**CYPHER CODE FOR BOOK DEPOSITORY**

//Create constraints for all nodes

CREATE CONSTRAINT ON (book:Book) ASSERT book.book\_id IS UNIQUE;

CREATE CONSTRAINT ON (l:Language) ASSERT l.InLanguage IS UNIQUE;

CREATE CONSTRAINT ON (a:Author) ASSERT a.author\_id IS UNIQUE;

CREATE CONSTRAINT ON (f:Format) ASSERT f.format\_id IS UNIQUE;

CREATE CONSTRAINT ON (c:Category) ASSERT c.category\_id IS UNIQUE;

CREATE CONSTRAINT ON (age: AgeGroup) ASSERT age.age\_group IS UNIQUE;

CREATE CONSTRAINT ON (b:BestSellers) ASSERT b.bestsellers\_rank IS UNIQUE;

CREATE CONSTRAINT ON (p:Publication) ASSERT p.publication\_date IS UNIQUE;

CREATE CONSTRAINT ON (i: Images) ASSERT i.image\_path IS UNIQUE;

CREATE CONSTRAINT ON (address:Address) ASSERT address.AddressKey IS UNIQUE;

//Create Address Node

                      :auto USING PERIODIC COMMIT 500

                      LOAD CSV With HEADERS FROM 'file:///AirbnbGrp4\_CleanedData.csv' AS row

                      MERGE

                      (address:Address{Addresskey:row.AddressKey})

                      ON CREATE SET

                     address.Street=row.Street,address.City=row.City,address.State=row.State, address.Country= row.Country,address.Longitude= row.Longitude, address.Latitude= row.Latitude;

//

//Create Book Node

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MERGE(book:Book{book\_id:row.id})

ON CREATE SET

book.title=row.title, book.edition=row.edition, book.edition\_statement=row.edition\_statement, book.illustrations\_note =row. illustrations\_note, book.index\_date= row.index\_date, book.imprint= row.imprint, book.isbn10=row.isbn10, book.isbn13=row.isbn13, book.rating\_avg= row.rating\_avg, book.rating\_count= row.rating\_count, book.x\_dimension= row.dimension\_x, book.y\_dimension= row.dimension\_y, book.z\_dimension= row.dimension\_z, book.weight\_kg= row.weight, book.url= row.final\_url;

//Visualize Book Node

MATCH (book:Book)

RETURN book LIMIT 300;

//Create Language Node

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MERGE(l:Language{InLanguage:row.lang});

//Visualize Language Node

MATCH (l:Language)

RETURN l LIMIT 100;

// Load Author

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MERGE

(a:Author{author\_id: toInteger(row.author\_id)})

ON CREATE SET

a.author\_name=row.author\_name;

//Visualize Author Node

MATCH (a:Author)

RETURN a LIMIT 1000;

//Create Format Node

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MERGE

(f:Format{format\_id: row.format\_id})

ON CREATE SET

f.format\_type=row.format\_name;

//Visualize Format Node

MATCH (f:Format)

RETURN f LIMIT 100;

//Create Category Node

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MERGE

(c:Category {category\_id: row.category\_id})

ON CREATE SET

c.category\_name=row.category\_name;

//Visualize Category Node

MATCH (c:Category)

RETURN c LIMIT 1000;

//Create BestSellers Node

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MERGE(b:BestSellers { bestsellers\_rank :row. bestsellers\_rank });

//Visualize BestSellers Node

MATCH (b:BestSellers)

RETURN b LIMIT 1000;

//Create Images Node

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MERGE

(i: Images { image\_path :row. image\_path })

ON CREATE SET

i.image\_path=row.image\_path, i.image\_url= row.image\_url;

//Visualize Images Node

MATCH (i:Images)

RETURN i LIMIT 1000;

//Create AgeGroup Node

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MERGE(age: AgeGroup { age\_group :row.for\_ages });

//Visualize AgeGroup Node

MATCH (age: AgeGroup)

RETURN age LIMIT 1000;

//Create Publication Node

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MERGE(p:Publication { publication\_date :row.publication\_date  });

//Visualize AgeGroup Node

MATCH (p:Publication)

RETURN p LIMIT 1000;

//Relationships

//Book-lang relationship

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv'  AS row    MATCH (book:Book {book\_id:row.id})

MATCH (l:Language{InLanguage: row.lang})

MERGE (book)-[:available\_in]->(l);

//Visualize book-lang relationship

MATCH (book:Book)-[rel:available\_in]->(lang:Language)

RETURN book, rel, lang LIMIT 50;

//Book-author relationship

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv'  AS row

MATCH (book:Book { book\_id: row.id})

MATCH (a: Author{author\_name: row.author\_name})

MERGE (book)-[:written\_by]->(a);

//Visualize Book-author relationship

MATCH (book:Book)-[rel:written\_by]->(a:Author)

RETURN book, rel, a LIMIT 100;

//Book-format relationship

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MATCH (book:Book { book\_id: row.id})

MATCH (f: Format{format\_type: row.format\_name})

MERGE (book)-[:available\_in]->(f);

//Visualize Book-format relationship

MATCH (book:Book)-[rel:available\_in]->(f:Format)

RETURN book, rel, f LIMIT 100;

//Book-category relationship

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MATCH (book:Book { book\_id: row.id})

MATCH (c:Category { category\_name : row.category\_name})

MERGE (book)-[:belongs\_to]->(c);

//Visualize Book-category relationship

MATCH (book:Book)-[rel:belongs\_to]->(c:Category)

RETURN book, rel, c LIMIT 100;

//Book-publication relationship

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MATCH (book:Book { book\_id: row.id})

MATCH (p:Publication {publication\_date: row. publication\_date })

MERGE (book)-[:published\_on]->(p);

//Visualize Book-publication relationship

MATCH (book:Book)-[rel: published\_on]->(p:Publication)

RETURN book, rel, p LIMIT 100;

//Book-age\_group relationship

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MATCH (book:Book { book\_id: row.id})

MATCH (age:AgeGroup {age\_group: row. for\_ages })

MERGE (book)-[:read\_by]->(age);

//Visualize Book-AgeGroup relationship

MATCH (book:Book)-[rel: read\_by]->( age:AgeGroup)

RETURN book, rel, age LIMIT 100;

//Book-Images relationship

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MATCH (book:Book { book\_id: row.id})

MATCH (i:Images {image\_path: row.image\_path })

MERGE (book)-[rel: has\_an]->(i);

//Visualize Book-Images relationship

MATCH (book:Book)-[rel: has\_an]->( i:Images)

RETURN book, rel, i LIMIT 20;

//Book-BestSellers relationship

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///file\_100000.csv' AS row

MATCH (book:Book { book\_id: row.id})

MATCH (b:BestSellers { bestsellers\_rank: row.bestsellers\_rank  })

MERGE (book)-[rel: has\_a]->(b);

//Visualize Book-BestSellers relationship

MATCH (book:Book)-[rel: has\_a]->(b:BestSellers)

RETURN book, rel, b LIMIT 100;