Database Theory and Applications for Biomedical Research and Practice

BMIN 502 / EPID 635
Week 12-13: Cypher queries

John H. Holmes, PhD

Institute for Biomedical Informatics

An important point about querying graphs

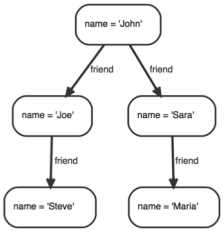
Remember: graphs contain *patterns*!

When querying a graph, you will be matching patterns determined by nodes and relationships

Some important Cypher clauses

- MATCH
  - Specifies a graph pattern to match
- WHERE
  - Works just like in SQL!
- RETURN
  - Specifies what to return
- The SQL command SELECT is like MATCH+RETURN

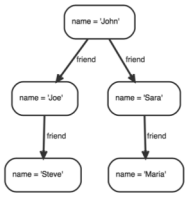
An example




```
MATCH (john {name: 'John'})-[:friend]->()-[:friend]->(fof)
RETURN john.name, fof.name
```

```
+-----------------------+
| john.name | fof.name  |
+-----------------------+
| "John"    | "Maria"   |
| "John"    | "Steve"   |
+-----------------------+
2 rows
```

Another example




```
MATCH (user)-[:friend]->(follower)
WHERE user.name IN ['Joe', 'John', 'Sara', 'Maria', 'Steve']
AND follower.name =~ 'S.*'
RETURN user.name, follower.name
```

```
+-------------------------+
| user.name | follower.name |
+-------------------------+
| "Joe"     | "Steve"       |
| "John"    | "Sara"        |
+-------------------------+
2 rows
```

And for updating the graph

- To operate on nodes and relationships
  - CREATE
  - DELETE
- To (un)assign values to properties and labels on nodes.
  - SET (can be used with CREATE)
  - REMOVE
- MERGE
  - Match existing nodes and patterns
  - Create new nodes and patterns

Setting up keys

- CREATE CONSTRAINT ON (*node*:*Label*) ASSERT exists(*node.property*)
  - All nodes with a given label have a given property
- CREATE CONSTRAINT ON (*node*:*Label*) ASSERT (*node.property1*, …, $node.property_n$) IS NODE KEY
  - Ensures that nodes with a given label have the specified properties and that the combination of property values is unique

Additional operators

- Arithmetic
- Boolean
- Comparison
- String operators
- List operators

Many functions!

- Predicate
- Scalar
- List
- Mathematical
- String
- Aggregate

Aggregate functions

- avg()- returns average of values for a given property
  - MATCH (n:Person)
  - RETURN avg(n.age)
- max()
- min()
- stDev()
- sum()

Specifying a pattern in a query

- Node patterns
  - (*x*): where x is a single node. Note the parentheses!
  - (*x*)->(*y*): relationship between x and y
- Label patterns
  - (*x*:*label*)->(*y*)
  - (*x*:*label1*:*label2*)->(*y*)
- Properties in patterns
  - (*x* {*property1*: '*value*', *property2*: '*value*'})
- Relationships in patterns
  - (*x*)-[*r*]->(*y*), where *r* is the name of the relationship

Working with external data

- LOAD CSV from '*filename.csv*' AS line
  - Loads data from filename with each line (row) instantiated as a new node in the graph
- LOAD CSV WITH HEADERS from '*filename.csv*' AS line
  - Use this when the .csv file has a header with column names
- LOAD CSV from '*filename.csv*' AS line FIELDTERMINATOR '*x*'
  - *x* is a field separator, such as , or ;
- USING PERIODIC COMMIT
  LOAD CSV from '*filename.csv*' AS line
  - This will cause an automatic commit of the data to the graph after a default of 1000 rows. You can change this by specifying a number after "COMMIT"

See this page for a complete list of clauses

https://neo4j.com/docs/developer-manual/current/cypher/keyword-glossary/

And now on to something a little more complicated

Importing relational data to create a graph database

Let's work with a clinical trial in ABIC

Shackford SR, et al.: Hypertonic saline resuscitation of patients with head injury: a prospective, randomized clinical trial. J Trauma 1998 44(1):50-8.

Abstract

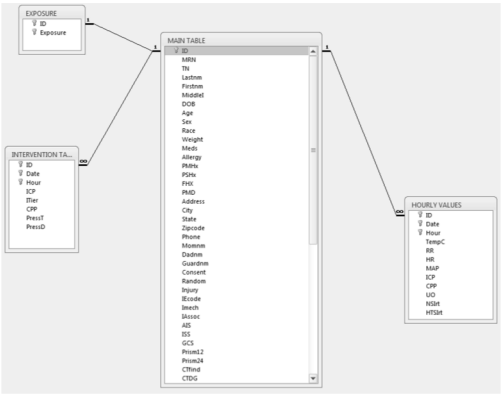
BACKGROUND: Experimental and clinical work has suggested that hypertonic saline (HTS) would be better than lactated Ringer's solution (LRS) for the resuscitation of patients with head injuries. No clinical study has examined the effect of HTS infusion on intracranial pressure (ICP) and outcome in patients with head injuries. We hypothesized that HTS infusion would result in a lower ICP and fewer medical interventions to lower ICP compared with LRS.

METHODS/DESIGN: Prospective, randomized clinical trial at two teaching hospitals.

RESULTS: Thirty-four patients were enrolled and were similar in age and Injury Severity Score. HTS patients had a lower admission Glasgow Coma Scale score (HTS: 4.7+/-0.7; LRS: 6.7+/-0.7; $p = 0.057$), a higher initial ICP (HTS: 16+/-2; LRS: 11+/-2; $p = 0.06$), and a higher initial mean maximum ICP (HTS: 31+/-3; LRS: 18+/-2; $p < 0.01$). Treatment effectively lowered ICP in both groups, and there was no significant difference between the groups in ICP at any time after entry. HTS patients required significantly more interventions (HTS: 31+/-4; LRS: 11+/-3; $p < 0.01$). During the study, the change in maximum ICP was positive in the LRS group but negative in the HTS group (LRS: +2+/-3; HTS: -9+/-4; $p < 0.05$).

CONCLUSION: As a group, HTS patients had more severe head injuries. HTS and LRS used with other therapies effectively controlled the ICP. The widely held conviction that sodium administration will lead to a sustained increase in ICP is not supported by this work.

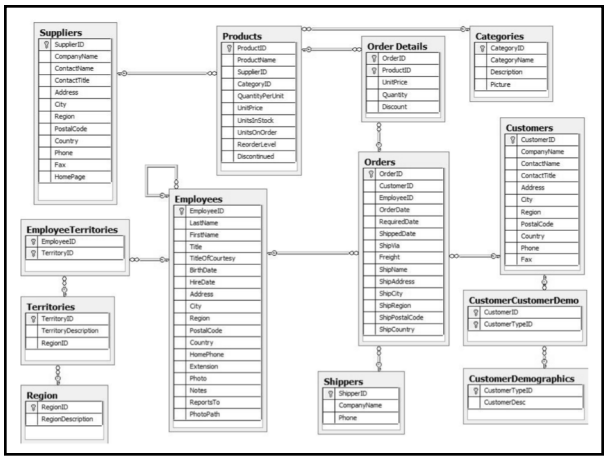
Here is the E-R model



But we are going to write queries to import these data

First, let's walk through a non-medical example:
The Northwind Database

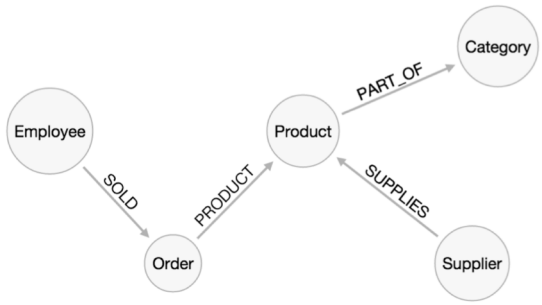
https://neo4j.com/developer/guide-importing-data-and-etl/



Steps to create graph model

1. Describe the domain
2. Identify the nodes
3. Identify the labels
4. Identify the properties
5. Construct the relationships

And the graph model



Step 0: Download the data from Canvas (all csv format)

1. Categories
2. Customers
3. Employees
4. Orders
5. Products
6. Suppliers

And place in an easily accessible folder

Step 1: Import each .csv file

```
// Create customers
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:customers.csv" AS row
CREATE (:Customer {companyName: row.CompanyName,
customerID: row.CustomerID, fax: row.Fax, phone: row.Phone});
```

Note: "file:customers.csv" will need to reflect the path, such as "/Users/Courses/BMIN 502-EPID 635/In-class exercises /Northwind"

Repeat for

- Categories
- Employees
- Products
- Suppliers

Step 1a: Orders, as the canonical file, needs special treatment

```
USING PERIODIC COMMIT LOAD CSV WITH HEADERS FROM
"file:orders.csv" AS row
MERGE (order:Order {orderID: row.OrderID}) ON CREATE SET
order.shipName = row.ShipName;
```

Step 2: Create indexes on each node

```
CREATE INDEX ON :Product(productID);
CREATE INDEX ON :Product(productName);
CREATE INDEX ON :Category(categoryID);
CREATE INDEX ON :Employee(employeeID);
CREATE INDEX ON :Supplier(supplierID);
CREATE INDEX ON :Customer(customerID);
CREATE INDEX ON :Customer(customerName);
```

And for Orders:

```
CREATE CONSTRAINT ON (o:Order) ASSERT o.orderID IS
UNIQUE;
```

After all of the nodes are indexed, type **`schema wait`** to delay populating the nodes until the indexes are created.

Step 3: Create the relationships between Orders, Products, Employees, and Customers

```
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:orders.csv" AS row
MATCH (order:Order {orderID: row.OrderID})
MATCH (product:Product {productID: row.ProductID})
MERGE (order)-[pu:PRODUCT]->(product)
ON CREATE SET pu.unitPrice = toFloat(row.UnitPrice),
pu.quantity = toFloat(row.Quantity);
```

Repeat for each of the child tables- Employees and Customers

Step 4: Create the relationships between Products, Suppliers, and Categories

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


USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:products.csv" AS row
MATCH (product:Product {productID: row.ProductID})
MATCH (category:Category {categoryID: row.CategoryID})
MERGE (product)-[:PART_OF]->(category);
```

Step 5: Create the REPORTS_TO relationship between Employees

```
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:employees.csv" AS row
MATCH (employee:Employee {employeeID: row.EmployeeID})
MATCH (manager:Employee {employeeID: row.ReportsTo})
MERGE (employee)-[:REPORTS_TO]->(manager);
```

Now, let's do some queries!

Which Employee had the Highest Cross-Selling Count of 'Chocolade' and Which Product?

```
MATCH (choc:Product {productName:'Chocolade'})<-
[:PRODUCT]-(:Order)<-[:SOLD]-(employee), (employee)-
[:SOLD]->(o2)-[:PRODUCT]->(other:Product)
RETURN employee.employeeID, other.productName,
count(distinct o2) as count
ORDER BY count
DESC LIMIT 5;
```

And another

How are Employees Organized?
Who Reports to Whom?

```
MATCH path = (e:Employee)<-[:REPORTS_TO]-(sub)
RETURN e.employeeID AS manager, sub.employeeID AS
employee;
```

Back to our clinical trial

Which is also Assignment 12!

Let's work with a clinical trial in ABIC

Shackford SR, et al.: Hypertonic saline resuscitation of patients with head injury: a prospective, randomized clinical trial. J Trauma 1998 44(1):50-8.

Abstract

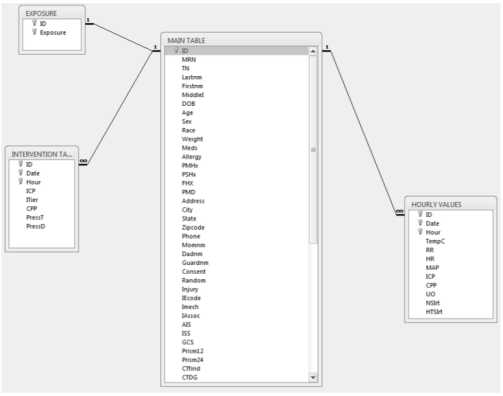
BACKGROUND: Experimental and clinical work has suggested that hypertonic saline (HTS) would be better than lactated Ringer's solution (LRS) for the resuscitation of patients with head injuries. No clinical study has examined the effect of HTS infusion on intracranial pressure (ICP) and outcome in patients with head injuries. We hypothesized that HTS infusion would result in a lower ICP and fewer medical interventions to lower ICP compared with LRS.

METHODS/DESIGN: Prospective, randomized clinical trial at two teaching hospitals.

RESULTS: Thirty-four patients were enrolled and were similar in age and Injury Severity Score. HTS patients had a lower admission Glasgow Coma Scale score (HTS: 4.7+/-0.7; LRS: 6.7+/-0.7; p = 0.057), a higher initial ICP (HTS: 16+/-2; LRS: 11+/-2; p = 0.06), and a higher initial mean maximum ICP (HTS: 31+/-3; LRS: 18+/-2; p < 0.01). Treatment effectively lowered ICP in both groups, and there was no significant difference between the groups in ICP at any time after entry. HTS patients required significantly more interventions (HTS: 31+/-4; LRS: 11+/-3; p < 0.01). During the study, the change in maximum ICP was positive in the LRS group but negative in the HTS group (LRS: +2+/-3; HTS: -9+/-4; p < 0.05).

CONCLUSION: As a group, HTS patients had more severe head injuries. HTS and LRS used with other therapies effectively controlled the ICP. The widely held conviction that sodium administration will lead to a sustained increase in ICP is not supported by this work.

Here is the E-R model

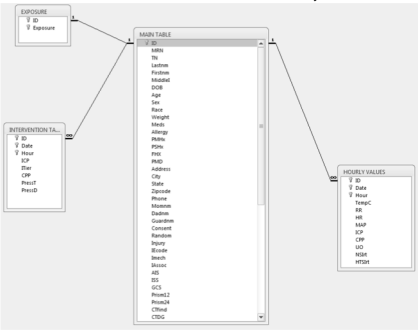


Let's review the data dictionary…

And take a look at the data

MAIN TABLE.CSV
EXPOSURE.CSV
HOURLY VALUES.CSV
INTERVENTION TABLE.CSV

Here is the E-R model, revisited



Let's create a graph model!

Steps to create graph model

1. Describe the domain
2. Identify the nodes
3. Identify the labels
4. Identify the properties
5. Construct the relationships

And on to Assignment 12 Hypertonic Saline Study

- Download from Canvas
  - The manuscript (Hypertonic Saline Resuscitation of Patients with Head Injury)
  - Data dictionary
  - The data
    - Exposure
    - Hourly values
    - Intervention
    - Main table
- Write the queries to import the data into a graph
- Write three queries of interest on the data
  - Examples:
    - What is the minimum GCS in the trial population?
    - What is the average mean arterial pressure for females between the ages of 18 and 45 who were treated with hypertonic saline?
    - …