

The Start-up:

```
C:\Users\ASUS>cassandra
WARNING! Powershell script execution unavailable.
Please use 'powershell Set-ExecutionPolicy Unrestricted'
on this user-account to run cassandra with fully featured
functionality on this platform.
Starting with legacy startup options
Starting Cassandra Server
INFO [main] 2021-07-14 16:37:12,873 YamlConfigurationLoader.java:89 - Configuration location: file:/C:/Cassandra/apache-cassandra-3.11.4/conf/cassandra.yaml
INFO [main] 2021-07-14 16:37:15,387 Config.java:496 - Node configuration:[allocate_tokens_for_keyspace=null; authenticator=AllowAllAuthenticator; authorizer=AllowAllAuthorizer; auto_bootstrap=true; auto_snapshot=true; back_pressure_enabled=false; back_pressure_strategy=org.apache.cassandra.net.RateBasedBackPressure{high_ratio=0.9, factor=5, flow=FAST}; batch_size_fail_threshold_in_kb=50; batch_size_warn_threshold_in_kb=5; batchlog_replay_throttle_in_kb=1024; broadcast_address=null; broadcast_rpc_address=null; buffer_pool_use_heap_if_exhausted=true; cas_contention_timeout_in_ms=1000; cdc_enabled=false; cdc_free_space_check_interval_ms=250; cdc_raw_directory=null; cdc_total_space_in_mb=0; client_encryption_options=<REDACTED>; cluster_name=Test Cluster; column_index_cache_size_in_kb=2; column_index_size_in_kb=64; commit_failure_policy=stop; commitlog_compression=null; commitlog_directory=null; commitlog_max_compression_buffers_in_pool=3; commitlog_periodic_queue_size=-1; commitlog_segment_size_in_mb=32; commitlog_sync=periodic; commitlog_sync_batch_window_in_ms=NaN; commitlog_sync_period_in_ms=10000; commitlog_total_space_in_mb=null; compaction_large_partition_warning_threshold_mb=100; compaction_throughput_mb_per_sec=16; concurrent_compactors=null; concurrent_counter_writes=32; concurrent_materialized_view_writes=32; concurrent_reads=32; concurrent_replicates=null; concurrent_writes=32; counter_cache_keys_to_save=2147483647; counter_cache_save_period=7200; counter_cache_size_in_mb=null; counter_write_request_timeout_in_ms=5000; credentials_cache_max_entries=1000; credentials_update_interval_in_ms=-1; credentials_validity_in_ms=2000; cross_node_timeout=false; data_file_directories=[Ljava.lang.String;@13c10b87; disk_access_mode=auto; disk_failure_policy=stop; disk_optimization_estimate_percentile=0.95; disk_optimization_page_cross_chance=0.1; disk_optimization_strategy=ssd; dynamic_snitch=true; dynamic_snitch_badness_threshold=0.1; dynamic_snitch_reset_interval_in_ms=600000; dynamic_snitch_update_interval_in_ms=100; enable_materialized_views=true; enable_scripted_user_defined_functions=true; enable_user_defined_functions=true; enable_user_defined_functions_threads=true; encryption_options=null; endpoint_snitch=SimpleSnitch; file_cache_round_up=128; netty_codec=netty-codec-4.0.44.Final.452812a; netty_codec_nio_proxy=netty-codec-nio-proxy-4.0.44.Final.452812a; netty_codec_http=netty-codec-http-4.0.44.Final.452812a, netty_codec_socks=netty-codec-socks-4.0.44.Final.452812a, netty_common=netty-common-4.0.44.Final.452812a, netty_handler=netty-handler-4.0.44.Final.452812a, netty_tcnative=netty-tcnative-1.1.33.Fork26.142ecbb, netty_transport=netty-transport-4.0.44.Final.452812a, netty_transport_native_epoll=netty-transport-native-epoll-4.0.44.Final.452812a, netty_transport_rxtx=netty-transport-rxtx-4.0.44.Final.452812a, netty_transport_sctp=netty-transport-sctp-4.0.44.Final.452812a, netty_transport_udt=netty-transport-udt-4.0.44.Final.452812a]
INFO [main] 2021-07-14 16:37:39,958 Server.java:156 - Starting listening for CQL clients on localhost/127.0.0.1:9042 (unencrypted)...
INFO [main] 2021-07-14 16:37:41,822 ThriftServer.java:116 - Binding thrift service to localhost/127.0.0.1:9160
INFO [Thread-2] 2021-07-14 16:37:41,834 ThriftServer.java:133 - Listening for thrift clients...
```

Starting CQLSH:

```
C:\Users\ASUS>cqlsh
WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.

Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh>
```

Creating Keyspaces:

```
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> CREATE KEYSPACE instagram
... WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : 3};
cqlsh> use instagram ;
cqlsh:instagram>
```

Creating tables:

```
cqlsh> use instagram ;
cqlsh:instagram> create table user (
    ... language text,
    ... createdAt timestamp,
    ... email text,
    ... phoneNo text,
    ... username text,
    ... password text,
    ... dob text,
    ... gender text,
    ... tokenVersion boolean,
    ... blockedUsers list<text>,
    ... savedPosts list<uuid>,
    ... followers list<text>,
    ... following list<text>,
    ... accountType boolean,
    ... accountType2 map<text,boolean>,
    ... requests list<text>,
    ... taggedIn list<uuid>,
    ... liked list<uuid>,
    ... account boolean,
    ... verified boolean,
    ... blockUserFromComment list<text>,
    ... allowReplyReaction map<text,boolean>.
```

```
    ... liveArchive boolean,
    ... restrictAccounts list<text>,
    ... NPS list<text>,
    ... profilePicture blob,
    ... fullName text,
    ... bio text,
    ... website text,
    ... blockedFrom list<text>,
    ... hideFromProfileGrid list<uuid>,
    ... posts list<uuid>,
    ... stories list<uuid>,
    ... recentStory list<uuid>,
    ... searchHistory list<text>,
    ... hashtags list<text>,
    ... countdown list<timeuuid>,
    ... primary key (email)
    ... );
cqlsh:instagram> create index username on user(username);
cqlsh:instagram>
cqlsh:instagram> create index phoneNo on user(phoneNo);
cqlsh:instagram>
```

```
cqlsh:instagram> create index location on post(location);
cqlsh:instagram> create table comment (
    ... p_id uuid,
    ... u_id text,
    ... likedBy list<text>,
    ... createdAt timestamp,
    ... rComment boolean,
    ... comment_id uuid,
    ... commentText text,
    ... owner_id text,
    ... disabled boolean,
    ... bComment boolean,
    ... primary key (comment_id)
    ... );
cqlsh:instagram> create index u_idComment on comment(u_id);
cqlsh:instagram>
cqlsh:instagram> create index p_idComment on comment(p_id);
cqlsh:instagram>
```

Queries

Insert User:

```
cqlsh:instagram> insert into user (username,email,phoneNo,password,dob,fullname,createdAt,language,gender,accountType,accountType2,account,verified,allowReplyReaction,manuallyApproveTags,allowTagsFrom,likeViewCount,allowCommentFrom,checkComment,allowAtMention,allowToShareOnStory,storyArchive,liveArchive) values ('millie.B.brown','eleven11@netflix.com','1323435555','upsideDown','2004-09-17','Millie Brown',toTimestamp(now()),'en-us','Female',true, {'Normal': true, 'Business' : false, 'Professional' : false}, true, true, {'All':true, 'none': false, 'following':false}, false, {'All':true, 'none': false, 'following':false}, true, {'everyone' : true,'following': true, 'followers':true,'followingFollowers' : true}, false, { 'All':true, 'none':false, 'following': false }, true, true, true);
```

Insert Post:

```
cqlsh:instagram> BEGIN BATCH
... insert into post (p_id,u_id,createdAt,modifiedAt,caption,location,tagged,type,hashTags,deleted,commenting,archived,count,title,music,disabled) values (78f738f6-e5a1-11eb-ba80-0242ac130004,'michealcollins@gmail.com',toTimestamp(now()),toTimestamp(now()),'test caption 5', 'test location 5', {'email1':true,'email2' : true}, 'Post',['#hashtag1','#hashtag2'],false,true,false,true,'test title 5',now(),false);
... update user set posts = posts + [78f738f6-e5a1-11eb-ba80-0242ac130004] where email = 'michealcollins@gmail.com';
... APPLY BATCH;
```

Insert Story:

```
cqlsh:instagram> BEGIN BATCH
... insert into story (s_id,u_id,createdat,viewtype,mention,location,archived,music,link,deleted,disabled,poll,question,eslider,quizcontent) values (01acd190-e612-11eb-ba80-0242ac130004,'michealcollins@gmail.com',toTimestamp(now()),'CloseFriends',[],'Winterfell',false,'GOT theme','',false,false,[],'Ask me anything about Winterfell ?','how much Cersi Lanister enjoyed in Winterfell ?',[]);
... update user set stories = stories + [01acd190-e612-11eb-ba80-0242ac130004] where email = 'michealcollins@gmail.com';
... APPLY BATCH;
```

Insert Comment:

```
cqlsh:instagram> insert into comment (p_id,u_id,createdat,rComment,comment_id,commentText,owner_id,disabled,bComment) values (78f738f6-e5a1-11eb-ba80-0242ac130004, 'eleven11@netflix.com',toTimestamp(now()),false,now(),'Hi Mike your post is amazing','michealcollins@gmail.com',false,false);
cqlsh:instagram> █
```

Send request for Verification:

```
cqlsh:instagram> insert into requestForVerification (id,u_id,fullname,knownas,category,image,status) values (now(),'michealcollins@gmail.com','Micheal Bruce Collins','Mike', 'Actor','www.greencardofUS.com','requested');
cqlsh:instagram>
```

Update Profile:

```
cqlsh:instagram> select username from user where username = 'mike#123';

username
-----
mike#123

(1 rows)
cqlsh:instagram> select username from user where username = 'tonyStark3000';

username
-----

(0 rows)
cqlsh:instagram>
```

```
cqlsh:instagram> select phoneNo from user where phoneNo = '1323435555';
```

```
phoneno
-----
1323435555
```

```
(1 rows)
```

```
cqlsh:instagram> select phoneNo from user where phoneNo = '1323433234';
```

```
phoneno
-----
```

```
(0 rows)
```

```
cqlsh:instagram>
```

```
cqlsh:instagram> update user set username = 'tonyStark3000',website = 'www.starkindustries.us',bio = 'Best Avenger' where email = 'mike123@gmail.com';
```

```
cqlsh:instagram> insert into logs (u_id,createdat,value,activityrelated,type) values ('mike123@gmail.com',toTimestamp(now()),'tonyStark3000',false,'username');
```

```
cqlsh:instagram> insert into logs (u_id,createdat,value,activityrelated,type) values ('mike123@gmail.com',toTimestamp(now()),'www.starkindustries.us',false,'website');
```

```
cqlsh:instagram> insert into logs (u_id,createdat,value,activityrelated,type) values ('mike123@gmail.com',toTimestamp(now()),'Best Avenger',false,'bio');
```

```
cqlsh:instagram> █
```

Like/Unlike Post/Comment:

```
cqlsh:instagram> BEGIN BATCH
```

```
... update post set likedBy = likedBy + ['eleven11@netflix.com'],modifiedat = toTimestamp(now()) where p_id = 78f738f6-e5a1-11eb-ba80-0242ac130004;
```

```
... update user set liked = liked + [78f738f6-e5a1-11eb-ba80-0242ac130004] where email = 'eleven11@netflix.com';
```

```
... insert into logs (u_id,createdat,value,activityrelated,type) values ('eleven11@netflix.com',toTimestamp(now()),'78f738f6-e5a1-11eb-ba80-0242ac130004',true,'post');
```

```
... APPLY BATCH;
```

```
cqlsh:instagram> update PostStats set likes = likes + 1 where p_id = 78f738f6-e5a1-11eb-ba80-0242ac130004;
```

```
cqlsh:instagram>
```

```
cqlsh:instagram> BEGIN BATCH
```

```
... update comment set likedBy = likedBy + ['eleven11@netflix.com'] where comment_id = cae37770-e627-11eb-86e6-4ff50b69c2d0;
```

```
... insert into logs (u_id,createdat,value,activityrelated,type) values ('eleven11@netflix.com',toTimestamp(now()),'cae37770-e627-11eb-86e6-4ff50b69c2d0',true,'comment');
```

```
... APPLY BATCH;
```

```
cqlsh:instagram> update Commentlikes set likes = likes + 1 where comment_id = cae37770-e627-11eb-86e6-4ff50b69c2d0;
```

```
cqlsh:instagram>
```

Block Profile: User1 blocks User2's profile:

```
cqlsh:instagram> BEGIN BATCH
```

```
... update user set blockedusers = blockedusers + ['eleven11@netflix.com'] where email = 'shatya0910@gmail.com';
```

```
... update user set blockedfrom = blockedfrom + ['shatya0910@gmail.com'] where email = 'eleven11@netflix.com';
```

```
... update user set following = following - ['eleven11@netflix.com'] where email = 'shatya0910@gmail.com';
```

```
... update user set followers = followers - ['eleven11@netflix.com'] where email = 'shatya0910@gmail.com';
```

```
... update user set following = following - ['shatya0910@gmail.com'] where email = 'eleven11@netflix.com';
```

```
... update user set followers = followers - ['shatya0910@gmail.com'] where email = 'eleven11@netflix.com';
```

```
... APPLY BATCH;
```

```
cqlsh:instagram> █
```

```

cqlsh:instagram> BEGIN BATCH
... update user set blockedusers = blockedusers - ['eleven11@netflix.com'] where email = 'shatya0910@gmail.com';
... update user set blockedfrom = blockedfrom - ['shatya0910@gmail.com'] where email = 'eleven11@netflix.com';
... APPLY BATCH;
cqlsh:instagram>

```

Save/Unsave Post:

```

cqlsh:instagram> BEGIN BATCH
... update user set savedPosts = savedPosts + [78f738f6-e5a1-11eb-ba80-0242ac130004] where email = 'eleven11@netflix.com';
... APPLY BATCH;
cqlsh:instagram> update PostStats set savedCount = savedCount + 1 where p_id = 78f738f6-e5a1-11eb-ba80-0242ac130004;
cqlsh:instagram>

```

```

cqlsh:instagram> BEGIN BATCH
... update user set savedPosts = savedPosts - [78f738f6-e5a1-11eb-ba80-0242ac130004] where email = 'eleven11@netflix.com';
... APPLY BATCH;
cqlsh:instagram> update PostStats set savedCount = savedCount - 1 where p_id = 78f738f6-e5a1-11eb-ba80-0242ac130004;
cqlsh:instagram>

```

Follow/Unfollow Profile: User 1 follows/unfollows user 2:

```

cqlsh:instagram>
cqlsh:instagram> select accounttype from user where email = 'eleven11@netflix.com';

accounttype
-----
True

(1 rows)
cqlsh:instagram>

```

```

cqlsh:instagram> BEGIN BATCH
... update user set following = following + ['eleven11@netflix.com'] where email = 'shatya0910@gmail.com';
... update user set followers = followers + ['shatya0910@gmail.com'] where email = 'eleven11@netflix.com';
... APPLY BATCH;
cqlsh:instagram>

```

```

cqlsh:instagram>
cqlsh:instagram> select accounttype from user where email = 'sanketjain123@gmail.com';

accounttype
-----
False

(1 rows)
cqlsh:instagram>

```

```

cqlsh:instagram>
cqlsh:instagram> update user set requests = requests + ['shatya0910@gmail.com'] where email = 'sanketjain123@gmail.com';
cqlsh:instagram>

```

Follow/Unfollow Hashtag:

```

cqlsh:instagram>
cqlsh:instagram> select hashtag from hashtag where hashtag = '#hashtag1';

hashtag
-----
#hashtag1

(1 rows)
cqlsh:instagram> update user set hashtags = hashtags + ['#hashtag1'] where email = 'eleven11@netflix.com';
cqlsh:instagram>

```

Archive/Unarchive:

```

cqlsh:instagram> BEGIN BATCH
... update post set archived = true,modifiedat = toTimestamp(now()) where p_id = 78f738f6-e5a1-11eb-ba80-02
42ac130004;
... APPLY BATCH;
cqlsh:instagram> update userCount set postCount = postCount - 1 where email = 'michealcollins@gmail.com';
cqlsh:instagram>

```

```

cqlsh:instagram> BEGIN BATCH
... update post set archived = false,modifiedat = toTimestamp(now()) where p_id = 78f738f6-e5a1-11eb-ba80-0
242ac130004;
... APPLY BATCH;
cqlsh:instagram> update userCount set postCount = postCount + 1 where email = 'michealcollins@gmail.com';
cqlsh:instagram>

```

Change Password:

```

cqlsh:instagram> BEGIN BATCH
... update user set password = 'shelltrial' where email = 'michealcollins@gmail.com';
... insert into logs (u_id,createdat,value,type,activityrelated) values ('michealcollins@gmail.com',toTimes
tamp(now()),'password','password',false);
... APPLY BATCH;
cqlsh:instagram>
cqlsh:instagram>

```

Change Profile Details Visibility:

```

cqlsh:instagram>
cqlsh:instagram> BEGIN BATCH
... update user set accounttype = true where email = 'sanketjain123@gmail.com';
... insert into logs (u_id,createdat,value,type,activityrelated) values ('sanketjain123@gmail.com',toTimest
amp(now()),'Public','privacy',false);
... APPLY BATCH;
cqlsh:instagram> █

```

Delete Post/Story:


```

cqlsh:instagram> select posts from user where email = 'michealcollins@gmail.com';

posts
-----
[d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0, e7f39010-e59c-11eb-86e6-4ff50b69c2d0, faed80e0-e59c-11eb-86e6-4ff50b69c2d0, 78f738f6-e5a1-11eb-ba80-0242ac130004]

(1 rows)
cqlsh:instagram> BEGIN BATCH
... update post set deleted = true, modifiedat = toTimestamp(now()) where p_id = faed80e0-e59c-11eb-86e6-4ff50b69c2d0;
... update user set posts = posts - [faed80e0-e59c-11eb-86e6-4ff50b69c2d0] where email = 'michealcollins@gmail.com';
... APPLY BATCH;
cqlsh:instagram> update userCount set postCount = postCount - 1 where email = 'michealcollins@gmail.com';
cqlsh:instagram>
cqlsh:instagram> select posts from user where email = 'michealcollins@gmail.com';

posts
-----
[d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0, e7f39010-e59c-11eb-86e6-4ff50b69c2d0, 78f738f6-e5a1-11eb-ba80-0242ac130004]

(1 rows)
cqlsh:instagram>

```

View Saved Posts:

```

cqlsh:instagram>
cqlsh:instagram> select savedPosts from user where email = 'eleven11@netflix.com';

savedposts
-----
[78f738f6-e5a1-11eb-ba80-0242ac130004]

(1 rows)
cqlsh:instagram> select p_id,u_id,createdat,modifiedat,caption,location,tagged,likedBy,hashtags,title,music,content from
post where p_id in (78f738f6-e5a1-11eb-ba80-0242ac130004) AND type = 'Post' AND deleted = false AND archived = false AND
disabled = false ALLOW FILTERING;

p_id          | caption | u_id          | location | tagged | createdat          | likedby          | modifiedat          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 78f738f6-e5a1-11eb-ba80-0242ac130004 | test caption 5 | michealcollins@gmail.com | test location 5 | {'email1': True, 'email2': True} | ['eleven11@netflix.com'] | ['#hashtag1', '#hashtag2'] | test title 5 | b278ab04-e5a1-11eb-86e6-4ff50b69c2d0 | null
(1 rows)
cqlsh:instagram>

```

View post(s) a particular user or the user himself/herself is/are tagged in:

```
cqlsh:instagram> select taggedIn from user where email = 'eleven11@netflix.com';

taggedin
-----
[78f738f6-e5a1-11eb-ba80-0242ac130004, d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0]

(1 rows)
cqlsh:instagram> select p_id,u_id,createdat,modifiedat,caption,location,tagged,likedBy,hashtags,title,music,content from
post where p_id in (78f738f6-e5a1-11eb-ba80-0242ac130004,d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0) AND type = 'Post'
... AND deleted = false AND archived = false AND disabled = false ALLOW FILTERING;

p_id          | u_id          | createdat          | modifiedat          |
+-----+-----+-----+-----+
| caption      | location      | tagged             | likedby             |
+-----+-----+-----+-----+
| content      |               | hashtags           | title               |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0 | michealcollins@gmail.com | 2021-07-15 18:45:23.834000+0000 | 2021-07-15 18:45:23
.834000+0000 | test caption | test location | {'eleven11@netflix.com': True, 'email1': T
rue, 'email2': True} | null | ['#hashtag1', '#hashtag2'] | test title | d05d0cb2-e59c-11eb-86e6-4f
f50b69c2d0 | null
78f738f6-e5a1-11eb-ba80-0242ac130004 | michealcollins@gmail.com | 2021-07-15 19:20:21.168000+0000 | 2021-07-16 12:49:05
.972000+0000 | test caption 5 | test location 5 | {'eleven11@gmail.com': True, 'eleven11@netflix.com': True, 'email1': T
rue, 'email2': True} | ['eleven11@netflix.com'] | ['#hashtag1', '#hashtag2'] | test title 5 | b278ab04-e5a1-11eb-86e6-4f
f50b69c2d0 | null

(2 rows)
cqlsh:instagram>
```

View posts I've liked:

```
cqlsh:instagram> select liked from user where email = 'eleven11@netflix.com';

liked
-----
[78f738f6-e5a1-11eb-ba80-0242ac130004]

(1 rows)
cqlsh:instagram> select p_id,u_id,createdat,modifiedat,caption,location,tagged,likedBy,hashtags,title,music,content from
post where p_id in (78f738f6-e5a1-11eb-ba80-0242ac130004) AND type = 'Post'
... AND deleted = false AND archived = false AND disabled = false ALLOW FILTERING;

p_id          | u_id          | createdat          | modifiedat          |
+-----+-----+-----+-----+
| caption      | location      | tagged             | likedby             |
+-----+-----+-----+-----+
| content      |               | hashtags           | title               |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
78f738f6-e5a1-11eb-ba80-0242ac130004 | michealcollins@gmail.com | 2021-07-15 19:20:21.168000+0000 | 2021-07-16 12:49:05
.972000+0000 | test caption 5 | test location 5 | {'eleven11@gmail.com': True, 'eleven11@netflix.com': True, 'email1': T
rue, 'email2': True} | ['eleven11@netflix.com'] | ['#hashtag1', '#hashtag2'] | test title 5 | b278ab04-e5a1-11eb-86e6-4f
f50b69c2d0 | null

(1 rows)
cqlsh:instagram>
```

View posts/stories I've archived:


```
cqlsh:instagram> select posts from user where email = 'michealcollins@gmail.com';
```

posts
[d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0, e7f39010-e59c-11eb-86e6-4ff50b69c2d0, 78f738f6-e5a1-11eb-ba80-0242ac130004]

```
(1 rows)
cqlsh:instagram> select p_id,u_id,createdat,modifiedat,caption,location,tagged,likedBy,hashtags,title,music,content from
post where p_id in (d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0, e7f39010-e59c-11eb-86e6-4ff50b69c2d0, 78f738f6-e5a1-11eb-ba80-0242ac130004) AND type = 'Post' AND deleted = false AND archived = true AND disabled = false ALLOW FILTERING;
```

p_id	caption	location	u_id	tagged	createdat	modifiedat	likedby	hashtags	title	music	content
e7f39010-e59c-11eb-86e6-4ff50b69c2d0	test caption 2	test location 2	michealcollins@gmail.com	{'email1': True, 'email2': True}	2021-07-15 18:46:03.409000+0000	2021-07-16 13:41:58.008000+0000	null	['#hashtag1', '#hashtag2']	test title 2		

```
(1 rows)
cqlsh:instagram>
cqlsh:instagram>
cqlsh:instagram>
```

```
cqlsh:instagram>
cqlsh:instagram> select stories from user where email = 'michealcollins@gmail.com';
```

stories
[01acd190-e612-11eb-ba80-0242ac130004, 14ff993c-e63d-11eb-ba80-0242ac130004]

```
(1 rows)
cqlsh:instagram> select s_id,u_id,createdat,viewedby,viewtype,mention,location,music,link,content,pollsResponses,question
nsResponses,emojisliderResponses,quizResponses,poll,
... question,eslider,quizContent from story where s_id in(01acd190-e612-11eb-ba80-0242ac130004, 14ff993c-e63d-11eb-ba80-0242ac130004) AND archived = true AND deleted = false AND disabled = false ALLOW FILTERING;
```

s_id	mention	location	music	u_id	link	content	pollsresponses	questionsresponses	emojisliderresponses	quizresponses	poll	question	eslider	quizcontent
14ff993c-e63d-11eb-ba80-0242ac130004	null	Winterfell	GOT theme	michealcollins@gmail.com								Ask me anything about Winterfell ?	how much Cersi Lanister enjoyed in Winterfell ?	

```
(1 rows)
cqlsh:instagram>
```

View activity related to my account:

```
cqlsh:instagram>
cqlsh:instagram> select * from logs where type in ('tagged','mentioned','pending','comment') and u_id = 'eleven11@netfli
x.com' and activityrelated = true ALLOW FILTERING ;
```

type	createdat	u_id	activityrelated	value
comment	2021-07-16 11:44:05.206000+0000	eleven11@netflix.com	True	cae37770-e627-11eb-86e6-4ff50b69c2d0
comment	2021-07-16 11:42:35.657000+0000	eleven11@netflix.com	True	cae37770-e627-11eb-86e6-4ff50b69c2d0

```
(2 rows)
cqlsh:instagram> _
```

Login Activity Log:

```
cqlsh:instagram> select * from logs where type = 'Login' and u_id = 'michealcollins@gmail.com' and activityrelated = false ALLOW FILTERING ;
```

type	createdat	u_id	activityrelated	value
Login	2021-07-16 14:16:38.967000+0000	michealcollins@gmail.com	False	

Switch to Professional Account:

```
cqlsh:instagram> update user set accounttype=true , accounttype2['Business'] = false, accounttype2['Normal'] = false, accounttype2['Professional'] = true where email = 'sanketjain123@gmail.com';
cqlsh:instagram> select requests from user where email = 'sanketjain123@gmail.com';
```

requests
['shatya0910@gmail.com']

(1 rows)

```
cqlsh:instagram> BEGIN BATCH
... update user set requests = requests - ['shatya0910@gmail.com'] where email = 'sanketjain123@gmail.com';
... update user set following = following + ['sanketjain123@gmail.com'] where email = 'shatya0910@gmail.com';
... update user set followers = followers + ['shatya0910@gmail.com'] where email = 'sanketjain123@gmail.com';
... APPLY BATCH;
```

Story Archive (Location):

```
cqlsh:instagram> select stories from user where email = 'michealcollins@gmail.com';
```

stories
[01acd190-e612-11eb-ba80-0242ac130004, 14ff993c-e63d-11eb-ba80-0242ac130004]

(1 rows)

```
cqlsh:instagram> select s_id,u_id,createdat,viewedby,viewtype,mention,location,music,link,content,pollsResponses,questionsResponses,emojisliderResponses,quizResponses,poll,
... question,eslider,quizContent from story where s_id in(01acd190-e612-11eb-ba80-0242ac130004, 14ff993c-e63d-11eb-ba80-0242ac130004) AND archived = true AND deleted = false AND disabled = false AND location = 'Winterfell' ALLOW FILTERING;
```

s_id	u_id	createdat	viewedby	viewtype	mention	location	music	link	content	pollsresponses	questionsresponses	emojisliderresponses	quizresponses	poll	question	eslider	quizcontent
14ff993c-e63d-11eb-ba80-0242ac130004	michealcollins@gmail.com	2021-07-16 14:04:41.047000+0000	null	All		Winterfell	GOT theme										
									Ask me anything about Winterfell ?	how much Cersi Lanister enjoyed in Winterfell ?							

(1 rows)

View Entire History of Account:

```
cqlsh:instagram>
cqlsh:instagram> select * from logs where u_id = 'michealcollins@gmail.com' and activityrelated = false ALLOW FILTERING;
```

type	createdat	u_id	activityrelated	value
password	2021-07-16 12:47:47.106000+0000	michealcollins@gmail.com	False	password
Login	2021-07-16 14:16:38.967000+0000	michealcollins@gmail.com	False	
privacy	2021-07-16 12:50:47.821000+0000	michealcollins@gmail.com	False	Private

```
(3 rows)
cqlsh:instagram>
```

Comments' features:

```
cqlsh:instagram> update user set blockuserfromcomment = blockuserfromcomment + ['shatya0910@gmail.com'] where email = 'michealcollins@gmail.com';
cqlsh:instagram> update user set checkcomment = true where email = 'michealcollins@gmail.com';
cqlsh:instagram> update user set offensivewords = offensivewords + ['hell','dumb'] where email = 'michealcollins@gmail.com';
cqlsh:instagram> update user set allowCommentFrom['everyone'] = false ,allowCommentFrom ['followers'] = false, allowCommentFrom['following'] = false, allowCommentFrom['followingFollowers'] = True where email = 'michealcollins@gmail.com';
cqlsh:instagram>
```

View and manage Search History:

```
cqlsh:instagram> update user set searchHistory = searchHistory + ['TEST 1','search content 1','#verify','test location'] where email = 'michealcollins@gmail.com';
cqlsh:instagram>
cqlsh:instagram> select searchHistory from user where email = 'michealcollins@gmail.com';
```

searchhistory
['TEST 1', 'search content 1', '#verify', 'test location']

```
(1 rows)
cqlsh:instagram> update user set searchHistory = searchHistory - ['TEST 1'] where email = 'michealcollins@gmail.com';
cqlsh:instagram> select searchHistory from user where email = 'michealcollins@gmail.com';
```

searchhistory
['search content 1', '#verify', 'test location']

```
(1 rows)
cqlsh:instagram>
```

View and manage Recently Deleted Posts:

```

cqlsh:instagram>
cqlsh:instagram> select p_id,u_id,createdat,modifiedat,caption,location,tagged,likedBy,hashtags,title,music,content from
post where u_id = 'michealcollins@gmail.com' AND type = 'Post'
... AND deleted = true AND archived = false AND disabled = false ALLOW FILTERING;

p_id          | caption          | location          | u_id          | tagged          | createdat          | likedby          | modifiedat
| title          | music            |                 |                 | content         |                   |                 |
-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
885a7040-e59e-11eb-ba80-0242ac130004 | michealcollins@gmail.com | 2021-07-15 18:58:53.969000+0000 | 2021-07-16 13:03:22
.867000+0000 | test caption 4 | test location 4 | {'email1': True, 'email2': True} | null | ['#hashtag1', '#hashtag2'
] | test title 3 | b33db014-e59e-11eb-86e6-4ff50b69c2d0 | null
faed80e0-e59c-11eb-86e6-4ff50b69c2d0 | michealcollins@gmail.com | 2021-07-15 18:46:35.247000+0000 | 2021-07-16 13:06:35
.055000+0000 | test caption 3 | test location 3 | {'email1': True, 'email2': True} | null | ['#hashtag1', '#hashtag2'
] | test title 3 | faeda7f4-e59c-11eb-86e6-4ff50b69c2d0 | null

(2 rows)
cqlsh:instagram> BEGIN BATCH
... update post set deleted = false where p_id = faed80e0-e59c-11eb-86e6-4ff50b69c2d0;
... update user set posts = posts + [faed80e0-e59c-11eb-86e6-4ff50b69c2d0] where email = 'michealcollins@gm
ail.com';
... APPLY BATCH;
cqlsh:instagram>

```

Edit Posts:

```

cqlsh:instagram>
cqlsh:instagram> update post set caption = 'Update caption test', tagged= tagged + {'sanketjain123@gmail.com' : true}, l
ocation = 'Kings Landing',
... hashtags = hashtags + ['#GOT'], modifiedat = toTimestamp(now()) where p_id = 78f738f6-e5a1-11eb-ba80-02
42ac130004;
cqlsh:instagram>

```

Remove Follower: User1 removes User2

```

cqlsh:instagram>
cqlsh:instagram> BEGIN BATCH
... update user set following = following - ['sanketjain123@gmail.com'] where email = 'shatya0910@gmail.com
';
... update user set followers = followers - ['shatya0910@gmail.com'] where email = 'sanketjain123@gmail.com
';
... APPLY BATCH;
cqlsh:instagram>

```

Hide/Unhide Like and View Counts:

```

cqlsh:instagram>
cqlsh:instagram> update user set likeviewcount = false where email = 'michealcollins@gmail.com';
cqlsh:instagram>

```

Allow tags from:

```

cqlsh:instagram>
cqlsh:instagram> update user set allowtagsfrom['All'] = false , allowtagsfrom['following'] = false , allowtagsfrom['none
s'] = true where email = 'michealcollins@gmail.com';
cqlsh:instagram>

```

Hide story from particular user(s): User1 hides from User2:

```

cqlsh:instagram>
cqlsh:instagram> update user set hidestory = hidestory + ['shatya0910@gmail.com'] where email = 'sanketjain123@gmail.com
';
cqlsh:instagram>

```

Remove Tag: User1 removes tag from User2's post:

```
cqlsh:instagram>
cqlsh:instagram> select taggedIn from user where email = 'eleven11@netflix.com';

taggedin
-----
[78f738f6-e5a1-11eb-ba80-0242ac130004, d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0]

(1 rows)
cqlsh:instagram>
cqlsh:instagram> BEGIN BATCH
... update user set taggedIn = taggedIn - [d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0] where email = 'eleven11@netflix.com';
... update post set tagged = tagged - {'eleven11@gmail.com'} where p_id = d05ce5a0-e59c-11eb-86e6-4ff50b69c2d0;
... APPLY BATCH;
cqlsh:instagram> select taggedIn from user where email = 'eleven11@netflix.com';

taggedin
-----
[78f738f6-e5a1-11eb-ba80-0242ac130004]

(1 rows)
cqlsh:instagram> _
```

Allow @Mentions From:

```
cqlsh:instagram> update user set allowatmention['All'] = false, allowatmention['none'] = true, allowatmention['following'] = false where email = 'sanketjain123@gmail.com';
cqlsh:instagram>
cqlsh:instagram> _
```

Check Allow @Mentions From:

```
cqlsh:instagram>
cqlsh:instagram> select allowatmention from user where email = 'sanketjain123@gmail.com';

allowatmention
-----
{'All': False, 'following': False, 'none': True}

(1 rows)
cqlsh:instagram> _
```

Close Friends: User1 adds/removes User2 to close friends

```
cqlsh:instagram> update user set closefriends = closefriends + ['michealcollins@gmail.com'] where email = 'eleven11@netflix.com';
cqlsh:instagram> _
```

Allow Replies and Reactions on stories:

```
cqlsh:instagram> update user set allowreplyreaction['All'] = false, allowreplyreaction['following'] = false, allowreplyreaction['none'] = true where email = 'sanketjain123@gmail.com';
cqlsh:instagram> _
```

Manage Notifications:

```
cqlsh:instagram> update manageNotification set stories = stories + ['eleven11@netflix.com'] where email = 'michealcollins@gmail.com';
cqlsh:instagram> _
```

Add a story(old/current) to highlights:

```

cqlsh:instagram>
cqlsh:instagram> select stories from user where email = 'michealcollins@gmail.com';

stories
-----
[01acd190-e612-11eb-ba80-0242ac130004, 14ff993c-e63d-11eb-ba80-0242ac130004]

(1 rows)
cqlsh:instagram> insert into collection (id,email,collectionname,p_id,type) values (now(),'michealcollins@gmail.com','Test Collection',[01acd190-e612-11eb-ba80-0242ac130004, 14ff993c-e63d-11eb-ba80-0242ac130004],'highlights');
cqlsh:instagram>

```

Polls:

```

cqlsh:instagram> BEGIN BATCH
... update story set pollsResponses = pollsResponses + {'eleven11@netflix.com': true} where s_id = 447a8186-e610-11eb-ba80-0242ac130004;
... insert into logs (u_id,createdat,value,activityrelated,type) values ('eleven11@netflix.com',toTimestamp(now()),'poll',false,'story');
... APPLY BATCH;
cqlsh:instagram>

```

Questions:

```

cqlsh:instagram> BEGIN BATCH
... update story set questionsResponses = questionsResponses + {'eleven11@netflix.com': 'How do you felt at the start of season 3 ?'} where s_id = 01acd190-e612-11eb-ba80-0242ac130004;
... insert into logs (u_id,createdat,value,activityrelated,type) values ('eleven11@netflix.com',toTimestamp(now()),'question',false,'story');
... APPLY BATCH;
cqlsh:instagram>

```

Emoji Slider:

```

cqlsh:instagram> BEGIN BATCH
... update story set EmojiSliderResponses = EmojiSliderResponses + {'eleven11@netflix.com': 90} where s_id = 01acd190-e612-11eb-ba80-0242ac130004;
... insert into logs (u_id,createdat,value,activityrelated,type) values ('eleven11@netflix.com',toTimestamp(now()),'emoji',false,'story');
... APPLY BATCH;
cqlsh:instagram>

```

Quiz:

```

cqlsh:instagram> BEGIN BATCH
... update story set quizResponses = quizResponses + {'eleven11@gmail.com': 'Millie Bobby Brown'} where s_id = 447a8186-e610-11eb-ba80-0242ac130004;
... insert into logs (u_id,createdat,value,activityrelated,type) values ('eleven11@gmail.com',toTimestamp(now()),'quiz',false,'story');
... APPLY BATCH;
cqlsh:instagram>

```