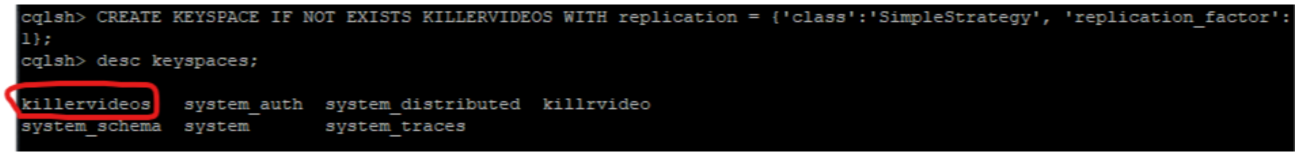
Report.

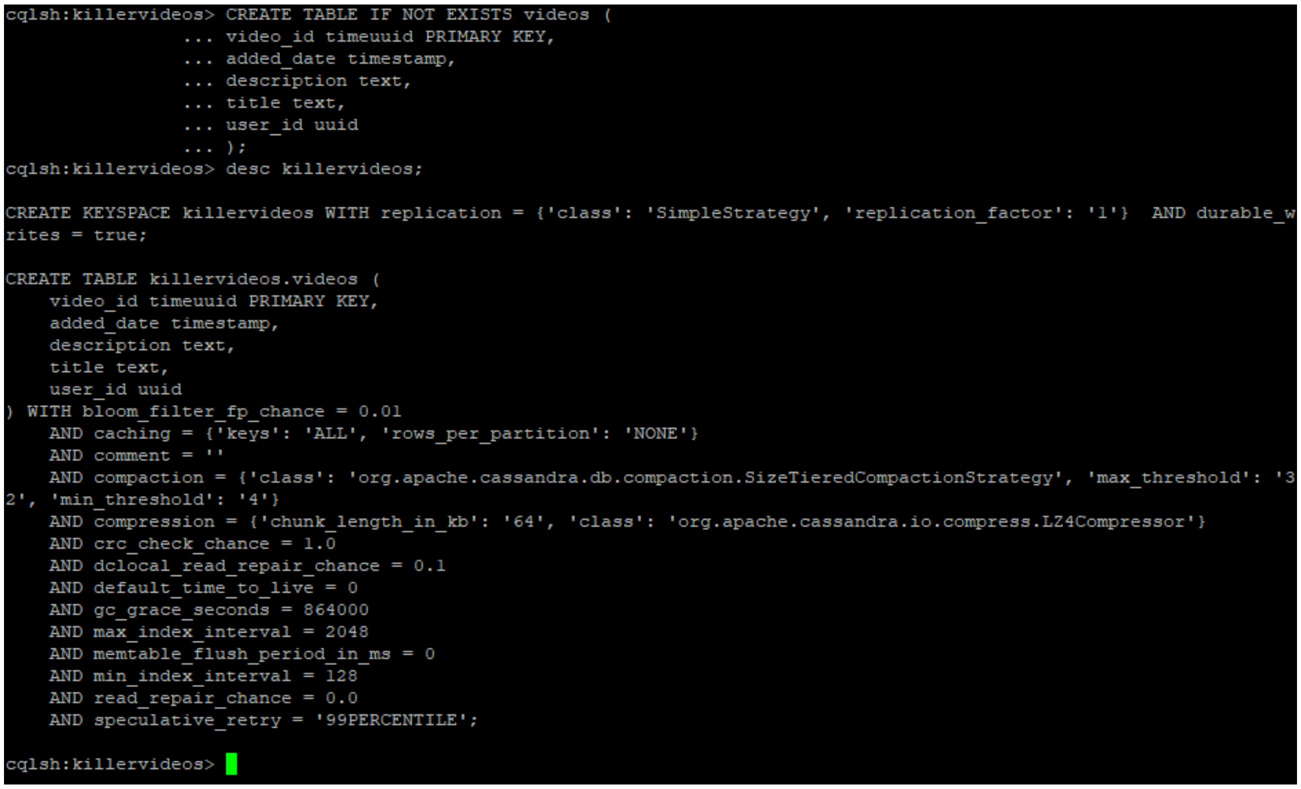
5. In 'cqlsh', create a keyspace called 'killrvideo' and switch to that keyspace. Use 'SimpleStrategy' for the replication class with a replication factor of one. Remember the 'use' command switches keyspaces.
NOTE: You can press the tab key within the CREATE KEYSPACE command to have 'cqlsh' autocomplete the replication parameters.



```
cqlsh> CREATE KEYSPACE IF NOT EXISTS KILLERVIDEOS WITH replication = {'class':'SimpleStrategy', 'replication_factor':
1};
cqlsh> desc keyspaces;

killervideos   system_auth  system_distributed  killrvideo
system_schema  system       system_traces
```

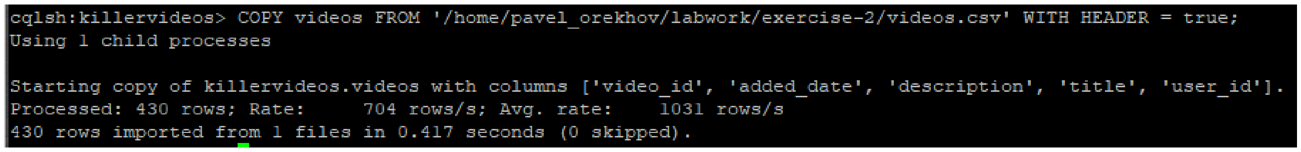
6. Create a single table called 'videos' with the same structure as shown in table above. 'video_id' is the primary key.

We can switch into the **killervideos** keyspace by doing **use killervideos;** at command prompt.



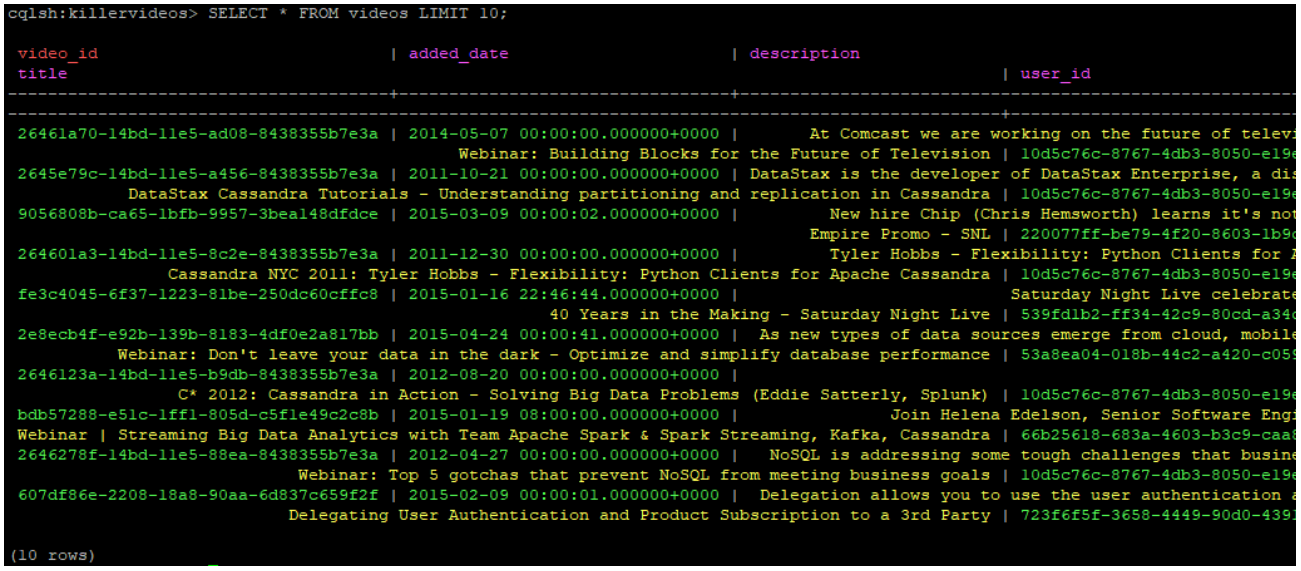
```
cqlsh:killervideos> CREATE TABLE IF NOT EXISTS videos (
                ... video_id timeuuid PRIMARY KEY,
                ... added_date timestamp,
                ... description text,
                ... title text,
                ... user_id uuid
                ... );
cqlsh:killervideos> desc killervideos;

CREATE KEYSPACE killervideos WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}  AND durable_w
rites = true;

CREATE TABLE killervideos.videos (
    video_id timeuuid PRIMARY KEY,
    added_date timestamp,
    description text,
    title text,
    user_id uuid
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '3
2', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';

cqlsh:killervideos>
```

7. Load the newly created table with the 'videos.csv' file using the 'COPY' command.
COPY videos FROM 'videos.csv' WITH HEADER=true;



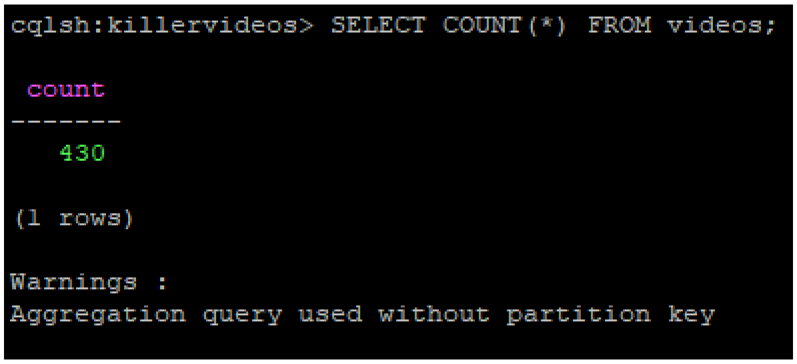
```
cqlsh:killervideos> COPY videos FROM '/home/pavel_orekhov/labwork/exercise-2/videos.csv' WITH HEADER = true;
Using 1 child processes

Starting copy of killervideos.videos with columns ['video_id', 'added_date', 'description', 'title', 'user_id'].
Processed: 430 rows; Rate:     704 rows/s; Avg. rate:    1031 rows/s
430 rows imported from 1 files in 0.417 seconds (0 skipped).
```

8. Use SELECT to verify the data loaded correctly. Include LIMIT to retrieve only the first 10 rows.



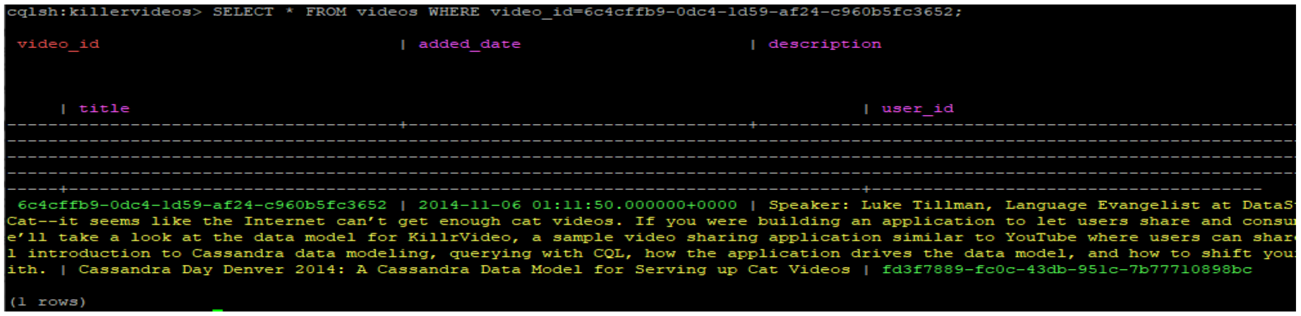
```
cqlsh:killervideos> SELECT * FROM videos LIMIT 10;

 video_id                             | added_date                      | description
 title                                                                                            | user_id
-------------------------------------+---------------------------------+-------------------------------------------------------
-------------------------------------------------------------------------------------------------+----------------------------
 26461a70-14bd-11e5-ad08-8438355b7e3a | 2014-05-07 00:00:00.000000+0000 |       At Comcast we are working on the future of televi
                                            Webinar: Building Blocks for the Future of Television | 10d5c76c-8767-4db3-8050-e19e
 2645e79c-14bd-11e5-a456-8438355b7e3a | 2011-10-21 00:00:00.000000+0000 | DataStax is the developer of DataStax Enterprise, a dis
           DataStax Cassandra Tutorials - Understanding partitioning and replication in Cassandra | 10d5c76c-8767-4db3-8050-e19e
 9056808b-ca65-1bfb-9957-3bea148dfdce | 2015-03-09 00:00:02.000000+0000 |         New hire Chip (Chris Hemsworth) learns it's not
                                                                               Empire Promo - SNL | 220077ff-be79-4f20-8603-1b9c
 264601a3-14bd-11e5-8c2e-8438355b7e3a | 2011-12-30 00:00:00.000000+0000 |         Tyler Hobbs - Flexibility: Python Clients for A
               Cassandra NYC 2011: Tyler Hobbs - Flexibility: Python Clients for Apache Cassandra | 10d5c76c-8767-4db3-8050-e19e
 fe3c4045-6f37-1223-81be-250dc60cffc8 | 2015-01-16 22:46:44.000000+0000 |                           Saturday Night Live celebrate
                                                     40 Years in the Making - Saturday Night Live | 539fd1b2-ff34-42c9-80cd-a34c
 2e8ecb4f-e92b-139b-8183-4df0e2a817bb | 2015-04-24 00:00:41.000000+0000 |  As new types of data sources emerge from cloud, mobile
          Webinar: Don't leave your data in the dark - Optimize and simplify database performance | 53a8ea04-018b-44c2-a420-c059
 2646123a-14bd-11e5-b9db-8438355b7e3a | 2012-08-20 00:00:00.000000+0000 |
                C* 2012: Cassandra in Action - Solving Big Data Problems (Eddie Satterly, Splunk) | 10d5c76c-8767-4db3-8050-e19e
 bdb57288-e51c-1ff1-805d-c5f1e49c2c8b | 2015-01-19 08:00:00.000000+0000 |              Join Helena Edelson, Senior Software Engi
 Webinar | Streaming Big Data Analytics with Team Apache Spark & Spark Streaming, Kafka, Cassandra | 66b25618-683a-4603-b3c9-caa8
 2646278f-14bd-11e5-88ea-8438355b7e3a | 2012-04-27 00:00:00.000000+0000 |   NoSQL is addressing some tough challenges that busine
                            Webinar: Top 5 gotchas that prevent NoSQL from meeting business goals | 10d5c76c-8767-4db3-8050-e19e
 607df86e-2208-18a8-90aa-6d837c659f2f | 2015-02-09 00:00:01.000000+0000 |  Delegation allows you to use the user authentication a
                           Delegating User Authentication and Product Subscription to a 3rd Party | 723f6f5f-3658-4449-90d0-4391

(10 rows)
```

9. Use SELECT to COUNT(*) the number of imported rows. It should match the number of rows COPY reported as imported.



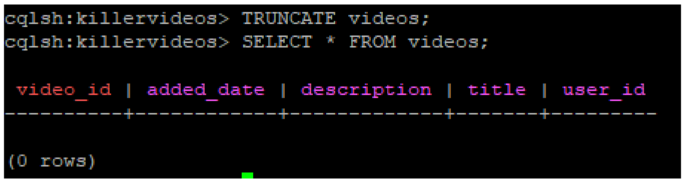
```
cqlsh:killervideos> SELECT COUNT(*) FROM videos;

 count
-------
   430

(1 rows)

Warnings :
Aggregation query used without partition key
```

10. Use SELECT to find a row where the video_id = 6c4cffb9-0dc4-1d59-af24-c960b5fc3652. Next we will explore some other CQL commands that will come in handy, like TRUNCATE in a later exercise, we will show you how to add/remove (non-primary key) columns.



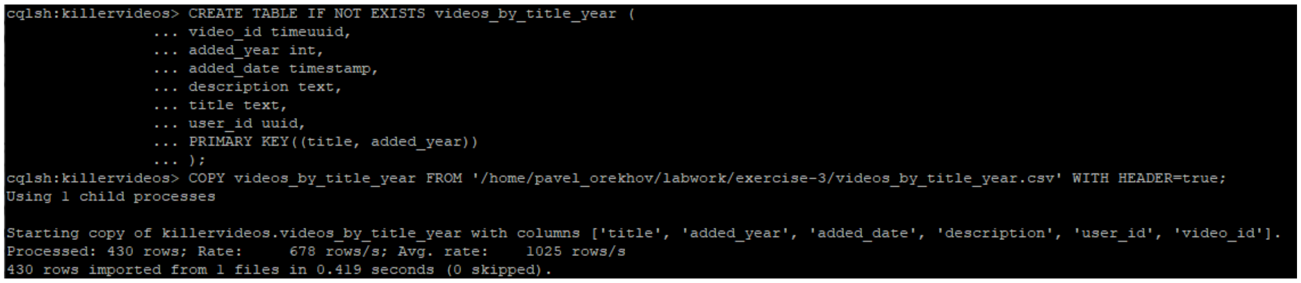
```
cqlsh:killervideos> SELECT * FROM videos WHERE video_id=6c4cffb9-0dc4-1d59-af24-c960b5fc3652;

 video_id                             | added_date                        | description


     | title                                                                           | user_id
--------------------------------------+-----------------------------------+-------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
-----+-----------------------------------------------------------------------------+-------------------------------------
 6c4cffb9-0dc4-1d59-af24-c960b5fc3652 | 2014-11-06 01:11:50.000000+0000 | Speaker: Luke Tillman, Language Evangelist at DataS
Cat--it seems like the Internet can’t get enough cat videos. If you were building an application to let users share and consu
e’ll take a look at the data model for KillrVideo, a sample video sharing application similar to YouTube where users can shar
l introduction to Cassandra data modeling, querying with CQL, how the application drives the data model, and how to shift you
ith. | Cassandra Day Denver 2014: A Cassandra Data Model for Serving up Cat Videos | fd3f7889-fc0c-43db-951c-7b77710898bc

(1 rows)
```

11. Let’s remove the data from our table using TRUNCATE .truncate videos;



```
cqlsh:killervideos> TRUNCATE videos;
cqlsh:killervideos> SELECT * FROM videos;

 video_id | added_date | description | title | user_id
----------+------------+-------------+-------+---------

(0 rows)
```

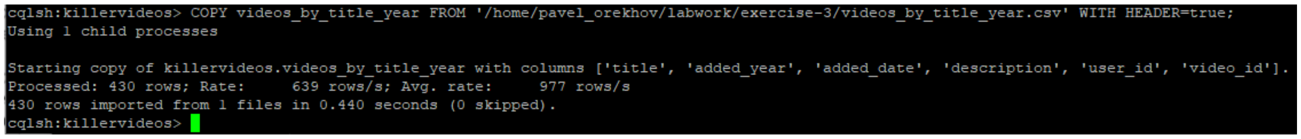
12. Create a second table in the 'killrvideo' keyspace called 'videos_by_title_year' with the structure shown in above table.



```
cqlsh:killervideos> CREATE TABLE IF NOT EXISTS videos_by_title_year (
                ... video_id timeuuid,
                ... added_year int,
                ... added_date timestamp,
                ... description text,
                ... title text,
                ... user_id uuid,
                ... PRIMARY KEY((title, added_year))
                ... );
cqlsh:killervideos> COPY videos_by_title_year FROM '/home/pavel_orekhov/labwork/exercise-3/videos_by_title_year.csv' WITH HEADER=true;
Using 1 child processes

Starting copy of killervideos.videos_by_title_year with columns ['title', 'added_year', 'added_date', 'description', 'user_id', 'video_id'].
Processed: 430 rows; Rate:     678 rows/s; Avg. rate:    1025 rows/s
430 rows imported from 1 files in 0.419 seconds (0 skipped).
```

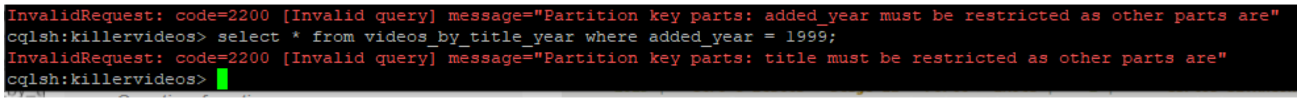
An extra set of parentheses is necessary, to combine **title** and **added_year** into 1 column.

13. Load the data from the 'videos_by_title_year.csv' file using the `COPY` command.
14. COPY videos_by_title_year FROM 'videos_by_title_year.csv' WITH HEADER=true;



```
cqlsh:killervideos> COPY videos_by_title_year FROM '/home/pavel_orekhov/labwork/exercise-3/videos_by_title_year.csv' WITH HEADER=true;
Using 1 child processes

Starting copy of killervideos.videos_by_title_year with columns ['title', 'added_year', 'added_date', 'description', 'user_id', 'video_id'].
Processed: 430 rows; Rate:     639 rows/s; Avg. rate:     977 rows/s
430 rows imported from 1 files in 0.440 seconds (0 skipped).
cqlsh:killervideos>
```

15. Try running queries on the 'videos_by_title_year' table to query on a specific 'title' and 'added_year'.



```
InvalidRequest: code=2200 [Invalid query] message="Partition key parts: added_year must be restricted as other parts are"
cqlsh:killervideos> select * from videos_by_title_year where added_year = 1999;
InvalidRequest: code=2200 [Invalid query] message="Partition key parts: title must be restricted as other parts are"
cqlsh:killervideos>
```

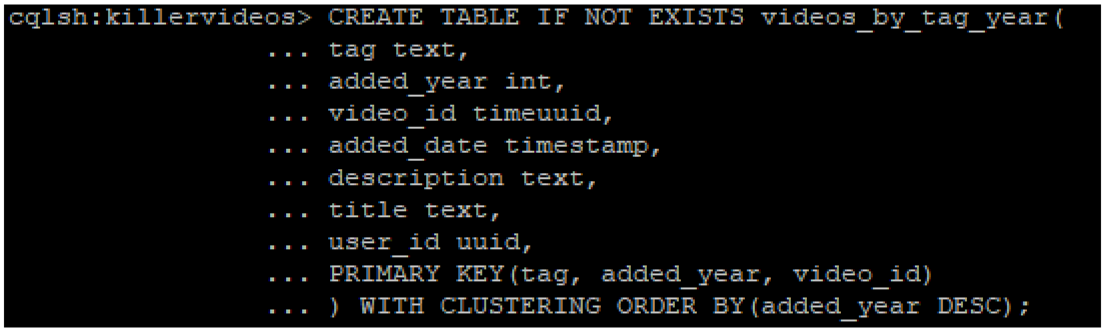
16. What error does Cassandra return when you try to query on just title or just year? Why?

It returns the error message I posted above. Because they have been combined into 1 compound partition key, hence the hash value for the partition was computed on this paired object, and not on the individual columns, hence, if you give it just one part of the paired object, it won't be able to retrieve the hash value of the pair.

17. Create a table with the columns above to facilitate querying for videos by tag within a given year range returning the results in descending order by year.
18. We wrote most of the CREATE TABLE for you. Fill in the PRIMARY KEY and CLUSTERING ORDER BY.

```
CREATE TABLE videos_by_tag_year (
tag text,
added_year int,
video_id timeuuid,
added_date timestamp,
description text,
title text,
```

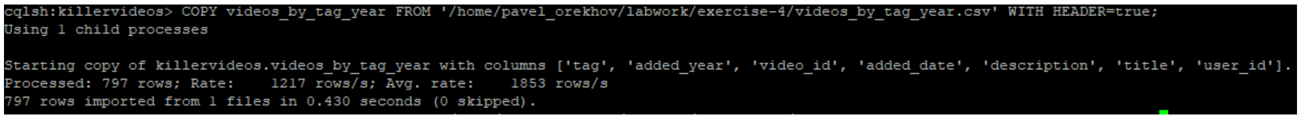
```
user_id uuid,
PRIMARY KEY ( )
) WITH CLUSTERING ORDER BY ( );
```



```
cqlsh:killervideos> CREATE TABLE IF NOT EXISTS videos_by_tag_year(
                ... tag text,
                ... added_year int,
                ... video_id timeuuid,
                ... added_date timestamp,
                ... description text,
                ... title text,
                ... user_id uuid,
                ... PRIMARY KEY(tag, added_year, video_id)
                ... ) WITH CLUSTERING ORDER BY(added_year DESC);
```

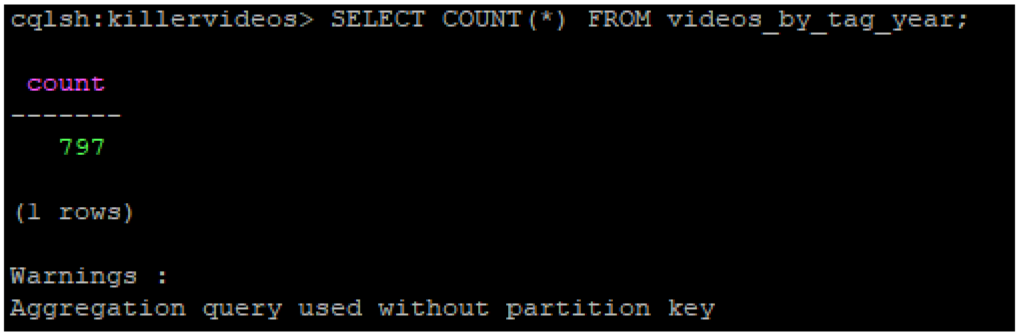
I added **video_id** into the primary key for simplicity, when copying data, otherwise, it is not kept in the right order.

19. Load the data from the 'videos_by_tag_year.csv' file in the provided 'exercise=4' directory using the COPY command.
COPY videos_by_tag_year FROM 'videos_by_tag_year.csv' WITH HEADER=true;



```
cqlsh:killervideos> COPY videos_by_tag_year FROM '/home/pavel_orekhov/labwork/exercise-4/videos_by_tag_year.csv' WITH HEADER=true;
Using 1 child processes

Starting copy of killervideos.videos_by_tag_year with columns ['tag', 'added_year', 'video_id', 'added_date', 'description', 'title', 'user_id'].
Processed: 797 rows; Rate:    1217 rows/s; Avg. rate:    1853 rows/s
797 rows imported from 1 files in 0.430 seconds (0 skipped).
```

20. Check the number of rows in the 'videos_by_tag_year' table.
NOTE: The number of rows should match the number of rows imported by the COPY command.

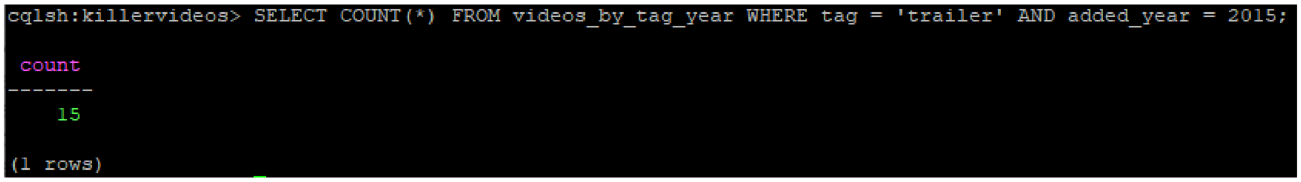


```
cqlsh:killervideos> SELECT COUNT(*) FROM videos_by_tag_year;

 count
-------
   797

(1 rows)

Warnings :
Aggregation query used without partition key
```

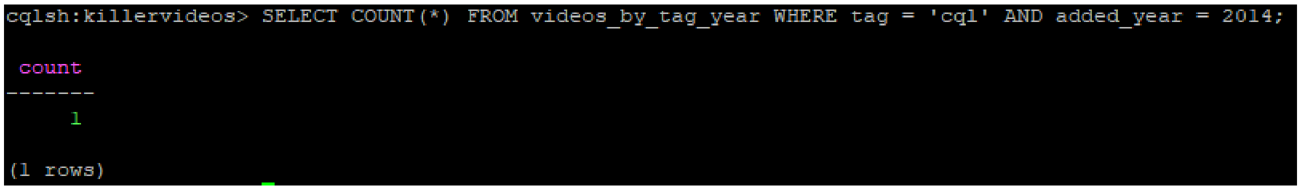
21. Try running queries on the 'videos_by_tag_year' table to query on a specific tag and added year.

Example queries:

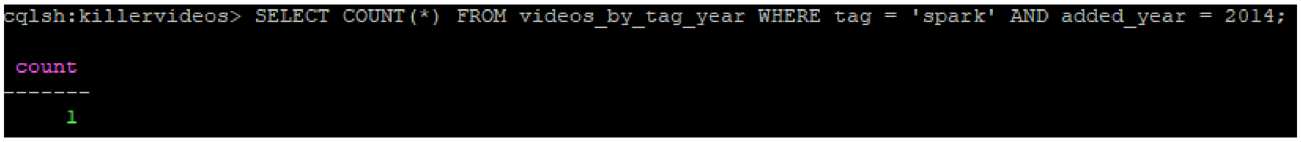
Tag	Added_year
trailer	2015
cql	2014
spark	2014



```
cqlsh:killervideos> SELECT COUNT(*) FROM videos_by_tag_year WHERE tag = 'trailer' AND added_year = 2015;

 count
-------
    15

(1 rows)
```

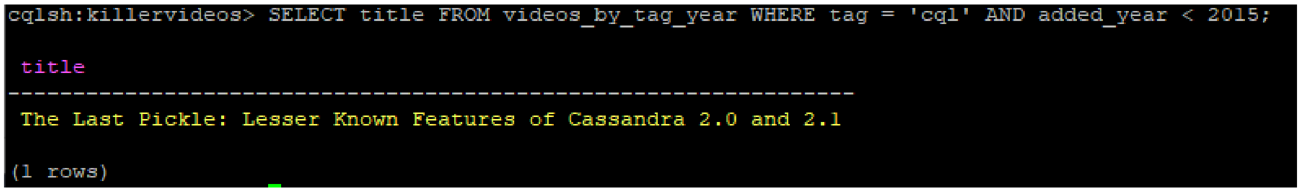


```
cqlsh:killervideos> SELECT COUNT(*) FROM videos_by_tag_year WHERE tag = 'cql' AND added_year = 2014;

 count
-------
     1

(1 rows)
```



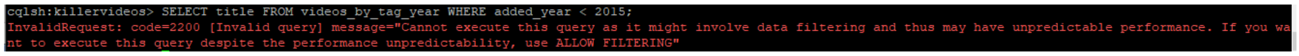
```
cqlsh:killervideos> SELECT COUNT(*) FROM videos_by_tag_year WHERE tag = 'spark' AND added_year = 2014;

 count
-------
     1
```

22. Try querying for all videos with tag "cql" added before the year 2015. Notice you can do range queries on clustering columns.



```
cqlsh:killervideos> SELECT title FROM videos_by_tag_year WHERE tag = 'cql' AND added_year < 2015;

 title
------------------------------------------------------------------
 The Last Pickle: Lesser Known Features of Cassandra 2.0 and 2.1

(1 rows)
```

23. Try querying for all videos added before 2015. The query will fail. What error message does cqlsh report? Why did the query fail whereas the previous query worked?

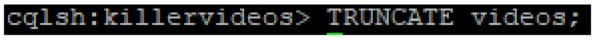


```
cqlsh:killervideos> SELECT title FROM videos_by_tag_year WHERE added_year < 2015;
InvalidRequest: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you wa
nt to execute this query despite the performance unpredictability, use ALLOW FILTERING"
```

Cassandra gives this message as a form of warning, telling us that the performance of our query will be bad and utilizing a lot of computing resources. Indeed, it will have to go through the entire table, filtering out the **added_year** values that we do not want, which is suboptimal in the case when there are a lot of such unwanted values.

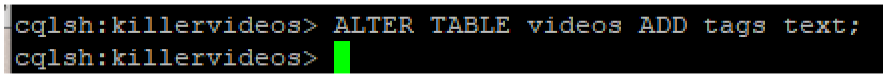
The previous query worked, because we filter on the field that our table is partitioned by, which is supposed to return only a small portion of data. It did not complain about the **added_year** range filter, because our rows are ordered by the **added_year** column within each partition.

25. Run the TRUNCATE command to erase the data from the 'videos' table.



```
cqlsh:killervideos> TRUNCATE videos;
```

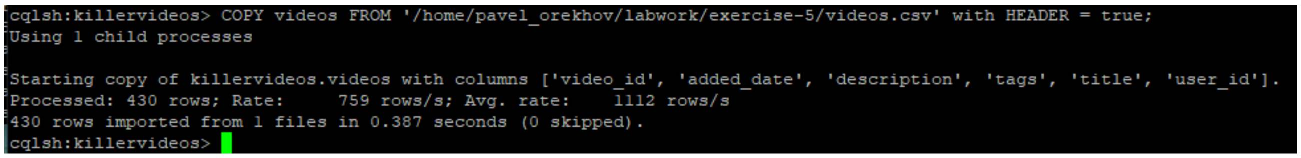
26. Alter the 'videos' table to add a 'tags' column.



```
cqlsh:killervideos> ALTER TABLE videos ADD tags text;
cqlsh:killervideos>
```

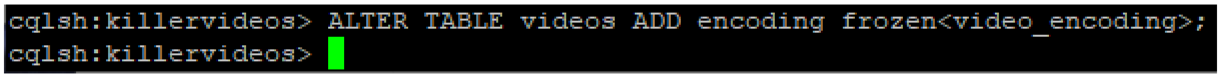
27. Load the data from the 'videos.csv' file using the COPY command. COPY videos FROM 'videos.csv' WITH HEADER=true; Remember, we do not need to create the user defined type called 'video_encoding' because we did so in the previous exercise. However, take a look at the code below as a refresher. Do not run it again or you will get an error!

```
CREATE TYPE video_encoding (
bit_rates SET<TEXT>,
encoding TEXT,
height INT,
width INT,
);
```



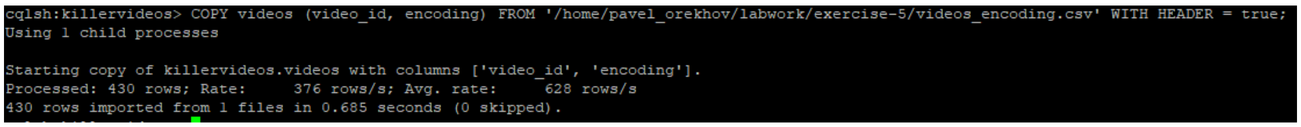
```
cqlsh:killervideos> COPY videos FROM '/home/pavel_orekhov/labwork/exercise-5/videos.csv' with HEADER = true;
Using 1 child processes

Starting copy of killervideos.videos with columns ['video_id', 'added_date', 'description', 'tags', 'title', 'user_id'].
Processed: 430 rows; Rate:     759 rows/s; Avg. rate:    1112 rows/s
430 rows imported from 1 files in 0.387 seconds (0 skipped).
cqlsh:killervideos>
```

28. Alter your table to add an 'encoding' column of the 'video_encoding' type.



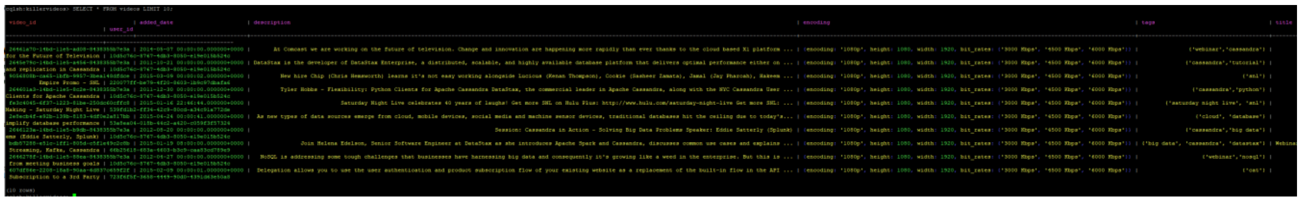
```
cqlsh:killervideos> ALTER TABLE videos ADD encoding frozen<video_encoding>;
cqlsh:killervideos>
```

29. Load the data from the 'videos_encoding.csv' file using the COPY command.
COPY videos (video_id, encoding) FROM 'videos_encoding.csv' WITH HEADER=true;



```
cqlsh:killervideos> COPY videos (video_id, encoding) FROM '/home/pavel_orekhov/labwork/exercise-5/videos_encoding.csv' WITH HEADER = true;
Using 1 child processes

Starting copy of killervideos.videos with columns ['video_id', 'encoding'].
Processed: 430 rows; Rate:     376 rows/s; Avg. rate:     628 rows/s
430 rows imported from 1 files in 0.685 seconds (0 skipped).
```

30. Run a query to retrieve the first 10 rows of the 'videos' table. Notice the altered table contains data for the new 'tags' and 'encoding' column.



```
cqlsh:killervideos> SELECT * FROM videos LIMIT 10;

 video_id                             | added_date                      | description                                                                                                                  | encoding                                                                                        | tags                                        | title
                          | user_id
--------------------------------------+---------------------------------+------------------------------------------------------------------------------------------------------------------------------+-------------------------------------------------------------------------------------------------+---------------------------------------------+--------
--------------------------+--------------------------------------
 2646la70-14bd-11e5-ad08-8438355b7e3a | 2014-05-07 00:00:00.000000+0000 |       At Comcast we are working on the future of television. Change and innovation are happening more rapidly than ever thanks to the cloud based X1 platform ... | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} |           {'webinar','cassandra'} |
for the Future of Television | 10d5c76c-8767-4db3-8050-e19e015b524c
 2645e79c-14bd-11e5-a456-8438355b7e3a | 2011-10-21 00:00:00.000000+0000 | DataStax is the developer of DataStax Enterprise, a distributed, scalable, and highly available database platform that delivers optimal performance either on ... | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} |          {'cassandra','tutorial'} |
and replication in Cassandra | 10d5c76c-8767-4db3-8050-e19e015b524c
 9056808b-ca65-1bfb-9957-3bea148dfdce | 2015-03-09 00:00:02.000000+0000 |        New hire Chip (Chris Hemsworth) learns it's not easy working alongside Lucious (Kenan Thompson), Cookie (Sasheer Zamata), Jamal (Jay Pharoah), Hakeem ... | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} |                           {'snl'} |
         Empire Promo - SNL | 220077ff-be79-4f20-8603-1b9c97dbafa6
 264601a3-14bd-11e5-8c2e-8438355b7e3a | 2011-12-30 00:00:00.000000+0000 |        Tyler Hobbs - Flexibility: Python Clients for Apache Cassandra DataStax, the commercial leader in Apache Cassandra, along with the NYC Cassandra User ... | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} |            {'cassandra','python'} |
Clients for Apache Cassandra | 10d5c76c-8767-4db3-8050-e19e015b524c
 fe3c4045-6f37-1223-81be-250dc60cffc8 | 2015-01-16 22:46:44.000000+0000 |                             Saturday Night Live celebrates 40 years of laughs! Get more SNL on Hulu Plus: http://www.hulu.com/saturday-night-live Get more SNL: ... | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} |     {'saturday night live', 'snl'} |
Making - Saturday Night Live | 539fd1b2-ff34-42c9-80cd-a34c91a772de
 2e5ecb4f-e92b-139b-8183-4df0e2a817bb | 2015-04-24 00:00:41.000000+0000 |  As new types of data sources emerge from cloud, mobile devices, social media and machine sensor devices, traditional databases hit the ceiling due to today's... | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} |            {'cloud', 'database'} |
implify database performance | 53a5ea04-018b-44c2-a420-c059f3f57324
 2646123e-14bd-11e5-b9db-8438355b7e3a | 2012-08-20 00:00:00.000000+0000 |                                                                   Session: Cassandra in Action - Solving Big Data Problems Speaker: Eddie Satterly (Splunk) | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} |          {'cassandra','big data'} |
ems (Eddie Satterly, Splunk) | 10d5c76c-8767-4db3-8050-e19e015b524c
 bdb87288-e51c-1ff1-805d-c5f1e49c2c8b | 2015-01-19 08:00:00.000000+0000 |                  Join Helena Edelson, Senior Software Engineer at DataStax as she introduces Apache Spark and Cassandra, discusses common use cases and explains ... | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} | {'big data', 'cassandra', 'datastax'} | Webina
 Streaming, Kafka, Cassandra | 66b25618-683a-4603-b3c9-caa83cd789e9
 2646278f-14bd-11e5-88ea-8438355b7e3a | 2012-04-27 00:00:00.000000+0000 |  NoSQL is addressing some tough challenges that businesses have harnessing big data and consequently it's growing like a weed in the enterprise. But this is ... | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} |              {'webinar','nosql'} |
 from meeting business goals | 10d5c76c-8767-4db3-8050-e19e015b524c
 607df86e-2208-18a8-90aa-6d837c659f2f | 2015-02-09 00:00:01.000000+0000 |  Delegation allows you to use the user authentication and product subscription flow of your existing website as a replacement of the built-in flow in the API ... | {encoding: '1080p', height: 1080, width: 1920, bit_rates: {'3000 Kbps', '4500 Kbps', '6000 Kbps'}} |                           {'oat'} |
 Subscription to a 3rd Party | 723f6f5f-3658-4449-90d0-4391d63e50a8

(10 rows)
cqlsh:killervideos>
```

It does indeed.