

### Importing Data to Hive

1. Connecting to existing MySQL Database

```
mysql --user=retail_dba --password=cloudera retail_db
```

2. Show all the available tables

```
show tables;
```

3. View/Count data from a table in MySQL

```
select * from categories;
```

4. Open Hive shell

```
hive
```

5. Show all available databases inside hive

```
SHOW DATABASES;
```

6. Show all available tables

```
SHOW TABLES;
```

7. Describe Hive table

```
describe sample_07;
```

8. Select data from hive table

```
select * from sample_07;
```

<code>--hive-home &lt;dir&gt;</code>	Override <code>\$HIVE_HOME</code>
<code>--hive-import</code>	Import tables into Hive (Uses Hive's default delimiters if none are set.)
<code>--hive-overwrite</code>	Overwrite existing data in the Hive table.
<code>--create-hive-table</code>	If set, then the job will fail if the target hive table exists. By default this property is false.
<code>--hive-table &lt;table-name&gt;</code>	Sets the table name to use when importing to Hive.
<code>--hive-drop-import-delims</code>	Drops <code>\n</code> , <code>\r</code> , and <code>\01</code> from string fields when importing to Hive.
<code>--hive-delims-replacement</code>	Replace <code>\n</code> , <code>\r</code> , and <code>\01</code> from string fields with user defined string when importing to Hive.
<code>--hive-partition-key</code>	Name of a hive field to partition are sharded on
<code>--hive-partition-value &lt;v&gt;</code>	String-value that serves as partition key for this imported into hive in this job.
<code>--map-column-hive &lt;map&gt;</code>	Override default mapping from SQL type to Hive type for configured columns.

9. Import Single table (Subset data) to hive table.

```
sqoop import \  
  --connect jdbc:mysql://quickstart:3306/retail_db \  
  --username=retail_dba \  
  --password=cloudera \  
  --table=categories \  
  --where "\`category_id\` between 1 and 22" \  
  --hive-import
```

**Note: Here the ` is the same you find on ~ key**

**This command will create a managed table and content will be created in the following directory.**

**/hive/warehouse/categories**

10. Check whether table is created or not.

```
show tables;
```

11. Check the data inside the table.

```
select * from categories;
```

12. Describe table

```
describe categories;
```

13. Run step 9 again and see what happen. It will be successful and data will be appended.

14. Overwrite existing data

```
sqoop import \  
  --connect jdbc:mysql://quickstart:3306/retail_db \  
  --username=retail_dba \  
  --password=cloudera \  
  --table=categories \  
  --where "\`category_id\` between 1 and 22" \  
  --hive-import \  
  --hive-overwrite
```

15. Check the data in hive table.

```
select * from categories;
```

16. Create Hive table with partition

```
CREATE TABLE CATEGORIES_PART (category_id int, category_name string)PARTITIONED BY  
(category_department_id int);
```

17. Now check the table metadata

```
describe categories_part;
```

Check the content

```
Select * from categories_part;
```

18. Create Hive Partitioned table.

**Note: Partition key category\_department\_id cannot be a column to import**

```
sqoop import \  
  --connect jdbc:mysql://quickstart:3306/retail_db \  
  --username=retail_dba \  
  --password=cloudera \  
  --table=categories \  
  --hive-import \  
  --hive-table categories_part \  
  --columns=category_id,category_name \  
  --where `category_department_id`=8 \  
  --hive-overwrite \  
  --hive-partition-key category_department_id \  
  --hive-partition-value 8
```

19. Check the directory structure, how it is created.

```
hdfs://quickstart.cloudera:8020/user/hive/warehouse/categories_part/category_department_id=8/part-m-00000  
hdfs://quickstart.cloudera:8020/user/hive/warehouse/categories_part/category_department_id=8/part-m-00001  
hdfs://quickstart.cloudera:8020/user/hive/warehouse/categories_part/category_department_id=8/part-m-00002  
hdfs://quickstart.cloudera:8020/user/hive/warehouse/categories_part/category_department_id=8/part-m-00003
```

20. Check the inserted data

```
Select * from categories_part;
```

### Replacing Special Delimiters during Hive Import

21. Insert a row in MySQL table

```
mysql --user=retail_dba --password=cloudera retail_db  
INSERT INTO categories values (59,8,"Hell \n Test");  
select * from categories;
```

22. Start Hive Import

```
sqoop import \  
  --connect jdbc:mysql://quickstart:3306/retail_db \  
  --username=retail_dba \  
  --password=cloudera \  
  --table=categories \  
  --hive-import \  
  --hive-table categories_part \  
  --columns=category_id,category_name \  
  --hive-partition-key category_department_id \  
  --hive-partition-value 8
```

```
--where `category_department_id`=8 \  
--hive-overwrite \  
--hive-partition-key category_department_id \  
--hive-partition-value 8
```

23. Now check the inserted data in Hive table

```
Select * from categories_part;
```

This \n is creating a problem.

24. Avoiding this issue

```
sqoop import \  
  --connect jdbc:mysql://quickstart:3306/retail_db \  
  --username=retail_dba \  
  --password=cloudera \  
  --table=categories \  
  --hive-import \  
  --hive-table categories_part \  
  --columns=category_id,category_name \  
  --where `category_department_id`=8 \  
  --hive-overwrite \  
  --hive-partition-key category_department_id \  
  --hive-partition-value 8 \  
  --hive-drop-import-delims
```

25. Now check the data in hive table

```
Select * from categories_part;
```

**Try your own, to just replace this character with some other value using `--hive-delims-replacement`**

### Correct NULL string in Hive

26. Insert following rows in the mysql table.

```
ALTER TABLE categories modify category_name varchar(45);  
INSERT INTO categories values (60,8,NULL);  
INSERT INTO categories values (61,8,null);  
Commit;
```

27. Now import the data

```
sqoop import \  
  --connect jdbc:mysql://quickstart:3306/retail_db \  
  --username=retail_dba \  
  --password=cloudera \  
  --table=categories \  
  --hive-import
```

```
--hive-import \  
--hive-table categories_part \  
--columns=category_id,category_name \  
--where `category_department_id`=8 \  
--hive-overwrite \  
--hive-partition-key category_department_id \  
--hive-partition-value 8 \  
--hive-drop-import-delims
```

28. Replace the null string with other value.

```
sqoop import \  
--connect jdbc:mysql://quickstart:3306/retail_db \  
--username=retail_dba \  
--password=cloudera \  
--table=categories \  
--hive-import \  
--hive-table categories_part \  
--columns=category_id,category_name \  
--where `category_department_id`=8 \  
--hive-overwrite \  
--hive-partition-key category_department_id \  
--hive-partition-value 8 \  
--hive-drop-import-delims \  
--null-string=""\N'
```

29. Now check the data in Hive table

```
Select * from categories_part;
```

30. Suggested exercise, please do exercise using following option as well.

```
--create-hive-table  
--hive-delims-replacement  
--map-column-hive
```

31. Cleanup

```
delete from categories where category_id in (59,60,61);
```

```
ALTER TABLE categories modify category_department_id int(11) NOT NULL;  
ALTER TABLE categories modify category_name varchar(45) NOT NULL;  
desc categories;
```

```
drop table categories;  
drop table categories_part;
```

 <b>430 Q &amp; A</b> <a href="#">Click Here</a> <b>Cloudera Hadoop Developer Certification</b> <b>CCD-410</b>	 <b>200 + Q &amp; A</b> <a href="#">Click Here</a> <b>Cloudera Hadoop Administrator Certification</b> <b>CCA-500</b>	 <b>262 Q &amp; A</b> <a href="#">Click Here</a> <b>Cloudera Hadoop HBase Certification</b> <b>CCB-400</b>	 <b>266 Q &amp; A</b> <a href="#">Click Here</a> <b>AWS Developer Certification Associate Level</b>
 <b>235 Q &amp; A</b> <a href="#">Click Here</a> <b>Cloudera Data Science Certification</b> <b>DS-200</b>	 <a href="#">Click Here</a> <b>Hadoop Training With HandsOn</b>	 <a href="#">Click Here</a> <b>Package Deal</b>	 <b>144 Q &amp; A</b> <a href="#">Click Here</a> <b>AWS Certified Solutions Architect Professional Level</b>

 <b>300 + Q &amp; A</b> <a href="#">Click Here</a> <b>AWS Certified SysOps Administrator Associate Level</b>	 <b>474 + Q &amp; A</b> <a href="#">Click Here</a> <b>AWS Certified Solutions Architect Associate Level</b>	 <a href="#">Click Here</a> <b>Click Here for AWS Package Deal</b>	 <b>490 Q &amp; A</b> <a href="#">Click Here</a> <b>SAS Base Certification A00-211</b>
 <b>365 Q &amp; A</b> <a href="#">Click Here</a> <b>SAS Advance Certification A00-212</b>	 <b>86+ Q &amp; A</b> <a href="#">Click Here</a> <b>SAS Certified Statistical Business Analyst A00-240</b>	 <b>85 Q &amp; A</b> <a href="#">Click Here</a> <b>SAS Certified Platform Administrator 9 A00-250</b>	 <a href="#">Click Here</a> <b>SAS Packaged Deal</b>



234 Q & A

[Click Here](#)

EMC Data Scientist  
Associate  
Certification  
E20-007 (EMCDSA)

**Data Science certification really needs a good and in depth knowledge of statistics cum BigData Hadoop knowledge.** It also require you to have

good knowledge in like the main phases of the Data Analytics Lifecycle, analyzing and exploring data with R, statistics for model building and evaluation, the theory and methods of advanced analytics and statistical modeling, the technology and tools that can be used for advanced analytics, operationalizing an analytics project, and data visualization techniques. Successful candidates will achieve the EMC Proven Professional – Data Science Associate credential. Hence to clear the real exam it really needs very well preparation. So HadoopExam Learning Resources brings Data Science Certification Simulator with 234 Practice Questions, which can help

you to prepare for this exam in lesser time. **Practice - practice - practice!** The EMC:DS E20-007 Exam Simulator offers you the opportunity to take 4 sample Exams before heading out for the real thing. Be ready to succeed on exam day!

### [Upcoming Releases](#)

1. [Apache Spark Training](#)
2. [Apache Spark Certification material](#)
3. [MongoDB Certification Material](#)
4. [Android Certification](#)
5. [Java Certification](#)
6. [AWS Trainings](#)
7. [Data Science Training](#)