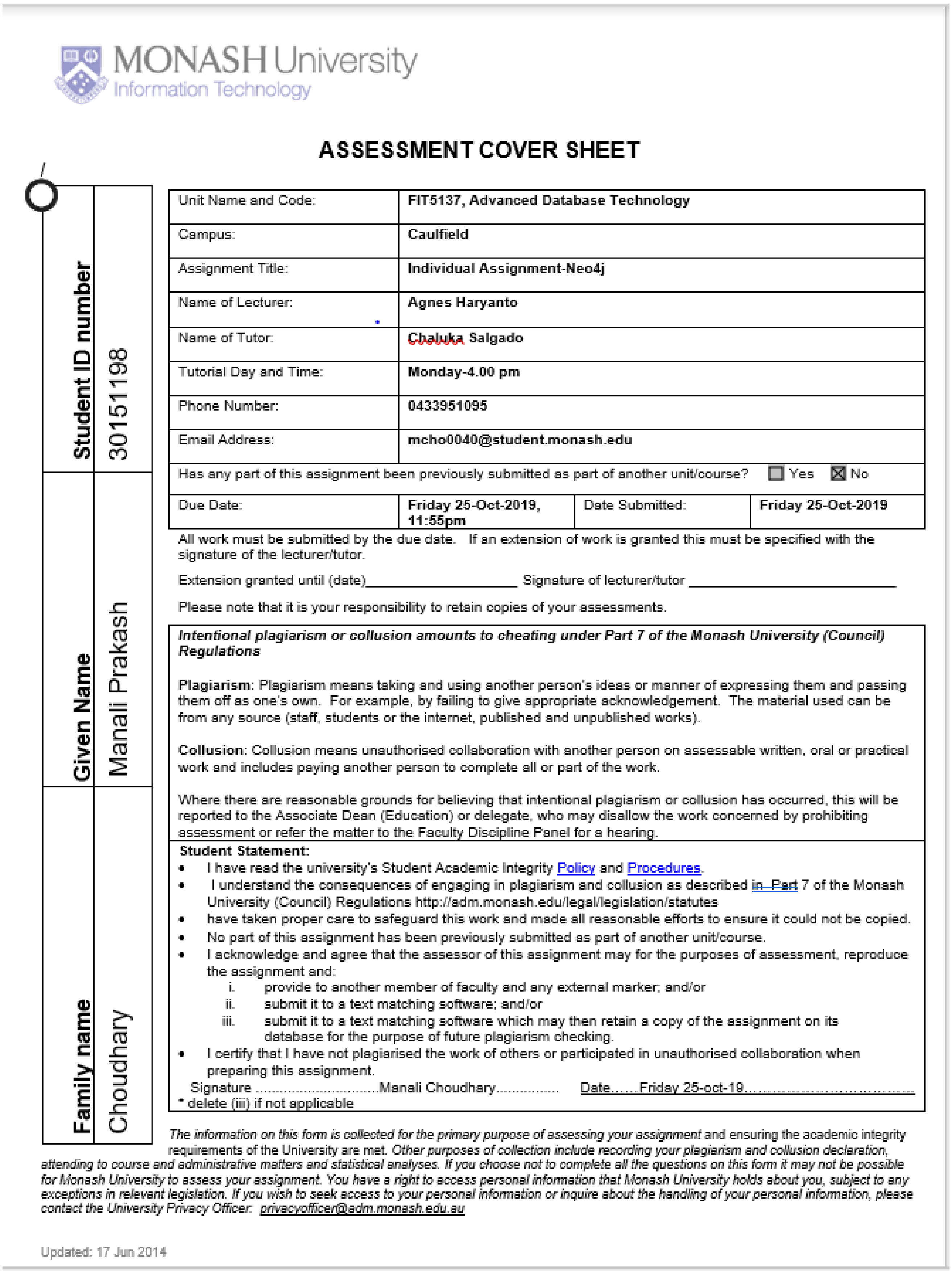
FIT5137, ADVANCED DATABASE TECHNOLOGY

Individual Assignment (Neo4j)- Sem 2/2019

Tutor
Chaluka Salgado

Submitted by
Manali Choudhary
30151198



MONASH University
Information Technology

ASSESSMENT COVER SHEET

Student ID number	30151198
Given Name	Manali Prakash
Family name	Choudhary

Unit Name and Code:	FIT5137, Advanced Database Technology		
Campus:	Caulfield		
Assignment Title:	Individual Assignment-Neo4j		
Name of Lecturer:	Agnes Haryanto		
Name of Tutor:	Chaluka Salgado		
Tutorial Day and Time:	Monday-4.00 pm		
Phone Number:	0433951095		
Email Address:	mcho0040@student.monash.edu		
Has any part of this assignment been previously submitted as part of another unit/course? ☐ Yes ☒ No			
Due Date:	Friday 25-Oct-2019, 11:55pm	Date Submitted:	Friday 25-Oct-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 The material used can be from any source (staff, students or the internet, published and unpublished works).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
- 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 ....Manali Choudhary............... Date……Friday 25-oct-19………………………………

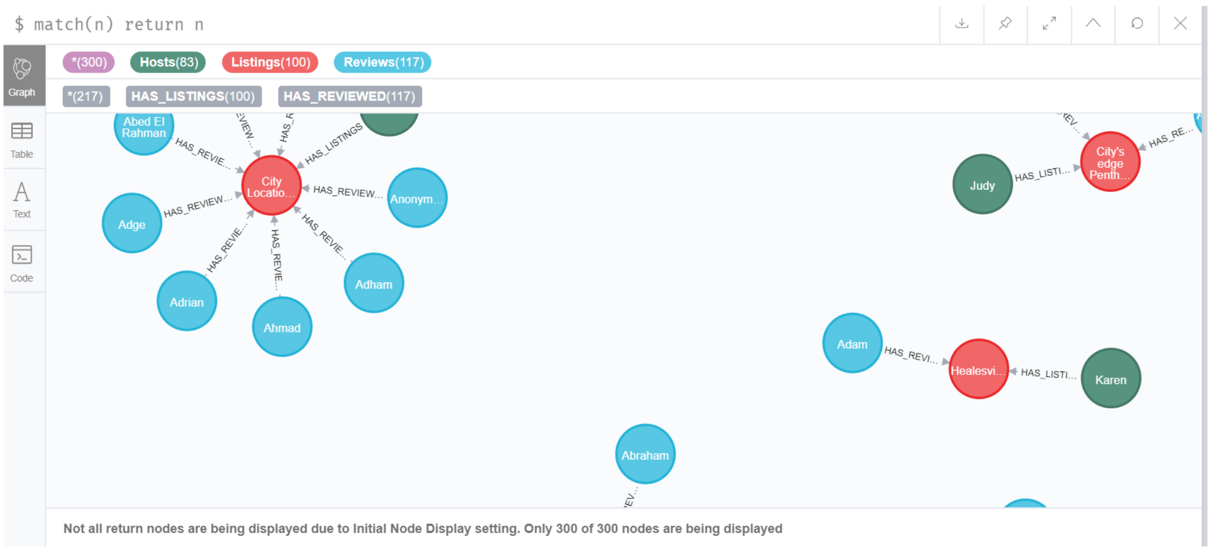
* delete (iii) if not applicable

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

C.1. Database Design.

```
--HOSTS--

LOAD CSV WITH HEADERS FROM "file:///host_v2.csv"

AS host

WITH host WHERE host.host_id IS NOT NULL

MERGE (h:Hosts {hostId: host.host_id})

ON CREATE SET h.hostUrl = host.host_url,

 h.hostName = host.host_name,

 h.hostVerifications = host.host_verifications,

 h.hostSince = host.host_since,

 h.hostLocation = host.host_location,

 h.hostResponseTime = host.host_response_time,

 h.hostIsSuperhost = host.host_is_superhost

RETURN h

MATCH (h:Hosts)

SET h.hostVerifications = replace(h.hostVerifications, "[", ""),

            h.hostVerifications = replace(h.hostVerifications, "]", ""),

            h.hostVerifications = replace(h.hostVerifications, " ", ""),

            h.hostVerifications = replace(h.hostVerifications, "'", ""),

            h.hostVerifications = split(h.hostVerifications, ","),

            h.hostIsSuperhost = (case h.hostIsSuperhost when 'f' then false else true end),

            h.hostSince = Date(h.hostSince),

            h.hostId = toInt(h.hostId)

RETURN h

--LISTINGS—

LOAD CSV WITH HEADERS FROM "file:///listing_v2.csv"

AS listing

WITH listing WHERE listing.id IS NOT NULL

MERGE (l:Listings {id: listing.id})

ON CREATE SET l.name = listing.name,

 l.summary        = listing.summary,
 l.listingUrl        =listing.listing_url,
 l.pictureUrl=listing.picture_url,
 l.hostId = listing.host_id,
 l.neighbourhood = listing.neighbourhood,
 l.street = listing.street,
 l.zipcode = listing.zipcode,
 l.latitude = listing.latitude,
 l.longitude = listing.longitude,
 l.roomType = listing.room_type,
 l.amenities = listing.amenities,
 l.price = listing.price,
 l.extraPeople = listing.extra_people,
 l.minimumNights = listing.minimum_nights,
 l.calculatedHostListingsCount = listing.calculated_host_listings_count,
 l.availability365 = listing.availability_365
 RETURN l

MATCH (h:Hosts),(l:Listings)
where h.hostId=toInt(l.hostId)
CREATE (h)-[:HAS_LISTINGS]->(l)
SET l.id = toInt(l.id),
        l.hostId = toInt(l.hostId),
        l.extraPeople = replace(l.extraPeople,"$",""),
        l.extraPeople = replace(l.extraPeople," ",""),
        l.extraPeople = toInt(l.extraPeople),
        l.amenities = replace(l.amenities, "{", ""),
        l.amenities = replace(l.amenities, "}", ""),
        l.amenities = replace(l.amenities, '"', ""),
        l.amenities = split(l.amenities, ","),
        l.zipcode = toInt(l.zipcode),
        l.price = toFloat(l.price),
        l.availability365 = toInt(l.availability365),
        l.minimumNights = toInt(l.minimumNights),
        l.latitude = toFloat(l.latitude),
        l.longitude = toFloat(l.longitude),
        l.calculatedHostListingsCount = toInt(l.calculatedHostListingsCount)
RETURN l,h
--REVIEW—
LOAD CSV WITH HEADERS FROM "file:///review_v2.csv"
AS review
WITH review WHERE review.id IS NOT NULL
MERGE (r: Reviews {id: review.id})
ON CREATE SET r.listingId = review.listing_id,
r.reviewerId = review.reviewer_id,
r.date = review.date,
r.reviewerName = review.reviewer_name,
r.reviewScoresRating = review.review_scores_rating,
r.comments = review.comments
RETURN r

MATCH (l:Listings),(r:Reviews)
where l.id=toInt(r.listingId)
CREATE (r)-[:HAS_REVIEWED]->(l)
SET r.id = toInt(r.id),
        r.listingId = toInt(r.listingId),
        r.reviewerId = toInt(r.reviewerId),
        r.reviewScoresRating = toInt(r.reviewScoresRating),
        r.date = Date(r.date)
RETURN r,l
```



```
$ match(n) return n
```

*(300) Hosts(83) Listings(100) Reviews(117)

*(217) HAS_LISTINGS(100) HAS_REVIEWED(117)

Not all return nodes are being displayed due to Initial Node Display setting. Only 300 of 300 nodes are being displayed

Explanation on Graph Design:

As required by MonashBnB and the given data, 3 nodes are mapped named- Hosts, Listings, Reviews. The nodes are connected by directed edges. Hosts and Listings share edge named - HAS_LISTINGS and Listings and Reviews with the edge - HAS_REVIEWED. The properties - hostId of Hosts, id of Listings and id of Reviews nodes are constrained with NOT NULL and are unique. The nodes are related to each other by ids of Host-Listing-Review. This mapping of nodes and edges together forms the Graph database of MonashBNB.

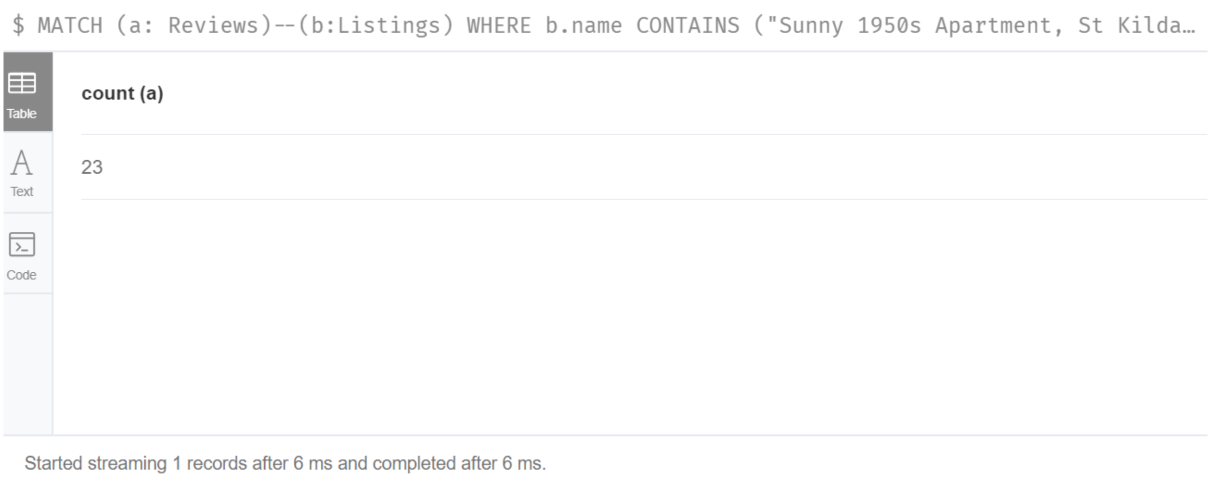
C.2. Queries

1.

MATCH (a: Reviews)--(b:Listings)

WHERE b.name CONTAINS ("Sunny 1950s Apartment, St Kilda East")

RETURN count (a)



```
$ MATCH (a: Reviews)--(b:Listings) WHERE b.name CONTAINS ("Sunny 1950s Apartment, St Kilda…
```

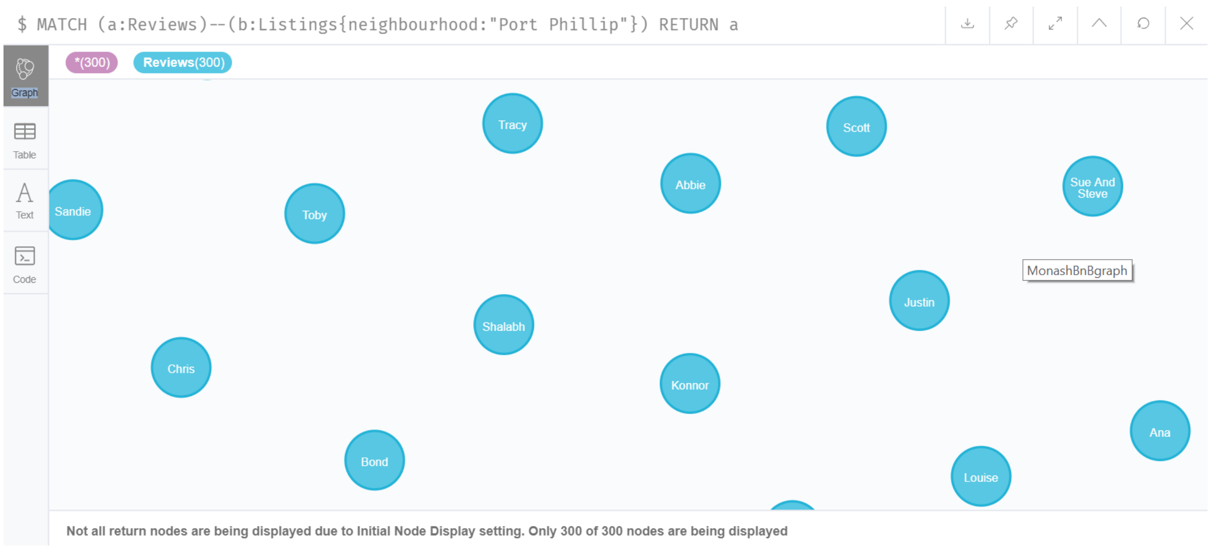
count (a)
23

Started streaming 1 records after 6 ms and completed after 6 ms.

2.

MATCH (a:Reviews)--(b:Listings{neighbourhood:"Port Phillip"})

RETURN a



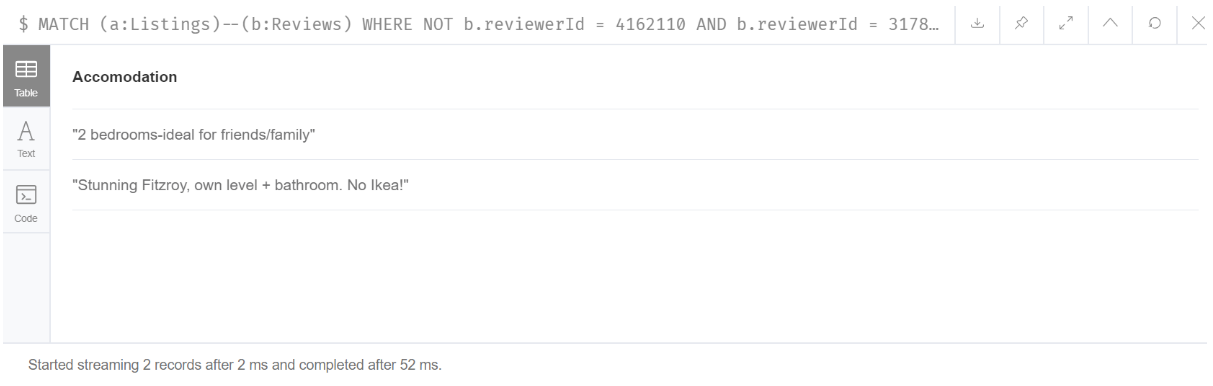
```
$ MATCH (a:Reviews)--(b:Listings{neighbourhood:"Port Phillip"}) RETURN a
```

Not all return nodes are being displayed due to Initial Node Display setting. Only 300 of 300 nodes are being displayed

3.

MATCH (a:Listings)--(b:Reviews) WHERE NOT b.reviewerId = 4162110 AND b.reviewerId = 317848 AND b.reviewScoresRating > 90

RETURN a.name AS Accomodation


```
$ MATCH (a:Listings)--(b:Reviews) WHERE NOT b.reviewerId = 4162110 AND b.reviewerId = 3178…
```

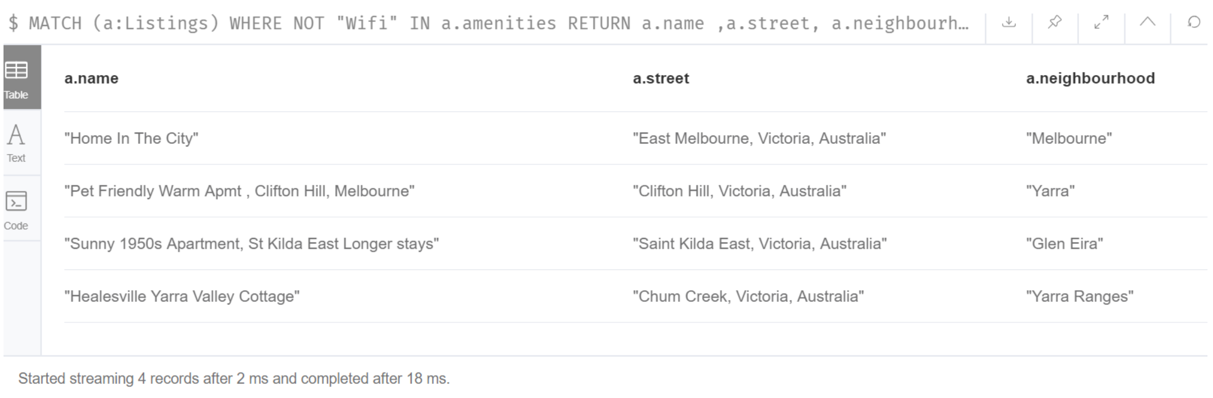
Accomodation
"2 bedrooms-ideal for friends/family"
"Stunning Fitzroy, own level + bathroom. No Ikea!"

Started streaming 2 records after 2 ms and completed after 52 ms.

4.

MATCH (a:Listings) WHERE NOT "Wifi" IN a.amenities

RETURN a.name ,a.street, a.neighbourhood


```
$ MATCH (a:Listings) WHERE NOT "Wifi" IN a.amenities RETURN a.name ,a.street, a.neighbourh…
```

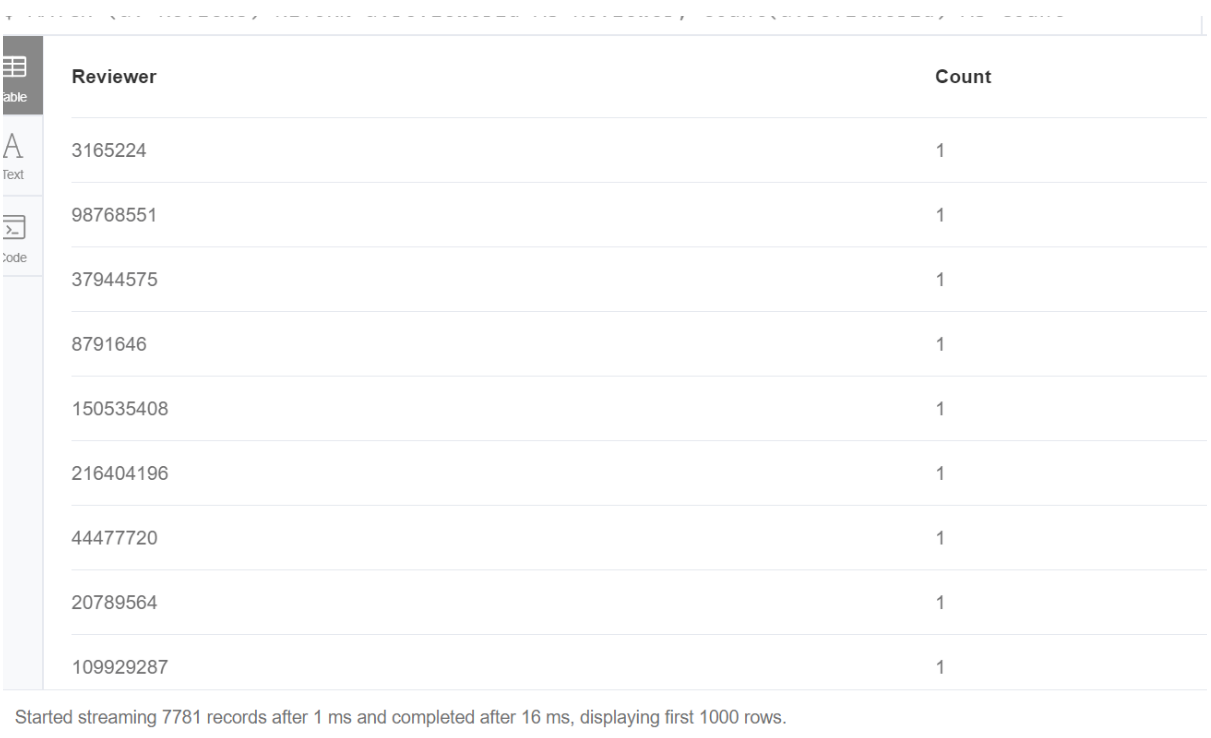
a.name	a.street	a.neighbourhood
"Home In The City"	"East Melbourne, Victoria, Australia"	"Melbourne"
"Pet Friendly Warm Apmt , Clifton Hill, Melbourne"	"Clifton Hill, Victoria, Australia"	"Yarra"
"Sunny 1950s Apartment, St Kilda East Longer stays"	"Saint Kilda East, Victoria, Australia"	"Glen Eira"
"Healesville Yarra Valley Cottage"	"Chum Creek, Victoria, Australia"	"Yarra Ranges"

Started streaming 4 records after 2 ms and completed after 18 ms.

5.

MATCH (a: Reviews)

RETURN a.reviewerId AS Reviewer, count(a.reviewerId) AS Count



Reviewer	Count
3165224	1
98768551	1
37944575	1
8791646	1
150535408	1
216404196	1
44477720	1
20789564	1
109929287	1

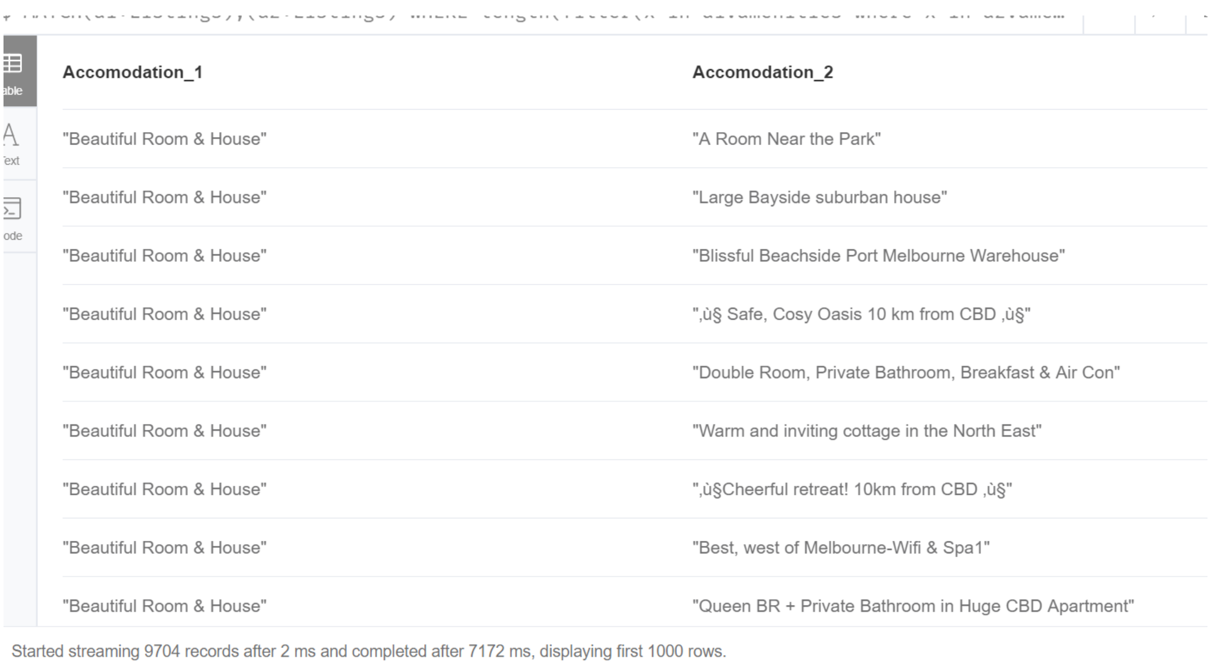
Started streaming 7781 records after 1 ms and completed after 16 ms, displaying first 1000 rows.

6.

MATCH(a1:Listings),(a2:Listings) WHERE

length(filter(x in a1.amenities where x in a2.amenities))>3 and NOT a1.name=a2.name

RETURN a1.name AS Accomodation_1,a2.name AS Accomodation_2

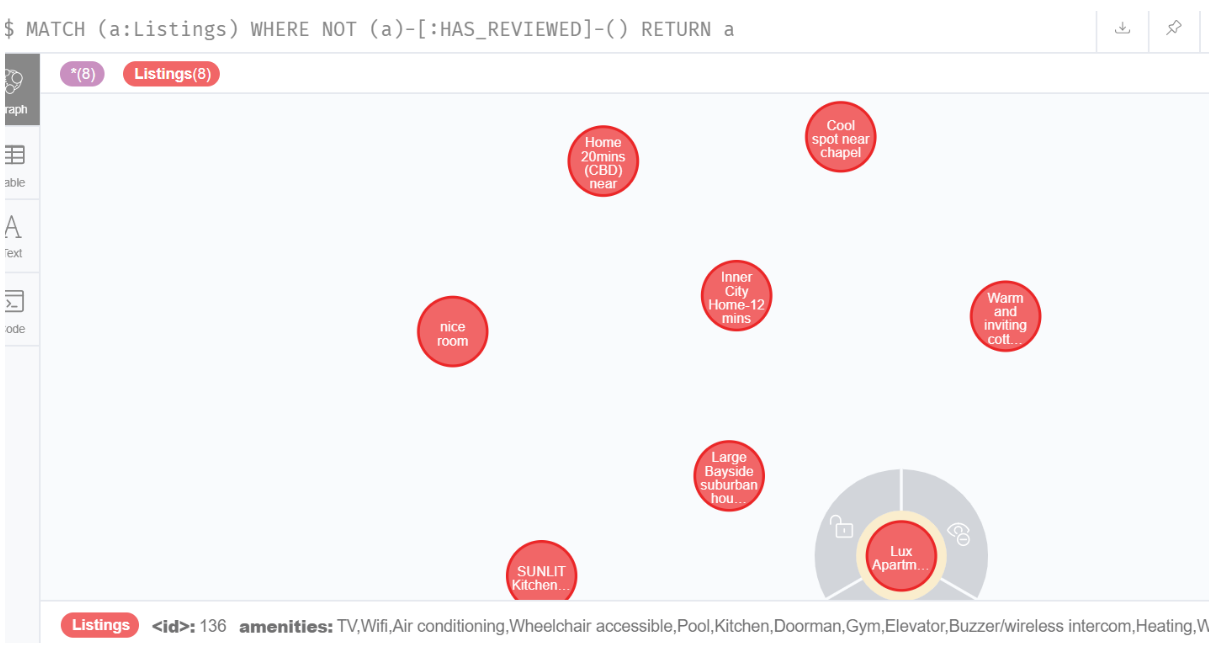


Accomodation_1	Accomodation_2
"Beautiful Room & House"	"A Room Near the Park"
"Beautiful Room & House"	"Large Bayside suburban house"
"Beautiful Room & House"	"Blissful Beachside Port Melbourne Warehouse"
"Beautiful Room & House"	",ù§ Safe, Cosy Oasis 10 km from CBD ,ù§"
"Beautiful Room & House"	"Double Room, Private Bathroom, Breakfast & Air Con"
"Beautiful Room & House"	"Warm and inviting cottage in the North East"
"Beautiful Room & House"	",ù§Cheerful retreat! 10km from CBD ,ù§"
"Beautiful Room & House"	"Best, west of Melbourne-Wifi & Spa1"
"Beautiful Room & House"	"Queen BR + Private Bathroom in Huge CBD Apartment"

Started streaming 9704 records after 2 ms and completed after 7172 ms, displaying first 1000 rows.

7.

MATCH (a:Listings) WHERE NOT (a)-[:HAS_REVIEWED]-() RETURN a


```
$ MATCH (a:Listings) WHERE NOT (a)-[:HAS_REVIEWED]-() RETURN a
```

```
Listings <id>: 136 amenities: TV,Wifi,Air conditioning,Wheelchair accessible,Pool,Kitchen,Doorman,Gym,Elevator,Buzzer/wireless intercom,Heating,W
```

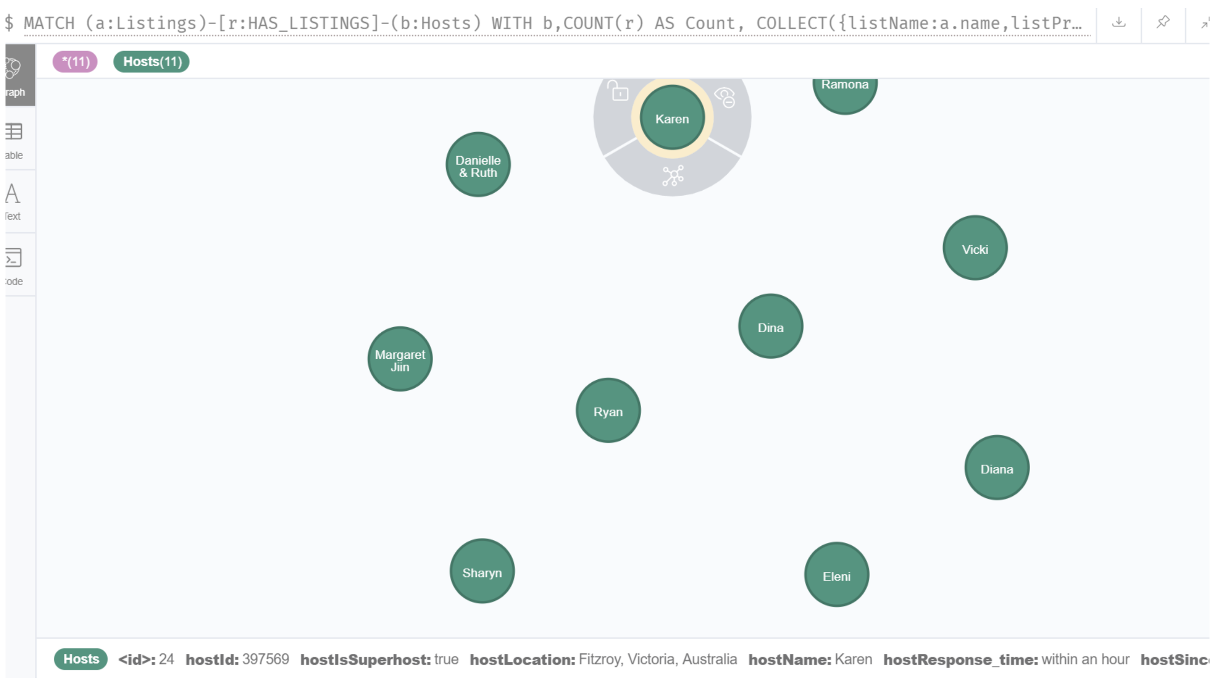
8.

MATCH (a:Listings)-[r:HAS_LISTINGS]-(b:Hosts) WITH b,COUNT(r) AS Count,

COLLECT({listName:a.name,listPrice:a.price}) AS Lists

WHERE Count >1

RETURN b AS Host,Lists,Count

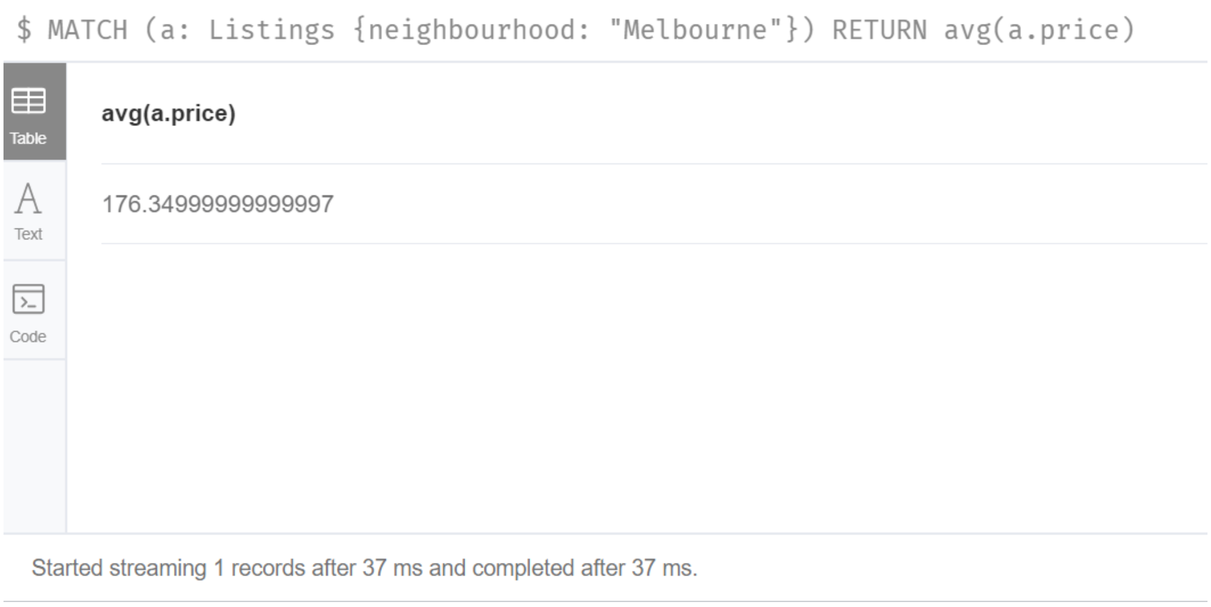

```
$ MATCH (a:Listings)-[r:HAS_LISTINGS]-(b:Hosts) WITH b,COUNT(r) AS Count, COLLECT({listName:a.name,listPr…
```

```
Hosts <id>: 24 hostId: 397569 hostIsSuperhost: true hostLocation: Fitzroy, Victoria, Australia hostName: Karen hostResponse_time: within an hour hostSinc
```

9.

MATCH (a: Listings {neighbourhood: "Melbourne"})

RETURN avg(a.price)



```
$ MATCH (a: Listings {neighbourhood: "Melbourne"}) RETURN avg(a.price)
```

avg(a.price)
176.34999999999997

Started streaming 1 records after 37 ms and completed after 37 ms.

10.

MATCH (a: Listings)--(b: Hosts)

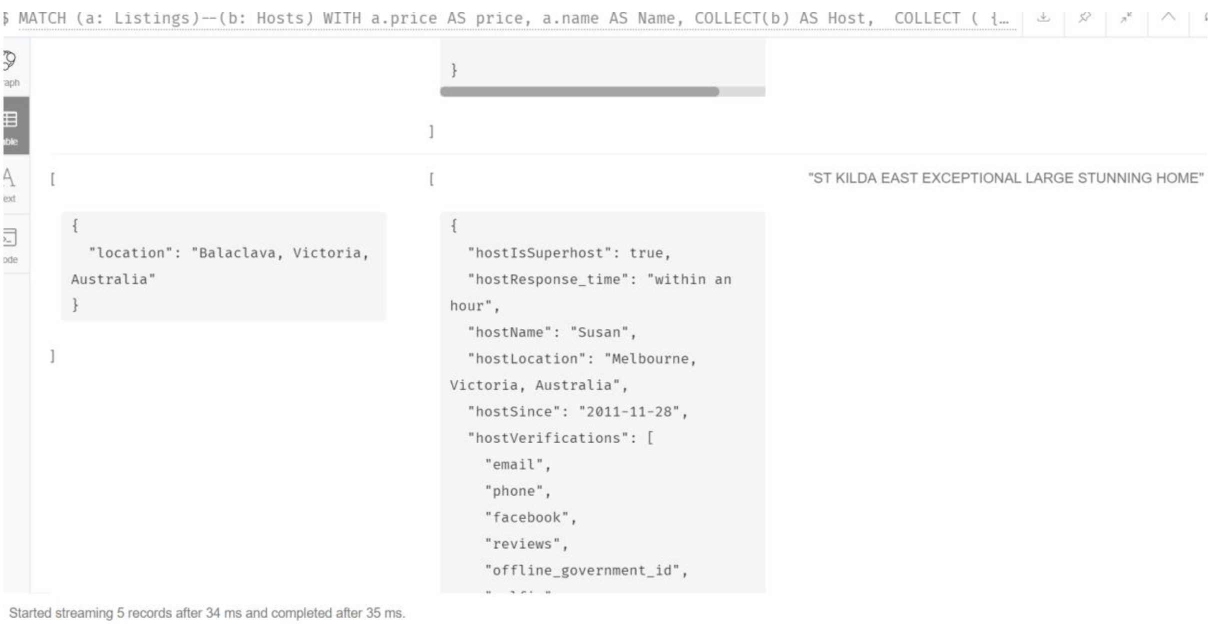
WITH a.price AS price, a.name AS Name, COLLECT(b) AS Host,

COLLECT ({location: a.street}) AS Location

RETURN Location, Host, Name

ORDER BY price DESC

LIMIT 5



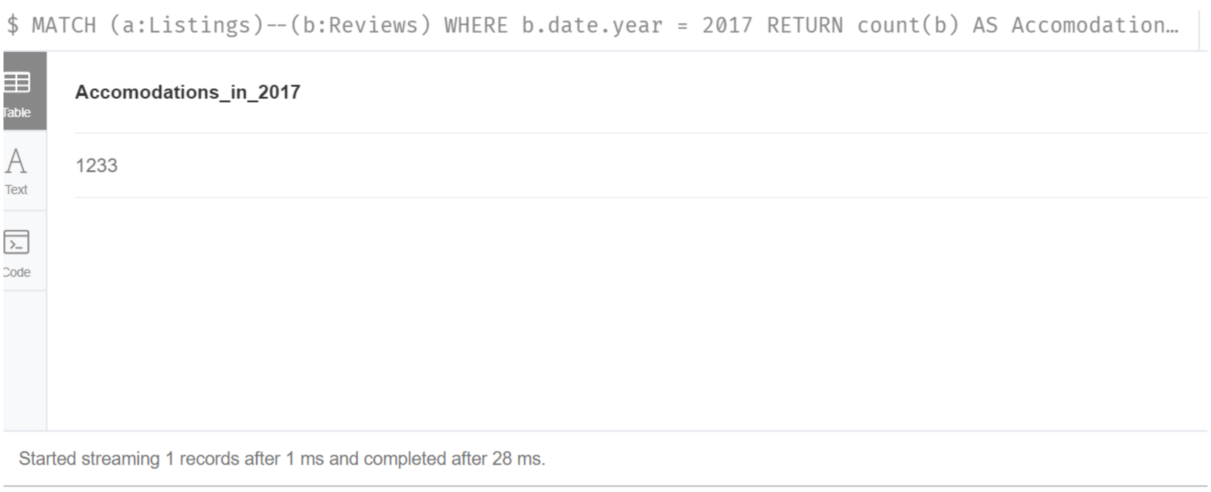
```
$ MATCH (a: Listings)--(b: Hosts) WITH a.price AS price, a.name AS Name, COLLECT(b) AS Host,  COLLECT ( {…
```

```
}
]
[                                   [                                   "ST KILDA EAST EXCEPTIONAL LARGE STUNNING HOME"
  {                                   {
    "location": "Balaclava, Victoria,   "hostIsSuperhost": true,
  Australia"                            "hostResponse_time": "within an
  }                                   hour",
]                                       "hostName": "Susan",
                                        "hostLocation": "Melbourne,
                                      Victoria, Australia",
                                        "hostSince": "2011-11-28",
                                        "hostVerifications": [
                                          "email",
                                          "phone",
                                          "facebook",
                                          "reviews",
                                          "offline_government_id",
```

Started streaming 5 records after 34 ms and completed after 35 ms.

11.

MATCH (a:Listings)--(b:Reviews) WHERE b.date.year = 2017 RETURN count(b) AS Accomodations_in_2017



```
$ MATCH (a:Listings)--(b:Reviews) WHERE b.date.year = 2017 RETURN count(b) AS Accomodation…
```

Accomodations_in_2017
1233

Started streaming 1 records after 1 ms and completed after 28 ms.

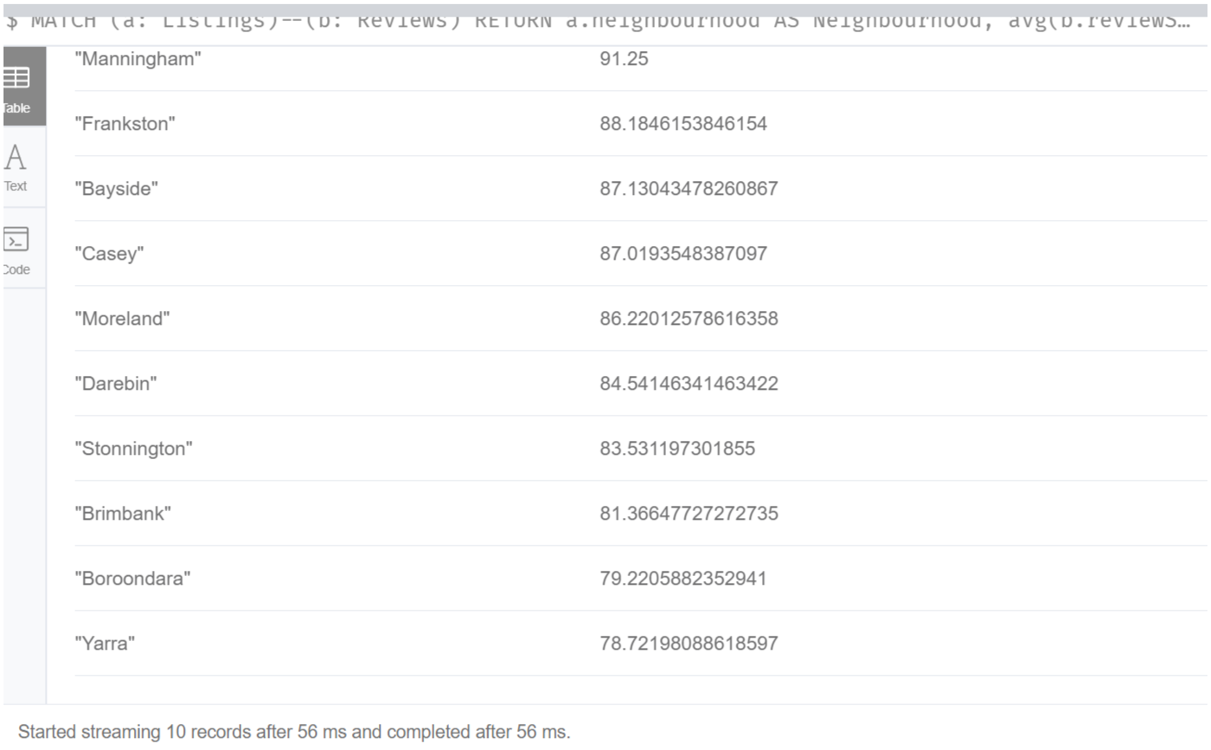
12.

MATCH (a: Listings)--(b: Reviews)

RETURN a.neighbourhood AS Neighbourhood, avg(b.reviewScoresRating) AS AverageReviewScoresRating

ORDER BY AverageReviewScoresRating DESC

LIMIT 10



```
$ MATCH (a: Listings)--(b: Reviews) RETURN a.neighbourhood AS Neighbourhood, avg(b.reviewS…
```

"Manningham"	91.25
"Frankston"	88.1846153846154
"Bayside"	87.13043478260867
"Casey"	87.0193548387097
"Moreland"	86.22012578616358
"Darebin"	84.54146341463422
"Stonnington"	83.531197301855
"Brimbank"	81.36647727272735
"Boroondara"	79.2205882352941
"Yarra"	78.72198088618597

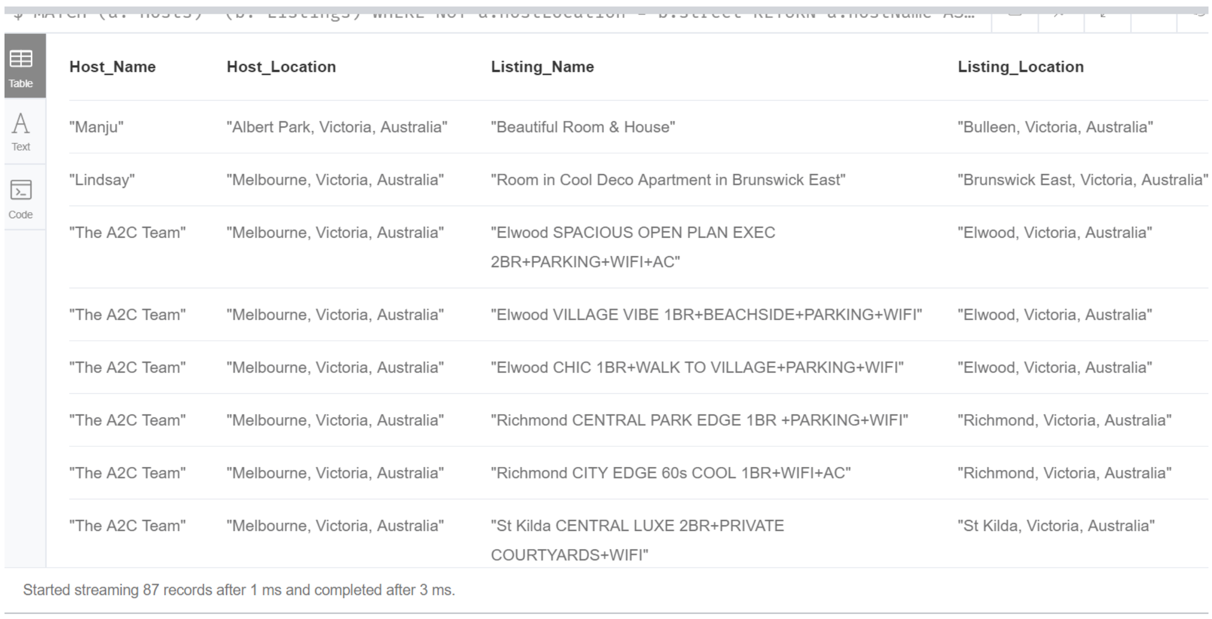
Started streaming 10 records after 56 ms and completed after 56 ms.

13.

MATCH (a: Hosts)--(b: Listings)

WHERE NOT a.hostLocation = b.street

RETURN a.hostName AS Host_Name, a.hostLocation AS Host_Location, b.name AS Listing_Name, b.street AS Listing_Location



```
$ MATCH (a: Hosts) (b: Listings) WHERE NOT a.hostLocation = b.street RETURN a.hostName AS…
```

Host_Name	Host_Location	Listing_Name	Listing_Location
"Manju"	"Albert Park, Victoria, Australia"	"Beautiful Room & House"	"Bulleen, Victoria, Australia"
"Lindsay"	"Melbourne, Victoria, Australia"	"Room in Cool Deco Apartment in Brunswick East"	"Brunswick East, Victoria, Australia"
"The A2C Team"	"Melbourne, Victoria, Australia"	"Elwood SPACIOUS OPEN PLAN EXEC 2BR+PARKING+WIFI+AC"	"Elwood, Victoria, Australia"
"The A2C Team"	"Melbourne, Victoria, Australia"	"Elwood VILLAGE VIBE 1BR+BEACHSIDE+PARKING+WIFI"	"Elwood, Victoria, Australia"
"The A2C Team"	"Melbourne, Victoria, Australia"	"Elwood CHIC 1BR+WALK TO VILLAGE+PARKING+WIFI"	"Elwood, Victoria, Australia"
"The A2C Team"	"Melbourne, Victoria, Australia"	"Richmond CENTRAL PARK EDGE 1BR +PARKING+WIFI"	"Richmond, Victoria, Australia"
"The A2C Team"	"Melbourne, Victoria, Australia"	"Richmond CITY EDGE 60s COOL 1BR+WIFI+AC"	"Richmond, Victoria, Australia"
"The A2C Team"	"Melbourne, Victoria, Australia"	"St Kilda CENTRAL LUXE 2BR+PRIVATE COURTYARDS+WIFI"	"St Kilda, Victoria, Australia"

Started streaming 87 records after 1 ms and completed after 3 ms.

14.

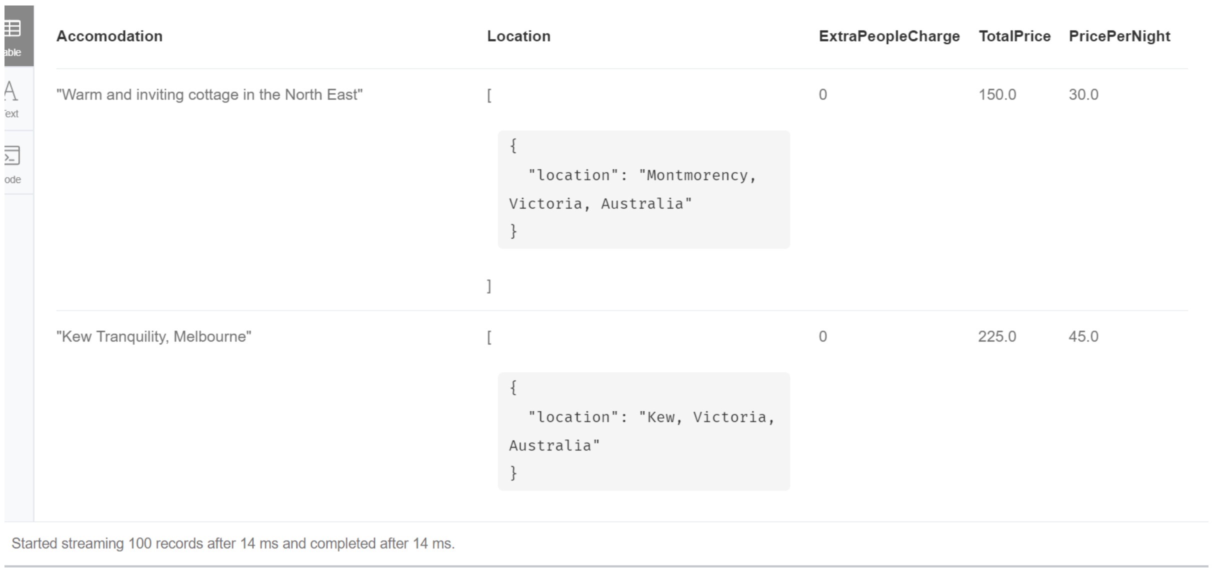
MATCH (a: Listings)

WITH a.name AS Accomodation, collect ({location: a.street}) AS Location,

a.price AS pricePerNight, a.extraPeople AS extraPeople

RETURN Accomodation, Location, extraPeople AS ExtraPeopleCharge, 5 * (pricePerNight + (2 * extraPeople)) AS TotalPrice, pricePerNight AS PricePerNight

ORDER BY TotalPrice



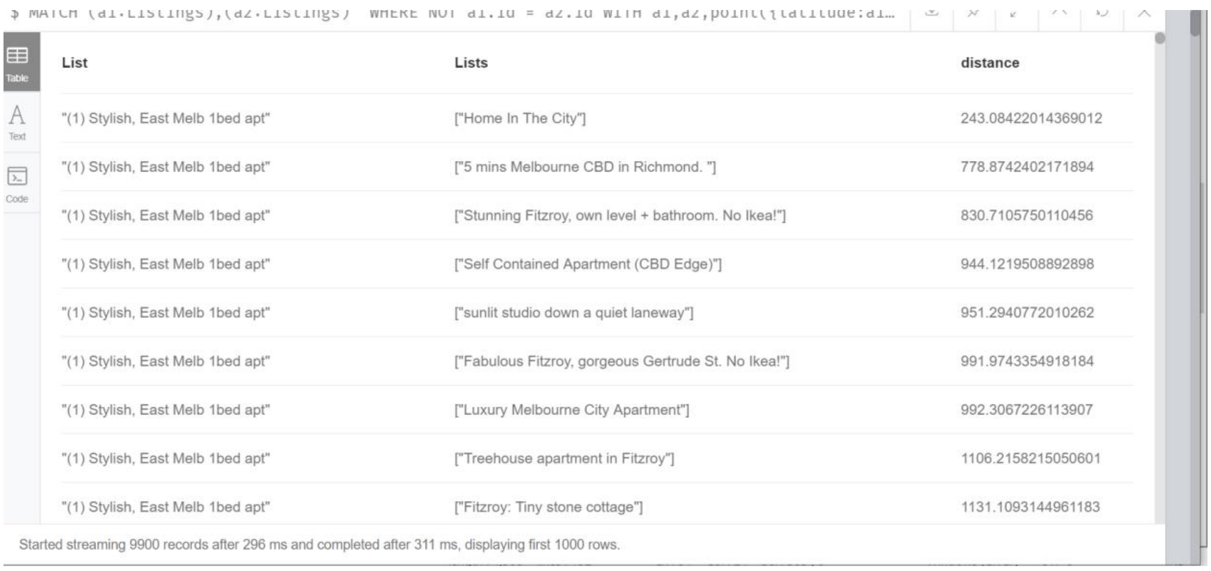
Accomodation	Location	ExtraPeopleCharge	TotalPrice	PricePerNight
"Warm and inviting cottage in the North East"	[{ "location": "Montmorency, Victoria, Australia" }]	0	150.0	30.0
"Kew Tranquility, Melbourne"	[{ "location": "Kew, Victoria, Australia" }	0	225.0	45.0

Started streaming 100 records after 14 ms and completed after 14 ms.

15.

MATCH (a1:Listings),(a2:Listings)

```
WHERE NOT a1.id = a2.id

WITH a1,a2,point({latitude:a1.latitude,longitude:a1.longitude}) AS p1,

point({latitude:a2.latitude,longitude:a2.longitude}) AS p2, COLLECT(a2.name) AS Lists

RETURN  a1.name AS List,Lists,distance(p1,p2) AS distance

ORDER BY List,distance
```



```
$ MATCH (a1:Listings),(a2:Listings) WHERE NOT a1.id = a2.id WITH a1,a2,point({latitude:a1…
```

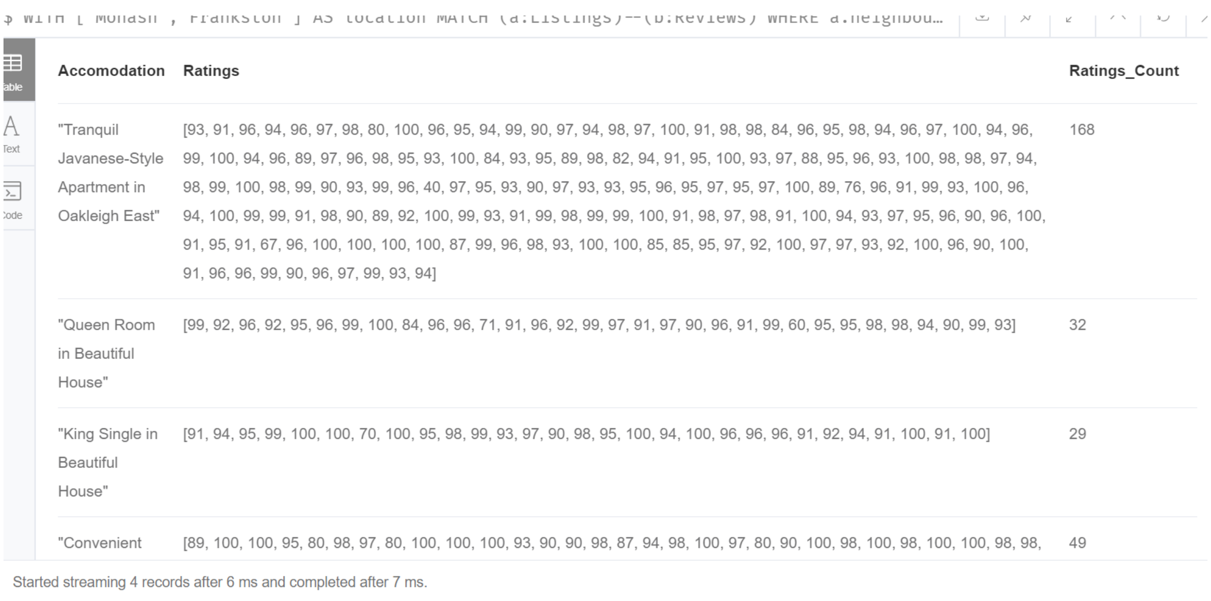
List	Lists	distance
"(1) Stylish, East Melb 1bed apt"	["Home In The City"]	243.08422014369012
"(1) Stylish, East Melb 1bed apt"	["5 mins Melbourne CBD in Richmond. "]	778.8742402171894
"(1) Stylish, East Melb 1bed apt"	["Stunning Fitzroy, own level + bathroom. No Ikea!"]	830.7105750110456
"(1) Stylish, East Melb 1bed apt"	["Self Contained Apartment (CBD Edge)"]	944.1219508892898
"(1) Stylish, East Melb 1bed apt"	["sunlit studio down a quiet laneway"]	951.2940772010262
"(1) Stylish, East Melb 1bed apt"	["Fabulous Fitzroy, gorgeous Gertrude St. No Ikea!"]	991.9743354918184
"(1) Stylish, East Melb 1bed apt"	["Luxury Melbourne City Apartment"]	992.3067226113907
"(1) Stylish, East Melb 1bed apt"	["Treehouse apartment in Fitzroy"]	1106.2158215050601
"(1) Stylish, East Melb 1bed apt"	["Fitzroy: Tiny stone cottage"]	1131.1093144961183

Started streaming 9900 records after 296 ms and completed after 311 ms, displaying first 1000 rows.

EXTRA 5 QUERIES

1. Display the Accommodations with all ratings which are not 0, and the count of ratings for all the listings which are in the neighbourhood Monash or Frankston.

```
WITH ['Monash','Frankston'] AS location

MATCH (a:Listings)--(b:Reviews)

WHERE a.neighbourhood in location AND b.reviewScoresRating >0

RETURN a.name AS Accomodation,collect(b.reviewScoresRating) AS
Ratings,count(b.reviewScoresRating) AS Ratings_Count
```



```
$ WITH [ Monash , Frankston ] AS location MATCH (a:Listings)--(b:Reviews) WHERE a.neighbou…
```

Accomodation	Ratings	Ratings_Count
"Tranquil Javanese-Style Apartment in Oakleigh East"	[93, 91, 96, 94, 96, 97, 98, 80, 100, 96, 95, 94, 99, 90, 97, 94, 98, 97, 100, 91, 98, 98, 84, 96, 95, 98, 94, 96, 97, 100, 94, 96, 99, 100, 94, 96, 89, 97, 96, 98, 95, 93, 100, 84, 93, 95, 89, 98, 82, 94, 91, 95, 100, 93, 97, 88, 95, 96, 93, 100, 98, 98, 97, 94, 98, 99, 100, 98, 99, 90, 93, 99, 96, 40, 97, 95, 93, 90, 97, 93, 93, 95, 96, 95, 97, 95, 97, 100, 89, 76, 96, 91, 99, 93, 100, 96, 94, 100, 99, 99, 91, 98, 90, 89, 92, 100, 99, 93, 91, 99, 98, 99, 99, 100, 91, 98, 97, 98, 91, 100, 94, 93, 97, 95, 96, 90, 96, 100, 91, 95, 91, 67, 96, 100, 100, 100, 100, 87, 99, 96, 98, 93, 100, 100, 85, 85, 95, 97, 92, 100, 97, 97, 93, 92, 100, 96, 90, 100, 91, 96, 96, 99, 90, 96, 97, 99, 93, 94]	168
"Queen Room in Beautiful House"	[99, 92, 96, 92, 95, 96, 99, 100, 84, 96, 96, 71, 91, 96, 92, 99, 97, 91, 97, 90, 96, 91, 99, 60, 95, 95, 98, 98, 94, 90, 99, 93]	32
"King Single in Beautiful House"	[91, 94, 95, 99, 100, 100, 70, 100, 95, 98, 99, 93, 97, 90, 98, 95, 100, 94, 100, 96, 96, 96, 91, 92, 94, 91, 100, 91, 100]	29
"Convenient	[89, 100, 100, 95, 80, 98, 97, 80, 100, 100, 100, 93, 90, 90, 98, 87, 94, 98, 100, 97, 80, 90, 100, 98, 100, 98, 100, 100, 98, 98,	49

Started streaming 4 records after 6 ms and completed after 7 ms.

2. Display the hosts who are superhosts, completed more than 5 years and also have submitted government ids for verification.

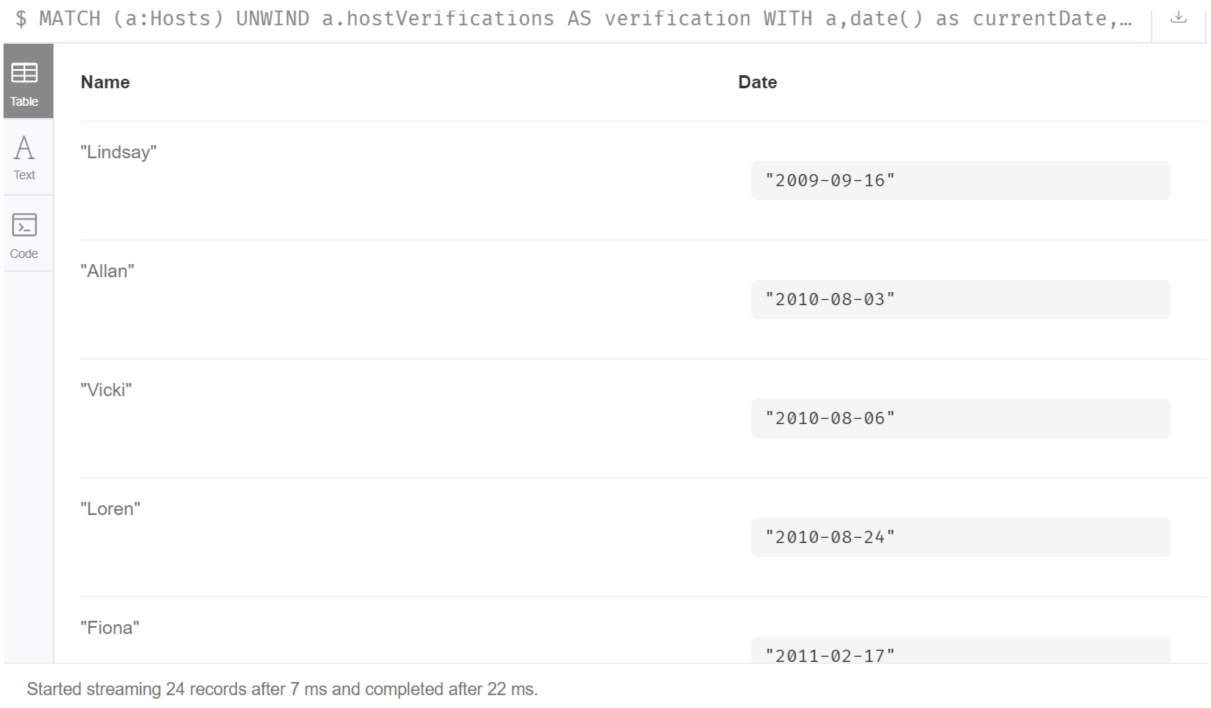
MATCH (a:Hosts)

UNWIND a.hostVerifications AS verification

WITH a,date() as currentDate,a.hostSince.year AS hostYear

WHERE a.hostIsSuperhost = true AND verification IN ["government_id"] AND (currentDate.year - hostYear) > 5

RETURN a.hostName AS Name, a.hostSince AS Date



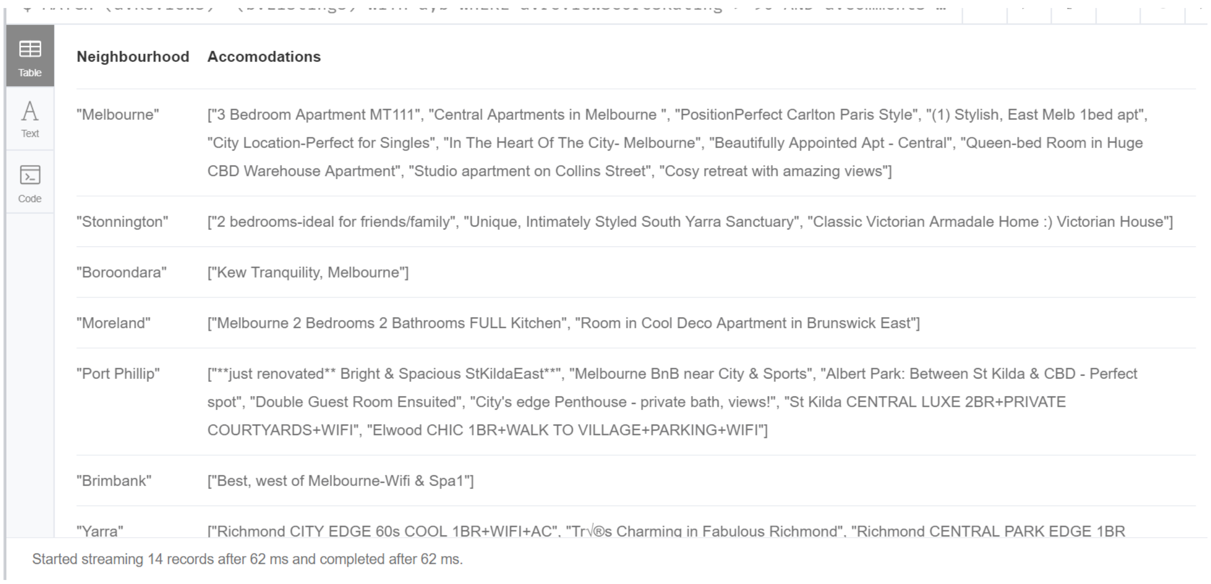
```
$ MATCH (a:Hosts) UNWIND a.hostVerifications AS verification WITH a,date() as currentDate,…
```

Name	Date
"Lindsay"	"2009-09-16"
"Allan"	"2010-08-03"
"Vicki"	"2010-08-06"
"Loren"	"2010-08-24"
"Fiona"	"2011-02-17"

Started streaming 24 records after 7 ms and completed after 22 ms.

3. Display the unique accomodations according to the review ratings greater than 90 and having "Good" in its comments

```
MATCH (a:Reviews)--(b:Listings)

WITH a,b

WHERE a.reviewScoresRating > 90 AND a.comments CONTAINS ("Good")

RETURN b.neighbourhood AS Neighbourhood,collect(DISTINCT(b.name)) AS Accommodations
```

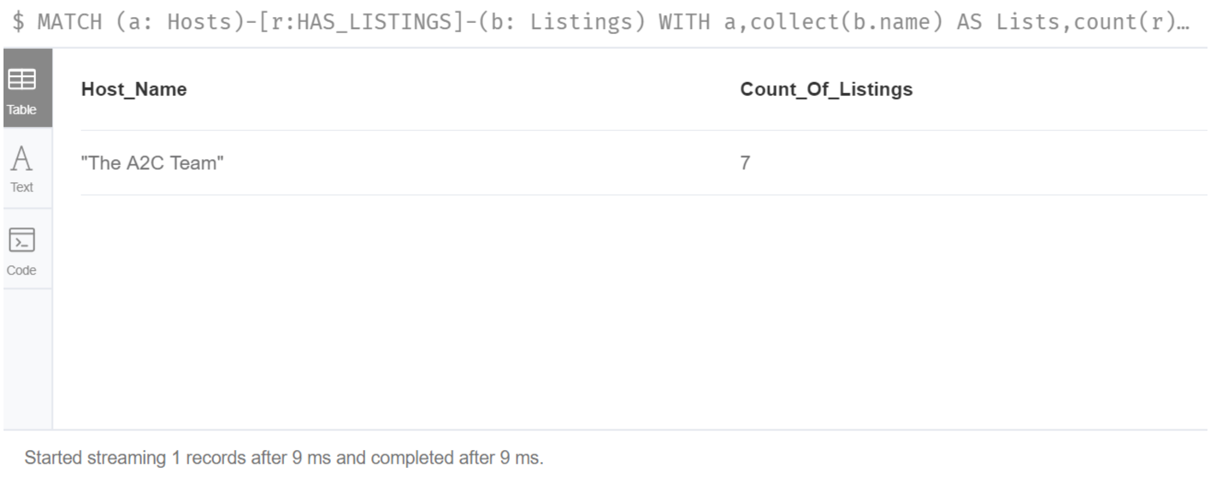


Neighbourhood	Accomodations
"Melbourne"	["3 Bedroom Apartment MT111", "Central Apartments in Melbourne ", "PositionPerfect Carlton Paris Style", "(1) Stylish, East Melb 1bed apt", "City Location-Perfect for Singles", "In The Heart Of The City- Melbourne", "Beautifully Appointed Apt - Central", "Queen-bed Room in Huge CBD Warehouse Apartment", "Studio apartment on Collins Street", "Cosy retreat with amazing views"]
"Stonnington"	["2 bedrooms-ideal for friends/family", "Unique, Intimately Styled South Yarra Sanctuary", "Classic Victorian Armadale Home :) Victorian House"]
"Boroondara"	["Kew Tranquility, Melbourne"]
"Moreland"	["Melbourne 2 Bedrooms 2 Bathrooms FULL Kitchen", "Room in Cool Deco Apartment in Brunswick East"]
"Port Phillip"	["**just renovated** Bright & Spacious StKildaEast**", "Melbourne BnB near City & Sports", "Albert Park: Between St Kilda & CBD - Perfect spot", "Double Guest Room Ensuited", "City's edge Penthouse - private bath, views!", "St Kilda CENTRAL LUXE 2BR+PRIVATE COURTYARDS+WIFI", "Elwood CHIC 1BR+WALK TO VILLAGE+PARKING+WIFI"]
"Brimbank"	["Best, west of Melbourne-Wifi & Spa1"]
"Yarra"	["Richmond CITY EDGE 60s COOL 1BR+WIFI+AC", "Tr√®s Charming in Fabulous Richmond", "Richmond CENTRAL PARK EDGE 1BR

Started streaming 14 records after 62 ms and completed after 62 ms.

4. Display the host and his number of listings having the maximum number of listings.

```
MATCH (a: Hosts)-[r:HAS_LISTINGS]-(b: Listings)

WITH a,collect(b.name) AS Lists,count(r) AS Count_Of_Listings

RETURN a.hostName AS Host_Name,Count_Of_Listings

ORDER BY Count_Of_Listings DESC

LIMIT 1
```



```
$ MATCH (a: Hosts)-[r:HAS_LISTINGS]-(b: Listings) WITH a,collect(b.name) AS Lists,count(r)…
```

Host_Name	Count_Of_Listings
"The A2C Team"	7

Started streaming 1 records after 9 ms and completed after 9 ms.

5. Display the Price with their listings according to the room types who have the price below 100 and minimum stay for 1 night.

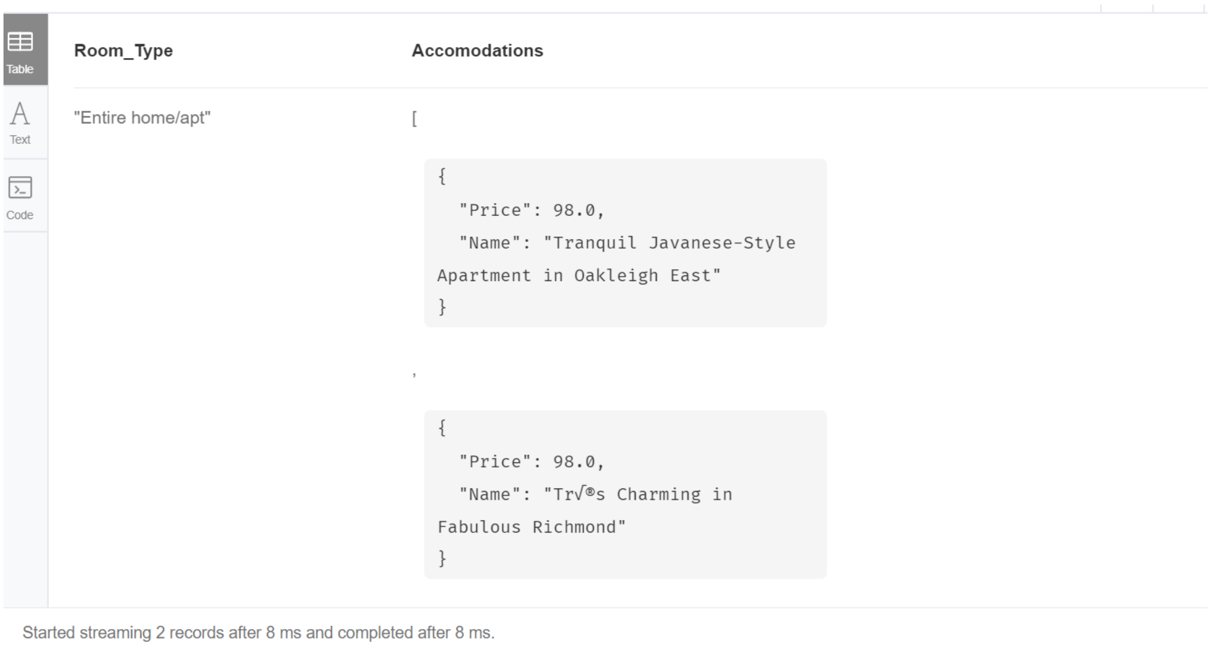
MATCH (a:Listings)

WITH a, a.price AS m

WHERE m< 100 AND a.minimumNights >1

RETURN a.roomType AS Room_Type ,collect({Name:a.name,Price:m}) AS Accomodations

ORDER BY Room_Type



Room_Type	Accomodations
"Entire home/apt"	[{ "Price": 98.0, "Name": "Tranquil Javanese-Style Apartment in Oakleigh East" } , { "Price": 98.0, "Name": "Tr√®s Charming in Fabulous Richmond" }

Started streaming 2 records after 8 ms and completed after 8 ms.

INDICES

CREATE INDEX ON: Reviews(reviewScoresRating)

CREATE INDEX ON: Hosts(hostIsSuperhost)

CREATE INDEX ON: Listings(street,price)

REASONING:

Indices are created on properties which are used frequently to make their operations efficient. Hence, reviewScoresRating in Reviews, hostIsSuperhost in Hosts and street,price in Listings are already used frequently and also are the key properties by which the data can be retrieved in future by MonashBnB being a travel accommodations booking company as ratings,hosts and search by street(location) and price for an accommodation are basic criteria for query processing .

C.3. Database Modifications

1.

```
//1a

MERGE (h: Hosts { hostId: 1900400})

ON CREATE SET

h.hostUrl = "https://www.airbnb.com.au/users/show/7211951",

h.hostName = "Mellisa",

h.hostVerifications = ["government_id", "selfie", "email", "phone"],

h.hostLocation = "Melbourne, Victoria, Australia",

h.hostResponseTime = "within an hour",

h.hostIsSuperhost = true,

h.hostSince = Date("2013-10-02")

RETURN h


MATCH (h: Hosts {hostId: 1900400})

MERGE (l: Listings {listingId: 400500})

ON CREATE SET

l.name = "Luxury Master Room w Stunning River View - MODERN",

l.summary	= "We are offering a master room (with private bathroom) in a modern 3-bedroom
room apartment in Southbank. Jane as host lives here. It is the best place in Melbourne to live
behind the beautiful yarra river with the French window in your private space.",

l.listingUrl	=
"https://www.airbnb.com.au/rooms/9516067?source_impression_id=p3_1571820797_wdj%2FkaEL
%2FpPRr40A",

l.pictureUrl =
"https://www.airbnb.com.au/rooms/9516067/slideshow/434651659?adults=1&children=0&infants
=0",

l.neighbourhood = "Southbank",

l.street = "Melbourne, Victoria, Australia",

l.zipcode = 3000,

l.latitude = -37.8290,

l.longitude = 144.9570,

l.roomType = "Private room",
l.amenities        = ["Air conditioning", "Dryer", "Play TV", "Essentials"],
l.extraPeople = 25,
l.price = 59,
l.minimumNights = 1,
l.calculatedHostListingsCount = 3,
l.availability365 = 230
MERGE(h)-[:HAS_LISTINGS]->(l)

MATCH (l:Listings {listingId: 400500})
MERGE (r: Reviews {reviewId: 300500})
ON CREATE SET
r.reviewerId = 1000500,
r.date = Date("2019-10-22"),
r.reviewerName = "Garry",
r.reviewScoresRating = 96,
r.comments = "Great location, view, master bedroom, robe, ensuite and hospitality. Looking forward
to staying there again sometime."
MERGE (r)-[:HAS_REVIEWED]->(l)
return r, l

MATCH (l:Listings {listingId: 400500})
MERGE (r: Reviews {reviewId: 300501})
ON CREATE SET
r.reviewerId = 1000545,
r.date = Date("2019-09-20"),
r.reviewerName = "Laurie",
r.reviewScoresRating = 85,
r.comments = "A truly top-notch Airbnb host and superb facilities, King Size bed, massive ensuite,
central to Southbank, Crown Casino and CBD just a 5-minute walk or get the tram just a 100 metres
away. What more could you ask for Highly recommended "
MERGE (r)-[:HAS_REVIEWED]->(l)
return r, l
```

```
//1b


MERGE (h: Hosts { hostId: 1900401})
ON CREATE SET
h.hostUrl = "https://www.airbnb.com.au/users/show/80621455",
h.hostName = "Bill Donghang",
h.hostVerifications = ["government_id", "selfie", "email", "phone"],
h.hostLocation = "Melbourne, Victoria, Australia",
h.hostResponseTime = "within a day",
h.hostIsSuperhost = false,
h.hostSince = Date("2016-11-03")
RETURN h


MATCH (h: Hosts {hostId: 1900401})
MERGE (l: Listings {listingId: 400501})
ON CREATE SET
l.name = "Holiday Apartment One Bedroom, Free Gym & Pool",
l.summary          = "1 bedroom apartment is only 3 minutes’ walk from Southern Cross Station. There
are free tram stops nearby as well as supermarket. Guests have access to an indoor heated pool, a
fitness centre and unlimited free wifi. The whole apartment internal build-in 45 sqm plus 5sqm
balconies. - In total 50sqm for this spacious apartment.",
l.listingUrl          =
"https://www.airbnb.com.au/rooms/15537910?source_impression_id=p3_1571825185_xFHMAK%2
BiJh%2FssQsn",
l.pictureUrl =
"https://www.airbnb.com.au/rooms/15537910/slideshow/222379656?adults=1&children=0&infant
s=0",
l.neighbourhood = "Southern Cross Station",
l.street = "Melbourne, Victoria, Australia",
l.zipcode = 3000,
l.latitude = -37.8184,
l.longitude = 144.9525,
l.roomType = "Entire home",
l.amenities         = ["Wi-Fi", "Kitchen", "Gym", "Pool"],
l.extraPeople = 0,
l.price = 81,
l.minimumNights = 2,
l.calculatedHostListingsCount = 32,
l.availability365 = 215
MERGE(h)-[:HAS_LISTINGS]->(l)

MATCH (l:Listings {listingId: 400501})
MERGE (r: Reviews {reviewId: 300503})
ON CREATE SET
r.reviewerId = 1000504,
r.date = Date("2017-10-01"),
r.reviewerName = "Wassaporn",
r.reviewScoresRating = 75,
r.comments = "Bill's place is great. Clean and modern. Good location."
MERGE (r)-[:HAS_REVIEWED]->(l)
return r, l

MATCH (l:Listings {listingId: 400501})
MERGE (r: Reviews {reviewId: 300504})
ON CREATE SET
r.reviewerId = 1000540,
r.date = Date("2019-06-20"),
r.reviewerName = "David",
r.reviewScoresRating = 55,
r.comments = "Great apartment and it's close to southern Cross and everything else."
MERGE (r)-[:HAS_REVIEWED]->(l)
return r, l
```

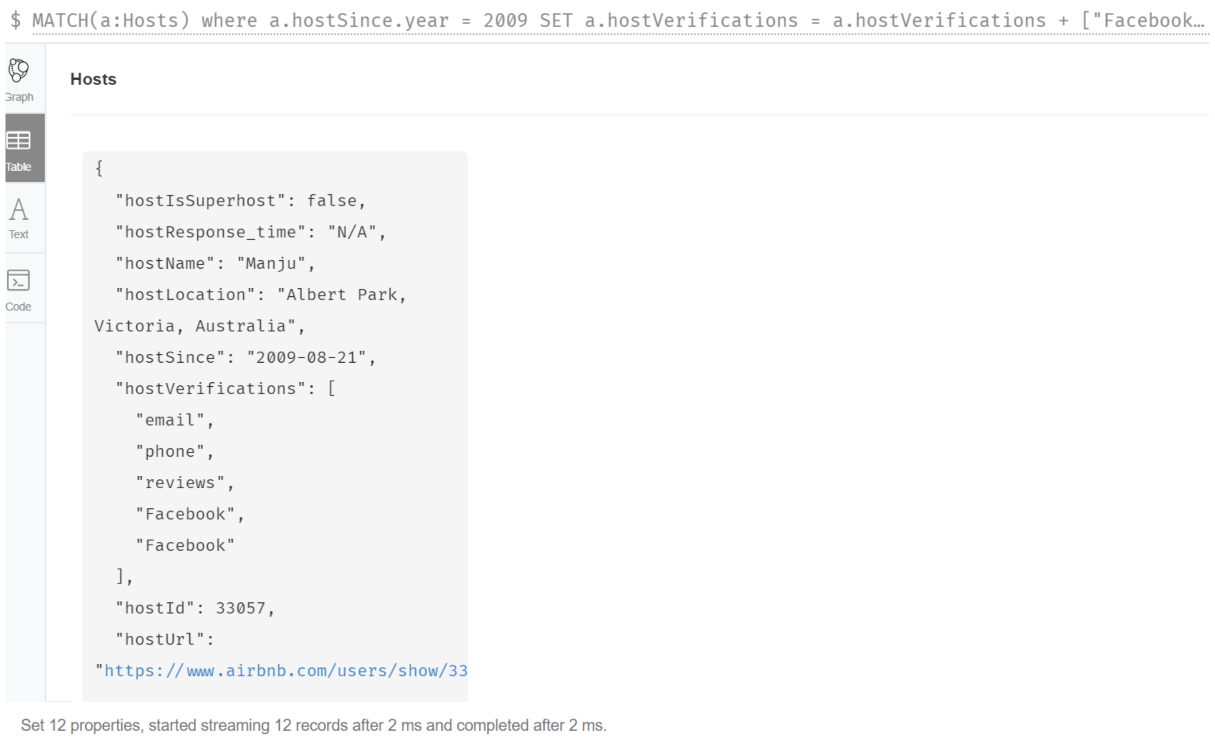
```
//1c

MERGE (h: Hosts { hostId: 1900402})
ON CREATE SET
h.hostUrl = "https://www.airbnb.com.au/users/show/80621455",
h.hostName = "Hadi",
h.hostVerifications = ["government_id", "email", "phone"],
h.hostLocation = "Melbourne, Victoria, Australia",
h.hostResponseTime = "within a day",
h.hostIsSuperhost = false,
h.hostSince = Date("2016-01-03")
RETURN h

MATCH (h: Hosts {hostId: 1900402})
MERGE (l: Listings {listingId: 400502})
ON CREATE SET
l.name = "Cityview Master Bedroom in The Green Abode",
l.summary          = "This is a private room listing where your host will be staying with you. The unit
has 2 bedrooms (a guest bedroom and a host bedroom), a joined kitchen space + living room +
dining, one bathroom, and a balcony with an awesome view of the city. Due to the unit facing East, it
receives plenty of morning sunlight, especially during summer. This space is complete with
entertainment features such as Netflix and unlimited NBN Wifi.",
l.listingUrl         =
"https://www.airbnb.com.au/rooms/17465305?source_impression_id=p3_1571826079_0UD71yEW
W7ftC0cq",
l.pictureUrl =
"https://www.airbnb.com.au/rooms/15537910/slideshow/222379656?adults=1&children=0&infant
s=0",
l.neighbourhood = "CBD",
l.street = "Melbourne, Victoria, Australia",
l.zipcode = 3000,
l.latitude = -37.8136,
l.longitude = 144.9631,
l.roomType = "Private room",
l.amenities         = ["Wi-Fi", "Kitchen", "Gym", "Lift"],
l.extraPeople = 2,
l.price = 50,
l.minimumNights = 2,
l.calculatedHostListingsCount = 1,
l.availability365 = 60
MERGE(h)-[:HAS_LISTINGS]->(l)

MATCH (l:Listings {listingId: 400502})
MERGE (r: Reviews {reviewId: 300507})
ON CREATE SET
r.reviewerId = 1000509,
r.date = Date("2019-07-01"),
r.reviewerName = "Joel",
r.reviewScoresRating = 90,
r.comments = "Very clean place and supe close to everything. So much great food and bars around."
MERGE (r)-[:HAS_REVIEWED]->(l)
return r, l

MATCH (l:Listings {listingId: 400502})
MERGE (r: Reviews {reviewId: 300508})
ON CREATE SET
r.reviewerId = 1000550,
r.date = Date("2019-06-01"),
r.reviewerName = "Elliott",
r.reviewScoresRating = 76,
r.comments = "Had a really nice stay at Hadi’s place, great location, sparkling cleaning. Also Hadi is
really helpful."
MERGE (r)-[:HAS_REVIEWED]->(l)
return r, l
```

2.

MATCH(a:Hosts) where a.hostSince.year = 2009

SET a.hostVerifications = a.hostVerifications + ["Facebook"]

return a AS Hosts



```
$ MATCH(a:Hosts) where a.hostSince.year = 2009 SET a.hostVerifications = a.hostVerifications + ["Facebook…
```

Hosts

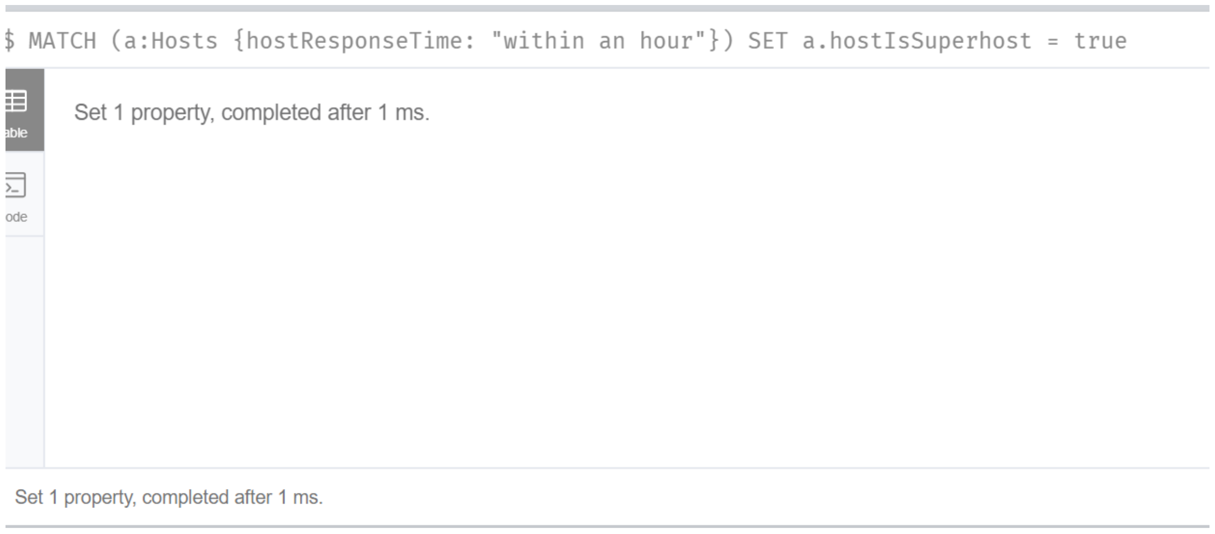
```
{
  "hostIsSuperhost": false,
  "hostResponse_time": "N/A",
  "hostName": "Manju",
  "hostLocation": "Albert Park,
Victoria, Australia",
  "hostSince": "2009-08-21",
  "hostVerifications": [
    "email",
    "phone",
    "reviews",
    "Facebook",
    "Facebook"
  ],
  "hostId": 33057,
  "hostUrl":
"https://www.airbnb.com/users/show/33
```

Set 12 properties, started streaming 12 records after 2 ms and completed after 2 ms.

3.

MATCH (a:Hosts {hostResponseTime: "within an hour"})

SET a.hostIsSuperhost = true



```
$ MATCH (a:Hosts {hostResponseTime: "within an hour"}) SET a.hostIsSuperhost = true
```

Set 1 property, completed after 1 ms.

Set 1 property, completed after 1 ms.

4.

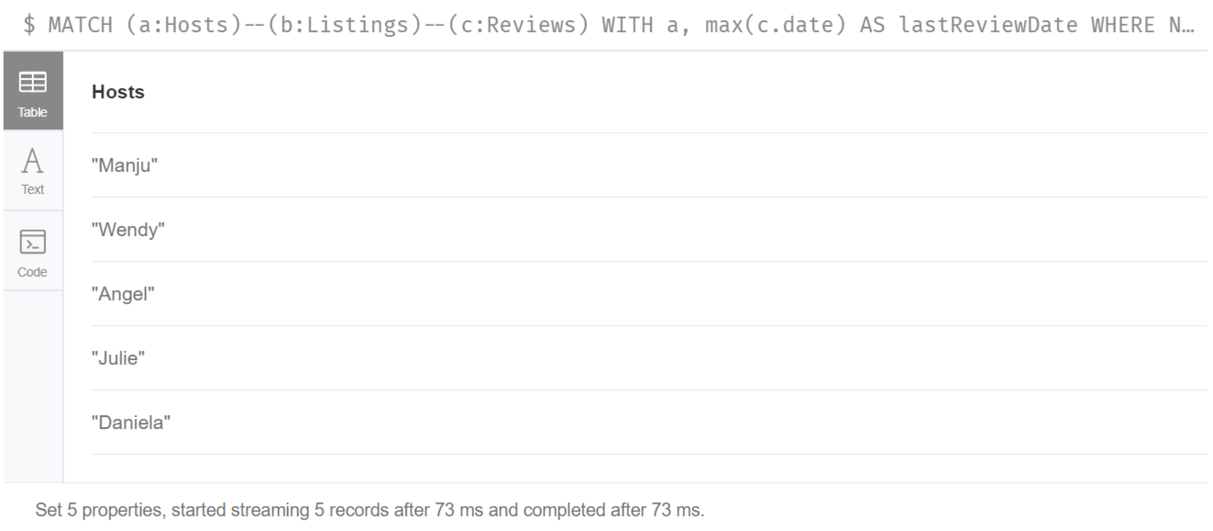
MATCH (a:Hosts)--(b:Listings)--(c:Reviews)

WITH a, max(c.date) AS lastReviewDate

WHERE NOT lastReviewDate.year > 2016

SET a.active = false

RETURN a.hostName AS Hosts



```
$ MATCH (a:Hosts)--(b:Listings)--(c:Reviews) WITH a, max(c.date) AS lastReviewDate WHERE N…
```

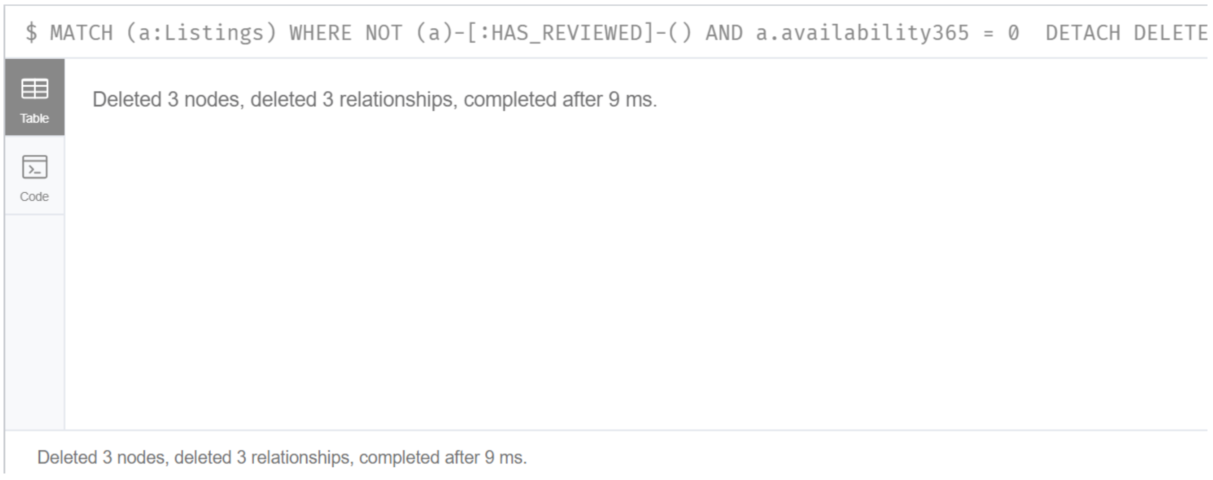
```
Hosts
"Manju"
"Wendy"
"Angel"
"Julie"
"Daniela"
```

Set 5 properties, started streaming 5 records after 73 ms and completed after 73 ms.

5.

MATCH (a:Listings) WHERE NOT (a)-[:HAS_REVIEWED]-() AND a.availability365 = 0

DETACH DELETE a



```
$ MATCH (a:Listings) WHERE NOT (a)-[:HAS_REVIEWED]-() AND a.availability365 = 0  DETACH DELETE
```

Deleted 3 nodes, deleted 3 relationships, completed after 9 ms.

Deleted 3 nodes, deleted 3 relationships, completed after 9 ms.

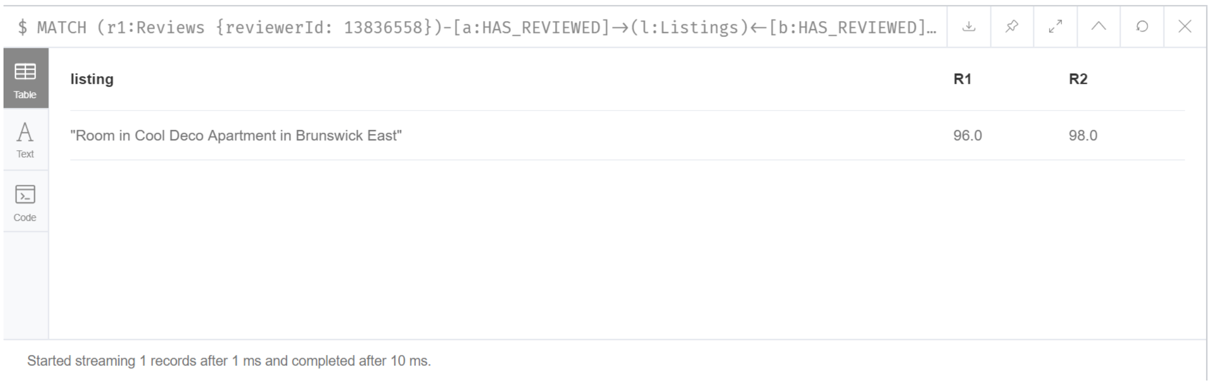
C.4. Advanced Topic.

(Option 1)

1. Match all the common listings between two reviewers.

MATCH (r1:Reviews {reviewerId: 13836558})-[a:HAS_REVIEWED]->(l:Listings)<-[b:HAS_REVIEWED]-(r2:Reviews {reviewerId:71026585 })

RETURN l.name AS listing, a.reviewScoresRating AS R1, b.reviewScoresRating AS R2



```
$ MATCH (r1:Reviews {reviewerId: 13836558})-[a:HAS_REVIEWED]→(l:Listings)←[b:HAS_REVIEWED]…
```

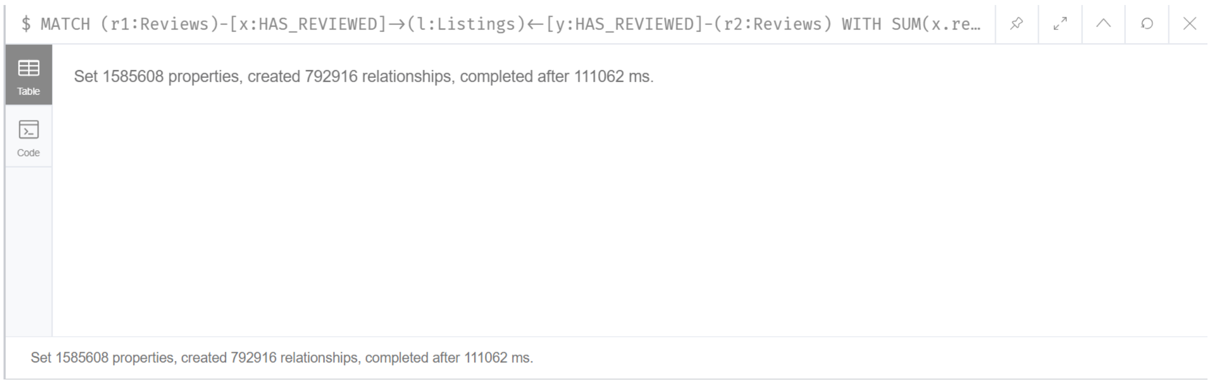
listing	R1	R2
"Room in Cool Deco Apartment in Brunswick East"	96.0	98.0

Started streaming 1 records after 1 ms and completed after 10 ms.

2. Create a relation “SIMILARITY” between Reviews and assign it the cosine formula as follows.

MATCH (r1:Reviews)-[x:HAS_REVIEWED]->(l:Listings)<-[y:HAS_REVIEWED]-(r2:Reviews)

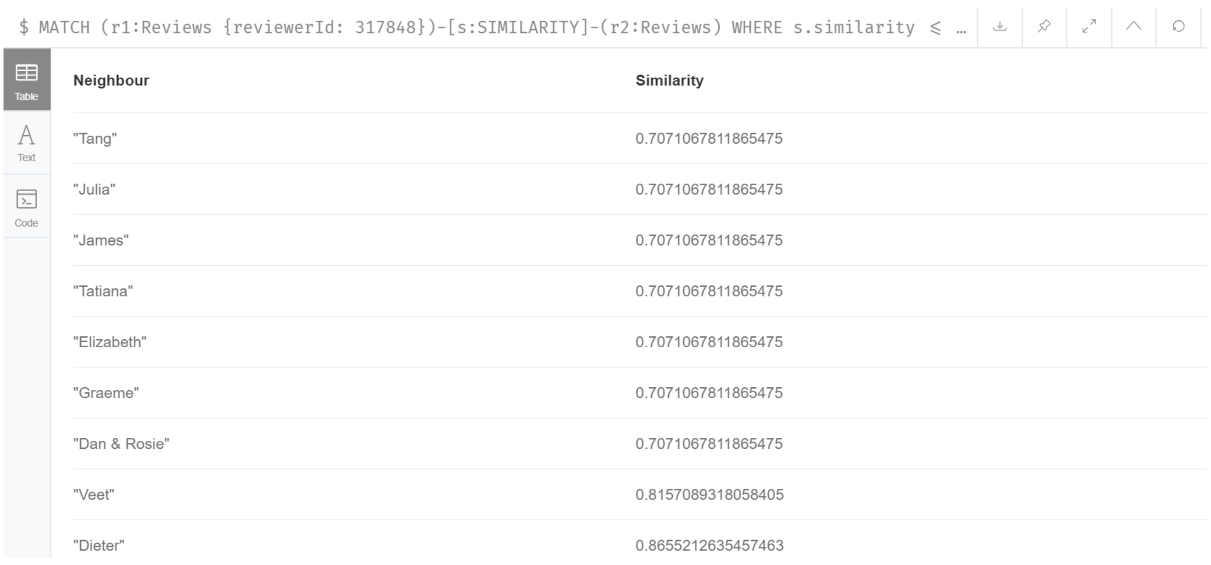
```
WITH  SUM(x.reviewScoresRating * y.reviewScoresRating) AS xyDotProduct,
SQRT(REDUCE(xDot = 0.0, a IN COLLECT(x.reviewScoresRating) | xDot + a^2)) AS xLength,
SQRT(REDUCE(yDot = 0.0, b IN COLLECT(y.reviewScoresRating) | yDot + b^2)) AS yLength, r1, r2
MERGE (r1)-[s:SIMILARITY]-(r2)
SET   s.similarity = xyDotProduct / (xLength * yLength)
```



```
$ MATCH (r1:Reviews)-[x:HAS_REVIEWED]→(l:Listings)←[y:HAS_REVIEWED]-(r2:Reviews) WITH SUM(x.re…
Set 1585608 properties, created 792916 relationships, completed after 111062 ms.
Set 1585608 properties, created 792916 relationships, completed after 111062 ms.
```

3.Display the neighbours (Reviewers) according to the similarity.

```
MATCH    (r1:Reviews {reviewerId: 317848})-[s:SIMILARITY]-(r2:Reviews)
WHERE    s.similarity <= 1
WITH     r2, s.similarity AS s
RETURN   r2.reviewerName AS Neighbour, s AS Similarity
ORDER BY Similarity
```



```
$ MATCH (r1:Reviews {reviewerId: 317848})-[s:SIMILARITY]-(r2:Reviews) WHERE s.similarity ≤ …
```

Neighbour	Similarity
"Tang"	0.7071067811865475
"Julia"	0.7071067811865475
"James"	0.7071067811865475
"Tatiana"	0.7071067811865475
"Elizabeth"	0.7071067811865475
"Graeme"	0.7071067811865475
"Dan & Rosie"	0.7071067811865475
"Veet"	0.8157089318058405
"Dieter"	0.8655212635457463

4. Query for displaying the recommendations based on similarity and non-booked accommodation.

MATCH (r1:Reviews)-[x:HAS_REVIEWED]->(l:Listings), (r1)-[s:SIMILARITY]-(r2:Reviews {reviewerId: 317848})

WHERE NOT((r2)-[:HAS_REVIEWED]->(l))

WITH l, s.similarity AS similarity, x.reviewScoresRating AS rating

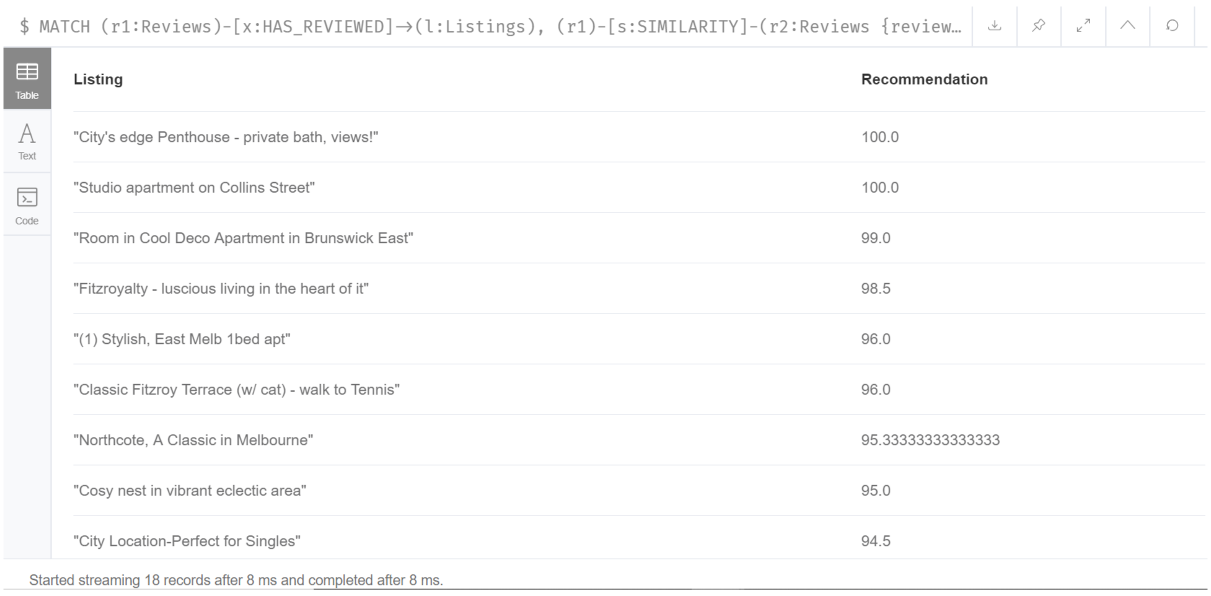
ORDER BY l.name, similarity

WITH l.name AS listing, COLLECT(rating)[0..3] AS ratings

WITH listing, REDUCE(s = 0, i IN ratings | s + i) * 1.0 / SIZE(ratings) AS reco

ORDER BY reco DESC

RETURN listing AS Listing, reco AS Recommendation



```
$ MATCH (r1:Reviews)-[x:HAS_REVIEWED]→(l:Listings), (r1)-[s:SIMILARITY]-(r2:Reviews {review…
```

Listing	Recommendation
"City's edge Penthouse - private bath, views!"	100.0
"Studio apartment on Collins Street"	100.0
"Room in Cool Deco Apartment in Brunswick East"	99.0
"Fitzroyalty - luscious living in the heart of it"	98.5
"(1) Stylish, East Melb 1bed apt"	96.0
"Classic Fitzroy Terrace (w/ cat) - walk to Tennis"	96.0
"Northcote, A Classic in Melbourne"	95.33333333333333
"Cosy nest in vibrant eclectic area"	95.0
"City Location-Perfect for Singles"	94.5

Started streaming 18 records after 8 ms and completed after 8 ms.