**Terna Engineering College**

**Computer Engineering Department**

Program: Sem VII

[**Course: Big Data Analytics & Computational Lab -I (BDA&CL-I)**](https://github.com/Amey-Thakur/BIG-DATA-ANALYTICS-AND-COMPUTATIONAL-LAB-I)

**Experiment No. 04**

**PART B**

**(PART B: TO BE COMPLETED BY STUDENTS)**

***(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)***

| Roll No. 50 | Name: AMEY THAKUR |
| --- | --- |
| Class: BE-COMPS-50 | Batch: B3 |
| Date of Experiment: 10-08-2021 | Date of Submission: 10-08-2021 |
| Grade : |  |

**Aim:** To study and run NoSql programs.

**B.1 Use Neo4j Graph Based NoSql database:**

For creating and showing the number of nodes and relations between them.

**(Paste your Search material completed during the 2 hours of practical in the lab here)**

**DATABASE**

| **NAME** | **TYPE** | **COLOR** | **FUEL** | **AFFORDABILITY** | **SPECIAL** |
| --- | --- | --- | --- | --- | --- |
| Chevrolet Bolt | SubCompact | Red | Petrol | Economic | BEST DEAL |
| Kia Rio | SubCompact | Blue | Diesel | Lavish | - |
| Toyota Yaris | SubCompact | Black | CNG | Economic | - |
| Mini Cooper | SubCompact | White | Diesel | Lavish | - |
| Toyota Corolla | Compact | Black | CNG | Economic | - |
| Volkswagen Golf | Compact | White | Petrol | Lavish | - |
| Honda Fit | Compact | Blue | Diesel | Economic | 20% Discount |
| Mazda 3 | Compact | Red | Electric | Economic | - |
| BMW 320i | Sedan | Black | Electric | Lavish | 20% Discount |
| Volkswagen Arteon | Sedan | Red | Diesel | Lavish | - |
| Mercedes-Benz C 300 | Sedan | Blue | Petrol | Economic | 20% Discount |
| KIA Stinger | Sedan | White | Electric | Economic | BEST DEAL |
| Audi A8 | Luxury | Green | Petrol | Lavish | - |
| Rolls-Royce Phantom | Luxury | Black | Electric | Lavish | - |
| Bently Bentayga | Luxury | Black | Petrol | Lavish | BEST DEAL |
| Mercedes-Benz S 550 | Luxury | Red | Electric | Lavish | - |

**B.2 Input and Output:**

**(Command and its output in graph)**

CREATE (n:Car{name:'Chevrolet Bolt',type:'Subcompact', color:'Red',fuel:'Petrol'})

CREATE (n:Car{name:'Kia Rio',type:'Subcompact', color:'Blue',fuel:'Diesel'})

CREATE (n:Car{name:'Toyota Yaris',type:'Subcompact', color:'Black',fuel:'CNG'})

.

.

.

CREATE (n:Car{name:'Bently Bentayga',type:'Luxury', color:'Black',fuel:'Petrol'})

CREATE (n:Car{name:'Mercedes-Benz S 550',type:'Luxury', color:'Red',fuel:'Electric'})

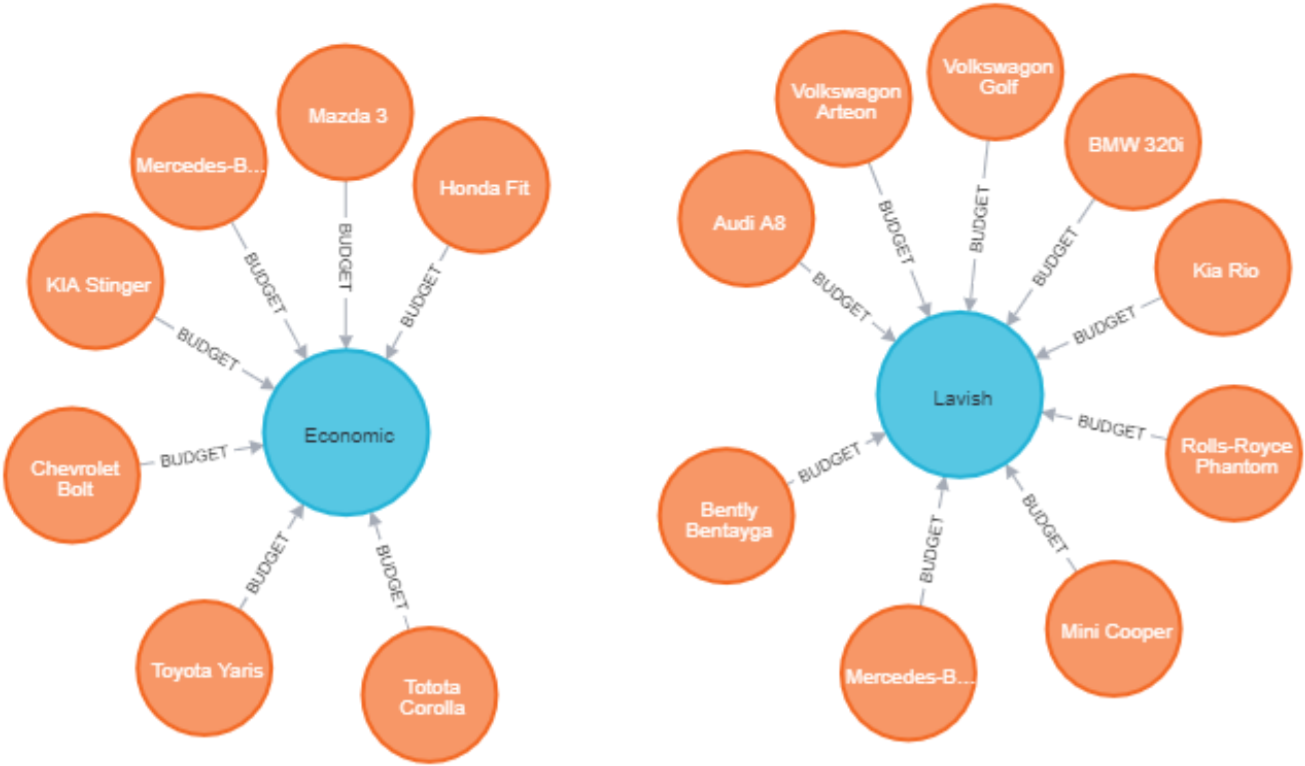
CREATE (n:affordability{name:’Lavish’, cost:’500$’}

CREATE (n:affordability{name:’Economic’, cost:’250$’}

MATCH (a:affordability), (c:Car)

WHERE a.name = 'Lavish' AND c.name='Mini Cooper'

CREATE (c)-[aff:BUDGET]->(a)



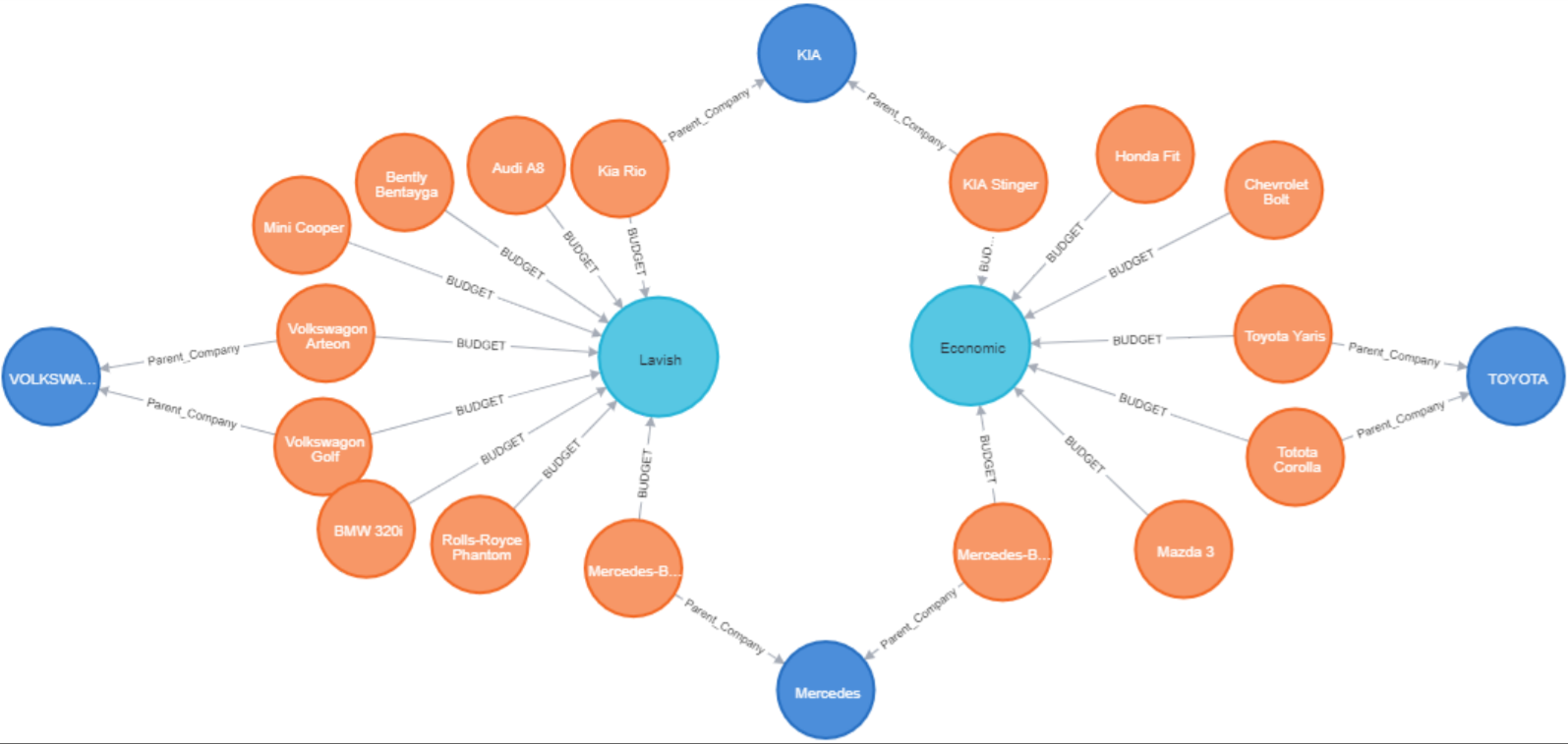
CREATE (n:Company{name:’Toyota’}

CREATE (n:Company{name:’Mercedes’}

MATCH (p:Company), (c:Car)

WHERE p.name = 'Mercedes' AND c.name='Mercedes-Benz C 300'

CREATE (c)-[same:Parent\_Company]->(p)

****

CREATE (n:Special{name:’BEST DEAL’}

CREATE (n:Special{name:’20% Discount’}

MATCH (s:special), (c:Car)

WHERE s.name = '20% Discount' AND c.name='Honda Fit'

CREATE (c)-[deal:OFFER]->(p)

****

**B.3 Observations and learning:**

**(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)**

Relationships are prioritised in graph databases, unlike conventional databases. This implies your programme won't have to rely on foreign keys or out-of-band processing techniques like MapReduce to infer data relationships.

A graph database's data model is also far more straightforward and expressive than that of relational or other NoSQL databases.

Graph databases are designed for use with transactional (OLTP) systems, with transactional integrity and operational availability as a priority.

**B.4 Conclusion:**

**(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)**

As a result, we were able to effectively construct a NoSQL graph using Neo4j.

**B.5 Question of Curiosity**

**(To be answered by student based on the practical performed and learning/observations)**

**Q1)** In which cases Neo4J is widely used?

**Ans:**

1. Fraud detection
2. Real-time recommendation engines
3. Master data management (MDM)
4. Network and IT operations
5. Identity and access management (IAM)

**Q2)** What is the difference between RDBMS and Graph Database?

**Ans:**

| **Index** | **Graph Database** | **RDBMS** |
| --- | --- | --- |
| **1.** | In the graph database, data is stored in graphs. | In RDBMS, data is stored in tables. |
| **2.** | In the graph database there are nodes. | In RDBMS, there are rows. |
| **3.** | In a graph database there are properties and their values. | In RDBMS, there are columns and data. |
| **4.** | In a graph database the connected nodes are defined by relationships. | In RDBMS, constraints are used instead of that. |
| **5.** | In graph database traversal is used instead of join. | In RDBMS, join is used instead of traversal. |