It’s Our Anniversary

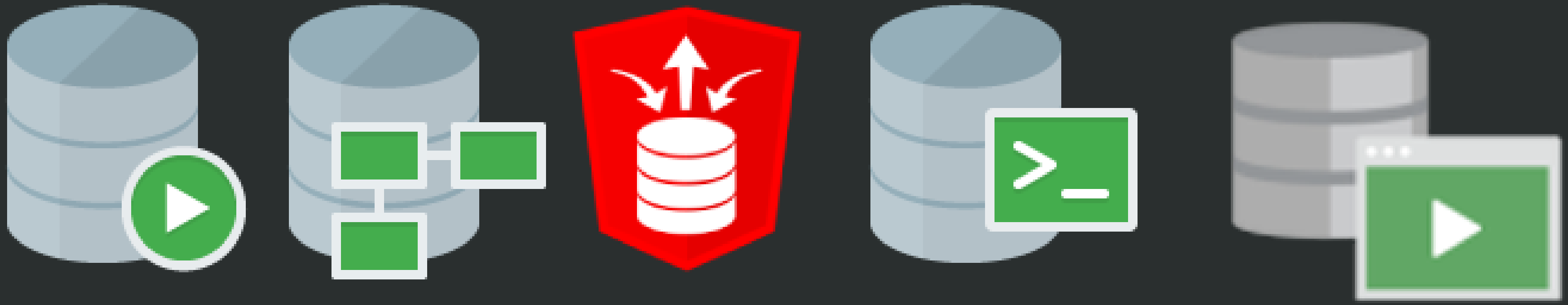
Oracle SQLcl

Proudly serving Oracle database developers since 2016

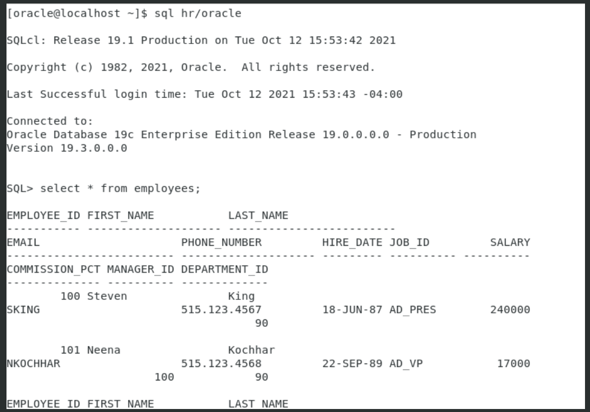
years

Modern Tooling Empowers Users, Increases Productivity

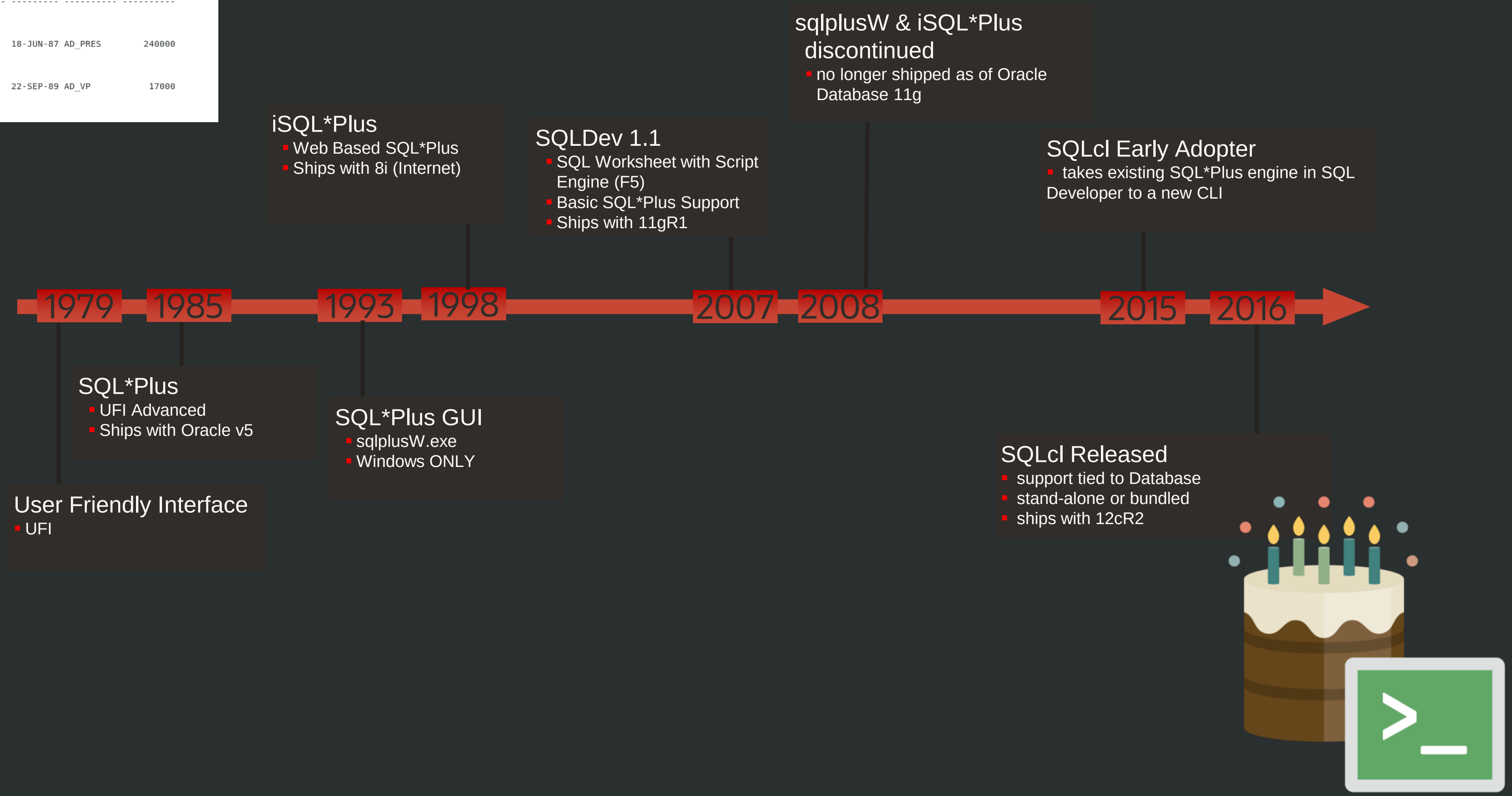
Use the right tool for the right job



Desktop REST APIs **CLI** WEB



```
[oracle@localhost ~]$ sql hr/oracle

SQLcl: Release 19.1 Production on Tue Oct 12 15:53:42 2021

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Last Successful login time: Tue Oct 12 2021 15:53:43 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0


SQL> select * from employees;

EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- -------------------------
EMAIL                     PHONE_NUMBER         HIRE_DATE JOB_ID         SALARY
------------------------- -------------------- --------- ---------- ----------
COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
-------------- ---------- -------------
        100 Steven               King
SKING                     515.123.4567         18-JUN-87 AD_PRES        240000
                                     90

        101 Neena                Kochhar
NKOCHHAR                  515.123.4568         22-SEP-89 AD_VP           17000
                      100            90

EMPLOYEE_ID FIRST_NAME           LAST_NAME
```



1979

User Friendly Interface
- UFI

1985

SQL*Plus
- UFI Advanced
- Ships with Oracle v5

1993

SQL*Plus GUI
- sqlplusW.exe
- Windows ONLY

1998

iSQL*Plus
- Web Based SQL*Plus
- Ships with 8i (Internet)

2007

SQLDev 1.1
- SQL Worksheet with Script Engine (F5)
- Basic SQL*Plus Support
- Ships with 11gR1

2008

sqlplusW & iSQL*Plus discontinued
- no longer shipped as of Oracle Database 11g

2015

SQLcl Early Adopter
- takes existing SQL*Plus engine in SQL Developer to a new CLI

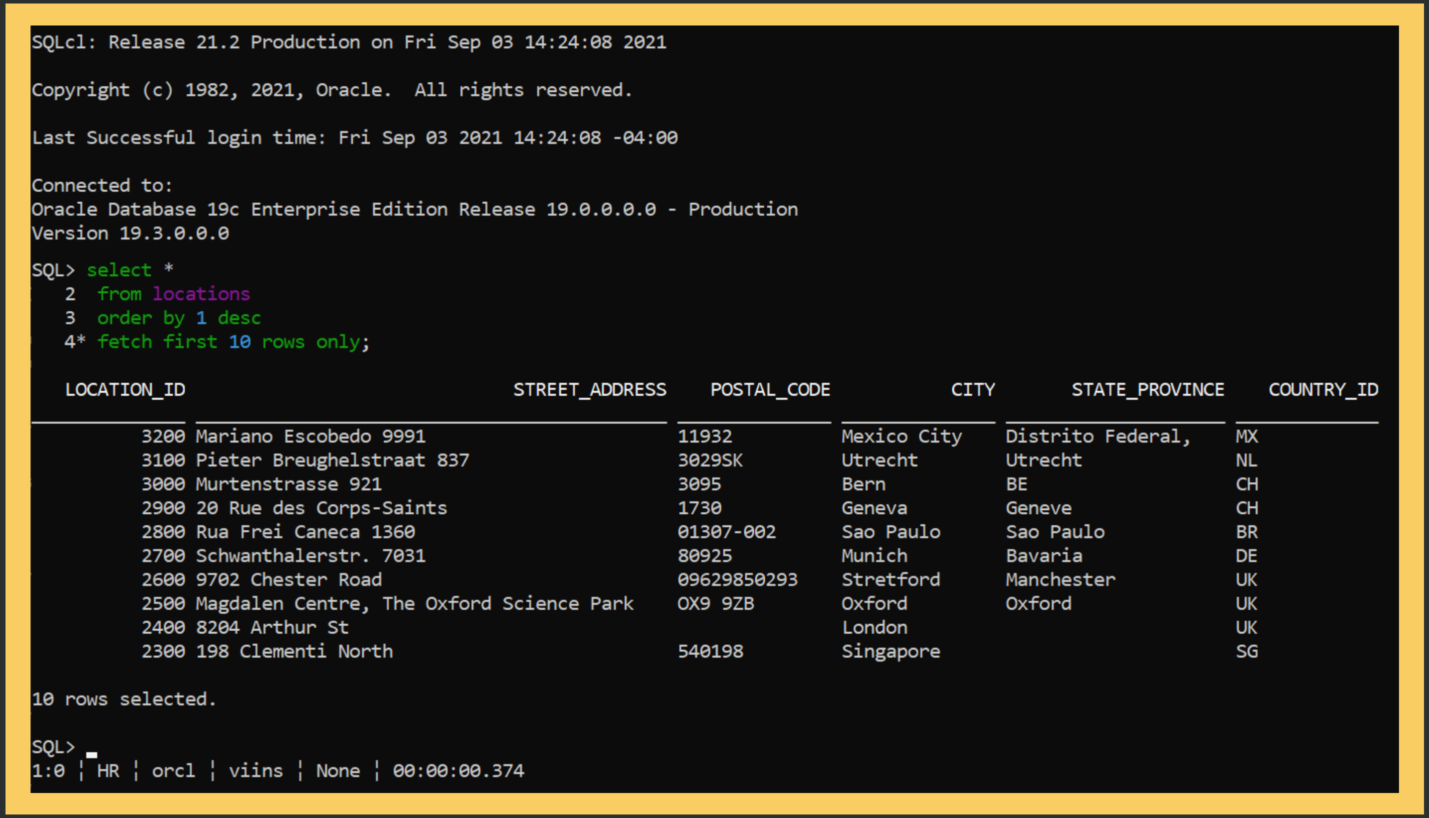
2016

SQLcl Released
- support tied to Database
- stand-alone or bundled
- ships with 12cR2

SQLcl

FREE

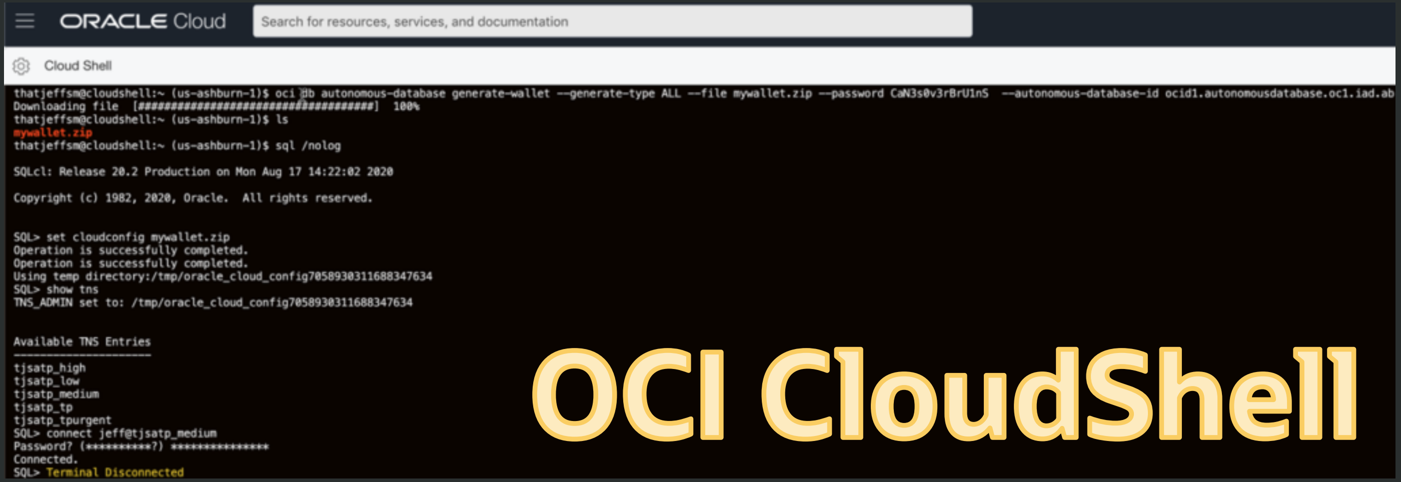
- Best of SQLDev & SQL*Plus
- Auto-Formatting
- Persistent Query History
- Power of JavaScript
- Custom Commands
- vi & Emacs
- Liquibase
- OCI Integration
- *Data Pump & Data Guard*



```
SQLcl: Release 21.2 Production on Fri Sep 03 14:24:08 2021

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Last Successful login time: Fri Sep 03 2021 14:24:08 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select *
  2  from locations
  3  order by 1 desc
  4* fetch first 10 rows only;

   LOCATION_ID                              STREET_ADDRESS    POSTAL_CODE           CITY       STATE_PROVINCE    COUNTRY_ID

______________ ____________________________________________ ______________ ______________ ____________________ _____________
          3200 Mariano Escobedo 9991                        11932          Mexico City    Distrito Federal,    MX
          3100 Pieter Breughelstraat 837                    3029SK         Utrecht        Utrecht              NL
          3000 Murtenstrasse 921                            3095           Bern           BE                   CH
          2900 20 Rue des Corps-Saints                      1730           Geneva         Geneve               CH
          2800 Rua Frei Caneca 1360                         01307-002      Sao Paulo      Sao Paulo            BR
          2700 Schwanthalerstr. 7031                        80925          Munich         Bavaria              DE
          2600 9702 Chester Road                            09629850293    Stretford      Manchester           UK
          2500 Magdalen Centre, The Oxford Science Park     OX9 9ZB        Oxford         Oxford               UK
          2400 8204 Arthur St                                              London                              UK
          2300 198 Clementi North                           540198         Singapore                           SG

10 rows selected.

SQL>
1:0 ¦ HR ¦ orcl ¦ viins ¦ None ¦ 00:00:00.374
```

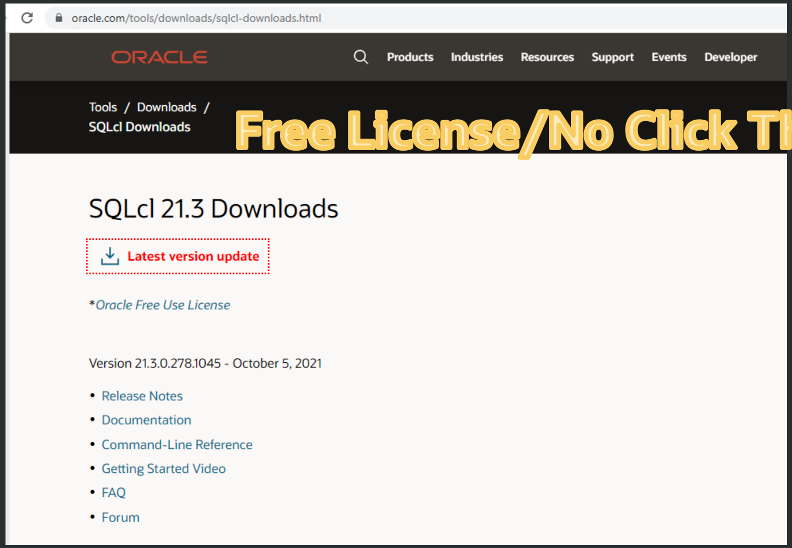
https://www.oracle.com/sqlcl

Accessible from anywhere, easily



```
thatjeffsm@cloudshell:~ (us-ashburn-1)$ oci db autonomous-database generate-wallet --generate-type ALL --file mywallet.zip --password CaN3s0v3rBrU1nS  --autonomous-database-id ocid1.autonomousdatabase.oc1.iad.ab
Downloading file  [####################################]  100%
thatjeffsm@cloudshell:~ (us-ashburn-1)$ ls
mywallet.zip
thatjeffsm@cloudshell:~ (us-ashburn-1)$ sql /nolog

SQLcl: Release 20.2 Production on Mon Aug 17 14:22:02 2020

Copyright (c) 1982, 2020, Oracle.  All rights reserved.


SQL> set cloudconfig mywallet.zip
Operation is successfully completed.
Operation is successfully completed.
Using temp directory:/tmp/oracle_cloud_config7058930311688347634
SQL> show tns
TNS_ADMIN set to: /tmp/oracle_cloud_config7058930311688347634


Available TNS Entries
---------------------
tjsatp_high
tjsatp_low
tjsatp_medium
tjsatp_tp
tjsatp_tpurgent
SQL> connect jeff@tjsatp_medium
Password? (**********?) ***************
Connected.
SQL> Terminal Disconnected
```

OCI CloudShell



Tools / Downloads /
SQLcl Downloads

Free License/No Click Thru or SSO

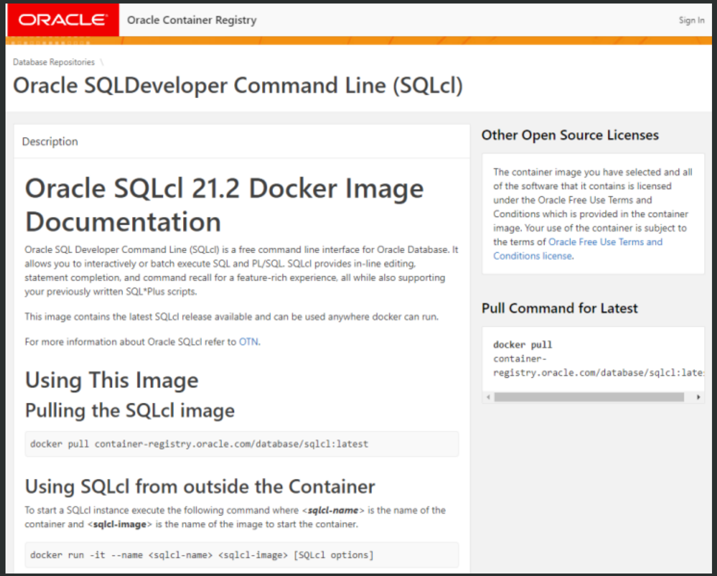
SQLcl 21.3 Downloads

Latest version update

*Oracle Free Use License

Version 21.3.0.278.1045 - October 5, 2021

- Release Notes
- Documentation
- Command-Line Reference
- Getting Started Video
- FAQ
- Forum



Database Repositories \

Oracle SQLDeveloper Command Line (SQLcl)

Description

Oracle SQLcl 21.2 Docker Image Documentation

Oracle SQL Developer Command Line (SQLcl) is a free command line interface for Oracle Database. It allows you to interactively or batch execute SQL and PL/SQL. SQLcl provides in-line editing, statement completion, and command recall for a feature-rich experience, all while also supporting your previously written SQL*Plus scripts.

This image contains the latest SQLcl release available and can be used anywhere docker can run.

For more information about Oracle SQLcl refer to OTN.

Using This Image

Pulling the SQLcl image

```
docker pull container-registry.oracle.com/database/sqlcl:latest
```

Using SQLcl from outside the Container

To start a SQLcl instance execute the following command where <***sqlcl-name***> is the name of the container and <**sqlcl-image**> is the name of the image to start the container.

```
docker run -it --name <sqlcl-name> <sqlcl-image> [SQLcl options]
```

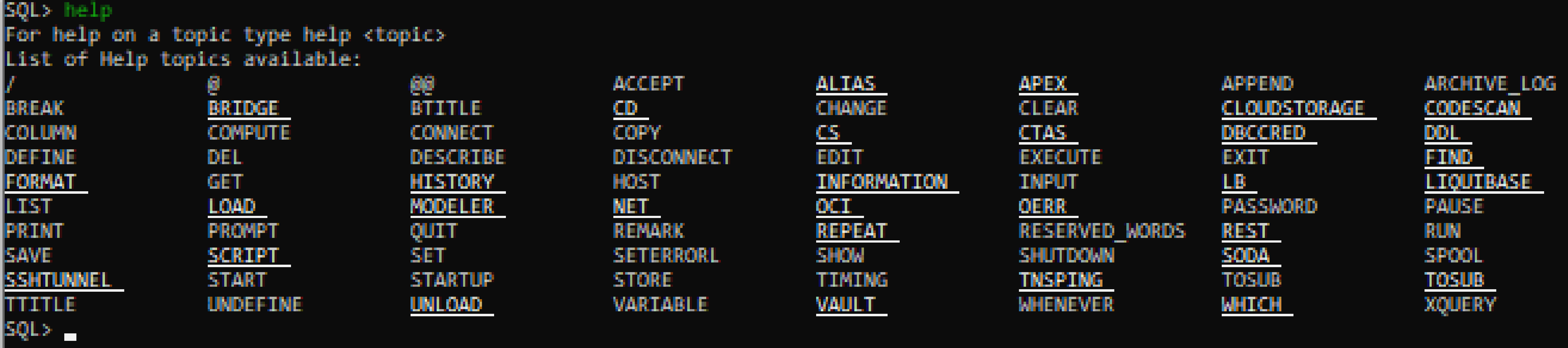
Other Open Source Licenses

The container image you have selected and all of the software that it contains is licensed under the Oracle Free Use Terms and Conditions which is provided in the container image. Your use of the container is subject to the terms of Oracle Free Use Terms and Conditions license.

Pull Command for Latest

```
docker pull
container-
registry.oracle.com/database/sqlcl:late
```

Oracle Container Registry (Docker)



```
SQL> help
For help on a topic type help <topic>
List of Help topics available:
/                @                @@               ACCEPT           ALIAS            APEX             APPEND           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>
```

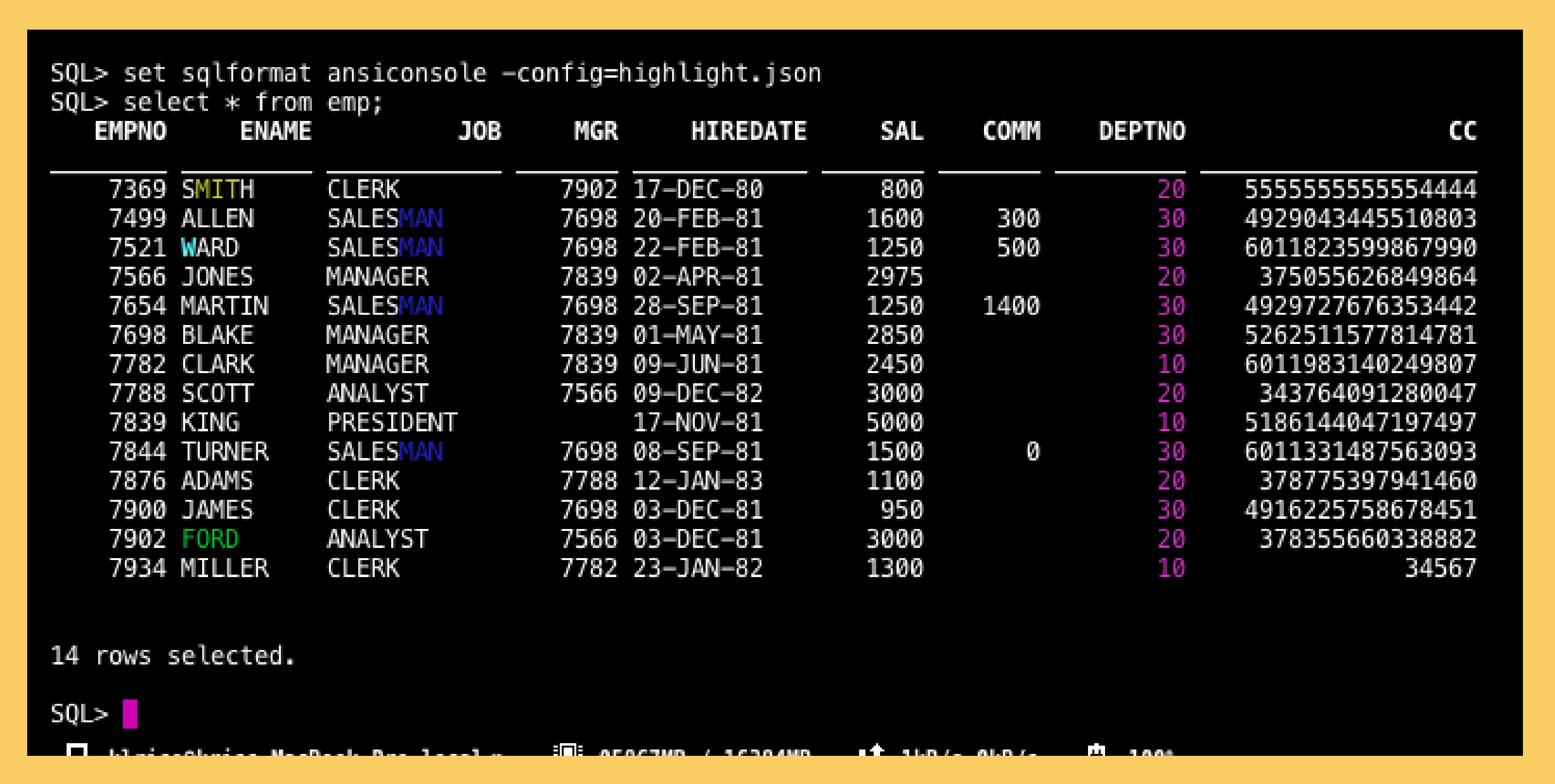
Commands, new from SQL*Plus are Underlined

```
1:0 ¦ HR ¦ orcl ¦ viins ¦ None ¦ 00:00:00.001
```

Query Output

It’s Automatic!

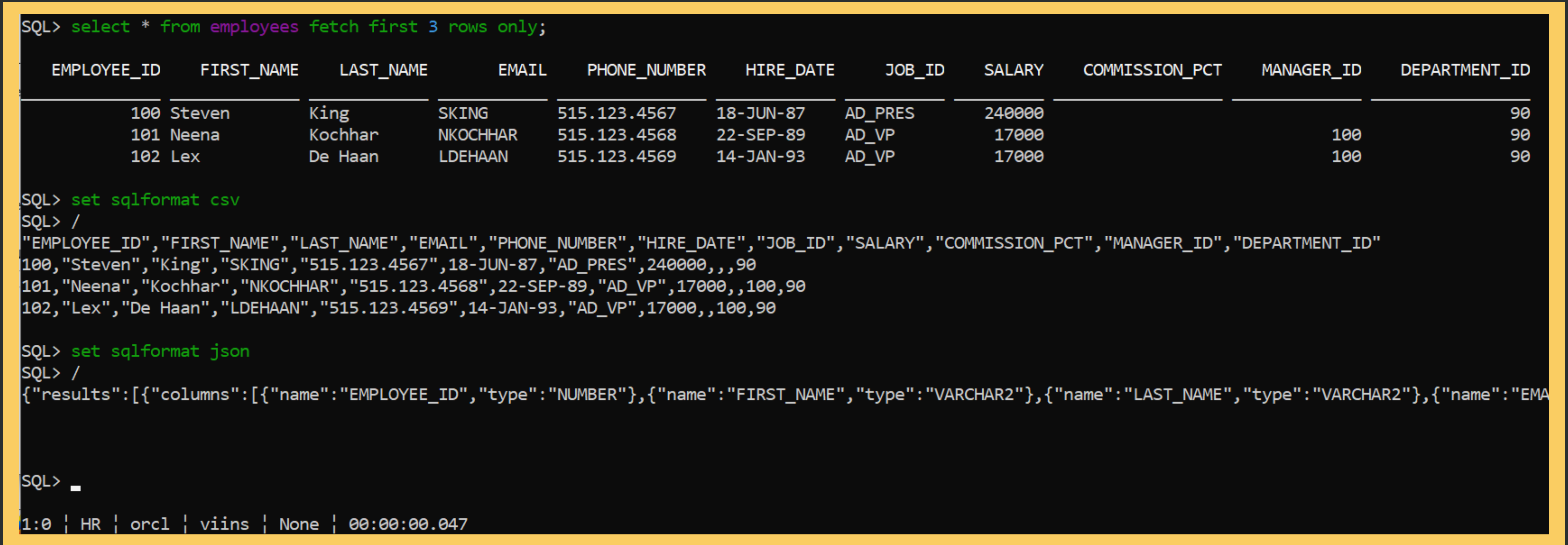
- Fit to screen, default (ANSICONSOLE)
  - Color Coded
- CSV
- DELIMITED
- HTML
  INSERT
- JSON
- JSON-FORMATTED
- XML



```
SQL> set sqlformat ansiconsole -config=highlight.json
SQL> select * from emp;
   EMPNO     ENAME          JOB     MGR     HIREDATE     SAL    COMM    DEPTNO                  CC
-------- --------- ------------ ------- ------------ ------- ------- --------- -------------------
    7369 SMITH     CLERK           7902 17-DEC-80        800                20    5555555555554444
    7499 ALLEN     SALESMAN        7698 20-FEB-81       1600     300        30    4929043445510803
    7521 WARD      SALESMAN        7698 22-FEB-81       1250     500        30    6011823599867990
    7566 JONES     MANAGER         7839 02-APR-81       2975                20     375055626849864
    7654 MARTIN    SALESMAN        7698 28-SEP-81       1250    1400        30    4929727676353442
    7698 BLAKE     MANAGER         7839 01-MAY-81       2850                30    5262511577814781
    7782 CLARK     MANAGER         7839 09-JUN-81       2450                10    6011983140249807
    7788 SCOTT     ANALYST         7566 09-DEC-82       3000                20     343764091280047
    7839 KING      PRESIDENT            17-NOV-81       5000                10    5186144047197497
    7844 TURNER    SALESMAN        7698 08-SEP-81       1500       0        30    6011331487563093
    7876 ADAMS     CLERK           7788 12-JAN-83       1100                20     378775397941460
    7900 JAMES     CLERK           7698 03-DEC-81        950                30    4916225758678451
    7902 FORD      ANALYST         7566 03-DEC-81       3000                20     378355660338882
    7934 MILLER    CLERK           7782 23-JAN-82       1300                10               34567


14 rows selected.

SQL>
```

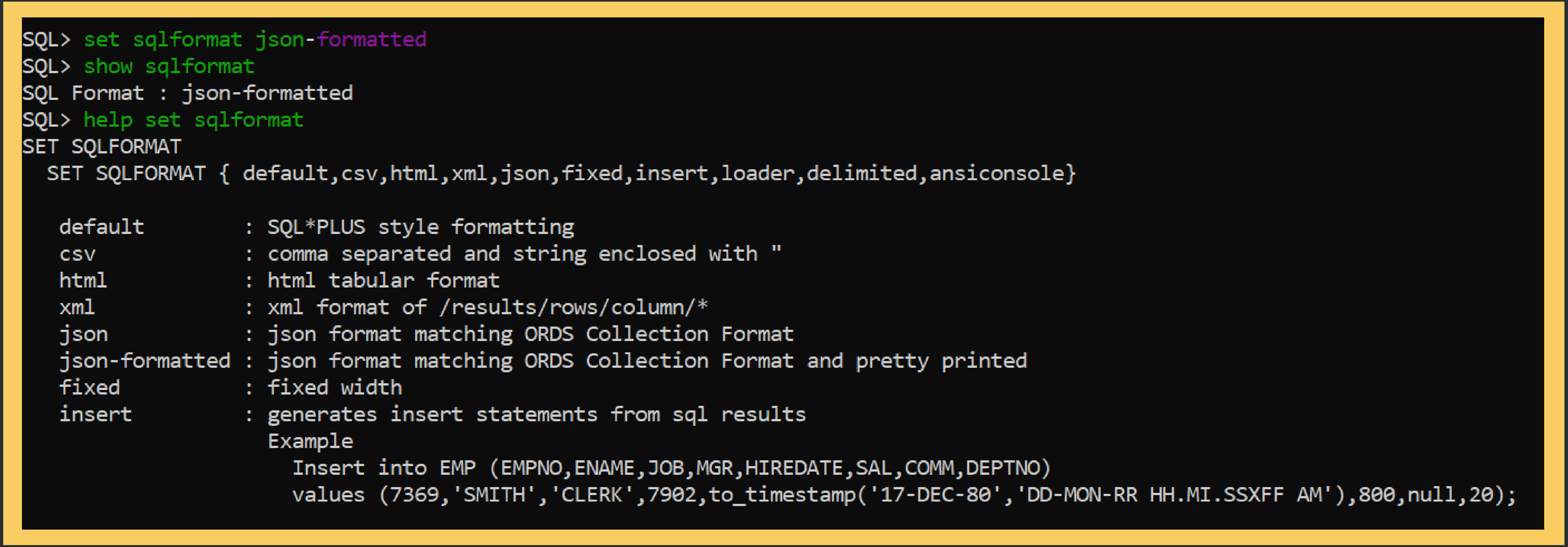
Formatting

set sqlformat or */*format*/*



```
SQL> select * from employees fetch first 3 rows only;

   EMPLOYEE_ID    FIRST_NAME    LAST_NAME        EMAIL    PHONE_NUMBER    HIRE_DATE       JOB_ID    SALARY    COMMISSION_PCT    MANAGER_ID    DEPARTMENT_ID
______________ _____________ ____________ ____________ _______________ ____________ ____________ _________ _________________ _____________ ________________
           100 Steven        King         SKING        515.123.4567    18-JUN-87    AD_PRES         24000                                                 90
           101 Neena         Kochhar      NKOCHHAR     515.123.4568    22-SEP-89    AD_VP           17000                                  100            90
           102 Lex           De Haan      LDEHAAN      515.123.4569    14-JAN-93    AD_VP           17000                                  100            90

SQL> set sqlformat csv
SQL> /
"EMPLOYEE_ID","FIRST_NAME","LAST_NAME","EMAIL","PHONE_NUMBER","HIRE_DATE","JOB_ID","SALARY","COMMISSION_PCT","MANAGER_ID","DEPARTMENT_ID"
100,"Steven","King","SKING","515.123.4567",18-JUN-87,"AD_PRES",240000,,,90
101,"Neena","Kochhar","NKOCHHAR","515.123.4568",22-SEP-89,"AD_VP",17000,,100,90
102,"Lex","De Haan","LDEHAAN","515.123.4569",14-JAN-93,"AD_VP",17000,,100,90

SQL> set sqlformat json
SQL> /
{"results":[{"columns":[{"name":"EMPLOYEE_ID","type":"NUMBER"},{"name":"FIRST_NAME","type":"VARCHAR2"},{"name":"LAST_NAME","type":"VARCHAR2"},{"name":"EMA



SQL> _

1:0 ¦ HR ¦ orcl ¦ viins ¦ None ¦ 00:00:00.047
```

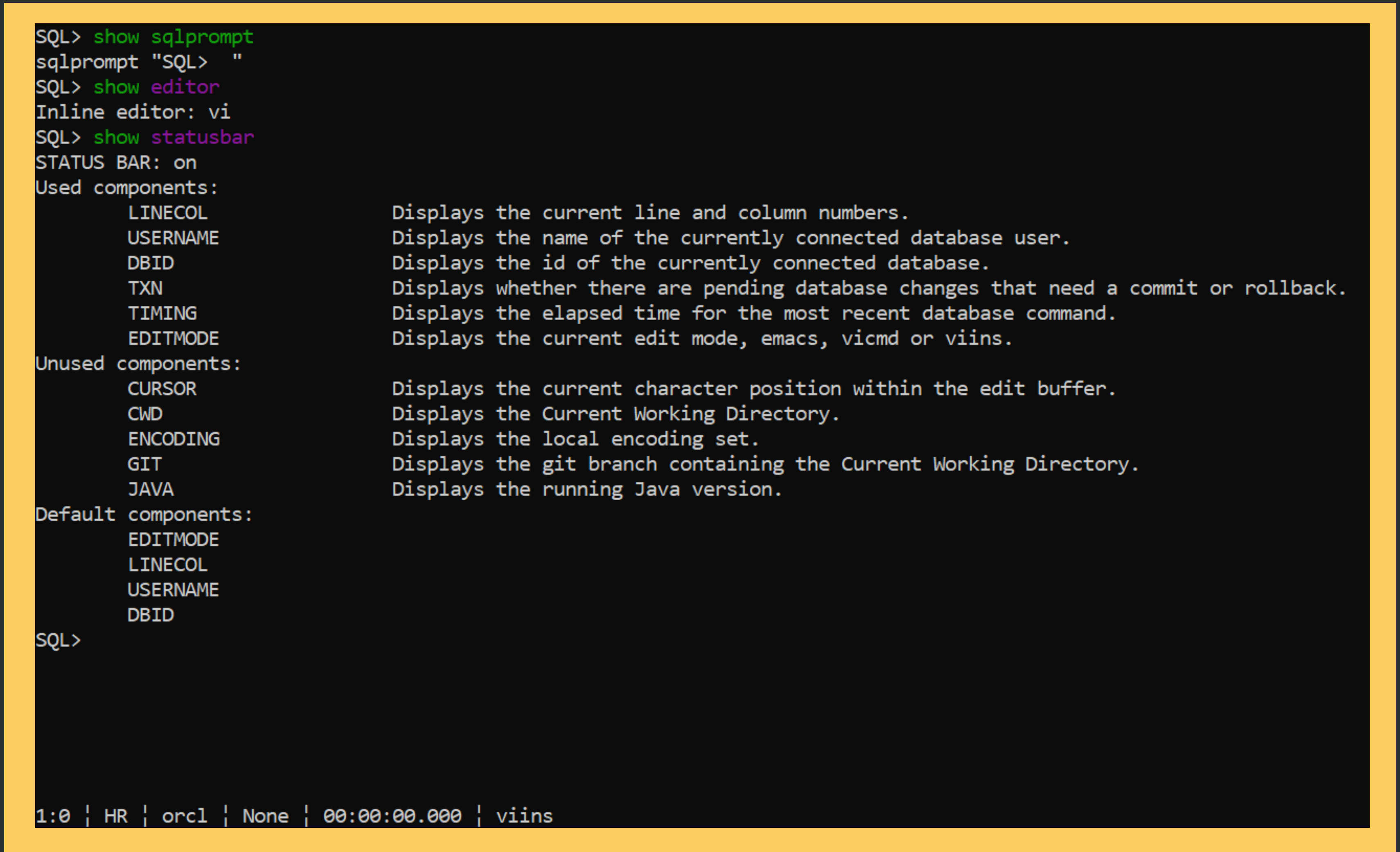
HELP, SET, & SHOW are your friends



```
SQL> set sqlformat json-formatted
SQL> show sqlformat
SQL Format : json-formatted
SQL> help set sqlformat
SET SQLFORMAT
  SET SQLFORMAT { default,csv,html,xml,json,fixed,insert,loader,delimited,ansiconsole}

   default        : SQL*PLUS style formatting
   csv            : comma separated and string enclosed with "
   html           : html tabular format
   xml            : xml format of /results/rows/column/*
   json           : json format matching ORDS Collection Format
   json-formatted : json format matching ORDS Collection Format and pretty printed
   fixed          : fixed width
   insert         : generates insert statements from sql results
                    Example
                      Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO)
                      values (7369,'SMITH','CLERK',7902,to_timestamp('17-DEC-80','DD-MON-RR HH.MI.SSXFF AM'),800,null,20);
```

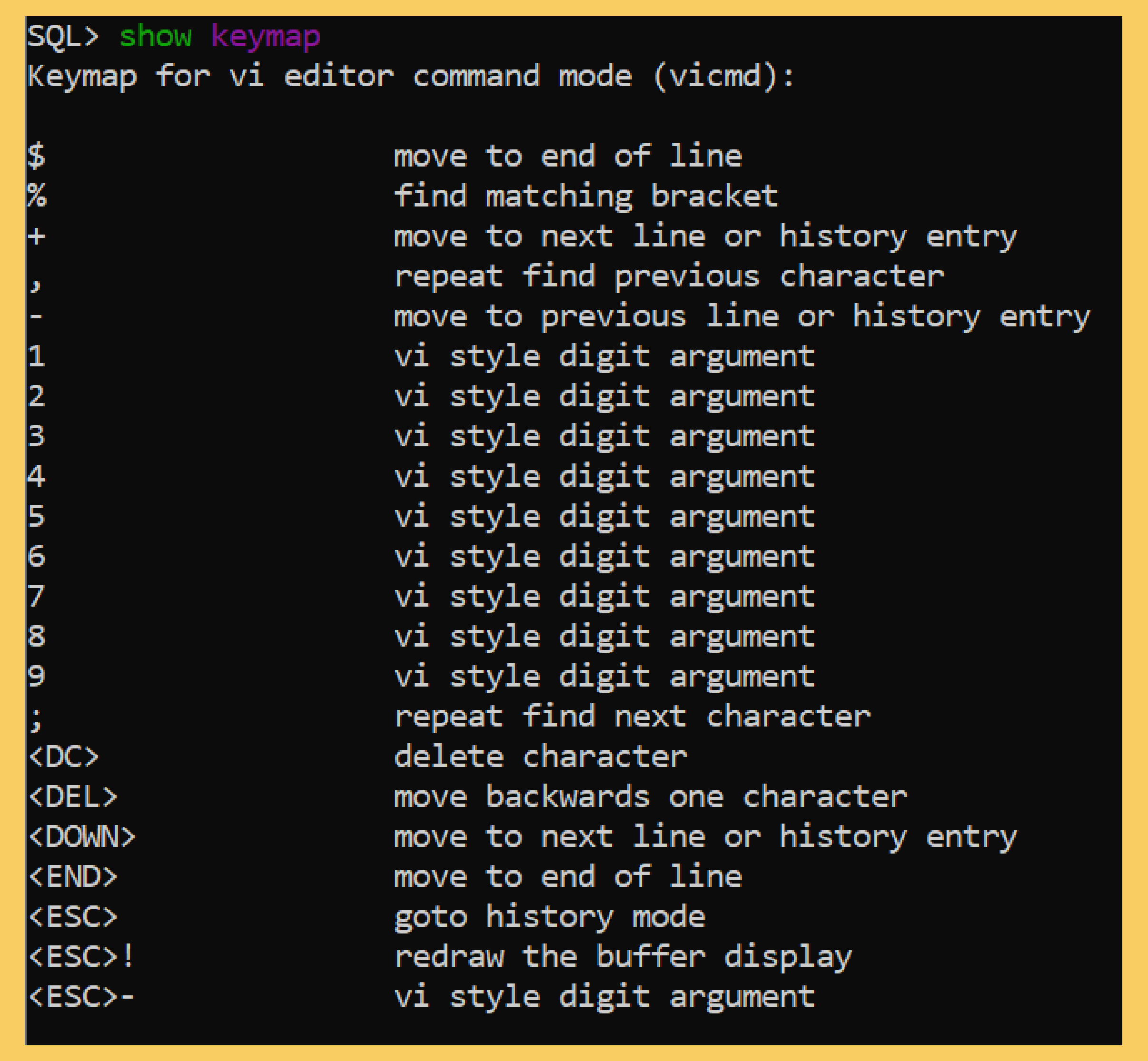
Editing

Edit Mode, Prompt, Statusbar



```
SQL> show sqlprompt
sqlprompt "SQL>  "
SQL> show editor
Inline editor: vi
SQL> show statusbar
STATUS BAR: on
Used components:
        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.
        TXN                    Displays whether there are pending database changes that need a commit or rollback.
        TIMING                 Displays the elapsed time for the most recent database command.
        EDITMODE               Displays the current edit mode, emacs, vicmd or viins.
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.
Default components:
        EDITMODE
        LINECOL
        USERNAME
        DBID
SQL>







1:0 ¦ HR ¦ orcl ¦ None ¦ 00:00:00.000 ¦ viins
```

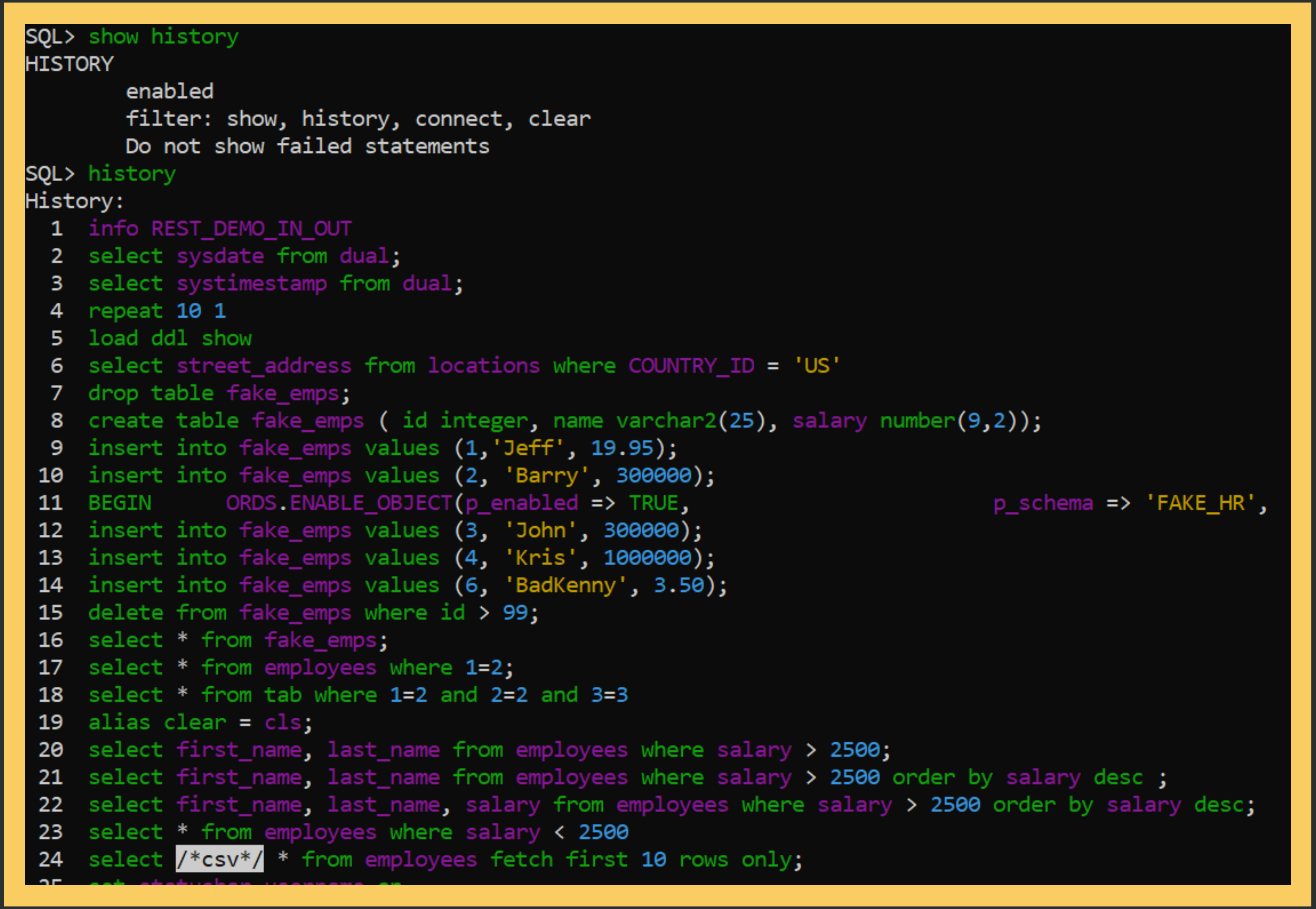
Inline editing with vi command support & the statusbar



```
SQL> show keymap
Keymap for vi editor command mode (vicmd):

$                    move to end of line
%                    find matching bracket
+                    move to next line or history entry
,                    repeat find previous character
-                    move to previous line or history entry
1                    vi style digit argument
2                    vi style digit argument
3                    vi style digit argument
4                    vi style digit argument
5                    vi style digit argument
6                    vi style digit argument
7                    vi style digit argument
8                    vi style digit argument
9                    vi style digit argument
;                    repeat find next character
<DC>                 delete character
<DEL>                move backwards one character
<DOWN>               move to next line or history entry
<END>                move to end of line
<ESC>                goto history mode
<ESC>!               redraw the buffer display
<ESC>-               vi style digit argument
```

History

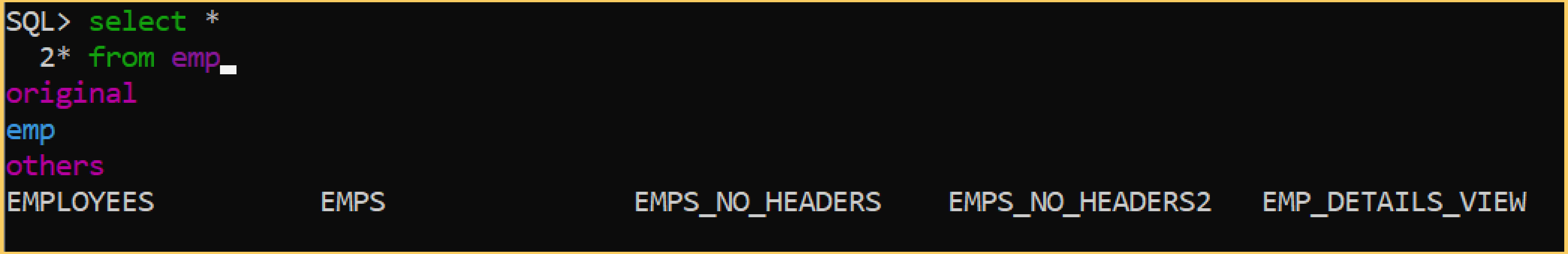
History persists between instances, except for FAILED statements/scripts



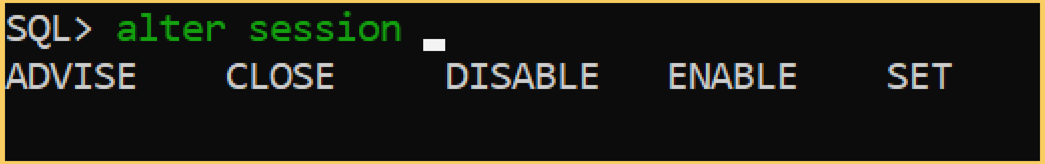
```
SQL> show history
HISTORY
        enabled
        filter: show, history, connect, clear
        Do not show failed statements
SQL> history
History:
  1  info REST_DEMO_IN_OUT
  2  select sysdate from dual;
  3  select systimestamp from dual;
  4  repeat 10 1
  5  load ddl show
  6  select street_address from locations where COUNTRY_ID = 'US'
  7  drop table fake_emps;
  8  create table fake_emps ( id integer, name varchar2(25), salary number(9,2));
  9  insert into fake_emps values (1,'Jeff', 19.95);
 10  insert into fake_emps values (2, 'Barry', 300000);
 11  BEGIN      ORDS.ENABLE_OBJECT(p_enabled => TRUE,                        p_schema => 'FAKE_HR',
 12  insert into fake_emps values (3, 'John', 300000);
 13  insert into fake_emps values (4, 'Kris', 1000000);
 14  insert into fake_emps values (6, 'BadKenny', 3.50);
 15  delete from fake_emps where id > 99;
 16  select * from fake_emps;
 17  select * from employees where 1=2;
 18  select * from tab where 1=2 and 2=2 and 3=3
 19  alias clear = cls;
 20  select first_name, last_name from employees where salary > 2500;
 21  select first_name, last_name from employees where salary > 2500 order by salary desc ;
 22  select first_name, last_name, salary from employees where salary > 2500 order by salary desc;
 23  select * from employees where salary < 2500
 24  select /*csv*/ * from employees fetch first 10 rows only;
 25  set statusbar username on
```

Code Completion/Insight

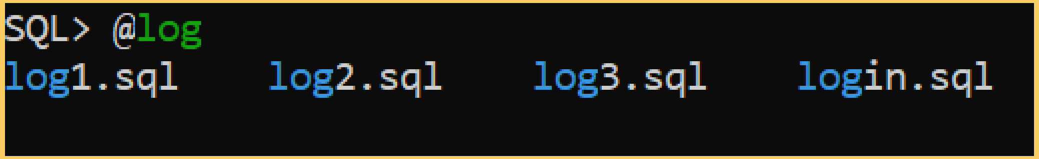
Object names, keywords, commands, and files!



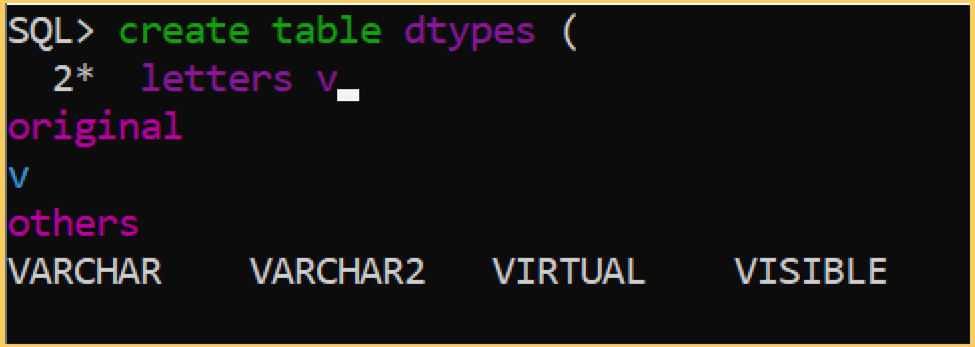
```
SQL> select *
  2* from emp
original
emp
others
EMPLOYEES           EMPS                EMPS_NO_HEADERS     EMPS_NO_HEADERS2    EMP_DETAILS_VIEW
```



```
SQL> alter session
ADVISE     CLOSE      DISABLE    ENABLE     SET
```



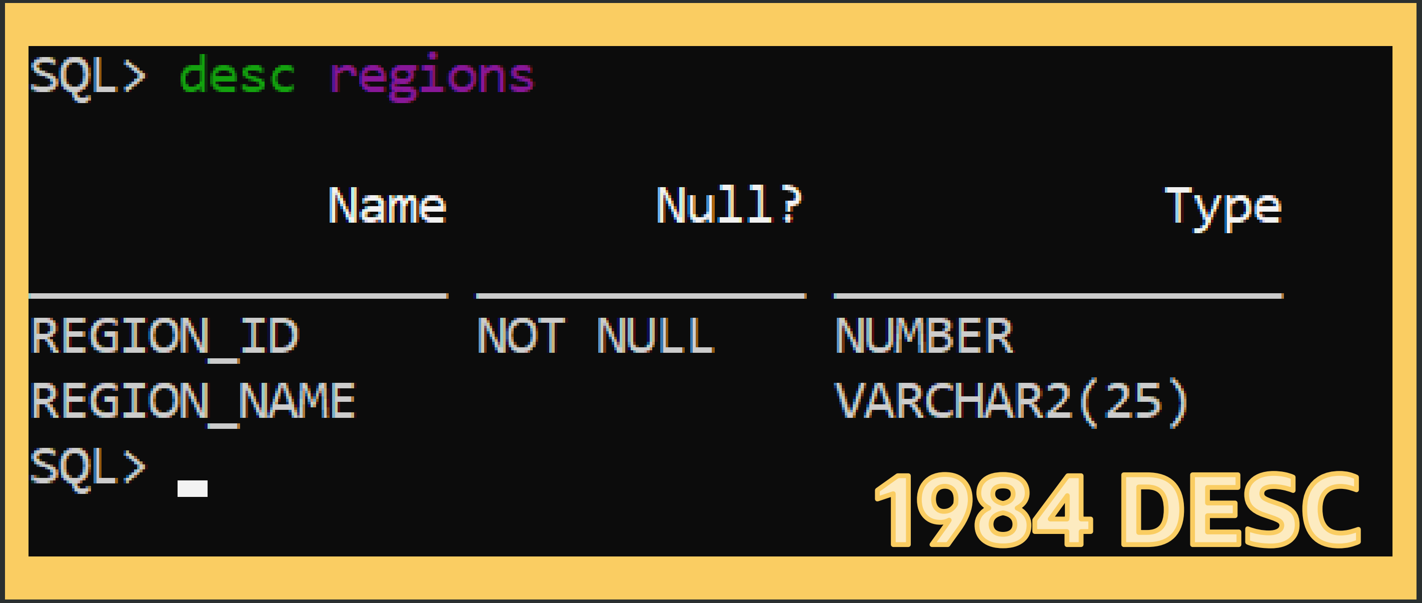
```
SQL> @log
log1.sql     log2.sql     log3.sql     login.sql
```



```
SQL> create table dtypes (
  2*  letters v
original
v
others
VARCHAR     VARCHAR2    VIRTUAL     VISIBLE
```

DESC, INFO, & INFO+

Tell me more about my database objects

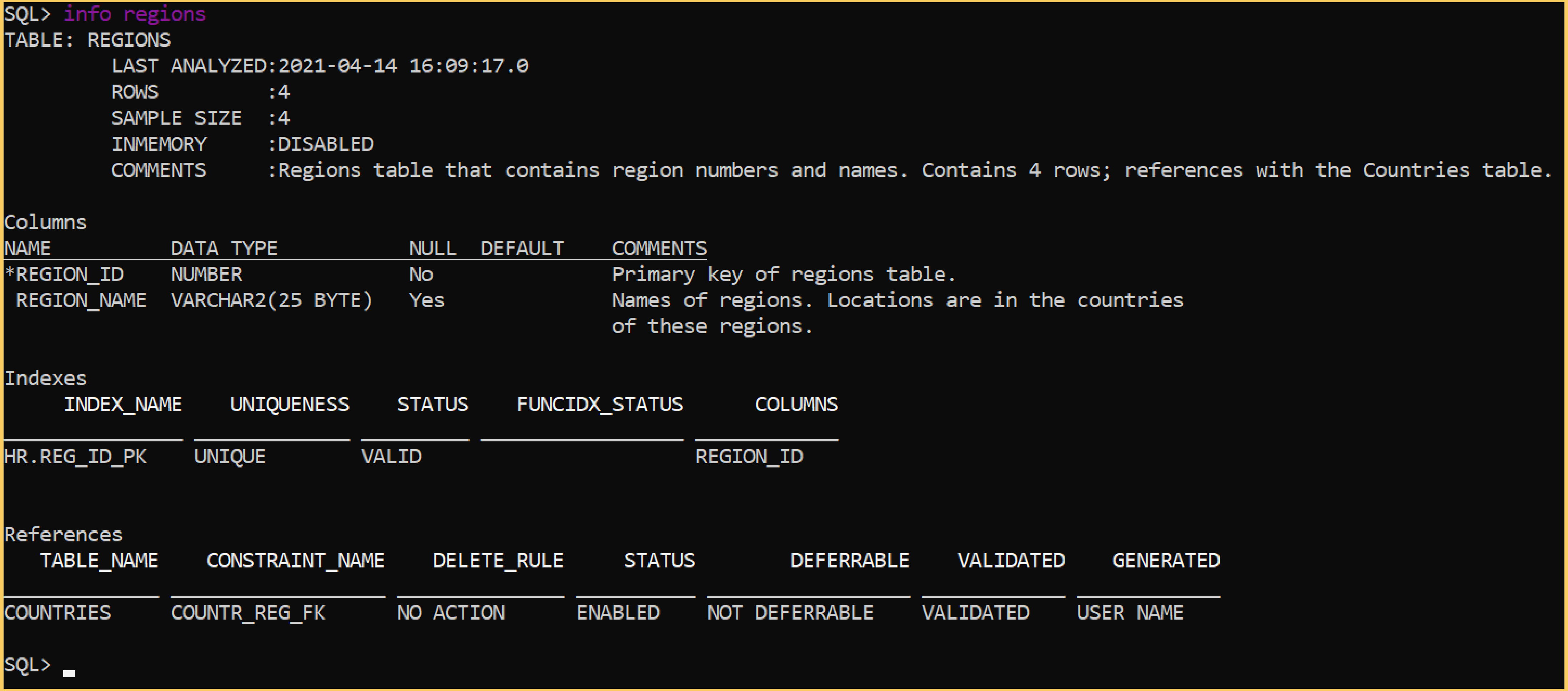


```
SQL> desc regions

           Name          Null?                 Type
______________ ___________ _______________
REGION_ID        NOT NULL     NUMBER
REGION_NAME                   VARCHAR2(25)
SQL> _
```

1984 DESC

INFO(RMATION) – DESC but so much

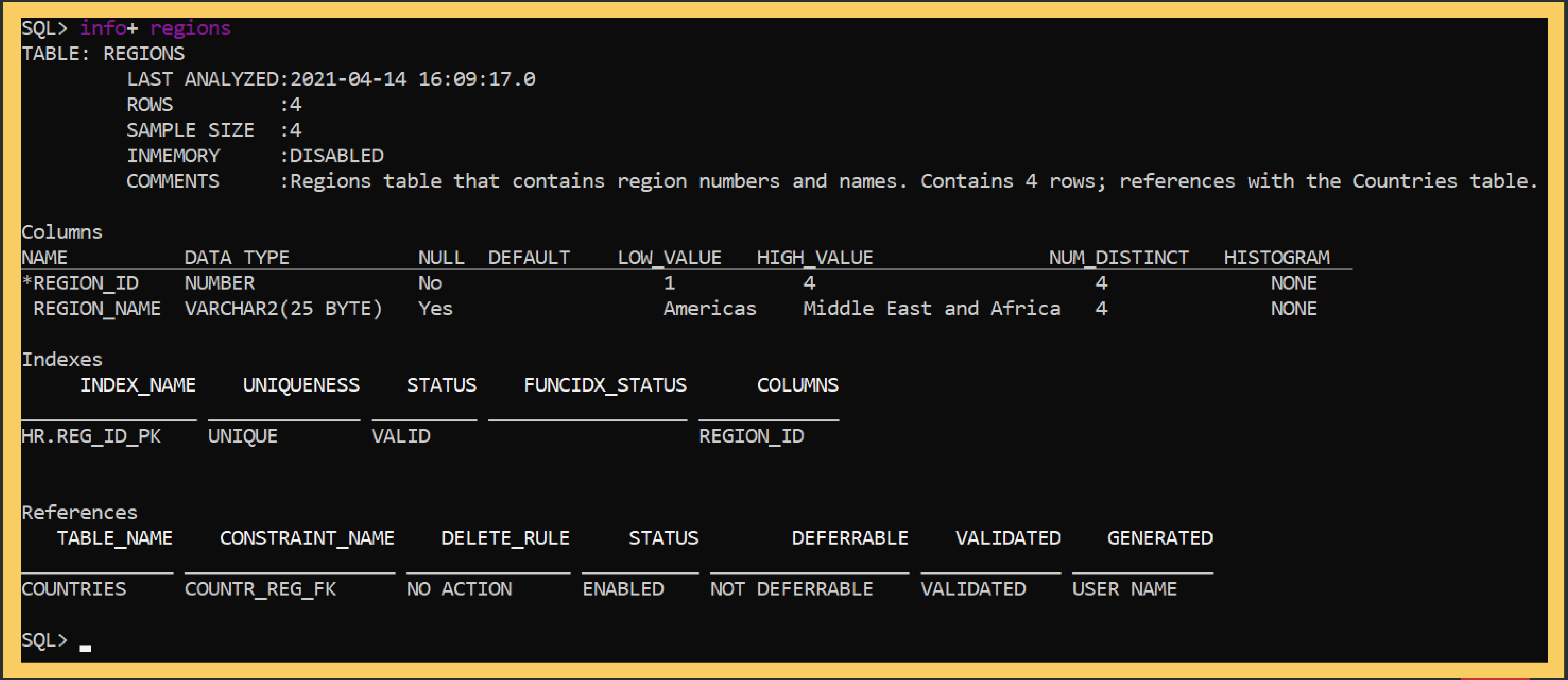
Object Metadata, Comments, Primary Keys, Indexes, and Foreign Keys



```
SQL> info regions
TABLE: REGIONS
         LAST ANALYZED:2021-04-14 16:09:17.0
         ROWS         :4
         SAMPLE SIZE  :4
         INMEMORY     :DISABLED
         COMMENTS     :Regions table that contains region numbers and names. Contains 4 rows; references with the Countries table.

Columns
NAME           DATA TYPE            NULL  DEFAULT    COMMENTS
*REGION_ID     NUMBER               No               Primary key of regions table.
 REGION_NAME   VARCHAR2(25 BYTE)    Yes              Names of regions. Locations are in the countries
                                                     of these regions.

Indexes
     INDEX_NAME     UNIQUENESS     STATUS     FUNCIDX_STATUS      COLUMNS

_______________ _____________ _________ _________________ ____________
HR.REG_ID_PK    UNIQUE        VALID                       REGION_ID


References
   TABLE_NAME     CONSTRAINT_NAME     DELETE_RULE      STATUS        DEFERRABLE     VALIDATED     GENERATED

_____________ _________________ _______________ __________ _________________ ____________ ____________
COUNTRIES     COUNTR_REG_FK     NO ACTION       ENABLED    NOT DEFERRABLE    VALIDATED    USER NAME

SQL> _
```

INFO+

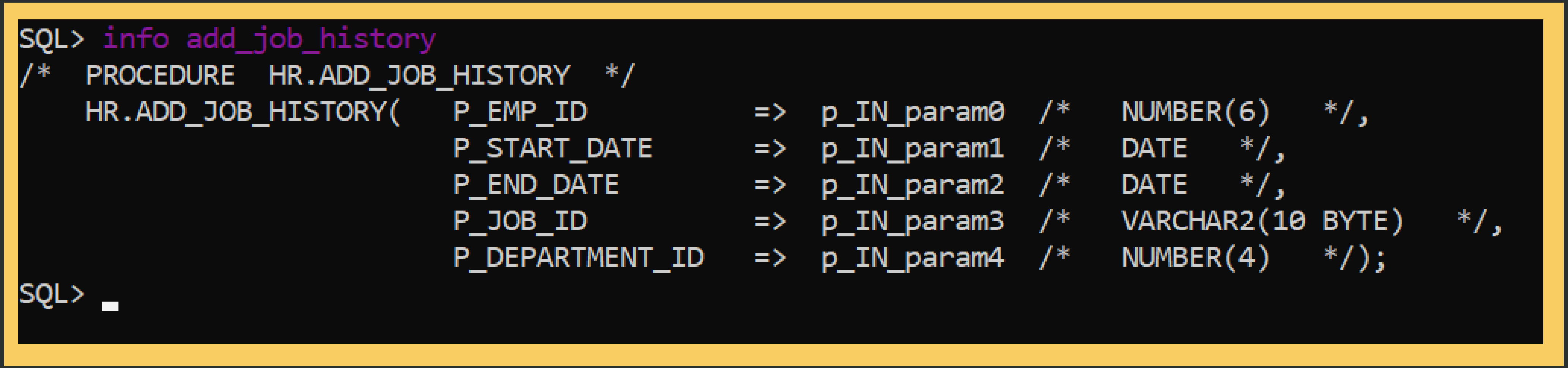
Instead of Column Comments, Show Statistics



```
SQL> info+ regions
TABLE: REGIONS
         LAST ANALYZED:2021-04-14 16:09:17.0
         ROWS         :4
         SAMPLE SIZE  :4
         INMEMORY     :DISABLED
         COMMENTS     :Regions table that contains region numbers and names. Contains 4 rows; references with the Countries table.

Columns
NAME          DATA TYPE           NULL  DEFAULT    LOW_VALUE   HIGH_VALUE               NUM_DISTINCT   HISTOGRAM
----------------------------------------------------------------------------------------------------------------
*REGION_ID    NUMBER              No                   1           4                        4              NONE
 REGION_NAME  VARCHAR2(25 BYTE)   Yes                  Americas    Middle East and Africa   4              NONE

Indexes
     INDEX_NAME    UNIQUENESS    STATUS    FUNCIDX_STATUS      COLUMNS

_______________ _____________ _________ _________________ _____________
HR.REG_ID_PK    UNIQUE        VALID                       REGION_ID


References
   TABLE_NAME    CONSTRAINT_NAME    DELETE_RULE     STATUS        DEFERRABLE    VALIDATED    GENERATED

_____________ __________________ ______________ __________ _________________ ____________ _____________
COUNTRIES     COUNTR_REG_FK      NO ACTION       ENABLED     NOT DEFERRABLE    VALIDATED    USER NAME

SQL> 
```

INFO for PL/SQL

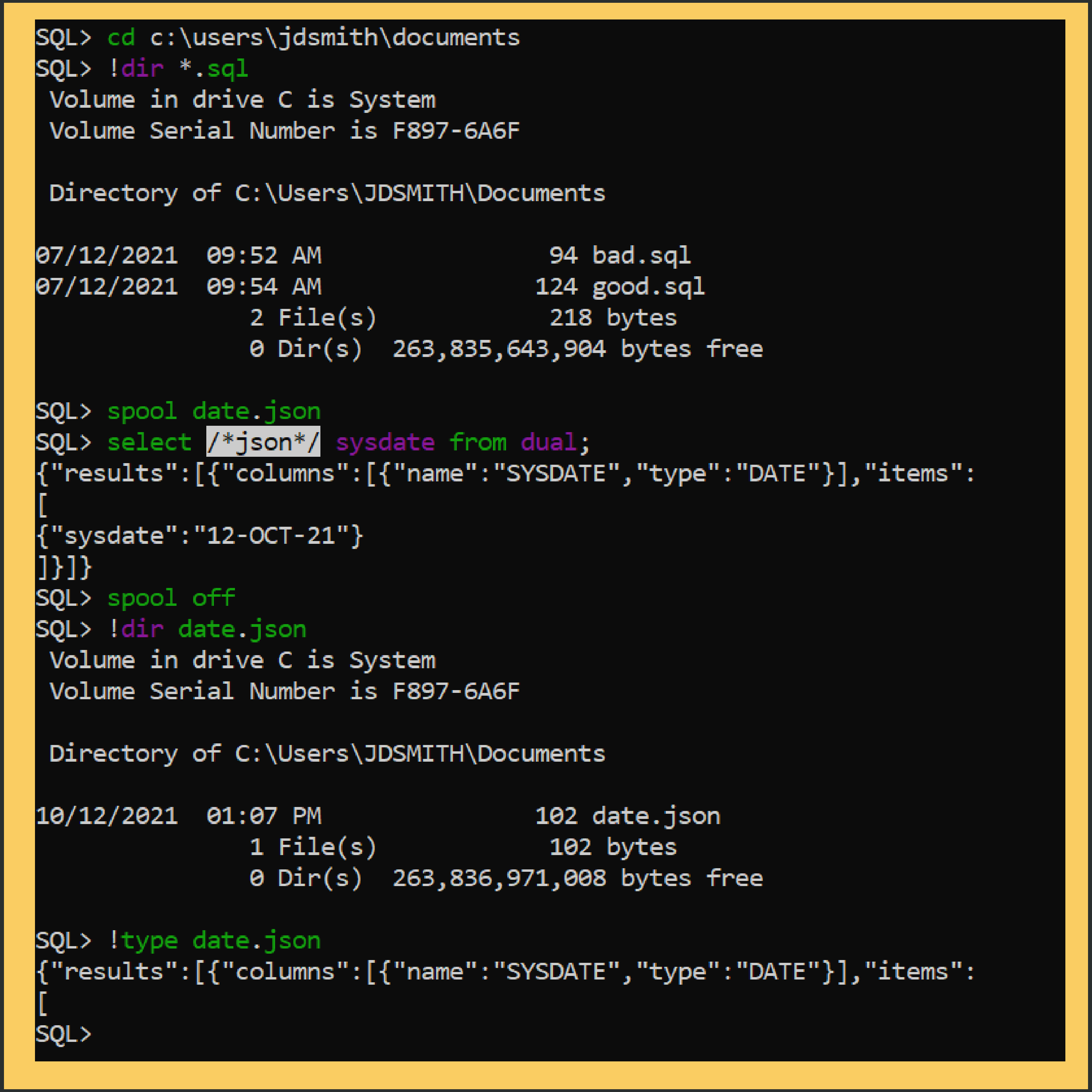
We print the ANON BLOCK required to execute said program



```
SQL> info add_job_history
/*  PROCEDURE  HR.ADD_JOB_HISTORY   */
    HR.ADD_JOB_HISTORY(   P_EMP_ID          =>  p_IN_param0  /*   NUMBER(6)   */,
                          P_START_DATE      =>  p_IN_param1  /*   DATE   */,
                          P_END_DATE        =>  p_IN_param2  /*   DATE   */,
                          P_JOB_ID          =>  p_IN_param3  /*   VARCHAR2(10 BYTE)   */,
                          P_DEPARTMENT_ID   =>  p_IN_param4  /*   NUMBER(4)   */);

SQL>
```

Commons tasks: CD

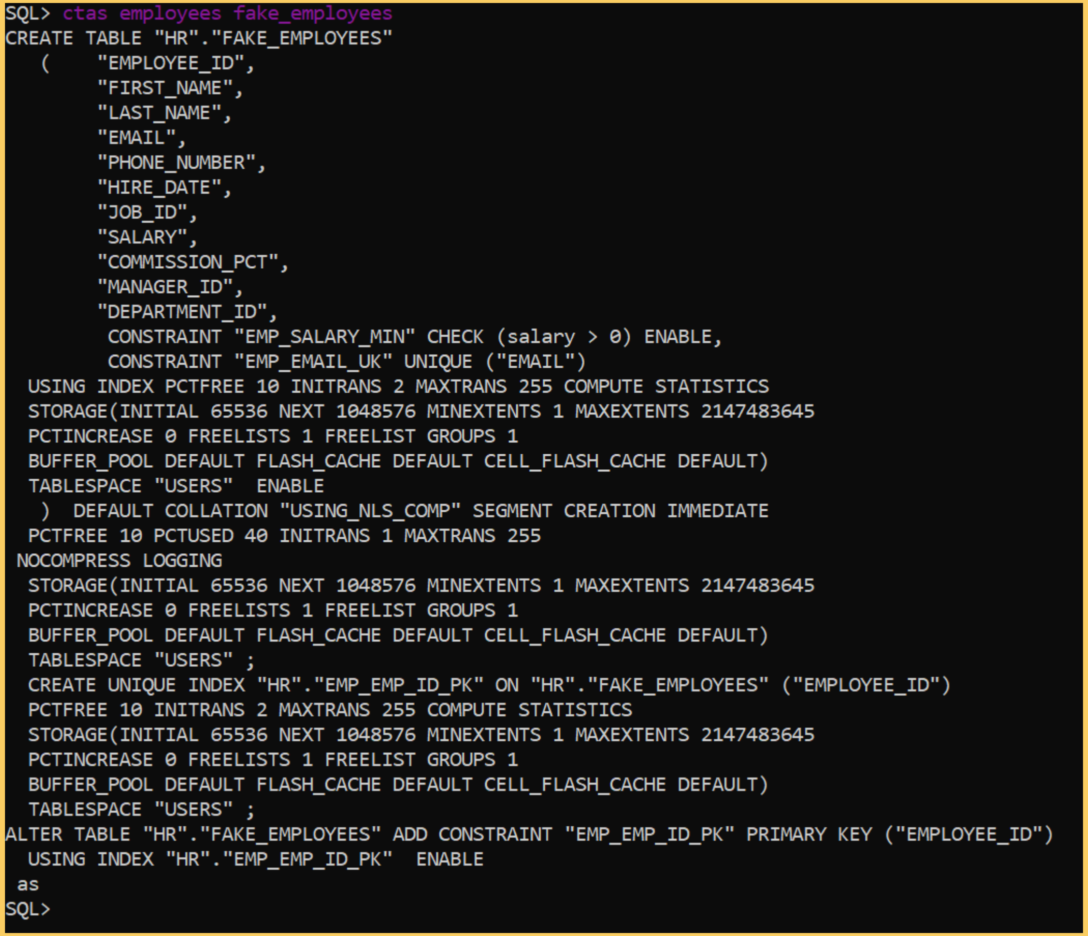
Change directory…



```
SQL> cd c:\users\jdsmith\documents
SQL> !dir *.sql
 Volume in drive C is System
 Volume Serial Number is F897-6A6F

 Directory of C:\Users\JDSMITH\Documents

07/12/2021  09:52 AM                94 bad.sql
07/12/2021  09:54 AM               124 good.sql
               2 File(s)            218 bytes
               0 Dir(s)  263,835,643,904 bytes free

SQL> spool date.json
SQL> select /*json*/ sysdate from dual;
{"results":[{"columns":[{"name":"SYSDATE","type":"DATE"}],"items":
[
{"sysdate":"12-OCT-21"}
]}]}
SQL> spool off
SQL> !dir date.json
 Volume in drive C is System
 Volume Serial Number is F897-6A6F

 Directory of C:\Users\JDSMITH\Documents

10/12/2021  01:07 PM               102 date.json
               1 File(s)            102 bytes
               0 Dir(s)  263,836,971,008 bytes free

SQL> !type date.json
{"results":[{"columns":[{"name":"SYSDATE","type":"DATE"}],"items":
[
SQL>
```

Create TABLE as SELECT

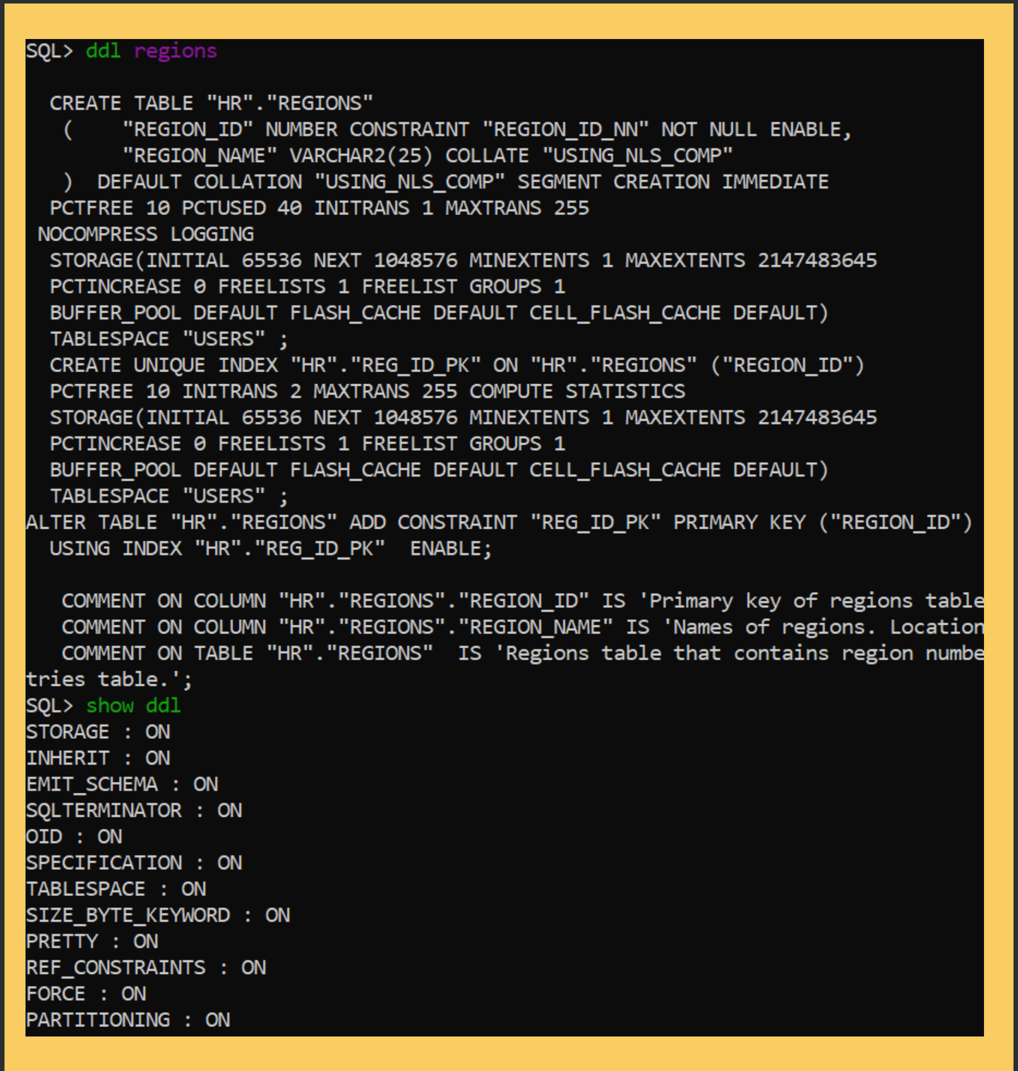
We grab ALL the DDL for the TABLE, just not the COLUMNs



```
SQL> ctas employees fake_employees
CREATE TABLE "HR"."FAKE_EMPLOYEES"
   (    "EMPLOYEE_ID",
        "FIRST_NAME",
        "LAST_NAME",
        "EMAIL",
        "PHONE_NUMBER",
        "HIRE_DATE",
        "JOB_ID",
        "SALARY",
        "COMMISSION_PCT",
        "MANAGER_ID",
        "DEPARTMENT_ID",
         CONSTRAINT "EMP_SALARY_MIN" CHECK (salary > 0) ENABLE,
         CONSTRAINT "EMP_EMAIL_UK" UNIQUE ("EMAIL")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE
   )  DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
  CREATE UNIQUE INDEX "HR"."EMP_EMP_ID_PK" ON "HR"."FAKE_EMPLOYEES" ("EMPLOYEE_ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
ALTER TABLE "HR"."FAKE_EMPLOYEES" ADD CONSTRAINT "EMP_EMP_ID_PK" PRIMARY KEY ("EMPLOYEE_ID")
  USING INDEX "HR"."EMP_EMP_ID_PK"  ENABLE
 as
SQL>
```

DDL – generate DDL for any object

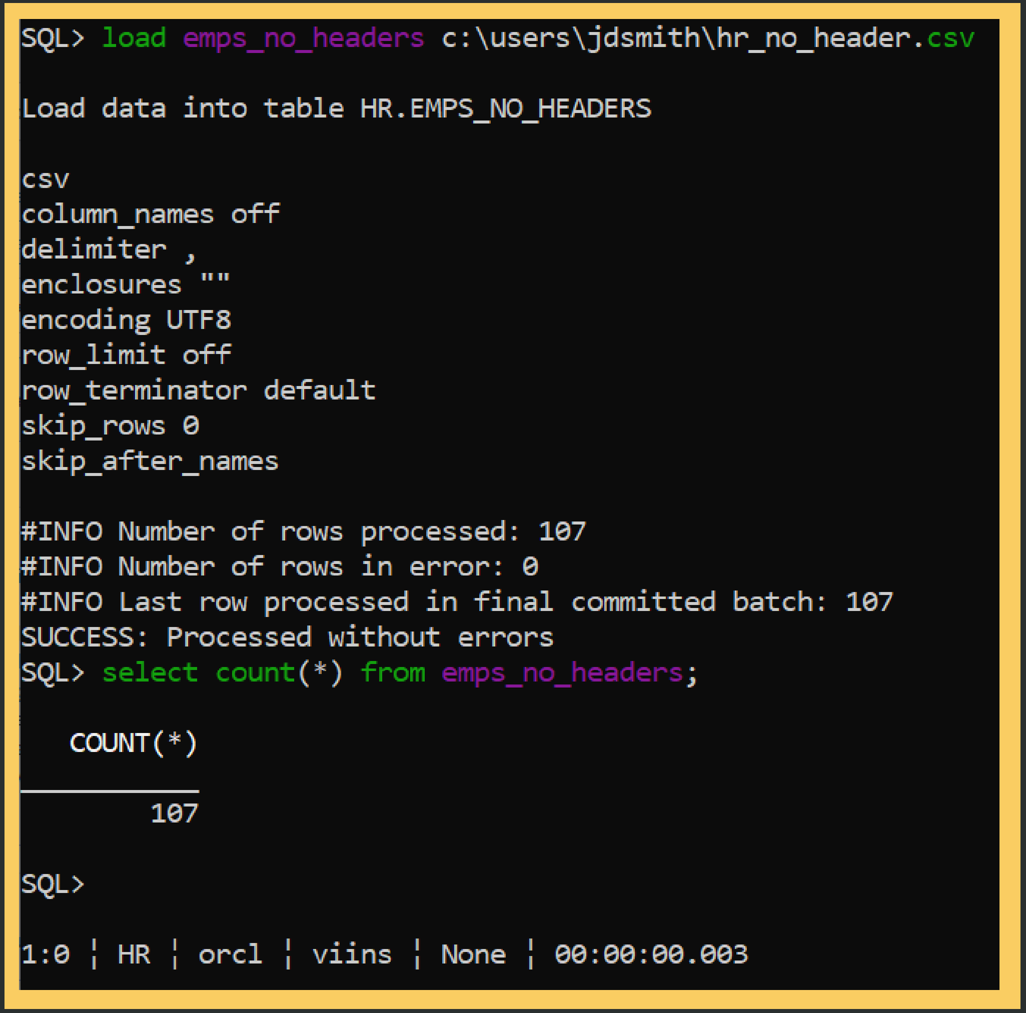
Use SET DDL to ‘shape’ how the code is generated



```
SQL> ddl regions

  CREATE TABLE "HR"."REGIONS"
   (    "REGION_ID" NUMBER CONSTRAINT "REGION_ID_NN" NOT NULL ENABLE,
        "REGION_NAME" VARCHAR2(25) COLLATE "USING_NLS_COMP"
   ) DEFAULT COLLATION "USING_NLS_COMP" SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
  CREATE UNIQUE INDEX "HR"."REG_ID_PK" ON "HR"."REGIONS" ("REGION_ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
ALTER TABLE "HR"."REGIONS" ADD CONSTRAINT "REG_ID_PK" PRIMARY KEY ("REGION_ID")
  USING INDEX "HR"."REG_ID_PK"  ENABLE;

   COMMENT ON COLUMN "HR"."REGIONS"."REGION_ID" IS 'Primary key of regions table
   COMMENT ON COLUMN "HR"."REGIONS"."REGION_NAME" IS 'Names of regions. Location
   COMMENT ON TABLE "HR"."REGIONS"  IS 'Regions table that contains region numbe
tries table.';
SQL> show ddl
STORAGE : ON
INHERIT : ON
EMIT_SCHEMA : ON
SQLTERMINATOR : ON
OID : ON
SPECIFICATION : ON
TABLESPACE : ON
SIZE_BYTE_KEYWORD : ON
PRETTY : ON
REF_CONSTRAINTS : ON
FORCE : ON
PARTITIONING : ON
```

LOAD – take delimited data and load to a table

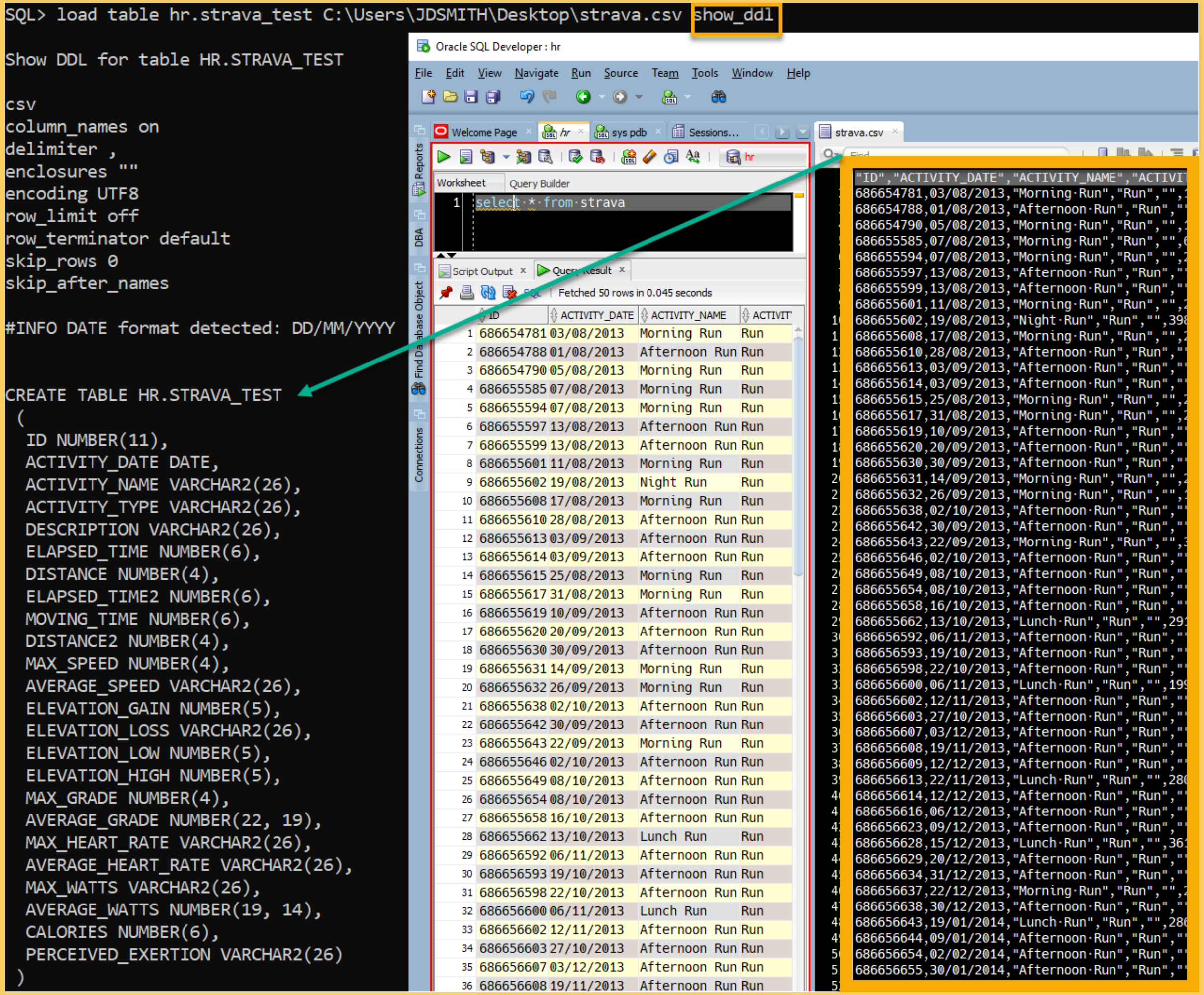
New or existing tables, file can be local or in Oracle Object Store (OSS)



```
SQL> load emps_no_headers c:\users\jdsmith\hr_no_header.csv

Load data into table HR.EMPS_NO_HEADERS

csv
column_names off
delimiter ,
enclosures ""
encoding UTF8
row_limit off
row_terminator default
skip_rows 0
skip_after_names

#INFO Number of rows processed: 107
#INFO Number of rows in error: 0
#INFO Last row processed in final committed batch: 107
SUCCESS: Processed without errors
SQL> select count(*) from emps_no_headers;

  COUNT(*)
__________
       107

SQL>

1:0 ¦ HR ¦ orcl ¦ viins ¦ None ¦ 00:00:00.003
```

Load, New Table, but…

…JUST show me the DDL for proposed table



```
SQL> load table hr.strava_test C:\Users\JDSMITH\Desktop\strava.csv show_ddl

Show DDL for table HR.STRAVA_TEST

csv
column_names on
delimiter ,
enclosures ""
encoding UTF8
row_limit off
row_terminator default
skip_rows 0
skip_after_names

#INFO DATE format detected: DD/MM/YYYY


CREATE TABLE HR.STRAVA_TEST
 (
  ID NUMBER(11),
  ACTIVITY_DATE DATE,
  ACTIVITY_NAME VARCHAR2(26),
  ACTIVITY_TYPE VARCHAR2(26),
  DESCRIPTION VARCHAR2(26),
  ELAPSED_TIME NUMBER(6),
  DISTANCE NUMBER(4),
  ELAPSED_TIME2 NUMBER(6),
  MOVING_TIME NUMBER(6),
  DISTANCE2 NUMBER(4),
  MAX_SPEED NUMBER(4),
  AVERAGE_SPEED VARCHAR2(26),
  ELEVATION_GAIN NUMBER(5),
  ELEVATION_LOSS VARCHAR2(26),
  ELEVATION_LOW NUMBER(5),
  ELEVATION_HIGH NUMBER(5),
  MAX_GRADE NUMBER(4),
  AVERAGE_GRADE NUMBER(22, 19),
  MAX_HEART_RATE VARCHAR2(26),
  AVERAGE_HEART_RATE VARCHAR2(26),
  MAX_WATTS VARCHAR2(26),
  AVERAGE_WATTS NUMBER(19, 14),
  CALORIES NUMBER(6),
  PERCEIVED_EXERTION VARCHAR2(26)
 )
```

```
1 select * from strava
```

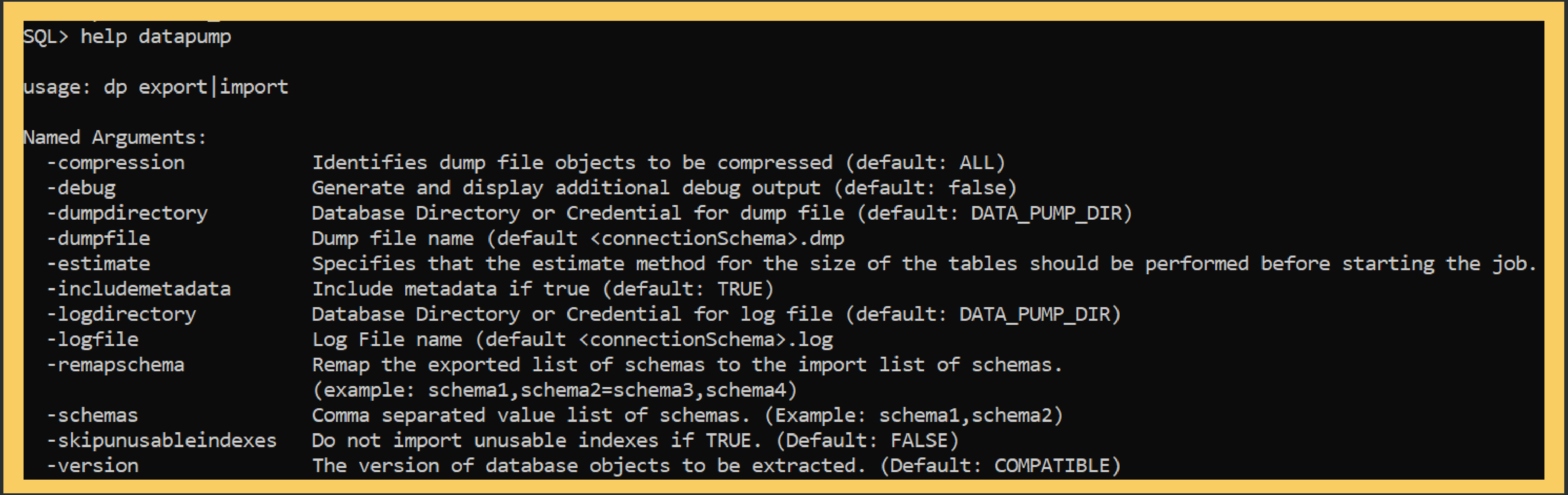
Fetched 50 rows in 0.045 seconds

	ID	ACTIVITY_DATE	ACTIVITY_NAME	ACTIVIT
1	686654781	03/08/2013	Morning Run	Run
2	686654788	01/08/2013	Afternoon Run	Run
3	686654790	05/08/2013	Morning Run	Run
4	686655585	07/08/2013	Morning Run	Run
5	686655594	07/08/2013	Morning Run	Run
6	686655597	13/08/2013	Afternoon Run	Run
7	686655599	13/08/2013	Afternoon Run	Run
8	686655601	11/08/2013	Morning Run	Run
9	686655602	19/08/2013	Night Run	Run
10	686655608	17/08/2013	Morning Run	Run
11	686655610	28/08/2013	Afternoon Run	Run
12	686655613	03/09/2013	Afternoon Run	Run
13	686655614	03/09/2013	Afternoon Run	Run
14	686655615	25/08/2013	Morning Run	Run
15	686655617	31/08/2013	Morning Run	Run
16	686655619	10/09/2013	Afternoon Run	Run
17	686655620	20/09/2013	Afternoon Run	Run
18	686655630	30/09/2013	Afternoon Run	Run
19	686655631	14/09/2013	Morning Run	Run
20	686655632	26/09/2013	Morning Run	Run
21	686655638	02/10/2013	Afternoon Run	Run
22	686655642	30/09/2013	Afternoon Run	Run
23	686655643	22/09/2013	Morning Run	Run
24	686655646	02/10/2013	Afternoon Run	Run
25	686655649	08/10/2013	Afternoon Run	Run
26	686655654	08/10/2013	Afternoon Run	Run
27	686655658	16/10/2013	Afternoon Run	Run
28	686655662	13/10/2013	Lunch Run	Run
29	686656592	06/11/2013	Afternoon Run	Run
30	686656593	19/10/2013	Afternoon Run	Run
31	686656598	22/10/2013	Afternoon Run	Run
32	686656600	06/11/2013	Lunch Run	Run
33	686656602	12/11/2013	Afternoon Run	Run
34	686656603	27/10/2013	Afternoon Run	Run
35	686656607	03/12/2013	Afternoon Run	Run
36	686656608	19/11/2013	Afternoon Run	Run

```
"ID","ACTIVITY_DATE","ACTIVITY_NAME","ACTIVIT
686654781,03/08/2013,"Morning·Run","Run","",1
686654788,01/08/2013,"Afternoon·Run","Run","
686654790,05/08/2013,"Morning·Run","Run","",1
686655585,07/08/2013,"Morning·Run","Run","",6
686655594,07/08/2013,"Morning·Run","Run","",2
686655597,13/08/2013,"Afternoon·Run","Run","
686655599,13/08/2013,"Afternoon·Run","Run","
686655601,11/08/2013,"Morning·Run","Run","",2
686655602,19/08/2013,"Night·Run","Run","",398
686655608,17/08/2013,"Morning·Run","Run","",2
686655610,28/08/2013,"Afternoon·Run","Run","
686655613,03/09/2013,"Afternoon·Run","Run","
686655614,03/09/2013,"Afternoon·Run","Run","
686655615,25/08/2013,"Morning·Run","Run","",2
686655617,31/08/2013,"Morning·Run","Run","",2
686655619,10/09/2013,"Afternoon·Run","Run","
686655620,20/09/2013,"Afternoon·Run","Run","
686655630,30/09/2013,"Afternoon·Run","Run","
686655631,14/09/2013,"Morning·Run","Run","",2
686655632,26/09/2013,"Morning·Run","Run","",1
686655638,02/10/2013,"Afternoon·Run","Run","
686655642,30/09/2013,"Afternoon·Run","Run","
686655643,22/09/2013,"Morning·Run","Run","",3
686655646,02/10/2013,"Afternoon·Run","Run","
686655649,08/10/2013,"Afternoon·Run","Run","
686655654,08/10/2013,"Afternoon·Run","Run","
686655658,16/10/2013,"Afternoon·Run","Run","
686655662,13/10/2013,"Lunch·Run","Run","",291
686656592,06/11/2013,"Afternoon·Run","Run","
686656593,19/10/2013,"Afternoon·Run","Run","
686656598,22/10/2013,"Afternoon·Run","Run","
686656600,06/11/2013,"Lunch·Run","Run","",199
686656602,12/11/2013,"Afternoon·Run","Run","
686656603,27/10/2013,"Afternoon·Run","Run","
686656607,03/12/2013,"Afternoon·Run","Run","
686656608,19/11/2013,"Afternoon·Run","Run","
686656609,12/12/2013,"Afternoon·Run","Run","
686656613,22/11/2013,"Lunch·Run","Run","",280
686656614,12/12/2013,"Afternoon·Run","Run","
686656616,06/12/2013,"Afternoon·Run","Run","
686656623,09/12/2013,"Afternoon·Run","Run","
686656628,15/12/2013,"Lunch·Run","Run","",361
686656629,20/12/2013,"Afternoon·Run","Run","
686656634,31/12/2013,"Afternoon·Run","Run","
686656637,22/12/2013,"Morning·Run","Run","",2
686656638,30/12/2013,"Afternoon·Run","Run","
686656643,19/01/2014,"Lunch·Run","Run","",286
686656644,09/01/2014,"Afternoon·Run","Run","
686656654,02/02/2014,"Afternoon·Run","Run","
686656655,30/01/2014,"Afternoon·Run","Run","
```

Data Pump (21.4)

- Very simple commands, e.g., `dp export`
- Send Exports to Oracle Object Store, automatically!
- Easy Imports for Databases in Oracle Cloud



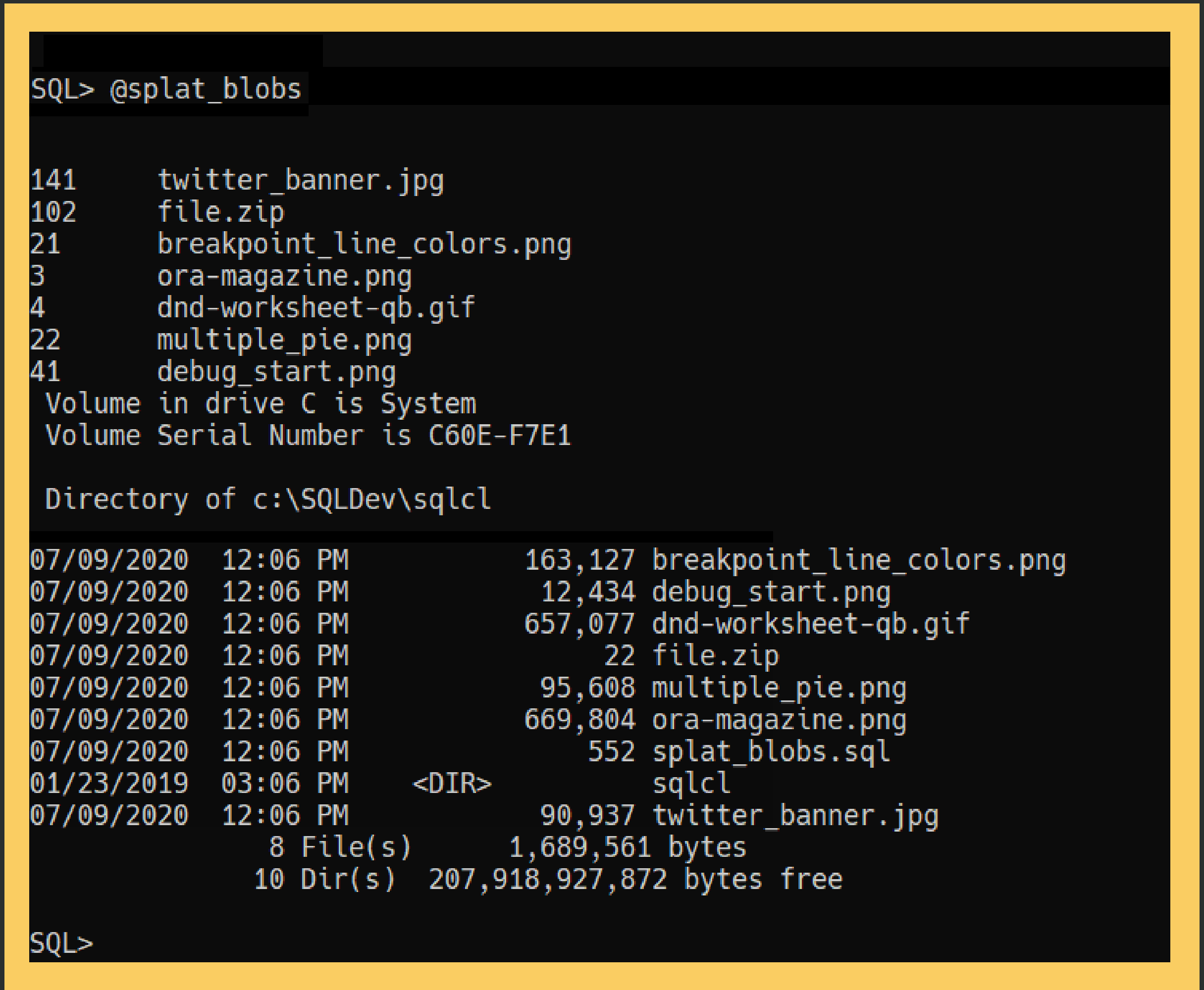
```
SQL> help datapump

usage: dp export|import

Named Arguments:
  -compression          Identifies dump file objects to be compressed (default: ALL)
  -debug                Generate and display additional debug output (default: false)
  -dumpdirectory        Database Directory or Credential for dump file (default: DATA_PUMP_DIR)
  -dumpfile             Dump file name (default <connectionSchema>.dmp
  -estimate             Specifies that the estimate method for the size of the tables should be performed before starting the job.
  -includemetadata      Include metadata if true (default: TRUE)
  -logdirectory         Database Directory or Credential for log file (default: DATA_PUMP_DIR)
  -logfile              Log File name (default <connectionSchema>.log
  -remapschema          Remap the exported list of schemas to the import list of schemas.
                        (example: schema1,schema2=schema3,schema4)
  -schemas              Comma separated value list of schemas. (Example: schema1,schema2)
  -skipunusableindexes  Do not import unusable indexes if TRUE. (Default: FALSE)
  -version              The version of database objects to be extracted. (Default: COMPATIBLE)
```

Extending the toolkit with JavaScript

```
script
// issue the SQL

var binds = {}
var  ret = util.executeReturnList('select id,file_name,content from 
media',binds);

// loop the results
FOR (i = 0; i < ret.length; i++) {
  // debug IS nice
  ctx.write( ret[i].ID  + "\t" + ret[i].FILE_NAME+ "\n");

  // GET the BLOB stream
  var blobStream =  ret[i].CONTENT.getBinaryStream(1);

  // GET the path/file handle TO WRITE TO
  var path = 
java.nio.file.FileSystems.getDefault().getPath(ret[i].FILE_NAME);

  // dump the file stream TO the file
  java.nio.file.Files.copy(blobStream,path);

}
/
!dir
```

8 lines of js…

Extract all BLOBs from a TABLE to local files on your OS



```
SQL> @splat_blobs


141     twitter_banner.jpg
102     file.zip
21      breakpoint_line_colors.png
3       ora-magazine.png
4       dnd-worksheet-qb.gif
22      multiple_pie.png
41      debug_start.png
 Volume in drive C is System
 Volume Serial Number is C60E-F7E1

 Directory of c:\SQLDev\sqlcl

07/09/2020  12:06 PM           163,127 breakpoint_line_colors.png
07/09/2020  12:06 PM            12,434 debug_start.png
07/09/2020  12:06 PM           657,077 dnd-worksheet-qb.gif
07/09/2020  12:06 PM                22 file.zip
07/09/2020  12:06 PM            95,608 multiple_pie.png
07/09/2020  12:06 PM           669,804 ora-magazine.png
07/09/2020  12:06 PM               552 splat_blobs.sql
01/23/2019  03:06 PM    <DIR>          sqlcl
07/09/2020  12:06 PM            90,937 twitter_banner.jpg
               8 File(s)      1,689,561 bytes
              10 Dir(s)  207,918,927,872 bytes free

SQL>
```

CI/CD, Schema Versioning with Liquibase

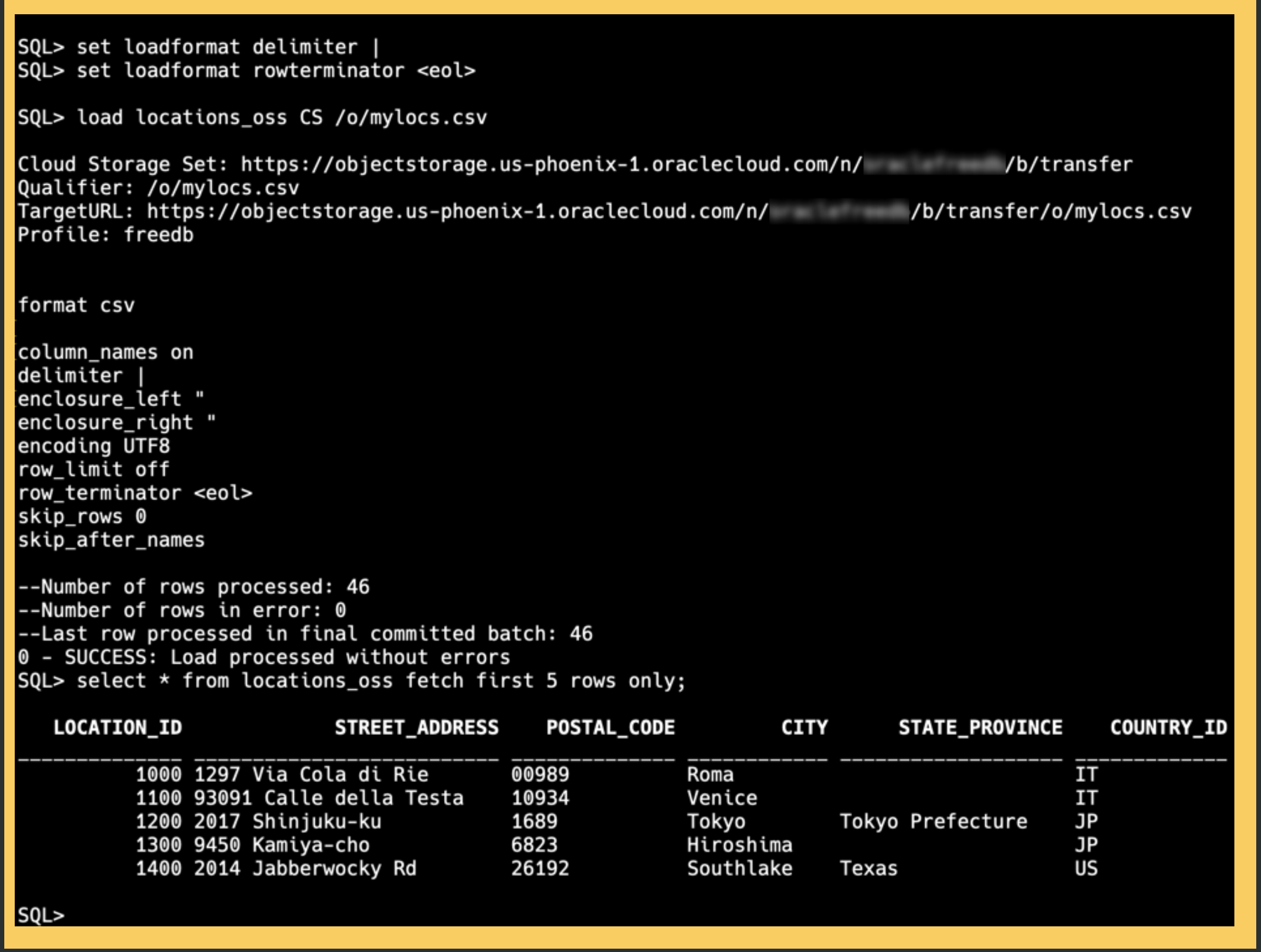
Free Liquibase Technology Enhanced for better Oracle Support

- Built into SQLcl via Liquibase (lb) command
- Automatically generate changeLogs
- changeLogs automatically sorted for creation order
- Automated rollbacks
- 100% Oracle data type, schema, and SQL scripting support

Oracle Cloud Support (OCI)

Interact with your OCI tenancies

- Object Store [read | write]
- SDK/APIs
- DBCCRED
  Manage credentials for
  DBMS_CLOUD

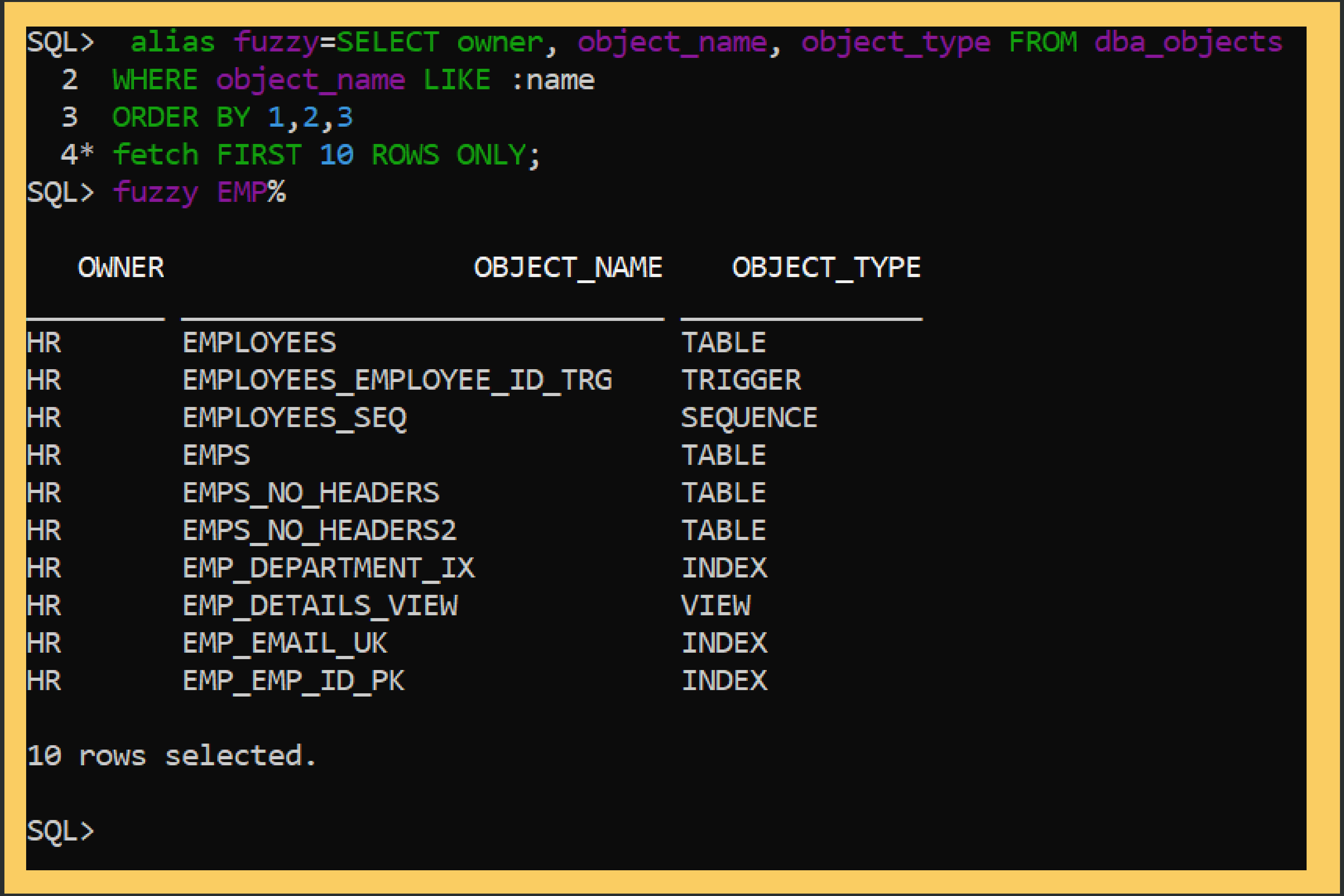


```
SQL> set loadformat delimiter |
SQL> set loadformat rowterminator <eol>

SQL> load locations_oss CS /o/mylocs.csv

Cloud Storage Set: https://objectstorage.us-phoenix-1.oraclecloud.com/n/[redacted]/b/transfer
Qualifier: /o/mylocs.csv
TargetURL: https://objectstorage.us-phoenix-1.oraclecloud.com/n/[redacted]/b/transfer/o/mylocs.csv
Profile: freedb


format csv

column_names on
delimiter |
enclosure_left "
enclosure_right "
encoding UTF8
row_limit off
row_terminator <eol>
skip_rows 0
skip_after_names

--Number of rows processed: 46
--Number of rows in error: 0
--Last row processed in final committed batch: 46
0 - SUCCESS: Load processed without errors
SQL> select * from locations_oss fetch first 5 rows only;

   LOCATION_ID             STREET_ADDRESS    POSTAL_CODE         CITY      STATE_PROVINCE    COUNTRY_ID

-------------- -------------------------- -------------- ------------ ------------------- -------------
          1000 1297 Via Cola di Rie       00989          Roma                             IT
          1100 93091 Calle della Testa    10934          Venice                           IT
          1200 2017 Shinjuku-ku           1689           Tokyo        Tokyo Prefecture    JP
          1300 9450 Kamiya-cho            6823           Hiroshima                        JP
          1400 2014 Jabberwocky Rd        26192          Southlake    Texas               US

SQL>
```

Potpourri

Alias

- Alias = shortcuts!
  - `alias xyz=select 1 from dual;`

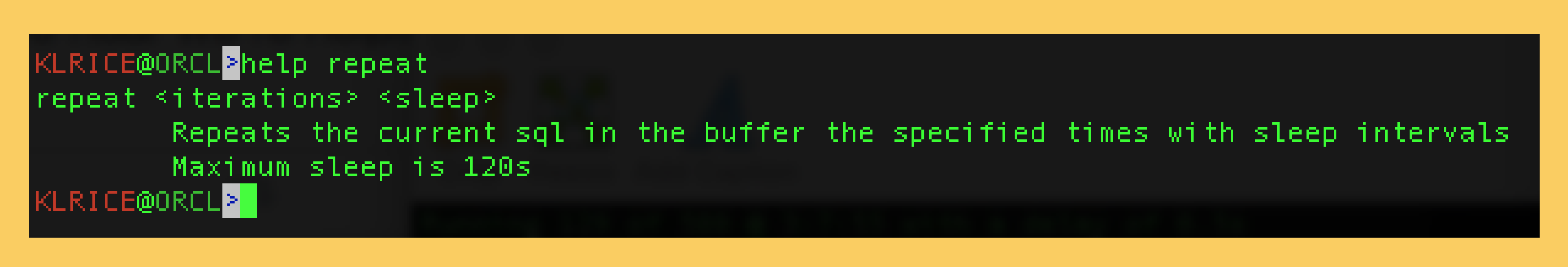


```
SQL>  alias fuzzy=SELECT owner, object_name, object_type FROM dba_objects
  2  WHERE object_name LIKE :name
  3  ORDER BY 1,2,3
  4* fetch FIRST 10 ROWS ONLY;
SQL> fuzzy EMP%

   OWNER                      OBJECT_NAME    OBJECT_TYPE

________ ____________________________ ______________
HR        EMPLOYEES                    TABLE
HR        EMPLOYEES_EMPLOYEE_ID_TRG    TRIGGER
HR        EMPLOYEES_SEQ                SEQUENCE
HR        EMPS                         TABLE
HR        EMPS_NO_HEADERS              TABLE
HR        EMPS_NO_HEADERS2             TABLE
HR        EMP_DEPARTMENT_IX            INDEX
HR        EMP_DETAILS_VIEW             VIEW
HR        EMP_EMAIL_UK                 INDEX
HR        EMP_EMP_ID_PK                INDEX

10 rows selected.

SQL>
```

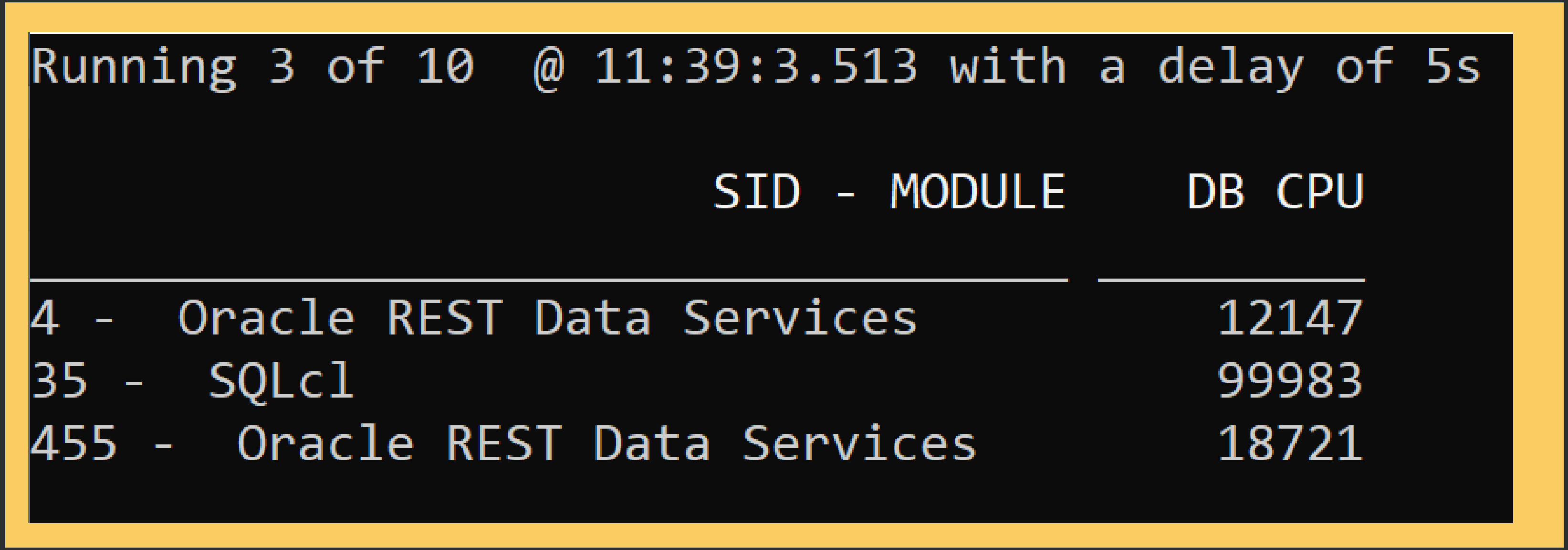
Potpourri

REPEAT

- Run a command…
- Repeatedly
- For X times with Y delay



```
KLRICE@ORCL>help repeat
repeat <iterations> <sleep>
       Repeats the current sql in the buffer the specified times with sleep intervals
       Maximum sleep is 120s
KLRICE@ORCL>
```



```
Running 3 of 10  @ 11:39:3.513 with a delay of 5s

                       SID - MODULE    DB CPU
___________________________________ _________
4 -  Oracle REST Data Services          12147
35 -  SQLcl                             99983
455 -  Oracle REST Data Services        18721
```

Thanks everyone!

Download: oracle.com/sqlcl

Current Release :21.3
    Next Release :21.4 (December 2021)