

Notes on the Data Management Course

Chapter 1

Data Warehouse

Chapter 2

NoSQL Graphs Databases

NoSQL is a movement borned in response to SQL movement. Relational Databases have difficult in managing **Big Data** (bigger data volume, more rapidly changing and more structural variety), for that reasons NoSQL movement is borned.

There are 4 types of NoSQL movements:

- **Graph Based**
- Key-Value stores
- Document-Based
- Column Oriented

Graph Based movement in turn are divided in two:

- Graph Databases (Neo4j)
- RDF Databases

2.1 Graph Databases

A graph database is a database that uses the graph structure with nodes, edges and graph properties to represent and store data.

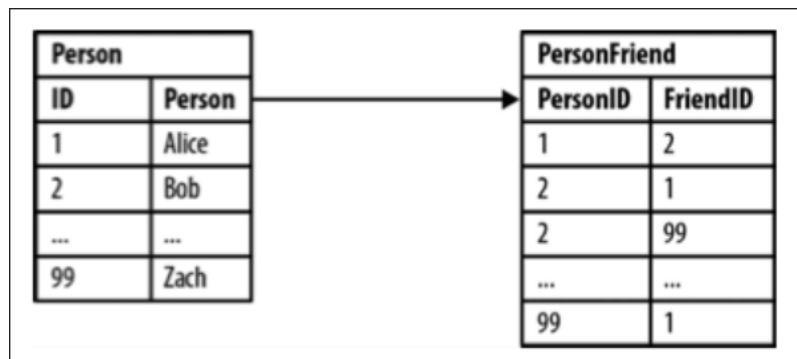
A management system (eg Neo4j) offers **CRUD** operations (Create, Read, Update and Delete) to acces and manipulate data.

Graph databases are **schemaless** \implies you can accumulate data without the need of a predefined rigid schema. You can add new nodes and new edges thanks to the preproperty of graphs.

Graph databases can be **queried** through declarative languages, they can provide very good performances because essentially the avoid classic **joins**.

Relational DB vs Graph DB

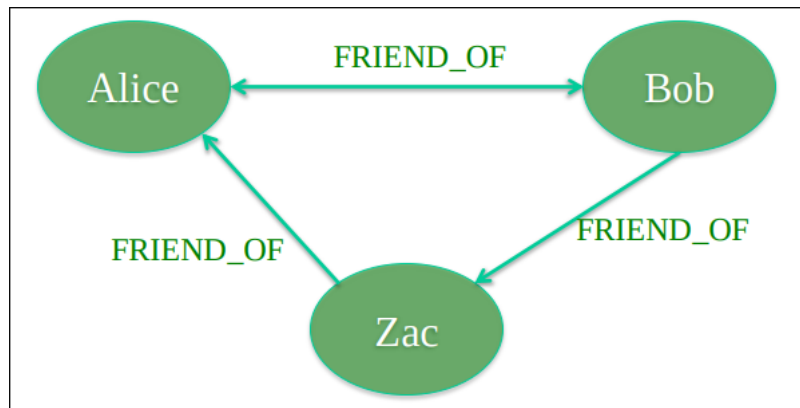
Target: Modeling friends of friends in a relational database.



```
SELECT p2.Person AS ALICE_FRIEND_OF_FRIEND
FROM Person p1 JOIN PersonFriend pf1 ON
    p1.ID = pf1.PersonID JOIN PersonFriend pf2 ON
    pf1.FriendID = pf2.PersonID JOIN Person p2 ON
    pf2.FriendID = p2.ID
WHERE p1.Person = 'Alice' AND pf2.FriendID <> p1.ID
```

The performances highly deteriorates when we go more in depth into the network of friends.

In the graph databases we can model the relation friend and friend of friend in this way:



Relationship in a graph naturally form paths \implies Querying means actual **traversing** the graph.

Neo4J Cypher

```
MATCH (p:Person) -[:FRIEND_OF*2..5] -> (fof:Person)
RETURN p, fof
```

Syntax of Neo4j language for querying

GraphDB vs Relational DB - Queries

- *Relational DB*: The join operation forms a graph that is dynamically constructed as one table linked to the other. The limitation is that the graph is not in explicit in the relational structure.
- *Graph DB*: There is no explicit join operation because vertices maintain direct references to their adjacent edges. The edges of the graph serve as explicit, "hard wired" join structure. Traversing (querying) the graph is a **constant** time operation

Chapter 3

NoSQL Graphs Document Based