

Modeling Guidelines



What are we going to do?



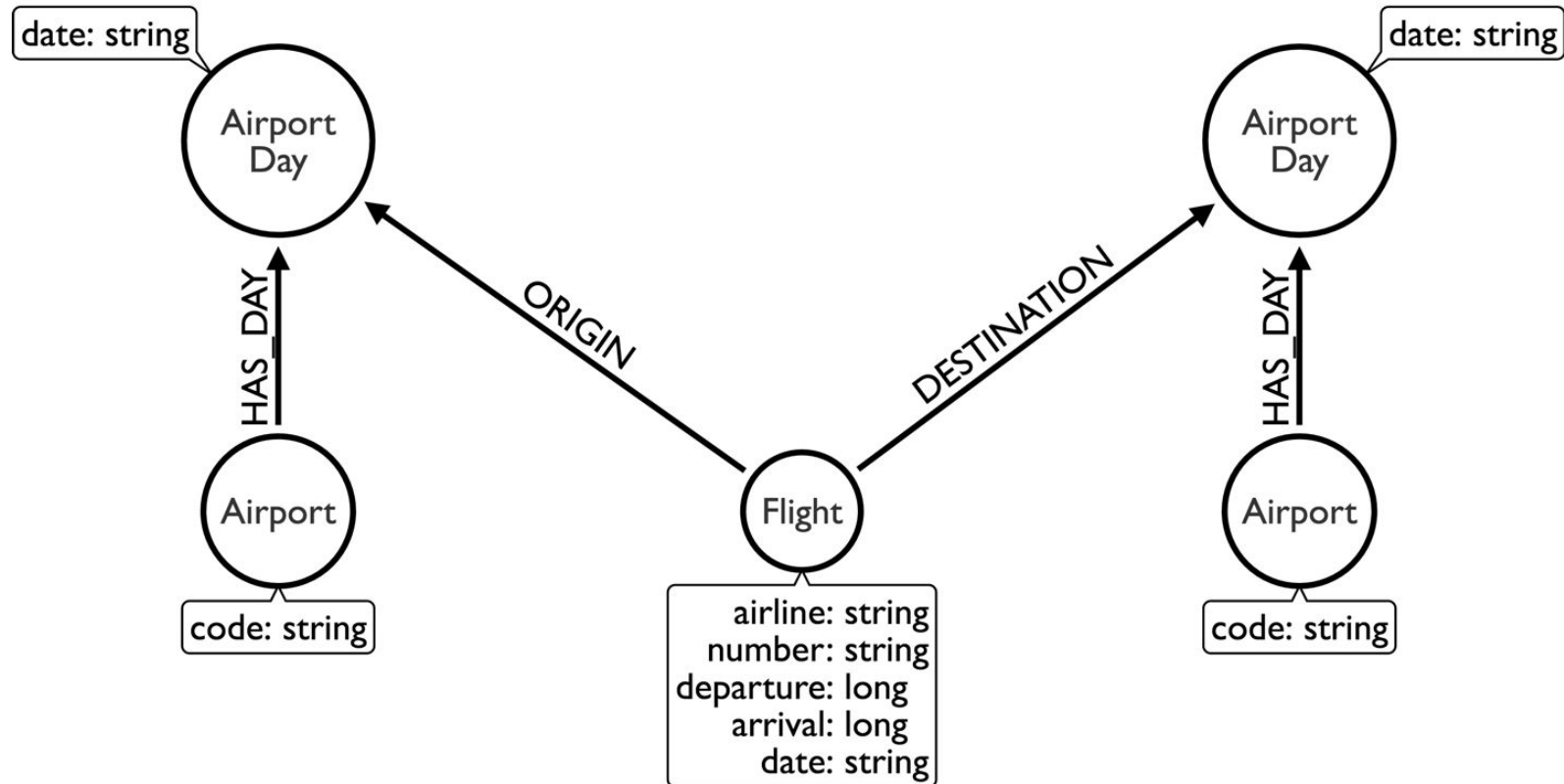
- Recap of the Property Graph Model
- Modeling choices
- Refactorings



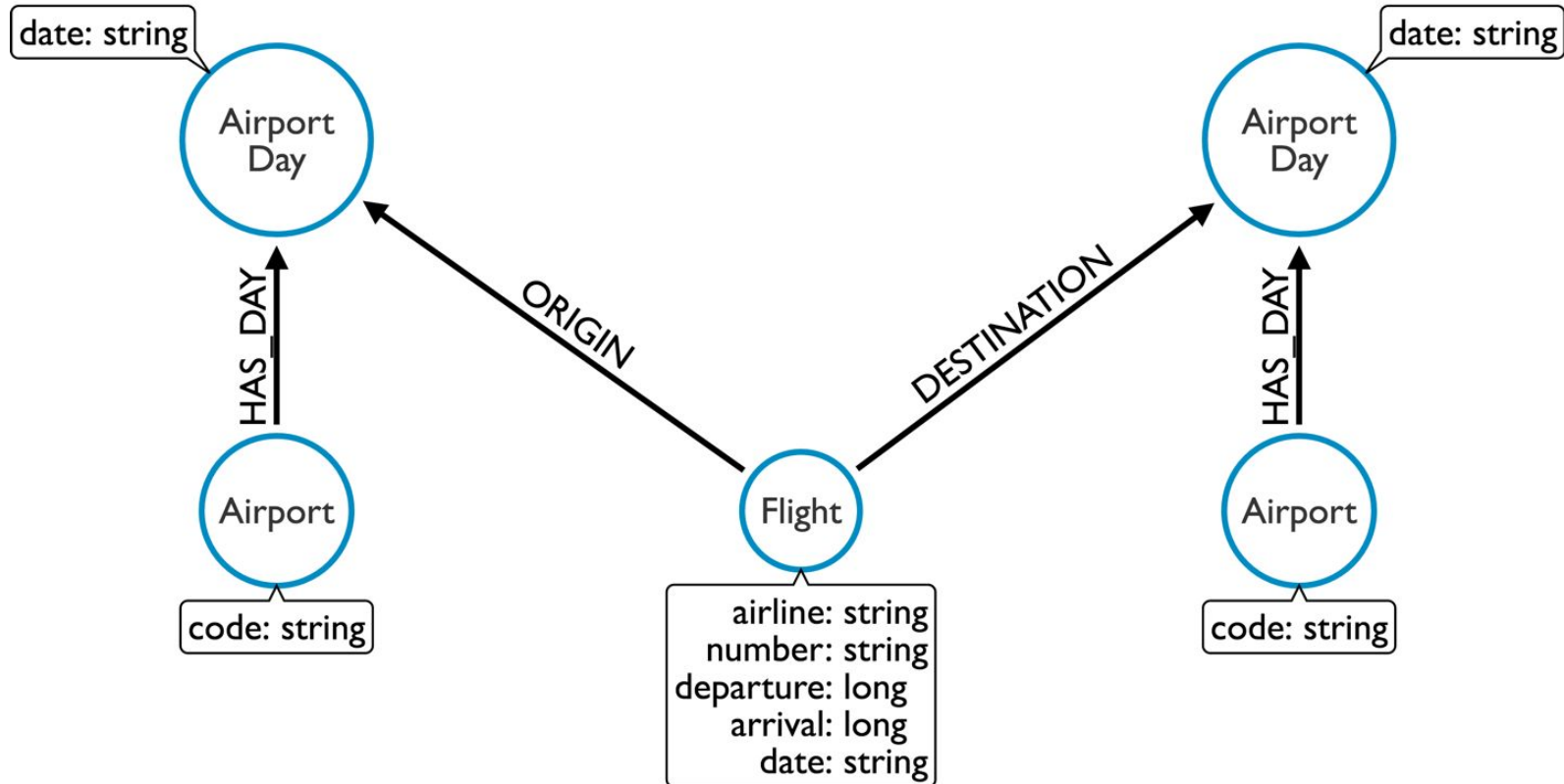
Nodes, relationships, labels, properties



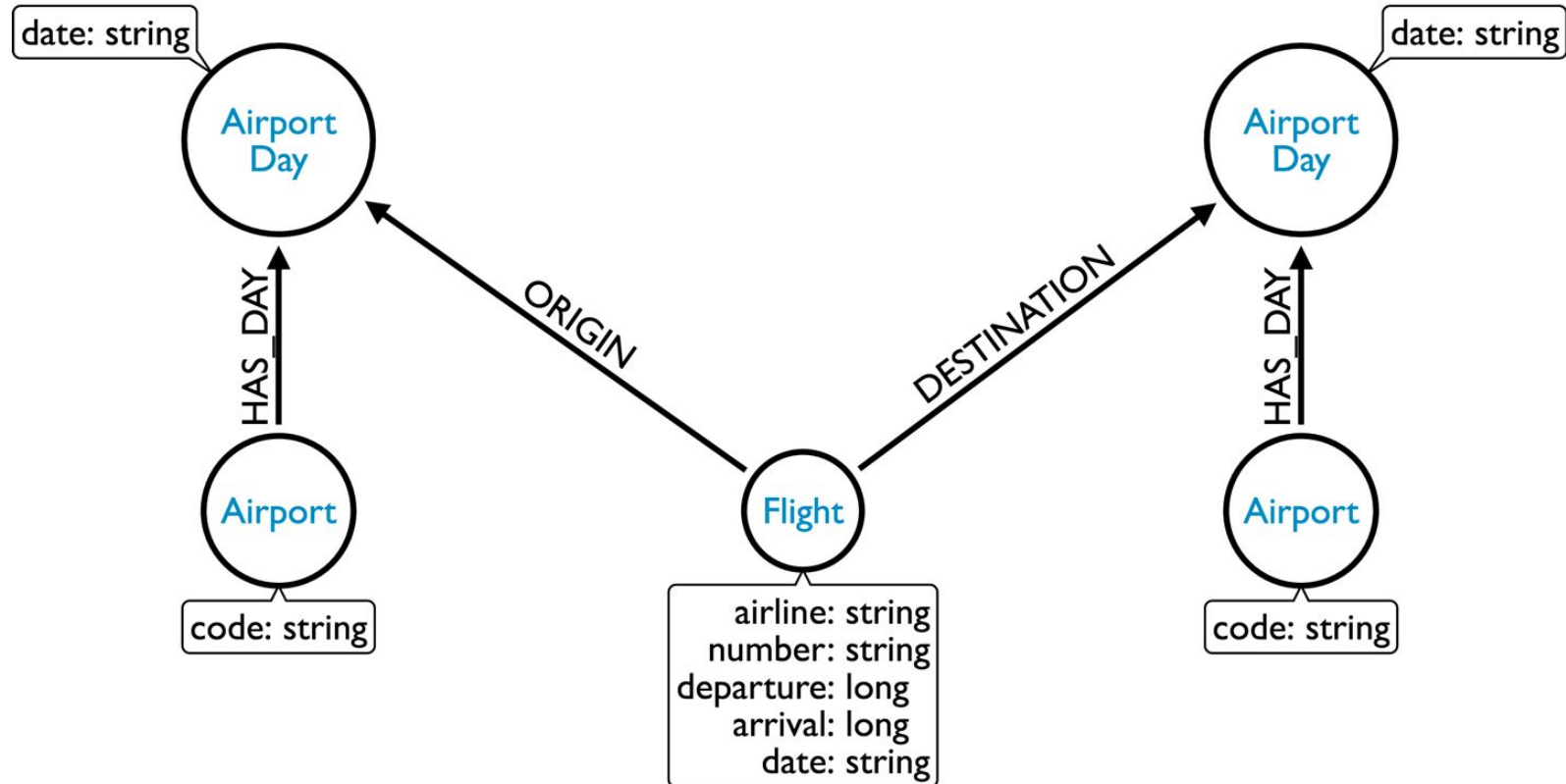
Our model



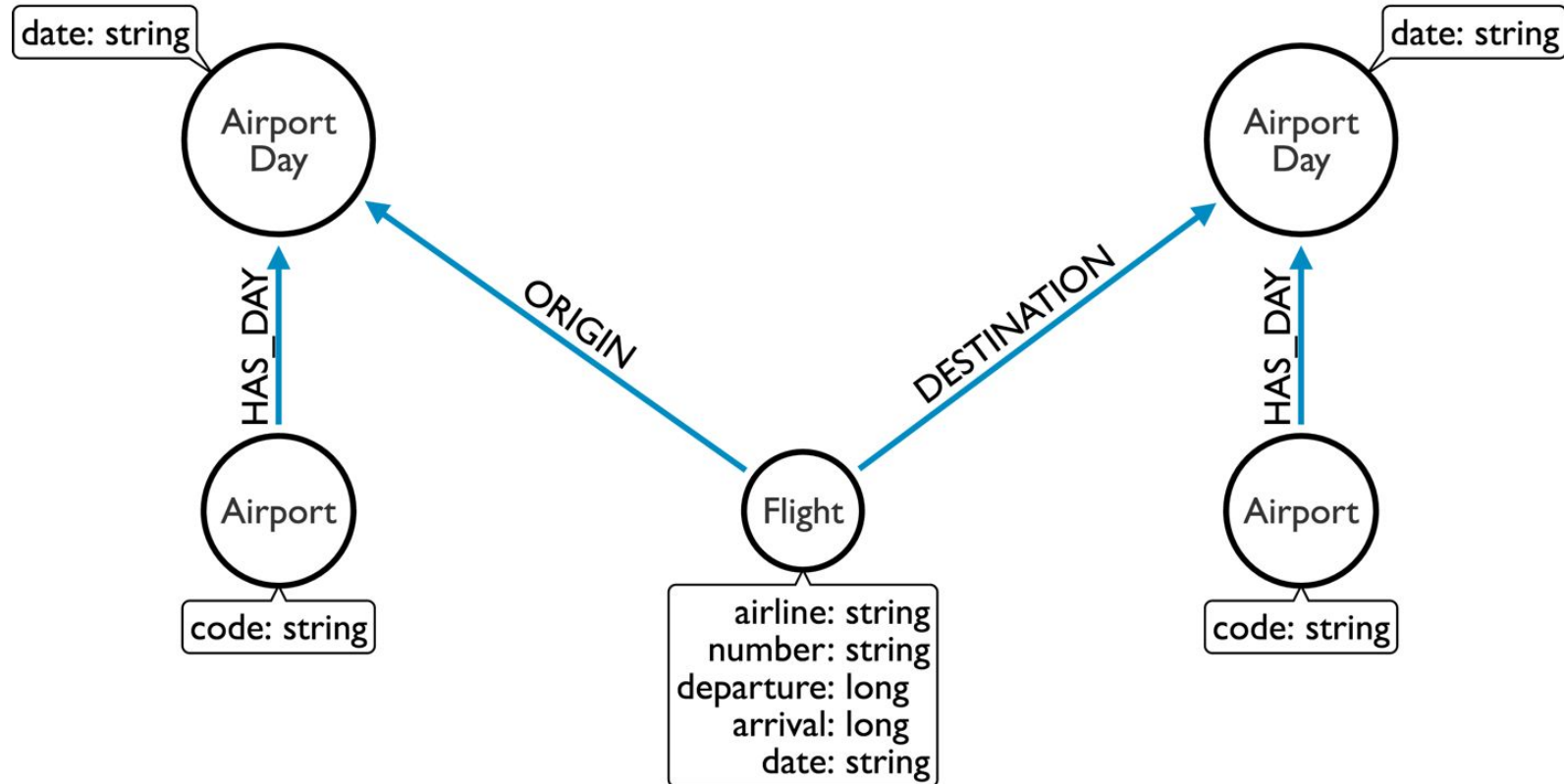
Nodes are for things



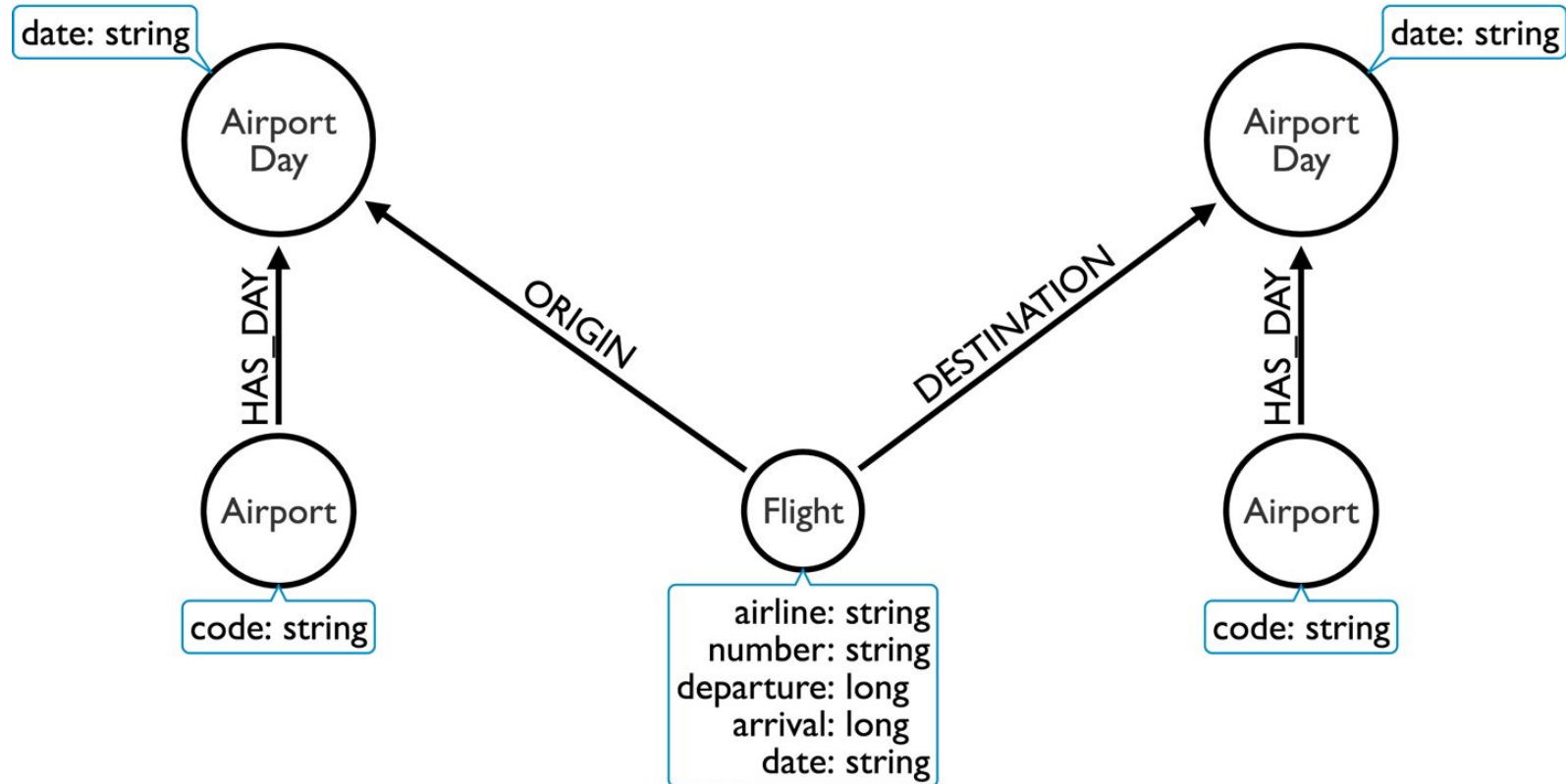
Labels for grouping



Relationships for structure



Properties for attributes



Modeling choices





Properties vs. Relationships

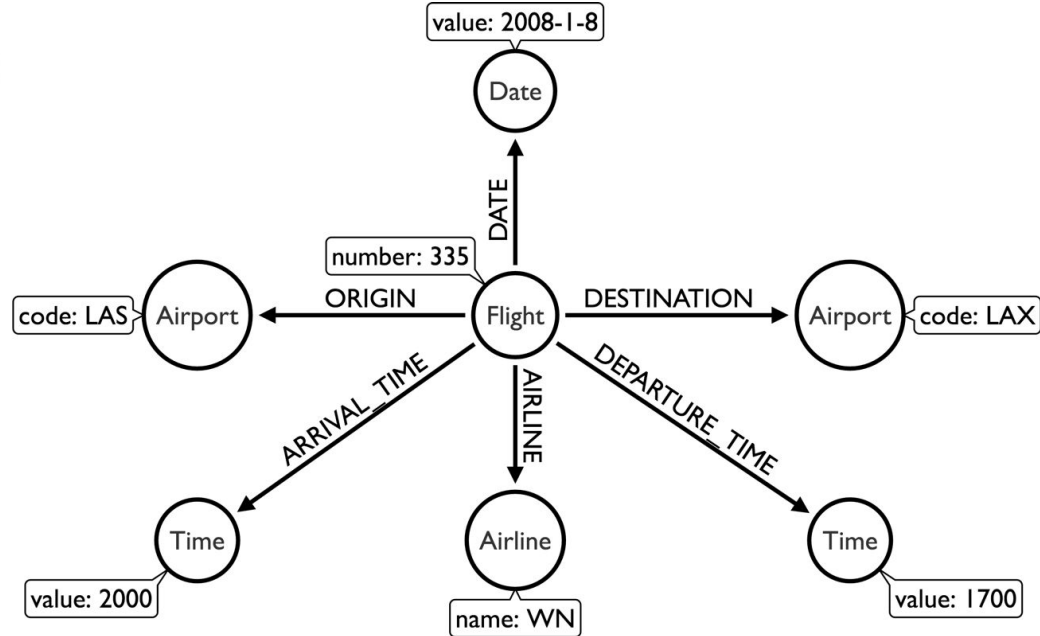


Properties vs Relationships



number: 335
origin: LAS
destination: LAX
airline: WN
date: 2008-1-8
departure: 1700
arrival: 2000

Flight

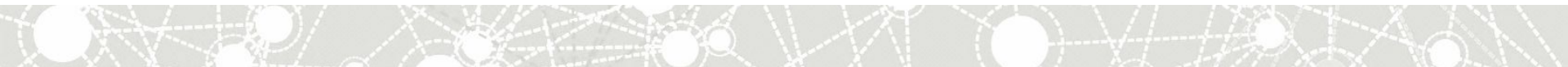


Properties vs Relationships



We only need to pull out a node if we're going to query through it, otherwise a property will suffice.

But if we pull out every single property then we end up with an RDF model and lose the benefit of the property graph

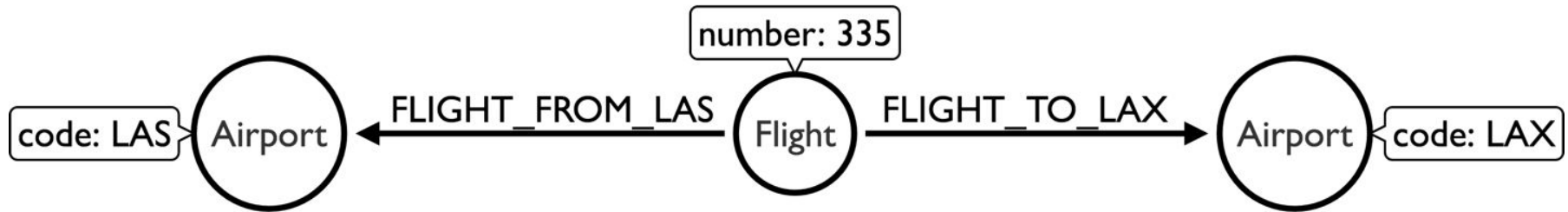
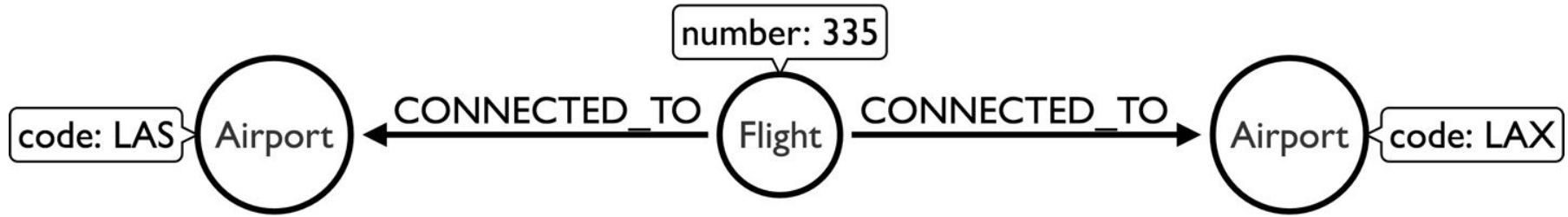




Relationship Granularity

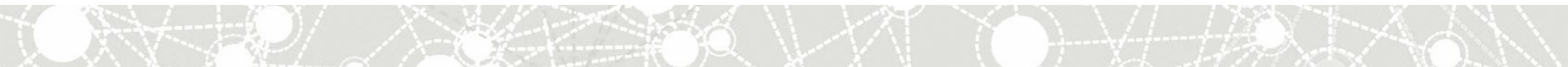


Relationship Granularity

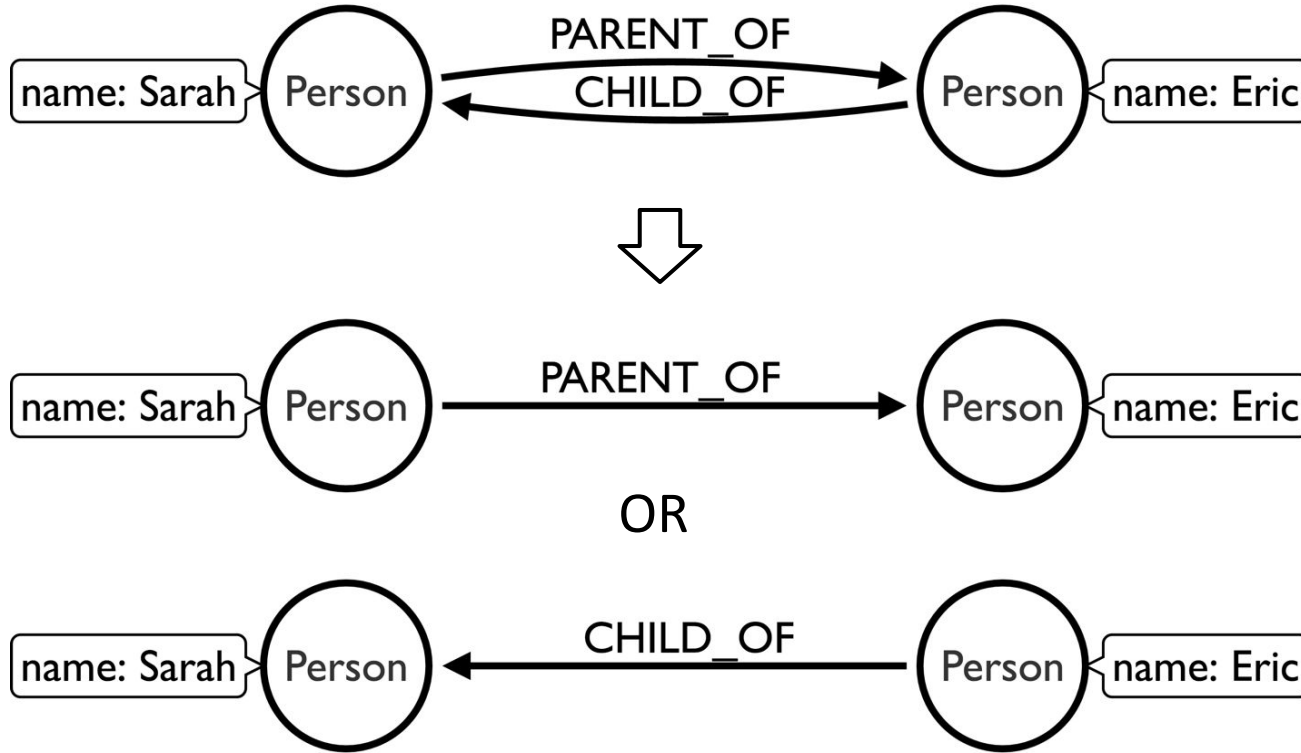




Symmetric Relationships

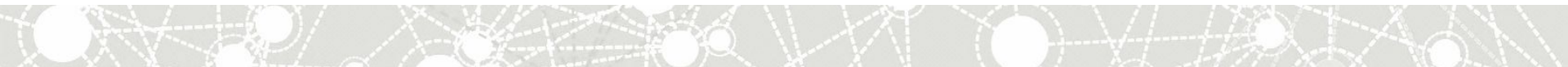


Symmetric relationships





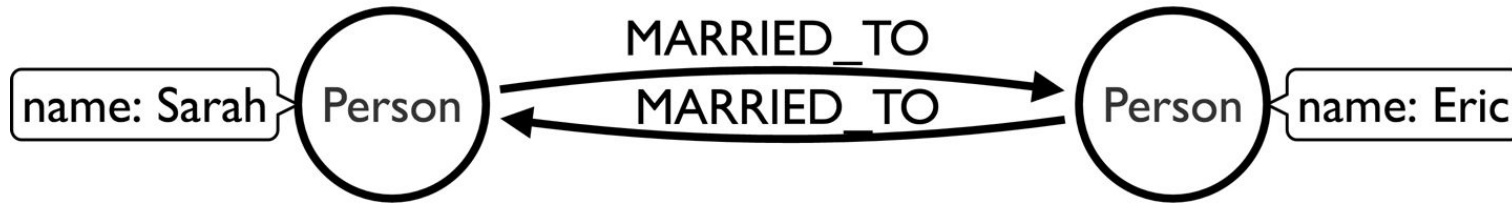
Bidirectional Relationships



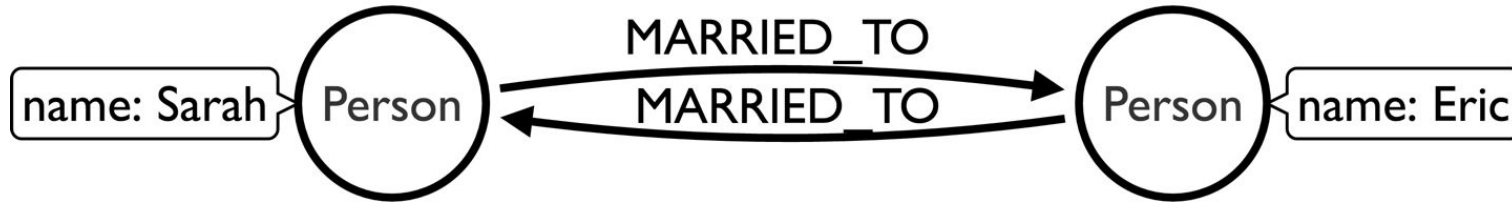
Bidirectional relationships



No need to have the relationship in both directions



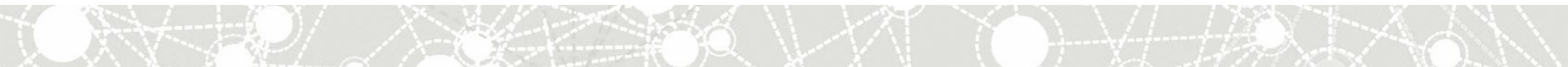
Use single relationship and ignore direction in queries



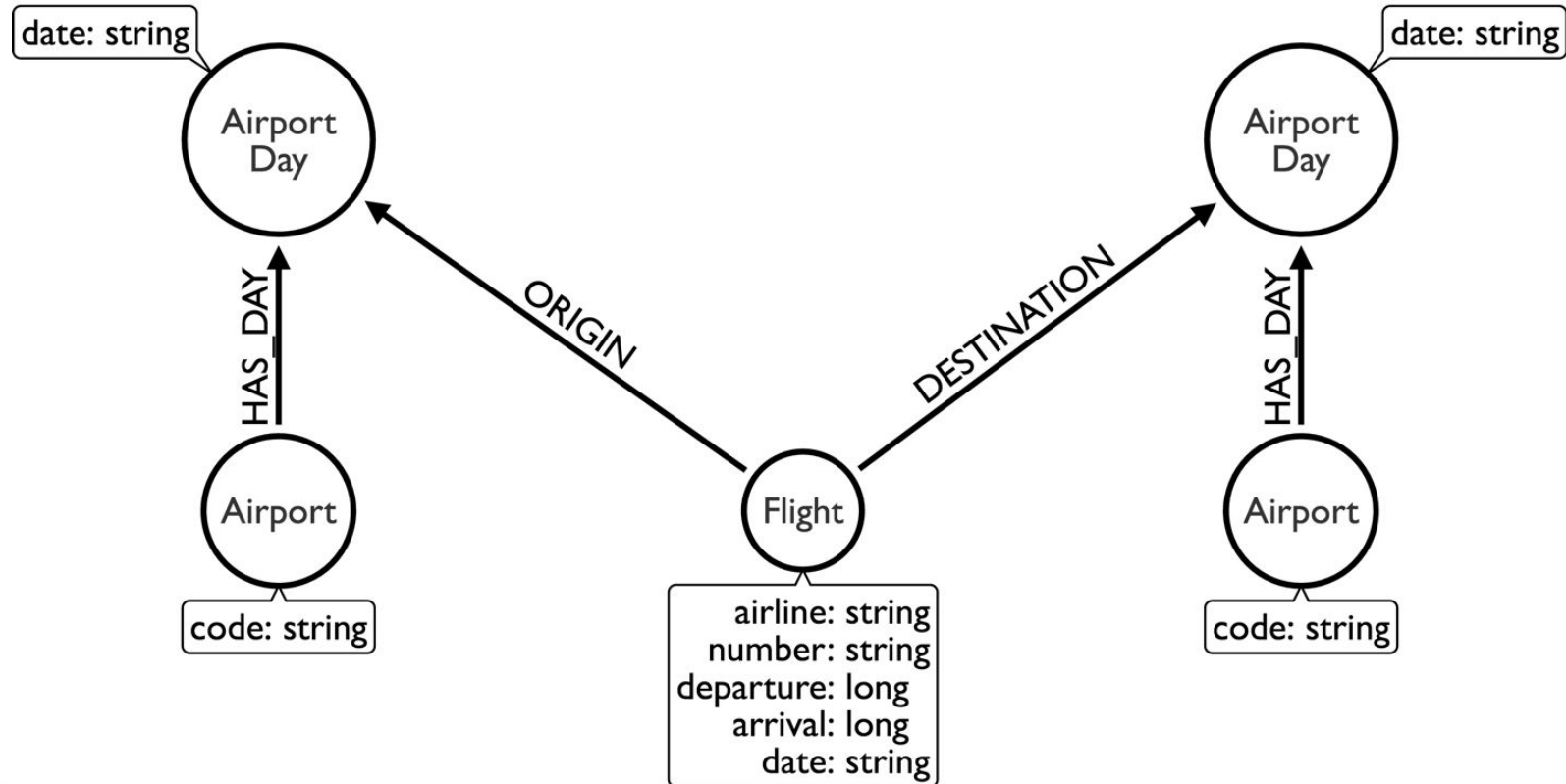
```
MATCH (:Person {name: 'Eric'})-[:MARRIED_TO]-(p2)  
RETURN p2
```




General vs. Specific Relationships



General Relationships

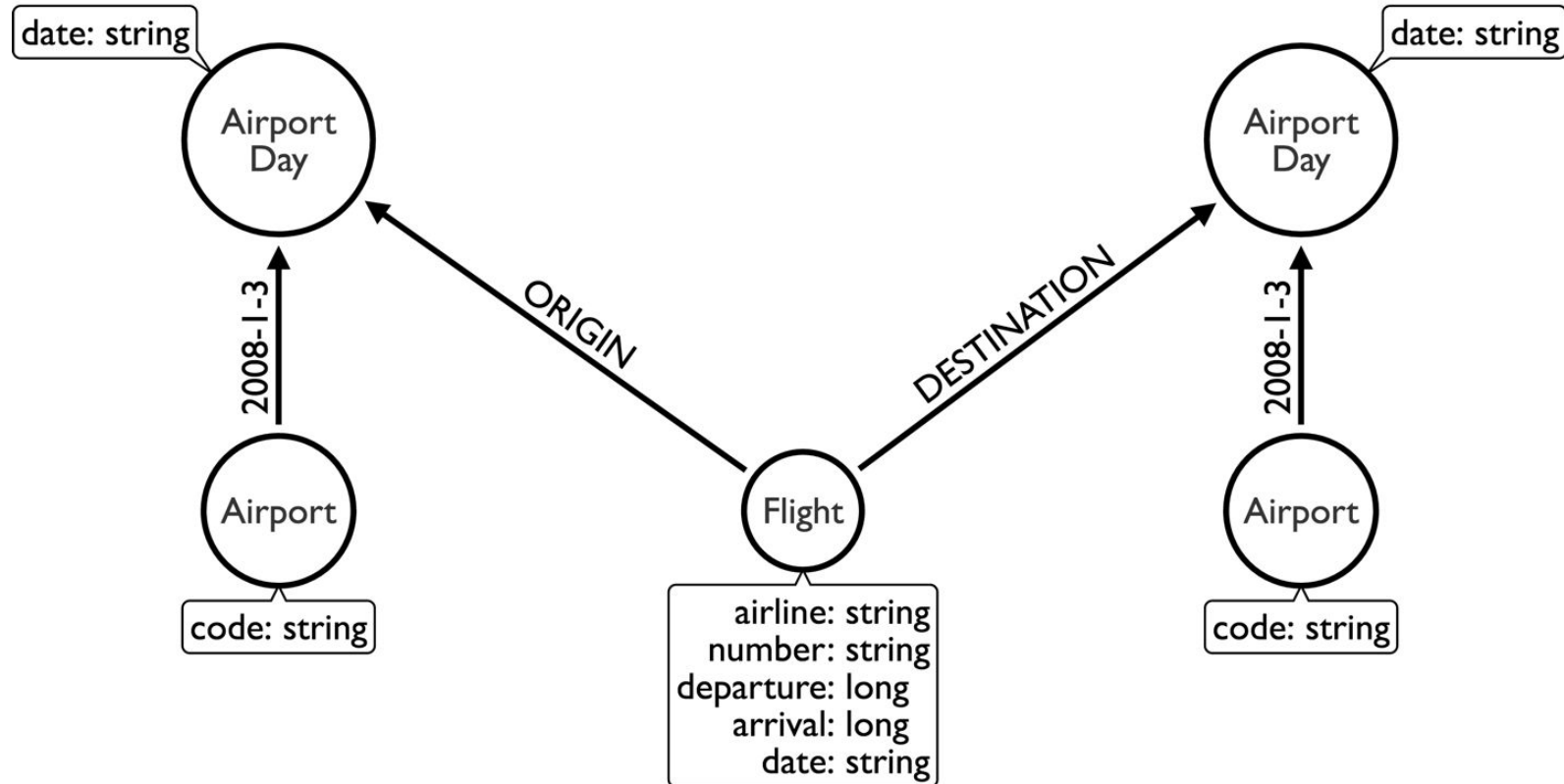


General: Find flights on a specific date



```
MATCH (origin:Airport {code: "LAS"})-[:HAS_DAY]->(originDay:AirportDay),  
        (originDay)<-[:ORIGIN]-(flight:Flight),  
        (flight)-[:DESTINATION]->(destinationDay),  
        (destinationDay:AirportDay)<-[:HAS_DAY]-(destination:Airport {code: "MDW"})  
  
WHERE originDay.date = "2008-1-3" AND destinationDay.date = "2008-1-3"  
  
RETURN flight.date, flight.number, flight.airline, flight.departure, flight.arrival  
  
ORDER BY flight.date, flight.departure
```

Specific Relationships



Specific: Find flights on a specific date



```
MATCH (origin:Airport {code: "LAS"})-[:`2008-1-3`]->(originDay:AirportDay),  
      (originDay)<-[:ORIGIN]-(flight:Flight),  
      (flight)-[:DESTINATION]->(destinationDay),  
      (destinationDay:AirportDay)<-[:`2008-1-3`]-(destination:Airport {code: "MDW"})  
  
RETURN flight.date, flight.number, flight.airline, flight.departure, flight.arrival  
  
ORDER BY flight.date, flight.departure;
```

General: Find flights by year and month



```
MATCH (origin:Airport {code: "LAS"})-[:HAS_DAY]->(originDay:AirportDay),  
        (originDay)<-[:ORIGIN]-(flight:Flight)
```

```
WHERE originDay.date STARTS WITH "2008-1"
```

```
RETURN flight.date, flight.number, flight.airline, flight.departure, flight.arrival
```

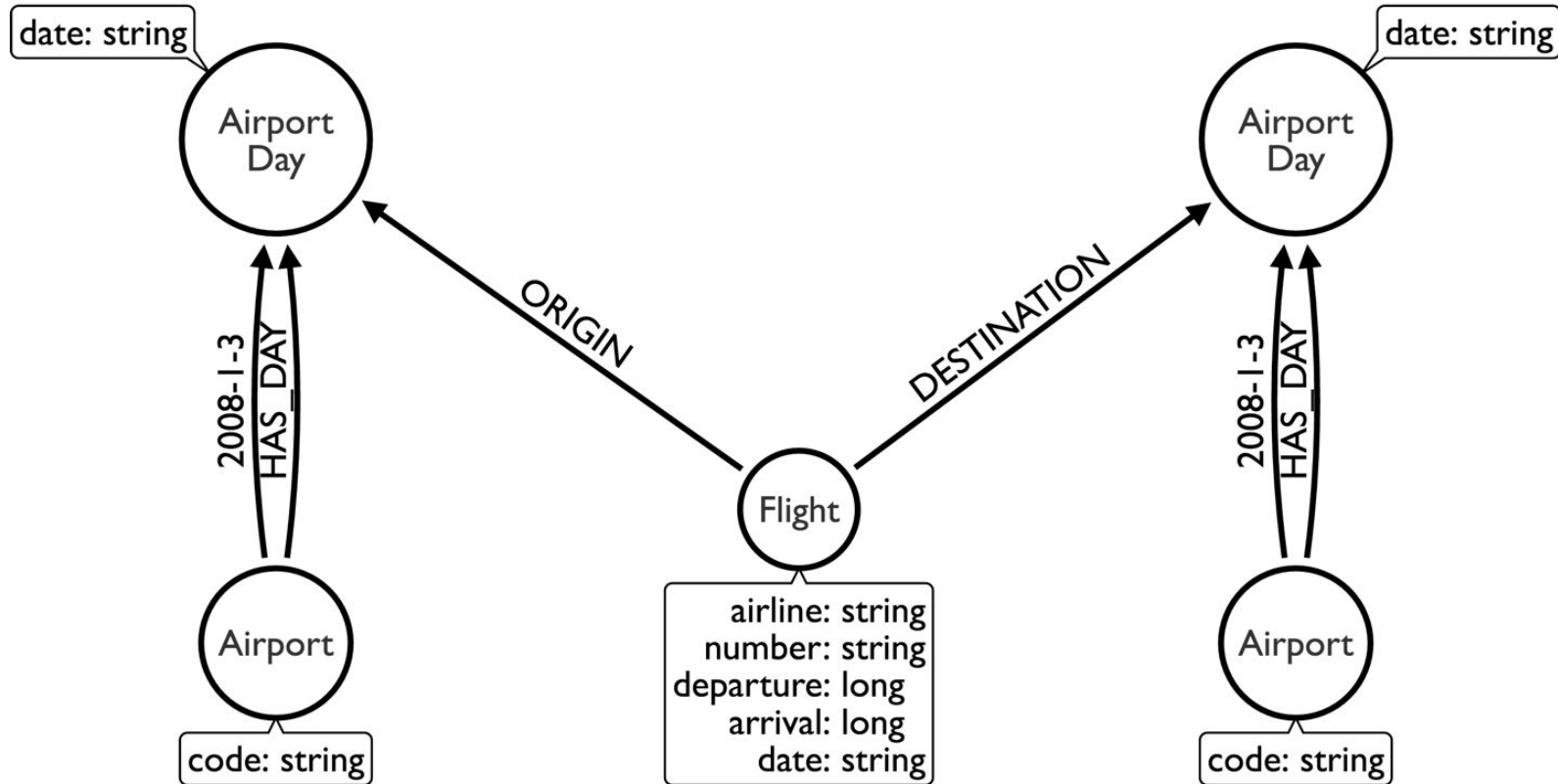
```
ORDER BY flight.date, flight.departure
```


Specific: Find flights by year and month



```
MATCH (origin:Airport {code: "LAS"})  
  -[:`2008-1-3`|:`2008-1-4`|:`2008-1-5`|:`2008-1-6`]->(originDay:AirportDay),  
  (originDay)<-[:ORIGIN]-(flight:Flight)  
  
RETURN flight.date, flight.number, flight.airline, flight.departure, flight.arrival  
  
ORDER BY flight.date, flight.departure
```

Best of both worlds?



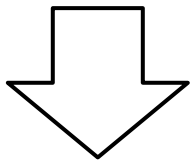
Refactorings



Derive node from relationship



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



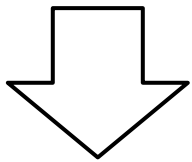
`(origin)<-[:ORIGIN]-(flight)-[:DESTINATION]->(destination)`



Derive node from property



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

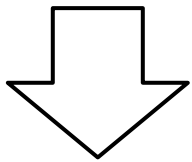


```
(origin)-[:HAS_DAY]->(originAirportDay)<-[:ORIGIN]-(flight),  
(destination)-[:HAS_DAY]->(destAirportDay)<-[:DESTINATION]-(flight)
```

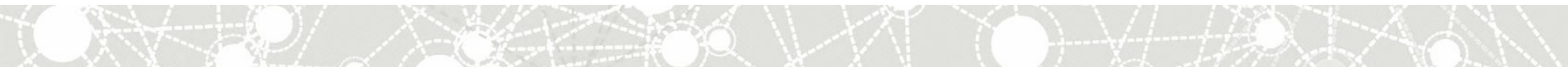
Derive relationship from property



```
(airport)-[:HAS_DAY]->(airportDay {date: "2008-1-3"})
```

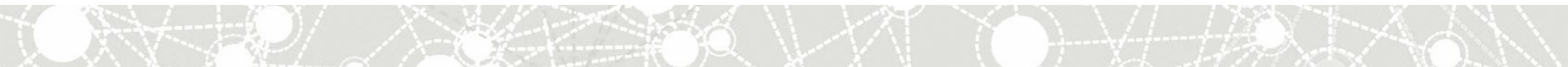


```
(airport)-[:`2008-1-3`]->(airportDay {date: "2008-1-3"})
```

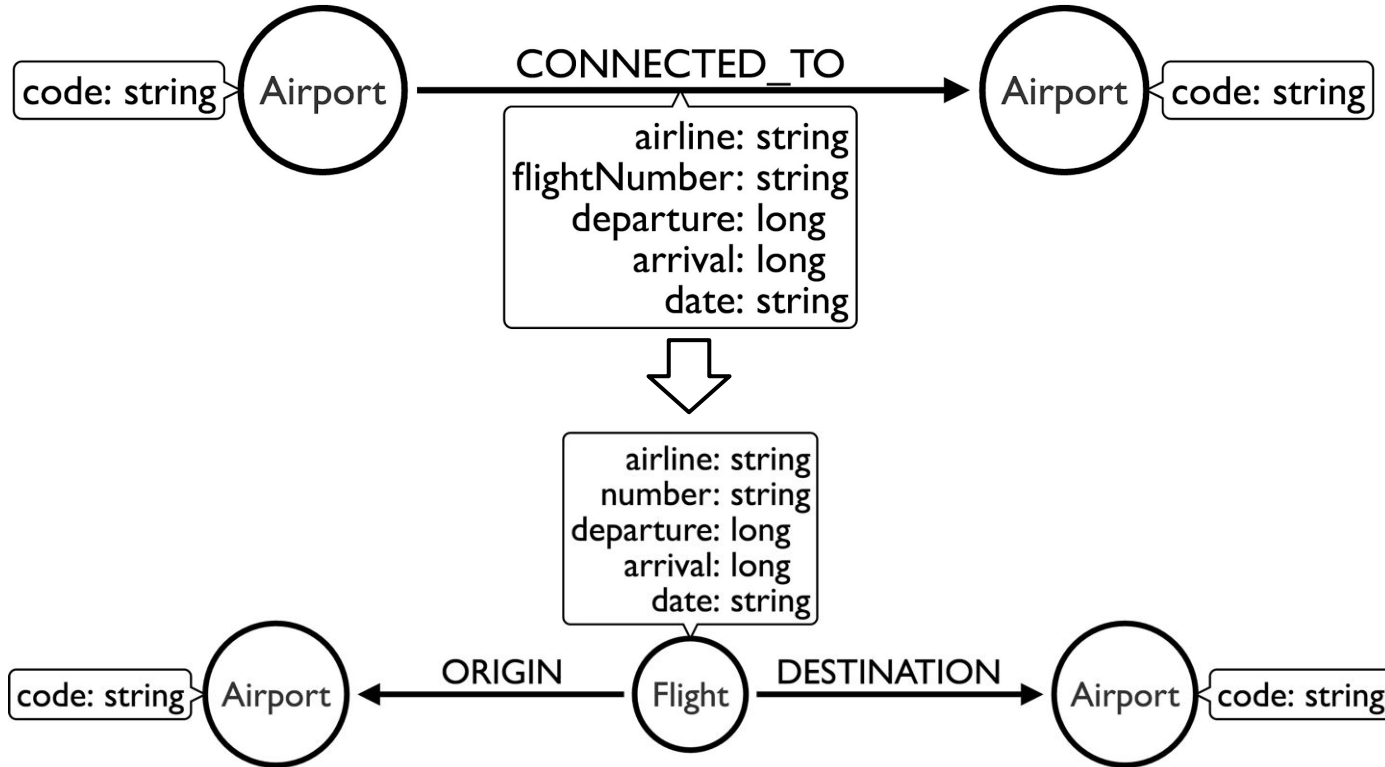




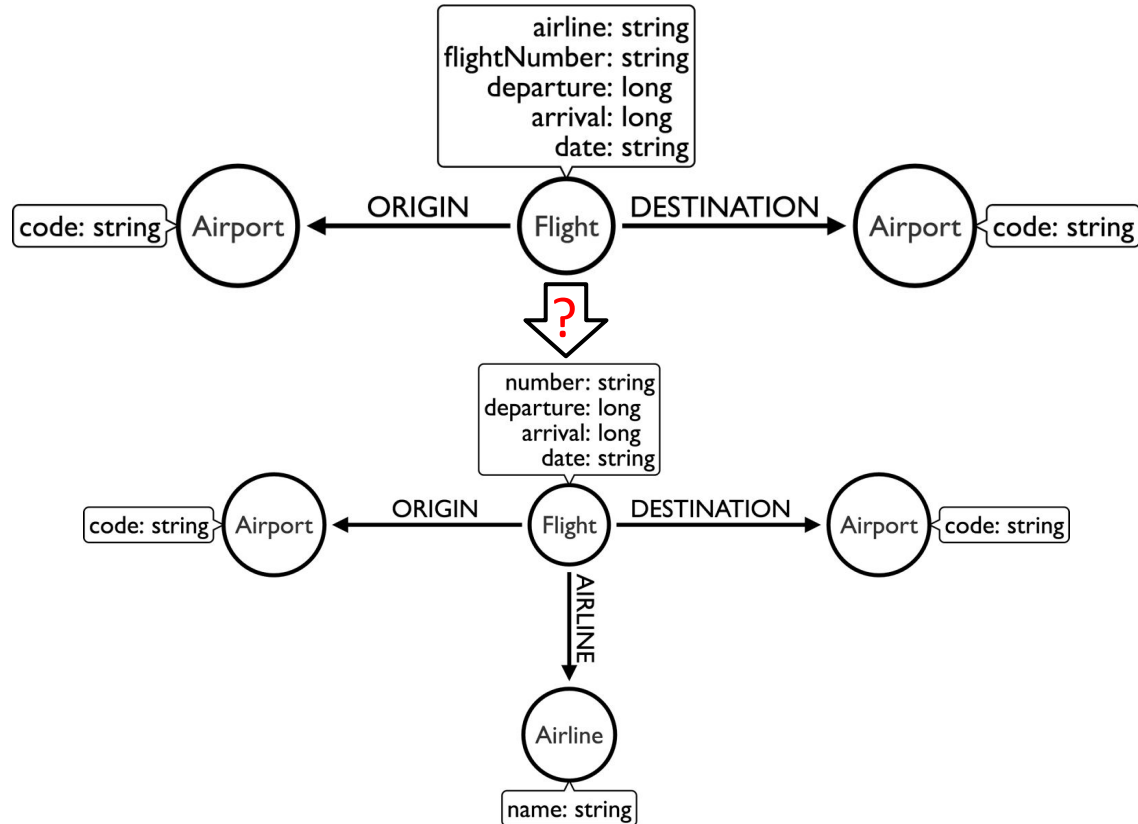
Evolving the Model



Revealing nodes



Revealing nodes?



End of Module Modeling Guidelines

Questions ?

