



Neo4j Status Codes v5

Table of Contents

Format	2
Classification.....	3
Error codes	3
Notification codes.....	10
Changes to status codes per Neo4j version.....	31

This document details all status codes that a Neo4j DBMS may return to indicate the outcome of a request.

Format

Each status code follows the same format:

```
Neo.[Classification].[Category].[Title]
```

Classification

The fact that a status code is returned by the server does not always mean there is a fatal error. Status codes can also indicate transient problems that may go away if you retry the request. The classification of the status code determines the effect on the transaction.

Table 1. Neo4j status code classification

Classification	Description	Effect on the transaction
<code>ClientError</code>	The Client sent a bad request - changing the request might yield a successful outcome.	Rollback
<code>TransientError</code>	The database cannot service the request right now, retrying later might yield a successful outcome.	Rollback
<code>DatabaseError</code>	The database failed to service the request.	Rollback
<code>ClientNotifications</code>	The query execution was successful, but there are notifications about the request sent by the client.	None

License: Creative Commons 4.0

Error codes

Error codes are returned by the server when the execution of a query fails. Error codes have severity **ERROR**.

Types of error codes

All errors returned are classified into one of the following categories:

Client errors

These errors are caused by the client and are usually related to the request itself. Client errors have the prefix `Neo.ClientError`.

Transient errors

These errors are detected by the server and are usually related to some kind of database unavailability, such as limits reached, out-of-memory, timeouts, etc. The error can be temporary and could therefore succeed if retrying the request. Transient errors have the prefix `Neo.TransientError`.

Database errors

These errors are caused by the database and are usually related to the database state. Database errors have the prefix `Neo.DatabaseError`.

List of all error codes

This page contains lists of all neo4j errors, grouped by classification.

Client errors

This is a complete list of all client error status codes Neo4j may return, and what they mean.

Status Code	Description
<code>Neo.ClientError.Cluster.NotALeader</code>	The request cannot be processed by this server. Write requests can only be processed by the leader.
<code>Neo.ClientError.Cluster.Routing</code>	Unable to route the request to the appropriate server
<code>Neo.ClientError.Database.DatabaseNotFound</code>	The request referred to a database that does not exist.
<code>Neo.ClientError.Database.ExistingAliasFound</code>	The request referred to a database with an alias.
<code>Neo.ClientError.Database.ExistingDatabaseFound</code>	The request referred to a database that already exists.
<code>Neo.ClientError.Fabric.AccessMode</code>	The request could not be completed due to an access mode violation
<code>Neo.ClientError.General.ForbiddenOnReadOnlyDatabase</code>	This is a read-only database, writing or modifying the database is not allowed.
<code>Neo.ClientError.General.InvalidArguments</code>	The request contained fields that were empty or are not allowed.
<code>Neo.ClientError.General.TransactionOutOfMemoryError</code>	The transaction used more memory than was allowed. The maximum allowed size for a transaction can be configured with <code>dbms.memory.transaction.max_size</code> in the Neo4j configuration file (normally in <code>conf/neo4j.conf</code> or, if you are using Neo4j Desktop, found through the user interface).
<code>Neo.ClientError.General.UpgradeRequired</code>	This transaction required the database to be of a higher kernel version than it is. Make sure that the Neo4j DBMS has been correctly upgraded before retrying this operation.
<code>Neo.ClientError.General.WriteOnReadOnlyAccessDatabase</code>	This database is in read-only mode, writing or modifying the database is not allowed.
<code>Neo.ClientError.Procedure.ProcedureCallFailed</code>	Failed to invoke a procedure. See the detailed error description for the exact cause.
<code>Neo.ClientError.Procedure.ProcedureNotFound</code>	A request referred to a procedure that is not registered with this database instance. If you are deploying custom procedures in a cluster setup, ensure all instances in the cluster have the procedure jar file deployed.
<code>Neo.ClientError.Procedure.ProcedureRegistrationFailed</code>	The database failed to register a procedure, refer to the associated error message for details.
<code>Neo.ClientError.Procedure.ProcedureTimedOut</code>	The procedure did not complete within the specified timeout. You may want to retry with a longer timeout.
<code>Neo.ClientError.Procedure.TypeError</code>	A procedure is using or receiving a value of an invalid type.
<code>Neo.ClientError.Request.Invalid</code>	The client provided an invalid request.
<code>Neo.ClientError.Request.InvalidFormat</code>	The client provided a request that was missing required fields or had values that are not allowed.

Status Code	Description
<code>Neo.ClientError.Request.InvalidUsage</code>	The client made a request but did not consume outgoing buffers in a timely fashion.
<code>Neo.ClientError.Schema.ConstraintAlreadyExists</code>	Unable to perform this operation because it will clash with a pre-existing constraint.
<code>Neo.ClientError.Schema.ConstraintNotFound</code>	The request (directly or indirectly) referred to a constraint that does not exist.
<code>Neo.ClientError.Schema.ConstraintValidationFailed</code>	A constraint imposed by the database was violated.
<code>Neo.ClientError.Schema.ConstraintViolation</code>	Added or changed index entry violates a constraint.
<code>Neo.ClientError.Schema.ConstraintWithNameAlreadyExists</code>	Unable to perform this operation because a constraint with that name already exists.
<code>Neo.ClientError.Schema.EquivalentSchemaRuleAlreadyExists</code>	Unable to perform this operation because an equivalent schema rule already exists.
<code>Neo.ClientError.Schema.ForbiddenOnConstraintIndex</code>	A requested operation cannot be performed on the specified index because the index is part of a constraint. If you want to drop the index, you must also drop the constraint.
<code>Neo.ClientError.Schema.IndexAlreadyExists</code>	Unable to perform this operation because it will clash with a pre-existing index.
<code>Neo.ClientError.Schema.IndexMultipleFound</code>	The request referenced an index by its schema and multiple matching indexes were found.
<code>Neo.ClientError.Schema.IndexNotApplicable</code>	The request did not contain the properties required by the index.
<code>Neo.ClientError.Schema.IndexNotFound</code>	The request (directly or indirectly) referred to an index that does not exist.
<code>Neo.ClientError.Schema.IndexWithNameAlreadyExists</code>	Unable to perform this operation because an index with that name already exists.
<code>Neo.ClientError.Schema.RepeatedLabelInSchema</code>	Unable to create an index or a constraint because the schema had a repeated label.
<code>Neo.ClientError.Schema.RepeatedPropertyInCompositeSchema</code>	Unable to create an index or a constraint because the schema had a repeated property.
<code>Neo.ClientError.Schema.RepeatedRelationshipTypeInSchema</code>	Unable to create an index or a constraint because the schema had a repeated relationship type.
<code>Neo.ClientError.Schema.TokenNameError</code>	An invalid name is used for a token (label, relationship type, property key). Tokens cannot be empty strings and cannot be null.
<code>Neo.ClientError.Security.AuthenticationRateLimit</code>	The client has provided incorrect authentication details too many times in a row.
<code>Neo.ClientError.Security.AuthorizationExpired</code>	The stored authorization info has expired. Please reconnect.
<code>Neo.ClientError.Security.CredentialsExpired</code>	The credentials have expired and need to be updated.
<code>Neo.ClientError.Security.Forbidden</code>	An attempt was made to perform an unauthorized action.

Status Code	Description
<code>Neo.ClientError.Security.TokenExpired</code>	The auth provider token has expired.
<code>Neo.ClientError.Security.Unauthorized</code>	The client is unauthorized due to authentication failure.
<code>Neo.ClientError.Statement.AccessMode</code>	The request could not be completed due to an access mode violation.
<code>Neo.ClientError.Statement.ArgumentError</code>	The statement is attempting to perform operations using invalid arguments.
<code>Neo.ClientError.Statement.ArithmeticError</code>	Invalid use of arithmetic, such as dividing by zero.
<code>Neo.ClientError.Statement.ConstraintVerificationFailed</code>	A constraint imposed by the statement is violated by the data in the database.
<code>Neo.ClientError.Statement.EntityNotFound</code>	The statement refers to a non-existent entity.
<code>Neo.ClientError.Statement.ExternalResourceFailed</code>	Access to an external resource failed.
<code>Neo.ClientError.Statement.NotSystemDatabaseError</code>	This is an administration command and it should be executed against the <code>system</code> database.
<code>Neo.ClientError.Statement.ParameterMissing</code>	The statement refers to a parameter that was not provided in the request.
<code>Neo.ClientError.Statement.PropertyNotFound</code>	The statement refers to a non-existent property.
<code>Neo.ClientError.Statement.RuntimeUnsupportedError</code>	This query is not supported by the chosen runtime.
<code>Neo.ClientError.Statement.SemanticError</code>	The statement is syntactically valid but expresses something that the database cannot do.
<code>Neo.ClientError.Statement.SyntaxError</code>	The statement contains invalid or unsupported syntax.
<code>Neo.ClientError.Statement.TypeError</code>	The statement is attempting to perform operations on values with types that are not supported by the operation.
<code>Neo.ClientError.Statement.UnsupportedAdministrationCommand</code>	This administration command is not supported.
<code>Neo.ClientError.Statement.UnsupportedOperationError</code>	This query performed an operation that is not supported in this context.
<code>Neo.ClientError.Transaction.ForbiddenDueToTransactionType</code>	The transaction is of the wrong type to service the request. For instance, a transaction that has had schema modifications performed in it cannot be used to subsequently perform data operations, and vice versa.
<code>Neo.ClientError.Transaction.InvalidBookmark</code>	The supplied bookmark cannot be interpreted. You should only supply a bookmark that was previously generated by Neo4j. Maybe you have generated your own bookmark, or modified a bookmark since it was generated by Neo4j.
<code>Neo.ClientError.Transaction.InvalidBookmarkMixture</code>	Mixing bookmarks generated by different databases is forbidden. You should only chain bookmarks that are generated from the same database. You may however chain bookmarks generated from the <code>system</code> database with bookmarks from another database.

Status Code	Description
<code>Neo.ClientError.Transaction.LockClientStopped</code>	The transaction has been terminated, so no more locks can be acquired. This may occur because the transaction ran longer than the configured transaction timeout, a human operator manually terminated the transaction, or the database shut down.
<code>Neo.ClientError.Transaction.Terminated</code>	Explicitly terminated by the user.
<code>Neo.ClientError.Transaction.TransactionAccessedConcurrently</code>	There were concurrent requests accessing the same transaction, which is not allowed.
<code>Neo.ClientError.Transaction.TransactionHookFailed</code>	Transaction hook failure.
<code>Neo.ClientError.Transaction.TransactionMarkedAsFailed</code>	The transaction was marked as both successful and failed. Because failure takes precedence, the transaction was rolled back.
<code>Neo.ClientError.Transaction.TransactionNotFound</code>	The request referred to a transaction that does not exist.
<code>Neo.ClientError.Transaction.TransactionTimedOut</code>	The transaction has not completed within the specified timeout (<code>dbms.transaction.timeout</code>). You may want to retry with a longer timeout.
<code>Neo.ClientError.Transaction.TransactionValidationFailed</code>	Transaction changes did not pass validation checks.

Transient errors

This is a complete list of all transient error status codes Neo4j may return, and what they mean.

Status Code	Description
<code>Neo.TransientError.Cluster.ReplicationFailure</code>	Replication failure.
<code>Neo.TransientError.Database.DatabaseUnavailable</code>	The database is not currently available to serve your request, refer to the database logs for more details. Retrying your request at a later time may succeed.
<code>Neo.TransientError.General.MemoryPoolOutOfMemoryError</code>	The memory pool limit was exceeded. The corresponding setting can be found in the error message.
<code>Neo.TransientError.General.OutOfMemoryError</code>	<p>There is not enough memory to perform the current task. Please try increasing <code>dbms.memory.heap.max_size</code> in the Neo4j configuration file (normally in <code>conf/neo4j.conf</code> or, if you are using Neo4j Desktop, found through the user interface).</p> <p>If you are running an embedded installation, you can increase the heap by using the <code>-Xmx</code> command line flag, and then restart the database.</p>

Status Code	Description
<code>Neo.TransientError.General.StackOverflowError</code>	<p>There is not enough stack size to perform the current task. This is generally considered to be a database error, so please contact Neo4j support.</p> <p>You could try increasing the stack size; for example to set the stack size to 2M, add <code>dbms.jvm.additional=-Xss2M</code> to in the Neo4j configuration file (normally in <code>conf/neo4j.conf</code> or, if you are using Neo4j Desktop, found through the user interface).</p> <p>If you are running an embedded installation, you can add <code>-Xss2M</code> as command line flag.</p>
<code>Neo.TransientError.General.TransactionMemoryLimit</code>	<p>There is not enough memory to perform the current task. Please try increasing <code>dbms.memory.off_heap.max_size</code> in the Neo4j configuration file (normally in <code>conf/neo4j.conf</code> or, if you are using Neo4j Desktop, found through the user interface), and then restart the database.</p>
<code>Neo.TransientError.Request.NoThreadsAvailable</code>	<p>There are no available threads to serve this request at the moment. You can retry at a later time, or consider increasing max thread pool size for bolt connector(s).</p>
<code>Neo.TransientError.Security.AuthProviderFailed</code>	<p>An auth provider request failed.</p>
<code>Neo.TransientError.Security.AuthProviderTimeout</code>	<p>An auth provider request timed out.</p>
<code>Neo.TransientError.Security.ModifiedConcurrently</code>	<p>The user was modified concurrently to this request.</p>
<code>Neo.TransientError.Statement.ExecutionTimeout</code>	<p>The database was unable to execute the statement in a timely fashion.</p>
<code>Neo.TransientError.Transaction.BookmarkTimeout</code>	<p>Bookmark wait timed out. Database has not reached the specified version.</p>
<code>Neo.TransientError.Transaction.ConstraintsChanged</code>	<p>Database constraints changed since the start of this transaction.</p>
<code>Neo.TransientError.Transaction.DeadlockDetected</code>	<p>This transaction, and at least one more transaction, has acquired locks in a way that it will wait indefinitely, and the database has aborted it. Retrying this transaction will most likely be successful.</p>
<code>Neo.TransientError.Transaction.Interrupted</code>	<p>Interrupted while waiting.</p>
<code>Neo.TransientError.Transaction.LeaseExpired</code>	<p>The lease under which this transaction was started is no longer valid.</p>
<code>Neo.TransientError.Transaction.LockAcquisitionTimeout</code>	<p>Unable to acquire lock within configured timeout (<code>dbms.lock.acquisition.timeout</code>).</p>
<code>Neo.TransientError.Transaction.MaximumTransactionLimit Reached</code>	<p>Unable to start new transaction since the maximum number of concurrently executing transactions is reached (<code>dbms.transaction.concurrent.maximum</code>). You can retry at a later time or consider increasing allowed maximum of concurrent transactions.</p>

Status Code	Description
<code>Neo.TransientError.Transaction.Outdated</code>	Transaction has seen state which has been invalidated by applied updates while the transaction was active. Transaction may succeed if retried.

Database error

This is a complete list of all database error status codes Neo4j may return, and what they mean.

Status Code	Description
<code>Neo.DatabaseError.Database.DatabaseLimitReached</code>	The limit to number of databases has been reached.
<code>Neo.DatabaseError.Database.UnableToStartDatabase</code>	Unable to start database.
<code>Neo.DatabaseError.Database.Unknown</code>	Unknown database management error.
<code>Neo.DatabaseError.Fabric.RemoteExecutionFailed</code>	The database was unable to execute a remote part of the statement.
<code>Neo.DatabaseError.General.IndexCorruptionDetected</code>	The request (directly or indirectly) referred to an index that is in a failed state. The index needs to be dropped and recreated manually.
<code>Neo.DatabaseError.General.SchemaCorruptionDetected</code>	A malformed schema rule was encountered. Please contact your support representative.
<code>Neo.DatabaseError.General.StorageDamageDetected</code>	Expected set of files not found on disk. Please restore from backup.
<code>Neo.DatabaseError.General.UnknownError</code>	An unknown error occurred.
<code>Neo.DatabaseError.Schema.ConstraintCreationFailed</code>	Creating a requested constraint failed.
<code>Neo.DatabaseError.Schema.ConstraintDropFailed</code>	The database failed to drop a requested constraint.
<code>Neo.DatabaseError.Schema.IndexCreationFailed</code>	Failed to create an index.
<code>Neo.DatabaseError.Schema.IndexDropFailed</code>	The database failed to drop a requested index.
<code>Neo.DatabaseError.Schema.LabelAccessFailed</code>	The request accessed a label that did not exist.
<code>Neo.DatabaseError.Schema.PropertyKeyAccessFailed</code>	The request accessed a property that does not exist.
<code>Neo.DatabaseError.Schema.RelationshipTypeAccessFailed</code>	The request accessed a relationship type that does not exist.
<code>Neo.DatabaseError.Schema.SchemaRuleAccessFailed</code>	The request referred to a schema rule that does not exist.
<code>Neo.DatabaseError.Schema.SchemaRuleDuplicateFound</code>	The request referred to a schema rule that is defined multiple times.
<code>Neo.DatabaseError.Schema.TokenLimitReached</code>	The maximum number of tokens of this type has been reached, no more tokens of this type can be created.
<code>Neo.DatabaseError.Statement.ExecutionFailed</code>	The database was unable to execute the statement.
<code>Neo.DatabaseError.Statement.RemoteExecutionFailed</code>	The database was unable to execute a remote part of the statement.

Status Code	Description
<code>Neo.DatabaseError.Transaction.LinkedTransactionError</code>	The transaction was terminated because another transaction executing the same query encountered an error.
<code>Neo.DatabaseError.Transaction.TransactionCommitFailed</code>	The database was unable to commit the transaction.
<code>Neo.DatabaseError.Transaction.TransactionLogError</code>	The database was unable to write transaction to log.
<code>Neo.DatabaseError.Transaction.TransactionRollbackFailed</code>	The database was unable to roll back the transaction.
<code>Neo.DatabaseError.Transaction.TransactionStartFailed</code>	The database was unable to start the transaction.

Notification codes

`ClientNotification` codes are messages returned by the server with severity `WARNING` or `INFORMATION`. Contrary to error messages, notifications are only returned if the execution of the query is successful.

Notification code fields

The notification object, which is part of the result summary, consists of the following fields:

- **Code** — The status code in the form of `Neo.ClientNotification.[Category].[Title]`.
- **Title** — The description of the status code.
- **Description** — The description of a specific notification, which can contain details about the query being executed.
- **Severity** — The severity can be one of the following:
 - `WARNING`: There might be a problem with your query. Please, take a look.
 - `INFORMATION`: The query is correct, but this information can still be useful.
- **Category** — The category of the notification.
- **Position** — The position, given by row and column, where the notification is relevant in the query text.

Notification code categories

All notifications are grouped by category, which can have severity `WARNING`, `WARNING OR INFORMATION`, or `INFORMATION`.

The following categories exist, ordered by severity:

CATEGORY	SEVERITY	EXPLANATION	RECOMMENDED ACTION
<code>DEPRECATION</code>	<code>WARNING</code>	The query or command uses deprecated features that should be replaced.	Update to use the new functionality.
<code>HINT</code>	<code>WARNING</code>	The given hint cannot be satisfied.	Remove the hint or fix the query so the hint can be used.

CATEGORY	SEVERITY	EXPLANATION	RECOMMENDED ACTION
UNSUPPORTED	WARNING	The query or command is trying to use features not supported by the current system or using experimental features that should not be used in production.	Unsupported features cannot be trusted and should not be used in production.
UNRECOGNIZED	WARNING OR INFORMATION	The query or command mentions entities that are unknown to the system.	Make sure you have not misspelled the entity.
GENERIC	WARNING OR INFORMATION	Notifications that are not part of a wider class.	Depends on the specific notification.
PERFORMANCE	INFORMATION	The query uses costly operations and might be slow. Consider if it is possible to rewrite the query in a different way.	

List of all notification codes

The following are all Neo4j notifications, grouped by category, when they are returned, and an example of improvement.

- [PERFORMANCE category](#)
- [HINT category](#)
- [UNRECOGNIZED category](#)
- [UNSUPPORTED category](#)
- [DEPRECATION category](#)
- [GENERIC category](#)

PERFORMANCE category

Performance notifications are returned whenever the query uses costly operations and the performance may be improved by changing the query or adding an index.

Neo.ClientNotification.Statement.CartesianProduct


	<p>When is this notification returned?</p> <p>Whenever there is a cartesian product in the plan.</p>
---	--

Table 2. Notification category details

Code	Neo.ClientNotification.Statement.CartesianProduct
Title	This query builds a cartesian product between disconnected patterns.
Severity	INFORMATION
Category	PERFORMANCE

Example 1. Cartesian product

Query

```
MATCH (c:Child), (p:Parent) RETURN c, p
```

Description of the returned code

If a part of a query contains multiple disconnected patterns, this will build a cartesian product between all those parts. This may produce a large amount of data and slow down query processing. While occasionally intended, it may often be possible to reformulate the query that avoids the use of this cross product, perhaps by adding a relationship between the different parts or by using `OPTIONAL MATCH` (identifier is: `(p)`)

Suggestions for improvement

In case a cartesian product is needed, nothing can be done to improve this query. In many cases, however, you might not need a combination of all children and parents, and that is when this query could be improved. If for example, you need the children and the children's parents, you can improve this query by rewriting it to the following:

```
MATCH (c:Child)-[:ChildOf]->(p:Parent) RETURN c, p
```

Neo.ClientNotification.Statement.UnboundedVariableLengthPattern



When is this notification returned?

When there is no upper bound specified on the variable length relationship.

Table 3. Notification category details

Code	<code>Neo.ClientNotification.Statement.UnboundedVariableLengthPattern</code>
Title	The provided pattern is unbounded, consider adding an upper limit to the number of node hops.
Severity	<code>INFORMATION</code>
Category	<code>PERFORMANCE</code>

Example 2. Shortest path with an unbounded pattern

Query

```
MATCH p=shortestPath((n)-[*]->(m)) RETURN p
```

Description of the returned code

Using shortest path with an unbounded pattern will likely result in long execution times. It is recommended to use an upper limit to the number of node hops in your pattern.

Suggestions for improvement

If you have a big graph, this query might be very slow. Consider adding an upper limit.

```
MATCH p=shortestPath((n)-[*..8]->(m)) RETURN p
```

Neo.ClientNotification.Statement.ExhaustiveShortestPath



When is this notification returned?

When a predicate, given on the shortest path, needs to inspect the whole path before deciding whether it is valid, the shortest path might fall back to the exhaustive search algorithm. For more information, see [Cypher manual → Shortest path planning](#).

Table 4. Notification category details

Code	<code>Neo.ClientNotification.Statement.ExhaustiveShortestPath</code>
Title	Exhaustive shortest path has been planned for your query that means that shortest path graph algorithm might not be used to find the shortest path. Hence an exhaustive enumeration of all paths might be used in order to find the requested shortest path.
Severity	INFORMATION
Category	PERFORMANCE

Example 3. Exhaustive shortest path

```
MATCH p = shortestPath(()-[*..42]-())
WHERE ANY(n in nodes(p) WHERE n:Label)
RETURN p
```

Description of the returned code

Using shortest path with an exhaustive search fallback might cause query slow down since shortest path graph algorithms might not work for this use case. It is recommended to introduce a **WITH** to separate the **MATCH** containing the shortest path from the existential predicates on that path.

Suggestions for improvement

Separate the predicate by introducing a **WITH** after the **MATCH** clause.

```
MATCH p = shortestPath(()-[*..42]-())
WITH p
WHERE ANY(n in nodes(p) WHERE n:Label)
RETURN p
```

Neo.ClientNotification.Statement.NoApplicableIndex



When is this notification returned?

Given a larger number of nodes, **LOAD CSV** together with a **MATCH** or a **MERGE** clause may speed up the query if you add an index.

Table 5. Notification category details

Code	Neo.ClientNotification.Statement.NoApplicableIndex
Title	Adding a schema index may speed up this query.
Severity	INFORMATION
Category	PERFORMANCE

Example 4. Load CSV with MATCH or MERGE

Query

```
LOAD CSV FROM 'file:///ignore/ignore.csv' AS line WITH * MATCH (n:Person{name:line[0]}) RETURN line, n
```

Description of the returned code

Using `LOAD CSV` followed by a `MATCH` or `MERGE` that matches a non-indexed label will most likely not perform well on large data sets. Please consider using a schema index.

Suggestions for improvement

Create an index on the label and property you match.

```
CREATE INDEX FOR (n:Person) ON (n.name)
```

Neo.ClientNotification.Statement.EagerOperator



When is this notification returned?

When the execution plan for a query contains an eager operator.

Table 6. Notification category details

Code	<code>Neo.ClientNotification.Statement.EagerOperator</code>
Title	The execution plan for this query contains the Eager operator, which forces all dependent data to be materialized in main memory before proceeding
Severity	<code>INFORMATION</code>
Category	<code>PERFORMANCE</code>

Example 5. Load CSV with MATCH or MERGE

`LOAD CSV` together with an Eager operator can take up a lot of memory.

Query

```
LOAD CSV FROM 'file:///ignore/ignore.csv' AS line MATCH (n:Person{name:line[0]}) DELETE n RETURN line
```

Description of the returned code

Using `LOAD CSV` with a large data set in a query where the execution plan contains the Eager operator could potentially consume a lot of memory and is likely to not perform well. See the Neo4j Manual entry on the Eager operator for more information and hints on how problems could be avoided.

Suggestions for improvement

See the [Cypher Manual → Eager operator](#) for more information and hints on how to avoid problems. In this specific case, the query could be rewritten to the following:

```
LOAD CSV FROM 'file:///ignore/ignore.csv' AS line
CALL {
  WITH line
  MATCH (n:Person{name:line[0]}) DELETE n
}
RETURN line
```

Neo.ClientNotification.Statement.DynamicProperty

Table 7. Notification category details

Code	<code>Neo.ClientNotification.Statement.DynamicProperty</code>
Title	Queries using dynamic properties will use neither index seeks nor index scans for those properties
Severity	INFORMATION
Category	PERFORMANCE

Example 6. Using a dynamic node property key makes it impossible to use indexes

Query

```
MATCH (n:Person) WHERE n[$prop] IS NOT NULL RETURN n;
```

Description of the returned code

Did not supply query with enough parameters. The produced query plan will not be cached and is not executable without **EXPLAIN**. (Missing parameters: **prop**)

Suggestions for improvement

If there is an index for **(n:Person) ON (n.name)**, it will not be used for the above query because the query is using a dynamic property. Therefore, if there is an index, it is better to use the constant value. For example, if **prop** is equal to **name**, the following query would be able to use the index:

```
MATCH (n:Person) WHERE n.name IS NOT NULL RETURN n;
```

Example 7. Using dynamic relationship property key makes it impossible to use indexes

Query

```
MATCH ()-[r: KNOWS]->() WHERE r[$prop] IS NOT NULL RETURN r
```

Description of the returned code

Did not supply query with enough parameters. The produced query plan will not be cached and is not executable without **EXPLAIN**. (Missing parameters: **prop**)

Suggestions for improvement

Similar to dynamic node properties, use a constant value if possible, especially when there is an index on the relationship property. For example, if **\$prop** is equal to **since**, you can rewrite the query to:

```
MATCH ()-[r: KNOWS]->() WHERE r.since IS NOT NULL RETURN r
```

Neo.ClientNotification.Statement.CodeGenerationFailed

Table 8. Notification category details

Code	Neo.ClientNotification.Statement.CodeGenerationFailed
Title	The database was unable to generate code for the query. A stack trace can be found in the debug.log.
Severity	INFORMATION

Category	PERFORMANCE
----------	-------------

The `CodeGenerationFailed` notification is created when it is not possible to generate a code for a query, for example, when the query is too big. To find more information about the specific query, see the stack trace in the `debug.log` file.

HINT category

HINT notifications are returned by default when the Cypher planner or runtime cannot create a query plan to fulfill a specified hint, for example, **JOIN** or **INDEX**. This behavior of the Cypher planner or runtime can be changed by setting the configuration `dbms.cypher.hints_error` to `true`, in which case, the query will return an error instead.

Neo.ClientNotification.Statement.JoinHintUnfulfillableWarning

Table 9. Notification category details

Code	Neo.ClientNotification.Statement.JoinHintUnfulfillableWarning
Title	The database was unable to plan a hinted join.
Severity	WARNING
Category	HINT

Example 8. A **JOIN** hint was given, but it was not possible to fulfill the hint.

Query

```
MATCH (a:A)
WITH a, 1 AS horizon
OPTIONAL MATCH (a)-[r]->(b:B)
USING JOIN ON a
OPTIONAL MATCH (a)--(c)
RETURN *
```

Description of the returned code

The hinted join was not planned. This could happen because no generated plan contained the join key, please try using a different join key or restructure your query. (hinted join key identifier is: `a`)

Suggestions for improvement

The join hint cannot be fulfilled because the given **JOIN** variable was introduced before the optional match and is therefore already bound. The only option for this query is to remove the hint or change the query so it is possible to use the hint.

Neo.ClientNotification.Schema.HintedIndexNotFound

Table 10. Notification category details

Code	Neo.ClientNotification.Schema.HintedIndexNotFound
Title	The request (directly or indirectly) referred to an index that does not exist.
Severity	WARNING
Category	HINT

Example 9. An index hint was given, but it was not possible to use the index.

Query

```
MATCH (a: Label)
USING INDEX a:Label(id)
WHERE a.id = 1
RETURN a
```

Description of the returned code

The hinted index does not exist, please check the schema (index is: INDEX FOR (a:`Label`) ON (a.id))

Suggestions for improvement

The hinted index does not exist, make sure the label and property are spelled correctly. If the spelling is correct, either create the index or remove the hint from the query.

Example 10. A relationship index hint was given, but it was not possible to use the index

Query

```
MATCH ()-[r:Rel]-()
USING INDEX r:Rel(id)
WHERE r.id = 1
RETURN r
```

Description of the returned code

The hinted index does not exist, please check the schema (index is: INDEX FOR ()-[r:`Rel`]-() ON (r.id))

Suggestions for improvement

The hinted index does not exist, make sure the label and property are spelled correctly. If the spelling is correct, either create the index or remove the hint from the query.

UNRECOGNIZED category

A notification has the unrecognized category if the query or command mentions entities that are unknown to the system.

Neo.ClientNotification.Database.HomeDatabaseNotFound

Table 11. Notification category details

Code	Neo.ClientNotification.Database.HomeDatabaseNotFound
Title	The request referred to a home database that does not exist.
Severity	INFORMATION
Category	UNRECOGNIZED

Example 11. Set the **home** database to a database that does not yet exist.

Query

```
CREATE USER linnea SET PASSWORD "password" SET HOME DATABASE NonExistingDatabase
```

Description of the returned code

The home database provided does not currently exist in the DBMS. This command will not take effect until this database is created. (HOME DATABASE: **nonexistingdatabase**)

Suggestions for improvement

Make sure the home database has not been misspelled.

Neo.ClientNotification.Statement.UnknownLabelWarning

Table 12. Notification category details

Code	Neo.ClientNotification.Statement.UnknownLabelWarning
Title	The provided label is not in the database.
Severity	WARNING
Category	UNRECOGNIZED

Example 12. Match on a node with a label that does not exist in the database.

Query

```
MATCH (n:Perso) RETURN n
```

Description of the returned code

One of the labels in your query is not available in the database, make sure you didn't misspell it or that the label is available when you run this statement in your application (the missing label name is: `Perso`)

Suggestions for improvement

Make sure you didn't misspell the label. If nodes with the given label will be created in the future, then no change is needed.

Neo.ClientNotification.Statement.UnknownRelationshipTypeWarning

Table 13. Notification category details

Code	Neo.ClientNotification.Statement.UnknownRelationshipTypeWarning
Title	The provided relationship type is not in the database.
Severity	WARNING
Category	UNRECOGNIZED

Example 13. Match on a relationship, where there are no relationships in the database with the given relationship type.

Query

```
MATCH (n)-[:NonExistingType]->() RETURN n
```

Description of the returned code

One of the relationship types in your query is not available in the database, make sure you didn't misspell it or that the label is available when you run this statement in your application (the missing relationship type is: `NonExistingType`)

Suggestions for improvement

Make sure you did not misspell the relationship type. If relationships will be created, with the given relationship type, in the future, then no change to the query is needed.

Neo.ClientNotification.Statement.UnknownPropertyKeyWarning

Table 14. Notification category details

Code	Neo.ClientNotification.Statement.UnknownPropertyKeyWarning
Title	The provided property key is not in the database
Severity	WARNING
Category	UNRECOGNIZED

Example 14. Match on a property key that does not exist.

Query

```
MATCH (n {nonExistingProp:1}) RETURN n
```

Description of the returned code

One of the property names in your query is not available in the database, make sure you didn't misspell it or that the label is available when you run this statement in your application (the missing property name is: `nonExistingProp`)

Suggestions for improvement

Make sure you didn't misspell the property key. If the property key will be created in the future, then no change is needed to the query.

UNSUPPORTED category

Notifications with the unsupported category are created if the query or command is trying to use features that are not supported by the current system or using experimental features that should not be used in production.

Neo.ClientNotification.Statement.RuntimeUnsupportedWarning

Table 15. Notification category details

Code	Neo.ClientNotification.Statement.RuntimeUnsupportedWarning
Title	This query is not supported by the chosen runtime.
Severity	WARNING
Category	UNSUPPORTED

Example 15. The chosen runtime is not supported for the given query

Query

```
CYPHER runtime=pipelined SHOW INDEXES YIELD *
```

Description of the returned code

Selected runtime is unsupported for this query, please use a different runtime instead or fallback to default. (Pipelined does not yet support the plans including `ShowIndexes`, use another runtime.)

Suggestions for improvement

Use a different runtime or remove the runtime option to run the query with the default runtime:

```
SHOW INDEXES YIELD *
```

Neo.ClientNotification.Statement.RuntimeExperimental

Table 16. Notification category details

Code	Neo.ClientNotification.Statement.RuntimeExperimental
Title	This feature is experimental and should not be used in production systems.
Severity	WARNING
Category	UNSUPPORTED

Example 16. Use of the parallel runtime

Query

```
CYPHER runtime=parallel MATCH (n) RETURN (n)
```

Description of the returned code

You are using an experimental feature (The parallel runtime is experimental and might suffer from instability and potentially correctness issues.)

Suggestions for improvement

The parallel runtime should not be used in production. Choose another runtime or remove the option to use the default runtime:

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

DEPRECATION category

Notifications within the deprecation category contain information about a feature or functionality which

has been deprecated. It is important to change to the new functionality, otherwise, the query might break in a future version.

Neo.ClientNotification.Statement.FeatureDeprecationWarning

Table 17. Notification category details

Code	Neo.ClientNotification.Statement.FeatureDeprecationWarning
Title	This feature is deprecated and will be removed in future versions.
Severity	WARNING
Category	DEPRECATION

Example 17. Repeated variable length relationship variable inside the same pattern, with same relationship type (when run on a 5.x release)

Query

```
MATCH ()-[r:PARENT*]-()  
MATCH ()-[r:PARENT*]-()  
RETURN *
```

Description of the returned code

Using an already bound variable for a variable length relationship is deprecated and will be removed in a future version. (the repeated variable is: `r`)

Suggestions for improvement

Inside a single pattern, we have relationship uniqueness, which means that this query never will return any rows and should therefore be rewritten to the wanted behavior.

Example 18. Repeated variable length relationship across patterns, with the same relationship type

Query

```
MATCH ({{name:"Molly"}})-[r:KNOWS*]->({name:"Kalle"})
MATCH ({{age:25}})-[r:KNOWS*]->({age:21})
RETURN r
```

Description of the returned code

Using an already bound variable for a variable length relationship is deprecated and will be removed in a future version. (the repeated variable is: *r*)

Suggestions for improvement

Since the relationships in the first and second match must be the same, we know that the start node and the end node are the same. In the query above, we could therefore concatenate the predicates from the two matches and create a single match:

```
MATCH ({{name:"Molly", age:25}})-[r:KNOWS*]->({name:"Kalle", age:21})
RETURN r
```

Example 19. Repeated variable length relationship across patterns with different relationship types

Query

```
MATCH ()-[r:PARENT*]-()
MATCH ()-[r:CHILD*]-()
RETURN *
```

Description of the returned code

Using an already bound variable for a variable length relationship is deprecated and will be removed in a future version. (the repeated variable is: *r*)

Suggestions for improvement

Since the relationships in the first and second match must be the same, and a single relationship can't have multiple relationship types, this query will never return any rows.

Example 20. Colon after the | in a relationship pattern

Query

```
MATCH (a)-[:A|:B|:C]-() RETURN *
```

Description of the returned code

The semantics of using colon in the separation of alternative relationship types will change in a future version. (Please use ':A|B|C' instead)

Suggestions for improvement

Remove the colon inside the relationship type expression.

```
MATCH (a)-[:A|B|C]-() RETURN *
```

Example 21. Setting properties using a node

Query

```
MATCH (a)-[]-(b)  
SET a = b
```

Description of the returned code

The use of nodes or relationships for setting properties is deprecated and will be removed in a future version. Please use `properties()` instead.

Suggestions for improvement

Use the `properties()` function to get all properties from `b`.

```
MATCH (a)-[]-(b)  
SET a = properties(b)
```

Example 22. Setting properties using a relationship

Query

```
MATCH (a)-[r]-(b)
SET a += r
```

Description of the returned code

The use of nodes or relationships for setting properties is deprecated and will be removed in a future version. Please use `properties()` instead.

Suggestions for improvement

Use the `properties()` function to get all properties from `r`

```
MATCH (a)-[r]-(b)
SET a += properties(r)
```

Example 23. Shortest path with a fixed relationship length

Query

```
MATCH (a:Start), shortestPath((a)-[r]->()) RETURN a
```

Description of the returned code

The use of `shortestPath` and `allShortestPaths` with fixed length relationships is deprecated and will be removed in a future version. Please use a path with a length of 1 `[r*1..1]` instead or a `Match` with a `limit`.

Suggestions for improvement

If the relationship length is fixed, there is no reason to search for the shortest path. Instead, you can rewrite it to the following:

```
MATCH (a: Start)-[r]->(b: End) RETURN b LIMIT 1
```

Example 24. Create a database with an unescaped name containing a dot

Query

```
CREATE DATABASE foo.bar
```

Description of the returned code

Databases and aliases with unescaped `.` are deprecated unless to indicate that they belong to a composite database. Names containing `.` should be escaped. (Name: `foo.bar`)

Suggestions for improvement

If not intended for a composite database, escape the name with the character ```.

```
CREATE DATABASE `foo.bar`
```

Example 25. A procedure has been deprecated or renamed

```
CALL unsupported.dbms.shutdown
```

Description of the returned code

The query used a deprecated procedure. (`'unsupported.dbms.shutdown'` is no longer supported)

Suggestions for improvement

Remove the use of the deprecated procedure. If there is a suggested replacement, update to use the replacement instead.

Example 26. Using a deprecated runtime option

Query

```
CYPHER runtime = interpreted MATCH (n) RETURN n
```

Description of the returned code

The query used a deprecated runtime option. (`'runtime=interpreted'` is deprecated, please use `'runtime=slotted'` instead)

Suggestions for improvement

Runtime `interpreted` is deprecated, use another runtime or remove the runtime option to use the default runtime.

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

Example 27. Using the `text-1.0` index provider when creating a text index

Query

```
CREATE TEXT INDEX FOR (n:Label) ON (n.prop) OPTIONS {indexProvider : 'text-1.0'}
```

Description of the returned code

The `text-1.0` provider for text indexes is deprecated and will be removed in a future version. Please use `text-2.0` instead.

Suggestions for improvement

Update the option `indexProvider` with the value `text-2.0`.

```
CREATE TEXT INDEX FOR (n:Label) ON (n.prop) OPTIONS {indexProvider : 'text-2.0'}
```

Neo.ClientNotification.Request.DeprecatedFormat

Table 18. Notification category details

Code	<code>Neo.ClientNotification.Request.DeprecatedFormat</code>
Title	The client requested a deprecated format.
Severity	<code>WARNING</code>
Category	<code>DEPRECATION</code>

GENERIC category

`GENERIC` notification codes do not belong to any wider category and do not have any connection to each other.

Neo.ClientNotification.Statement.SubqueryVariableShadowing

Table 19. Notification category details

Code	<code>Neo.ClientNotification.Statement.SubqueryVariableShadowing</code>
Title	Variable in subquery is shadowing a variable with the same name from the outer scope.
Severity	<code>INFORMATION</code>
Category	<code>GENERIC</code>

Example 28. Shadowing of a variable from the outer scope

Query

```
MATCH (n)
CALL {
  MATCH (n)--(m)
  RETURN m
}
RETURN *
```

Description of the returned code

Variable in subquery is shadowing a variable with the same name from the outer scope. If you want to use that variable instead, it must be imported into the subquery using importing **WITH** clause. (the shadowing variable is: **n**)

Suggestions for improvement

If the intended behavior of the query is for the variable in the subquery to be a new variable, then nothing needs to be done. If the intended behavior is to use the variable from the outer query, it needs to be imported to the subquery using the with clause.

```
MATCH (n)
CALL {
  WITH n
  MATCH (n)--(m)
  RETURN m
}
RETURN *
```

Neo.ClientNotification.Statement.ParameterNotProvided

Table 20. Notification category details

Code	Neo.ClientNotification.Statement.ParameterNotProvided
Title	The statement refers to a parameter that was not provided in the request.
Severity	WARNING
Category	GENERIC

Example 29. Using an **EXPLAIN** query with parameters without providing them

Query

```
EXPLAIN WITH $param as param RETURN param
```

Description of the returned code

Did not supply query with enough parameters. The produced query plan will not be cached and is not executable without **EXPLAIN**. (Missing parameters: **param**)

Suggestions for improvement

Provide the parameter to be able to cache the plan.

Neo.ClientNotification.Procedure.ProcedureWarning

Table 21. Notification category details

Code	Neo.ClientNotification.Procedure.ProcedureWarning
Title	The query used a procedure that generated a warning.
Severity	WARNING
Category	GENERIC

Changes to status codes per Neo4j version

Neo4j 5.0

New:

```
neo.ClientNotification.Database.HomeDatabaseNotFound  
Neo.ClientNotification.Request.DeprecatedFormat  
neo.ClientNotification.Schema.HintedIndexNotFound  
neo.ClientNotification.Statement.CartesianProduct  
neo.ClientNotification.Statement.CodeGenerationFailed  
neo.ClientNotification.Statement.DynamicProperty  
neo.ClientNotification.Statement.EagerOperator  
neo.ClientNotification.Statement.ExhaustiveShortestPath  
neo.ClientNotification.Statement.NoApplicableIndex  
neo.ClientNotification.Statement.RuntimeExperimental  
neo.ClientNotification.Statement.SubqueryVariableShadowing  
neo.ClientNotification.Statement.UnboundedVariableLengthPattern
```

Removed:

```
neo.ClientError.Statement.CodeGenerationFailed
neo.ClientNotification.Statement.CartesianProductWarning
neo.ClientNotification.Statement.DynamicPropertyWarning
neo.ClientNotification.Statement.EagerOperatorWarning
neo.ClientNotification.Statement.ExhaustiveShortestPathWarning
neo.ClientNotification.Statement.ExperimentalFeature
Neo.ClientNotification.Statement.MissingAlias
neo.ClientNotification.Statement.NoApplicableIndexWarning
neo.ClientNotification.Statement.SubqueryVariableShadowingWarning
neo.ClientNotification.Statement.UnboundedVariableLengthPatternWarning
```

Neo4j 4.4

New:

```
Neo.ClientError.Statement.UnsupportedAdministrationCommand
Neo.DatabaseError.Transaction.LinkedTransactionError
```

Neo4j 4.3

New:

```
Neo.ClientError.General.UpgradeRequired
Neo.TransientError.Statement.ExecutionTimeout
```

Neo4j 4.2

New:

```
Neo.ClientNotification.Statement.MissingAlias
Neo.ClientNotification.Statement.SubqueryVariableShadowingWarning
Neo.TransientError.Transaction.TransientTransactionFailure
```

Neo4j 4.1

New:

```
Neo.ClientError.Cluster.Routing
Neo.ClientError.General.TransactionOutOfMemoryError
Neo.ClientError.Statement.AccessMode
Neo.DatabaseError.Statement.RemoteExecutionFailed
Neo.TransientError.General.MemoryPoolOutOfMemoryError
```

Removed:

```
Neo.ClientError.Cluster.TransactionSizeExceedsLimit
Neo.DatabaseError.Cluster.SerializationFailure
Neo.TransientError.General.TransactionOutOfMemoryError
```

Neo4j 4.0

New:

```
Neo.ClientError.Cluster.TransactionSizeExceedsLimit
Neo.ClientError.Database.DatabaseNotFound
Neo.ClientError.Database.ExistingDatabaseFound
Neo.ClientError.Fabric.AccessMode
Neo.ClientError.Schema.ConstraintViolation
Neo.ClientError.Schema.ConstraintWithNameAlreadyExists
Neo.ClientError.Schema.EquivalentSchemaRuleAlreadyExists
Neo.ClientError.Schema.IndexMultipleFound
Neo.ClientError.Schema.IndexWithNameAlreadyExists
Neo.ClientError.Statement.NotSystemDatabaseError
Neo.ClientError.Statement.RuntimeUnsupportedError
Neo.ClientError.Transaction.InvalidBookmarkMixture
Neo.DatabaseError.Cluster.SerializationFailure
Neo.DatabaseError.Database.DatabaseLimitReached
Neo.DatabaseError.Database.UnableToStartDatabase
Neo.DatabaseError.Database.Unknown
Neo.DatabaseError.Fabric.RemoteExecutionFailed
Neo.DatabaseError.Schema.TokenLimitReached
Neo.DatabaseError.Statement.CodeGenerationFailed
Neo.TransientError.Database.DatabaseUnavailable
Neo.TransientError.General.TransactionMemoryLimit
Neo.TransientError.General.TransactionOutOfMemoryError
Neo.TransientError.Transaction.BookmarkTimeout
Neo.TransientError.Transaction.LeaseExpired
Neo.TransientError.Transaction.MaximumTransactionLimitReached
```

Removed:

```
Neo.ClientError.LegacyIndex.LegacyIndexNotFound
Neo.ClientError.Request.TransactionRequired
Neo.ClientError.Schema.ConstraintVerificationFailed
Neo.ClientError.Security.EncryptionRequired
Neo.ClientError.Statement.LabelNotFound
Neo.ClientNotification.Statement.CreateUniqueUnavailableWarning
Neo.ClientNotification.Statement.JoinHintUnsupportedWarning
Neo.ClientNotification.Statement.PlannerUnavailableWarning
Neo.ClientNotification.Statement.PlannerUnsupportedWarning
Neo.DatabaseError.Schema.LabelLimitReached
Neo.TransientError.Cluster.NoLeaderAvailable
Neo.TransientError.General.DatabaseUnavailable
Neo.TransientError.Network.CommunicationError
Neo.TransientError.Schema.SchemaModifiedConcurrently
Neo.TransientError.Transaction.InstanceStateChanged
Neo.TransientError.Transaction.LockSessionExpired
```

License

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

You are free to

Share

copy and redistribute the material in any medium or format

Adapt

remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms

Attribution

You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial

You may not use the material for commercial purposes.

ShareAlike

If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

See <https://creativecommons.org/licenses/by-nc-sa/4.0/> for further details. The full license text is available at <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>.