# Research - Week 4 - Tanmay Munigala – 12742995

Graph Databases

Their advantages, disadvantages over relational databases as well as common use cases. If use cases match with our use case of providing tailored recommendation, we can provision an additional graph database resource to serve this purpose.

## What are they?

1. Based on graph theory - consist of Nodes (Vertices) and Relationships (Edges)

2. Nodes can have labels and properties to describe them.

3. Relationships can also have labels to describe them, as well as a direction.

## Advantages

1. Graph databases are more powerful and faster due to indexing.

2. You can add a lot more relationships and still be performant.

3. Helps you answer unanticipated questions about data and locate indirect relationships 4. Data captured can be easily changed dynamically.

## Use Cases

1. **Retail** - Shopping recommendations

2. **Hospitality/Travel** - Pricing recommendations

3. **Logistics** - Package routing

4. **Finance** - Financial Fraud Detection

5. **Cybersecurity** - Network vulnerability checks

6. **Social** **networks** - Post recommendations

## Limitations

Not efficient at processing high volumes of transactions.

Not good at handling queries that span entire databases.

Not optimized to store business entities. So will need to be combined with a relational DB.

Not optimized for mass analytics across all relationships and records, better for finding relationships of a single record.

Relational DB

Designed to answer known and anticipated questions, so difficult to track wide-spanning relationships.

AWS Neptune

Fully managed Graph DB-as-a-service. Good option for us to get up and running quickly if the rest of our infrastructure ends up being hosted on AWS.

## Neo4j

Open-source, native graph database management system.

Most popular graph database with lots of supporting documentation and resources.

## Conclusion

Based on the research, graph databases are not always ideal, but in our team’s case, they perfectly serve the purpose. This is because we need to build our database keeping the central use case of providing smart trip recommendations to customers in mind, and as graph databases are better suited for answering questions which are unknown and require traversal over large amounts of neighbouring records. Ideally, we should select Neo4j as our graph database provider due to the comprehensive documentation and supporting material available. If the rest of our project infrastructure is hosted on AWS, then we can look into AWS Neptune as our primary provider as well.