TEST DATA MANAGEMENT
CONCEPTS & METHODS
Key requirements

- Automation: Define once, execute regularly
- Create different environments that do not interfere
- Easily add more environments
- Goal: Better QA with reasonable costs
Limitations of Tools for Test Data Management

• Test cases
• Application logic
• What needs to be tested?
• How often a copy is required?
Who needs what?
Who needs what?

- PRODUCTION
- ACCEPT
- REGRESSION
- INTEGRATION
- COMPONENT
- FUNCTION
- DEVELOPMENT

Amount of Data
Who needs what?

- Amount of Data
- Degree of Modification
Row Level Processing
Unit Tests

Table Copying
Component / Integration Tests

Database / Subsystem Cloning
Release / System Tests / Preproduction
Subsystem/Database Cloning
• Create a pre-production environment
• Data source for further copy operations
• Strict separation between prod and test
• Acceptance tests
• RDBMS upgrades
Preproduction Clone for DB2, Oracle, etc.
How does a clone become consistent?
How does a clone become consistent?

- FlashCopy1
- SnapShot
- EMC TimeFinder
- IBM PPRC
- HDS ShadowImage
- Softek TDMF
- Fujitsu Equivalent Copy

Time
Transactions
Prepare & (Re)Start

Rebuild current status

Time
Cloning

- Well defined set of scripts / jobs that read production
- Executed periodically by scheduler
- Strict decoupling of production from preproduction
- Rework after cloning is required
Characteristic of Cloning

• Need access to physical hard drives
• No need to stop or start databases
• Hardware assisted copy tools are very beneficial
• DBMS tools can also be used alternatively
UBS HAINER

TEST DATA MANAGEMENT

SUBSYSTEM CLONING | TABLE COPYING | ROW LEVEL PROCESSING

Database / Subsystem Cloning
Release / System Tests / Preproduction

Table Copying
Component / Integration Tests

Row Level Processing
Unit Tests
Table Level Copying
Feeding the other Environments

Production → Preproduction

Preproduction → Test Environments

Preproduction → Dev
Wish list for file system level copies:

- Specify one schema,
- copy 1000 tables
Characteristic Table Copying

- Replace/copy/rename selected objects
- Need authorization to move TS / to use utilities
- Hardware assisted copy tools not applicable
- But optimized for copy of mass data
- Masking/Anonymization feasible
**Row Level Processing**

- Unit Tests

**Table Copying**

- Component / Integration Tests

**Database / Subsystem Cloning**

- Release / System Tests / Preproduction
Row Level Processing
Production (DB2, Oracle, etc.) → Pre-Production (DB2, Oracle, etc.) → Test Environments

- Clone
- Copies
- Inserts

Function Testing

Unit Testing
Provide data for the next application version

Version 1.0:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Constraints</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>INTEGER</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>FIRSTNAME</td>
<td>VARCHAR(256)</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>LASTNAME</td>
<td>VARCHAR(256)</td>
<td>NOT NULL</td>
<td></td>
</tr>
</tbody>
</table>

Version 2.0:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Constraints</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>INTEGER</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>FIRSTNAME</td>
<td>VARCHAR(256)</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>LASTNAME</td>
<td>VARCHAR(256)</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>DATE_OF_BIRTH</td>
<td>DATE</td>
<td>NOT NULL</td>
<td></td>
</tr>
</tbody>
</table>
Automatic RI check

• All RI dependencies should be brought to test/development
Insert Options

• Completely replace target
• Append to target tables
• Merge based on primary key
Characteristic of Row Level Processing

- RI identification and transfer of relevant data to test system
- Flexible data customization
- Automates refresh of test case data
- Not suitable for mass data
- Developer tool
Manage Test Data Versions
Make Tests Reproducible
The Dilemma

- Tests should run under same preconditions
- Tests modify the data
VERSIONING OF TEST DATA

BACKUP BEFORE TEST EXECUTION

FROZEN TEST DATA
TEST DATA MANAGEMENT

VERSIONING OF TEST DATA

BACKUP BEFORE TEST EXECUTION

FROZEN TEST DATA

ON DEMAND
Characteristic of Versioning Management

- Availability of data at the push of a button
- Separate test/development from backup procedures
- Reduce error rates when doing acceptance tests
- Fast provisioning of test data instead of finding modified rows
Anonymization/Masking
What is the benefit of Anonymization?

- Compliance with legislation
- Preservation of reputation
- Avoid to provide competition with critical data
## Judicial directives

<table>
<thead>
<tr>
<th>Directive</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>European Privacy Directive 95/46/EC</td>
<td>European Union</td>
</tr>
<tr>
<td>PIPEDA, Privacy ACT</td>
<td>Canada</td>
</tr>
<tr>
<td>HIPAA, PCI DSS</td>
<td>USA</td>
</tr>
<tr>
<td>IFAI</td>
<td>Mexico</td>
</tr>
</tbody>
</table>
What must be anonymized and where?
Anonymization/Masking in the data landscape

- Strict separation
- Enforce anonymization
Anonymization/Masking in the data landscape

- Strict separation
- Enforce anonymization

Production
DB2, Oracle, etc.

Preproduction
DB2, Oracle, etc.

Anonymized Base Stock

Test Environment

Development
Existing methods
Unload – Modification - Load

While copying
- Fast
- Automated
- Higher safety standard
Rewarding of Anonymization/Masking?

- Keep the danger potential low
- Match compliance rules: yes, we anonymize!
- Improvement of process with every new release
Characteristic of Anonymization/Masking

- Anonymization/Masking cannot be right or wrong
- Every application is different, no silver bullet existing
- Risk / Effort / Costs / Effect must be considered realistically
- No definite rules what **exactly** has to be anonymized exist
Time Journey for Test Data
OS: z/OS, Linux, Unix, Windows

DBMS: DB2/Oracle/SQL-Server
TIME JOURNEY FOR TEST DATA

OS

App Prog

Dates in Data

DBMS

Time-Date Modifier

Tester

Specs List
Test Data Provision

(Pre)Production

Test Environment
Thank you for your attention!