# 

[DB2 Auditing - Operations](#_bivae1g9eefs)

[Introduction](#_6tb9r4lhblob)

[Operational Steps](#_96ktezhdgla6)

[How Auditing Works](#_n6xvyuvny0ti)

[Discover what is active and what is audited](#_dvxel11979gx)

[Storage and analysis of audit logs](#_2q0ohjpdiyry)

[Archiving active audit logs](#_7fgki0wpvpw6)

[Creating Extract Files](#_i1udt5qaqoxy)

[Creating the database tables for loading the audit extract files](#_1gb1vq5z9g9f)

[Loading the tables From the Extract Files](#_aa9oufbboqs6)

[Analyze audit data:](#_c4cugbapmvzm)

[Demo](#_q4nnootbe7d9)

# 

# DB2 Auditing - Operations

## Introduction

This is intended to simplify a complex topic into a subset. How to use auditing of a DB2 database effectively when required.

In this case, auditing of a role and the users assigned to the role is tracked. This assumes all the preparation steps are completed.   
  
See this document for details of setup and modification of the audited operations.  
  
<https://docs.google.com/document/d/1bnZrru67WzTjtJtIQcqEYKKB1drbOuCyrVGOoZZ4KJs/edit>

We will break down the steps to extract the auditing data in detail.

### Operational Steps

1. Overview of how this works
2. Discover what is active and what is audited
3. Extract the files to an archive
4. Load the archives to “\*.del” files
5. Load the archives to database tables
6. Reporting

Automation of the process is possible from point #3 through point #6. Thus far it’s not in scope,

### How Auditing Works

You can decide what is audited and who is audited through a number of strategies.

In this case a specific role is audited which is assigned to DBAs and an auditing policy is created.. There are many other options and no actual limit to the number of policies.

Policies can be on the following:

* 1. The entire database
* 2. Tables  
  3. Trusted contexts  
  4. Authorization IDs representing users, groups, or roles  
  Much of this potential is beyond the scope of the current document, please see the URL below for much more detail:

<https://www.ibm.com/docs/en/db2/11.5?topic=facility-audit-policies>

and

<https://www.ibm.com/docs/en/db2/11.5?topic=activities-introduction-db2-audit-facility>

### Discover what is active and what is audited

You can check instance level audit settings with the describe option of the db2audit command.

* Log in to the database server as the instance owner. (“db2cai” in this case)
* Execute the following command.

db2audit describe **(as the instance owner ex: db2cai)**

jq03a010:db2cai 1> db2audit describe

DB2 AUDIT SETTINGS:

Audit active: "TRUE "

Log audit events: "BOTH"

Log checking events: "BOTH"

Log object maintenance events: "BOTH"

Log security maintenance events: "BOTH"

Log system administrator events: "BOTH"

Log validate events: "BOTH"

Log context events: "BOTH"

Return SQLCA on audit error: "TRUE "

Audit Data Path: "/db2/CAI/AUDIT/"

Audit Archive Path: "/db2/CAI/AUDIT/audarchive/"

AUD0000I Operation succeeded.

The settings above indicate that instance level auditing is enabled that audits audit, checking, objmaint, secmaint, sysadmin, and validate success and failures, and also signals that audit context is active.

Also, instance and database level audit logs are output to the /db2/CAI/AUDIT directory and the audit log archives go to the /db2/CAI/AUDIT/audarchive directory.

You can also modify the above configuration:

db2audit configure archivepath '/db2/CAD/AUDIT/audarchive'

### Database level db2audit settings:

Database level audit settings can be found in the SYSCAT.AUDITPOLICIES and SYSCAT.AUDITUSE catalog tables.

* Connect to the target database. (eg: db2 connect to <database\_name>)
* Query the SYSCAT.AUDITPOLICIES and SYSCAT.AUDITUSE tables.

SELECT substr(AUDITPOLICYNAME,1,12) as "Policy ", \

OBJECTTYPE as "Obj Type", \

SUBOBJECTTYPE as "Sub Object Type", \

substr(OBJECTSCHEMA,1, 10) as "Schema", \

substr(OBJECTNAME,1,10) as "Object Name" \

FROM SYSCAT.AUDITUSE;

Policy Obj Type Sub Object Type Schema Object Name

---------------- -------- -------------------- ---------- -----------

ADMINSPOLICY i R - DB2\_SYSADM

Here the instance and database are audited with the ADMINSPOLICY on the role DB2\_SYSADM  
  
See the [Appendix](#_ze8c6uf0b3qj) for [column values for the column OBJECTTYPE in the AUDITUSE table](#_uc92fae1z473)

And for the [column values for the **SUBOBJECTTYPE** column in the **SYSCAT.AUDITUSE** table](#_fc8ifugeh6tf)

We now query the ADMINSPOLICY to see what is audited. DB2 uses a primitive subset of SQL and does not have a PIVOT function. Since it’s easier to read this as a list, the below query simulated this.

For example from the db2 prompt:

SELECT 'AUDITPOLICYNAME' AS "Policy ", CAST(AUDITPOLICYNAME AS VARCHAR(18)) AS "Value" \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY' \

UNION all \

SELECT 'AUDITSTATUS' AS Policy, CAST(AUDITSTATUS AS VARCHAR(18)) AS VALUE \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY' \

UNION all \

SELECT 'CONTEXTSTATUS' as Policy, CAST(CONTEXTSTATUS AS VARCHAR(18)) AS VALUE \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY' \

UNION all \

SELECT 'VALIDATESTATUS' as Policy, CAST(VALIDATESTATUS AS VARCHAR(18)) AS VALUE \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY' \

UNION all \

SELECT 'CHECKINGSTATUS' as Policy, CAST(CHECKINGSTATUS AS VARCHAR(18)) AS VALUE \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY' \

UNION all \

SELECT 'SECMAINTSTATUS' as Policy, CAST(SECMAINTSTATUS AS VARCHAR(18)) AS VALUE \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY' \

UNION all \

SELECT 'OBJMAINTSTATUS' as Policy, CAST(OBJMAINTSTATUS AS VARCHAR(18)) AS VALUE \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY' \

UNION all \

SELECT 'SYSADMINSTATUS' as Policy, CAST(SYSADMINSTATUS AS VARCHAR(18)) AS VALUE \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY' \

UNION all \

SELECT 'EXECUTESTATUS' as Policy, CAST(EXECUTESTATUS AS VARCHAR(18)) AS VALUE \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY' \

UNION all \

SELECT 'EXECUTEWITHDATA' as Policy, CAST(EXECUTEWITHDATA AS VARCHAR(18)) AS VALUE \

FROM SYSCAT.AUDITPOLICIES WHERE AUDITPOLICYNAME = 'ADMINSPOLICY'

You can see that the audit policy ADMINSPOLICY is defined in the database (as opposed to the instance level) also and this policy audits the success and failure of the EXECUTE category with data.with **both** (‘B’) as well as the other categories. Other values possible are ‘N’ = None and for “EXECUTEWITHDATA” the value ‘Y’. As usual for this lovely dinosaur, the values are hard to find for someone not employed by IBM.

POLICY VALUE

--------------- ------------------

AUDITPOLICYNAME ADMINSPOLICY

AUDITSTATUS B

CONTEXTSTATUS B

VALIDATESTATUS B

CHECKINGSTATUS B

SECMAINTSTATUS B

OBJMAINTSTATUS B

SYSADMINSTATUS B

EXECUTESTATUS B

EXECUTEWITHDATA Y

<https://www.ibm.com/support/pages/db2how-check-current-db2audit-settings>

### Storage and analysis of audit logs

Archiving the audit log moves the active audit log to an archive directory while the server begins writing to a new, active audit log. Later, you can extract data from the archived log into delimited files and then load data from these files into Db2® database tables for analysis.

As noted earlier, the paths defined are akin to the following:

“Datapath” = /db2/CAI/AUDIT

“Archivepath” = /db2/CAI/AUDIT/audarchive

What this means is that auditing is actively writing logs to the “Datapath” and when you flush and archive the data the contents are copied to the “Archivepath”

#### Archiving active audit logs

There are both instance and database audit logs.

The general naming convention is in this general format:

db2audit.db.CAI.log.0 <- database audit logs

db2audit.instance.log.0 <- instance audit logs

Archived logs have a “database member ID” plus timestamp appended to the copies of the active log.

These can be extracted one of two ways, by a db2audit command or via a stored procedure to individual files to be loaded into tables. Please note that any large CLOB or BLOB field will have a secondary or companion files called “auditlobs” by default which stores the results and is dependent upon the the pointers in the files for the context and execute sections of auditing.

“The security administrator can use the SYSPROC.AUDIT\_ARCHIVE stored procedure and table function, the SYSPROC.AUDIT\_DELIM\_EXTRACT stored procedure, and the SYSPROC.AUDIT\_LIST\_LOGS table function to archive audit logs and extract data to delimited files.“

You really want to flush the records first.

Before archiving and extracting the db2 audit log file, you can flush all pending audit records by using the flush option of *db2audit*.  
**Do this as the instance owner (ex. db2cai)**

db2audit flush

db2audit archive <- INSTANCE LEVEL LOGS

db2audit archive database <SID> <- DATABASE LEVEL LOGS

Or for just the database level

CALL SYSPROC.AUDIT\_ARCHIVE ( '/db2/CAI/AUDIT/audarchive', -2 ) <- use the archive path!

It will create the archive for the database.

db2audit.db.CAI.log.0.20240626232249

Additional information:   
<https://www.ibm.com/docs/en/db2/11.5?topic=facility-storage-analysis-audit-logs>

#### Creating Extract Files

Now that you have archived files for your analysis, you need to load the archives into the audit tables you created earlier. You will want to perform this as the DB2 instance owner (ex. db2cai) and not one of the audited users.

By convention at Colgate, the schema is AUDIT.

You’ll be extracting the selected archives to the following files:

validate.del

secmaint.del

objmaint.del

audit.del

sysadmin.del

checking.del

For these two below, if active and extracted there is likely an “auditlob” file as well.

context.del

execute.del

db2audit extract delasc to /db2/CAI/AUDIT/audarchive from files /db2/CAI/AUDIT/audarchive/db2audit.db.CAI.log.0.<timestamp>

You can wildcard any of that and you’ll see that in the next example using a stored procedure.

Here’s an example of an extract for a year for all types of logs, be it instance or database.

CALL SYSPROC.AUDIT\_DELIM\_EXTRACT(

'', '', '/auditarchive', 'db2audit.%.200604%', '' )

In another example, they can call the SYSPROC.AUDIT\_DELIM\_EXTRACT stored procedure to extract the archived audit records with success events from the EXECUTE category and failure events from the CHECKING category, from a file with the timestamp they are interested in:

CALL SYSPROC.AUDIT\_DELIM\_EXTRACT( '', '', '/auditarchive',

'db2audit.%.20060419034937', 'category

execute status success, checking status failure );

More examples are here:  
  
<https://www.ibm.com/docs/en/db2/11.5?topic=facility-storage-analysis-audit-logs>

#### Creating the database tables for loading the audit extract files

This should have already been done, so if you are in doubt, please refer to this document:

[Setting Up DB2 Audit in CAI](https://docs.google.com/document/d/1bnZrru67WzTjtJtIQcqEYKKB1drbOuCyrVGOoZZ4KJs/edit)

#### Loading the tables From the Extract Files

Most of this is given here [Setting Up DB2 Audit in CAI](https://docs.google.com/document/d/1bnZrru67WzTjtJtIQcqEYKKB1drbOuCyrVGOoZZ4KJs/edit) but we will repeat the important points.

db2audit extract delasc to /db2/CAI/AUDIT/audarchive from files /db2/CAI/AUDIT/audarchive/db2audit.instance.log.0\*

db2audit extract delasc to /db2/CAI/AUDIT/audarchive from files /db2/CAI/AUDIT/audarchive/db2audit.db.CAI.log.0\*

You can also wildcard the file names like so.   
/db2/CAI/AUDIT/audarchive/db2audit.\*

How to look at audit data. Schema in this case is “AUDIT”. First we load the data, preferably without being one of the audited users.  **Please note the “nonrecoverable” clause.** You must include that or you will put whatever tablespace the tables exist in into LOAD PENDING state which requires you to remove the state at some point by [various techniques](https://docs.google.com/document/d/1bnZrru67WzTjtJtIQcqEYKKB1drbOuCyrVGOoZZ4KJs/edit#heading=h.6pvgtpavis9w), As these are standalone tables it’s much simpler to add the clause.

You can check the tablespace states quickly by   
db2 list tablespaces |grep -i state

Or you can look for anything not equal to State = 0x0000

db2 list tablespaces |grep -i state | grep -v 0x0000

db2 set schema audit;

db2 “LOAD FROM /db2/CAI/AUDIT/audarchive/audit.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.audit nonrecoverable”

db2 “LOAD FROM /db2/CAI/AUDIT/audarchive/checking.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.checking nonrecoverable”

db2 “LOAD FROM /db2/CAI/AUDIT/audarchive/context.del OF del MODIFIED BY LOBSINFILE INSERT INTO audit.context nonrecoverable”

db2 “LOAD FROM /db2/CAI/AUDIT/audarchive/execute.del OF del MODIFIED BY LOBSINFILE INSERT INTO audit.execute nonrecoverable”

db2 “LOAD FROM /db2/CAI/AUDIT/audarchive/objmaint.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.objmaint nonrecoverable”

db2 “LOAD FROM /db2/CAI/AUDIT/audarchive/secmaint.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.secmaint nonrecoverable”

db2 “LOAD FROM /db2/CAI/AUDIT/audarchive/sysadmin.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.sysadmin nonrecoverable”

db2 “LOAD FROM /db2/CAI/AUDIT/audarchive/validate.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.validate nonrecoverable”

Note that the tables have no indexes and may contain duplicate records.

#### Analyze audit data:

It’s important to log in as a user that isn’t really audited.

db2 connect to <sid> user db2<sid> using <password>

Let’s start with a sample where we want the executed SQL Statements with the variable data in the statement (such as the where clause). You can pick and choose the columns you want to use, these are only suggestions.

These are sample queries which highlight a possible request for data. Pick and choose the columns you want. The “SUBSTR” clause is there as a sample to truncate large column definitions like USERID which is defined as 128 characters to a readable size.

For the first two statements to load tables, the load command should specify the “MODIFIED BY LOBSINFILE” as shown above. Otherwise you will only return a pointer to the lobfile. The lob file(s) are created at OS level when the db2audit extract … command is run.   
  
This applies to the two tables that have a column called STMTTEXT of type CLOB (EXECUTE and CONTEXT)

db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10) AS USER, SUBSTR(STMTTEXT,1,110) AS SQL FROM AUDIT.EXECUTE WHERE USERID = ('uskmf90') and STMTTEXT != '- '"

Example output:   
2024-06-13-01.28.44.499784 uskmf90 delete from sapcap.basis\_admins

2024-06-13-01.46.00.194267 uskmf90 SELECT COLNAME, TYPESCHEMA, TYPENAME, LENGTH, SCALE, NULLS FROM SYSCAT.COLUMNS WHERE TABSCHEMA = 'AUDIT' AND T

2024-06-26-01.57.00.889045 uskmf90 INSERT INTO DB2CAP.BASIS\_ADMINS (age, Name) VALUES ('22', 'Gack')

For the context table it’s strongly advised that you filter out the “hvr<sid>” user and filter out the STMTTEXT values of “-”. In the case below, you do not want to include the “hvr<sid>” user as these entries are ridiculously prolific.

db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10)AS USER, SUBSTR(STMTTEXT,1,110) AS SQL FROM AUDIT.context WHERE USERID != 'hvrcai' and STMTTEXT != '- '"

The remaining tables

db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10) AS USER, SUBSTR(AUTHID, 1, 10) AS AUTHID,CATEGORY, EVENT FROM AUDIT.AUDIT WHERE USER IN ('uskmf90', 'usdjv01')"

db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10) AS USER, SUBSTR(AUTHID, 1, 10) AS AUTHID, CATEGORY, EVENT FROM AUDIT.SYSADMIN WHERE USER IN ('uskmf90', 'usdjv01')"

db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10) AS USER, SUBSTR(AUTHID, 1, 10) AS AUTHID, CATEGORY, EVENT FROM AUDIT.CHECKING WHERE USER IN ('uskmf90', 'usdjv01')"

db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10) AS USER, SUBSTR(AUTHID, 1, 10) AS AUTHID, CATEGORY, EVENT FROM AUDIT.OBJMAINT WHERE USER IN ('uskmf90', 'usdjv01')"

db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10) AS USER, SUBSTR(AUTHID, 1, 10) AS AUTHID, CATEGORY, EVENT FROM AUDIT.VALIDATE WHERE USER IN ('uskmf90', 'usdjv01')"

db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10) AS USER, SUBSTR(AUTHID, 1, 10) AS AUTHID, CATEGORY, EVENT FROM AUDIT.SECMAINT WHERE USERID IN ('uskmf90', 'usdjv01')"

## Demo

Connect as your audited user (in my case “SAPCAP”)  
  
jq03a010:db2cai 2> db2 connect to cai user SAPCAP using [redacted]

Database Connection Information

Database server = DB2/AIX64 11.5.9.0

SQL authorization ID = USKMF90

Local database alias = CAI

jq03a010:db2cai 3> db2 "select \* from DB2CAP.BASIS\_ADMINS"

AGE NAME

------ ------------------------------

22 Gack

1 record(s) selected.

jq03a010:db2cai 4> db2 'delete from DB2CAP.BASIS\_ADMINS'

DB20000I The SQL command completed successfully.

jq03a010:db2cai 5> db2 commit

DB20000I The SQL command completed successfully.

jq03a010:db2cai 6> db2 "INSERT INTO db2cap.BASIS\_ADMINS (age, Name) VALUES ('55', 'JamesM')"

DB20000I The SQL command completed successfully.

jq03a010:db2cai 7> db2 commit

DB20000I The SQL command completed successfully.

jq03a010:db2cai 8> db2 "select \* from DB2CAP.BASIS\_ADMINS"

AGE NAME

------ ------------------------------

55 JamesM

1 record(s) selected.

jq03a010:db2cai 9> db2audit flush

AUD0000I Operation succeeded.

jq03a010:db2cai 10> db2 commit

DB20000I The SQL command completed successfully.

Did we get output written recently?   
  
jq03a010:db2cai 19> ls -ltra db2audit.\*log\*

-rw------- 1 db2cai dbcaiadm 18862 Jun 27 01:47 db2audit.db.CAI.log.0

-rw------- 1 db2cai dbcaiadm 159594183 Jun 27 01:48 db2audit.instance.log.0

Let’s archive and look up in the audarchive subdirectory

jq03a010:db2cai 20> db2audit archive database CAI

Member DB Partition AUD Archived or Interim Log File

Number Number Message

-------- -------------- -------- -------------------------------------------------

0 0 AUD0000I db2audit.db.CAI.log.0.20240627015015

AUD0000I Operation succeeded.

jq03a010:db2cai 21> db2audit archive

Member DB Partition AUD Archived or Interim Log File

Number Number Message

-------- -------------- -------- -------------------------------------------------

0 0 AUD0000I db2audit.instance.log.0.20240627015023

AUD0000I Operation succeeded.

jq03a010:db2cai 22> cd audarch\*

jq03a010:db2cai 23> ls -ltra db2audit.\*log\*.20240627\*

-rw------- 1 db2cai dbcaiadm 18862 Jun 27 01:50 db2audit.db.CAI.log.0.20240627015015

-rw------- 1 db2cai dbcaiadm 159845761 Jun 27 01:50 db2audit.instance.log.0.20240627015023

jq03a010:db2cai 24>

Success.

Let’s extract these to \*.del files

db2audit extract delasc to /db2/CAI/AUDIT/audarchive from files /db2/CAI/AUDIT/audarchive/db2audit.instance.log.0.20240627015023

db2audit extract delasc to /db2/CAI/AUDIT/audarchive from files /db2/CAI/AUDIT/audarchive/db2audit.db.CAI.log.0.20240627015015

jq03a010:db2cai 24> db2audit extract delasc to /db2/CAI/AUDIT/audarchive from files /db2/CAI/AUDIT/audarchive/db2audit.instance.log.0.20240627015023

AUD0000I Operation succeeded.

jq03a010:db2cai 25> db2audit extract delasc to /db2/CAI/AUDIT/audarchive from files /db2/CAI/AUDIT/audarchive/db2audit.db.CAI.log.0.20240627015015

AUD0000I Operation succeeded.

jq03a010:db2cai 26> ls -ltra

Let’s check the \*.del files

[...]

-rw-rw---- 1 db2cai dbcaiadm 1567857 Jun 27 01:56 validate.del

-rw-rw---- 1 db2cai dbcaiadm 102357 Jun 27 01:56 sysadmin.del

-rw-rw---- 1 db2cai dbcaiadm 7013 Jun 27 01:56 audit.del

-rw-rw---- 1 db2cai dbcaiadm 223853223 Jun 27 01:56 context.del

-rw-rw---- 1 db2cai dbcaiadm 115409253 Jun 27 01:56 checking.del

-rw-rw---- 1 db2cai dbcaiadm 96903 Jun 27 01:56 execute.del

-rw-rw---- 1 db2cai dbcaiadm 148291 Jun 27 01:56 auditlobs

Just a minute ago (at the time of this writing .

Let’s load up the most recent archive for the context and execute sections only.

Remember the EXECUTE and CONTEXT likely have LOB content.

jq03a010:db2cai 32> db2 "LOAD FROM /db2/CAI/AUDIT/audarchive/audit.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.audit nonrecoverable";

SQL3109N The utility is beginning to load data from file

"/db2/CAI/AUDIT/audarchive/audit.del".

SQL3500W The utility is beginning the "LOAD" phase at time "06/27/2024

01:59:37.255788".

SQL3519W Begin Load Consistency Point. Input record count = "0".

SQL3520W Load Consistency Point was successful.

SQL3110N The utility has completed processing. "40" rows were read from the

input file.

SQL3519W Begin Load Consistency Point. Input record count = "40".

SQL3520W Load Consistency Point was successful.

SQL3515W The utility has finished the "LOAD" phase at time "06/27/2024

01:59:37.293007".

Number of rows read = 40

Number of rows skipped = 0

Number of rows loaded = 40

Number of rows rejected = 0

Number of rows deleted = 0

Number of rows committed = 40

jq03a010:db2cai 33> db2 "LOAD FROM /db2/CAI/AUDIT/audarchive/checking.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.checking nonrecoverable";

SQL3109N The utility is beginning to load data from file

"/db2/CAI/AUDIT/audarchive/checking.del".

SQL3500W The utility is beginning the "LOAD" phase at time "06/27/2024

01:59:48.561474".

SQL3519W Begin Load Consistency Point. Input record count = "0".

SQL3520W Load Consistency Point was successful.

SQL3110N The utility has completed processing. "372597" rows were read from

the input file.

SQL3519W Begin Load Consistency Point. Input record count = "372597".

SQL3520W Load Consistency Point was successful.

SQL3515W The utility has finished the "LOAD" phase at time "06/27/2024

01:59:49.058189".

Number of rows read = 372597

Number of rows skipped = 0

Number of rows loaded = 372597

Number of rows rejected = 0

Number of rows deleted = 0

Number of rows committed = 372597

jq03a010:db2cai 34> db2 "LOAD FROM /db2/CAI/AUDIT/audarchive/context.del OF del MODIFIED BY LOBSINFILE INSERT INTO audit.context nonrecoverable";

SQL3109N The utility is beginning to load data from file

"/db2/CAI/AUDIT/audarchive/context.del".

SQL3500W The utility is beginning the "LOAD" phase at time "06/27/2024

01:59:57.603664".

SQL3519W Begin Load Consistency Point. Input record count = "0".

SQL3520W Load Consistency Point was successful.

SQL3110N The utility has completed processing. "1021431" rows were read from

the input file.

SQL3519W Begin Load Consistency Point. Input record count = "1021431".

SQL3520W Load Consistency Point was successful.

SQL3515W The utility has finished the "LOAD" phase at time "06/27/2024

02:00:03.623823".

Number of rows read = 1021431

Number of rows skipped = 0

Number of rows loaded = 1021431

Number of rows rejected = 0

Number of rows deleted = 0

Number of rows committed = 1021431

jq03a010:db2cai 35> db2 "LOAD FROM /db2/CAI/AUDIT/audarchive/execute.del OF del MODIFIED BY LOBSINFILE INSERT INTO audit.execute nonrecoverable";

SQL3109N The utility is beginning to load data from file

"/db2/CAI/AUDIT/audarchive/execute.del".

SQL3500W The utility is beginning the "LOAD" phase at time "06/27/2024

02:00:11.503333".

SQL3519W Begin Load Consistency Point. Input record count = "0".

SQL3520W Load Consistency Point was successful.

SQL3110N The utility has completed processing. "354" rows were read from the

input file.

SQL3519W Begin Load Consistency Point. Input record count = "354".

SQL3520W Load Consistency Point was successful.

SQL3515W The utility has finished the "LOAD" phase at time "06/27/2024

02:00:11.560783".

Number of rows read = 354

Number of rows skipped = 0

Number of rows loaded = 354

Number of rows rejected = 0

Number of rows deleted = 0

Number of rows committed = 354

jq03a010:db2cai 36> db2 "LOAD FROM /db2/CAI/AUDIT/audarchive/objmaint.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.objmaint nonrecoverable";

SQL3109N The utility is beginning to load data from file

"/db2/CAI/AUDIT/audarchive/objmaint.del".

SQL3500W The utility is beginning the "LOAD" phase at time "06/27/2024

02:00:20.758845".

SQL3519W Begin Load Consistency Point. Input record count = "0".

SQL3520W Load Consistency Point was successful.

SQL3110N The utility has completed processing. "2" rows were read from the

input file.

SQL3519W Begin Load Consistency Point. Input record count = "2".

SQL3520W Load Consistency Point was successful.

SQL3515W The utility has finished the "LOAD" phase at time "06/27/2024

02:00:20.808787".

Number of rows read = 2

Number of rows skipped = 0

Number of rows loaded = 2

Number of rows rejected = 0

Number of rows deleted = 0

Number of rows committed = 2

jq03a010:db2cai 37> db2 "LOAD FROM /db2/CAI/AUDIT/audarchive/secmaint.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.secmaint nonrecoverable";

SQL3109N The utility is beginning to load data from file

"/db2/CAI/AUDIT/audarchive/secmaint.del".

SQL3500W The utility is beginning the "LOAD" phase at time "06/27/2024

02:00:29.451099".

SQL3519W Begin Load Consistency Point. Input record count = "0".

SQL3520W Load Consistency Point was successful.

SQL3110N The utility has completed processing. "0" rows were read from the

input file.

SQL3519W Begin Load Consistency Point. Input record count = "0".

SQL3520W Load Consistency Point was successful.

SQL3515W The utility has finished the "LOAD" phase at time "06/27/2024

02:00:29.493448".

Number of rows read = 0

Number of rows skipped = 0

Number of rows loaded = 0

Number of rows rejected = 0

Number of rows deleted = 0

Number of rows committed = 0

jq03a010:db2cai 38> db2 "LOAD FROM /db2/CAI/AUDIT/audarchive/sysadmin.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.sysadmin nonrecoverable";

SQL3109N The utility is beginning to load data from file

"/db2/CAI/AUDIT/audarchive/sysadmin.del".

SQL3500W The utility is beginning the "LOAD" phase at time "06/27/2024

02:00:38.629709".

SQL3519W Begin Load Consistency Point. Input record count = "0".

SQL3520W Load Consistency Point was successful.

SQL3110N The utility has completed processing. "650" rows were read from the

input file.

SQL3519W Begin Load Consistency Point. Input record count = "650".

SQL3520W Load Consistency Point was successful.

SQL3515W The utility has finished the "LOAD" phase at time "06/27/2024

02:00:38.667842".

Number of rows read = 650

Number of rows skipped = 0

Number of rows loaded = 650

Number of rows rejected = 0

Number of rows deleted = 0

Number of rows committed = 650

jq03a010:db2cai 39> db2 "LOAD FROM /db2/CAI/AUDIT/audarchive/validate.del OF del MODIFIED BY DELPRIORITYCHAR INSERT INTO audit.validate nonrecoverable";

SQL3109N The utility is beginning to load data from file

"/db2/CAI/AUDIT/audarchive/validate.del".

SQL3500W The utility is beginning the "LOAD" phase at time "06/27/2024

02:00:46.402146".

SQL3519W Begin Load Consistency Point. Input record count = "0".

SQL3520W Load Consistency Point was successful.

SQL3110N The utility has completed processing. "8358" rows were read from

the input file.

SQL3519W Begin Load Consistency Point. Input record count = "8358".

SQL3520W Load Consistency Point was successful.

SQL3515W The utility has finished the "LOAD" phase at time "06/27/2024

02:00:46.444148".

Number of rows read = 8358

Number of rows skipped = 0

Number of rows loaded = 8358

Number of rows rejected = 0

Number of rows deleted = 0

Number of rows committed = 8358

jq03a010:db2cai 40> jq03a010:db2cai 40>

Now we can query the tables.   
  
Let’s just take the execute and context tables for the demo with sample SQL.

jq03a010:db2cai 40> db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10) AS USER, SUBSTR(STMTTEXT,1,110) AS SQL FROM AUDIT.EXECUTE WHERE USERID = ('uskmf90') and STMTTEXT != '- '"

TIMESTAMP USER SQL

-------------------------- ---------- --------------------------------------------------------------------------------------------------------------

2024-06-07-20.04.39.034917 uskmf90 select count(\*) from SAPCAP.USR40

2024-06-11-20.48.42.485924 uskmf90 INSERT INTO SAPCAP.BASIS\_ADMINS (age, Name) VALUES ('44','CujoDeSockpuppet')

[...]

**2024-06-27-01.36.22.812758 uskmf90 select \* from DB2CAP.BASIS\_ADMINS**

**2024-06-27-01.37.54.506190 uskmf90 delete from DB2CAP.BASIS\_ADMINS**

**2024-06-27-01.40.10.429853 uskmf90 INSERT INTO db2cap.BASIS\_ADMINS (age, Name) VALUES ('55', 'JamesM')**

**2024-06-27-01.40.23.422336 uskmf90 select \* from DB2CAP.BASIS\_ADMINS**

518 record(s) selected.

And  
  
jq03a010:db2cai 48> db2 "SELECT TIMESTAMP, SUBSTR(USERID, 1, 10)AS USER, SUBSTR(STMTTEXT,1,110) AS SQL FROM AUDIT.context WHERE USERID != 'hvrcai' and STMTTEXT != '- '"

TIMESTAMP USER SQL

-------------------------- ---------- --------------------------------------------------------------------------------------------------------------

2024-06-11-21.07.52.081826 uskmf90 INSERT INTO SAPCAP.BASIS\_ADMINS (age, Name) VALUES ('45','CujodeSockpuppet')

2024-06-12-01.14.52.044072 uskmf90 SELECT COLNAME, TYPESCHEMA, TYPENAME, LENGTH, SCALE, NULLS FROM SYSCAT.COLUMNS WHERE TABSCHEMA = 'AUDIT' AND T

2024-06-12-01.14.52.053279 uskmf90 SQLCUR201

2024-06-12-01.14.52.059079 uskmf90 SQLCUR201

2024-06-13-00.12.35.947221 uskmf90 INSERT INTO SAPCAP.BASIS\_ADMINS (age, Name) VALUES ('55', 'JamesM')

2024-06-11-21.07.52.081826 uskmf90 INSERT INTO SAPCAP.BASIS\_ADMINS (age, Name) VALUES ('45','CujodeSockpuppet')

2024-06-12-01.14.52.044072 uskmf90 SELECT COLNAME, TYPESCHEMA, TYPENAME, LENGTH, SCALE, NULLS FROM SYSCAT.COLUMNS WHERE TABSCHEMA = 'AUDIT' AND T

[...]

2024-06-27-01.36.22.807866 uskmf90 SQLCUR201

2024-06-27-01.36.22.811346 uskmf90 select \* from DB2CAP.BASIS\_ADMINS

2024-06-27-01.36.22.812698 uskmf90 SQLCUR201

2024-06-27-01.37.54.500459 uskmf90 delete from DB2CAP.BASIS\_ADMINS

2024-06-27-01.39.09.489564 uskmf90 INSERT INTO db2cai..BASIS\_ADMINS (age, Name) VALUES ('55', 'JamesM')

2024-06-27-01.39.31.638590 uskmf90 INSERT INTO db2cai.BASIS\_ADMINS (age, Name) VALUES ('55', 'JamesM')

2024-06-27-01.40.10.427987 uskmf90 INSERT INTO db2cap.BASIS\_ADMINS (age, Name) VALUES ('55', 'JamesM')

2024-06-27-01.40.23.420495 uskmf90 select \* from DB2CAP.BASIS\_ADMINS

2024-06-27-01.40.23.420680 uskmf90 SQLCUR201

2024-06-27-01.40.23.420853 uskmf90 select \* from DB2CAP.BASIS\_ADMINS

2024-06-27-01.40.23.422225 uskmf90 SQLCUR201

## Appendix

### Values for the **OBJECTTYPE** column in the **SYSCAT.AUDITUSE** table

Here are the possible values for the **OBJECTTYPE** column in the **SYSCAT.AUDITUSE** table and their meanings:

* A: Alias
* C: Table or View
* F: Routine (Function or Procedure)
* G: Global Variable
* I: Index
* L: Sequence
* M: Materialized Query Table (MQT)
* N: Nickname
* O: Module
* P: Package
* R: Routine (Function or Procedure)
* S: Server
* T: Trigger
* U: User-defined Type (UDT)
* V: View
* W: Wrapper
* X: Index Extension

Note: The additional value of lowercase “i” value in the **OBJECTTYPE** column of the **SYSCAT.AUDITUSE** table indicates an "inline SQL function." Inline SQL functions are a type of user-defined function (UDF) that are defined using SQL and are inlined into the calling SQL statement, which can improve performance by reducing function call overhead.

* Ex: i: Inline SQL function

This distinction is important for understanding the specific type of object being audited, especially when dealing with user-defined functions and their performance implications.

These values help identify the specific type of database object that is being audited, which can be useful for auditing and monitoring purposes. The **SYSCAT.AUDITUSE** table is part of the system catalog tables in Db2, which store metadata about the database objects and their usage.

The **SUBOBJECTTYPE** column provides additional granularity about the type of sub-object within the main object type specified in the **OBJECTTYPE** column.

### Values for the **SUBOBJECTTYPE** column in the **SYSCAT.AUDITUSE** table

Here are the possible values for the **SUBOBJECTTYPE** column in the **SYSCAT.AUDITUSE** table and their meanings:

* A: Alias
* C: Column
* D: Distinct type
* F: Function
* G: Global variable
* I: Index
* L: Label
* M: Module
* N: Nickname
* O: Constraint
* P: Procedure
* Q: Sequence

R: Routine

* S: Server
* T: Trigger
* U: User-defined type (UDT)
* V: View
* W: Wrapper
* X: Index extension

These values help identify the specific type of sub-object that is being audited, which can be useful for detailed auditing and monitoring purposes. The **SYSCAT.AUDITUSE** table is part of the system catalog tables in Db2, which store metadata about the database objects and their usage.

1. 2872218 - DB6: Using db2audit with SAP Applications for Audit polices

<https://me.sap.com/notes/2872218/E>

1. SAP note for Trusted Context Db2 12.1

<https://me.sap.com/notes/3484724/E>

1. Audit Guide link :

<https://help.sap.com/doc/7367f81b468e4480b3c550669b3534aa/CURRENT_VERSION/en-US/DB6_Admin_Guide.pdf>

<https://help.sap.com/docs/DB6/e3eefec5d20740f4872652a475457348/f81d9eba344145e488bcb55ec01e7f60.html>

## 