

- MATCH (n) RETURN (n);

```
MATCH (r:Retailer) RETURN (r);
```

```
neo4j$ MATCH (r:Retailer) RETURN (r);
```

r
(:Retailer {zip: "60610",city: "Chicago",street: "240 North Clark",name: "Target",state: "IL"})
(:Retailer {zip: "60612",city: "Chicago",street: "388 East Jackson",name: "Gap",state: "IL"})
(:Retailer {zip: "60617",city: "Chicago",street: "1285 West Harison",name: "Walmart",state: "IL"})
(:Retailer {zip: "60615",city: "Chicago",street: "287 Oakly",name: "BestBuy",state: "IL"})
(:Retailer {zip: "60615",city: "Chicago",street: "10 East Erie",name: "Express",state: "IL"})
(:Retailer {zip: "60616",city: "Chicago",street: "45 North LaSalle",name: "Nordstrom",state: "IL"})
(:Retailer {zip: "60620",city: "Chicago",street: "310 East Monroe",name: "Kohls",state: "IL"})
(:Retailer {zip: "60606",city: "Chicago",street: "610 South LaSalle",name: "Neiman Marcus",state: "IL"})
(:Retailer {zip: "60612",city: "Chicago",street: "20 West Randolph St",name: "Foot Locker",state: "IL"})
(:Retailer {zip: "60610",city: "Chicago",street: "510 Waker",name: "Coach",state: "IL"})

3. Execute the following Cypher code to the get the list of employees: **(0 point)**

MATCH (e:Employee) RETURN (e);

neo4j\$ MATCH (e:Employee) RETURN (e);

Graph	e
Table	(:Employee {phone: "312-566-3234",name: "Seth Snow",age: "21"})
Text	(:Employee {phone: "312-426-3234",name: "Donny Pollard",age: "32"})
Code	(:Employee {phone: "708-322-8712",name: "Roxie Aguilar",age: "21"})
	(:Employee {phone: "630-452-8756",name: "Sonny Horn",age: "27"})
	(:Employee {phone: "312-314-7751",name: "Isaac Mendez",age: "19"})
	(:Employee {phone: "708-622-8145",name: "Ricky Bond",age: "35"})
	(:Employee {phone: "708-348-3592",name: "Don Howe",age: "41"})
	(:Employee {phone: "312-983-3452",name: "Carmen Dixon",age: "24"})
	(:Employee {phone: "312-237-9872",name: "Steven Carney",age: "19"})
	(:Employee {phone: "312-862-9866",name: "Andrea Gallegos",age: "23"})
	(:Employee {phone: "630-876-5429",name: "Bryon Ramos",age: "27"})
	(:Employee {phone: "312-790-3651",name: "Malik Copeland",age: "38"})
	(:Employee {phone: "708-298-3441",name: "Letha Hardy",age: "44"})
	(:Employee {phone: "312-439-3282",name: "Denver Glover",age: "27"})
	(:Employee {phone: "312-348-7258",name: "Deidre Duke",age: "37"})
	(:Employee {phone: "708-827-7761",name: "Lina Rivers",age: "24"})
	(:Employee {phone: "630-628-2771",name: "Kraig Hensley",age: "47"})
	(:Employee {phone: "312-318-3951",name: "Jerold Mccarthy",age: "49"})
	(:Employee {phone: "708-827-2945",name: "Kelvin Haney",age: "25"})
	(:Employee {phone: "708-748-5792",name: "Irvin Clayton",age: "32"})

4. Execute the following Cypher code to the get the list of customers: **(0 point)**

MATCH (c:Customer) RETURN (c);

```
neo4j$ MATCH (c:Customer) RETURN (c);
```



c
(:Customer {zip: "60615",credit_card: "5500 1000 0000 1004",city: "Chicago",street: "10 West 93rd st",name: "Edgar Haroop",state: "IL"})
(:Customer {zip: "60616",credit_card: "5500 2000 0000 7005",city: "Chicago",street: "4471 Tator Patch Road",name: " Alex Buel ",state: "IL"})
(:Customer {zip: "60610",credit_card: "5500 3000 0000 3006",city: "Chicago",street: "1137 Nash Street",name: " Torri Pettway ",state: "IL"})
(:Customer {zip: "60609",credit_card: "5500 4000 0000 4907",city: "Chicago",street: "694 Point Street",name: " Rigoberto Kinchen ",state: "IL"})
(:Customer {zip: "60605",credit_card: "5500 5000 0000 2808",city: "Chicago",street: "4725 John Calvin Drive",name: " Cary Moenaney ",state: "IL"})
(:Customer {zip: "60618",credit_card: "5500 6000 0000 3109",city: "Chicago",street: "3681 Oakmound Road",name: " Nicola Castanon ",state: "IL"})
(:Customer {zip: "60620",credit_card: "5500 7000 0000 0105",city: "Chicago",street: "2795 Vesta Drive",name: " Marceline Westfield ",state: "IL"})
(:Customer {zip: "60601",credit_card: "5500 8000 0000 5905",city: "Chicago",street: "1274 Poplar Street",name: " Zenaida Sitzes ",state: "IL"})
(:Customer {zip: "60603",credit_card: "5500 9000 0000 8705",city: "Chicago",street: "3781 Oakmound Road",name: " Aura Schiel ",state: "IL"})
(:Customer {zip: "60624",credit_card: "5500 0100 0000 0025",city: "Chicago",street: "4355 Virginia Street",name: " Corinne Suman ",state: "IL"})
(:Customer {zip: "60640",credit_card: "5500 0100 0000 8805",city: "Chicago",street: "1114 Oakmound Drive",name: "Jonathan Rinka",state: "IL"})
(:Customer {zip: "60643",credit_card: "5500 0200 0000 4905",city: "Chicago",street: "3830 Cherry Camp Road",name: "Richard Smith",state: "IL"})
(:Customer {zip: "60615",credit_card: "5500 0300 0000 1805",city: "Chicago",street: "234 East 55th st",name: "Sarah Radovic",state: "IL"})
(:Customer {zip: "60610",credit_card: "5500 0400 0000 4205",city: "Chicago",street: "520 North wells st",name: "Lucy Scheller",state: "IL"})
(:Customer {zip: "60610",credit_card: "5500 0400 0000 2976",city: "Chicago",street: "320 Virginia Street",name: "Carol Rose",state: "IL"})
(:Customer {zip: "60610",credit_card: "5500 0400 0000 6721",city: "Chicago",street: "180 Vesta Drive",name: "Lacy Grant",state: "IL"})
(:Customer {zip: "60610",credit_card: "5500 0400 0000 8923",city: "Chicago",street: "360 Taylor st",name: "Ashlee Reid",state: "IL"})
(:Customer {zip: "60610",credit_card: "5500 0400 0000 7823",city: "Chicago",street: "430 Cherry Camp Road",name: "Sallie Bauer",state: "IL"})
(:Customer {zip: "60610",credit_card: "5500 0400 0000 8938",city: "Chicago",street: "180 Nash Street",name: "Wade Boyer",state: "IL"})
(:Customer {zip: "60610",credit_card: "5500 0400 0000 6182",city: "Chicago",street: "301 Oakmound Road",name: "Larry Grahamr",state: "IL"})

5. Execute the following Cypher code to get the list of all disputed transactions: **(0 point)**

MATCH (customer:Customer)-[transaction:SHOPPED_AT]->(retailer) WHERE transaction.status = "Disputed"

RETURN customer.name AS `Customer Name`, retailer.name AS `Retailer Name`, transaction.amount AS `Transaction Amount`, transaction.date AS `Transaction date` ORDER BY `Transaction date` DESC

```
1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]->(retailer) WHERE transaction.status = "Disputed"
2 RETURN customer.name AS `Customer Name`, retailer.name AS `Retailer Name`, transaction.amount AS `Transaction Amount`, transaction.date AS `Transaction date`
3 ORDER BY `Transaction date` DESC
```



Customer Name	Retailer Name	Transaction Amount	Transaction date
" Nicola Castanon "	"Coach"	"721"	"7/17/2020"
" Zenaida Sitzes "	"Express"	"1884"	"5/7/2020"
" Marceline Westfield "	"Express"	"533"	"5/6/2020"
"Edgar Haroop"	"Neiman Marcus"	"1732"	"5/26/2020"
"Edgar Haroop"	"Kohls"	"1021"	"5/23/2020"
"Lucy Scheller"	"BestBuy"	"424"	"5/20/2020"
"Larry Grahamr"	"Walmart"	"425"	"5/19/2020"
"Larry Grahamr"	"Neiman Marcus"	"475"	"5/19/2020"
"Richard Smith"	"Kohls"	"875"	"5/13/2020"
" Rigoberto Kinchen "	"BestBuy"	"424"	"5/10/2020"
"Jonathan Rinka"	"Neiman Marcus"	"375"	"4/19/2020"
" Torri Pettway "	"Foot Locker"	"62"	"4/17/2020"
"Carol Rose"	"Express"	"721"	"4/13/2020"
"Edgar Haroop"	"Nordstrom"	"1415"	"4/1/2020"
" Rigoberto Kinchen "	"Express"	"721"	"4/1/2020"
"Edgar Haroop"	"Walmart"	"654"	"3/20/2020"

```

1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed"
2 RETURN customer.name AS 'Customer Name', retailer.name AS 'Retailer Name', transaction.amount AS 'Transaction Amount', transaction.date AS 'Transaction date'
3 ORDER BY 'Transaction date' DESC

```

"Edgar Haroop"	"Nordstrom"	"1415"	"4/1/2020"
"Rigoberto Kinchen "	"Express"	"721"	"4/1/2020"
"Edgar Haroop"	"Walmart"	"654"	"3/20/2020"
"Rigoberto Kinchen "	"Walmart"	"914"	"3/18/2020"
"Zenaida Sitzes "	"Walmart"	"1149"	"3/18/2020"
"Richard Smith"	"Coach"	"1145"	"3/18/2020"
"Ashlee Reid"	"Walmart"	"1149"	"3/18/2020"
"Sarah Radovic"	"Nordstrom"	"516"	"3/15/2020"
"Aura Schiel "	"Neiman Marcus"	"830"	"3/13/2020"
"Cary Moenaney "	"Kohls"	"468"	"2/29/2020"
"Edgar Haroop"	"Walmart"	"1849"	"2/20/2020"
"Rigoberto Kinchen "	"Nordstrom"	"1003"	"2/20/2020"
"Corinne Suman "	"Nordstrom"	"816"	"2/20/2020"
"Lacy Grant"	"Nordstrom"	"1003"	"2/20/2020"
"Jonathan Rinka"	"Kohls"	"1345"	"2/18/2020"
"Zenaida Sitzes "	"BestBuy"	"378"	"2/10/2020"
"Sallie Bauer"	"Foot Locker"	"378"	"2/10/2020"

"Sallie Bauer"	"Foot Locker"	"378"	"2/10/2020"
"Torri Pettway "	"Target"	"605"	"1/27/2020"
"Jonathan Rinka"	"Walmart"	"945"	"1/27/2020"
"Zenaida Sitzes "	"Nordstrom"	"1790"	"1/20/2020"

6. Write the Cypher code to get the number of disputed transactions for every retailer. The output should show the Retailer name and the number of disputes. Sort with highest number of disputes on top. **(10 points)**

```

1 MATCH (:Customer)-[transaction:SHOPPED_AT]-(retailer:Retailer)
2 WHERE transaction.status = "Disputed"
3 RETURN retailer.name AS 'Retailer Name', COUNT(transaction) AS 'Number of Disputes'
4 ORDER BY 'Number of Disputes' DESC
5

```

Retailer Name	Number of Disputes
"Walmart"	7
"Nordstrom"	6
"Express"	4
"Kohls"	4
"Neiman Marcus"	4
"BestBuy"	3
"Foot Locker"	2
"Coach"	2
"Target"	1

MAX COLUMN WIDTH:

7. Write the Cypher code to get the number of disputed transactions and the list of customer names for these disputed transactions for every retailer. The output should show the Retailer and the customer name(s). You can consider using a collect() container, but it is not required. **(10 points)**

```

1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer:Retailer)
2 WHERE transaction.status = "Disputed"
3 WITH retailer, COLLECT(customer.name) AS CustomerNames, COUNT(transaction) AS DisputeCount
4 RETURN retailer.name AS Retailer, CustomerNames, DisputeCount
5 ORDER BY DisputeCount DESC
6

```

Retailer	CustomerNames	DisputeCount
"Walmart"	["Ashlee Reid", "Edgar Haroop", "Jonathan Rinka", "Edgar Haroop", "Zenaida Sitzes ", "Larry Grahamr", "Rigoberto Kinchen "]	7
"Nordstrom"	["Zenaida Sitzes ", "Rigoberto Kinchen ", "Corinne Suman ", "Edgar Haroop", "Sarah Radovic", "Lacy Grant"]	6
"Express"	["Marceline Westfield ", "Rigoberto Kinchen ", "Zenaida Sitzes ", "Carol Rose"]	4
"Kohls"	["Jonathan Rinka", "Richard Smith", "Cary Moenaney ", "Edgar Haroop"]	4
"Neiman Marcus"	["Jonathan Rinka", "Edgar Haroop", "Aura Schiel ", "Larry Grahamr"]	4
"BestBuy"	["Lucy Scheller", "Rigoberto Kinchen ", "Zenaida Sitzes "]	3
"Foot Locker"	["Torri Pettway ", "Sallie Bauer"]	2
"Coach"	["Nicola Castanon ", "Richard Smith"]	2
"Target"	["Torri Pettway "]	1

8. Write the Cypher code to get the number of disputed transactions for every customer that has more than one disputed transaction **(10 points)**

```

1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(:Retailer)
2 WHERE transaction.status = "Disputed"
3 WITH customer, COUNT(transaction) AS Disputes
4 WHERE Disputes > 1
5 RETURN customer.name AS `Customer Name`, Disputes
6 ORDER BY Disputes DESC
7

```

Customer Name	Disputes
"Edgar Haroop"	5
"Zenaida Sitzes "	4
"Rigoberto Kinchen "	4
"Jonathan Rinka"	3
"Torri Pettway "	2
"Larry Grahamr"	2
"Richard Smith"	2

9. Write the Cypher code to get the list of stores on LaSalle Street that have disputed transactions and the number of disputed transactions for every store; the store list must be sorted by store name in ascending order. **(10 points)**

```

1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(store:Retailer)
2 WHERE store.street CONTAINS 'LaSalle' AND transaction.status = 'Disputed'
3 WITH store.name AS StoreName, store.street AS Street, COUNT(transaction) AS DisputeCount
4 RETURN StoreName, Street, DisputeCount
5 ORDER BY StoreName ASC
6

```

StoreName	Street	DisputeCount
"Neiman Marcus"	"610 South LaSalle"	4
"Nordstrom"	"45 North LaSalle"	6

10. Write the Cypher code to get the list of Employees who work in at least 2 stores where disputed transactions reported in these retailers. (10 points)

```

1 MATCH (employee:Employee)-[:WORKS_AT]→(store:Retailer)←[:SHOPPED_AT {status: 'Disputed'}]-(customer:Customer)
2 WITH employee, COUNT(DISTINCT store) AS StoresWithDisputes
3 WHERE StoresWithDisputes ≥ 2
4 RETURN employee.name AS EmployeeName, StoresWithDisputes
5 ORDER BY StoresWithDisputes DESC
6

```

EmployeeName	StoresWithDisputes
"Irvin Clayton"	3
"Ricky Bond"	2
"Carmen Dixon"	2
"Bryon Ramos"	2

11. Write the Cypher code to show the total amount customers spent shopping at retailers. List the customer's name and the total amount spent. (10 points)

```

1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer:Retailer)
2 WITH customer, SUM(ToInteger(transaction.amount)) AS TotalAmountSpent
3 RETURN customer.name AS CustomerName, TotalAmountSpent
4 ORDER BY TotalAmountSpent DESC
5

```

CustomerName	TotalAmountSpent
"Edgar Haroop"	8371
" Zenaïda Sitzes "	7172
" Rigoberto Kinchen "	4937
" Corinne Suman "	3722
" Cary Moenaney "	3159
" Nicola Castanon "	2738
"Jonathan Rinka"	2665
" Marceline Westfield "	2651
" Alex Buel "	2551
"Richard Smith"	2285
" Aura Schiel "	2043
"Wade Boyer"	1884
"Ashlee Reid"	1762
"Lucy Scheller"	1272
"Larry Grahamr"	1224

"Larry Grahamr"	1224
"Lacy Grant"	1003
" Torri Pettway "	843
"Carol Rose"	721
"Sallie Bauer"	721
"Sarah Radovic"	516

12. Write the Cypher code to show the average amount spent at each Retailer. List the Retailer and the average amount spent. Sort with the highest amount on top. **(10 points)**

```

1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer:Retailer)
2 WITH retailer.name AS RetailerName, AVG(TOFLOAT(transaction.amount)) AS AvgAmountSpent
3 RETURN RetailerName, ROUND(100 * AvgAmountSpent) / 100 AS RoundedAvgAmountSpent
4 ORDER BY RoundedAvgAmountSpent DESC
5

```

RetailerName	RoundedAvgAmountSpent
"Nordstrom"	888.08
"Walmart"	801.43
"Kohls"	734.0
"Express"	713.5
"Coach"	603.5
"Neiman Marcus"	578.55
"BestBuy"	448.83
"Target"	343.67
"Gap"	308.5
"Foot Locker"	292.22