Assignment 1

TASK 1:

Create a database named 'custom'.

Create a table named temperature_data inside custom having below fields:

1. date (mm-dd-yyyy) format

2. zip code

3. temperature

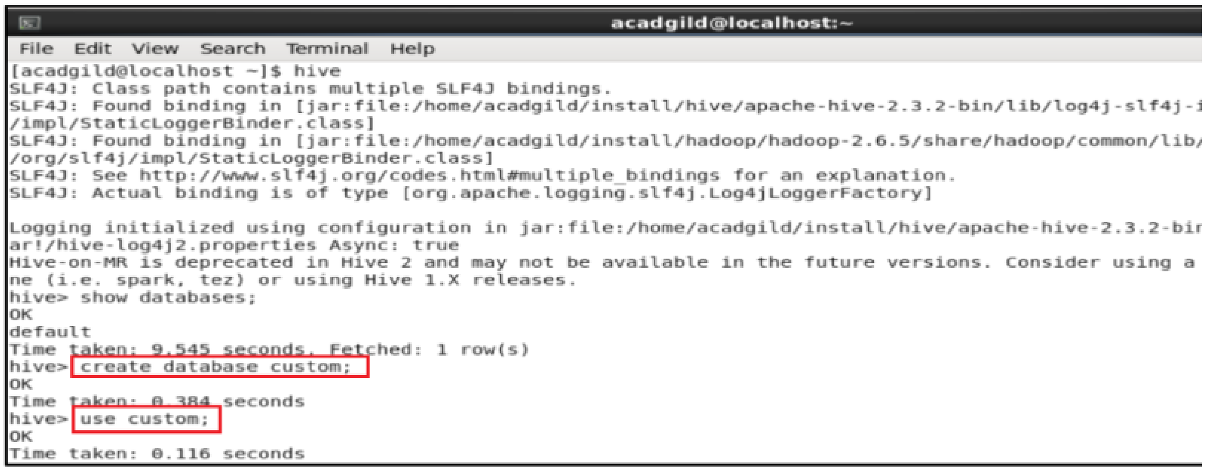
The table will be loaded from comma-delimited file.

Load the dataset.txt (which is ',' delimited) in the table.

Database creation :

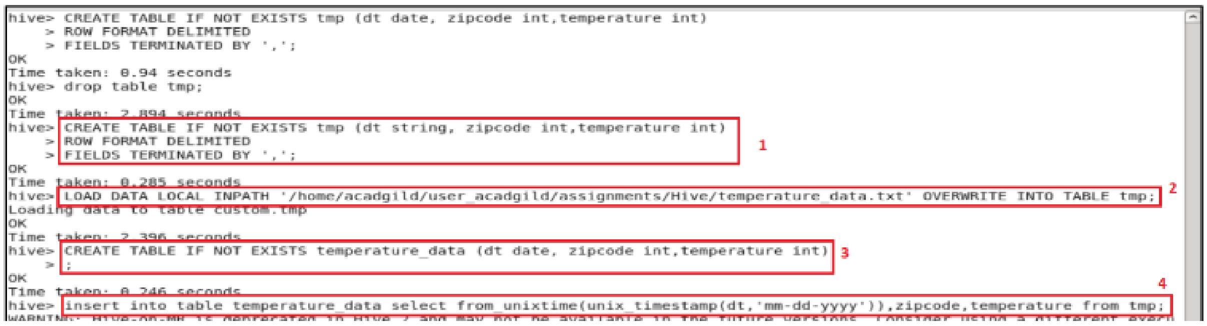
Command :

create database custom;



```
[acadgild@localhost ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-i
/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin
ar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a
ne (i.e. spark, tez) or using Hive 1.X releases.
hive> show databases;
OK
default
Time taken: 9.545 seconds, Fetched: 1 row(s)
hive> create database custom;
OK
Time taken: 0.384 seconds
hive> use custom;
OK
Time taken: 0.116 seconds
```

Table Creation :



```
hive> CREATE TABLE IF NOT EXISTS tmp (dt date, zipcode int,temperature int)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.94 seconds
hive> drop table tmp;
OK
Time taken: 2.894 seconds
hive> CREATE TABLE IF NOT EXISTS tmp (dt string, zipcode int,temperature int)       1
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.285 seconds
hive> LOAD DATA LOCAL INPATH '/home/acadgild/user_acadgild/assignments/Hive/temperature_data.txt' OVERWRITE INTO TABLE tmp;   2
Loading data to table custom.tmp
OK
Time taken: 2.396 seconds
hive> CREATE TABLE IF NOT EXISTS temperature_data (dt date, zipcode int,temperature int)   3
    > ;
OK
Time taken: 0.246 seconds                                                                                        4
hive> insert into table temperature_data select from_unixtime(unix_timestamp(dt,'mm-dd-yyyy')),zipcode,temperature from tmp;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execu
```

With reference to the screenshot above,

1 : To convert the date to 'mm-dd-yyyy' format, a temporary table is used. Initially, the date is loaded from the file as a string.

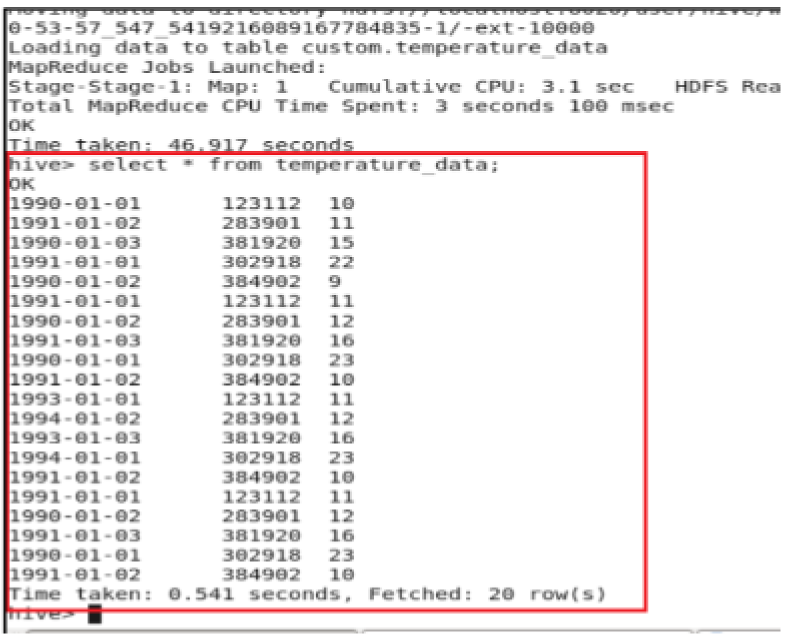
2 : The data from the text file is loaded into the temporary table.

LOAD DATA LOCAL INPATH 'location/of/text/file' OVERWRITE INTO TABLE table_name;

3 : The actual Table 'temperature_data' is created with the column data type as 'date'.

4 : Now the date format transformation is applied on the date in string format in temporary table and inserted into the 'temperature_data' table.

Insert into table temperature_data select ***from_unixtime(unix_time(dt,'mm-dd-yyyy')),*** *zipcode, temperature from tmp;*

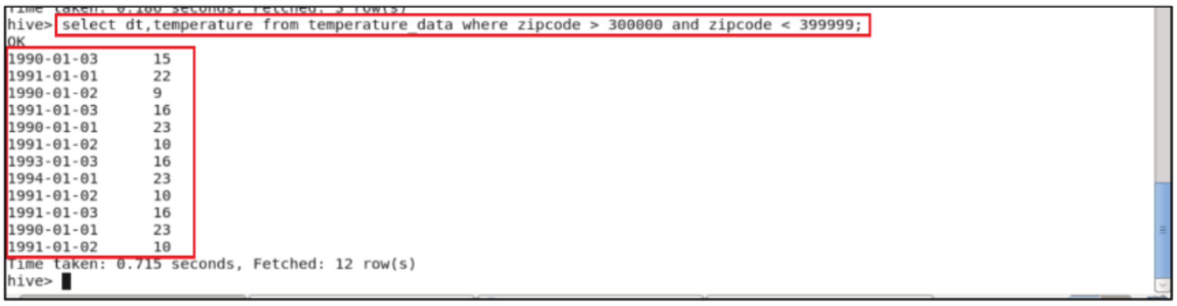


```
0-53-57_547_5419216089167784835-1/-ext-10000
Loading data to table custom.temperature_data
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.1 sec   HDFS Rea
Total MapReduce CPU Time Spent: 3 seconds 100 msec
OK
Time taken: 46.917 seconds
hive> select * from temperature_data;
OK
1990-01-01      123112  10
1991-01-02      283901  11
1990-01-03      381920  15
1991-01-01      302918  22
1990-01-02      384902  9
1991-01-01      123112  11
1990-01-02      283901  12
1991-01-03      381920  16
1990-01-01      302918  23
1991-01-02      384902  10
1993-01-01      123112  11
1994-01-02      283901  12
1993-01-03      381920  16
1994-01-01      302918  23
1991-01-02      384902  10
1991-01-01      123112  11
1990-01-02      283901  12
1991-01-03      381920  16
1990-01-01      302918  23
1991-01-02      384902  10
Time taken: 0.541 seconds, Fetched: 20 row(s)
hive>
```

Data is loaded and on performing a select on the table 'temperature_data', the data in the table can be seen.

TASK 2:

1. Fetch date and temperature from temperature_data where zip code is greater than 300000 and less than 399999.

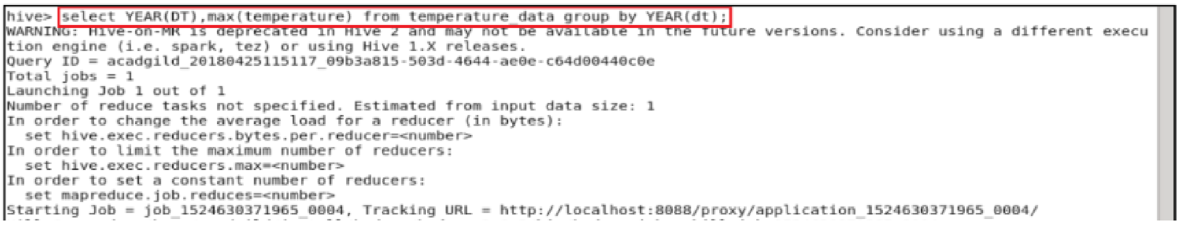


```
hive> select dt,temperature from temperature_data where zipcode > 300000 and zipcode < 399999;
OK
1990-01-03      15
1991-01-01      22
1990-01-02      9
1991-01-03      16
1990-01-01      23
1991-01-02      10
1993-01-03      16
1994-01-01      23
1991-01-02      10
1991-01-03      16
1990-01-01      23
1991-01-02      10
Time taken: 0.715 seconds, Fetched: 12 row(s)
hive> █
```

HQL statement :

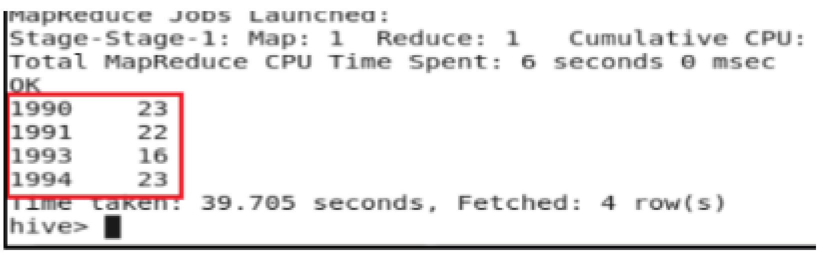
select dt, temperature from temperature_data where zipcode > 300000 and zipcode < 399999;

2. Calculate maximum temperature corresponding to every year from temperature_data table.



```
hive> select YEAR(DT),max(temperature) from temperature_data group by YEAR(dt);
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execu
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180425115117_09b3a815-503d-4644-ae0e-c64d00440c0e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1524630371965_0004, Tracking URL = http://localhost:8088/proxy/application_1524630371965_0004/
```

Output:



```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU:
Total MapReduce CPU Time Spent: 6 seconds 0 msec
OK
1990     23
1991     22
1993     16
1994     23
Time taken: 39.705 seconds, Fetched: 4 row(s)
hive> █
```

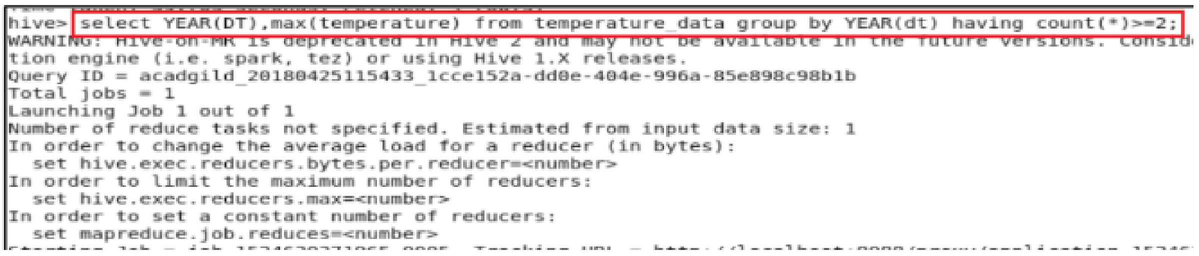
HQL statement :

select YEAR(dt), max(temperature) from temperature_data group by YEAR(dt);

YEAR(dt) : to extract the 'yyyy' part from the timestamp/date.

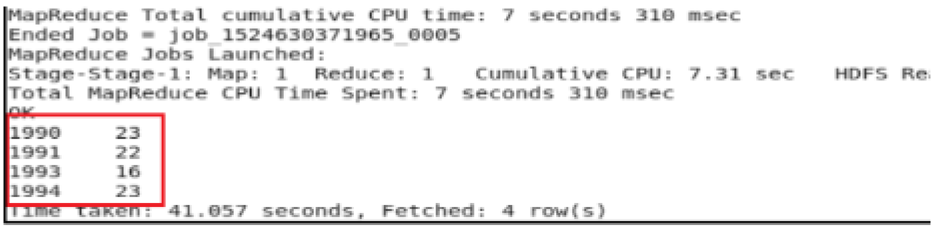
Group by : used to group together and perform max on each group.

3. Calculate maximum temperature from temperature_data table corresponding to those years which have at least 2 entries in the table.



```
hive> select YEAR(DT),max(temperature) from temperature_data group by YEAR(dt) having count(*)>=2;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consid
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180425115433_1cce152a-dd0e-404e-996a-85e898c98b1b
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
```

Output:



```
MapReduce Total cumulative CPU time: 7 seconds 310 msec
Ended Job = job_1524630371965_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.31 sec   HDFS Re
Total MapReduce CPU Time Spent: 7 seconds 310 msec
OK
1990    23
1991    22
1993    16
1994    23
Time taken: 41.057 seconds, Fetched: 4 row(s)
```

HQL statement :

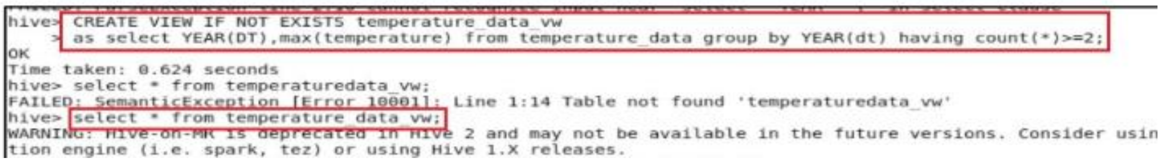
select YEAR(dt), max(temperature) from temperature_data group by YEAR(dt) having count(*) >= 2;

YEAR(dt) : to extract the 'yyyy' part from the timestamp/date.

group by : used to group together and perform max on each group.

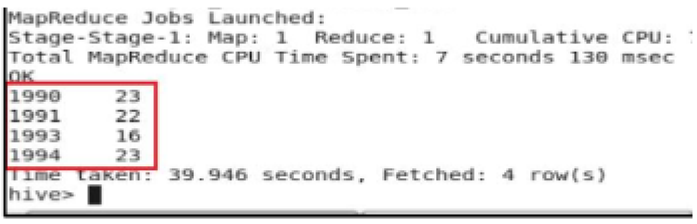
having count(*) >=2 : to filter in the groups that have 2 or more values in it.

4. Create a view on the top of last query, name it temperature_data_vw.



```
hive> CREATE VIEW IF NOT EXISTS temperature_data_vw
    > as select YEAR(DT),max(temperature) from temperature_data group by YEAR(dt) having count(*)>=2;
OK
Time taken: 0.624 seconds
hive> select * from temperaturedata_vw;
FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'temperaturedata_vw'
hive> select * from temperature_data_vw;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider usin
tion engine (i.e. spark, tez) or using Hive 1.X releases.
```

Output:



```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU:
Total MapReduce CPU Time Spent: 7 seconds 130 msec
OK
1990     23
1991     22
1993     16
1994     23
Time taken: 39.946 seconds, Fetched: 4 row(s)
hive>
```

HQL Statement :

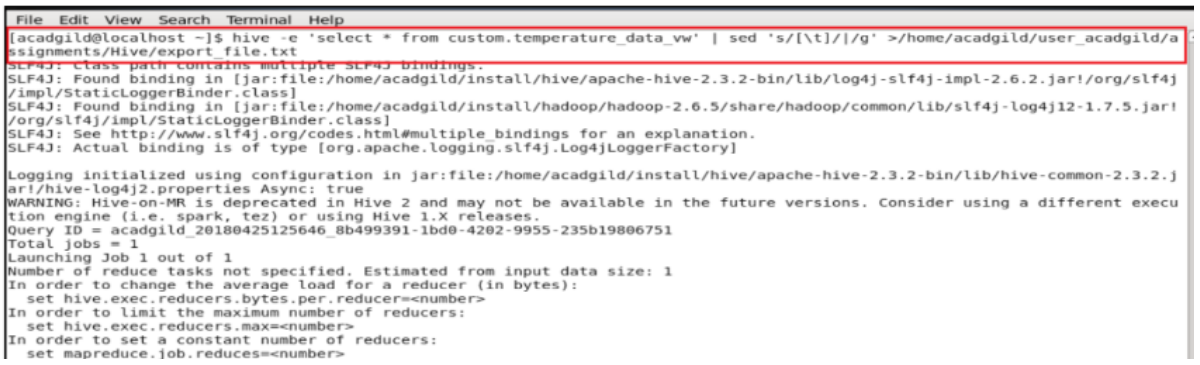
CREATE VIEW IF NOT EXISTS temperature_data_vw as

select YEAR(dt), max(temperature) from temperature_data group by YEAR(dt) having count(*) >= 2;

View named 'temperature_data_vw' is created.

When queried, the data can be seen in the view.

5. Export contents from temperature_data_vw to a file in local file system, such that each file is '|' delimited.



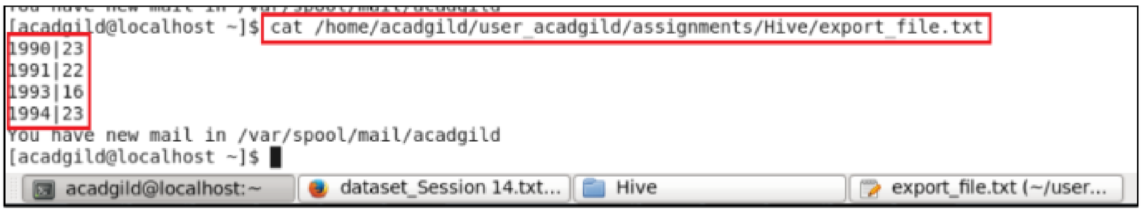
```
[acadgild@localhost ~]$ hive -e 'select * from custom.temperature_data_vw' | sed 's/[\t]/|/g' >/home/acadgild/user_acadgild/a
ssignments/Hive/export_file.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.j
ar!/hive-log4j2.properties Async: true
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execu
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180425125646_8b499391-1bd0-4202-9955-235b19806751
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
```

HQL Statement :

*hive -e 'select * from custom.temperature_data_vw' | sed 's/[\t]/|/g' >/home/acadgild/assignments/Hive/export_file.txt*

The query select * from custom.temperature_data_vw is run in hive and the resultant fields are seperated by '|' and the final result is put into the file 'export_file.txt'

The query select * from custom.temperature_data_vw is run in hive and the resultant fields are seperated by '|' and the final result is put into the file 'export_file.txt'



```
[acadgild@localhost ~]$ cat /home/acadgild/user_acadgild/assignments/Hive/export_file.txt
1990|23
1991|22
1993|16
1994|23
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$
```

When a cat is run on the exported file, |-separated values can be seen.