SQL - the Dark Side

Tamas Piros

2019

Drawbacks of SQL

- They are not designed to handle change
- They are not designed for heterogeneous data
- They are not designed for scale
- They are note designed for mixed workloads
- They are a mismatch for modern app development

RDBMS are not designed to handle change

- Data is created faster and it also changes faster
- Businesses must keep up with this pace
  - Market dynamics, on-demand services, acquisitions
- Data modelling is a lengthy and daunting process in itself, change(s) add(s) further complexity
- Just remember, data models almost always change

Matt Allen (MarkLogic) - http://www.marklogic.com/blog/relational-databases-change/

RDBMS are not designed for heterogeneous data

- Businesses need to store everything
  - Customers, partners, regulators
- Creation of data silos to store heterogeneous data is not the right solution
- Most relational databases only support certain application or certain type of data
- “duct tape” together data sources which leads to additional complexity and frustration
- Unstructured data is complex by nature

Matt Allen (MarkLogic) - http://www.marklogic.com/blog/relational-databases-heterogeneous-data/

RDBMS are not designed for scale

- Scalability and elasticity is challenging for relational systems
- Relational systems are designed to run on single servers in order to maintain table mapping integrity and to avoid distributed computing
    - Scaling is therefore achieved by buying more expensive hardware
- Database vendors apply improvements such as master-slave model, sharding, shared storage, distributed caching
- Flexibility is often sacrificed for performance

Matt Allen (MarkLogic) - http://www.marklogic.com/blog/relational-databases-scale/

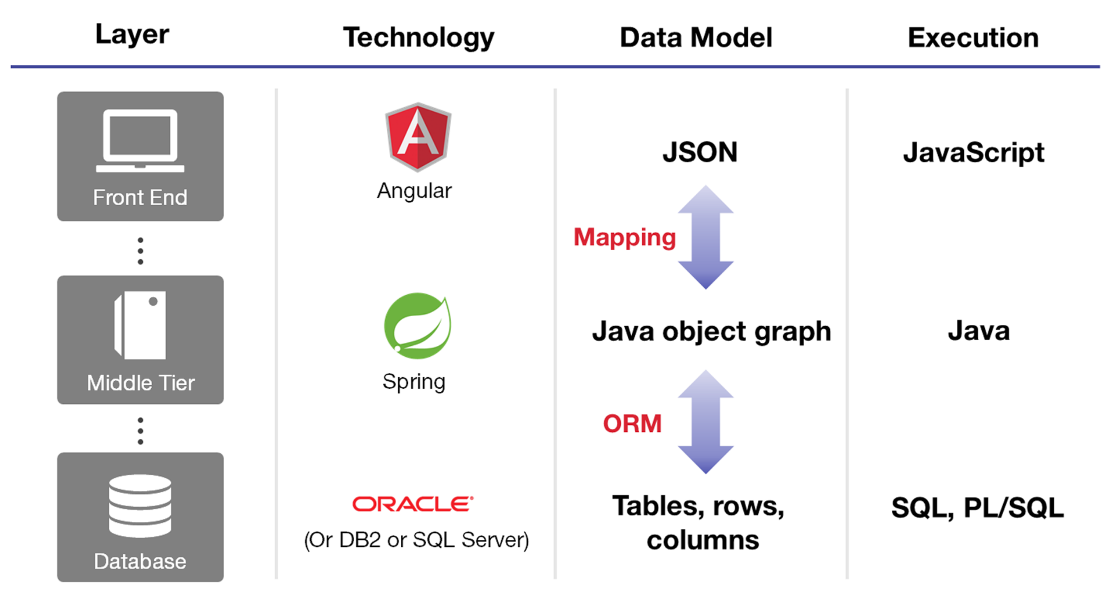
RDBMS are not designed for mixed workloads

- Relational databases are designed to handle two types of workloads: operational and analytical but they can’t do both
  - OLTP (online transaction processing) is designed for fast transactions - “who bought X?”
  - OLAP (online analytical processing) is designed for aggregating information – “How many people bought X?”
- Data is moved from an operational system to a central data warehouse, then moved again to a data mart to answer specific questions
- Problems start to creep in – maintaining systems, enforcing security

Matt Allen (MarkLogic) - http://www.marklogic.com/blog/relational-databases-are-not-designed-for-mixed-workloads/

RDBMS are a mismatch for modern app development

- ORMs are required – adds mor process
- Performance loss, potentially difficult to maintain codebase
- Legacy databases require lega



Layer	Technology	Data Model	Execution
Front End	Angular	JSON	JavaScript
		Mapping	
Middle Tier	Spring	Java object graph	Java
		ORM	
Database	ORACLE (Or DB2 or SQL Server)	Tables, rows, columns	SQL, PL/SQL

Matt Allen (MarkLogic) - http://www.marklogic.com/blog/rdbms-mismatch-modern-app-development/

Conclusion

- Relational databases do have their own use cases
    - When data is truly relational use relational databases
- In today’s rapidly changing environments, where complex data is used, relational databases lag behind
- Remember: use the right technology to the right problem