

Java Database Difference Checker

BY PETER KAUFMAN

ADVISED BY DR. SESSIONS

EXPECTED DATE OF GRADUATION IS MAY 2020

FOR SENIOR PROJECT DEFENSE(CSCI 499)



Problem Statement

- Many companies have a live and development database
- When updated, the live database should reflect the development one
- How to do this?
- Set up a process that updates the live database



Research and Background

- SQL
 - MySQL
 - Prior experience
 - SQLite
 - What was common?
- NoSQL
 - Couchbase
 - Prior experience, but API problems
 - MongoDB
 - Prior experience, but API problems
 - Graphical User Interface (GUI)
 - Implementation while still learning

Types of Headache

Migraine



Hypertension



Stress



Java GUI



Hardware and Software Requirements

Hardware

- A working computer from the 21st century

Software

- Java
- Python 2.7 or later
- Some kind of hosting service for the database implementation if need:
 - MySQL - WAMP
 - SQLite - NA
 - Couchbase - Couchbase Server Community Edition
 - MongoDB - MongoDB Compass Community

Program Description

- The program compares two databases
 - The databases can be connected in two ways
- Can update live database
- Keeps track of run based data like errors and commands used to update the live database
- Works on NoSQL and SQL databases
 - Couchbase & MongoDB and MySQL & SQL Lite

Default Application Page

The screenshot shows a window titled "Database Difference Checker" with standard Windows window controls (minimize, maximize, close). The interface includes a tabbed menu at the top with "Compare to Database" selected, and other tabs for "Compare to Snapshot", "Create Snapshot", "Logs", and "Last Run". Below the tabs is a section header "Development and Live Database Information". Under this header is a dropdown menu labeled "Select Database Type". The main area is divided into two vertical panels: "Development Information" on the left and "Live Information" on the right. At the bottom of these panels are two buttons: "Produce Statements" and "Run Statements". Below these buttons is a section labeled "Preview:" followed by a large, empty rectangular box.

Two Database Comparison

Database Difference Checker

Compare to Database Compare to Snapshot Create Snapshot Logs Last Run

Development and Live Database Information

MySQL

Development Information	Live Information
Username:	Username:
Password:	Password:
Host:	Host:
Port:	Port:
Database Name:	Database Name:

Produce Statements Run Statements

Preview:

Database Difference Checker

Compare to Database Compare to Snapshot Create Snapshot Logs Last Run

Development and Live Database Information

MySQL

Development Information	Live Information
Username: pjkaufman	Username: pjkaufman
Password:	Password:
Host: localhost	Host: localhost
Port: 3308	Port: 3308
Database Name: live	Database Name: live2

Produce Statements Run Statements

Database Comparison Complete

Preview:

```
`new_tablecol1` varchar(45) DEFAULT NULL,  
PRIMARY KEY (`idnew_table`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;  
SET FOREIGN_KEY_CHECKS=1;
```

Creating a Database Snapshot

Database Difference Checker

Compare to Database Compare to Snapshot **Create Snapshot** Logs Last Run

Development Database Information

MySQL

Development Information

Username:

Password:

Host:

Port:

Database Name:

Take Snapshot

Database Difference Checker

Compare to Database Compare to Snapshot **Create Snapshot** Logs Last Run

Development Database Information

MySQL

Development Information

Username: pjkaufman

Password:

Host: localhost

Port: 3308

Database Name: live

Take Snapshot

Database Snapshot Complete

Comparing a Database with a Snapshot

Database Difference Checker

Compare to Database **Compare to Snapshot** Create Snapshot Logs Last Run

Live Database Information

MySQL

Live Information

Username:

Password:

Host:

Port:

Database Name:

Produce Statements Run Statements

Preview:

Database Difference Checker

Compare to Database **Compare to Snapshot** Create Snapshot Logs Last Run

Live Database Information

MySQL

Live Information

Username:

Password:

Host:

Port:

Database Name:

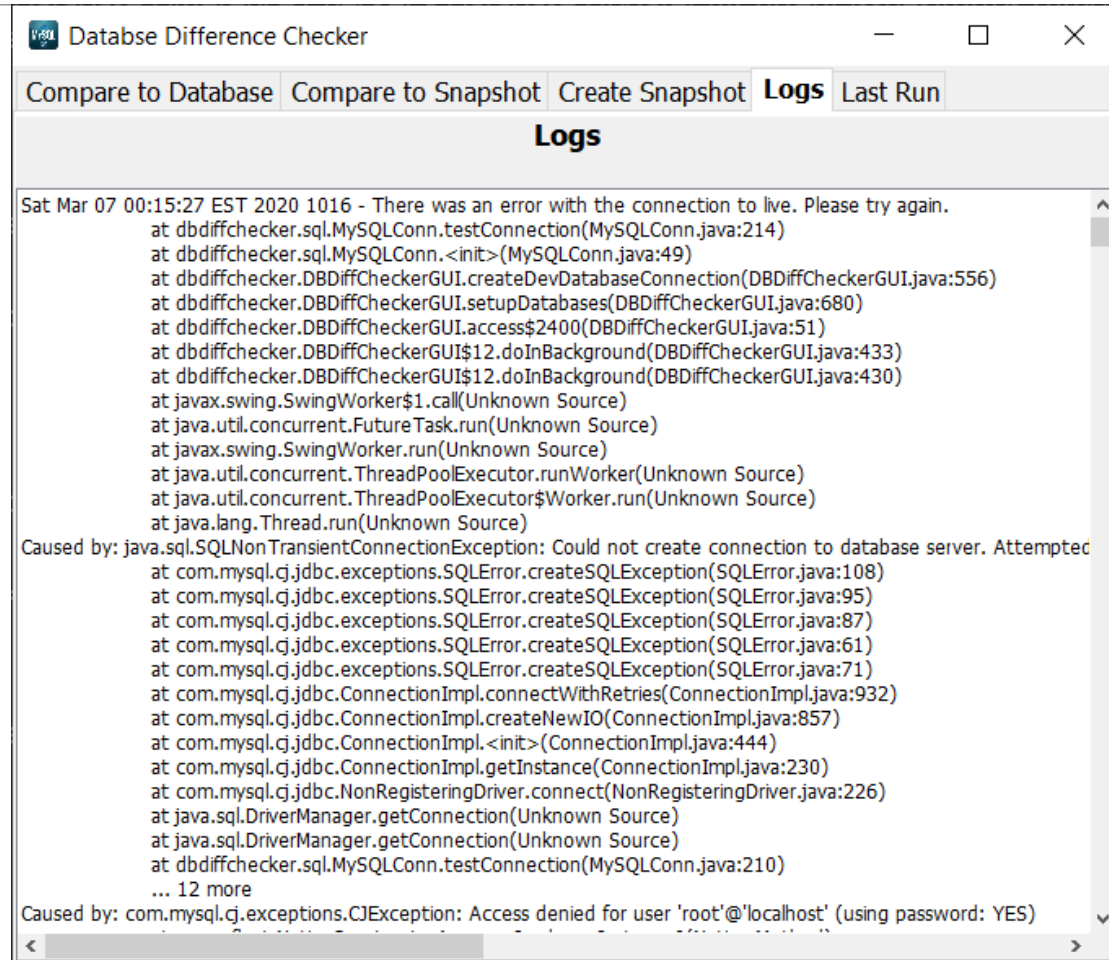
Produce Statements Run Statements

Database Comparison Complete

Preview:

```
new_tablecol1` varchar(45) DEFAULT NULL,  
PRIMARY KEY (`idnew_table`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;  
SET FOREIGN_KEY_CHECKS=1;
```

Logs



Database Difference Checker

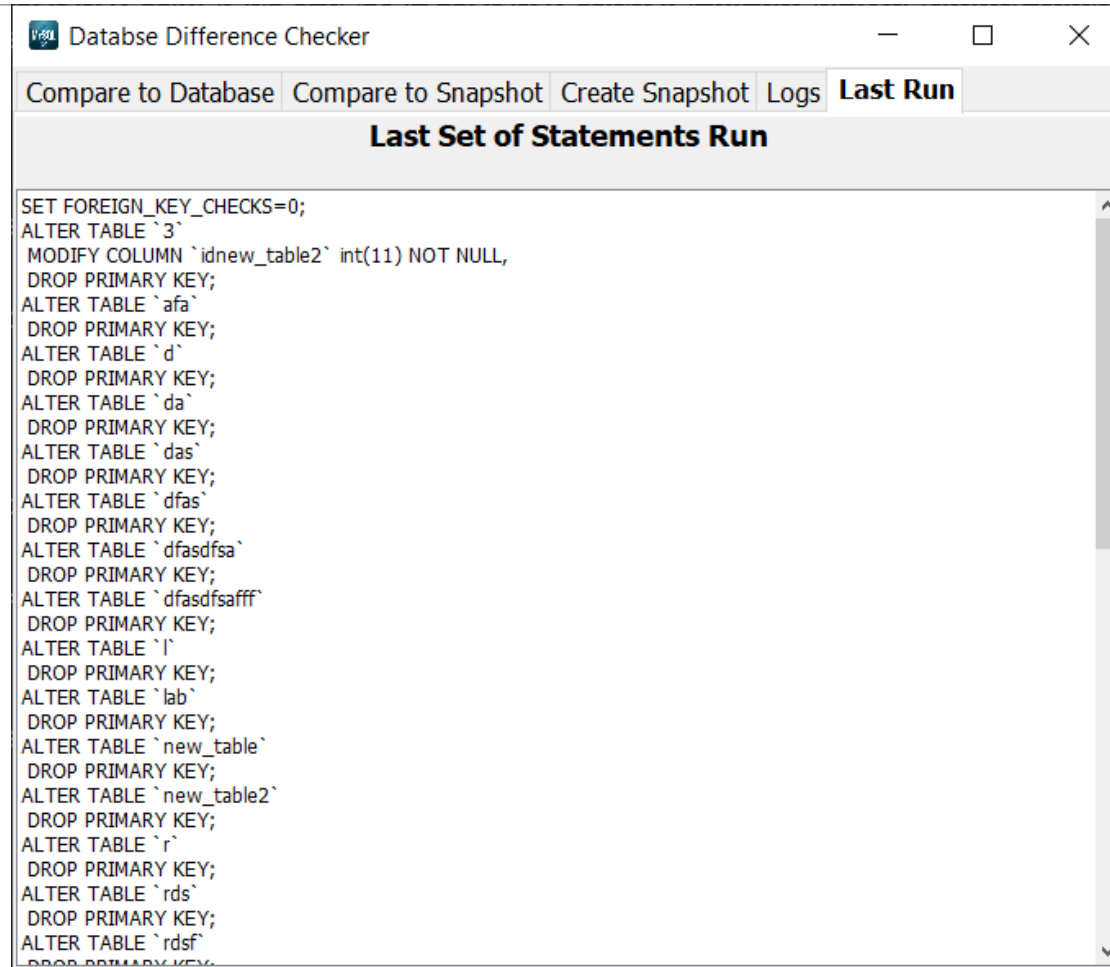
Compare to Database Compare to Snapshot Create Snapshot **Logs** Last Run

Logs

Sat Mar 07 00:15:27 EST 2020 1016 - There was an error with the connection to live. Please try again.

```
at dbdiffchecker.sql.MySQLConn.testConnection(MySQLConn.java:214)
at dbdiffchecker.sql.MySQLConn.<init>(MySQLConn.java:49)
at dbdiffchecker.DBDiffCheckerGUI.createDevDatabaseConnection(DBDiffCheckerGUI.java:556)
at dbdiffchecker.DBDiffCheckerGUI.setupDatabases(DBDiffCheckerGUI.java:680)
at dbdiffchecker.DBDiffCheckerGUI.access$2400(DBDiffCheckerGUI.java:51)
at dbdiffchecker.DBDiffCheckerGUI$12.doInBackground(DBDiffCheckerGUI.java:433)
at dbdiffchecker.DBDiffCheckerGUI$12.doInBackground(DBDiffCheckerGUI.java:430)
at javax.swing.SwingWorker$1.call(Unknown Source)
at java.util.concurrent.FutureTask.run(Unknown Source)
at javax.swing.SwingWorker.run(Unknown Source)
at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
at java.lang.Thread.run(Unknown Source)
Caused by: java.sql.SQLException: Could not create connection to database server. Attempted
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:108)
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:95)
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:87)
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:61)
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:71)
at com.mysql.cj.jdbc.ConnectionImpl.connectWithRetries(ConnectionImpl.java:932)
at com.mysql.cj.jdbc.ConnectionImpl.createNewIO(ConnectionImpl.java:857)
at com.mysql.cj.jdbc.ConnectionImpl.<init>(ConnectionImpl.java:444)
at com.mysql.cj.jdbc.ConnectionImpl.getInstance(ConnectionImpl.java:230)
at com.mysql.cj.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java:226)
at java.sql.DriverManager.getConnection(Unknown Source)
at java.sql.DriverManager.getConnection(Unknown Source)
at dbdiffchecker.sql.MySQLConn.testConnection(MySQLConn.java:210)
... 12 more
Caused by: com.mysql.cj.exceptions.CJException: Access denied for user 'root'@'localhost' (using password: YES)
```

Last Run Statements



```
SET FOREIGN_KEY_CHECKS=0;
ALTER TABLE `3`
MODIFY COLUMN `idnew_table2` int(11) NOT NULL,
DROP PRIMARY KEY;
ALTER TABLE `afa`
DROP PRIMARY KEY;
ALTER TABLE `d`
DROP PRIMARY KEY;
ALTER TABLE `da`
DROP PRIMARY KEY;
ALTER TABLE `das`
DROP PRIMARY KEY;
ALTER TABLE `dfas`
DROP PRIMARY KEY;
ALTER TABLE `dfasdfsaf`
DROP PRIMARY KEY;
ALTER TABLE `dfasdfsafff`
DROP PRIMARY KEY;
ALTER TABLE `l`
DROP PRIMARY KEY;
ALTER TABLE `lab`
DROP PRIMARY KEY;
ALTER TABLE `new_table`
DROP PRIMARY KEY;
ALTER TABLE `new_table2`
DROP PRIMARY KEY;
ALTER TABLE `r`
DROP PRIMARY KEY;
ALTER TABLE `rds`
DROP PRIMARY KEY;
ALTER TABLE `rdsf`
DROP PRIMARY KEY;
```

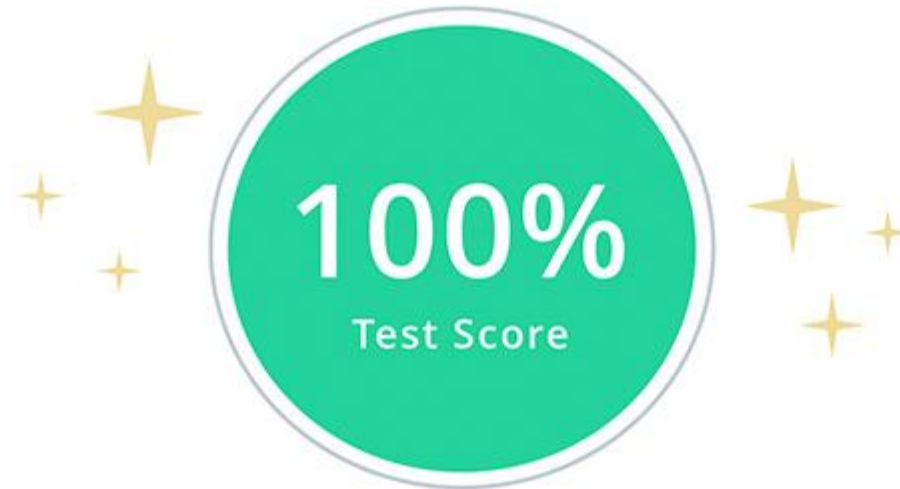
Test Results and Plan

Test Include:

- SQL Tests
 - MySQL Tests
 - SQLite Tests
 - Common Tests
- NoSQL Tests
 - Couchbase Tests
 - MongoDB Tests
- General Tests
 - FileHandler Tests

A total of 48 tests

All tests passed



Congratulations, you passed!

Challenges Overcome

CHALLENGES

1. Other peoples documentation (APIs)
2. SQL Statements
3. GUI on Multiple OSes
4. Automation Script for Multiple OSes
5. Automation Script for Version 2.7 and 3.x
6. Scaling Text for GUI
7. Let User Know that Comparison is Occurring

SOLUTIONS

1. Tutorials and Ctrl + f
2. Research
3. Change JFrame type
4. Had to se `shell=True`
5. Lower level python functions
6. Custom Algorithm
7. Multithreading (SwingWorker)

Future Enhancements

1. Do not actually drop tables or delete documents
2. Add nested view comparison functionality
3. Make the GUI more intuitive*
4. Data input validation*
5. Consistent GUI size*

* These were completed in a separate project

[illegible]

Works Cited

<https://cdn.dribbble.com/users/167717/screenshots/2830953/congrats.jpg>