Databases

>Relational Database Service

RDS offers a managed and highly scalable database environment for most popular relational database engines (including MySQL, MariaDB, and Oracle).

The Amazon Relational Database Service (RDS) is Amazon's managed relational database service. RDS lets you provision a number of popular relational database management systems (RDBMSs) including Microsoft SQL Server, Oracle, MySQL, and PostgreSQL

When you create an RDS database instance, Amazon sets up one or more compute instances and takes care of installing and configuring the RDBMS of your choice.

Automated patching, backups, redundancy, failover and disaster recovery. RDS will automatically fail over to the standby during a failure so database operations can resume quickly without admin intervention.

Joining multiple tables to get the bigger picture. RDS is generally used for online transaction processing (OLTP) workloads

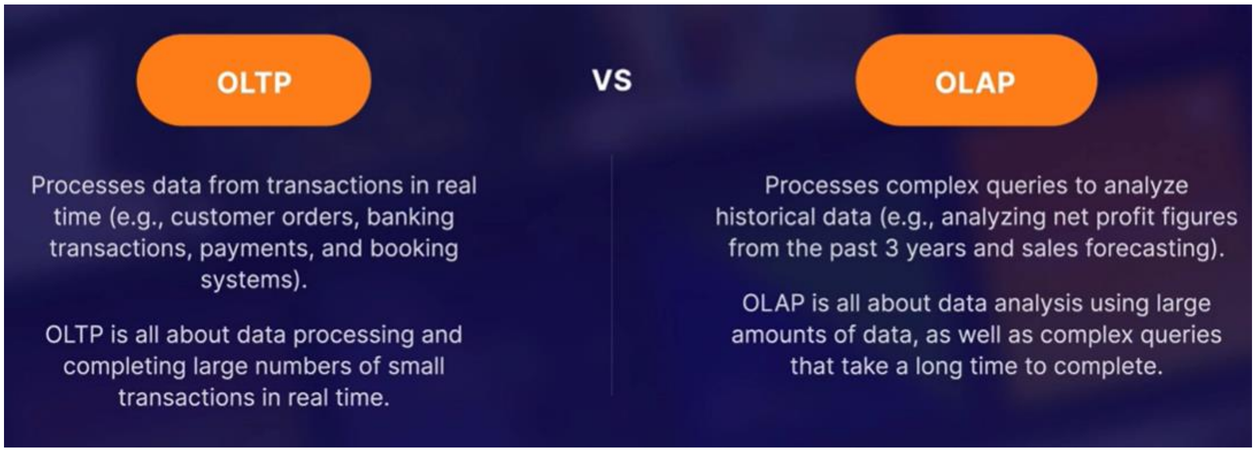
Able to launch read replicas across regions, which is faster for faster querying

>RDS read replicas have their own endpoints

RDS can also perform manual or automatic EBS snapshots that you can easily restore to new RDS instances. RDS can also handle the hard work of installing patches and upgrades during scheduled maintenance windows.

Up and running in minutes, mult-AZ, automated failover capability, automated backups,

A manual install in your own data center could take 8 days or longer. But with the great thing about RDS you can have it up and running in 5-10 mins.



OLTP

VS

OLAP

Processes data from transactions in real time (e.g., customer orders, banking transactions, payments, and booking systems).

OLTP is all about data processing and completing large numbers of small transactions in real time.

Processes complex queries to analyze historical data (e.g., analyzing net profit figures from the past 3 years and sales forecasting).

OLAP is all about data analysis using large amounts of data, as well as complex queries that take a long time to complete.

*RDS is not suitable for analyzing large amounts of data. Use a data warehouse like redshift which is optimized for OLAP

Anything about OLAP & OLTP

- Think that online transactions are through RDS (OLTP)
- Product that is called RedShift (OLAP)

Amazon RDS supports the following six database engines:

- MySQL
- MariaDB
- Oracle
- PostgreSQL
- Microsoft SQL Server
- Amazon Aurora

- Aurora
  - With the exception of Amazon Aurora, these database engines are either open source or commercially available products found in many data center environments. Amazon Aurora is a proprietary database designed for RDS, but it's compatible with existing MySQL (5 read), Aurora Replicas (15 read) and PostgreSQL databases (5 read)
  - If you use the Amazon Aurora database engine—Amazon's proprietary database engine designed for and available exclusively with RDS—you can take advantage of additional benefits when using multi-AZ. When you use Aurora, your RDS instances are part of an Aurora cluster.
  - 5x faster than normal MySQL and 3x faster than normal prostgreSQL
  - Managed by RDS
  - Continuous backup to Amazon S3
  - 2 copies of your data are contained in each AZ, with a min of 3 AZ. 6 copies of your data
  - Compute resources can scale up to 96 vCPUs and 768 GB of memory
  - Point-in-time recovery
  - Self healing
  - Aurora stores copies of the data across multiple Availability Zones in a single AWS Region.
  - Aurora Serverless is a supported feature of Aurora.
  - Automatic Backups is a supported feature of Aurora

Aurora Serverless

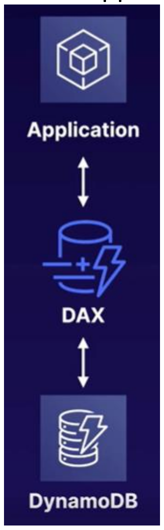
An Aurora Serverless DB cluster automatically starts up, shuts down, and scales capacity up or down based on your applications needs. Might be some questions of spiky workloads, you want to look at this

Use Cases

- Provides relatively simple, cost-effective option for infrequent, intermittent, or unpredictiable workloads.

- Dynamo DB
  - DynamoDB is Amazon's managed nonrelational database service. It's designed for highly transactional applications that need to read from or write to a database tens of thousands of times a second.
  - analogous to a row or record in a relational database. DynamoDB stores items in tables.
  - Other than the primary key, an item doesn't have to have any particular attributes.

        - This flexibility makes DynamoDB the database of choice for applications that need to store a wide variety of data without having to know the nature of that data in advance.
    - Non-Relational databases such as DynamoDB are designed to scale horizontally by spreading your data across more partitions, allowing for thousands of reads and writes per second
    - DynamoDB stores data as items in tables.
    - Each item must have primary key whose values are unique within the table. This is how DynamoDB uniquely identifies an item.
    - Great for new product launches
    - Pay-per-request pricing, but you pay more per request than with provisioned capacity
    - DynamoDB uses the primary key to distribute items across different partitions. The number of partitions allocated to a table depends on the number of WCU and RCU you configure.
    - DynamoDB is a NoSQL database
    - The minimum monthly availability for DynamoDB is 99.99 percent in a single Region. It's not 99.95 percent, 99.9 percent, or 99.0 percent.
- DynamoDB Accelerator (DAX)
    - Fully managed, highly available, in-memory cache
    - 10x performance improvement
    - No need for developers to manage caching logic (This is done by DynamoDB)
    - Caching with DAX
        - The application speaks directly to DAX to see if the information is cached inside of DAX > If not in cached in DAX it will basically interrogate DynamoDB and bring it up to DAX and back to the application



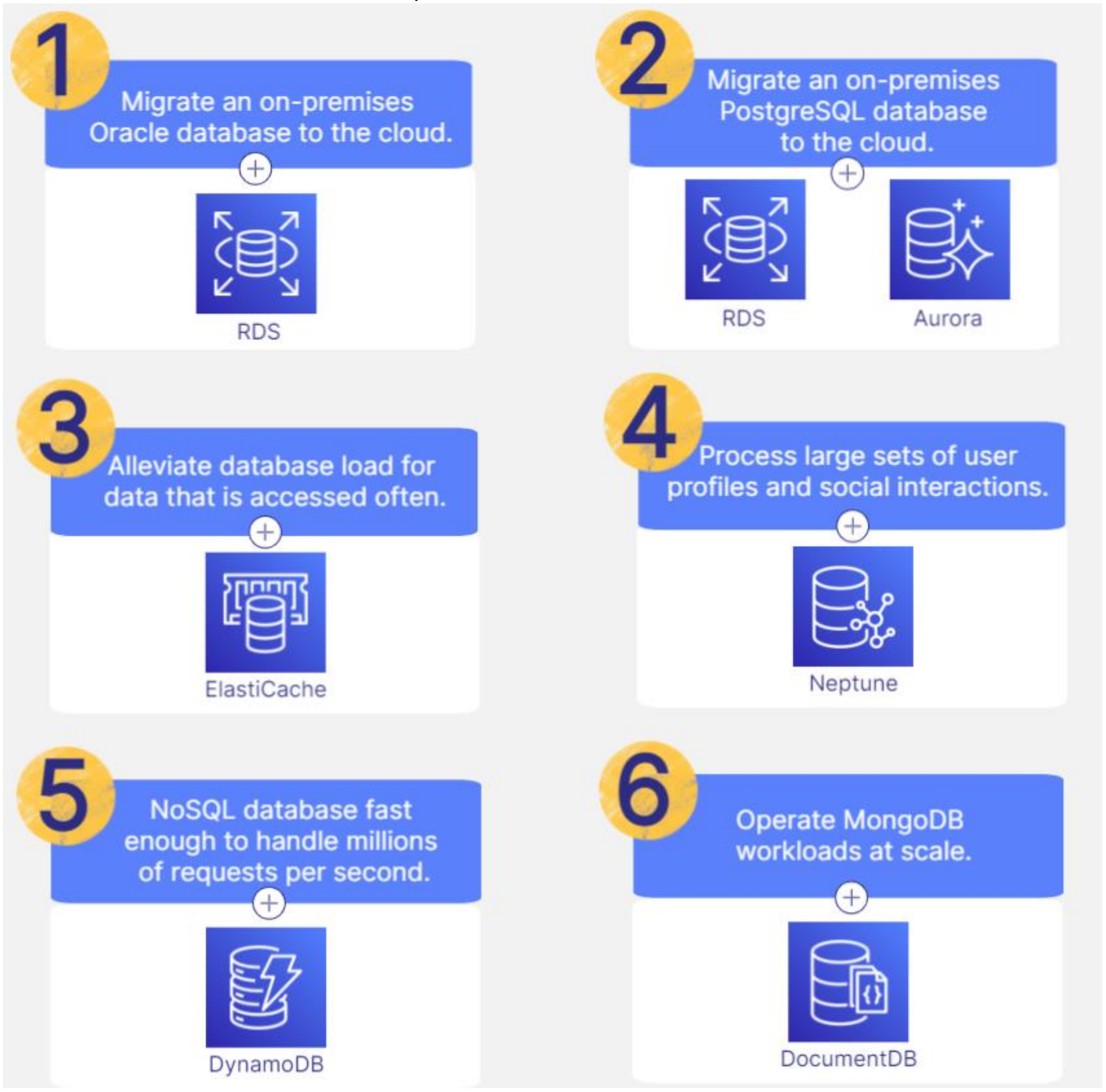
    - Security in DynoDB
        - Encryption at rest using KMS
        - CloudWatch and CloudTraill
        - Site-to-site VPN

- Document database
    - MongoDB compatible
- ElastiCache

- Fully managed in-memory datastore compatible with redis or memcached
  - Data can be lost because it 's stored in-memory
- High performance and low latency

- Neptune
  - Graph database service
  - Highly connected datasets like social media sites

Down below are what are the best option for each use case.


1
Migrate an on-premises Oracle database to the cloud.
RDS
2
Migrate an on-premises PostgreSQL database to the cloud.
RDS
Aurora
3
Alleviate database load for data that is accessed often.
ElastiCache
4
Process large sets of user profiles and social interactions.
Neptune
5
NoSQL database fast enough to handle millions of requests per second.
DynamoDB
6
Operate MongoDB workloads at scale.
DocumentDB

Mult-AZ is for disaster recovery, not for improving performance, so you cannot connect to the standby when the primary database is active.

>Increasing Read Performance with Read Replicas

Read replicas is a read-only copy of your primary database. Which great for read-heavy workloads and takes the load off your primary database. Each replica has its own unique DNS endpoint. This can also be promoted to be their own databases.

*There may be scenarios where they try and trick you in between read replica and Multi-AZ

Multi-AZ is for disaster recovery only, and in the event of a failure, RDS will automatically fail over to the standby instance.

Read replicas are perfect for read-heavy workloads

Read-Replica is for boosting performance, scaling

- Requires automatic backups
- Multiple read replicas are supported. Up to 5 read replicas to each DB instance

> When Do We Use DynamoDB Transactions?

The ACID Diagram

Name	Includes/Function
Atomic	All changes to the data must be performed successfully or not at all.
Consistent	Data must be in a consistent state before and after the transaction
Isolated	No other process can change the data while the transaction is running
Durable	The changes made by a transaction must be persist

You can use ACID with DynamoDB, but you need to use DynamoDB transactions which provides developers atomicity, consistency, isolation and durability.

When using DynamoDB transactions

- Multiple "all-or-nothing" operations
- Financial transactions
- Fullfilling orders
- 3 options for reads: eventual, strong, and transactional consistency
- 2 options for writes: standard and transactional
- Up to 25 items of 4 MB of data

>Savings your data with DynamoDB backups

This on-demand backup and restore can do full backups at any time. Which has ZERO impact on table performance or availability. The consistency is within seconds and retained until deleted. Operates within the same region as the source table

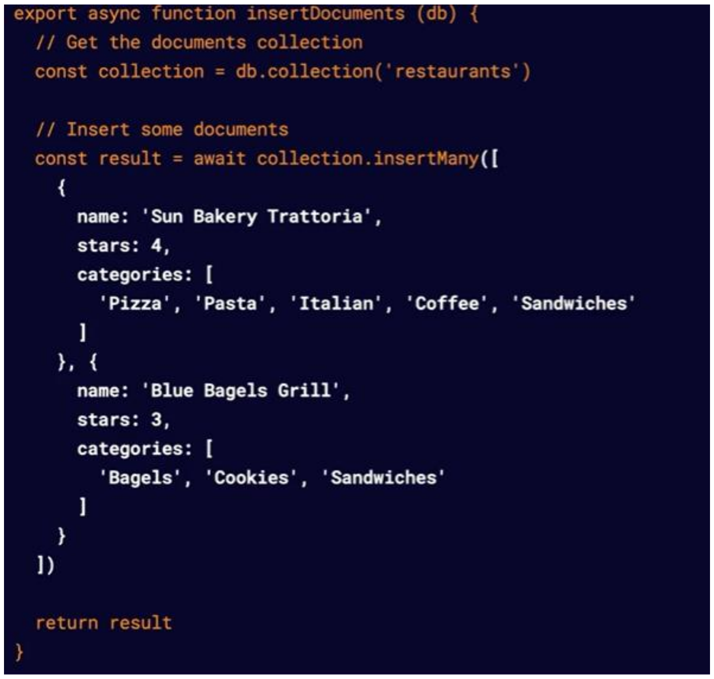
- Point-in-Time recovery (PITR)
    - Protects against accidental writes or deletes
    - Restore to any point in the last 35 days
    - Incremental backups
    - This is not enabled by default
    - Latest restorable: 5 mins in the past

>Operating MongoDB Compatible Databases in Amazon DocumentDB

What is MongoDB?

- MongoDB is a document database that allows for scalability and flexibility with your data as well as robust querying and indexing features.

Down below is an example of MongoDB



```
export async function insertDocuments (db) {
  // Get the documents collection
  const collection = db.collection('restaurants')

  // Insert some documents
  const result = await collection.insertMany([
    {
      name: 'Sun Bakery Trattoria',
      stars: 4,
      categories: [
        'Pizza', 'Pasta', 'Italian', 'Coffee', 'Sandwiches'
      ]
    }, {
      name: 'Blue Bagels Grill',
      stars: 3,
      categories: [
        'Bagels', 'Cookies', 'Sandwiches'
      ]
    }
  ])

  return result
}
```

Amazon DocumentDB

- This service allows you to run MongoDB on the AWS cloud. It’s a managed database service that scales with your workloads and safely and durably stores your database

>Taking your data global with DynamoDB Streams and Global Tables

- Streams
  - Time-ordered sequence of item-level changes in a table
  - Stored for 24 hours
  - So every time you make a change to a Dynamo DB table, that data is going to be stored sequentially in a stream record which is broken up into shards
  - Inserts, updates, and deletes
    - This gives us that time-ordered sequence
  - Combine with Lambda functions for functionality like stored procedures
- Global Tables
  - A way to replicate DynamoDB tables from one region to another
  - Based on DynamoDB streams, need this turned on
  - Globally distributed applications
  - Multi-region redundancy for disaster recovery or high availability
  - Replication latency under 1 second

>Running Apache Cassandra

- What is Cassandra?
  - This is a distributed database that uses NoSQL. It's primarily used for big data solutions. Enterprises such as Netflix use Cassandra on their backend.

So think of big database workloads that's distributed across many servers

- What is Amazon Keyspaces
  - Amazons Apache Cassandra database service. This allows you to run Cassandra workloads on AWS. Fully managed database service.

You should use Apache Cassandra because it is a fully managed database service. You don't need to worry about managing servers, software, patching, etc.

Keyspaces is serverless!!!!

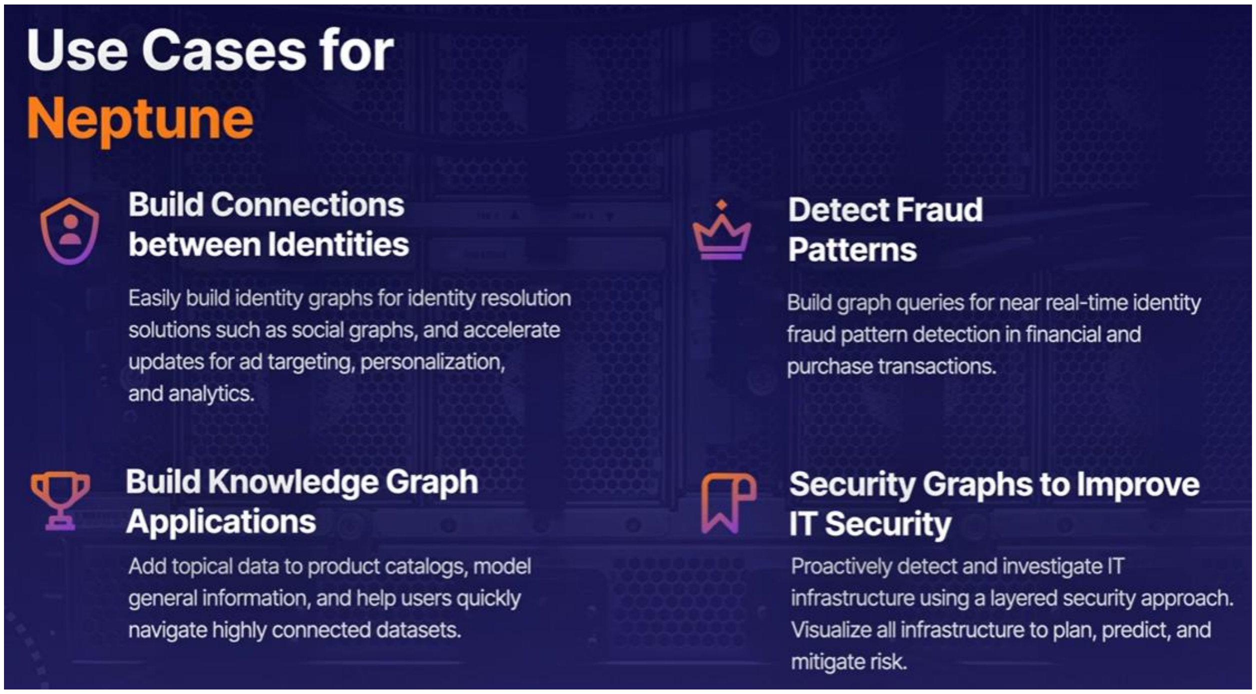
You also pay for only the resources you use. This service can automatically scale tables up and down in response to your applications.

> Implementing Graph Databases Using Amazon Neptune

What is a Graph Database?

- This database is where data is stored like a sketch of ideas on a whiteboard. Basically a database that stores nodes and relationships instead of tables of documents

- Netptune
  - Graph database service that is fast, reliable, fully managed that makes it easy to build and run applications



Use Cases for Neptune

Build Connections between Identities

Easily build identity graphs for identity resolution solutions such as social graphs, and accelerate updates for ad targeting, personalization, and analytics.

Detect Fraud Patterns

Build graph queries for near real-time identity fraud pattern detection in financial and purchase transactions.

Build Knowledge Graph Applications

Add topical data to product catalogs, model general information, and help users quickly navigate highly connected datasets.

Security Graphs to Improve IT Security

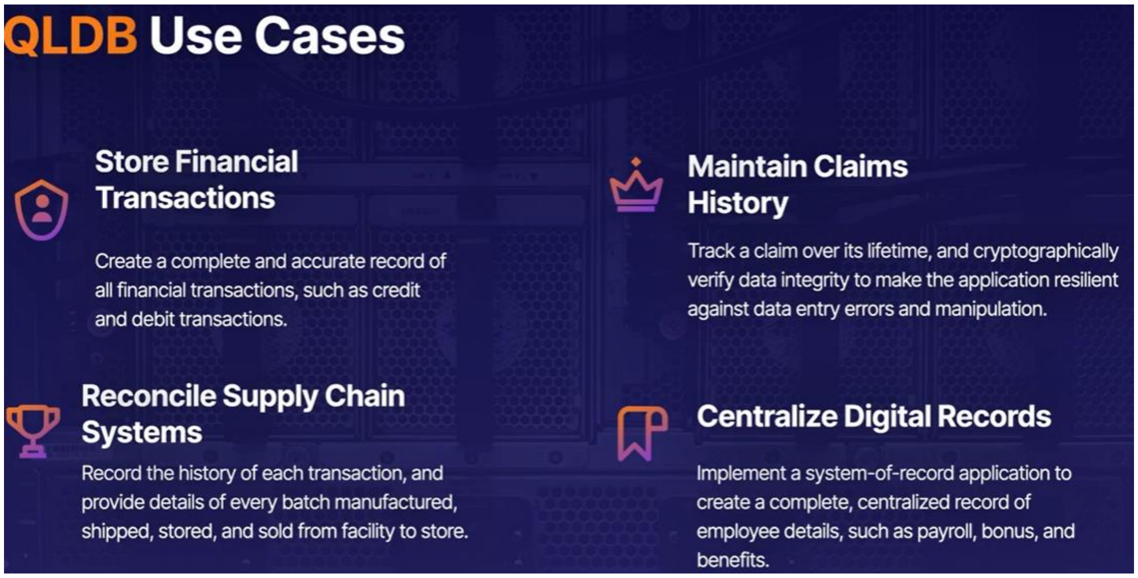
Proactively detect and investigate IT infrastructure using a layered security approach. Visualize all infrastructure to plan, predict, and mitigate risk.

> Leveraging Amazon Quantum Ledger Database (Amazon QLDB) for Ledger Databases

- Ledger Database
    - This is a NoSQL that is immutable, transparent and has cryptographically verifiable transaction log

Ledger is good for tracking items, boxes, shipping containers and more. Used for cryptocurrencies such as bitcoin. Even used in the pharmaceutical field to track creation and distribution of drugs.

- Amazon Quantum Ledger Database



QLDB Use Cases

Store Financial Transactions

Create a complete and accurate record of all financial transactions, such as credit and debit transactions.

Maintain Claims History

Track a claim over its lifetime, and cryptographically verify data integrity to make the application resilient against data entry errors and manipulation.

Reconcile Supply Chain Systems

Record the history of each transaction, and provide details of every batch manufactured, shipped, stored, and sold from facility to store.

Centralize Digital Records

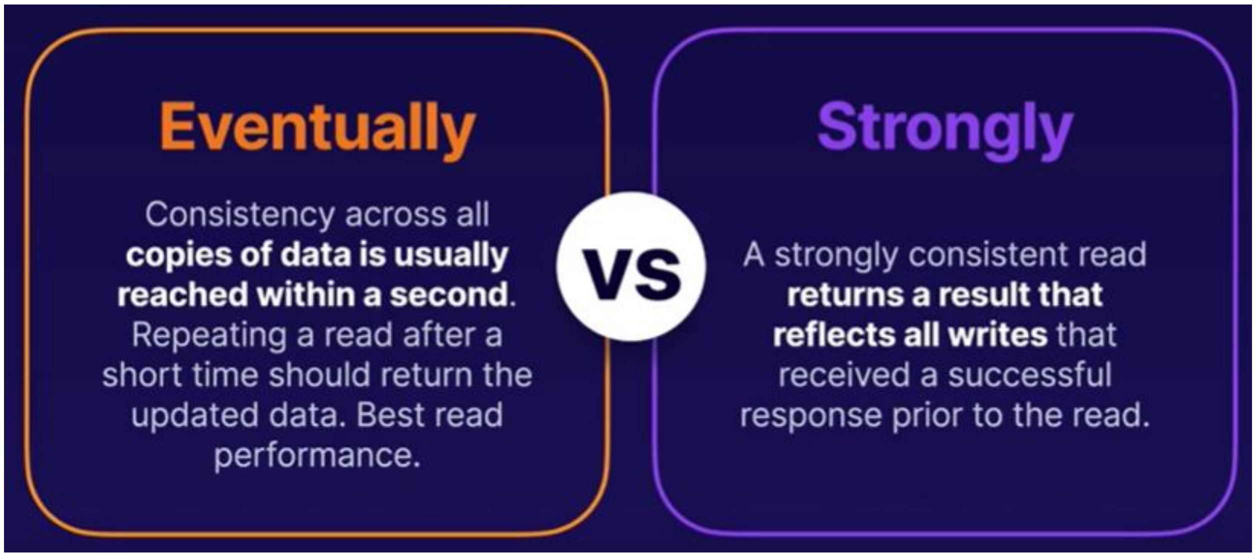
Implement a system-of-record application to create a complete, centralized record of employee details, such as payroll, bonus, and benefits.

> Analyzing Time-Series Data with Amazon Timestream

- Amazon Timestream
    - Serverless, fully managed database service for time-series data. Analyze trillions of events per day.

Exam Tips

1. Relational Database Service
   a. 6 RDS Datatypes
   b. RDS is for OLTP Workloads
   c. Not suitable for OLAP workloads
2. Aurora
   a. At least 6 copies of your data
   b. Share snapshots with other AWS accounts
   c. 3 types of replicas: MySQL (5 read), Aurora Replicas (15 read) and PostgreSQL databases (5 read)
   d. Automated backups turned on by default
   e. Aurora serverless if you want a simple, cost-effective option for infrequent, intermittent or unpredictable workloads
3. DynamoDB
   a. Stored on SSD storage
   b. Spread across 3 data centers
   c. Eventually consistent reads (default)
   d. Strongly consistent reads
   e. This on-demand backup and restore can do full backups at any time.



Eventually

Consistency across all **copies of data is usually reached within a second.** Repeating a read after a short time should return the updated data. Best read performance.

VS

Strongly

A strongly consistent read **returns a result that reflects all writes** that received a successful response prior to the read.

4. DynamoDB Transactions
   a. Any scenario questions that mentions ACID requirements, think of this
   b. Transactions provide developers, atomicity, consistency, isolation and durability across 1 or more tables within a single AWS account and region
   c. All-or-nothing transactions

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadConsistency.html

5. MongoDB database to AWS

- a. Steps below on the process
    - i. So you have MongoDB running on-prem
    - ii. So you might want to use AWS database migration service
        1. Remember this is a way of automating database migration from on-prem to AWS
    - iii. And then you have to select the database engine that you would do it which might be Amazon DocumentDB
- b. So any scenario questions about migrating MongoDB from on-prem to AWS think of Amazon DocumentDB

*Tip… you cannot run MongoDB workloads on DynamoDB because Dynamo is a NoSQL Database

6. Amazon Keyspaces
    - a. Any scenario questions about migrating a big data Cassandra cluster to AWS… Think Amazon Keyspaces talking about Cassandra workloads
7. Implementing Graph Databases
    - a. Neptune
        - i. This is often used as a distractor. If the scenario is NOT talking about graph databases. Do not select Neptune.
8. QLDB
    - a. This is often used as a distractor. Any scenario NOT talking about immutable databases. Do not select Amazon QLDB
9. Time-series Data
    - a. Any scenario question about where to store a large amount of time-series data… Think of Timestream