

Introduction into Database Forensics – Issues to be solved

Antje Raab-Düsterhöft

University of Applied Science Wismar, Technology,
Business and Design
Dep. of Computer Science

antje.duesterhoeft@hs-wismar.de
<https://it-forensik.fiw.hs-wismar.de/>





Database Forensics...

- subfield of digital forensics
- deals with forensic examination of databases

Databases....

- structured collections of data
- allows efficient storage, retrieval and manipulation of information
- controlled by database management systems (DBMS)



Why Databases are critical assets?

- Databases contains important information
- Database servers now store a greater volume of sensitive data than ever before
- Database security breaches are everywhere and also used in the context of various attacks
 - Use of credit cards
 - Use of medical personal data
 - Use of industrial data
 - ...
- Insider attacks: persons can manipulate data for their own purposes

→ Databases are “interesting sources”



The Road to Database Forensics...

1. Goals
2. Preparation of database forensic investigation
 - Scenarios
 - Kinds of DBMS & Database models
3. Incident verification & Artefact collection
4. Artifact analysis
 - Overview of database artifacts
 - Physical and logical database artifacts
 - Tools
5. Conclusion – Issues to be solved

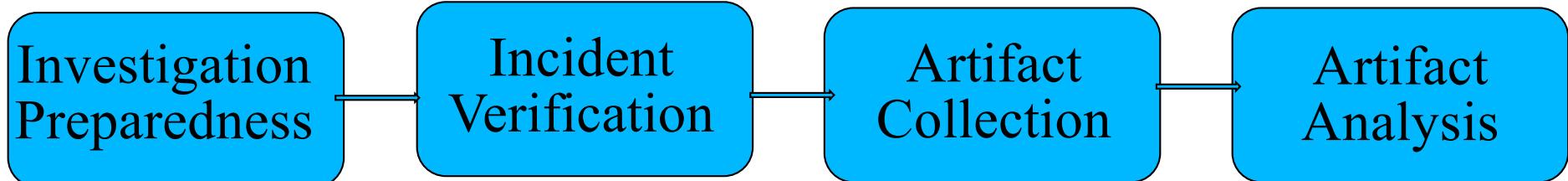
strategy
↓
focus



1. Goals

- Prove or disprove an occurrence of a data security breach
- Determine the scope of a database intrusion
- Retrace user operations (SQL operations)
- Identify data pre- and post-transactions
- Recover previously manipulated or deleted data
- ...

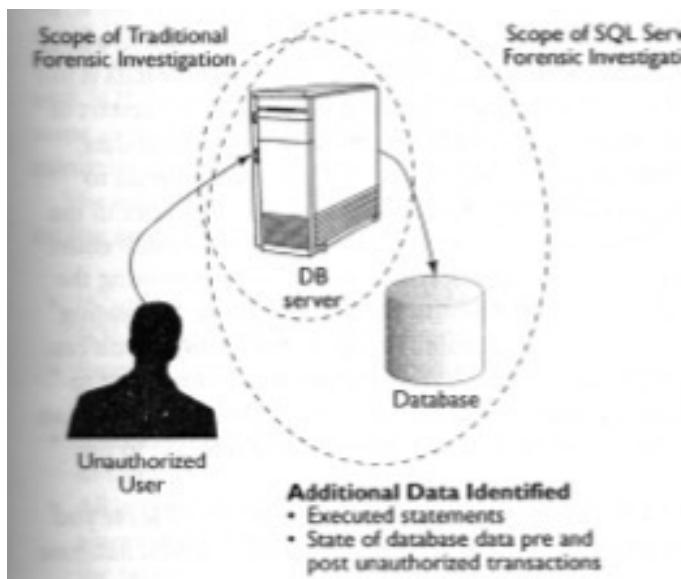
Kevvie Fowler: SQLServer Forensic Analysis, p.49



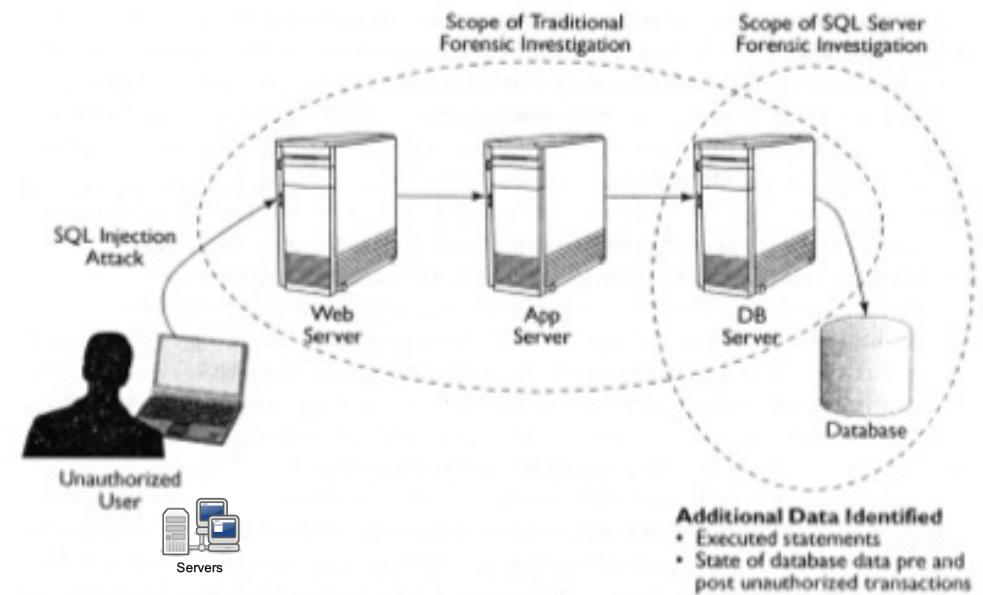
Kevvie Fowler: SQLServer Forensic Analysis, p.60



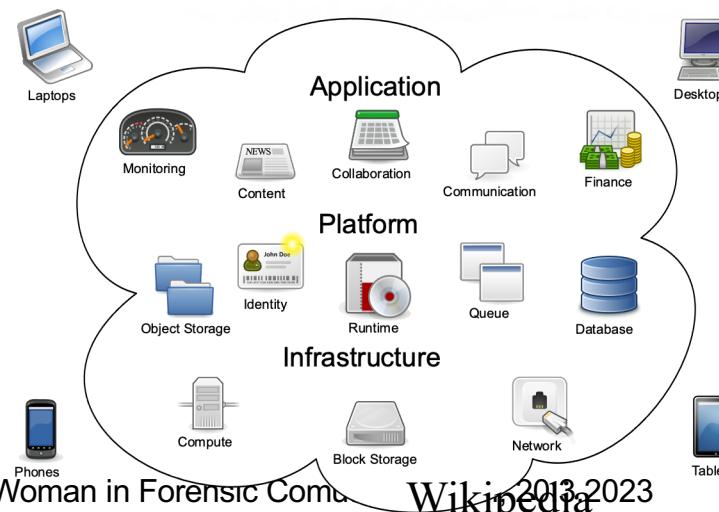
2. Preparation: Scenario 1

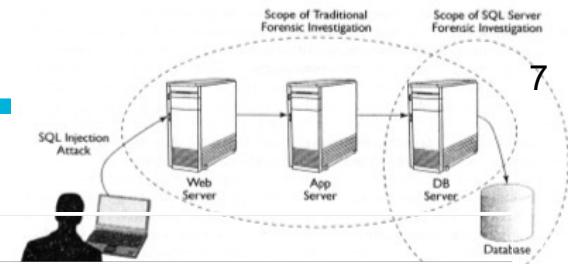


Kevvie Fowler: SQLServer Forensic Analysis, p.53
Scenario 2

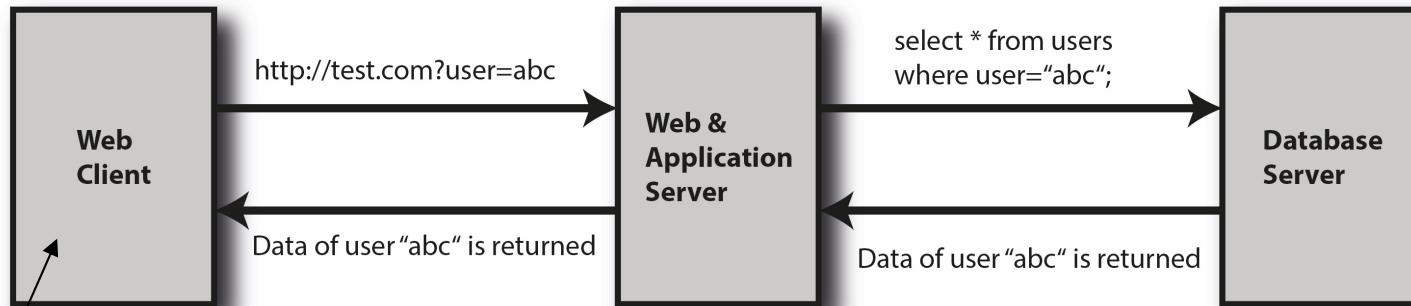


Scenario 3 *Cloud system (new)*

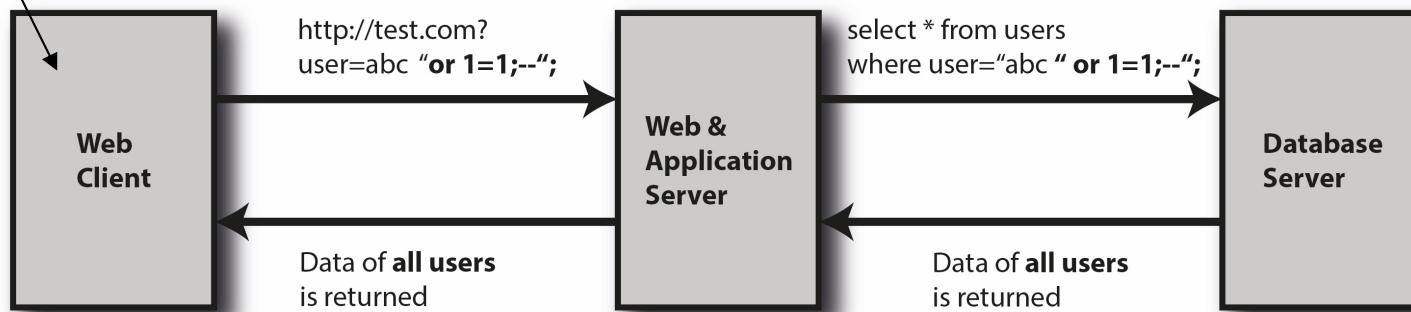




Scenario 2: Example SQL Injection



Typical Scenario of Webapplication and database Communication



SQL Injection Example

- Application Logs
- Webserver-Log
- Error Log
- Page-File
- Memory



2. Preparation: Database Management Systems (RDMS)

The screenshot shows the DB-Engines website with the URL <https://db-engines.com/en/ranking>. The page displays the 'Relational DBMS' ranking for March 2023. The table includes columns for Rank, Mar 2023, Feb 2023, Mar 2022, DBMS, Database Model, and Score. The top 21 entries are:

Rank	Mar 2023	Feb 2023	Mar 2022	DBMS	Database Model	Score
1.	1.	1.	1.	Oracle	Relational, Multi-model	1261.29 +13.77
2.	2.	2.	2.	MySQL	Relational, Multi-model	1182.79 -12.66
3.	3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	922.01 -7.08
4.	4.	4.	4.	PostgreSQL	Relational, Multi-model	613.83 -2.67
5.	5.	5.	5.	MongoDB	Document, Multi-model	458.78 +6.02
6.	6.	6.	6.	Redis	Key-value, Multi-model	172.45 -1.39
7.	7.	7.	7.	IBM Db2	Relational, Multi-model	142.92 -0.04
8.	8.	8.	8.	Elasticsearch	Search engine, Multi-model	139.07 +0.47
9.	9.	↑ 10.	10.	SQLite	Relational	133.82 +1.15
10.	10.	↓ 9.	9.	Microsoft Access	Relational	132.06 +1.03
11.	↑ 12.	↑ 14.	14.	Snowflake	Relational	114.40 -1.26
12.	↓ 11.	↓ 11.	11.	Cassandra	Wide column	113.79 -2.43
13.	13.	↓ 12.	12.	MariaDB	Relational, Multi-model	96.84 +0.03
14.	14.	↓ 13.	13.	Splunk	Search engine	87.97 +0.89
15.	15.	↑ 16.	16.	Amazon DynamoDB	Multi-model	80.77 +1.08
16.	16.	↓ 15.	15.	Microsoft Azure SQL Database	Relational, Multi-model	77.44 -1.31
17.	17.	17.	17.	Hive	Relational	70.91 -1.21
18.	18.	18.	Teradata		Relational, Multi-model	63.74 +0.71
19.	19.		Databricks		Multi-model	60.86 +0.52
20.	20.	↓ 19.	Neo4j		Graph	53.51 -1.92
21.	↑ 22.	↑ 24.	Google BigQuery		Relational	53.44 +0.99



2. Preparation: Types of database models

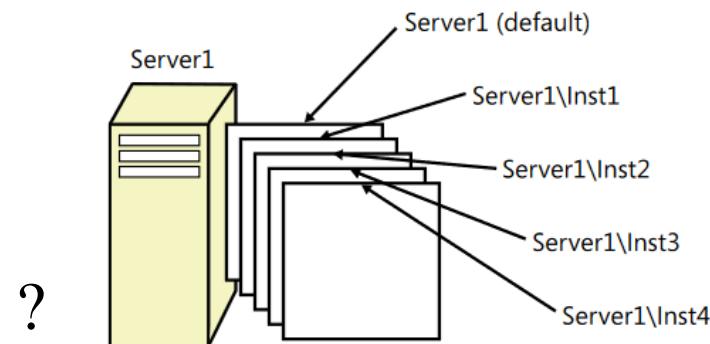
	Relational	NoSQL	Others (deductive, object oriented, ...)
Data model	Structured (table)	Free/ different	Free/ different
Audit/ Log Mechanismen	many (some standardized, some different)	some/ few	?
Storage	storage management, Some differences	file-based	?
Standard	SQL	no/ own	?

focus



2. Preparation: Strategy

1. Defining or identifying the scope/ infrastructure
2. Identifying DBMS, version, configuration, instances, ...
3. Classification of the database model



Kevvie Fowler: SQLServer Forensic Analysis, Appendix B, p. 439ff



2. Preparation: Strategy

4. Configure a forensic workstation for the database system

- Use forensic software
- Use/ create/ integrate forensic scripts
 - for live acquisition/ automated live forensic
 - for memory acquisition (volatile data)
 - (first) artifact collection



2. Preparation: Database Artifacts

Database artifacts refer to the objects that are created in a database system as part of structure and functionality.

Physical artifacts:

- the actual files that are created on the storage medium when a database is created
- data files, log files, backup files, system information, ...

Logical artifacts:

- the database objects that are created and managed within the database
- Scheme, tables, views, indexes, stored procedures, triggers, constraints,



2. Preparation: Forensic database scripts

- for live acquisition/ automated live forensics
- for memory acquisition (volatile data)
 - Caches (data, plan, ...)
 - Databases
 - SQL statements
 - Connections
 - Transaction log
 - Users
 - ...
 - DBObjects
 - Configurations
 - Jobs
 - TimeConfigs
 - Triggers
 - Sessions
 - ...

e.g.

SQL Server: `SELECT * FROM sys.configurations`

SQL Server: `SELECT * FROM sys.dm_exec_cached_plans cross apply sys.dm_exec_sql_text(plan_handle)`

Postgres: `SELECT current_user`

Oracle: `SELECT owner,table_name,column_name,data_type
FROM dba_tab_columns
WHERE data_type='DATE' or data_type like 'TIMESTAMP%';`



2. Preparation: Forensic database scripts

The screenshot displays four windows from the pgAdmin 4 interface:

- Left Window:** Shows the "Servers" tree with three servers: PostgreSQL 10, PostgreSQL 11, and PostgreSQL 12. PostgreSQL 12 is expanded to show "Databases" (postgres), "Casts", "Catalogs", "Event Triggers", "Extensions", "Foreign Data Wrappers", "Languages", "Publications", and "Schemas" (fahrrad, fussball, geheimdienst).
- Middle Left Window:** A "Data Output" tab showing the result of the query `SHOW DATA_DIRECTORY`. It shows one row: `1 /Library/PostgreSQL/11/data`.
- Middle Right Window:** A "Browser" window showing the structure of the "public" schema. It lists "geheimdienst", "public" (expanded to show Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (16) including Kunden, assistenten, bestellungen, hoeren, and karten), and "geheimdienst".
- Bottom Window:** A "Query Editor" window containing the query `SELECT CURRENT_USER`. The result shows the user name `postgres`.

Bottom Left Window: A terminal window titled "Eingabeaufforderung" showing the command `sqlcmd -L` and its output, which lists the local server as "CBDESKTOP\SQLEXPRESS".

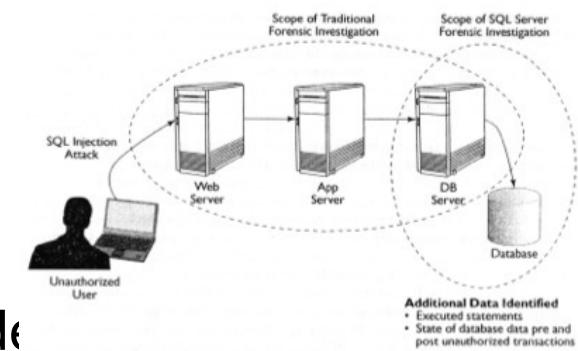
Bottom Right Window: A "Nach Servern suchen" (Search for servers) dialog box showing a list of available servers under "Datenbankmodul": CBDESKTOP\SQLEXPRESS, CBDESKTOP, Analysis Services, Reporting Services, and Integration Services.



3. Incident Verification + Artifact Collection

Identifying first signs of breaches

- Server problems, server traffic
- Network traffic
- An (unauthorized) modification of data
- Compare checksums of database source code
- Confirm the breach/ incident
- Made a decision about shut-down/ separate the database server



First collection of forensic data (forensically usable)

- Collecting volatile and non-volatile artifacts
- Bit-to-bit image of the database data, query files, query output,..
- Collecting data from the infrastructure



3. Incident Verification + Artifact Collection

General rules for collecting database artifacts:

1. Do not use system libraries, they might be compromised
2. Do not execute DDL/DML statements to avoid changes to data
3. Do not create SQL Server logins, database users or objects
4. Restrict the use of temporary tables and variables in order not to overwrite existing information in the SQL Server buffer
5. Do not modify existing corrections or database objects
6. Do not execute untested statements, errors will be logged and may change the event log



3. Artifact Collection: Starting point

Results of collecting database artifacts

Decision of analysing methods:

1. Actual database
2. Images, that can be used for starting the DBMS with databases
3. Parts of the databases (data), that can be integrated into a virtual environment with the according DBMS



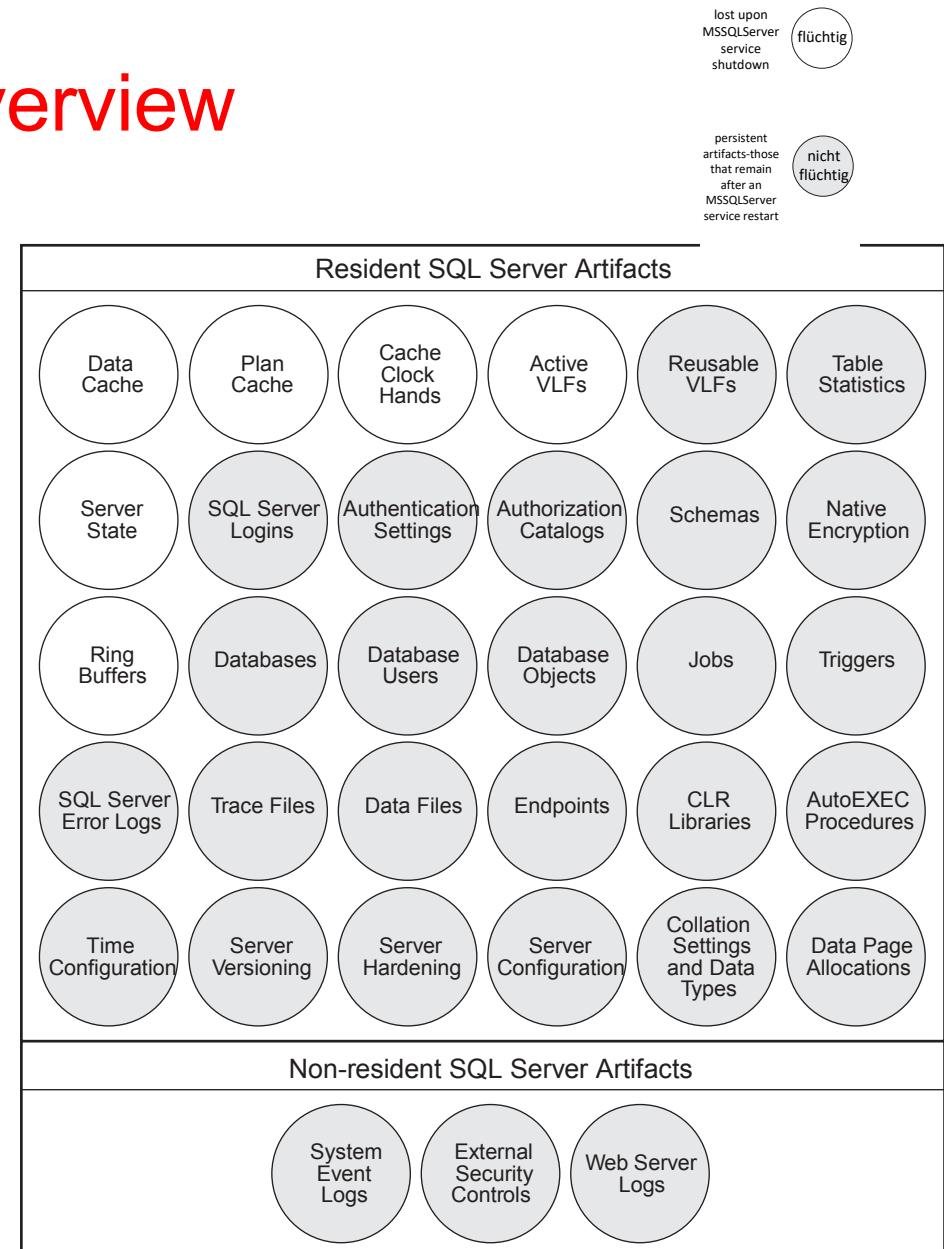
Further data from the short artifact analysis

- Incident SQL queries, that were found
- Data, that was queried or modified
- Application Logs/ Web Server Log with IP Addresses (connections to the servers)



3. Artifact Collection: Overview

- DBMS specific
- Artifacts are
 - Logical
 - Physical
- Volatile
- Non-volatile
- Resident
- Non-resident





4. Artifact Analysis: Logical Artifacts

Starting point: e.g. Incident SQL queries in Web Server Log

First: Get an overview of the logical artifacts

- Which database is involved?
- Which data are stored in this database?
- Which users were active?
- Which rights does a special user have?
- Which queries were executed?
- ...

→ SQL-Queries for analysis



4. Artifact Analysis: Database structure

Information Schema:
All information about the database

General Catalogs:
Information about the dbms

The screenshot shows the pgAdmin 4 interface. The left pane, titled 'Browser', displays a tree view of database objects under 'public.Kunden...'. The 'tables' node is expanded, showing 12 columns: commit_action, is_insertable_into, is_typed, reference_generation, self_referencing_column_name, table_catalog, table_name, table_schema, table_type, user_defined_type_catalog, user_defined_type_name, and user_defined_type_schema. The right pane, titled 'Data Output', shows a table with 10 rows of data from the 'information_schema.tables' catalog. The columns are table_catalog, table_schema, table_name, and table_type. The data includes various tables like 'professoren', 'assistenten', etc., under the 'postgres' catalog and 'public' schema.

table_catalog	table_schema	table_name	table_type
postgres	public	professoren	BA
postgres	public	assistenten	BA
postgres	public	vorlesungen	BA
postgres	public	studenten	BA
postgres	public	hoeren	BA
postgres	public	voraussetzen	BA
postgres	public	pruefen	BA
postgres	geheimdienst	agenten	BA
postgres	geheimdienst	decknamen	BA
postgres	geheimdienst	einsaetze	BA

→ Scope: what the attacker might have done



4. Artifact Analysis: Database functions, user, logs, trigger, ...

The screenshot shows the pgAdmin 4 interface. The left pane, titled 'Browser', displays a tree view of database objects under 'public.Kunden/'. The right pane, titled 'Query Editor', contains a table named 'information_schema.tables/postgres' with 10 rows. Below the table is a query window with the following SQL:

```
1 SELECT * FROM information_schema.tables
2
```

The screenshot shows the pgAdmin 4 interface. The left pane, titled 'Browser', has 'Tables (16)' selected. The right pane, titled 'Data Output', shows the result of the query 'SELECT CURRENT_USER'. The result table has one row with the value 'postgres'.

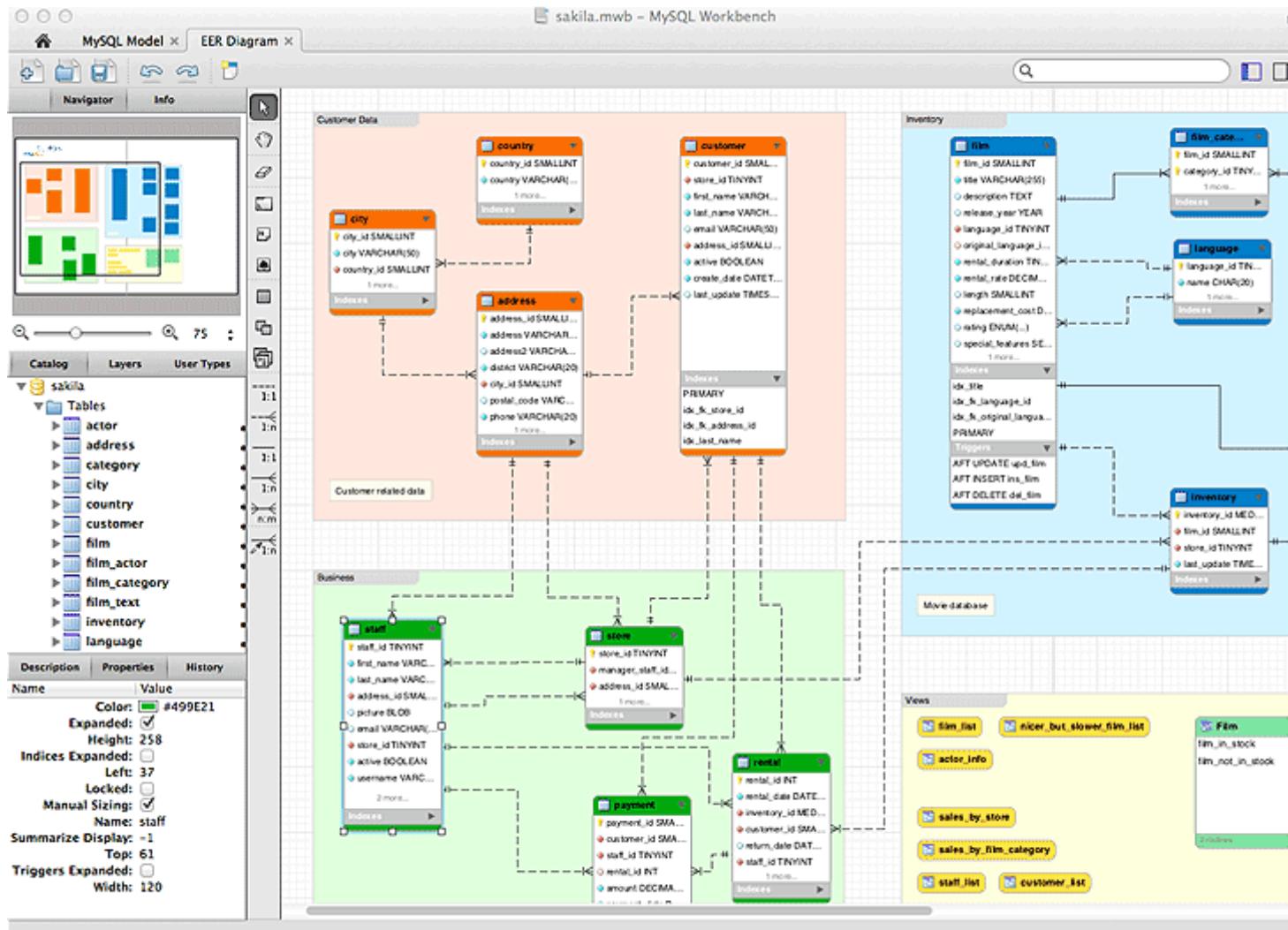
name
postgres

The screenshot shows the pgAdmin 4 interface. The left pane, titled 'Browser', has 'Schemas (1)' selected. The right pane, titled 'Data Output', shows the result of the query 'show log_destination'. The result table has one row with the value 'stderr'.

log_destination
stderr



4. Artifact Analysis: Visualization of the db structure





4. Artifact Analysis: Query plan (Postgres)

The screenshot shows the pgAdmin 4 interface. On the left, the Browser pane displays a tree view of databases, tables, and objects. The 'public' schema under 'PostgreSQL 12' is currently selected. On the right, the main window shows the results of an 'Explain Analyze' command. The 'QUERY PLAN' tab displays the execution plan with 13 numbered steps, including Hash Joins, Hash Cond, Seq Scans, and Hash operations. The 'Text' tab shows the raw SQL query. Below the plan, the 'Planning Time' and 'Execution Time' are listed. The 'Query Editor' tab at the bottom contains the original SQL code.

```
1 explain analyze select s.Name, v.Titel
2 from Studenten s, hoeren h, Vorlesungen v
3 where s. MatrNr = h. MatrNr and
4       h.VorlNr = v.VorlNr;
```



4. Artifact Analysis: Physical Artifacts

Logs: must be enabled before an incident (!)

- SQL Logs
- Error Logs
- Transaction logs/ Write Ahead Logs (WAL)
- Special Query logs

Transaction logs (used for roll-back and recover databases)...

- Specific for DBMS
- Detailed information about SQL operation and data
 - Insert/ update/ delete data, create structure/ functions, user, ...
- Interpretation tools only partially available



MS SQL Server: Transaction Log

Inside the transaction log:

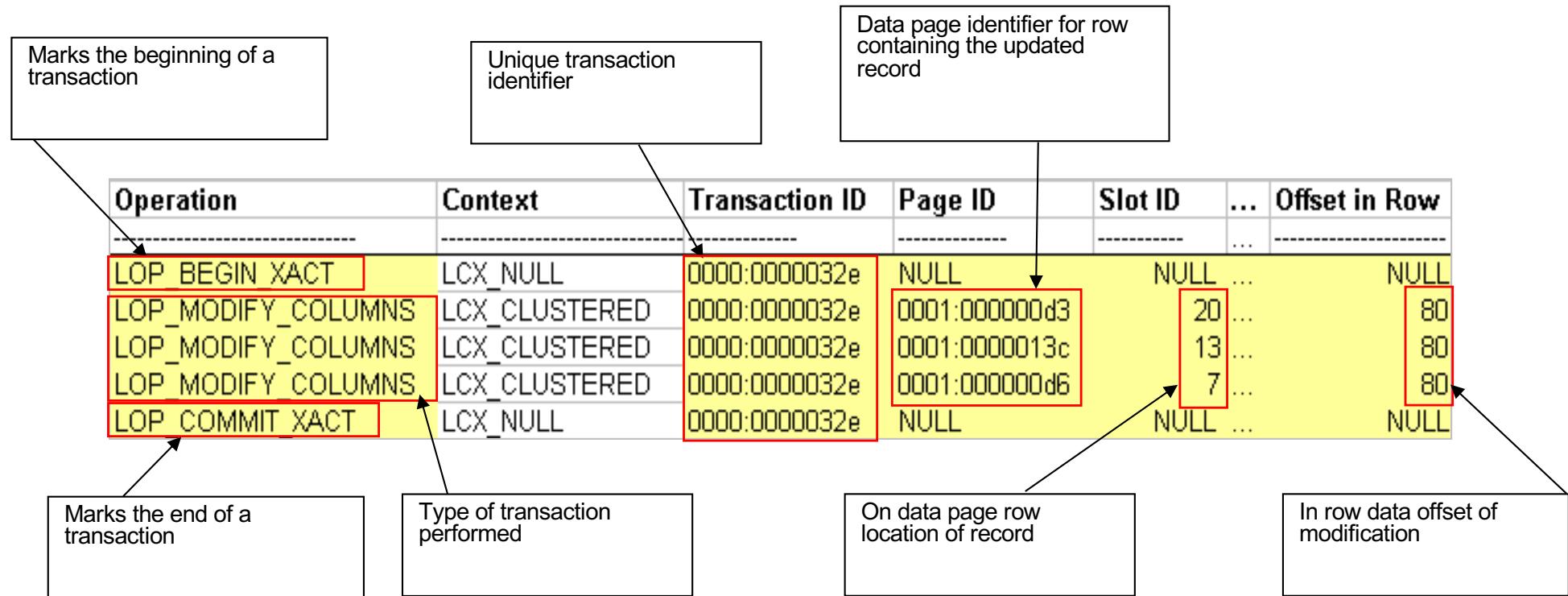
1. CurrentLSN
2. Operation
3. Context
4. Transaction ID
5. Tag Bits
6. Log Record Fixed Length
7. Log Record Length
8. PreviousLSN
9. Flag Bits
10. Log Reserve
11. AllocUnitID
12. AllocUnitName
13. Page ID
14. Slot ID
15. Previous Page LSN
16. PartitionID
17. RowFlags
18. Num Elements
19. Offset in Row
20. Checkpoint Begin
21. CHKPT Begin DB Version
22. MaxXDESID
23. Num Transactions
24. Checkpoint End
25. CHKPT End DB Version
26. Minimum LSN
27. Dirty Pages
28. Oldest Replicated Begin LSN
29. Next Replicated End LSN
30. Last Distributed End LSN
31. Server UID
32. UID
33. SPID
34. BeginLog Status
35. Xact Type
36. Begin Time
37. Transaction Name
38. Transaction SID
39. End Time
40. Transaction Begin
41. Replicated Records
42. Oldest Active LSN
43. Server Name
44. Database Name
45. Mark Name
46. Master XDESID
47. Master DBID
48. PrepLogBegin LSN
49. PrepareTime
50. Virtual Clock
51. Previous Point

Kevvie Fowler: SQLServer Forensic Analysis

50. Savepoint Name
51. Rowbits First Bit
52. Rowbits Bit Count
53. Rowbits Bit Value
54. Number of Locks
55. Lock Information
56. LSN Before Writes
57. Pages Written
58. Data Pages Delta
59. Reserved Pages Delta
60. Used Pages Delta
61. Data Rows Delta
62. Command Type
63. Publication ID
64. Article ID
65. Partial Status
66. Command
67. Byte Offset
68. New Value
69. Old Value
70. New Split Page
71. Rows Deleted
72. Bytes Freed
73. CI Table ID
74. CI Index ID
75. Filegroup ID
76. Meta Status
77. File Status
78. File ID
79. Physical Name
80. Logical Name
81. Format LSN
82. RowsetID
83. TextPtr
84. Column Offset
85. Flags
86. Text Size
87. Offset
88. Old Size
89. New Size
90. Description
91. Bulk allocated extent count
92. Bulk rowinsertID
93. Bulk allocationunitID
94. Bulk allocation first IAM Page ID
95. Bulk allocated extent ids
96. RowLog Contents 0
97. RowLog Contents 1
98. RowLog Contents 2
99. RowLog Contents 3
100. Rowlog Contents 4
101. Log Record



MS SQL Server: Transaction Log



Identifier	Hex	Decimal
Transaction ID	0000:00000032e	0:814
Data Page	0001:000000d3	1:211



MS SQL Server: Transaction Log –Recovering SQL is possible

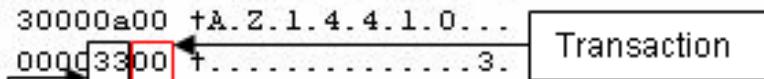
```
Slot 20 Offset 0x147f Length 237

Record Type = PRIMARY_RECORD          Record Attributes = NULL_BITMAP VARIABLE_COLUMNS

Memory Dump @0x2F3AD47F

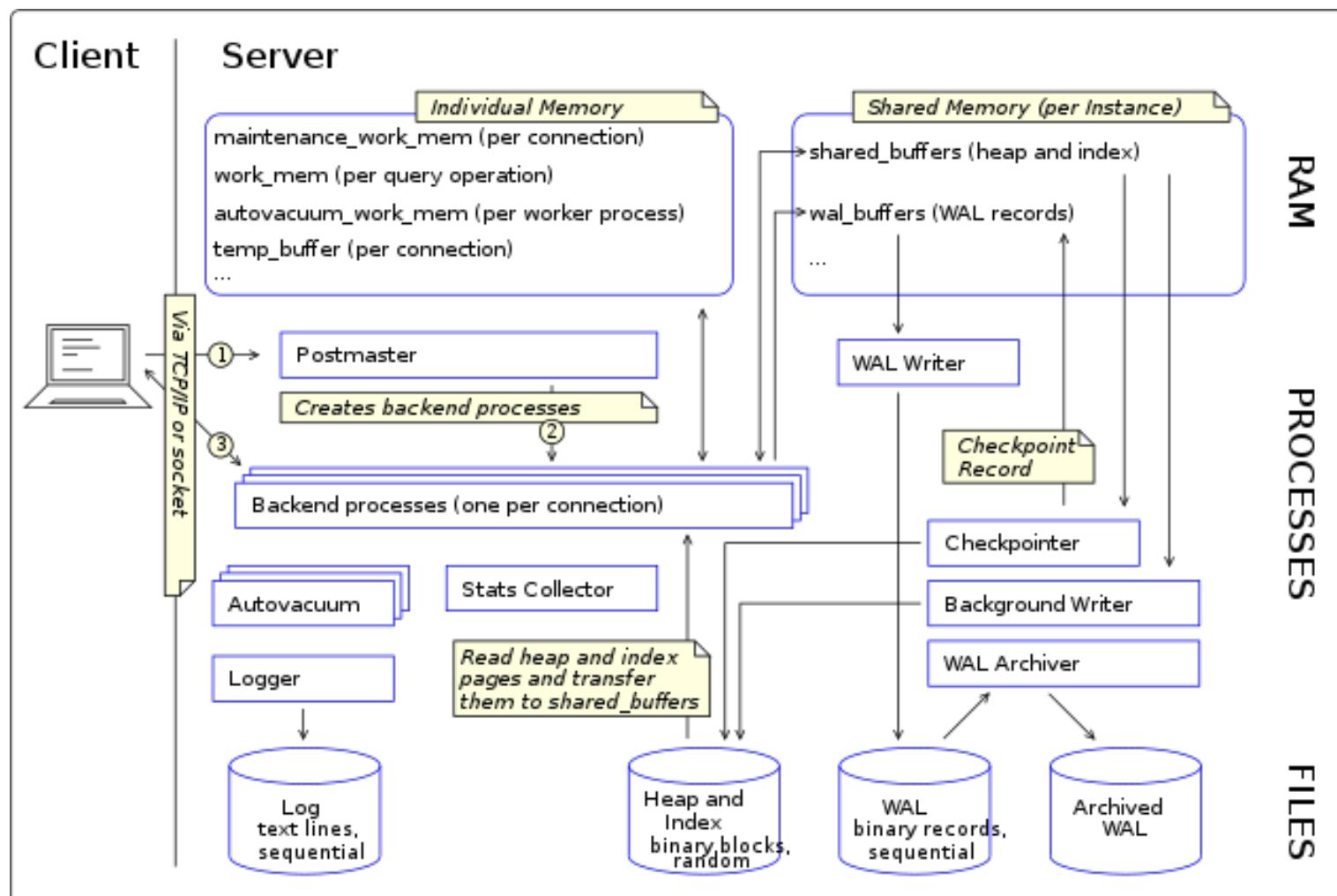
00000000: 30006c00 6f000000 53007000 72006900 +0.1.o...S.p.r.i.
00000010: 6e006700 4c006100 6b006500 20002000 tn.g.L.a.k.e. .
00000020: 20002000 20002000 20002000 20002000 t . . . . .
00000030: 41005a00 31003400 34003100 30000a00 tA.Z.1.4.4.1.0...
00000040: 00000100 00000000 0000e498 00003300 t.....3. Transaction
00000050: 2e003500 30002000 20002000 20002000 t..5.0. . . .
00000060: 20002000 20002000 20002000 0e0000c0 t . . . . .
00000070: 06008400 88009900 9d00ad00 ed00416e t.....An
00000080: 6f736f6e 456d696c 37322053 74617266 tosonEmil72 Starf
00000090: 656c6c20 44726976 65566973 61343931 tell DriveVisa491
000000A0: 36383833 38343033 38323330 3056006f t6883840382300V.o
000000B0: 006c0063 0061006e 006f0020 00360032 t.l.c.a.n.o. .6.2
000000C0: 00200069 006e0063 00680020 0050006c t. .i.n.c.h. .P.1
000000D0: 00610073 006d0061 00200054 00560020 t.a.s.m.a. .T.V.
000000E0: 00560043 00320033 00330032 00ttttttt.V.C.2.3.3.2.
```

Start of column



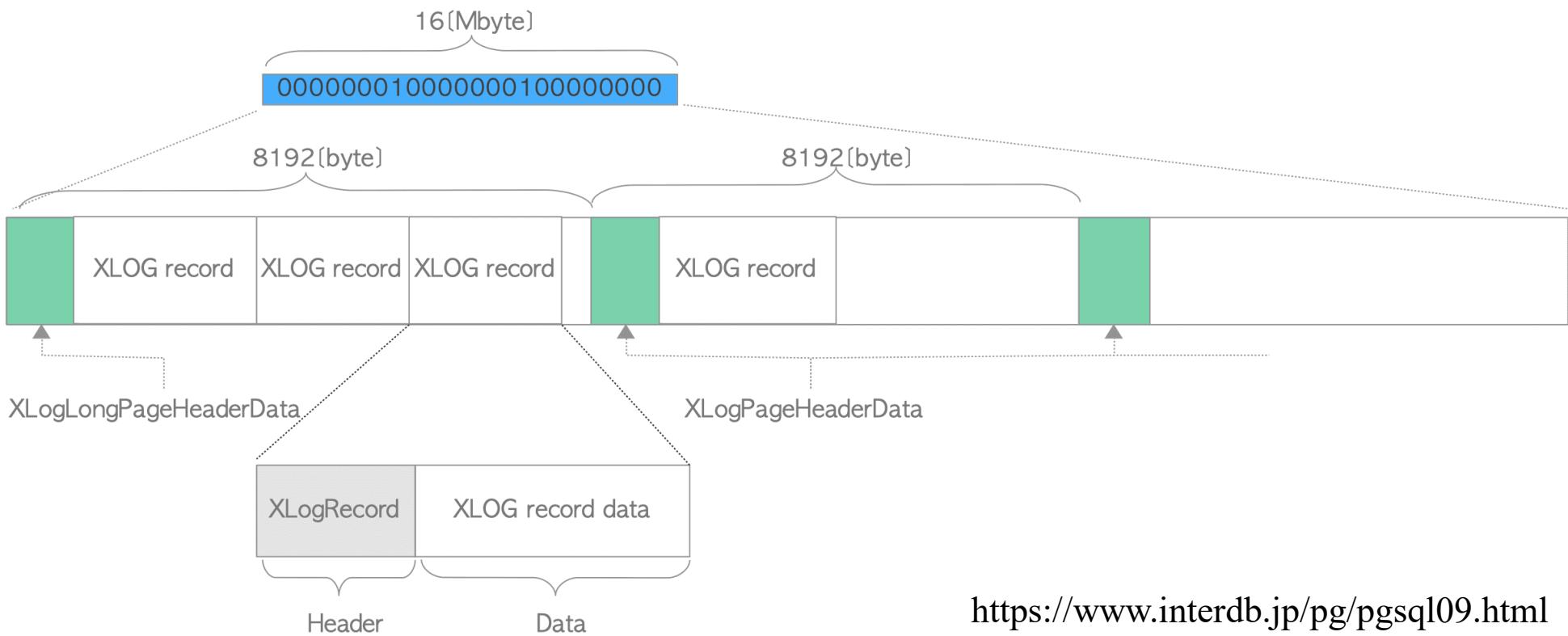


Postgres internal structure





Postgres - Internal layout of a WAL segment file



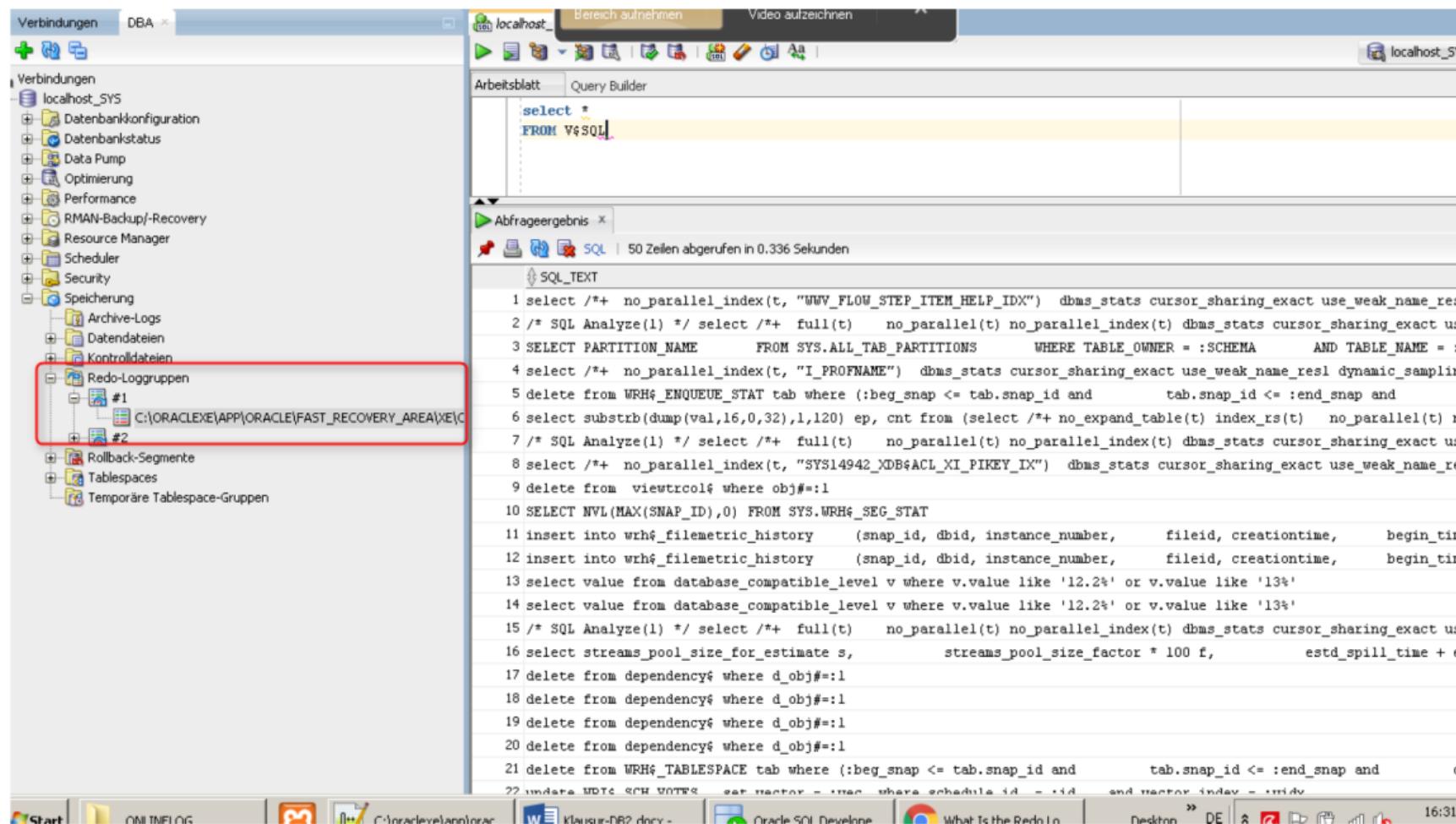
<https://www.interdb.jp/pg/pgsql109.html>

```
typedef struct XLogRecord
{
    uint32          xl_tot_len;    /* total len of entire record */
    TransactionId  xl_xid;        /* xact id */
    uint32          xl_len;         /* total len of rmgr data */
    uint8           xl_info;       /* flag bits, see below */
    RmgrId          xl_rmid;      /* resource manager for this record */
    /* 2 bytes of padding here, initialize to zero */
    XLogRecPtr      xl_prev;       /* ptr to previous record in log */
    pg_crc32        xl_crc;        /* CRC for this record */
} XLogRecord;
```



4. Artifact Analysis: Tools for the interpretation of logs

Oracle: LOG File from the LOG-MINER



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Verbindungen' sidebar is open, showing a tree view of database connections. A red box highlights the 'Redo-Loggruppen' node under the 'localhost_SYS' connection, which contains entries for redo log groups #1 and #2, both pointing to the path 'C:\ORACLE\APP\ORACLE\FAST_RECOVERY_AREA\XE'. The main workspace shows a query builder window with the following SQL command:

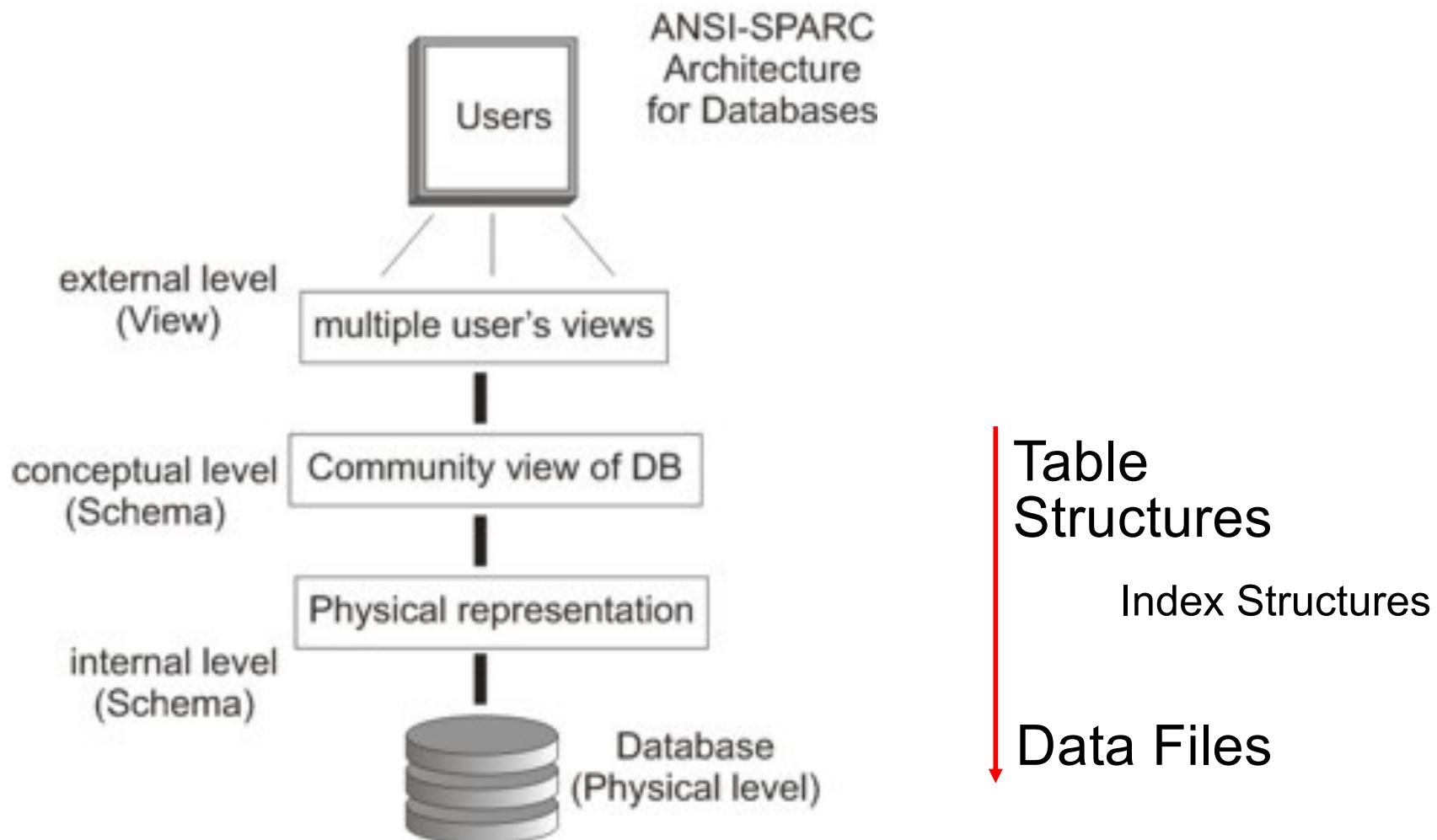
```
select *  
FROM V$SQL
```

Below it, the 'Abfrageergebnis' (Query Results) window displays the output of the query, showing 50 rows returned in 0.336 seconds. The results are presented in a table with columns for 'SQL_TEXT' and 'Rows'.

SQL_TEXT	Rows
1 select /*+ no_parallel_index(t, "WWW_FLOW_STEP_ITEM_HELP_IDX") */ dbms_stats cursor_sharing_exact use_weak_name_res	50
2 /* SQL Analyze(1) */ select /*+ full(t) */ no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_exact us	
3 SELECT PARTITION_NAME FROM SYS.ALL_TAB_PARTITIONS WHERE TABLE_OWNER = :SCHEMA AND TABLE_NAME = :	
4 select /*+ no_parallel_index(t, "I_PROFNAME") */ dbms_stats cursor_sharing_exact use_weak_name_resl dynamic_sampli	
5 delete from WRH\$_ENQUEUE_STAT tab where (:beg_snap <= tab.snap_id and tab.snap_id <= :end_snap and	
6 select substrb(dump(val,16,0,32),1,120) ep, cnt from (select /*+ no_expand_table(t) index_rs(t) */ no_parallel(t) n	
7 /* SQL Analyze(1) */ select /*+ full(t) */ no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_exact us	
8 select /*+ no_parallel_index(t, "SYS14942_XDB\$ACL_XI_PIKEY_IX") */ dbms_stats cursor_sharing_exact use_weak_name_re	
9 delete from vewtrol\$ where obj#:1	
10 SELECT NVL(MAX(SNAP_ID),0) FROM SYS.WRH\$_SEG_STAT	
11 insert into wrh\$_filemetric_history (snap_id, dbid, instance_number, fileid, creationtime, begin_time)	
12 insert into wrh\$_filemetric_history (snap_id, dbid, instance_number, fileid, creationtime, begin_time)	
13 select value from database_compatible_level v where v.value like '12.2%' or v.value like '13%	
14 select value from database_compatible_level v where v.value like '12.2%' or v.value like '13%	
15 /* SQL Analyze(1) */ select /*+ full(t) */ no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_exact us	
16 select streams_pool_size_for_estimate s, streams_pool_size_factor * 100 f, estd_spill_time + e	
17 delete from dependency\$ where d_obj#:1	
18 delete from dependency\$ where d_obj#:1	
19 delete from dependency\$ where d_obj#:1	
20 delete from dependency\$ where d_obj#:1	
21 delete from WRH\$_TABLESPACE tab where (:beg_snap <= tab.snap_id and tab.snap_id <= :end_snap and d	
22 update NOTE SCM_VNOTE set nsector = tsec where schedule_id = :id and nsector_inday = :tiny	



4. Artefact Analysis: Physical Artifacts - Data



The ANSI-SPARC three-level architecture (Wikipedia)



4. Artefact Analysis: Data files (Postgres)

Table Structures

Index structures

Data Files

Postgres:

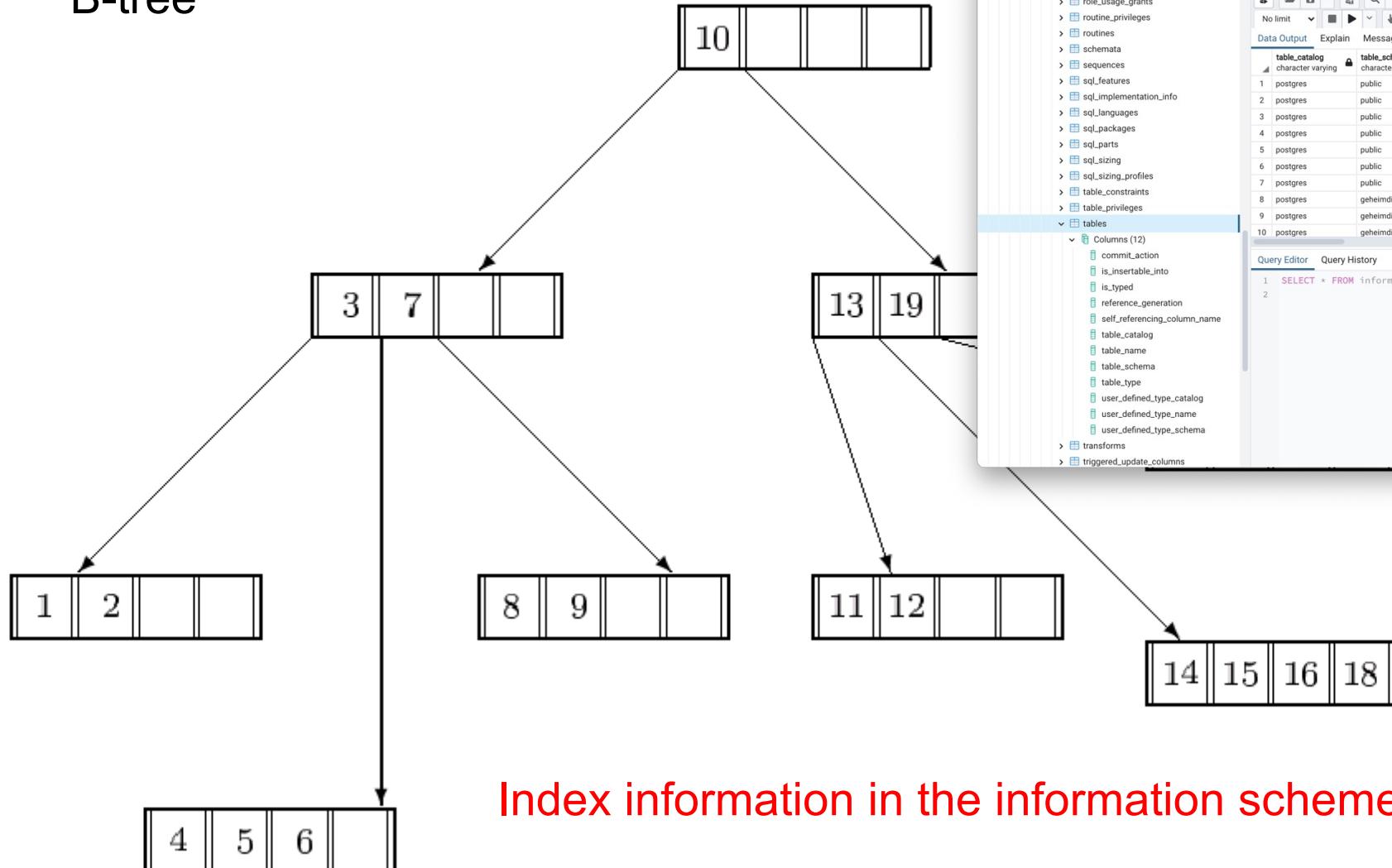
- Data directory: **/var/lib/pgsql/data**
- Each PSQL database table is a separate file with a default file extension of **.mfd**.
- Different index structures:
B-tree, Hash, GiST, SP-GiST, GIN, BRIN, the extension bloom

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays the database structure under 'PostgreSQL 12': Servers (3), Databases (1) containing 'postgres' (Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (4)), and Schemas (4) containing 'fahrrad', 'fussball', and 'geheimdienst'. On the right, the 'Data Output' tab shows the result of the query 'SHOW DATA_DIRECTORY' with the value '/Library/PostgreSQL/11/data'. Below it, the 'Query Editor' shows the same query.



4. Artefact Analysis: Data files (Postgres)

B-tree



A screenshot of the pgAdmin 4 interface. The left sidebar shows a tree view of database objects under 'Browser'. The 'tables' node under 'information_schema' is currently selected. The main pane displays a table with the following data:

table_catalog	table_schema	table_name	character varying
1	postgres	professoren	BA
2	postgres	assistenten	BA
3	postgres	vorlesungen	BA
4	postgres	studenten	BA
5	postgres	hoeren	BA
6	postgres	voraussetzen	BA
7	postgres	pruefen	BA
8	postgres	geheimdienst	agenten
9	postgres	geheimdienst	decknamen
10	postgres	geheimdienst	einsatze

The 'Query Editor' tab at the bottom contains the query: `1 SELECT * FROM information_schema.tables`.



4. Artifact Analysis: Query plan (Postgres)

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays a tree view of database objects. Under 'PostgreSQL 12' > 'Databases' > 'public', the 'Aggregates' node is selected. The main pane shows the 'Explain' tab of the 'QUERY PLAN' section, which displays the execution plan for the query. The plan includes various stages such as Hash Join, Hash Cond, Seq Scan, and Hash, along with their costs, row counts, and actual execution times. A red annotation 'search indexes' is placed over the plan. Below the plan, the 'Query Editor' tab contains the SQL code:

```
1 explain analyze select s.Name, v.Titel
2 from Studenten s, hoeren h, Vorlesungen v
3 where s. MatrNr = h. MatrNr and
4       h.VorlNr = v.VorlNr;
```



Conclusion: Strategy for a forensic database analysis

1. Preparation

- Defining or identifying the scope/ infrastructure
- Identifying DBMS, version, configuration, instances, ...
- Classification of the database model
- Preparing forensic scripts and a forensic machine

2. Incident verification

- First data collection from the database
- Using infrastructure information
- Confirm the breach or incident
- Decision about shut-down/ operating the database server



Conclusion: Strategy for a forensic database analysis

3. Artifact collection

- Collecting volatile and non-volatile database artifacts
- Bit-to-bit image of the database data, query files, query output,..
- Collecting data from the infrastructure

4. Artifact analysis

- Decision of the analysis methods
- Overview of the database use logical artifacts
- Analyse logical artifacts
- Analyse physical artifacts

5. Reporting

Antje Raab-Düsterhöft, DFRWS-EU, "Woman in Forensic Computing", Bonn, 20.3.2023



Conclusion: Issues to be solved

1. Different Database systems
 - Relational:
 - Different versions, configurations, ...
 - Frequent updates
 - Different internals and file structures
 - NoSQL (and others):
 - No/ few forensic strategies in general
 - Very different structure of the data systems
 - No/ few management functions
 - (probably easy) forensic examination of data files
2. Need of tools for data interpretation (logs, structures, internals)



Thank you for your attention!