Using the TUNE_MVIEW Advisor

Purpose

This module shows you how to use the TUNE_MVIEW Advisor to make recommendations on what parameters to specify when creating a materialized view to optimize performance.

Topics

This module will discuss the following topics:

Place the cursor on this icon to display all screenshots. You can also place the cursor on each icon to see only the screenshot associated with it.

Overview

Back to List

Oracle9i Database introduced the DBMS_MVIEW.EXPLAIN_MVIEW API which is used to explain whether a materialized view is fast refreshable or eligible for general query rewrite. Oracle Database 10g introduces the new PL/SQL API, DBMS_ADVISOR.TUNE_MVIEW, to facilitate the materialized view creation process by delivering a set of SQL statements that can be used to:

- Automatically fix any materialized view log problems such as non-existence of the materialized view log, or missing columns in the materialized view log required for materialized view fast refresh.
- Redefine fast refreshable materialized view with optimized defining queries to enable fast refresh and general query rewrite.
- Redefine, if possible, a non-fast refreshable materialized view by decomposing its original defining query into a number of fast refreshable sub-materialized views referenced by the original one.

The new TUNE_MVIEW API advises what changes you need to make to a materialized view to make it fast refreshable and eligible for advanced query rewrite techniques.

Prerequisites

Back to List

Before starting this module, you should have:

☐ **1.** Completed the Configuring Linux for the Installation of Oracle Database 10g lesson

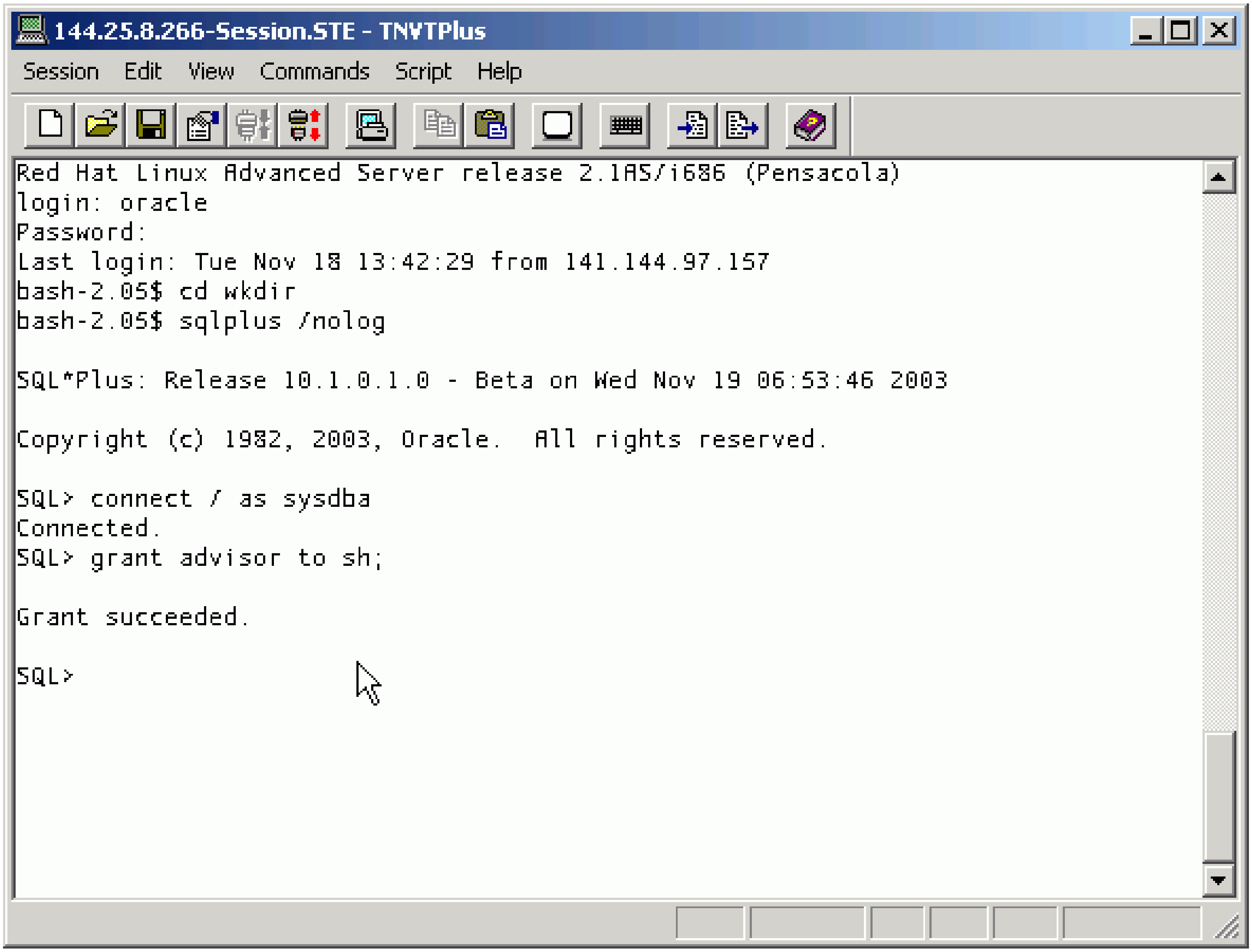
☐ **2.** Completed the Installing the Oracle Database 10g on Linux lesson

☐ **3.** Completed the Postinstallation Tasks lesson.

☐ **4.** Download and unzip tunemview.zip into your working directory (i.e. /home/oracle/wkdir)

☐ **5.** Open a terminal window and execute the following:

```
cd wkdir
sqlplus /nolog
connect / as sysdba
grant advisor to sh;
```


144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

```
Red Hat Linux Advanced Server release 2.1AS/i686 (Pensacola)
login: oracle
Password:
Last login: Tue Nov 18 13:42:29 from 141.144.97.157
bash-2.05$ cd wkdir
bash-2.05$ sqlplus /nolog

SQL*Plus: Release 10.1.0.1.0 - Beta on Wed Nov 19 06:53:46 2003

Copyright (c) 1982, 2003, Oracle.  All rights reserved.

SQL> connect / as sysdba
Connected.
SQL> grant advisor to sh;

Grant succeeded.

SQL>
```

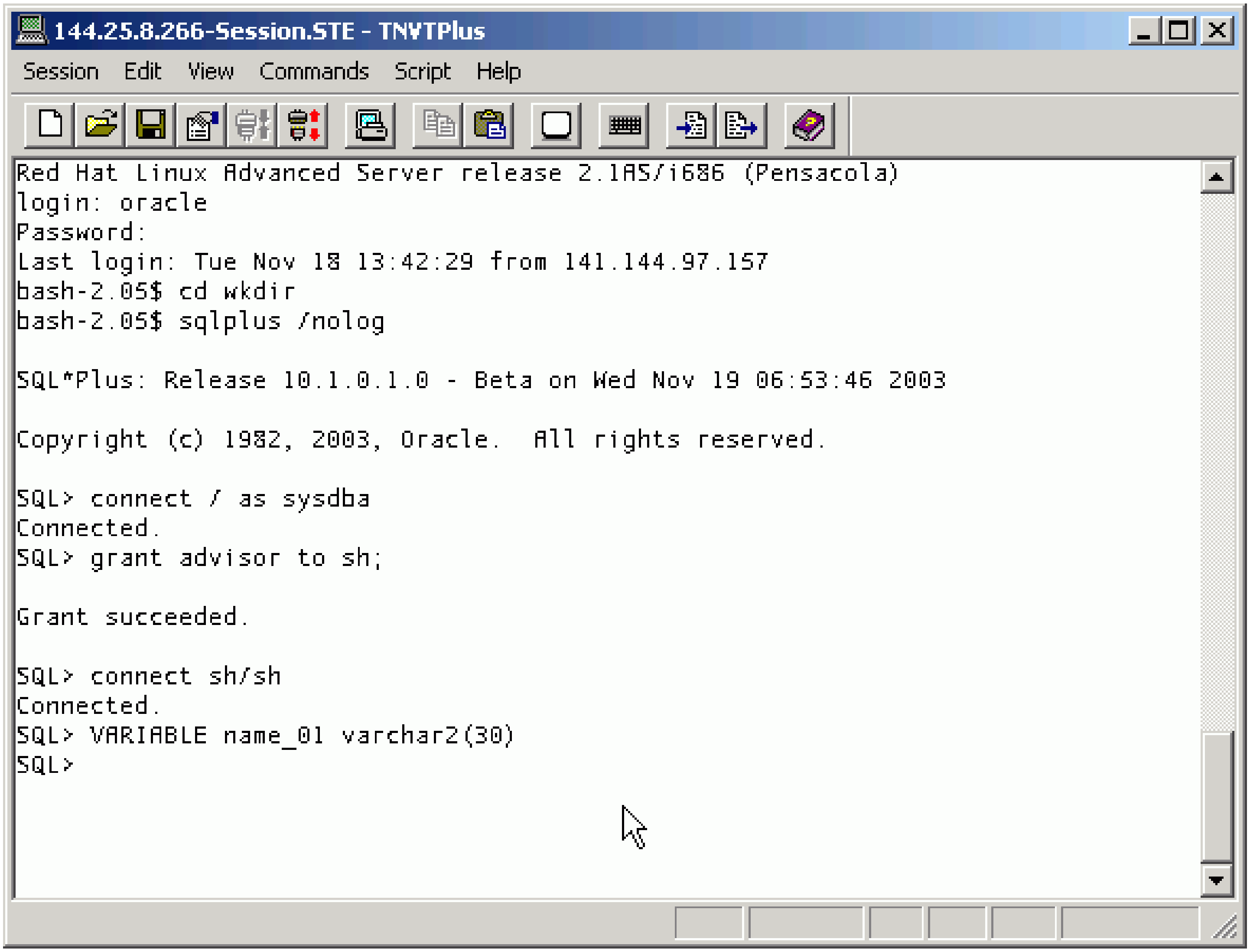
Generating Materialized View Suggestions using TUNE_MVIEW

Back to Topic List

You will generate materialized view suggestions using the new Oracle Database 10g dbms_advisor.TUNE_MVIEW procedure. Perform the following steps:

☐ **1.** You need to define a bind variable to capture a task name from the dbms_advisor.TUNE_MVIEW procedure. From your terminal window, execute the following commands:

```
connect sh/sh
VARIABLE name_01 varchar2(30)
```


144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

```
Red Hat Linux Advanced Server release 2.1AS/i686 (Pensacola)
login: oracle
Password:
Last login: Tue Nov 18 13:42:29 from 141.144.97.157
bash-2.05$ cd wkdir
bash-2.05$ sqlplus /nolog

SQL*Plus: Release 10.1.0.1.0 - Beta on Wed Nov 19 06:53:46 2003

Copyright (c) 1982, 2003, Oracle.  All rights reserved.

SQL> connect / as sysdba
Connected.
SQL> grant advisor to sh;

Grant succeeded.

SQL> connect sh/sh
Connected.
SQL> VARIABLE name_01 varchar2(30)
SQL>
```

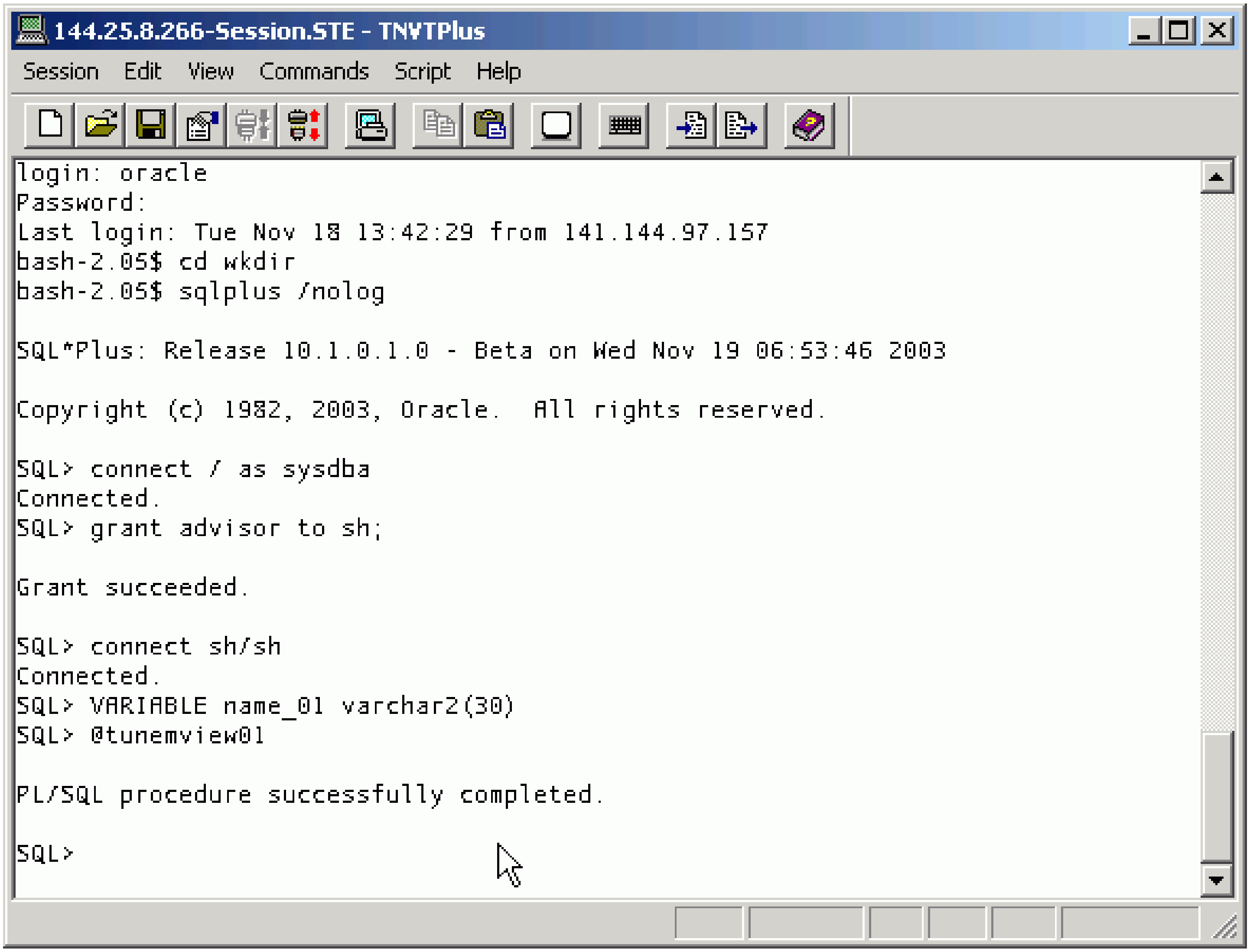
☐ **2.** Now you can execute the procedure by executing the following script from your terminal window:

```
@tunemview01
```

The query in the **`tunemview01.sql`** script is as follows:

```
EXECUTE dbms_advisor.TUNE_MVIEW     -
( :name_01                          -
, 'CREATE MATERIALIZED VIEW prod_mv -
   REFRESH FAST WITH ROWID          -
   ENABLE QUERY REWRITE             -
   AS                               -
   SELECT DISTINCT                  -
          prod_name, prod_category  -
   FROM   products'                 -
) ;
```

Note that the query above contains a DISTINCT clause.


144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

```
login: oracle
Password:
Last login: Tue Nov 18 13:42:29 from 141.144.97.157
bash-2.05$ cd wkdir
bash-2.05$ sqlplus /nolog

SQL*Plus: Release 10.1.0.1.0 - Beta on Wed Nov 19 06:53:46 2003

Copyright (c) 1982, 2003, Oracle.  All rights reserved.

SQL> connect / as sysdba
Connected.
SQL> grant advisor to sh;

Grant succeeded.

SQL> connect sh/sh
Connected.
SQL> VARIABLE name_01 varchar2(30)
SQL> @tunemview01

PL/SQL procedure successfully completed.

SQL>
```

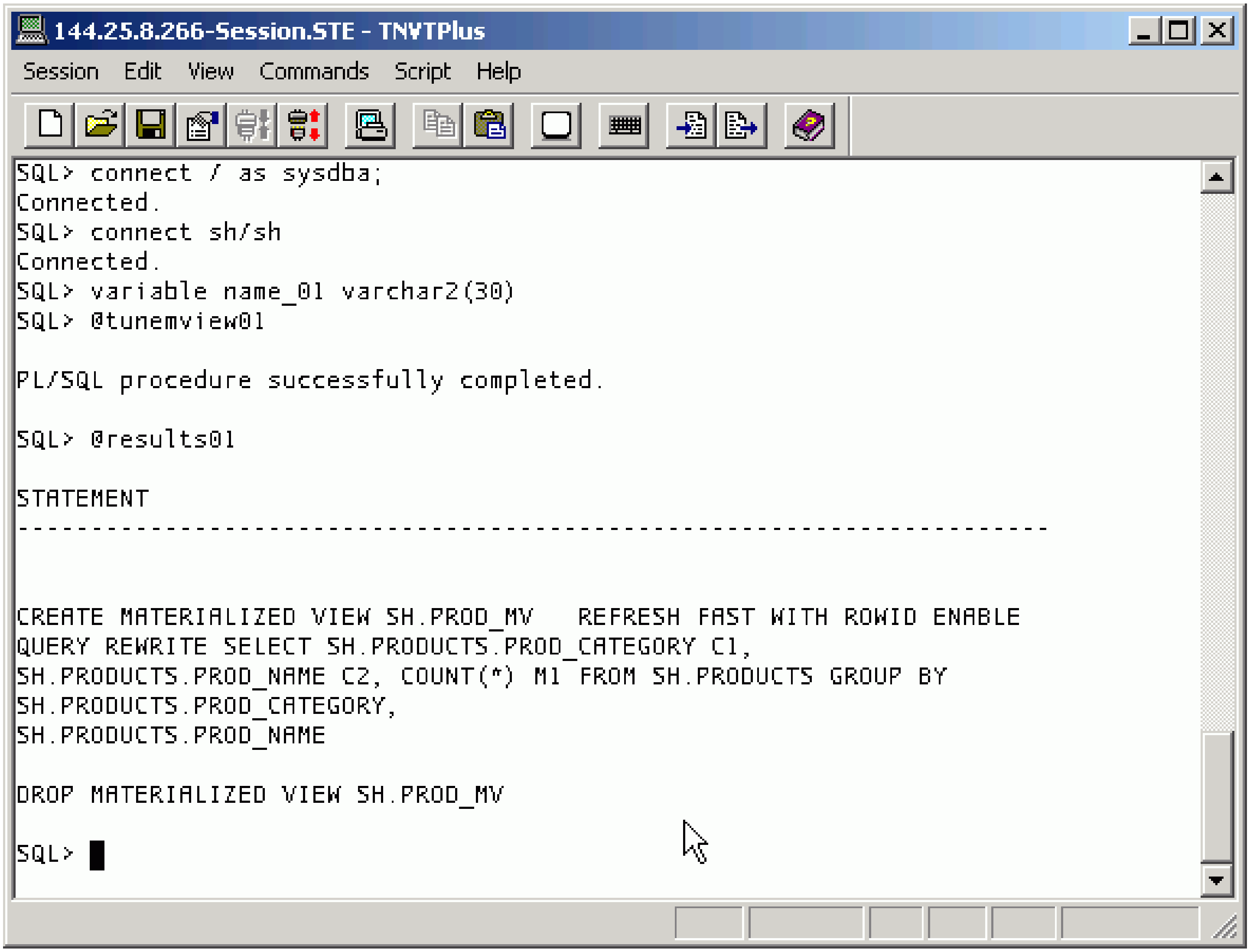
☐ **3.** The TUNE_MVIEW procedure returns a task name into the name_01 variable and the results are stored in the data dictionary. Now you can query DBA_TUNE_MVIEW to view the suggested rewrite of the MV definition. From your terminal window, execute the following script:

```
@results01
```

The query in the **`results 01.sql`** script is as follows:

```
column statement format a70 word

set long 999
SELECT statement

FROM   DBA_TUNE_MVIEW
```

```
WHERE  task_name = :name_01

ORDER  BY script_type, action_id;
```


144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

```
SQL> connect / as sysdba;
Connected.
SQL> connect sh/sh
Connected.
SQL> variable name_01 varchar2(30)
SQL> @tunemview01

PL/SQL procedure successfully completed.

SQL> @results01

STATEMENT
----------------------------------------------------------------------


CREATE MATERIALIZED VIEW SH.PROD_MV   REFRESH FAST WITH ROWID ENABLE
QUERY REWRITE SELECT SH.PRODUCTS.PROD_CATEGORY C1,
SH.PRODUCTS.PROD_NAME C2, COUNT(*) M1 FROM SH.PRODUCTS GROUP BY
SH.PRODUCTS.PROD_CATEGORY,
SH.PRODUCTS.PROD_NAME

DROP MATERIALIZED VIEW SH.PROD_MV

SQL>
```

Note that the DISTINCT clause is replaced by the GROUP BY construct, and the COUNT(*) is added to the SELECT clause. This makes the Materialized view eligible for fast refresh and usable for general rewrite.

Using TUNE_MVIEW to Make a Materialized View Fast Refreshable

Back to Topic List

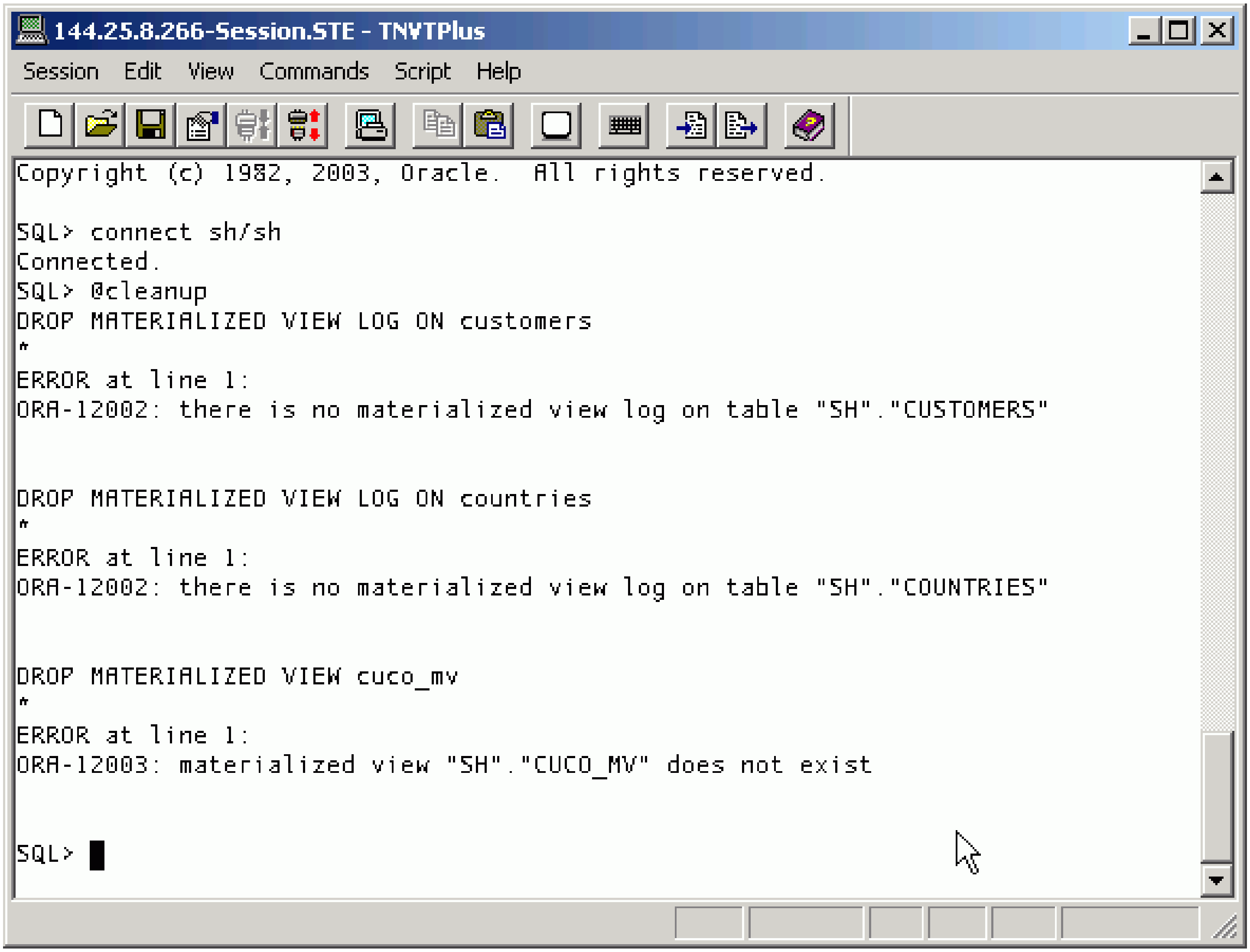
You will try to create a materialized view with the REFRESH FAST option that fails. You will use TUNE_MVIEW to figure out what needs to be changed. Perform the following steps:

☐ **1.** Before you perform the steps in this example, you need to clean up your database. From your terminal window, execute the following script:

```
@cleanup
```

The script `cleanup.sql` contains the following:

```
DROP MATERIALIZED VIEW LOG ON customers;

DROP MATERIALIZED VIEW LOG ON countries;

DROP MATERIALIZED VIEW cuco_mv;
```



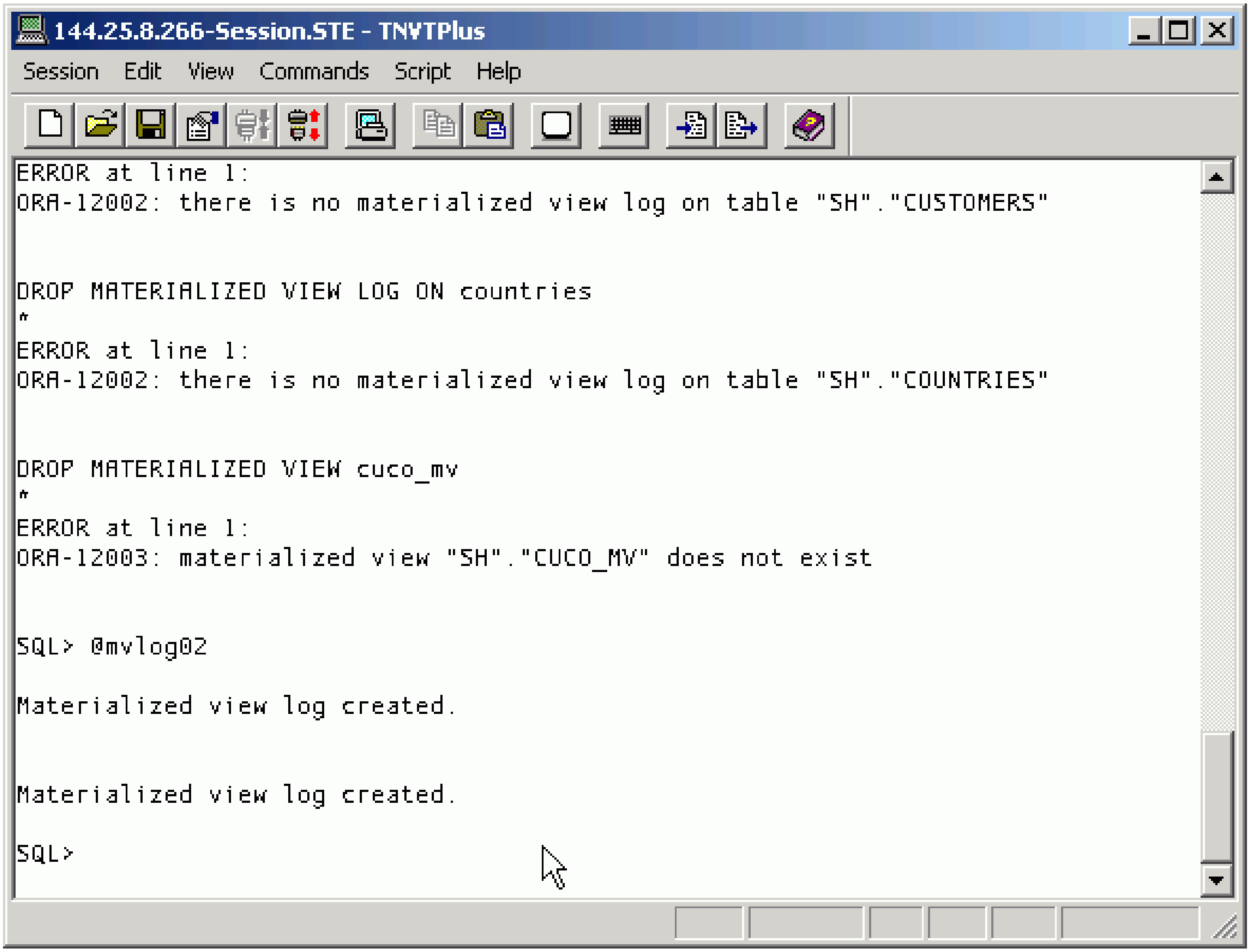
```
Copyright (c) 1982, 2003, Oracle.  All rights reserved.

SQL> connect sh/sh
Connected.
SQL> @cleanup
DROP MATERIALIZED VIEW LOG ON customers
*
ERROR at line 1:
ORA-12002: there is no materialized view log on table "SH"."CUSTOMERS"


DROP MATERIALIZED VIEW LOG ON countries
*
ERROR at line 1:
ORA-12002: there is no materialized view log on table "SH"."COUNTRIES"


DROP MATERIALIZED VIEW cuco_mv
*
ERROR at line 1:
ORA-12003: materialized view "SH"."CUCO_MV" does not exist


SQL>
```

☐ **2.** You need to create two MV logs on CUSTOMERS and COUNTRIES. From your terminal window, execute the following script:

```
@mvlog02
```

The script **`mvlog02 .sql`** contains the following:

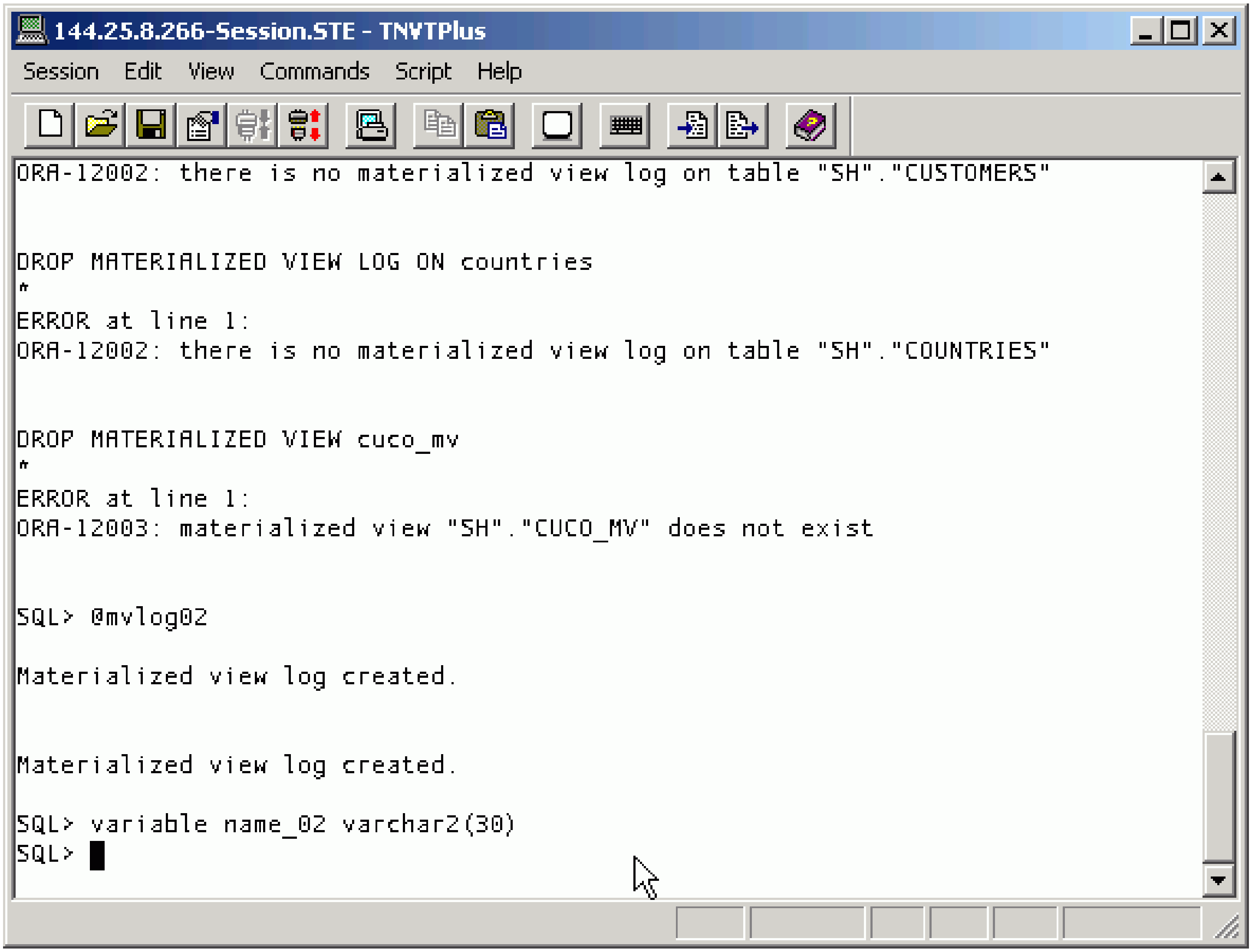
```
CREATE MATERIALIZED VIEW LOG ON customers

   WITH SEQUENCE, ROWID INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON countries
   WITH SEQUENCE, ROWID INCLUDING NEW VALUES;
```



```
ERROR at line 1:
ORA-12002: there is no materialized view log on table "SH"."CUSTOMERS"


DROP MATERIALIZED VIEW LOG ON countries
*
ERROR at line 1:
ORA-12002: there is no materialized view log on table "SH"."COUNTRIES"


DROP MATERIALIZED VIEW cuco_mv
*
ERROR at line 1:
ORA-12003: materialized view "SH"."CUCO_MV" does not exist


SQL> @mvlog02

Materialized view log created.


Materialized view log created.

SQL>
```

☐ **3.** You need to define another bind variable. From your SQL*Plus session, execute the following command:

```
VARIABLE name_02 varchar2(30)
```


144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

```
ORA-12002: there is no materialized view log on table "SH"."CUSTOMERS"


DROP MATERIALIZED VIEW LOG ON countries
*
ERROR at line 1:
ORA-12002: there is no materialized view log on table "SH"."COUNTRIES"


DROP MATERIALIZED VIEW cuco_mv
*
ERROR at line 1:
ORA-12003: materialized view "SH"."CUCO_MV" does not exist


SQL> @mvlog02

Materialized view log created.


Materialized view log created.

SQL> variable name_02 varchar2(30)
SQL>
```

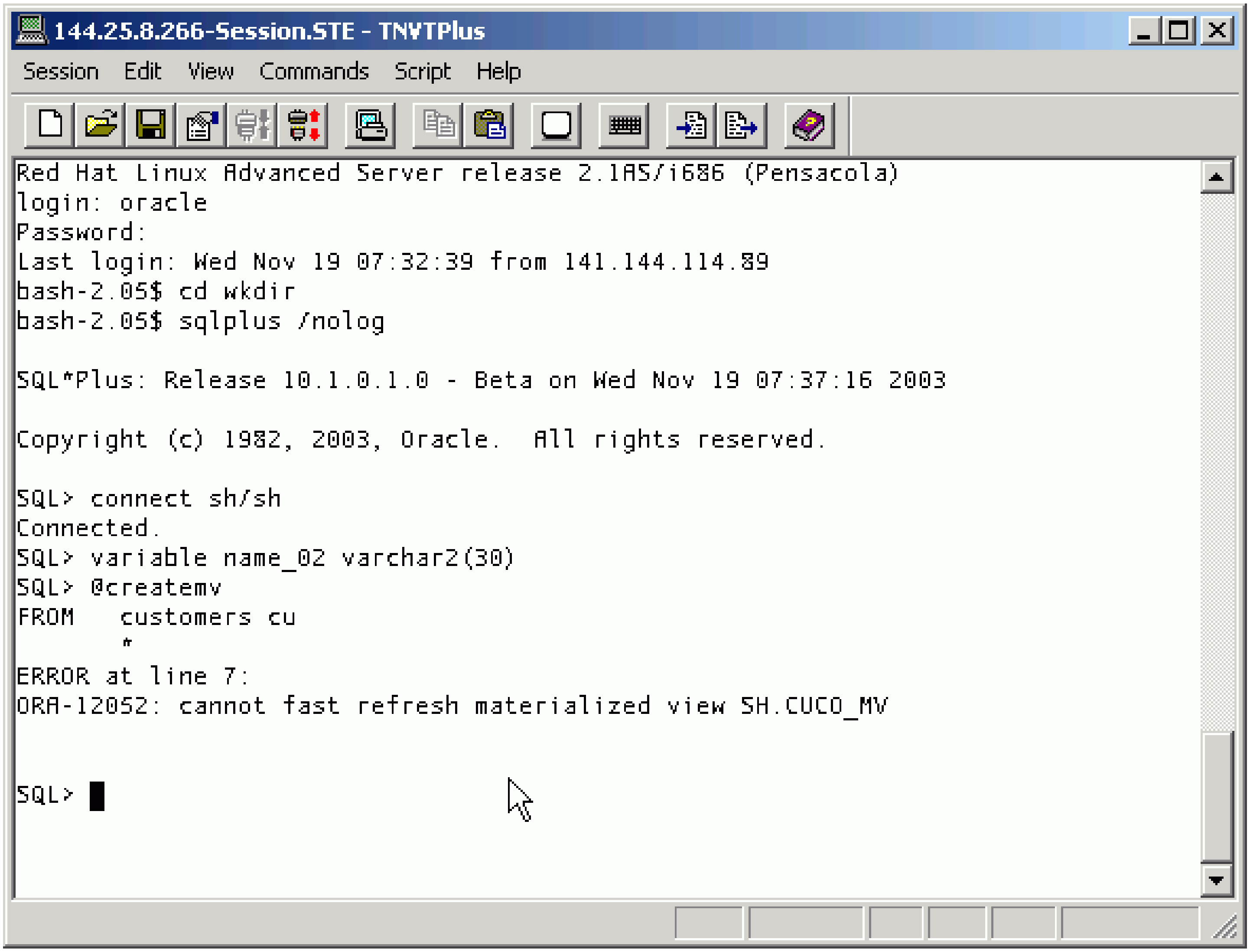
☐ **4.** Now you can try to create the MV by executing the following script:

```
@createmv
```

The command in the **`createmv .sql`** script is as follows:

```
CREATE MATERIALIZED VIEW cuco_mv

       REFRESH FAST

       ENABLE QUERY REWRITE

AS

SELECT cu.cust_last_name

,      co.country_name

FROM   customers cu

,      countries co

WHERE  cu.country_id = co.country_id;
```

Note that you explicitly ask for REFRESH FAST.


144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

```
Red Hat Linux Advanced Server release 2.1AS/i686 (Pensacola)
login: oracle
Password:
Last login: Wed Nov 19 07:32:39 from 141.144.114.89
bash-2.05$ cd wkdir
bash-2.05$ sqlplus /nolog

SQL*Plus: Release 10.1.0.1.0 - Beta on Wed Nov 19 07:37:16 2003

Copyright (c) 1982, 2003, Oracle.  All rights reserved.

SQL> connect sh/sh
Connected.
SQL> variable name_02 varchar2(30)
SQL> @createmv
FROM   customers cu
       *
ERROR at line 7:
ORA-12052: cannot fast refresh materialized view SH.CUCO_MV


SQL> 
```

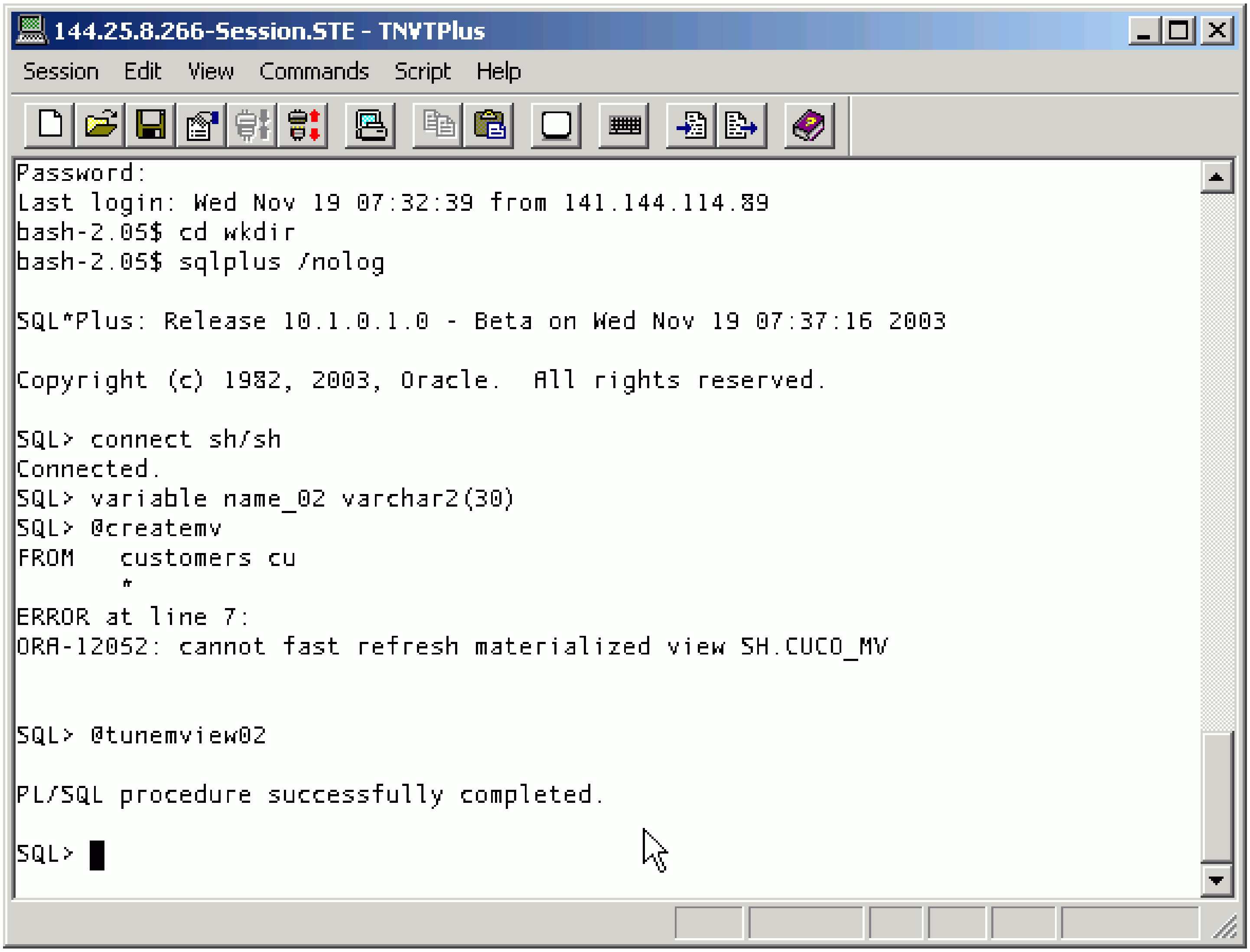
Notice that you received an error even though you created the necessary MV logs on the two tables. Any idea what might be the problem?

☐ **5.** Now you can use the TUNE_MVIEW procedure to see what recommendations can be made. Execute the following script:

```
@tunemview02
```

The command in the **`tunemview02.sql`** script is as follows:

```
EXECUTE dbms_advisor.TUNE_MVIEW              -

( :name_02                                   -
, 'CREATE MATERIALIZED VIEW cuco_mv      -

          REFRESH FAST                  -

          ENABLE QUERY REWRITE          -

   AS                                   -

   SELECT cu.cust_last_name             -

   ,      co.country_name               -

   FROM   customers cu                  -

   ,      countries co                  -

   WHERE  cu.country_id = co.country_id' -

) ;
```



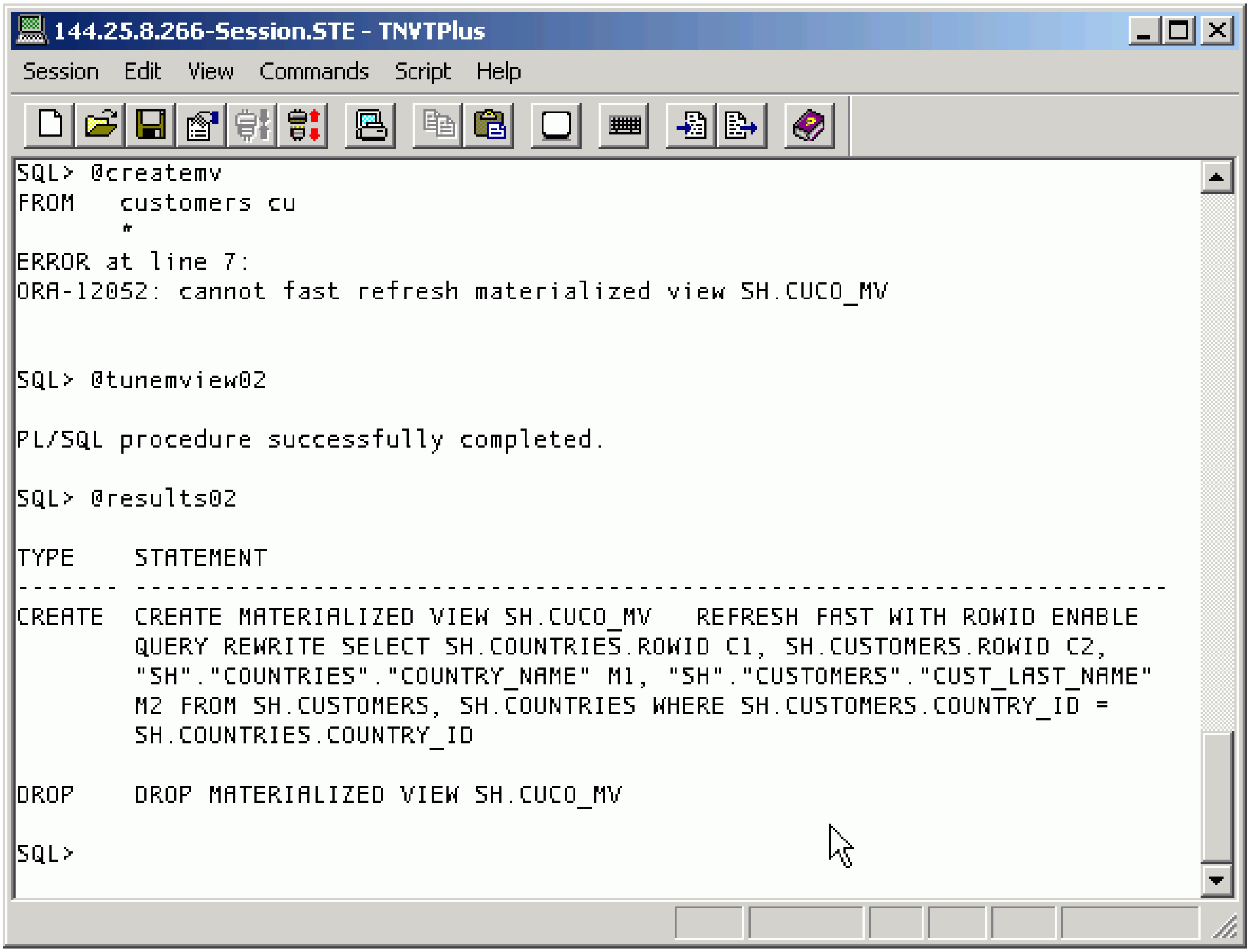
```
Password:
Last login: Wed Nov 19 07:32:39 from 141.144.114.89
bash-2.05$ cd wkdir
bash-2.05$ sqlplus /nolog

SQL*Plus: Release 10.1.0.1.0 - Beta on Wed Nov 19 07:37:16 2003

Copyright (c) 1982, 2003, Oracle.  All rights reserved.

SQL> connect sh/sh
Connected.
SQL> variable name_02 varchar2(30)
SQL> @createmv
FROM   customers cu
       *
ERROR at line 7:
ORA-12052: cannot fast refresh materialized view SH.CUCO_MV


SQL> @tunemview02

PL/SQL procedure successfully completed.

SQL> 
```

☐ **6.** Now you can query DBA_TUNE_MVIEW to view the suggested rewrite of the MV definition. From your terminal window, execute the following script:

```
@results02
```

The query in the **`results 02.sql`** script is as follows:

```
column statement format a70 word

set long 999
SELECT script_type as type, statement

FROM   DBA_TUNE_MVIEW

WHERE  task_name = :name_02

ORDER  BY script_type, action_id;
```


144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

```
SQL> @createmv
FROM   customers cu
       *
ERROR at line 7:
ORA-12052: cannot fast refresh materialized view SH.CUCO_MV


SQL> @tunemview02

PL/SQL procedure successfully completed.

SQL> @results02

TYPE    STATEMENT
------- ----------------------------------------------------------------------
CREATE  CREATE MATERIALIZED VIEW SH.CUCO_MV   REFRESH FAST WITH ROWID ENABLE
        QUERY REWRITE SELECT SH.COUNTRIES.ROWID C1, SH.CUSTOMERS.ROWID C2,
        "SH"."COUNTRIES"."COUNTRY_NAME" M1, "SH"."CUSTOMERS"."CUST_LAST_NAME"
        M2 FROM SH.CUSTOMERS, SH.COUNTRIES WHERE SH.CUSTOMERS.COUNTRY_ID =
        SH.COUNTRIES.COUNTRY_ID

DROP    DROP MATERIALIZED VIEW SH.CUCO_MV

SQL>
```

Two explicit ROWID references are added to the SELECT clause. This makes the materialized view fast refreshable.

Generating a Script of TUNE_MVIEW Recommendations

Back to Topic List

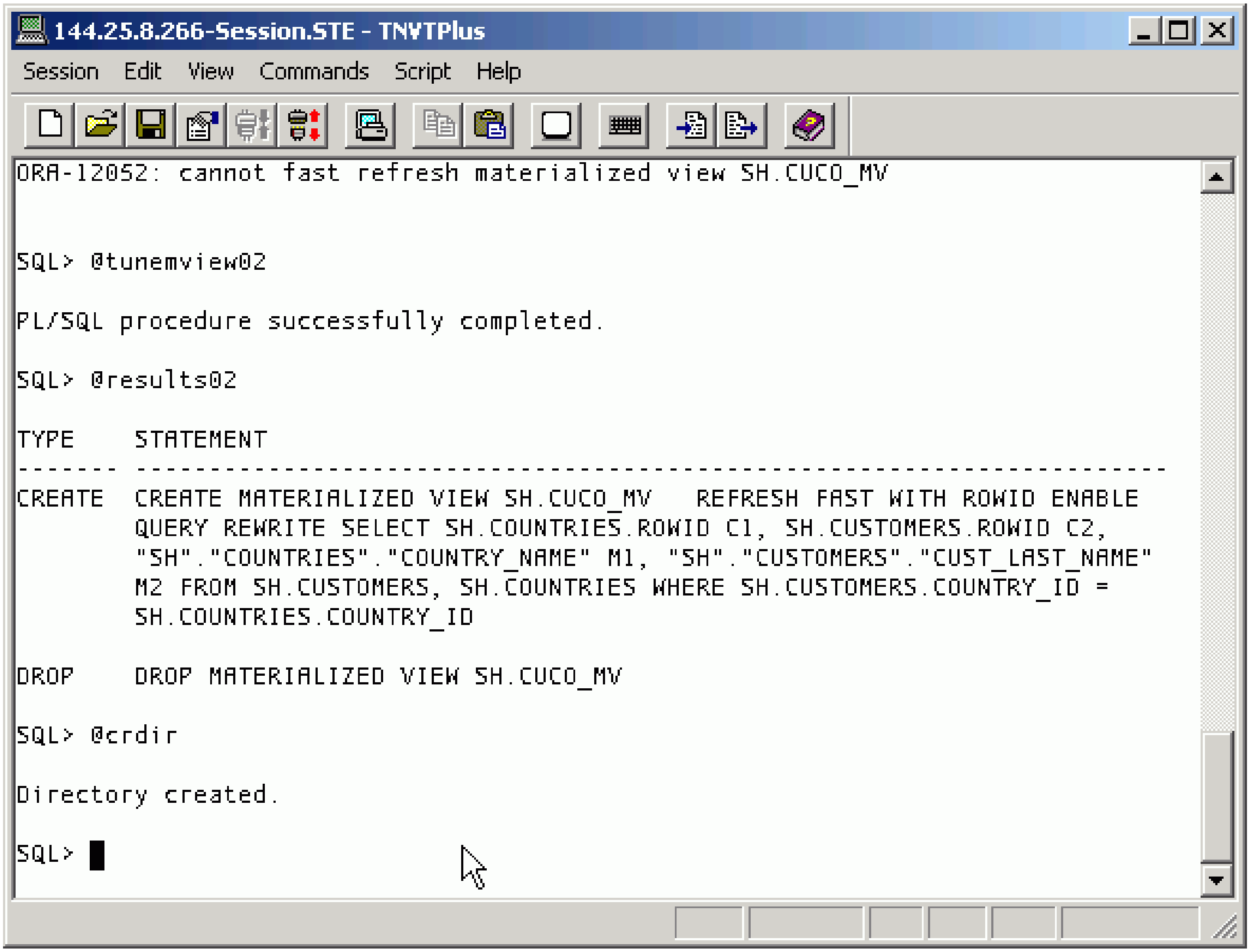
At this point, you want to generate a script of the recommendations from DBA_TUNE_MVIEW so that you can execute them in a SQL*Plus session. Perform the following steps:

☐ **1.** Before you perform the steps in this example, you need to clean up your database. From your terminal window, execute the following script:

```
@crdir
```

The script **`crdir.sql`** contains the following:

```
create or replace directory ext_tab_dir as '/home/oracle/wkdir';
```


144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

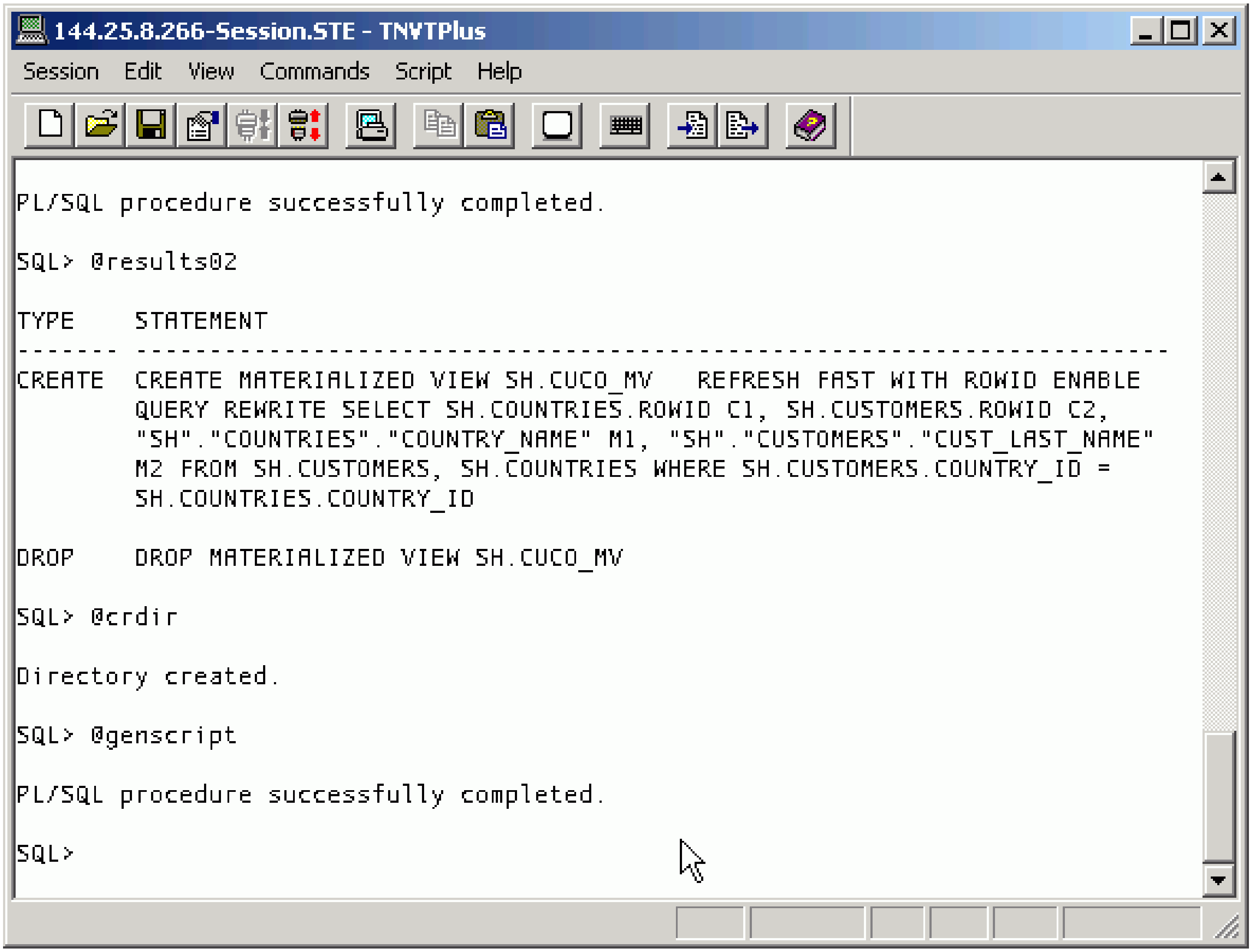
```
ORA-12052: cannot fast refresh materialized view SH.CUCO_MV


SQL> @tunemview02

PL/SQL procedure successfully completed.

SQL> @results02

TYPE    STATEMENT
------- --------------------------------------------------------------------
CREATE  CREATE MATERIALIZED VIEW SH.CUCO_MV   REFRESH FAST WITH ROWID ENABLE
        QUERY REWRITE SELECT SH.COUNTRIES.ROWID C1, SH.CUSTOMERS.ROWID C2,
        "SH"."COUNTRIES"."COUNTRY_NAME" M1, "SH"."CUSTOMERS"."CUST_LAST_NAME"
        M2 FROM SH.CUSTOMERS, SH.COUNTRIES WHERE SH.CUSTOMERS.COUNTRY_ID =
        SH.COUNTRIES.COUNTRY_ID

DROP    DROP MATERIALIZED VIEW SH.CUCO_MV

SQL> @crdir

Directory created.

SQL>
```

☐ **2.** Now you can generate the script. From your terminal window, execute the following script:

```
@genscript
```

The script **`genscript.sql`** contains the following:

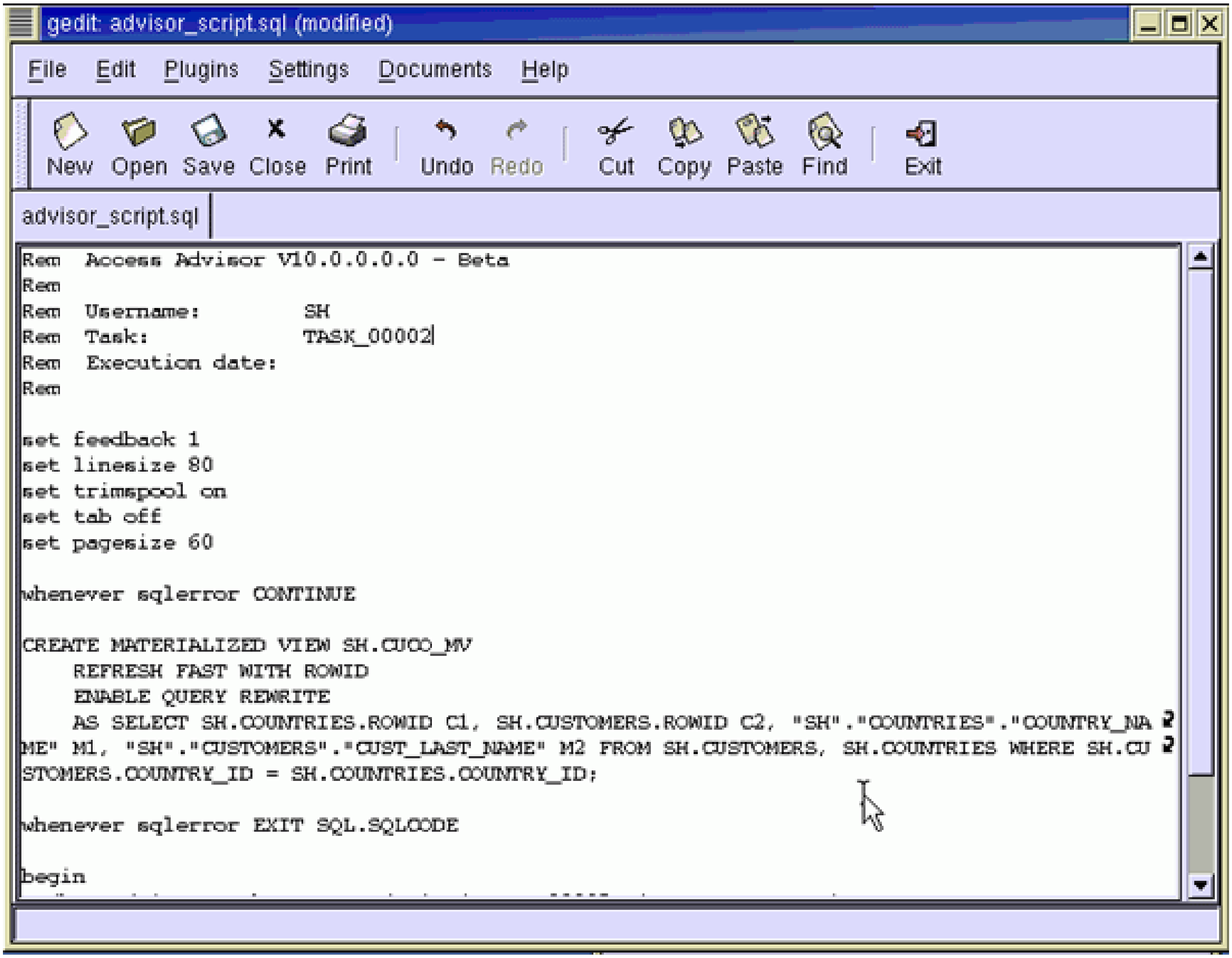
```
EXECUTE dbms_advisor.CREATE_FILE         -
( dbms_advisor.get_task_script(:name_02) -
, location => 'EXT_TAB_DIR'              -
, filename => 'advisor_script.sql'       -
) ;
```



144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

```
PL/SQL procedure successfully completed.

SQL> @results02

TYPE    STATEMENT
------- ------------------------------------------------------------------------------
CREATE  CREATE MATERIALIZED VIEW SH.CUCO_MV   REFRESH FAST WITH ROWID ENABLE
        QUERY REWRITE SELECT SH.COUNTRIES.ROWID C1, SH.CUSTOMERS.ROWID C2,
        "SH"."COUNTRIES"."COUNTRY_NAME" M1, "SH"."CUSTOMERS"."CUST_LAST_NAME"
        M2 FROM SH.CUSTOMERS, SH.COUNTRIES WHERE SH.CUSTOMERS.COUNTRY_ID =
        SH.COUNTRIES.COUNTRY_ID

DROP    DROP MATERIALIZED VIEW SH.CUCO_MV

SQL> @crdir

Directory created.

SQL> @genscript

PL/SQL procedure successfully completed.

SQL>
```

☐ **3.** You can now view the script file **`advisor_script.sql`** . Open the file in gedit.



```
Rem  Access Advisor V10.0.0.0.0 - Beta
Rem
Rem  Username:         SH
Rem  Task:             TASK_00002
Rem  Execution date:
Rem

set feedback 1
set linesize 80
set trimspool on
set tab off
set pagesize 60

whenever sqlerror CONTINUE

CREATE MATERIALIZED VIEW SH.CUCO_MV
    REFRESH FAST WITH ROWID
    ENABLE QUERY REWRITE
    AS SELECT SH.COUNTRIES.ROWID C1, SH.CUSTOMERS.ROWID C2, "SH"."COUNTRIES"."COUNTRY_NA
ME" M1, "SH"."CUSTOMERS"."CUST_LAST_NAME" M2 FROM SH.CUSTOMERS, SH.COUNTRIES WHERE SH.CU
STOMERS.COUNTRY_ID = SH.COUNTRIES.COUNTRY_ID;

whenever sqlerror EXIT SQL.SQLCODE

begin
```

Place the cursor on this icon to hide all screenshots.