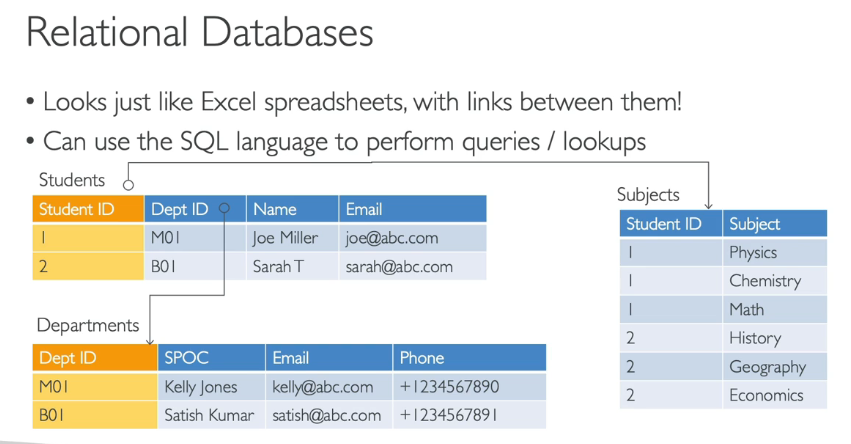
**Databases**

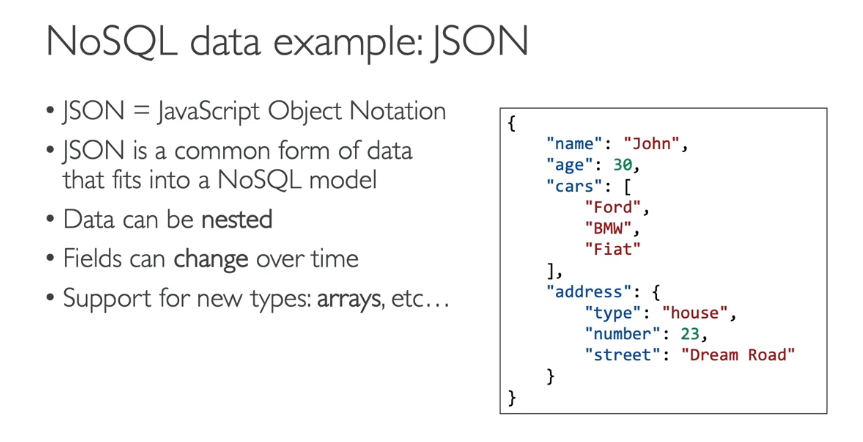
Storing data on disk (EFS, EBS, EC2 Instance Store, S3) can have its limits

* Sometimes, you want to store data in a database…
* You can structure the data
* You build indexes to efficiently query / search through the data
* You define relationships between your datasets
* Databases are optimized for a purpose and come with different features, shapes and constraint



**NoSQL Databases**

* NoSQL = non-SQL = non relational databases
* NoSQL databases are purpose built for specific data models and have flexible schemas for building modern applications.
* Benefits:
  + Flexibility: easy to evolve data model
  + Scalability: designed to scale-out by using distributed clusters
  + High-performance: optimized for a specific data model
  + Highly functional: types optimized for the data model
* Examples: Key-value, document, graph, in-memory, search databases



**Databases & Shared Responsibility on AWS**

* AWS offers use to manage different databases
* Benefits include:
  + Quick Provisioning, High Availability, Vertical and Horizontal Scaling
  + Automated Backup & Restore, Operations, Upgrades
  + Operating System Patching is handled by AWS
  + Monitoring, alerting
* Note: many databases technologies could be run on EC2, but you must handle yourself the resiliency, backup, patching, high availability, fault tolerance, scaling

**AWS RDS Overview**

* RDS stands for Relational Database Service
* It’s a managed DB service for DB use SQL as a query language.
* It allows you to create databases in the cloud that are managed by AWS
  + Postgres
  + MySQL
  + MariaDB
  + Oracle
  + Microsoft SQL Server
  + **Aurora (AWS Proprietary database)**

### Advantage over using RDS versus deploying DB on EC2

**Reduced administrative burden:**

* **Automated tasks:** With RDS, you don't have to worry about manually provisioning servers, patching the operating system, or setting up backups and restores. This frees up your time to focus on other important tasks.
* **Monitoring and scaling:** RDS provides built-in monitoring dashboards so you can easily track your database's performance. Scaling is also simplified, allowing you to adjust resources (storage, compute) vertically or horizontally as needed.

**Improved performance and reliability:**

* **Read replicas:** RDS enables you to create read replicas for improved read performance, which can be crucial for applications with high read traffic.
* **Disaster recovery:** Multi-AZ (Availability Zone) deployments in RDS offer built-in disaster recovery. If one AZ goes down, your database can automatically failover to another, minimizing downtime.
* **Storage:** RDS leverages EBS volumes for storage, which is reliable and scalable.

**Security considerations:**

* **Limited access:** While you can't directly SSH into RDS instances, this can actually be a security benefit. It restricts unauthorized access to the database server itself. You can still manage and configure the database using the RDS console or API.

**Choosing the right option depends on your needs:**

* **RDS is ideal for:** Most users who prioritize ease of use, automated management, and cost-effectiveness.
* **EC2 might be preferable for:** Scenarios where you need fine-grained control over the underlying infrastructure or require specific database features not supported by RDS.

**Amazon Aurora**

* Aurora is a proprietary technology from AWS (not open sourced)
* PostgreSQL and MySQL are both supported as Aurora DB
* Aurora is “AWS cloud optimized” and claims 5x performance improvement over MySQL on RDS, over 3x the performance of Postgres on RDS
* Aurora storage automatically grows in increments of 10GB, up to 64 TB.
* Aurora costs more than RDS (20% more) – but is more efficient
* Not in the free tier

Amazon Aurora Serverless

• Automated database instantiation and

auto-scaling based on actual usage

• PostgreSQL and MySQL are both

supported as Aurora Serverless DB

• No capacity planning needed

• Least management overhead

• Pay per second, can be more cost-

effective

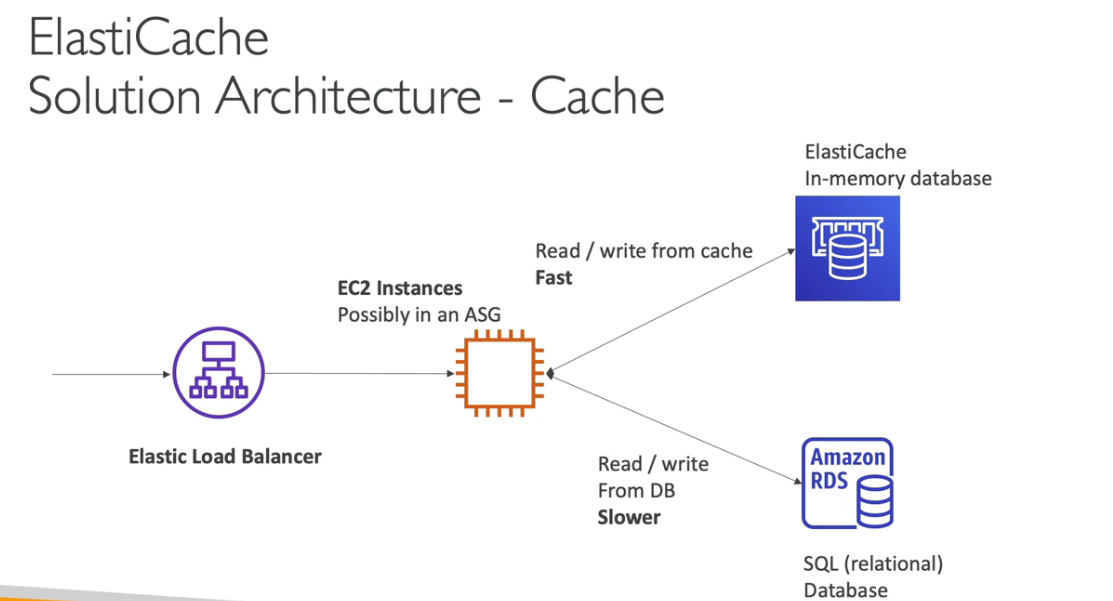
• Use cases: good for infrequent,

intermittent or unpredictable

workloads...

**Amazon ElastiCache Overview**

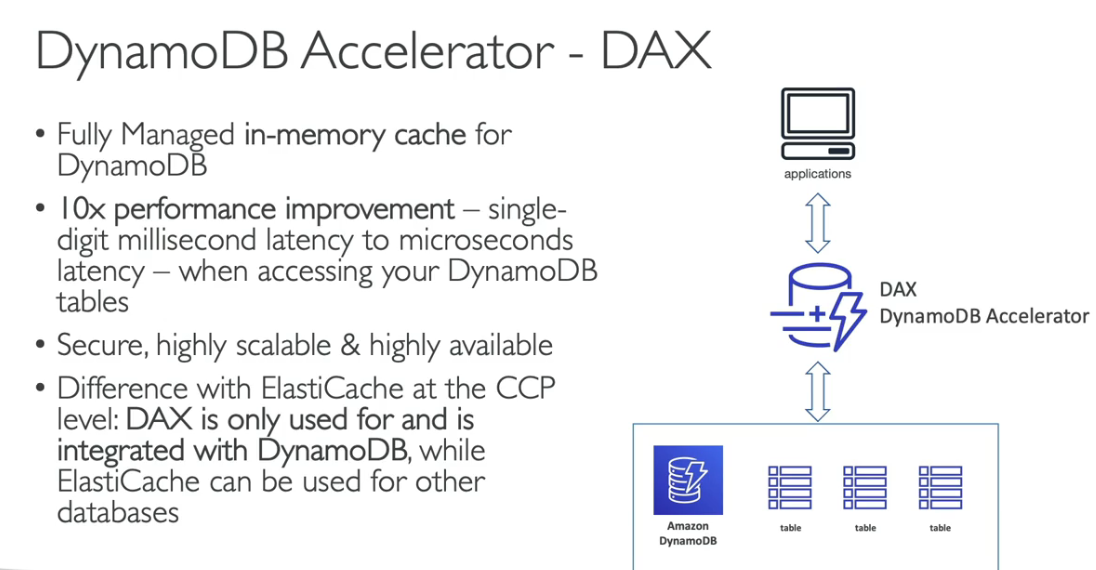
* The same way RDS is to get managed Relational Databases…
* ElastiCache is to get managed Redis or Memcached
* Caches are in-memory databases with high performance, low latency
* Helps reduce load off databases for read intensive workloads
* AWS takes care of OS maintenance / patching, optimizations, setup, configuration, monitoring, failure recovery and backup

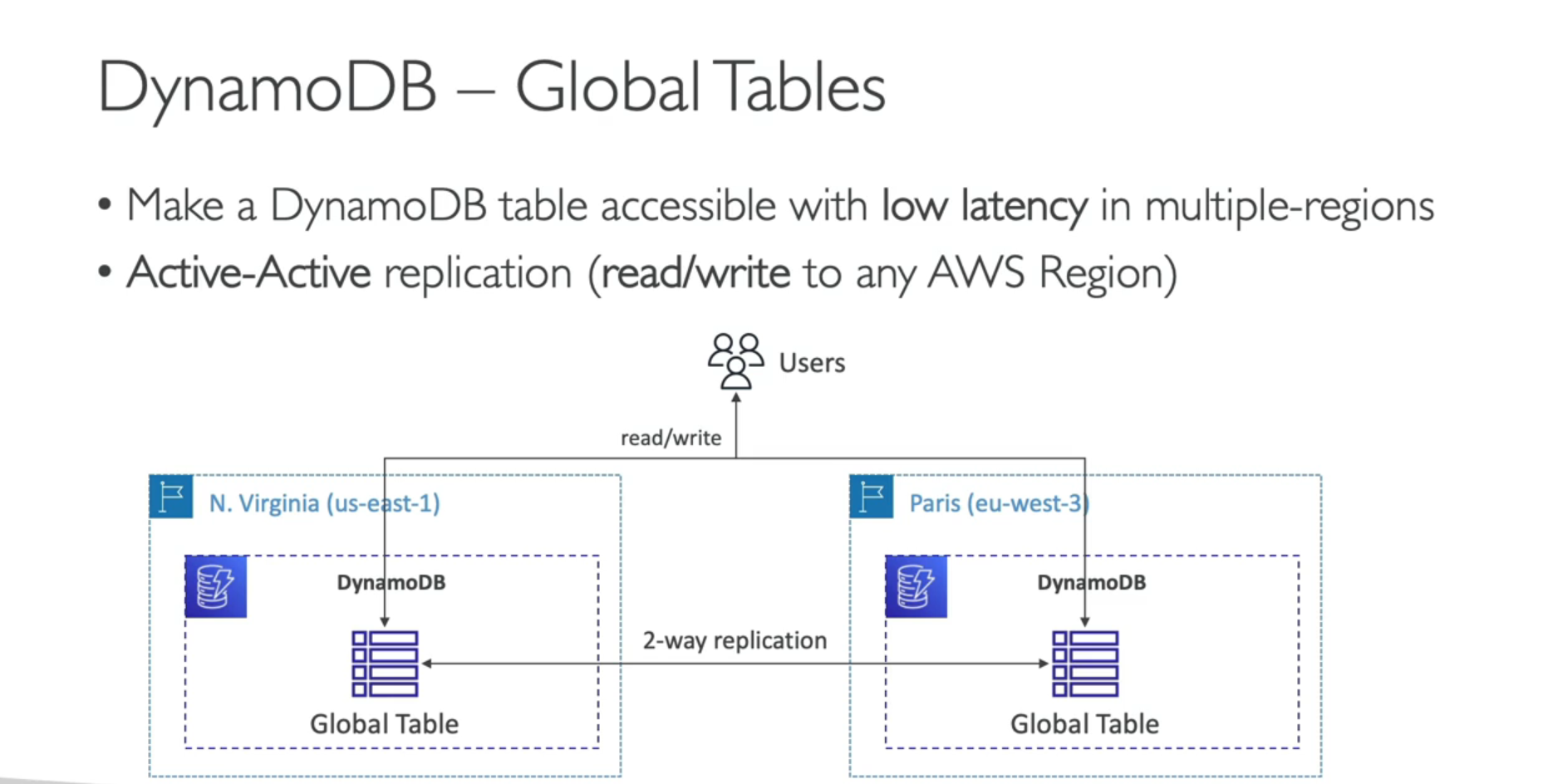


**DynamoDB**

* Fully Managed Highly available with replication across 3 AZ
* NoSQL database - not a relational database
* Scales to massive workloads, distributed “serverless” database
* Millions of requests per seconds, trillions of row, 100s of TB of storage
* Fast and consistent in performance
* Single-digit millisecond latency – low latency retrieval
* Integrated with IAM for security, authorization and administration
* Low cost and auto scaling capabilities
* Standard & Infrequent Access (IA) Table Class

DYNAMO DB ACCELATOR

Amazon DynamoDB Accelerator (DAX) is a fully managed, in-memory cache specifically designed to boost the performance of Amazon DynamoDB. It can significantly improve read performance, bringing response times down from milliseconds to microseconds, even when handling millions of requests per second.

Imagine a global app with users everywhere. Regular DynamoDB might slow things down because users access data far away. Global tables fix this by creating copies of your data in different regions, like having a local copy of a book in different libraries.

**Redshift Overview**

* Redshift is based on PostgreSQL, but it’s not used for OLTP (Online Transactional Processing)
* It’s OLAP – online analytical processing (analytics and data warehousing)
* Load data once every hour, not every second
* 10x better performance than other data warehouses, scale to PBs of data
* Columnar storage of data (instead of row based)
* Massively Parallel Query Execution (MPP), highly available
* Pay as you go based on the instances provisioned
* Has a SQL interface for performing the queries
* BI tools such as AWS Quicksight or Tableau integrate with it

AWS Redshift is a data warehouse service. Imagine you have a giant storeroom with all your business information scattered around in boxes. Redshift helps you organize all that information into a big, searchable database. This lets you analyze huge amounts of data quickly and easily, like finding patterns in sales figures or customer behavior.

**Amazon EMR**

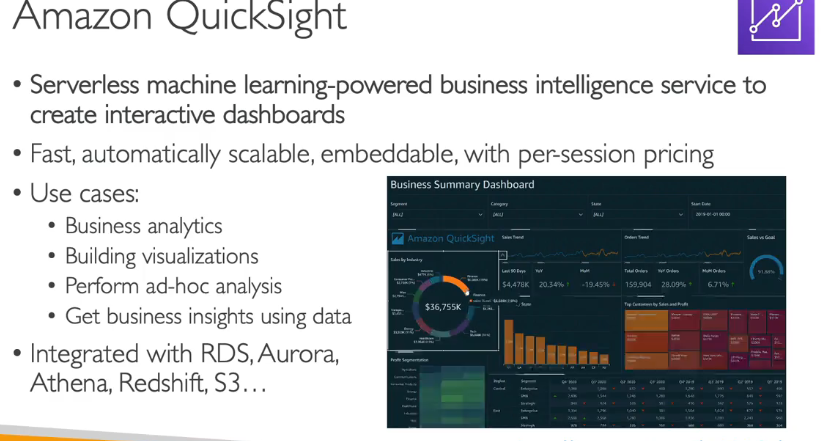
* EMR stands for “Elastic MapReduce”
* EMR helps creating clusters (Big Data) to analyze and process vast amount of data
* The clusters can be made of hundreds of EC2 instances
* Also supports Apache Spark, HBase, Presto, Flink
* EMR takes care of all the provisioning and configuration
* Auto-scaling and integrated with Spot instances
* Use cases: data processing, machine learning, web indexing, big data

**Amazon Athena**

* Serverless query service to analyze data stored in Amazon S3
* Uses standard SQL language to query the files
* Supports CSV, JSON, ORC, Avro, and Parquet (built on Presto)
* Pricing: $5.00 per TB of data scanned
* Use compressed or columnar data for cost-savings (less scan)
* Use cases: Business intelligence / analytics / reporting, analyze & query VPC Flow Logs, ELB Logs, CloudTrail trails, etc...
* **analyze data in S3 using serverless SQL, use Athena**

**Amazon QuickSight**

* Serverless machine learning-powered business intelligence service to create interactive dashboards
* Fast, automatically scalable, embeddable, with per-session pricing
* Use cases:
  + Business analytics
  + Building visualizations
  + Perform ad-hoc analysis
  + Get business insights using data
* Integrated with RDS, Aurora, Athena, Redshift, S3

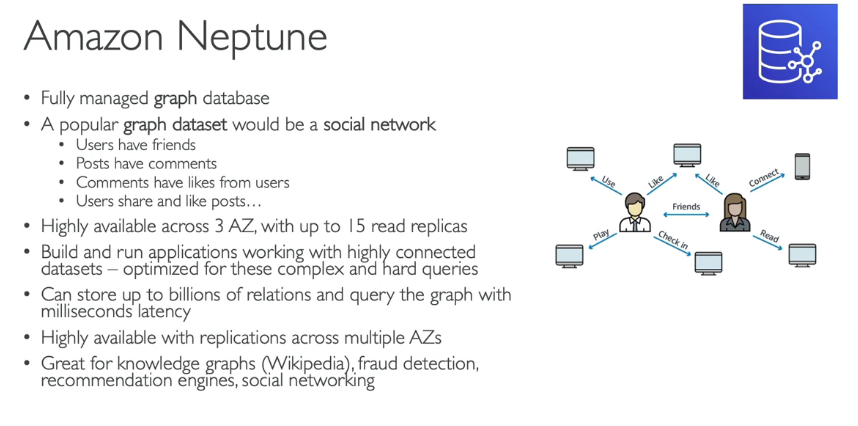


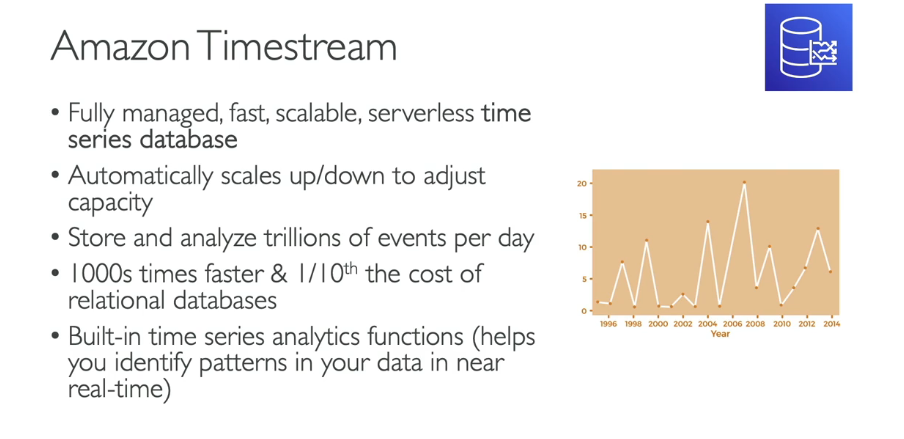
**DocumentDB**

* Aurora is an “AWS-implementation” of PostgreSQL / MySQL …
* DocumentDB is the same for MongoDB (which is a NoSQL database)
* MongoDB is used to store, query, and index JSON data
* Similar “deployment concepts” as Aurora
* Fully Managed, highly available with replication across 3 AZ
* Aurora storage automatically grows in increments of 10GB, up to 64 TB.
* Automatically scales to workloads with millions of requests per seconds

**Amazon Neptune**

* Fully managed graph database
* A popular graph dataset would be a social network
  + Users have friends
  + Posts have comments
  + Comments have likes from users
  + Users share and like posts…
* Highly available across 3 AZ, with up to 15 read replicas
* Build and run applications working with highly connected datasets – optimized for these complex and hard queries
* Can store up to billions of relations and query the graph with milliseconds latency
* Highly available with replications across multiple AZs
* Great for knowledge graphs (Wikipedia), fraud detection, recommendation engines, social networking





Amazon Timestream

It's a powerful tool for working with time series data, which is data that's collected over time at regular intervals

**Amazon QLDB**

* QLDB stands for ”Quantum Ledger Database”
* A ledger is a book **recording financial transactions**
* Fully Managed, Serverless, High available, Replication across 3 AZ
* Used to **review history of all the changes made to your application data** over time
* **Immutable** system: no entry can be removed or modified, cryptographically verifiable
* 2-3x better performance than common ledger blockchain frameworks, manipulate data using SQL
* Difference with Amazon Managed Blockchain: no decentralization component, in accordance with financial regulation rules

**Amazon Managed Blockchain**

* Blockchain makes it possible to build applications where multiple parties can execute transactions without the need for a trusted, central authority.
* Amazon Managed Blockchain is a managed service to:
  + Join public blockchain networks
  + Or create your own scalable private network
* Compatible with the frameworks Hyperledger Fabric & Ethereum

**AWS Glue**

* Managed extract, transform, and load (ETL) service
* Useful to prepare and transform data for analytics
* Fully serverless service
* Glue Data Catalog: catalog of datasets
  + can be used by Athena, Redshift, EMR

**DMS - Database Migration Service**

* Quickly and securely migrate databases to AWS, resilient, self healing
* The source database remains available during the migration
* Supports:
  + Homogeneous migrations: ex Oracle to Oracle
  + Heterogeneous migrations: ex Microsoft SQL Server to Aurora

