

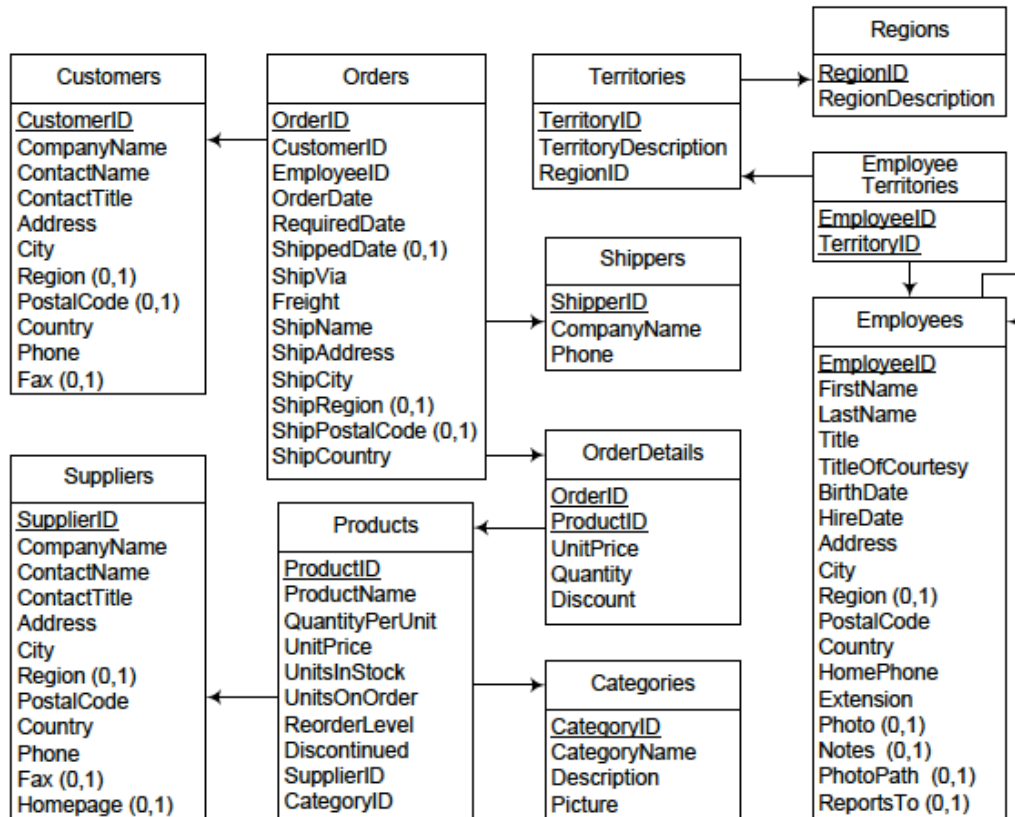


Graph Database Seminar

Part 3 – Neo4j - Practice

Alejandro Vaisman
avaisman@itba.edu.ar

Neo4j Practice – Northwind (OLTP)



Neo4j Practice – Northwind

- Must convert this DB to Neo4j
- CSV files available elsewhere
- Cypher has instructions to load CSV files

```
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:/NWdata/city.csv" AS row
CREATE (:City {cityID:row.citykey,cityName: row.cityname});
```

```
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:/NWdata/territories.csv" AS row
CREATE (:Territory {territoryID: row.territoryID,
name: row.territoryDescription});
```

...

```
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:/NWdata/employee-territories.csv" AS row
MATCH (territory:Territory{territoryID: row.territoryID})
MATCH (employee:Employee {employeeID: row.employeeID})
MERGE (employee)-[:AssignedTo]->(territory);
```

Neo4j Practice

```
-- First join orders with order details.
```

```
CREATE VIEW order1 AS (SELECT o.orderid AS orderID,o.orderdate AS  
  orderDate,o.shippeddate AS shippedDate,o.shipname AS shipName, sum(quantity)  
AS totqty,sum(unitprice*quantity) AS totAmount FROM orders o,orderdetails ol  
WHERE o.orderid=ol.orderid  
group by o.orderid,o.orderdate,o.shippeddate,o.shipname  
order by orderid asc)  
SELECT * INTO ordershg FROM order1  
  
COPY ordershg to 'C:\tmp\ordershg.csv' delimiter ',' CSV header USING PERIODIC COMMIT  
  
LOAD CSV WITH HEADERS FROM "file:/NWdata/ordershg.csv" AS row  
CREATE (:Order {orderID: row.orderid, orderDate: row.orderdate,  
ShippedDate: row.shippeddate,shipName:row.shipname,totalQty:row.totqty,  
totalAmount:row.totamount});  
  
USING PERIODIC COMMIT  
LOAD CSV WITH HEADERS FROM "file:/NWdata/orders.csv" AS row  
MATCH (order:Order {orderID: row.orderID})  
MATCH (employee:Employee {employeeID: row.employeeID})  
MERGE (employee)-[:Sold]->(order);
```

Neo4j Practice

```
LOAD CSV WITH HEADERS FROM "file:/NWdata/order-details.csv" AS row
MATCH (order:Order {orderID: row.orderID})
MATCH (product:Product {productID: row.productID})
MERGE (order)-[:Contains{unitPrice:row.unitPrice,quantity:row.quantity,
discount:row.discount}]->(product);
```

```
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:/NWdata/products.csv" AS row
MATCH (product:Product {productID: row.productID})
MATCH (supplier:Supplier {supplierID: row.supplierID})
MERGE (supplier)-[:Supplies]->(product);
```

You can also connect to a PostgreSQL database

```
CALL apoc.load.jdbc('jdbc:postgresql://localhost:5434/NWDW?user=postgres&password=postgres','select *
from ordershg') YIELD row
```

```
CREATE (:Order {orderID: row.orderid, orderDate: row.orderdate,
ShippedDate: row.shippeddate,shipName:row.shipname,totalQty:row.totqty,
totalAmount:row.totamount});
```

- Check consistency

```
MATCH (c:City) WHERE NOT (c) - [:BelongsTo]-> (:Country) WITH c
MATCH (c1:Supplier) WHERE c.cityName=c1.city return
c.cityName,c1.city,c1.country order by c.cityName asc
```

Schema: Northwindhg database

Database Information

Node Labels

- * Category City Country
- Customer Employee Order
- Product Supplier Territory

Relationship Types

- * AssignedTo BelongsTo
- Contains HasCategory LivesIn
- Purchased ReportsTo Sold
- Supplies isFrom

Property Keys

- Phone ShippedDate address
- categoryID categoryName
- city cityID cityName

\$
☆ × ▶

```
$ call db.schema
```

*(9) Order(1) Territory(1) Category(1) Employee(1) Customer(1) Product(1)

*(10) AssignedTo(1) Sold(1) isFrom(1) Contains(1) LivesIn(1) HasCategory(1)

Graph

Table

Text

Code

```

graph TD
    Territory((Territory)) -- ReportsTo --> Employee((Employee))
    Employee -- Sold --> Order((Order))
    Order -- Contains --> Product((Product))
    Supplier((Supplier)) -- Supplies --> Product
    Customer((Customer)) -- Purchased --> Order
    Customer -- LivesIn --> City((City))
    City -- BelongsTo --> Country((Country))
    Product -- HasCategory --> Category((Category))
  
```

Problem 1. Northwindhg database

- Query 1. List products and their unit price.

```
MATCH (p:Product)
RETURN p.productName, p.unitPrice
ORDER BY p.unitPrice DESC
```

- Query 2. List information about products 'Chocolate' & 'Pavlova'.

```
MATCH (p:Product)
WHERE p.productName IN ['Chocolate','Pavlova']
RETURN p
```

- Query 3. List information about products with names starting with a "C", whose unit price is greater than 50.

```
MATCH (p:Product)
WHERE p.productName STARTS WITH "C" AND tofloat(p.unitPrice) > 50
RETURN p.productName, p.unitPrice;
```

- Query 4. Same as 3, but considering the sales price, not the product's price.

```
MATCH (p:Product) <- [c:Contains] - (o:Order)
WHERE p.productName STARTS WITH "C" AND tofloat(c.unitPrice) > 50
RETURN distinct p.productName, p.unitPrice,c.unitPrice;
```

Problem 1. Northwindhg database

- Query 5. Total purchased by customer and product.

```
MATCH (c:Customer)
MATCH (p:Product)-[pu:Contains]-(:Order)-[:Purchased]-(c)
RETURN c.companyName,p.productName, tofloat(sum(tofloat(pu.unitPrice) *toInt (pu.quantity))) AS volume
ORDER BY volume DESC;
```

- Query 6. Top 10 employees, considering the number of orders sold.

- Query 7. For each employee, list the assigned territories.

```
MATCH (t:Territory)-[:AssignedTo]-(:Employee)
RETURN e.lastName, COLLECT(t.name);
```

- Query 8. For each city, list the companies settled in that city.

Problem 1. Northwindhg database

- Query 10. How many persons an employee reports to, either directly or transitively?
- Query 11. To whom do persons called “Robert” report to?
- Query 12. Who does not report to anybody?
- Query 13. Suppliers, number of categories they supply, and a list of such categories

Problem 1. Northwindhg database

- Query 14. Suppliers who supply beverages
- Query 15. Customer who purchases the largest amount of beverages
- Query 16. List the 5 most popular products (considering the number of orders)
- Query 17. Products ordered by customers from the same country than their suppliers

Problem 2 – Movies database

The screenshot shows a web browser at localhost:7474/browser/. The interface is divided into three main sections:

- Database Information (Left Panel):**
 - Node Labels:** Actor, Director, Movie, Person, User.
 - Relationship Types:** ACTS_IN, DIRECTED, FRIEND, RATED.
 - Property Keys:** A list of properties such as biography, birthday, birthplace, comment, description, genre, homepage, id, imageUrl, imdbId, language, lastModified, login, name, password, profileImageUrl, and releaseDate.
- Query Editor (Top Right):** Contains a Cypher query:

```
$ MATCH p=(v1:Actor)-[r1:ACTS_IN]->(m1:Movie)-[dm1:DIRECTED]-(d1:Director)-[dm2:DIRECTED]-(m2:Movie)-[r2:ACTS_IN]-(v2:Actor) WHERE m1.title <> m2.title RETURN p LIMIT 100
```
- Graph Visualization (Bottom Right):** A network graph showing nodes for actors and movies. The central node is "Aliens", which is connected to many other nodes. Other nodes include "The Abyss", "James Cameron", "Michael Biehn", "Paul Reiser", "Bill Paxton", "Lance Henriks...", "William Hope", "AJ Matthews", "Jenette Goldstein", "Carrie Henn", "Colette Hiller", "Daniel Kash", "Cynthia Dale Sc...", "Mark Rolston", and "Sigourney". Edges are labeled with relationship types like "ACTS_IN" and "DIRECTED".

Problem 2 – Movies database

Query 1 - Actors who played in two movies directed by the same director. Return the actor, the films, and the director.

Query 2 - Actors who played in the same film with Kevin Bacon.

Query 3 - Actors who played in a movie directed by Robert De Niro.

Query 4 - For each actor, list the number of actors she played with in a movie.

Query 5 - Actors who played in a movie with Samuel L. Jackson.

Query 6 - Shortest path between Robert De Niro and Kevin Bacon.

Query 7 - Shortest path between Kevin Bacon and Stephen Lang.

Query 8 - Shortest path between Kevin Bacon and any other actor.

Problem 3 – Worldcup database

Query 1 - Who hosted the world cup?

Query 2 - Who hosted the World Cup more than once, and when.

```
MATCH (host:Country)-[:HOSTED_BY]-(wc)
WITH wc, host ORDER BY wc.year
WITH host, count(*) AS times, collect(wc.year) AS years WHERE times > 1
RETURN host.name, times, years
```

Query 3 - Hosts that won the World Cup, and the result of the final match.

Query 4 - Top scorers per world cup.

```
MATCH (player)-->(stats)-[:SCORED_GOAL]->(goal), (stats)-[:IN_MATCH]->()-[:CONTAINS_MATCH]-(wc:WorldCup)
WHERE goal.type IN ["goal", "penalty"]
WITH player.name AS player, count(*) AS goals, collect(DISTINCT wc.year) AS competitions
UNWIND competitions AS competition
WITH player, goals, competition ORDER BY player, goals, competition
RETURN player, goals, collect(competition) AS competitions
ORDER BY goals DESC
LIMIT 5
```

Problem 3 – Worldcup database

Query 5 - Top scorer playing in the 2018 World Cup.

Query 6 - Which hosts won the World Cup that they hosted?

Query 7 - Which countries have never won a match at a World Cup?

Query 8 - What's the highest number of goals scored in a World Cup match?

Query 9 - Which stadium has hosted the most World Cup matches?

Query 10 - Which country has scored the most goals across all World Cups?

Query 11 - Which country has participated in the most World Cups?