



# Hadoop File Formats Practical

...

# IMPORTANT

## **Copyright Infringement and Illegal Content Sharing Notice**

All course content designs, video, audio, text, graphics, logos, images are Copyright© and are protected by India and international copyright laws. All rights reserved.

Permission to download the contents (wherever applicable) for the sole purpose of individual reading and preparing yourself to crack the interview only. Any other use of study materials – including reproduction, modification, distribution, republishing, transmission, display – without the prior written permission of Author is strictly prohibited.

**Trendytech Insights** legal team, along with thousands of our students, actively searches the Internet for copyright infringements. Violators subject to prosecution.



# Orc File Format

Create a table with orc file format named “orders\_orc”

```
CREATE TABLE orders_orc(  
  id bigint,  
  product_id string,  
  customer_id bigint,  
  quantity int,  
  amount double) stored as orc;
```

```
hive> CREATE TABLE orders_orc(  
  >   id bigint,  
  >   product_id string,  
  >   customer_id bigint,  
  >   quantity int,  
  >   amount double) stored as orc;  
OK  
Time taken: 0.82 seconds  
hive> █
```



# Orc File Format

Now insert the data in this table from orders table

**insert into orders\_orc select \* from orders;**

```
hive> insert into orders_orc select * from orders;
Query ID = cloudera_20200507031616_8722e716-a7ec-46a1-9e83-ac1b8221618a
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1588458683533_0011, Tracking URL = http://quickstart.cloudera
88/proxy/application_1588458683533_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1588458683533_0011
```



# Orc File Format

Now try to see the data in hdfs

```
hadoop fs -ls /user/hive/warehouse/trendytech.db/orders_orc/000000_0
```

```
hadoop fs -cat /user/hive/warehouse/trendytech.db/orders_orc/*
```

```
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/trendytech.db/orders_orc
Found 1 items
-rwxrwxrwx   1 cloudera supergroup          645 2020-05-07 03:29 /user/hive/warehouse/trendytech.db/orders_orc/000000_0
[cloudera@quickstart ~]$
```



# Orc File Format

```
[cloudera@quickstart ~]$ hadoop fs -cat /user/hive/warehouse/trendytech.db/orders_orc/*
ORC
P7
CPC
broomt-shirtBP3
P+
PBbb`FVW/broomcameraphonet-shir^"
Fbsoc`CtCM tC -0100
```

```
broomt-shirtBP3
P+
PBbb`FVW/broomcameraphonet-shir^"
Fbsoc`CtCM tC -0100
4IK(
`C`C;0+0C@lnX0V%!.C\!C000*0b0`
60P5C:A0C00Ex0"lLlC0C04'0000-0`F0N0R
0Q"A0K0UI00G000000U0S>9700B(e
0L000X0003000
k00C0 ^0i'x'n0p0Ip0* q0S00
000gd0HXe00080
0p00700000`0000090000A0llq003V:00L00000000
(n000ORC cloudera@quickstart
```



# Orc File Format

To get information about an ORC file, use the `orcfiledump` command  
`hive --orcfiledump /user/hive/warehouse/trendytech.db/orders_orc/000000_0`

```
[cloudera@quickstart ~]$ hive --orcfiledump /user/hive/warehouse/trendytech.db/orders_orc/000000_0
Structure for /user/hive/warehouse/trendytech.db/orders_orc/000000_0
File Version: 0.12 with HIVE_8732
20/05/07 03:38:54 INFO orc.ReaderImpl: Reading ORC rows from /user/hive/warehouse/trendytech.db/orders_orc/000000_0 with {include: null, offset: 0, length: 9223372036854775807}
Rows: 5
Compression: ZLIB
Compression size: 262144
Type: struct<_col0:bigint,_col1:string,_col2:bigint,_col3:int,_col4:double>

Stripe Statistics:
Stripe 1:
  Column 0: count: 5 hasNull: false
  Column 1: count: 5 hasNull: false min: 111111 max: 111115 sum: 555565
  Column 2: count: 5 hasNull: false min: broom max: t-shirt sum: 28
  Column 3: count: 5 hasNull: false min: 1111 max: 4444 sum: 9999
  Column 4: count: 5 hasNull: false min: 1 max: 3 sum: 9
  Column 5: count: 5 hasNull: false min: 10.0 max: 5200.0 sum: 6496.0

File Statistics:
  Column 0: count: 5 hasNull: false
```



# Orc File Format

To display the data in the ORC file, use

```
hive --orcfiledump -d /user/hive/warehouse/trendytech.db/orders_orc/000000_0
```

```
[cloudera@quickstart ~]$ hive --orcfiledump -d /user/hive/warehouse/trendytech.db/orders_orc/000000_0
20/05/07 03:47:20 INFO orc.ReaderImpl: Reading ORC rows from /user/hive/warehouse/trendytech.db/order
s_orc/000000_0 with {include: null, offset: 0, length: 9223372036854775807}
{"_col0":111111,"_col1":"phone","_col2":1111,"_col3":3,"_col4":1200}
{"_col0":111112,"_col1":"camera","_col2":1111,"_col3":1,"_col4":5200}
{"_col0":111113,"_col1":"broom","_col2":1111,"_col3":1,"_col4":10}
{"_col0":111114,"_col1":"broom","_col2":2222,"_col3":2,"_col4":20}
{"_col0":111115,"_col1":"t-shirt","_col2":4444,"_col3":2,"_col4":66}
[cloudera@quickstart ~]$
```





# Parquet File Format

Create a table with parquet file format named “orders\_parquet”

```
CREATE TABLE orders_parquet(  
  id bigint,  
  product_id string,  
  customer_id bigint,  
  quantity int,  
  amount double) stored as parquet;
```

```
insert into orders_parquet select * from orders;
```

```
hive>  
hive> CREATE TABLE orders_parquet(  
  > id bigint,  
  > product_id string,  
  > customer_id bigint,  
  > quantity int,  
  > amount double) stored as parquet;  
OK  
Time taken: 0.415 seconds  
hive> insert into orders_parquet select * from orders;  
Query ID = cloudera_20200507040000_e1c710a0-8422-42b1-a531-ab465c945739  
Total jobs = 3
```



# Parquet File Format

Now try to see the data in hdfs

```
hadoop fs -cat /user/hive/warehouse/trendytech.db/orders_parquet/*
```

```
[cloudera@quickstart ~]$ hadoop fs -cat /user/hive/warehouse/trendytech.db/orders_parquet/000000_0
PAR1
phone
camerabroomt-shirt
-shirtbroom
w
Pese400Pa
hive_schema
product_id
customer_id
quantity
amount
id
```



# Parquet File Format

Now get the data from hdfs to local using get command.

```
hadoop fs -get /user/hive/warehouse/trendytech.db/orders_parquet/000000_0 .
```

Now try to see the metadata using below command.

```
parquet-tools meta 000000_0
```

```
[cloudera@quickstart ~]$ parquet-tools meta 000000_0
creator:      parquet-mr version 1.5.0-cdh5.13.0 (build ${buildNumber})
file schema:  hive_schema
=====
id:           OPTIONAL INT64 R:0 D:1
product_id:   OPTIONAL BINARY O:UTF8 R:0 D:1
customer_id:  OPTIONAL INT64 R:0 D:1
quantity:     OPTIONAL INT32 R:0 D:1
amount:       OPTIONAL DOUBLE R:0 D:1

row group 1:  RC:5 TS:429
=====
id:           INT64 UNCOMPRESSED DO:0 FPO:4 SZ:87/87/1.00 VC:5 ENC:RLE,PLAIN,BIT_PACKED
product_id:   BINARY UNCOMPRESSED DO:0 FPO:91 SZ:99/99/1.00 VC:5 ENC:RLE,PLAIN DICTIONARY,BIT_PACKED
customer_id:  INT64 UNCOMPRESSED DO:0 FPO:190 SZ:88/88/1.00 VC:5 ENC:RLE,PLAIN DICTIONARY,BIT_PACKED
quantity:     INT32 UNCOMPRESSED DO:0 FPO:278 SZ:68/68/1.00 VC:5 ENC:RLE,PLAIN DICTIONARY,BIT_PACKED
amount:       DOUBLE UNCOMPRESSED DO:0 FPO:346 SZ:87/87/1.00 VC:5 ENC:RLE,PLAIN,BIT_PACKED
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ █
```



# Parquet File Format

Now try to see the data using below command.

**parquet-tools cat 000000\_0**

```
[cloudera@quickstart ~]$ parquet-tools cat 000000_0
id = 111111
product_id = phone
customer_id = 1111
quantity = 3
amount = 1200.0

id = 111112
product_id = camera
customer_id = 1111
quantity = 1
amount = 5200.0

id = 111113
product_id = broom
customer_id = 1111
quantity = 1
amount = 10.0
```



# Json Serde

**Ser + de**

**Serialization + Deserialization**

**There is no support for Json by default. So we need to add a jar and add that in hive.**

**Download the jar.**

**[www.congiu.net/hive-json-serde/1.3.7/cdh5/json-serde-1.3.7-jar-with-dependencies.jar](http://www.congiu.net/hive-json-serde/1.3.7/cdh5/json-serde-1.3.7-jar-with-dependencies.jar)**



# Json Serde

Index of /hive-json-serd... x +

← | www.congiu.net/hive-json-serde/1.3.7/cdh5/

## Index of /hive-json-serde/1.3.7/cdh5

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>		-	
<a href="#">json-serde-1.3.7-jar...&gt;</a>	2016-01-30 20:28	80K	
<a href="#">json-udf-1.3.7-jar-w...&gt;</a>	2016-01-30 20:28	83K	



# Json Serde

**Now add the jar in hive using the below command as show.**

```
hive> add jar /home/cloudera/Downloads/json-serde-1.3.7-jar-with-dependencies.jar;  
Added [/home/cloudera/Downloads/json-serde-1.3.7-jar-with-dependencies.jar] to class path  
Added resources: [/home/cloudera/Downloads/json-serde-1.3.7-jar-with-dependencies.jar]  
hive> █
```



# Json Serde

Create a table using json serde

```
CREATE TABLE orders_json(  
  id bigint,  
  product_id string,  
  customer_id bigint,  
  quantity int,  
  amount double) ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe';
```

```
hive> CREATE TABLE orders_json(  
  > id bigint,  
  > product_id string,  
  > customer_id bigint,  
  > quantity int,  
  > amount double) ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe';  
OK  
Time taken: 0.24 seconds  
hive> █
```





# Json Serde

Now insert the data in this table from orders table

**insert overwrite table orders\_json select \* from orders;**

```
hive> insert overwrite table orders_json select * from orders;
Query ID = cloudera_20200507044141_4adaa906-a1bc-4620-9b5e-b1c98f19d44f
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1588847091075_0003, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1588847091075_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1588847091075_0003
```



# Json Serde

Now try to see the data in hdfs

**hadoop fs -cat /user/hive/warehouse/trendytech.db/orders\_json/\***

```
[cloudera@quickstart ~]$ hadoop fs -cat /user/hive/warehouse/trendytech.db/orders_json/*
{"amount":1200,"id":111111,"product_id":"phone","quantity":3,"customer_id":1111}
{"amount":5200,"id":111112,"product_id":"camera","quantity":1,"customer_id":1111}
{"amount":10,"id":111113,"product_id":"broom","quantity":1,"customer_id":1111}
{"amount":20,"id":111114,"product_id":"broom","quantity":2,"customer_id":2222}
{"amount":66,"id":111115,"product_id":"t-shirt","quantity":2,"customer_id":4444}
[cloudera@quickstart ~]$
```



# Json Serde

To see all details related to table run the below command

**show create table orders\_json;**

```
CREATE TABLE `orders_json`(  
  `id` bigint COMMENT 'from deserializer',  
  `product_id` string COMMENT 'from deserializer',  
  `customer_id` bigint COMMENT 'from deserializer',  
  `quantity` int COMMENT 'from deserializer',  
  `amount` double COMMENT 'from deserializer')  
ROW FORMAT SERDE  
  'org.openx.data.jsonserde.JsonSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.mapred.TextInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat'  
LOCATION  
  'hdfs://quickstart.cloudera:8020/user/hive/warehouse/trendytech.db/orders_json'  
TBLPROPERTIES (  
  'COLUMN_STATS_ACCURATE'='true',  
  'numFiles'='1',  
  'numRows'='5',  
  'rawDataSize'='0',  
  'totalSize'='402',  
  'transient lastDdlTime'='158851707')  
Time taken: 0.043 seconds, Fetched: 21 row(s)
```



**We have seen hadoop file formats practically**

**Happy Learning!!!**



**5** Star Google Rated  
Big Data Course

**LEARN FROM THE EXPERT**



**9108179578**

**Call for more details**



# Follow US

**Trainer** Mr. Sumit Mittal

**Phone** 9108179578

**Email** trendytech.sumit@gmail.com

**Website** <https://trendytech.in/courses/big-data-online-training/>

**LinkedIn** <https://www.linkedin.com/in/bigdatabysumit/>

**Twitter** @BigdataBySumit

**Instagram** bigdatabysumit

**Facebook** <https://www.facebook.com/trendytech.in/>

**Youtube** [https://www.youtube.com/channel/UCbTggJVf0NDTfWX-C\\_gUGSg](https://www.youtube.com/channel/UCbTggJVf0NDTfWX-C_gUGSg)