FIT5137 Group Assignment Report

Semester 2, 2019

Students Name: Dawei Gu, Jiexuan Du

Students ID: 29910226, 29218020

Tutorial Section: FIT5137 Laboratory 05,
12:00m-14:00pm Tue

Tutor: Agnes Haryanto

Contents

Signed Contribution Declaration Form

Contribution Declaration Form
(to be completed by all team members)

Please fill in the form with the contribution from each student towards the assignment.

1 NAME AND CONTRIBUTION DETAILS

Student ID	Student Name	Contribution Percentage
29910226	Dawei Gu	50%
29218020	Jiexuan Du	50%

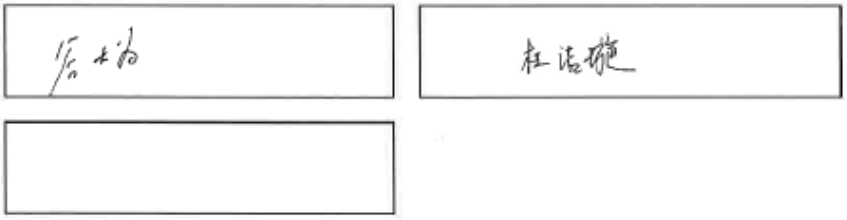
2 DECLARATION

We declare that:

- The information we have supplied in or with this form is complete and correct.
- We understand that the information we have provided in this form will be used for individual assessment of the assignment.

3 SIGNATURE

Signatures



Date

Day Month Year

15 / 9 / 2019

Contribution Declaration Detail

Percentage of contribution:

- Name: Dawei Gu, ID: 29910226, Contribution: 50%
- Name: Jiexuan Du, ID: 29218020, Contribution: 50%

List of parts that each student did:

I. Dawei Gu: list the parts that Dawei did:

- Task C.1 (MongoDB Host data create collections and insert data, Cassandra create table and insert data)
- Task C.2 (1, 3, 5, 7)
- Task C.3 (1, 3, 5, 7, 9, 11, 13, 15, 17, 19, Q1, Q2, Q4)
- Task C.4

□ Task C.3

II. Jiexuan Du: list the parts that Jiexuan did:

- Task C.1 (MongoDB listing data create collections and insert data Cassandra create table)
- Task C.2 (2, 4, 6)
- Task C.3 (2, 4, 6, 8, 10, 12, 14, 16, 18, 20, Q3, Q5)
- Task C.4

Report

C.1 Database Design

MongoDB:

```
# create database
use FIT5137_Assign

#create collection host
db.createCollection('Host')

#insert data to host
db.Host.insertMany([
{'host_id': 'MONHOS01', 'host_name':'Manju',
'host_vertifications':['email', 'phone', 'reviews'],
'host_since': new Date('2009-08-21'),
 'host_location': {'city': 'Clayton', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within a day', 'host_is_superhost':false},
{'host_id': 'MONHOS02', 'host_name':'Lindsay',
 'host_vertifications':['email', 'phone', 'reviews', 'jumio', 'government id'],
 'host_since': new Date('2009-09-16'),
 'host_location': {'city': 'Clifton Hill', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within an hour', 'host_is_superhost':true},
{'host_id': 'MONHOS03', 'host_name':'Adam',
 'host_vertifications':['email', 'phone', 'google', 'reviews', 'jumio', 'government id', 'work
email'],
 'host_since': new Date('2009-10-31'),
 'host_location': {'city': 'Port Melbourne', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within an hour', 'host_is_superhost':false},
{'host_id': 'MONHOS04', 'host_name':'Eleni',
 'host_vertifications':['email', 'phone', 'facebook', 'reviews', 'jumio', 'offline government
id', 'government id', 'work email'],
 'host_since': new Date('2009-12-03'),
 'host_location': {'city': 'Fitzroy', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within a day', 'host_is_superhost':false},
{'host_id': 'MONHOS05', 'host_name':'Colin',
 'host_vertifications':['email', 'phone', 'facebook', 'reviews', 'jumio', 'offline government
id', 'government id'],
 'host_since': new Date('2009-12-22'),
 'host_location': {'city': 'Saint Kilda East', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within an hour', 'host_is_superhost':false},
{'host_id': 'MONHOS06', 'host_name':'Daryl',
 'host_vertifications':['email', 'phone', 'manual online', 'reviews', 'manual offline', 'work
```

```
email'],
 'host_since': new Date('2010-07-12'),
'host_location': {'city': 'Berwick', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within an hour', 'host_is_superhost':true},
{'host_id': 'MONHOS07', 'host_name':'Diana',
 'host_vertifications':['email', 'phone', 'facebook', 'reviews', 'jumio', 'offline government
id', 'government id', 'work email'],
 'host_since': new Date('2010-07-27'),
 'host_location': {'city': 'Parkdale', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within a day', 'host_is_superhost':false},
{'host_id': 'MONHOS08', 'host_name':'Belinda',
 'host_vertifications':['email', 'phone', 'facebook', 'reviews', 'jumio', 'offline government
id', 'selfie', 'government id', 'identity manual', 'work email'],
 'host_since': new Date('2010-08-03'),
 'host_location': {'city': 'Prahran', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within a few hours', 'host_is_superhost':false},
{'host_id': 'MONHOS09', 'host_name':'Allan',
 'host_vertifications':['email', 'phone', 'facebook', 'reviews', 'jumio', 'offline government
id', 'selfie', 'government id', 'identity manual'],
 'host_since': new Date('2010-08-03'),
 'host_location': {'city': 'South Melbourne', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within an hour', 'host_is_superhost':true},
{'host_id': 'MONHOS10', 'host_name':'Vicki',
 'host_vertifications':['email', 'phone', 'facebook', 'reviews', 'jumio', 'government id'],
 'host_since': new Date('2010-08-06'),
 'host_location': {'city': 'Frankston', 'state':'Victoria', 'country':'Australia'},
 'host_response_rate':'within an hour', 'host_is_superhost':true}
])

#create collection listing
db.createCollection('Listing')

#insert data into listing
db.Listing.insertMany([
{'listing_id':'MONLST01', 'name':'Monash Beautiful House',
 'host_id':'MONHOS14', 'neighbourhood':'Manningham',
 'address':{'city':'Clayton', 'state':'VIC', 'post_code':3800},
 'latitude': -37.773, 'longitude':145.09213, 'room_type':'Entire home',
 'amenities':['TV', 'Wifi', 'Pets Allowed', 'Family friendly', '24-hour check-in', 'Self check-
in'],
 'price':61, 'extra_people':22, 'minimum_nights':1, 'availability_365':365},
{'listing_id':'MONLST02', 'name':'Monash Brunswick Deco',
 'host_id':'MONHOS08', 'neighbourhood':'Moreland',
```

'address':{'city':'Brunswick East', 'state':'VIC', 'post_code':3057},
'latitude': -37.767, 'longitude':144.98074, 'room_type':'Private room',
'amenities':['Kitchen', 'Toiletries', 'Hair dryer', 'Iron', 'Microwave', 'Coffee Maker', 'Refrigerator', 'Cooking basics', 'Stove', 'Garden'],
'price':35, 'extra_people':15, 'minimum_nights':3, 'availability_365':194},
{'listing_id':'MONLST03', 'name':'Monash Beachside Retreat',
'host_id':'MONHOS01', 'neighbourhood':'Port Phillip',
'address':{'city':'St Kilda', 'state':'VIC', 'post_code':3182},
'latitude': -37.86, 'longitude':144.97737, 'room_type':'Entire home',
'amenities':['Cooking basics', 'Oven', 'Stove', 'Dishwasher', 'Balcony', 'Wifi', 'Kitchen', 'Washer', 'Toiletries', 'Hair dryer', 'Iron', 'Microwave', 'Coffee Maker', 'Refrigerator', 'Long term stay allowed', 'Luggage dropoff', '24-hour check-in', 'Self check-in'],
'price':159, 'extra_people':0, 'minimum_nights':2, 'availability_365':82},
{'listing_id':'MONLST04', 'name':'Monash Close2City',
'host_id':'MONHOS04', 'neighbourhood':'Darebin',
'address':{'city':'Thornbury', 'state':'VIC', 'post_code':3071},
'latitude': -37.759, 'longitude':144.98923, 'room_type':'Private room',
'amenities':['TV', 'Internet', 'Wifi', 'Kitchen', 'Free parking on premises', 'Breakfast', 'Heating', 'Washer', 'Toiletries', 'Hair dryer'],
'price':50, 'extra_people':20, 'minimum_nights':2, 'availability_365':0},
{'listing_id':'MONLST05', 'name':'Monash City and Sports',
'host_id':'MONHOS05', 'neighbourhood':'Port Phillip',
'address':{'city':'St Kilda East', 'state':'VIC', 'post_code':3183},
'latitude': -37.865, 'longitude':144.99224, 'room_type':'Private room',
'amenities':['Internet', 'Wifi', 'Heating', 'Washer', 'Toiletries', 'Bed linens', 'Gym Access', '24-hour check-in', 'Self check-in'],
'price':69, 'extra_people':20, 'minimum_nights':1, 'availability_365':274},
{'listing_id':'MONLST06', 'name':'Monash Trafford Apartment',
'host_id':'MONHOS06', 'neighbourhood':'Casey',
'address':{'city':'Berwick', 'state':'VIC', 'post_code':3806},
'latitude': -38.057, 'longitude':145.33936, 'room_type':'Entire home',
'amenities':['Microwave', 'Coffee Maker', 'Wifi', 'Kitchen', 'Toiletries', 'Hair dryer'],
'price':99, 'extra_people':30, 'minimum_nights':1, 'availability_365':353},
{'listing_id':'MONLST07', 'name':'Monash Close2Airport',
'host_id':'MONHOS07', 'neighbourhood':'Darebin',
'address':{'city':'Reservoir', 'state':'VIC', 'post_code':3073},
'latitude': -37.697, 'longitude':145.00082, 'room_type':'Private room',
'amenities':['Long term stay allowed', 'Luggage dropoff', '24-hour check-in', 'Self check-in', 'Airport shuttle service', 'Host greets you', 'Wifi', 'Kitchen', 'Heating', 'Toiletries', 'Microwave', 'Coffee Maker', 'Refrigerator', 'Cooking basics', 'Oven', 'Stove', 'Dishwasher', 'Garden'],
'price':50, 'extra_people':20, 'minimum_nights':7, 'availability_365':0},
{'listing_id':'MONLST08', 'name':'Monash Home In The City',

'host_id':'MONHOS02', 'neighbourhood':'Melbourne',
'address':{'city':'East Melbourne', 'state':'VIC', 'post_code':3002},
'latitude': -37.81, 'longitude':144.98592, 'room_type':'Private room',
'amenities':['Cooking basics', 'TV', 'Wifi', 'Pets Allowed', 'Family friendly', '24-hour check-in', 'Self check-in'],
'price':99, 'extra_people':25, 'minimum_nights':15, 'availability_365':62},
{'listing_id':'MONLST09', 'name':'Monash Japanese-Style',
'host_id':'MONHOS11', 'neighbourhood':'Monash',
'address':{'city':'Oakleigh East', 'state':'VIC', 'post_code':3166},
'latitude': -37.9, 'longitude':145.11447, 'room_type':'Entire home',
'amenities':['Kitchen', 'Toiletries', 'Hair dryer', 'Iron', 'Microwave', 'Garden', 'Coffee Maker', 'Refrigerator', 'Cooking basics', 'Stove', 'Dishwasher', 'Balcony', 'Paid Parking', 'Long term stay allowed'],
'price':98, 'extra_people':0, 'minimum_nights':2, 'availability_365':219},
{'listing_id':'MONLST10', 'name':'Beautiful Monash House',
'host_id':'MONHOS10', 'neighbourhood':'Frankston',
'address':{'city':'Frankston', 'state':'VIC', 'post_code':3199},
'latitude': -38.149, 'longitude':145.14157, 'room_type':'Entire home',
'amenities':['TV', 'Wifi', 'AC', 'Kitchen', 'Heating', 'Washer', 'Toiletries', 'Hair dryer', 'Iron', 'Microwave', 'Coffee Maker', 'Refrigerator', 'Cooking basics', 'Oven', 'Stove', 'Dishwasher', 'Balcony', 'Paid Parking', 'Long term stay allowed', 'Luggage dropoff', '24-hour check-in', 'Self check-in'],
'price':59, 'extra_people':10, 'minimum_nights':2, 'availability_365':318},
{'listing_id':'MONLST11', 'name':'Fabulous Monash Richmond',
'host_id':'MONHOS09', 'neighbourhood':'Yarra',
'address':{'city':'Richmond,', 'state':'VIC', 'post_code':3121},
'latitude': -37.818, 'longitude':145.00442, 'room_type':'Entire home',
'amenities':['Bed linen', 'Wifi', 'AC', 'Kitchen', 'Heating', 'Washer', 'Toiletries', 'Hair dryer', 'Iron', 'Microwave', 'Coffee Maker', 'Free Parking', 'Long term stay allowed', 'Luggage dropoff', '24-hour check-in', 'Self check-in', 'Host greets you', 'Refrigerator', 'Cooking basics', 'Oven', 'Stove', 'Dishwasher', 'Balcony', 'Garden'],
'price':98, 'extra_people':30, 'minimum_nights':14, 'availability_365':16},
{'listing_id':'MONLST12', 'name':'Monash Central Lux',
'host_id':'MONHOS12', 'neighbourhood':'Port Phillip',
'address':{'city':'St Kilda', 'state':'VIC', 'post_code':3182},
'latitude': -37.861, 'longitude':144.98038, 'room_type':'Entire home',
'amenities':['Wifi', 'AC', 'Kitchen', 'Heating', 'Toiletries', 'Hair dryer', 'Iron', 'Coffee Maker', 'Refrigerator', 'Stove', 'Dishwasher', 'Beachside view', 'Free Parking 24-hour check-in', 'Self check-in'],
'price':189, 'extra_people':29, 'minimum_nights':2, 'availability_365':6},
{'listing_id':'MONLST13', 'name':'Central Monash Warehouse Apartment',
'host_id':'MONHOS13', 'neighbourhood':'Melbourne',
'address':{'city':'Melbourne', 'state':'VIC', 'post_code':3000},

'latitude': -37.815, 'longitude':144.96267, 'room_type':'Entire home',
'amenities':['Dishwasher', 'Garden', 'Paid Parking', 'Long term stay allowed', 'Coffee Maker', 'Refrigerator', 'Cooking basics', 'Oven', 'Stove', 'Wifi', 'AC', 'Kitchen', 'Heating', 'Washer', 'Toiletries', 'Hair dryer', 'Iron', 'Microwave', 'Luggage dropoff', '24-hour check-in', 'Self check-in'],
'price':249, 'extra_people':40, 'minimum_nights':2, 'availability_365':353},
{'listing_id':'MONLST14', 'name':'Monash Near the Park',
'host_id':'MONHOS03', 'neighbourhood':'Bayside',
'address':{'city':'Melbourne', 'state':'VIC', 'post_code':3187},
'latitude': -37.928, 'longitude':145.02518, 'room_type':'Private room',
'amenities':['Pets Allowed', 'Breakfast', 'Garden', 'Family friendly', '24-hour check-in', 'Self check-in'],
'price':40, 'extra_people':11, 'minimum_nights':2, 'availability_365':365}
])

Cassandra:

```
# create keyspace
CREATE KEYSPACE FIT5137_Assign with
replication = {'class':'SimpleStrategy', 'replication_factor':1};

#swich to keyspace
USE FIT5137_Assign;

#create table review
CREATE TABLE review (listing_id  text,
                     review_id  text,
                     review_date date,
                     sequence  time,
                     reviewer_id  text,
                     reviewer_name  text,
                     rating  int,
                     reason  set<text>,
                     comments text,
                     PRIMARY KEY (review_id));

#create second index
CREATE INDEX ON review (reviewer_id );

#insert data into reiview
BEGIN BATCH
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
```

```
        VALUES('MONLST02', 'REV01', '2017-03-22', '10:37:50', '500001', 'Miriam',
90, {'location', 'amenities'}, 'Beautiful View');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST02', 'REV02', '2017-03-22', '11:37:50', '500002', 'Johannes',
90, {'host'}, 'Good Host');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST02', 'REV03', '2017-03-22', '11:37:50', '500003', 'Camille',
100, {'location', 'view'}, 'Nice View and Location');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST02', 'REV04', '2017-03-22', '12:37:50', '500004', 'Paige', 95,
{'price'}, 'Excellent Price');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST01', 'REV05', '2017-03-22', '15:37:50', '500005', 'Adele', 93,
{'location', 'price'}, 'Good Location');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST03', 'REV06', '2017-03-22', '17:37:50', '500006', 'Greg', 87,
{'host', 'view'}, 'Very Clean House');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST04', 'REV07', '2017-03-22', '19:37:50', '500007', 'Wolfgang',
91, {'location', 'price'}, 'Nice Location');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST05', 'REV08', '2017-03-22', '20:37:50', '500008', 'Klaus', 96,
{'location', 'view'}, 'Nice Building');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST06', 'REV09', '2017-03-23', '11:37:50', '500009', 'Rox', 100,
{'host', 'price'}, 'Friendly Host');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST05', 'REV10', '2017-03-23', '12:37:50', '500010', 'Elisabeth',
98, {'host', 'price'}, 'Friendly Host');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST09', 'REV11', '2017-03-23', '19:37:50', '500011', 'Derek',
100, {'space', 'clean'}, 'Very Clean and comfortable');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
```

```
reviewer_name, rating, reason, comments)
        VALUES('MONLST09', 'REV12', '2017-03-25', '10:07:40', '500012', 'Joy', 92,
{'host', 'clean'}, 'Friendly and Nice Host');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST10', 'REV13', '2017-03-26', '10:02:10', '500013', 'Anouck',
93, {'host', 'view'}, 'Very Comfortable');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST12', 'REV14', '2017-03-26', '10:49:40', '500014', 'Jerome',
85, {'location', 'clean'}, 'Friendly Host');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST13', 'REV15', '2017-03-26', '10:48:40', '500015', 'Jehan', 98,
{'location', 'amenities'}, 'Beautiful View');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST14', 'REV16', '2017-03-26', '10:48:10', '500012', 'Joy', 97,
{'amenities', 'view'}, 'Good Location');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST14', 'REV17', '2017-03-26', '10:47:40', '500014', 'Jerome',
30, {'price', 'view'}, 'Bad Location');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST10', 'REV18', '2017-03-26', '10:47:10', '500002', 'Johannes',
20, {'amenities', 'view'}, 'Bad Service');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST12', 'REV19', '2017-03-27', '09:37:50', '500013', 'Anouck',
87, {'space', 'clean'}, 'Good Location');
INSERT INTO review(listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, rating, reason, comments)
        VALUES('MONLST05', 'REV20', '2017-03-27', '10:45:10', '500011', 'Derek', 96,
{'host', 'view'}, 'Nice Building');
APPLY BATCH;
```

C.2 Database Modifications

MonogoDB:

1.

Code:

db.Host.update({'host_name':'Adam'}, {$push:{'host_vertifications': 'facebook'}})

2.

Code:

db.Host.insertMany([
{'host_id': 'MONHOS11', 'host_name':'Alison',
'host_vertifications':['email', 'phone','facebook', 'reviews'],
'host_since': new Date('2019-01-9'),
'host_location': {'city': 'Caulfield', 'state':'Victoria', 'country':'Australia'},
'host_response_rate':'within an hour', 'host_is_superhost':false},
{'host_id': 'MONHOS12', 'host_name':'Mike',
'host_vertifications':['email', 'phone'],
'host_since': new Date('2019-01-9'),
'host_location': {'city': 'Clayton', 'state':'Victoria', 'country':'Australia'},
'host_response_rate':'within a day', 'host_is_superhost':true},
{'host_id': 'MONHOS13', 'host_name':'Robyn',
'host_vertifications':['facebook', 'reviews'],
'host_since': new Date('2019-01-9'),
'host_location': {'city': 'Berwick', 'state':'Victoria', 'country':'Australia'},
'host_response_rate':'within an hour', 'host_is_superhost':false},
{'host_id': 'MONHOS14', 'host_name':'Daniel',
'host_vertifications':['email', 'manual offline','work email'],
'host_since': new Date('2019-01-9'),
'host_location': {'city': 'Frankston', 'state':'Victoria', 'country':'Australia'},

'host_response_rate':'within a day', 'host_is_superhost':true},
{'host_id': 'MONHOS15', 'host_name':'Ron',
'host_vertifications':['facebook'],
'host_since': new Date('2019-01-9'),
'host_location': {'city': 'Caulfield', 'state':'Victoria', 'country':'Australia'},
'host_response_rate':'within a day', 'host_is_superhost':false}
])

```
...   'host_vertifications':['facebook'],
...   'host_since': new Date('2019-01-9'),
...   'host_location': {'city': 'Caulfield', 'state':'Victori
...   'host_response_rate':'within a day',
...   'host_is_superhost':false}
... ])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5d7b18313208cd71fae81dec"),
                ObjectId("5d7b18313208cd71fae81ded"),
                ObjectId("5d7b18313208cd71fae81dee"),
                ObjectId("5d7b18313208cd71fae81def"),
                ObjectId("5d7b18313208cd71fae81df0")
        ]
}
>
```

3.

Code:

db.Host.updateMany({'host_response_rate':'within an hour'}, {$set:{'host_is_superhost': true}})

```
}
> db.Host.updateMany({'host_response_rate':'within an hour'}, {$set:{'host_is_superhost': true}})
{ "acknowledged" : true, "matchedCount" : 8, "modifiedCount" : 4 }
>
```

4.

Code:

db.Listing.deleteMany({'availability_365':0})

```
> db.Listing.deleteMany({'availability_365':0})
{ "acknowledged" : true, "deletedCount" : 2 }
>
```

5.

Code:

db.Listing.updateMany({'neighbourhood':'Monash'}, {$set:{'neighbourhood':'Monash City'}})

```
> db.Listing.updateMany({'neighbourhood':'Monash'}, {$set:{'neighbourhood':'Monash City'}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
```

Cassandra:

6.

Code:

```
UPDATE review SET reason = {'space', 'price'} WHERE review_id = 'REV11';
```

```
cqlsh:fit5137_assign> SELECT reason FROM review WHERE review_id = 'REV11';

 reason
--------------------
 {'clean', 'space'}

(1 rows)
cqlsh:fit5137_assign> UPDATE review SET reason = {'space', 'price'} WHERE review_id = 'REV11';
cqlsh:fit5137_assign> SELECT reason FROM review WHERE review_id = 'REV11';

 reason
--------------------
 {'price', 'space'}

(1 rows)
cqlsh:fit5137_assign>
```

7.

Code:

```
SELECT review_id FROM review WHERE reviewer_id='500015';

DELETE FROM review WHERE review_id = 'REV15';
```

```
cqlsh:fit5137_assign> SELECT review_id FROM review WHERE reviewer_id='500015';

 review_id
-----------
     REV15

(1 rows)
cqlsh:fit5137_assign> DELETE FROM review WHERE  review_id = 'REV15';
cqlsh:fit5137_assign> select * from review where reviewer_id ='500015';

 review_id | comments | listing_id | rating | reason | review_date | reviewer_id | reviewer_name | sequence
-----------+----------+------------+--------+--------+-------------+-------------+---------------+----------

(0 rows)
cqlsh:fit5137_assign>
```

C.3 Queries

MongoDB

```
#preparing

#create database
use FIT5137_Assign_C3

#quit the mongoDB
quit()

#insert data into Host
mongoimport --db FIT5137_Assign_C3 --collection Host --file D:\host.json

#insert data into Listing
mongoimport --db FIT5137_Assign_C3 --collection Listing --file D:\listing.json

#run shall
mongod

#enter shall
mongo

#enter database
use FIT5137_Assign_C3

#create embedding model collection called Listing_host
db.Listing.aggregate([
        {$lookup: {from:'Host',
                localField:'host_id',
                foreignField:'host_id',
                as: 'host'}},
        {$out: ''Listing_host''}
])

#create index for Host collection
db.Host.createIndex({'host_name':1})
db.Host.createIndex({'host_id':1})
```

```
> db.Host.createIndex({'host_name':1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.Host.createIndex({'host_id':1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "ok" : 1
}
```

#create index for Listing collection

db.Listing.createIndex({'price':1})

db.Listing.createIndex({'number_of_reviews':1})

db.Listing.createIndex({'neighbourhood ':1})

```
> db.Listing.createIndex({'price':1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.Listing.createIndex({'number_of_reviews':1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "ok" : 1
}
> db.Listing.createIndex({'neighbourhood ':1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 3,
        "numIndexesAfter" : 4,
        "ok" : 1
}
```

#create index for Listing_host collection

db.Listing_host.createIndex({'price':1})

db.Listing_host.createIndex({'host.host_name':1})

db.Listing_host.createIndex({'number_of_reviews ':1})

```
> db.Listing_host.createIndex({'price':1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.Listing_host.createIndex({'host.host_name':1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "ok" : 1
}
> db.Listing_host.createIndex({'number_of_reviews ':1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 3,
        "numIndexesAfter" : 4,
        "ok" : 1
}
```

1.

Referencing code:

```
db.Listing.aggregate([
{$match: {"last_review": {"$gte": new Date('2018-12-01'), "$lte":
            new Date('2018-12-31')}}},
{$group:{'_id':null, 'total':{$sum:1}}}
])
```

```
> db.Listing.aggregate([
... {$match: {"last_review": {"$gte": new Date('2018-12-01'), "$lte": new Date('2018-12-31')}}},
... {$group:{'_id':null, 'total':{$sum:1}}}
... ])
{ "_id" : null, "total" : 2 }
```

Embedding code:

```
db.Listing_host.aggregate([
{$match: {"last_review": {"$gte": new Date('2018-12-01'), "$lte": new Date('2018-12-31')}}},
{$group:{'_id':null, 'total':{$sum:1}}}
])
```

```
> db.Listing_host.aggregate([
... {$match: {"last_review": {"$gte": new Date('2018-12-01'), "$lte": new Date('2018-12-31')}}},
... {$group:{'_id':null, 'total':{$sum:1}}}
... ])
{ "_id" : null, "total" : 2 }
```

2.

Referencing code:

```
db.Listing.aggregate([
   { "$match": {"neighbourhood":"Port Phillip"}},
   { "$group": {"_id": null, "averageprice": { "$avg": "$price" }}}
])
```

```
> db.Listing.aggregate([
...     { "$match": {"neighbourhood":"Port Phillip"}},
...     { "$group": {"_id": null, "averageprice": { "$avg": "$price" }}}
... ])
{ "_id" : null, "averageprice" : 134.5 }
>
```

Embedding code:

```
db.Listing_host.aggregate([
   { "$match": {"neighbourhood":"Port Phillip"}},
   { "$group": {"_id": null, "averageprice": { "$avg": "$price" }}}
])
```

```
> db.Listing_host.aggregate([
...     { "$match": {"neighbourhood":"Port Phillip"}},
...     { "$group": {"_id": null, "averageprice": { "$avg": "$price" }}}
... ])
{ "_id" : null, "averageprice" : 134.5 }
>
```

3.

Referencing code:

```
db.Listing.aggregate([
{$group:{'_id':'$neighbourhood', 'average':{$avg:'$reviews_per_month'}}},
{ $sort: { 'average': -1 } },
{ $limit: 10 }
])
```

```
> db.Listing.aggregate([
...  {$group:{'_id':'$neighbourhood', 'average':{$avg:'$reviews_per_month'}}},
...  { $sort: { 'average': -1 } },
...  { $limit: 10 }
... ])
{ "_id" : "Stonnington", "average" : 2.1233333333333335 }
{ "_id" : "Brimbank", "average" : 1.9449999999999998 }
{ "_id" : "Yarra", "average" : 1.65375 }
{ "_id" : "Monash", "average" : 1.2850000000000001 }
{ "_id" : "Maribyrnong", "average" : 1.09 }
{ "_id" : "Melbourne", "average" : 1.0510526315789475 }
{ "_id" : "Casey", "average" : 0.905 }
{ "_id" : "Hobsons Bay", "average" : 0.855 }
{ "_id" : "Kingston", "average" : 0.8433333333333334 }
{ "_id" : "Port Phillip", "average" : 0.7835714285714286 }
>
```

Embedding code:

```
db.Listing_host.aggregate([
{$group:{'_id':'$neighbourhood', 'average':{$avg:'$reviews_per_month'}}},
{ $sort: { 'average': -1 } },
{ $limit: 10 }
])
```

```
> db.Listing_host.aggregate([
...  {$group:{'_id':'$neighbourhood', 'average':{$avg:'$reviews_per_month'}}},
...  { $sort: { 'average': -1 } },
...  { $limit: 10 }
... ])
{ "_id" : "Stonnington", "average" : 2.1233333333333335 }
{ "_id" : "Brimbank", "average" : 1.9449999999999998 }
{ "_id" : "Yarra", "average" : 1.65375 }
{ "_id" : "Monash", "average" : 1.2850000000000001 }
{ "_id" : "Maribyrnong", "average" : 1.09 }
{ "_id" : "Melbourne", "average" : 1.0510526315789475 }
{ "_id" : "Casey", "average" : 0.905 }
{ "_id" : "Hobsons Bay", "average" : 0.855 }
{ "_id" : "Kingston", "average" : 0.8433333333333334 }
{ "_id" : "Port Phillip", "average" : 0.7835714285714286 }
>
```

4.

Referencing code:

```
db.Listing.aggregate([
  {"$match": {"street" : {$regex : ".*Melbourne.*"}}},
  { "$group": { "_id": null, "max": { "$max": "$number_of_reviews" }, "min": { "$min":
"$number_of_reviews" } }}
])
```

```
> db.Listing.aggregate([
...     {"$match": {"street" : {$regex : ".*Melbourne.*"}}},
...     { "$group": { "_id": null, "max": { "$max": "$number_of_reviews" }, "min": { "$min": "$number_of_reviews" } }}
... ])
{ "_id" : null, "max" : 397, "min" : 0 }
>
```

Embedding code:

```
db.Listing_host.aggregate([
  {"$match": {"street" : {$regex : ".*Melbourne.*"}}},
  { "$group": { "_id": null, "max": { "$max": "$number_of_reviews" }, "min": { "$min": "$number_of_reviews" } }}
])
```

```
> db.Listing_host.aggregate([
...     {"$match": {"street" : {$regex : ".*Melbourne.*"}}},
...     { "$group": { "_id": null, "max": { "$max": "$number_of_reviews" }, "min": { "$min": "$number_of_reviews" } }}
... ])
{ "_id" : null, "max" : 397, "min" : 0 }
>
```

5.

Referencing code:

```
db.Listing.aggregate([
{$group:{'_id':'$room_type', 'total':{$sum:1}}},
{ $sort: { 'total': -1 } },
{ $limit: 1 }
])
```

```
> db.Listing.aggregate([
...  {$group:{'_id':'$room_type', 'total':{$sum:1}}},
...  { $sort: { 'total': -1 } },
...  { $limit: 1 }
... ])
{ "_id" : "Private room", "total" : 50 }
>
```

Embedding code:

```
db.Listing_host.aggregate([
{$group:{'_id':'$room_type', 'total':{$sum:1}}},
{ $sort: { 'total': -1 } },
{ $limit: 1 }
])
```

```
> db.Listing_host.aggregate([
...  {$group:{'_id':'$room_type', 'total':{$sum:1}}},
...  { $sort: { 'total': -1 } },
...  { $limit: 1 }
... ])
{ "_id" : "Private room", "total" : 50 }
>
```

6.

Referencing code:

db.Listing.find({},{"_id":0,"street":1}).sort({"price":-1}).limit(5)

```
> db.Listing.find({},{"_id":0,"street":1}).sort({"price":-1}).limit(5)
{ "street" : "Southbank, VIC, Australia" }
{ "street" : "Belgrave, VIC, Australia" }
{ "street" : "Caulfield North, VIC, Australia" }
{ "street" : "Southbank, VIC, Australia" }
{ "street" : "Balaclava, VIC, Australia" }
>
```

Embedding code:

db.Listing_host.find({},{"_id":0, "street":1}).sort({"price":-1}).limit(5)

```
> db.Listing_host.find({},{"_id":0, "street":1}).sort({"price":-1}).limit(5)
{ "street" : "Southbank, VIC, Australia" }
{ "street" : "Belgrave, VIC, Australia" }
{ "street" : "Caulfield North, VIC, Australia" }
{ "street" : "Southbank, VIC, Australia" }
{ "street" : "Balaclava, VIC, Australia" }
>
```

7.

Referencing code:

```
db.Listing.aggregate([
{$lookup:
        {from:'Host',
         localField:'host_id',
         foreignField:'host_id',
         as: 'host'}},
{$match:{'host.host_name':'Eleni'}},
{$project:{'name':1, 'host.host_name':1}}
      ])
```

```
> db.Listing.aggregate([
... {$lookup:
...         {from:'Host',
...          localField:'host_id',
...          foreignField:'host_id',
...          as: 'host'}},
... {$match:{'host.host_name':'Eleni'}},
... {$project:{'name':1, 'host.host_name':1}}
...       ])
{ "_id" : ObjectId("5d7b1de66f3f3c6235b0ba39"), "name" : "Charming house inner Melbourne", "host" : [ { "host_name" : "Eleni" } ] }
{ "_id" : ObjectId("5d7b1de66f3f3c6235b0ba3a"), "name" : "Large private room-close to city", "host" : [ { "host_name" : "Eleni" } ] }
>
```

Embedding code:

db.Listing_host.find({'host.host_name':'Eleni'},{'name':1, 'host.host_name':1})

```
> db.Listing_host.find({'host.host_name':'Eleni'},{'name':1, 'host.host_name':1})
{ "_id" : ObjectId("5d7b1de66f3f3c6235b0ba39"), "name" : "Charming house inner Melbourne", "host" : [ { "host_name" : "Eleni" } ] }
{ "_id" : ObjectId("5d7b1de66f3f3c6235b0ba3a"), "name" : "Large private room-close to city", "host" : [ { "host_name" : "Eleni" } ] }
>
```

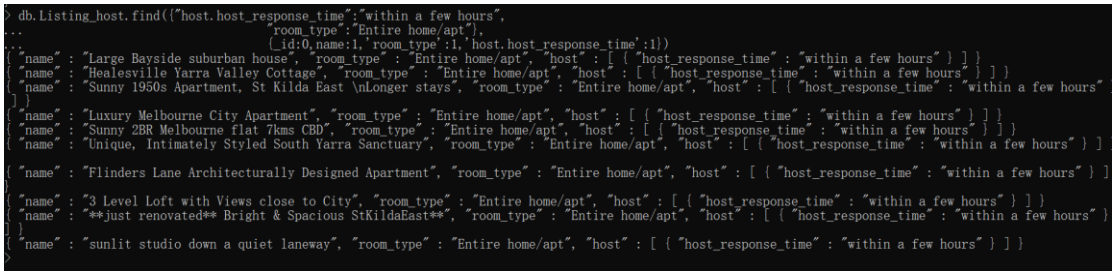
8.

Referencing code:

```
db.Listing.aggregate([
{"$match":{"room_type":"Entire home/apt"}},
{$lookup:{from:'Host',localField:'host_id',foreignField:'host_id',as:'result'}},
{"$match":{"result.host_response_time":"within a few hours"}},
{$project:{_id:0,name:1,'room_type':1,'result.host_response_time':1}}
])
```

Embedding code:

```
db.Listing_host.find({"host.host_response_time":"within a few hours",
                      "room_type":"Entire home/apt"},
                      {_id:0,name:1,'room_type':1,'host.host_response_time':1})
```



9.

Referencing code:

```
db.Listing.aggregate([
{$lookup:
        {from:'Host',
         localField:'host_id',
         foreignField:'host_id',
         as: 'host'}},
{$match:{'host.host_name':'Colin', 'amenities':{$regex:'.*Internet'},
'amenities':{$regex:'.*Gym'}}}]).pretty()
```

```
> db.Listing.aggregate([
... {$lookup:
...               {from:'Host',
...                localField:'host_id',
...                foreignField:'host_id',
...                as: 'host'}},
... {$match:{'host.host_name':'Colin', 'amenities':{$regex:'.*Internet'}, 'amenities':{$regex:'.*Gym'}}}
... ]).pretty()
>
```

Embedding code:

db.Listing_host.find({'host.host_name':'Colin', 'amenities':{$regex:'.*Internet'}, 'amenities':{$regex:'.*Gym'}}).pretty()

```
> db.Listing_host.find({'host.host_name':'Colin', 'amenities':{$regex:'.*Internet'}, 'amenities':{$regex:'.*Gym'}}).pretty()
>
```

10.

Referencing code:

db.Listing.find({"name" : {$regex : ".*Beautiful.*"},"street" : {$regex : ".*Clayton.*"}},{"room_type":1,"price":1})

```
> db.Listing.find({"name" : {$regex : ".*Beautiful.*"},"street" : {$regex : ".*Clayton.*"}},{"room_type":1,"price":1})
>
```

Embedding code:

db.Listing_host.find({"name" : {$regex : ".*Beautiful.*"},"street" : {$regex : ".*Clayton.*"}},{"room_type":1,"price":1})

```
> db.Listing_host.find({"name" : {$regex : ".*Beautiful.*"},"street" : {$regex : ".*Clayton.*"}},{"room_type":1,"price":1})
>
```

11.

Referencing code:

db.Listing.aggregate([
{$project:{'list_information':{$concat:['$name', ',' , '$street', ',' , '$neighbourhood']}}}
]).pretty()

```
> db.Listing.aggregate([
... {$project:{'list_information':{$concat:['$name', ',' , '$street', ',' , '$neighbourhood']}}}
... ]).pretty()
{
        "_id" : ObjectId("5d7486cf077f14e20090aff9"),
        "list_information" : "St Kilda 1BR+BEACHSIDE+BALCONY+GARAGE+WIFI+AC,St Kilda, VIC, Australia,Port Phillip"
}
{
        "_id" : ObjectId("5d7486cf077f14e20090affa"),
        "list_information" : "Room in Cool Deco Apartment in Brunswick East,Brunswick East, VIC, Australia,Moreland"
}
{
        "_id" : ObjectId("5d7486cf077f14e20090affb"),
        "list_information" : "Beautiful Room & House,Bulleen, VIC, Australia,Manningham"
}
{
        "_id" : ObjectId("5d7486cf077f14e20090affc"),
        "list_information" : "Melbourne BnB near City & Sports,St Kilda East, VIC, Australia,Port Phillip"
}
{
        "_id" : ObjectId("5d7486cf077f14e20090affd"),
        "list_information" : "Melbourne - Old Trafford Apartment,Berwick, VIC, Australia,Casey"
```

Embedding code:

db.Listing_host.aggregate([
{$project:{'list_information':{$concat:['$name', ',' , '$street', ',' , '$neighbourhood']}}}
]).pretty()

```
> db.Listing_host.aggregate([
... {$project:{'list_information':{$concat:['$name', ',' , '$street', ',' , '$neighbourhood']}}}
... ]).pretty()
{
        "_id" : ObjectId("5d7486cf077f14e20090aff9"),
        "list_information" : "St Kilda 1BR+BEACHSIDE+BALCONY+GARAGE+WIFI+AC,St Kilda, VIC, Australia,Port Phillip"
}
{
        "_id" : ObjectId("5d7486cf077f14e20090b000"),
        "list_information" : "Large private room-close to city,Thornbury, VIC, Australia,Darebin"
}
{
        "_id" : ObjectId("5d7486cf077f14e20090affe"),
        "list_information" : "Kew Tranquility, Melbourne,Kew, VIC, Australia,Boroondara"
}
{
        "_id" : ObjectId("5d7486cf077f14e20090affb"),
        "list_information" : "Beautiful Room & House,Bulleen, VIC, Australia,Manningham"
}
{
        "_id" : ObjectId("5d7486cf077f14e20090affc"),
        "list_information" : "Melbourne BnB near City & Sports,St Kilda East, VIC, Australia,Port Phillip"
}
{
        "_id" : ObjectId("5d7486cf077f14e20090affa"),
        "list_information" : "Room in Cool Deco Apartment in Brunswick East,Brunswick East, VIC, Australia,Moreland"
}
```

12.

Referencing code:

```
db.Listing.aggregate([
{$sort:{"availability_365":-1}},
{$limit:1},
{$project:{'_id':0, 'name':1, 'id':1, 'host_id':1, 'street':1, 'price':1, 'availability_365':1,
'generation_time':Date()}}
]).pretty()
```

```
> db.Listing.aggregate([
... {$sort:{"availability_365":-1}},
... {$limit:1},
... {$project:{'_id':0, 'name':1, 'id':1, 'host_id':1, 'street':1, 'price':1, 'availability_365':1, 'generation_time':Date()}}
... ]).pretty()
{
        "id" : 9835,
        "name" : "Beautiful Room & House",
        "host_id" : 33057,
        "street" : "Bulleen, VIC, Australia",
        "price" : 61,
        "availability_365" : 365,
        "generation_time" : "Sun Sep 15 2019 13:04:46 GMT+1000"
}
>
```

Embedding code:

```
db.Listing_host.aggregate([
{$sort:{"availability_365":-1}},
{$limit:1},
{$project:{'_id':0, 'name':1, 'id':1, 'host_id':1, 'street':1, 'price':1, 'availability_365':1,
'generation_time':Date()}}
]).pretty()
```

```
> db.Listing_host.aggregate([
... {$sort:{"availability_365":-1}},
... {$limit:1},
... {$project:{'_id':0, 'name':1, 'id':1, 'host_id':1, 'street':1, 'price':1, 'availability_365':1, 'generation_time':Date()}}
... ]).pretty()
{
        "id" : 9835,
        "name" : "Beautiful Room & House",
        "host_id" : 33057,
        "street" : "Bulleen, VIC, Australia",
        "price" : 61,
        "availability_365" : 365,
        "generation_time" : "Sun Sep 15 2019 13:06:17 GMT+1000"
}
>
```

13.

Referencing code:

```
db.Listing.aggregate([
{$group:{'_id':'$neighbourhood', 'price':{$avg:'$price'}}},
{$match:{'price':{$gt:50}}},
{ $sort: { '_id': -1 } },
])
```

```
> db.Listing.aggregate([
... {$group:{'_id':'$neighbourhood', 'price':{$avg:'$price'}}},
... {$match:{'price':{$gt:50}}},
... { $sort: { '_id': -1 } },
... ])
{ "_id" : "Yarra Ranges", "price" : 238 }
{ "_id" : "Yarra", "price" : 131.1875 }
{ "_id" : "Wyndham", "price" : 71 }
{ "_id" : "Stonnington", "price" : 85.5 }
{ "_id" : "Port Phillip", "price" : 134.5 }
{ "_id" : "Moreland", "price" : 72.75 }
{ "_id" : "Monash", "price" : 71.5 }
{ "_id" : "Melton", "price" : 72 }
{ "_id" : "Melbourne", "price" : 176.35 }
{ "_id" : "Maribyrnong", "price" : 65 }
{ "_id" : "Manningham", "price" : 61 }
{ "_id" : "Kingston", "price" : 64.33333333333333 }
{ "_id" : "Hobsons Bay", "price" : 86.33333333333333 }
{ "_id" : "Glen Eira", "price" : 191.66666666666666 }
{ "_id" : "Frankston", "price" : 59 }
{ "_id" : "Darebin", "price" : 71.57142857142857 }
{ "_id" : "Casey", "price" : 79 }
{ "_id" : "Boroondara", "price" : 57.5 }
{ "_id" : "Bayside", "price" : 169.5 }
{ "_id" : "Banyule", "price" : 64 }
```

Embedding code:

```
db.Listing_host.aggregate([
{$group:{'_id':'$neighbourhood', 'price':{$avg:'$price'}}},
{$match:{'price':{$gt:50}}},
{ $sort: { '_id': -1 } },
])
```

```
> db.Listing_host.aggregate([
... {$group:{'_id':'$neighbourhood', 'price':{$avg:'$price'}}},
... {$match:{'price':{$gt:50}}},
... { $sort: { '_id': -1 } },
... ])
{ "_id" : "Yarra Ranges", "price" : 238 }
{ "_id" : "Yarra", "price" : 131.1875 }
{ "_id" : "Wyndham", "price" : 71 }
{ "_id" : "Stonnington", "price" : 85.5 }
{ "_id" : "Port Phillip", "price" : 134.5 }
{ "_id" : "Moreland", "price" : 72.75 }
{ "_id" : "Monash", "price" : 71.5 }
{ "_id" : "Melton", "price" : 72 }
{ "_id" : "Melbourne", "price" : 176.35 }
{ "_id" : "Maribyrnong", "price" : 65 }
{ "_id" : "Manningham", "price" : 61 }
{ "_id" : "Kingston", "price" : 64.33333333333333 }
{ "_id" : "Hobsons Bay", "price" : 86.33333333333333 }
{ "_id" : "Glen Eira", "price" : 191.66666666666666 }
{ "_id" : "Frankston", "price" : 59 }
{ "_id" : "Darebin", "price" : 71.57142857142857 }
{ "_id" : "Casey", "price" : 79 }
{ "_id" : "Boroondara", "price" : 57.5 }
{ "_id" : "Bayside", "price" : 169.5 }
{ "_id" : "Banyule", "price" : 64 }
```

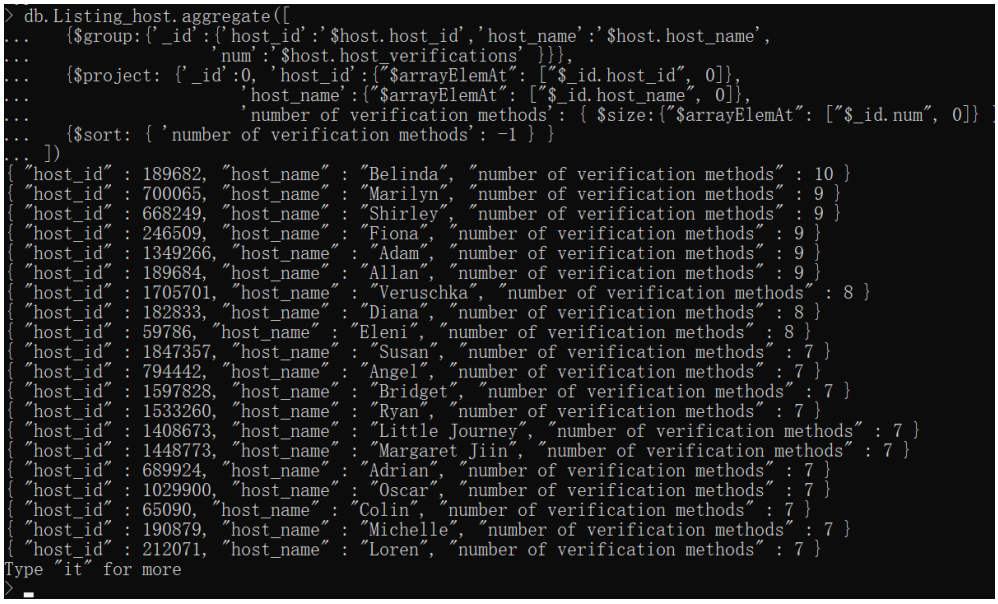
14.

Referencing code:

```
db.Host.aggregate([
  {$project: {_id:0,host_id: 1,host_name:1,
       number_of_verification_methods: { $cond: { if: { $isArray: "$host_verifications" }, then:
                                                  { $size: "$host_verifications" }, else: "NA"} }}},
  {$sort:{ number_of_verification_methods:-1}}
] )
```

Embedding code:

```
db.Listing_host.aggregate([
 {$group:{'_id':{'host_id':'$host.host_id','host_name':'$host.host_name',
          'num':'$host.host_verifications' }}},
 {$project: {'_id':0, 'host_id':{"$arrayElemAt": ["$_id.host_id", 0]},
             'host_name':{"$arrayElemAt": ["$_id.host_name", 0]},
             'number of verification methods': { $size:{"$arrayElemAt": ["$_id.num", 0]} }}},
 {$sort: { 'number of verification methods': -1 } }
])
```



Cassandra:

```
#preparation
#create the keyspace
CREATE KEYSPACE FIT5137_Assign_C3 with
 replication = {'class':'SimpleStrategy', 'replication_factor':1};

#swich to keyspace
USE FIT5137_Assign_C3;

#create table in keyspace
CREATE TABLE review (listing_id  text,
                     review_id text,
                     review_date date,
                     reviewer_id  text,
                     reviewer_name text,
                     review_scores  int,
                     comments text,
                     PRIMARY KEY (listing_id, review_date));

#insert data into table
COPY review (listing_id, review_id, review_date, reviewer_id, reviewer_name,
review_scores, comments)
FROM 'D:\review.csv' WITH HEADER = true AND ESCAPE = '"' ;

#check the number of rows is equal to 8208 or not
SELECT COUNT(*) FROM review;
```

```
cqlsh:fit5137_assign_c3> SELECT COUNT(*) FROM review;

 count
-------
  8130

(1 rows)

Warnings :
Aggregation query used without partition key
```

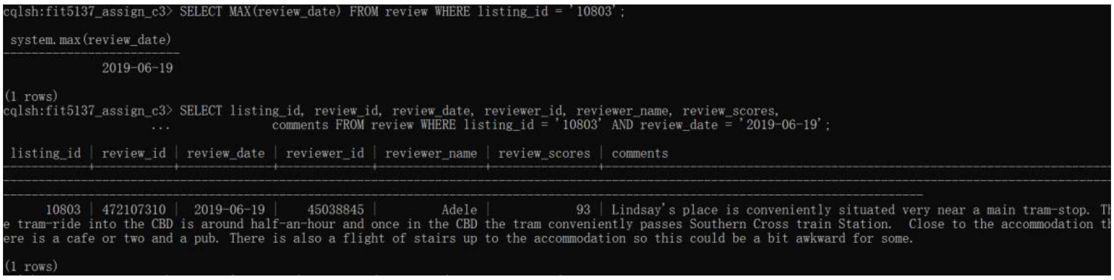
```
#create second index
CREATE INDEX ON review (review_date);
```

15.

Since we need to displace the whole review of the most recent one, the max function only returning the most recent date, the information of other may not be same row. If we use the order by review_date, it could work. The problem is that question 19 also need order by score which will be conflict with this question. So this question will be done by two queries.

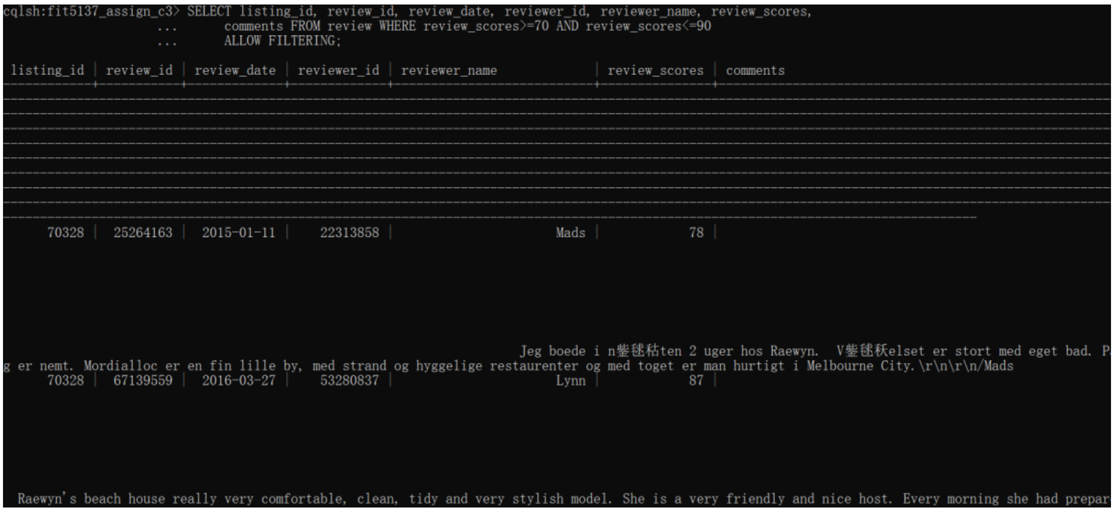
#find the recent date first
SELECT MAX(review_date) FROM review WHERE listing_id = '10803';

#base on the listing id and recent date find the recent review
SELECT listing_id, review_id, review_date, reviewer_id, reviewer_name, review_scores, comments FROM review WHERE listing_id = '10803' AND review_date = '2019-06-19';



```
cqlsh:fit5137_assign_c3> SELECT MAX(review_date) FROM review WHERE listing_id = '10803';

 system.max(review_date)
-------------------------
              2019-06-19

(1 rows)
cqlsh:fit5137_assign_c3> SELECT listing_id, review_id, review_date, reviewer_id, reviewer_name, review_scores,
                     ...           comments FROM review WHERE listing_id = '10803' AND review_date = '2019-06-19';

 listing_id | review_id | review_date | reviewer_id | reviewer_name | review_scores | comments
------------+-----------+-------------+-------------+---------------+---------------+---------

      10803 | 472107310 |  2019-06-19 |    45038845 |         Adele |            93 | Lindsay's place is conveniently situated very near a main tram-stop. Th
e tram-ride into the CBD is around half-an-hour and once in the CBD the tram conveniently passes Southern Cross train Station.  Close to the accommodation th
ere is a cafe or two and a pub. There is also a flight of stairs up to the accommodation so this could be a bit awkward for some.

(1 rows)
```

16.
SELECT listing_id, review_id, review_date, reviewer_id, reviewer_name, review_scores, comments FROM review WHERE review_scores>=70 AND review_scores<=90 ALLOW FILTERING;



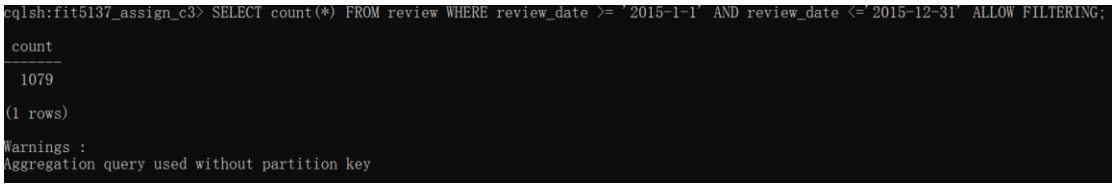
```
cqlsh:fit5137_assign_c3> SELECT listing_id, review_id, review_date, reviewer_id, reviewer_name, review_scores,
                     ...      comments FROM review WHERE review_scores>=70 AND review_scores<=90
                     ...      ALLOW FILTERING;

 listing_id | review_id | review_date | reviewer_id | reviewer_name               | review_scores | comments
------------+-----------+-------------+-------------+-----------------------------+---------------+---------

      70328 |  25264163 |  2015-01-11 |    22313858 |                        Mads |            78 |

                                                 Jeg boede i n鐢毯秸ten 2 uger hos Raewyn.  V鐢毯秼elset er stort med eget bad. P
g er nemt. Mordialloc er en fin lille by, med strand og hyggelige restaurenter og med toget er man hurtigt i Melbourne City.\r\n\r\n/Mads
      70328 |  67139559 |  2016-03-27 |    53280837 |                        Lynn |            87 |

 Raewyn's beach house really very comfortable, clean, tidy and very stylish model. She is a very friendly and nice host. Every morning she had prepar
```

17.
SELECT listing_id, reviewer_name, comments FROM review WHERE review_scores < 50 ALLOW FILTERING;

```
cqlsh:fit5137_assign_c3> SELECT listing_id, reviewer_name, comments FROM review WHERE review_scores < 50 ALLOW FILTERING;

 listing_id | reviewer_name | comments
------------+---------------+---------

      16760 |         Linda |

                                                       Tell others in the Airbnb community about your stay.
      16760 |          Dean |

                Colin was great, stayed up late to meet us & show us around & our room was lovely, comfortable & clean. Thanks a heap!
     268849 |       Charlie |

                                                       Mia has a fantastic apartment, which is set up with all that you'l
need for a great city stay. The cafe downstairs was a bonus, as is the parking. Mia was super friendly and very accommodating. Thanks Mia....
```

18.

SELECT count(*) FROM review WHERE review_date >= '2015-1-1' AND review_date <='2015-12-31' ALLOW FILTERING;



19.

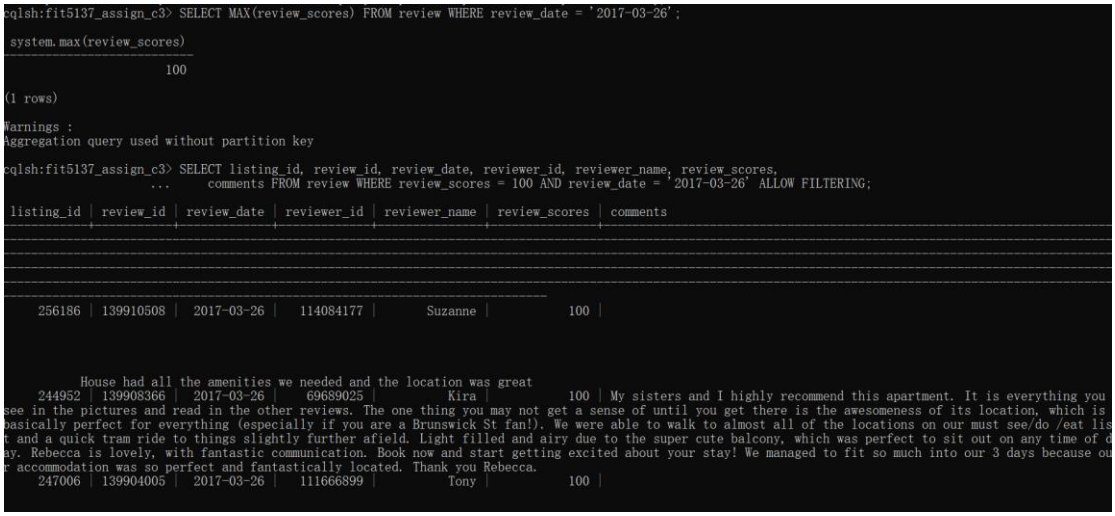
The reason of using 2 query is same as question 15.

#find the highest scores in that day

SELECT MAX(review_scores) FROM review WHERE review_date = '2017-03-26';

#base on the highest scores and date to dispace the review

SELECT listing_id, review_id, review_date, reviewer_id, reviewer_name, review_scores,
comments FROM review WHERE review_scores = 100
AND review_date = '2017-03-26' ALLOW FILTERING;



20.

#find the highest scores in table

SELECT MAX(review_scores) FROM review;

#dispace the review with the highest review scores

SELECT listing_id,reviewer_name, review_scores FROM review WHERE review_scores = 100;

```
cqlsh:fit5137_assign_c3> SELECT MAX(review_scores) FROM review;

 system.max(review_scores)
---------------------------
                       100

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:fit5137_assign_c3> SELECT listing_id,reviewer_name, review_scores FROM review WHERE review_scores = 100;

 listing_id | reviewer_name | review_scores
------------+---------------+---------------
      70328 |           Lee |           100
      70328 |         Diane |           100
      70328 |         Marie |           100
      70328 |         Jo Jo |           100
      70328 |       Barbara |           100
      70328 |       Josique |           100
      70328 |      Michaela |           100
      70328 |         Renee |           100
      70328 |      Gai Sing |           100
      70328 |       Barbara |           100
      70328 |          Ross |           100
      70328 |          Doug |           100
      70328 | Suzanne-Wayne |           100
      70328 |           Ken |           100
     261558 |          鈥扳垙脛鈥扳埇呡 |           100
     261558 |          Chlo鐾毬?|           100
[Errno 0] Error
```

Additional 5 questions:
MongoDB

Additional Q1. Number of listing, average price, max price, min price in each city, and sort by the number of listing order.

Referencing code:

```
db.Listing.aggregate([
{$group:{'_id':'$street',  'total':{$sum:1}, 'average_price':{$avg:'$price'},
'max_price':{$max:'$price'}, 'min_price':{$min:'$price'},}},
{ $sort: { 'total': -1 } }
])
```

```
> db.Listing.aggregate([
... {$group:{'_id':'$street',  'total':{$sum:1}, 'average_price':{$avg:'$price'},
... 'max_price':{$max:'$price'}, 'min_price':{$min:'$price'},}},
... { $sort: { 'total': -1 } }
... ])
{ "_id" : "Melbourne, VIC, Australia", "total" : 14, "average_price" : 133.78571428571428, "max_price" : 249, "min_price" : 36 }
{ "_id" : "Richmond, VIC, Australia", "total" : 6, "average_price" : 139, "max_price" : 249, "min_price" : 94 }
{ "_id" : "Fitzroy, VIC, Australia", "total" : 6, "average_price" : 115.83333333333333, "max_price" : 160, "min_price" : 71 }
{ "_id" : "Parkville, VIC, Australia", "total" : 3, "average_price" : 106, "max_price" : 190, "min_price" : 44 }
{ "_id" : "Elwood, VIC, Australia", "total" : 3, "average_price" : 155.66666666666666, "max_price" : 199, "min_price" : 130 }
{ "_id" : "Thornbury, VIC, Australia", "total" : 3, "average_price" : 106.66666666666667, "max_price" : 140, "min_price" : 50 }
{ "_id" : "St Kilda, VIC, Australia", "total" : 3, "average_price" : 139.66666666666666, "max_price" : 189, "min_price" : 71 }
{ "_id" : "Derrimut, VIC, Australia", "total" : 2, "average_price" : 45, "max_price" : 45, "min_price" : 45 }
{ "_id" : "Menzies Creek, VIC, Australia", "total" : 2, "average_price" : 224.5, "max_price" : 249, "min_price" : 200 }
{ "_id" : "Southbank, VIC, Australia", "total" : 2, "average_price" : 529, "max_price" : 701, "min_price" : 357 }
{ "_id" : "North Melbourne, VIC, Australia", "total" : 2, "average_price" : 122, "max_price" : 155, "min_price" : 89 }
{ "_id" : "Belgrave, VIC, Australia", "total" : 2, "average_price" : 330, "max_price" : 500, "min_price" : 160 }
{ "_id" : "South Melbourne, VIC, Australia", "total" : 2, "average_price" : 83, "max_price" : 85, "min_price" : 81 }
{ "_id" : "Prahran, VIC, Australia", "total" : 2, "average_price" : 85.5, "max_price" : 98, "min_price" : 73 }
{ "_id" : "Caulfield North, VIC, Australia", "total" : 2, "average_price" : 238, "max_price" : 419, "min_price" : 57 }
{ "_id" : "St Kilda East, VIC, Australia", "total" : 2, "average_price" : 97, "max_price" : 125, "min_price" : 69 }
{ "_id" : "East Melbourne, VIC, Australia", "total" : 2, "average_price" : 104.5, "max_price" : 110, "min_price" : 99 }
{ "_id" : "Reservoir, VIC, Australia", "total" : 2, "average_price" : 40, "max_price" : 50, "min_price" : 30 }
{ "_id" : "Newport, VIC, Australia", "total" : 2, "average_price" : 50, "max_price" : 50, "min_price" : 50 }
{ "_id" : "Port Melbourne, VIC, Australia", "total" : 2, "average_price" : 107.5, "max_price" : 120, "min_price" : 95 }
Type "it" for more
```

Embedding code:

```
db.Listing_host.aggregate([
{$group:{'_id':'$street',  'total':{$sum:1}, 'average_price':{$avg:'$price'},
'max_price':{$max:'$price'}, 'min_price':{$min:'$price'},}},
{ $sort: { 'total': -1 } }
])
```

```
> db.Listing_host.aggregate([
... {$group:{'_id':'$street',  'total':{$sum:1}, 'average_price':{$avg:'$price'},
... 'max_price':{$max:'$price'}, 'min_price':{$min:'$price'},}},
... { $sort: { 'total': -1 } }
... ])
{ "_id" : "Melbourne, VIC, Australia", "total" : 14, "average_price" : 133.78571428571428, "max_price" : 249, "min_price" : 36 }
{ "_id" : "Richmond, VIC, Australia", "total" : 6, "average_price" : 139, "max_price" : 249, "min_price" : 94 }
{ "_id" : "Fitzroy, VIC, Australia", "total" : 6, "average_price" : 115.83333333333333, "max_price" : 160, "min_price" : 71 }
{ "_id" : "Parkville, VIC, Australia", "total" : 3, "average_price" : 106, "max_price" : 190, "min_price" : 44 }
{ "_id" : "Elwood, VIC, Australia", "total" : 3, "average_price" : 155.66666666666666, "max_price" : 199, "min_price" : 130 }
{ "_id" : "Thornbury, VIC, Australia", "total" : 3, "average_price" : 106.66666666666667, "max_price" : 140, "min_price" : 50 }
{ "_id" : "St Kilda, VIC, Australia", "total" : 3, "average_price" : 139.66666666666666, "max_price" : 189, "min_price" : 71 }
{ "_id" : "Derrimut, VIC, Australia", "total" : 2, "average_price" : 45, "max_price" : 45, "min_price" : 45 }
{ "_id" : "Menzies Creek, VIC, Australia", "total" : 2, "average_price" : 224.5, "max_price" : 249, "min_price" : 200 }
{ "_id" : "Southbank, VIC, Australia", "total" : 2, "average_price" : 529, "max_price" : 701, "min_price" : 357 }
{ "_id" : "North Melbourne, VIC, Australia", "total" : 2, "average_price" : 122, "max_price" : 155, "min_price" : 89 }
{ "_id" : "Belgrave, VIC, Australia", "total" : 2, "average_price" : 330, "max_price" : 500, "min_price" : 160 }
{ "_id" : "South Melbourne, VIC, Australia", "total" : 2, "average_price" : 83, "max_price" : 85, "min_price" : 81 }
{ "_id" : "Prahran, VIC, Australia", "total" : 2, "average_price" : 85.5, "max_price" : 98, "min_price" : 73 }
{ "_id" : "Caulfield North, VIC, Australia", "total" : 2, "average_price" : 238, "max_price" : 419, "min_price" : 57 }
{ "_id" : "St Kilda East, VIC, Australia", "total" : 2, "average_price" : 97, "max_price" : 125, "min_price" : 69 }
{ "_id" : "East Melbourne, VIC, Australia", "total" : 2, "average_price" : 104.5, "max_price" : 110, "min_price" : 99 }
{ "_id" : "Reservoir, VIC, Australia", "total" : 2, "average_price" : 40, "max_price" : 50, "min_price" : 30 }
{ "_id" : "Newport, VIC, Australia", "total" : 2, "average_price" : 50, "max_price" : 50, "min_price" : 50 }
{ "_id" : "Port Melbourne, VIC, Australia", "total" : 2, "average_price" : 107.5, "max_price" : 120, "min_price" : 95 }
Type "it" for more
>
```

Additional Q2. Count the number of amernities in each listings.

Referencing code:

```
db.Listing.aggregate([
  { $project : { id:'$id', name:'$name', amernites : { $split: ["$amenities", ","] }, qty : 1 } },
  { $unwind : "$amernites" },
  { $group : { _id: { "listing_id" : "$id", 'name':'$name' }, amernities_num: { "$sum" : 1 } } },
  { $sort : { listing_id : -1 } }
])
```

```
> db.Listing.aggregate([
...     { $project : { id:'$id', name:'$name', amernites : { $split: ["$amenities", ","] }, qty : 1 } },
...     { $unwind : "$amernites" },
...     { $group : { _id: { "listing_id" : "$id", 'name':'$name' }, amernities_num: { "$sum" : 1 } } },
...     { $sort : { listing_id : -1 } }
... ])
{ "_id" : { "listing_id" : 289467, "name" : "The Suite @ Angelus Retreat" }, "amernities_num" : 18 }
{ "_id" : { "listing_id" : 288147, "name" : "LARGE RM ENSUITE NEAR CTY & AIRPORT" }, "amernities_num" : 16 }
{ "_id" : { "listing_id" : 365832, "name" : "ST KILDA EAST EXCEPTIONAL LARGE STUNNING HOME" }, "amernities_num" : 41 }
{ "_id" : { "listing_id" : 334678, "name" : "Room + Own Bathroom - 7km from city" }, "amernities_num" : 32 }
{ "_id" : { "listing_id" : 334891, "name" : "Attractive room in leafy Deepdene" }, "amernities_num" : 22 }
{ "_id" : { "listing_id" : 363060, "name" : "SUNLIT Kitchen/Vegie Garden/CLEAN" }, "amernities_num" : 8 }
{ "_id" : { "listing_id" : 357740, "name" : "House in Pristine Coastal Village" }, "amernities_num" : 20 }
{ "_id" : { "listing_id" : 323280, "name" : "Classic Victorian Armadale Home :) Victorian House" }, "amernities_num" : 22 }
{ "_id" : { "listing_id" : 309093, "name" : "3 Level Loft with Views close to City" }, "amernities_num" : 23 }
{ "_id" : { "listing_id" : 307630, "name" : "Queen-bed Room in Huge CBD Warehouse Apartment" }, "amernities_num" : 26 }
{ "_id" : { "listing_id" : 315513, "name" : "Best, west of Melbourne-Wifi & spa2" }, "amernities_num" : 48 }
{ "_id" : { "listing_id" : 297350, "name" : "Best, west of Melbourne-Wifi & Spa1" }, "amernities_num" : 39 }
{ "_id" : { "listing_id" : 307615, "name" : "Queen BR + Private Bathroom in Huge CBD Apartment" }, "amernities_num" : 37 }
{ "_id" : { "listing_id" : 300679, "name" : "Comfy room in Oakleigh South Melbourne" }, "amernities_num" : 23 }
{ "_id" : { "listing_id" : 313648, "name" : "(1) Stylish, East Melb 1bed apt" }, "amernities_num" : 34 }
{ "_id" : { "listing_id" : 317351, "name" : "**just renovated** Bright & Spacious StKildaEast**" }, "amernities_num" : 35 }
{ "_id" : { "listing_id" : 284210, "name" : "Fitzroyalty - luscious living in the heart of it" }, "amernities_num" : 30 }
{ "_id" : { "listing_id" : 281890, "name" : "¸ù§Cheerful retreat! 10km from CBD¸ù§" }, "amernities_num" : 24 }
{ "_id" : { "listing_id" : 232088, "name" : "Melbourne 2  Bedrooms 2 Bathrooms FULL Kitchen" }, "amernities_num" : 59 }
{ "_id" : { "listing_id" : 280698, "name" : "Elwood SPACIOUS OPEN PLAN EXEC 2BR+PARKING+WIFI+AC" }, "amernities_num" : 33 }
Type "it" for more
>
```

Embedding code:

```
db.Listing_host.aggregate([
  { $project : { id:'$id', name:'$name', amernites : { $split: ["$amenities", ","] }, qty : 1 } },
  { $unwind : "$amernites" },
  { $group : { _id: { "listing_id" : "$id", 'name':'$name' }, amernities_num: { "$sum" : 1 } } },
  { $sort : { listing_id : -1 } }
])
```

```
> db.Listing_host.aggregate([
...    { $project : { id:'$id', name:'$name', amernites : { $split: ["$amenities", ","] }, qty : 1 } },
...    { $unwind : "$amernites" },
...    { $group : { _id: { "listing_id" : "$id", 'name':'$name' }, amernities_num: { "$sum" : 1 } } },
...    { $sort : { listing_id : -1 } }
... ])
{ "_id" : { "listing_id" : 289467, "name" : "The Suite @ Angelus Retreat" }, "amernities_num" : 18 }
{ "_id" : { "listing_id" : 288147, "name" : "LARGE RM ENSUITE NEAR CTY & AIRPORT" }, "amernities_num" : 16 }
{ "_id" : { "listing_id" : 365832, "name" : "ST KILDA EAST EXCEPTIONAL LARGE STUNNING HOME" }, "amernities_num" : 41 }
{ "_id" : { "listing_id" : 334678, "name" : "Room + Own Bathroom - 7km from city" }, "amernities_num" : 32 }
{ "_id" : { "listing_id" : 334891, "name" : "Attractive room in leafy Deepdene" }, "amernities_num" : 22 }
{ "_id" : { "listing_id" : 363060, "name" : "SUNLIT Kitchen/Vegie Garden/CLEAN" }, "amernities_num" : 8 }
{ "_id" : { "listing_id" : 357740, "name" : "House in Pristine Coastal Village" }, "amernities_num" : 20 }
{ "_id" : { "listing_id" : 323280, "name" : "Classic Victorian Armadale Home :) Victorian House" }, "amernities_num" : 22 }
{ "_id" : { "listing_id" : 309093, "name" : "3 Level Loft with Views close to City" }, "amernities_num" : 23 }
{ "_id" : { "listing_id" : 307630, "name" : "Queen-bed Room in Huge CBD Warehouse Apartment" }, "amernities_num" : 26 }
{ "_id" : { "listing_id" : 315513, "name" : "Best, west of Melbourne-Wifi & spa2" }, "amernities_num" : 48 }
{ "_id" : { "listing_id" : 297350, "name" : "Best, west of Melbourne-Wifi & Spa1" }, "amernities_num" : 39 }
{ "_id" : { "listing_id" : 307615, "name" : "Queen BR + Private Bathroom in Huge CBD Apartment" }, "amernities_num" : 37 }
{ "_id" : { "listing_id" : 300679, "name" : "Comfy room in Oakleigh South Melbourne" }, "amernities_num" : 23 }
{ "_id" : { "listing_id" : 313648, "name" : "(1) Stylish, East Melb 1bed apt" }, "amernities_num" : 34 }
{ "_id" : { "listing_id" : 317351, "name" : "**just renovated** Bright & Spacious StKildaEast**" }, "amernities_num" : 35 }
{ "_id" : { "listing_id" : 284210, "name" : "Fitzroyalty - luscious living in the heart of it" }, "amernities_num" : 30 }
{ "_id" : { "listing_id" : 281890, "name" : ",ù§Cheerful retreat! 10km from CBD ,ù§" }, "amernities_num" : 24 }
{ "_id" : { "listing_id" : 232088, "name" : "Melbourne 2  Bedrooms 2 Bathrooms FULL Kitchen" }, "amernities_num" : 59 }
{ "_id" : { "listing_id" : 280698, "name" : "Elwood SPACIOUS OPEN PLAN EXEC 2BR+PARKING+WIFI+AC" }, "amernities_num" : 33 }
Type "it" for more
>
```

Additional Q3. Count how many host which room_type is "Private room" and the host_verification allows email.

Referencing code:

```
db.Listing.aggregate([
{"$match":{"room_type":"Private room"}},
{$lookup:{from:'Host',localField:'host_id',foreignField:'host_id',as:'result'}},
{"$match":{"result.host_verifications":"email"}},
{$count: "name"}
])
```

```
> db.Listing.aggregate([
...  {"$match":{"room_type":"Private room"}},
...  {$lookup:{from:'Host',localField:'host_id',foreignField:'host_id',as:'result'}},
...  {"$match":{"result.host_verifications":"email"}},
...  {$count: "name"}
... ])
{ "name" : 48 }
>
```

Embedding code:

```
db.Listing_host.count({"host.host_verifications":"email","room_type":"Private room"})
```

```
> db.Listing_host.count({"host.host_verifications":"email","room_type":"Private room"})
48
>
```

Cassandra

Additional Q4. What is the number of review of each listing?

```
SELECT listing_id, COUNT(*) FROM review GROUP BY listing_id;
```

```
cqlsh:fit5137_assign_c3> SELECT listing_id, COUNT(*) FROM review GROUP BY listing_id;

 listing_id | count
------------+-------
      70328 |   108
     189293 |     1
     261558 |    13
      74715 |    90
      51592 |   218
      16760 |    75
     288147 |    11
     241263 |   202
     313648 |    46
     307630 |   121
     309093 |    10
      15246 |    29
      47100 |   177
     365832 |    15
      41836 |   156
     227964 |     7
     257149 |   109
      67211 |   160
     268849 |   406
     280698 |    50
      78143 |   131
     256186 |   133
     108032 |   233
     133617 |    13
      68482 |     5
     307615 |   102
```

Additional Q5. Display the highest rating and lowest rating left in 2014.

SELECT min(review_scores), max(review_scores) FROM review WHERE review_date >= '2014-1-1' AND review_date <='2014-12-31' ALLOW FILTERING;

```
cqlsh:fit5137_assign_c3> SELECT min(review_scores), max(review_scores) FROM review WHERE  review_date >= '2014-1-1' AND review_date <='2014-12-31' ALLOW FILT
ERING;

 system.min(review_scores) | system.max(review_scores)
---------------------------+---------------------------
                        20 |                       100

(1 rows)

Warnings :
Aggregation query used without partition key
```

C.4 Database Comparison

Through the practice the demo of two database, we have basic idea about the features of two types of Non-SQL database. Following two tables are the advantages and disadvantages of MongoDB and Cassandra:

MongoDB

Advantages	Disadvantages
1. Very friendly for querying. MongoDB can handle different kinds of query without changing collection a lot.	**1. Don't have proper join function.** MongoDB only have left outer join.
2. Data is easy to handle. because the smallest unit is document, did not have a lot of limit in data type.	
3. MongoDB can scale out.	

Cassandra

Advantages	Disadvantages
1. The Cassandra is good at Scaling. data can be partitioning into different data centers, racks and nodes.	**1. Data need to be predefined.**
2. Cassandra have fault tolerance. The duplication feature ensures the data will not lost, even one of node, rack or data center is down.	**2. The system somehow is not very stable.**
3. Cassandra have good availability. The duplication ensures that there is always the node available to query, even the oriental data storage node is down.	**3. A lot of restrictions on querying.** - The delate, update and group by can only using first partitioning key, even the second index will not work. - For order by, we must specify the first partitioning key in restriction condition. - Cassandra did not support sub query or nested query. - If there is null value in certain column, that column cannot be second index or partitioning key. For query, we have to add 'ALLOW FILTERING', it causes the query inefficient.

Comparing the strength and weakness of MongoDB and Cassandra, and the situation of MonashBnB also have been considering, the MongoDB is more suitable for MonashBnB. There are two reasons. The first one is about query, since the MonashBnB is a start-up website, the query function is not very stable, we need to try all different queries to see which queries are more useful for staff and consumer. The Cassandra is not very suitable for this condition, its query is mainly depending on

table structure, the different queries may need to changing data table a lot. In the other hand, the MongoDB is much more flexible, the query did not have a lot of restrictions. The second reason is about the size of website, the MonashBnB is start-up website. The data volume of website is not very big, we don't need to estimate the other datacenter, so the MongoDB should be enough to handling current data size. In the future, Cassandra may can be considered as database for this website when the data volume is very big.

Following is the merging step for merging review table from Cassandra into MongoDB:

- Export the data from Cassandra into csv file by using copy + out function.
- Using pandas in Python to load csv file to wrangling data.
- Define a function to convert the data into library type data which looks like json format, then using json package in Python completely convert the data into json format, export the json format data into json file.
- Insert json file into new mongodb collection, then the merging step is done.