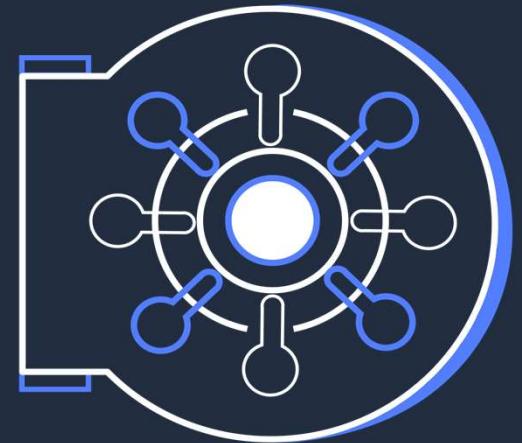


AWS TECHSHIFT

EMBARK



Data Management

Data Management is the practice of storing, validating, indexing, retrieving, and collating data used by a computer system.

Relational DBMS (database management system) refers to a database that stores data in a structured format, using rows and columns in tables. Also known as SQL database.

A **NoSQL** database provides a mechanism for storage and retrieval of data that is structured other than the tabular relations used in relational databases.

Relational

Traditional applications, ERP, CRM, e-commerce



Amazon Aurora



Amazon Redshift



Amazon RDS

Key-value

High-traffic web apps, e-commerce systems, gaming applications



Amazon DynamoDB

Document

Content management, catalogs, user profiles



Amazon DocumentDB

Graph

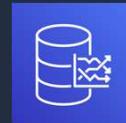
Fraud detection,
social networking,
recommendation
engines



Amazon Neptune

Time series

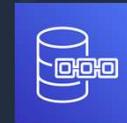
IoT applications,
DevOps, industrial
telemetry



Amazon Timestream

Ledger

Systems of record,
supply chain,
registrations,
banking
transactions

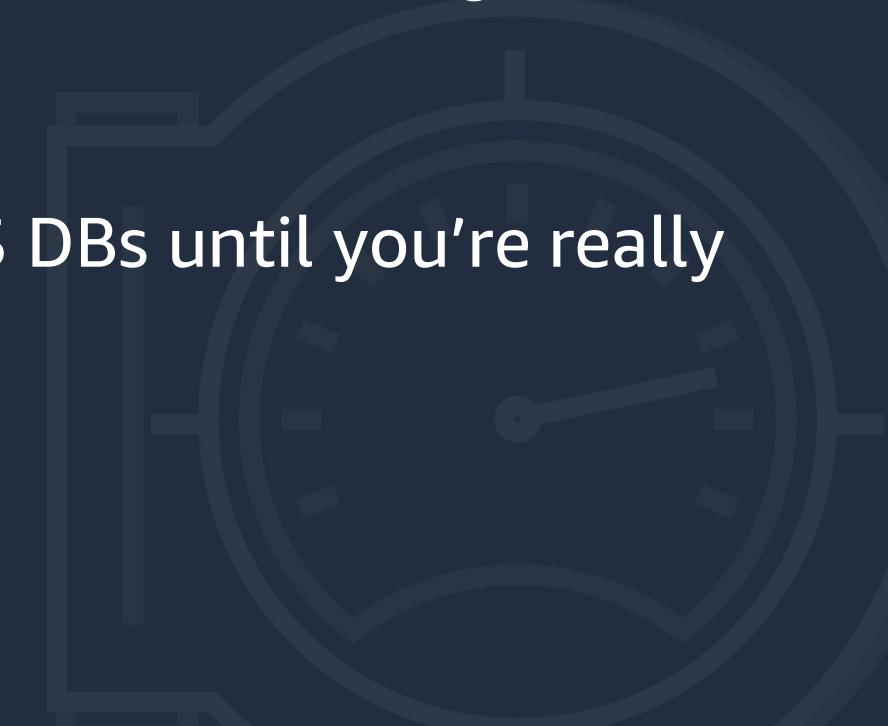


Amazon Quantum Ledger
Database

But how do I choose what DB technology I need?
RDBMS? NoSQL?

- Not a binary decision !
- Blended approach can reduce technical debt as you grow
- Start with RDBMS databases where it makes sense.

- Established and well worn technology
- Lots of existing code, communities, books, background, tools, etc
- You aren't going to break RDBMS DBs until you're really BIG!!!
- General purpose



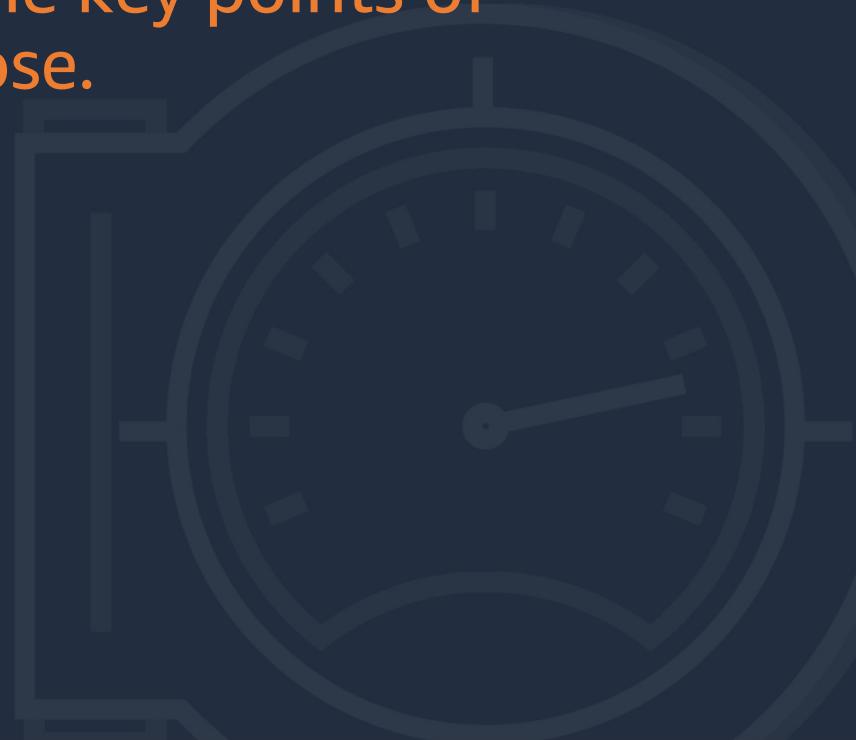
- Trades storage for CPU time through relationships / joins
- Difficult to scale and shard
- Requires more work to optimize for specific workloads:
 - Large number of transactions per second
 - Time based indexing

- Super low latency applications
- Metadata driven datasets
- Highly un-relational data
- Need schema-less data constructs*
- Massive amounts of data (again, in the TB range)
- Rapid ingest of data (thousands of records/sec)

*Need != “its easier to do dev without schemas”

- Generally uses more storage space
- Data modelled on usage, harder to remodel
- Optimizes read / create over updates

So decide wisely. Look for the key points of scale and purpose.





Amazon RDS

- Relational Databases
 - Amazon Aurora, MySQL, PostgreSQL, Oracle, SQL Server, MariaDB
- Scale vertically through instance sizes
- Scale horizontally through read replicas
- Single click Multi-AZ support





Amazon Aurora

- High Performance and Scalability
- High Availability and Durability
- MySQL and PostgreSQL Compatible

Serverless

- Pay per second for the database capacity you use
- Automatically scales with no instances to manage
- Starts up on demand, shuts down when not in use



Amazon Redshift

- Petabyte-scale data warehousing
- Columnar storage database
- Massively parallel leader nodes route and speed up queries using machine learning and results caching

Redshift Spectrum

- Exabyte-scale data lake analytics
- Query S3 using SQL



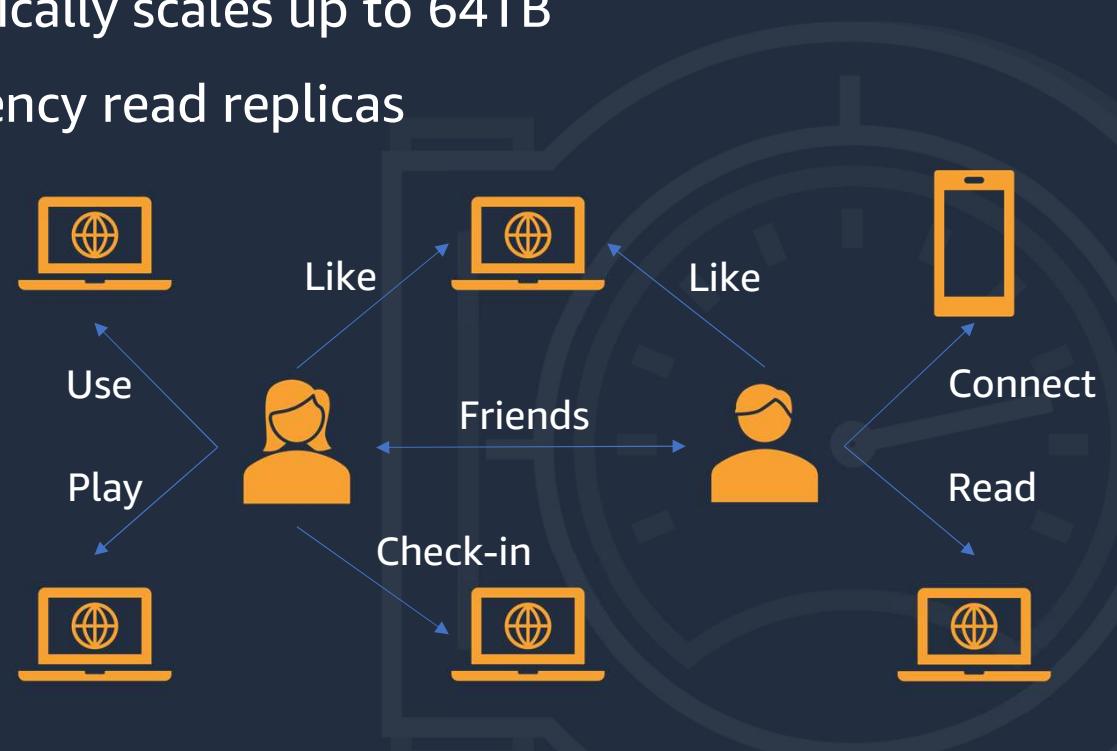
Amazon DynamoDB

- Key-value and document data models
- Microsecond latency
- Automated global replication with global tables
- Real-time data processing with DynamoDB Streams
- No servers to manage
- Reserved or on-demand capacity with auto-scaling
- DynamoDB Accelerator (DAX) provides a fully managed in-memory cache



Amazon Neptune

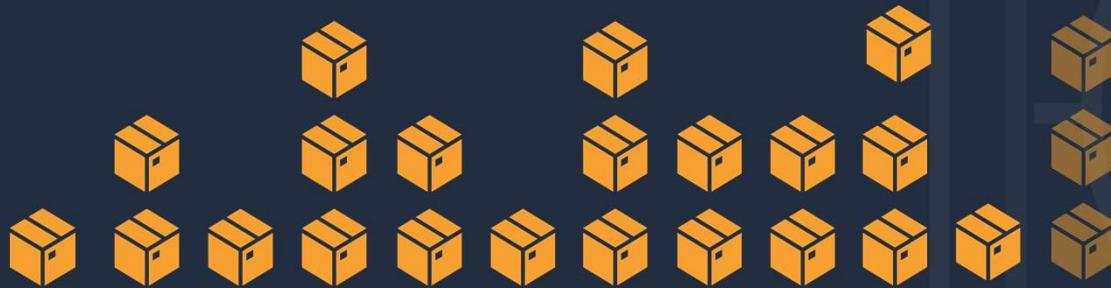
- Supports open graph APIs
 - Gremlin & SPARQL
- Storage automatically scales up to 64TB
- Up to 15 low latency read replicas





Amazon Timestream

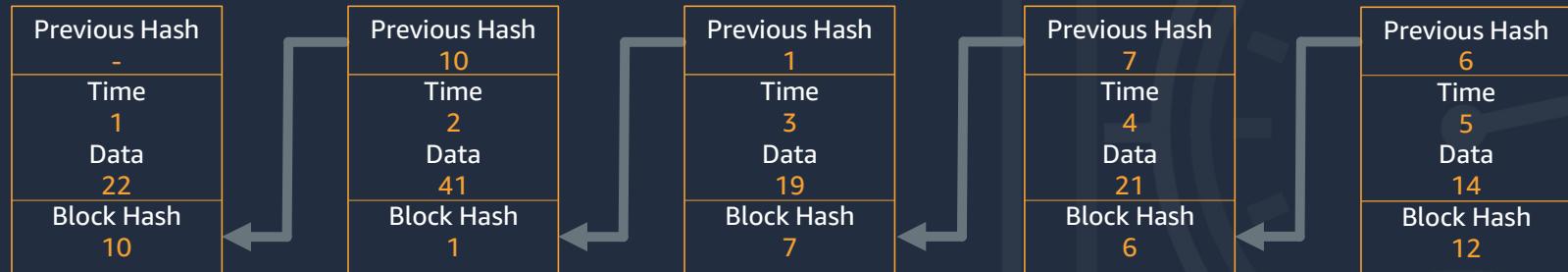
- Supports open graph APIs
 - Gremlin & SPARQL
- Storage automatically scales up to 64TB
- Up to 15 low latency read replicas
- Data stored keyed based on time intervals





Amazon Quantum Ledger Database

- Immutable and Transparent
- Cryptographically Verifiable
- QLDB is a database, not a blockchain framework
 - Provides better performance and scale
 - Up to 15 low latency read replicas



End

