

Oh no!

Unfortunately you have not passed. Don't worry though, you can try again in 24 hours.

You answered **59** of **80** questions correctly, giving you a total score of **73.8%**.

You can review your answers and gather feedback below.

▼ General

10 of 12 **83%**

 Relationships

True or false, a relationship can start and end at the same node.

True

False

Feedback

The answer is **True**.

 Properties

What statement best describes properties in Neo4j?

Properties are the key-value pairs used for nodes and relationships.

Properties are the key-value pairs used exclusively for nodes.

Properties are the key-value pairs used exclusively for relationships.

None of the above.

Feedback

Properties are the key-value pairs used for nodes and relationships.

✓ Graph vs RDBMS

Which of the following is a benefit of using a graph database over a relational database?

Enhanced storage capacity for large binary files

To perform fast and complex traversals of highly interconnected data

Simplified handling of unstructured text data

Better compatibility with legacy SQL applications

Feedback

Graph databases like Neo4j are optimized for querying and traversing complex and interconnected relationships, offering significantly better performance compared to relational databases for such tasks.

✓ **Which of the following elements make up a property graph?**

Select all of the options that apply

Collections

Foreign keys

Nodes

Labels

Relationships

Properties

Rows

Feedback

These key elements of a Neo4j property graph are **Nodes** and **Relationships**. Nodes and relationships hold **Properties**.

Nodes are grouped by their **Labels** and relationships are grouped by their type.

✓ Cypher

What statement best describes Cypher, Neo4j's graph query language?

It's an SQL plugin for Neo4j.

It is a regular expression-like programming language for interfacing with Neo4j.

It is a declarative query language designed for graph pattern matching and traversals.

It is a procedural programming language for interfacing with Neo4j.

Feedback

Correct. Cypher is a declarative query language designed for graph pattern matching and traversals.

You can learn more in [Cypher Fundamentals](#).

Indexes

True or False - Neo4j uses indexes to store relationships between nodes.

True

False

Feedback

Neo4j does not use indexes to store relationships between nodes.

ACID

True or False - Neo4j supports ACID transactions.

True

False

The Seven Bridges problem

True or False - It is possible to visit each landmass in the city of Königsberg while crossing each of the seven bridges only once.

The Seven Bridges of Königsberg

True

False

Feedback

Using Graph Theory, Leonhard Euler was able to definitively demonstrate that it was not possible to visit each landmass in the city of Königsberg while crossing each of the seven bridges only once.

Reference: The Seven Bridges

What is Neo4j?

What type of database is Neo4j?

Key-value store.

Document database.

Graph database.

Relational database.

Semantic database.

Feedback

Correct. Neo4j is a **Graph Database**.

You can learn more in **Neo4j Fundamentals**.

✓ Index-free Adjacency

What statement best describes index-free adjacency for a Neo4j database?

Nodes and their relationships are stored as pointers so that access is very fast without the need for an index to traverse nodes.

~~You never have to add indexes to a Neo4j database to perform faster queries.~~

~~Each relationship is stored as a table where nodes related to each are stored in the same table.~~

~~Each group of nodes is stored as a table where the relationships are mapped to nodes in the same group.~~

Feedback

Index-free adjacency means that the query engine uses pointers to traverse paths (nodes connected by relationships) in the graph, which makes queries across relationships performant.

✓ Protocols

What protocol do the official Neo4j drivers use to access the Neo4j database?

Bolt

TLS RMI NFJ

Feedback

Drivers connect to Neo4j using the **Bolt protocol**.

Query caching

To take advantage of the Query cache, what should you do?

Select all of the appropriate answers.

 Use parameters in your queries. Use literals in your queries. Ensure the casing of keywords and parameters is followed identically in queries. Define an ID for query reuse.

✓ The MERGE clause

Which of the following best describes the behavior of `MERGE` in Neo4j?

~~It always creates new nodes and relationships.~~

It matches existing patterns or creates new ones if they do not exist.

~~It only updates existing nodes and relationships.~~

~~It deletes existing nodes and relationships before creating new ones.~~

Feedback

Correct! `MERGE` will match existing patterns or create new ones if they do not exist.

Source: <https://graphacademy.neo4j.com/courses/cypher-fundamentals/2-writing/7-merge-processing/>

✓ Duplicate rows

How can you eliminate duplicate rows in the results of a Cypher query?

Using the DISTINCT keyword

~~Using the UNIQUE keyword~~

~~Using the REMOVE DUPLICATES function~~

Using the DELETE clause

Feedback

The DISTINCT keyword eliminates duplicate rows in the results. Source:

<https://graphacademy.neo4j.com/courses/cypher-intermediate-queries/2-controlling-results/05-limiting/>

✓ The MATCH clause

Which of the following statements best describes the pattern described in this MATCH clause?

`MATCH (p:Person)-[r]->(m:Movie)`

It finds nodes with the label Person with an outgoing relationship to a Movie node.

It finds nodes with the label Movie with an outgoing relationship to a Person node.

It finds nodes with the label Person with a relationship in any direction.

It is an invalid MATCH clause.

Feedback

Cypher is a declarative language that allows you to identify patterns in your data using an ASCII-art-style syntax consisting of parentheses, dashes and arrows.

The parentheses represent **nodes**.

You can learn more in [Cypher Fundamentals](#).

✓ Relationships in Cypher

In the following `MATCH` clause, which of the elements represents **the relationship between two nodes?**

`MATCH (a)-[b]->(c)`

(a)

(a)-

-[b]→

(c)

→(c)

Feedback

Relationships are drawn with two dashes, plus greater than or less than arrows to denote the direction and square brackets to enclose information about the relationship.

Learn more in **Cypher Fundamentals**.

✓ Property existence

How can you ensure that a property exists in a Cypher query?

~~Use the 'IS NULL' operator~~

Use the 'IS NOT NULL' operator

~~Use the 'EXISTS' function~~

~~Use the 'NOT EXISTS' function~~

Feedback

You can ensure that a Movie node has a 'tagline' property by using the 'IS NOT NULL' operator in a Cypher query. This filters out Movie nodes that do not have a 'tagline' property. Source: <https://graphacademy.neo4j.com/courses/cypher-fundamentals/1-reading/6-filtering-queries/>

✓ Deleting nodes

How can you delete a specific node in Neo4j using Cypher?

Use the MATCH clause to find the node and then DELETE it

~~Use the DELETE clause with the pattern you want to delete~~

~~Use the MERGE clause to delete the node~~

~~There is no way to delete a specific node in Neo4j~~

Feedback

To delete a specific node in Neo4j, you can use the MATCH clause to find the node and then DELETE it. For more information, refer to the lesson at:

<https://graphacademy.neo4j.com/courses/cypher-fundamentals/2-writing/9-delete/>

Filtering results

Which of the following Cypher statements will find people who acted in the movie Toy Story?

Select all of the options that apply.

~~MATCH people → acted in → movie(toy story)~~

MATCH (p:Person)-[:ACTED_IN]→(m:Movie) WHERE m.title = 'Toy Story' RETURN p

MATCH (p:Person)-[:ACTED_IN]→(m:Movie {title: 'Toy Story'}) RETURN p

```
SELECT path=(p:Person)-[:ACTED_IN]-(m:Movie {title: 'Toy Story'}) RETURN path
```

Feedback

The last three options are valid Cypher statements. The first two return the (:Person) node (p), while the third returns a Path ; an object consisting of a list of Relationships .

You can learn more by enrolling in [Cypher Fundamentals](#).

Parentheses in Cypher

In the following Cypher statement, what do the parentheses represent?

```
MATCH p = ()-->() RETURN p
```

Nodes

Relationships

Label

Tables

Feedback

Cypher is a declarative language that allows you to identify patterns in your data using an ASCII-art-style syntax consisting of parentheses, dashes and arrows.

The parentheses represent **nodes**.

✓ Find or create

Update the following Cypher statement to find or create a node in the database.

```
/*select:MERGE */ (m:Movie {title: 'Toy Story'})  
RETURN m
```

INSERT

MATCH OR CREATE

MERGE

UPSERT

Feedback

If Neo4j cannot find a node when using the MERGE statement, it creates a new node with the specified properties.

Reference: <https://graphacademy.neo4j.com/courses/cypher-fundamentals/2-writing/7-merge-processing/>

✓ Optional Matches

How can optional matches be useful in a Neo4j query?

Optional matches allow you to specify patterns that may or may not exist in the graph.

~~Optional matches are only used for nodes that have a specific property.~~

~~Optional matches force a pattern to exist in the graph, otherwise the query will return an error.~~

~~Optional matches are not supported in Neo4j queries.~~

Feedback

Optional matches allow you to specify patterns that may or may not exist in the graph, ensuring flexibility in query results. See more at:

<https://graphacademy.neo4j.com/courses/cypher-intermediate-queries/1-filtering-queries/13-multiple-match/>

Cypher

Which of the following Cypher statements would you use to count the number of nodes with the label Person ?

~~MATCH (n:Person) RETURN avg(n)~~

```
MATCH (n:Person) RETURN count(n)
```

```
MATCH (n:Person) RETURN sum(n)
```

```
RETURN COUNT { (:Person) }
```

Feedback

The correct answer is `MATCH (n:Person) RETURN count(n)`. It counts the number of nodes with the label `Person`.

You can also use a `COUNT` as an expression and return it directly from a `MATCH` clause.

Source: <https://graphacademy.neo4j.com/courses/cypher-intermediate-queries/3-working-with-cypher-data/02-c1-counting-results/>

✓ MERGE statements

If you run the following Cypher statement on an empty database, which properties will exist on the node?

```
MERGE (m:Movie {title: "Toy Story"})
ON CREATE SET m.released = 1995
ON MATCH SET m.tagline = "The adventure takes off! Hang on for the comedy tha
SET m.imdbRating = 8.3
```



title

released

tagline

imdbRating

Feedback

The `MERGE` command will find or create a pattern in the database.

Because this statement is run on an empty database, the node will be created.

Therefore, properties defined in `ON CREATE SET` will be set.

A previously created node has not been matched so the `ON MATCH SET` will not be executed.

The `SET` clause will be run in both cases.

Behaviour of the `MERGE` command

Say you open up a new database and run the following Cypher command to create a new `(:Movie)` node.

```
CREATE (:Movie {title: "Barbie"})
```

Then, you run the following `MERGE` Cypher statement.

```
MERGE (p:Person {name: "Margot Robbie"})-[:ACTED_IN]-(m:Movie {title: "Barbie"})
```

How many nodes will exist in your database?

~~None, you need to create a schema first~~

~~Two nodes, one (:Movie) and one (:Person)~~

Three nodes, one (:Person) and two (:Movie)

~~Four nodes, two (:Person) and two (:Movie)~~

Feedback

When you run a `MERGE` statement, Neo4j will attempt to find the entire pattern. If *any* of the elements within the pattern do not exist, the entire path will be created.

For more information, [see the `MERGE` section of the Cypher Manual](#).

✓ Returning Data

Complete the following Cypher statement to return the `name` property of the Person node.

```
MATCH (:Movie{title: 'The Matrix'})-[:ACTED_IN]-(p:Person)  
RETURN /*input:p.name*/ AS name
```

```
p.name
```

Feedback

Properties are accessed using the variable bound to the node or relationship.

The WITH clause

In Cypher, what is the primary purpose of the `WITH` clause?

To divide the query into multiple parts, allowing intermediate results to be passed to the next part of the query.

~~To filter nodes and relationships before returning results.~~

~~To specify the database to execute the query on.~~

~~To create and return new nodes and relationships in the database.~~

Feedback

The `WITH` clause is used to define or redefine the scope of variables in a Cypher query. Source: <https://graphacademy.neo4j.com/courses/cypher-intermediate-queries/5-pipelining-queries/01-with-scoping/>

✓ Union queries

What is the difference between UNION and UNION ALL in Neo4j?

~~UNION ALL returns distinct results, while UNION returns all results including duplicates~~

UNION returns distinct results, while **UNION ALL** returns all results including duplicates

~~UNION and UNION ALL are interchangeable in Neo4j~~

~~UNION and UNION ALL have no difference in Neo4j~~

Feedback

In Neo4j, UNION returns distinct results while UNION ALL returns all results, including duplicates.

Source: <https://graphacademy.neo4j.com/courses/cypher-intermediate-queries/6-subqueries/01-subquery/>

✓ Multiple Match Clauses

Why would Tom Hanks appear in this Cypher statement?

```
MATCH (p:Person)-[:ACTED_IN]->(m)
MATCH (m)<-[:ACTED_IN]-(p2)
RETURN p2.name AS coactor
```

But not in this one?

```
MATCH (p:Person)-[:ACTED_IN]->(m),  
(m)<-[ :ACTED_IN]-(p2)  
RETURN p2.name AS coactor
```

~~It's a bug in Neo4j. You should report it to Neo4j Customer Support~~

Relationships are unique within the scope of a single MATCH clause only

~~The import process probably created two Tom Hanks nodes in the database~~

~~It would never happen~~

Feedback

Relationships are unique within the scope of a single MATCH clause only. By adding a second MATCH clause, the original relationship will be traversed again.

Indexes

Which Cypher command can be used to list the indexes in a Neo4j database?

SHOW INDEXES

CREATE INDEX

```
MATCH (n) RETURN n
```

```
CALL db.indexes()
```

Feedback

The correct command is `SHOW INDEXES`.

Refer to <https://graphacademy.neo4j.com/courses/cypher-indexes-constraints/3-indexes/08-c-create-text-index/> for more details.

✓ Subqueries in Cypher

Which Cypher clause would you use to create a subquery?

```
CALL { /.../ }
```

```
CREATE { /.../ }
```

```
IN SUBQUERY { /.../ }
```

```
SUBSEQUENTLY { /.../ }
```

Feedback

See: <https://neo4j.com/docs/cypher-manual/current/clauses/call-subquery/>

Variables in subqueries

Complete the Cypher statement below to pass the `p` identifier to the subquery.

```
MATCH (p:Person)-[:ACTED_IN]->()-[:IN_GENRE]->(:Genre {name: 'Comedy'})  
CALL {  
    /*select:WITH p*/  
    SET p:FunnyActor  
} IN TRANSACTIONS OF 1000 ROWS
```

WITH p

HAVING p

USING p

SELECT p FROM \$mainQuery

Feedback

For more information, see the [Cypher manual](#).

Cypher

Which of the following commands would you use to drop a constraint?

DROP CONSTRAINT [constraint_name]

DELETE CONSTRAINT [constraint_name]

REMOVE CONSTRAINT [constraint_name]

DROP INDEX [constraint_name]

Feedback

The `DROP CONSTRAINT [constraint_name]` command is used to drop a constraint in Neo4j. Source: <https://graphacademy.neo4j.com/courses/cypher-indexes-constraints/2-constraints/11-c-managing-constraints/>

✓ MERGE statements

How do you ensure that the same relationship is not created twice between two nodes?

Use the `DISTINCT` keyword when creating the relationship.

Use the `UNIQUE` keyword when creating the relationship.

Use the `ONLY` keyword when creating the relationship.

Use the MERGE keyword when creating the relationship.

Feedback

The MERGE command will find or create a pattern in the database.

Query Pagination

In a query requiring pagination, why is it essential to use an ORDER BY clause before SKIP and LIMIT ?

It ensures consistent and predictable results across different pages.

~~It improves the query execution time.~~

~~It groups the results by a specified property.~~

~~It avoids duplicate results.~~

Feedback

Using an ORDER BY clause before SKIP and LIMIT ensures that the paginated results are consistent and predictable. Source:

<https://graphacademy.neo4j.com/courses/cypher-intermediate-queries/2-controlling-results/05-limiting/>

✓ Cypher

What is the difference between using EXPLAIN and PROFILE?

~~EXPLAIN provides exact steps and number of rows retrieved, while PROFILE provides estimates.~~

EXPLAIN provides estimates, while PROFILE provides exact steps and number of rows retrieved.

~~EXPLAIN is used for updating data, while PROFILE is used for querying.~~

~~EXPLAIN is used for querying, while PROFILE is used for updating data.~~

Feedback

The difference between EXPLAIN and PROFILE is that EXPLAIN provides estimates, while PROFILE provides exact steps and number of rows retrieved in a Neo4j query.

Reference: <https://graphacademy.neo4j.com/courses/cypher-intermediate-queries/1-filtering-queries/10-patterns/>

Escaping Special Characters

How can you escape special characters such as spaces in database, user and role names?

With double quotes (").

With single quotes (').

With backticks (`).

With backslashes (\).

Feedback

The answer is backticks (`). For information, see <https://neo4j.com/docs/operations-manual/current/manage-databases/configuration/>

✓ Cypher and GraphQL

Which of the following statements best describes the difference between Cypher and GraphQL?

Cypher is a declarative graph query language designed for querying and updating data in Neo4j, while GraphQL is a query language for APIs that allows clients to specify exactly what data they need and is used for building web APIs.

Cypher is a query language for relational databases, while GraphQL is a query language specifically designed for graph databases.

Cypher is used for data modeling, while GraphQL is used for data querying and retrieval.

Cypher provides a flexible syntax for requesting data through APIs, while GraphQL is optimized for complex graph traversals and pattern matching in Neo4j.

Feedback

Cypher is a declarative graph query language designed for querying and updating data in Neo4j, while GraphQL is a query language for APIs that allows clients to specify exactly what data they need and is used for building web APIs.

You can [learn more about Neo4j & GraphQL in Introduction to Neo4j & GraphQL](#).

✓ Behaviour of the MERGE command

Say you have a database with a unique constraint on the `.name` property for the `(:Person)` label.

You run the following Cypher statement twice:

```
MERGE (p:Person {name: "Michelle Yeoh", createdAt: datetime()})
```

What will happen?

A single node will be created with the `createdAt` property set to the latest datetime value.

Two nodes will be created with the same name property but different createdAt properties

A Neo.ClientError.Schema.ConstraintViolation error will be thrown

The database will crash and all changes will be lost

Feedback

If an operation within a transaction violates a unique constraint the transaction will be terminated, all changes will be rolled back and a

`Neo.ClientError.Schema.ConstraintViolation` will be returned to the client.

For a full list of error codes, see <https://neo4j.com/docs/status-codes/current/errors/all-errors/>

✓ Query performance

Why is it recommended to avoid unnecessary labels in query patterns for better performance?

Labels on non-anchor nodes are not allowed.

Labels on non-anchor nodes force a label check, which may not be necessary.

Labels on non-anchor nodes reduce the need for WHERE clauses.

Labels on non-anchor nodes are forbidden as part of the software licensing agreement.

Feedback

Avoiding unnecessary labels on non-anchor nodes reduces the overhead of label checks, improving query performance. Reference:

<https://graphacademy.neo4j.com/courses/cypher-intermediate-queries/4-graph-traversal/01-graph-traversal/>

Bad Cypher Statements

The following Cypher statement will match all nodes in the database and return their relationships regardless, resulting in a cartesian product. Each node will be returned as both `n` and `m` and each relationship will be returned twice.

```
MATCH (n)--(m)  
RETURN *
```

How could this statement be improved?

Select all of the options below that apply.

Define a label for one of the nodes to narrow down the initial search area

Define a relationship type

Define a direction for the relationship

Feedback

All of the answers are correct. When writing a `MATCH` clause, you should define at least one label to narrow the initial search area. You should then be specific about which relationship types and directions you would like to follow.

Both of these will ensure the amount of the graph that is explored is as small as possible.

Subqueries in Cypher

Suppose we want to return a list of all movies for each actor whose name contains "Tom". What is wrong with this code? Select the correct answer,

```
MATCH (a:Person)
WHERE a.name CONTAINS "Tom"
WITH a, a.name AS actorName
CALL
{
    MATCH (a)-[:ACTED_IN]->(m:Movie)
    RETURN collect(m.title) as movies
}
RETURN actorName, movies
```

The syntax is invalid. You must use GraphQL.

You must add `WITH movies, actorName` after the `CALL {}` block.

You can only pass property values to a subquery, not nodes.

You must specify WITH in the subquery for any variables you are passing to the subquery.

Feedback

You must specify WITH in the subquery for any variables you are passing to the subquery.

For more information, see: <https://neo4j.com/docs/cypher-manual/current/clauses/call-subquery/>

✓ Subqueries

How many records will the following Cypher statement return?

```
UNWIND range(1, 3) AS outer
CALL {
    WITH outer
    UNWIND range(1, 3) AS inner

    RETURN inner
}
RETURN outer, inner
```

+

3

6

9

12

Feedback

For each row returned by the subquery, the value in the outer query will be duplicated in the result set. The results will look like this.

outer inner

1 1

1 2

1 3

2 1

2 2

2 3

3 1

3 2

3 3

Traversal Behaviour

Which method of traversal behavior does Neo4j use during a Cypher query?

breadth-first

depth-first

topological

sequential

Feedback

The query engine always completes the first path before moving on to the next, streaming each result as it is found.(source:

<https://graphacademy.neo4j.com/courses/cypher-intermediate-queries/4-graph-traversal/01-graph-traversal/>)

Modeling

6 of 11 55%

Data modeling

Given the following domain question.

Which customers have purchased *Product X*?

Which of the following could you represent as relationships in your data model?

Customer

Purchased

Product

*

Feedback

Verbs will typically become relationships in your data model.

You can learn more in [Modeling Relationships in Modeling Fundamentals](#).

Red flags

What are some things to look for as red flags to investigate in a Cypher query plan?

AllNodesScan

NodeByLabelScan with a lot of DB hits

~~NodeByLabelScan with a few DB hits~~

Placement of eager operators

Reading property values early in the query plan.

✓ Data modeling

Given the following domain question.

What parts of type X are required to make product Y?

Which of the following elements could you **represent as nodes** in your data model?

Part

Type

Product

*

¥

Feedback

All of the nouns used in the statement could be nodes in your instance model.

✓ Intermediate nodes

You have a graph consisting of (:Customer) nodes with :ORDERED relationships to (:Products).

Instead of using a direct relationship between Customer and Product to represent an order, what is the benefit of extracting this relationship into an intermediate node Order? Which of the following optimizations involves enhancing the data model by extracting the 'ordered' relationship into a separate node?

~~It allows us to only link products to customers.~~

It enables linking the order to multiple entities, such as products and shipping companies.

~~It simplifies the graph by reducing the number of nodes.~~

~~It prevents the need for any relationships between Customer and Product.~~

Feedback

You can learn more in **Intermediate Nodes**

Bi-directional relationships

In modeling, the use of bi-directional relationships is a **good practice** when:

The semantics of the relationship in one direction is different from the other direction.

~~You want to show the same relationship between two nodes in each direction.~~

~~A relationship between two nodes could be represented in either direction.~~

~~This is never a good idea.~~

Feedback

Bi-directional relationships make sense when the semantics of the relationship in one direction is different from the other direction. For example, using different weights for the relationships.

Adding additional labels

You have a database consisting of (:Person) nodes connected to (:Movie) nodes through :ACTED_IN and :DIRECTED relationships?

What is the benefit of adding additional (:Actor) labels for any (:Person) node with an :ACTED_IN relationship?

~~It reduces the number of nodes in the graph~~

It provides a faster way to find a subset of (:Person) nodes

~~You can remove the :ACTED_IN relationships altogether~~

~~There are no benefits~~

Feedback

Adding a label will reduce the search space at the start of the query, reducing the number of db hits and making the statement more performant.

Labels

What is the recommended limit for the number of labels for a node in a data model?

2 labels

4 labels

6 labels

8 labels

Feedback

Incorrect. The recommended limit is 4 labels for a node in a data model. If you assign more than four labels against a node, the labels will be stored in a separate file which can increase the number of db hits per query. Source:

<https://graphacademy.neo4j.com/courses/modeling-fundamentals/5-refactoring-graph/2-labels-in-graph/>

Modeling categories

You are building a recommendation engine for an online store.

The store organizes products into departments, each containing sub-categories.

An excerpt of their category list is as follows:

Computers & Accessories > Accessories
Computers & Accessories > Computers & Laptops
Computers & Accessories > Computers & Laptops > Apple
Computers & Accessories > Computers & Laptops > Samsung
Computers & Accessories > Data Storage
Computers & Accessories > Input Devices
Computers & Accessories > Input Devices > Keyboards
Computers & Accessories > Input Devices > Mice
Computers & Accessories > Printing
Computers & Accessories > Printing > Printers
Computers & Accessories > Printing > Cartridges

Which of the following actions should you take when modeling the data?

~~Assign a new label for each category to (:Product) nodes, for example, :Product_Accessories~~

Create a hierarchy of (:Category) nodes

~~Create a category property on each node to hold an array of categories~~

Feedback

Because the categories represent a hierarchy, it makes sense to create nodes for each category and use relationships to create the hierarchy. This means you can quickly find products in a category, but also traverse the hierarchy to find products in sub-categories.

See: [Graph Data Modeling Fundamentals](#).

Query tuning

Before you start a query tuning exercise, you must make sure that the set of queries you will be measuring represents the correct syntax. You can use `EXPLAIN` to ensure that unnecessary `NodeByLabelScan` steps occur.

For each query, you should confirm that:

An index will be used as expected.

Use correct names for labels, property keys, and property values.

Use correct relationship types.

~~Use literal strings to test values.~~

Feedback

You should make sure that you:

- An index will be used as expected.
- Use correct names for labels, property keys, and property values.
- Use correct relationship types.

✓ Modeling categories

You are building a recommendation engine for an online store.

Some of their products are perishable and need to be sold within a certain timeframe.

Managers would like to find a list of perishable products for stock control purposes.

Which of the following actions should you take when modeling the data?

Add a :Perishable label to perishable products

Create a boolean property named perishable

Create a :Status node with the name Perishable and create relationships from each product node

Feedback

Managers need to quickly identify a subset of products. Because products are either perishable or not, a boolean and not hierarchical, using a label will narrow down the search area within the graph.

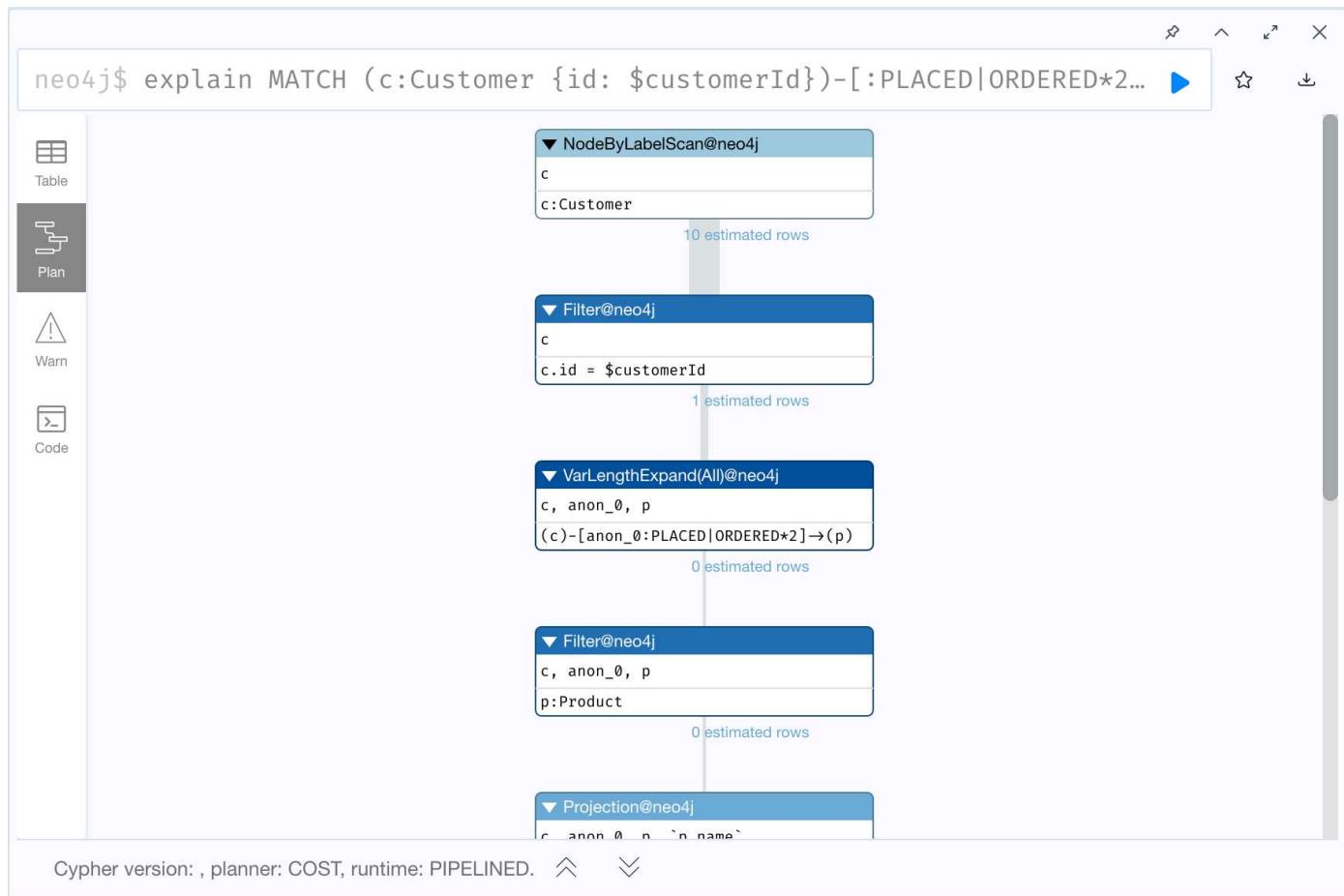
See: [Graph Data Modeling Fundamentals](#).

✓ Slow running queries

You receive reports from users that your website is slow. You narrow down the problem to a single Cypher statement used on the home page.

```
MATCH (c:Customer {id: $customerId})-[:PLACED|ORDERED*2]->(p:Product)
RETURN p.name
ORDER BY p.name ASC
```

After running the `EXPLAIN` command, you spot a `NodeByLabelScan` as the first step in the query plan.



Which of the following actions should you take to improve the performance of the query?

Add query caching to your application server

Add an index to the :Customer.id property

Change the runtime to PARALLEL

Delete some data from the database

Feedback

A `NodeByLabelScan` step at the start of a query plan followed immediately by a `Filter` is a sign that the database is scanning all nodes with that label before. You should consider adding an index to this node and property combination.

To learn more, [see Modeling Fundamentals](#).

Schema

7 of 11 **64%**

Unique constraints

How do you create a unique constraint on the `email` property for any node with the label `User`?

`CREATE INDEX ON :User(email)`

`CREATE CONSTRAINT ON (u:User) ASSERT u.email IS UNIQUE`

`CREATE CONSTRAINT ON (u:User) ASSERT u.email IS NODE KEY`

`CREATE CONSTRAINT FOR (u:User) REQUIRE (u.email) IS UNIQUE`

Feedback

The Cypher statement

CREATE CONSTRAINT FOR (u:User) REQUIRE (u.email) IS UNIQUE creates a unique constraint on the 'email' property for nodes with the 'User' label. For more details, refer to the lesson at <https://graphacademy.neo4j.com/courses/modeling-fundamentals/7-using-specific-relationships/1-specific-relationships/>

Indexes

True or False - You can create an index on nodes or relationships.

True

False

Feedback

This is true, you can create a wide range of indexes and constraints on nodes and relationships.

Schemas in Neo4j

Which of the following statements is correct?

Neo4j does not support schemas

Neo4j allows for an optional schema

~~Neo4j enforces a strict schema~~

Feedback

You can operate a Neo4j database with no schema, but applying a database schema will improve performance.

You can learn more about applying a schema to Neo4j in [Cypher Indexes & Constraints](#).

Unique constraints

Complete the Cypher statement below to create an existence constraint on the `rated` property on the `()-[:RATED] -> ()` relationship.

```
CREATE CONSTRAINT rated_rating  
FOR ()-[r:RATED]-() REQUIRE r.rating /*select:IS NOT NULL*/
```

EXISTS

IS UNIQUE

IS NOT NULL

:= NOT NULL

Feedback

You can learn all about the indexes and constraints available in Neo4j in the [Cypher Indexes and Constraints course](#).

Indexes

Which of the following indexes are available in Neo4j?

BLOOM

CURRENCY

TEXT

RANGE

VECTOR

Feedback

Learn how to make your graph more performant in the [Cypher Indexes and Constraints course](#).

✓ Schema visualizations

Which of the following Cypher statements will visualize your database schema in Workspace or Neo4j Browser?

`CALL db.schema.visualization()`

`SHOW SCHEMA`

`MATCH schema = VISUALIZE ()-[]->() RETURN schema`

`CALL schema.visualize()`

Feedback

To visualize your graph data schema, call the `db.schema.visualization()` procedure. To get a general view, you can also return a random sample of nodes and relationships.

```
MATCH p = ()-[ ]->()
RETURN p
LIMIT 1000
```

✓ Indexes

How do you create a RANGE index on a relationship property?

```
CREATE RANGE INDEX index_name FOR ()-[relType:RELATIONSHIP_TYPE]-() ON  
(relType.property_key)
```

```
CREATE BTREE INDEX index_name FOR (n:NODE_LABEL) ON (n.property_key)
```

```
CREATE FULLTEXT INDEX index_name FOR (n:NODE_LABEL) ON (n.property_key)
```

```
CREATE TEXT INDEX index_name FOR ()-[relType:RELATIONSHIP_TYPE]-() ON  
(relType.property_key)
```

Feedback

To create a RANGE index on a relationship property in Neo4j, you use the syntax:

```
CREATE RANGE INDEX index_name FOR ()-[relType:RELATIONSHIP_TYPE]-() ON  
(relType.property_key)
```

. Source: <https://graphacademy.neo4j.com/courses/cypher-indexes-constraints/3-indexes/08-c-create-text-index/>

Property combination index

What type of constraint can you use to ensure that **two or more properties** are unique for any given label?

Combination Constraint

Unique Constraint

Node Key Constraint

NodePropertyConst

Feedback

You can learn all about the indexes and constraints available in Neo4j in the [Cypher Indexes and Constraints course](#).

✓ Full-text indexes

Complete the Cypher statement below to query a `fulltext` index called `movieInfo` with the value in the parameter `$query` .

```
/*select:CALL db.index.fulltext.queryNodes*/("movieInfo", $query)  
YIELD node, score  
RETURN node.title, score
```

CALL db.index.fulltext.queryNodes

QUERY FULLTEXT INDEX WITH

SELECT INDEX

MATCH fulltext

```
CALL fulltext.queryNodes
```

Feedback

You can learn all about `fulltext` indexes in the [Cypher Indexes and Constraints course](#).

✓ Full-text indexes

What makes `fulltext` and `vector` indexes different from other indexes in Neo4j?

They can be specified for properties of relationships.

They must be queried by calling a procedure.

They can be used to constrain uniqueness of a relationship property value.

They can be used to index a combination of node and relationship properties.

Feedback

You can learn all about the indexes and constraints available in Neo4j in the [Cypher Indexes and Constraints course](#).

To learn about Vector indexes, check out [Neo4j & LLM Fundamentals](#).

Vector indexes

Complete the Cypher statement below to create a Vector index on the `:Movie.embedding` property.

```
/*select:CREATE VECTOR INDEX*/ moviePlots IF NOT EXISTS  
FOR (m:Movie) ON m.embedding  
OPTIONS {indexConfig: {  
    `vector.dimensions`: 1536,  
    `vector.similarity_function`: 'cosine'  
}}
```

CREATE VECTOR INDEX

CREATE INDEX

CREATE CONSTRAINT

CALL db.vector.createIndex

Feedback

To create a vector index, call the `CREATE VECTOR INDEX` command.

To learn about Vector indexes, check out [Neo4j & LLM Fundamentals](#).

▽ Importing

5 of 6 83%

✓ Complex data imports

In a complex data import scenario involving both nodes and relationships, which of the following sequence of steps is recommended?

Import nodes first, then relationships

Import relationships first, then nodes

Import nodes and relationships simultaneously

Import nodes and relationships in separate transactions

Feedback

A multi-pass approach to importing data will ensure that larger imports are more performant.

<https://graphacademy.neo4j.com/courses/importing-fundamentals/>

✓ Importing CSV data

Which Cypher command would you use to load data from a CSV file?

LOAD CSV

IMPORT CSV

READ CSV

GET CSV

Feedback

LOAD CSV is the Cypher command that loads data from a CSV file. Source:

<https://graphacademy.neo4j.com/courses/importing-fundamentals/>

✓ Importing CSV files

If you omit WITH HEADERS from a LOAD CSV statement, which of the following will happen?

The statement will lead to a Syntax Error

An LLM will be to generate headers for you

Each row will be treated as a list

The database will crash

Feedback

If `WITH HEADERS` is not present, the data will be returned as a list of values. You can access these values using a zero-based index, for example, `row[0]`.

✓ Importing CSV files

Complete the following Cypher statement to instruct Neo4j that the first row in the CSV file contains headers.

```
LOAD CSV /*select:WITH HEADERS*/ FROM 'file:///people.csv' AS row  
CREATE (p:Person)  
SET p += row
```

~~HAS HEADERS~~

~~HEADERROW=1~~

~~USE FIRST AS HEADERS~~

WITH HEADERS

Feedback

`LOAD CSV` is the Cypher command used to load data from a CSV file. The command has a `WITH HEADERS` option that instructs Neo4j that the first row contains headers.

Source: <https://graphacademy.neo4j.com/courses/importing-fundamentals/>

✓ Data types

Which of the following statements applies when using the `LOAD CSV` statement?

Select all of the options below that apply.

~~The database will automatically determine the data type when reading the CSV file~~

All values are cast as a string

~~You must define the data types for each file in import.conf~~

You must cast any values other than a string using the appropriate Cypher function

Feedback

When loading a CSV file using `LOAD CSV`, all values will be treated as strings. This ensures that no information is lost during the import process. You must use the appropriate function to convert the value into the correct data type, for example the `toInteger()` and `toBoolean()` functions.

Batch Imports

Complete the Cypher statement below to execute the `LOAD CSV` command in batches of 1000 rows.

```
LOAD CSV WITH HEADERS FROM 'file:///people.csv' AS row
CALL {
    WITH row
    MERGE (p:Person {id: row.id})
    SET p += row
} /*select:IN TRANSACTIONS OF 1000 ROWS*/
```

COMMIT ON 1000

IN TRANSACTIONS OF 1000 ROWS

OPTIONS { batchSize: 1000 }

ROWS=1000

Feedback

The correct syntax for setting the transaction size in a subquery is

```
CALL {} IN TRANSACTIONS OF 1000 ROWS
```

`CALL { ... } IN TRANSACTIONS` is only allowed in **implicit transactions**.

If you are using **Neo4j Browser**, you must prepend any queries using

```
CALL { ... } IN TRANSACTIONS with :auto .
```

Drivers6 of 8 **75%****Driver Protocols**

Which of the following protocols are valid to create a secure connection to Neo4j using the Driver?

bolt+s**graphdb+s****neo4j+s****neo4j+ssc****https****Feedback**

The correct answers are **bolt+s**, **neo4j+s** and **neo4j+ssc**.

Bolt Port

What is the default port for connecting to a Neo4j database using the driver?

7474

7687

7987

8501

Feedback

The correct answer is 7687 .

Fun fact, the port number is used because the two developers who coined the bolt protocol were born in 1976 and 1987.

Drivers

What is the role of a driver in Neo4j?

To establish a connection with the Neo4j database

To store data in the graph database

To define the data model in Neo4j

To visualize nodes and relationships in a forced graph layout

Feedback

The role of a driver in Neo4j is to establish a connection with the Neo4j database. For more information, refer to: <https://graphacademy.neo4j.com/courses/modeling-fundamentals/1-getting-started/3-purpose-of-model/>

Drivers

When establishing a driver connection using the `neo4j+ssc`, which of the following will occur?

The driver will establish an unencrypted connection

The driver will establish an encrypted connection and it will verify the SSL certificate

The driver will establish an encrypted connection but it will not verify the SSL certificate

Feedback

The `neo4j+ssc` protocol securely connects to a Neo4j database using SSL/TLS encryption with a self-signed certificate. This ensures encrypted data is transmitted between the client and server, providing security and privacy, typically in development or testing environments.

You can [learn more about connecting an application to Neo4j in our Development courses.](#)

✓ Verifying Database Credentials

Complete the following Python code to verify the database credentials passed to the `GraphDatabase.driver()` method are correct.

```
driver = GraphDatabase.driver(  
    "neo4j://localhost:7687",  
    auth=("neo4j", "1$ecurepassword")  
)
```

```
driver.#select:verify_authentication()
```

```
assert_auth()
```

```
check_authentication()
```

```
verify_authentication()
```

```
verify()
```

Feedback

The `verify_authentication()` method will try to establish a working connection to a remote database or cluster method. The method will return `True` if the credentials are valid, otherwise, it will return `False`.

✓ Drivers

When using a language driver, a statement of results comprises a stream of...

nodes

rows

entries

records

Feedback

You can [learn more about how to connect an application to Neo4j in our Development courses.](#)

Retrying Failed Transactions

When using a language driver, which forms of transaction functions will be automatically retried on failure?

Auto-commit transactions

Transaction functions

Explicit transactions

Feedback

Transaction functions will automatically retry when they encounter transient errors.

You can **learn more about how to connect an application to Neo4j in our Development courses.**

✓ Reading Data from Neo4j

Complete the following Python code to execute a one-off Cypher statement and consume the results as a Pandas DataFrame .

```
driver = GraphDatabase.driver(  
    "neo4j://localhost:7687" auth=("neo4j", "letmein")  
)  
  
df = driver.#select:execute_query  
(  
    "MATCH (n) RETURN count(n) AS count",  
    result_transformer_=Result.to_df  
)
```

cypher

to_df

execute_query

run

```
query
```

Feedback

To execute a Cypher statement in the Python driver, run the `execute_query` method.

[Return to Neo4j Certified Professional](#)

Copyright © 2025 Neo4j Inc.