9/18/2019

# FIT5137 - Advanced Database Technology

Semester 2, 2019

## Assignment 1

## MongoDB and Cassandra


Tutor Name:

Chaluka Salgado

Submitted by:

Sameeksha Ingle – 30036585

Abhilash Kale – 30254140

MONASH University
Information Technology

# GROUP ASSIGNMENT COVER SHEET

| Student ID Number | Surname | Given Names |
|---|---|---|
| 30036585 | Ingle | Sameeksha |
| 30254140 | Kale | Abhilash Anil |
| | | |
| | | |

* Please include the names of all other group members.

| | | |
|---|---|---|
| **Unit name and code** | FIT5137 – Advanced Database Technology | |
| **Title of assignment** | MongoDB and Cassandra | |
| **Lecturer/tutor** | Chaluka Salgado | |
| **Tutorial day and time** | Monday, 4PM to 6PM | **Campus:** Caulfield |
| **Is this an authorised group assignment?** ☒ Yes ☐ No | | |
| **Has any part of this assignment been previously submitted as part of another unit/course?** ☐ Yes ☒ No | | |
| **Due Date:** 18th September 2019 | | **Date submitted:** 18th September 2019 |

All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor.

**Extension granted until (date)** ..... **Signature of lecturer/tutor** .....

Please note that it is your responsibility to retain copies of your assessments.

***Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations***

**Plagiarism**: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

**Collusion**: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

**Student Statement:**

- I have read the university's Student Academic Integrity Policy and Procedures.
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://adm.monash.edu/legal/legislation/statutes
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:
  - i. provide to another member of faculty and any external marker; and/or
  - ii. submit it to a text matching software; and/or
  - iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

*Signature* ..... *Date*.....

* delete (iii) if not applicable

Signature Sameeksha Ingle Date: 18/09/19 Signature _____ Date: _____

Signature Abhilash Kale Date: 18/09/19 Signature _____ Date: _____

Signature _____ Date: _____ Signature _____ Date: _____

**Privacy Statement**

The information on this form is collected for the primary purpose of assessing your assignment and ensuring the academic integrity requirements of the University are met. Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au

Updated: 17 Jun 2014

## 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 | List of parts student did |
|---|---|---|---|
| **30036585** | Sameeksha Ingle | 50% | C.1. MongoDB create collection and data insert for listing<br>C.2. Performed all queries together<br>C.3. Performed and created all queries together<br>C.4. Worked on the report together |
| **30254140** | Abhilash Kale | 50% | C.1. MongoDB create collection and data insert for host. Cassandra create and data insert for review.<br>C.2. Performed all queries together<br>C.3. Performed and created all queries together<br>C.4. Worked on the report together |

## 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.

## 2 SIGNATURE

**Signatures** Sameeksha Ingle | Abhilash Kale

**Date** Day Month Year: 18 / 09 / 2019

## C.1. Database Design

### C.1.1. MongoDB

```
use FIT5137_Assign

db.createCollection("host")

db.host.insertMany([
{
hostId: "MONHOS01",
hostName: "Manju",
hostVerifications: ["email", "phone", "reviews"],
hostSince: new Date("2009/08/21"),
hostLocation: {suburb: "Clayton", state: "Victoria", country: "Australia"},
hostResponseRate: "within a day",
hostIsSuperhost: false
},
{
hostId: "MONHOS02",
hostName: "Lindsay",
hostVerifications: ["email", "phone", "reviews", "jumio", "government id"],
hostSince: new Date("2009/09/16"),
hostLocation: {suburb: "Clifton Hill", state: "Victoria", country: "Australia"},
hostResponseRate: "within an hour",
hostIsSuperhost: true
},
{
hostId: "MONHOS03",
hostName: "Adam",
hostVerifications: ["email", "phone", "google", "reviews", "jumio", "government id", "work
email"],
hostSince: new Date("2009/10/31"),
hostLocation: {suburb: "Port Melbourne", state: "Victoria", country: "Australia"},
```

hostResponseRate: "within an hour",
hostIsSuperhost: false
},
{
hostId: "MONHOS04",
hostName: "Eleni",
hostVerifications: ["email", "phone", "facebook", "reviews", "jumio", "offline government id", "government id", "work email"],
hostSince: new Date("2009/12/03"),
hostLocation: {suburb: "Fitzroy", state: "Victoria", country: "Australia"},
hostResponseRate: "within a day",
hostIsSuperhost: false
},
{
hostId: "MONHOS05",
hostName: "Colin",
hostVerifications: ["email", "phone", "facebook", "reviews", "jumio", "offline government id", "government id"],
hostSince: new Date("2009/12/22"),
hostLocation: {suburb: "Saint Kilda East", state: "Victoria", country: "Australia"},
hostResponseRate: "within an hour",
hostIsSuperhost: false
},
{
hostId: "MONHOS06",
hostName: "Daryl",
hostVerifications: ["email", "phone", "manual online", "reviews", "manual offline", "work email"],
hostSince: new Date("2010/07/12"),
hostLocation: {suburb: "Berwick", state: "Victoria", country: "Australia"},
hostResponseRate: "within an hour",
hostIsSuperhost: true
},
{

hostId: "MONHOS07",
hostName: "Diana",
hostVerifications: ["email", "phone", "facebook", "reviews", "jumio", "offline government id", "government id", "work email"],
hostSince: new Date("2010/07/27"),
hostLocation: {suburb: "Parkdale", state: "Victoria", country: "Australia"},
hostResponseRate: "within a day",
hostIsSuperhost: false
},
{
hostId: "MONHOS08",
hostName: "Belinda",
hostVerifications: ["email", "phone", "facebook", "reviews", "jumio", "offline government id", "selfie", "government id", "identity manual", "work email"],
hostSince: new Date("2010/08/03"),
hostLocation: {suburb: "Prahran", state: "Victoria", country: "Australia"},
hostResponseRate: "within a few hours",
hostIsSuperhost: false
},
{
hostId: "MONHOS09",
hostName: "Allan",
hostVerifications: ["email", "phone", "facebook", "reviews", "jumio", "offline government id", "selfie", "government id", "identity manual"],
hostSince: new Date("2010/08/03"),
hostLocation: {suburb: "South Melbourne", state: "Victoria", country: "Australia"},
hostResponseRate: "within an hour",
hostIsSuperhost: true
},
{
hostId: "MONHOS10",
hostName: "Vicki",
hostVerifications: ["email", "phone", "facebook", "reviews", "jumio", "government id"],
hostSince: new Date("2010/08/06"),

```
hostLocation: {suburb: "Frankston", state: "Victoria", country: "Australia"},
hostResponseRate: "within an hour",
hostIsSuperhost: true
}
]);

db.createCollection ("listing");

db.listing.insertMany([
{
  listingID: "MONLST01",
  name: "Monash Beautiful House",
  hostID: "MONHOS14",
  neighbourhood: "Manningham",
  address: {
          suburb: "Clayton",
          state: "VIC",
          postCode: 3800
  },
  latitude: -37.773,
  longitude: 145.09213,
  roomType: "Entire home",
  amenities: ["TV", "Wifi", "Pets Allowed", "Family friendly", "24-hour check-in", "Self
check-in"],
  "pricePerNight($)": 61,
  "priceForExtraPeople($)": 22,
  minimumNightsRequiredForBooking: 1,
  availabilityPer365Days: 365
},
{
  listingID: "MONLST02",
  name: "Monash Brunswick Deco",
  hostID: "MONHOS08",
  neighbourhood: "Moreland",
```

```
  address: {
    suburb: "Brunswick East",
    state: "VIC",
    postCode: 3057
  },
  latitude: -37.767,
  longitude: 144.98074,
  roomType: "Private room",
  amenities: ["Kitchen", "Toiletries", "Hair dryer", "Iron", "Microwave", "Coffee Maker",
"Refrigerator", "Cooking basics", "Stove", "Garden"],
  "pricePerNight($)": 35,
  "priceForExtraPeople($)": 15,
  minimumNightsRequiredForBooking: 3,
  availabilityPer365Days: 194
},
{
  listingID: "MONLST03",
  name: "Monash Beachside Retreat",
  hostID: "MONHOS01",
  neighbourhood: "Port Philip",
  address: {
    suburb: "St Kilda",
    state: "VIC",
    postCode: 3182
  },
  latitude: -37.86,
  longitude: 144.97737,
  roomType: "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"],
  "pricePerNight($)": 159,
  "priceForExtraPeople($)": 0,
```

```
  minimumNightsRequiredForBooking: 2,
  availabilityPer365Days: 82
},
{
  listingID: "MONLST04",
  name: "Monash Close2City",
  hostID: "MONHOS04",
  neighbourhood: "Darebin",
  address: {
    suburb: "Thornbury",
    state: "VIC",
    postCode: 3071
  },
  latitude: -37.759,
  longitude: 144.98923,
  roomType: "Private room",
  amenities: ["TV", "Internet", "Wifi", "Kitchen Free parking on premises", "Breakfast",
"Heating", "Washer", "Toiletries", "Hair dryer"],
  "pricePerNight($)": 50,
  "priceForExtraPeople($)": 20,
  minimumNightsRequiredForBooking: 2,
  availabilityPer365Days: 0
},
{
  listingID: "MONLST05",
  name: "Monash City and Sports",
  hostID: "MONHOS05",
  neighbourhood: "Port Phillip",
  address: {
    suburb: "St Kilda East",
    state: "VIC",
    postCode: 3183
  },
  latitude: -37.865,
```

```
  longitude: 144.99224,
  roomType: "Private room",
  amenities: ["Internet", "Wifi", "Heating", "Washer", "Toiletries", "Bed linens", "Gym
Access", "24-hour check-in", "Self check-in"],
  "pricePerNight($)": 69,
  "priceForExtraPeople($)": 20,
  minimumNightsRequiredForBooking: 1,
  availabilityPer365Days: 274
},
{
  listingID: "MONLST06",
  name: "Monash Trafford Apartment",
  hostID: "MONHOS06",
  neighbourhood: "Casey",
  address: {
    suburb: "Berwick",
    state: "VIC",
    postCode: 3806
  },
  latitude: -38.057,
  longitude: 145.33936,
  roomType: "Entire home",
  amenities: ["Microwave", "Coffee Maker", "Wifi", "Kitchen", "Toiletries", "Hair dryer"],
  "pricePerNight($)": 99,
  "priceForExtraPeople($)": 30,
  minimumNightsRequiredForBooking: 1,
  availabilityPer365Days: 353
},
{
  listingID: "MONLST07",
  name: "Monash Close2Airport",
  hostID: "MONHOS07",
  neighbourhood: "Darebin",
  address: {
```

```
    suburb: "Reservoir",
    state: "VIC",
    postCode: 3073
  },
  latitude: -37.697,
  longitude: 145.00082,
  roomType: "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"],
  "pricePerNight($)": 50,
  "priceForExtraPeople($)": 20,
  minimumNightsRequiredForBooking: 7,
  availabilityPer365Days: 0
},
{
  listingID: "MONLST08",
  name: "Monash Home In The City",
  hostID: "MONHOS02",
  neighbourhood: "Melbourne",
  address: {
    suburb: "East Melbourne",
    state: "VIC",
    postCode: 3002
  },
  latitude: -37.81,
  longitude: 144.98592,
  roomType: "Private room",
  amenities: ["Cooking basics", "TV", "Wifi", "Pets Allowed", "Family friendly", "24-hour check-in", "Self check-in"],
  "pricePerNight($)": 99,
  "priceForExtraPeople($)": 25,
  minimumNightsRequiredForBooking: 15,
```

```
  availabilityPer365Days: 62
},
{
  listingID: "MONLST09",
  name: "Monash Japanese-Style",
  hostID: "MONHOS11",
  neighbourhood: "Monash",
  address: {
    suburb: "Oakleigh East",
    state: "VIC",
    postCode: 3166
  },
  latitude: -37.9,
  longitude: 145.11447,
  roomType: "Entire home",
  amenities: ["Kitchen", "Toiletries", "Hair dryer", "Iron", "Microwave", "Garden", "Coffee
Maker", "Refrigerator", "Cooking basics", "Stove", "Dishwasher"],
  "pricePerNight($)": 98,
  "priceForExtraPeople($)": 0,
  minimumNightsRequiredForBooking: 2,
  availabilityPer365Days: 219
},
{
  listingID: "MONLST10",
  name: "Beautiful Monash House",
  hostID: "MONHOS10",
  neighbourhood: "Frankston",
  address: {
    suburb: "Frankston",
    state: "VIC",
    postCode: 3199
  },
  latitude: -38.149,
  longitude: 145.14157,
```

```
  roomType: "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"],
  "pricePerNight($)": 59,
  "priceForExtraPeople($)": 10,
  minimumNightsRequiredForBooking: 2,
  availabilityPer365Days: 318
},
{
  listingID: "MONLST11",
  name: "Fabulous Monash Richmond",
  hostID: "MONHOS09",
  neighbourhood: "Yarra",
  address: {
    suburb: "Richmond",
    state: "VIC",
    postCode: 3121
  },
  latitude: -37.818,
  longitude: 145.0044,
  roomType: "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"],
  "pricePerNight($)": 98,
  "priceForExtraPeople($)": 30,
  minimumNightsRequiredForBooking: 14,
  availabilityPer365Days: 16
},
{
  listingID: "MONLST12",
```

name: "Monash Central Lux",
hostID: "MONHOS12",
neighbourhood: "Port Philip",
address: {
suburb: "St Kila",
state: "VIC",
postCode: 3182
},
latitude: -37.861,
longitude: 144.98038,
roomType: "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"],
"pricePerNight($)": 189,
"priceForExtraPeople($)": 29,
minimumNightsRequiredForBooking: 2,
availabilityPer365Days: 6
},
{
listingID: "MONLST13",
name: "Central Monash Warehouse Apartment",
hostID: "MONHOS13",
neighbourhood: "Melbourne",
address: {
suburb: "Melbourne",
state: "VIC",
postCode: 3000
},
latitude: -37.815,
longitude: 144.96267,
roomType: "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"],
  "pricePerNight($)": 249,
  "priceForExtraPeople($)": 40,
  minimumNightsRequiredForBooking: 2,
  availabilityPer365Days: 353
},
{
  listingID: "MONLST14",
  name: "Monash Near the Park",
  hostID: "MONHOS03",
  neighbourhood: "Bayside",
  address: {
    suburb: "Melbourne",
    state: "VIC",
    postCode: 3187
  },
  latitude: -37.928,
  longitude: 145.02518,
  roomType: "Private room",
  amenities: ["Pets Allowed", "Breakfast", "Garden", "Family friendly", "24-hour check-in", "Self check-in"],
  "pricePerNight($)": 40,
  "priceForExtraPeople($)": 11,
  minimumNightsRequiredForBooking: 2,
  availabilityPer365Days: 365
}
]);

Command Prompt - mongo

```
> db
FIT5137_Assign
> db.host.find().pretty();
{
        "_id" : ObjectId("5d7daa5a15886b2a8a3d4c3f"),
        "hostId" : "MONHOS01",
        "hostName" : "Manju",
        "hostVerifications" : [
                "email",
                "phone",
                "reviews"
        ],
        "hostSince" : ISODate("2009-08-20T14:00:00Z"),
        "hostLocation" : {
                "suburb" : "Clayton",
                "state" : "Victoria",
                "country" : "Australia"
        },
        "hostResponseRate" : "within a day",
        "hostIsSuperhost" : false
}
{
        "_id" : ObjectId("5d7daa5a15886b2a8a3d4c40"),
        "hostId" : "MONHOS02",
        "hostName" : "Lindsay",
        "hostVerifications" : [
                "email",
                "phone",
                "reviews",
                "jumio",
                "government id"
        ],
        "hostSince" : ISODate("2009-09-15T14:00:00Z"),
        "hostLocation" : {
                "suburb" : "Clifton Hill",
                "state" : "Victoria",
                "country" : "Australia"
        },
        "hostResponseRate" : "within an hour",
        "hostIsSuperhost" : true
}
{
        "_id" : ObjectId("5d7daa5a15886b2a8a3d4c41"),
        "hostId" : "MONHOS03",
        "hostName" : "Adam",
        "hostVerifications" : [
                "email",
                "phone",
                "google",
                "reviews",
                "jumio",
                "government id",
```

MongoDB Compass Community - localhost:27017/FIT5137_Assign.host

Connect View Collection Help

My Cluster — 10 DBS, 13 COLLECTIONS

- FIT5137_Assign
  - host
  - listing
- FIT5137_Assign_C3
- FIT5137_Lecture2
- admin
- config
- lecture3
- local
- restaurantdb
- test
- testDB2

localhost:27017 STANDALONE — MongoDB 4.0.11 Community

FIT5137_Assign.host

DOCUMENTS 15 — TOTAL SIZE 4.9KB — AVG. SIZE 335B — INDEXES 1 — TOTAL SIZE 32.0KB — AVG. SIZE 32.0KB

Documents | Aggregations | Explain Plan | Indexes

FILTER — OPTIONS — FIND — RESET

INSERT DOCUMENT — VIEW: LIST, TABLE — Displaying documents 1 - 15 of 15

host

| | _id ObjectId | hostId String | hostName String | hostVerifications Array | hostSince Date | hostLocation Object | host |
|---|---|---|---|---|---|---|---|
| 1 | 5d7daa5a15886b2a8a3d4c3f | "MONHOS01" | "Manju" | [] 3 elements | 2009-08-21T00:00:00.000+10:00 | {} 3 fields | "wit |
| 2 | 5d7daa5a15886b2a8a3d4c40 | "MONHOS02" | "Lindsay" | [] 5 elements | 2009-09-16T00:00:00.000+10:00 | {} 3 fields | "wit |
| 3 | 5d7daa5a15886b2a8a3d4c41 | "MONHOS03" | "Adam" | [] 8 elements | 2009-10-31T00:00:00.000+11:00 | {} 3 fields | "wit |
| 4 | 5d7daa5a15886b2a8a3d4c42 | "MONHOS04" | "Eleni" | [] 8 elements | 2009-12-03T00:00:00.000+11:00 | {} 3 fields | "wit |
| 5 | 5d7daa5a15886b2a8a3d4c43 | "MONHOS05" | "Colin" | [] 7 elements | 2009-12-22T00:00:00.000+11:00 | {} 3 fields | "wit |
| 6 | 5d7daa5a15886b2a8a3d4c44 | "MONHOS06" | "Daryl" | [] 6 elements | 2010-07-12T00:00:00.000+10:00 | {} 3 fields | "wit |
| 7 | 5d7daa5a15886b2a8a3d4c45 | "MONHOS07" | "Diana" | [] 8 elements | 2010-07-27T00:00:00.000+10:00 | {} 3 fields | "wit |
| 8 | 5d7daa5a15886b2a8a3d4c46 | "MONHOS08" | "Belinda" | [] 10 elements | 2010-08-03T00:00:00.000+10:00 | {} 3 fields | "wit |
| 9 | 5d7daa5a15886b2a8a3d4c47 | "MONHOS09" | "Allan" | [] 9 elements | 2010-08-03T00:00:00.000+10:00 | {} 3 fields | "wit |
| 10 | 5d7daa5a15886b2a8a3d4c48 | "MONHOS10" | "Vicki" | [] 6 elements | 2010-08-06T00:00:00.000+10:00 | {} 3 fields | "wit |
| 11 | 5d7dce0829a59a40aba50c4a | "MONHOS11" | "Alison" | [] 4 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |
| 12 | 5d7dce0829a59a40aba50c4b | "MONHOS12" | "Mike" | [] 2 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |
| 13 | 5d7dce0829a59a40aba50c4c | "MONHOS13" | "Robyn" | [] 2 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |
| 14 | 5d7dce0829a59a40aba50c4d | "MONHOS14" | "Daniel" | [] 3 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |
| 15 | 5d7dce0829a59a40aba50c4e | "MONHOS15" | "Ron" | [] 1 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |

Command Prompt - mongo

```
> db
FIT5137_Assign
> db.listing.find().pretty();
{
        "_id" : ObjectId("5d7daac415886b2a8a3d4c49"),
        "listingID" : "MONLST01",
        "name" : "Monash Beautiful House",
        "hostID" : "MONHOS14",
        "neighbourhood" : "Manningham",
        "address" : {
                "suburb" : "Clayton",
                "state" : "VIC",
                "postCode" : 3800
        },
        "latitude" : -37.773,
        "longitude" : 145.09213,
        "roomType" : "Entire home",
        "amenities" : [
                "TV",
                "Wifi",
                "Pets Allowed",
                "Family friendly",
                "24-hour check-in",
                "Self check-in"
        ],
        "pricePerNight($)" : 61,
        "priceForExtraPeople($)" : 22,
        "minimumNightsRequiredForBooking" : 1,
        "availabilityPer365Days" : 365
}
{
        "_id" : ObjectId("5d7daac415886b2a8a3d4c4a"),
        "listingID" : "MONLST02",
        "name" : "Monash Brunswick Deco",
        "hostID" : "MONHOS08",
        "neighbourhood" : "Moreland",
        "address" : {
                "suburb" : "Brunswick East",
                "state" : "VIC",
                "postCode" : 3057
        },
        "latitude" : -37.767,
        "longitude" : 144.98074,
        "roomType" : "Private room",
        "amenities" : [
                "Kitchen",
                "Toiletries",
                "Hair dryer",
                "Iron",
                "Microwave",
                "Coffee Maker",
                "Refrigerator",
```

MongoDB Compass Community - localhost:27017/FIT5137_Assign.listing

Connect View Collection Help

My Cluster — 10 DBS, 13 COLLECTIONS

- FIT5137_Assign
  - host
  - listing
- FIT5137_Assign_C3
- FIT5137_Lecture2
- admin
- config
- lecture3
- local
- restaurantdb
- test
- testDB2

localhost:27017 STANDALONE — MongoDB 4.0.11 Community

FIT5137_Assign.listing

DOCUMENTS 12 — TOTAL SIZE 7.5KB — AVG. SIZE 641B — INDEXES 1 — TOTAL SIZE 32.0KB — AVG. SIZE 32.0KB

Documents | Aggregations | Explain Plan | Indexes

FILTER — OPTIONS — FIND — RESET

INSERT DOCUMENT — VIEW: LIST, TABLE — Displaying documents 1 - 12 of 12

host

| | _id ObjectId | listingID String | name String | hostID String | neighbourhood String | address Object | lat |
|---|---|---|---|---|---|---|---|
| 1 | 5d7daac415886b2a8a3d4c49 | "MONLST01" | "Monash Beautiful House" | "MONHOS14" | "Manningham" | {} 3 fields | -37. |
| 2 | 5d7daac415886b2a8a3d4c4a | "MONLST02" | "Monash Brunswick Deco" | "MONHOS08" | "Moreland" | {} 3 fields | -37. |
| 3 | 5d7daac415886b2a8a3d4c4b | "MONLST03" | "Monash Beachside Retreat" | "MONHOS01" | "Port Philip" | {} 3 fields | -37. |
| 4 | 5d7daac415886b2a8a3d4c4d | "MONLST05" | "Monash City and Sports" | "MONHOS05" | "Port Phillip" | {} 3 fields | -37. |
| 5 | 5d7daac415886b2a8a3d4c4e | "MONLST06" | "Monash Trafford Apartment" | "MONHOS06" | "Casey" | {} 3 fields | -38. |
| 6 | 5d7daac415886b2a8a3d4c50 | "MONLST08" | "Monash Home In The City" | "MONHOS02" | "Melbourne" | {} 3 fields | -37. |
| 7 | 5d7daac415886b2a8a3d4c51 | "MONLST09" | "Monash Japanese-Style" | "MONHOS11" | "Monash City" | {} 3 fields | -37. |
| 8 | 5d7daac415886b2a8a3d4c52 | "MONLST10" | "Beautiful Monash House" | "MONHOS10" | "Frankston" | {} 3 fields | -38. |
| 9 | 5d7daac415886b2a8a3d4c53 | "MONLST11" | "Fabulous Monash Richmond" | "MONHOS09" | "Yarra" | {} 3 fields | -37. |
| 10 | 5d7daac415886b2a8a3d4c54 | "MONLST12" | "Monash Central Lux" | "MONHOS12" | "Port Philip" | {} 3 fields | -37. |
| 11 | 5d7daac415886b2a8a3d4c55 | "MONLST13" | "Central Monash Warehouse Apart | "MONHOS13" | "Melbourne" | {} 3 fields | -37. |
| 12 | 5d7daac415886b2a8a3d4c56 | "MONLST14" | "Monash Near the Park" | "MONHOS03" | "Bayside" | {} 3 fields | -37. |

### C.1.2. Cassandra

```
CREATE KEYSPACE FIT5137_Assign
WITH REPLICATION = {'class': 'SimpleStrategy',
'replication_factor' : 1};

USE FIT5137_Assign;

CREATE TABLE review (
  listing_id text,
  review_id text,
  review_date date,
  sequence text,
  reviewer_id int,
  reviewer_name text,
  review_scores_rating int,
  satisfied_reason set<text>,
  comments text,
  PRIMARY KEY ((review_id), listing_id, reviewer_id)
);

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST02',
'REV01', '2017-03-22', '10:37:50+1300', 500001, 'Miriam', 90, {'location', 'amenities'},
'Beautiful View');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST02',
'REV02', '2017-03-22', '11:37:50+1300', 500002, 'Johannes', 90, {'host'}, 'Good Host');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id,
reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST02',
```

'REV03', '2017-03-22', '11:37:50+1300', 500003, 'Camille', 100, {'location', 'view'}, 'Nice View and Location');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST02', 'REV04', '2017-03-22', '12:37:50+1300', 500004, 'Paige', 95, {'price'}, 'Excellent Price');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST01', 'REV05', '2017-03-22', '15:37:50+1300', 500005, 'Adele', 93, {'location', 'price'}, 'Good Location');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST03', 'REV06', '2017-03-22', '17:37:50+1300', 500006, 'Greg', 87, {'host', 'view'}, 'Very Clean House');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST04', 'REV07', '2017-03-22', '19:37:50+1300', 500007, 'Wolfgang', 91, {'location', 'price'}, 'Nice Location');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST05', 'REV08', '2017-03-22', '20:37:50+1300', 500008, 'Klaus', 96, {'location', 'view'}, 'Nice Building');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST06', 'REV09', '2017-03-23', '11:37:50+1300', 500009, 'Rox', 100, {'host', 'price'}, 'Friendly Host');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST05', 'REV10', '2017-03-23', '12:37:50+1300', 500010, 'Elizabeth', 98, {'host', 'price'}, 'Friendly Host');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST09',

'REV11', '2017-03-23', '19:37:50+1300', 500011, 'Derek', 100, {'space', 'clean'}, 'Very Clean and comfortable');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST09', 'REV12', '2017-03-25', '10:07:40+1300', 500012, 'Joy', 92, {'host', 'clean'}, 'Friendly and Nice Host');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST10', 'REV13', '2017-03-26', '10:02:10+1300', 500013, 'Anouck', 93, {'host', 'view'}, 'Very Comfortable');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST12', 'REV14', '2017-03-26', '10:49:40+1300', 500014, 'Jerome', 85, {'location', 'clean'}, 'Friendly Host');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST13', 'REV15', '2017-03-26', '10:48:40+1300', 500015, 'Jehan', 98, {'location', 'amenities'}, 'Beautiful View');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST14', 'REV16', '2017-03-26', '10:48:10+1300', 500016, 'Joy', 97, {'amenities', 'view'}, 'Good Location');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST14', 'REV17', '2017-03-26', '10:47:40+1300', 500014, 'Jerome', 30, {'price', 'view'}, 'Bad Location');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST10', 'REV18', '2017-03-26', '10:47:10+1300', 500002, 'Johannes', 20, {'amenities', 'view'}, 'Bad Service');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST12', 'REV19', '2017-03-27', '09:37:50+1300', 500013, 'Anouck', 87, {'space', 'clean'}, 'Good Location');

INSERT INTO review (listing_id, review_id, review_date, sequence, reviewer_id, reviewer_name, review_scores_rating, satisfied_reason, comments) VALUES ('MONLST05', 'REV20', '2017-03-27', '10:45:10+1300', 500011, 'Derek', 96, {'host', 'view'}, 'Nice Building');

Command Prompt - cqlsh

```
cqlsh:fit5137_assign> select * from review;

 review_id | listing_id | reviewer_id | comments                    | review_date | review_scores_rating | reviewer_name | satisfied_reason           | sequence
-----------+------------+-------------+-----------------------------+-------------+----------------------+---------------+----------------------------+--------------
     REV03 |   MONLST02 |      500003 |     Nice View and Location  |  2017-03-22 |                  100 |       Camille |      {'location', 'view'}  | 11:37:50+1300
     REV19 |   MONLST12 |      500013 |              Good Location  |  2017-03-27 |                   87 |        Anouck |        {'clean', 'space'}  | 09:37:50+1300
     REV10 |   MONLST05 |      500010 |              Friendly Host  |  2017-03-23 |                   98 |     Elizabeth |         {'host', 'price'}  | 12:37:50+1300
     REV06 |   MONLST03 |      500006 |          Very Clean House   |  2017-03-22 |                   87 |          Greg |          {'host', 'view'}  | 17:37:50+1300
     REV13 |   MONLST10 |      500013 |          Very Comfortable   |  2017-03-26 |                   93 |        Anouck |          {'host', 'view'}  | 10:02:10+1300
     REV08 |   MONLST05 |      500008 |              Nice Building  |  2017-03-22 |                   96 |         Klaus |      {'location', 'view'}  | 20:37:50+1300
     REV02 |   MONLST02 |      500002 |                   Good Host |  2017-03-22 |                   90 |      Johannes |                  {'host'}  | 11:37:50+1300
     REV11 |   MONLST09 |      500011 | Very Clean and comfortable  |  2017-03-23 |                  100 |         Derek |        {'price', 'space'}  | 19:37:50+1300
     REV20 |   MONLST05 |      500011 |              Nice Building  |  2017-03-27 |                   96 |         Derek |          {'host', 'view'}  | 10:45:10+1300
     REV14 |   MONLST12 |      500014 |              Friendly Host  |  2017-03-26 |                   85 |        Jerome |     {'clean', 'location'}  | 10:49:40+1300
     REV09 |   MONLST06 |      500009 |              Friendly Host  |  2017-03-23 |                  100 |           Rox |         {'host', 'price'}  | 11:37:50+1300
     REV12 |   MONLST09 |      500012 |     Friendly and Nice Host  |  2017-03-25 |                   92 |           Joy |         {'clean', 'host'}  | 10:07:40+1300
     REV07 |   MONLST04 |      500007 |              Nice Location  |  2017-03-22 |                   91 |      Wolfgang |     {'location', 'price'}  | 19:37:50+1300
     REV05 |   MONLST01 |      500005 |              Good Location  |  2017-03-22 |                   93 |         Adele |     {'location', 'price'}  | 15:37:50+1300
     REV01 |   MONLST02 |      500001 |             Beautiful View  |  2017-03-22 |                   90 |        Miriam | {'amenities', 'location'}  | 10:37:50+1300
     REV16 |   MONLST14 |      500016 |              Good Location  |  2017-03-26 |                   97 |           Joy |     {'amenities', 'view'}  | 10:48:10+1300
     REV18 |   MONLST10 |      500002 |                Bad Service  |  2017-03-26 |                   20 |      Johannes |     {'amenities', 'view'}  | 10:47:10+1300
     REV04 |   MONLST02 |      500004 |           Excellent Price   |  2017-03-22 |                   95 |         Paige |                  {'price'}  | 12:37:50+1300
     REV17 |   MONLST14 |      500014 |               Bad Location  |  2017-03-26 |                   30 |        Jerome |         {'price', 'view'}  | 10:47:40+1300

(19 rows)
cqlsh:fit5137_assign>
```

## C.2. Database Modifications

***Q1***

```
db.host.updateOne(
{hostName: "Adam"},
{
        $addToSet: { "hostVerifications": "facebook" }
}
);
```

***Q2***

```
db.host.insertMany([
{
hostId: "MONHOS11",
hostName: "Alison",
hostVerifications: ["email", "phone", "facebook", "reviews"],
hostSince: new Date("2019/01/09"),
hostLocation: {suburb: "Caulfield", state: "Victoria", country: "Australia"},
hostResponseRate: "within an hour",
hostIsSuperhost: false
},
{
hostId: "MONHOS12",
hostName: "Mike",
hostVerifications: ["email", "phone"],
hostSince: new Date("2019/01/09"),
```

hostLocation: {suburb: "Clayton", state: "Victoria", country: "Australia"},
hostResponseRate: "within a day",
hostIsSuperhost: true
},
{
hostId: "MONHOS13",
hostName: "Robyn",
hostVerifications: ["facebook", "reviews"],
hostSince: new Date("2019/01/09"),
hostLocation: {suburb: "Berwick", state: "Victoria", country: "Australia"},
hostResponseRate: "within an hour",
hostIsSuperhost: false
},
{
hostId: "MONHOS14",
hostName: "Daniel",
hostVerifications: ["email", "manual offline", "work email"],
hostSince: new Date("2019/01/09"),
hostLocation: {suburb: "Frankston", state: "Victoria", country: "Australia"},
hostResponseRate: "within a day",
hostIsSuperhost: true
},
{
hostId: "MONHOS15",
hostName: "Ron",
hostVerifications: ["facebook"],
hostSince: new Date("2019/01/09"),
hostLocation: {suburb: "Caulfield", state: "Victoria", country: "Australia"},
hostResponseRate: "within a day",
hostIsSuperhost: false

```
}
]);
```

***Q3***

```
db.host.updateMany(
{"hostResponseRate": "within an hour"},
{
  $set: {"hostIsSuperhost": true}
}
);
```

***Q4***

```
db.listing.deleteMany(
        {"availabilityPer365Days": 0}
)
```

***Q5***

```
db.listing.updateOne(
{ neighbourhood: "Monash" },
{
  $set: { "neighbourhood": "Monash City" }
}
);
```

***Q6***

```
UPDATE review SET satisfied_reason = {'space', 'price'}
WHERE review_id = 'REV11' AND listing_id = 'MONLST09' AND reviewer_id = 500011;
```

***Q7***

```
CREATE INDEX ON review (reviewer_id);

SELECT review_id, listing_id FROM review WHERE reviewer_id = 500015;

DELETE from review
WHERE review_id = 'REV15'
AND listing_id = 'MONLST13'
AND reviewer_id = 500015;
```

Command Prompt - mongo

```
> db
FIT5137_Assign
> db.host.find().pretty();
{
        "_id" : ObjectId("5d7daa5a15886b2a8a3d4c3f"),
        "hostId" : "MONHOS01",
        "hostName" : "Manju",
        "hostVerifications" : [
                "email",
                "phone",
                "reviews"
        ],
        "hostSince" : ISODate("2009-08-20T14:00:00Z"),
        "hostLocation" : {
                "suburb" : "Clayton",
                "state" : "Victoria",
                "country" : "Australia"
        },
        "hostResponseRate" : "within a day",
        "hostIsSuperhost" : false
}
{
        "_id" : ObjectId("5d7daa5a15886b2a8a3d4c40"),
        "hostId" : "MONHOS02",
        "hostName" : "Lindsay",
        "hostVerifications" : [
                "email",
                "phone",
                "reviews",
                "jumio",
                "government id"
        ],
        "hostSince" : ISODate("2009-09-15T14:00:00Z"),
        "hostLocation" : {
                "suburb" : "Clifton Hill",
                "state" : "Victoria",
                "country" : "Australia"
        },
        "hostResponseRate" : "within an hour",
        "hostIsSuperhost" : true
}
{
        "_id" : ObjectId("5d7daa5a15886b2a8a3d4c41"),
        "hostId" : "MONHOS03",
        "hostName" : "Adam",
        "hostVerifications" : [
                "email",
                "phone",
                "google",
                "reviews",
                "jumio",
                "government id",
```

MongoDB Compass Community - localhost:27017/FIT5137_Assign.host

Connect View Collection Help

My Cluster — 10 DBS 13 COLLECTIONS

- FIT5137_Assign
  - host
  - listing
- FIT5137_Assign_C3
- FIT5137_Lecture2
- admin
- config
- lecture3
- local
- restaurantdb
- test
- testDB2

localhost:27017 STANDALONE — MongoDB 4.0.11 Community

FIT5137_Assign.host

DOCUMENTS 15 — TOTAL SIZE 4.9KB — AVG. SIZE 335B — INDEXES 1 — TOTAL SIZE 32.0KB — AVG. SIZE 32.0KB

Documents | Aggregations | Explain Plan | Indexes

FILTER — OPTIONS — FIND — RESET

INSERT DOCUMENT — VIEW: LIST, TABLE — Displaying documents 1 - 15 of 15

host

| | _id ObjectId | hostId String | hostName String | hostVerifications Array | hostSince Date | hostLocation Object | host |
|---|---|---|---|---|---|---|---|
| 1 | 5d7daa5a15886b2a8a3d4c3f | "MONHOS01" | "Manju" | [] 3 elements | 2009-08-21T00:00:00.000+10:00 | {} 3 fields | "wit |
| 2 | 5d7daa5a15886b2a8a3d4c40 | "MONHOS02" | "Lindsay" | [] 5 elements | 2009-09-16T00:00:00.000+10:00 | {} 3 fields | "wit |
| 3 | 5d7daa5a15886b2a8a3d4c41 | "MONHOS03" | "Adam" | [] 8 elements | 2009-10-31T00:00:00.000+11:00 | {} 3 fields | "wit |
| 4 | 5d7daa5a15886b2a8a3d4c42 | "MONHOS04" | "Eleni" | [] 8 elements | 2009-12-03T00:00:00.000+11:00 | {} 3 fields | "wit |
| 5 | 5d7daa5a15886b2a8a3d4c43 | "MONHOS05" | "Colin" | [] 7 elements | 2009-12-22T00:00:00.000+11:00 | {} 3 fields | "wit |
| 6 | 5d7daa5a15886b2a8a3d4c44 | "MONHOS06" | "Daryl" | [] 6 elements | 2010-07-12T00:00:00.000+10:00 | {} 3 fields | "wit |
| 7 | 5d7daa5a15886b2a8a3d4c45 | "MONHOS07" | "Diana" | [] 8 elements | 2010-07-27T00:00:00.000+10:00 | {} 3 fields | "wit |
| 8 | 5d7daa5a15886b2a8a3d4c46 | "MONHOS08" | "Belinda" | [] 10 elements | 2010-08-03T00:00:00.000+10:00 | {} 3 fields | "wit |
| 9 | 5d7daa5a15886b2a8a3d4c47 | "MONHOS09" | "Allan" | [] 9 elements | 2010-08-03T00:00:00.000+10:00 | {} 3 fields | "wit |
| 10 | 5d7daa5a15886b2a8a3d4c48 | "MONHOS10" | "Vicki" | [] 6 elements | 2010-08-06T00:00:00.000+10:00 | {} 3 fields | "wit |
| 11 | 5d7dce0829a59a40aba50c4a | "MONHOS11" | "Alison" | [] 4 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |
| 12 | 5d7dce0829a59a40aba50c4b | "MONHOS12" | "Mike" | [] 2 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |
| 13 | 5d7dce0829a59a40aba50c4c | "MONHOS13" | "Robyn" | [] 2 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |
| 14 | 5d7dce0829a59a40aba50c4d | "MONHOS14" | "Daniel" | [] 3 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |
| 15 | 5d7dce0829a59a40aba50c4e | "MONHOS15" | "Ron" | [] 1 elements | 2019-01-09T00:00:00.000+11:00 | {} 3 fields | "wit |

Command Prompt - mongo

```
> db
FIT5137_Assign
> db.listing.find().pretty();
{
        "_id" : ObjectId("5d7daac415886b2a8a3d4c49"),
        "listingID" : "MONLST01",
        "name" : "Monash Beautiful House",
        "hostID" : "MONHOS14",
        "neighbourhood" : "Manningham",
        "address" : {
                "suburb" : "Clayton",
                "state" : "VIC",
                "postCode" : 3800
        },
        "latitude" : -37.773,
        "longitude" : 145.09213,
        "roomType" : "Entire home",
        "amenities" : [
                "TV",
                "Wifi",
                "Pets Allowed",
                "Family friendly",
                "24-hour check-in",
                "Self check-in"
        ],
        "pricePerNight($)" : 61,
        "priceForExtraPeople($)" : 22,
        "minimumNightsRequiredForBooking" : 1,
        "availabilityPer365Days" : 365
}
{
        "_id" : ObjectId("5d7daac415886b2a8a3d4c4a"),
        "listingID" : "MONLST02",
        "name" : "Monash Brunswick Deco",
        "hostID" : "MONHOS08",
        "neighbourhood" : "Moreland",
        "address" : {
                "suburb" : "Brunswick East",
                "state" : "VIC",
                "postCode" : 3057
        },
        "latitude" : -37.767,
        "longitude" : 144.98074,
        "roomType" : "Private room",
        "amenities" : [
                "Kitchen",
                "Toiletries",
                "Hair dryer",
                "Iron",
                "Microwave",
                "Coffee Maker",
                "Refrigerator",
```

MongoDB Compass Community - localhost:27017/FIT5137_Assign.listing

Connect View Collection Help

My Cluster — 10 DBS 13 COLLECTIONS

- FIT5137_Assign
  - host
  - listing
- FIT5137_Assign_C3
- FIT5137_Lecture2
- admin
- config
- lecture3
- local
- restaurantdb
- test
- testDB2

localhost:27017 STANDALONE — MongoDB 4.0.11 Community

FIT5137_Assign.listing

DOCUMENTS 12 — TOTAL SIZE 7.5KB — AVG. SIZE 641B — INDEXES 1 — TOTAL SIZE 32.0KB — AVG. SIZE 32.0KB

Documents | Aggregations | Explain Plan | Indexes

FILTER — OPTIONS — FIND — RESET

INSERT DOCUMENT — VIEW: LIST, TABLE — Displaying documents 1 - 12 of 12

host

| | _id ObjectId | listingID String | name String | hostID String | neighbourhood String | address Object | lat |
|---|---|---|---|---|---|---|---|
| 1 | 5d7daac415886b2a8a3d4c49 | "MONLST01" | "Monash Beautiful House" | "MONHOS14" | "Manningham" | {} 3 fields | -37. |
| 2 | 5d7daac415886b2a8a3d4c4a | "MONLST02" | "Monash Brunswick Deco" | "MONHOS08" | "Moreland" | {} 3 fields | -37. |
| 3 | 5d7daac415886b2a8a3d4c4b | "MONLST03" | "Monash Beachside Retreat" | "MONHOS01" | "Port Philip" | {} 3 fields | -37. |
| 4 | 5d7daac415886b2a8a3d4c4d | "MONLST05" | "Monash City and Sports" | "MONHOS05" | "Port Phillip" | {} 3 fields | -37. |
| 5 | 5d7daac415886b2a8a3d4c4e | "MONLST06" | "Monash Trafford Apartment" | "MONHOS06" | "Casey" | {} 3 fields | -38. |
| 6 | 5d7daac415886b2a8a3d4c50 | "MONLST08" | "Monash Home In The City" | "MONHOS02" | "Melbourne" | {} 3 fields | -37. |
| 7 | 5d7daac415886b2a8a3d4c51 | "MONLST09" | "Monash Japanese-Style" | "MONHOS11" | "Monash City" | {} 3 fields | -37. |
| 8 | 5d7daac415886b2a8a3d4c52 | "MONLST10" | "Beautiful Monash House" | "MONHOS10" | "Frankston" | {} 3 fields | -38. |
| 9 | 5d7daac415886b2a8a3d4c53 | "MONLST11" | "Fabulous Monash Richmond" | "MONHOS09" | "Yarra" | {} 3 fields | -37. |
| 10 | 5d7daac415886b2a8a3d4c54 | "MONLST12" | "Monash Central Lux" | "MONHOS12" | "Port Philip" | {} 3 fields | -37. |
| 11 | 5d7daac415886b2a8a3d4c55 | "MONLST13" | "Central Monash Warehouse Apart | "MONHOS13" | "Melbourne" | {} 3 fields | -37. |
| 12 | 5d7daac415886b2a8a3d4c56 | "MONLST14" | "Monash Near the Park" | "MONHOS03" | "Bayside" | {} 3 fields | -37. |

Command Prompt - cqlsh

```
cqlsh:fit5137_assign> select * from review;

 review_id | listing_id | reviewer_id | comments                    | review_date | review_scores_rating | reviewer_name | satisfied_reason           | sequence
-----------+------------+-------------+-----------------------------+-------------+----------------------+---------------+----------------------------+--------------
     REV03 |   MONLST02 |      500003 |     Nice View and Location  |  2017-03-22 |                  100 |       Camille |      {'location', 'view'}  | 11:37:50+1300
     REV19 |   MONLST12 |      500013 |               Good Location |  2017-03-27 |                   87 |        Anouck |        {'clean', 'space'}  | 09:37:50+1300
     REV10 |   MONLST05 |      500010 |               Friendly Host |  2017-03-23 |                   98 |     Elizabeth |         {'host', 'price'}  | 12:37:50+1300
     REV06 |   MONLST03 |      500006 |           Very Clean House  |  2017-03-22 |                   87 |          Greg |          {'host', 'view'}  | 17:37:50+1300
     REV13 |   MONLST10 |      500013 |           Very Comfortable  |  2017-03-26 |                   93 |        Anouck |          {'host', 'view'}  | 10:02:10+1300
     REV08 |   MONLST05 |      500008 |               Nice Building |  2017-03-22 |                   96 |         Klaus |      {'location', 'view'}  | 20:37:50+1300
     REV02 |   MONLST02 |      500002 |                   Good Host |  2017-03-22 |                   90 |      Johannes |                  {'host'}  | 11:37:50+1300
     REV11 |   MONLST09 |      500011 | Very Clean and comfortable  |  2017-03-23 |                  100 |         Derek |        {'price', 'space'}  | 19:37:50+1300
     REV20 |   MONLST05 |      500011 |               Nice Building |  2017-03-27 |                   96 |         Derek |          {'host', 'view'}  | 10:45:10+1300
     REV14 |   MONLST12 |      500014 |               Friendly Host |  2017-03-26 |                   85 |        Jerome |     {'clean', 'location'}  | 10:49:40+1300
     REV09 |   MONLST06 |      500009 |               Friendly Host |  2017-03-23 |                  100 |           Rox |         {'host', 'price'}  | 11:37:50+1300
     REV12 |   MONLST09 |      500012 |      Friendly and Nice Host |  2017-03-25 |                   92 |           Joy |         {'clean', 'host'}  | 10:07:40+1300
     REV07 |   MONLST04 |      500007 |               Nice Location |  2017-03-22 |                   91 |      Wolfgang |     {'location', 'price'}  | 19:37:50+1300
     REV05 |   MONLST01 |      500005 |               Good Location |  2017-03-22 |                   93 |         Adele |     {'location', 'price'}  | 15:37:50+1300
     REV01 |   MONLST02 |      500001 |              Beautiful View |  2017-03-22 |                   90 |        Miriam | {'amenities', 'location'}  | 10:37:50+1300
     REV16 |   MONLST14 |      500016 |               Good Location |  2017-03-26 |                   97 |           Joy |     {'amenities', 'view'}  | 10:48:10+1300
     REV18 |   MONLST10 |      500002 |                 Bad Service |  2017-03-26 |                   20 |      Johannes |     {'amenities', 'view'}  | 10:47:10+1300
     REV04 |   MONLST02 |      500004 |            Excellent Price  |  2017-03-22 |                   95 |         Paige |                {'price'}  | 12:37:50+1300
     REV17 |   MONLST14 |      500014 |               Bad Location  |  2017-03-26 |                   30 |        Jerome |         {'price', 'view'}  | 10:47:40+1300

(19 rows)
cqlsh:fit5137_assign>
```

# C.3. Queries

## C.3.1. MongoDB

### C.3.1.1 Import the data

```
mongoimport --db FIT5137_Assign_C3 --collection host --file
"C:\Users\Kale\Documents\Abhilash\Monash - MBIS_C6003\Sem II\FIT5137_ADB\Assignment
1\assignment_data\host.json"
```

```
mongoimport --db FIT5137_Assign_C3 --collection listing --file
"C:\Users\Kale\Documents\Abhilash\Monash - MBIS_C6003\Sem II\FIT5137_ADB\Assignment
1\assignment_data\listing.json"
```

### C.3.1.2 Create the embedded model listing_host

```
db.listing.aggregate( [
        { $lookup: { from: "host", localField: "host_id", foreignField: "host_id", as: "hostDocs" } },
        { $out : "listing_host" }
] );
```

### C.3.1.3. Create the indices

***Referencing***

db.listing.createIndex( { room_type: 1 } );

- ➔ The index created on ‘room_type’ helps increase the efficiency of the various queries inquiring about the type of the room of the listings, which are indexed in an ascending order, in the ‘listing’ collection.

db.listing.createIndex( { neighbourhood: 1, price: -1} );

- ➔ The compound index created on ‘neighbourhood’ and ‘price’ helps the various queries operating on these 2 fields in improving their processing time. ‘price’ field has been indexed in a descending order as the queries mostly focus on the higher range of priced listings in the ‘listing’ collection.

db.host.createIndex( {host_name: 1} );

- ➔ The index created on ‘host_name’ supports the efficient execution of the multiple queries which need to inspect the documents based on the host’s name of any listing in the ‘host’ collection.

***Embedding***

- ➔ Following is a set of similar indices created above for the embedded model of the 2 collections, ‘listing_host’. They also serve the same purpose as mentioned above in terms of the efficiency of the execution.

db.listing_host.createIndex( { room_type: 1 } );

db.listing_host.createIndex( { neighbourhood: 1, price: -1} );

db.listing_host.createIndex( { "hostDocs.host_name": 1} );

### C.3.1.4. Answered Queries

***Q1 - Referencing***

```
db.listing.count( {
        $and: [
        { last_review: { $gte: new Date("2018-12-01") } },
        { last_review: { $lte: new Date("2018-12-31") } }
        ]
} );
```

Command Prompt - mongo

```
> db.listing.count( {
... $and: [
... { last_review: { $gte: new Date("2018-12-01") } },
... { last_review: { $lte: new Date("2018-12-31") } }
... ]
... } );
2
>
```

***Q1 - Embedding***

```
db.listing_host.count( {
        $and: [
        { last_review: { $gte: new Date( "2018-12-01" ) } },
        { last_review: { $lte: new Date( "2018-12-31" ) } }
        ]
} );
```

Command Prompt - mongo

```
> db.listing_host.count( {
... $and: [
... { last_review: { $gte: new Date( "2018-12-01" ) } },
... { last_review: { $lte: new Date( "2018-12-31" ) } }
... ]
... } );
2
>
```

***Q2 - Referencing***

```
db.listing.aggregate( [
                { $match: { neighbourhood: "Port Phillip" } },
                { $group: { _id: "$neighbourhood", avgPrice: { $avg: "$price" } } },
                { $project: {_id: 0, neighbourhood: "$_id", avgPrice: "$avgPrice" } }
] );
```

Command Prompt - mongo

```
> db.listing.aggregate( [
... { $match: { neighbourhood: "Port Phillip" } },
... { $group: { _id: "$neighbourhood", avgPrice: { $avg: "$price" } } },
... { $project: {_id: 0, neighbourhood: "$_id", avgPrice: "$avgPrice" } }
... ] );
{ "neighbourhood" : "Port Phillip", "avgPrice" : 134.5 }
>
```

***Q2 - Embedding***

```
db.listing_host.aggregate( [
        { $match: { neighbourhood: "Port Phillip" } },
        { $group: { _id: "$neighbourhood", avgPrice: { $avg: "$price" } } },
        { $project: {_id: 0, neighbourhood: "$_id", avgPrice: "$avgPrice" } }
] );
```

Command Prompt - mongo

```
> db.listing_host.aggregate( [
... { $match: { neighbourhood: "Port Phillip" } },
... { $group: { _id: "$neighbourhood", avgPrice: { $avg: "$price" } } },
... { $project: {_id: 0, neighbourhood: "$_id", avgPrice: "$avgPrice" } }
... ] );
{ "neighbourhood" : "Port Phillip", "avgPrice" : 134.5 }
>
```

***Q3 - Referencing***

```
db.listing.aggregate( [
        { $group: { _id : "$neighbourhood" } },
        { $project: { _id: 0, neighbourhood: "$_id" } },
        { $sort: { "reviews_per_month" : -1 } },
        { $limit: 10 }
] ).pretty();
```

Command Prompt - mongo

```
> db.listing.aggregate( [
... { $group: { _id : "$neighbourhood" } },
... { $project: { _id: 0, neighbourhood: "$_id" } },
... { $sort: { "reviews_per_month" : -1 } },
... { $limit: 10 }
... ] ).pretty();
{ "neighbourhood" : "Glen Eira" }
{ "neighbourhood" : "Hobsons Bay" }
{ "neighbourhood" : "Stonnington" }
{ "neighbourhood" : "Boroondara" }
{ "neighbourhood" : "Yarra Ranges" }
{ "neighbourhood" : "Whitehorse" }
{ "neighbourhood" : "Melton" }
{ "neighbourhood" : "Wyndham" }
{ "neighbourhood" : "Brimbank" }
{ "neighbourhood" : "Bayside" }
>
```

***Q3 - Embedding***

```
db.listing_host.aggregate( [
        { $group: { _id : "$neighbourhood" } },
        { $project: { _id: 0, neighbourhood: "$_id" } },
        { $sort: { "reviews_per_month" : -1 } },
        { $limit: 10 }
] ).pretty();
```

Command Prompt - mongo

```
> db.listing_host.aggregate( [
... { $group: { _id : "$neighbourhood" } },
... { $project: { _id: 0, neighbourhood: "$_id" } },
... { $sort: { "reviews_per_month" : -1 } },
... { $limit: 10 }
... ] ).pretty();
{ "neighbourhood" : "Glen Eira" }
{ "neighbourhood" : "Hobsons Bay" }
{ "neighbourhood" : "Stonnington" }
{ "neighbourhood" : "Boroondara" }
{ "neighbourhood" : "Yarra Ranges" }
{ "neighbourhood" : "Whitehorse" }
{ "neighbourhood" : "Melton" }
{ "neighbourhood" : "Wyndham" }
{ "neighbourhood" : "Brimbank" }
{ "neighbourhood" : "Bayside" }
>
```

***Q4 - Referencing***

db.listing.aggregate( [

{ $group: { _id: null, minReviews: { $min: "$number_of_reviews" }, maxReviews: { $max: "$number_of_reviews" } } },

{ $project: { _id: 0, minReviews: 1, maxReviews: 1, rangeOfNumberOfReviews: { $subtract: [ "$maxReviews", "$minReviews" ] } } }

] );

Command Prompt - mongo

```
> db.listing.aggregate( [
... { $group: { _id: null, minReviews: { $min: "$number_of_reviews" }, maxReviews: { $max: "$number_of_reviews" } } },
... { $project: { _id: 0, minReviews: 1, maxReviews: 1, rangeOfNumberOfReviews: { $subtract: [ "$maxReviews", "$minReviews" ] } } }
... ] );
{ "minReviews" : 0, "maxReviews" : 527, "rangeOfNumberOfReviews" : 527 }
>
```

***Q4 - Embedding***

db.listing_host.aggregate( [

{ $group: { _id: null, minReviews: { $min: "$number_of_reviews" }, maxReviews: { $max: "$number_of_reviews" } } },

{ $project: { _id: 0, minReviews: 1, maxReviews: 1, rangeOfNumberOfReviews: { $subtract: [ "$maxReviews", "$minReviews" ] } } }

] );

Command Prompt - mongo

```
> db.listing_host.aggregate( [
... { $group: { _id: null, minReviews: { $min: "$number_of_reviews" }, maxReviews: { $max: "$number_of_reviews" } } },
... { $project: { _id: 0, minReviews: 1, maxReviews: 1, rangeOfNumberOfReviews: { $subtract: [ "$maxReviews", "$minReviews" ] } } }
... ] );
{ "minReviews" : 0, "maxReviews" : 527, "rangeOfNumberOfReviews" : 527 }
>
```

***Q5 - Referencing***

```
db.listing.aggregate( [
        { $group : { _id : "$room_type", numberOfOccurrence: { $sum: 1 } } },
        { $project: { _id: 0, roomType: "$_id", numberOfOccurrence: "$numberOfOccurrence" } },
        { $sort : { numberOfOccurrence : -1 } },
        { $limit : 1 }
] );
```

Command Prompt - mongo

```
> db.listing.aggregate( [
... { $group : { _id : "$room_type", numberOfOccurrence: { $sum: 1 } } },
... { $project: { _id: 0, roomType: "$_id", numberOfOccurrence: "$numberOfOccurrence" } },
... { $sort : { numberOfOccurrence : -1 } },
... { $limit : 1 }
... ] );
{ "roomType" : "Private room", "numberOfOccurrence" : 50 }
>
```

***Q5 - Embedding***

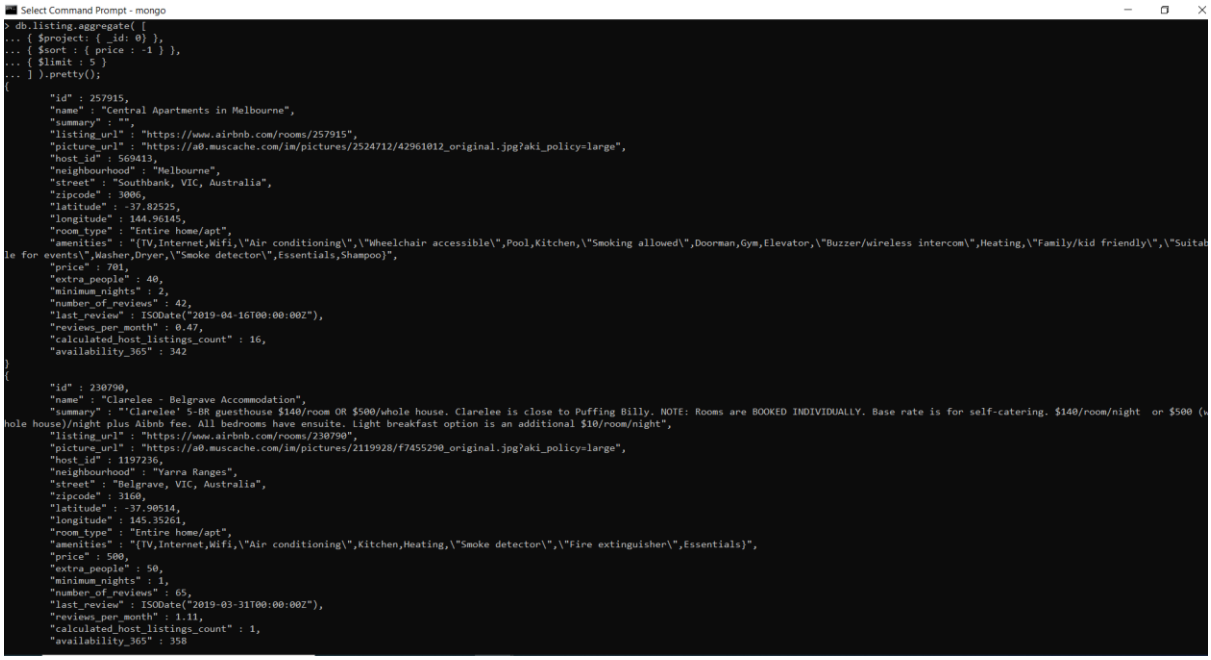
```
db.listing_host.aggregate( [
        { $group : { _id : "$room_type", numberOfOccurrence: { $sum: 1 } } },
        { $project: { _id: 0, roomType: "$_id", numberOfOccurrence: "$numberOfOccurrence" } },
        { $sort : { numberOfOccurrence : -1 } },
        { $limit : 1 }
] );
```

Command Prompt - mongo

```
> db.listing_host.aggregate( [
... { $group : { _id : "$room_type", numberOfOccurrence: { $sum: 1 } } },
... { $project: { _id: 0, roomType: "$_id", numberOfOccurrence: "$numberOfOccurrence" } },
... { $sort : { numberOfOccurrence : -1 } },
... { $limit : 1 }
... ] );
{ "roomType" : "Private room", "numberOfOccurrence" : 50 }
>
```

***Q6 - Referencing***

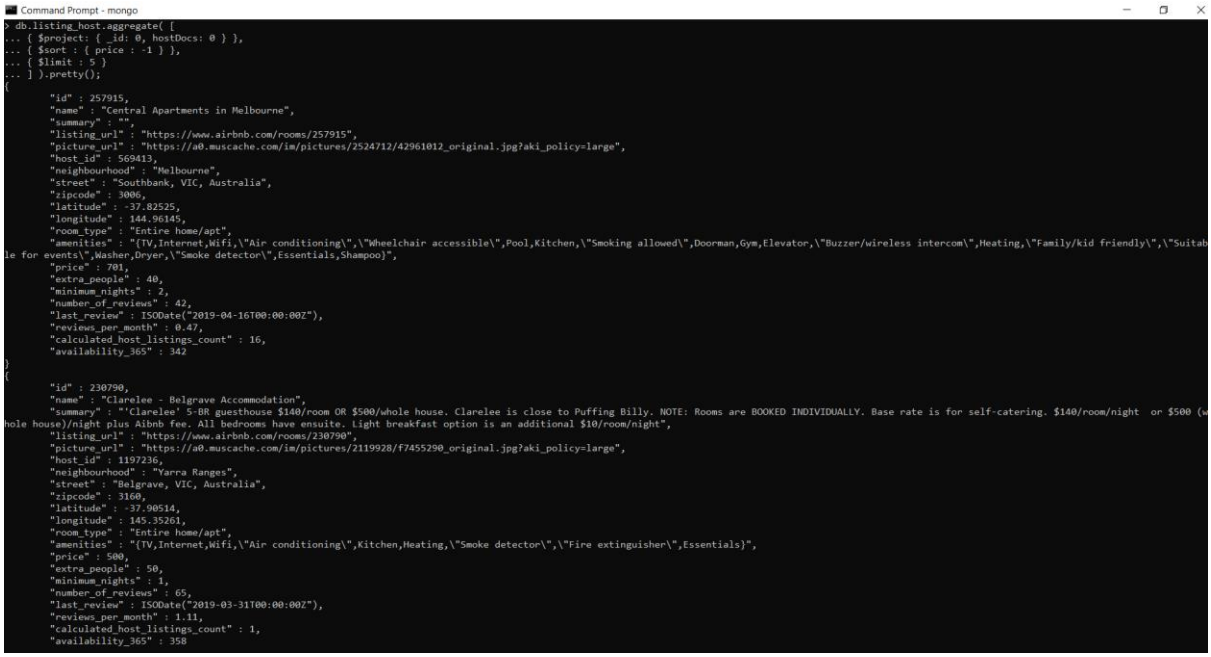
```
db.listing.aggregate( [
        { $project: { _id: 0} },
        { $sort : { price : -1 } },
        { $limit : 5 }
] ).pretty();
```



```
Select Command Prompt - mongo
> db.listing.aggregate( [
... { $project: { _id: 0} },
... { $sort : { price : -1 } },
... { $limit : 5 }
... ] ).pretty();
{
        "id" : 257915,
        "name" : "Central Apartments in Melbourne",
        "summary" : "",
        "listing_url" : "https://www.airbnb.com/rooms/257915",
        "picture_url" : "https://a0.muscache.com/im/pictures/2524712/42961012_original.jpg?aki_policy=large",
        "host_id" : 569413,
        "neighbourhood" : "Melbourne",
        "street" : "Southbank, VIC, Australia",
        "zipcode" : 3006,
        "latitude" : -37.82525,
        "longitude" : 144.96145,
        "room_type" : "Entire home/apt",
        "amenities" : "{TV,Internet,Wifi,\"Air conditioning\",\"Wheelchair accessible\",Pool,Kitchen,\"Smoking allowed\",Doorman,Gym,Elevator,\"Buzzer/wireless intercom\",Heating,\"Family/kid friendly\",\"Suitab
le for events\",Washer,Dryer,\"Smoke detector\",Essentials,Shampoo}",
        "price" : 701,
        "extra_people" : 40,
        "minimum_nights" : 2,
        "number_of_reviews" : 42,
        "last_review" : ISODate("2019-04-16T00:00:00Z"),
        "reviews_per_month" : 0.47,
        "calculated_host_listings_count" : 16,
        "availability_365" : 342
}
{
        "id" : 230790,
        "name" : "Clarelee - Belgrave Accommodation",
        "summary" : "'Clarelee' 5-BR guesthouse $140/room OR $500/whole house. Clarelee is close to Puffing Billy. NOTE: Rooms are BOOKED INDIVIDUALLY. Base rate is for self-catering. $140/room/night  or $500 (w
hole house)/night plus Aibnb fee. All bedrooms have ensuite. Light breakfast option is an additional $10/room/night",
        "listing_url" : "https://www.airbnb.com/rooms/230790",
        "picture_url" : "https://a0.muscache.com/im/pictures/2119928/f7455290_original.jpg?aki_policy=large",
        "host_id" : 1197236,
        "neighbourhood" : "Yarra Ranges",
        "street" : "Belgrave, VIC, Australia",
        "zipcode" : 3160,
        "latitude" : -37.90514,
        "longitude" : 145.35261,
        "room_type" : "Entire home/apt",
        "amenities" : "{TV,Internet,Wifi,\"Air conditioning\",Kitchen,Heating,\"Smoke detector\",\"Fire extinguisher\",Essentials}",
        "price" : 500,
        "extra_people" : 50,
        "minimum_nights" : 1,
        "number_of_reviews" : 65,
        "last_review" : ISODate("2019-03-31T00:00:00Z"),
        "reviews_per_month" : 1.11,
        "calculated_host_listings_count" : 1,
        "availability_365" : 358
```

***Q6 - Embedding***

```
db.listing_host.aggregate( [
        { $project: { _id: 0, hostDocs: 0 } },
        { $sort : { price : -1 } },
        { $limit : 5 }
] ).pretty();
```



```
Command Prompt - mongo
> db.listing_host.aggregate( [
... { $project: { _id: 0, hostDocs: 0 } },
... { $sort : { price : -1 } },
... { $limit : 5 }
... ] ).pretty();
{
        "id" : 257915,
        "name" : "Central Apartments in Melbourne",
        "summary" : "",
        "listing_url" : "https://www.airbnb.com/rooms/257915",
        "picture_url" : "https://a0.muscache.com/im/pictures/2524712/42961012_original.jpg?aki_policy=large",
        "host_id" : 569413,
        "neighbourhood" : "Melbourne",
        "street" : "Southbank, VIC, Australia",
        "zipcode" : 3006,
        "latitude" : -37.82525,
        "longitude" : 144.96145,
        "room_type" : "Entire home/apt",
        "amenities" : "{TV,Internet,Wifi,\"Air conditioning\",\"Wheelchair accessible\",Pool,Kitchen,\"Smoking allowed\",Doorman,Gym,Elevator,\"Buzzer/wireless intercom\",Heating,\"Family/kid friendly\",\"Suitab
le for events\",Washer,Dryer,\"Smoke detector\",Essentials,Shampoo}",
        "price" : 701,
        "extra_people" : 40,
        "minimum_nights" : 2,
        "number_of_reviews" : 42,
        "last_review" : ISODate("2019-04-16T00:00:00Z"),
        "reviews_per_month" : 0.47,
        "calculated_host_listings_count" : 16,
        "availability_365" : 342
}
{
        "id" : 230790,
        "name" : "Clarelee - Belgrave Accommodation",
        "summary" : "'Clarelee' 5-BR guesthouse $140/room OR $500/whole house. Clarelee is close to Puffing Billy. NOTE: Rooms are BOOKED INDIVIDUALLY. Base rate is for self-catering. $140/room/night  or $500 (w
hole house)/night plus Aibnb fee. All bedrooms have ensuite. Light breakfast option is an additional $10/room/night",
        "listing_url" : "https://www.airbnb.com/rooms/230790",
        "picture_url" : "https://a0.muscache.com/im/pictures/2119928/f7455290_original.jpg?aki_policy=large",
        "host_id" : 1197236,
        "neighbourhood" : "Yarra Ranges",
        "street" : "Belgrave, VIC, Australia",
        "zipcode" : 3160,
        "latitude" : -37.90514,
        "longitude" : 145.35261,
        "room_type" : "Entire home/apt",
        "amenities" : "{TV,Internet,Wifi,\"Air conditioning\",Kitchen,Heating,\"Smoke detector\",\"Fire extinguisher\",Essentials}",
        "price" : 500,
        "extra_people" : 50,
        "minimum_nights" : 1,
        "number_of_reviews" : 65,
        "last_review" : ISODate("2019-03-31T00:00:00Z"),
        "reviews_per_month" : 1.11,
        "calculated_host_listings_count" : 1,
        "availability_365" : 358
```

***Q7 - Referencing***

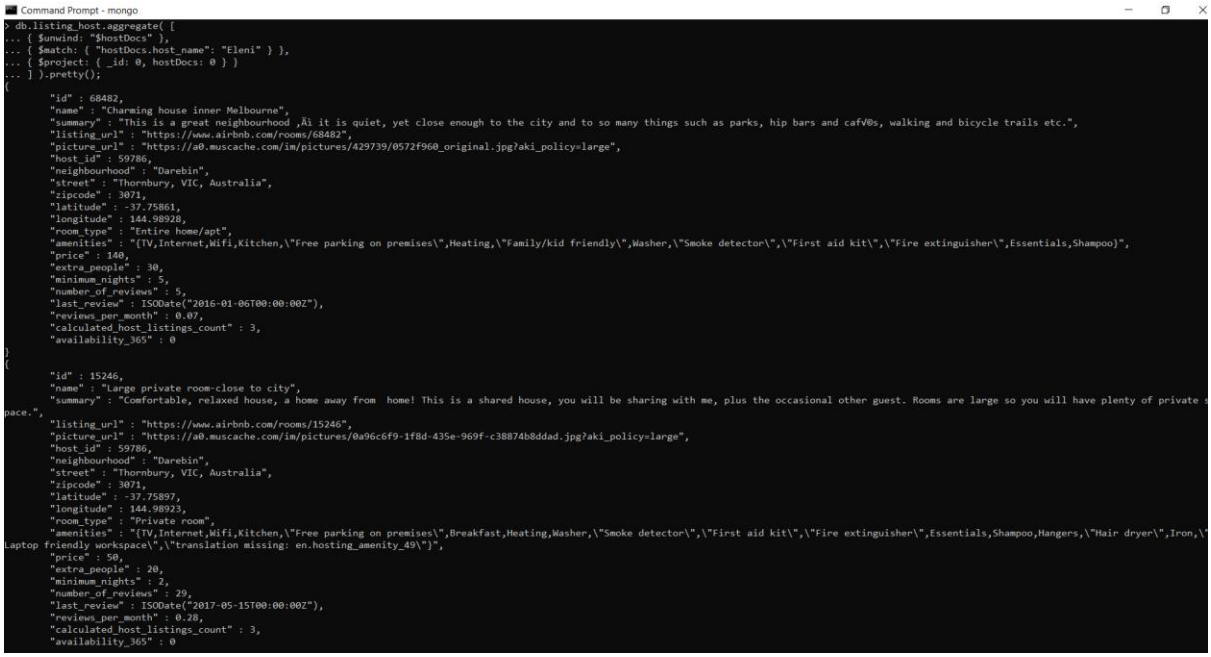
```
db.host.aggregate( [
        { $lookup : { from : "listing", localField : "host_id", foreignField : "host_id", as : "listingDocs" } },
        { $match: { host_name: "Eleni" } },
        { $project: { _id: 0, host_name: 1, listingDocs: 1 } }
] ).pretty();
```

Command Prompt - mongo

```
> db.host.aggregate( [
... { $lookup : { from : "listing", localField : "host_id", foreignField : "host_id", as : "listingDocs" } },
... { $match: { host_name: "Eleni" } },
... { $project: { _id: 0, host_name: 1, listingDocs: 1 } }
... ] ).pretty();
{
        "host_name" : "Eleni",
        "listingDocs" : [
                {
                        "_id" : ObjectId("5d7f2720e834bf739087a3a9"),
                        "id" : 68482,
                        "name" : "Charming house inner Melbourne",
                        "summary" : "This is a great neighbourhood ,Äì it is quiet, yet close enough to the city and to so many things such as parks, hip bars and cafv0s, walking and bicycle trails etc.",
                        "listing_url" : "https://www.airbnb.com/rooms/68482",
                        "picture_url" : "https://a0.muscache.com/im/pictures/429739/0572f960_original.jpg?aki_policy=large",
                        "host_id" : 59786,
                        "neighbourhood" : "Darebin",
                        "street" : "Thornbury, VIC, Australia",
                        "zipcode" : 3071,
                        "latitude" : -37.75861,
                        "longitude" : 144.98928,
                        "room_type" : "Entire home/apt",
                        "amenities" : "{TV,Internet,Wifi,Kitchen,\"Free parking on premises\",Heating,\"Family/kid friendly\",Washer,\"Smoke detector\",\"First aid kit\",\"Fire extinguisher\",Essentials,Shampoo}
",
                        "price" : 140,
                        "extra_people" : 30,
                        "minimum_nights" : 5,
                        "number_of_reviews" : 5,
                        "last_review" : ISODate("2016-01-06T00:00:00Z"),
                        "reviews_per_month" : 0.07,
                        "calculated_host_listings_count" : 3,
                        "availability_365" : 0
                },
                {
                        "_id" : ObjectId("5d7f2720e834bf739087a3bc"),
                        "id" : 15246,
                        "name" : "Large private room-close to city",
                        "summary" : "Comfortable, relaxed house, a home away from  home! This is a shared house, you will be sharing with me, plus the occasional other guest. Rooms are large so you will have ple
nty of private space.",
                        "listing_url" : "https://www.airbnb.com/rooms/15246",
                        "picture_url" : "https://a0.muscache.com/im/pictures/0a96c6f9-1f8d-435e-969f-c38874b8ddad.jpg?aki_policy=large",
                        "host_id" : 59786,
                        "neighbourhood" : "Darebin",
                        "street" : "Thornbury, VIC, Australia",
                        "zipcode" : 3071,
                        "latitude" : -37.75897,
                        "longitude" : 144.98923,
                        "room_type" : "Private room",
                        "amenities" : "{TV,Internet,Wifi,Kitchen,\"Free parking on premises\",Breakfast,Heating,Washer,\"Smoke detector\",\"First aid kit\",\"Fire extinguisher\",Essentials,Shampoo,Hangers,\"Hair
 dryer\",Iron,\"Laptop friendly workspace\",\"translation missing: en.hosting_amenity_49\"}",
                        "price" : 50,
                        "extra_people" : 20,
```

***Q7 - Embedding***

```
db.listing_host.aggregate( [
        { $unwind: "$hostDocs" },
        { $match: { "hostDocs.host_name": "Eleni" } },
        { $project: { _id: 0, hostDocs: 0 } }
] ).pretty();
```



```
Command Prompt - mongo
> db.listing_host.aggregate( [
... { $unwind: "$hostDocs" },
... { $match: { "hostDocs.host_name": "Eleni" } },
... { $project: { _id: 0, hostDocs: 0 } }
... ] ).pretty();
{
        "id" : 68482,
        "name" : "Charming house inner Melbourne",
        "summary" : "This is a great neighbourhood ,Âì it is quiet, yet close enough to the city and to so many things such as parks, hip bars and cafv©s, walking and bicycle trails etc.",
        "listing_url" : "https://www.airbnb.com/rooms/68482",
        "picture_url" : "https://a0.muscache.com/im/pictures/429739/0572f960_original.jpg?aki_policy=large",
        "host_id" : 59786,
        "neighbourhood" : "Darebin",
        "street" : "Thornbury, VIC, Australia",
        "zipcode" : 3071,
        "latitude" : -37.75861,
        "longitude" : 144.98928,
        "room_type" : "Entire home/apt",
        "amenities" : "{TV,Internet,Wifi,Kitchen,\"Free parking on premises\",Heating,\"Family/kid friendly\",Washer,\"Smoke detector\",\"First aid kit\",\"Fire extinguisher\",Essentials,Shampoo}",
        "price" : 140,
        "extra_people" : 30,
        "minimum_nights" : 5,
        "number_of_reviews" : 5,
        "last_review" : ISODate("2016-01-06T00:00:00Z"),
        "reviews_per_month" : 0.07,
        "calculated_host_listings_count" : 3,
        "availability_365" : 0
}
{
        "id" : 15246,
        "name" : "Large private room-close to city",
        "summary" : "Comfortable, relaxed house, a home away from  home! This is a shared house, you will be sharing with me, plus the occasional other guest. Rooms are large so you will have plenty of private s
pace.",
        "listing_url" : "https://www.airbnb.com/rooms/15246",
        "picture_url" : "https://a0.muscache.com/im/pictures/0a96c6f9-1f8d-435e-969f-c38874b8ddad.jpg?aki_policy=large",
        "host_id" : 59786,
        "neighbourhood" : "Darebin",
        "street" : "Thornbury, VIC, Australia",
        "zipcode" : 3071,
        "latitude" : -37.75897,
        "longitude" : 144.98923,
        "room_type" : "Private room",
        "amenities" : "{TV,Internet,Wifi,Kitchen,\"Free parking on premises\",Breakfast,Heating,Washer,\"Smoke detector\",\"First aid kit\",\"Fire extinguisher\",Essentials,Shampoo,Hangers,\"Hair dryer\",Iron,\"
Laptop friendly workspace\",\"translation missing: en.hosting_amenity_49\"}",
        "price" : 50,
        "extra_people" : 20,
        "minimum_nights" : 2,
        "number_of_reviews" : 29,
        "last_review" : ISODate("2017-05-15T00:00:00Z"),
        "reviews_per_month" : 0.28,
        "calculated_host_listings_count" : 3,
        "availability_365" : 0
```

***Q8 - Referencing***

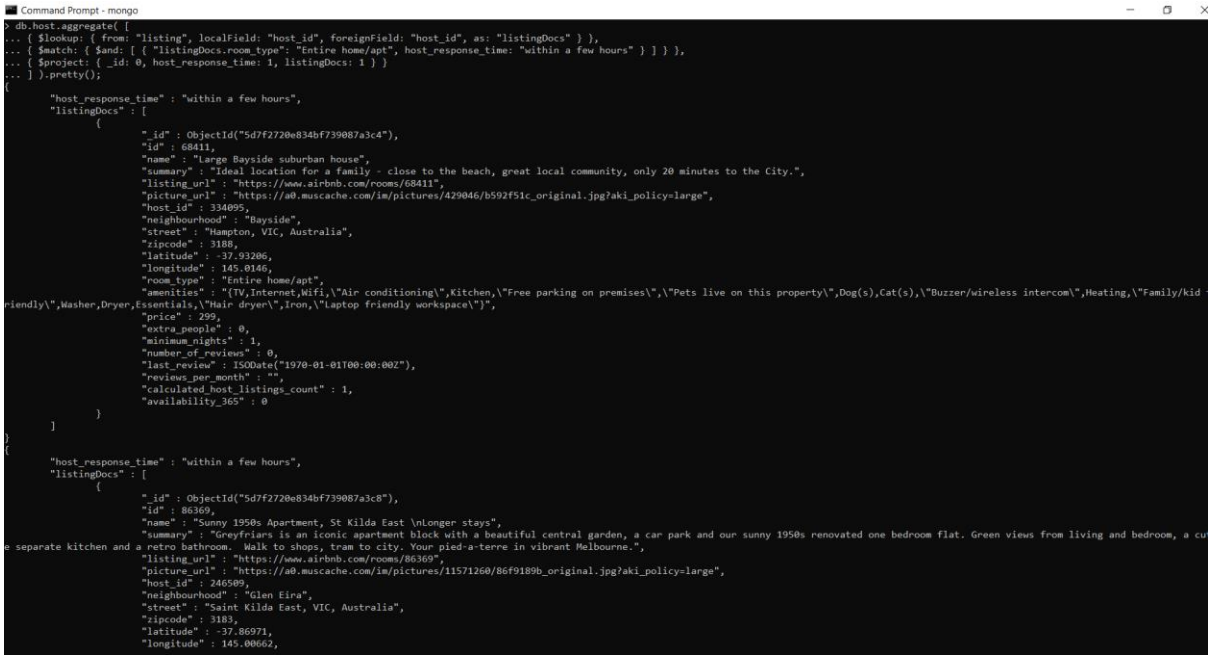
db.host.aggregate( [

{ $lookup: { from: "listing", localField: "host_id", foreignField: "host_id", as: "listingDocs" } },

{ $match: { $and: [ { "listingDocs.room_type": "Entire home/apt", host_response_time: "within a few hours" } ] } },

{ $project: { _id: 0, host_response_time: 1, listingDocs: 1 } }

] ).pretty();



```
Command Prompt - mongo
> db.host.aggregate( [
... { $lookup: { from: "listing", localField: "host_id", foreignField: "host_id", as: "listingDocs" } },
... { $match: { $and: [ { "listingDocs.room_type": "Entire home/apt", host_response_time: "within a few hours" } ] } },
... { $project: { _id: 0, host_response_time: 1, listingDocs: 1 } }
... ] ).pretty();
{
        "host_response_time" : "within a few hours",
        "listingDocs" : [
                {
                        "_id" : ObjectId("5d7f2720e834bf739087a3c4"),
                        "id" : 68411,
                        "name" : "Large Bayside suburban house",
                        "summary" : "Ideal location for a family - close to the beach, great local community, only 20 minutes to the City.",
                        "listing_url" : "https://www.airbnb.com/rooms/68411",
                        "picture_url" : "https://a0.muscache.com/im/pictures/429046/b592f51c_original.jpg?aki_policy=large",
                        "host_id" : 334095,
                        "neighbourhood" : "Bayside",
                        "street" : "Hampton, VIC, Australia",
                        "zipcode" : 3188,
                        "latitude" : -37.93206,
                        "longitude" : 145.0146,
                        "room_type" : "Entire home/apt",
                        "amenities" : "{TV,Internet,Wifi,\"Air conditioning\",Kitchen,\"Free parking on premises\",\"Pets live on this property\",Dog(s),Cat(s),\"Buzzer/wireless intercom\",Heating,\"Family/kid f
riendly\",Washer,Dryer,Essentials,\"Hair dryer\",Iron,\"Laptop friendly workspace\"}",
                        "price" : 299,
                        "extra_people" : 0,
                        "minimum_nights" : 1,
                        "number_of_reviews" : 0,
                        "last_review" : ISODate("1970-01-01T00:00:00Z"),
                        "reviews_per_month" : "",
                        "calculated_host_listings_count" : 1,
                        "availability_365" : 0
                }
        ]
}
{
        "host_response_time" : "within a few hours",
        "listingDocs" : [
                {
                        "_id" : ObjectId("5d7f2720e834bf739087a3c8"),
                        "id" : 86369,
                        "name" : "Sunny 1950s Apartment, St Kilda East \nLonger stays",
                        "summary" : "Greyfriars is an iconic apartment block with a beautiful central garden, a car park and our sunny 1950s renovated one bedroom flat. Green views from living and bedroom, a cu
e separate kitchen and a retro bathroom.  Walk to shops, tram to city. Your pied-a-terre in vibrant Melbourne.",
                        "listing_url" : "https://www.airbnb.com/rooms/86369",
                        "picture_url" : "https://a0.muscache.com/im/pictures/11571260/86f9189b_original.jpg?aki_policy=large",
                        "host_id" : 246509,
                        "neighbourhood" : "Glen Eira",
                        "street" : "Saint Kilda East, VIC, Australia",
                        "zipcode" : 3183,
                        "latitude" : -37.86971,
                        "longitude" : 145.00662,
```

***Q8 - Embedding***

```
db.listing_host.aggregate( [
        { $unwind: "$hostDocs" },
        { $match: { $and: [ { room_type: "Entire home/apt", "hostDocs.host_response_time": "within a few hours" } ] } },
        { $project: { _id: 0, hostDocs: 0 } }
] ).pretty();
```

Command Prompt - mongo

```
> db.listing_host.aggregate( [
... { $unwind: "$hostDocs" },
... { $match: { $and: [ { room_type: "Entire home/apt", "hostDocs.host_response_time": "within a few hours" } ] } },
... { $project: { _id: 0, hostDocs: 0 } }
... ] ).pretty();
{
        "id" : 68411,
        "name" : "Large Bayside suburban house",
        "summary" : "Ideal location for a family - close to the beach, great local community, only 20 minutes to the City.",
        "listing_url" : "https://www.airbnb.com/rooms/68411",
        "picture_url" : "https://a0.muscache.com/im/pictures/429046/b592f51c_original.jpg?aki_policy=large",
        "host_id" : 334095,
        "neighbourhood" : "Bayside",
        "street" : "Hampton, VIC, Australia",
        "zipcode" : 3188,
        "latitude" : -37.93206,
        "longitude" : 145.0146,
        "room_type" : "Entire home/apt",
        "amenities" : "{TV,Internet,Wifi,\"Air conditioning\",Kitchen,\"Free parking on premises\",\"Pets live on this property\",Dog(s),Cat(s),\"Buzzer/wireless intercom\",Heating,\"Family/kid friendly\",Washer
,Dryer,Essentials,\"Hair dryer\",Iron,\"Laptop friendly workspace\"}",
        "price" : 299,
        "extra_people" : 0,
        "minimum_nights" : 1,
        "number_of_reviews" : 0,
        "last_review" : ISODate("1970-01-01T00:00:00Z"),
        "reviews_per_month" : "",
        "calculated_host_listings_count" : 1,
        "availability_365" : 0
}
{
        "id" : 161033,
        "name" : "Healesville Yarra Valley Cottage",
        "summary" : "The cottage is a small self contained cabin on a picturesque property. It is simple, comfortable, clean and private. This space is perfect for people  looking for a quiet stay in a rural loc
ation with access to all the Yarra Valley offers.",
        "listing_url" : "https://www.airbnb.com/rooms/161033",
        "picture_url" : "https://a0.muscache.com/im/pictures/17370249/68633870_original.jpg?aki_policy=large",
        "host_id" : 770883,
        "neighbourhood" : "Yarra Ranges",
        "street" : "Chum Creek, VIC, Australia",
        "zipcode" : 3777,
        "latitude" : -37.63282,
        "longitude" : 145.49766,
        "room_type" : "Entire home/apt",
        "amenities" : "{\"Free parking on premises\",Breakfast,Heating,Washer,\"Smoke detector\",Essentials,Shampoo,\"Lock on bedroom door\",Hangers,\"Hair dryer\",\"Laptop friendly workspace\",\"translation mis
sing: en.hosting_amenity_49\",\"translation missing: en.hosting_amenity_50\",\"Private living room\",\"Private entrance\",\"Hot water\",\"Bed linens\",\"Extra pillows and blankets\",Microwave,\"Coffee maker\",Re
frigerator,\"Dishes and silverware\",\"Patio or balcony\",\"Garden or backyard\",\"Luggage dropoff allowed\"}",
        "price" : 81,
        "extra_people" : 0,
        "minimum_nights" : 2,
        "number_of_reviews" : 219,
        "last_review" : ISODate("2019-07-07T00:00:00Z"),
        "reviews_per_month" : 2.29,
```

***Q9 - Referencing***

```
db.host.aggregate( [
        { $lookup: { from: "listing", localField: "host_id", foreignField: "host_id", as: "listingDocs" } },
        { $match: { $and: [ { host_name: "Colin" }, { "listingDocs.amenities": { $regex: "Internet" } }, { "listingDocs.amenities": { $regex: "Gym" } } ] } },
        { $project: { _id: 0, host_name: 1, listingDocs: 1 } }
] ).pretty();
```

Command Prompt - mongo

```
> db.host.aggregate( [
... { $lookup: { from: "listing", localField: "host_id", foreignField: "host_id", as: "listingDocs" } },
... { $match: { $and: [ { host_name: "Colin" }, { "listingDocs.amenities": { $regex: "Internet" } }, { "listingDocs.amenities": { $regex: "Gym" } } ] } },
... { $project: { _id: 0, host_name: 1, listingDocs: 1 } }
... ] ).pretty();
>
```

***Q9 - Embedding***

```
db.listing_host.aggregate( [
        { $unwind: "$hostDocs" },
        { $match: { $and: [ { "hostDocs.host_name": "Colin" }, { amenities: { $regex: "Internet" } }, { amenities: { $regex: "Gym" } } ] } },
        { $project: { _id: 0, hostDocs: 0 } }
] ).pretty();
```

Command Prompt - mongo

```
> db.listing_host.aggregate( [
... { $unwind: "$hostDocs" },
... { $match: { $and: [ { "hostDocs.host_name": "Colin" }, { amenities: { $regex: "Internet" } }, { amenities: { $regex: "Gym" } } ] } },
... { $project: { _id: 0, hostDocs: 0 } }
... ] ).pretty();
>
```

***Q10 - Referencing***

db.listing.aggregate( [

{ $match: { $and: [ {street: { $regex: "Clayton" }, name: { $regex: "Beautiful" } } ] } },

{ $project: { _id: 0, price: 1, room_type: 1 } }

] ).pretty();

Command Prompt - mongo

```
> db.listing.aggregate( [
... { $match: { $and: [ {street: { $regex: "Clayton" }, name: { $regex: "Beautiful" } } ] } },
... { $project: { _id: 0, price: 1, room_type: 1 } }
... ] ).pretty();
>
```

***Q10 - Embedding***

db.listing_host.aggregate( [

{ $match: { $and: [ { street: { $regex: "Clayton" }, name: { $regex: "Beautiful" } } ] } },

{ $project: { _id: 0, price: 1, room_type: 1 } }

] ).pretty();

Command Prompt - mongo

```
> db.listing_host.aggregate( [
... { $match: { $and: [ { street: { $regex: "Clayton" }, name: { $regex: "Beautiful" } } ] } },
... { $project: { _id: 0, price: 1, room_type: 1 } }
... ] ).pretty();
>
```

***Q11 - Referencing***

db.listing.aggregate( [

{ $project: { _id: 0, listingDetails: { $concat: [ "$name", ", ", { $rtrim: { input: "$street", chars: ", Australia"} }, ", ", { $substr: [ "$zipcode", 0, -1 ] }, ", ", "$neighbourhood" ] } } },

{ $sort: { "listingDetails": 1 } }

] );

Command Prompt - mongo

```
> db.listing.aggregate( [
... { $project: { _id: 0, listingDetails: { $concat: [ "$name", ", ", { $rtrim: { input: "$street", chars: ", Australia"} }, ", ", { $substr: [ "$zipcode", 0, -1 ] }, ", ", "$neighbourhood" ] } } },
... { $sort: { "listingDetails": 1 } }
... ] );
{ "listingDetails" : "(1) Stylish, East Melb 1bed apt, East Melbourne, VIC, 3002, Melbourne" }
{ "listingDetails" : "**just renovated** Bright & Spacious StKildaEast**, St Kilda East, VIC, 3183, Port Phillip" }
{ "listingDetails" : "2 bedrooms-ideal for friends/family, Prahran, VIC, 3181, Stonnington" }
{ "listingDetails" : "3 Bedroom Apartment MT111, Southbank, VIC, 3006, Melbourne" }
{ "listingDetails" : "3 Level Loft with Views close to City, Richmond, VIC, 3121, Yarra" }
{ "listingDetails" : "5 mins Melbourne CBD in Richmond., Richmond, VIC, 3121, Yarra" }
{ "listingDetails" : "A POP-UP BEDROOM NEAR CITY AND AIRPORT, Reservoir, VIC, 3073, Darebin" }
{ "listingDetails" : "A Room Near the Park, Melbourne, VIC, 3187, Bayside" }
{ "listingDetails" : "AWESOME BEACH HOUSE BnB -, Mordialloc, VIC, 3195, Kingston" }
{ "listingDetails" : "Albert Park: Between St Kilda & CBD - Perfect spot, Albert Park, VIC, 3206, Port Phillip" }
{ "listingDetails" : "Angelus Retreat @ Mount Dandenong, Menzies Creek, VIC, 3159, Yarra Ranges" }
{ "listingDetails" : "Attractive room in leafy Deepdene, Balwyn, VIC, 3103, Boroondara" }
{ "listingDetails" : "Beautiful Northcote home with heart -close to city, Melbourne, VIC, 3070, Darebin" }
{ "listingDetails" : "Beautiful Room & House, Bulleen, VIC, 3105, Manningham" }
{ "listingDetails" : "Beautifully Appointed Apt - Central, North Melbourne, VIC, 3051, Melbourne" }
{ "listingDetails" : "Best, west of Melbourne-Wifi & Spa1, Derrimut, VIC, 3030, Brimbank" }
{ "listingDetails" : "Best, west of Melbourne-Wifi & spa2, Derrimut, VIC, 3030, Brimbank" }
{ "listingDetails" : "Blissful Beachside Port Melbourne Warehouse, Port Melbourne, VIC, 3207, Port Phillip" }
{ "listingDetails" : "Bohemian Bungalow in Brunswick, Melbourne, VIC, 3056, Moreland" }
{ "listingDetails" : "CLOSE TO CITY & MELBOURNE AIRPORT, Reservoir, VIC, 3073, Darebin" }
Type "it" for more
> it
{ "listingDetails" : "Carnarvon, Armadale/St KldaE/Caulfl, Caulfield North, VIC, 3161, Glen Eira" }
{ "listingDetails" : "Central Apartments in Melbourne, Southbank, VIC, 3006, Melbourne" }
{ "listingDetails" : "Central City Warehouse Apartment, Melbourne, VIC, 3000, Melbourne" }
{ "listingDetails" : "Charming house inner Melbourne, Thornbury, VIC, 3071, Darebin" }
{ "listingDetails" : "City Location-Perfect for Singles, Melbourne, VIC, 3000, Melbourne" }
{ "listingDetails" : "City's edge Penthouse - private bath, views!, South Melbourne, VIC, 3205, Port Phillip" }
{ "listingDetails" : "Clarelee - Belgrave Accommodation, Belgrave, VIC, 3160, Yarra Ranges" }
{ "listingDetails" : "Classic Fitzroy Terrace (w/ cat) - walk to Tennis, Fitzroy, VIC, 3065, Yarra" }
{ "listingDetails" : "Classic Victorian Armadale Home :) Victorian House, Armadale, VIC, 3143, Stonnington" }
{ "listingDetails" : "Collingwood 2 bedrm Warehouse Apt, Collingwood, VIC, 3066, Yarra" }
{ "listingDetails" : "Comfy room in Oakleigh South Melbourne, Oakleigh South, VIC, 3167, Kingston" }
{ "listingDetails" : "Convenient Spot in Mt Waverley, Mount Waverley, VIC, 3149, Monash" }
{ "listingDetails" : "Cool spot near chapel st!, Prahran, VIC, 3181, Stonnington" }
{ "listingDetails" : "Cosy nest in vibrant eclectic area, Abbotsford, VIC, 3067, Yarra" }
{ "listingDetails" : "Cosy retreat with amazing views, Parkville, VIC, 3052, Melbourne" }
{ "listingDetails" : "Double Guest Room Ensuited, South Melbourne, VIC, 3205, Port Phillip" }
{ "listingDetails" : "Double Room, Private Bathroom, Breakfast & Air Con, Hawthorn East, VIC, 3123, Boroondara" }
{ "listingDetails" : "Elwood CHIC 1BR+WALK TO VILLAGE+PARKING+WIFI, Elwood, VIC, 3184, Port Phillip" }
{ "listingDetails" : "Elwood SPACIOUS OPEN PLAN EXEC 2BR+PARKING+WIFI+AC, Elwood, VIC, 3184, Port Phillip" }
{ "listingDetails" : "Elwood VILLAGE VIBE 1BR+BEACHSIDE+PARKING+WIFI, Elwood, VIC, 3184, Port Phillip" }
Type "it" for more
>
```

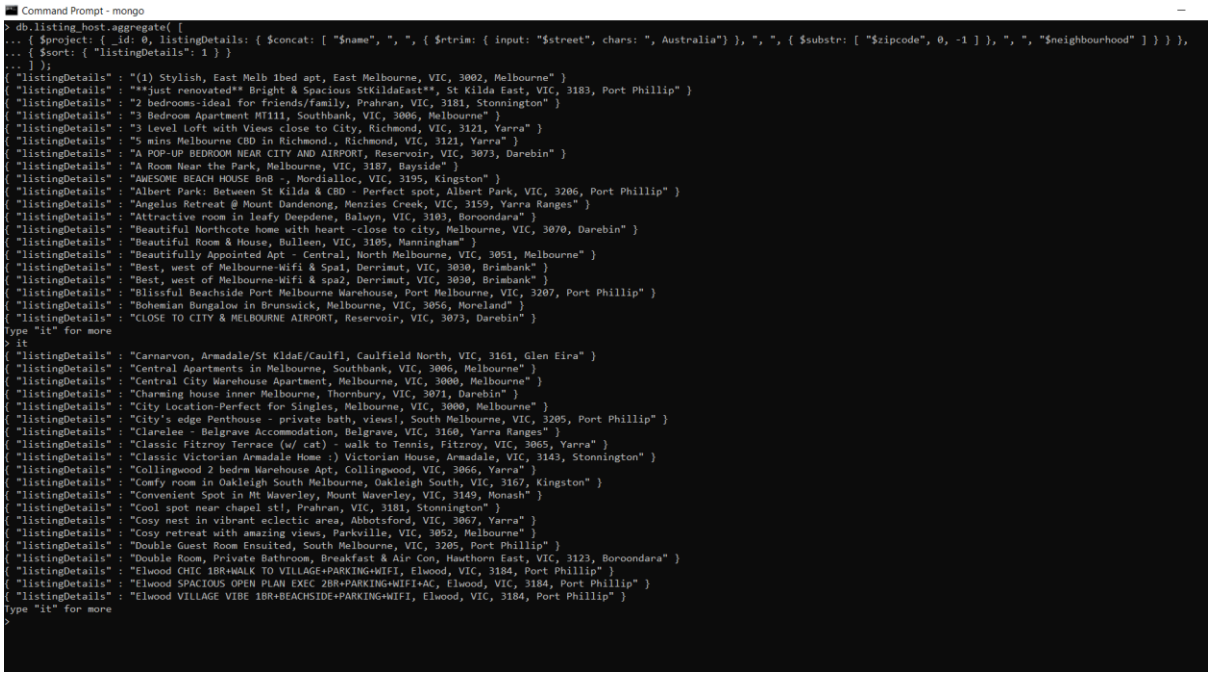
***Q11 - Embedding***

db.listing_host.aggregate( [

{ $project: { _id: 0, listingDetails: { $concat: [ "$name", ", ", { $rtrim: { input: "$street", chars: ", Australia"} }, ", ", { $substr: [ "$zipcode", 0, -1 ] }, ", ", "$neighbourhood" ] } } },

{ $sort: { "listingDetails": 1 } }

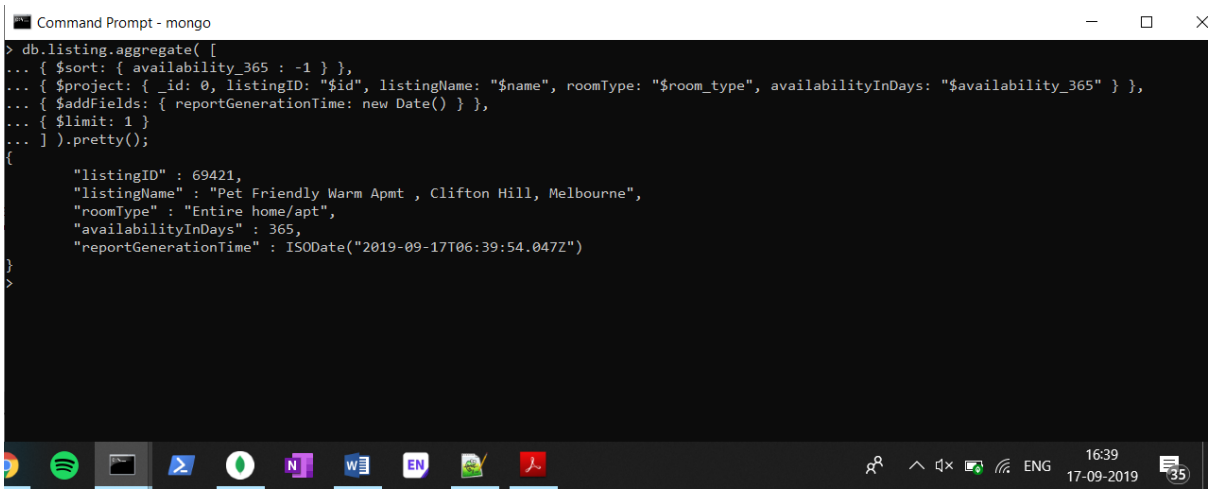
] );



```
Command Prompt - mongo
> db.listing_host.aggregate( [
... { $project: { _id: 0, listingDetails: { $concat: [ "$name", ", ", { $rtrim: { input: "$street", chars: ", Australia"} }, ", ", { $substr: [ "$zipcode", 0, -1 ] }, ", ", "$neighbourhood" ] } } },
... { $sort: { "listingDetails": 1 } }
... ] );
{ "listingDetails" : "(1) Stylish, East Melb 1bed apt, East Melbourne, VIC, 3002, Melbourne" }
{ "listingDetails" : "**just renovated** Bright & Spacious StKildaEast**, St Kilda East, VIC, 3183, Port Phillip" }
{ "listingDetails" : "2 bedrooms-ideal for friends/family, Prahran, VIC, 3181, Stonnington" }
{ "listingDetails" : "3 Bedroom Apartment MT111, Southbank, VIC, 3006, Melbourne" }
{ "listingDetails" : "3 Level Loft with Views close to City, Richmond, VIC, 3121, Yarra" }
{ "listingDetails" : "5 mins Melbourne CBD in Richmond., Richmond, VIC, 3121, Yarra" }
{ "listingDetails" : "A POP-UP BEDROOM NEAR CITY AND AIRPORT, Reservoir, VIC, 3073, Darebin" }
{ "listingDetails" : "A Room Near the Park, Melbourne, VIC, 3187, Bayside" }
{ "listingDetails" : "AWESOME BEACH HOUSE BnB -, Mordialloc, VIC, 3195, Kingston" }
{ "listingDetails" : "Albert Park: Between St Kilda & CBD - Perfect spot, Albert Park, VIC, 3206, Port Phillip" }
{ "listingDetails" : "Angelus Retreat @ Mount Dandenong, Menzies Creek, VIC, 3159, Yarra Ranges" }
{ "listingDetails" : "Attractive room in leafy Deepdene, Balwyn, VIC, 3103, Boroondara" }
{ "listingDetails" : "Beautiful Northcote home with heart -close to city, Melbourne, VIC, 3070, Darebin" }
{ "listingDetails" : "Beautiful Room & House, Bulleen, VIC, 3105, Manningham" }
{ "listingDetails" : "Beautifully Appointed Apt - Central, North Melbourne, VIC, 3051, Melbourne" }
{ "listingDetails" : "Best, west of Melbourne-Wifi & Spa1, Derrimut, VIC, 3030, Brimbank" }
{ "listingDetails" : "Best, west of Melbourne-Wifi & spa2, Derrimut, VIC, 3030, Brimbank" }
{ "listingDetails" : "Blissful Beachside Port Melbourne Warehouse, Port Melbourne, VIC, 3207, Port Phillip" }
{ "listingDetails" : "Bohemian Bungalow in Brunswick, Melbourne, VIC, 3056, Moreland" }
{ "listingDetails" : "CLOSE TO CITY & MELBOURNE AIRPORT, Reservoir, VIC, 3073, Darebin" }
Type "it" for more
> it
{ "listingDetails" : "Carnarvon, Armadale/St KldaE/Caulfl, Caulfield North, VIC, 3161, Glen Eira" }
{ "listingDetails" : "Central Apartments in Melbourne, Southbank, VIC, 3006, Melbourne" }
{ "listingDetails" : "Central City Warehouse Apartment, Melbourne, VIC, 3000, Melbourne" }
{ "listingDetails" : "Charming house inner Melbourne, Thornbury, VIC, 3071, Darebin" }
{ "listingDetails" : "City Location-Perfect for Singles, Melbourne, VIC, 3000, Melbourne" }
{ "listingDetails" : "City's edge Penthouse - private bath, views!, South Melbourne, VIC, 3205, Port Phillip" }
{ "listingDetails" : "Clarelee - Belgrave Accommodation, Belgrave, VIC, 3160, Yarra Ranges" }
{ "listingDetails" : "Classic Fitzroy Terrace (w/ cat) - walk to Tennis, Fitzroy, VIC, 3065, Yarra" }
{ "listingDetails" : "Classic Victorian Armadale Home :) Victorian House, Armadale, VIC, 3143, Stonnington" }
{ "listingDetails" : "Collingwood 2 bedrm Warehouse Apt, Collingwood, VIC, 3066, Yarra" }
{ "listingDetails" : "Comfy room in Oakleigh South Melbourne, Oakleigh South, VIC, 3167, Kingston" }
{ "listingDetails" : "Convenient Spot in Mt Waverley, Mount Waverley, VIC, 3149, Monash" }
{ "listingDetails" : "Cool spot near chapel st!, Prahran, VIC, 3181, Stonnington" }
{ "listingDetails" : "Cosy nest in vibrant eclectic area, Abbotsford, VIC, 3067, Yarra" }
{ "listingDetails" : "Cosy retreat with amazing views, Parkville, VIC, 3052, Melbourne" }
{ "listingDetails" : "Double Guest Room Ensuited, South Melbourne, VIC, 3205, Port Phillip" }
{ "listingDetails" : "Double Room, Private Bathroom, Breakfast & Air Con, Hawthorn East, VIC, 3123, Boroondara" }
{ "listingDetails" : "Elwood CHIC 1BR+WALK TO VILLAGE+PARKING+WIFI, Elwood, VIC, 3184, Port Phillip" }
{ "listingDetails" : "Elwood SPACIOUS OPEN PLAN EXEC 2BR+PARKING+WIFI+AC, Elwood, VIC, 3184, Port Phillip" }
{ "listingDetails" : "Elwood VILLAGE VIBE 1BR+BEACHSIDE+PARKING+WIFI, Elwood, VIC, 3184, Port Phillip" }
Type "it" for more
>
```

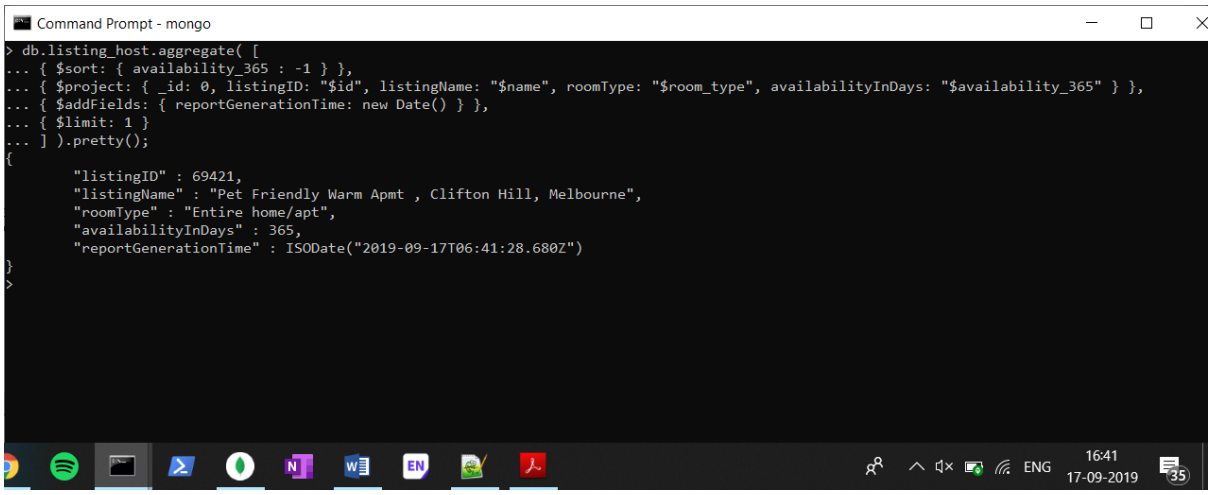
***Q12 - Referencing***

```
db.listing.aggregate( [
        { $sort: { availability_365 : -1 } },
        { $project: { _id: 0, listingID: "$id", listingName: "$name", roomType: "$room_type",
availabilityInDays: "$availability_365" } },
        { $addFields: { reportGenerationTime: new Date() } },
        { $limit: 1 }
] ).pretty();
```



***Q12 - Embedding***

```
db.listing_host.aggregate( [
        { $sort: { availability_365 : -1 } },
        { $project: { _id: 0, listingID: "$id", listingName: "$name", roomType: "$room_type",
availabilityInDays: "$availability_365" } },
        { $addFields: { reportGenerationTime: new Date() } },
        { $limit: 1 }
] ).pretty();
```



***Q13 - Referencing***

```
db.listing.aggregate( [
        { $match: { price: { $gt: 50 } } },
        { $group: { _id: "$neighbourhood", prices: { $push: "$price" } } },
        { $project: { _id: 0, neighbourhood: "$_id", prices: "$prices" } },
        { $sort: { neighbourhood: -1 } }
] );
```

Command Prompt - mongo

```
> db.listing.aggregate( [
... { $match: { price: { $gt: 50 } } },
... { $group: { _id: "$neighbourhood", prices: { $push: "$price" } } },
... { $project: { _id: 0, neighbourhood: "$_id", prices: "$prices" } },
... { $sort: { neighbourhood: -1 } }
... ] );
{ "neighbourhood" : "Yarra Ranges", "prices" : [ 81, 160, 200, 500, 249 ] }
{ "neighbourhood" : "Yarra", "prices" : [ 180, 110, 98, 138, 160, 120, 135, 89, 125, 120, 94, 140, 145, 249, 125, 71 ] }
{ "neighbourhood" : "Wyndham", "prices" : [ 71 ] }
{ "neighbourhood" : "Stonnington", "prices" : [ 73, 115, 98, 56 ] }
{ "neighbourhood" : "Port Phillip", "prices" : [ 189, 159, 85, 69, 95, 138, 71, 81, 120, 92, 130, 199, 125, 330 ] }
{ "neighbourhood" : "Moreland", "prices" : [ 89, 78, 89 ] }
{ "neighbourhood" : "Monash", "prices" : [ 98 ] }
{ "neighbourhood" : "Melton", "prices" : [ 72 ] }
{ "neighbourhood" : "Melbourne", "prices" : [ 69, 99, 249, 140, 135, 140, 245, 190, 155, 84, 225, 89, 701, 125, 221, 75, 110, 357, 74 ] }
{ "neighbourhood" : "Maribyrnong", "prices" : [ 65 ] }
{ "neighbourhood" : "Manningham", "prices" : [ 61 ] }
{ "neighbourhood" : "Kingston", "prices" : [ 71, 89 ] }
{ "neighbourhood" : "Hobsons Bay", "prices" : [ 159 ] }
{ "neighbourhood" : "Glen Eira", "prices" : [ 99, 57, 419 ] }
{ "neighbourhood" : "Frankston", "prices" : [ 59, 59 ] }
{ "neighbourhood" : "Darebin", "prices" : [ 140, 65, 130 ] }
{ "neighbourhood" : "Casey", "prices" : [ 99, 59 ] }
{ "neighbourhood" : "Boroondara", "prices" : [ 61, 75 ] }
{ "neighbourhood" : "Bayside", "prices" : [ 299 ] }
{ "neighbourhood" : "Banyule", "prices" : [ 98 ] }
>
```

***Q13 - Embedding***

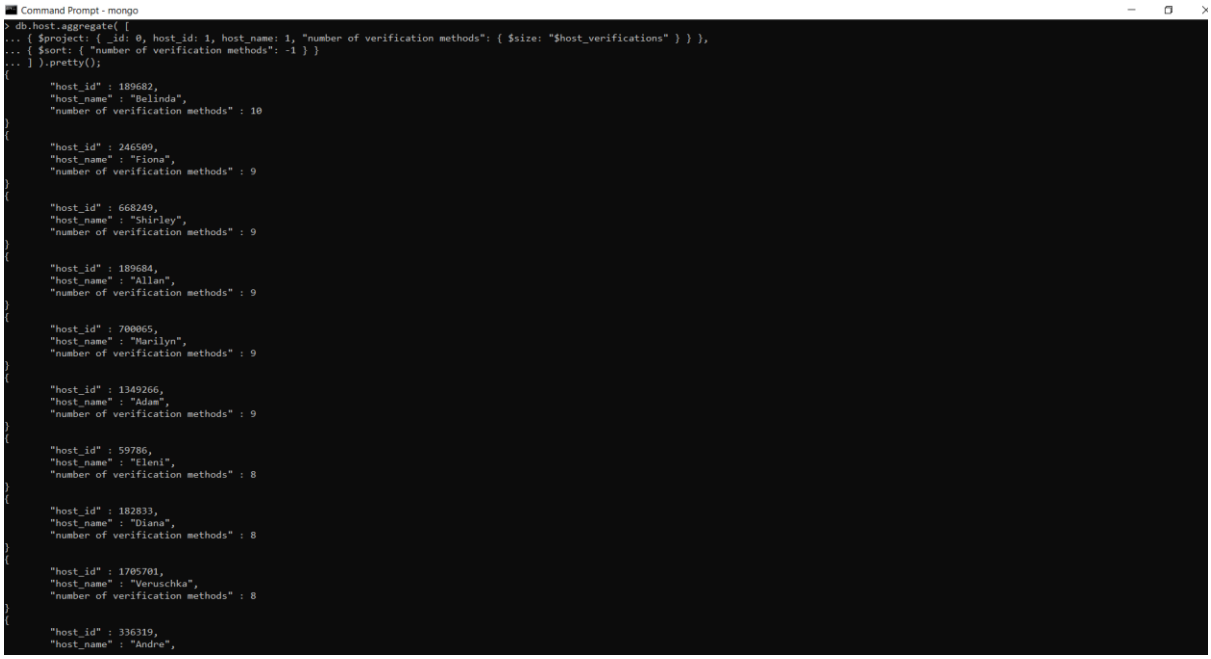
```
db.listing_host.aggregate( [
        { $match: { price: { $gt: 50 } } },
        { $group: { _id: "$neighbourhood", prices: {$push: "$price" } } },
        { $project: { _id: 0, neighbourhood: "$_id", prices: "$prices" } },
        { $sort: { neighbourhood: -1 } }
] );
```

Command Prompt - mongo

```
> db.listing_host.aggregate( [
... { $match: { price: { $gt: 50 } } },
... { $group: { _id: "$neighbourhood", prices: {$push: "$price" } } },
... { $project: { _id: 0, neighbourhood: "$_id", prices: "$prices" } },
... { $sort: { neighbourhood: -1 } }
... ] );
{ "neighbourhood" : "Yarra Ranges", "prices" : [ 81, 160, 200, 500, 249 ] }
{ "neighbourhood" : "Yarra", "prices" : [ 180, 110, 98, 138, 160, 120, 135, 89, 125, 120, 94, 140, 145, 249, 125, 71 ] }
{ "neighbourhood" : "Wyndham", "prices" : [ 71 ] }
{ "neighbourhood" : "Stonnington", "prices" : [ 73, 115, 98, 56 ] }
{ "neighbourhood" : "Port Phillip", "prices" : [ 189, 159, 85, 69, 95, 138, 71, 81, 120, 92, 130, 199, 125, 330 ] }
{ "neighbourhood" : "Moreland", "prices" : [ 89, 78, 89 ] }
{ "neighbourhood" : "Monash", "prices" : [ 98 ] }
{ "neighbourhood" : "Melton", "prices" : [ 72 ] }
{ "neighbourhood" : "Melbourne", "prices" : [ 69, 99, 249, 140, 135, 140, 245, 190, 155, 84, 225, 89, 701, 125, 221, 75, 110, 357, 74 ] }
{ "neighbourhood" : "Maribyrnong", "prices" : [ 65 ] }
{ "neighbourhood" : "Manningham", "prices" : [ 61 ] }
{ "neighbourhood" : "Kingston", "prices" : [ 71, 89 ] }
{ "neighbourhood" : "Hobsons Bay", "prices" : [ 159 ] }
{ "neighbourhood" : "Glen Eira", "prices" : [ 99, 57, 419 ] }
{ "neighbourhood" : "Frankston", "prices" : [ 59, 59 ] }
{ "neighbourhood" : "Darebin", "prices" : [ 140, 65, 130 ] }
{ "neighbourhood" : "Casey", "prices" : [ 99, 59 ] }
{ "neighbourhood" : "Boroondara", "prices" : [ 61, 75 ] }
{ "neighbourhood" : "Bayside", "prices" : [ 299 ] }
{ "neighbourhood" : "Banyule", "prices" : [ 98 ] }
>
```

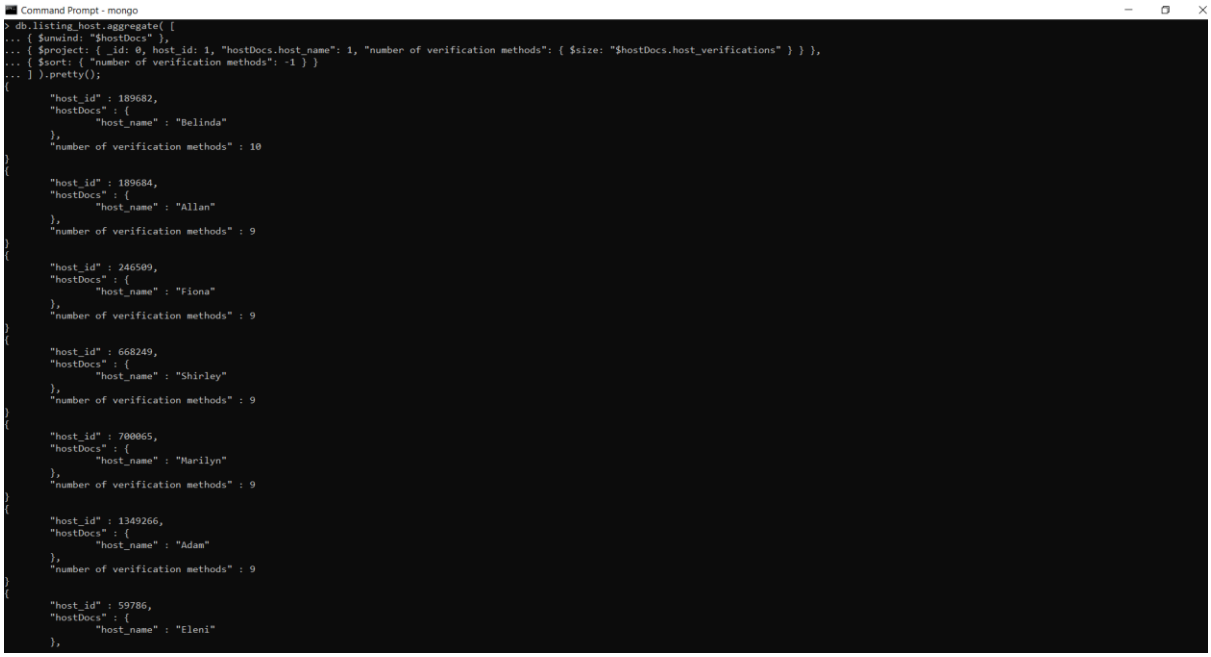
***Q14 - Referencing***

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



***Q14 - Embedding***

```
db.listing_host.aggregate( [
        { $unwind: "$hostDocs" },
        { $project: { _id: 0, host_id: 1, "hostDocs.host_name": 1, "number of verification methods": { $size: "$hostDocs.host_verifications" } } },
        { $sort: { "number of verification methods": -1 } }
] ).pretty();
```



### C.3.1.5. Additional Queries

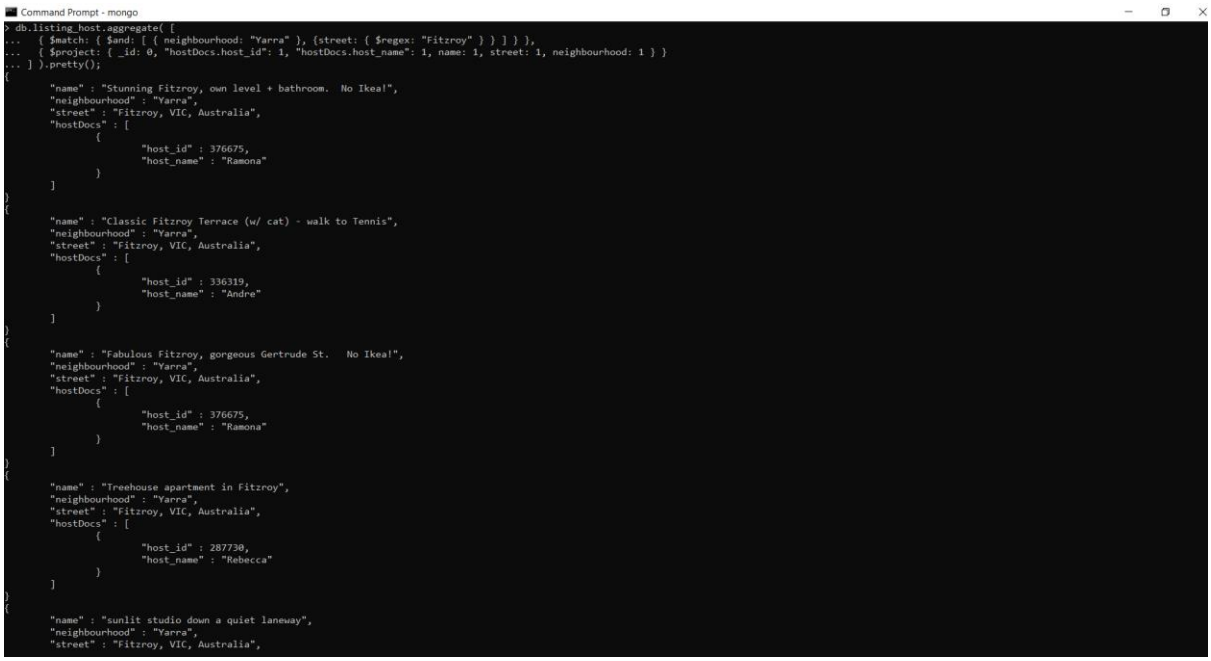
***Query 1***

Display the host's ID, host's name and the name of those listings that are located in Fitzroy and has Yarra as the neighbourhood

```
db.listing_host.aggregate( [
  { $match: { $and: [ { neighbourhood: "Yarra" }, {street: { $regex: "Fitzroy" } } ] } },
  { $project: { _id: 0, "hostDocs.host_id": 1, "hostDocs.host_name": 1, name: 1, street: 1, neighbourhood: 1 } }
] ).pretty();
```

---OR---

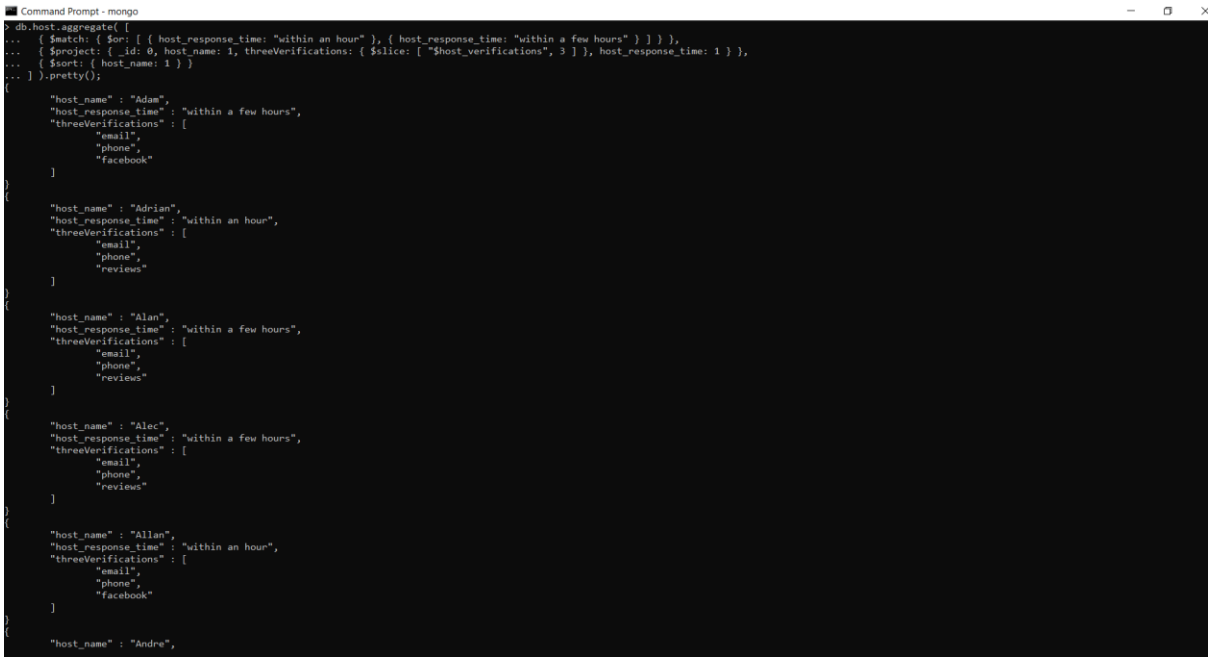
```
db.listing.aggregate( [
        { $lookup : { from : "host", localField : "host_id", foreignField : "host_id", as : "hostDocs" } },
        { $match: { $and: [ { neighbourhood: "Yarra" }, {street: { $regex: "Fitzroy" } } ] } },
        { $project: { _id: 0, "hostDocs.host_id": 1, "hostDocs.host_name": 1, name: 1, street: 1, neighbourhood: 1 } }
] ).pretty();
```



***Query 2***

Display the first three verifications for those hosts who respond within an hour or within a few hours and sort them by host name in ascending order.

```
db.host.aggregate( [
  { $match: { $or: [ { host_response_time: "within an hour" }, { host_response_time: "within a few hours" } ] } },
  { $project: { _id: 0, host_name: 1, threeVerifications: { $slice: [ "$host_verifications", 3 ] }, host_response_time: 1 } },
  { $sort: { host_name: 1 } }
] ).pretty();
```



***Query 3***

Find the top 5 listings along with their neighbourhoods with the most reviewed accomodations that were last reviewed after 10th January 2019.

```
db.listing.aggregate( [
  { $match: { last_review: { $gt: new Date("2019-01-10") } } },
  { $project: {_id: 0, listingID: "$id", listingName: "$name", neighbourhood: "$neighbourhood",
numberOfReviews: "$number_of_reviews", lastReviewDate: "$last_review" } },
  { $sort: { numberOfReviews: -1 } },
  { $limit : 5 }
] ).pretty();
```

Command Prompt - mongo

```
> db.listing.aggregate( [
...   { $match: { last_review: { $gt: new Date("2019-01-10") } } },
...   { $project: {_id: 0, listingID: "$id", listingName: "$name", neighbourhood: "$neighbourhood", numberOfReviews: "$number_of_reviews", lastReviewDate: "$last_review" } },
...   { $sort: { numberOfReviews: -1 } },
...   { $limit : 5 }
... ] ).pretty();
{
        "listingID" : 150729,
        "listingName" : "Fabulous Fitzroy, gorgeous Gertrude St.   No Ikea!",
        "neighbourhood" : "Yarra",
        "numberOfReviews" : 527,
        "lastReviewDate" : ISODate("2019-07-08T00:00:00Z")
}
{
        "listingID" : 72576,
        "listingName" : "Stunning Fitzroy, own level + bathroom.  No Ikea!",
        "neighbourhood" : "Yarra",
        "numberOfReviews" : 476,
        "lastReviewDate" : ISODate("2019-07-07T00:00:00Z")
}
{
        "listingID" : 268849,
        "listingName" : "Unique, Intimately Styled South Yarra Sanctuary",
        "neighbourhood" : "Stonnington",
        "numberOfReviews" : 416,
        "lastReviewDate" : ISODate("2019-07-07T00:00:00Z")
}
{
        "listingID" : 74548,
        "listingName" : "City Location-Perfect for Singles",
        "neighbourhood" : "Melbourne",
        "numberOfReviews" : 397,
        "lastReviewDate" : ISODate("2019-06-21T00:00:00Z")
}
{
        "listingID" : 244952,
        "listingName" : "Treehouse apartment in Fitzroy",
        "neighbourhood" : "Yarra",
        "numberOfReviews" : 269,
        "lastReviewDate" : ISODate("2019-05-05T00:00:00Z")
}
>
```

## C.3.2. Cassandra

### C.3.2.1. Answered Queries

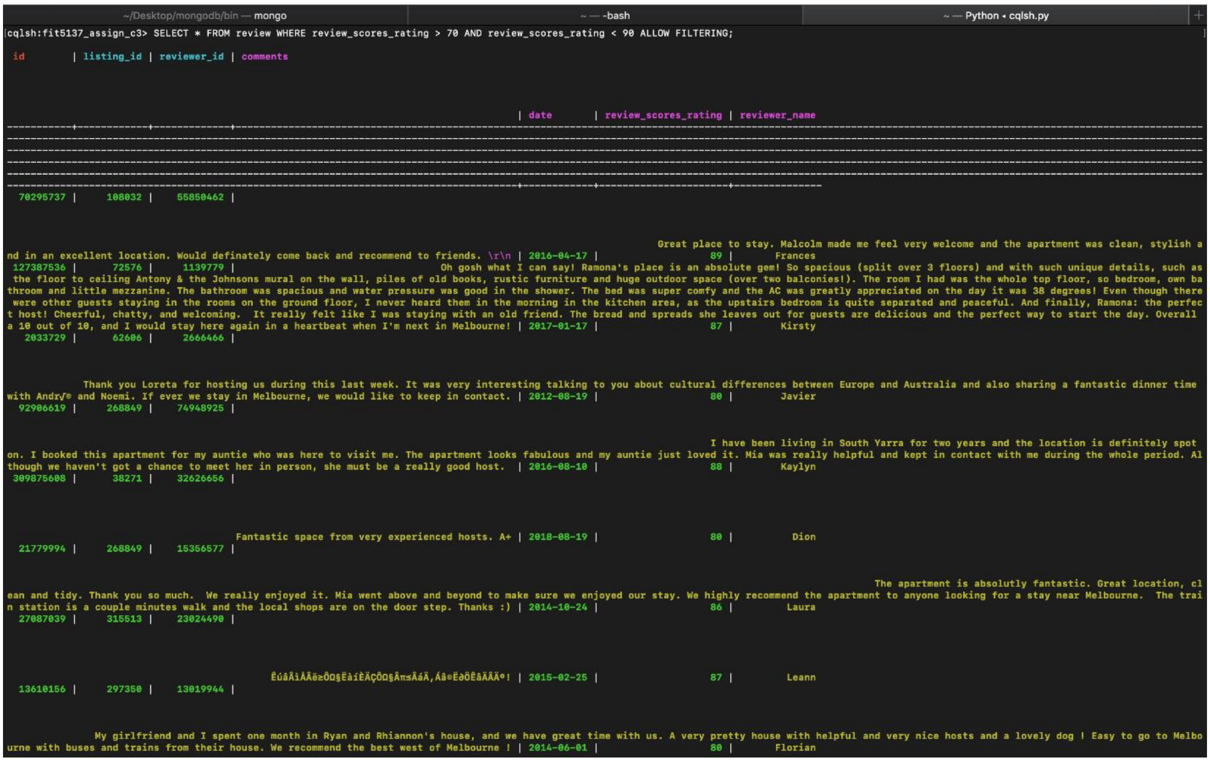
***Q15***

CREATE INDEX ON review (listing_ID);

SELECT listing_ID, id, max(date), reviewer_id, review_scores_rating, comments FROM review WHERE listing_ID = 10803;

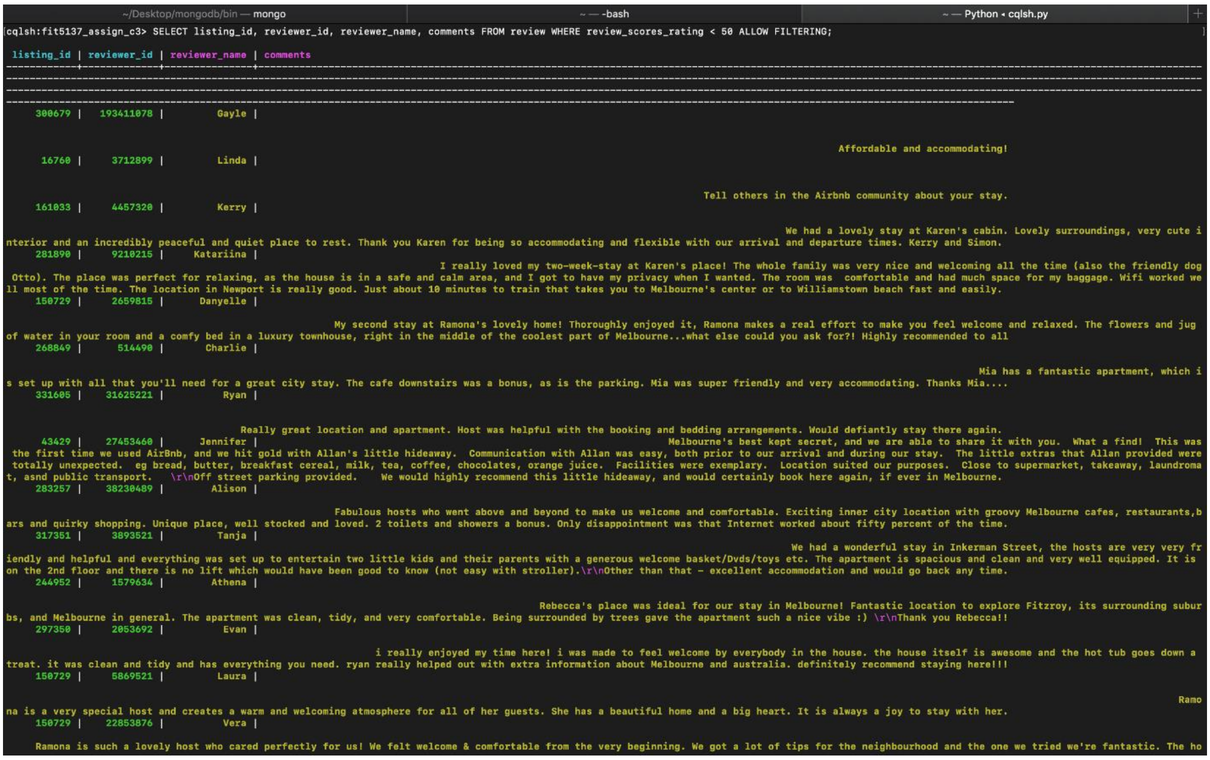
***Q16***

SELECT * FROM review WHERE review_scores_rating > 70 AND review_scores_rating < 90 ALLOW FILTERING;



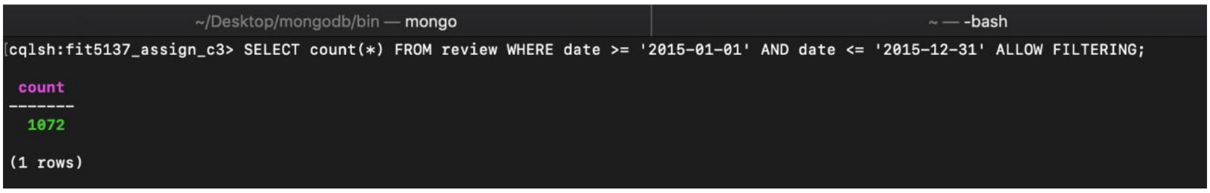
***Q17***

SELECT listing_id, reviewer_id, reviewer_name, comments FROM review WHERE review_scores_rating < 50 ALLOW FILTERING;




***Q18***

SELECT count(*) FROM review WHERE date >= '2015-01-01' AND date <= '2015-12-31' ALLOW FILTERING;



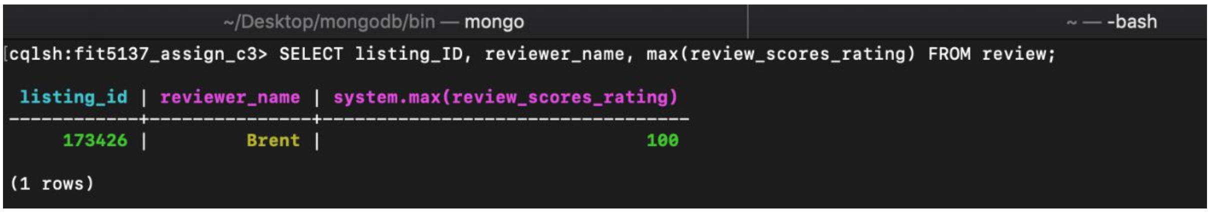
***Q19***

CREATE INDEX ON review (date);

SELECT id, reviewer_id, max(review_scores_rating), comments FROM review WHERE date = '2017-03-26';

***Q20***

SELECT listing_ID, reviewer_name, max(review_scores_rating) FROM review;



### C.3.2.2. Additional Queries

***Query 4***

Display the reviewers along with their ratings who have provided the reviews 219559474 and 46191285.

SELECT id, reviewer_id, reviewer_name, review_scores_rating FROM review WHERE id IN (219559474, 46191285) ALLOW FILTERING;

```
~/Desktop/mongodb/bin — mongo                                   ~ — -bash
cqlsh:fit5137_assign_c3> SELECT id, reviewer_id, reviewer_name, review_scores_rating FROM review WHERE id IN (219559474, 46191285) ALLOW FILTERING;

 id        | reviewer_id | reviewer_name | review_scores_rating
-----------+-------------+---------------+----------------------
  46191285 |    11135213 |       Michael |                   90
 219559474 |    79447678 |          Tate |                   99

(2 rows)
cqlsh:fit5137_assign_c3>
```

***Query 5***

What is the average of all the ratings provided for the listing?

CREATE INDEX on review (listing_ID);

SELECT avg(review_scores_rating) from review WHERE listing_ID = 51592;

```
~/Desktop/mongodb/bin — mongo                                   ~ — -bash
cqlsh:fit5137_assign_c3> CREATE INDEX on review (listing_ID);
cqlsh:fit5137_assign_c3> SELECT avg(review_scores_rating) from review WHERE listing_ID = 51592;

 system.avg(review_scores_rating)
---------------------------------
                               94

(1 rows)
```

# C.4. Database Comparison

## C.4.1. The Decision

We have selected Cassandra for MonashBnB, considering various aspects of both MongoDB and Cassandra. Cassandra is designed in a way that can help in handling high volumes of data and most importantly, provide high availability without a single point of failure. It is also highly scalable and consistent. The availability is due to Cassandra's peer-to-peer approach which allows easy access to data as the read and write operations can be performed on all the nodes in the cluster.

## C.4.2. Merging Process

Merging process for importing JSON data in Cassandra:

There are multiple processes for importing JSON data in Cassandra or exporting MongoDB data to Cassandra. Two of these methods are:

1) Importing JSON data by using the "JSON Support" provided by Cassandra. In this approach, we can either use the "INSERT JSON" or the "fromJson()" function.
2) Exporting the MongoDB data through JSON or CSV export of that data.

The first process is a good one but, in our thought, the greatest limitation here is efficiency and processing speed due to executing 'n' number of queries. Therefore, we have selected the second method for the merging process.

Steps to merge the data:

1. Start the mongo shell on the terminal.
2. Write the mongo export command ("mongoDB - Documentation," 2019):

   > 'mongoexport --collection <collection_name> --db <db_name> --out <file_name.csv>'

   where:

   - --db is used to specify the database name
   - --collection is used to specify the collection name
   - --out is used to specify the path of the output file (Batra, 2018)

3. Once the csv file is generated, create a KEYSPACE in Cassandra.
4. Create a table with all the fields similar to the fields present in csv file.
5. Copy the csv file data in the table, using the copy command:

   > COPY <keyspace_name.table_name (field1, field2, field3)> FROM '<path_name/file_name.csv>' WITH HEADER = TRUE;

6. Check whether all the data has been stored correctly.

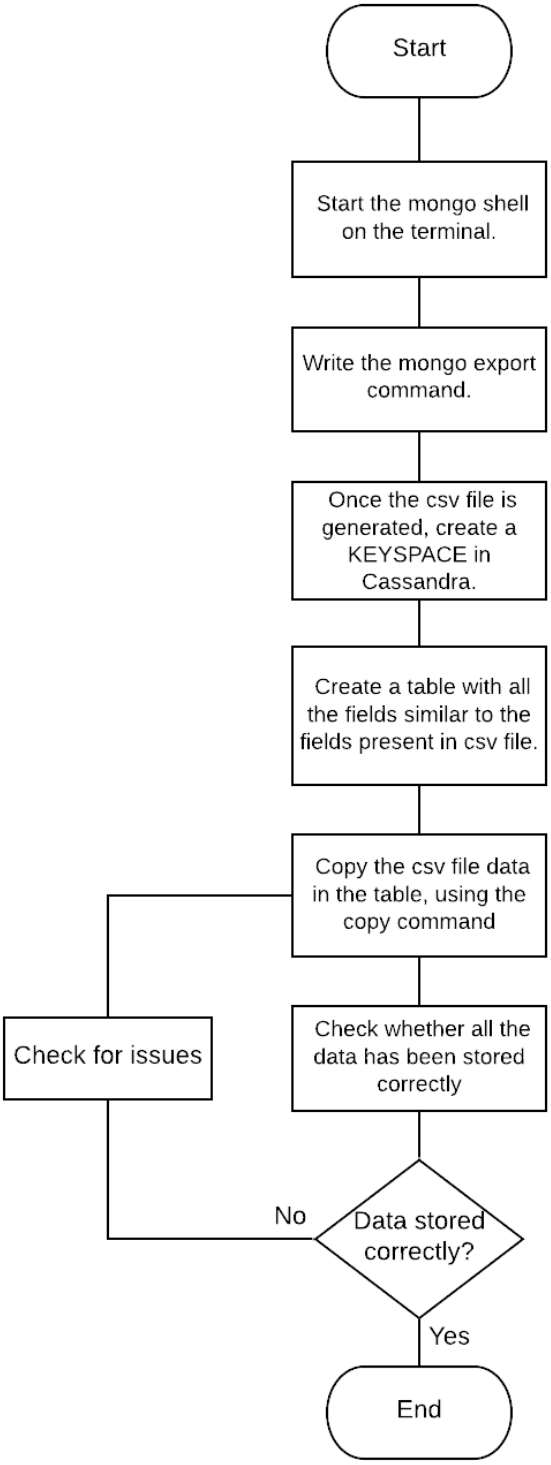



*Fig. 1 Flowchart for Merging Process*

## C.4.3. Strengths and Weaknesses

| | Strengths | Weaknesses |
|---|---|---|
| **MongoDB** | • Supports secondary indexes ("ScaleGrid," 2016)<br>• Is an expressive object model and is very 'object-oriented'<br>• Supports auto-sharding<br>• Good for unstructured data<br>• Easy to use | • Is a 'single-master' model with just 1 primary server<br>• Low write scalability (low write-throughput)<br>• No 'JOIN' support, '$lookup' function is used instead<br>• High data consumption |
| **Cassandra** | • Is a 'multiple-master' model with no single point-of-failure<br>• Is highly available with 100% uptime<br>• High write scalability with the acceptance of writes on all the servers in the cluster (high write-throughput) (Al-Saeedi, 2016)<br>• In comparison, high in performance<br>• High query language support (CQL is similar to SQL)<br>• Easy to use | • No 'JOIN' support, No 'OR' support<br>• No ad-hoc query support<br>• No subquery support<br>• No 'regex' support<br>• Cannot sort by a partition key column<br>• A query needs to be run across all the declared partitions<br>• Data modelling is tough |

## C.4.4. Comparison Report

MonashBnB is a residential service, offering the students and staff of Monash University a short time lodging experience. This field of service usually faces high amount of traffic in terms of users. Various new users and their data is going to be frequently added and removed in the MonashBnB's database. The users here are the hosts, with their property listings, and the customers with their hunt of properties where they can find a stay. In such a case, continuous updating is required in the database. High traffic of users accessing a system and its database may lead to problems like congestion.

Acting towards the suggestion, MonashBnB's database system was built and handled using MongoDB and Cassandra technologies. Both the technologies were involved in controlling the entire data of the company. The data was mainly divided into 3 parts, namely, hosts, listings and reviews. The hosts and listings parts were managed in MongoDB, whereas, the reviews were entirely worked on Cassandra. Specifically, managing the data included a new creation of the MonashBnB database and also updating the same. Further, few queries were resolved on the existing database provided by the company, and some new were designed as well.

As mentioned, we have decided to cruise further with Cassandra as the technology for MonashBnB's database system. Working on both the technologies helped in identifying the various pros and cons of MongoDB and Cassandra. These steered us in choosing the appropriate technology for the given case study. Cassandra, as a technology is more robust when compared with MongoDB. Cassandra is known to handle huge amounts of data efficiently. Implied, the high number of users will be handled by Cassandra effectively. Also, Cassandra supports a peer-to-peer approach which makes the available data easily accessible, allowing the respective users perform the read and write operations effectively.

The data managed on Cassandra is spread across all the assigned servers with equal operation rights. This makes the users experience a seamless performance while operating the MonashBnB system. The access to multiple servers avoids a total breakdown of the system, making it highly available for the end users. Additionally, CQL (Cassandra Query Language) has a high query language support. CQL syntax is quite similar to SQL (Structured Query Language), which is one of the most popular and commonly used query languages (Conrad, Misenar, & Feldman, 2016). This makes Cassandra a much easy-to-learn and easy-to-use query language. This feature can be used by the various database analysts and database administrators which might have a prior experience of working with SQL.

The data provided and generated by MonashBnB is leaned towards being more structured and has a clear understanding of the data with respect to their fields. There are a negligible number of null attributes, but high anomalies in terms of the data types of the information. The data storage can be classified into flexible and wide-column. Considering these factors, the decision of choosing Cassandra over MongoDB is justified even more.

To summarise, the information stored in the MonashBnB database system needs to be available 24 x 7 for the end users. The data should also be highly scalable, easily accessible and readily operable. The high number of users and huge amount of data should not affect the efficiency and effectiveness of the database system. To maintain this performance throughout, with least interruptions and difficulties for the audience, Cassandra is a clear winner of choice in comparison with MongoDB for this particular case study.

# References


Al-Saeedi, B. (2016). Factors Influencing NoSQL Adoption. Retrieved from http://alronz.github.io/Factors-Influencing-NoSQL-Adoption/site/Cassandra/Results/Strengths%20and%20Weaknesses/

Batra, Y. (2018). Java Code Geeks. *MongoDB Import and Export JSON Data Example.* Retrieved from https://examples.javacodegeeks.com/software-development/mongodb/mongodb-import-export-json-data-example/

Conrad, E., Misenar, S., & Feldman, J. (2016). Chapter 9 - Domain 8: Software Development Security (Understanding, Applying, and Enforcing Software Security). In E. Conrad, S. Misenar, & J. Feldman (Eds.), *CISSP Study Guide (Third Edition)* (pp. 429-477). Boston: Syngress.

mongoDB - Documentation. (2019). *mongoexport — MongoDB Manual.* Retrieved from https://docs.mongodb.com/manual/reference/program/mongoexport/

ScaleGrid. (2016). *Cassandra vs. MongoDB.* Retrieved from https://scalegrid.io/blog/cassandra-vs-mongodb/