Using External Tables for Data Unload and Projected Columns

Purpose

This module shows you how to use two new Oracle10g features: Data Unloading and Projected Columns.

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.ated with it.

Overview

Back to List

External Table Population

Prior to Oracle Database 10g, external tables were read-only. In Oracle Database 10g, external tables can also be written to. Although neither data manipulation language (DML) operations nor index creation are allowed on an external table, it is possible to use the CREATE TABLE AS SELECT command to populate an external table composed of proprietary format (Direct Path API) flat files that are operating system independent.

In the context of external tables, loading data refers to the act of data being read from an external table and loaded into a table in the database. Unloading data refers to the act of reading data from a table in the database and inserting it into an external table. Both these operations can be used with external tables using the new Data Pump access driver.

Projected Columns

When dealing with external table files which contain rows of data that may be rejected, the projected column feature allows you to get a consistent result set independent of the columns referenced by the SQL statement accessing your external table. Prior to Oracle Database 10g, only the columns referenced by the SQL statement were projected out by the access driver. Due to the access driver parsing the input data stream, and the external table service doing data conversions from the external representation to the internal representation, some rows may get rejected due to conversion errors or data format errors.

Prerequisites

Back to List

Before starting this module, you should have:

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

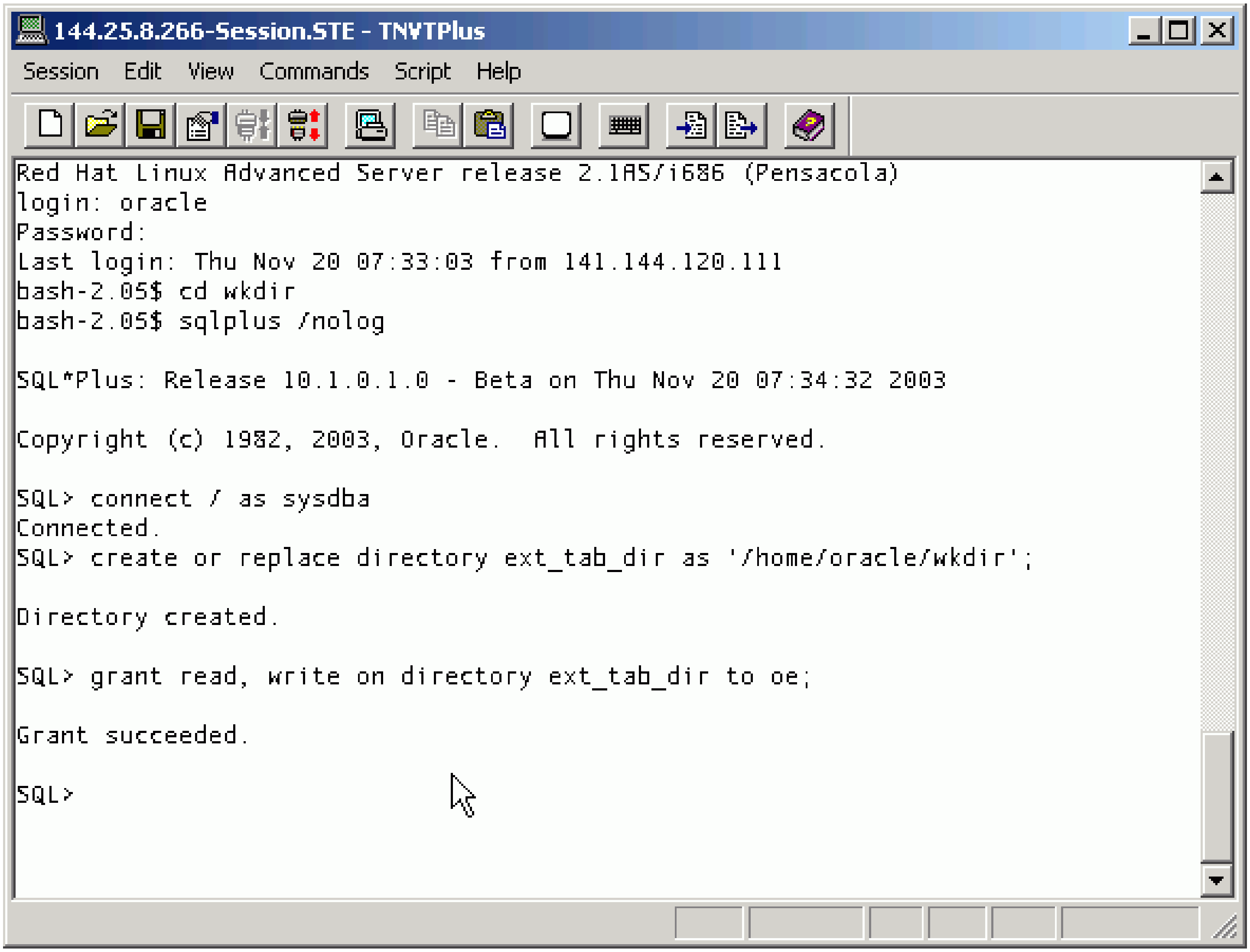
☐ **2.** Completed the Installing the Oracle Database 10g on Linux lesson

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

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

☐ **5.** For this lesso, you need to create a directory. Open a terminal window and execute the following:

```
cd wkdir
sqlplus /nolog
connect / as sysdba
create or replace directory ext_tab_dir as '/home/oracle/wkdir';
grant read, write on directory ext_tab_dir to oe;
```



```
Red Hat Linux Advanced Server release 2.1AS/i686 (Pensacola)
login: oracle
Password:
Last login: Thu Nov 20 07:33:03 from 141.144.120.111
bash-2.05$ cd wkdir
bash-2.05$ sqlplus /nolog

SQL*Plus: Release 10.1.0.1.0 - Beta on Thu Nov 20 07:34:32 2003

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

SQL> connect / as sysdba
Connected.
SQL> create or replace directory ext_tab_dir as '/home/oracle/wkdir';

Directory created.

SQL> grant read, write on directory ext_tab_dir to oe;

Grant succeeded.

SQL>
```

Populating External Tables

Back to Topic List

You will create an external table, unload data into it and then query from the external table. Perform the following steps:

☐ **1.** You need to create an external table that will unload data into two files. From your terminal window, execute the following commands:

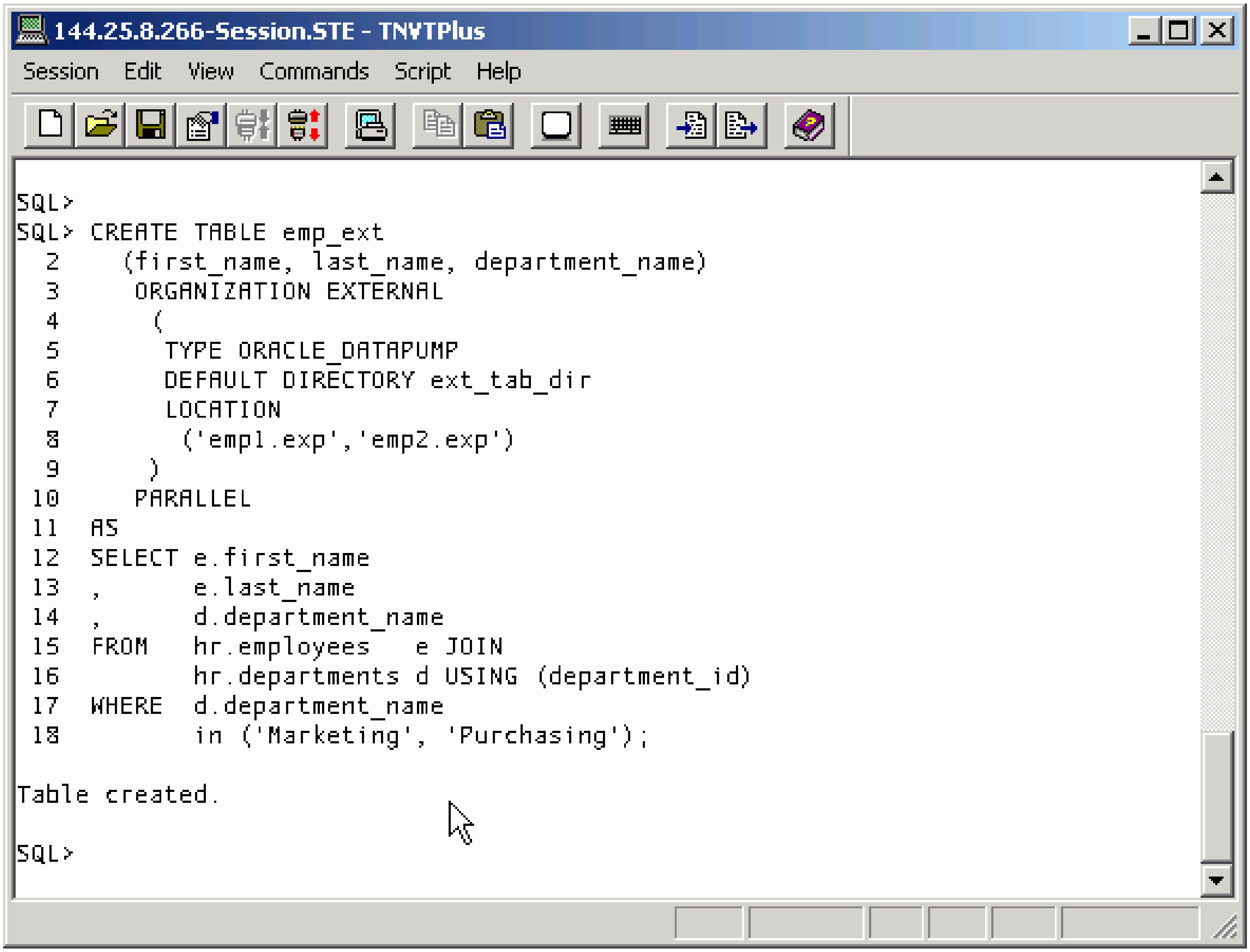
```
connect oe/oe
@crtabl01
```

The query in the **`crtabl01.sql`** script is as follows:

```
DROP
```

```
 TABLE emp_ext;

CREATE TABLE emp_ext

  (first_name, last_name, department_name)

   ORGANIZATION EXTERNAL

    (


TYPE ORACLE_DATAPUMP

     DEFAULT DIRECTORY ext_tab_dir

     LOCATION

      ('emp1.exp','emp2.exp')

    )

   PARALLEL

AS

SELECT e.first_name

,      e.last_name

,      d.department_name

FROM   hr.employees   e JOIN

       hr.departments d USING (department_id)

WHERE  d.department_name

       in ('Marketing', 'Purchasing');
```

Notice that you are using ORACLE_DATAPUMP as the type.


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

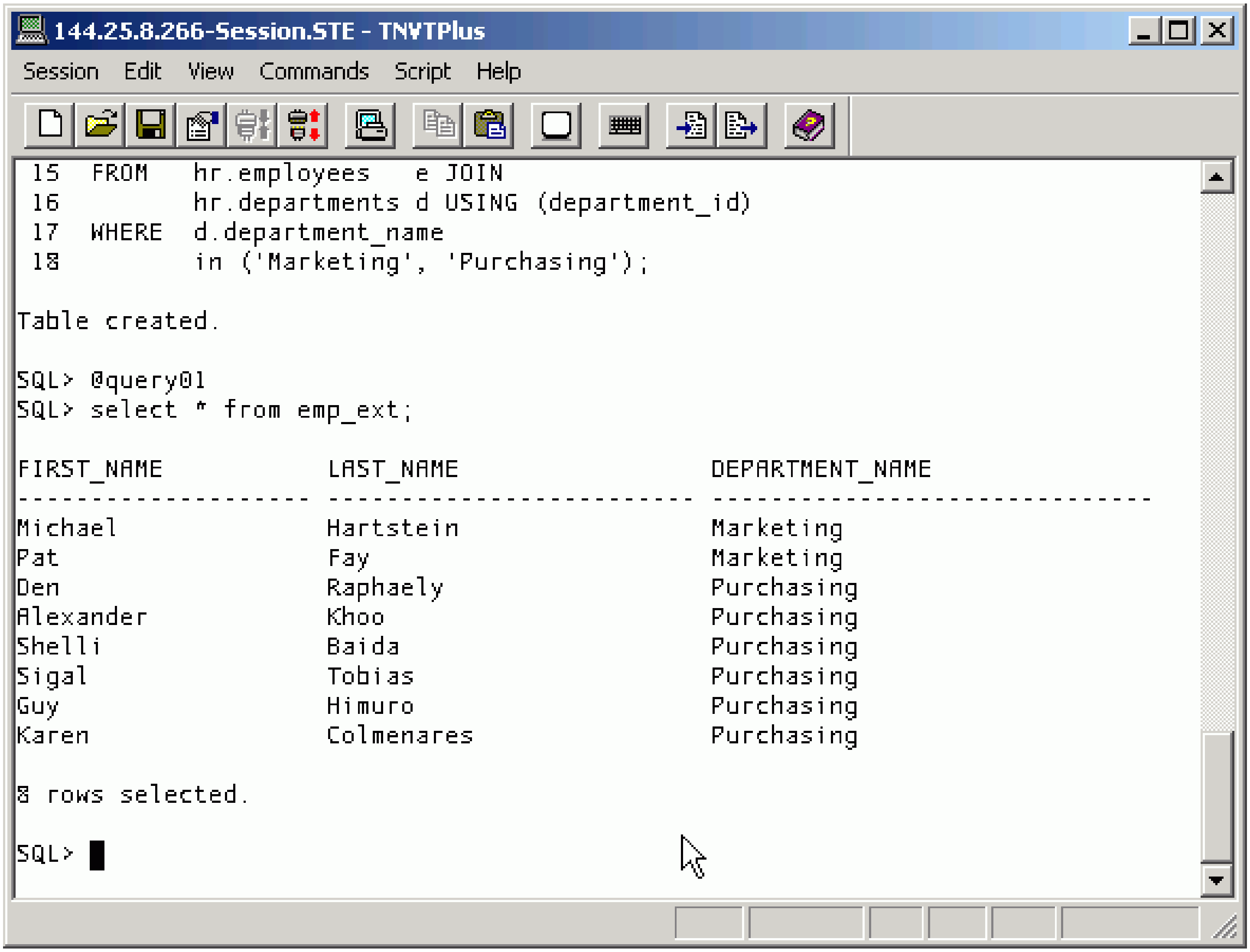
```
SQL>
SQL> CREATE TABLE emp_ext
  2    (first_name, last_name, department_name)
  3     ORGANIZATION EXTERNAL
  4      (
  5       TYPE ORACLE_DATAPUMP
  6       DEFAULT DIRECTORY ext_tab_dir
  7       LOCATION
  8        ('emp1.exp','emp2.exp')
  9      )
 10     PARALLEL
 11  AS
 12  SELECT e.first_name
 13  ,      e.last_name
 14  ,      d.department_name
 15  FROM   hr.employees   e JOIN
 16         hr.departments d USING (department_id)
 17  WHERE  d.department_name
 18         in ('Marketing', 'Purchasing');

Table created.

SQL>
```

☐ **2.** Now you can query the external table by executing the following script from your terminal window:

`@query01`

The query in the **`query01.sql`** script is as follows:

```
select * from emp_ext;
exit
```

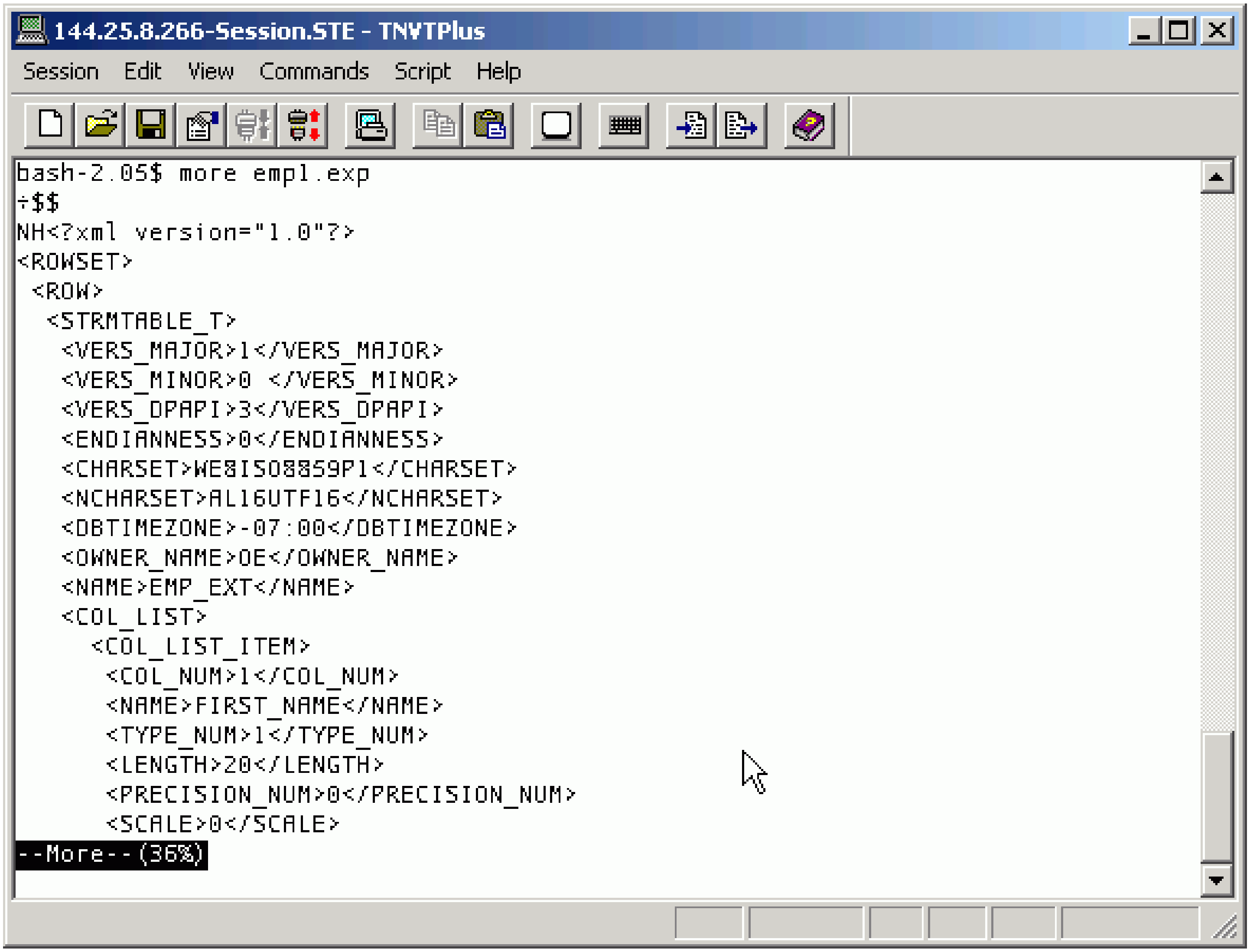


```
 15  FROM   hr.employees   e JOIN
 16         hr.departments d USING (department_id)
 17  WHERE  d.department_name
 18         in ('Marketing', 'Purchasing');

Table created.

SQL> @query01
SQL> select * from emp_ext;

FIRST_NAME           LAST_NAME                 DEPARTMENT_NAME
-------------------- ------------------------- ------------------------------
Michael              Hartstein                 Marketing
Pat                  Fay                       Marketing
Den                  Raphaely                  Purchasing
Alexander            Khoo                      Purchasing
Shelli               Baida                     Purchasing
Sigal                Tobias                    Purchasing
Guy                  Himuro                    Purchasing
Karen                Colmenares                Purchasing

8 rows selected.

SQL>
```

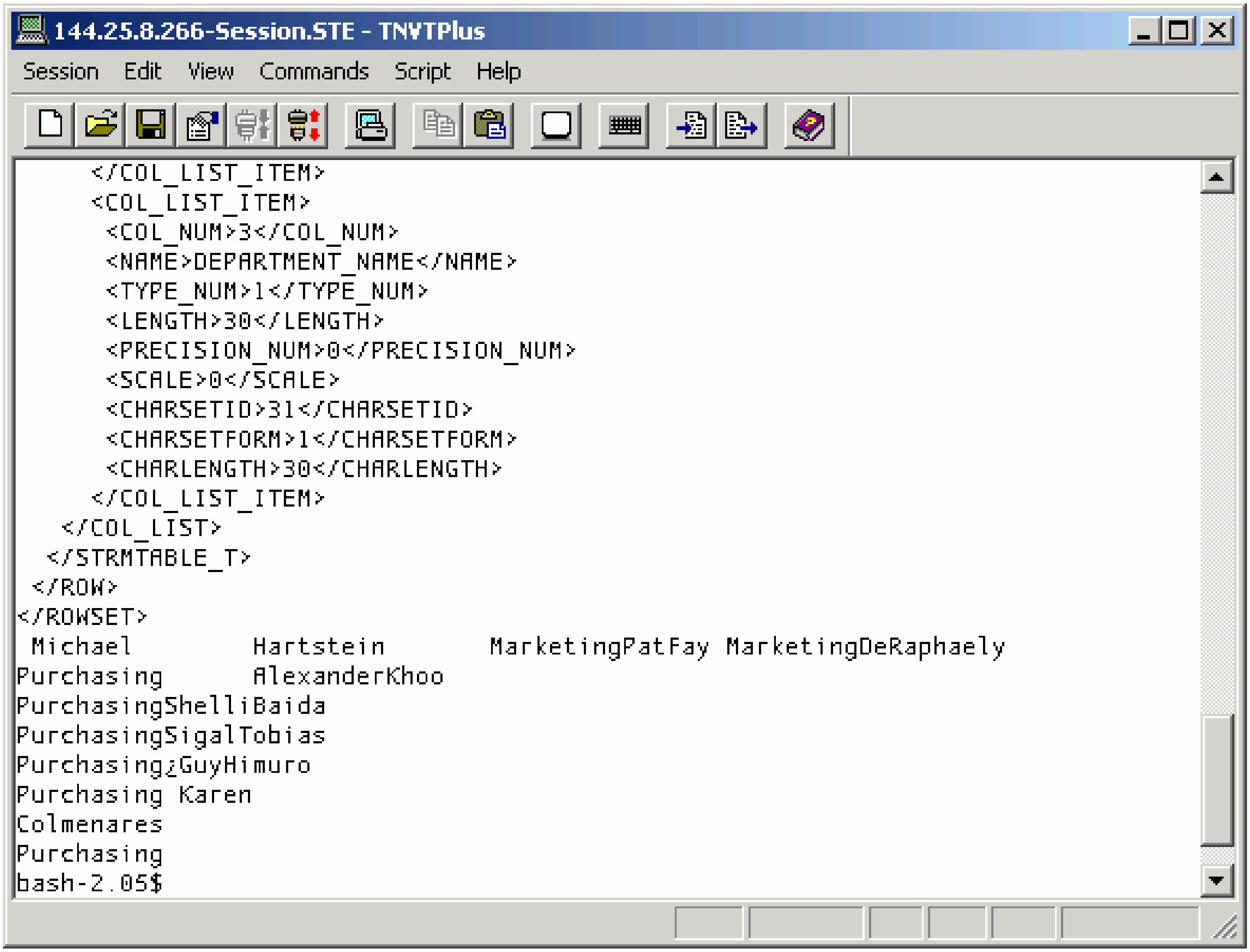
Note that the external table shows two employees in Marketing and six employees in Purchasing.

☐ **3.** You can also look at the two files that were generated. From your terminal window, execute the following command:

```
more emp1.exp
```

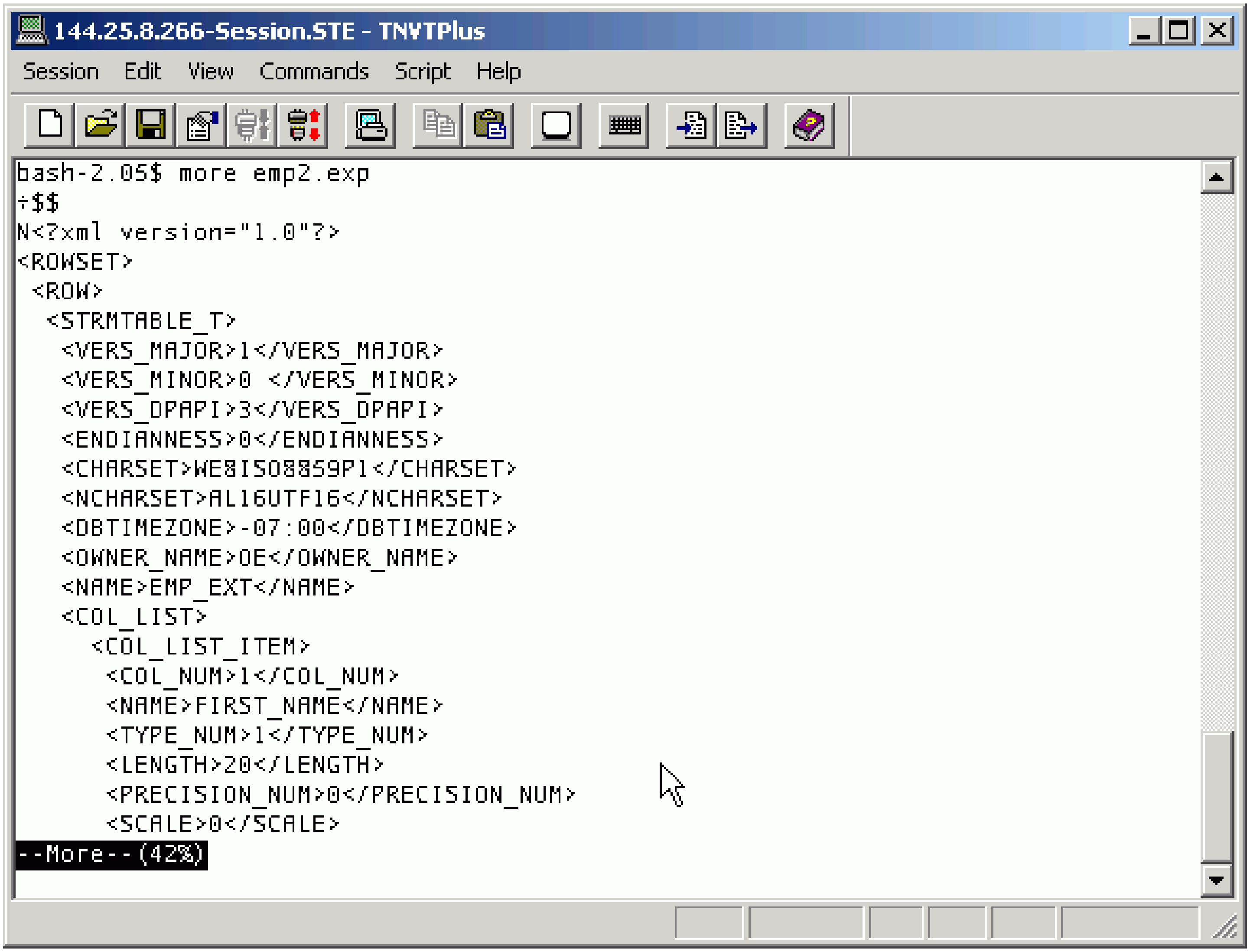


```
bash-2.05$ more empl.exp
÷$$
NH<?xml version="1.0"?>
<ROWSET>
 <ROW>
  <STRMTABLE_T>
   <VERS_MAJOR>1</VERS_MAJOR>
   <VERS_MINOR>0 </VERS_MINOR>
   <VERS_DPAPI>3</VERS_DPAPI>
   <ENDIANNESS>0</ENDIANNESS>
   <CHARSET>WE8ISO8859P1</CHARSET>
   <NCHARSET>AL16UTF16</NCHARSET>
   <DBTIMEZONE>-07:00</DBTIMEZONE>
   <OWNER_NAME>OE</OWNER_NAME>
   <NAME>EMP_EXT</NAME>
   <COL_LIST>
     <COL_LIST_ITEM>
      <COL_NUM>1</COL_NUM>
      <NAME>FIRST_NAME</NAME>
      <TYPE_NUM>1</TYPE_NUM>
      <LENGTH>20</LENGTH>
      <PRECISION_NUM>0</PRECISION_NUM>
      <SCALE>0</SCALE>
--More--(36%)
```

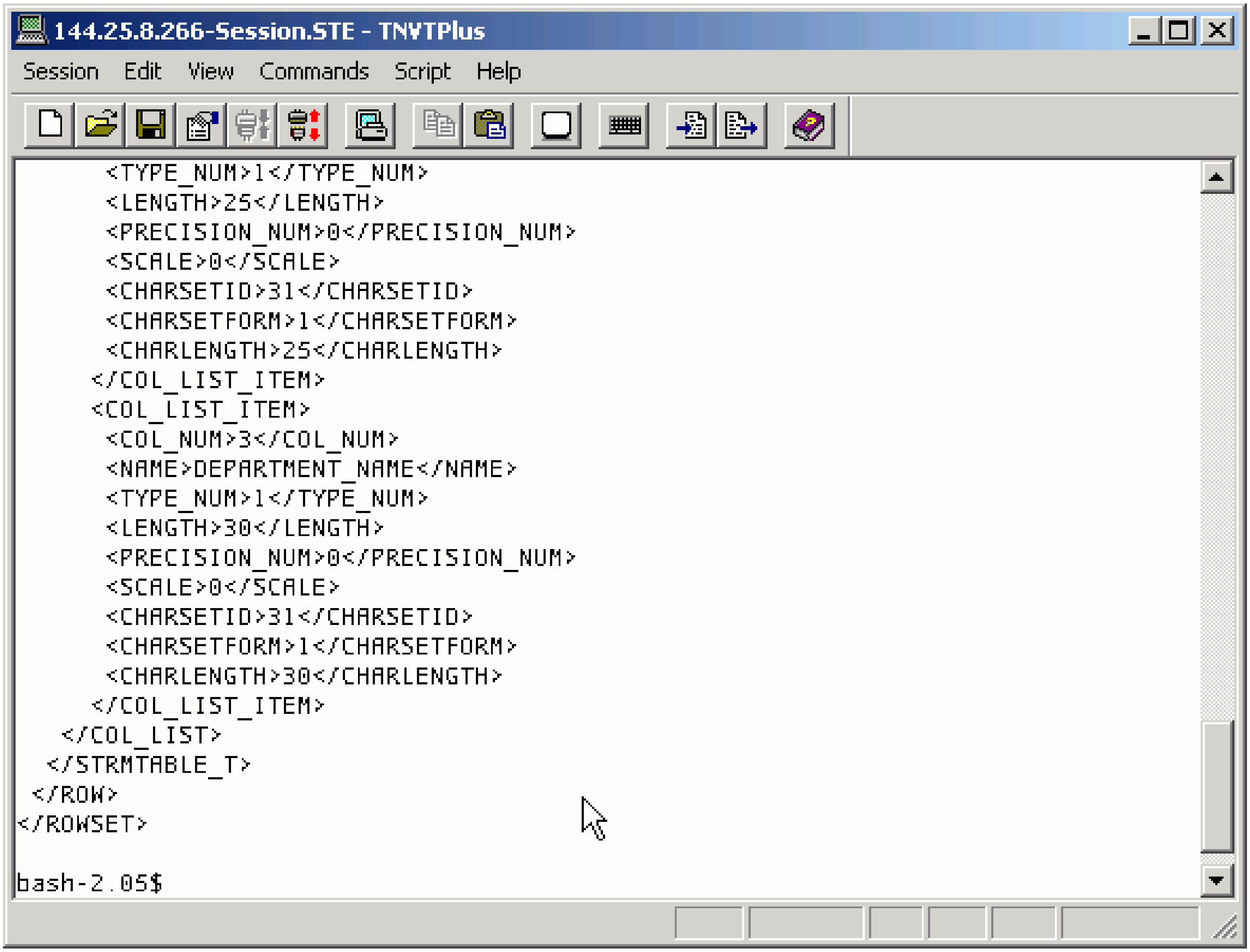


```
     </COL_LIST_ITEM>
     <COL_LIST_ITEM>
      <COL_NUM>3</COL_NUM>
      <NAME>DEPARTMENT_NAME</NAME>
      <TYPE_NUM>1</TYPE_NUM>
      <LENGTH>30</LENGTH>
      <PRECISION_NUM>0</PRECISION_NUM>
      <SCALE>0</SCALE>
      <CHARSETID>31</CHARSETID>
      <CHARSETFORM>1</CHARSETFORM>
      <CHARLENGTH>30</CHARLENGTH>
     </COL_LIST_ITEM>
   </COL_LIST>
  </STRMTABLE_T>
 </ROW>
</ROWSET>
 Michael        Hartstein       MarketingPatFay MarketingDeRaphaely
Purchasing      AlexanderKhoo
PurchasingShelliBaida
PurchasingSigalTobias
Purchasing¿GuyHimuro
Purchasing Karen
Colmenares
Purchasing
bash-2.05$
```

☐ **4.** Now you can take a look at the second file. Open **emp2.exp** using gedit.


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

```
bash-2.05$ more emp2.exp
÷$$
N<?xml version="1.0"?>
<ROWSET>
 <ROW>
  <STRMTABLE_T>
   <VERS_MAJOR>1</VERS_MAJOR>
   <VERS_MINOR>0 </VERS_MINOR>
   <VERS_DPAPI>3</VERS_DPAPI>
   <ENDIANNESS>0</ENDIANNESS>
   <CHARSET>WE8ISO8859P1</CHARSET>
   <NCHARSET>AL16UTF16</NCHARSET>
   <DBTIMEZONE>-07:00</DBTIMEZONE>
   <OWNER_NAME>OE</OWNER_NAME>
   <NAME>EMP_EXT</NAME>
   <COL_LIST>
     <COL_LIST_ITEM>
      <COL_NUM>1</COL_NUM>
      <NAME>FIRST_NAME</NAME>
      <TYPE_NUM>1</TYPE_NUM>
      <LENGTH>20</LENGTH>
      <PRECISION_NUM>0</PRECISION_NUM>
      <SCALE>0</SCALE>
--More--(42%)
```


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

```
      <TYPE_NUM>1</TYPE_NUM>
      <LENGTH>25</LENGTH>
      <PRECISION_NUM>0</PRECISION_NUM>
      <SCALE>0</SCALE>
      <CHARSETID>31</CHARSETID>
      <CHARSETFORM>1</CHARSETFORM>
      <CHARLENGTH>25</CHARLENGTH>
     </COL_LIST_ITEM>
     <COL_LIST_ITEM>
      <COL_NUM>3</COL_NUM>
      <NAME>DEPARTMENT_NAME</NAME>
      <TYPE_NUM>1</TYPE_NUM>
      <LENGTH>30</LENGTH>
      <PRECISION_NUM>0</PRECISION_NUM>
      <SCALE>0</SCALE>
      <CHARSETID>31</CHARSETID>
      <CHARSETFORM>1</CHARSETFORM>
      <CHARLENGTH>30</CHARLENGTH>
     </COL_LIST_ITEM>
   </COL_LIST>
  </STRMTABLE_T>
 </ROW>
</ROWSET>

bash-2.05$
```

The row data and metadata are stored in XML format.

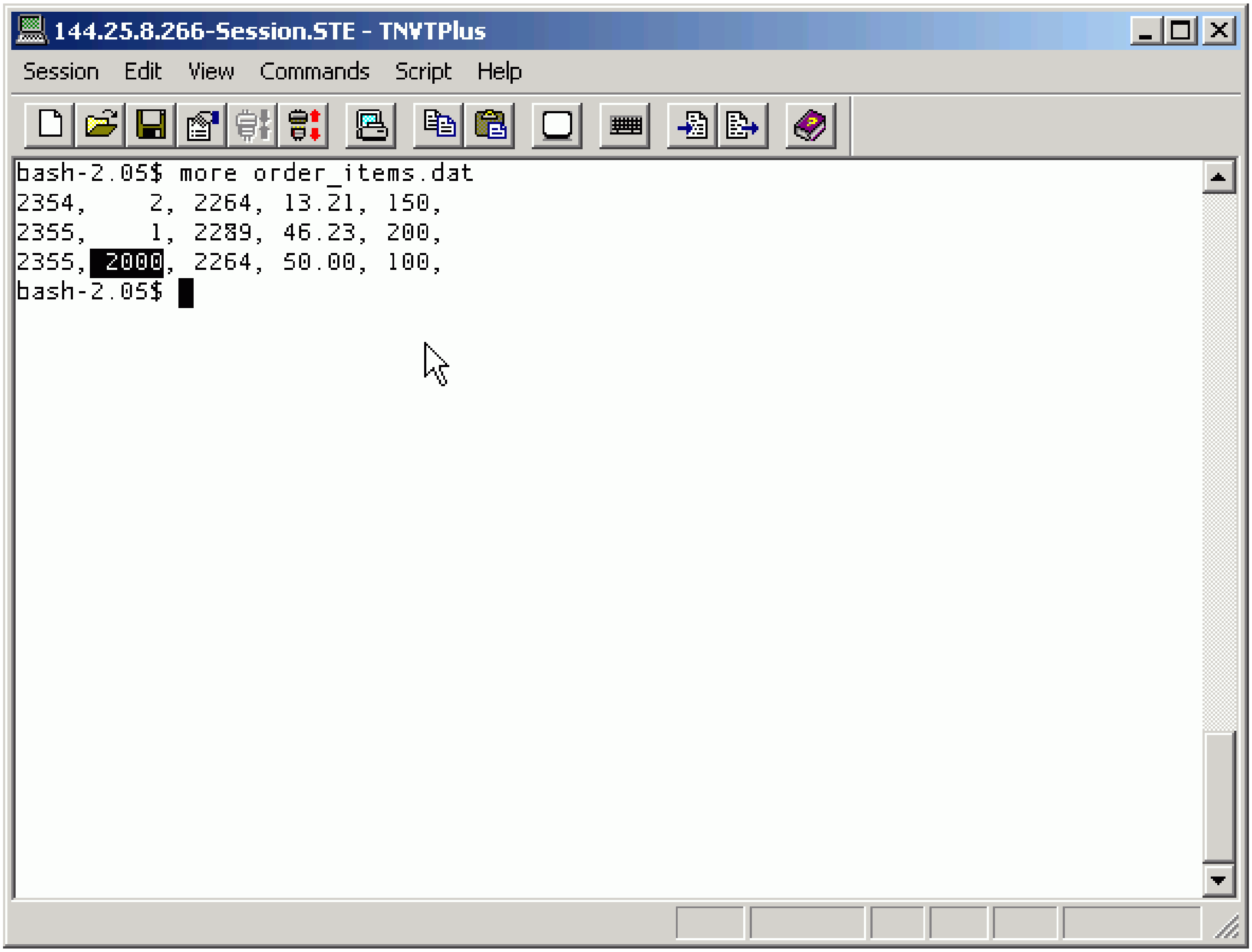
Using Projected Columns

Back to Topic List

You will use external tables to project column data. Perform the following steps:

☐ **1.** Review the contents of the **order_items.dat** external data file. From your terminal window, execute the following command:

```
more order_items.dat
```



```
bash-2.05$ more order_items.dat
2354,    2, 2264, 13.21, 150,
2355,    1, 2289, 46.23, 200,
2355, 2000, 2264, 50.00, 100,
bash-2.05$
```

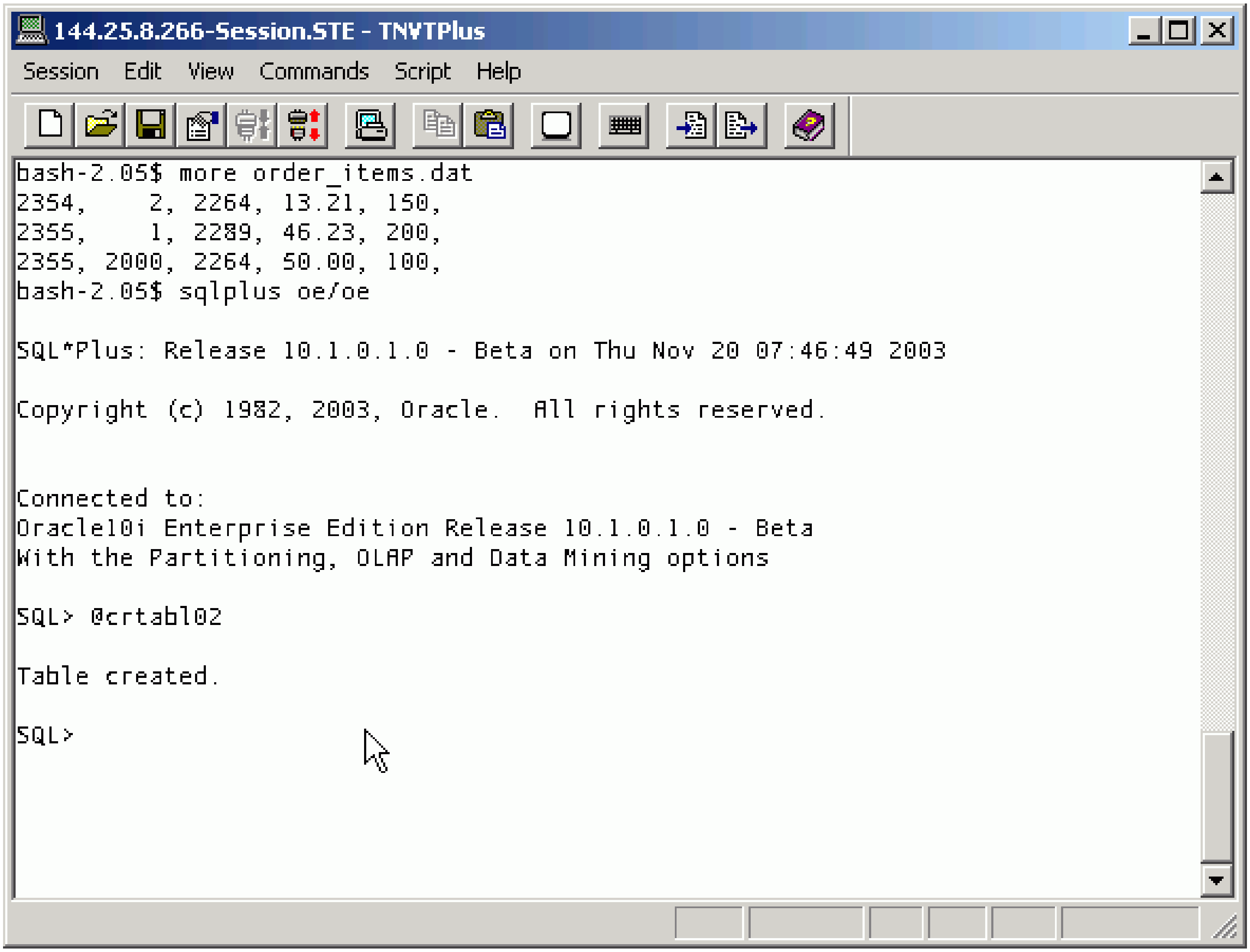
Note the value 2000 in row 3 which is 4 characters. This may cause some trouble.

☐ **2.** Now you can create the external table by executing the following commands:

```
sqlplus oe/oe
@crtabl02
```

The command in the **`crtabl02.sql`** script is as follows:

```
drop table order_items_ext;

create table order_items_ext

( order_id   number(12)

, line_id
NUMBER(3)

, product_id number(6)

, unit_price number(8,2)

, quantity   number(8)

)

ORGANIZATION EXTERNAL

 (TYPE ORACLE_LOADER

  DEFAULT DIRECTORY ext_tab_dir

  ACCESS PARAMETERS (RECORDS DELIMITED BY NEWLINE

                     FIELDS TERMINATED BY ',')

  LOCATION ('order_items.dat')

 )

REJECT LIMIT UNLIMITED;
```

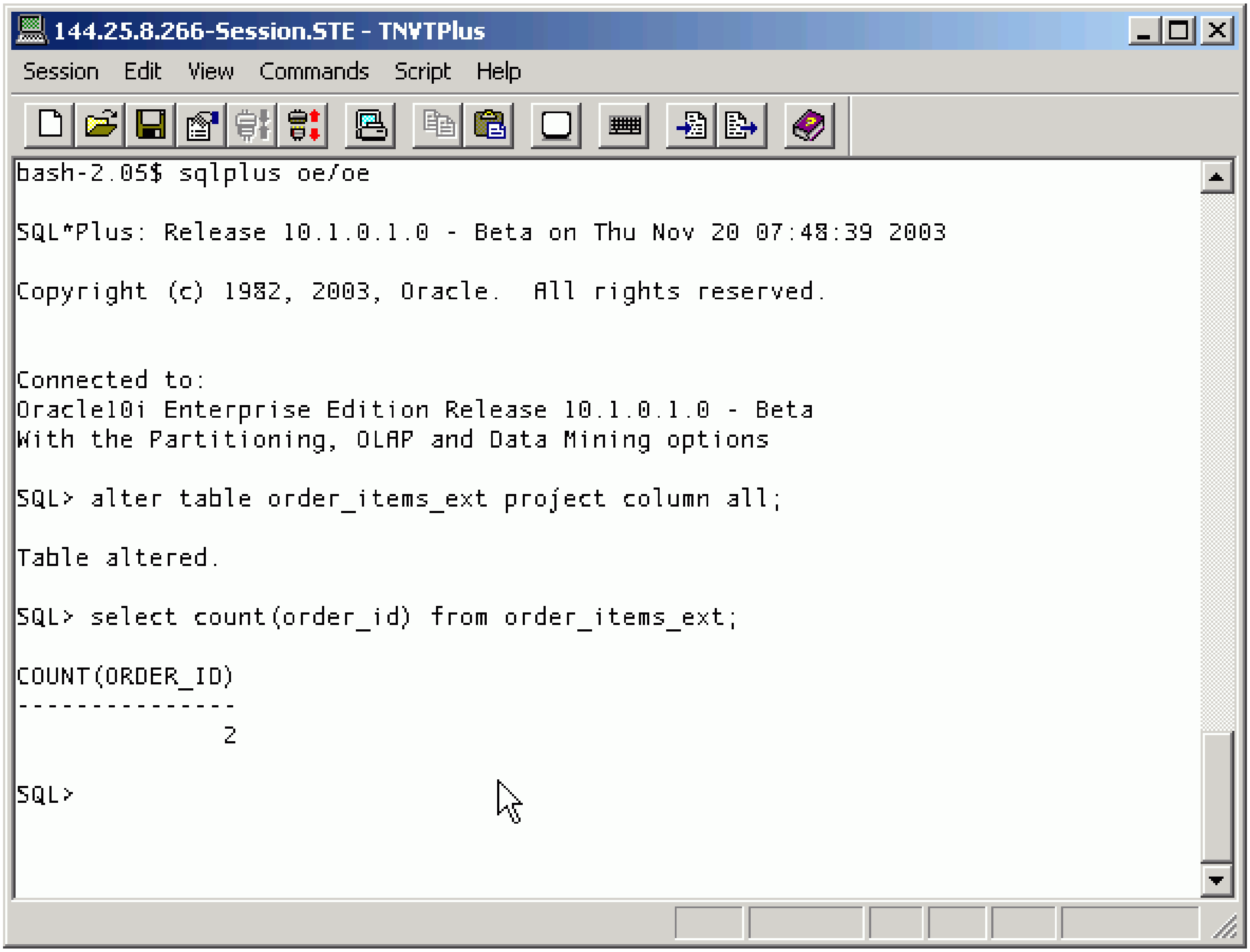


```
bash-2.05$ more order_items.dat
2354,    2, 2264, 13.21, 150,
2355,    1, 2289, 46.23, 200,
2355, 2000, 2264, 50.00, 100,
bash-2.05$ sqlplus oe/oe

SQL*Plus: Release 10.1.0.1.0 - Beta on Thu Nov 20 07:46:49 2003

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


Connected to:
Oracle10i Enterprise Edition Release 10.1.0.1.0 - Beta
With the Partitioning, OLAP and Data Mining options

SQL> @crtabl02

Table created.

SQL>
```

Notice that the format of the LINE_ID is NUMBER(3).

☐ **3.** You will alter the table to set the PROJECT COLUMN attribute to ALL. This is the default. Then you can query the external table. Execute the following command:

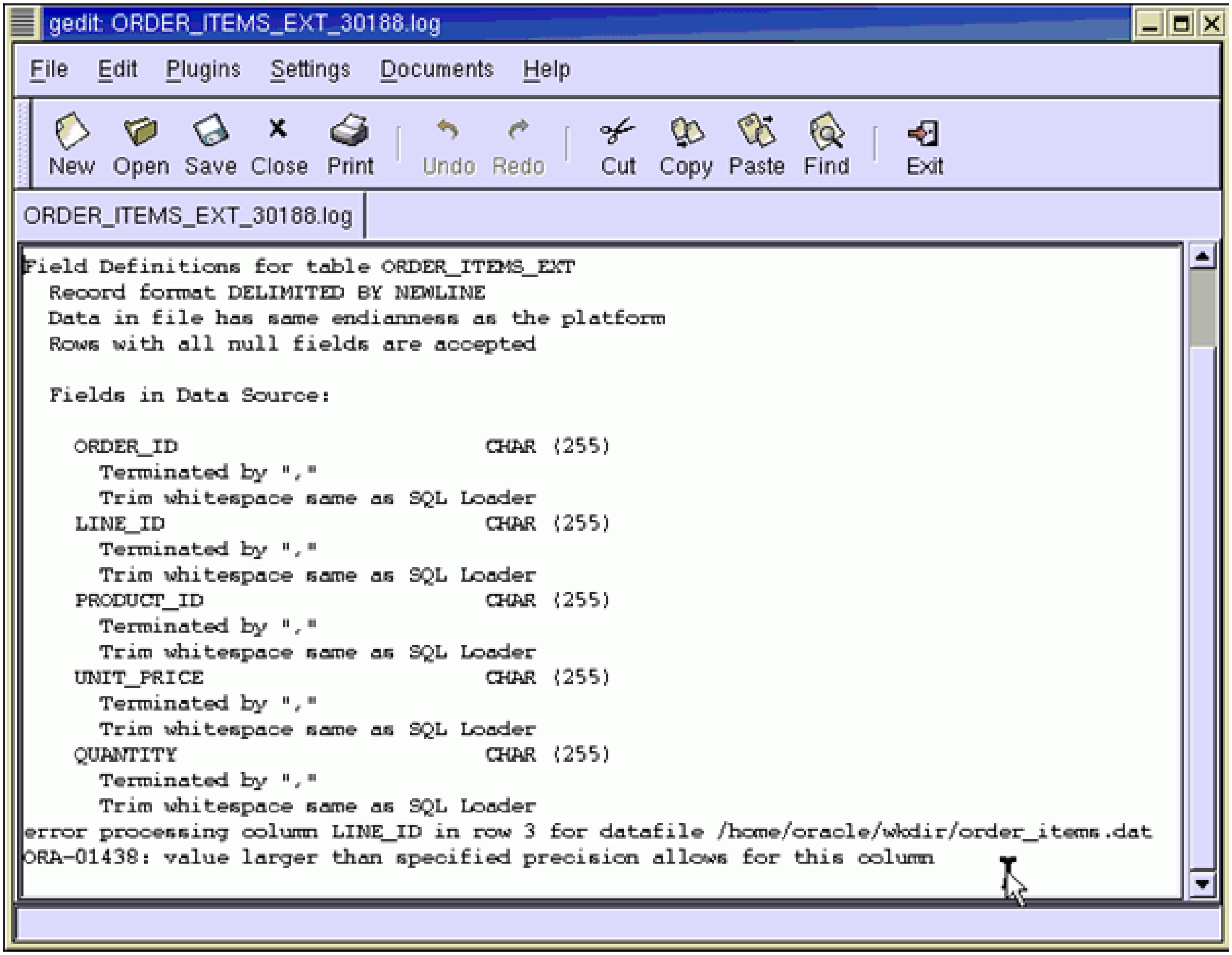
```
alter table order_items_ext project column all;
select count(order_id)from order_items_ext;
```


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

```
bash-2.05$ sqlplus oe/oe

SQL*Plus: Release 10.1.0.1.0 - Beta on Thu Nov 20 07:48:39 2003

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


Connected to:
Oracle10i Enterprise Edition Release 10.1.0.1.0 - Beta
With the Partitioning, OLAP and Data Mining options

SQL> alter table order_items_ext project column all;

Table altered.

SQL> select count(order_id) from order_items_ext;

COUNT(ORDER_ID)
---------------
              2

SQL>
```

Two rows were found.

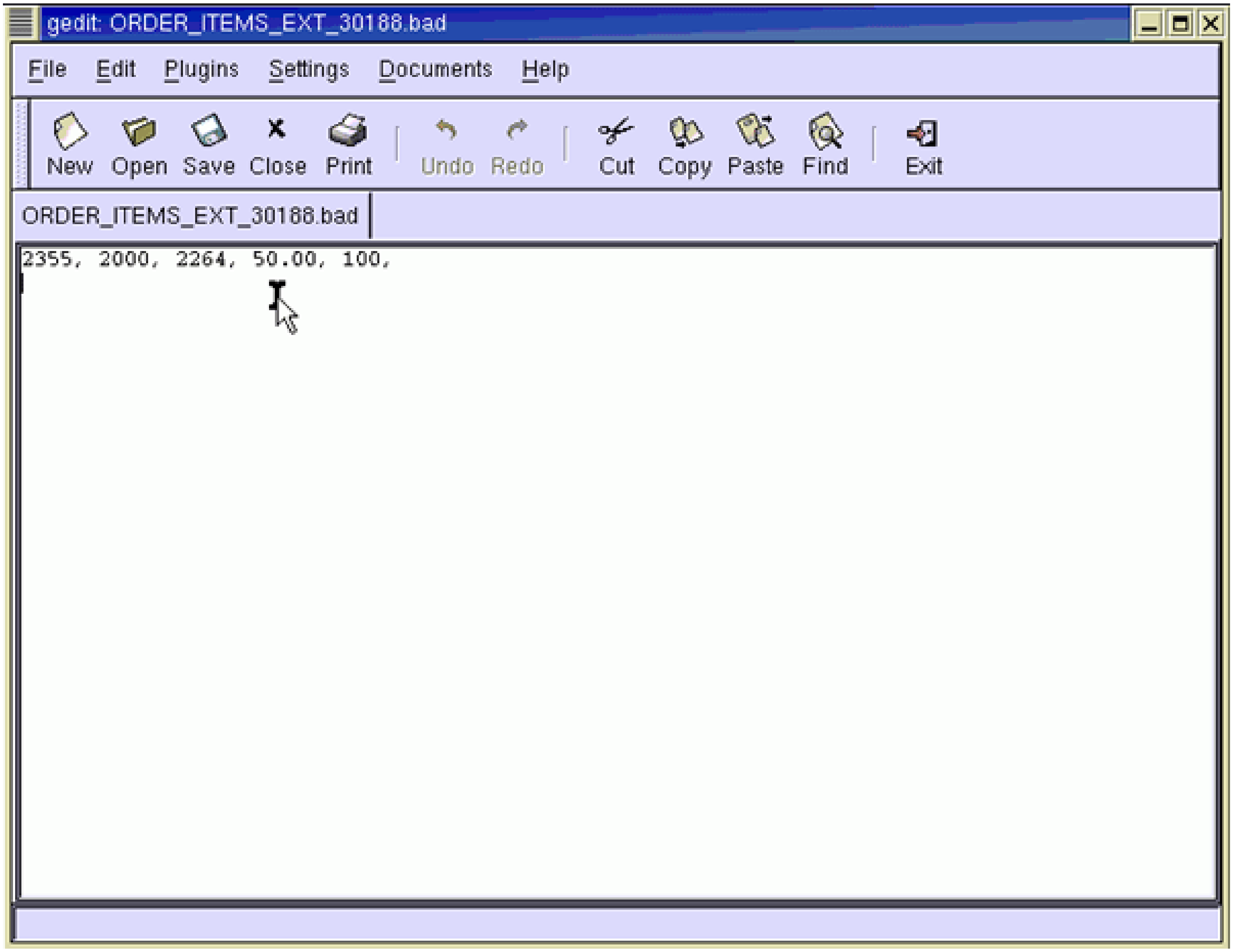
☐ **4.** To see why only two rows were found when there are three rows in the order_items.dat file, you need to look at the Loader log file. Open **ORDER_ITEMS_EXT_#####.log** from gedit.



```
Field Definitions for table ORDER_ITEMS_EXT
  Record format DELIMITED BY NEWLINE
  Data in file has same endianness as the platform
  Rows with all null fields are accepted

  Fields in Data Source:

    ORDER_ID                        CHAR (255)
      Terminated by ","
      Trim whitespace same as SQL Loader
    LINE_ID                         CHAR (255)
      Terminated by ","
      Trim whitespace same as SQL Loader
    PRODUCT_ID                      CHAR (255)
      Terminated by ","
      Trim whitespace same as SQL Loader
    UNIT_PRICE                      CHAR (255)
      Terminated by ","
      Trim whitespace same as SQL Loader
    QUANTITY                        CHAR (255)
      Terminated by ","
      Trim whitespace same as SQL Loader
error processing column LINE_ID in row 3 for datafile /home/oracle/wkdir/order_items.dat
ORA-01438: value larger than specified precision allows for this column
```

Notice that the record that had a LINE_ID greater than 3 numbers was rejected when the external table was accessed. That row never reached the SQL query processing.

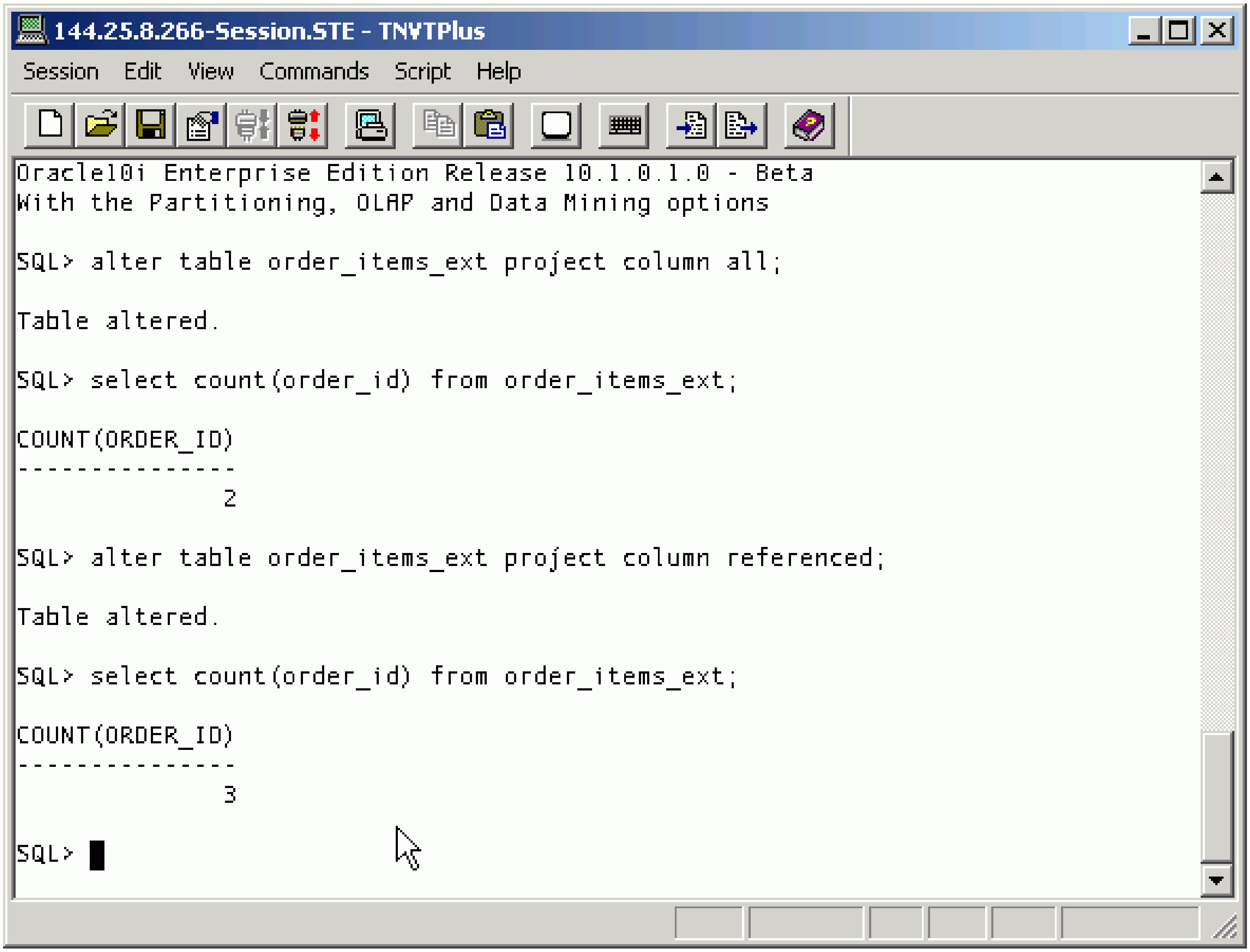
☐ **5.** To see the row that was rejected, open **ORDER_ITEMS_EXT_#####.bad** from gedit.



```
2355, 2000, 2264, 50.00, 100,
```

☐ **6.** Now you can change the PROJECT COLUMN attribute to REFERENCED and run the same query to see what happens. Execute the following command:

```
alter table order_items_ext project column referenced;
select count(order_id)from order_items_ext;
```



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

```
Oracle10i Enterprise Edition Release 10.1.0.1.0 - Beta
With the Partitioning, OLAP and Data Mining options

SQL> alter table order_items_ext project column all;

Table altered.

SQL> select count(order_id) from order_items_ext;

COUNT(ORDER_ID)
---------------
              2

SQL> alter table order_items_ext project column referenced;

Table altered.

SQL> select count(order_id) from order_items_ext;

COUNT(ORDER_ID)
---------------
              3

SQL>
```

Three rows were found. This query projected the external data to the field needed (ORDER_ID) so all three records were accepted.

Place the cursor on this icon to hide all screenshots.