HrOUG 2021

25. hroug godišnja konferencija

"I don't need SQLcl, I have SQLPlus" You're wrong!

Erik van Roon

EvROCS
COMPLETING THE PUZZLE

Who Am I? Erik van Roon

- Originally analyst of microbiology and biochemistry
- Oracle developer since 1995, self-employed since 2009

EvROCS
COMPLETING THE PUZZLE

- Most of my work takes place inside the database
- Did several successful major datamigration projects
- Core member of the MASH program

ORACLE ACE

: erik.van.roon@evrocs.nl
: www.evrocs.nl
: @evrocs_nl

Mentor and Speaker Hub

Our goal is to *connect* speakers with mentors to assist in *preparing* technical sessions and *improving* presentation skills

Interested? Read more and get in touch

https://mashprogram.wordpress.com

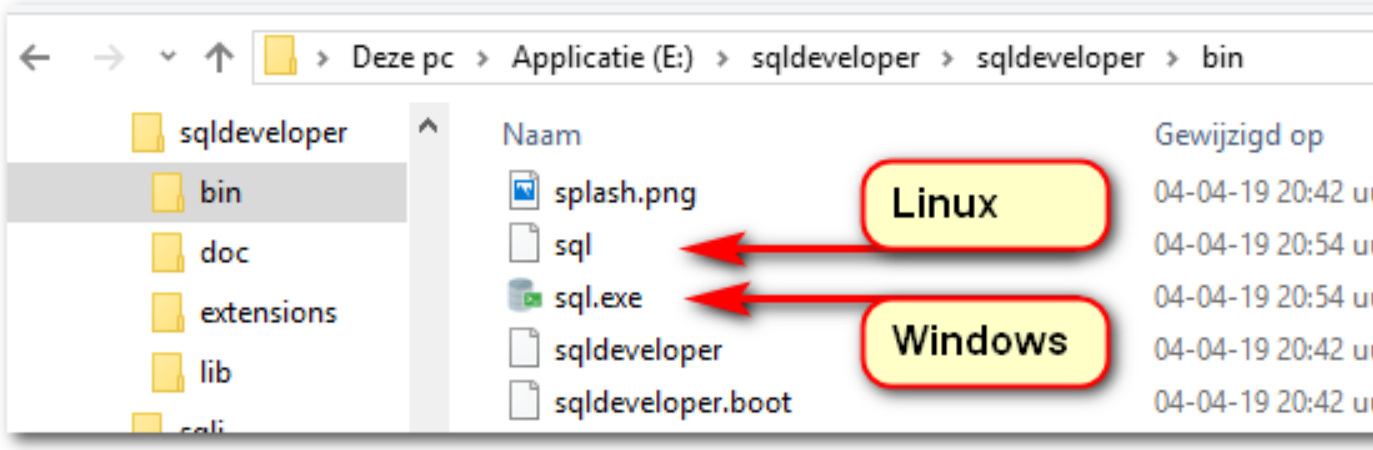
What is SQLcl?

- a command line Interface like SQL*Plus
- Built in Java by the SQL-Developer team
- Built around the SQLDev Java library for executing SQL
- Runs anywhere a JRE is available
  (Windows, Linux, Mac,…)

Q: Will it replace SQL*Plus?
A: No! “SQL*plus will always exist alongside SQLcl”

How do you get SQLcl?

- Included in SQL Developer



- So, also in DBMS installation (Recent versions)

- Available from Oracle Software Downloads
  - https://www.oracle.com/tools/downloads/sqldev-downloads.html
  - ~~https://www.oracle.com/nl/tools/downloads/sqlcl-downloads.html~~
  - https://www.oracle.com/tools/downloads/sqlcl-downloads.html

On June 20, 2021

Command Line - SQLcl 21.1.1

Version 21.1.1.113.1704 - May 4, 2021

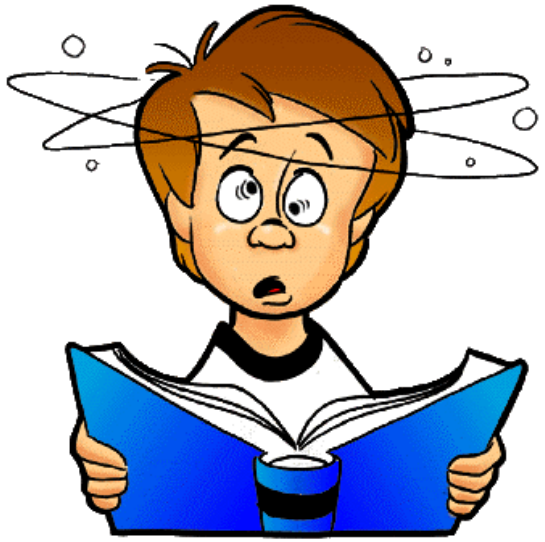
~~SQLcl 20.4.2 Downloads~~

~~Version 20.4.2.35.2359 - February 22, 2021~~

  - https://download.oracle.com/otn_software/java/sqldeveloper/sqlcl-latest.zip

Only place it's missing currently is the Oracle Client Software

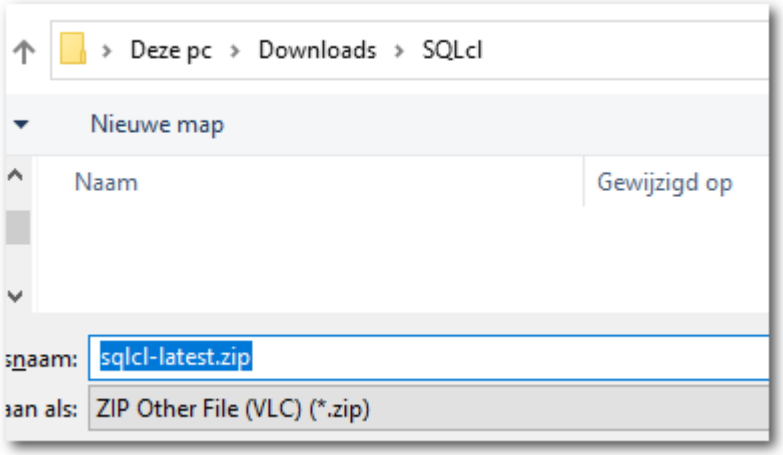
Installation



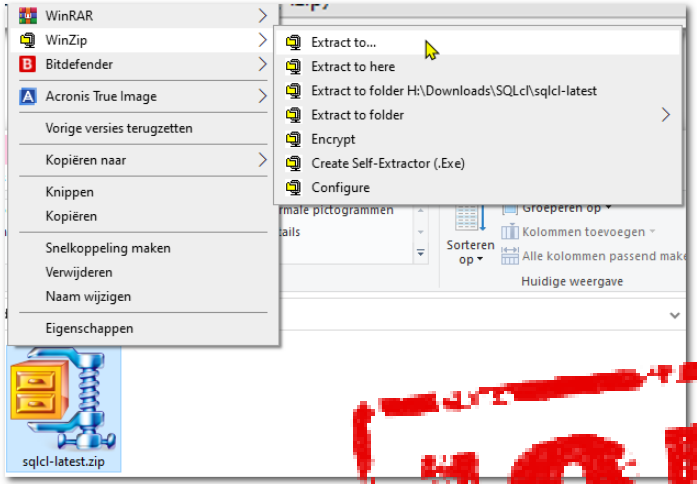
https://download.oracle.com/otn_software/java/sqldeveloper/sqlcl-latest.zip

1. Use direct download link

2. Save file

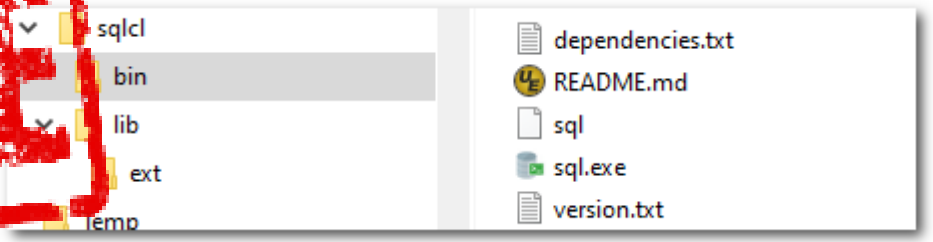


3. Unzip



JOB DONE

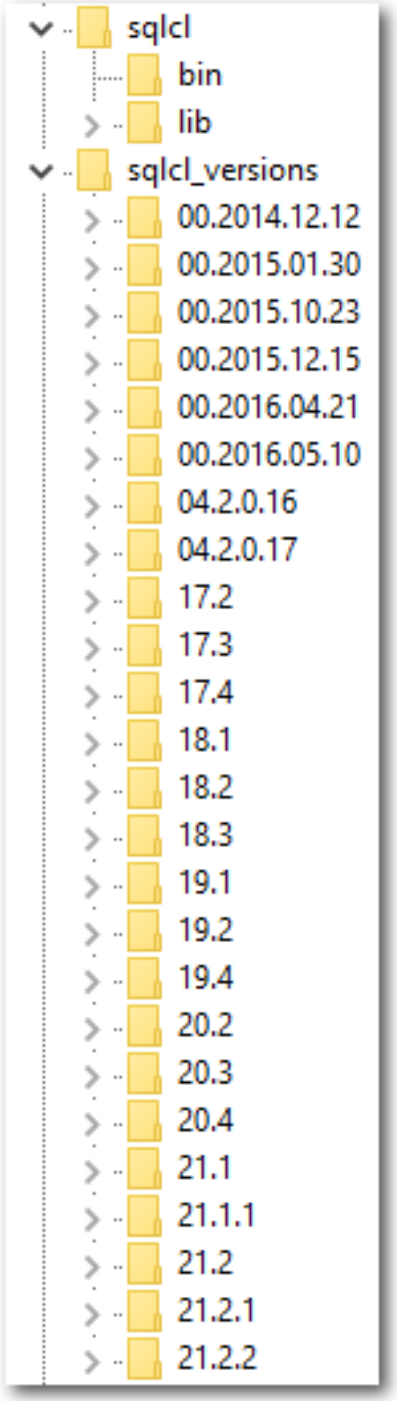
4. Run SQLcl



Can run multiple versions next to each other just unzip to different directories

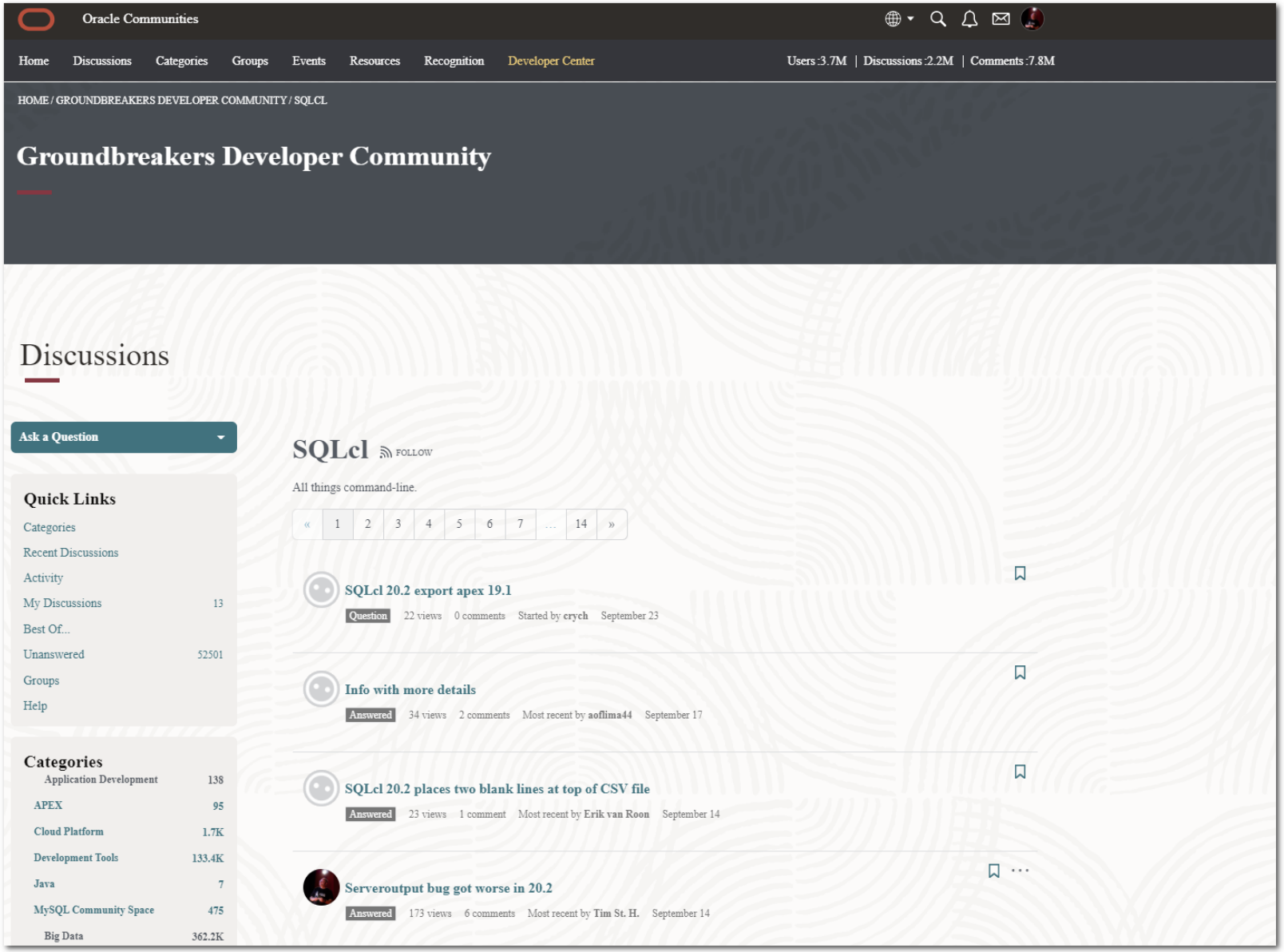
No need for

- Oracle client installation 🤔 (See tips, later)
- tnsnames.ora
  Though you can use one of course
- JDK (But need a JRE >= 1.8)



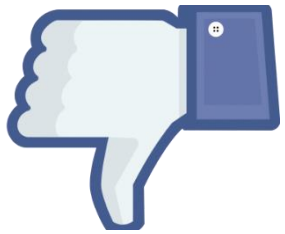
Dedicated Forum

https://community.oracle.com/tech/developers/categories/sqlcl




Are there any down sides?

Disadvantages? - Bugs

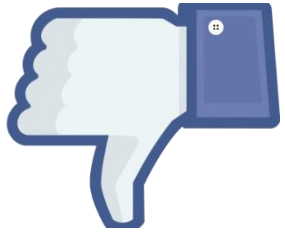


True

But remember:

- SQLcl is frequently updated
- SQL*Plus not immune to bugs either
  (e.g. unpatched version in 12.2 ignores SQLPATH)

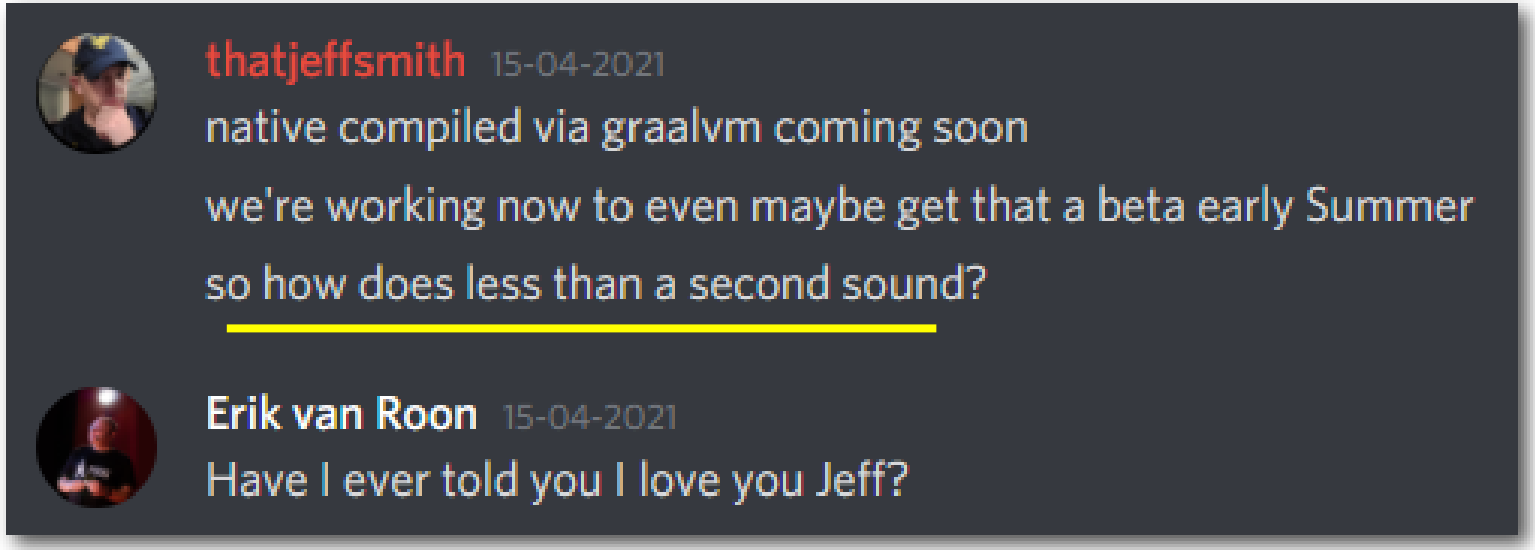
Disadvantages? - Startup slower dan SqlPlus



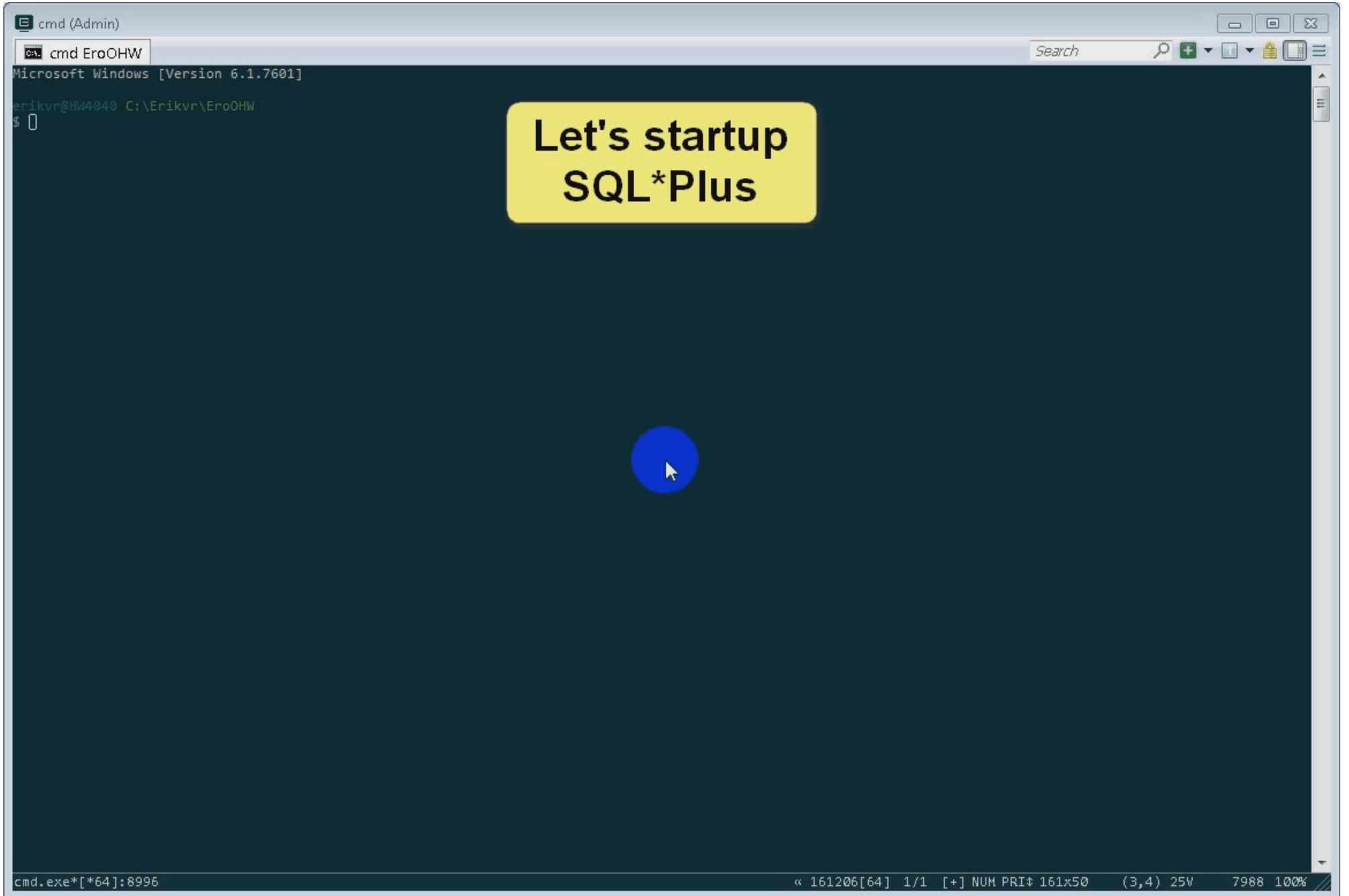
But remember:

- On average computer
  difference not enough to worry about
- On my laptop:
  Start Sql*Plus ⇨ < 1 sec
  Start SQLcl ⇨ 3 - 4 sec

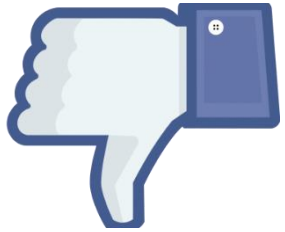
Good times are coming……



Slow startup (on VERY slow machine), the motion picture



Disadvantages? Does not support everything SqlPlus



21.1, "SQL*Plus List of Unsupported Commands and Features"

- REPHEADER
- REPFOOTER
- 21 Variables like COPYTYPECHECK, RECSEP, SHIFTINOUT

"Column" command accepts entire SqlPlus syntax…

But apart from "new_value" everything is ignored.

SqlPlus

```
ERO@EVROCS>set feedback off
ERO@EVROCS>column test noprint new_value test_var
ERO@EVROCS>select 'whatever'   test    from   dual;



ERO@EVROCS>
```

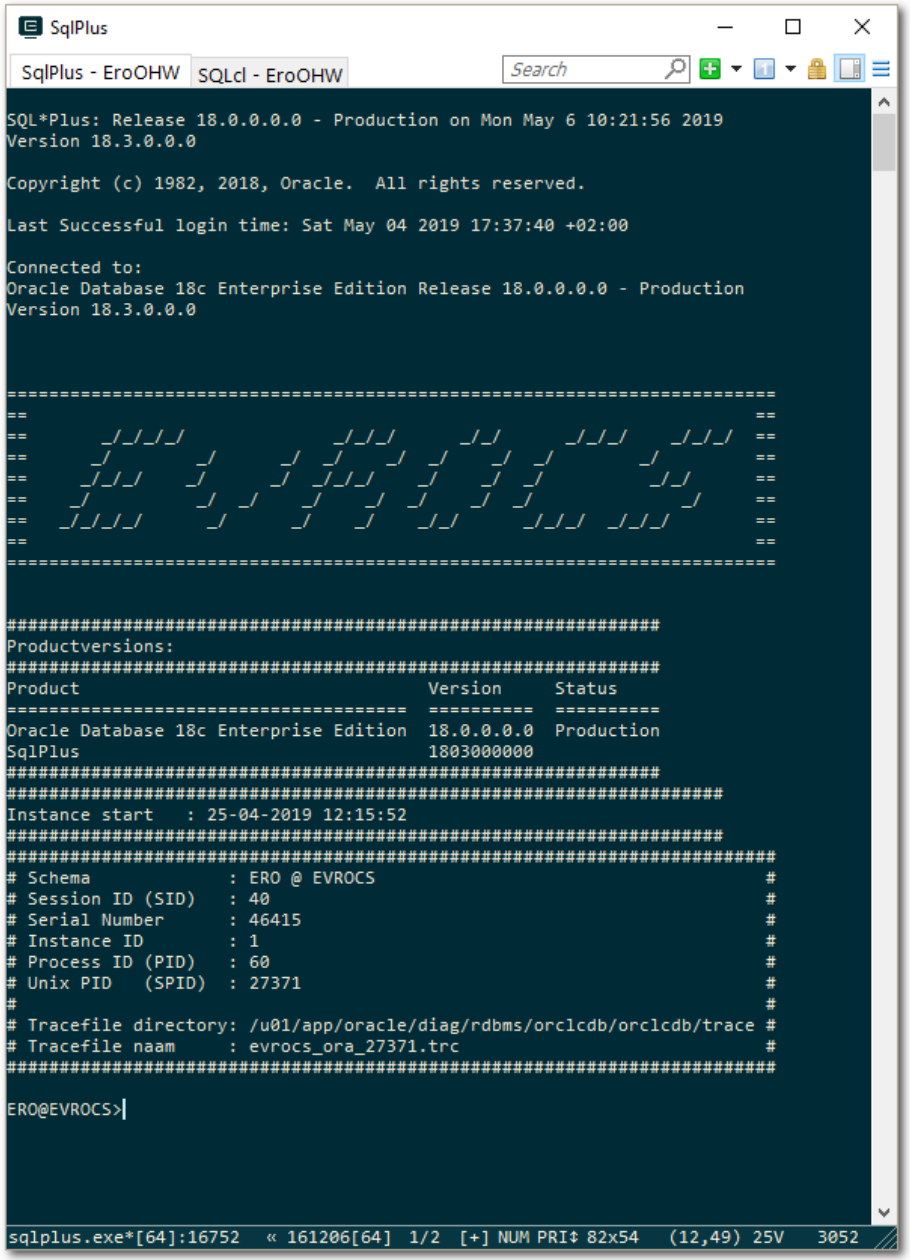
SQLcl

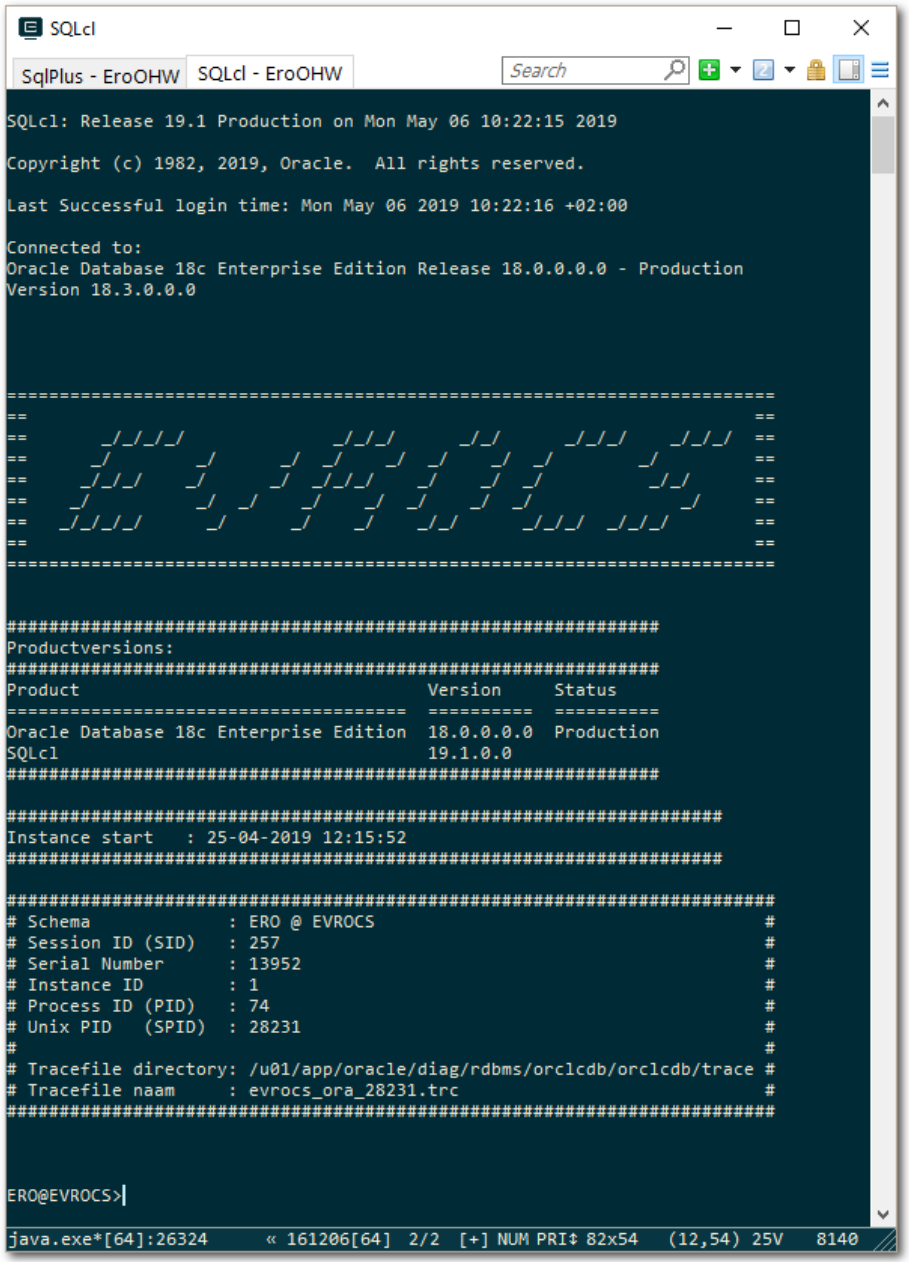
```
ERO@EVROCS>set feedback off
ERO@EVROCS>column test noprint new_value test_var
ERO@EVROCS>select 'whatever'   test    from   dual;
       TEST
____________
whatever

ERO@EVROCS>
```

What makes SQLcl the Cool Kid?

Find the differences






HELP !!!

Help

```
SQL> help
For help on a topic type help <topic>
List of Help topics available:
/                @                @@               ACCEPT           ALIAS            APEX             APPEND           ARBORI           ARCHIVE_LOG
BREAK            BRIDGE           BTITLE           CD               CHANGE           CLEAR            CLOUDSTORAGE     CODESCAN         COLUMN
COMPUTE          CONNECT          COPY             CS               CTAS             DBCCRED          DDL              DEFINE           DEL
DESCRIBE         DISCONNECT       EDIT             EXECUTE          EXIT             FIND             FORMAT           GET              HISTORY
HOST             INFORMATION      INPUT            LB               LIQUIBASE        LIST             LOAD             MODELER          NET
OCI              OERR             PASSWORD         PAUSE            PRINT            PROMPT           QUIT             REMARK           REPEAT
RESERVED_WORDS   REST             RUN              SAVE             SCRIPT           SET              SETERRORL        SHOW             SHUTDOWN
SODA             SPOOL            SSHTUNNEL        START            STARTUP          STORE            TIMING           TNSPING          TOSUB
TOSUB            TTITLE           UNDEFINE         UNLOAD           VARIABLE         VAULT            WHENEVER         WHICH            XQUERY
SQL>
```

Information on (almost) all functionality

Highlighted/underlined/asterisked commands are unique for SQLcl

Help is (really) good, but not perfect:

- Modifiers for Bridge not mentioned in help
- Seterrorl is not a command. "help seterrorl" leads to help for
  "set errorlogging …"
  which used to be missing in "help set" 😜, but not anymore


Command Completion

Command Completion

Start typing a statement

```
ERO@EVROCS>select * from evr
```

and hit [Tab]

'Hits' are displayed and if all hits have a common start (in this case EVROCS_), it is already filled in…

```
EVROCS_SNAPSHOTS                    EVROCS_SNAPSHOT_PARAMETERS
EVROCS_SNAPSHOT_SOURCES             EVROCS_SNAPSHOT_VERGELIJKINGEN
EVROCS_ADMIN.
ERO@EVROCS>select * from EVROCS_
```

- Query needs to 'make sense':
  SELECT FROM EVR [Tab]
  Won't work

Completion works for

- Table names/View names
- Keywords
- Column names
- After an **@** at the start of the line: files and directories in workdirectory

Command Completion, the motion picture

Status

Bar

Statusbar

Activate with: `set statusbar on`

```
ERO@EVROCS>
ERO@EVROCS>set statusbar on
```

```
ERO@EVROCS>set statusbar on
ERO@EVROCS>
viins ¦ 1:0 ¦ ERO ¦ evrocs
```

Available and used components: `show statusbar`

```
ERO@EVROCS>show statusbar
STATUS BAR: on
Used components:
        EDITMODE                Displays the current edit mode, emacs, vicmd or viins.
        LINECOL                 Displays the current line and column numbers.
        USERNAME                Displays the name of the currently connected database user.
        DBID                    Displays the id of the currently connected database.
Unused components:
        CURSOR                  Displays the current character position within the edit buffer.
        CWD                     Displays the Current Working Directory.
        ENCODING                Displays the local encoding set.
        GIT                     Displays the git branch containing the Current Working Directory.
        JAVA                    Displays the running Java version.
        TIMING                  Displays the elapsed time for the most recent database command.
        TXN                     Displays whether there are pending database changes that need a commit or rollback.
Default components:
        EDITMODE
        LINECOL
        USERNAME
        DBID
ERO@EVROCS>
viins ¦ 1:0 ¦ ERO ¦ evrocs
```

Statusbar

Add transaction component: **`set statusbar add txn`**

```
ERO@EVROCS>set statusbar add txn
ERO@EVROCS>
viins ¦ 1:0 ¦ ERO ¦ evrocs ¦ None
```

When there are uncommited changes:

```
ERO@EVROCS>delete from ero_test_load;

6 rows deleted.

ERO@EVROCS>
viins ¦ 1:0 ¦ ERO ¦ evrocs ¦ Changes Pending
```

Undo statusbar changes: **`set statusbar default`**

```
ERO@EVROCS>set statusbar default
ERO@EVROCS>
viins ¦ 1:0 ¦ ERO ¦ evrocs
```

Customize the default: **`set statusbar default dbid username txn`**

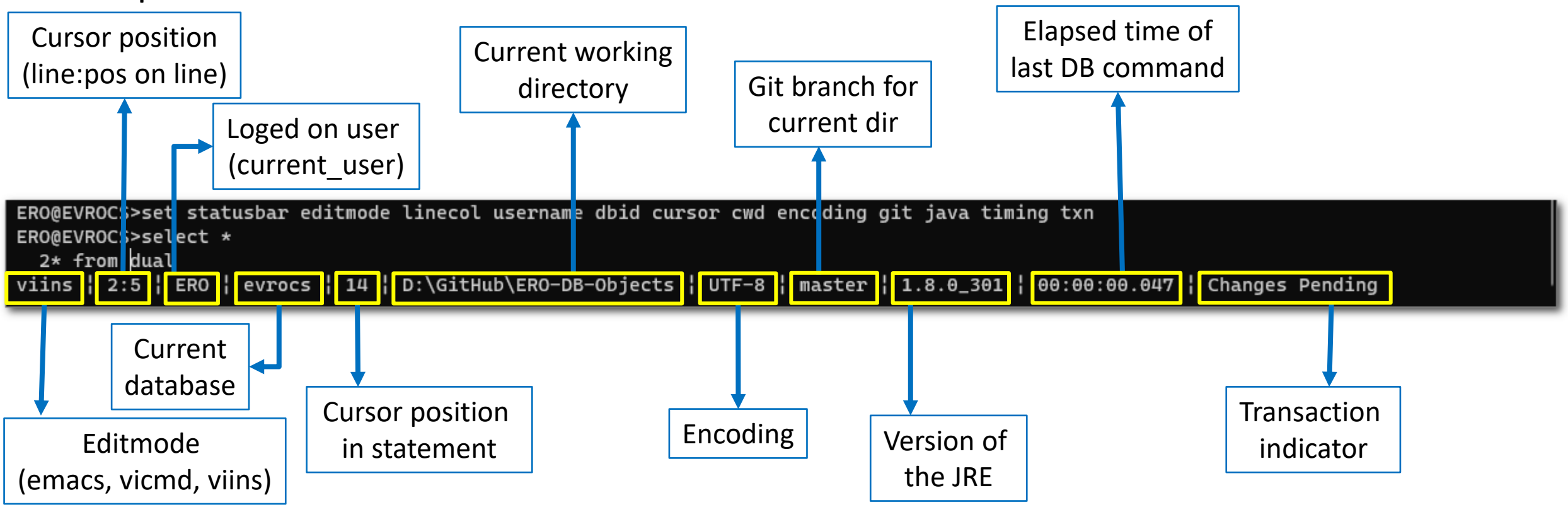
```
ERO@EVROCS>set statusbar default dbid username txn
ERO@EVROCS>set statusbar default
ERO@EVROCS>
evrocs ¦ ERO ¦ None
```

Statusbar

Componentname is case sensitive (lowercase!!) when overriding the current statusbar

```
ERO@EVROCS>set statusbar add TXN
ERO@EVROCS>set statusbar TXN
Statusbar: 'TXN' not found.
ERO@EVROCS>set statusbar txn
ERO@EVROCS>
Changes Pending
```

All components…..



Syntax Highlighting

Syntax Highlighting

Toggling highlighting
```
set highlighting on
set highlighting off
```
Resetting to defaults
```
set highlighting reset
```

Configuring highlighting

```
set highlighting [element type] foreground [color]
set highlighting [element type] background [color]
set highlighting [element type] [style] on|off|reset
```

Element types

- DEFAULT
- COMMENT
- STRING
- NUMBER
- PUNCTUATION
- KEYWORD
- IDENTIFIER

Colors

- RED
- BLUE
- BLACK
- CYAN
- GREEN
- MAGENTA
- WHITE
- YELLOW

Styles

- BOLD
- ULINE
- INVERSE

Syntax Highlighting In Action

Highlighting is off

Configuring Highlighting

Set Highlighting on

Same query, with highlighting

```
ERO@EVROCS>select object_name
  2  ,      object_type  -- and some comments
  3  from   user_objects
  4* where  object_type = 'whatever';


0 rows selected.

ERO@EVROCS>set highlighting keyword foreground yellow
ERO@EVROCS>set highlighting keyword bold on
ERO@EVROCS>set highlighting comment inverse on
ERO@EVROCS>set highlighting identifier foreground blue
ERO@EVROCS>set highlighting string background red
ERO@EVROCS>set highlighting string foreground black
ERO@EVROCS>set highlighting on
ERO@EVROCS>
ERO@EVROCS>select object_name
  2  ,      object_type  -- and some comments
  3  from   user_objects
  4* where  object_type = 'whatever';


0 rows selected.

ERO@EVROCS>
```


History

History

Browse through your previous statements using arrow-up & arrow-down
Per statement!! Not per entered line as in SqlPlus.

`history` Show list of executed statements, 1 per line

```
ERO@EVROCS>history
History:
  1  select * from   employees fetch first 2 rows only /
  2  select count(*) from   all_objects where  object_name like 'E%';
  3  select table_name from   user_tables order by 1 /
ERO@EVROCS>
```

`history full` Show list of executed statements, as entered (multiline)

```
ERO@EVROCS>history full
  1  select *
  >  from   employees
  >  fetch first 2 rows only
  >  /

  2  select count(*)
  >  from   all_objects
  >  where  object_name like 'E%';

  3  select table_name
  >  from   user_tables
  >  order by 1
  >  /
```

history usage Same as "history" but also displays number of executions

```
ERO@EVROCS>history usage
  1  (1) select * from   employees fetch first 2 rows only /
  2  (1) select table_name from   user_tables order by 1 /
  3  (5) select count(*) from   all_objects where  object_name like 'E%';
ERO@EVROCS>
```

history time Same as "history" but also displays execution time

```
ERO@EVROCS>history time
  1  (00.471) select * from   employees fetch first 2 rows only /
  2  (00.199) select table_name from   user_tables order by 1 /
  3  (00.151) select count(*) from   all_objects where  object_name like 'E%';
ERO@EVROCS>
```

history clear Erases the history buffer

history # Loads statement with number # into SQLcl buffer

show history Shows the current history settings

```
ERO@EVROCS>show history
HISTORY
        enabled
        blacklist: show,history,connect,set
        Do not show failed statements
ERO@EVROCS>
```

History – some remarks

- is **enabled** (haven't found a way to disable)
- **Doesn't store duplicates** of statements, each unique statement is stored only once
- Stores **multiline** statements **as one entry**, not individual lines
- stores a maximum of **100** statements, increase/decrease with for example:

  ```
  set history limit 500
  set history limit default
  ```
  to return to default behavior
- **Doesn't *store* failed queries**

  Toggle this behavior with

  ```
  set history fails limit 10
  ```
  to have a max of 10 failed statements in history

  ```
  set history nofails
  ```
  to go back to default behavior
- Filters out **show**, **history**, **connect** and **set** commands.

  Change this list with for example: (note, pre 20.3 'filter' should be 'blacklist')

  ```
  set history filter show,history,connect,set,commit,rollback
  ```
- History is **saved at end of SQLcl session**

  But **only** if you end it with **exit** or **quit**, not if you close the window

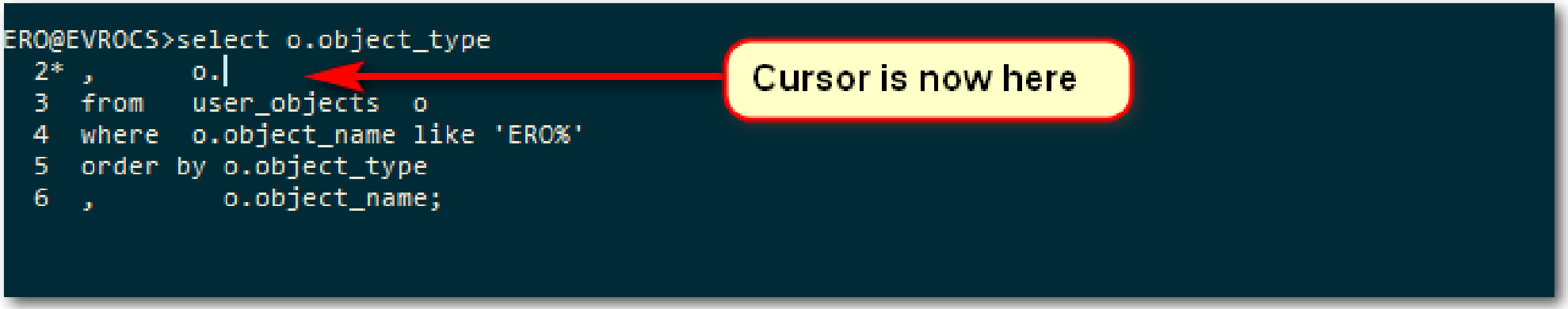
  Undocumented option: history save

Command Line Editing

command line editing

Edit **complete statements** (not just lines) without external editor

- Use up & down keys to get to statement of interest in history
- Use Left key to enter editing mode
- Modify statement however you want
- Hit CTRL-R to run the modified statement



command line editing, the motion picture

```
ERO@EVROCS>select count(*)
  2  from   user_objects
  3  where  object_name like 'ERO%'
  4  ;
  COUNT(*)
----------
        17


1 row selected.

ERO@EVROCS>
```


Code
For
ma
ting

Code Formatting

Format code using settings of SQL Developer formatter

```
ERO@EVROCS>select m.last_name, count(*) employees from MyEmp e join MyEmp m on e.manager_id=m.employee_id group by m.last_name order by 2 desc;
   LAST_NAME    EMPLOYEES
____________ ____________
King                   14
Fripp                   8
Vollman                 8
Mourgos                 8
Kaufling                8
```

```
ERO@EVROCS>format buffer
  1  SELECT
  2      m.last_name,
  3      COUNT(*) employees
  4  FROM
  5           myemp e
  6      JOIN myemp m ON e.manager_id = m.employee_id
  7  GROUP BY
  8      m.last_name
  9  ORDER BY
 10*     2 DESC;
ERO@EVROCS>
```

- Change formatter rules in SQL Developer
- **`Format rules <filename>`**
  loads a saved file with formatter rules
- **`Format buffer`**
  formats the statement in SQLcl buffer
- **`Format file <inputfile> <outputfile>`**
  formats contents of inputfile, writes to outputfile (inputfile = outputfile is allowed)

Formatting Query Output

Sqlformat

formats the **output** of all following sql statements

```
set sqlformat [format]
```

All formats are also available as comments in a query

Like:

```
select /*csv*/ * from user_tables;
```

Sqlformat - Formats

- csv — Output as comma separated values
- delimited — Like csv, but separator/delimiters can be defined
- xml — Output as xml
- json — Output as json
- json-formatted — Output as formatted json
- html — Output as html
- fixed — Fixed width columns, double-quoted values
- loader — Pipe (|) delimited values
- insert — Output as insert statements
- default — Clears any set sqlformat
- ansiconsole — Pretty output (see later)

Sqlformat – delimited

Returns character separated values.

Separator and delimiters can be defined in parameters

```
Set sqlformat delimited [delimiter [encloser-left [encloser-right]]]
```

- Delimiter can not be a non-printable character (like Tab)
- Enclosers can not be empty, defaults to double quote

```
Set sqlformat delimited * < >

    select 'A'  alias_a
    ,      'B'  alias_b
    ,      'C'  alias_c
    from   dual;
```

```
ERO@EVROCS>select 'A'  alias_a
  2  ,      'B'  alias_b
  3  ,      'C'  alias_c
  4* from   dual;


<ALIAS_A>*<ALIAS_B>*<ALIAS_C>
<A>*<B>*<C>

1 row selected.
```

Sqlformat – Ansiconsole

Output is formatted more readable

Column widths are based on actual data and headers

```
ERO@EVROCS>set sqlformat
SQL Format Cleared
ERO@EVROCS>select object_type,object_name from user_objects;

OBJECT_TYPE
-----------------------
OBJECT_NAME
--------------------------------------------------------------------
TABLE
MYEMP

TABLE
EMPLOYEES
```

```
ERO@EVROCS>set sqlformat ansiconsole
ERO@EVROCS>select object_type,object_name from user_objects;
    OBJECT_TYPE                          OBJECT_NAME
_______________ ____________________________________
TABLE           MYEMP
TABLE           EMPLOYEES
TABLE           EVROCS_SNAPSHOT_SOURCES
INDEX           SYS_IL0000091463C00005$$
LOB             SYS_LOB0000091463C00005$$
TABLE           MGMT$CM_SCHEMAMAP_GTT
TABLE           NESTED_TABLE
TABLE           COL1_TAB
INDEX           SYS_C0018342
TABLE           CUSTOMERS
TABLE           CUSTOMER_ADDRESSES
INDEX           SYS_C0018343
```

Sqlformat – Ansiconsole

Ugly edge case

- Data containing EndOfLine not handled correctly yet

```
ERO@EVROCS>select * from ero_eol;
   ID                              TEXT
_____ ___________________________________
    1 first row
    2 second row
contains an eol
    3 third row
```

Sqlformat – Ansiconsole Number format

Set a default format for your number columns

```
set sqlformat ansiconsole [format]
```

Format as in Java (URL of webpage with formats is in help text)

```
ERO@EVROCS>select object_id
  2  from   dba_objects
  3  where object_id is not null
  4  order by 1 desc
  5  fetch first 5 rows only
  6  ;
   OBJECT_ID
____________
      111695
      111694
      111507
      111506
      111505
```

```
ERO@EVROCS>set sqlformat ansiconsole ###,###,###
ERO@EVROCS>select object_id
  2  from   dba_objects
  3  where object_id is not null
  4  order by 1 desc
  5  fetch first 5 rows only
  6  ;
   OBJECT_ID
____________
      111,695
      111,694
      111,507
      111,506
      111,505
```

Sqlformat – Ansiconsole Highlight

Highlight data returned by queries

Define rules in a json file (in this case named highlight.json)

```
{"highlights":
  [
    {"type":"startWith" ,"test":"ERO"        ,"color":"INTENSITY_BOLD,YELLOW"},
    {"type":"endWith"   ,"test":"GTT"        ,"color":"BLUE"                  },
    {"type":"contains"  ,"test":"SNAPSHOT"   ,"color":"CYAN"                  },
    {"type":"exact"     ,"test":"CUSTOMERS"  ,"color":"GREEN"                 },
    {"type":"regex"     ,"test":".*[0-9].*"  ,"color":"MAGENTA"               }
  ]
}
```

Then Do

```
set sqlformat ansiconsole -config=highlight.json
```

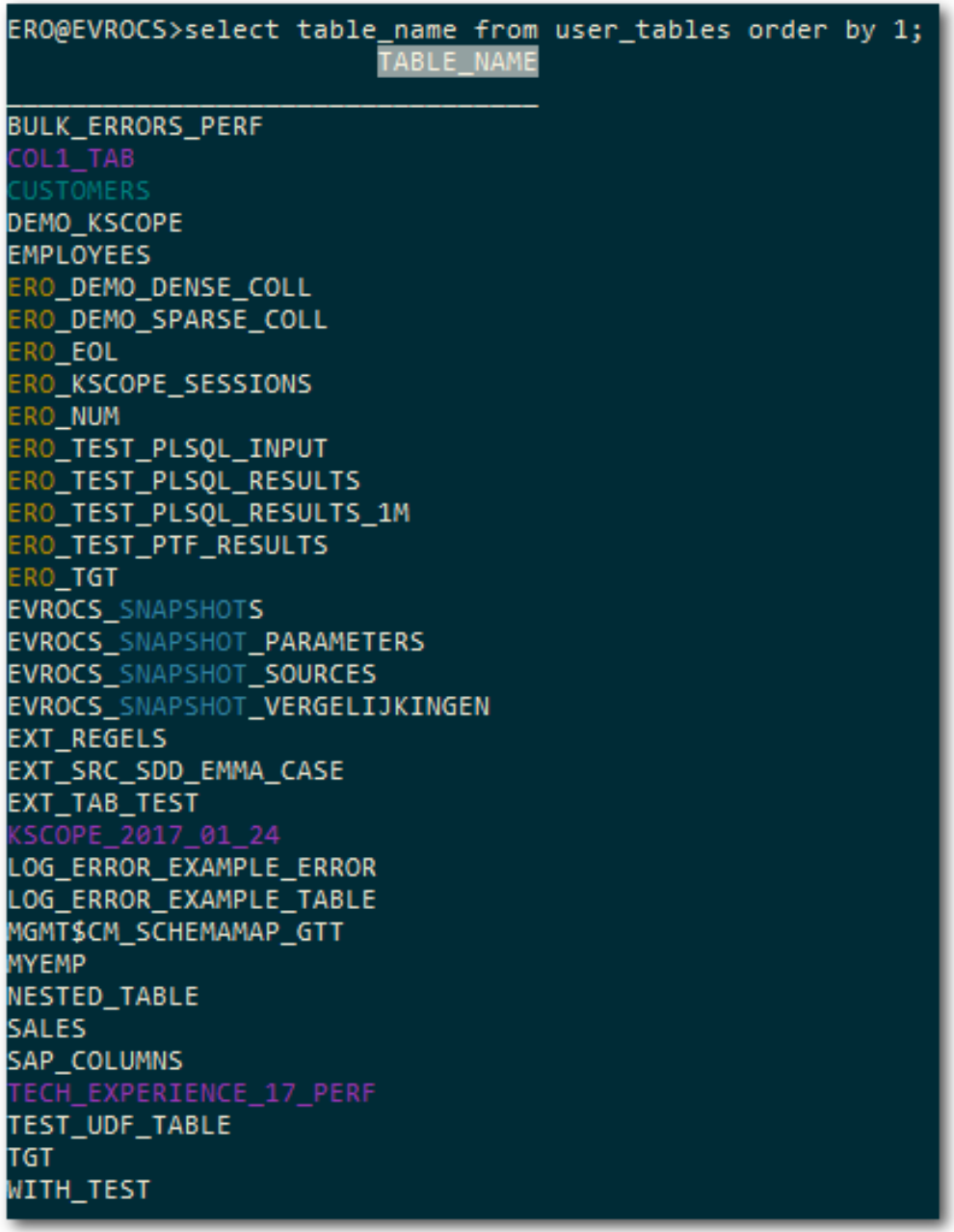
Sqlformat – Ansiconsole Data Highlight

Note

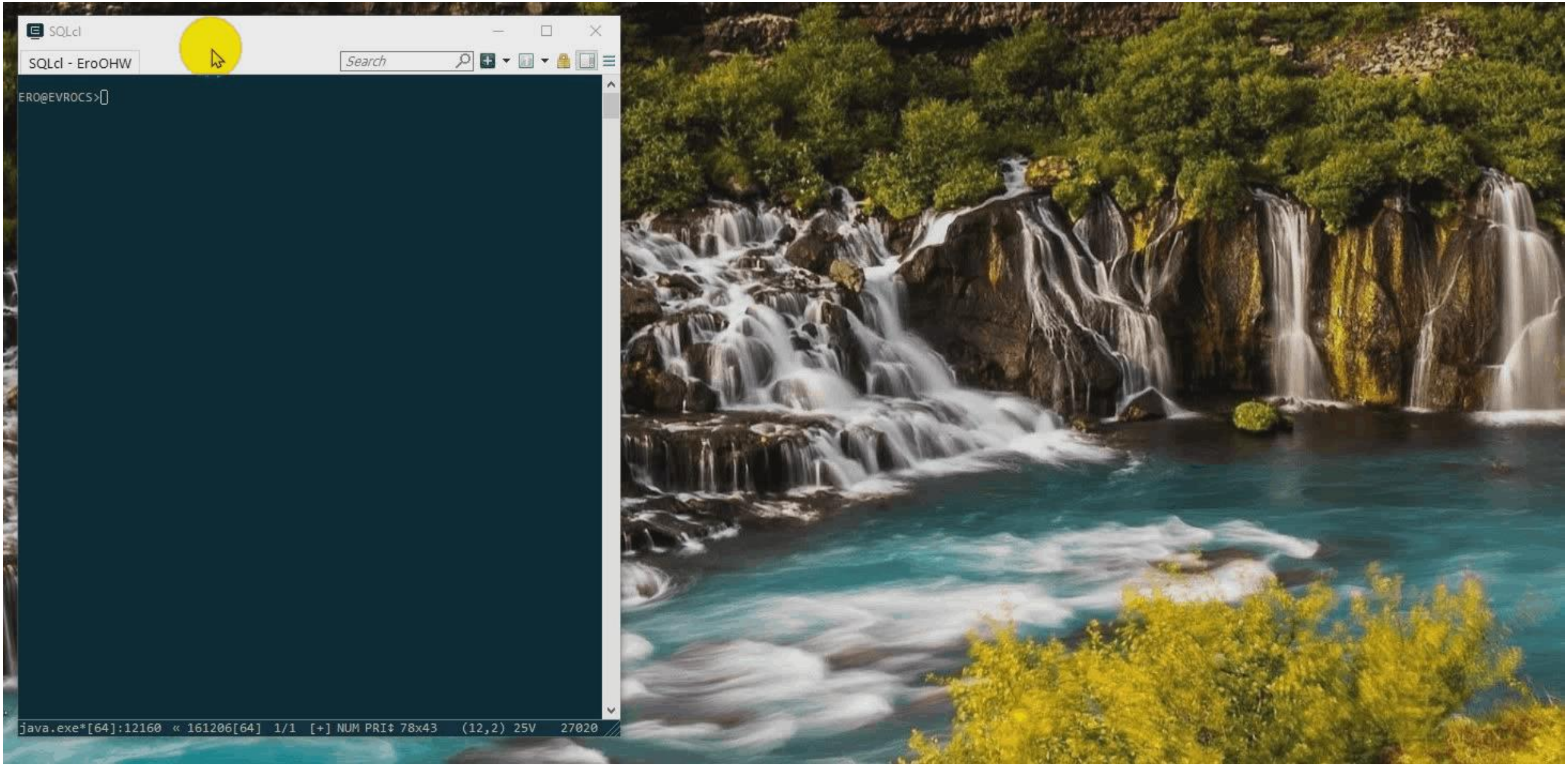
All tests only highlight the matching part,
except RegEx which highlights the entire string with a match

Note

The first rule that applies is used.
See "ero_test_plsql_results_1m"
1st and 5th rule apply, only 1st is used



Ansiconsole And Resizing Window




APEX Export
HYK LOGISTICS
msc

Apex

Lists all applications with

Either : `apex`

Or : `apex list`

```
SYS@ORCL>apex
```

WORKSPACE	APPLICATION_ID	APPLICATION_NAME	BUILD_STATUS	LAST_UPDATED_ON
INTERNAL	4411	Oracle APEX System Messages and Native Types	Run and Develop	2018-09-20 23:03:58
INTERNAL	4155	Scheme Authentication Login	Run and Develop	2018-09-20 23:03:58
COM.ORACLE.CUST.REPOSITORY	8842	Universal Theme Sample Application	Run and Develop	2018-08-06 08:55:17
COM.ORACLE.CUST.REPOSITORY	8851	Mobile Master Theme	Run and Develop	2018-02-12 07:04:53
INTERNAL	4000	Oracle APEX AppBuilder	Run and Develop	2018-10-15 16:15:04
INTERNAL	4020	Oracle APEX - Create App Wizard	Run and Develop	2018-10-15 16:15:04
INTERNAL	4350	Oracle APEX Workspace Administration	Run and Develop	2018-10-15 16:15:04
INTERNAL	4050	Oracle APEX Internal Administration	Run and Develop	2018-10-15 16:15:04
INTERNAL	4550	Oracle APEX Login	Run and Develop	2018-10-15 16:15:04
INTERNAL	4700	Oracle APEX New Service Signup	Run and Develop	2018-09-20 23:03:58
INTERNAL	4750	Oracle APEX Productivity and Sample Applications	Run and Develop	2018-10-15 16:15:04
INTERNAL	4500	Oracle APEX SQL Workshop	Run and Develop	2018-10-15 16:15:04
INTERNAL	4300	Oracle APEX Data Workshop	Run and Develop	2018-10-15 16:15:04
INTERNAL	4400	Oracle APEX Application Migration	Run and Develop	2018-10-15 16:15:04
INTERNAL	4900	Oracle APEX Websheets	Run and Develop	2018-09-20 23:03:58
INTERNAL	4800	Oracle APEX Team Development	Run and Develop	2018-10-15 16:15:04
INTERNAL	4850	Oracle APEX RESTful Services	Run and Develop	2018-10-15 16:15:04

Apex export

Help (used to) just say:

apex export <app id> - Exports the application which could be combined with spool for writing to a file

Then in 19.4 it says:

```
ERO@EVROCS>help apex
null
ERO@EVROCS>
```

In 20.2 We had a little help again:

export - Export APEX applications and or workspaces.

In 21.1 We get:

```
ERO@EVROCS>help apex
        Available options:
        list:           Lists applications installed in the database.
        log:            Reports APEX application usage from APEX_WORKSPACE_LOG_SUMMARY.
        export:         Export APEX applications and or workspaces.
```

But there's more to it than this….

In version <= 19.2 you get more help after typing in rubbish like

```
apex export 0 alkjsdhfkjh
```

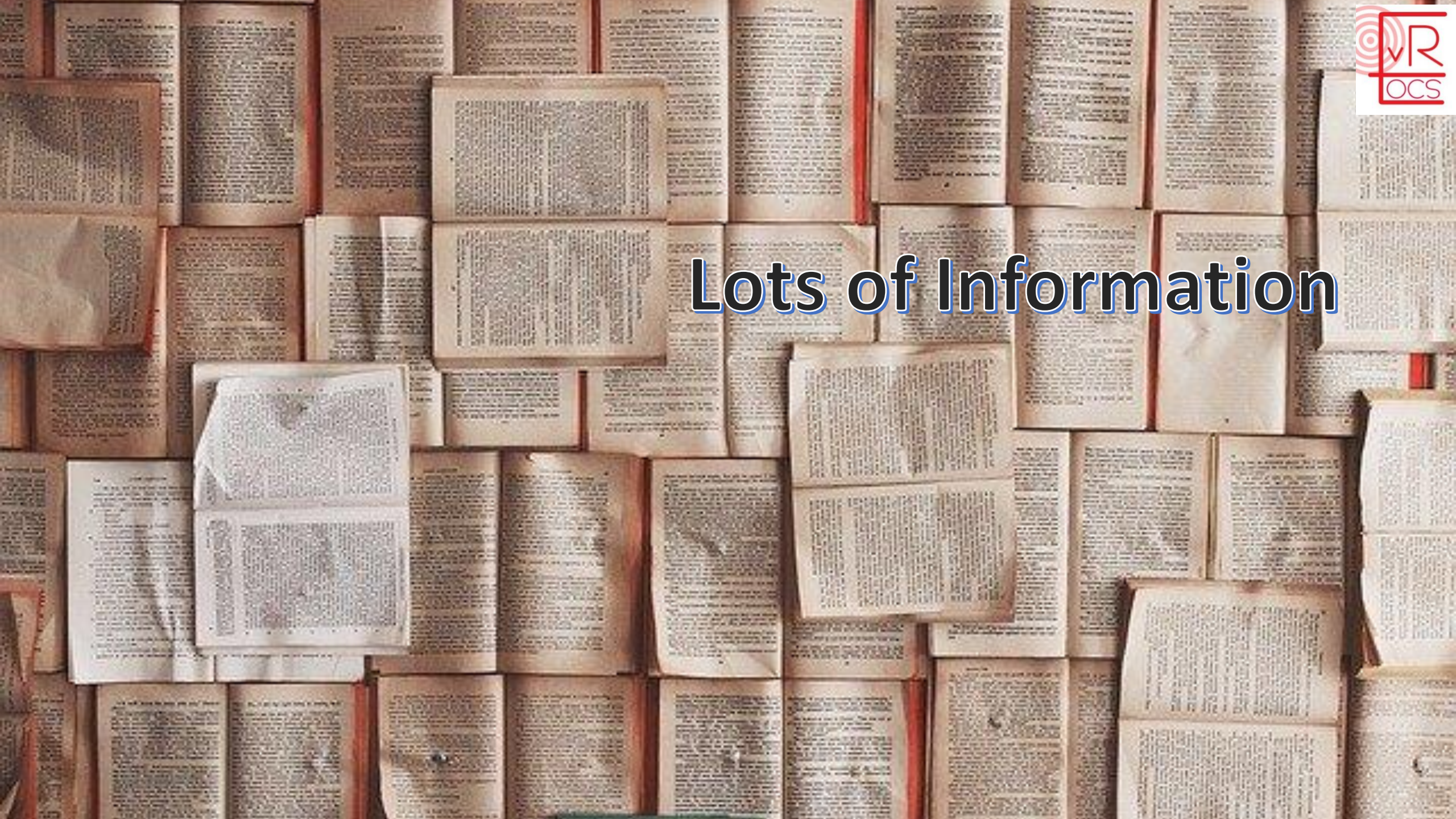
```
SYS@ORCL>apex export 0 alkjsdhfkjh
    -applicationid:    ID for application to be exported
    -workspaceid:      Workspace ID for which all applications to be exported or the workspace to be exported
    -instance:         Export all applications
    -expWorkspace:     Export workspace identified by -workspaceid or all workspaces if -workspaceid not specified
    -expMinimal:       Only export workspace definition, users, and groups
    -expFiles:         Export all workspace files identified by -workspaceid
    -skipExportDate:   Exclude export date from application export files
    -expPubReports:    Export all user saved public interactive reports
    -expSavedReports:  Export all user saved interactive reports
    -expIRNotif:       Export all interactive report notifications
    -expTranslations:  Export the translation mappings and all text from the translation repository
    -expFeedback:      Export team development feedback for all workspaces or identified by -workspaceid to development or deployment
    -expTeamdevdata:   Export team development data for all workspaces or identified by -workspaceid
    -deploymentSystem: Deployment system for exported feedback
    -expFeedbackSince: Export team development feedback since date in the format YYYYMMDD
    -expOriginalIds:   If specified, the application export will emit ids as they were when the application was imported

    -split:            Split the exported file
    -splitFlat:        Split with no directory strucure
    -splitUpdate:      Generate update.sql file while splitting
    -splitNoCheckSum:  Overwrite all files
 Application Example:           apex export -applicationid 31500
 Workspace Example:             apex export -workspaceid 9999
  Instance Example:              apex export -instance
 Export All Workspaces Example:  apex export -expWorkspace
 Export Feedback to development environment:
        apex export -workspaceid 9999 -expFeedback
 Export Feedback to deployment environment EA2 since 20100308:
        apex export -workspaceid 9999 -expFeedback -deploymentSystem EA2 -expFeedbackSince 20100308
```

In 19.4 this produces java-errors

In >= 20.2 it says "`APEX: Unable to process request. Verify specified options`"

But the options still work!!!


Lots of Information

Info(rmation)

Info / Information shows information about most object types

For a table it shows regular 'desc' information plus:

- table analysis information (last analyzed, row count, ...)
- table and column comments
- column default values
- indexes
- foreign key constraints referencing the table

Note: It is obviously slower than a regular DESC

Info(rmation)

```
ERO@EVROCS>info evrocs_snapshots
TABLE: EVROCS_SNAPSHOTS
         LAST ANALYZED:2016-06-07 01:58:20.0
         ROWS          :1
         SAMPLE SIZE   :1
         INMEMORY      :DISABLED
         COMMENTS      :

Columns
NAME                DATA TYPE             NULL  DEFAULT    COMMENTS
*ID                 NUMBER(9,0)           No
 SNAPSHOT_MOMENT    DATE                  No
 DATABASE_NAME      VARCHAR2(30 BYTE)     No
 OBJECT_OWNER       VARCHAR2(30 BYTE)     No
 OMSCHRIJVING       VARCHAR2(100 BYTE)    Yes
 OS_USER            VARCHAR2(30 BYTE)     No
 COMPUTERNAME       VARCHAR2(30 BYTE)     No

Indexes
            INDEX_NAME    UNIQUENESS    STATUS    FUNCIDX_STATUS                                    COLUMNS
_______________________ _____________ _________ _________________ ____________________________________________
ERO.EVROCS_SST_PK       UNIQUE        VALID                       ID
ERO.EVROCS_SST_UK1      UNIQUE        VALID                       SNAPSHOT_MOMENT, DATABASE_NAME, OBJECT_OWNER


References
                        TABLE_NAME       CONSTRAINT_NAME    DELETE_RULE     STATUS        DEFERRABLE    VALIDATED    GENERATED
_____________________________________ ____________________ _____________ _________ _________________ _____________ ____________
EVROCS_SNAPSHOT_PARAMETERS            EVROCS_SPR_SST_FK1    NO ACTION      ENABLED    NOT DEFERRABLE    VALIDATED    USER NAME
EVROCS_SNAPSHOT_SOURCES               EVROCS_SSE_SST_FK1    NO ACTION      ENABLED    NOT DEFERRABLE    VALIDATED    USER NAME
EVROCS_SNAPSHOT_VERGELIJKINGEN        EVROCS_SVG_SST_FK1    NO ACTION      ENABLED    NOT DEFERRABLE    VALIDATED    USER NAME
EVROCS_SNAPSHOT_VERGELIJKINGEN        EVROCS_SVG_SST_FK2    NO ACTION      ENABLED    NOT DEFERRABLE    VALIDATED    USER NAME
```

Info+

Info+ (currently there is no Information+)

When used for tables the column comments are replaced by some statistics

- low value
- high value
- number of distinct values
- type of histogram

Info+

```
ERO@EVROCS>info+ evrocs_snapshots
TABLE: EVROCS_SNAPSHOTS
         LAST ANALYZED:2016-06-07 01:58:20.0
         ROWS          :1
         SAMPLE SIZE   :1
         INMEMORY      :DISABLED
         COMMENTS      :

Columns
NAME                DATA TYPE             NULL  DEFAULT  LOW_VALUE              HIGH_VALUE             NUM_DISTINCT  HISTOGRAM
*ID                 NUMBER(9,0)           No             4001                   4001                   1             NONE
 SNAPSHOT_MOMENT    DATE                  No             2016.03.15.14.25.13    2016.03.15.14.25.13    1             NONE
 DATABASE_NAME      VARCHAR2(30 BYTE)     No             evrocs                 evrocs                 1             NONE
 OBJECT_OWNER       VARCHAR2(30 BYTE)     No             ero                    ero                    1             NONE
 OMSCHRIJVING       VARCHAR2(100 BYTE)    Yes            x                      x                      1             NONE
 OS_USER            VARCHAR2(30 BYTE)     No             Erik                   Erik                   1             NONE
 COMPUTERNAME       VARCHAR2(30 BYTE)     No             LAPTOP-EVROCS          LAPTOP-EVROCS          1             NONE

Indexes
          INDEX_NAME    UNIQUENESS    STATUS    FUNCIDX_STATUS  COLUMNS
_____________________ _____________ _________ _______________ ____________________________________________
ERO.EVROCS_SST_PK     UNIQUE        VALID                     ID
ERO.EVROCS_SST_UK1    UNIQUE        VALID                     SNAPSHOT_MOMENT, DATABASE_NAME, OBJECT_OWNER


References
                    TABLE_NAME       CONSTRAINT_NAME    DELETE_RULE     STATUS        DEFERRABLE     VALIDATED     GENERATED
_____________________________________ ____________________ ___________ _________ _________________ _____________ _____________
EVROCS_SNAPSHOT_PARAMETERS          EVROCS_SPR_SST_FK1    NO ACTION     ENABLED    NOT DEFERRABLE    VALIDATED    USER NAME
EVROCS_SNAPSHOT_SOURCES             EVROCS_SSE_SST_FK1    NO ACTION     ENABLED    NOT DEFERRABLE    VALIDATED    USER NAME
EVROCS_SNAPSHOT_VERGELIJKINGEN      EVROCS_SVG_SST_FK1    NO ACTION     ENABLED    NOT DEFERRABLE    VALIDATED    USER NAME
EVROCS_SNAPSHOT_VERGELIJKINGEN      EVROCS_SVG_SST_FK2    NO ACTION     ENABLED    NOT DEFERRABLE    VALIDATED    USER NAME
```

Aliases

Alias

Assign a user-defined name to sql, plsql and scripts

alias	show list of currently loaded aliases
alias **list**	same as "alias" but includes alias descriptions
alias **list MyAlias**	Shows the statement that MyAlias is an alias for
alias **MyAlias=(pl)sql**	Defines alias MyAlias for some sql or plsql
alias **MyAlias=@script**	Defines alias MyAlias for some sql script
alias **drop MyAlias**	Removes alias MyAlias from memory
alias **save [filename]**	Saves ALL currently loaded aliases to filename default file is %APPDATA%\sqlcl\aliases.xml
alias **load [filename]**	Loads aliases from filename default file is %APPDATA%\sqlcl\aliases.xml Appends to already loaded aliases, overwrites already existing ones

Unless '**save**' is used, alias is for the duration of the **SQLcl** session

Alias - Example

PLSQL block, including bind variables

```
ERO@EVROCS>alias anatab=begin
  2    dbms_stats.gather_table_stats
  3                (ownname           => coalesce(upper(:table_owner),user)
  4                ,tabname           => upper(:table_to_analyze)
  5                ,estimate_percent => dbms_stats.auto_sample_size
  6                ,method_opt        =>'FOR ALL COLUMNS SIZE AUTO'
  7                ,degree            => dbms_stats.default_degree
  8                ,cascade           => true
  9                );
 10    dbms_output.put_line ('Table analyzed');
 11  end;
 12  /
ERO@EVROCS>anatab ero employees
Table analyzed


PL/SQL procedure successfully completed.

ERO@EVROCS>
```

Alias - Case

Watch out: aliases are case-sensitive

```
ERO@EVROCS>alias hello=select 'hi there' from dual;
ERO@EVROCS>hello
   'HITHERE'
_______________
hi there


1 row selected.

ERO@EVROCS>HELLO
SP2-0042: unknown command "HELLO" - rest of line ignored.
ERO@EVROCS>
```

Alias – Host command

Even host commands can be aliased
Provided you end them with a semicolon

```
ERO@EVROCS>alias dirlist=$dir /o/a;
ERO@EVROCS>dirlist
 Volume in drive D is Document
 Volume Serial Number is 00C6-0170

 Directory of D:\EroOHW\Temp

06-05-19  22:21    <DIR>          .
06-05-19  22:21    <DIR>          ..
26-01-18  14:59         1.024.755 junk.LST
05-04-18  11:05             3.744 test.ctl
05-04-18  11:21            14.726 test.log
26-01-18  14:43               237 try.sql
               4 File(s)      1.043.462 bytes
               2 Dir(s)  15.332.974.592 bytes free

ERO@EVROCS>
```

Alias – Location of SQLcl

If SQLcl is in a path with spaces or 'special characters' like [and]
(**e.g.** c:\Program Files\SQLcl)

```
SQL> alias
2021-06-20 19:57:43.101 SEVERE oracle.dbtools.raptor.newscriptrunner.commands.alias.Aliases load_metrics C:\Program%20Files\sqlcl\lib\dbtools-common.jar (Het sys
teem kan het opgegeven pad niet vinden)
OsStat/Memory                  OsStat/Network                 OsStat/Processor               SgaInfo                        SysMetric/Long/Cache
SysMetric/Long/Cpu             SysMetric/Long/Cr              SysMetric/Long/DbBlock         SysMetric/Long/Enqueue         SysMetric/Long/Execute
SysMetric/Long/GlobalCache     SysMetric/Long/IndexScans      SysMetric/Long/Io              SysMetric/Long/LogicalReads    SysMetric/Long/Logon
SysMetric/Long/NodeSplits      SysMetric/Long/Parse           SysMetric/Long/PhysicalRead    SysMetric/Long/PhysicalReads   SysMetric/Long/PhysicalWrite
SysMetric/Long/Px              SysMetric/Long/Redo            SysMetric/Long/Service         SysMetric/Long/Session         SysMetric/Long/Sort
SysMetric/Long/TableScan       SysMetric/Long/User            SysMetric/Short/Consistent     SysMetric/Short/Cpu            SysMetric/Short/DbBlock
SysMetric/Short/Logon          SysMetric/Short/Memory         SysMetric/Short/Misc           SysMetric/Short/ParseScan      SysMetric/Short/ReadWrite
SysMetric/Short/User           SysStat/Cache/Commit           SysStat/Cache/Consistent       SysStat/Cache/DataWarehouse    SysStat/Cache/DbBlock
SysStat/Cache/FlashCache       SysStat/Cache/Flashback        SysStat/Cache/Lob              SysStat/Cache/Misc             SysStat/Cache/PhysicalRead
SysStat/Cache/PhysicalWrite    SysStat/Cache/PhysicalWrites   SysStat/Cache/Prefetch         SysStat/Cache/Recovery         SysStat/Cache/WriteClones
SysStat/Os                     SysStat/Rac/Cache/GcCr         SysStat/Rac/Cache/GcCurrent    SysStat/Rac/Cache/Other        SysStat/Rac/GlobalEnqueue
SysStat/Rac/Parallel           SysStat/Rac/User               SysStat/Redo/Blocks            SysStat/Redo/KbRead            SysStat/Redo/Misc
SysStat/Redo/Write             SysStat/Redo/Writes            SysStat/Sql/Cache              SysStat/Sql/Cell/Cus           SysStat/Sql/Cell/Num
SysStat/Sql/Cell/PhysicalIo    SysStat/Sql/Cell/SmartIo       SysStat/Sql/Hsc                SysStat/Sql/Index              SysStat/Sql/Misc
SysStat/Sql/Parse              SysStat/Sql/Sorts              SysStat/Sql/Table              SysStat/Sql/Workarea           SysStat/User/Commit
SysStat/User/JavaCall          SysStat/User/JavaSession       SysStat/User/Redaction         SysStat/User/Session           SysStat/User/Sql
SysStat/User/Time              SysStat/User/User              SysStat/User/Workload          SysTimeModel/Parse             SysTimeModel/Processor
SysTimeModel/Util              \!                             \?                             \c                             \cd
\dp                            \dt                            \dt+                           \e                             \echo
\i                             \o                             \p                             \prompt                        \q
\r                             \save                          \timing                        \w                             \z
cls                            ddl_constraint_inline          ddl_constraint_separate        ddl_nosegment                  ddl_segment
locks                          sessions                       tables                         tables2
SQL>
```

Error only on first call to Alias, and everything just works


Loading Data From CSV

LOAD

Easily load CSV files into tables

```
load ero.mytable@mylink d:\mypath\myfile.csv
```

By Default

Separator is a comma
Optionally enclosed, double quote
Processing in batches of 50 rows
Commit after every 10 batches
Terminates on > 50 errors

LOAD

Restrictions

Table must exist (Not necessarily anymore in version >= 21.3)

Names in header **must** match table column names

Names in header **must** be in uppercase (rather: must match case in repository)

Table columns that are not in the file **must be nullable**

Even when they have a default (and identity columns can't be nullable)

Options introduced in version 21.3

```
load tablename filename [option]
```

(No option)		- Table must exist and comply with header line.		File will be loaded.
new		- Create table statement displayed.	Table is created.	File will be loaded.
create	create_ddl	- Create table statement displayed.	Table is created.	File will **NOT** be loaded.
show	show_ddl	- Create table statement displayed.	Table is **NOT** created.	File will **NOT** be loaded.

LOAD - Example

```
ERO@EVROCS>desc ero_load
              Name        Null?            Type
_________________ ____________ _______________
FIRST_COLUMN                   VARCHAR2(1)
DEPARTMENT_NAME     NOT NULL   VARCHAR2(30)
MANAGER             NOT NULL   VARCHAR2(46)
SALARY              NOT NULL   NUMBER(8,2)
CITY                NOT NULL   VARCHAR2(30)
EXTRA_COLUMN                   VARCHAR2(1)

ERO@EVROCS>$type ero_load_data.csv
DEPARTMENT_NAME,MANAGER,SALARY,CITY
"Administration","Jennifer Whalen","4400","Seattle"
"Marketing","Michael Hartstein","13000","Toronto"
"Human Resources","Susan Mavris","6500","London"
"Public Relations","Hermann Baer","10000","Munich"

ERO@EVROCS>select * from ero_load;


0 rows selected.

ERO@EVROCS>load ero_load ero_load_data.csv
--Number of rows processed: 4
--Number of rows in error: 0
0 - SUCCESS: Load processed without errors
ERO@EVROCS>select * from ero_load;
   FIRST_COLUMN     DEPARTMENT_NAME                MANAGER    SALARY       CITY    EXTRA_COLUMN
_______________ _____________________ _____________________ _________ __________ _______________
                Administration        Jennifer Whalen          4400 Seattle
                Marketing             Michael Hartstein       13000 Toronto
                Human Resources       Susan Mavris             6500 London
                Public Relations      Hermann Baer            10000 Munich


4 rows selected.
```

Tweaking the LOAD statement

Two variables control how LOAD works

LOAD

Changes the way the load is performed

LOADFORMAT

Changes the accepted format of the inputfile

Variable "LOAD"

```
SET LOAD default
SET LOAD <<options>>
```

Options that can be set

batch_rows {###}
batches_per_commit {###}
commit on / commit off
truncate on / truncate off
errors {###} / errors unlimited
locale {language} {country}
method insert
date_format {format}
timestamp_format {format}
timestamptz_format {format}

Synonyms

- All underscores can be left out
- The ones with a blue part can be abreviated to just that part

Default

- Returns to the default settings

IMHO

- The *format options should have been implemented with the LOADFORMAT variable

Commit

- Is set to ON if batches_per_commit > 0
  So no longer depending on setting autocommit

Variable "LOAD" New in 21.3

Options that can be set

clean_names

Cleanup of column names in header row. Various strategies available, see help

column_size [actual | round | max]

Column sizing strategy for table to be created based on file analysis

map_column_names

Comma separated list of manual column mappings (file-column=table-column)

scan_rows

Number of lines to scan for analysis, max=5000, default=100

unknown_columns_fail on/off

Whether to fail on unmapped columns in file, or just ignore them

Variable "LOADFORMAT"

```
SET LOADFORMAT default
SET LOADFORMAT <<fileformat>> <<options>>
```

Options that can be set

column_**names** on / column_**names** off
delimiter {chrs}
enclosures off / enclosures {chr1}[{chr2}]
enclosure_**left** off / enclosure_**left** {chrs}
enclosure_**right** off / enclosure_**right** {chrs}
encoding off / encoding {encoding}
row_**limit** off / row_**limit** {###}
skip_rows off / **skip**_rows {###}
skip_**after**_names / skip_**before**_names
row_**termi**nator {chr} / row_**termi**nator ""

File Formats

Both load and unload
csv | delimited
Unload only
html insert json json-formatted
loader t2 xml

Synonyms

- "csv" and "delimited" are synonyms!
- All underscores can be left out
- The ones with a blue part can be abreviated to just that part

Default

- Returns to the default settings

Variable "LOADFORMAT"

Setting delimiter to semicolon => ;

```
ERO@EVROCS>set loadformat default
Load Format Cleared
ERO@EVROCS>show loadformat
default

format CSV
column_names on
delimiter ,
enclosure_left "
enclosure_right "
encoding UTF8
row_limit off
row_terminator
skip_rows 0
skip_after_names
ERO@EVROCS>set loadformat delimiter ;;
ERO@EVROCS>show loadformat

format CSV
column_names on
delimiter ;
enclosure_left "
enclosure_right "
encoding UTF8
row_limit off
row_terminator
skip_rows 0
skip_after_names
ERO@EVROCS>
```

+ <CTRL-R>

The semicolon needs to be escaped by typing two semicolons

```
set LOADFORMAT delimiter ;;
```

NOTE

- Works if it's in a script that is executed
- Works if typed at command line and executed with <CTRL-R>
- Does not work if typed at command line and executed with <ENTER>

```
ERO@EVROCS>set loadformat default
Load Format Cleared
ERO@EVROCS>show loadformat
default

format CSV
column_names on
delimiter ,
enclosure_left "
enclosure_right "
encoding UTF8
row_limit off
row_terminator
skip_rows 0
skip_after_names
ERO@EVROCS>set loadformat delimiter ;;
A value is required for option delimiter.


ERO@EVROCS>show loadformat
default

format CSV
column_names on
delimiter ,
enclosure_left "
enclosure_right "
encoding UTF8
row_limit off
row_terminator
skip_rows 0
skip_after_names
ERO@EVROCS>
```

+ <Enter>

Fixed in version 21.3

UNLOAD

Easily unload tables into CSV files

```
load ero.mytable@mylink dir d:\mypath
```

Also controlled with LOADFORMAT variable
Filename will be <table name>_DATA_TABLE.csv

```
ERO@EVROCS>desc ero_tst

   Name     Null?          Type
_______ _________ _____________
ONE               NUMBER
TWO               NUMBER(38)
ERO@EVROCS>
```

```
ERO@EVROCS>unload ero_tst dir D:\EroScripts\Data

format csv

column_names on
delimiter ,
enclosure_left "
enclosure_right "
encoding UTF8
row_terminator default

** UNLOAD Start ** at 2021.06.20-20.19.34
Export Separate Files to D:\EroScripts\Data
DATA TABLE ERO_TST
File Name: D:\EroScripts\Data\ERO_TST_DATA_TABLE.csv
Number of Rows Exported: 4
** UNLOAD End ** at 2021.06.20-20.19.34
ERO@EVROCS>
```

How about views / queries??

Nope!
Just tables.

```
ERO@EVROCS>$type D:\EroScripts\Data\ERO_TST_DATA_TABLE.csv
"ONE","TWO"
1,1001
2,1002
3,1003
4,1004

ERO@EVROCS>
```


The little things are important too

Working directory

PWD (Print Working Directory)

- displays the current working directory, like the linux/unix pwd command
- saves user one keystroke (pwd instead of !pwd or $dir)
- saves windows users a lot of unwanted output compared to $dir

CD (Change Directory)

- changes the working directory **while inside SQLcl**
- Even to other disk,
  without the usual dos /d parameter
- No more need to exit,
  change directory and start again

```
ERO@EVROCS>pwd
D:\EroOHW\Temp
ERO@EVROCS>cd e:\sqlcl\bin
ERO@EVROCS>pwd
E:\sqlcl\bin
ERO@EVROCS>cd ..\lib
ERO@EVROCS>pwd
E:\sqlcl\lib
ERO@EVROCS>
```

Files

Find filename

Is supposed to list all occurrences of filename in the SQLPATH

But lists all in:

- SQLPATH + all subdirs
- Working dir + all subdirs

No wildcards allowed

```
ERO@EVROCS>show sqlpath
SQLPATH : D:\EroOHW\;.;D:\EroScripts\Tools\Login_Sql1;D:\EroScripts;D:\Ero
Scripts\Tools;D:\EroScripts\ProjectTools;D:\EroScripts\Connect;D:\EroScript
s\Cloning
ERO@EVROCS>
ERO@EVROCS>pwd
D:\EroOHW\
ERO@EVROCS>
ERO@EVROCS>find comp.sql
D:\EroScripts\Tools\comp.sql
D:\EroScripts\IAS\comp.sql
ERO@EVROCS>
```

Which filename

displays the pathname of the script that will be executed if called with @filename

```
ERO@EVROCS>which comp.sql
D:\EroScripts\Tools\comp.sql
ERO@EVROCS>
```

SQL_ID for every query

set feedback on sql_id

Turns on reporting of the SQL ID of the just executed query

```
ERO@EVROCS>set feedback on sql_id
ERO@EVROCS>select count(*) from dba_objects;

  COUNT(*)
----------
     73031

1 row selected.

SQL_ID: g4pkmrqrgxg3b
ERO@EVROCS>
```

set feedback on

Turns the reporting of SQL ID off

Spool to zip file

If you spool to a filename with *.zip extension

SQLcl will create the zip automatically

Spool To	Then Datafile	In Zipfile
mytest.zip	mytest.out	mytest.zip
mytest.json.zip	mytest.json	mytest.json.zip
mytestzip.json.zip	mytes.json	mytestzip.json.zip

OERR

ORA-24381

What is that error? Let's google it.

Or……..

```
SQL> oerr ora 24381

24381. 00000 -  "error(s) in array DML"
*Cause:    One or more rows failed in the DML.
*Action:   Refer to the error stack in the error handle.
SQL>
```

Only works for ORA and TNS message codes

Codescan

```
create or replace procedure tst_codescan
(p_userinput   in  varchar2
)
is
  t_result  varchar2(100);
begin   -- tst_codescan
  execute immediate 'select dymmy '||chr(10)||
                    'from   dual'  ||chr(10)||
                    'where  ''A'' = '||p_userinput
  into t_result
  ;
  dbms_output.put_line ('result = '||t_result);
end tst_codescan;
/
```

SQL Injection vulnerability!

Set codescan

And you will be **warned** about it:

```
ERO@EVROCS>set codescan all

ERO@EVROCS>create or replace procedure tst_codescan
  2  (p_userinput   in  varchar2
  3  )
  4  is
  5    t_result  varchar2(100);
  6  begin   -- tst_codescan
  7    execute immediate 'select dymmy '||chr(10)||
  8                      'from   dual'  ||chr(10)||
  9                      'where  ''A'' = '||p_userinput
 10    into t_result
 11    ;
 12    dbms_output.put_line ('result = '||t_result);
 13  end tst_codescan;
 14  /

SQLcl security warning: SQL injection P_USERINPUT line 2 -> P_USERINPUT line 7


Procedure TST_CODESCAN compiled

ERO@EVROCS>
```

Turn the warnings on with either of these:

```
set codescan all
set codescan sqlinjection on
```

Turn them off with either of these:

```
set codescan none
set codescan sqlinjection off
```

Data Definition Language

DDL

Basically just shorthand for dbms_metadata.get_ddl

```
ddl object_name [object_type] [save-filename]
```

Behavior can be altered with dbms_metadata like:

```
exec dbms_metadata.set_transform_param
  (dbms_metadata.session_transform, 'CONSTRAINTS_AS_ALTER', true);

exec dbms_metadata.set_transform_param
  (dbms_metadata.session_transform, 'SEGMENT_ATTRIBUTES'  , false);
```

If used often: make aliases for such transforms.

DDL

```
ERO@EVROCS>ddl_nosegatt

PL/SQL procedure successfully completed.

ERO@EVROCS>ddl ero_num

  CREATE TABLE "ERO"."ERO_NUM"
   (    "N_FLOAT" NUMBER,
        "N_REAL" NUMBER
   )  DEFAULT COLLATION "USING_NLS_COMP" ;
ERO@EVROCS>
```

DDL

Not ***just*** dbms_metadata.get_ddl

For a table, what gets executed is

So,

- The table
- Its comments
- Its column-comments
- Its indexes (except if linked to a constraint)
- Its triggers

```
select DBMS_METADATA.GET_DDL('TABLE',OBJECT_NAME,OWNER)
FROM   all_objects
where  owner = :OWNER
and    object_name = :NAME
and    object_type = 'TABLE'
union all
select dbms_metadata.GET_DEPENDENT_DDL ('COMMENT', TABLE_NAME, OWNER )
FROM   (select table_name
        ,      owner
        from   all_col_comments
        where  owner      = :OWNER
          and  table_name = :NAME
          and  comments is not null
        union
        select table_name
        ,      owner
        from   sys.all_TAB_comments
        where  owner = :OWNER
          and  table_name = :NAME
          and  comments is not null
       )
union all
select DBMS_METADATA.GET_DDL('INDEX',INDEX_NAME, OWNER)
FROM   (select index_name
        ,      owner
        from   sys.all_indexes
        where  table_owner = :OWNER
          and  table_name = :NAME
          and  generated = 'N'
        minus
        select index_name
        ,      owner
        from   sys.all_constraints
        where  owner = :OWNER
          and  table_name = :NAME
       )
union all
select dbms_metadata.GET_DDL ('TRIGGER', trigger_name ,owner )
from   all_triggers
where  table_owner = :OWNER
  and  table_name = :NAME
;
```


Bridge

Bridge

Bridge gets data from a remote connection
into a local table without needing a database link

Somewhat similar to SqlPlus command "copy"

```
bridge targettable as "jdbcURL" (query)[modifier];
```

jdbcURL :
jdbc:oracle:thin:username/password@host:port/servicename

e.g. : **jdbc:oracle:thin:ero/ero@localhost:1521/evrocs**

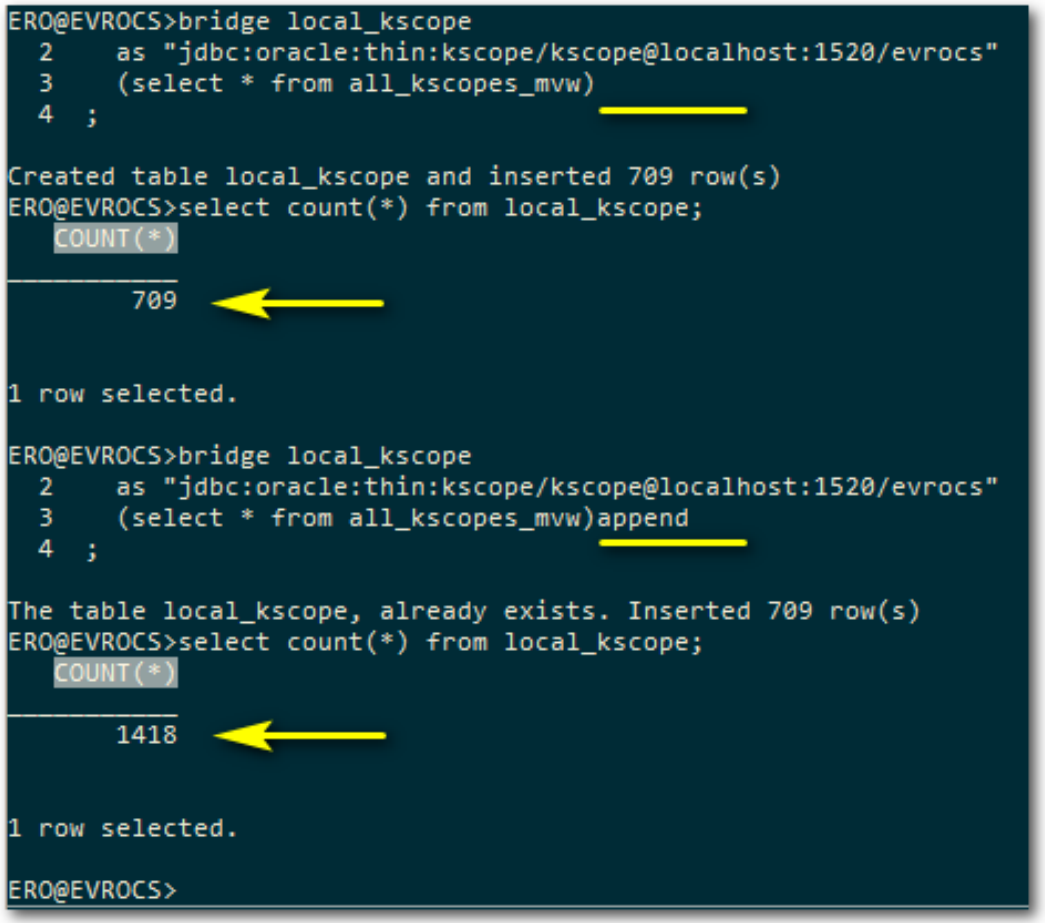
By default targettable is created.

Bridge – modifier

The modifiers are not in the help yet!!!

	(= no modifier) Always do create target table. Ora-00955 if it already exists
replace	Drop target table if it exists, then re-create it and add the data
append	Add the data of the query to possibly existing data in target table
truncate	Truncate the target table before adding the data
skip	If table exists: it does nothing (reports "table already exists, inserted 0 rows") If table doesn't exist: will create it and add the data

Modifier HAS (!!) to be directly after closing bracket after query



Repeat

Repeat

Repeat runs the statement that's in the buffer

For a requested number of times

With a requested wait time between executions

```
repeat number_of_times seconds_between_executions
```

The wait time maximum is 120 seconds

Repeat – The Motion Picture

```
ERO@EVROCS>select username,status,action from v$session where sid = 257;
```


Liquibase

Liquibase (LB)

Available since version 19.2: Liquibase support

Liquibase: Tool for change management of database schemas

Only in standalone version, not the one included in Sql Developer


But what if you want more... ?

SQLcl Scripting with nashorn

If you miss any functionality….

….you can just build it yourself

1. The Nashorn JavaScript engine is included in Java
1. SQLcl is built in Java

1 + 1 =

You can build your own functionality in JavaScript and make use of the Java libraries.

!! Warning !!

Version 21.2.*

Issue that makes running javascript in some setups impossible

Nothing happens and "!ScriptCommand.1!" is outputted

Version 21.3

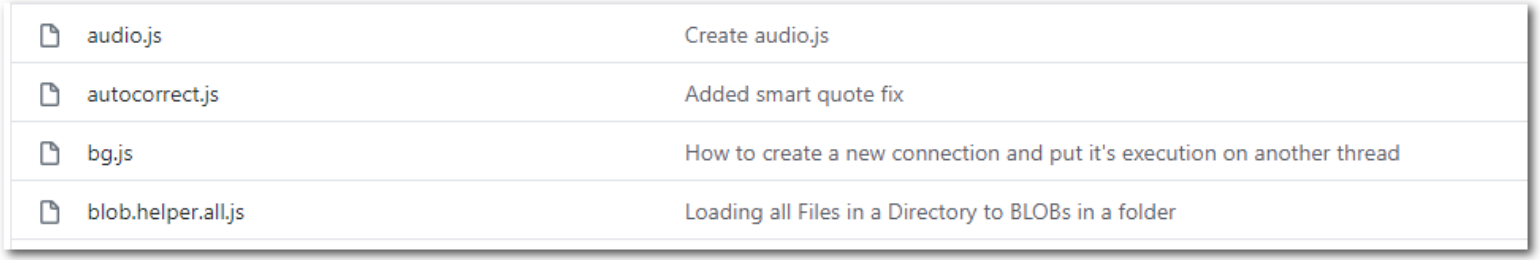
Fixed !!

Scripts to start with

Learning how to build JavaScripts for SQLcl is beyond the scope of this presentation

But
Lots of example scripts online e.g.

Kris Rice
https://github.com/oracle/oracle-db-tools/tree/master/sqlcl/examples
Menno Hoogendijk
https://github.com/mennooo/SQLcl



audio.js	Create audio.js
autocorrect.js	Added smart quote fix
bg.js	How to create a new connection and put it's execution on another thread
blob.helper.all.js	Loading all Files in a Directory to BLOBs in a folder

And, my blog has a couple of long articles to help take the first steps

(Very) Basic intro to scripting : https://bit.ly/2SPkQnK
Building file2bind script : https://bit.ly/34SlkyK
Creating a custom command : https://bit.ly/3drl4L9
Unregistering a custom command : https://bit.ly/35HOT7f
Library to make (un)registering of custom commands easier : https://bit.ly/3xzOQ9l

Home | Events | Resume | Quotes | Contact | EScope | Downloads | Blog

Home→SQLcl→*When great just isn't good enough*

When great just isn't good enough

2020-08-19 Erik van Roon

This post is the first one of three that together are an introduction to scripting in SQLcl.
Also read part 2 and part 3.

For many years now SQLcl is my favorite command line tool to connect to Oracle databases. (available at the bottom of the SQL Developer download page)
It does (practically) everything SQL*Plus does and a whole lot more. It has many capabilities that SQL*Plus just doesn't have.
But this post is not about those goodies SQLcl delivers, it's more or less about what is doesn't....

How to execute JavaScripts

Option 1, at command prompt:

- Type: script <<Enter>>
- Type or paste the JavaScript
- Type: / <<Enter>>

```
ERO@EVROCS>script
  2  var today = new Date();
  3  var day = today.getDay();
  4
  5  if ((day == 0) || (day == 6)) {
  6    ctx.write ("It's weekend, go get a beer!\n");
  7  } else {
  8    ctx.write ("It's a workday, go get a beer!\n");
  9  }
 10  /
It's a workday, go get a beer!
ERO@EVROCS>
```

Simple JavaScript

```
var today = new Date();
var day = today.getDay();

if ((day == 0) || (day == 6)) {
  ctx.write ("It's weekend, go get a beer!\n");
} else {
  ctx.write ("It's a workday, go get a beer!\n");
}
```

Option 2, from a script file:

- Save the JavaScript to a file
- Execute with: script filename.js

```
ERO@EVROCS>script silly.js
It's a workday, go get a beer!
ERO@EVROCS>
```

How useful can this be?

Very!

```
ERO@EVROCS>$type ero_load_data.csv
```

Example (from the blog)

script file2bind

Accepts a filename and bindvariable name as parameters
Uses Java class **FileReader** to access client files
Reads a client file, places its contents in a clob bind variable
Now the filecontents is available for your (PL)SQL

SQLcl has nothing like that 'out of the box'

Using scripting the script can also be registered as a new command
So, no need for : **script my_script_file.js**
Just : **my_command**

How useful can this be?

Very!

Example (from the blog)

script file2bind

Accepts a filename and bindvariable name as parameters
Uses Java class **FileReader** to access client files
Reads a client file, places its contents in a clob bind variable
Now the filecontents is available for your (PL)SQL

SQLcl has nothing like that 'out of the box'

Using scripting the script can also be registered as a new command
So, no need for : **script my_script_file.js**
Just : **my_command**

```
ERO@EVROCS>$type ero_load_data.csv
DEPARTMENT_NAME;MANAGER;SALARY;CITY
"Administration";"Jennifer Whalen";"4400";"Seattle"
"Marketing";"Michael Hartstein";"13000";"Toronto"
"Human Resources";"Susan Mavris";"6500";"London"
"Public Relations";"Hermann Baer";"10000";"Munich"

ERO@EVROCS>
```

How useful can this be?

Very!

Example (from the blog)

script file2bind

Accepts a filename and bindvariable name as parameters
Uses Java class **FileReader** to access client files
Reads a client file, places its contents in a clob bind variable
Now the filecontents is available for your (PL)SQL

SQLcl has nothing like that 'out of the box'

Using scripting the script can also be registered as a new command
So, no need for : **script my_script_file.js**
Just : **my_command**

```
ERO@EVROCS>$type ero_load_data.csv
DEPARTMENT_NAME;MANAGER;SALARY;CITY
"Administration";"Jennifer Whalen";"4400";"Seattle"
"Marketing";"Michael Hartstein";"13000";"Toronto"
"Human Resources";"Susan Mavris";"6500";"London"
"Public Relations";"Hermann Baer";"10000";"Munich"

ERO@EVROCS>
ERO@EVROCS>print my_data
```

How useful can this be?

Very!

Example (from the blog)

script file2bind

Accepts a filename and bindvariable name as parameters
Uses Java class **FileReader** to access client files
Reads a client file, places its contents in a clob bind variable
Now the filecontents is available for your (PL)SQL

```
ERO@EVROCS>$type ero_load_data.csv
DEPARTMENT_NAME;MANAGER;SALARY;CITY
"Administration";"Jennifer Whalen";"4400";"Seattle"
"Marketing";"Michael Hartstein";"13000";"Toronto"
"Human Resources";"Susan Mavris";"6500";"London"
"Public Relations";"Hermann Baer";"10000";"Munich"

ERO@EVROCS>
ERO@EVROCS>print my_data

SP2-0552: Bind variable "MY_DATA" not declared.
ERO@EVROCS>
```

SQLcl has nothing like that 'out of the box'

Using scripting the script can also be registered as a new command
So, no need for : **script my_script_file.js**
Just : **my_command**

How useful can this be?

Very!

Example (from the blog)

script file2bind

Accepts a filename and bindvariable name as parameters
Uses Java class **FileReader** to access client files
Reads a client file, places its contents in a clob bind variable
Now the filecontents is available for your (PL)SQL

SQLcl has nothing like that 'out of the box'

Using scripting the script can also be registered as a new command
So, no need for : **script my_script_file.js**
Just : **my_command**

```
ERO@EVROCS>$type ero_load_data.csv
DEPARTMENT_NAME;MANAGER;SALARY;CITY
"Administration";"Jennifer Whalen";"4400";"Seattle"
"Marketing";"Michael Hartstein";"13000";"Toronto"
"Human Resources";"Susan Mavris";"6500";"London"
"Public Relations";"Hermann Baer";"10000";"Munich"

ERO@EVROCS>
ERO@EVROCS>print my_data

SP2-0552: Bind variable "MY_DATA" not declared.
ERO@EVROCS>
ERO@EVROCS>file2bind ero_load_data.csv my_data
```

How useful can this be?

Very!

Example (from the blog)

script file2bind

Accepts a filename and bindvariable name as parameters
Uses Java class **FileReader** to access client files
Reads a client file, places its contents in a clob bind variable
Now the filecontents is available for your (PL)SQL

SQLcl has nothing like that 'out of the box'

Using scripting the script can also be registered as a new command
So, no need for : **script my_script_file.js**
Just : **my_command**

```
ERO@EVROCS>$type ero_load_data.csv
DEPARTMENT_NAME;MANAGER;SALARY;CITY
"Administration";"Jennifer Whalen";"4400";"Seattle"
"Marketing";"Michael Hartstein";"13000";"Toronto"
"Human Resources";"Susan Mavris";"6500";"London"
"Public Relations";"Hermann Baer";"10000";"Munich"

ERO@EVROCS>
ERO@EVROCS>print my_data

SP2-0552: Bind variable "MY_DATA" not declared.
ERO@EVROCS>
ERO@EVROCS>file2bind ero_load_data.csv my_data
ERO@EVROCS>
```

How useful can this be?

Very!

Example (from the blog)

script file2bind

Accepts a filename and bindvariable name as parameters
Uses Java class **FileReader** to access client files
Reads a client file, places its contents in a clob bind variable
Now the filecontents is available for your (PL)SQL

SQLcl has nothing like that 'out of the box'

Using scripting the script can also be registered as a new command
So, no need for : **script my_script_file.js**
Just : **my_command**

```
ERO@EVROCS>$type ero_load_data.csv
DEPARTMENT_NAME;MANAGER;SALARY;CITY
"Administration";"Jennifer Whalen";"4400";"Seattle"
"Marketing";"Michael Hartstein";"13000";"Toronto"
"Human Resources";"Susan Mavris";"6500";"London"
"Public Relations";"Hermann Baer";"10000";"Munich"

ERO@EVROCS>
ERO@EVROCS>print my_data

SP2-0552: Bind variable "MY_DATA" not declared.
ERO@EVROCS>
ERO@EVROCS>file2bind ero_load_data.csv my_data
ERO@EVROCS>
ERO@EVROCS>print my_data
```

How useful can this be?

Very!

Example (from the blog)

script file2bind

Accepts a filename and bindvariable name as parameters
Uses Java class **FileReader** to access client files
Reads a client file, places its contents in a clob bind variable
Now the filecontents is available for your (PL)SQL

SQLcl has nothing like that 'out of the box'

Using scripting the script can also be registered as a new command
So, no need for : **script my_script_file.js**
Just : **my_command**

```
ERO@EVROCS>$type ero_load_data.csv
DEPARTMENT_NAME;MANAGER;SALARY;CITY
"Administration";"Jennifer Whalen";"4400";"Seattle"
"Marketing";"Michael Hartstein";"13000";"Toronto"
"Human Resources";"Susan Mavris";"6500";"London"
"Public Relations";"Hermann Baer";"10000";"Munich"

ERO@EVROCS>
ERO@EVROCS>print my_data

SP2-0552: Bind variable "MY_DATA" not declared.
ERO@EVROCS>
ERO@EVROCS>file2bind ero_load_data.csv my_data
ERO@EVROCS>
ERO@EVROCS>print my_data

MY_DATA
--------------------------------------------------------------------------------
DEPARTMENT_NAME;MANAGER;SALARY;CITY
"Administration";"Jennifer Whalen";"4400";"Seattle"
"Marketing";"Michael Hartstein";"13000";"Toronto"
"Human Resources";"Susan Mavris";"6500";"London"
"Public Relations";"Hermann Baer";"10000";"Munich"


ERO@EVROCS>
```

Tips

TIP – login.sql

SQLcl uses same SQLPATH environment variable as SqlPlus

So

- will execute the same (g)login.sql

So what?

- **sqlcl-specific** stuff in login.sql will cause errors when starting **SqlPlus**
- **sqlplus-specific** stuff in login.sql that is not (yet/fully) supported in sqlcl will cause errors when starting **sqlcl**

But: easily fixed….

Fix common SQLPATH

1. All common settings in **x:\login\common_login.sql**

2. Create x:\login**PLUS****login.sql**
   - All **SQLPLUS** specific settings
   - @@..\common_login.sql

3. Create x:\login**CL****login.sql**
   - All **SQLCL** specific settings
   - @@..\common_login.sql

4. Create batchfiles sp.bat & sc.bat
   - Add the login directory to SQLPATH
   - Run the executable passing all parameters

Fix common SQLPATH

x:\login\common_login.sql

```
-- ************************************************************
-- Funktion    : Initial settings for sql*plus and sqlcl
-- ************************************************************

@@ login_variables.sql
@@ MySettings

-- Show evrocs banner
@@ evrocs.sql

prompt
-- Show product versions for current database
@@ version
-- Show startup date/time of the instance
@@ startup
-- Show identification of the session
@@ session
prompt
```

x:\login**PLUS****login.sql**

```
@@ ..\login_common.sql

define_editor="notepad.exe"
```

x:\login**CL****login.sql**

```
@@ ..\login_common.sql

set sqlformat ansiconsole
set classic on
```

sp.bat

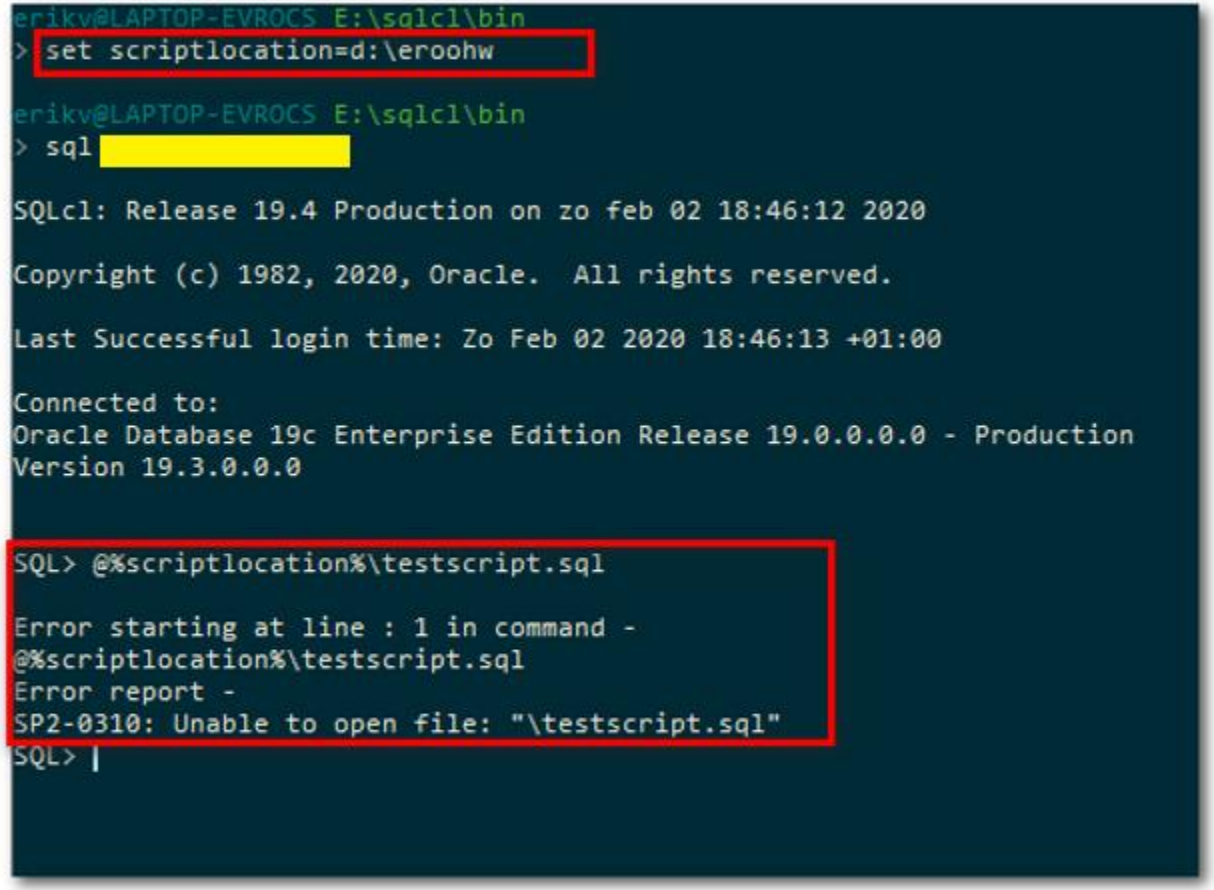
```
@echo off
set sqlpath=x:\login\plus;%sqlpath%
sqlplus.exe %*
```

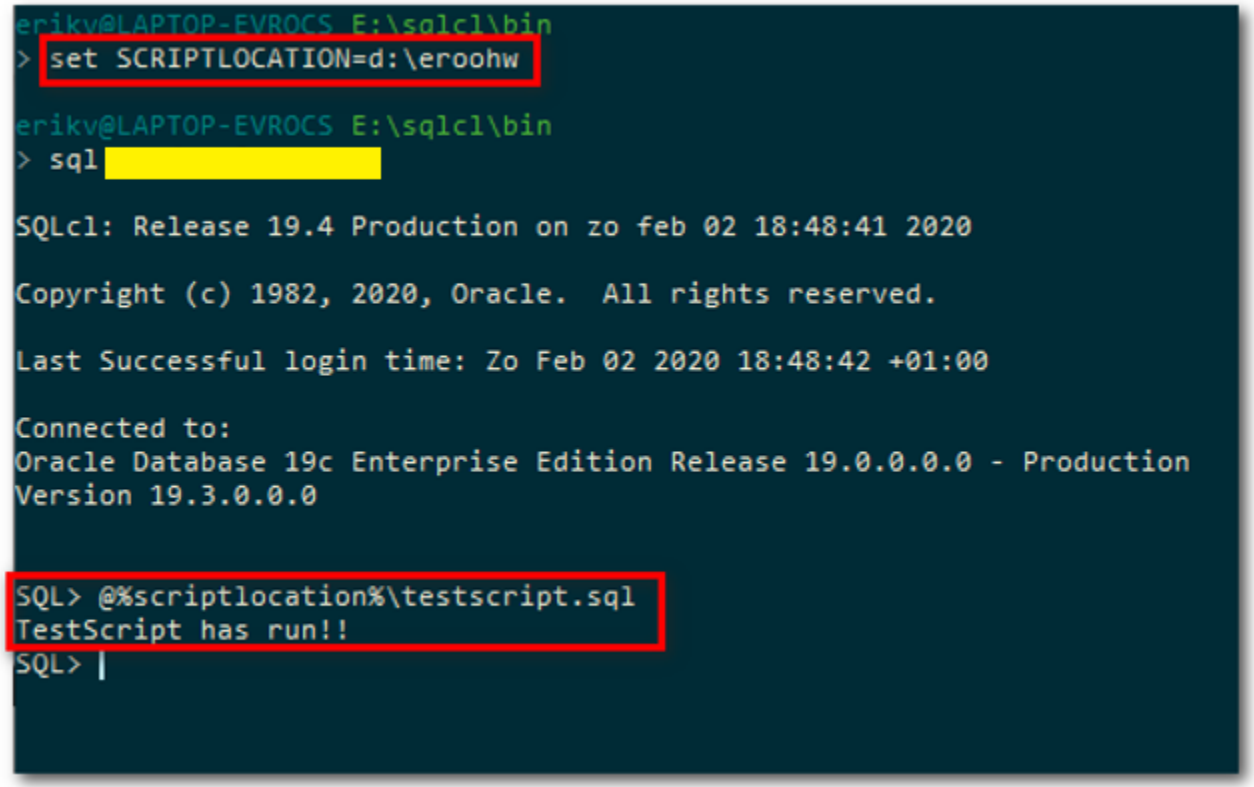
sc.bat

```
@echo off
set sqlpath=x:\login\cl;%sqlpath%
E:\SqlCL\bin\sql.exe %*
```

TIP – environment variables

When running a script using an environment variable, the variable must be declared with a name in uppercase, or SQLcl won't see it





TIP - language

In **SQLPLUS** language of messages is controlled by NLS_LANG

In **SQLcl** the language is determined by OS language

To change this, add to previously mentioned batch file:

```
set JAVA_TOOL_OPTIONS=-Duser.language=en -Duser.region=US
```

Replace ‘en’ and ‘US’ as desired

TIP – classic mode

By default error messages show more than in SqlPlus

If you don't like that:

```
set classic on
```

```
ERO@EVROCS>begin
  2    dbms_output.put_line('x')
  3  end;
  4  /

Error starting at line : 1 in command -
begin
  dbms_output.put_line('x')
end;
Error report -
ORA-06550: line 3, column 1:
PLS-00103: Encountered the symbol "END" when expecting one of the following:

   := . ( % ;
The symbol ";" was substituted for "END" to continue.
06550. 00000 -  "line %s, column %s:\n%s"
*Cause:    Usually a PL/SQL compilation error.
*Action:
ERO@EVROCS>
ERO@EVROCS>set classic on
ERO@EVROCS>
ERO@EVROCS>begin
  2    dbms_output.put_line('x')
  3  end;
  4  /
end;
*
ERROR at line 3:
ORA-06550: line 3, column 1:
PLS-00103: Encountered the symbol "END" when expecting one of the following:

   := . ( % ;
The symbol ";" was substituted for "END" to continue.


ERO@EVROCS>
```

TIP - tnsnames

Multiple DB-Aliases in a TNSNAMES.ORA entry?

```
ORCL , ORCL.WORLD=
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1520))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

Leads to errors on connecting

Solution:

Create environment variable

ORACLE_HOME

Pointing to location of Oracle Client software

```
501677@V-AZ-CENH-05056 C:\Program Files\sqlcl\bin
$ sql [redacted]

SQLcl: Release 19.1 Production on di jun 18 12:35:23 2019

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

  USER          = mvs_own
  URL           = jdbc:oracle:thin:@[redacted]
  Error Message = IO-fout: could not resolve the connect identifier  "[redacted]"
  USER          = [redacted]
  URL           = jdbc:oracle:thin:@[redacted]
  Error Message = IO-fout: could not resolve the connect identifier  "[redacted]"
  USER          = [redacted]
  URL           = jdbc:oracle:thin:@[redacted]:1521/[redacted]
  Error Message = IO-fout: could not resolve the connect identifier  "[redacted]:1521/[redacted]"
Username? (RETRYING) ('[redacted]/*********@[redacted]'?) ^C
501677@V-AZ-CENH-05056 C:\Program Files\sqlcl\bin
$ set ORACLE_HOME=C:\OracleInstantClient

501677@V-AZ-CENH-05056 C:\Program Files\sqlcl\bin
$ sql [redacted]

SQLcl: Release 19.1 Production on di jun 18 12:35:58 2019

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Last Successful login time: Di Jun 18 2019 12:36:02 +02:00

Connected to:
Oracle Database 18c EE Extreme Perf Release 18.0.0.0.0 - Production
Version 18.5.0.0.0


SQL>
```

“Stupid questions do exist.
But it takes a lot more time and energy to correct a stupid mistake than it takes to answer a stupid question, so please ask your stupid questions.”

a wise teacher who taught me more than just physics

Thanks !!