
ORACLE®

# AGENDA

- Presentation – approximately 45 minutes
- Q&A Session – approximately 15 minutes
  - Web attendees can ask questions via Q&A panel
  - Phone attendees can ask questions via Q&A panel or phone (operator assisted)

# ATTENTION – AUDIO Options

**You can**

- **either listen the audio broadcast on your computer**
- **or join teleconference (dial in)**

# Voice Streaming – Audio Broadcast

- **Listen only mode**
- **Advantage: no need to dial in**
- **What about Questions?**
  **Type your questions into WebEx Q&A panel**
- **If you prefer full audio access in order to ask questions directly, please connect to our teleconference**
- **Connect details you will find at next slide**

# ATTENTION – AUDIO INFORMATION

**Teleconference Connect details:**

1. **Conference ID: 47629030**
2. **International dial in: +44 (0) 1452 562 665**
   **US Free Call: 1866 230 1938**
   **US Local Call: 1845 608 8023**
3. **List with national toll free numbers is available in note 1148600.1**

**You can view this info anytime during the conference using**

**`Communicate > Teleconference > Join Teleconference`**

**from your WebEx menu**

ORACLE

# How to create in 5 minutes a SQL Tuning Test Case using SQLTXPLAIN

Carlos Sierra
Consulting Technical Advisor

ORACLE

# AGENDA

- Presentation – approximately 45 minutes
- Q&A Session – approximately 15 minutes
  - Web attendees can ask questions via Q&A panel only
  - Phone attendees can ask questions via Q&A panel or phone (operator assisted)
- Advisor Webcast Archived Recordings DocID 740964.1

# AUDIO INFO - Teleconference

**Connect details:**

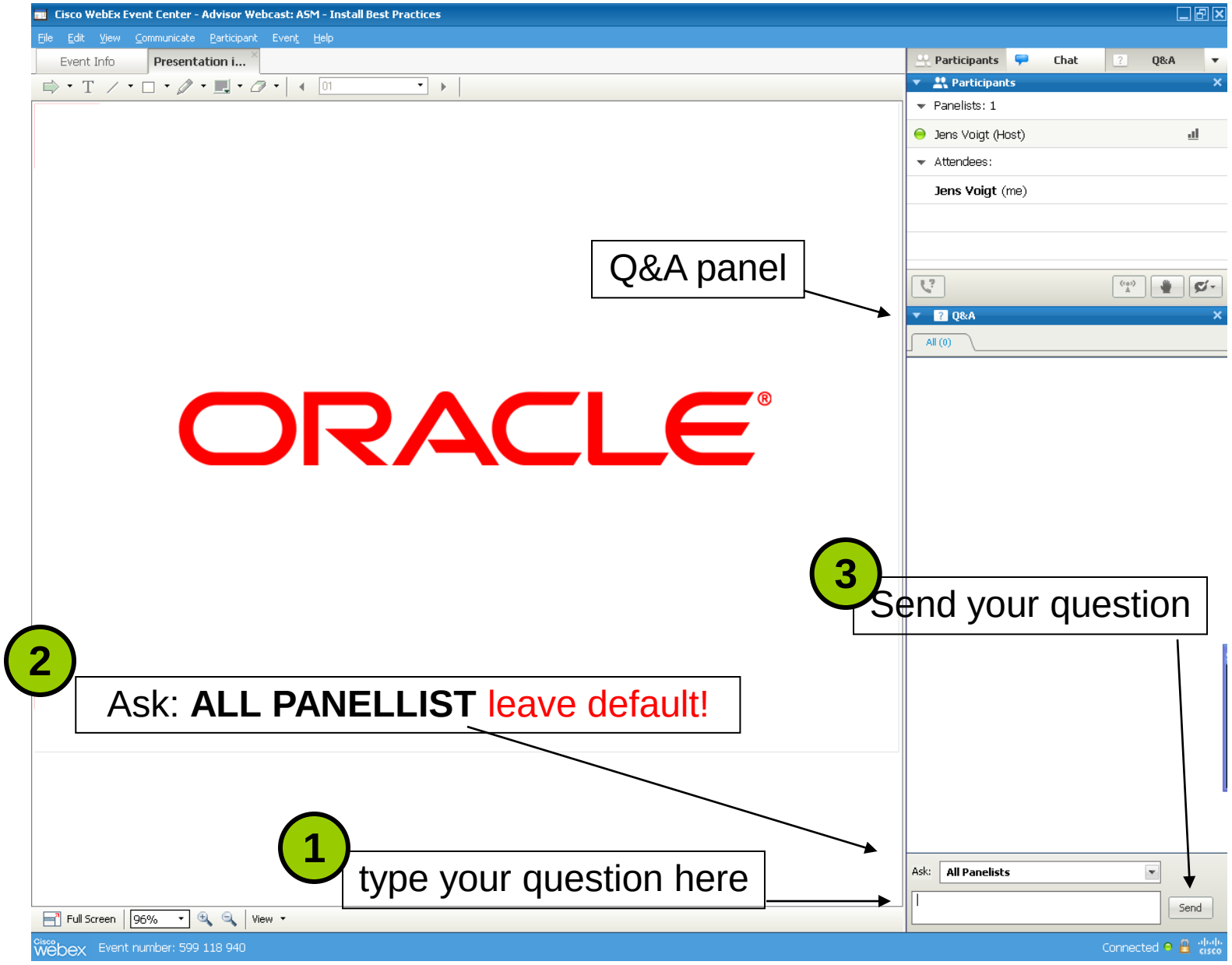
1. **Conference ID: 47629030**
2. **International dial in: +44 (0) 1452 562 665**
   **US Free Call: 1866 230 1938**
   **US Local Call: 1845 608 8023**
3. **List with national toll free numbers is available in note 1148600.1**

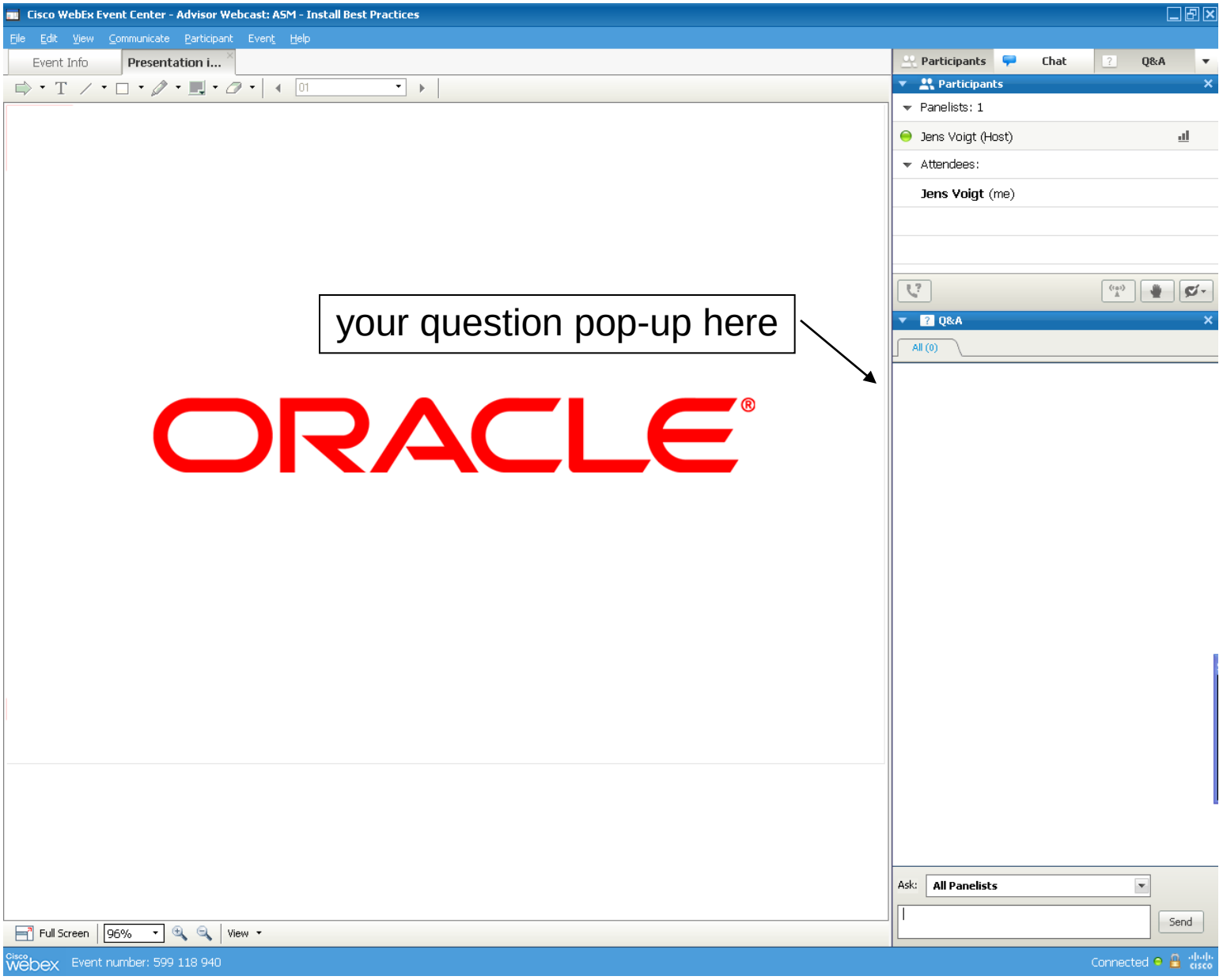
**You can view this info anytime during the conference using**

**`Communicate > Teleconference > Join Teleconference`**

**from your WebEx menu**

ORACLE





# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decision. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

ORACLE

# How to create in 5 minutes a SQL Tuning Test Case using SQLTXPLAIN

Carlos Sierra
Consulting Technical Advisor

ORACLE

# About SQLTXPLAIN (SQLT)

- Tool to diagnose SQL statements performing poorly
  - Comprehensive and consistent set of diagnostics reports
  - **Automatic Test Case (TC) extraction on each use**
  - Provides a SQLT repository for further SQL tuning actions
- Not a product but an add-on tool
  - Needs to be installed before its first use
  - Very fast life cycle (monthly?)
  - Used by Oracle Support since 1999 (as coe_xplain.sql)
- Works on 10.2+ (UNIX, Linux and Windows)
  - Prior versions for 8.1, 9i and 10.1 exist

# Topics

- **A Test Case (TC) for SQL Tuning?**
- How SQLTXPLAIN (SQLT) helps with a TC?
- What is included in a SQLT TC?
- Demo of SQLT TC implementation
- Troubleshooting a SQLT TC
- Beyond a SQLT TC

# A Test Case (TC) for SQL Tuning?

- First Objective
  - Reproduce an Execution Plan from a Source into a Target system
    - Source and Target  are usually different systems
    - Target is usually a Test system or an Oracle box
- What for?
  - Apply WHAT-IF scenarios which are not feasible to test in Source system
  - Investigate “unexpected results”
    - This type of analysis usually requires a subset of data
- Two-phases (extraction and implementation)

# What is needed to put together a TC?

- A similar system in terms of database release
  - Be aware of possible one-offs
  - In some cases a TC can be implemented on same system
- SQL text including binds declaration and values
- Metadata “create” SQL commands
  - Tables, Indexes, Views, Functions, Packages, Constraints…
  - Schema owner(s) re-map (common but not required)
- CBO Statistics
  - For schema objects and System Statistics
- CBO Environment
  - Parameters and Fix Control

# Common Methods to extract a TC

- Manual
  - Easy if Target contains same schema objects
  - Export and Import of CBO schema object statistics
  - Difficult to synchronize CBO environment and System stats
- 11g Test Case Builder (TCB)
  - Overloaded `DBMS_SQLDIAG.EXPORT_SQL_TESTCASE`
  - Inputs SQL from Memory, Text or Incident
- SQLTXPLAIN (SQLT) Test Case (TC)
  - Available for 9i, 10g and 11g
  - Inputs SQL from Memory, AWR or Text

# Topics

- A Test Case (TC) for SQL Tuning?
- **How SQLTXPLAIN (SQLT) helps with a TC?**
- What is included in a SQLT TC?
- Demo of SQLT TC implementation
- Troubleshooting a SQLT TC
- Beyond a SQLT TC

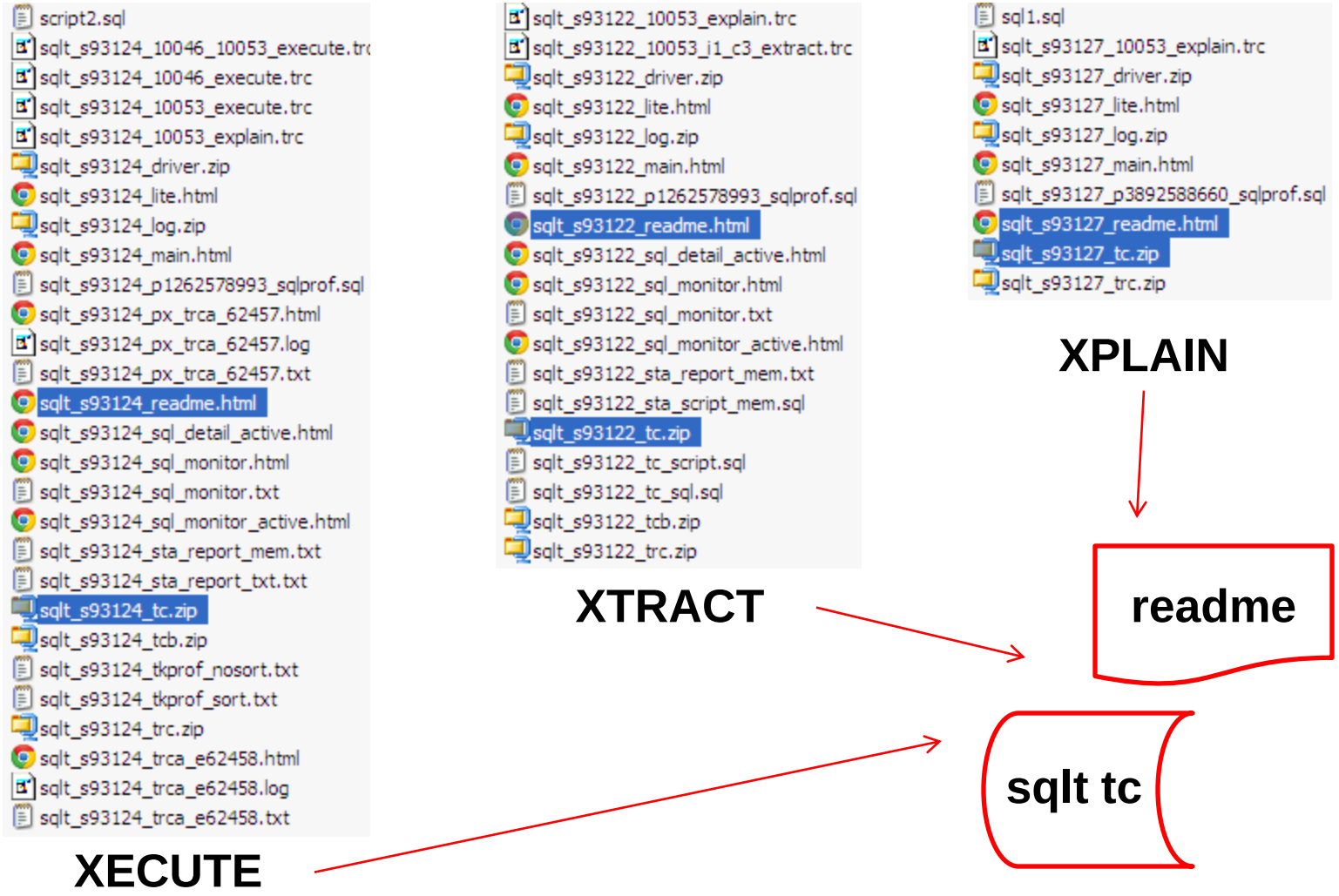
# How SQLTXPLAIN helps with a TC?

- SQLT XTRACT, XECUTE and XTRXEC
  - Automatic 11g **TCB** export on every 11g execution
    - Only if SQL still resides in memory
  - Automatic SQLT **TC** extraction on every 9i-11g execution
    - When SQL is found in memory or AWR
    - SQLT TC is also extracted with XPLAIN method
- SQLT TC provides two implementation modes
  - XPRESS
    - One driver script for a faster TC implementation
  - CUSTOM
    - Seven steps for better control during TC implementation

# Where do I find the SQL TC?

- XTRACT, XECUTE, XTRXEC and XPLAIN generate a `sqlt_sNNNNN.zip` which includes a SQLT TC
- Review first `sqlt_sNNNNN_main.html` diagnostics report
- For SQLT TC implementation you will need two files
  - Readme file `sqlt_sNNNNN_readme.html`
  - SQLT TC zip file `sqlt_sNNNNN_tc.zip`
- Look for “Implement SQLT Test Case (TC)” on readme file
  - Proceed with “Preparation” then with XPRESS or CUSTOM

# SQLT TC implementation files



# SQLT TC implementation steps

- Requirements
  - A `sqlt_sNNNNN.zip` file from Source system
  - A Target system with same database version (or higher)
    - Be aware of possible one-offs
  - SQLTXPLAIN installed on Target system
- Preparation
  - Unzip `sqlt_sNNNNN.zip`
  - Review “main” diagnostics report and “readme”
    - Find “Implement SQLT Test Case (TC)” section in readme
  - Copy `sqlt_sNNNNN_tc.zip` file to Target server
  - Follow Preparation then XPRESS or CUSTOM mode

# Typical menu from a SQLT readme

## 215187.1 SQLT XTRACT 11.4.4.1 Report: sqlt_s93896_readme.html

Instructions to perform the following:

- Export SQLT repository
- Import SQLT repository
- Using SQLT COMPARE
- Restore CBO schema statistics
- Restore CBO system statistics
- Implement SQLT Test Case (TC)
- Create TC with no SQLT dependencies
- Restore SQL Set
- Create SQL Plan Baseline from SQL Set
- Gather CBO statistics without Histograms
- Gather CBO statistics with Histograms
- List generated files

# SQLT TC Preparation

## Implement SQLT Test Case (TC)

SOURCE and TARGET systems should be similar. Proceed with Preparation followed by Express or Custom mode.

### Preparation

1. Unzip `sqlt_s93896_tc.zip` in server and navigate to TC directory.

```
unzip sqlt_s93896_tc.zip -d TC93896

cd TC93896
```

# XPRESS implementation mode

**Express (XPRESS) mode**

1. Review and execute `xpress.sh` from OS or `xpress.sql` from sqlplus.

   Option 1: `./xpress.sh`

   Option 2: `sqlplus / as sysdba @xpress.sql`

# CUSTOM implementation mode

## Custom mode

1. Create test case user and schema objects connecting as SYSDBA:

```
sqlplus / as sysdba

START sqlt_s93896_metadata.sql
```

2. Purge pre-existing s93896 from local SQLT repository connected as SYSDBA:

```
START sqlt_s93896_purge.sql
```

3. Import SQLT repository for s93896 (provide SQLTXPLAIN password):

```
HOS imp sqltxplain FILE=sqlt_s93896_exp.dmp TABLES=sqlt% IGNORE=Y
```

4. Restore CBO schema statistics for test case user connected as SYSDBA:

```
START sqlt_s93896_restore.sql
```

5. Restore CBO system statistics connected as SYSDBA:

```
START sqlt_s93896_system_stats.sql
```

6. Set the CBO environment connecting as test case user TC93896 (include optional test case user suffix):

```
CONN TC93896/TC93896

START sqlt_s93896_set_cbo_env.sql
```

7. Execute test case:

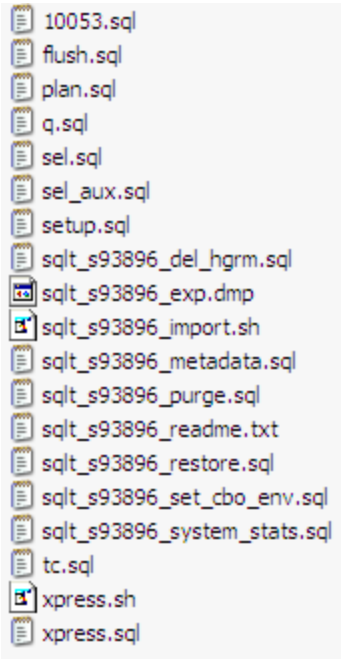
```
START tc.sql
```

# Topics

- A Test Case (TC) for SQL Tuning?
- How SQLTXPLAIN (SQLT) helps with a TC?
- **What is included in a SQLT TC?**
- Demo of SQLT TC implementation
- Troubleshooting a SQLT TC
- Beyond a SQLT TC

# What is included in a SQLT TC?

- TC zip contains several files
  - Most are used for the TC implementation
  - Some miscellaneous utilities
  - A readme file in text format



# What is included in a SQLT TC?

- Metadata script to create schema objects
- Purge script to avoid collisions on `statement_id`
- Export DMP file with SQLT repository from Source
- Restore script to map and import CBO schema stats
- System Statistics script to set them in Target
- Set CBO environment script to set CBO Parameters and Fix Control
- Test Case script to execute Query and produce Plan
- XPRESS shell and sql scripts
- Miscellaneous utilities and a readme file

# About Metadata script

- Performs a schema owner re-map into TCNNNNN
  - Re-map can be avoided by modifying “Customization Section”
- Test Case user defaults to TCNNNNN
  - A suffix can be specified to branch one TC into M cases
    - TC12345CS, TC12345A…
- Most times both defaults are fine (re-map no-suffix)
- Always review “Invalid” section at end of execution
  - Invalid synonyms are common and mostly safe to ignore
  - Invalid libraries are common and sometimes require attention
  - Invalid views require attention
    - Eliminate references to schema names on FROM

# Metadata Customization Section

```
42
43 /*****************************************************************************/
44 /* CUSTOMIZATION - BEGIN
45 /*****************************************************************************/
46
47 -- Use line below to define a NULL test case user suffix, or a specific value within the double-quotes.
48 DEF TC_USER_SUFFIX = "";
49
50 -- Remove "accept" command below if you dont need to ask for a test case user suffix. It will use the DEF above.
51 REM
52 REM Test case user suffix. Enter your initials or hit "Enter" for NULL.
53 REM
54 ACC TC_USER_SUFFIX PROMPT 'TC user suffix (opt): ';
55
56 -- Uppercase test case user suffix and remove special characters including space, quotes, etc.
57 COL TC_USER_SUFFIX NEW_V TC_USER_SUFFIX FOR A100;
58 SELECT TRANSLATE(UPPER('^^TC_USER_SUFFIX'), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ''`~!@#$%^*()-_=+[]{}\|;:",.<>/
59
60 -- Create test case user.
61 DEF TC_USER = TC93896^^TC_USER_SUFFIX.;
62 GRANT DBA TO ^^TC_USER. IDENTIFIED BY ^^TC_USER.;
63 GRANT CTXAPP TO ^^TC_USER.;
64
65 -- Use DEF command(s) below if you want to consolidate objects into one test case user (recommended).
66 DEF SCHEMA_SH                                = '^^TC_USER.'
67
68 -- Un-comment DEF command(s) below ONLY if you want to create objects into original owner(s). (not recommended).
69 -- DEF SCHEMA_SH                                = 'SH'
70
71 /*****************************************************************************/
72 /* CUSTOMIZATION - END
73 /*****************************************************************************/
74
```

# About Purge script

- SQLT repository for `statement_id` (NNNNN) is about to be imported
- Purge script deletes prior repository entries for `statement_id` NNNNN
  - Avoids possible but very infrequent collisions on NNNNN
- Purge script becomes important when TC is refreshed
  - In such case manually drop user TCNNNNN before hand
- Purge script only executes a SQLT library

```
EXEC sqltxplain.sqlt$a.purge_repository(93896,
  93896);
```

# About Exported DMP file

- An export DMP file is created on Source when SQLT is executed (XTRACT/XECUTE/XTRXEC/XPLAIN)
- This DMP file contains the entries on SQLT repository for `statement_id` NNNNN
- No application data is included into the DMP file
- Import command gets all entries for NNNNN including a STATTAB staging table

```
imp sqltxplain FILE=sqlt_s93896_exp.dmp TABLES=sqlt% IGNORE=Y
```

- Default utility is original export/import
  - SQLT can be configured to use Data Pump instead

# About Restore script

- Restore (import) schema object CBO statistics for all tables identified by SQLT
- Re-maps CBO statistics into TCNNNNN user
- Re-maps extended statistics columns as per expression or column group
- Uses `DBMS_STATS.IMPORT_TABLE_STATS`
- If re-map of “owner” is not desired use `sqlt/utl/sqltimp.sql` instead

# About System Statistics script

- Deletes CBO System Statistics on Target then restores (imports) using values from Source
- Calls `DBMS_STATS.DELETE_SYSTEM_STATS` first
- Calls then `DBMS_STATS.SET_SYSTEM_STATS`

# About Set CBO Environment script

- Contains CBO Environment from SQLT session in Source
- Executed after connecting as TC user
- Sets Optimizer Features Enable (OFE)
- Sets non-default CBO Parameters
  - Using `ALTER SESSION` and `ALTER SYSTEM` commands
- Sets non-default Bug Fix Control
- Sets default CBO Parameters
- Sets default Bug Fix Control

# About TC script

- Executes SQL statement script `q.sql` followed by plan script `plan.sql`
- Script `q.sql` contains
  - Optimizer environment for particular SQL when captured from memory
  - Bind variables
    - Declaration and values assignment (if applicable)
  - SQL text for SQL statement being analyzed
- Script `plan.sql` displays execution plan calling `DBMS_XPLAN.DISPLAY_CURSOR`

# About XPRESS sql and sh scripts

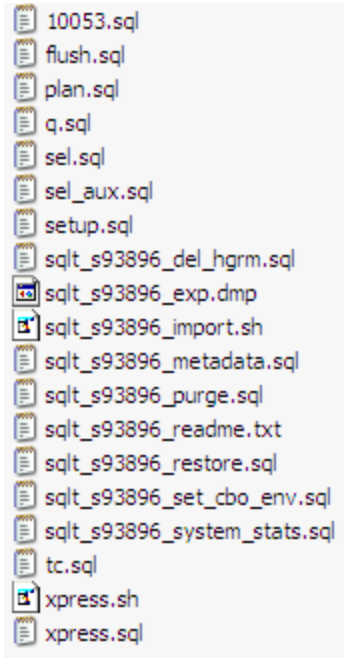
- Shell script xpress.sh executes SQL script xpress.sql connected as SYSDBA

```
1  # Implements SQLT TC Express Mode.
2  # Just execute ". xpress.sh" from OS.
3  sqlplus / as sysdba @xpress.sql
4
```

- Script xpress.sql executes all same 7 steps from CUSTOM mode
- TC implementation can be restarted from intermediate steps

# XPRESS script xpress.sql

```
REM Implements SQLT TC Express Mode.
REM Just execute "./xpress.sh" or "sqlplus / as sysdba @xpress.sql" from OS.
SET ECHO OFF;
CL SCR
PAU 1/7 Press ENTER to create TC user and schema objects for statement_id 93896.
SET ECHO ON;
@@sqlt_s93896_metadata.sql
SET ECHO OFF;
PRO
PAU 2/7 Press ENTER to purge statement_id 93896 from SQLT repository.
SET ECHO ON;
@@sqlt_s93896_purge.sql
SET ECHO OFF;
PRO
PAU 3/7 Press ENTER to import SQLT repository for statement_id 93896.
SET ECHO ON;
HOS imp sqltxplain FILE=sqlt_s93896_exp.dmp TABLES=sqlt% IGNORE=Y
SET ECHO OFF;
PRO
PAU 4/7 Press ENTER to restore schema object stats for &&tc_user..
SET ECHO ON;
@@sqlt_s93896_restore.sql
SET ECHO OFF;
PRO
PAU 5/7 Press ENTER to restore system statistics.
SET ECHO ON;
@@sqlt_s93896_system_stats.sql
SET ECHO OFF;
PRO
PAU 6/7 Press ENTER to connect as &&tc_user. and set CBO env.
SET ECHO ON;
CONN &&tc_user./&&tc_user.
@@sqlt_s93896_set_cbo_env.sql
SET ECHO OFF;
PRO
PAU 7/7 Press ENTER to execute test case.
SET ECHO ON;
@@tc.sql
```



# CUSTOM implementation mode

## Custom mode

1. Create test case user and schema objects connecting as SYSDBA:

```
sqlplus / as sysdba

START sqlt_s93896_metadata.sql
```

2. Purge pre-existing s93896 from local SQLT repository connected as SYSDBA:

```
START sqlt_s93896_purge.sql
```

3. Import SQLT repository for s93896 (provide SQLTXPLAIN password):

```
HOS imp sqltxplain FILE=sqlt_s93896_exp.dmp TABLES=sqlt% IGNORE=Y
```

4. Restore CBO schema statistics for test case user connected as SYSDBA:

```
START sqlt_s93896_restore.sql
```

5. Restore CBO system statistics connected as SYSDBA:

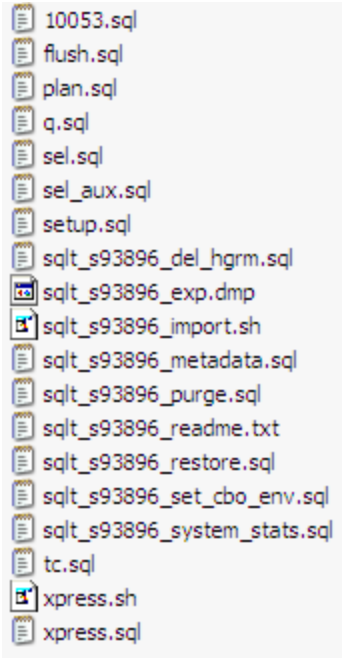
```
START sqlt_s93896_system_stats.sql
```

6. Set the CBO environment connecting as test case user TC93896 (include optional test case user suffix):

```
CONN TC93896/TC93896

START sqlt_s93896_set_cbo_env.sql
```

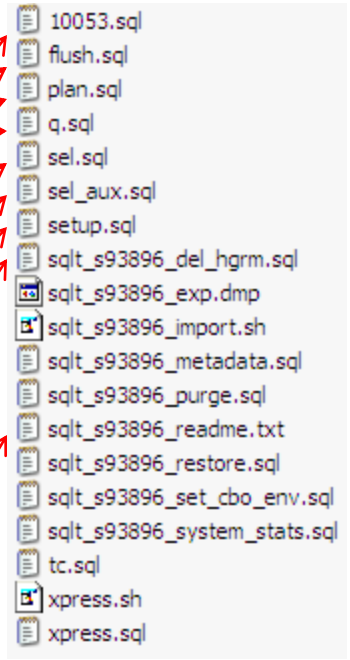
7. Execute test case:

```
START tc.sql
```



# About Miscellaneous Utilities and Files

- SQL statement script `q.sql`
- Display plan script `plan.sql`
- Flush shared pool script `flush.sql`
- EVENT 10053 script `10053.sql`
- Selectivity script `sel.sql`
- Auxiliary selectivity script `sel_aux.sql`
- Delete TC Histograms script `sqlt_sNNNNN_del_hgrm.sql`
- Text readme `sqlt_sNNNNN_readme.txt`
- Setup script `setup.sql` for stand-alone TC



# Topics

- A Test Case (TC) for SQL Tuning?
- How SQLTXPLAIN (SQLT) helps with a TC?
- What is included in a SQLT TC?
- **Demo of SQLT TC implementation**
- Troubleshooting a SQLT TC
- Beyond a SQLT TC

# D E M O N S T R A T I O N

# Plan from source

**Execution Plan phv:2874174940 [B] [W] sqlt_phv:43213 sqlt_phv2:79490 source:GV$SQL_PLAN**

SQL Text: [-]

```
SELECT /*+ gather_plan_statistics monitor */
       SUBSTR(NVL(r.country_name, 'GRAND')||' - '||
       NVL(c.cust_state_province, 'TOTAL'), 1, 50) place,
       SUM(p.total_cost) total_cost
  FROM countries r,
       customers c,
       profits p
 WHERE r.country_region = :b1
   AND r.country_id = c.country_id
   AND c.cust_id = p.cust_id
 GROUP BY ROLLUP
       (r.country_name, c.cust_state_province)
```

SQL: [+]

| ID | Exec Ord | Operation | Go To | More | Peek Bind | Capt Bind | Cost[2] | Estim Card | LAST Starts | LAST Output Rows | L Ove Est |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 10 | SELECT STATEMENT | | | | | 1432 | 411 | 1 | 0 | |
| 1 | 9 | SORT GROUP BY ROLLUP | | [+] | | | 1432 | 411 | 1 | 0 | |
| 2 | 8 | . HASH JOIN | | [+] | | | 1430 | 16566 | 1 | 0 | |
| 3 | 1 | .. TABLE ACCESS FULL COUNTRIES | [+] | [+] | [+] | [+] | 3 | 4 | 1 | 1 | |
| 4 | 7 | .. HASH JOIN | | [+] | | | 1426 | 82112 | 1 | 916039 | |
| 5 | 2 | ... TABLE ACCESS FULL CUSTOMERS | [+] | [+] | | | 406 | 55500 | 1 | 55500 | |
| 6 | 6 | ... PARTITION RANGE ALL | | [+] | | | 687 | 82112 | 1 | 916039 | |
| 7 | 5 | .... HASH JOIN | | [+] | | | 687 | 82112 | 28 | 916039 | |
| 8 | 3 | ....+ TABLE ACCESS FULL COSTS | [+] | [+] | | | 137 | 82112 | 28 | 82112 | |
| 9 | 4 | ....+ TABLE ACCESS FULL SALES | [+] | [+] | | | 530 | 918843 | 16 | 918843 | |

# Follow Preparation and XPRESS mode

## Implement SQLT Test Case (TC)

SOURCE and TARGET systems should be similar. Proceed with Preparation followed by Express or Custom mode.

### Preparation

1. Unzip `sqlt_s93896_tc.zip` in server and navigate to TC directory.

```
unzip sqlt_s93896_tc.zip -d TC93896

cd TC93896
```

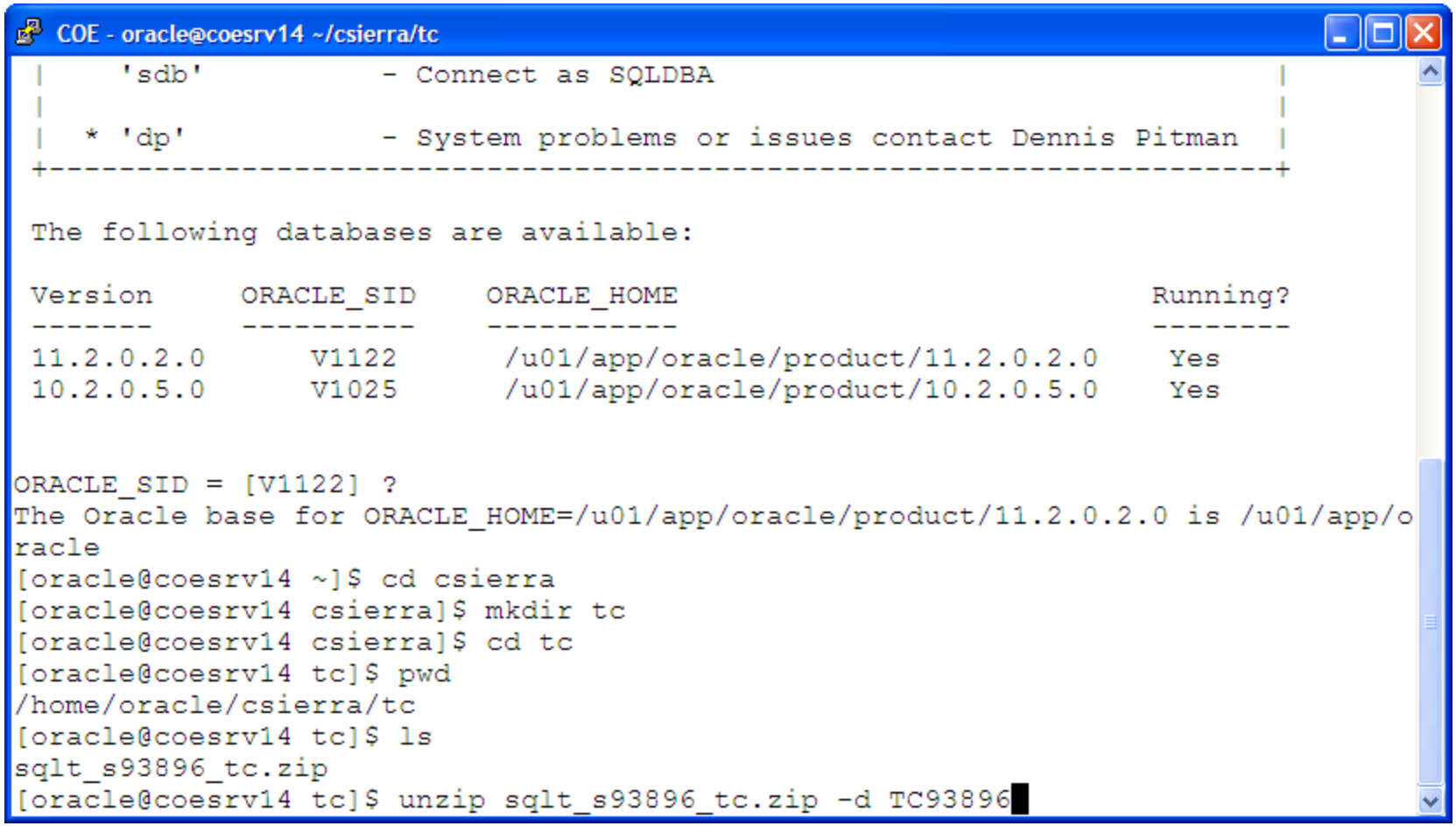
### Express (XPRESS) mode

1. Review and execute `xpress.sh` from OS or `xpress.sql` from sqlplus.

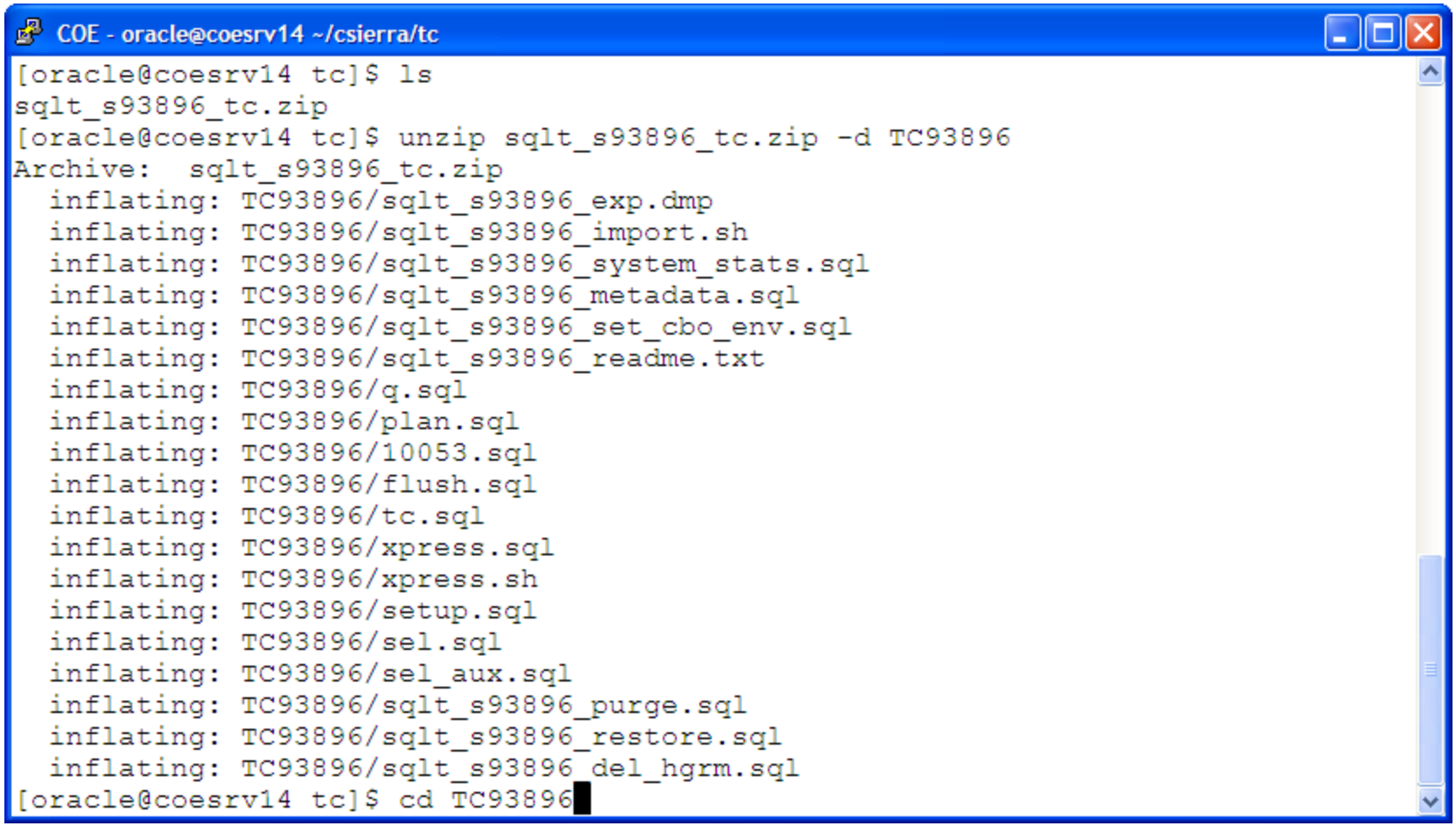
   Option 1: `./xpress.sh`

   Option 2: `sqlplus / as sysdba @xpress.sql`

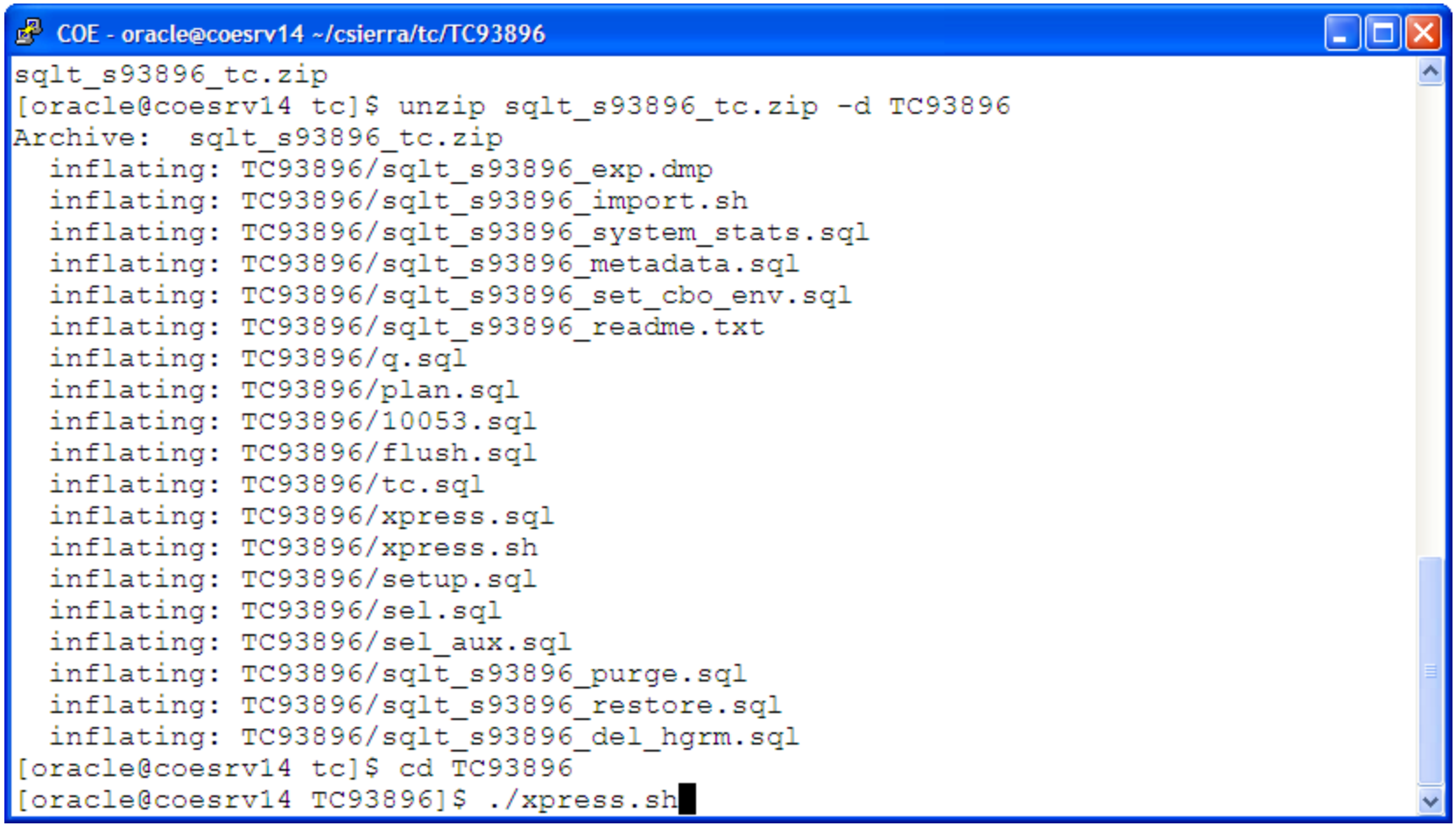
# Copy TC zip file and unzip it



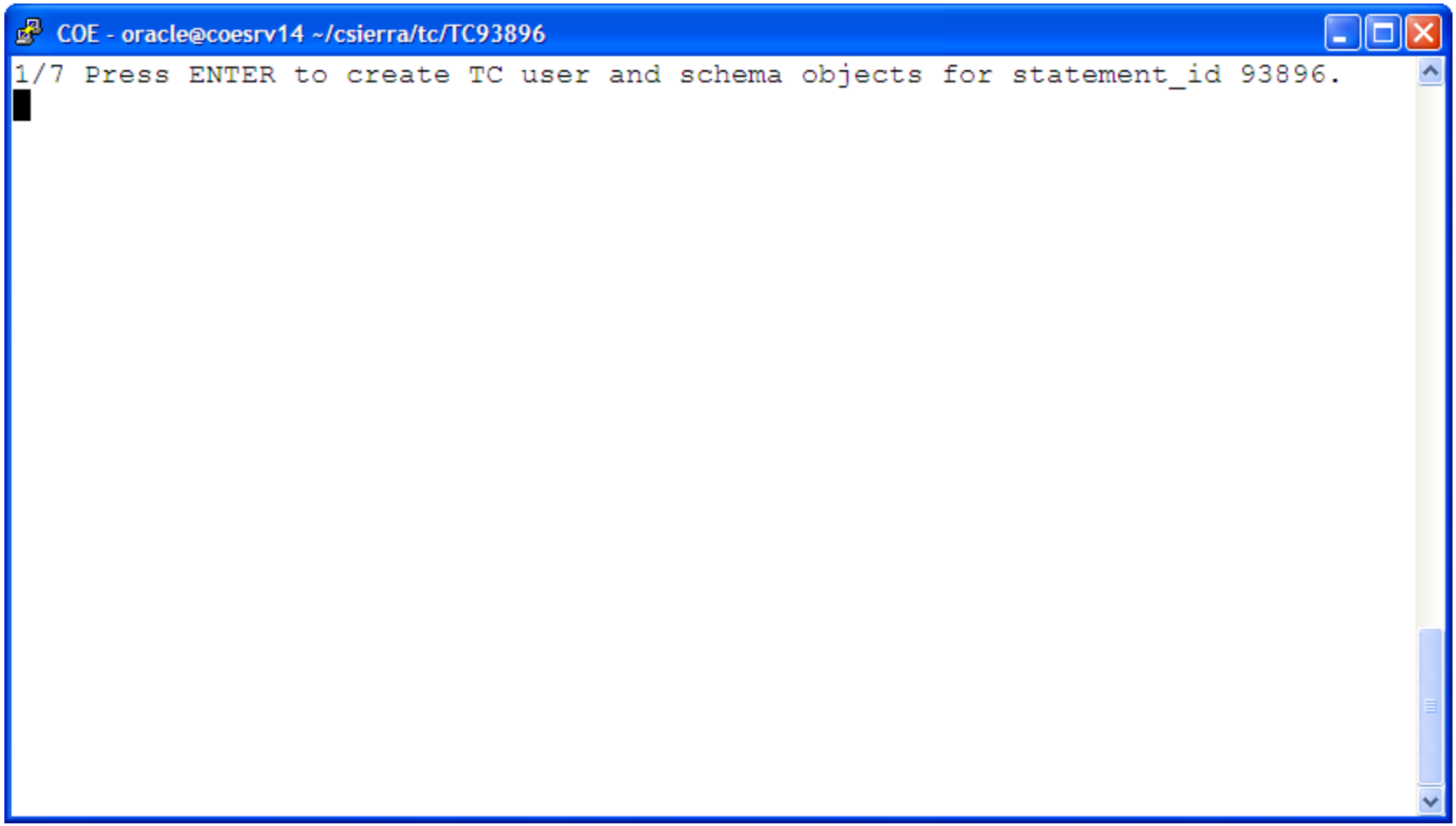
# Navigate to TC directory



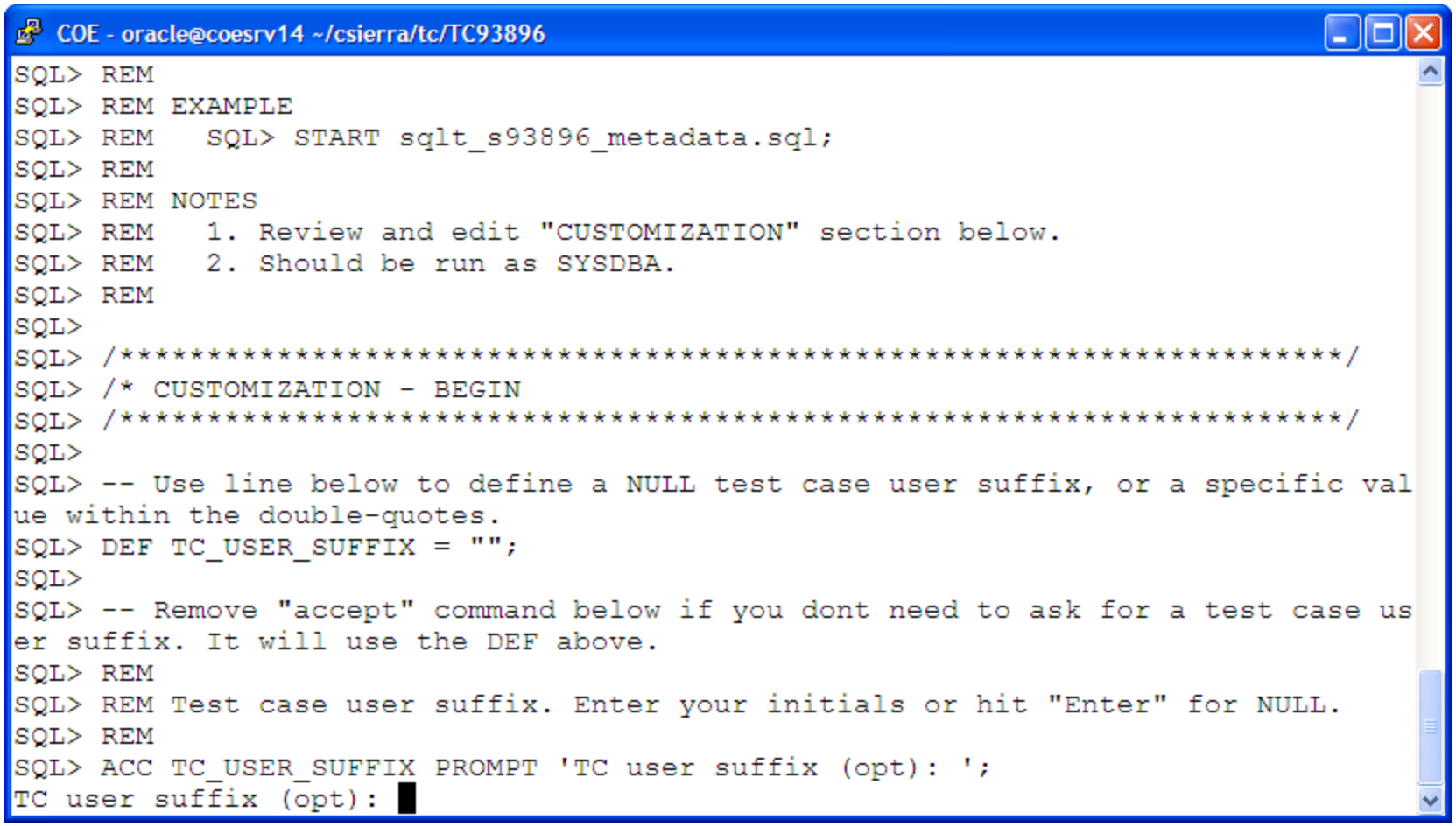
# Execute xpress.sh



# Step 1: Create TC user and schema objects



# Leave TC user suffix empty and hit enter

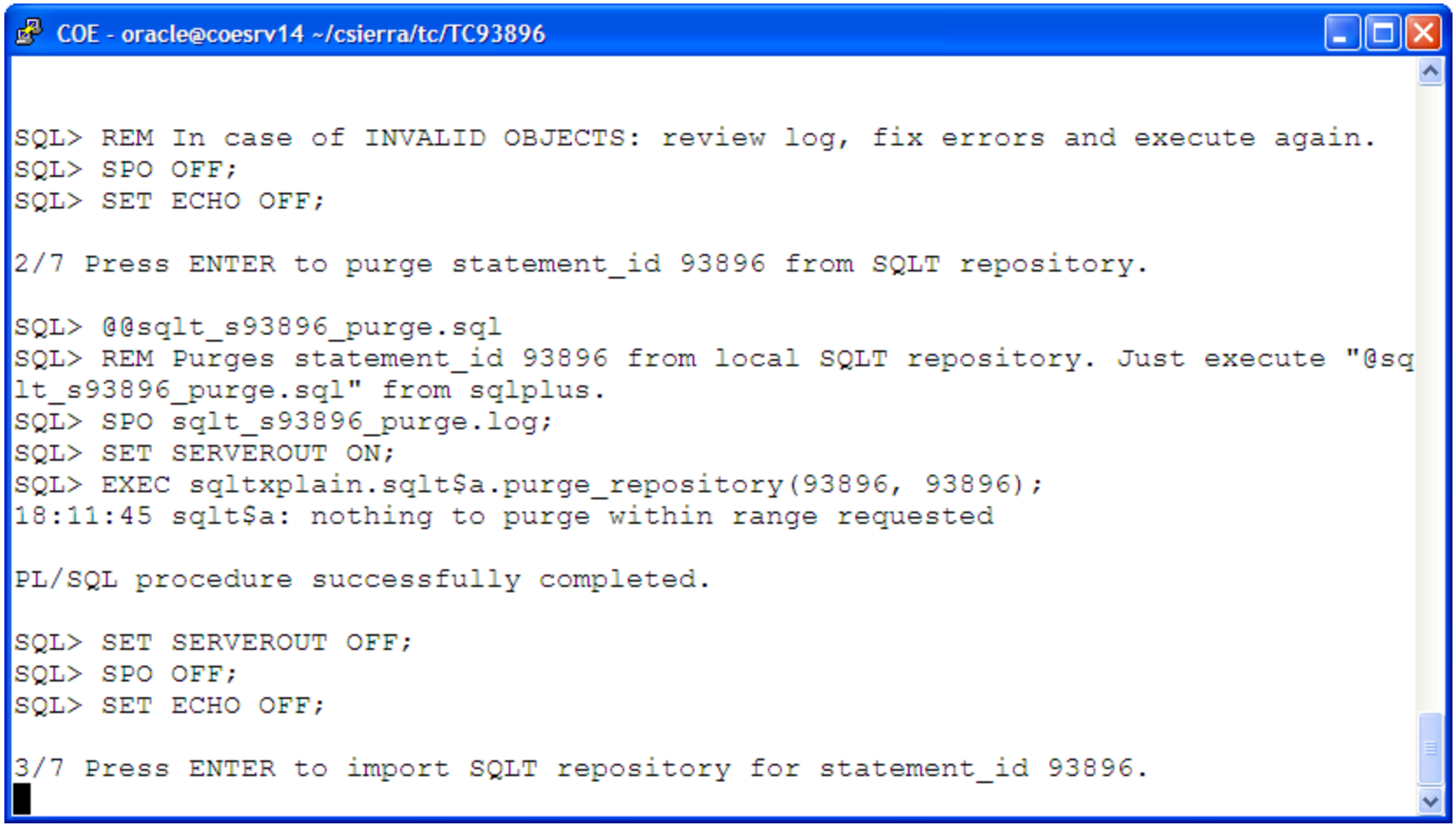


# Step 2: Purge statement_id from SQLT repository

COE - oracle@coesrv14 ~/csierra/tc/TC93896

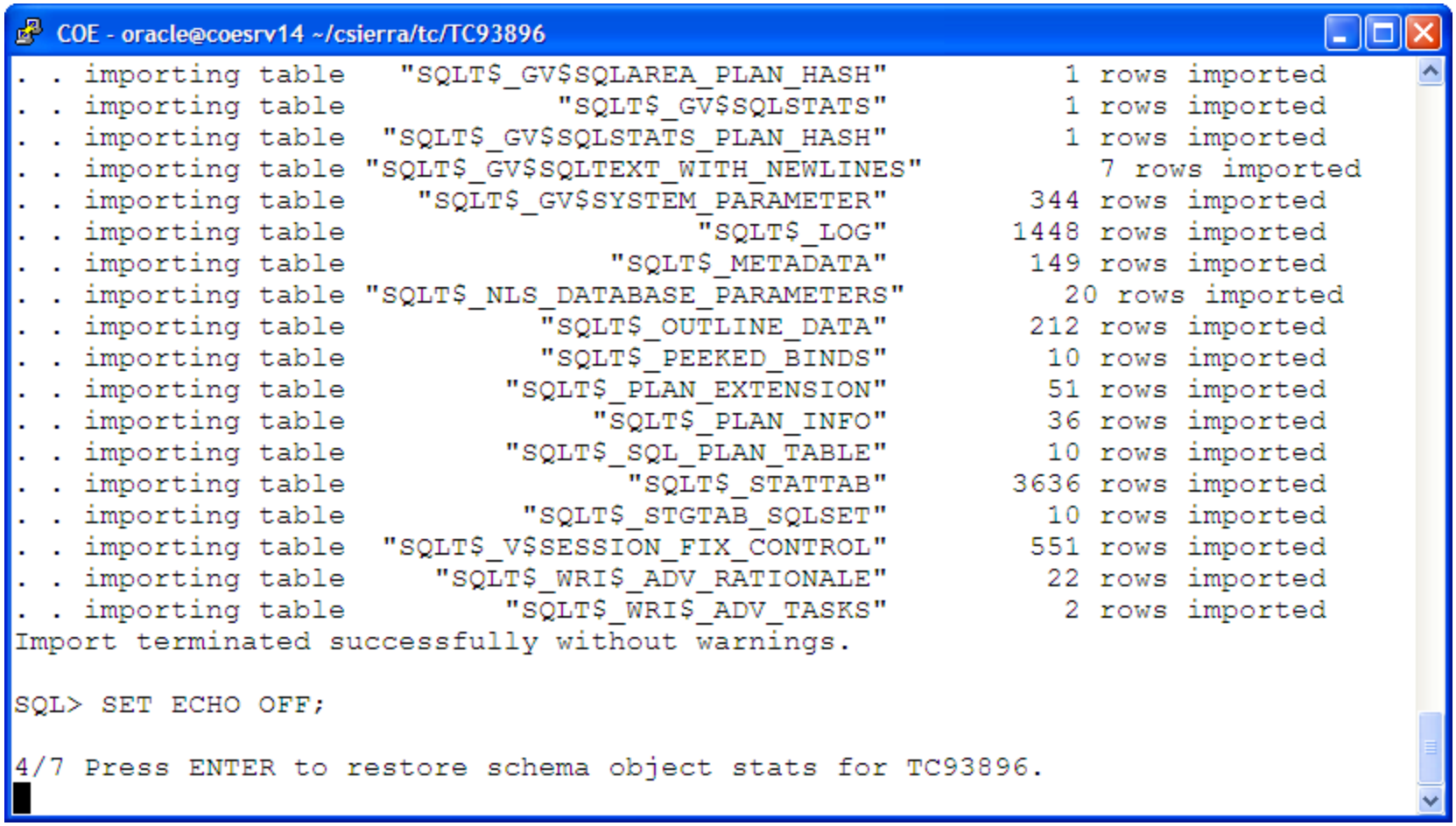
```
VALID INDEX TC93896 COUNTRIES_PK
VALID INDEX TC93896 CUSTOMERS_GENDER_BIX
VALID INDEX TC93896 CUSTOMERS_MARITAL_BIX
VALID INDEX TC93896 CUSTOMERS_PK
VALID INDEX TC93896 CUSTOMERS_YOB_BIX
VALID INDEX TC93896 SALES_CHANNEL_BIX
VALID INDEX TC93896 SALES_CUST_BIX
VALID INDEX TC93896 SALES_PROD_BIX
VALID INDEX TC93896 SALES_PROMO_BIX
VALID INDEX TC93896 SALES_TIME_BIX
VALID VIEW TC93896 PROFITS



:INVALID_OBJECTS
--------------------------------------------------------------------------------


SQL> REM In case of INVALID OBJECTS: review log, fix errors and execute again.
SQL> SPO OFF;
SQL> SET ECHO OFF;

2/7 Press ENTER to purge statement_id 93896 from SQLT repository.
```

# Step 3: Import SQLT repository

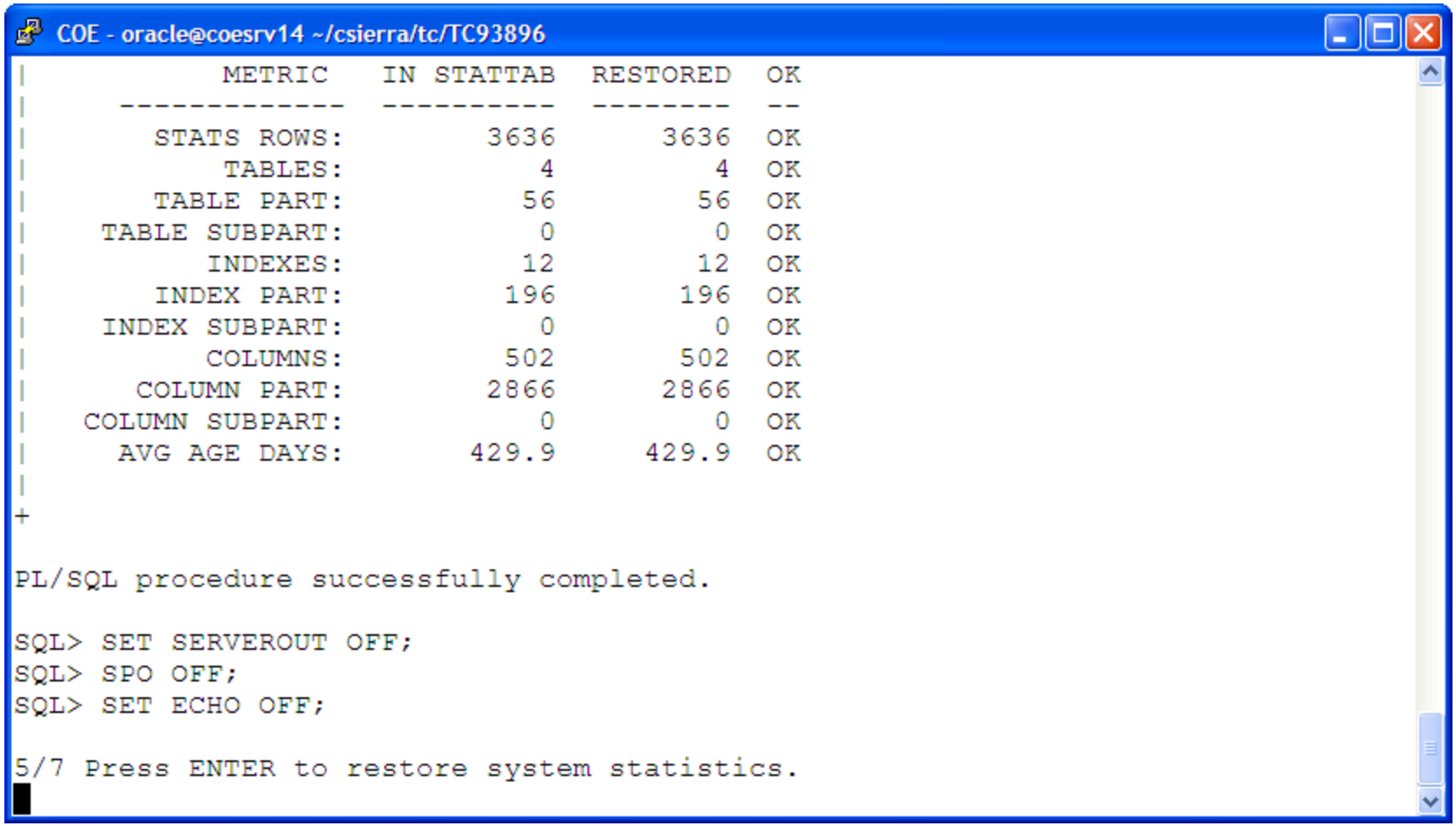


# Provide SQLTXPLAIN password

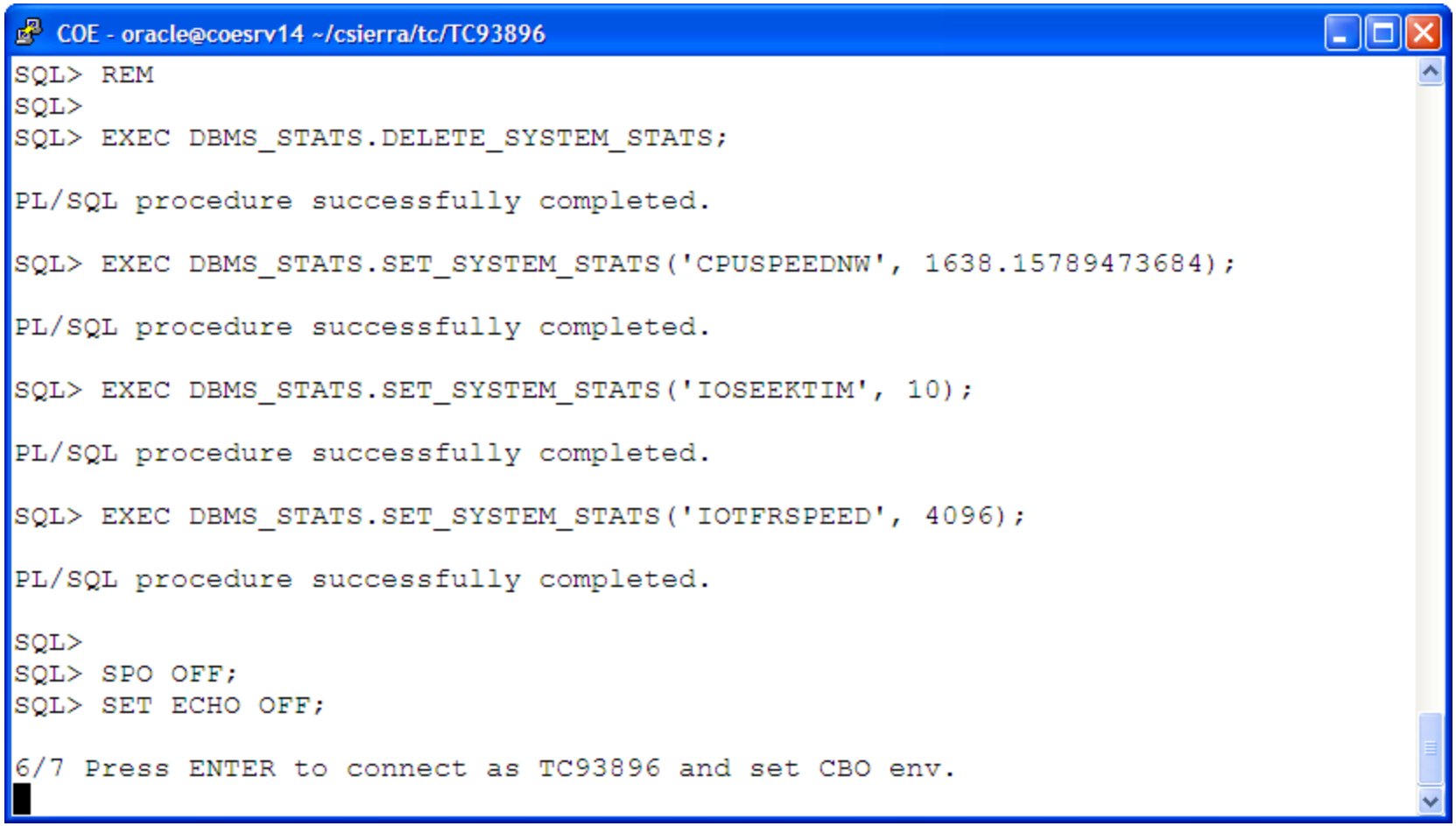
# Step 4: Restore CBO schema statistics



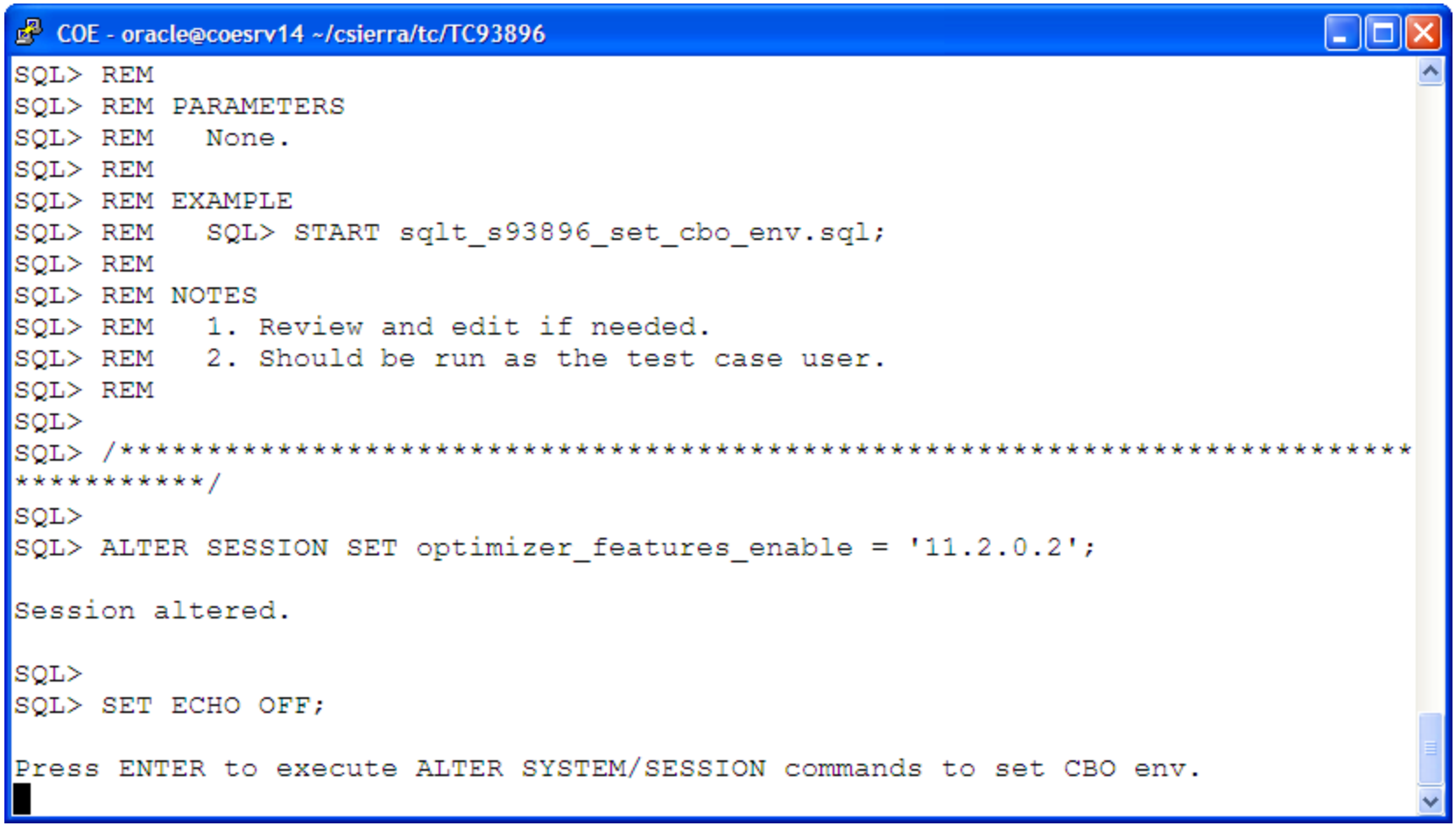
# Step 5: Restore System Statistics



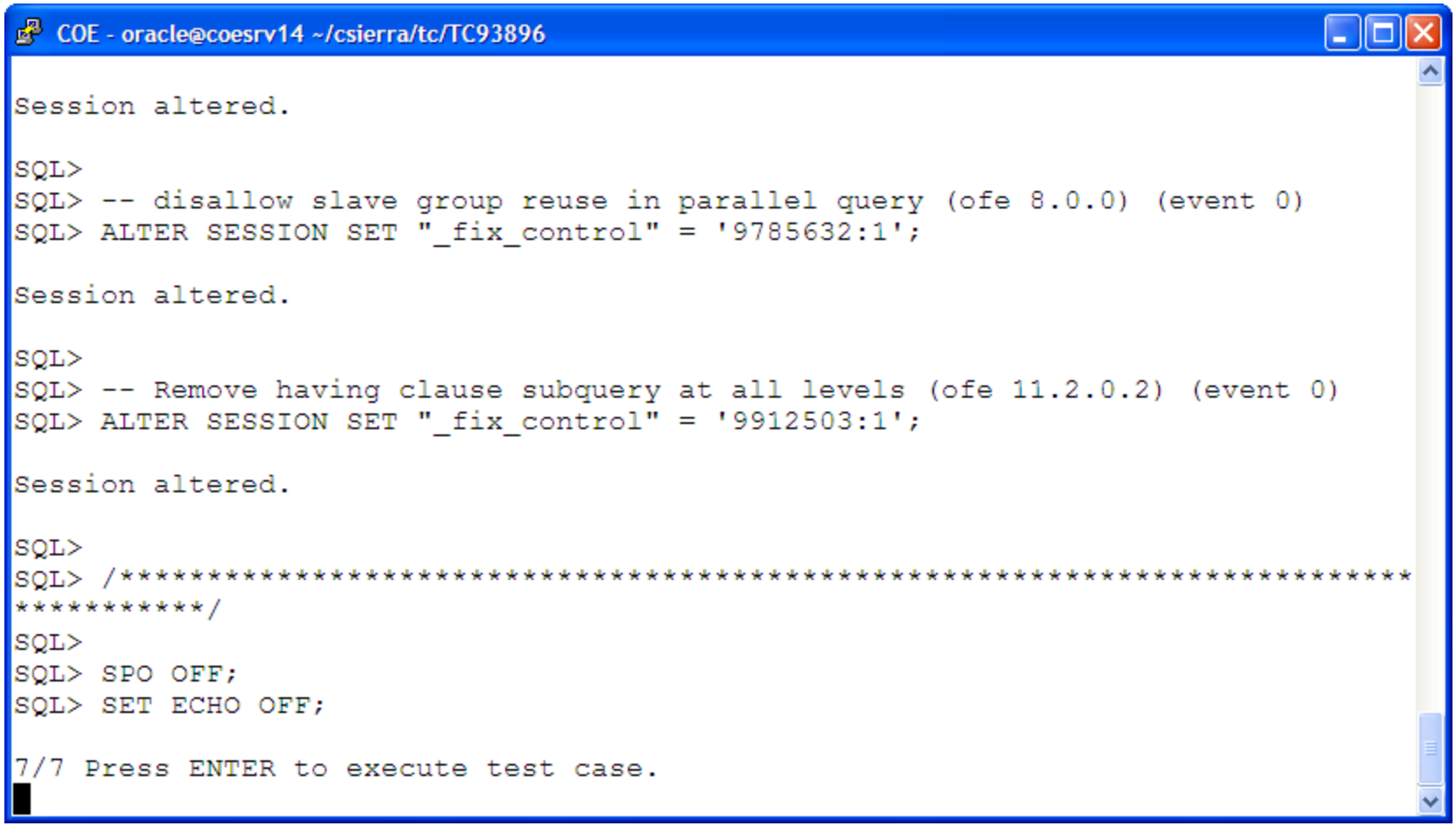
# Step 6: Connect as TC user and set CBO environment



# Acknowledge ALTER SYSTEM commands



# Step 7: Execute Test Case statement and get Plan



# Review Plan

COE - oracle@coesrv14 ~/csierra/tc/TC93896

```
|*  7 |      HASH JOIN          |           | 82112 |  4089K|       |   687   (5)
| 00:00:01 |       |       |
|   8 |       TABLE ACCESS FULL| COSTS     | 82112 |  1924K|       |   137   (1)
| 00:00:01 |     1 |    28 |
|   9 |       TABLE ACCESS FULL| SALES     |   918K|    23M|       |   530   (3)
| 00:00:01 |     1 |    28 |
---------------------------------------------------------------------------------
----------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("R"."COUNTRY_ID"="C"."COUNTRY_ID")
   3 - filter("R"."COUNTRY_REGION"=:B1)
   4 - access("C"."CUST_ID"="S"."CUST_ID")
   7 - access("C"."TIME_ID"="S"."TIME_ID" AND "C"."PROD_ID"="S"."PROD_ID" AND
              "C"."CHANNEL_ID"="S"."CHANNEL_ID" AND "C"."PROMO_ID"="S"."PROMO_ID
")


36 rows selected.

SQL> SPO OFF;
SQL>
```

# Plan on target

```
Plan hash value: 2874174940

-------------------------------------------------------------------------------------------------------------
| Id  | Operation              | Name      | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |           |       |       |       |  1432 (100)|          |       |       |
|   1 |  SORT GROUP BY ROLLUP  |           |   411 | 39045 |       |  1432   (3)| 00:00:01 |       |       |
|*  2 |   HASH JOIN            |           | 16566 |  1536K|       |  1430   (3)| 00:00:01 |       |       |
|*  3 |    TABLE ACCESS FULL   | COUNTRIES |     4 |    92 |       |     3   (0)| 00:00:01 |       |       |
|*  4 |    HASH JOIN           |           | 82112 |  5773K|  1792K|  1426   (3)| 00:00:01 |       |       |
|   5 |     TABLE ACCESS FULL  | CUSTOMERS | 55500 |  1138K|       |   406   (1)| 00:00:01 |       |       |
|   6 |     PARTITION RANGE ALL|           | 82112 |  4089K|       |   687   (5)| 00:00:01 |     1 |    28 |
|*  7 |      HASH JOIN         |           | 82112 |  4089K|       |   687   (5)| 00:00:01 |       |       |
|   8 |       TABLE ACCESS FULL| COSTS     | 82112 |  1924K|       |   137   (1)| 00:00:01 |     1 |    28 |
|   9 |       TABLE ACCESS FULL| SALES     |   918K|    23M|       |   530   (3)| 00:00:01 |     1 |    28 |
-------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("R"."COUNTRY_ID"="C"."COUNTRY_ID")
   3 - filter("R"."COUNTRY_REGION"=:B1)
   4 - access("C"."CUST_ID"="S"."CUST_ID")
   7 - access("C"."TIME_ID"="S"."TIME_ID" AND "C"."PROD_ID"="S"."PROD_ID" AND
              "C"."CHANNEL_ID"="S"."CHANNEL_ID" AND "C"."PROMO_ID"="S"."PROMO_ID")
```

# D E M O N S T R A T I O N

**end**

# Topics

- A Test Case (TC) for SQL Tuning?
- How SQLTXPLAIN (SQLT) helps with a TC?
- What is included in a SQLT TC?
- Demo of SQLT TC implementation
- **Troubleshooting a SQLT TC**
- Beyond a SQLT TC

# Troubleshooting a SQLT TC

- Known issues
  - Missing SQLT repository
  - Invalid view
  - Invalid package
  - Invalid SQL statement
  - Distributed SQL
  - SQL statement references identical table names from multiple owners
- SQLT TC can be restarted from intermediate steps

# Missing SQLT repository

- When using SQLT XTRACT/XECUTE/XPLAIN password for SQLTXPLAIN was not passed
  - Export SQLT repository as per dynamic readme

### Export SQLT repository

Steps:

1. Unzip `sqlt_s93896_driver.zip` in order to get `sqlt_s93896_export_parfile.txt`.
2. Copy `sqlt_s93896_export_parfile.txt` to SOURCE server (TEXT).
3. Execute export on server:

```
exp sqltxplain parfile=sqlt_s93896_export_parfile.txt
```

# Invalid view

- View references tables including schema name
  - Modify metadata script removing schema name
  - Re-execute metadata script

# Invalid package

- Isolate function and compile it individually
- If package is small and number of errors is small then fix package in metadata script and re-execute

# Invalid SQL statement

- If SQL statement is an INSERT and inserted table is missing from metadata then convert statement into a SELECT by removing INSERT clause
  - Be aware that SELECT part may be parsed differently by the CBO than when part of a DML
    - Some Query Transformations (QT) are not applied to DML
- Remove schema names on FROM clause

# Distributed SQL

- Implement TC in local and remote
  - You may need to process export from remote Source
    - Refer to readme for instructions
- Create database links manually

# SQL references identical tables from multiple owners

- Metadata script maps multiple instances of same table into only one TC user
- Consider customizing metadata to preserve original schema names
  - Then use `sqlt/utl/sqltimp.sql` to restore CBO schema statistics to these multiple schema owners

# Topics

- A Test Case (TC) for SQL Tuning?
- How SQLTXPLAIN (SQLT) helps with a TC?
- What is included in a SQLT TC?
- Demo of SQLT TC implementation
- Troubleshooting a SQLT TC
- **Beyond a SQLT TC**

# Beyond a SQLT TC

- SQLT TC enabled a SQL Tuning environment on a test system
- What is next after Execution Plan from Source system has been reproduced in Target?
  - If dealing with a regression test OFE to pre-upgrade
    - If OFE produces a prior “good” plan consider then using SQLT XPLORE
  - If suspecting schema object statistics consider using SQLT XHUME on your disposable test system
  - If suspecting histograms consider using SQLT XGRAM

# Removing SQLTXPLAIN dependencies from Test Case

- To isolate the TC from any SQLTXPLAIN dependencies refer to “Create TC with no SQLT dependencies” on readme

# Create TC with no SQLT dependencies

After creating a local test case using SQLT files, you can create a stand-alone TC with no dependencies on SQLT.

Steps:

1. Export TC schema object statistics to staging table within TC schema:

```
EXEC DBMS_STATS.EXPORT_SCHEMA_STATS('TC93896','CBO_STAT_TAB_4TC',NULL,'TC93896');
```

2. Export TC schema object statistics from staging table:

```
HOS exp TC93896/TC93896 FILE=cbo_stat_tab_4tc.dmp TABLES=cbo_stat_tab_4tc STATISTICS=NONE
```

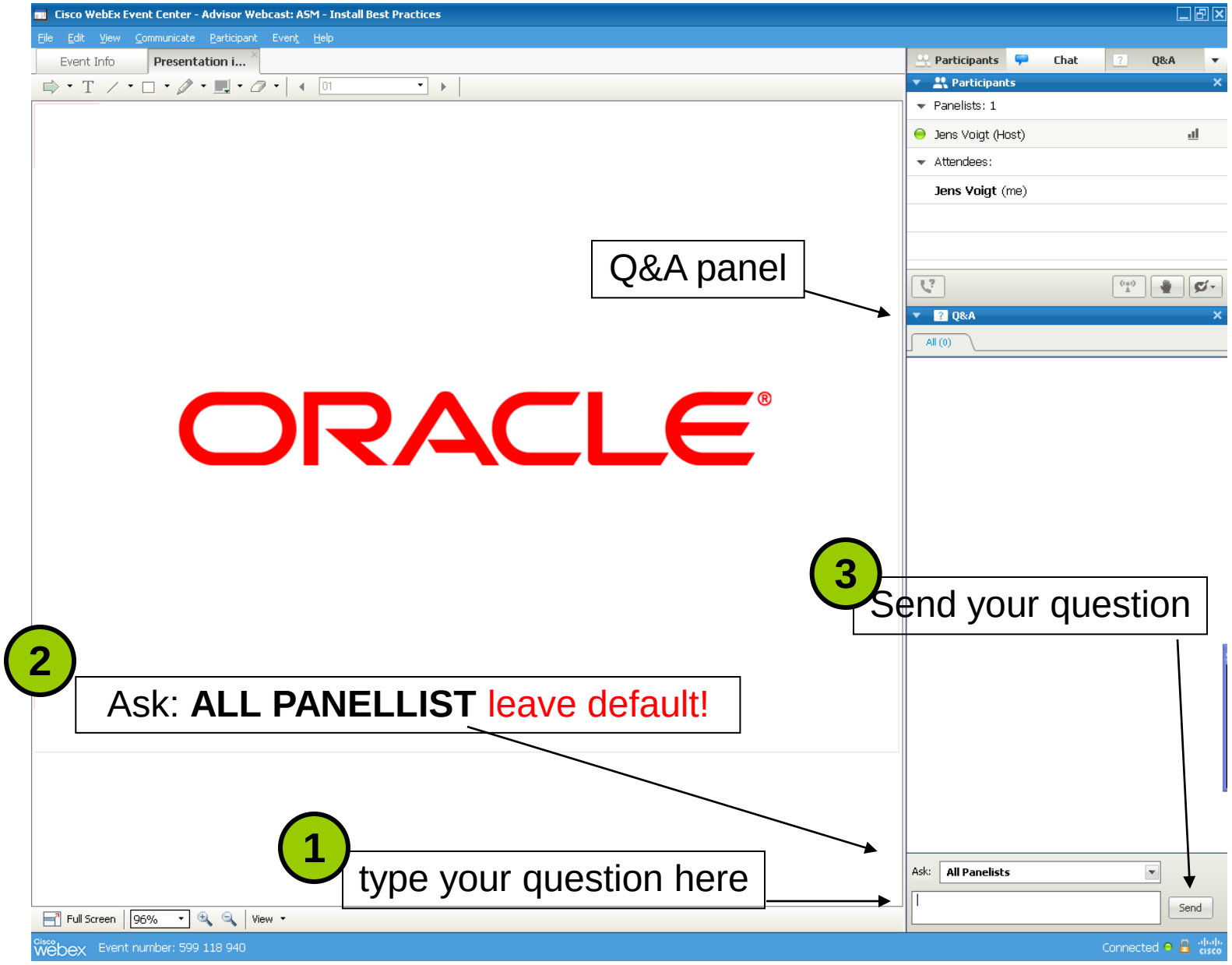
3. Review and adjust `sqlt_s93896_metadata.sql` script. You may want to set the TC_USER_SUFFIX as a constant and remove the ACCEPT command.
4. Review and adjust `setup.sql` (may remove comment on set_cbo_env) and `sqlt_s93896_set_cbo_env.sql` (remove pause).
5. Create a new `readme.txt` file with simple instructions, like example below:

```
connect as sys and execute setup.sql
```

6. Create and zip a new directory with the following files:

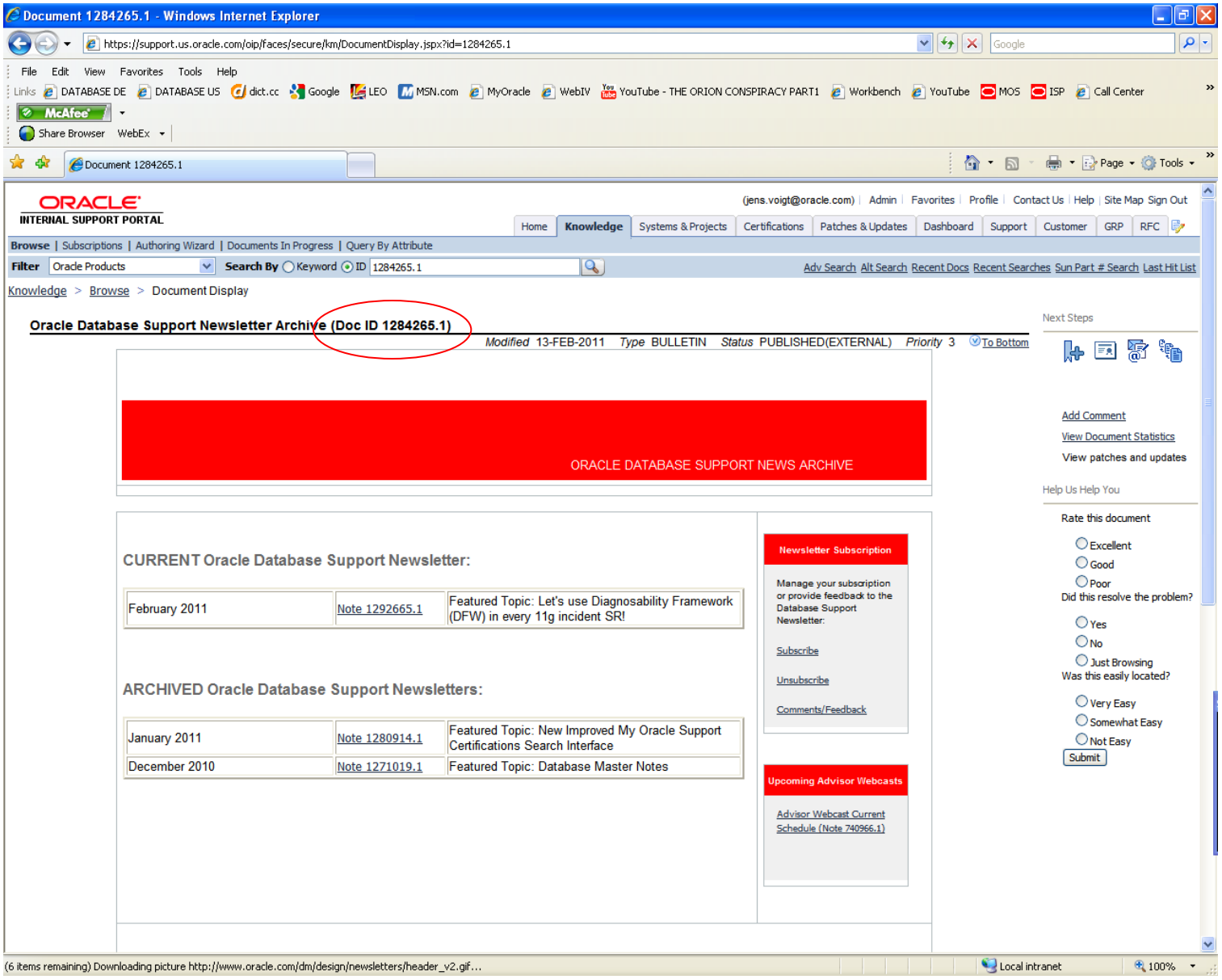
```
CBO schema object statistics dump: cbo_stat_tab_4tc.dmp
Plan script:                       plan.sql
Query script:                      q.sql
Instructions:                      readme.txt
Setup script:                      setup.sql
Metadata script:                   sqlt_s93896_metadata.sql
OPatch (if needed):                sqlt_s93896_opatch.zip
Set CBO env script (if needed):    sqlt_s93896_set_cbo_env.sql
System statistics setup:           sqlt_s93896_system_stats.sql
Test case script:                  tc.sql
```

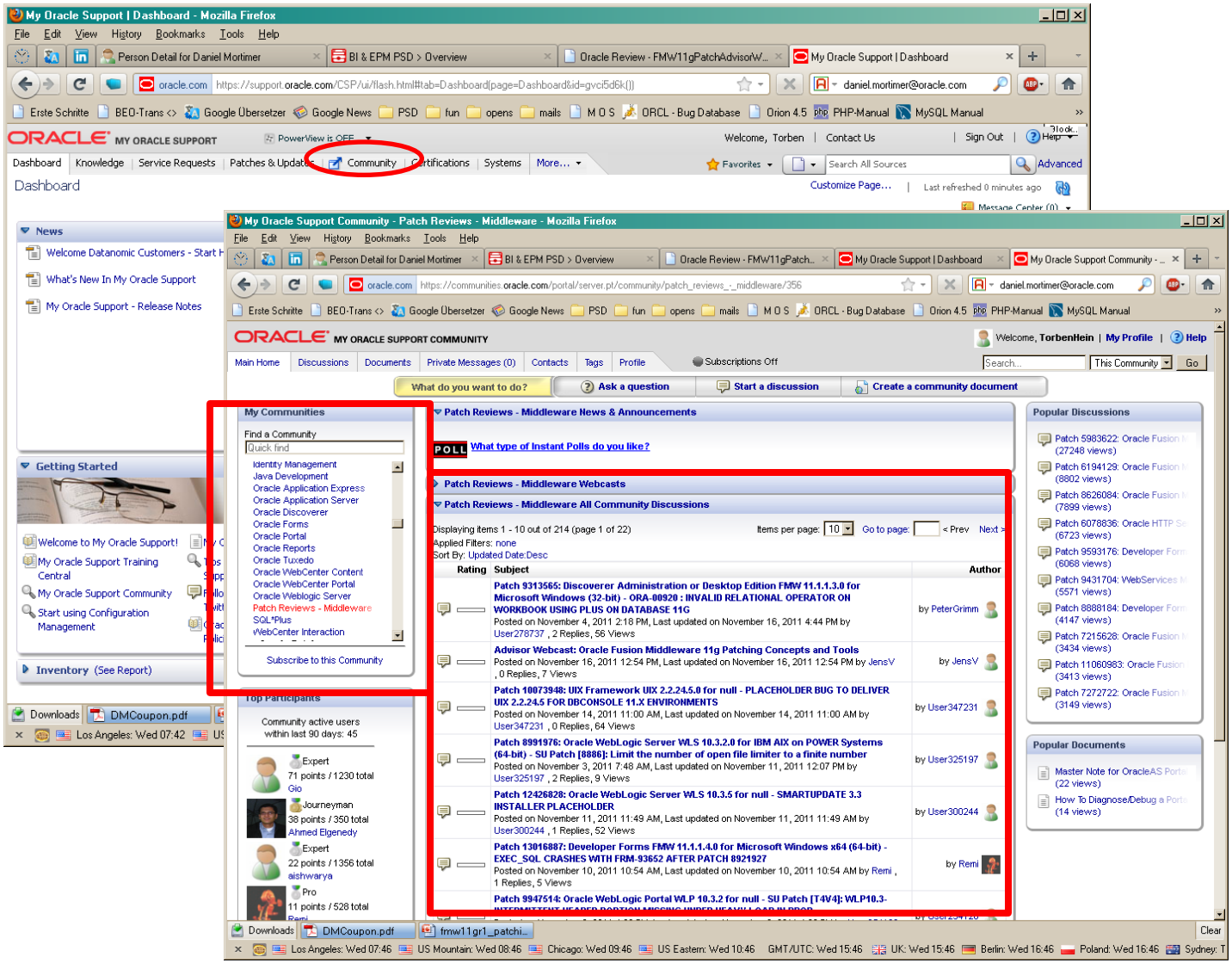
7. Test your new stand-alone TC following your own `readme.txt` in another system.



# Further Info & Help

- Advisor Webcast Archived Recordings (Doc ID 740964.1)
- Advisor Webcast Current Schedule (Doc ID 740966.1)
- DB Newsletter (Doc ID 1284265.1)
- MOS Community Database Tuning





# THANK YOU

# BACKUP

# ATTENTION – AUDIO STREAMING IS NOT AVAILABLE !

**All, connect to the teleconference please:**

1. **Conference ID: 47629030**
2. **International dial in: +44 (0) 1452 562 665**
   **US Free Call: 1866 230 1938**
   **US Local Call: 1845 608 8023**
3. **List with national toll free numbers is available in note 1148600.1**

**You can view this info anytime during the conference using**

`Communicate > Teleconference > Join Teleconference`

**from your WebEx menu**