## Site search

Custom Search

## User login

**Username** *

**Password** *

- **Request new password**

Log in

## Site navigation

- **About**
- **Blogs**
- **Feed aggregator**
- **Books**
- **Events**
- **FAQ's**
- **Forums**
- **Mailing Lists**
- **Papers**
- **Scripts**
- **Tools**
- **USENET News**
- **Wiki**
- **XML Feeds**

## Recent blog posts

- **Tablespace Capacity Planning Report**
- **Apriori Algorithm in pl/sql**
- **Linear Regression Algorithm in pl/sql**
- **NAIVE BAYES ALGORITHM WITH SQL & PL/SQL**
- **Help Santa with SQL and get gifts from Devart!**
- **weblogic administration online training**
- **Oracle performance tuning -A Foolproof Query Execution plan with incorrect statistics**
- **12.2 upgrade - it can break all your outgoing https calls**
- **Configuring Oracle**

**Home** » **Blogs** » **Natalka Roshak's blog**

# External Tables

Submitted by Natalka Roshak on Sun, 2006-03-05 18:00

**articles:**
**RDBMS Server**

External Tables let you query data in a flat file as though the file were an Oracle table. In 9i, only read operations were permitted; in 10g, you can also write out data to an external table, although you can't write to an existing table.

While external tables can be queried, they're not usable in many ways regular Oracle tables are. You cannot perform any DML operations on external tables other than table creation; one consequence is that you can't create an index on an external table. External tables are largely used as a convenient way of moving data into and out of the database.

Oracle uses SQL*Loader functionality, through the ORACLE_LOADER access driver to move data from the flat file into the database; it uses a Data Pump access driver to move data out of the db into a file in an Oracle-proprietary format, and back into the database from files of that format. While there are some behaviour differences and restrictions, you can think of external tables as a convenient, SQL-based way to use SQL*Loader and Data Pump functionality.

For example, suppose that you receive a daily .csv report from another department. Instead of writing a SQL*Loader script to import each day's .csv file into your database, you can simply create an external table and write an "insert ... select" SQL query to insert the data directly into your tables. Place the day's CSV file in the location specified in the external table definition, run the query, and you're done.

## Creating an external table

Since an external table's data is in the operating system, its data file needs to be in a place Oracle can access it. So the first step is to create a directory and grant access to it.

First create the directory in the operating system, or choose an existing directory. It must be a real directory, not a symlink. Make sure that the OS user that the Oracle binaries run as has read-write access to this directory. Note: Be sure not to use a directory you should be keeping secure, such as an Oracle datafile, program, log or configuration file directory. And if the data you'll be putting there is sensitive, make sure that other OS users don't have permissions on this directory.

```
$ cd /oracle/feeds/
$ mkdir xtern
$ mkdir xtern/mySID
$ mkdir xtern/mySID/data
$ ls  -l /oracle/feeds/xtern/mySID
total 8
drwx------   2 oracle oinstall 4096 Mar  1 17:05 data
```

Put the external table's data file in the data directory. In this example, I'll use the following CSV file:

employee_report.csv:

```
001,Hutt,Jabba,896743856,jabba@thecompany.com,18
002,Simpson,Homer,382947382,homer@thecompany.com,20
003,Kent,Clark,082736194,superman@thecompany.com,5
004,Kid,Billy,928743627,billythkid@thecompany.com,9
005,Stranger,Perfect,389209831,nobody@thecompany.com,23
006,Zoidberg,Dr,094510283,crustacean@thecompany.com,1
```

You must actually move or copy the file to the data directory; symlinks won't cut it. Again, make sure that if the data is sensitive, only the Oracle user can read or write to it.

The next step is to create this directories in Oracle, and grant read/write access on it to the Oracle user who will be creating the external table. When you create the directory, be sure to use the directory's full path, and don't include any symlinks in the path -- use the actual full path.

```
SQL> connect sys as sysdba
```

Financial Accounting Hub
- Sending Email from PLSQL from Gmail

More

```
Enter password:
Connected.
SQL> create or replace directory xtern_data_dir
  2  as '/oracle/feeds/xtern/mySID/data';

Directory created.

SQL> grant read,write on directory xtern_data_dir to bulk_load;

Grant succeeded.
```

The last step is to create the table. The CREATE TABLE statement for an external table has two parts. The first part, like a normal CREATE TABLE, has the table name and field specs. This is followed by a block of syntax specific to external tables, which lets you tell Oracle how to interpret the data in the external file.

```
SQL> connect bulkload
Enter password:
Connected.
SQL> create table xtern_empl_rpt
  2      ( empl_id varchar2(3),
  3        last_name varchar2(50),
  4        first_name varchar2(50),
  5        ssn varchar2(9),
  6       email_addr varchar2(100),
  7        years_of_service number(2,0)
  8      )
  9      organization external
 10      ( default directory xtern_data_dir
 11        access parameters
 12        ( records delimited by newline
 13          fields terminated by ','
 14        )
 15        location ('employee_report.csv')
 16    );

Table created.
```

At this point, Oracle hasn't actually tried to load any data. It doesn't attempt to check the validity of many of the external-table-specific parameters you pass it. The CREATE TABLE statement will succeed even if the external data file you specify doesn't actually exist.

With the create table statement, you've created table metadata in the data dictionary and instructed Oracle how to direct the ORACLE_LOADER access driver to parse the data in the datafile. Now, kick off the load by accessing the table:

```
SQL> select * from xtern_empl_rpt ;

EMP LAST_NAME  FIRST_NAME SSN       EMAIL_ADDR                     YEARS_OF_SERVICE
--- ---------- ---------- --------- ------------------------------ ----------------
001 Hutt       Jabba      896743856 jabba@thecompany.com                         18
002 Simpson    Homer      382947382 homer@thecompany.com                         20
003 Kent       Clark      082736194 superman@thecompany.com                       5
004 Kid        Billy      928743627 billythkid@thecompany.com                     9
005 Stranger   Perfect    389209831 nobody@thecompany.com                        23
006 Zoidberg   Dr         094510283 crustacean@thecompany.com                     1

6 rows selected.
```

Oracle used the ORACLE_LOADER driver to process the file, and just as with SQL*Loader, it's created a log file that you can inspect to see what just happened. The log file -- and the "bad" and "discard" files -- will have been written to the directory you specified as the "default directory" in your CREATE TABLE statement, and the file names default to tablename_ospid :

```
$ ls -l
total 16
-rw-r--r--   1 oracle oinstall 3652 Mar  1 19:41 XTERN_EMPL_RPT_26797.log
-rw-------   1 oracle oinstall  313 Mar  1 18:34 employee_report.csv
```

If Oracle was unable to process the data given the access parameters you specified, you'll get an error on the command line and in the log file, and there will also be a bad and/or discard file. (Note: if you're copying and pasting data into your external data file, be sure not to put a newline after the last record, or SQL*Loader will expect a seventh record, and you'll get an error when you try to select from the external table.)

You may want to configure separate directories for the SQL*Loader output files -- the LOG file, the DISCARD file and the BAD file -- as well as for the external table data. You can lump all four in the same directory, as we did in the previous example, although it's a bad idea: a naming mishap could have you overwriting one external table's data file with another's bad file. I like to have one directory for data files, and one for log/bad/discard files:

```
$ cd xtern/mySID
$ mkdir log
$ ls -l
total 16
drwx------  2 oracle oinstall 4096 Mar  1 17:33 data
drwx------  2 oracle oinstall 4096 Mar  1 17:32 log
```

Again, these must be actual directories, not symlinks, and be sure to set the permissions appropriately. To eliminate the possibility of any naming mishap, you can grant READ access only on /.../data, and WRITE access only on /..../log, to the user creating the external tables.

You can use ALTER TABLE to change the access parameters without dropping and redefining the whole table:

```
SQL> alter table xtern_empl_rpt
  2  access parameters
  3        ( records delimited by newline
  4          badfile xtern_log_dir:'xtern_empl_rpt.bad'
  5          logfile xtern_log_dir:'xtern_empl_rpt.log'
  6          discardfile xtern_log_dir:'xtern_empl_rpt.dsc'
  7          fields terminated by ','
  8        ) ;

Table altered.
```

Alternatively, you can set up the table so that no log, discard or bad files are generated. SELECTing data from the table will still fail if the maximum number of rejects is exceeded, just as in SQL*Loader. You can change the reject limit for an external table with an ALTER TABLE statement:

```
SQL> ALTER TABLE XTERN_EMPL_RPT SET REJECT_LIMIT 100;
```

## Loading the data into your tables

Where external tables really shine are in the ease with which you can load their data into your tables. A particularly nice feature is that you can use any valid function that the current Oracle user has rights on to transform the raw data before loading it into your database tables. For example, suppose you had a function, get_bday_from_ssn (ssn in varchar2) that looked up an employee's birth date given their SSN. You can use that function to populate a BIRTH_DATE column in your local database table in the same step as you load the data into it.

```
SQL> create table empl_info as
  2  (select empl_id, last_name, first_name, ssn, get_bday_from_ssn (ssn) birth_dt
  3* from xtern_empl_rpt)
SQL> /

Table created.

SQL> select * from empl_info ;

EMP LAST_NAME  FIRST_NAME SSN       BIRTH_DT
--- ---------- ---------- --------- ----------
001 Hutt       Jabba      896743856 03/11/1939
002 Simpson    Homer      382947382 11/01/1967
003 Kent       Clark      082736194 01/15/1925
004 Kid        Billy      928743627 07/20/1954
005 Stranger   Perfect    389209831 10/23/1980
006 Zoidberg   Dr         094510283 04/04/2989

6 rows selected.
```

## Unloading data into an external file...

Oracle 10g lets you create a new external table from data in your database, which goes into a flat file pushed from the database using the ORACLE_DATAPUMP access driver. This flat file is in an Oracle-proprietary format that can be read by DataPump. The syntax is similar to the CREATE TABLE... ORGANIZATION EXTERNAL above, but simpler -- since you can't specify the data format, you can specify very few access_parameters. The key difference is that you must specify the access driver, ORACLE_DATAPUMP, since the access driver defaults to ORACLE_LOADER.

```
SQL> create table export_empl_info
  2  organization external
  3  ( type oracle_datapump
  4    default directory xtern_data_dir
  5    location ('empl_info_rpt.dmp')
  6* ) as select * from empl_info
SQL> /
```

```
Table created.

SQL> select * from export_empl_info ;

EMPL_ID LAST_NAME       FIRST_NAME      SSN       BIRTH_DT
------- --------------- --------------- --------- ----------
001     Hutt            Jabba           896743856 01/01/1979
002     Simpson         Homer           382947382 01/01/1979
003     Kent            Clark           082736194 01/01/1979
004     Kid             Billy           928743627 01/01/1979
005     Stranger        Perfect         389209831 01/01/1979
006     Zoidberg        Dr              094510283 01/01/1979

6 rows selected.
```

## ... and back in again

You can now move the file you just created, empl_info_rpt.dmp, to another system and create an external table to read the data:

```
SQL> connect joe/some.where.else
Connected.
SQL> create table import_empl_info
  2  ( empl_id varchar2(3),
  3    last_name varchar2(50),
  4    first_name varchar2(50),
  5    ssn varchar2(9),
  6    birth_dt date
  7  )
  8  organization external
  9  ( type oracle_datapump
 10    default directory xtern_data_dir
 11    location ('empl_info_rpt.dmp')
 12  ) ;

Table created.

SQL> select * from import_empl_info ;

EMPL_ID LAST_NAME       FIRST_NAME      SSN       BIRTH_DT
------- --------------- --------------- --------- ----------
001     Hutt            Jabba           896743856 01/01/1979
002     Simpson         Homer           382947382 01/01/1979
003     Kent            Clark           082736194 01/01/1979
004     Kid             Billy           928743627 01/01/1979
005     Stranger        Perfect         389209831 01/01/1979
006     Zoidberg        Dr              094510283 01/01/1979

6 rows selected.
```

## Conclusion

We've seen an introduction to loading and unloading data with external tables. External tables in 9i and 10g provide a convenient, seamless way to move data in and out of the database, integrating SQL*Loader and Data Pump functionality with the power, scriptability and ease of SQL statements. It's definitely worth considering external tables the next time you have a daily upload or download to arrange.

## About the author

Natalka Roshak is a senior Oracle and Sybase database administrator, analyst, and architect. She is based in Kingston, Ontario and consults across North America. More of her scripts and tips can be found in her online DBA toolkit at **http://toolkit.rdbms-insight.com/.**

» **Natalka Roshak's blog** | **Log in** to post comments

## Comments

### Thanks

**Permalink** Submitted by Jim Lu (not verified) on Tue, 2006-03-28 10:02.

This is the best summary of external table I have ever read. It is simple and clear. It really helped me understanding the ETL concept. Thank you very much for doing this.

» **Log in** to post comments

### External Tables

**Permalink** Submitted by Kevin Closson (not verified) on Wed, 2006-06-07 15:10.

In 2002, PolyServe and XIOtech partnered to perform a huge Proof of Concept (audited by IDC) of a 10TB database on a 10 node cluster and ramped the active OLTP users count to 10,000 users. The project was called The Tens.

The entire setup was stored in PolyServe Matrix Server CFS, to include External Tables. With RAC, External Tables can be accessed via IPQO (Intra-node Parallel Query). The source data for the load was pipe-delimited and instered with APPEND as select /*+ PARALLEL */ * from External_table.

The paper is a good reference for VLDB with Linux/RAC and can be obtained at:

**http://www.polyserve.com/requestinfo_form.php?PDF3=checked**

» **Log in** to post comments

### Fantastic stuff

**Permalink** Submitted by Arun (not verified) on Thu, 2006-08-10 07:28.

Thanks a lot for this wonderful explanation.
It's so easy and efficient.

» **Log in** to post comments

### Clear Cut

**Permalink** Submitted by Maria Antony (not verified) on Thu, 2006-08-24 09:43.

I always like the manuals that are 'Clear Cut' and easy to understand for all level of people. I could see that in your paper.

Good work; Keep it up!

» **Log in** to post comments

### Great Stuff

**Permalink** Submitted by Prasad (not verified) on Sat, 2006-09-16 21:04.

The most structured Oracle article I have ever seen - each line answered the next question that popped up in my mind.

A big thank you!

» **Log in** to post comments

### Helpful!

**Permalink** Submitted by Guy Conan (not verified) on Wed, 2006-11-15 20:57.

I'm now working on a project where I have to deal with data transfert between different databases on different systems using Oracle and this Great explanation saved me hours!
Thanks again!

» **Log in** to post comments

### Sweet !!!

**Permalink** Submitted by amardeep.sidhu on Thu, 2007-12-27 10:41.

Excellent article. Simple and precise...

Thanks :)

Sidhu

» **Log in** to post comments

### Really greate but i need help.

**Permalink** Submitted by bandarna on Mon, 2008-03-17 02:19.

I used above example and finally got error. can any one help me?

1. created new folder in my c drive as:
C:\oracle\xtrn_data

2. created a file employee.csv using above data (from this artical) to this directory
C:\oracle\xtrn_data\employee.csv

3. connected as sys and created xtern_data_dir directory object:

```
SQL> conn sys as sysdba;
Enter password: ***
Connected.
SQL> create or replace Directory xtern_data_dir as 'C:\oracle\xtrn_data';

Directory created.
```

4. Issued grant read and write privileges to directory object to Scott.

```
SQL> grant read, write on directory xtern_data_dir to Scott;

Grant succeeded.
```

5. connected as Scott and created external table:

```
SQL> create table xtern_empl_rpt
2 ( empl_id varchar2(3),
3 last_name varchar2(50),
4 first_name varchar2(50),
5 ssn varchar2(9),
6 email_addr varchar2(100),
7 years_of_service number(2,0)
8 )
9 organization external
10 ( default directory xtern_data_dir
11 access parameters
12 ( records delimited by newline
13 fields terminated by ','
14 )
15 location ('employee_report.csv')
16 );
Table created.
```

6. I have error when i issue select command.

```
SQL>select * from xtern_empl_rpt;
select * from xtern_empl_rpt
*
ERROR at line 1:
ORA-29913: error in executing ODCIEXTTABLEOPEN callout
ORA-29400: data cartridge error
KUP-04040: file employee.csv in XTERN_DATA_DIR not found
ORA-06512: at "SYS.ORACLE_LOADER", line 14
ORA-06512: at line 1
```

7. log file created:

```
LOG file opened at 03/13/08 22:58:10

Field Definitions for table XTERN_EMPL_RPT
Record format DELIMITED BY NEWLINE
```

```
Data in file has same endianness as the platform

Fields in Data Source:

EMPL_ID CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
LAST_NAME CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
FIRST_NAME CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
SSN CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
EMAIL_ADDR CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
YEARS_OF_SERVICE CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
KUP-04040: file employee.csv in XTERN_DATA_DIR not found
```

» **Log in** to post comments

### reason for error

**Permalink** Submitted by meuser on Tue, 2008-03-18 15:44.

The reason for the error is a mismatch between the filename in the directory (employee.csv) and the filename in the create external table statement location ('employee_report.csv')

Hopes this helps to resolve the problem

Martin

» **Log in** to post comments

### Nice Example and a suggestion...

**Permalink** Submitted by debrucer on Wed, 2008-03-26 19:46.

This is definitely something I needed to know about since I have been asked to complete a task that this will be perfect for :) Thank you! Thank you!

Petty perhaps, when you've had something spoon fed to you like this; but, it would be a nice touch when articles with SQL examples gave you a shot without the SQL> Prompt so you could cut and paste the entire code. As it was, I had to do 18 individual lines (okay, 16, I didn't cut and past the parenthesis).

Maybe a link to "download code" or something like that would be the solution.

At any rate, did I say "thanks"?

David

» **Log in** to post comments

### Really a great article with clear understanding................

**Permalink** Submitted by singh.neerajin on Mon, 2008-04-28 01:19.

Is there any kind of similar thing we can use with XML file. Presently I am using the XSLT to transform the data into text file & then loading it to database through CONTROL file using SQL loader. Is there any way we cna directly fetch the XML file to database.

» **Log in** to post comments

### Help

**Permalink** Submitted by jluziano on Thu, 2008-06-12 14:40.

I created a External table with 3 fileds, but in some register the last filed is empty, then this register is BAD.

example:

aa|bb|cc -> register OK
dd|ee|ff -> register OK
gg|hh| -> register BAD
ii|jj|kk -> register OK

How can I do to the third register to be OK.

Sorry my english
Thank you

» **Log in** to post comments

### Neatly explained article.

**Permalink** Submitted by renu4dwh on Thu, 2008-07-03 06:29.

Neatly explained article. Thank you so much

» **Log in** to post comments

### External Datapump managed tables correction

**Permalink** Submitted by hrivera99 on Thu, 2008-08-28 11:31.

Just a correction. You stated on: "Unloading data into an external file..."

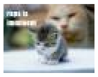
"Oracle 10g lets you create a new external table from data in your database, which goes into a flat file "

It actually doesn't go to a flat file, but to a binary proprietary format file which is seen, from an OEL4.5 OS perspective, as a "PDP-11 UNIX/RT ldp" file type.

~ Madrid
**http://hrivera99.blogspot.com**

» **Log in** to post comments

### I'm getting same error as



**Permalink** Submitted by barrywalkors on Sat, 2008-09-06 19:24.

I'm getting same error as bardarna...

I used above example and finally got error. can any one help me?

1. created new folder in my c drive as:
C:\oracle\xtrn_data

2. created a file employee.csv using above data (from this artical) to this directory
C:\oracle\xtrn_data\employee.csv

3. connected as sys and created xtern_data_dir directory object:

```
SQL> conn sys as sysdba;
Enter password: ***
Connected.
SQL> create or replace Directory xtern_data_dir as 'C:\oracle\xtrn_data';

Directory created.
```

4. Issued grant read and write privileges to directory object to Scott.

```
SQL> grant read, write on directory xtern_data_dir to Scott;

Grant succeeded.
```

5. connected as Scott and created external table:

```
SQL> create table xtern_empl_rpt
2 ( empl_id varchar2(3),
3 last_name varchar2(50),
4 first_name varchar2(50),
5 ssn varchar2(9),
6 email_addr varchar2(100),
7 years_of_service number(2,0)
8 )
9 organization external
10 ( default directory xtern_data_dir
11 access parameters
12 ( records delimited by newline
13 fields terminated by ','
14 )
15 location ('employee_report.csv')
16 );
Table created.
```

6. I have error when i issue select command.

```
SQL>select * from xtern_empl_rpt;
select * from xtern_empl_rpt
*
ERROR at line 1:
ORA-29913: error in executing ODCIEXTTABLEOPEN callout
ORA-29400: data cartridge error
KUP-04040: file employee.csv in XTERN_DATA_DIR not found
ORA-06512: at "SYS.ORACLE_LOADER", line 14
ORA-06512: at line 1
```

7. log file created:

```
LOG file opened at 03/13/08 22:58:10

Field Definitions for table XTERN_EMPL_RPT
Record format DELIMITED BY NEWLINE
Data in file has same endianness as the platform

Fields in Data Source:

EMPL_ID CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
LAST_NAME CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
FIRST_NAME CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
SSN CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
```

EMAIL_ADDR CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
YEARS_OF_SERVICE CHAR (255)
Terminated by ","
Trim whitespace same as SQL Loader
KUP-04040: file employee.csv in XTERN_DATA_DIR not found

» **Log in** to post comments

### I created a External table

**Permalink** Submitted by Daniel Saltman on Thu, 2008-12-18 18:03.

I created a External table with 3 fileds, but in some register the last filed is empty, then this register is BAD.

example:

aa|bb|cc -> register OK
dd|ee|ff -> register OK
gg|hh| -> register BAD
ii|jj|kk -> register OK

How can I do to the third register to be OK.

-Daniel Saltman
BostonDan@gmail.com

» **Log in** to post comments

### Use the 'missing field values

**Permalink** Submitted by yelluh on Sat, 2011-06-11 11:07.

Use the 'missing field values are null' fields-clause.
See also **http://download.oracle.com/docs/cd/B10501_01/server.920/a96652/ch12.htm.**

Rgds, Jelle

» **Log in** to post comments

### Comma as a character

**Permalink** Submitted by navkrish on Wed, 2009-04-01 00:03.

What if my column has comma (,) as one of the character in it?

employee_report.csv:

001,"Hutt",Jabba,896743856,jabba@thecompany.com,18
002,"Simpson,A",Homer,382947382,homer@thecompany.com,20
003,"Kent",Clark,082736194,superman@thecompany.com,5
004,"Kid,K",Billy,928743627,billythkid@thecompany.com,9
005,"Stranger",Perfect,389209831,nobody@thecompany.com,23
006,"Zoidberg,S",Dr,094510283,crustacean@thecompany.com,1

Watch for "Simpson,A" in Row 2.

In this case the column gets broke as Simpson and A. I want it to be together in Column 2.

» **Log in** to post comments

### Worked with having optionally

**Permalink** Submitted by navkrish on Wed, 2009-04-01 00:43.

Worked with having
optionally enclosed by '"' clause in access parameters

» **Log in** to post comments

### Nice article but...

**Permalink** Submitted by wakula on Thu, 2009-04-16 08:31.

This is a very nice article. Thanks Natalka.
However the data security is an issue - if only Oracle has access to the .csv file then all the users having access to the database have access to this file. Or am I missing something?

» **Log in** to post comments

### External Tables

**Permalink** Submitted by hsherif on Tue, 2009-11-10 13:41.

I am new to Oracle external tables and I found this article very illumnating in a short and concise manner. Thank you indeed!

» **Log in** to post comments

### Data files and log files

**Permalink** Submitted by shadab on Wed, 2010-04-28 07:34.

Can I use CREATE or REPLACE EXTERNAL_TABLE ...?
so that the previous data file is replaced by the new one.

Else
Could anybody tell me how do to delete the files created so as to free the space.

» **Log in** to post comments

### Tab seperated values

**Permalink** Submitted by soagra on Mon, 2010-07-12 08:05.

How can I handle tab separated values.
I'm also finding difficulty in converting formats (char to number , char to date) while putting data in external table.

Please help me out.

» **Log in** to post comments

### Tab separated values

**Permalink** Submitted by udaikiran_g on Wed, 2010-09-08 04:35.

In order to handle tab separated values while reading data from the file use FIELDS TERMINATED BY 0X'09' (zeroX'zero9').

» **Log in** to post comments

### How to dump data from multiple files?

**Permalink** Submitted by vineetarora3 on Tue, 2011-03-15 03:22.

Hi,

Am new to external table concept but got an idea from your well written article.

I have some doubts.

You have created the below table:

```
create table empl_info as
(select empl_id, last_name, first_name, ssn, get_bday_from_ssn (ssn) birth_dt
from xtern_empl_rpt)
```

Is it possible to add more columns to empl_info table apart from those present in the csv file?

Also, I would like to know the best way to dump data from multiple xls/csv files into oracle data tables? This dumping might be done daily as content of files will change. So shall we have multiple external tables for this?

Thanks,
Vineet

» **Log in** to post comments

.:: **Blogger Home** :: **Wiki Home** :: **Forum Home** :: **Privacy** :: **Contact** ::.