

Designing Data Intensive Applications

CHAPTER 2

Data models and Query Languages

Relational Model - Data organized in relations ~ tables in SQL.

why NOSQL?

- ① More scalability as compared to SQL databases
- ② open source instead of paid software
- ③ Query operations that are not supported by SQL.

The way we write code → most applications use OOP paradigm to write code and you'll require extra translation layer between objects in code and data models in SQL tables.

users table						
user_id	first_name	last_name	summary			
251	Bill	Gates	Co-chair of ... blogger.			
	region_id	industry_id	photo_id			
	us:91	131	57817532			
regions table						
id	region_name					
us:7	Greater Boston Area					
us:91	Greater Seattle Area					
industries table						
id	industry_name					
43	Financial Services					
48	Construction					
131	Philanthropy					
positions table						
id	user_id	job_title	organization			
458	251	Co-chair	Bill & Melinda Gates F...			
457	251	Co-founder, Chairman	Microsoft			
education table						
id	user_id	school_name	start	end		
807	251	Harvard University	1973	1975		
806	251	Lakeside School, Seattle	NULL	NULL		
contact_info table						
id	user_id	type	url			
155	251	blog	http://thegatesnotes.com			
156	251	twitter	http://twitter.com/BillGates			

```
{ user_id : 251,  
  first_name : Bill  
  last_name : Gates  
  positions : [  
    { jobtitle : Co-Chair },  
    { jobtitle : Co-founder }  
  ]  
  ...  
}
```

Image - Wikimedia Commons

Fetch profile for Bill Gates

- ① Relational → perform multiple queries in tables
or perform complex join
- ② Document → all information at single place.

Comparison of document Model vs Relational Model

- ① Application code simplicity →
↳ It will depend on application, what is the usecase it is solving.
↳

Property	Document	Relational
① Data in application ?	→ requirement is document like structure. → poor joins support.	→ requirement is many to many relationships.
② Schema flexibility	→ Don't enforce strict schema. → schema on read - schema is maintained in application code. → Adding new fields is easy task.	→ strict schema → schema on write. → Adding new columns require proper migration strategy ↳ could be a real pain point if data size is huge. → If all records are expected to have same structure, RDBMS is good way to enforce it.

③ Data Locality → document is stored as continuous string. for queries	→ If data is split into multiple tables and entire data needs to be retrieved, there are performance issues.
	→ Oracle has feature "multi-table index cluster tables" which provide locality property in RDBMS.

Similarity

- ↳ There can always be feature similarity between SQL & NOSQL.
- It totally depends on usecase we're trying to solve.
- ↳ NOSQL database has support for joins.
- ↳ SQL databases can store data in document form.

Query Languages for Data

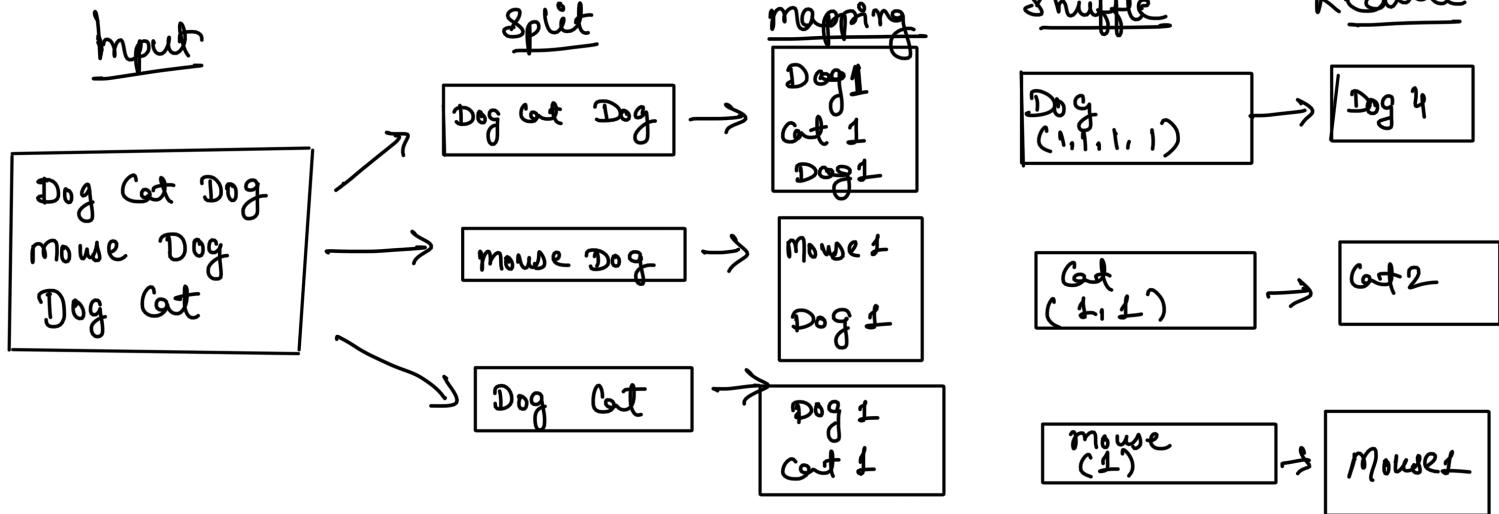
SQL - declarative query Language

Declarative Language	Imperative Language
<p>→ You specify pattern of data you need.</p> <p>↳ but you don't specify how these results are achieved.</p> <p>↳ Database system's query optimizer to decide how to gather this data.</p>	<p>→ Tells computer to perform certain operations in certain order.</p>

Map Reduce Querying

→ programming model for processing large amounts of data.

↳ in later chapters → CH 10

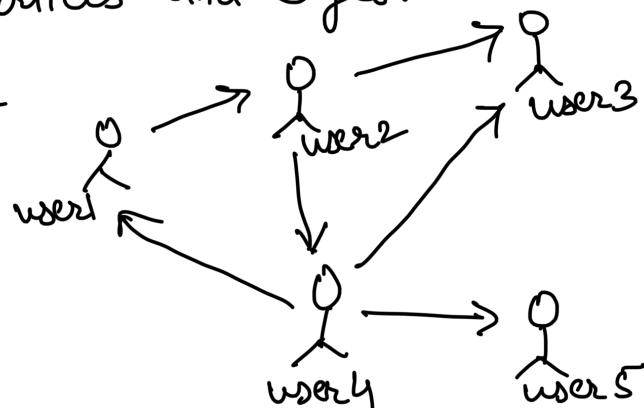


Graph Data Model

- ↳ Document is good if there are mostly 1-many relationships or no relationships between records.
- ↳ many-many relationships can be very well handled by RDBMS.
- ↳ But what if these relationships become very complex and nested.
 - Then graph data model is more suitable.

Graph → vertices and edges.

Example



↳ Graph provides flexibility in data modelling as compared to RDBMS.

Example Neo4j

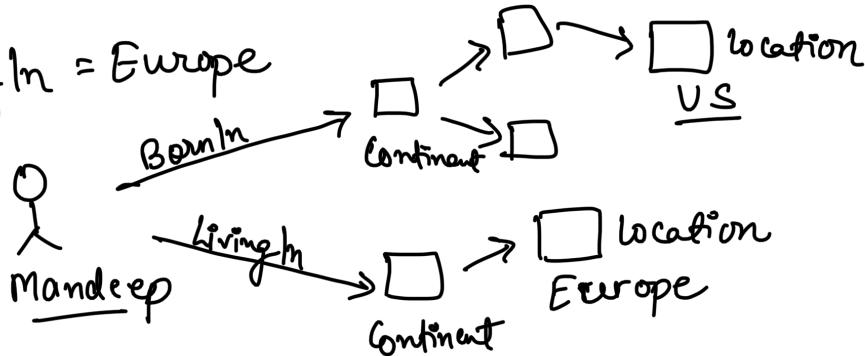
query language - Cypher (declarative)

example - find all people who immigrated from US to Europe.

↳ BornIn = US

&&

LivingIn = Europe



Graph Databases Compared to Network Model

Network	Graph
<ul style="list-style-type: none">① Schema specifies record type could be nested within each record type.② To reach particular record, traverse the access path.③ Children of record are ordered set.④ Imperative queries	<ul style="list-style-type: none">① No such restriction. A vertex can have edge to any other vertex.② You can directly go to any vertex by its unique identifier.③ vertices & edges are not ordered.④ Support for both declarative & imperative queries.

Stay Tuned for follow up chapters.

Subscribe for more. YouTube - Ms Deep Singh

Happy Learning 😊