Missing domain concept

What are we going to do?

- Profile queries
- Introduce a missing concept in the domain
- Constraints and indexes
- Write our first refactoring query

Start playing the next guide….

…if you aren't playing it already

▷ Flight as a first class citizen

```
:play http://guides.neo4j.com/modeling_airports/02_flight.html
```

Profiling queries

How do I profile a query?

By prefixing the query with:

`EXPLAIN`

shows the execution plan without actually executing it or returning any results.

How do I profile a query?

By prefixing the query with:

```
EXPLAIN
```

shows the execution plan without actually executing it or returning any results.

```
PROFILE
```

executes the statement and returns the results along with profiling information.

How do I profile a query?

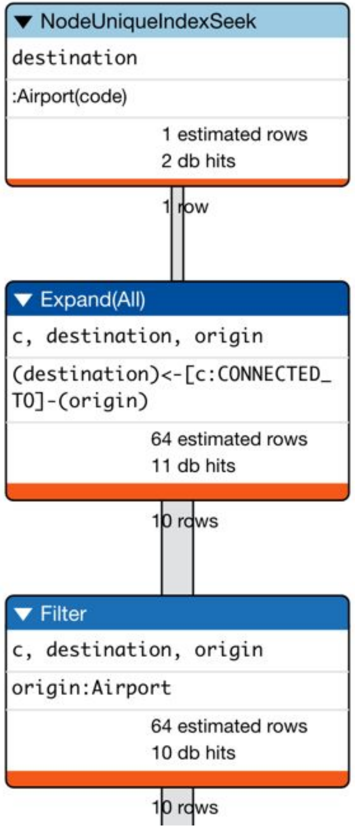
```
EXPLAIN
MATCH (origin:Airport)-[c:CONNECTED_TO]
     ->(destination:Airport)
WHERE c.code = "LAS"
RETURN origin, destination, c
LIMIT 10
```

How do I profile a query?

```
PROFILE
MATCH (origin:Airport)-[c:CONNECTED_TO]
     ->(destination:Airport)
WHERE destination.code = "LAS"
RETURN origin, destination, c
LIMIT 10
```

Anatomy of an execution plan

```
PROFILE
MATCH (origin:Airport)-[c:CONNECTED_TO]
     ->(destination:Airport)
WHERE destination.code = "LAS"
RETURN origin, destination, c
LIMIT 10
```



▼ NodeUniqueIndexSeek

destination

:Airport(code)

1 estimated rows
2 db hits

1 row

▼ Expand(All)

c, destination, origin

(destination)<-[c:CONNECTED_TO]-(origin)

64 estimated rows
11 db hits

10 rows

▼ Filter

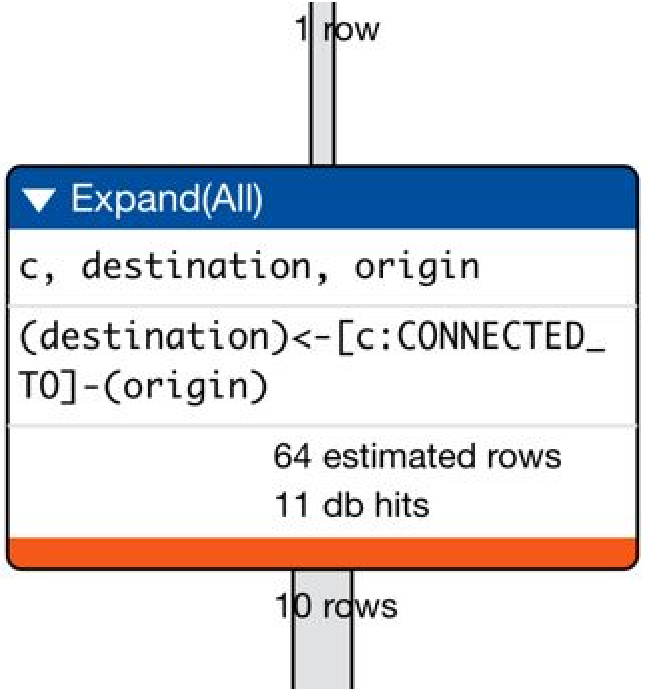
c, destination, origin

origin:Airport

64 estimated rows
10 db hits

10 rows

Operations



1 row

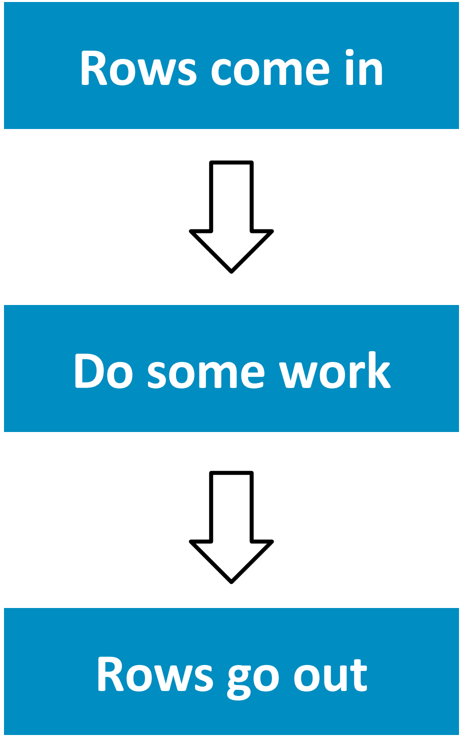
▼ Expand(All)

c, destination, origin

(destination)<-[c:CONNECTED_TO]-(origin)

64 estimated rows
11 db hits

10 rows



Rows come in

Do some work

Rows go out

What’s our goal?

At a high level, the goal is simple: **get the number of db hits down.**

What is a database hit?

> “an abstract unit of storage engine work.”

Let’s profile some queries

Continue playing the guide in your browser

Introducing a missing domain concept

Introducing a missing domain concept

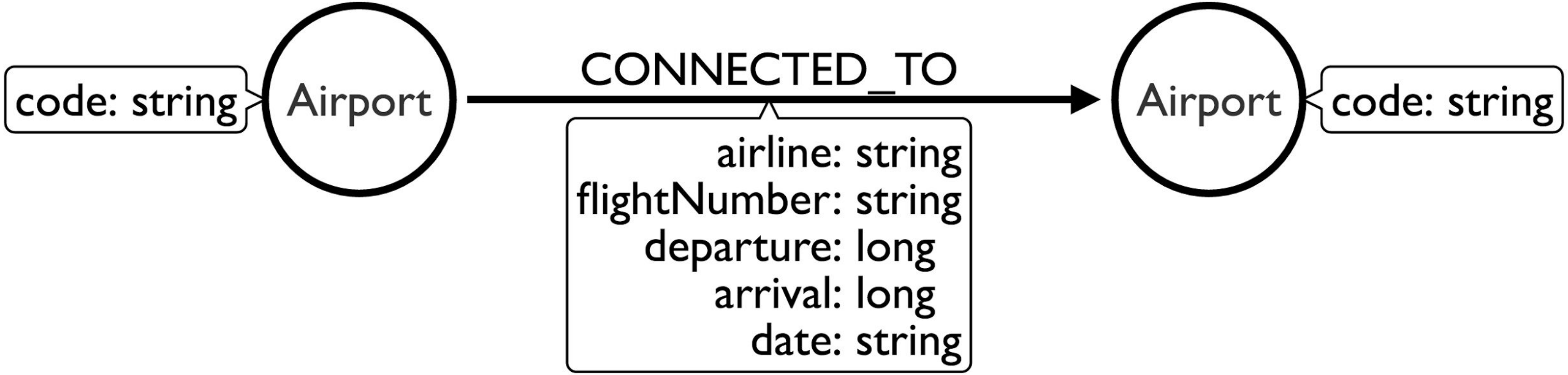
Sometimes we design a model and after working with it for a bit we realise that we’ve missed an important domain concept.

Introducing a missing domain concept

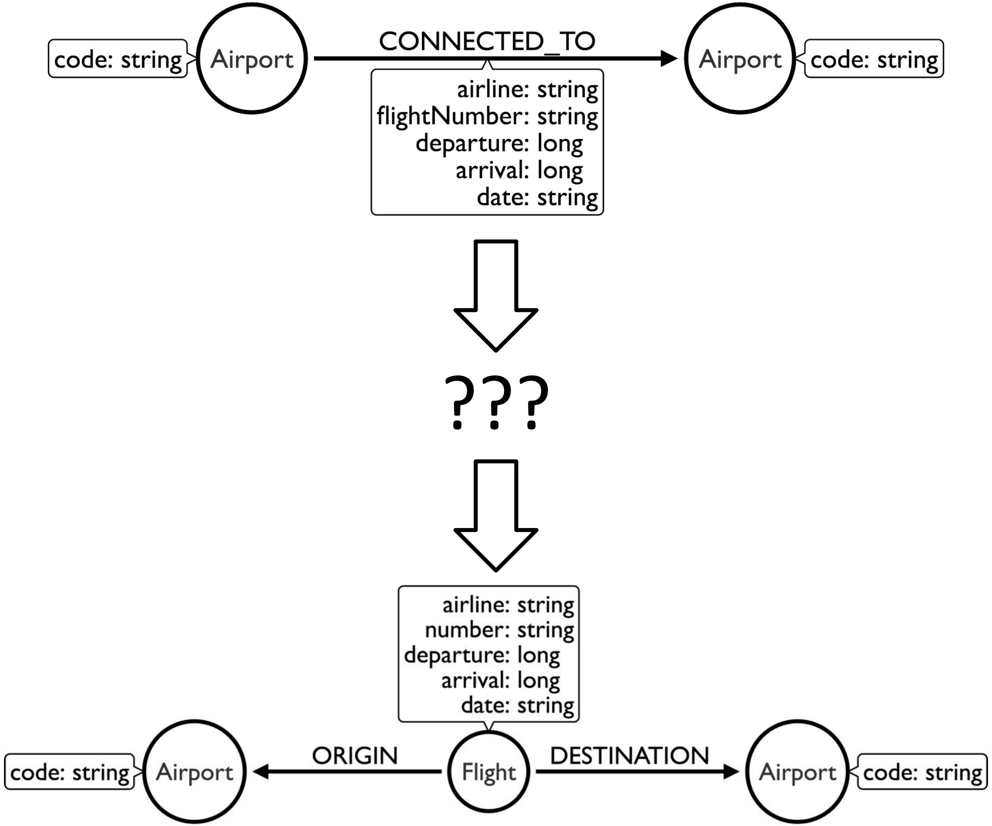
Sometimes we design a model and after working with it for a bit we realise that we’ve missed an important domain concept.

In our case we’re missing nodes to represent **Flights**

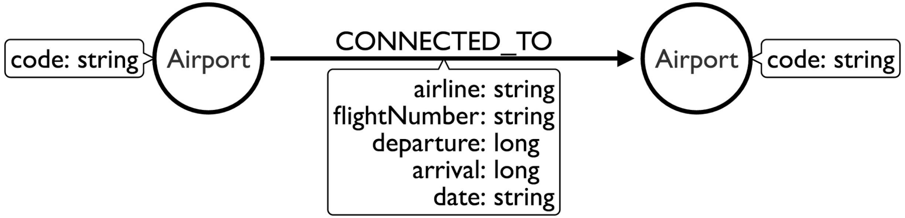
Introducing a missing domain concept



Flight as a first class citizen



Flight as a first class citizen



code: string

Airport

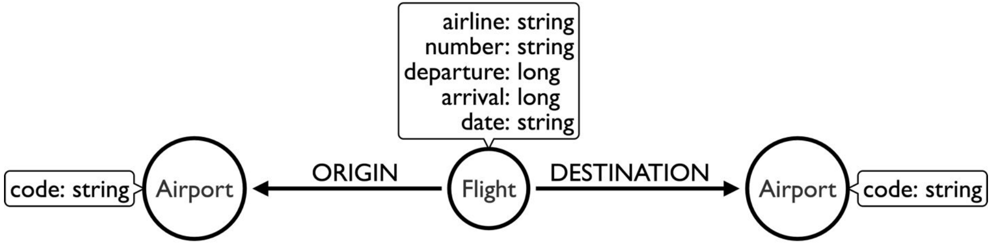
CONNECTED_TO

airline: string
flightNumber: string
departure: long
arrival: long
date: string

Airport

code: string

```
MATCH  (o:Airport)-[:CONNECTED_TO]->(d:Airport)
MERGE...
```



airline: string
number: string
departure: long
arrival: long
date: string

code: string

Airport

ORIGIN

Flight

DESTINATION

Airport

code: string

Refactoring: Derive node from relationship

This is the first of the refactoring patterns that we'll encounter today.

We have this pattern:

```
(origin)-[:CONNECTED_TO]->(destination)
```

And we'll create a new node using the properties of the `CONNECTED_TO` relationship:

```
(origin)<-[:ORIGIN]-(flight)-[:DESTINATION]->(destination)
```

Constraints and indexes

Unique Constraints

We create unique constraints to:

- ensure uniqueness
- allow fast lookup of nodes which match label-property pairs.

Unique Constraints

We create unique constraints to:

- ensure uniqueness
- allow fast lookup of nodes which match label-property pairs.

```
CREATE CONSTRAINT ON (label:Label)
ASSERT label.property IS UNIQUE
```

Indexes

We create indexes to:

- allow fast lookup of nodes which match label-property pairs.

Indexes

We create indexes to:

- allow fast lookup of nodes which match label-property pairs.

```
CREATE INDEX ON :Label(property)
```

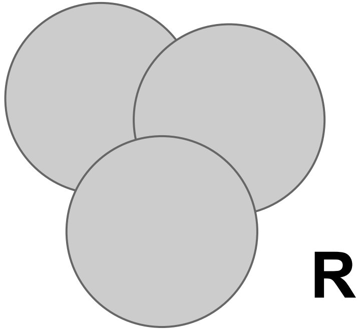
What gets indexed?

The following are index backed:

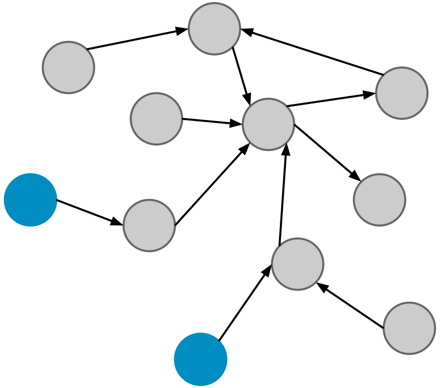
- Equality
- `STARTS WITH`
- `CONTAINS`
- `ENDS WITH`
- Range searches
- (Non-) existence checks

How are indexes used in neo4j?

Indexes are **only** used to find the starting point for queries.



Relational



Graph

Use index scans to look up rows in tables and join them with rows from other tables

Use indexes to find the starting points for a query.

Let’s get on with the refactoring

Continue playing the guide in your browser

End of Module
Missing Concept

Questions?