
Anleitung zur Experiment-Durchführung

Signature-Based Inference-Usability Confinement for Relational Databases under Functional and Join Dependencies

Martin Bring und Jaouad Zarouali



VERSION 1.0

März 2013

Inhaltsverzeichnis

1	Einleitung	3
2	Wichtige Vorbemerkung	3
3	Einrichtung	3
3.1	Voraussetzungen	3
3.2	Einrichtung des Prototypen	3
3.3	Einrichtung der Datenbank	4
4	Experiment	4
4.1	Konfiguration	5
4.2	Durchführung	5

1 Einleitung

Diese Anleitung beschreibt die Durchführung des in [1] beschriebenen Experiments zur signaturbasierten Inferenzkontrolle. Die Implementierung der signaturbasierten Inferenzkontrollen ist Teil eines in Java implementierten Prototypen, der am Lehrstuhl 6 der Fakultät für Informatik der TU Dortmund entwickelt wird. Für die Durchführung des Experiments ist deshalb eine lauffähige Version des Prototypen erforderlich. Die Einrichtung des Prototypen ist in Abschnitt 3 beschrieben. Im Anschluss daran behandelt Abschnitt 4 die Konfiguration und Durchführung des Experiments. Bevor jedoch auf die Einrichtung des Prototypen und Durchführung der Experimente eingegangen wird, beinhaltet Abschnitt 2 eine wichtige Vorbemerkung.

2 Wichtige Vorbemerkung

Bei der Archivierung der Experimente ist ein kleinerer, die Ergebnisse nicht wesentlich verfälschender, Fehler im ursprünglichen Aufbau der Experimente aufgefallen. Die in den Experimenten in [1] zur Anwendung gekommene Instanz hat aufgrund eines Programmierfehlers die Verbundabhängigkeiten nicht vollständig erfüllt. Dies hat zur Folge, dass in der verwendeten Instanz 28 Tupel fehlen (79 anstatt 107 Tupel). Die in dieser Anleitung beschriebenen Experimente berücksichtigen (im Gegensatz zu den in [1] beschriebenen Experimenten) die korrekte 107 Tupel umfassende Instanz. Bzgl. der Ergebnisse ergeben sich bei der vergrößerten Instanz bedingt durch die zusätzlichen Tupel geringfügig erhöhte jedoch nicht signifikant höhere Laufzeiten.

3 Einrichtung

Dieser Abschnitt beschreibt die Einrichtung des Prototypen. Die Voraussetzungen sind in Unterabschnitt 3.1 aufgeführt. Die Einrichtung des Prototypen behandelt Unterabschnitt 3.2. Das Erstellen des vom Prototypen verwendete Datenbankschemas ist in Unterabschnitt 3.3 beschrieben.

3.1 Voraussetzungen

Für die Ausführung des Prototypen wird die Oracle-Java-Laufzeitumgebung in der Version SE 6 benötigt. Als zugrundeliegende Datenbank wird eine Oracle-Datenbank in der Enterprise Edition der Version 11g Release 1 benötigt. In dieser Datenbank müssen für den Prototypen zwei Datenbankbenutzer existieren. Einen Maintenance-Datenbank-Benutzer für die Maintenance-Datenbank und einen Applikationsdatenbank-Benutzer für die Applikationsdatenbank. Die Namen der Benutzer sind dabei frei wählbar, müssen jedoch wie in den folgenden Abschnitten beschrieben in den Konfigurationsdateien hinterlegt werden. Die Benutzer müssen mit entsprechenden Privilegien ausgestattet sein. Der Maintenance-Datenbank-Benutzer benötigt die folgenden Systemprivilegien: `CREATE TABLE`, `CREATE TRIGGER`, `CREATE SEQUENCE`, `CREATE PROCEDURE` und `CONNECT`. Der Applikationsdatenbank-Benutzer benötigt die beiden Systemprivilegien `CREATE TABLE` und `CONNECT`. Zusätzlich benötigt der Maintenance-Datenbank-Benutzer ein `SELECT`-Recht auf einer Tabelle des Applikationsdatenbank-Benutzers, dessen Gewährung in Unterabschnitt 3.3 beschrieben wird. Im Folgenden wird davon ausgegangen, dass sowohl die Java-Laufzeitumgebung als auch eine Oracle-Datenbank zur Verfügung stehen.

3.2 Einrichtung des Prototypen

Für die Einrichtung des Prototypen muss zunächst die ZIP-Datei entpackt werden. Diese enthält den folgenden Inhalt:

- `external/`
- `cie_prototype.jar`
- `createTrigger.sql`
- `db.xml`

- `experiment.properties`

Zunächst müssen zwei externe Bibliotheken heruntergeladen und als jar-Datei in das Verzeichnis `external/` kopiert werden. Dies ist zum einen das vom Prototypen verwendete Logging-Framework `log4j`¹ in der Version 1.2.17 und zum anderen der Oracle JDBC Datenbanktreiber² für Java SE 6.

Anschließend müssen die Zugangsdaten der beiden Benutzer für die Oracle-Datenbank in der Datei `db.xml` für den Prototypen hinterlegt werden. Das Element `<MaintainDB>` muss die Zugangsdaten für den Maintenance-Datenbank-Benutzer und das Element `<CQEDB>` die Zugangsdaten für den Applikationsdatenbank-Benutzer enthalten.

Die Einrichtung des Prototypen ist hiermit abgeschlossen.

3.3 Einrichtung der Datenbank

Für das Anlegen der vom Prototypen benötigten Relationen in der Datenbank bietet der Prototyp eine grafische Oberfläche. Diese lässt sich unter UNIX-basierten Betriebssystemen durch den folgenden Befehl aufrufen:

```
java -cp ../*:external/* server.start.SchemaAdmin
```

Unter Windows-basierten Betriebssystemen lautet der Befehl wie folgt:

```
java -cp .;.*\*;external\* server.start.SchemaAdmin
```

Im obersten Drop-Down-Feld ist die zu verwendende Datenbankkonfiguration auszuwählen. Das zu generierende Datenbankschema kann mittels des zweiten Drop-Down-Feldes ausgewählt werden. Für die Durchführung des Experiments muss das „Signature Experiment Schema“ ausgewählt werden. Ein Klick auf den Button „Create Schema“ stößt die Erstellung des ausgewählten Schemas an. Nach Abschluss der Schemaerstellung füllt ein Klick auf den Button „Fill with contents“ das neu erstellte Schema mit Inhalt. Anschließend kann die grafische Oberfläche geschlossen werden.

Für die signaturbasierte Inferenzkontrolle, die durch Trigger realisiert wird, müssen zusätzlich noch einige weitere Schritte durchgeführt werden. Da die Trigger vom Maintenance-Datenbank-Benutzer ausgeführt werden, müssen diesen entsprechende `SELECT`-Rechte auf den Tabellen des Applikationsdatenbank-Benutzers gewährt werden. Angemeldet als Applikationsdatenbank-Benutzer ist hierzu der folgende Befehl an die Datenbank zu senden:

```
GRANT SELECT ON SIG_KRANKHEIT TO Maintenance-Datenbank-Benutzer;
```

Maintenance-Datenbank-Benutzer muss im obigen Befehl durch den tatsächlich gewählten Benutzernamen ersetzt werden.

Anschließend müssen die Trigger in die Maintenance-Datenbank eingefügt werden. Hierzu ist das mitgelieferte SQL-Skript `createTrigger.sql` zu modifizieren, indem in den Zeilen 15 und 16 die Benutzernamen des Maintenance-Datenbank- und Applikationsdatenbank-Benutzers angegeben werden, d.h. die Konstanten `appDBUser` und `maintainDBUser` gesetzt werden. Daran anschließend muss das SQL-Skript als Maintenance-Datenbank-Benutzer ausgeführt werden, sodass die Trigger in der Maintenance-Datenbank angelegt werden. Dabei erstellt das SQL-Skript die beiden Trigger `SIGS_FLEX` und `SIGS_SIG_KRANKHEIT` sowie die Prozeduren `SIGNATURE_TRANSFORMATION` und `POT_SEC_TRANSFORMATION`.

Die Datenbank-Einrichtung ist hiermit vollständig.

4 Experiment

Ausgehend von einem lauffähigen Prototypen beschreibt dieser Abschnitt die Konfiguration (Unterabschnitt 4.1) und Durchführung (Unterabschnitt 4.2) des Experiments.

¹<http://logging.apache.org/log4j/1.2/>

²<http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>

4.1 Konfiguration

Die Konfiguration des Experiments erfolgt über die Datei `experiment.properties`. Als `Approach` ist stets der unflexible Ansatz (Wert 0) auszuwählen, da die Experimente in [1] mit diesem Ansatz durchgeführt worden sind.

4.2 Durchführung

Nachdem das Experiment über die Datei `experiment.properties` konfiguriert worden ist, kann es unter UNIX-basierten Betriebssystemen über den folgenden Befehl gestartet werden:

```
java -cp ../*:external/*  
      server.censor.signature.experiment.ExperimentExecutionDBSEC12 experiment.properties
```

Für Windows-basierte Betriebssysteme lautet der Befehl wie folgt:

```
java -cp .;.*;external\*  
      server.censor.signature.experiment.ExperimentExecutionDBSEC12 experiment.properties
```

Nach Abschluss des Experiments können die Ergebnisse in der Tabelle `SIGS_EXPERIMENT_EVAL` bzw. `SIGS_EXPERIMENT_EVAL_CUM` eingesehen werden.

Literatur

- [1] Joachim Biskup, Sven Hartmann, Sebastian Link, Jan-Hendrik Lochner, and Torsten Schlotmann. Signature-based inference-usability confinement for relational databases under functional and join dependencies. In Nora Cuppens-Boulahia, Frédéric Cuppens, and Joaquin Garcia-Alfaro, editors, *Data and Applications Security and Privacy XXVI*, volume 7371 of *Lecture Notes in Computer Science*, pages 56–73. Springer Berlin Heidelberg, 2012.