

Recommendation Engine With Neo4j

[recommendation](#), [neo4j](#)

Algorithm Types

- [Collaborative Filtering](#)
 - An algorithm that considers users interactions with products, with the assumption that other users will behave in similar ways.
- [Content Based Filtering](#)
 - An algorithm that considers similarities between products and categories of products

Using Data Relationships for Recommendations

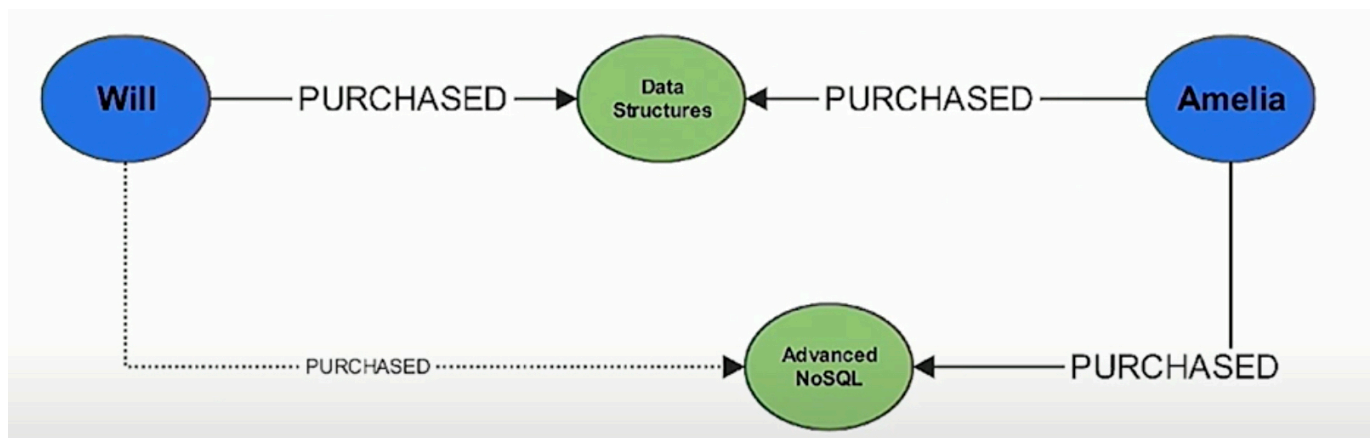
Content based filtering

- Recommend items based on what users have liked in the past

Collaborative filtering

- Predict what users like based on the similarity of their behaviors, activities and preference to others

Collaborative Filtering



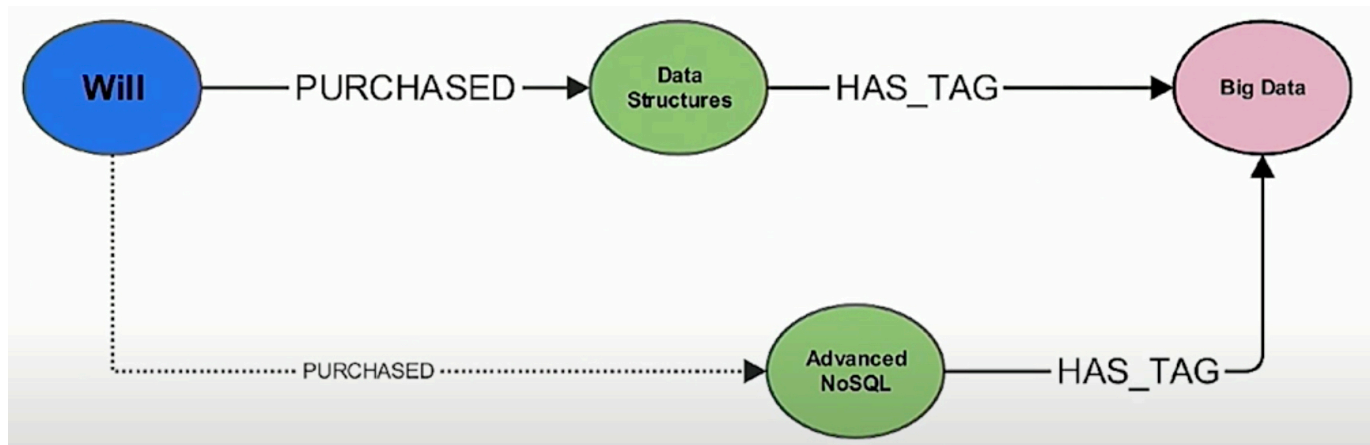
In Cypher

```
MATCH (will:Person {name:"Will"}) -[:PURCHASED]->(b:Book) <-[:PURCHASED]- (o:Person)
MATCH (o) -[:PURCHASED]->(rec:Book)
WHERE NOT exists((will) -[:PURCHASED]->(rec))
RETURN rec
```

Basic initial approach. Improvements:

- aggregate across all purchases
- scoring / normalize
- compute similarity metrics

Content Filtering

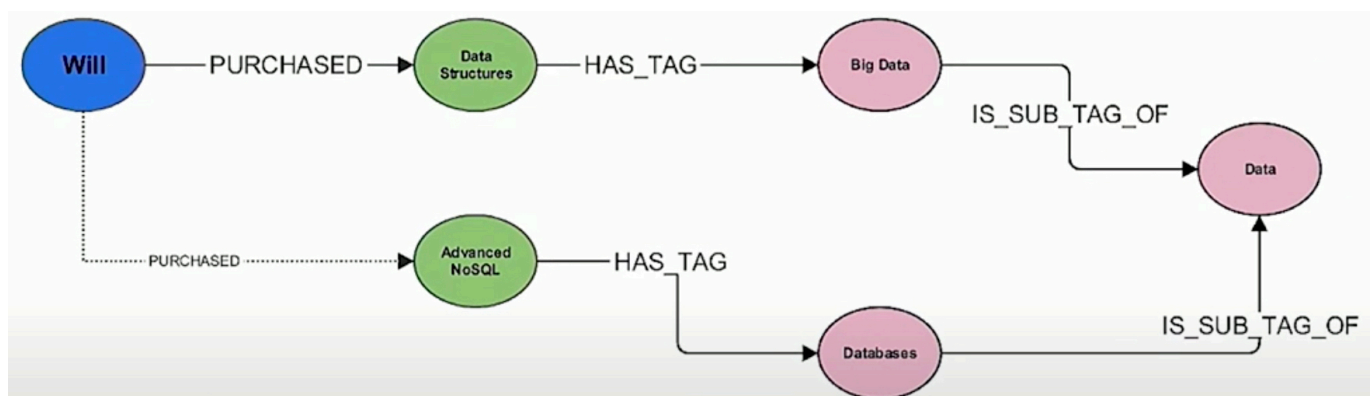


In Cypher

```

MATCH (will:Person {name:"Will"})-[:PURCHASED]->(b:Book)<-[:HAS_TAG]-(t:Tag)
MATCH (t)<-[:HAS_TAG]-(other:Book)
WHERE NOT exists((will)-[:PURCHASED]->(other))
RETURN other
  
```

Content Filtering - Concept Hierarchy



```

MATCH (will:Person {name: "Will"})-[:PURCHASED]->(b:Book)<-[:HAS_TAG]-(t:Tag)
MATCH (t)-[:IS_SUB_TAG_OF]->(:Tag)<-[:IS_SUB_TAG_OF]-(ot:Tag)
MATCH (ot)<-[:HAS_TAG]-(other:Book) WHERE NOT exists((will)-[:PURCHASED]->(other))
RETURN other
  
```

Ref

- <https://www.youtube.com/watch?v=wbI5JwIFYEM>
- <https://bit.ly/neo4josconslides>
- <https://bit.ly/neo4jnotebook>

Plugin Backlinks:

From:

<http://jace.link/> - **Various Ways**

Permanent link:

<http://jace.link/open/recommendation-engine-with-neo4j>

Last update: **2021/07/07 10:14**

