

# Elastic Enterprise Data Warehouse Query Log Analysis on a Secure Private Cloud

Dr. Kurt Stockinger  
Data Warehouse and Business Intelligence Architect  
Credit Suisse, Zurich

Joint research between Credit Suisse and ETH Zurich:

Willy Lai, Maria Grineva, Maxim Grinev, Donald Kossmann, Georg Polzer, Kurt Stockinger

# Agenda

- Auditing in Enterprises
- Traditional Data Warehouse Approach vs. Cloud Approach
- Query Log Analysis with Xadoop
- Results of Security Analysis on Real Data

# The Challenge that Many Companies Face



- **Requirement:**
  - Every operation against the core database (data warehouse) must be **traceable and explainable**.
  
- **Audits are performed at random points in time:**
  - Which user accessed attribute A1, A3 and A6 of tables T1 and T2?
  - Which user deleted attribute A4 of view V2?
  - Which user updated the value of attribute A5 of table T3?
  
- **Capacity and performance management:**
  - Which table partitions were never accessed over the last year?
    - Candidate for archiving.
  - What are the top 10 longest running queries?
    - Candidate for query optimization.

# Data Warehouse Application Platform

## Data Warehouse Application Platform

- Shared platform for **integrating data** from multiple internal and external sources for developing, deploying and operating applications that implement reporting, analysis and data mining functions.

## Scope

- **Reporting and analysis** (standard and ad-hoc reporting, Online Analytical Processing (OLAP) and **data mining** (in special areas (CRM))
- Data from last **end-of-day** processing (4500 jobs per day)
- No operational/transactional reporting or direct initiation of business transactions

## Key figures

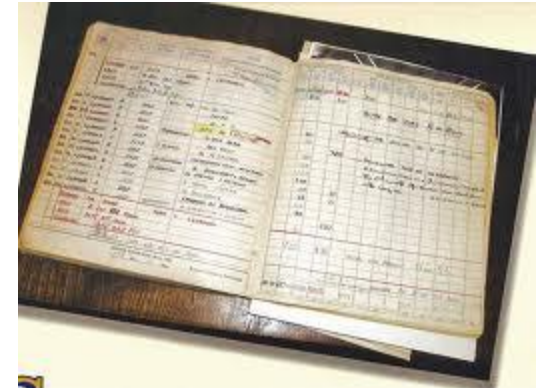
- Servers: 2 M9000, 25 M5000, several V490 / T2000, > 1000 CPUs, 4TB main memory
- ~700 TB storage with growth rate of **15-20 TB/month** (overall ET, IT, UAT and P)  
Throughput between DWH production server and HDS: **10-20 TB/day** (>1 GB/s)

## Users and applications

- **~100 applications** on the platform with some 14'000 users
  - Accounting (Management Accounting, Financial Accounting)
  - Legal & Compliance (Anti Money Laundering, Basel II, Swiss National Reporting)
  - Customer Relationship Management (Front Support , Marketing)
  - Operations (Credit-MIS, Credit Risk) etc.
  - Systems Management (IT DWH)

# Full Logging of Database Queries in Production

- All the **queries against the databases of the data warehouse** are logged via Oracle Audit Option:
  - One XML-file per user session.
  - Compressed at regular intervals.
  - Many Terabytes of uncompressed XML files per month.
  - Data is kept for n days then it is archived.



# Possible Solutions: Data Warehouse vs. Cloud Approach

- Build a **data warehouse**:
  - Not cost effective for a few queries per a year.
- Use **private cloud-computing** approach based on **Hadoop/PIG**:
  - Ad-hoc analysis at unknown time periods:
    - Compute resources are only needed for short periods.
  - **Hadoop** is open source software with proven track record:
    - Parallel file system where data is split into several chunks.
    - Allows parallel analysis/querying of data.
  - **PIG** is XML-based query language that runs on top of Hadoop:
    - Query logs are already stored in XML files.
    - Hadoop/PIG approach can be leveraged to analyze the queries in parallel.
  - Framework can easily be deployed.

# Data Warehouse and Private Cloud Approach



Audit required:  
shipping of XML-based  
query log files



Data warehouse processing  
(typical business analytics)

Hadoop/PIG-based analysis  
of XML query logs  
(large XML file distributed  
over all cloud nodes)

# Data: Audit Logs and Database Schema Snapshots

## Audit logs:

- Contain all database accesses (queries) of the data warehouse.
- The information logged in a single access contains:
  - Audit type
  - Session-, statement-, entry id
  - Extended timestamp
  - DB User, OS User, User Host
  - OS Process
  - ...
  - SQL Text

## Database schema snapshots:

- Generated once a day.
- Capture the entire schema of the data warehouse:
  - Table owners, names and configuration
  - View owners, names, SQL statement and configuration
  - View dependencies
  - Synonyms

# Illustration: Query Processing Steps

## Audit log entry:

```
<AuditRecord>
...
<Extended_Timestamp>2010-06-30 ... </Extended_Timestamp>
<DB_User>DEMOUSER</DB_User>
<Sql_Text>
  select e.*, het.salary, het.title
  from employees e, highsal_engineer_titles het
  where e.emp_no = het.emp_no
</Sql_Text>
<AuditRecord>
```

## Schema snapshot of views:

```
...
<VIEW_NAME>HIGHSAL_ENGINEER_TITLES</VIEW_NAME>
<TEXT_LENGTH>100</TEXT_LENGTH>
<TEXT>
  select e.emp_no, t.title, s.salary
  from employees e, salaries s, title t
  where e.emp_no = s.emp_no and e.emp_no = t.emp_no
  and s.salary >= 200000
  and UPPER(t.title) LIKE '%ENGINEER%'
</TEXT>
...
```

Resolving

Matching

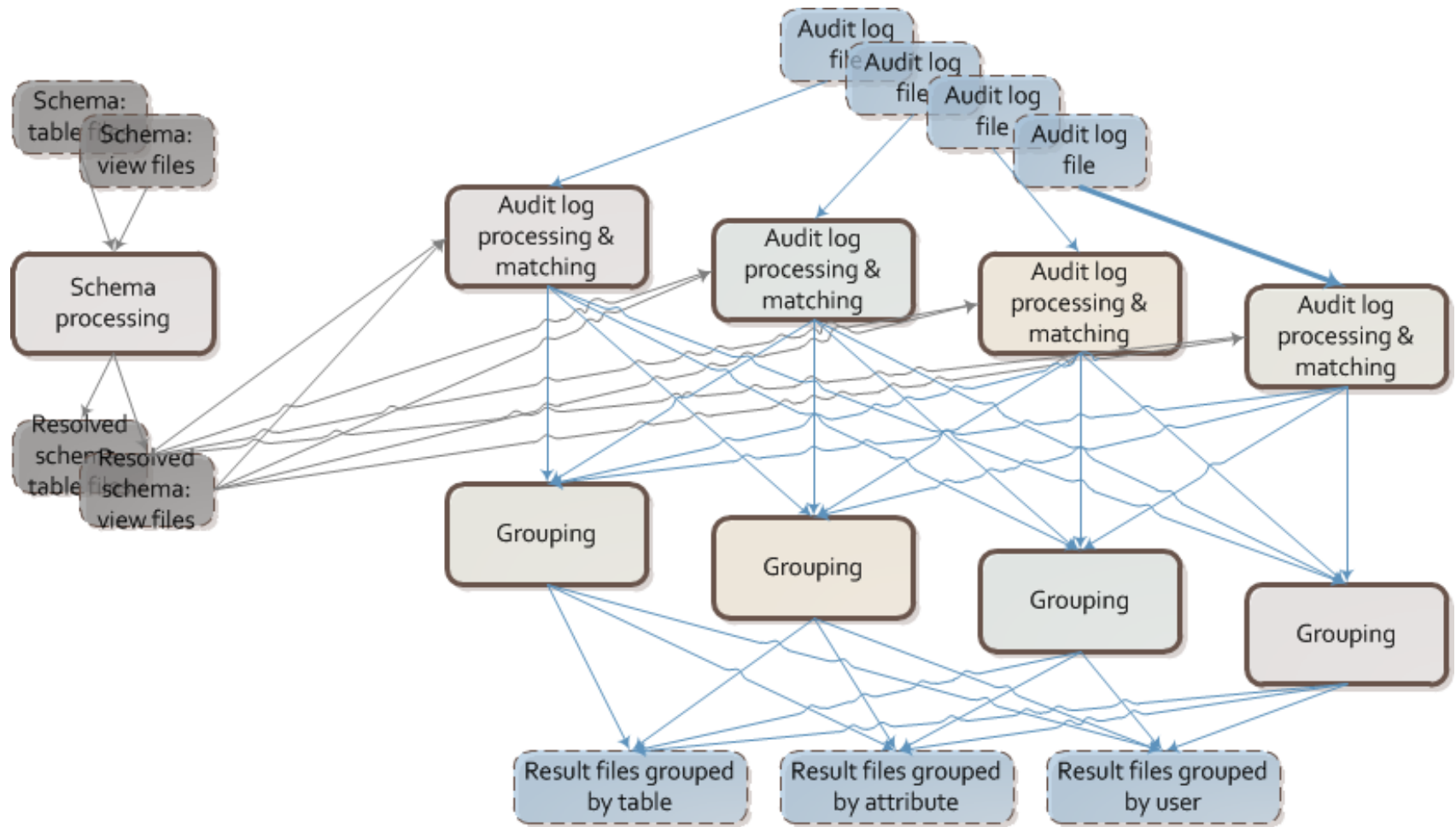
## Grouped & aggregated results

```
tables:
employees {(DEMOUSER,2010-06-30 ... )}
salaries {(DEMOUSER,2010-06-30 ... )}
title {(DEMOUSER,2010-06-30 ... )}
highsal_engineer_titles {(DEMOUSER,2010-06-30 ... )}

attributes:
employees.* {(DEMOUSER,2010-06-30 ... )}
employees.emp_no {(DEMOUSER,2010-06-30 ... )}
title.emp_no {(DEMOUSER,2010-06-30 ... )}
title.title {(DEMOUSER,2010-06-30 ... )}
salaries.salary {(DEMOUSER,2010-06-30 ... )}
salaries.emp_no {(DEMOUSER,2010-06-30 ... )}
highsal_engineer_titles.salary {(DEMOUSER,2010-06-30 ... )}
highsal_engineer_titles.title {(DEMOUSER,2010-06-30 ... )}
  {(DEMOUSER,2010-06-30 ... )}

user:
DEMOUSER {(2010-06-30,
  {(employees),(salaries),(title)},{(employees.*),(employees.emp_no),(title.emp_no),(title.title),(salaries.salary),(salaries.emp_no),(highsal_engineer_titles.salary),(highsal_engineer_titles.title),(highsal_engineer_title
s.emp_no )})}
```

# Architecture of Xadoop-Based Query Processing



# Typical Audit Query: "Was TableX Accessed During Lunch Time?"

```
register ./pigxml.jar
define DATECOMP ch.ethz.xadoodf.DATECOMP();

A = load '/user/DrWho/querylogs1.xml' using ch.ethz.xadoodloader.XMLLoader() as
    (..);
```

# Typical Audit Query: "Was TableX Accessed During Lunch Time?"

```
register ./pigxml.jar
define DATECOMP ch.ethz.xadood.udf.DATECOMP();

A = load '/user/DrWho/querylogs1.xml' using ch.ethz.xadood.loader.XMLLoader() as
    (...);

B = filter A by sql_text matches '*TableX*' and
    DATECOMP((chararray)extended_timestamp, '2011-05-17T12:00:00.000000')>0
    and
    DATECOMP((chararray)extended_timestamp, '2011-05-17T14:00:00.000000')<0;
```

# Typical Audit Query: "Was TableX Accessed During Lunch Time?"

```
register ./pigxml.jar
define DATECOMP ch.ethz.xadoodf.DATECOMP();

A = load '/user/DrWho/querylogs1.xml' using ch.ethz.xadoodloader.XMLLoader() as
    (...);

B = filter A by sql_text matches '*TableX*' and
    DATECOMP((chararray)extended_timestamp, '2011-05-17T12:00:00.000000')>0
    and
    DATECOMP((chararray)extended_timestamp, '2011-05-17T14:00:00.000000')<0;

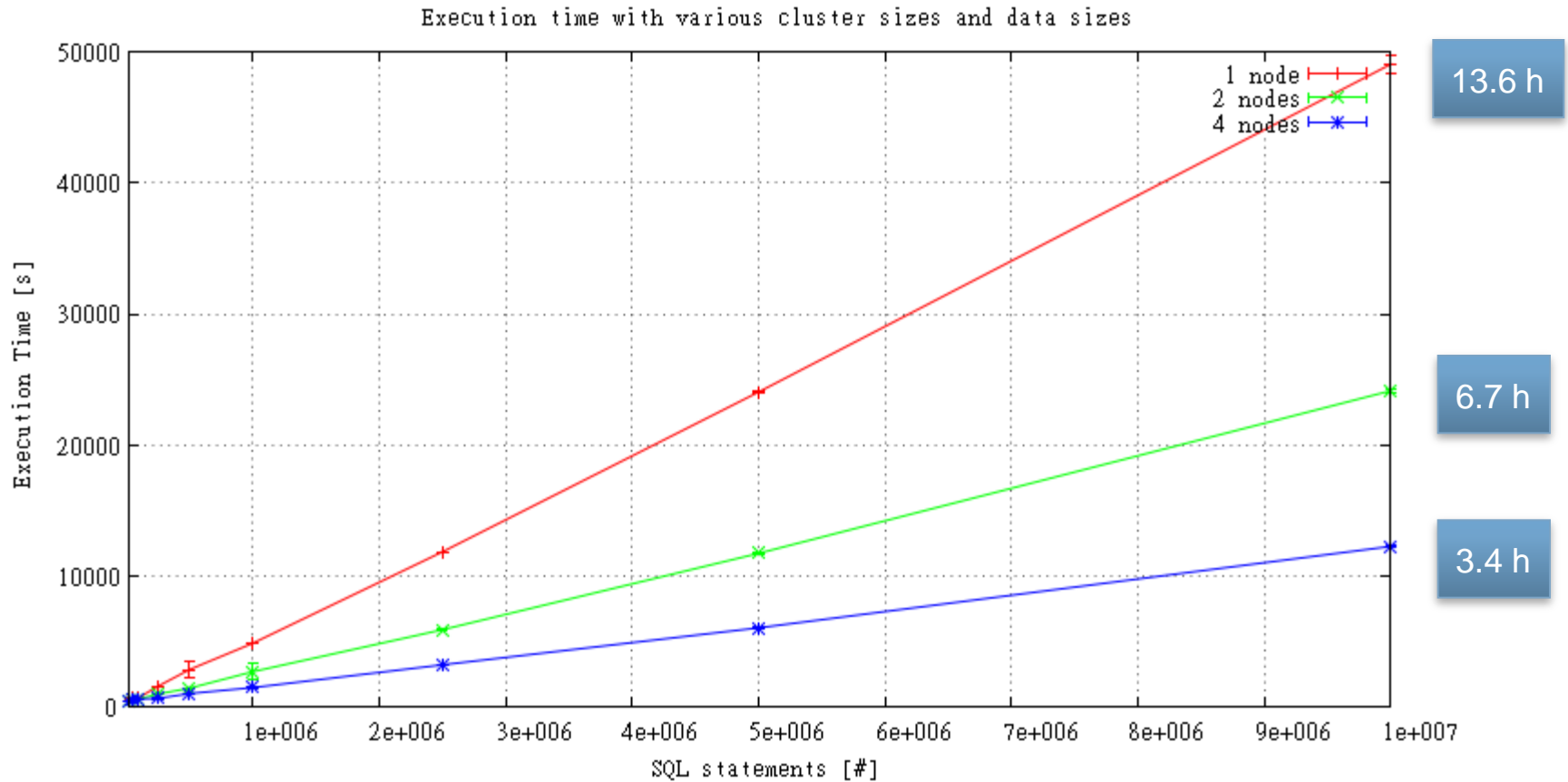
C = foreach B generate db_user, sql_text, extended_timestamp;

dump C;
store C into '/user/DrWho/analysis_querylogs1_2011_05_17.res';
```

## Synthetic Data and Queries at ETH Zurich - Measurement Results on 1 Node

Test set input	Input data size	Output data size		Execution time
1'000'000	1'215 MB	Tables: Attributes: Users:	423 MB 1'196 MB 637 MB	1.4 h
2'500'000	3'038 MB	Tables: Attributes: Users:	1057 MB 2'982 MB 1'392 MB	3.3 h
5'000'000	6'075 MB	Tables: Attributes: Users:	2'114 MB 5'958 MB 3'185 MB	6.7 h
10'000'000	12'151 MB	Tables: Attributes: Users:	4'226 MB 11'620 MB 6'369 MB	13.6 h

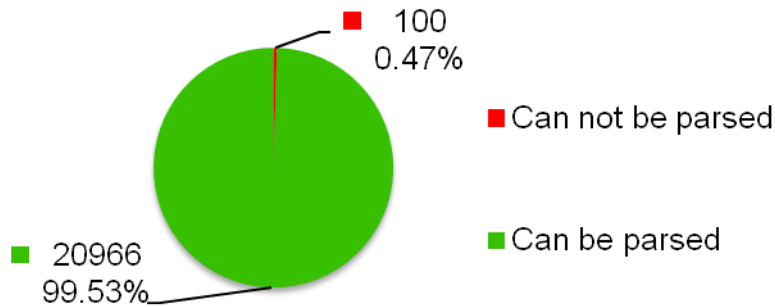
# Execution Times Measured at ETH Zurich



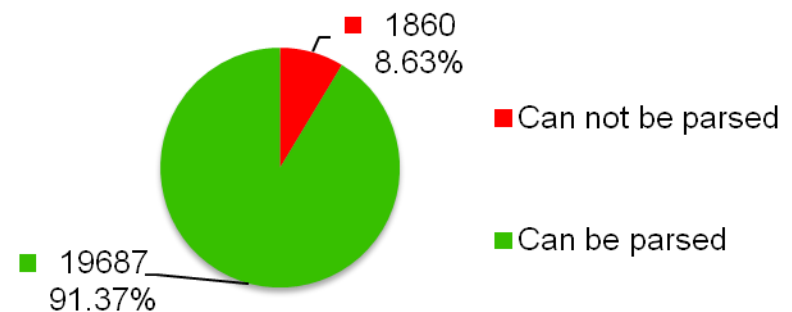
# Experiments inside Credit Suisse on Real Data #1

- We ran our **SQL parser** on real **audit log/schema view snapshot files**.
- We measured the number of audit records/view SQL statements that could be parsed.

Parsing audit log entries



Parsing schema views

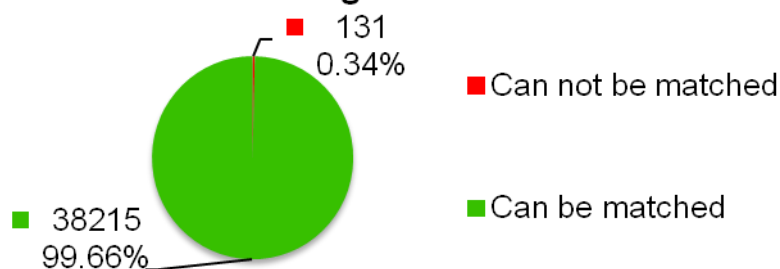


- **8.63%** of view SQL statements could not be parsed due to:
  - **Syntactically incorrect SQL statements.**
  - Issues in the generation of the schema snapshots (**Oracle bug**).

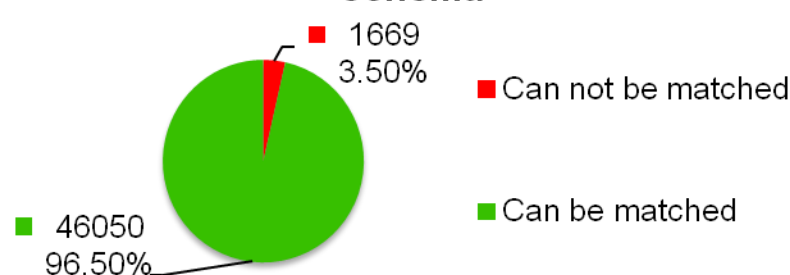
# Experiments inside Credit Suisse on Real Data #2

- We extracted **object names (tables and views)** from successfully parsed SQLs.
- Measured **matching objects** between audit log entries and schema snapshots.

Matching accessed tables in audit log entries



Matching objects from views to schema



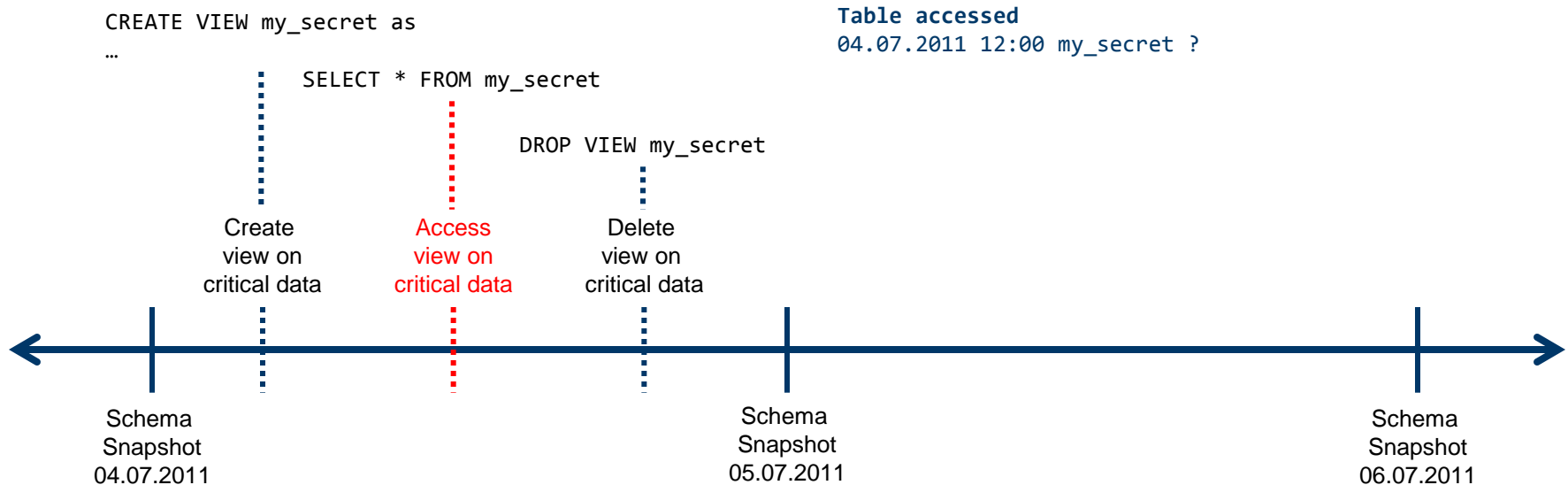
- 99.66% and 96.5% success rates, respectively.

- Limitations:

- Newly created **objects** not present in the schema cannot be matched.
- Neither can be table functions and DB-links.

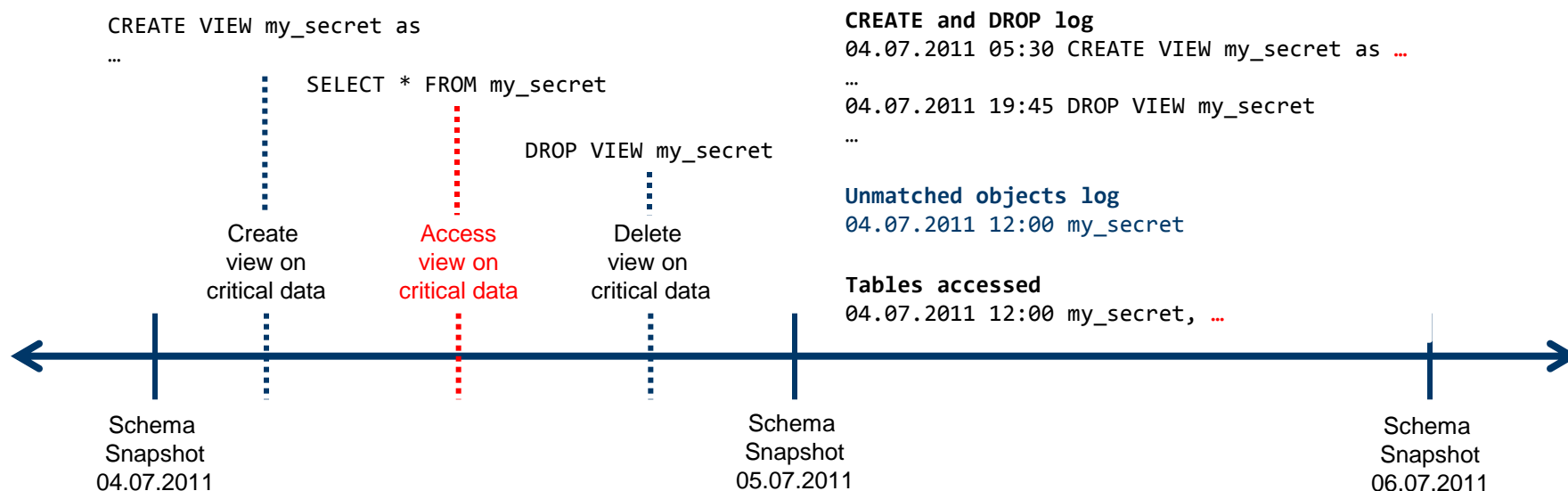
# Dealing with Intraday Schema Modification: The Problem

Assume we have a vicious person that does not want anybody to know that he/she has accessed some critical data.



# Dealing with Intraday Schema Modification: The Counter Measure

- Log all create and drop statements in a separate file.
- Output all tables and views that could not be matched with the schema.
- For all these tables check whether they can be matched to some create or drop statements and resolve accordingly.



# Conclusions

- Prototype implementation was performed on **parts of the production query logs** of the Credit Suisse data warehouse.
- **Xadoop-based analysis with above 95% accuracy and linear scalability:**
  - Several different groups within Credit Suisse in Zurich and New York provided excellent feedback and are ready for further collaboration.
- **Next steps:**
  - **Perform large-scale query log analysis** against the warehouses covering data volumes of several months.
  - Implement more **advanced use cases** with several new stakeholders in Credit Suisse based on machine learning techniques.