Assignment 1: MongoDB and Cassandra

Group ID: 16
Student ID: An Ju (30391113)
Jia Shi (28515056)
Assignment 1 A1 Report

Contents

Assignment 1: MongoDB and Cassandra 1
Cover Sheet 3
Contribution Declaration Form 4
Signed form 4
List of contribution 4
Report 5
C1 5
C 1.1 5
C1.2 6
C1.3 7
C1.4 7
C1.5 10
C1.7 13
C1.8 14
C2 14
C2.1 15
C2.2 15
C2.3 18
C2.4 19
C2.5 21
C2.6 21
C3 Report 22
Strength and weakness 24
Data Architecture Diagram 25
Reference 26

Cover Sheet

MONASH University
Information Technology

GROUP ASSIGNMENT COVER SHEET

Student ID Number	Surname	Given Names
30391113	*Ju*	**An**
28515056	*Shi*	**Jia**

* Please include the names of all other group members.

Unit name and code	FIT3176 Advanced Database Design	
Title of assignment	Assignment 1: MongoDB and Cassandra	
Lecturer/tutor	Farah Kabir	
Tutorial day and time	06_Online Wed 2 pm - 4 pm	**Campus:** Clayton
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: Wednesday, 21 September 2022, 11:55 PM		**Date submitted:** 17/10/2022

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 The material used can be from any source (staff, students or the internet, published and unpublished works).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 .....*An Ju*.....*Jia Shi*..... *Date*.....*23/09/2022*.....

* delete (iii) if not applicable

Signature ___An Ju_____Date:_____23/09/2022_____ Signature _____Date:_____

Signature ___Jia Shi_____Date:____ 23/09/2022_____ 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

Signed form

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

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

1 NAME AND CONTRIBUTION DETAILS

Student ID	Student Name	Contribution Percentage
30391113	An Ju	90%
28515056	Jia Shi	10%

2 DECLARATION

We declare that:

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

3 SIGNATURE

Signatures An Ju

Jia Shi

Date Day Month Year 23 / 09 / 2022

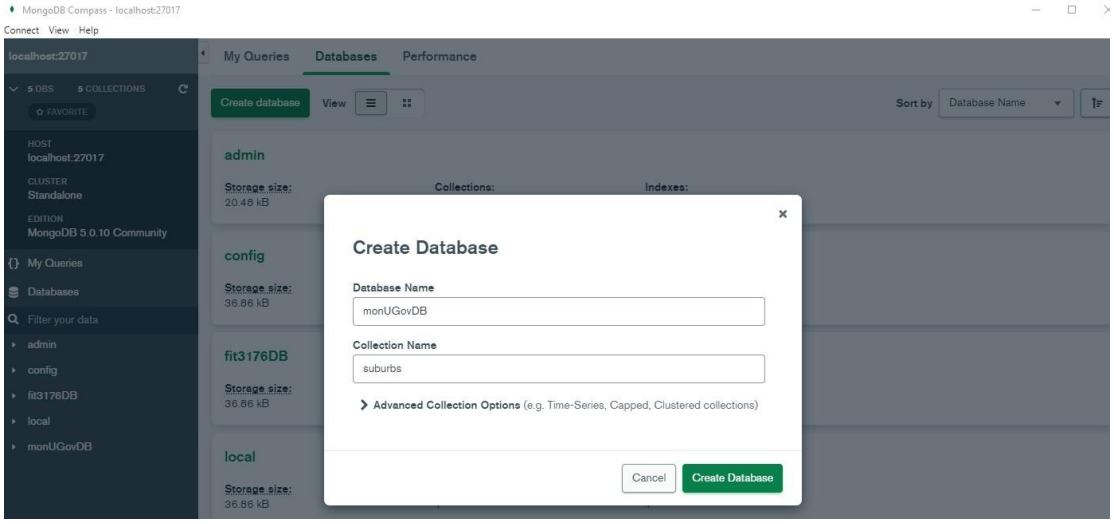
List of contribution

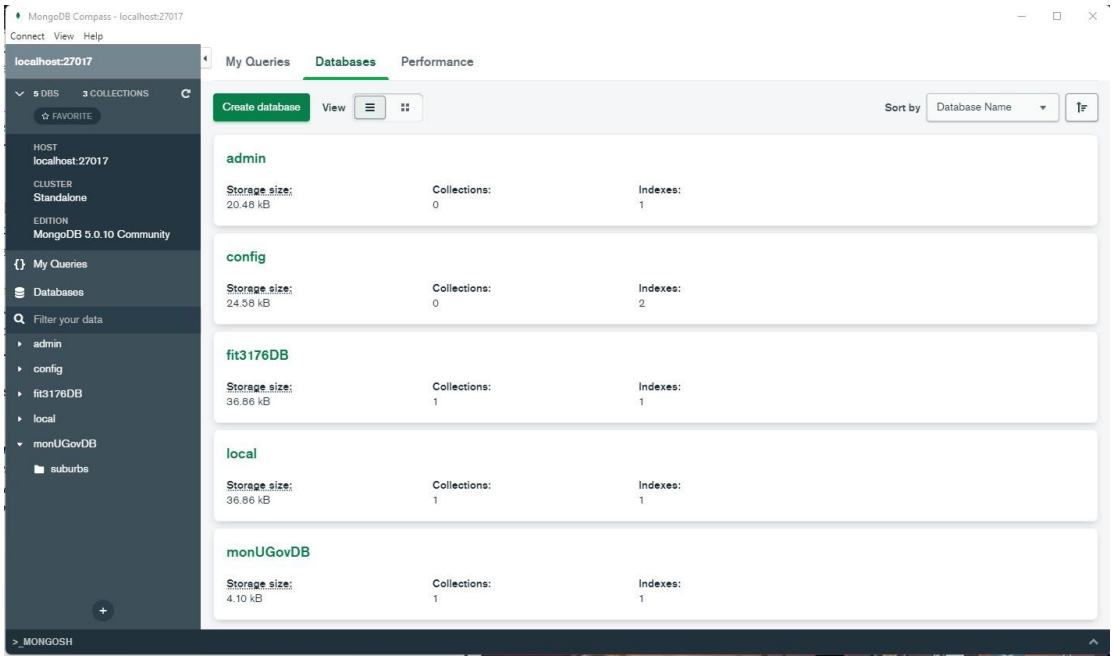
I. An Ju: Task C.1
Task C.2.
II. Jia Shi: Task C.3

Report

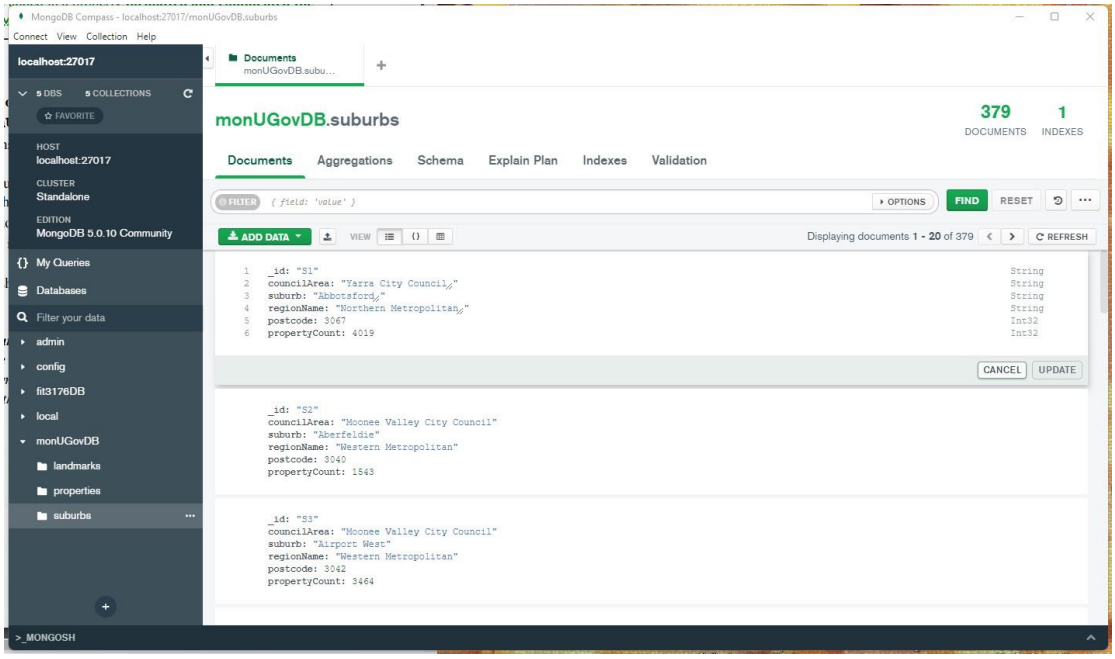
C1

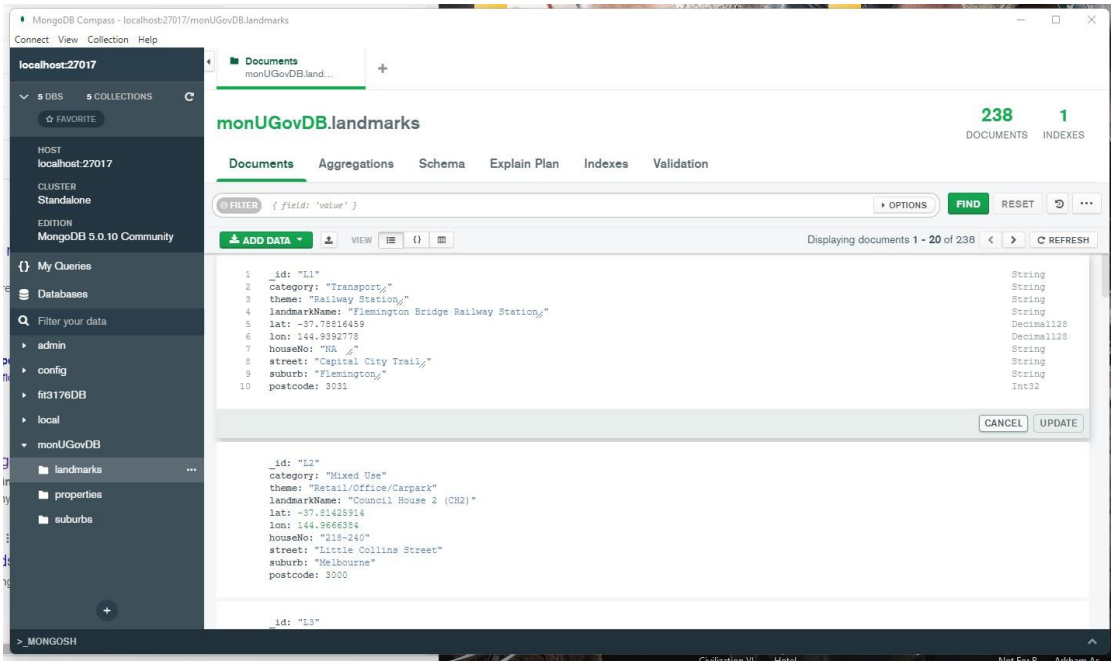
C 1.1

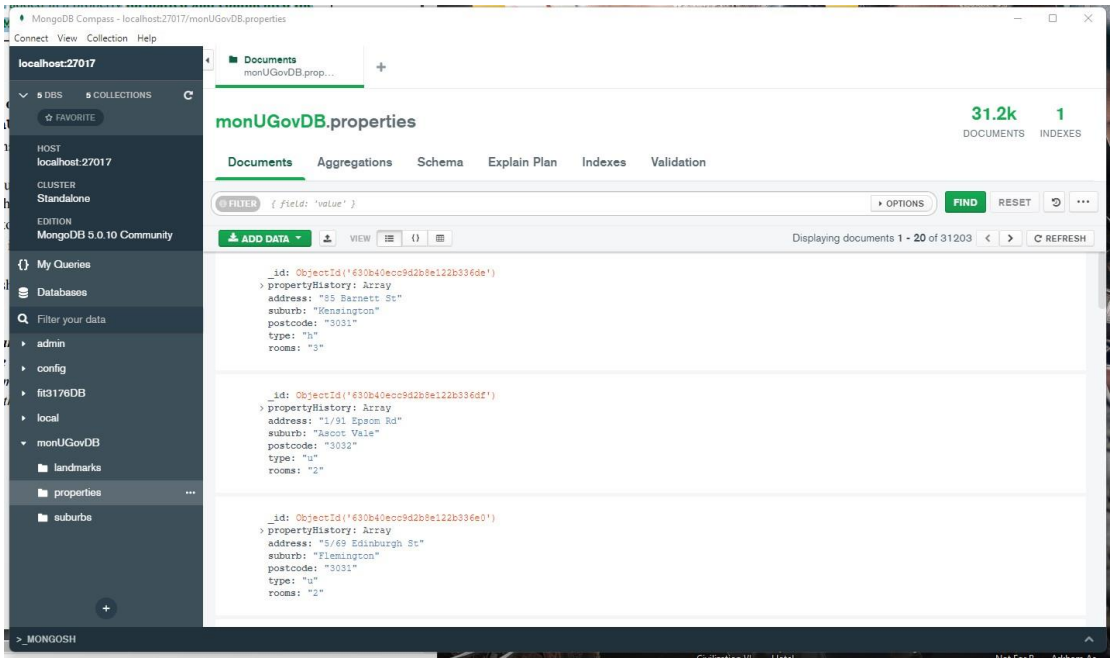




C1.2


MongoDB Compass - localhost:27017/monUGovDB.suburbs
Connect View Collection Help
localhost:27017
5 DBS 5 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 5.0.10 Community
My Queries
Databases
Filter your data
admin
config
fit3176DB
local
monUGovDB
landmarks
properties
suburbs
Documents
monUGovDB.subu...
monUGovDB.suburbs
379 DOCUMENTS
1 INDEXES
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 20 of 379
REFRESH
_id: "S1"
councilArea: "Yarra City Council"
suburb: "Abbotsford"
regionName: "Northern Metropolitan"
postcode: 3067
propertyCount: 4019
String
String
String
String
Int32
Int32
CANCEL
UPDATE
_id: "S2"
councilArea: "Moonee Valley City Council"
suburb: "Aberfeldie"
regionName: "Western Metropolitan"
postcode: 3040
propertyCount: 1543
_id: "S3"
councilArea: "Moonee Valley City Council"
suburb: "Airport West"
regionName: "Western Metropolitan"
postcode: 3042
propertyCount: 3464
>_MONGOSH


MongoDB Compass - localhost:27017/monUGovDB.landmarks
Connect View Collection Help
localhost:27017
5 DBS 5 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 5.0.10 Community
My Queries
Databases
Filter your data
admin
config
fit3176DB
local
monUGovDB
landmarks
properties
suburbs
Documents
monUGovDB.land...
monUGovDB.landmarks
238 DOCUMENTS
1 INDEXES
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 20 of 238
REFRESH
_id: "L1"
category: "Transport"
theme: "Railway Station"
landmarkName: "Flemington Bridge Railway Station"
lat: -37.78816459
lon: 144.9392778
houseNo: "NA"
street: "Capital City Trail"
suburb: "Flemington"
postcode: 3031
String
String
String
String
Decimal128
Decimal128
String
String
String
Int32
CANCEL
UPDATE
_id: "L2"
category: "Mixed Use"
theme: "Retail/Office/Carpark"
landmarkName: "Council House 2 (CH2)"
lat: -37.81425914
lon: 144.9666384
houseNo: "218-240"
street: "Little Collins Street"
suburb: "Melbourne"
postcode: 3000
_id: "L3"
>_MONGOSH



C1.3

use monUGovDB

// C1.3
db.landmarks.createIndex({ theme: "text" })
db.properties.createIndex({"address":1, "suburb":1}, {name:"addressWhole"})

C1.4

*//*C1.4.i

db.landmarks.find({ $text: { $search: "\"School\"" } }, {"_id":0, "landmarkName":1, "theme":1 })

Output C1.4.i

8 rows

	landmarkName	theme
1	North Melbourne Primary School	Primary Schools
2	Kensington Primary School	Primary Schools
3	University High School	Secondary Schools
4	Melbourne Grammar School	Secondary Schools
5	Carlton Primary School	Primary Schools
6	Carlton Gardens Primary School	Primary Schools
7	Melbourne Girls Grammar School	School - Primary and Secondary Education
8	Wesley College	School - Primary and Secondary Education

//C1.4.ii

db.suburbs.find({},{"_id": 0, "suburb": 1, "propertyCount": 1}).sort({ propertyCount: -1})

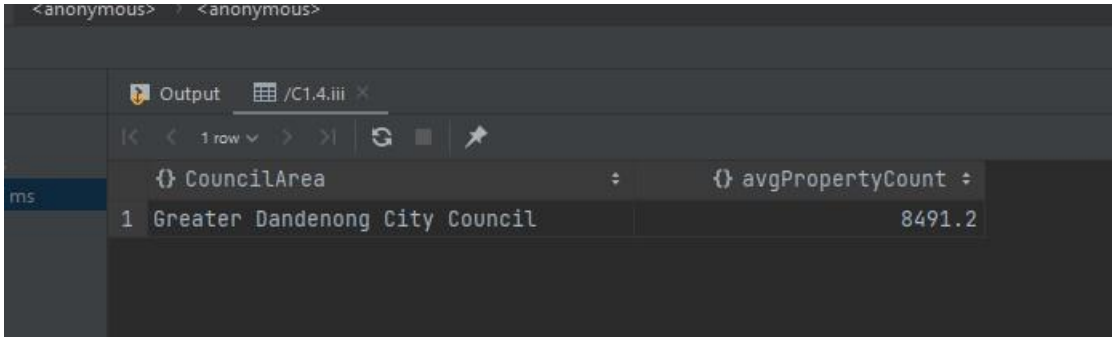
Output C1.4.ii

1-20 of 21+

	propertyCount	suburb
1	21650	Reservoir
2	17496	Melbourne
3	17384	Pakenham
4	17093	Berwick
5	17055	Frankston
6	16166	Werribee
7	15542	Point Cook
8	15510	Craigieburn
9	15321	Glen Waverley
10	14949	Richmond
11	14887	South Yarra
12	14577	Preston
13	14092	Sunbury
14	14042	St Albans
15	13830	Hoppers Crossing
16	13366	Mount Waverley
17	13240	St Kilda
18	11925	Croydon
19	11925	croydon
20	11918	Brunswick

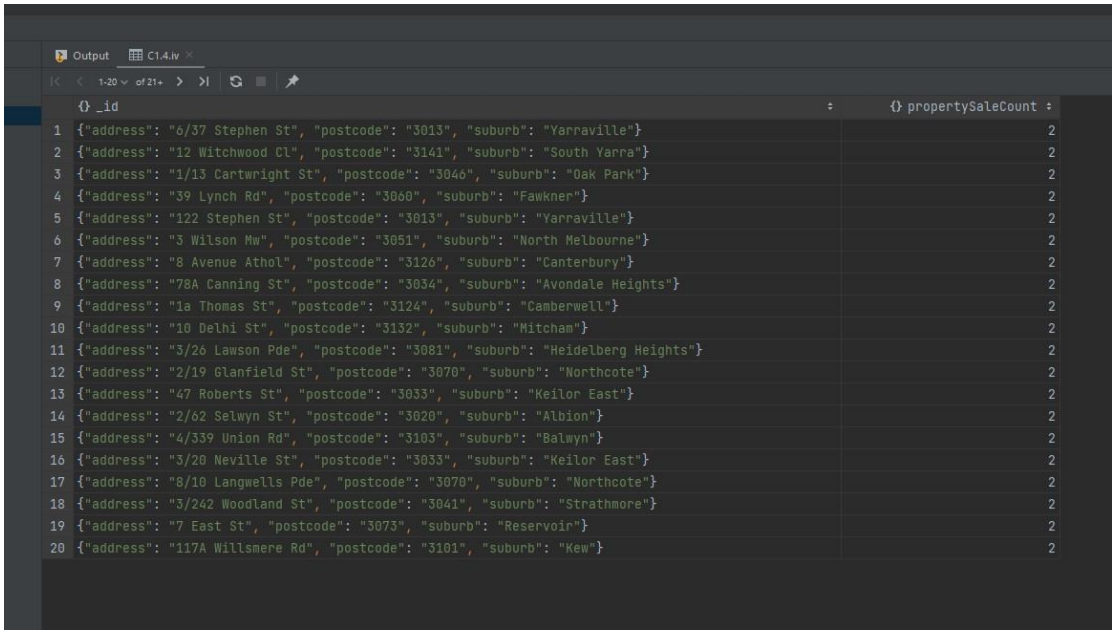
///C1.4.iii

db.suburbs.aggregate([
{ $group: { _id: "$councilArea",
avgPropertyCount: {$avg: "$propertyCount" } } },{$project: {_id:0, CouncilArea: "$_id",avgPropertyCount:{$round:["$avgPropertyCount", 1]}}}, {$sort:{avgPropertyCount:-1}}, {$skip:1}, {$limit:1}
])



//C1.4.iv

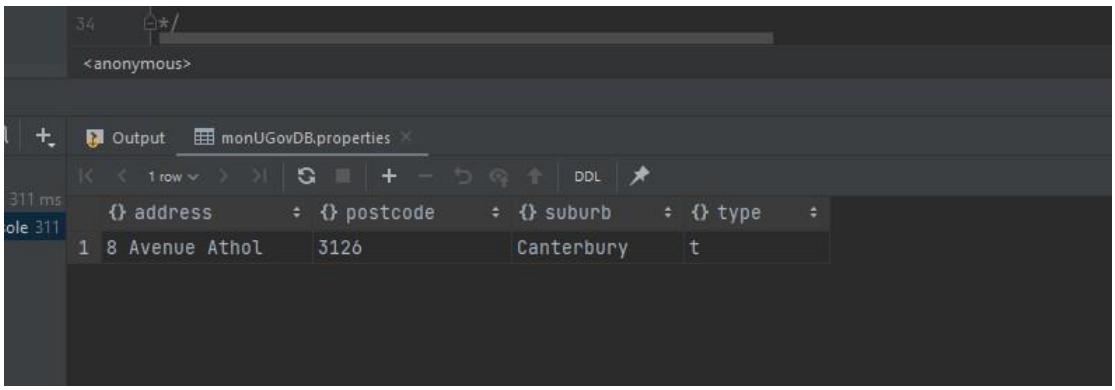
db.properties.aggregate([{$group:{_id: {address:"$address",postcode: "$postcode", suburb:"$suburb"}, propertySaleCount:{$sum:1}}}, {$project:{propertySaleCount:"$propertySaleCount"}},{$sort:{propertySaleCount:-1}}]).pretty()



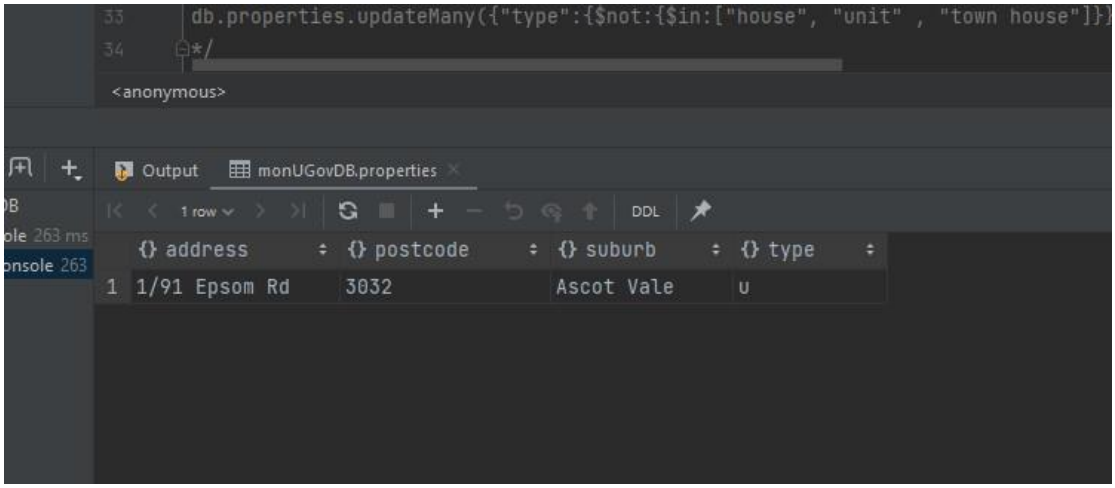
C1.5

//output before update

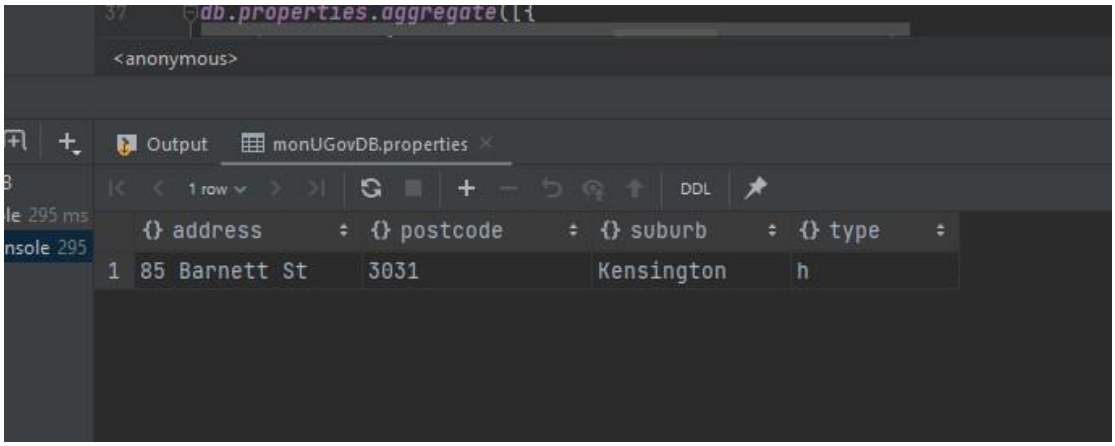
db.properties.find({"type":"t"},{_id:0,address:1,postcode:1,suburb:1,type:1}).limit(1)



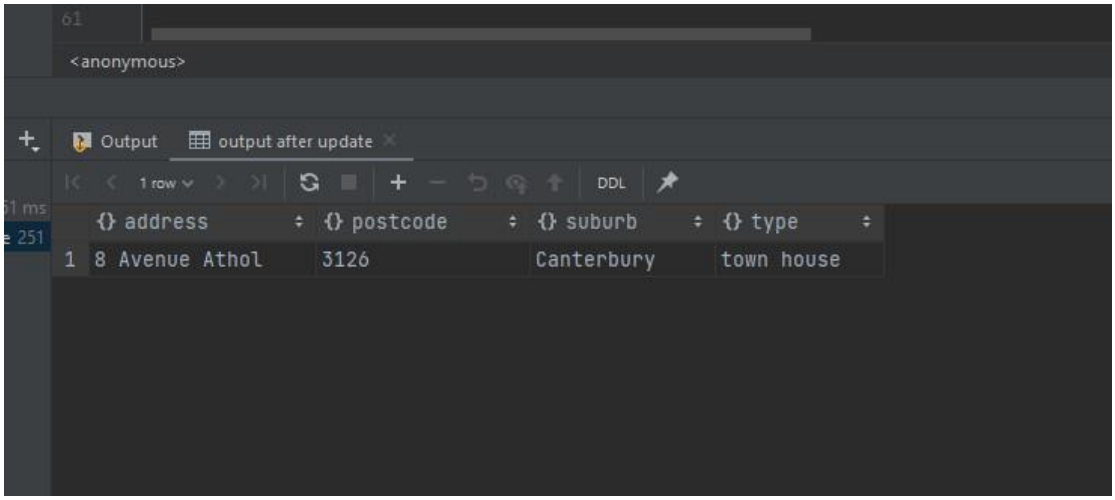
db.properties.find({"type":"u"},{_id:0,address:1,postcode:1,suburb:1,type:1}).limit(1)



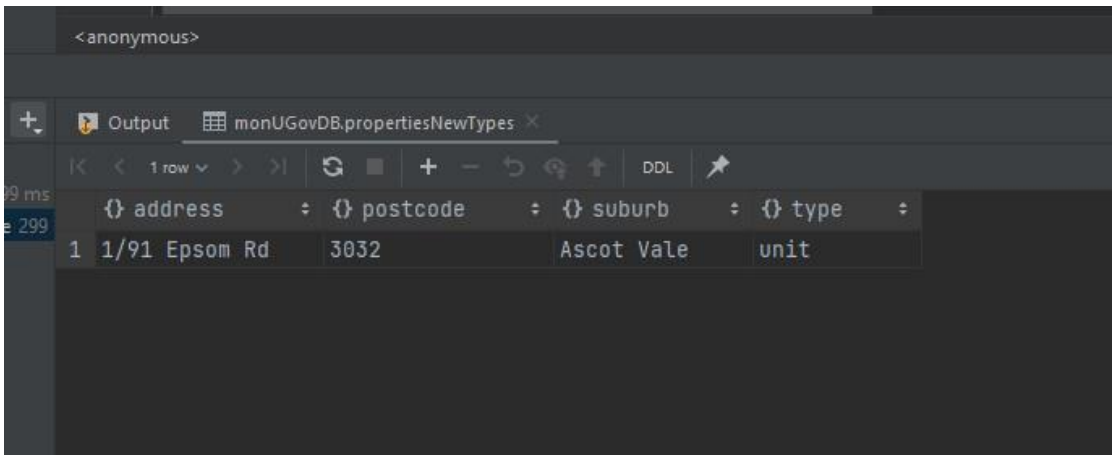
db.properties.find({"type":"h"},{_id:0,address:1,postcode:1,suburb:1,type:1}).limit(1)



```
/*
//update
db.properties.updateMany({"type":"house"},{$set:{"type": "h"}})
db.properties.updateMany({"type":"unit"},{$set:{"type": "u"}})
db.properties.updateMany({"type":"town house"},{$set:{"type": "t"}})
db.properties.updateMany({"type":{$not:{$in:["house", "unit" , "town house"]}}},{$set:{"type": "other"}})
*/

//update use nested if conditions
db.properties.aggregate([{
    $project: { _id:0, address:"$address", postcode:"$postcode", suburb:"$suburb",
        "type": {
            "$cond": [{"$eq": ["$type", "t"]}, "town house", {
                "$cond": [{"$eq": ["$type", "h"]}, "house", {
                    "$cond": [{"$eq": ["$type", "u"]},
                        "unit", "other"]
                }]
            }]
        }
    }
}, {$out:{db:"monUGovDB",coll:"propertiesNewTypes"}}])

//output after update
db.propertiesNewTypes.find({"type":"town house"},{_id:0,address:1,postcode:1,suburb:1,type:1}).limit(1)
```

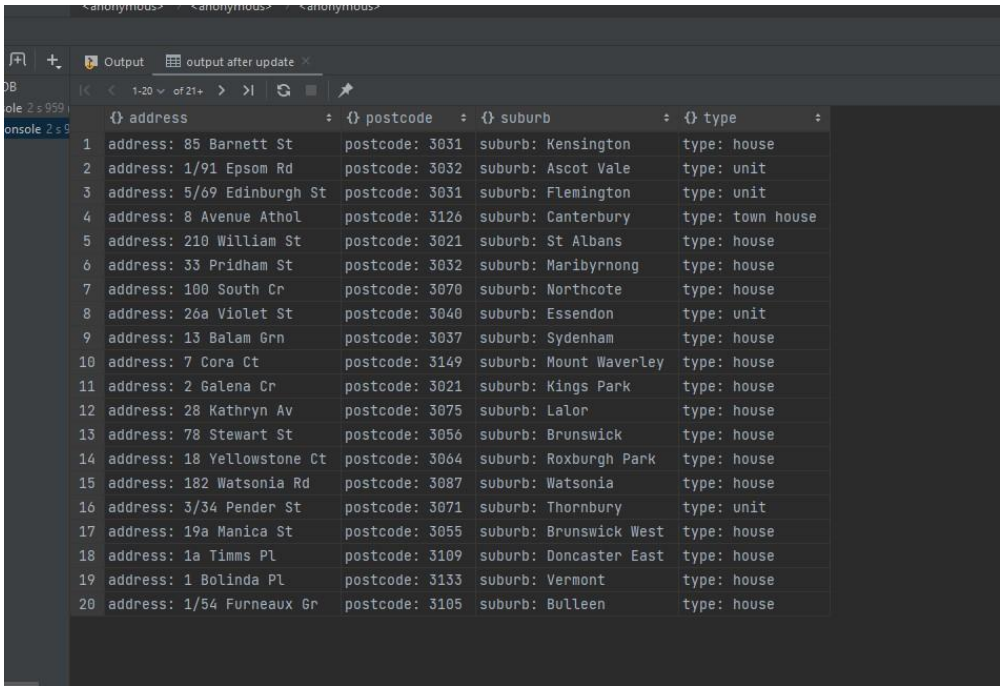



```
db.propertiesNewTypes.find({"type":"unit"},{_id:0,address:1,postcode:1,suburb:1,type:1}).limit(1)
```



	address	postcode	suburb	type
1	1/91 Epsom Rd	3032	Ascot Vale	unit

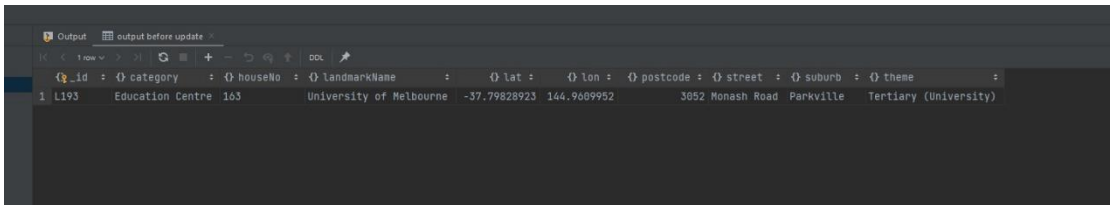
db.propertiesNewTypes.find({"type":"house"},{_id:0,address:1,postcode:1,suburb:1,type:1 }).limit(1)



	address	postcode	suburb	type
1	address: 85 Barnett St	postcode: 3031	suburb: Kensington	type: house
2	address: 1/91 Epsom Rd	postcode: 3032	suburb: Ascot Vale	type: unit
3	address: 5/69 Edinburgh St	postcode: 3031	suburb: Flemington	type: unit
4	address: 8 Avenue Athol	postcode: 3126	suburb: Canterbury	type: town house
5	address: 210 William St	postcode: 3021	suburb: St Albans	type: house
6	address: 33 Pridham St	postcode: 3032	suburb: Maribyrnong	type: house
7	address: 100 South Cr	postcode: 3070	suburb: Northcote	type: house
8	address: 26a Violet St	postcode: 3040	suburb: Essendon	type: unit
9	address: 13 Balam Grn	postcode: 3037	suburb: Sydenham	type: house
10	address: 7 Cora Ct	postcode: 3149	suburb: Mount Waverley	type: house
11	address: 2 Galena Cr	postcode: 3021	suburb: Kings Park	type: house
12	address: 28 Kathryn Av	postcode: 3075	suburb: Lalor	type: house
13	address: 78 Stewart St	postcode: 3056	suburb: Brunswick	type: house
14	address: 18 Yellowstone Ct	postcode: 3064	suburb: Roxburgh Park	type: house
15	address: 182 Watsonia Rd	postcode: 3087	suburb: Watsonia	type: house
16	address: 3/34 Pender St	postcode: 3071	suburb: Thornbury	type: unit
17	address: 19a Manica St	postcode: 3055	suburb: Brunswick West	type: house
18	address: 1a Timms Pl	postcode: 3109	suburb: Doncaster East	type: house
19	address: 1 Bolinda Pl	postcode: 3133	suburb: Vermont	type: house
20	address: 1/54 Furneaux Gr	postcode: 3105	suburb: Bulleen	type: house

//C1.6

//output before update
db.landmarks.find({"street":"Monash Road"})



	_id	category	houseNo	landmarkName	lat	lon	postcode	street	suburb	theme
1	L193	Education Centre	163	University of Melbourne	-37.79828923	144.9609952	3052	Monash Road	Parkville	Tertiary (University)

//update and output after update
db.landmarks.updateMany({"street":"Monash Road"}, {$set:{"homeGround": true, "team": 16}})

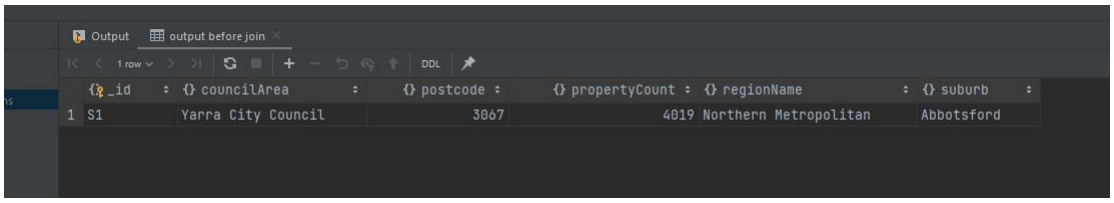
//output after update
db.landmarks.find({"street":"Monash Road"})

	_id	category	homeGround	houseNo	landmarkName	lat	lon	postcode	street	suburb	team	theme
1	L193	Education Centre	true	163	University of Melbourne	-37.79828923	144.9609952	3052	Monash Road	Parkville	16	Tertiary (University)

C1.7

//C1.7 i

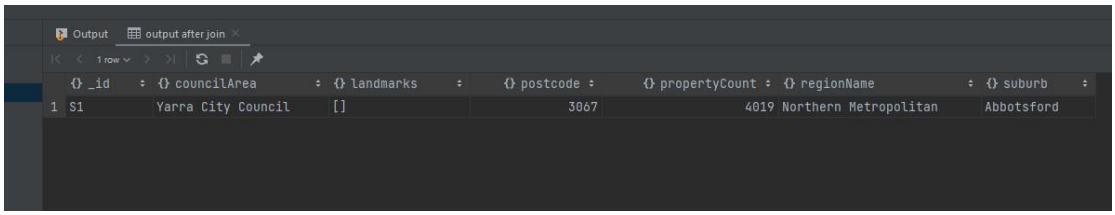
//output before join
db.suburbs.find()



	_id	councilArea	postcode	propertyCount	regionName	suburb
1	S1	Yarra City Council	3067	4019	Northern Metropolitan	Abbotsford

//join and saved as a new collection
db.landmarks.updateMany({}, {$rename:{"suburb":"landmark_suburb"}})
db.landmarks.find()
db.suburbs.aggregate([{$lookup:{
from:"landmarks",
localField:"suburb",
foreignField:"landmark_suburb",
as:"landmarks"
}},{$out:{db:"monUGovDB",coll:"suburbLandmarks"}}])

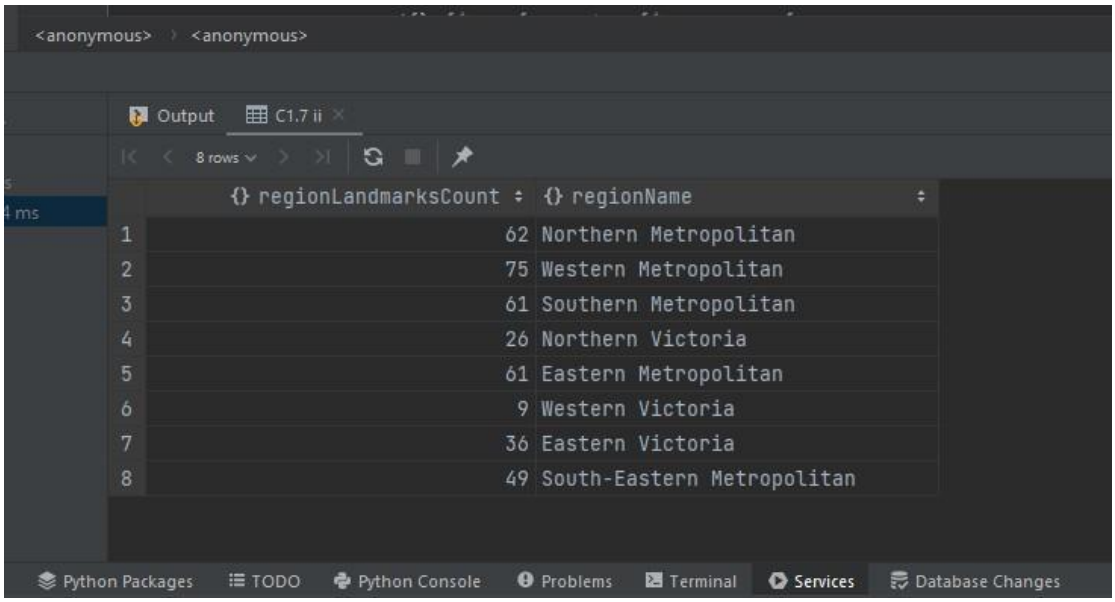
//output after join
db.suburbLandmarks.aggregate([{$match:{"landmarks":{$ne:null}}}]).pretty()



	_id	councilArea	landmarks	postcode	propertyCount	regionName	suburb
1	S1	Yarra City Council	[]	3067	4019	Northern Metropolitan	Abbotsford

//C1.7 ii

db.suburbLandmarks.aggregate([{$group:{_id:"$regionName", regionLandmarksCount:{$sum:1}}},{$project: {_id:0,regionLandmarksCount:"$regionLandmarksCount", regionName:"$_id"}}])



	regionLandmarksCount	regionName
1	62	Northern Metropolitan
2	75	Western Metropolitan
3	61	Southern Metropolitan
4	26	Northern Victoria
5	61	Eastern Metropolitan
6	9	Western Victoria
7	36	Eastern Victoria
8	49	South-Eastern Metropolitan

C1.8

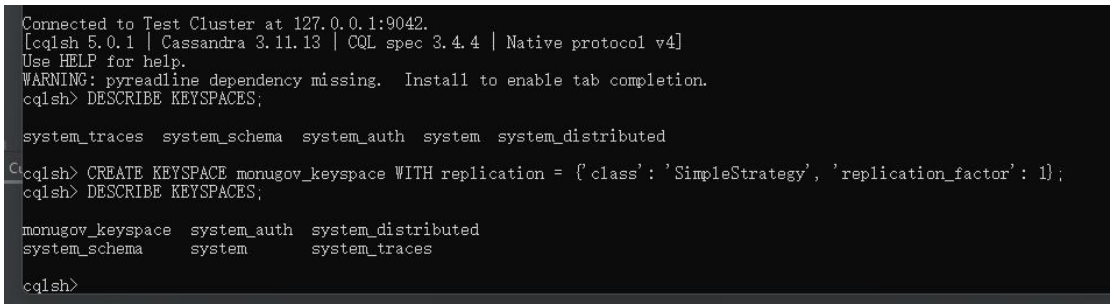
//C1.8 i

```
db.places.createIndex( { location: "2dsphere" } )
db.landmarks.updateMany({},{$set:{location:{$geometry: {
                    type: "point",
                    coordinates: ["$lat", "$lon"]
                }}}})
db.landmarks.find()
```

C2

C2.1

DESCRIBE KEYSPACES;
CREATE KEYSPACE monugov_keyspace WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
DESCRIBE KEYSPACES;



```
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.13 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cqlsh> DESCRIBE KEYSPACES;

system_traces  system_schema  system_auth  system  system_distributed

cqlsh> CREATE KEYSPACE monugov_keyspace WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> DESCRIBE KEYSPACES;

monugov_keyspace  system_auth  system_distributed
system_schema     system       system_traces

cqlsh>
```

C2.2

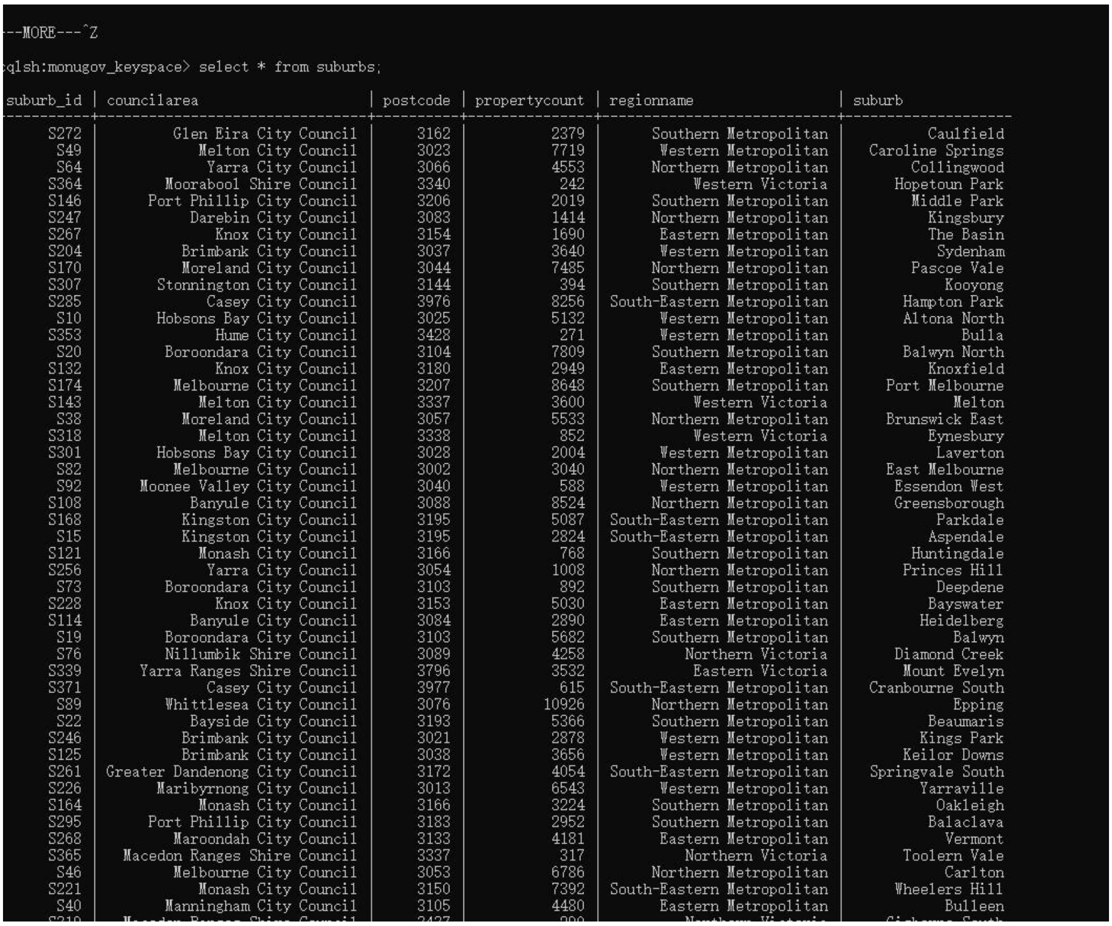
USE monugov_keyspace;

```
/*
 create table
 */

CREATE TABLE landmarks (
 landmark_id VARCHAR,
 category TEXT,
 theme TEXT,
 landmarkName TEXT,
 lat DECIMAL,
 lon DECIMAL,
 houseNo TEXT,
 street TEXT,
 suburb TEXT,
 postcode INT,
 PRIMARY KEY(landmark_id,landmarkName)
);
CREATE TABLE suburbs (
 suburb_id VARCHAR,
 councilArea TEXT,
 suburb TEXT,
 regionName TEXT,
```

```
  postcode INT,
  propertyCount INT,
  PRIMARY KEY(suburb_id, suburb, postcode)
);

/*
  define a type for propertyHistory in properties
  */
CREATE TYPE history (
  sold_by TEXT,
  date TIMESTAMP,
  price INT);

/*
  create table with frozen datatype
  */

CREATE TABLE properties (
  property_id VARCHAR,
  address TEXT,
  postcode INT,
  propertyHistory set<frozen<history>>,
  rooms INT,
  suburb TEXT,
  type TEXT,
  PRIMARY KEY(property_id)
);

/*
  import the csv file
  */

COPY landmarks (landmark_id, category, theme, landmarkName, lat, lon, houseNo, street,
suburb, postcode) FROM 'C:\Users\user\Desktop\landmarks.csv' WITH HEADER = TRUE;
```

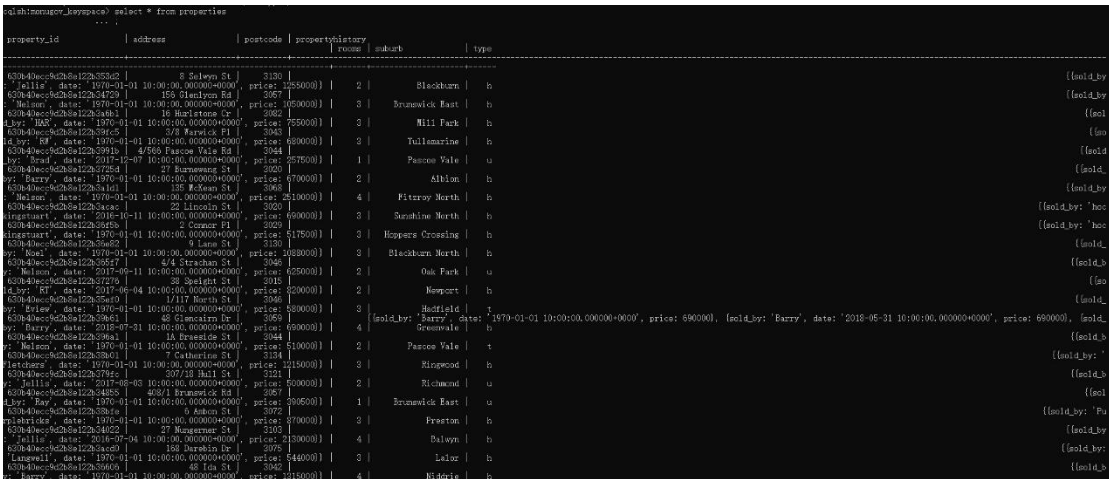
```
cqlsh:monugov_keyspace> select * from landmarks
                    ... ;

 landmark_id | category                             | houseno | landmarkname                                           | lat          | lon         | postcode | street                  | suburb          | theme
-------------+--------------------------------------+---------+--------------------------------------------------------+--------------+-------------+----------+-------------------------+-----------------+---------------------------
------------------------
        L113 | Specialist Residential Accommodation |     717 |                      The Mission To Seafarers Victoria |  -37.8224996 | 144.9512548 |     3008 |          Wurundjeri Way |       Docklands |
      Hostel
        L124 |
 Public Buildings                  Community Use |      NA |                                 Elisabeth Murdoch Hall | -37.82397337 |  144.967451 |     3006 |            Sturt Street |       Southbank |
         L35 |
   Police Station                  Community Use |     226 |                                        Victoria Police |  -37.8164849 | 144.9664868 |     3000 |           Flinders Lane |       Melbourne |
         L80 |
te Sports Club/Facility        Leisure/Recreation |      NA |                    Melbourne International Shooting Club | -37.82928449 | 144.9134158 |     3207 |              Todd Road |  Port Melbourne |                      Priva
        L120 |
  Art Gallery/Museum           Place Of Assembly |      11 |                                       Melbourne Museum | -37.80355181 | 144.9716907 |     3053 |        Nicholson Street |         Carlton |
        L141 |
    Public Hospital              Health Services |      50 |                               Royal Childrens Hospital | -37.79420608 | 144.9500483 |     3051 |          Flemington Road | North Melbourne |
        L123 |
       Theatre Live            Place Of Assembly |     140 |                                            MTC Theatre | -37.82426618 | 144.9681744 |     3006 |     Southbank Boulevard |       Southbank |
         L36 |
     Police Station                Community Use |      36 |                                        Victoria Police | -37.80031213 | 144.9546975 |     3051 |           Wreckyn Street | North Melbourne |
        L134 |
    Railway Station                    Transport |      NA | Jolimont-MCG Railway Station (East Melbourne) - Train stop | -37.81656518 |  144.983641 |     3002 |             Event Gates |  East Melbourne |
         L12 |
             Church            Place of Worship |     339 |                                        Church of Christ | -37.81045241 | 144.9638892 |     3000 |         Swanston Street |       Melbourne |
        L232 |
   Department Store                       Retail |     310 |                                            David Jones | -37.81331273 | 144.9643735 |     3000 |           Bourke Street |       Melbourne |
        L137 |
    Railway Station                    Transport |     308 |    Macaulay Railway Station (North Melbourne) - Train stop |   -37.794964 | 144.9361908 |     3051 |            Macaulay Road | North Melbourne |
        L135 |
y (Park/Garden/Reserve)       Leisure/Recreation |      NA |                       North Melbourne Recreation Reserve | -37.79883453 | 144.9414521 |     3051 |            Macaulay Road | North Melbourne | Informal Outdoor Facilit
         L38 |
     Visitor Centre                Community Use |      NA |                                  Melbourne Visitor Booth | -37.81395251 | 144.9644259 |     3000 |       Bourke Street Mall |       Melbourne |
         L79 |
Land - Undeveloped Site              Vacant Land |      NA |                  Melbourne International Karting Complex | -37.83107464 | 144.9138229 |     3207 |              Cook Street |  Port Melbourne |                     Vacant
        L167 |
lity (Zoo, Golf Course)       Leisure/Recreation |      NA |                                           Riverside Park | -37.79483753 | 144.9156071 |     3031 |        Riverside Reserve |      Kensington |  Outdoor Recreation Faci
        L203 |
y (Park/Garden/Reserve)       Leisure/Recreation |      42 |                                        Murchinson Square |  -37.8002735 | 144.9730593 |     3053 |              Owen Street |         Carlton | Informal Outdoor Facilit
         L70 |
y (Park/Garden/Reserve)       Leisure/Recreation |      54 |                                              Batman Park | -37.82184609 | 144.9566658 |     3000 |            Rebecca Walk |       Melbourne | Informal Outdoor Facilit
        L126 |
   Public Buildings                Community Use |     180 |                                        NGV International |  -37.8230135 | 144.9693425 |     3006 |            St Kilda Road |       Southbank |
        L147 |
   Dwelling (House)  Residential Accommodation |       1 |                                         Government House | -37.82756359 | 144.9766551 |     3000 |   Government House Drive |       Melbourne |
        L197 |
Retail/Office/Carpark                  Mixed Use |     211 |                                        Melbourne Central | -37.81053718 | 144.9627497 |     3000 |         La Trobe Street |       Melbourne |
         L21 |
             Church            Place of Worship |     110 |                               Melbourne Unitarian Church | -37.81144878 | 144.9846649 |     3002 |              Grey Street |  East Melbourne |
        L228 |                        Transport |      NA |                                              Webb Bridge | -37.82320524 |  144.947291 |     3008 |              Webb Bridge |       Docklands |
```

COPY suburbs (suburb_id, councilArea, suburb, regionName, postcode, propertyCount) FROM 'C:\Users\user\Desktop\suburbs.csv' WITH HEADER = TRUE;



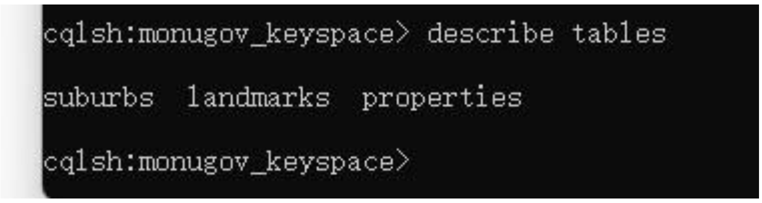
```
---MORE---^Z

cqlsh:monugov_keyspace> select * from suburbs;

 suburb_id | councilarea                  | postcode | propertycount | regionname                 | suburb
-----------+------------------------------+----------+---------------+----------------------------+--------------------
      S272 |       Glen Eira City Council |     3162 |          2379 |       Southern Metropolitan |          Caulfield
       S49 |          Melton City Council |     3023 |          7719 |        Western Metropolitan |   Caroline Springs
       S64 |           Yarra City Council |     3066 |          4553 |       Northern Metropolitan |        Collingwood
      S364 |      Moorabool Shire Council |     3340 |           242 |            Western Victoria |      Hopetoun Park
      S146 |    Port Phillip City Council |     3206 |          2019 |       Southern Metropolitan |        Middle Park
      S247 |         Darebin City Council |     3083 |          1414 |       Northern Metropolitan |          Kingsbury
      S267 |            Knox City Council |     3154 |          1690 |        Eastern Metropolitan |          The Basin
      S204 |        Brimbank City Council |     3037 |          3640 |        Western Metropolitan |           Sydenham
      S170 |        Moreland City Council |     3044 |          7485 |       Northern Metropolitan |        Pascoe Vale
      S307 |     Stonnington City Council |     3144 |           394 |       Southern Metropolitan |            Kooyong
      S285 |           Casey City Council |     3976 |          8256 | South-Eastern Metropolitan |        Hampton Park
       S10 |     Hobsons Bay City Council |     3025 |          5132 |        Western Metropolitan |       Altona North
      S353 |            Hume City Council |     3428 |           271 |        Western Metropolitan |              Bulla
       S20 |      Boroondara City Council |     3104 |          7809 |       Southern Metropolitan |      Balwyn North
      S132 |            Knox City Council |     3180 |          2949 |        Eastern Metropolitan |          Knoxfield
      S174 |       Melbourne City Council |     3207 |          8648 |       Southern Metropolitan |     Port Melbourne
      S143 |          Melton City Council |     3337 |          3600 |            Western Victoria |             Melton
       S38 |        Moreland City Council |     3057 |          5533 |       Northern Metropolitan |     Brunswick East
      S318 |          Melton City Council |     3338 |           852 |            Western Victoria |          Eynesbury
      S301 |     Hobsons Bay City Council |     3028 |          2004 |        Western Metropolitan |           Laverton
       S82 |       Melbourne City Council |     3002 |          3040 |       Northern Metropolitan |     East Melbourne
       S92 |    Moonee Valley City Council |     3040 |           588 |        Western Metropolitan |       Essendon West
      S108 |         Banyule City Council |     3088 |          8524 |       Northern Metropolitan |      Greensborough
      S168 |        Kingston City Council |     3195 |          5087 | South-Eastern Metropolitan |           Parkdale
       S15 |        Kingston City Council |     3195 |          2824 | South-Eastern Metropolitan |          Aspendale
      S121 |          Monash City Council |     3166 |           768 |       Southern Metropolitan |       Huntingdale
      S256 |           Yarra City Council |     3054 |          1008 |       Northern Metropolitan |       Princes Hill
       S73 |      Boroondara City Council |     3103 |           892 |       Southern Metropolitan |           Deepdene
      S228 |            Knox City Council |     3153 |          5030 |        Eastern Metropolitan |          Bayswater
      S114 |         Banyule City Council |     3084 |          2890 |        Eastern Metropolitan |         Heidelberg
       S19 |      Boroondara City Council |     3103 |          5682 |       Southern Metropolitan |             Balwyn
       S76 |      Nillumbik Shire Council |     3089 |          4258 |           Northern Victoria |      Diamond Creek
      S339 |   Yarra Ranges Shire Council |     3796 |          3532 |            Eastern Victoria |       Mount Evelyn
      S371 |           Casey City Council |     3977 |           615 | South-Eastern Metropolitan |    Cranbourne South
       S89 |      Whittlesea City Council |     3076 |         10926 |       Northern Metropolitan |              Epping
       S22 |         Bayside City Council |     3193 |          5366 |       Southern Metropolitan |          Beaumaris
      S246 |        Brimbank City Council |     3021 |          2878 |        Western Metropolitan |          Kings Park
      S125 |        Brimbank City Council |     3038 |          3656 |        Western Metropolitan |        Keilor Downs
      S261 |   Greater Dandenong City Council |     3172 |          4054 | South-Eastern Metropolitan |   Springvale South
      S226 |     Maribyrnong City Council |     3013 |          6543 |        Western Metropolitan |          Yarraville
      S164 |          Monash City Council |     3166 |          3224 |       Southern Metropolitan |            Oakleigh
      S295 |    Port Phillip City Council |     3183 |          2952 |       Southern Metropolitan |           Balaclava
      S268 |       Maroondah City Council |     3133 |          4181 |        Eastern Metropolitan |             Vermont
      S365 |  Macedon Ranges Shire Council |     3337 |           317 |           Northern Victoria |        Toolern Vale
       S46 |       Melbourne City Council |     3053 |          6786 |       Northern Metropolitan |             Carlton
      S221 |          Monash City Council |     3150 |          7392 | South-Eastern Metropolitan |      Wheelers Hill
       S40 |      Manningham City Council |     3105 |          4480 |        Eastern Metropolitan |             Bulleen
```

COPY properties (property_id, address, postcode, propertyHistory, rooms, suburb, type) FROM 'C:\Users\user\Desktop\properties.csv' WITH DELIMITER = '|' AND HEADER = TRUE;



```
cqlsh:monugov_keyspace> select * from properties
                    ... ;

 property_id              | address              | postcode | propertyhistory
                                                                       | rooms | suburb             | type
--------------------------+----------------------+----------+----------------------------------------
-----------------------------------------------------------------------+-------+--------------------+------
 630b40ecc9d2b8e122b353d2 |          8 Selwyn St |     3130 |                                          {{sold_by
: 'Jellis', date: '1970-01-01 10:00:00.000000+0000', price: 1255000}} |     2 |          Blackburn |    h
 630b40ecc9d2b8e122b34729 |      156 Glenlyon Rd |     3057 |                                          {{sold_by
: 'Nelson', date: '1970-01-01 10:00:00.000000+0000', price: 1050000}} |     3 |     Brunswick East |    h
 630b40ecc9d2b8e122b3a6b1 |      16 Hurlstone Cr |     3082 |                                              {{sol
d_by: 'HAR', date: '1970-01-01 10:00:00.000000+0000', price: 755000}} |     3 |          Mill Park |    h
 630b40ecc9d2b8e122b39fc5 |        3/8 Warwick Pl |     3043 |                                              {{so
ld_by: 'RW', date: '1970-01-01 10:00:00.000000+0000', price: 680000}} |     3 |        Tullamarine |    h
 630b40ecc9d2b8e122b3991b |  4/566 Pascoe Vale Rd |     3044 |                                             {{sold
_by: 'Brad', date: '2017-12-07 10:00:00.000000+0000', price: 257500}} |     1 |        Pascoe Vale |    u
 630b40ecc9d2b8e122b3725d |      27 Burnewang St |     3020 |                                            {{sold_
by: 'Barry', date: '1970-01-01 10:00:00.000000+0000', price: 670000}} |     2 |             Albion |    h
 630b40ecc9d2b8e122b3a1d1 |        135 McKean St |     3068 |                                          {{sold_by
: 'Nelson', date: '1970-01-01 10:00:00.000000+0000', price: 2510000}} |     4 |      Fitzroy North |    h
 630b40ecc9d2b8e122b3acac |        22 Lincoln St |     3020 |                                    {{sold_by: 'hoc
kingstuart', date: '2016-10-11 10:00:00.000000+0000', price: 690000}} |     3 |     Sunshine North |    h
 630b40ecc9d2b8e122b36f5b |          2 Connor Pl |     3029 |                                    {{sold_by: 'hoc
kingstuart', date: '1970-01-01 10:00:00.000000+0000', price: 517500}} |     3 |   Hoppers Crossing |    h
 630b40ecc9d2b8e122b36e82 |            9 Lane St |     3130 |                                            {{sold_
by: 'Noel', date: '1970-01-01 10:00:00.000000+0000', price: 1088000}} |     3 |    Blackburn North |    h
 630b40ecc9d2b8e122b365f7 |       4/4 Strachan St |     3046 |                                           {{sold_b
y: 'Nelson', date: '2017-09-11 10:00:00.000000+0000', price: 625000}} |     2 |           Oak Park |    u
 630b40ecc9d2b8e122b37276 |        38 Speight St |     3015 |                                              {{so
ld_by: 'RT', date: '2017-06-04 10:00:00.000000+0000', price: 820000}} |     2 |            Newport |    h
 630b40ecc9d2b8e122b35ef0 |       1/117 North St |     3046 |                                            {{sold_
by: 'Eview', date: '1970-01-01 10:00:00.000000+0000', price: 580000}} |     3 |           Hadfield |    t
 630b40ecc9d2b8e122b39b61 |      48 Glencairn Dr |     3059 | {{sold_by: 'Barry', date: '1970-01-01 10:00:00.000000+0000', price: 690000}, {sold_by: 'Barry', date: '2018-05-31 10:00:00.000000+0000', price: 690000}, {sold_
by: 'Barry', date: '2018-07-31 10:00:00.000000+0000', price: 690000}} |     4 |          Greenvale |    h
 630b40ecc9d2b8e122b396a1 |       1A Braeside St |     3044 |                                           {{sold_b
y: 'Nelson', date: '1970-01-01 10:00:00.000000+0000', price: 510000}} |     2 |        Pascoe Vale |    t
 630b40ecc9d2b8e122b38b01 |      7 Catherine St |     3134 |                                       {{sold_by: '
Fletchers', date: '1970-01-01 10:00:00.000000+0000', price: 1215000}} |     3 |           Ringwood |    h
 630b40ecc9d2b8e122b379fc |       307/18 Hull St |     3121 |                                           {{sold_b
y: 'Jellis', date: '2017-08-03 10:00:00.000000+0000', price: 500000}} |     2 |           Richmond |    u
 630b40ecc9d2b8e122b34855 |    408/1 Brunswick Rd |     3057 |                                              {{sol
d_by: 'Ray', date: '1970-01-01 10:00:00.000000+0000', price: 390500}} |     1 |     Brunswick East |    u
 630b40ecc9d2b8e122b38bfe |          6 Ambon St |     3072 |                                     {{sold_by: 'Pu
rplebricks', date: '1970-01-01 10:00:00.000000+0000', price: 870000}} |     3 |            Preston |    h
 630b40ecc9d2b8e122b34022 |       27 Nungerner St |     3103 |                                         {{sold_by
: 'Jellis', date: '2016-07-04 10:00:00.000000+0000', price: 2130000}} |     4 |             Balwyn |    h
 630b40ecc9d2b8e122b3acd0 |      168 Darebin Dr |     3075 |                                        {{sold_by:
 'Langwell', date: '1970-01-01 10:00:00.000000+0000', price: 544000}} |     3 |              Lalor |    h
 630b40ecc9d2b8e122b36606 |            48 Ida St |     3042 |                                           {{sold_b
y: 'Barry', date: '1970-01-01 10:00:00.000000+0000', price: 1315000}} |     4 |            Niddrie |    h
```

Show Table in list of table



```
cqlsh:monugov_keyspace> describe tables

suburbs  landmarks  properties

cqlsh:monugov_keyspace>
```

C2.3

*/**
 insert into different tables
 **/*

insert into properties (property_id, address, postcode, propertyHistory, rooms, suburb, type)
VALUES ('insert1', '19 Kinlock St', 3085, {{sold_by: 'Darren', date: '1970-01-01T00:00:00Z', price: 1120000}}, 5, 'Macleod', 'house');
insert into landmarks (landmark_id, landmarkName, category, theme, lat, lon, houseNo, street)
VALUES ('insert2','Gresswell Theatre','Place Of Assembly', 'Theatre Live', -37.712422, 145.072617, '1', 'Forrest Road');

*/**
 output after insert
 **/*

*SELECT **
FROM properties
WHERE property_id = 'insert1';

*SELECT **

```
FROM landmarks
WHERE landmark_id = 'insert2';

/*
 Alternate solution
 */
CREATE INDEX ON properties ( address );
SELECT *
FROM properties
WHERE address = '19 Kinlock St';
CREATE INDEX ON landmarks ( landmarkName );
SELECT *
FROM landmarks
WHERE landmarkName = 'Gresswell Theatre';
```

```
cqlsh:monugov_keyspace> select * from properties where property_id = 'insert1';

 property_id | address       | postcode | propertyhistory                                                                  | rooms | suburb  | type
-------------+---------------+----------+----------------------------------------------------------------------------------+-------+---------+-------
    insert1 | 19 Kinlock St |     3085 | {{sold_by: 'Darren', date: '1970-01-01 00:00:00.000000+0000', price: 1120000}} |     5 | Macleod | house

(1 rows)
```

```
cqlsh:monugov_keyspace> select * from landmarks where landmark_id = 'insert2';

 landmark_id | category          | houseno | landmarkname      | lat        | lon        | postcode | street       | suburb | theme
-------------+-------------------+---------+-------------------+------------+------------+----------+--------------+--------+-------------
    insert2 | Place Of Assembly |       1 | Gresswell Theatre | -37.712422 | 145.072617 |     null | Forrest Road |   null | Theatre Live
```

C2.4

C2.4 i

```
CREATE INDEX ON suburbs ( suburb );
SELECT * FROM suburbs WHERE suburb = 'Caulfield';

/*
 alternate solution
 */
SELECT * FROM suburbs WHERE suburb = 'Caulfield' ALLOW FILTERING;
```

```
cqlsh:monugov_keyspace> SELECT * FROM suburbs WHERE suburb = 'Caulfield' ALLOW FILTERING;

 suburb_id | suburb    | councilarea            | postcode | propertycount | regionname
-----------+-----------+------------------------+----------+---------------+-----------------------
      S272 | Caulfield | Glen Eira City Council |     3162 |          2379 | Southern Metropolitan

(1 rows)
```

C2.4 ii

add new columns

ALTER TABLE suburbs ADD otherHomeGround BOOLEAN;
ALTER TABLE suburbs ADD team INT;

/*
update columns
*/

UPDATE suburbs
SET otherHomeGround = true, team = 16
where suburb = 'Caulfield';

/*
display related field
*/
*SELECT **
FROM suburbs
WHERE suburb = 'Caulfield';

```
cqlsh:monugov_keyspace> SELECT * FROM suburbs WHERE suburb = 'Caulfield' ALLOW FILTERING;

 suburb_id | suburb    | councilarea            | otherhomeground | postcode | propertycount | regionname            | team
-----------+-----------+------------------------+-----------------+----------+---------------+-----------------------+------
      S272 | Caulfield | Glen Eira City Council |            True |     3162 |          2379 | Southern Metropolitan |   16

(1 rows)
cqlsh:monugov_keyspace>
```

C2.4 iii

/* add TTL */

UPDATE suburbs USING TTL 300000
SET otherHomeGround = true, team = 16
WHERE suburb = 'Caulfield' AND suburb_id = 'S272';

SELECT otherHomeGround, TTL(otherHomeGround), writetime(otherHomeGround), team, TTL(team), writetime(team)
FROM suburbs
WHERE suburb = 'Caulfield'
AND suburb_id = 'S272';

```
cqlsh:monugov_keyspace> UPDATE suburbs USING TTL 300000
                   ... SET otherHomeGround = true, team = 16
                   ... WHERE suburb = 'Caulfield' AND suburb_id = 'S272';
cqlsh:monugov_keyspace> SELECT otherHomeGround, TTL(otherHomeGround), team, TTL(team)
                   ... FROM suburbs
                   ... WHERE suburb = 'Caulfield'
                   ...   AND suburb_id = 'S272';

 otherhomeground | ttl(otherhomeground) | team | ttl(team)
-----------------+----------------------+------+-----------
            True |               299992 |   16 |    299992

(1 rows)
cqlsh:monugov_keyspace> SELECT otherHomeGround, TTL(otherHomeGround), writetime(otherHomeGround), team, TTL(team), writetime(team)
                   ... FROM suburbs
                   ... WHERE suburb = 'Caulfield'
                   ...   AND suburb_id = 'S272';

 otherhomeground | ttl(otherhomeground) | writetime(otherhomeground) | team | ttl(team) | writetime(team)
-----------------+----------------------+----------------------------+------+-----------+------------------
            True |               299855 |          1665986850549000 |   16 |    299855 | 1665986850549000

(1 rows)
cqlsh:monugov_keyspace>
```

C2.5

CREATE INDEX ON properties (type);

```
cqlsh:monugov_keyspace> CREATE INDEX ON properties ( type );
cqlsh:monugov_keyspace> SELECT * FROM properties WHERE type = 'u';
```

After create the index:

```
cqlsh:monugov_keyspace> CREATE INDEX ON properties ( type );
InvalidRequest: Error from server: code=2200 [Invalid query] message="Index properties_type_idx_1 is a duplicate of existing index properties_type_idx"
cqlsh:monugov_keyspace>
```

C2.6

C2.6 i

SELECT COUNT(*) FROM properties WHERE type = 'u';

```
cqlsh:monugov_keyspace> SELECT COUNT(*) FROM properties WHERE type = 'u';

 count
-------
  5817

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:monugov_keyspace>
```

C2.6 ii

*SELECT * FROM landmarks WHERE postcode > 3200 ALLOW FILTERING;*

```
cqlsh:monugov_keyspace> SELECT * FROM landmarks WHERE postcode > 3200 ALLOW FILTERING;

 landmark_id | landmarkname                          | category          | houseno | lat          | lon          | postcode | street       | suburb         | theme
-------------+---------------------------------------+-------------------+---------+--------------+--------------+----------+--------------+----------------+-------------------------------------------
         L80 |   Melbourne International Shooting Club | Leisure/Recreation |      NA | -37.82928449 | 144.9134158 |     3207 |    Todd Road | Port Melbourne |                   Private Sports Club/Facility
         L79 | Melbourne International Karting Complex |       Vacant Land |      NA | -37.83107464 | 144.9138229 |     3207 |  Cook Street | Port Melbourne |                   Vacant Land - Undeveloped Site
        L210 |                     Westgate Park | Leisure/Recreation |      NA | -37.83149186 | 144.9038248 |     3207 |     Bay Trail | Port Melbourne | Informal Outdoor Facility (Park/Garden/Reserve)
        L175 |                              Kraft |        Industrial |       1 | -37.82623131 | 144.9234757 |     3207 | Vegemite Way | Port Melbourne |                   Industrial (Manufacturing)

(4 rows)
cqlsh:monugov_keyspace>
```

C3 Report

Three databases (RDBMS (Oracle), Document-Oriented Database (MongoDB), and Column-Oriented Database (Cassandra)) have been chosen as the data storage technology for MonUGov. MonUGov will provide a unified query platform that allows users to search for content from different databases at once. When a user sends a request on the platform, several agents will start working in the background. These agents translate user requests into different statements through a translation model built by artificial intelligence. The agents send database requests in different languages to different databases. When these requests are answered, they are stored in a separate database. The contents of the separate database are merged and sent to the agent according to the user's needs. The agent will send the results to the user side and transform them into a user search result page. At the same time, the proxy will send the results to the algorithm so that the algorithm can learn again and upgrade the model.
The above is a more complex approach. Much simpler is to manually specify how the software translates user requests into professional syntax.

To meet the needs of multi-tier data storage, we will compare the advantages and disadvantages of three databases in a multi-dimensional manner and select the appropriate data storage technology for each type of data. The analysis dimension is taken as the CAP theorem, which includes consistency, availability, and partition tolerance, and the three databases we choose to correspond to three combinations of CAP: CA, CP, and AP[1].

RDBMS is a data storage technology that focuses more on the CA side. The advantage is that the data in the database is always the same because there is only one server. Also, the data is always valid and there is no version difference because there is only one server. However, as a result, partition tolerance is not allowed. if the only server fails, all the data will fail. The type of data suitable for traditional databases should look at immediate

consistency and availability of data.

1. Cafes and restaurant data should be placed in an RDBMS. From the data table, the number of seats should change in real-time. Users may choose whether to make a reservation for a restaurant or cafe based on the number of seats left. After the user submits the reservation, the changes should also be synchronized in real-time to ensure consistency when everyone accesses. Also, the reservation platform should always allow users to view the number of seats remaining. Therefore, availability and consistency are crucial for this data type. We choose to assign the RDBMS to it.

Next is the CP type of database. Mongodb is document oriented database, which has CP attribute. A database with CP attribute will value data consistency and partial availability more. In other words, when a part of data is inconsistent with the latest data, it will prohibit users from accessing this part of data and pay availability to ensure consistency.

1. Properties.json data should be stored in MongoDB because document oriented databases are very good at JSON type data. Also, Properties.json has documents with different contents, which is a good fit for MongoDB, a data storage technology without a specific schema. Also, for users, any change in properties is important, and immediate consistency becomes important. And when a part of the cluster is inconsistent with the latest version, users can be disabled from accessing the lagging version. Thus, MongoDB, which focuses on consistency and partition tolerance, is particularly well suited for properties.json.
2. Suburb.csv is also suitable for storage in a CP database. Suburb is data that will not be rewritten frequently. If the contents of a suburb are rewritten, the government is the only entity that has the authority to perform this action. When a government rewrite is published, one server will have the latest version, while the rest of the servers that are behind will immediately stop any access to prevent incorrect data from being read. Only the latest version of the server is allowed to be accessed. This is partitioning tolerance, and all servers that have had access stopped are sacrificing availability for consistency. So it is a good idea to store the suburb.csv in the CP database (MongoDB).
3. Similarly, landmarks.csv should be stored in MongoDB. Landmarks do not change often. Also, landmarks stored in government systems are approved by the government before they are entered. The government is, therefore, the only entity that has permission to write data, similar to suburb.csv. So I chose MongoDB database for this data.

Finally, Cassandra is a column oriented database with the AP property. Databases with AP attributes give up immediate consistency in exchange for availability and partition tolerance, and are suitable for storing data types that are not often written to but are frequently accessed.

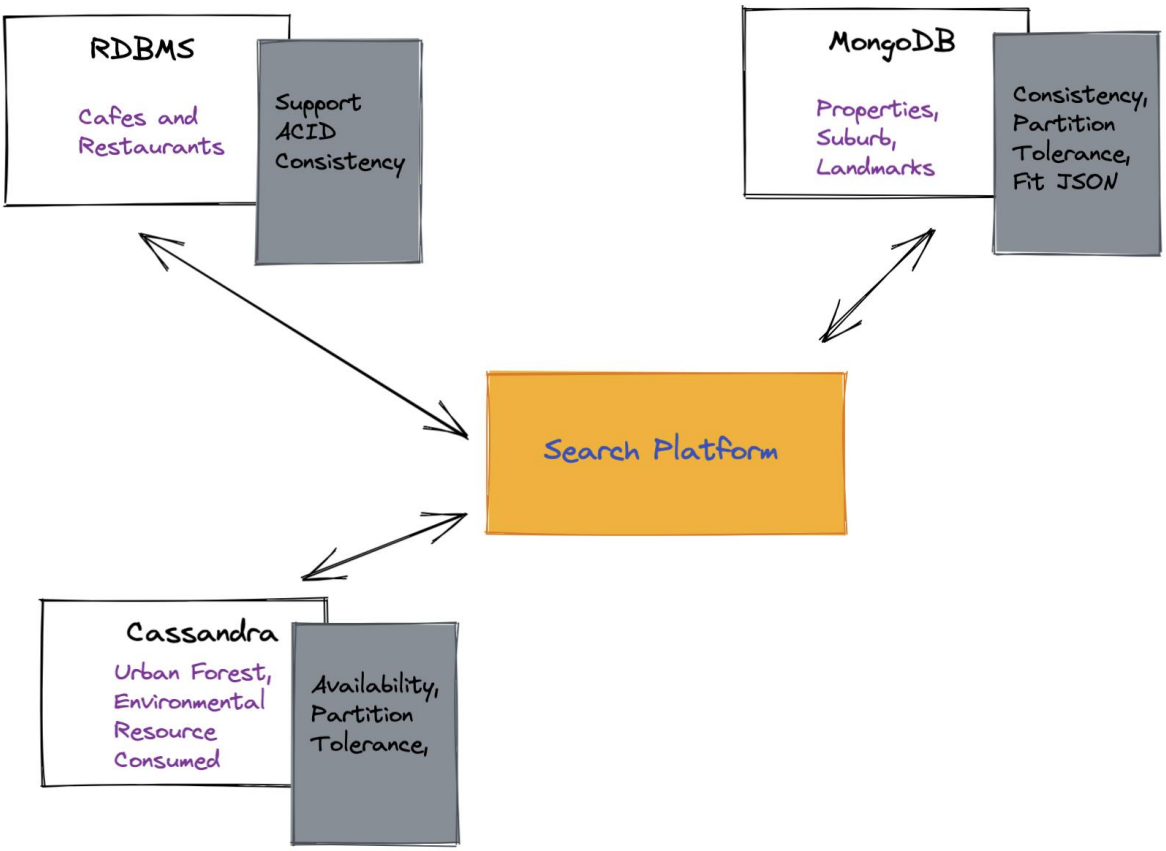
1. The urban forest data should be stored in an AP database represented by Cassandra. Suppose that there are seven forest rangers in Australia who are responsible for writing forest data for seven regions. These rangers need to upload the forest data for their area to the database by 5:00 pm every night. After five o'clock, the data from different regions are uploaded to the servers in different regions. Availability is very important to prevent Ranger A from uploading data and other rangers from uploading it. The whole cluster should still be allowed to write when the data is not uniform. Also, partition management is exceptionally important in order to prevent conflicts when rangers in different regions upload data. After all the data is uploaded to the server, the whole cluster then handles the consistency. Therefore, Cassandra is well suited for urban forest data.
2. Environmental resources consumed should be stored in Cassandra. Similar to forest data, environmental resource data should be managed by region. Similar to forest data, administrators read and write to the database will also happen to environmental resource data. Therefore, the consumed environmental resources belong to this database.

Strength and weakness

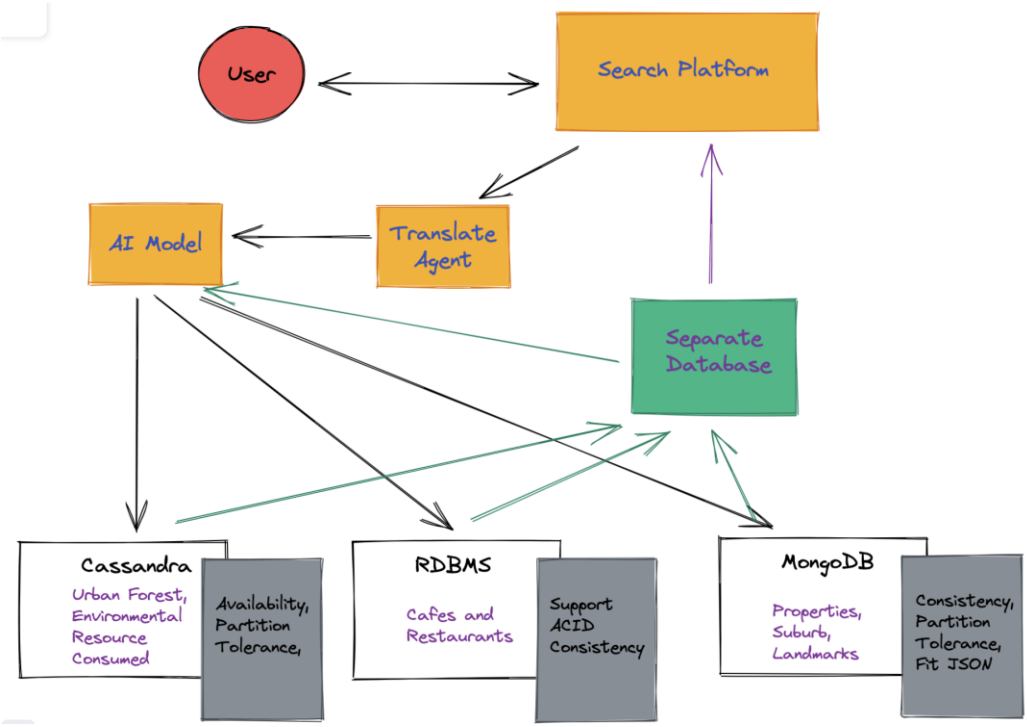
Database and Dataset	Strength	Weakness
RDBMS + Cafes and Restaurants	Consistency Availability	Partition Tolerance
MongoDB + Properties.json	JSON Type Consistency Partition Tolerance	Availability
MongoDB + Suburb	Consistency Partition Tolerance	Availability
MongoDB + Landmarks	Consistency Partition Tolerance	Availability
Cassandra + Urban Forest Data	Availability Partition Tolerance	Consistency
Cassandra + Environmental Resource Consumed	Availability Partition Tolerance	Consistency

Data Architecture Diagram

Simple Polyglot Persistence



Detailed Polyglot Persistence



Reference

1. Bikas Katwal. (2019, September 28). What is the CAP Theorem? MongoDB vs Cassandra vs RDBMS, where do they stand in the CAP theorem? Retrieved September 22, 2022, from Medium website: https://bikas-katwal.medium.com/mongodb-vs-cassandra-vs-rdbms-where-do-they-stand-in-the-cap-theorem-1bae779a7a15
2. Unit: FIT3176 Advanced database design - S2 2022. (2022). Retrieved September 23, 2022, from Monash.edu website: https://lms.monash.edu/course/view.php?id=140830