

Testen von Datenbanken

Inhalt

Einleitung.....	2
Mocking.....	2
Testen gegen echte Datenbanken.....	2
Datenbankvalidierungen	3
ACID.....	3
Types of Testing.....	4
Black Box Testing.....	4
White Box Testing	4
Types of Database Testing.....	5
Database Schema Testing.....	6
Database Testing Tools.....	8
Datenbank-Testschritte.....	8
Database Migrations	8
Quellen	9

Einleitung

Die Datenbank ist ein wesentlicher Bestandteil eines Systems.

Das Testen von Datenbanken hilft uns Schwachstellen in einem Datenbanksystem herauszufinden und die Datenbank vor einem instabilen Zustand zu schützen.

Mocking

Stellen Sie sich eine Reihe von Tests vor, die aufgesetzt wurden, um gegen JDBC zu mocken und außerdem, dass Ihr Code zum Spring JdbcTemplate, zu jOOQ oder zu Hibernate wechselt. Die Tests müssen nun umgeschrieben werden um gegen diese Frameworks zu mocken – keine perfekte Lösung.

Testen gegen echte Datenbanken

In-Memory-Datenbanken: Einer der einfachsten und schnellsten Wege um eine Datenbank aufzusetzen, gegen die dann getestet wird, ist die Verwendung einer der verfügbaren In-Memory-Versionen, z. B. H2, HSQL oder Derby.

Einer der großen Vorteile dieses Ansatzes ist, dass er schnell ist. Sie können eine neue Datenbankinstanz für wirklich jeden Test erstellen, der eine benötigt, wodurch Sie eine absolute Garantie haben, dass die Daten sauber sind. Sie brauchen auch keine extra Infrastruktur auf ihrem Entwicklungsrechner dafür, weil alles in der JVM erledigt wird. Dieser Weg hat aber auch Schattenseiten.

Wenn Sie nicht auf derselben In-Memory-Datenbank deployen, die Sie in Ihrem Test nutzen, bekommen sie automatisch Probleme mit der Kompatibilität, die sich nicht zeigen, bevor Sie ein höheres Level der Tests erreichen – oder, Gott bewahre, die Produktion! Weil Sie eine andere Datenquelle für ihre Produktionsinstanz verwenden, passiert es ganz schnell, Optionen in der Konfiguration zu übersehen, die aber notwendig sind, damit der Treiber korrekt läuft.

Glücklicherweise sind in den letzten Jahren einige Tools herausgekommen, die diesen Vorgang vereinfachen. Die wichtigsten davon sind Vagrant und Docker. Eine lokale MySQL-Version aufzusetzen, kann mit Docker einfach nur über das Ausführen des folgenden Kommandos erledigt werden:

```
1$ docker run -p 3306:3306 -e MYSQL_ROOT_PASSWORD=bob mysql
```

Datenbankvalidierungen

Das Testen von Datenbanken umfasst die folgenden vier Validierungen:

- Data mapping.
- ACID (Atomicity, Consistency, Isolation, Durability) properties validation.
- Data Integrity.
- Business Rule Conformance.

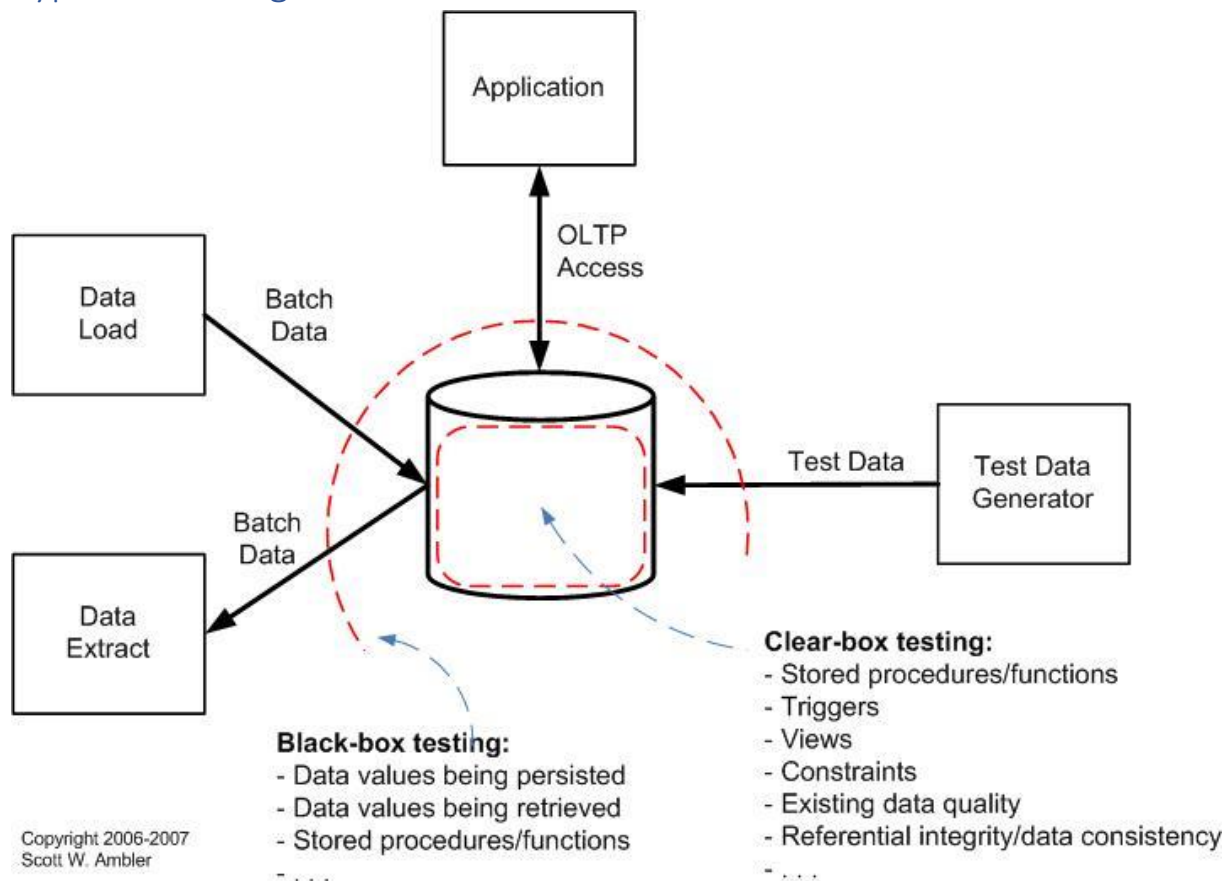
ACID

Beschreibt die Eigenschaften von Transaktionen in Datenbankmanagementsystemen (DBMS).

ACID



Types of Testing



Black Box Testing

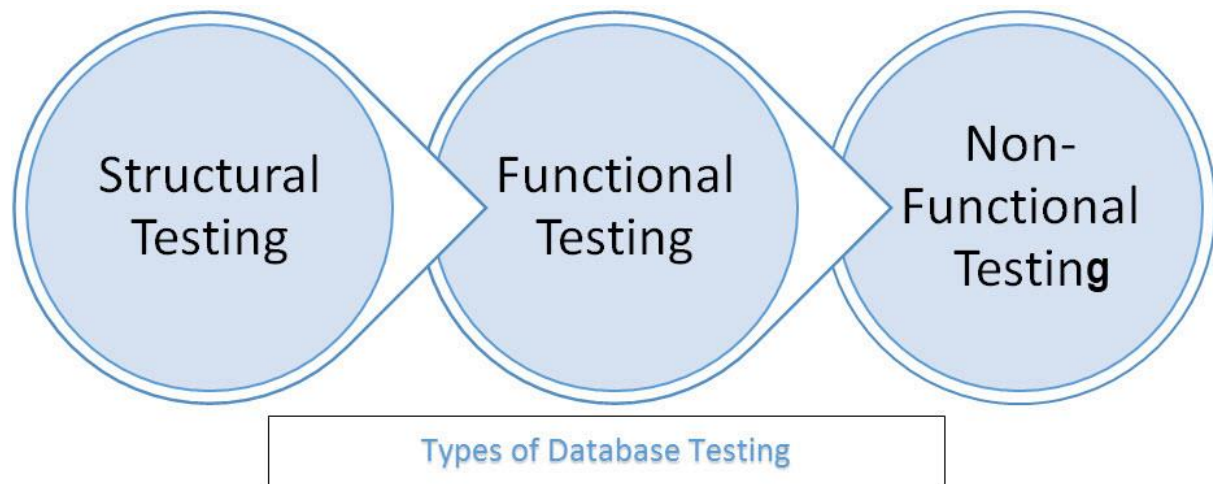
- Data Mapping.
- Verifying stored and retrieved data.
- Uses black box testing techniques such as equivalence partitioning and boundary value analysis.

White Box Testing

- Testing database triggers and logical views.
- Validating data models and database schema.
- Checking referential integrity and database consistency.
- Use white box testing techniques such as statement coverage, decision coverage, condition coverage.

Types of Database Testing

Funktion und Struktur bestimmen, welche Art von Tests auf einem Datenbanksystem durchgeführt werden müssen. Auf der Grundlage dieser Faktoren kann das Testen von Datenbanken weiter unterteilt werden in:



Structural database testing: Komponenten der Datenbank, die für die Benutzer nicht sichtbar sind, müssen getestet werden. Dazu kann der strukturelle Datenbanktest verwendet werden. Ein weiterer Vorteil dieser Art von Tests ist, dass sie zur Validierung der Datenbank verwendet werden können.

It deals with table and column testing, schema testing, stored procedures and views testing, checking triggers, etc.

Functional testing: Aus Benutzersicht sind viele Funktionalitäten mit der Datenbank verbunden. White-Box- und Black-Box-Tests sind die Arten von Tests, die zu diesem Zweck verwendet werden

It involves checking functionality of database from user point of view.

Stored procedures and triggers. Stored procedures and triggers should be tested just like your application code would be.

Non-functional testing: Risikofaktor der Datenbank, Anforderungen, Performance, etc. der Datenbank muss getestet werden und das ist der nicht-funktionale Faktor jeder Datenbank.

It involves load-testing, risk testing in database, stress testing, minimum system requirements, and deals with the performance of the database.

Database Schema Testing

Tables, columns, column types check:

- Name of all the tables in database
- Column names for each table
- Column types for each table
- **NULL** value checked or not
- Whether a default is bound to correct table columns

Schlüssel und Indizes:

- Primärschlüssel für jede Tabelle
- Fremdschlüssel für jede Tabelle
- Datentypen zwischen einer Fremdschlüsselspalte und einer Spalte in anderen Tabellenindizes, gruppiert oder nicht gruppiert, eindeutig oder nicht eindeutig

Stored Procedure Tests:

- Stored procedure name
- Parameter names, parameter types, etc.
- **Output** – Whether the output contains many records. Zero rows are effected or only a few records are extracted.
- What is the function of Stored Procedure and what a stored procedure is not supposed to do?
- Passing sample input queries to check if a stored procedure extracts correct data.
- **Stored Procedure Parameters** – Call stored procedure with boundary data and with valid data. Make each parameter invalid once and run a procedure.
- **Return values** – Check the values that are returned by stored procedure. In case of a failure, nonzero must be returned.
- **Error messages check** – Make changes in such a way that the stored procedure fails and generate every error message at least once. Check any exception scenarios when there is no predefined error message.

Trigger Tests:

- Make sure the trigger name is correct.
- Validate the trigger if it is generated for a specific table column.
- Trigger's update validation.
- Update a record with a valid data.
- Update a record with invalid data and cover every trigger error.
- Update a record when it is still referenced by a row in other table.
- Ensure rolling back transactions when a failure occurs.
- Find out any cases in which a trigger is not supposed to roll back transactions.

Database Testing Tools

Some of the tools mentioned below that help to execute these type of testing on a database.

- Database Benchmark
- Database Rider
- Db stress
- DbUnit
- DB Test Driven
- HammerDB
- JdbcSlim
- JDBC (Java DataBase Delta Testing)
- Nbi
- NoSQLMap
- NoSQLUnit
- ruby-plsql-spec
- SeLite
- sqlmap
- tSQLt
- [Tsung](#)
- utPLSQL

More on: <https://www.softwaretestinghelp.com/tools/40-best-database-testing-tools/>

Datenbank-Testschritte

1. Bereiten Sie die Umgebung vor
2. Führen Sie einen Test durch
3. Testergebnis prüfen
4. Validierung entsprechend der erwarteten Ergebnisse
5. Berichterstattung an die jeweiligen Stakeholder

Database Migrations (Flyway)

Es besteht die Notwendigkeit einer Datenbankmigration während der Durchführung von Datenbanktests. Die DB-Migration ermöglicht folgende Dinge:

- Die Datenbank von Grund auf neu zu erstellen.
- Den Zustand der Datenbank festzulegen.
- Von der aktuellen Version der Datenbank zu einer neueren Version zu wechseln.

Quellen

<https://www.xenonstack.com/insights/what-is-database-testing/>

<https://jaxenter.de/wohl-oder-uebel-es-muss-gemacht-werden-datenbank-tests-26109>

<http://www.agiledata.org/essays/databaseTesting.html>

<http://www.agiledata.org/essays/whatToTest.html>

<https://www.testbytes.net/blog/database-testing/>

<https://www.guru99.com/data-testing.html>

https://www.tutorialspoint.com/database_testing/

<https://www.softwaretestinghelp.com/database-testing-process/>

<https://www.techbeamers.com/database-testing/#what-is-database-testing-and-why-is-it-important>

<https://de.wikipedia.org/wiki/ACID>

<https://image.slidesharecdn.com/bigdata20153-150312093513-conversion-gate01/95/bigdataanalyse-und-nosqldatenbanken-44-638.jpg?cb=1426153163>