

**NORTAL**

# **Micro-migration: transferring sets of test data between database schemas**

Jüri Harju

2013

# Outline

- Problem specification
- Solution specification
- Installing and usage
- Usage scenarios
- Usage results
- Future plans

# Problem specification I

## Developer schemas

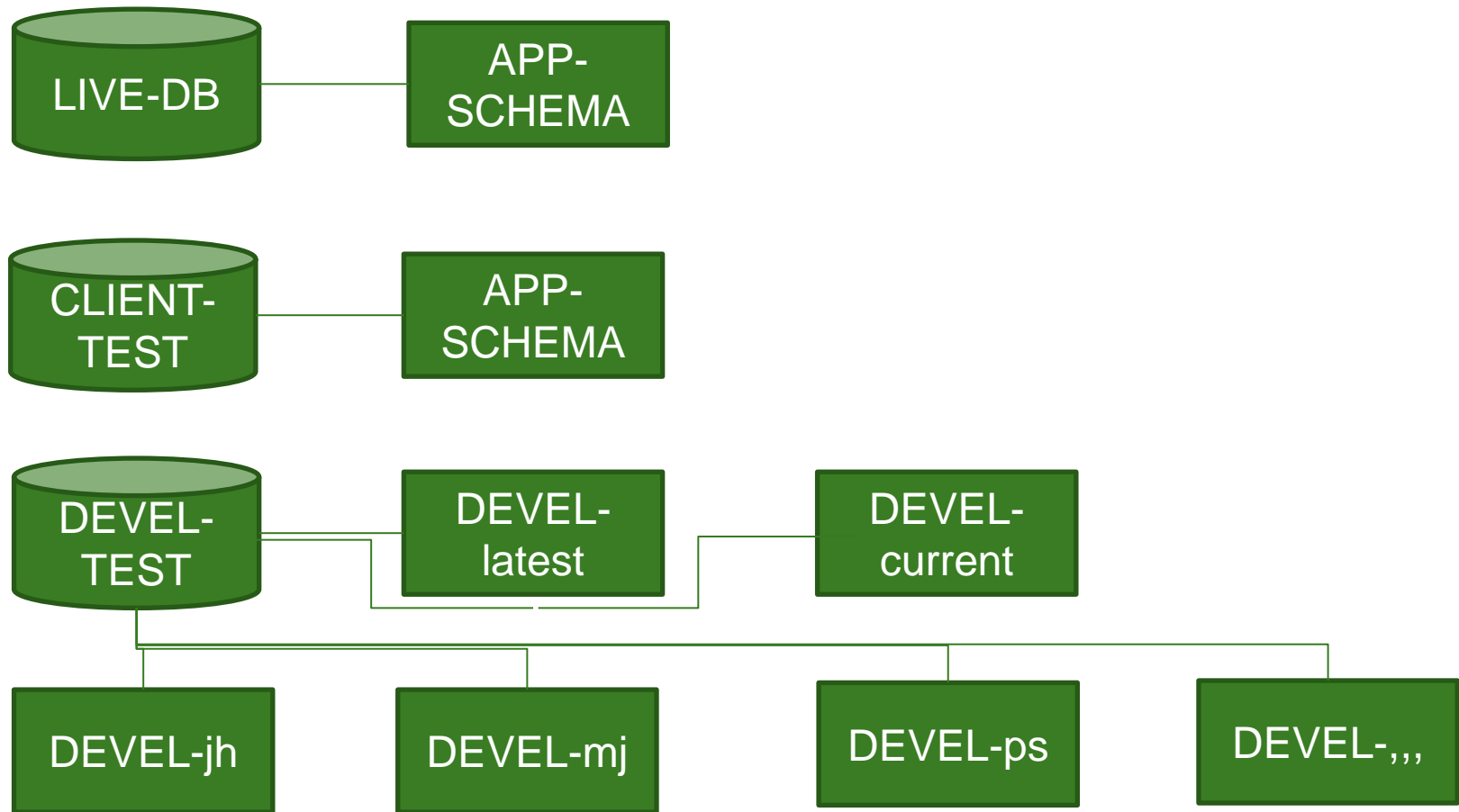
- Each developer has his own schema
- Minimal amount of data
- Data has poor quality
- “Artificial” data with inadequate cases

# Problem specification II

## Testers schemas

- Large amount of data
- Realistic data cases
- Used with more stable application versions

# Bases and schemas



# Problem specification III

Complicated test cases in tester schemas

- Re-creation takes additional time

Very specific test cases

- Re-creation of test case too difficult

# Problem specification IV

## Full-scale data migrating

- LIVE-DB -> data obfuscation -> TEST-DB
- TEST-DB -> DEVEL-TEST

## Selected data migrating

- SQL Insert generation
- Writing dedicated migrating scripts

# Solution specification I

Create a PL/SQL package to transfer data based on the record from a one table

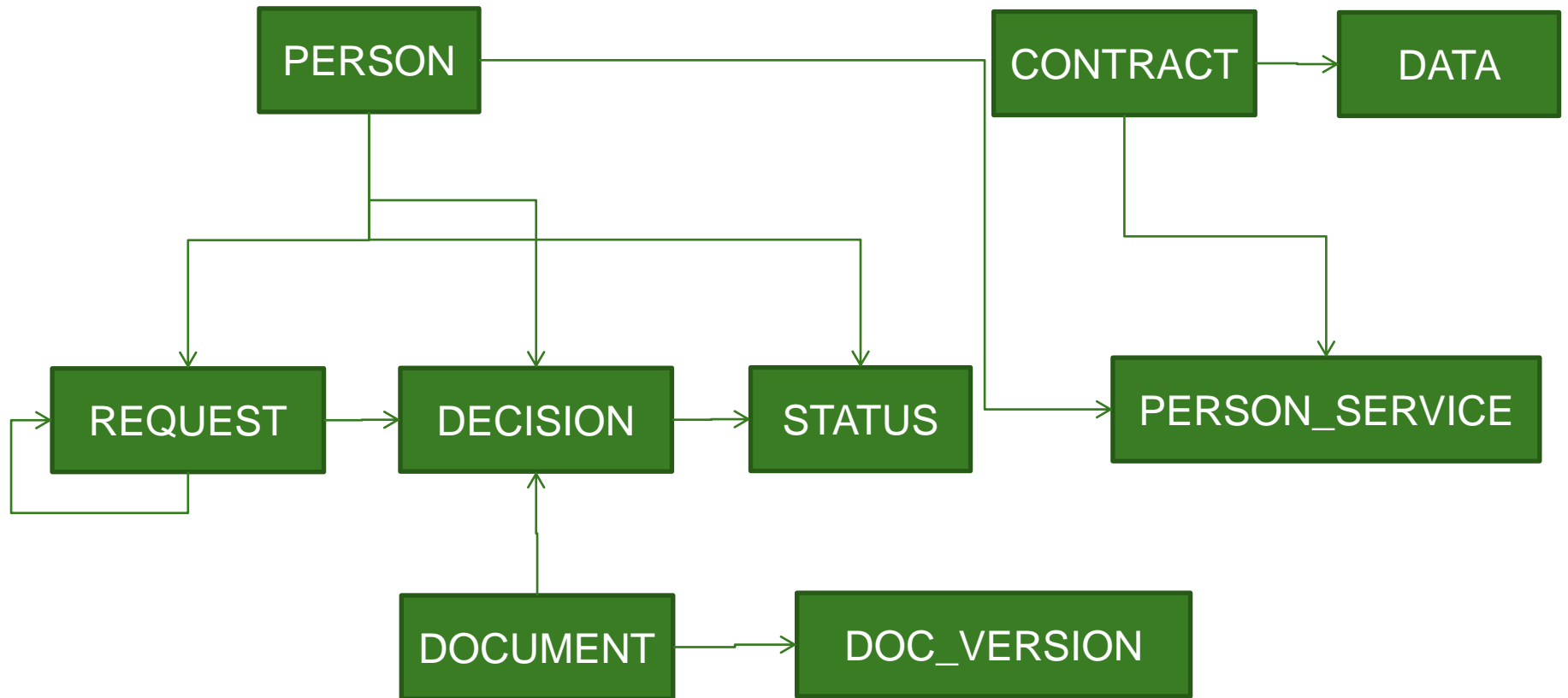


# Solution specification II

PL/SQL package: main steps

- Find all FK relations based on given record
- Collect related ID's of related records
- Insert data into destination schema

# Data model example



# Solution specification III

PL/SQL package: resolving relations

- Parent relations
- Child relations
- Relation cycles
- De-normalizations

# Installing

- DB user with DBA privileges
- Micro-migration tables script
- Micro-migration package
- Execution-script adaptation

# Usage: simple

Prepared execution script

- Schema 1 – source schema
- Schema 2 – target schema
- Table – main table for given data set
- PK values for main table

# Usage: additional features I

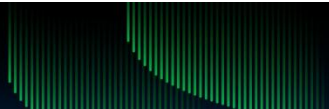
- Child relations exclusion
- Parent relations exclusion
- Other child tables of parent tables inclusion
- Migrating with new ID's
- Same schema data duplication
- Recursive delete

# Usage: additional possibilities II

FK unrelated data migration

- Domain values etc.

Synchronization of sequences



# Usage scenarios I

## Component development

- Selecting realistic cases from testers schema
- More effective developer basic testing
- More effective debugging while running application on developer schema



# Usage scenarios II

## Easier bug reporting

- Only ID from tester schema and actual problem
- Detailed description for problematic data is not needed

# Usage in real project

- 12 developers in project
- 4 testers in project
- >300 tables in data model
- “PERSON” table with >350000 records

# Usage results I

PERSON table with one given ID (simple case)

- 27 tables
- 57 records
- ~1 min

# Usage results II

PERSON with one given ID (more complicated cases)

- 73 tables
- 500-600 records
- ~2-3 min

# Usage results III

PERSON with one ID + CONTRACT table  
child relations

- 73 tables
- ~2500 records
- ~10 min

# Usage results IV

Top 3 usage in 6 month pilot period  
(in this period related project had 4 deliveries)

- 91 cases and 50802 records
- 56 cases and 11114 records
- 39 cases and 8549 records

# Future plans

- Resolving relations based on UQ keys
- Migration processes binding
- DB-link usage possibility
- “Off-line” migration between different databases

# Questions





**NORTAL**