

Automated Tests with Databases

Helves Domingues e Paulo Cheque
Translation: Paulo Cheque

Summer 2009

License:

Creative Commons: Attribution-Share Alike 3.0 Unported
<http://creativecommons.org/licenses/by-sa/3.0/>



Why Test Databases?

- Data => \$
- Logic
 - Stored Procedures
 - Triggers
- SQL
- Design => Maintenance and Evolution
- ORM Mapping
- Security

Why Test Databases?

- What about evolutive databases?
 - Evolutive data modeling
 - Test with databases
 - Configuration management (SVN, CVS, GIT, etc)
 - Database refactoring

Concerns

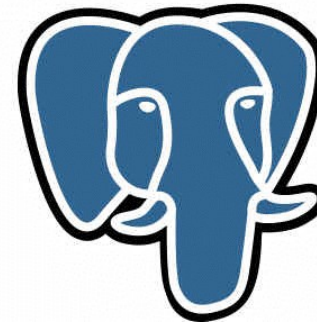
- Database sharing
- Permissions: Schemas, CRUD
- Concurrent execution of test
- A lot of systems using the same base
- Legacy systems
- Connection – Network - Distribution:
Performance
- Replication and Auditing

Test Environment

- Shared databases
- Local databases
- Memory databases



PostgreSQL



ORACLE®

Shared Databases

- Advantages:
 - Real data of users
 - Environment closer to a real one
 - Existent data: Lots of scenarios
- Disadvantages:
 - Not controlled environment for tests
 - Permission problems
 - Concern to the “original” data
 - Concern to the data created by tests
 - Slow

Local Databases

- Advantages:

- Controlled environment for tests
- Better performance

- Disadvantages:

- Cost to prepare data
- Additional queries
- Data created by developers, not users
- Acceptance tests?
- Concern to data created by tests
- Licenses

Memory Databases

- Advantages:

- Performance
- Controlled environment for tests

- Disadvantages:

- Not real database
- Cost to prepare data
- Additional Queries
- Data created by developers, not users
- Acceptance tests?
- Limitations: Stored Procedures, Triggers...

Questions

- Is it possible to choose the environment of tests?
- What's the best way to choose?
- What should be tested?
- Where are most of the errors?
- What is the cost/benefit ratio?

Strategies

- Developer Environment:
 - Version Control
 - Scripts
 - Individual databases

- Repository \Leftrightarrow Artifacts \Leftrightarrow Control

Strategies

- Environment, databases, transitions
- Development (Frequent)
- Integration (Frequent)
- Approval (Controlled)
- Production (Approved by final user)

Strategies

- Transition to integration environment:
- 1) Continuous Integration Tool reads Repository
- 2) Continuous Integration Tool applies database refactoring
- 3) Continuous Integration Tool runs tests using database

Patterns

- **Organization:**
 - Testcase Superclass
 - Ex: GrailsUnitTestCase
 - Test Utility Method
 - Test Helper
- Database Sandbox: 1 per developer
- Transaction Rollback Teardown
- Table Truncation Teardown

+ Patterns

- Generated Values
 - IDs
 - Timestamp
 - Random

```
new Person(IDGenerator.uniqueID()).save()
```

```
CreditCardGenerator.generate(CreditCardFactory.getMasterCard())
```

Anti-patterns

- Fragile tests
- Data/Environment not controlled
- Ids hard-coded
- Slow
- Test data in database

System Under Test

- ... Database Connectivity (DBC)
 - ODBC, JDBC ...
- Frameworks for queries
 - Spring-JDBC ...
- Object-Relational Mapping
 - Hibernate
 - ActiveRecord
- Stored Procedures - Triggers

DBC

- String errors
 - Conversion
 - Null values
- Low coupling
 - Code replication
- Connection
- Transactions

Object-Relational Mapping (ORM)

- Types
- Validations
 - Min, max, range, null, empty, default, precision
- Conversion: Primitive types or objects
- Metadata: XML, Annotations, Conventions ...

ORM: Mapping

- Mapping:
 - 1-1, 1-N, N-N
- Inheritance:
 - Unique table
 - Table per type
- Lists, maps ...

ORM: Facilities

- Cascade of Insert, Update and Delete
- Objects loading:
 - Lazy, eager
- Cache
- Events
 - Before/After: insert / update / delete / load

ORM: Frameworks

- Errors / Limitations
- SQL Grammar specific to a DB
- Database particularity
- Hibernate example:

```
@GeneratedValue(strategy = GenerationType.AUTO)  
@GeneratedValue(strategy = GenerationType.SEQUENCE)  
// Works on MySQL?
```

ORM: Problems

- Reflection (entities with basic constructor...)
- Serializable entities
- Correct metadata?
- Bi-directional mapping without owner?
- Is design correct?
 - TablePerClass, SingleTable...

Stored Procedures

- Tests inside databases
- Tests at the application level
 - Stored Procedure Test
 - Facility
 - Little control
 - Input and output

Triggers

- Delay
- Replication
- Common DB? Multiple users?
- Multiple developers?
- Validation
- Generated values

ALTER TABLE ENABLE/DISABLE TRIGGER

Performance

- Queries:
 - Join slow
 - Heavy query
 - Tens of millions of registers
- Connection: Pool of connections, # of threads and Timeout
- Solution: Collect access plan and make improvements of performance

Security

- SQLInjection I e II

```
....'; DROP TABLE Tabela; --
```

- Memory overflow:

```
select * from Tabela where x = ' ' or 1 = 1
```

- Errors in drivers, bases...

Contact

<http://www.agilcoop.org.br>

<http://ccsl.ime.usp.br>

<http://qualipso.org>

helves@ime.usp.br

paulocheque@agilcoop.org.br



License:

Creative Commons: Attribution-Share Alike 3.0 Unported

<http://creativecommons.org/licenses/by-sa/3.0/>

