

# Advanced Database Design and Implementation

## Assignment 5

Boxiong TAN  
300300835

## Question 2

```
CREATE KEYSPACE S
```

```
WITH REPLICATION = {'class': 'NetworkTopologyStrategy', 'DC1': 3, 'DC2': 3};
```

```
CREATE TABLE S.user (  
    email_address varchar primary key,  
    password varchar  
);
```

```
CREATE TABLE S.vehicles (  
    email_address varchar,  
    vehicle_id varchar,  
    colour varchar,  
    type varchar,  
    PRIMARY KEY(email_address, vehicle_id)  
);
```

```
CREATE TABLE S.datapoint (  
    sequence timestamp,  
    email_address varchar,  
    vehicle_id varchar,  
    latitude double,  
    longitude double,  
    speed double,  
    primary key((email_address, vehicle_id), sequence)  
) WITH CLUSTERING ORDER BY(sequence ASC);
```

```
CREATE TABLE S.vehicle_count (  
    email_address varchar,  
    number_of_vehicle counter,  
    primary key(email_address)  
);
```

```
CREATE TABLE S.user_activate_count (  
    email_address varchar,  
    date Timestamp,  
    number_of_datapoint counter,  
    primary key(email_address, date)  
);
```

```
CREATE TABLE S.user_activate_vehicle (  
    email_address varchar,  
    date Timestamp,
```

```
vehicle_id varchar,  
count counter,  
primary key((email_address, vehicle_id), date));
```

```
CREATE TABLE S.vehicle_recent_location (  
    email_address varchar,  
    sequence timestamp,  
    vehicle_id varchar,  
    latitude double,  
    longitude double,  
    speed double,  
    primary key(email_address, vehicle_id)  
);
```

Question 1 & Question 3 & Question 4

Read:

1.

Retrieve password when user login

Example: email\_address = 'fred@fred.com'

Require strong consistency

CQL:

```
consistency QUORUM;  
select password from S.user  
where email_address='fred@fred.com';
```

Result:

```
password  
-----  
1234
```

(1 rows)

2.

When user want to see information about the vehicles they have registered. The list of the vehicles they have registered.

Require strong consistency

CQL:

```
consistency QUORUM;
```

```
select vehicle_id from vehicles
where email_address='fred@fred.com';
```

```
vehicle_id
-----
      abc123
      def123
```

(2 rows)

3.

When user want to check vehicle the most recent location and speed for a given day.

Timestamp will be read.

Latitude and longitude, which will be used for calculating location.

Speed information will be read as well.

In Question 2, we create a table named `vehicle_recent_location`, each time we add a datapoint record, this table will be update as well. So we only need to retrieve the information we need.

Require low consistency.

CQL([fred@fred.com](mailto:fred@fred.com), vehicle\_id='abc123', 2014-05-26) :

consistency one;

```
select latitude, longitude, speed from vehicle_recent_location
where email_address='fred@fred.com'
and vehicle_id='abc123'
and date='2014-05-26';
```

result:

```
latitude | longitude | speed
-----+-----+-----
      -41.3 | 175.95 |      34
```

(1 rows)

4

For each vehicle a track of locations for a specified time span for a given

day.

Timestamp, Latitude and longitude will be read.

Require low consistency.

CQL:

consistency ONE;

select sequence, latitude, longitude from datapoint

where email\_address='jane@jane.com' and vehicle\_id='jkl123' and

sequence >= '2014-05-26 10:30:00+1200' and sequence < '2014-05-26 10:40:00+1200';

Result:

sequence	latitude   longitude
2014-05-26 10:32:10+1200	-40.118   174.8
2014-05-26 10:32:40+1200	-39.937   174.93
2014-05-26 10:33:10+1200	-39.325   174.95
2014-05-26 10:33:40+1200	-39.015   175.83
2014-05-26 10:34:10+1200	-41.3   175.86
2014-05-26 10:34:40+1200	-41.099   176.43
2014-05-26 10:35:10+1200	-40.164   176.61
2014-05-26 10:35:40+1200	-39.696   176.81
2014-05-26 10:36:10+1200	-41.3   174.8
2014-05-26 10:36:40+1200	-40.686   175.49
2014-05-26 10:37:10+1200	-39.737   176.34
2014-05-26 10:37:40+1200	-39.357   176.91
2014-05-26 10:38:10+1200	-39.074   174.8
2014-05-26 10:38:40+1200	-39.041   175.62
2014-05-26 10:39:10+1200	-41.3   176.31
2014-05-26 10:39:40+1200	-40.565   176.97

(16 rows)

5.

For each user, the number of vehicles they have registered. In Question2, we have created a table for counting each user's vehicles named vehicle\_count. We could retrieve information from that table.

Require a strong consistency.

CQL: (for a particular user: [fred@fred.com](mailto:fred@fred.com))

consistency QUORUM;

select number\_of\_vehicle from vehicle\_count

where email\_address='fred@fred.com';

Result:

number\_of\_vehicle

-----

2

(1 rows)

6.

For each user, the number of data points they have recorded each day.

In Question2, we create a table for counting user's activate number named user\_activate\_count.

Require eventual consistency.

CQL: (See all users)

consistency ONE;

select \* from user\_activate\_count;

Result:

email\_address | date | number\_of\_datapoint

-----+-----+-----

fred@fred.com | 2014-05-26 00:00:00+1200 | 4

jane@jane.com | 2014-05-26 00:00:00+1200 | 35

(2 rows)

7.

For each vehicle a list of the days it has been active.

In Question2, we have created a table for this requirement.

Require strong consistency.

CQL:(for a particular user and a particular car: [fred@fred.com](mailto:fred@fred.com), vehicle\_id='def123')

consistency QUORUM;

select date from user\_activate\_vehicle

where email\_address='fred@fred.com' and vehicle\_id='def123';

Result:

date

-----

2014-05-26 00:00:00+1200

(1 rows)

Write:

1.

Create personal information.

Require strong consistency.

CQL:

consistency LOCAL\_QUORUM;

INSERT INTO S.user (

email\_address,

password

)

values (

'fred@fred.com',

'1234'

);

2.

Register vehicles.

The count of vehicle increment at the same time.

Require strong consistency

CQL:

consistency LOCAL\_QUORUM;

INSERT INTO S.vehicles (

email\_address,

vehicle\_id,

colour,

type

)

values (

'fred@fred.com',

'abc123',

'red',

'sedan'

);

UPDATE S.vehicle\_count

SET

number\_of\_vehicle = number\_of\_vehicle + 1

WHERE email\_address='fred@fred.com';

3.

Each time the phone is in the vehicle, the application will collect Data Points and send it to the server. Then the server write these information into the database.

Require eventual consistency

consistency ONE;

INSERT INTO S.datapoint (

sequence,

email\_address,

vehicle\_id,

latitude,

longitude,

speed

)

values(

'2014-05-26 10:49:10+1200',

'jane@jane.com',

'jkl123',

-40.66,

176.5,

34

);

UPDATE S.user\_activate\_count

SET

number\_of\_datapoint = number\_of\_datapoint + 1

WHERE email\_address='jane@jane.com' and date = '2014-05-26';

UPDATE S.user\_activate\_vehicle

SET count = count + 1

WHERE email\_address='jane@jane.com' and vehicle\_id = 'jkl123' and date='2014-05-26';

UPDATE S.vehicle\_recent\_location



SET latitude = -40.66, longitude = 176.5, speed = 34  
WHERE email\_address='jane@jane.com' and vehicle\_id = 'jkl123' and date='2014-05-26';

Question 5:

If 2 nodes are down.

Using nodetool:

Datacenter: DC1

=====

Status=Up/Down

|/ State=Normal/Leaving/Joining/Moving

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UN	127.0.0.1	83.1 KB	256	50.0%	8e7eae54-9baf-4fb6-b273-d60f87cae076	RAC1
DN	127.0.0.2	83.32 KB	256	50.0%	79842aa3-8829-4a69-b88d-d7907c0a1c54	RAC1
DN	127.0.0.3	83.54 KB	256	50.0%	10901ca7-1030-4ad6-8a89-dc45bbe2cc04	RAC1
UN	127.0.0.4	83.27 KB	256	50.0%	1571aca4-fc87-42e1-9f5a-19a8121707cf	RAC1
UN	127.0.0.5	83.36 KB	256	50.0%	25eb47e2-d109-44a0-950b-67158b968c81	RAC1
UN	127.0.0.6	83.31 KB	256	50.0%	4bf6fa39-3de6-4e90-8e41-a958bbfe503e	RAC1

Example1(Succeed):

consistency QUORUM;

INSERT INTO S.user (

email\_address,

password

)

values (

'may@may.com',

'9999'

);

consistency QUORUM;

select \* from S.user;

Feedback:

Consistency level set to QUORUM.

Consistency level set to QUORUM.

email\_address | password

-----+-----

must@must.com | 9999

(1 rows)

```

Example 2(Fails):
consistency ALL;
INSERT INTO S.user (
    email_address,
    password
)
values (
    'monk@monk.com',
    '9988'
);
select * from S.user;
Feedback:

```

Consistency level set to ALL.  
 Unable to complete request: one or more nodes were unavailable.

Question 6:

nodetool:

Datacenter: DC1

=====

Status=Up/Down

|/ State=Normal/Leaving/Joining/Moving

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UN	127.0.0.1	123.52 KB	256	0.0%	44e3aac0-9417-42db-a921-9dfca28a87df	RAC1
DN	127.0.0.2	105.2 KB	256	0.0%	2232bbda-de44-43fb-8e99-6a734bc17741	RAC1
UN	127.0.0.3	121.61 KB	256	0.0%	ceb6ee9c-7b6a-444a-9002-b607001989fb	RAC1
UN	127.0.0.4	121.56 KB	256	0.0%	69bedcc0-32fd-485c-a64e-db18cc8978b6	RAC1
UN	127.0.0.5	123.61 KB	256	0.0%	50293b45-2101-4d40-96f3-ec02950fdd81	RAC1
UN	127.0.0.6	116.65 KB	256	0.0%	eb656e29-d924-4032-a7a5-9cd9fe0a0d9f	RAC1

Success

consistency any;

```

insert into S.vehicles (

```

```

    email_address,
    vehicle_id,
    colour,
    type

```

```

)

```

```

values (

```

```

    'bob@bob.com',
    'bob123',

```

```
'fire',  
'truck'  
);
```

consistency one;

```
select * from S.vehicles
```

```
where email_address='bob@bob.com' and vehicle_id='bob123';
```

Feedback:

Consistency level set to ANY.

Consistency level set to ONE.

```
email_address | vehicle_id | colour | type
```

```
-----+-----+-----+-----
```

```
bob@bob.com |    bob123 |  fire | truck
```

(1 rows)

Question7:

There should be 2 AWS Availability Zones. Each zones has 3 nodes.