters

JUnit TC TFS

sonarqube COVERALLS





# Test Declaration

trivadis

```
CREATE OR REPLACE PACKAGE test_package_name AS
   --%suite

   --%test
   PROCEDURE procedure_name;
END;
```

--%suite(<description>)
--%suitepath(<path>)
--%tags(<tag>[,...]
--%displayame(<description>)
--%beforeall([...])
--%afterall([...])
--%beforeeach([...])
--%aftereach([...])
--%rollback(manual)
--%disabled
--%context
--%endcontext

--%displayname(<description>)
--%test(<description>)
--%tags(<tag>[,...]
--%throws(<exception>[,...])
--%beforeall
--%afterall
--%beforeeach
--%aftereach
--%beforetest([...])
--%aftertest([...])
--%rollback(manual)
--%disabled

# Test Implementation

trivadis

```
CREATE OR REPLACE PACKAGE BODY test_package_name AS
   PROCEDURE procedure_name IS
      l_actual   SYS_REFCURSOR;
      l_expected SYS_REFCURSOR;
   BEGIN
      open l_actual for select * from dept where deptno != 30;
      open l_expected for select * from dept;
      ut.expect(l_actual).to_equal(l_expected).join_by('DEPTNO');
   END procedure_name;
END;
```

Matcher:

be_between, be_empty, be_false, be_greater_than, be_greater_or_equal, be_less_or_equal, be_less_than, be_like, be_not_null, be_null, be_true, contain, **equal**, have_count, match

Extended options for refcursor, object type, JSON, nested table and varray:

- include(<items>)
- exclude(<items>)
- unordered
- **join_by**(<items>)

# Test Run

trivadis

```
SET SERVEROUTPUT ON SIZE UNLIMITED
EXEC ut.run('test_package_name')
```

List of ...
- package[.procedure]
- schema[.package[.procedure]]
- schema:suitepath[.context][.procedure]

List of ...
- tag (include)
- -tag (exclude)

```
test_package_name
  procedure_name [.03 sec] (FAILED - 1)

Failures:

  1) procedure_name
       Actual: refcursor [ count = 3 ] was expected to equal: refcursor [ count = 4 ]
       Diff:
       Rows: [ 1 differences ]
         PK <DEPTNO>30</DEPTNO> - Missing:  <DEPTNO>30</DEPTNO><DNAME>SALES</DNAME><LOC>CHICAGO</LOC>
       at "SCOTT.TEST_PACKAGE_NAME.PROCEDURE_NAME", line 8 ut.expect(l_actual).to_equal(l_expected)
                                                                                  .join_by('DEPTNO');

Finished in .034792 seconds
1 tests, 1 failed, 0 errored, 0 disabled, 0 warning(s)
```

trivadis

# The Problem with Database Development

trivadis

# Stateless Functionality

# BETWNSTR

trivadis

```
CREATE OR REPLACE PACKAGE util IS
   /**
    * A variation of substring that will return the portion of a
    * string between specified start and end locations.
    */
   FUNCTION betwnstr(
       in_string           IN VARCHAR2,
       in_start_pos        IN INTEGER,
       in_end_pos          IN INTEGER
   ) RETURN VARCHAR2 DETERMINISTIC;
END util;
```


trivadis

# Stateful Functionality

# Stateful Session

trivadis

```
CREATE OR REPLACE VIEW v IS
SELECT …
  FROM …
 WHERE tenant_id = sys_context ('APP_CONTEXT', 'TENANT_ID');
```

# Stateful Package

trivadis

```
CREATE OR REPLACE PACKAGE BODY pkg IS
   g_a BOOLEAN DEFAULT FALSE;
   …
   PROCEDURE do (in_x VARCHAR2) IS
   BEGIN
      IF g_a THEN
         …
      END IF;
   END do;
BEGIN
   …
END pkg;
```

# Code Under Test Requires DML

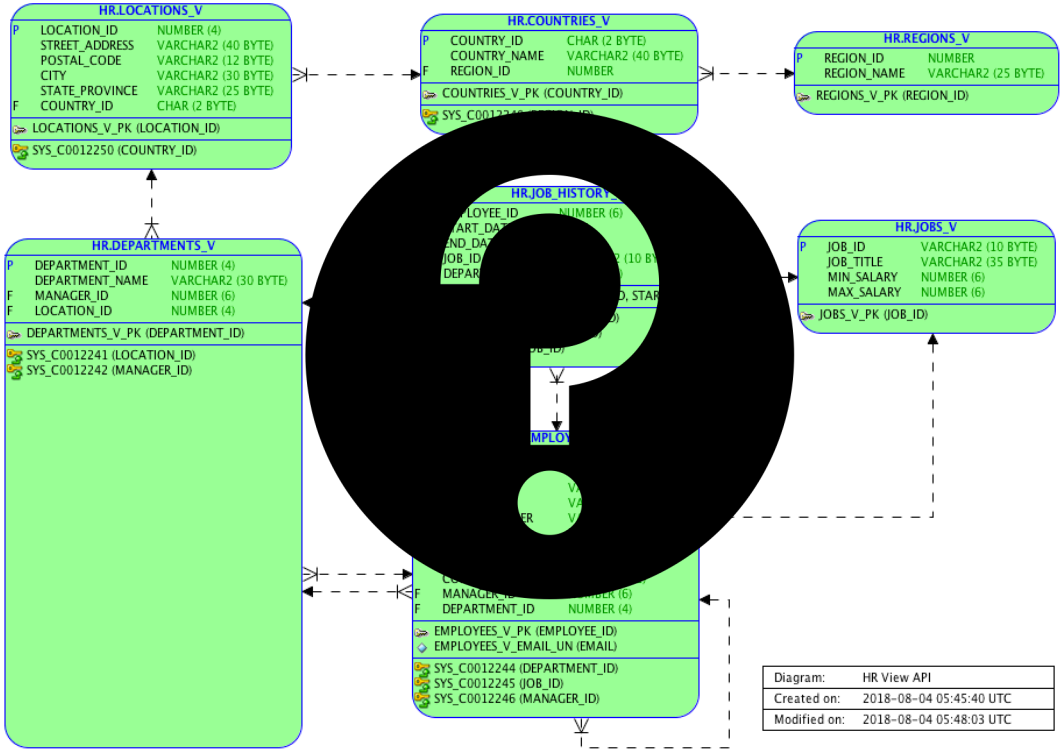
trivadis

- Requirement
  - Only employees working in the SALES department are eligible for a commission

```
ALTER TABLE emp
   ADD CONSTRAINT emp_comm_ck
      CHECK (comm IS NULL or deptno = 30);
```

# DML

- SELECT
- INSERT
- UPDATE
- DELETE
- MERGE
- …

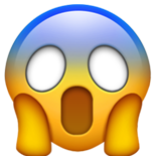


# Transaction Control Statements

trivadis

**Implicit (via DDL)**

- ALTER (except SESSION/SYSTEM)
- DROP
- TRUNCATE
- FLASHBACK
- GRANT
- REVOKE
- …



**Explicit**

- COMMIT
- ROLLBACK
- …

Source: https://www.youtube.com/watch?v=EXGE5btmLVo&t=43

trivadis

# Solution Approaches

trivadis

# Extended Scope of a Unit Test in the Database

# Database Unit Test

trivadis

- Written and maintained by the developer
- Covers a part of the system
- Fast
- **Sociable** (rely on other units to fulfill its behavior)
  - Tables, Constraints, Triggers, Views, Functions, Packages, Types
- Deterministic

Based on: https://martinfowler.com/bliki/UnitTest.html

# Run Tests With Same Package States

trivadis

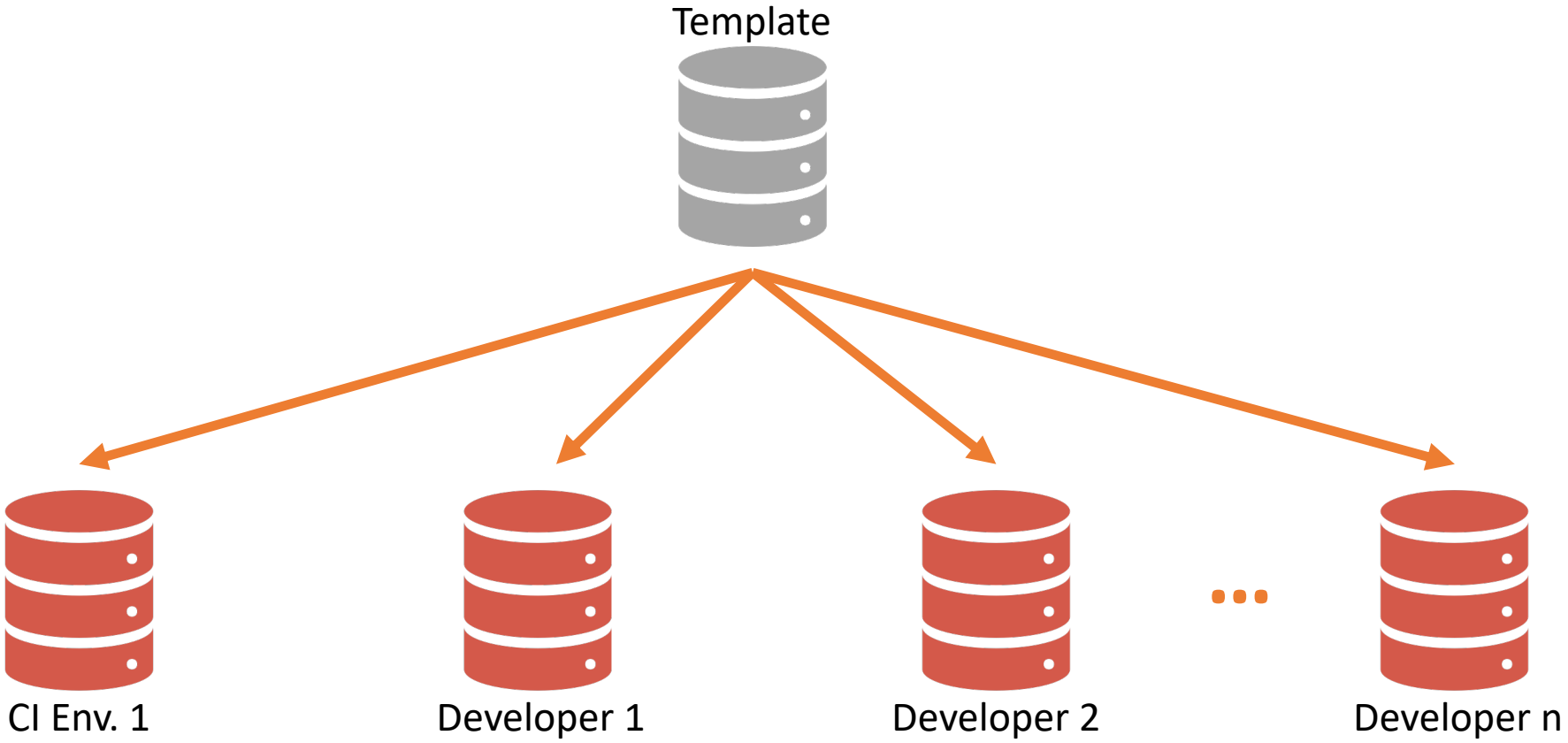
- If existing code does not allow to reset package states
  - Are public reset procedures an issue?
  - Introduce reset procedures accessible by test code only
- Group tests using
  - Suites
  - Tags
- Run tests in dedicated "fresh" connections

trivadis

# Personal Databases

# Database Environments

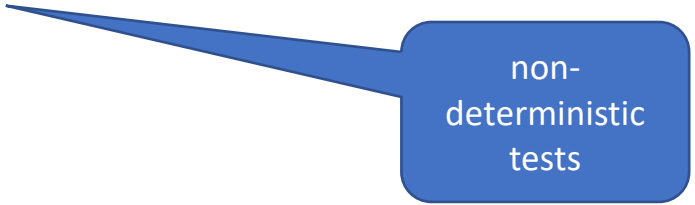
trivadis



# Template Based on Production Data

trivadis

- Full Scope
- Reduced Scope
- Anonymized / Redacted (client identifying data)
- Potential Issues
  - Size (can be amended with copy on write technologies)
  - Changes (might lead to test failures)



# Template Based on Model Database

trivadis

- Manually maintained data of a potential production database
- Reduced scope compared to typical production databases
- Known, stable master data (no client identifying data)
- Known, stable transactional data

# Template Based on Empty Database

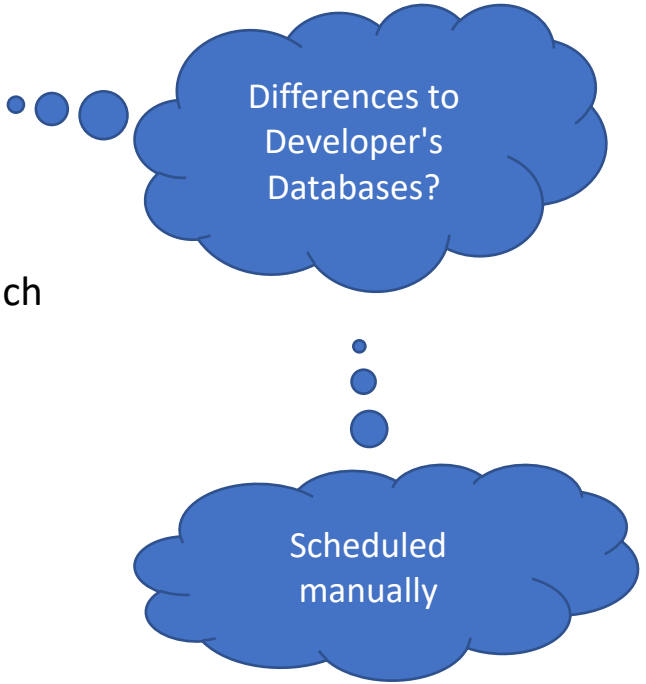
trivadis

- Full install of database structures
- Full installation of database code
- Optional
  - Full installation of referential data
    - Code tables
    - Subset of master data that could be seen as part of the application (stable)
  - Partial installation of vital master data
    - Configuration of a tenant
    - Subset of master data that could be seen as application configuration

# Continuous Integration Environments

trivadis

- Schedule Build based on the following events
  - New database template
  - Commit on CI relevant branch (e.g. develop)
- Build Job
  - Create database (shortcuts via snapshots possible)
  - Install utPLSQL
  - Install database components based on the CI relevant branch
  - Run all tests using the utPLSQL-cli



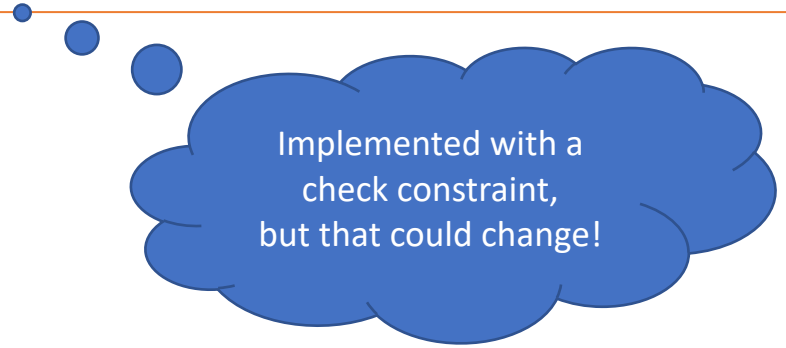
# Set Up

# For Test Specification

trivadis

```
CREATE OR REPLACE PACKAGE test_emp IS
   --%suite

   --%test(commission is not allowed for non-sales employees)
   --%throws(-02290)
   PROCEDURE comm_for_non_sales_emp_failure;
END test_emp;
```



# Using Existing Data

trivadis

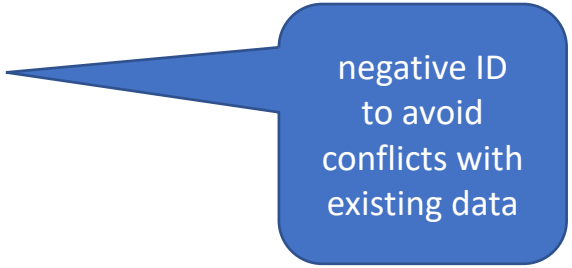
```
CREATE OR REPLACE PACKAGE BODY test_emp IS
   PROCEDURE comm_for_non_sales_emp_failure IS
      l_empno emp.empno%type;
   BEGIN
      SELECT empno
        INTO l_empno
        FROM emp
       WHERE deptno != 30 AND ROWNUM = 1;
      UPDATE emp
         SET comm = 1
       WHERE empno = l_empno;
   END comm_for_non_sales_emp_failure;
END test_emp;
```

Test will fail, when no data is found

# From Scratch Using Plain SQL

trivadis

```
CREATE OR REPLACE PACKAGE BODY test_emp IS
   PROCEDURE comm_for_non_sales_emp_failure IS
   BEGIN
      INSERT INTO dept (deptno)
           VALUES (-10);
      INSERT INTO emp (empno, deptno)
           VALUES (-1, -10);
      UPDATE emp
         SET comm = 1
       WHERE empno = -1;
   END comm_for_non_sales_emp_failure;
END test_emp;
```



# From Scratch Using Data Builder

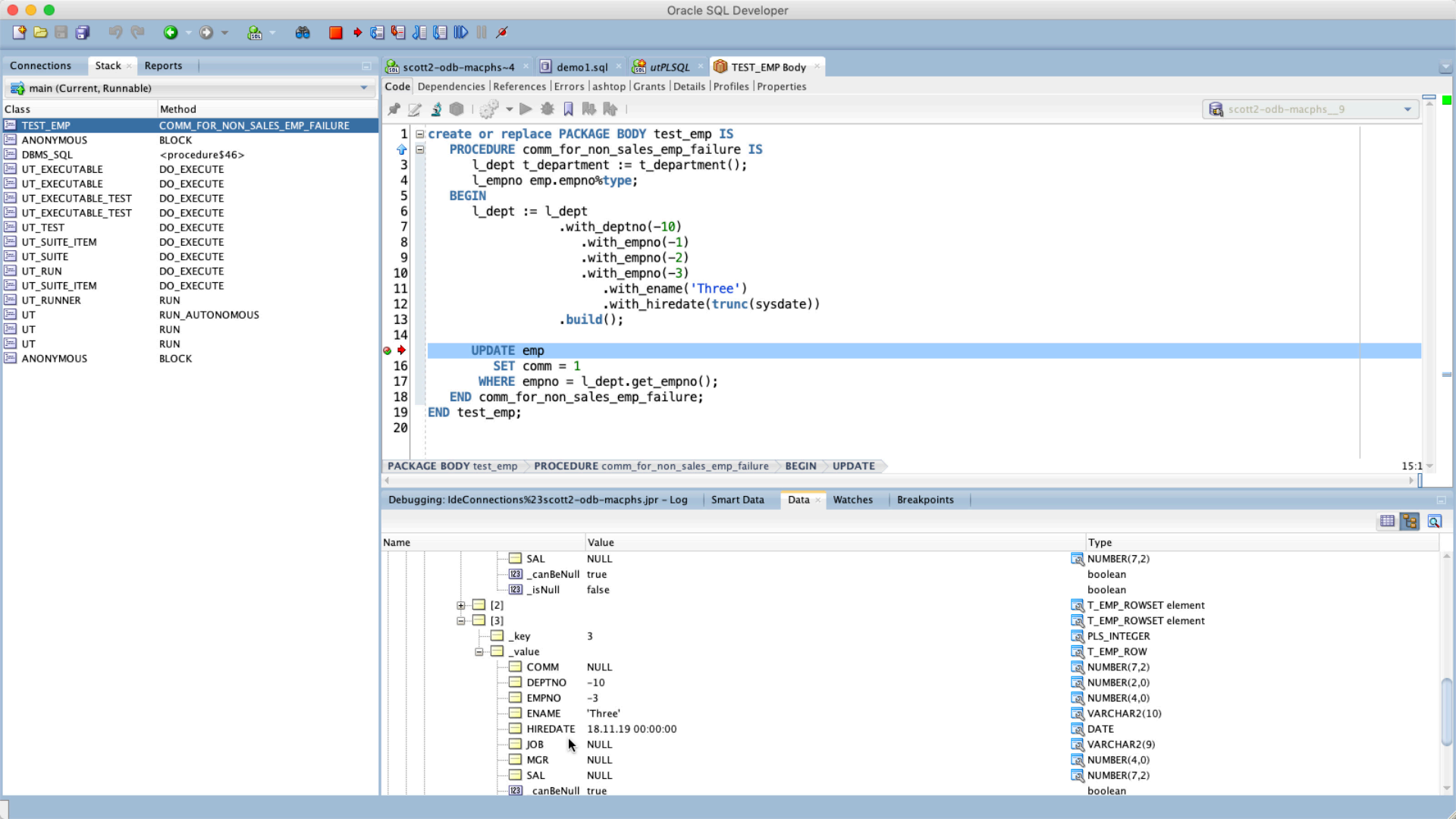
trivadis

```
CREATE OR REPLACE PACKAGE BODY test_emp IS
   PROCEDURE comm_for_non_sales_emp_failure IS
      l_dept t_department := t_department();
   BEGIN
      l_dept := l_dept
                  .with_deptno(-10)
                  .with_empno(-1)
                  .build();
      UPDATE emp
         SET comm = 1
       WHERE empno = l_dept.get_empno();
   END comm_for_non_sales_emp_failure;
END test_emp;
```

Based on: https://github.com/mathewbutler/data-builder-simple

trivadis

# Demo


Oracle SQL Developer
Connections
Stack
Reports
main (Current, Runnable)
Class
Method
TEST_EMP
COMM_FOR_NON_SALES_EMP_FAILURE
ANONYMOUS
BLOCK
DBMS_SQL
<procedure$46>
UT_EXECUTABLE
DO_EXECUTE
UT_EXECUTABLE
DO_EXECUTE
UT_EXECUTABLE_TEST
DO_EXECUTE
UT_EXECUTABLE_TEST
DO_EXECUTE
UT_TEST
DO_EXECUTE
UT_SUITE_ITEM
DO_EXECUTE
UT_SUITE
DO_EXECUTE
UT_RUN
DO_EXECUTE
UT_SUITE_ITEM
DO_EXECUTE
UT_RUNNER
RUN
UT
RUN_AUTONOMOUS
UT
RUN
UT
RUN
ANONYMOUS
BLOCK
scott2-odb-macphs~4
demo1.sql
utPLSQL
TEST_EMP Body
Code
Dependencies
References
Errors
ashtop
Grants
Details
Profiles
Properties
scott2-odb-macphs__9
1 create or replace PACKAGE BODY test_emp IS
2 PROCEDURE comm_for_non_sales_emp_failure IS
3 l_dept t_department := t_department();
4 l_empno emp.empno%type;
5 BEGIN
6 l_dept := l_dept
7 .with_deptno(-10)
8 .with_empno(-1)
9 .with_empno(-2)
10 .with_empno(-3)
11 .with_ename('Three')
12 .with_hiredate(trunc(sysdate))
13 .build();
14
15 UPDATE emp
16 SET comm = 1
17 WHERE empno = l_dept.get_empno();
18 END comm_for_non_sales_emp_failure;
19 END test_emp;
20
PACKAGE BODY test_emp
PROCEDURE comm_for_non_sales_emp_failure
BEGIN
UPDATE
15:1
Debugging: IdeConnections%23scott2-odb-macphs.jpr - Log
Smart Data
Data
Watches
Breakpoints
Name
Value
Type
SAL NULL NUMBER(7,2)
_canBeNull true boolean
_isNull false boolean
[2] T_EMP_ROWSET element
[3] T_EMP_ROWSET element
_key 3 PLS_INTEGER
_value T_EMP_ROW
COMM NULL NUMBER(7,2)
DEPTNO -10 NUMBER(2,0)
EMPNO -3 NUMBER(4,0)
ENAME 'Three' VARCHAR2(10)
HIREDATE 18.11.19 00:00:00 DATE
JOB NULL VARCHAR2(9)
MGR NULL NUMBER(4,0)
SAL NULL NUMBER(7,2)
_canBeNull true boolean

trivadis

# Tear Down

# --rollback(auto)

trivadis

- utPLSQL default behaviour
- Changes are isolated by savepoints
- Rollback is performed automatically at the end, in any case

# --rollback(manual)

trivadis

- utPLSQL can throw the following warning:

```
Unable to perform automatic rollback after test. An implicit or explicit
commit/rollback occurred in procedures:
  …
Use the "--%rollback(manual)" annotation or remove commit/rollback/ddl
statements that are causing the issue.
```

- The annotation suppresses the warning
- You have to manually undo all changes in the database

# Outside of utPLSQL

trivadis

- Recreate the Personal Database based on the template
- Shortcuts possible via snapshots
  - Flashback Database
  - Virtual Machine
  - …

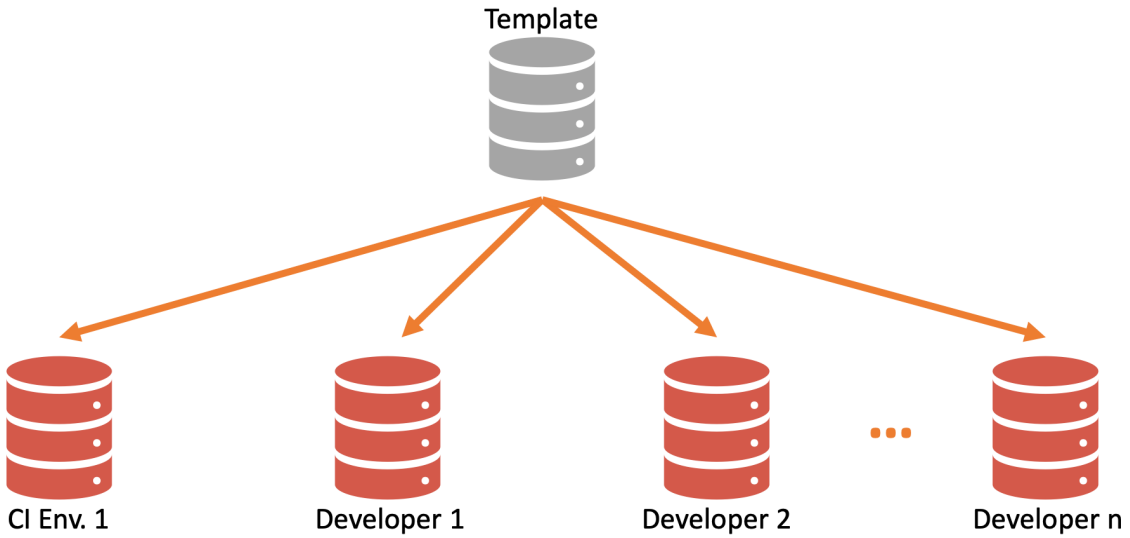
trivadis

# Core Messages

# Personal Databases

trivadis

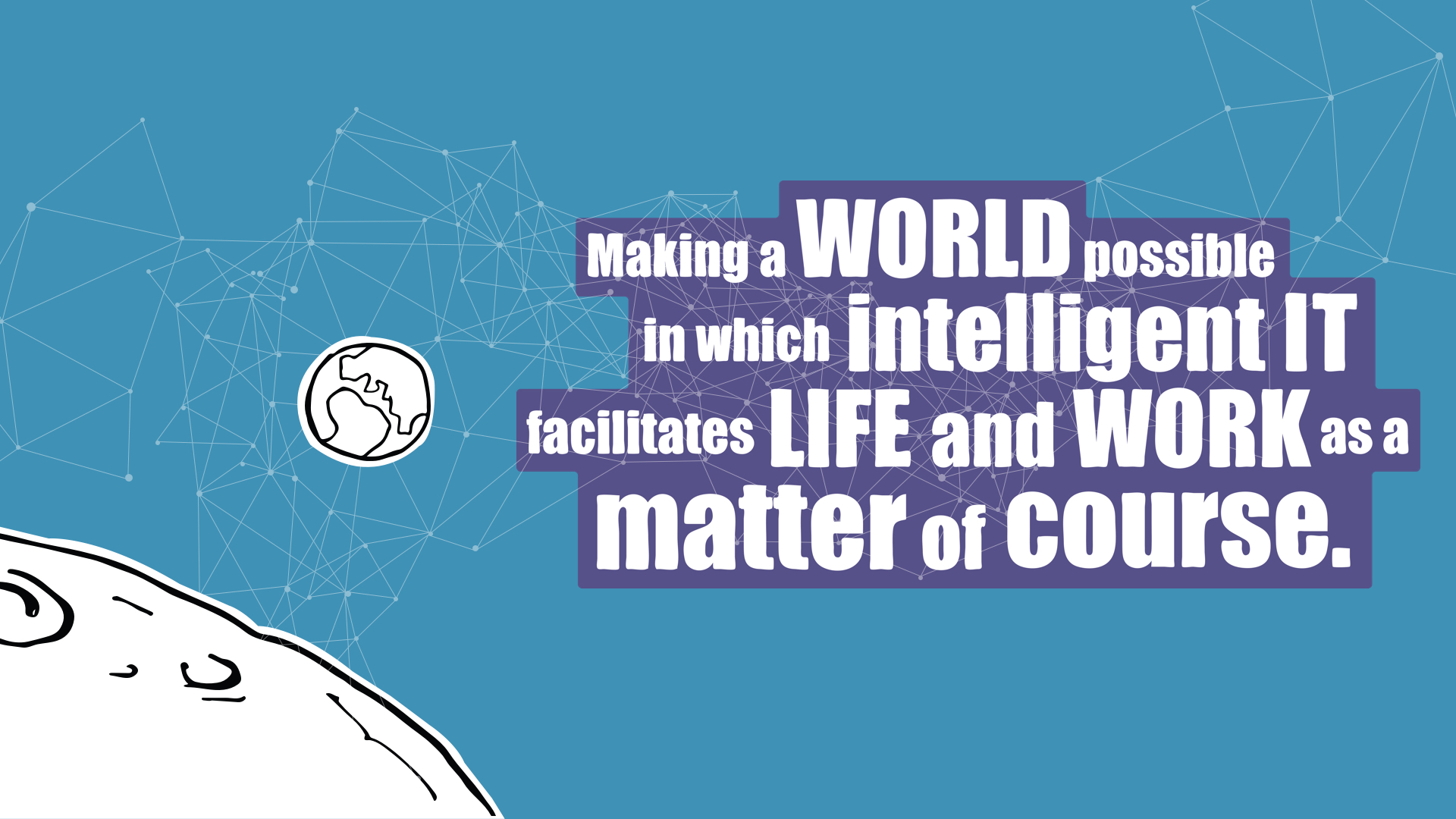
- Self Service for Each Developer
- Fully Automated for Continuous Integration
- With utPLSQL



# Database Unit Tests

trivadis

- Written and Maintained by Developers
- Deterministic
- Use Tooling for Data Set Up
- No Commits = No Manual Data Tear Downs


Making a WORLD possible
in which intelligent IT
facilitates LIFE and WORK as a
matter of course.