

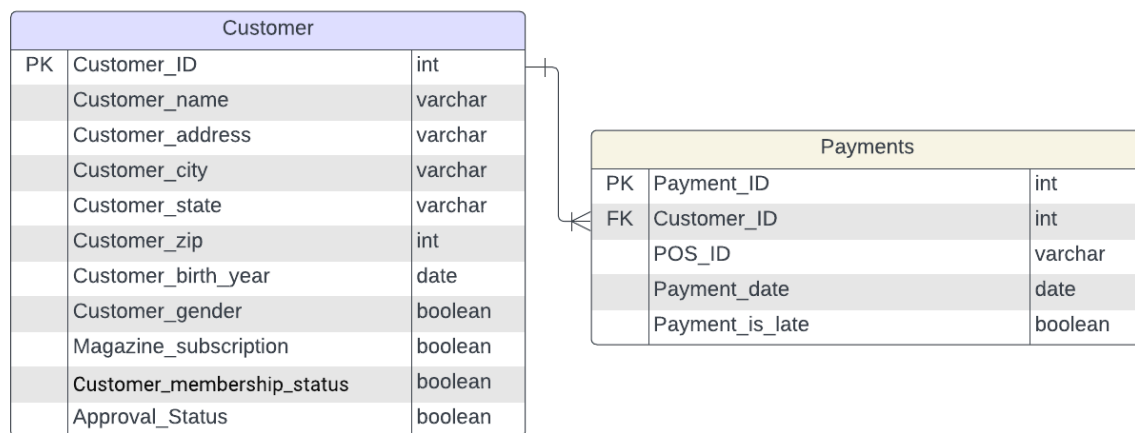
NoSQL Assignment

Farzan - 235819710

Q1. Design an ERD that represents the entities and relationships of the database that contains the data from those two files.

Answer:

The ERD diagram is as follows:



The data files primarily focus on various fields for customers as well as their payment logs. I have used the EOD to transcribe the data as well as list down their keys and types of input.

Q2. Store the records in a key-value format in **Redis**

Answer:

The commands are as follows:

```
HMSET customer:171 name "Madelyn Hensley" address "10075 Thierer Plaza" city "New York" state "New York" zip "81377" birth_year "1976" gender "M" flag1 "TRUE" flag2 "FALSE" flag3 "FALSE"
```

```
HMSET customer:172 name "Lonny Foster" address "23901 Park Meadow Dr" city "Austin" state "Texas" zip "13498" birth_year "1981" gender "F" flag1 "TRUE" flag2 "FALSE" flag3 "FALSE"
```

```
HMSET customer:173 name "Karina Livingston" address "95 Anderson Park" city "Chattanooga" state "Tennessee" zip "94518" birth_year "1977" gender "F" flag1 "TRUE" flag2 "TRUE" flag3 "TRUE"
```

```
HMSET customer:174 name "Avery McCormick" address "09992 Sunfield Parkway" city "Chicago" state "Illinois" zip "38300" birth_year "1992" gender "F" flag1 "TRUE" flag2 "FALSE" flag3 "FALSE"
```

HMSET customer:175 name "Peter King" address "0486 Dryden Road" city "Chicago" state "Illinois" zip "21012" birth_year "1991" gender "M" flag1 "TRUE" flag2 "FALSE" flag3 "TRUE"

HMSET customer:176 name "Bret Ibarra" address "14 Transport Place" city "San Diego" state "California" zip "12865" birth_year "1984" gender "M" flag1 "TRUE" flag2 "FALSE" flag3 "FALSE"

HMSET customer:177 name "Leonardo Wheeler" address "806 Corry Crossing" city "New York" state "New York" zip "44464" birth_year "1972" gender "F" flag1 "FALSE" flag2 "FALSE" flag3 "TRUE"

HMSET customer:178 name "Bennett Noble" address "8 South Terrace" city "Hixson" state "Tennessee" zip "52890" birth_year "1989" gender "F" flag1 "FALSE" flag2 "FALSE" flag3 "TRUE"

HMSET customer:179 name "Marcia Mathews" address "0380 Knutson Road" city "Dallas" state "Texas" zip "80477" birth_year "1967" gender "F" flag1 "TRUE" flag2 "FALSE" flag3 "TRUE"

HMSET customer:180 name "Avis Kramer" address "49624 Hanover Junction" city "New York" state "New York" zip "62542" birth_year "1965" gender "M" flag1 "TRUE" flag2 "FALSE" flag3 "TRUE"

HMSET customer:181 name "Lynnette Tate" address "30741 Paget Court" city "New York" state "New York" zip "34886" birth_year "1987" gender "M" flag1 "TRUE" flag2 "FALSE" flag3 "TRUE"

HMSET customer:182 name "Lakisha Estrada" address "50 Dahle Crossing" city "Dallas" state "Texas" zip "16042" birth_year "1991" gender "F" flag1 "FALSE" flag2 "TRUE" flag3 "FALSE"

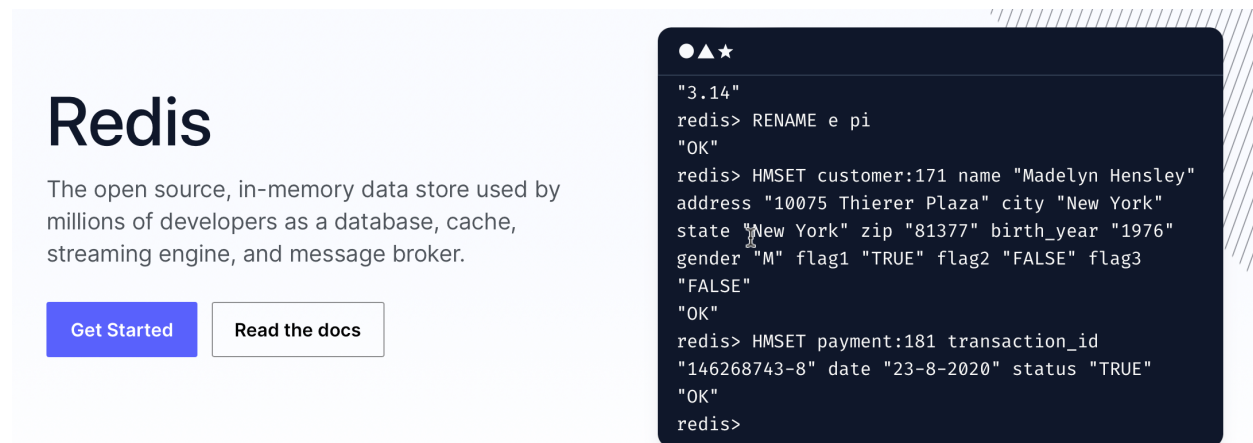
HMSET customer:183 name "Bill Silva" address "4 McBride Crossing" city "Detroit" state "Michigan" zip "15871" birth_year "1968" gender "M" flag1 "TRUE" flag2 "TRUE" flag3 "TRUE"

Set Payment Fields

HMSET payment:181 transaction_id "146268743-8" date "23-8-2020" status "TRUE"
HMSET payment:172 transaction_id "396589804-7" date "28-9-2020" status "FALSE"
HMSET payment:183 transaction_id "553753031-8" date "25-9-2020" status "FALSE"
HMSET payment:183 transaction_id "559786593-4" date "13-12-2018" status "TRUE"
HMSET payment:175 transaction_id "108659198-9" date "13-9-2016" status "TRUE"
HMSET payment:176 transaction_id "360007723-2" date "27-9-2016" status "FALSE"
HMSET payment:177 transaction_id "238309554-X" date "28-11-2014" status "FALSE"
HMSET payment:178 transaction_id "694690715-8" date "31-5-2014" status "TRUE"
HMSET payment:179 transaction_id "318241713-5" date "15-7-2016" status "TRUE"
HMSET payment:180 transaction_id "84360172-2" date "6-7-2016" status "FALSE"
HMSET payment:181 transaction_id "807633856-8" date "5-7-2018" status "FALSE"
HMSET payment:182 transaction_id "845886260-4" date "23-4-2014" status "FALSE"
HMSET payment:183 transaction_id "270161074-X" date "19-7-2016" status "TRUE"
HMSET payment:171 transaction_id "887428332-0" date "4-7-2020" status "FALSE"
HMSET payment:172 transaction_id "401129380-4" date "23-8-2015" status "TRUE"
HMSET payment:173 transaction_id "277406266-7" date "27-12-2019" status "FALSE"
HMSET payment:174 transaction_id "851112155-2" date "4-12-2019" status "TRUE"

* Note that I have used HMSET instead of HSET as discussed in class since HMSET allows me to set multiple files in a single command. Even from an efficiency perspective, HMSET set reduces the number

of trips between the client and server, which gives it an advantage over HSET in my opinion. A screenshot of the commands being executed on the online Redis server can be found below:



Q3. Write the first two records into a JSON document and store in **MongoDB**.

Answer:

To initiate working on this question, I created a new database on MongoDB Compass (farzans_database) and I formed a collection within this database (called my_collection). Each entry contains diverse customer information such as name, address, zip, etc.

The code for the first two entries in JSON is:

```
{
  "id": 171,
  "name": "Madelyn Hensley",
  "address": "10075 Thierer Plaza",
  "city": "New York",
  "state": "New York",
  "zip": 81377,
  "birth_year": 1976,
  "gender": "M",
  "flag1": true,
  "flag2": false,
  "flag3": false
}

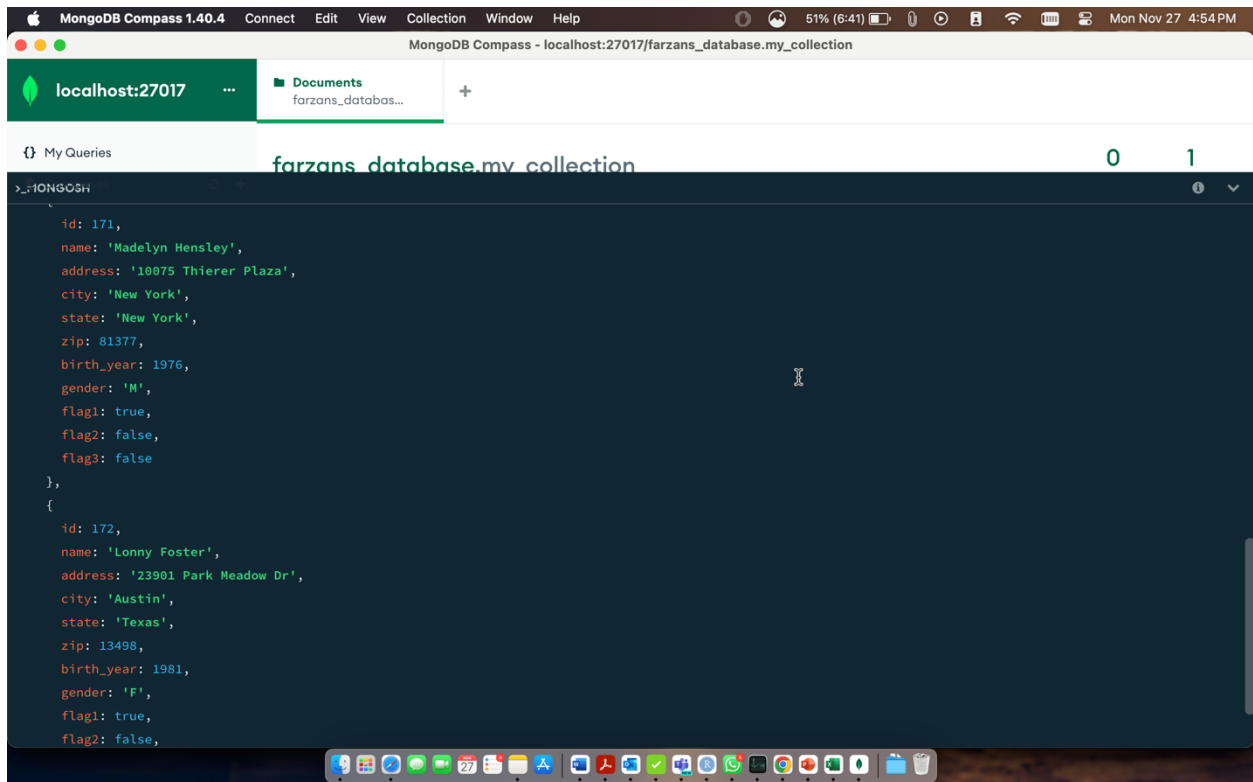
{
  "id": 172,
  "name": "Lonny Foster",
  "address": "23901 Park Meadow Dr",
```

```

"city": "Austin",
"state": "Texas",
"zip": 13498,
"birth_year": 1981,
"gender": "F",
"flag1": true,
"flag2": false,
"flag3": false
}

```

Here's a screenshot of the code:



Q4. Delete the collection(s) from point 3, rewrite them into an XML document, and store again in **MongoDB**.

Answer:

In order to tackle this question, I chose to use the “drop collection” feature in MongoDB compass since the platform does not support XML natively. I then made a new document in XML and used the following code:

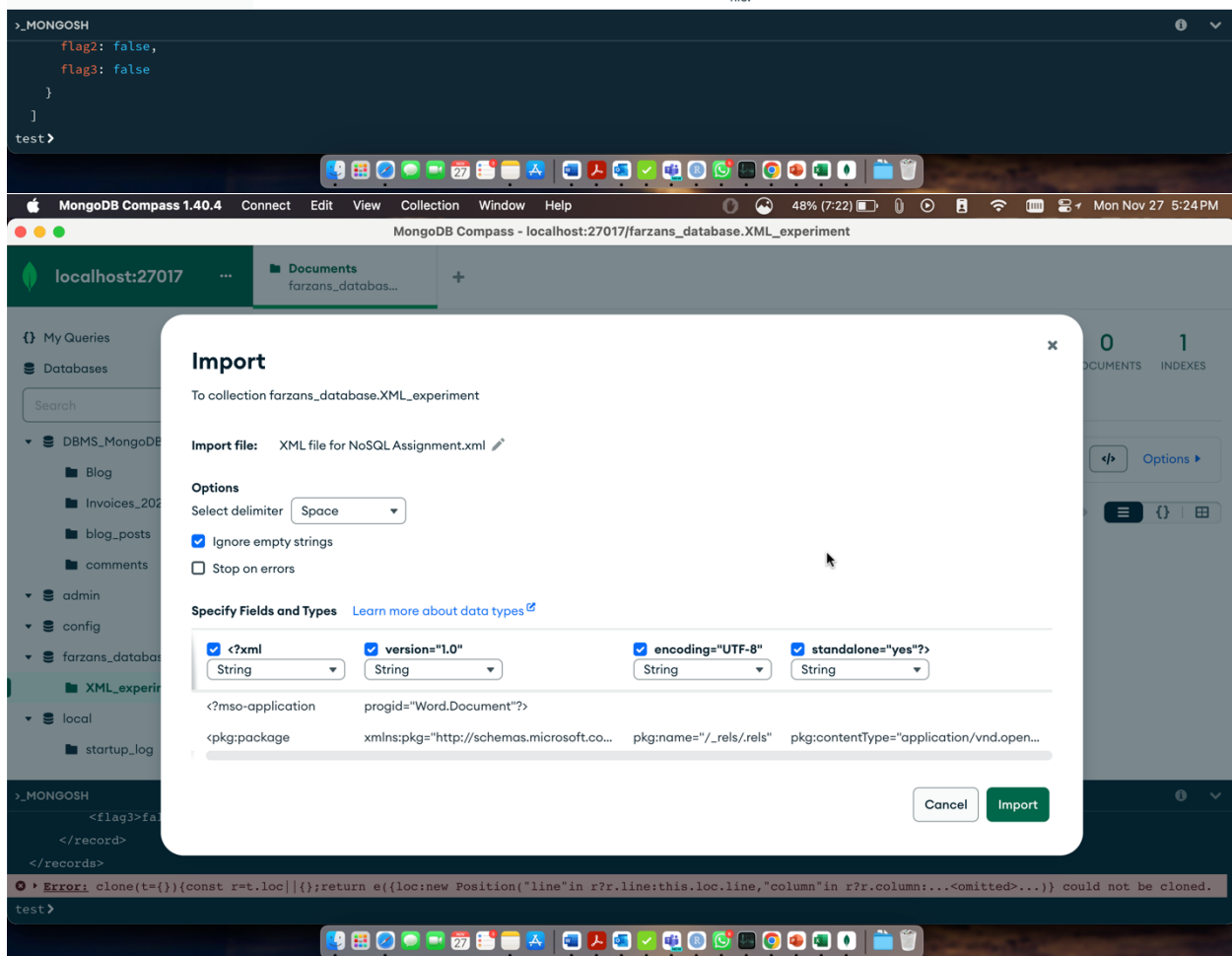
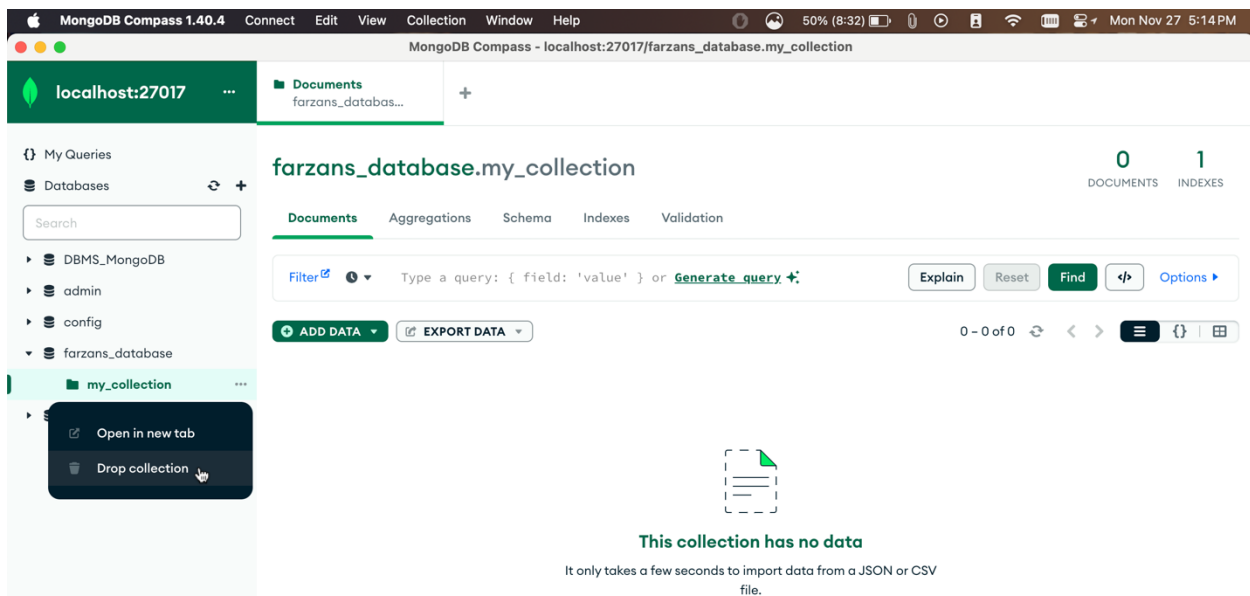
```

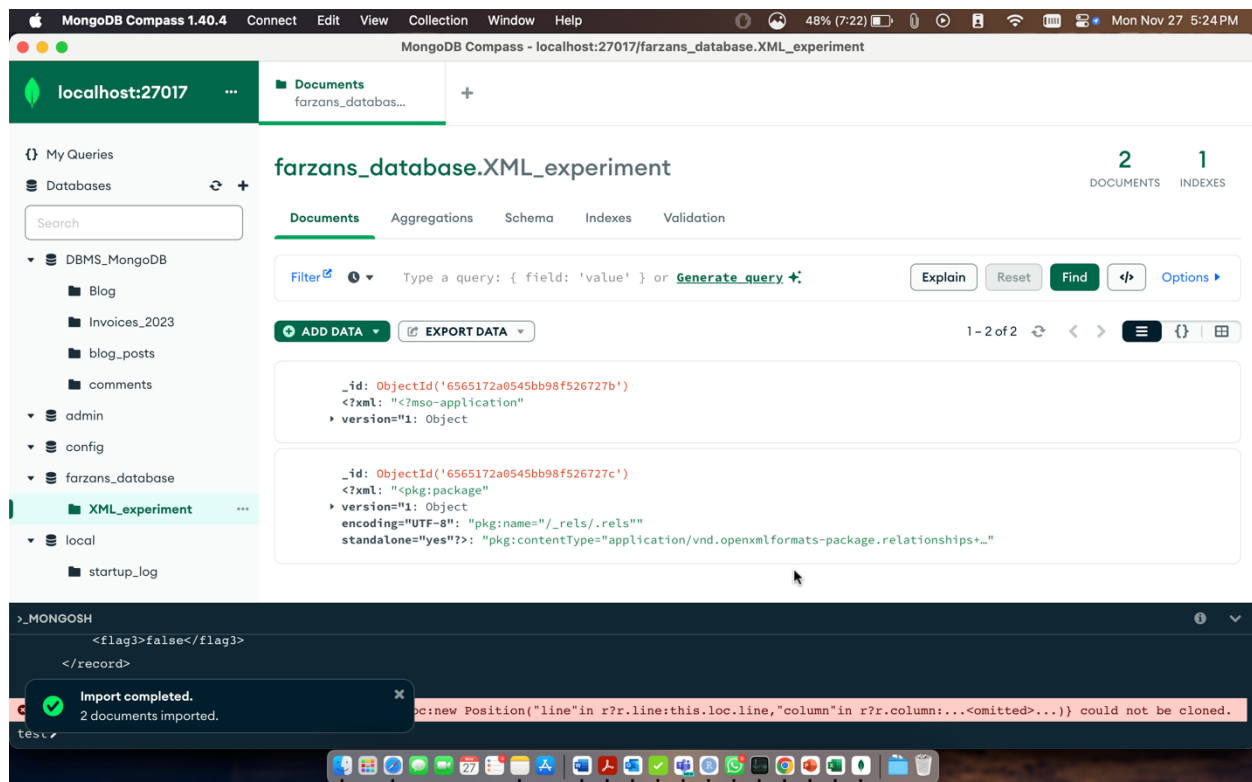
<records>
  <recender>
    <id>171</id>
    <name>Madelyn Hensley</name>

```

```
<address>10075 Thierer Plaza</address>
<city>New York</city>
<state>New York</state>
<zip>81377</zip>
<birth_year>1976</birth_year>
<gender>M</gender>
<flag1>true</flag1>
<flag2>false</flag2>
<flag3>false</flag3>
</record>
<record>
  <id>172</id>
  <name>Lonny Foster</name>
  <address>23901 Park Meadow Dr</address>
  <city>Austin</city>
  <state>Texas</state>
  <zip>13498</zip>
  <birth_year>1981</birth_year>
  <gender>F</gender>
  <flag1>true</flag1>
  <flag2>false</flag2>
  <flag3>false</flag3>
</record>
</records>
```

After saving the XML file (file to be shared along with the assignment), I created a new collection on MongoDB (called XML_experiment) and then manually inserted the data from the XML document. The following screenshots corroborate this working:





Q5. Design a graph-based diagram. Use nodes to represent entities and edges to represent relationships
 Answer:

I used the following code to create nodes:

```
CREATE (:Customer {id: 171, name: 'Madelyn Hensley', city: 'New York'})
```

```
CREATE (:Customer {id: 172, name: 'Lonny Foster', city: 'Austin'})
```

```
MATCH (c:Customer {id: 171})
```

```
CREATE (:Payment {transaction_id: '146268743-8', date: '23-8-2020', status: 'TRUE'})-[:MADE]->(c)
```

```
MATCH (c:Customer {id: 172})
```

```
CREATE (:Payment {transaction_id: '396589804-7', date: '28-9-2020', status: 'FALSE'})-[:MADE]->(c)
```

The following screenshot shows this in action:

Query took 7 ms and returned no rows.
Updated the graph - deleted 6 nodes and 6 relationships [Result Details](#)

Query:
`CREATE (:Customer {id: 171, name: 'Madelyn Hensley', city: 'New York'})`

Query took 6 ms and returned no rows.
Updated the graph - created 1 node set 3 properties [Result Details](#)

Query:
`CREATE (:Customer {id: 172, name: 'Lonny Foster', city: 'Austin'})`

Query took 2 ms and returned no rows.
Updated the graph - created 1 node set 3 properties [Result Details](#)

Query:
`MATCH (c:Customer {id: 171}) CREATE (:Payment {transaction_id: '146268743-8', date: '23-8-2020', status: 'TRUE'})-[:MADE]->(c)`

Query took 9 ms and returned no rows.
Updated the graph - created 1 node and 1 relationship set 3 properties [Result Details](#)

Query:
`MATCH (c:Customer {id: 172}) CREATE (:Payment {transaction_id: '396589804-7', date: '28-9-2020', status: 'FALSE'})-[:MADE]->(c)`

Query took 10 ms and returned no rows.
Updated the graph - created 1 node and 1 relationship set 3 properties [Result Details](#)

`MATCH (c:Customer {id: 172}) CREATE (:Payment {transaction_id: '396589804-7', date: '28-9-2020', status: 'FALSE'})-[:MADE]->(c)`

Q6. Use create statements to implement the nodes and relationships for the diagram you have designed in **Neo4j**

Answer:

The following script creates nodes for customers:

```
CREATE (:Customer {id: 171, name: 'Madelyn Hensley', city: 'New York', state: 'New York'})
CREATE (:Customer {id: 172, name: 'Lonny Foster', city: 'Austin', state: 'Texas'})
CREATE (:Customer {id: 173, name: 'Karina Livingston', city: 'Chattanooga', state: 'Tennessee'})
CREATE (:Customer {id: 174, name: 'Avery McCormick', city: 'Chicago', state: 'Illinois'})
CREATE (:Customer {id: 175, name: 'Peter King', city: 'Chicago', state: 'Illinois'})
CREATE (:Customer {id: 176, name: 'Bret Ibarra', city: 'San Diego', state: 'California'})
CREATE (:Customer {id: 177, name: 'Leonardo Wheeler', city: 'New York', state: 'New York'})
CREATE (:Customer {id: 178, name: 'Bennett Noble', city: 'Hixson', state: 'Tennessee'})
CREATE (:Customer {id: 179, name: 'Marcia Mathews', city: 'Dallas', state: 'Texas'})
CREATE (:Customer {id: 180, name: 'Avis Kramer', city: 'New York', state: 'New York'})
CREATE (:Customer {id: 181, name: 'Lynnette Tate', city: 'New York', state: 'New York'})
CREATE (:Customer {id: 182, name: 'Lakisha Estrada', city: 'Dallas', state: 'Texas'})
CREATE (:Customer {id: 183, name: 'Bill Silva', city: 'Detroit', state: 'Michigan'})
```

```
MATCH (c:Customer {id: 171})
CREATE (:Payment {transaction_id: '146268743-8', date: '23-8-2020', status: TRUE})-[:MADE]->(c)
```


MATCH (c:Customer {id: 172})
CREATE (:Payment {transaction_id: '396589804-7', date: '28-9-2020', status: FALSE})-[:MADE]->(c)
MATCH (c:Customer {id: 183})
CREATE (:Payment {transaction_id: '553753031-8', date: '25-9-2020', status: FALSE})-[:MADE]->(c)
MATCH (c:Customer {id: 183})
CREATE (:Payment {transaction_id: '559786593-4', date: '13-12-2018', status: TRUE})-[:MADE]->(c)
MATCH (c:Customer {id: 175})
CREATE (:Payment {transaction_id: '108659198-9', date: '13-9-2016', status: TRUE})-[:MADE]->(c)
MATCH (c:Customer {id: 176})
CREATE (:Payment {transaction_id: '360007723-2', date: '27-9-2016', status: FALSE})-[:MADE]->(c)
MATCH (c:Customer {id: 177})
CREATE (:Payment {transaction_id: '238309554-X', date: '28-11-2014', status: FALSE})-[:MADE]->(c)
MATCH (c:Customer {id: 178})
CREATE (:Payment {transaction_id: '694690715-8', date: '31-5-2014', status: TRUE})-[:MADE]->(c)
MATCH (c:Customer {id: 179})
CREATE (:Payment {transaction_id: '318241713-5', date: '15-7-2016', status: TRUE})-[:MADE]->(c)
MATCH (c:Customer {id: 180})
CREATE (:Payment {transaction_id: '84360172-2', date: '6-7-2016', status: FALSE})-[:MADE]->(c)
MATCH (c:Customer {id: 181})
CREATE (:Payment {transaction_id: '807633856-8', date: '5-7-2018', status: FALSE})-[:MADE]->(c)
MATCH (c:Customer {id: 182})
CREATE (:Payment {transaction_id: '845886260-4', date: '23-4-2014', status: FALSE})-[:MADE]->(c)
MATCH (c:Customer {id: 183})
CREATE (:Payment {transaction_id: '270161074-X', date: '19-7-2016', status: TRUE})-[:MADE]->(c)
MATCH (c:Customer {id: 171})
CREATE (:Payment {transaction_id: '887428332-0', date: '4-7-2020', status: FALSE})-[:MADE]->(c)
MATCH (c:Customer {id: 172})
CREATE (:Payment {transaction_id: '401129380-4', date: '23-8-2015', status: TRUE})-[:MADE]->(c)
MATCH (c:Customer {id: 173})
CREATE (:Payment {transaction_id: '277406266-7', date: '27-12-2019', status: FALSE})-[:MADE]->(c)

```

MATCH (c:Customer {id: 174})
CREATE (:Payment {transaction_id: '851112155-2', date: '4-12-2019', status: TRUE})-[:MADE]->(c)

```

The following screenshot showcases the relation between the nodes:

The screenshot displays the Neo4j Console interface. On the left, a list of queries is shown, each followed by a status message and a 'Result Details' button. The queries are:

- Query: `MATCH (c:Customer {id: 171}) CREATE (:Payment {transaction_id: '887428332-0', date: '4-7-2020', status: FALSE})-[:MADE]->(c)`
Query took 3 ms and returned no rows.
Updated the graph - created 1 node and 1 relationship set 3 properties
- Query: `MATCH (c:Customer {id: 172}) CREATE (:Payment {transaction_id: '401129380-4', date: '23-8-2019', status: TRUE})-[:MADE]->(c)`
Query took 3 ms and returned no rows.
Updated the graph - created 1 node and 1 relationship set 3 properties
- Query: `MATCH (c:Customer {id: 173}) CREATE (:Payment {transaction_id: '277406266-7', date: '27-12-2019', status: TRUE})-[:MADE]->(c)`
Query took 3 ms and returned no rows.
Updated the graph - created 1 node and 1 relationship set 3 properties
- Query: `MATCH (c:Customer {id: 174}) CREATE (:Payment {transaction_id: '851112155-2', date: '4-12-2019', status: TRUE})-[:MADE]->(c)`
Query took 3 ms and returned no rows.
Updated the graph - created 1 node and 1 relationship set 3 properties

On the right, a graph visualization shows the relationships between nodes. Nodes are represented by colored circles (orange for customers, purple for payments) and are connected by blue arrows labeled 'MADE'. The graph includes nodes such as Brett Barra, Marcia Mathews, Karina Livingston, Madelyn Hensley, Leonardo Whiskey, Peter King, Avery McCormick, Avis Kramer, Bill Silva, and Lakisha Estrada. The graph shows a complex network of 'MADE' relationships between these entities.