Databases

Relational vs. Non-Relational Databases

Relational Databases organize data into structured tables with rows and columns, using a rigid schema that enforces strict rules about data types and relationships. These SQL-based databases like Amazon RDS, Oracle, and MySQL excel at handling structured data and complex queries with joins across tables. They typically scale vertically by adding more power to a single server.

Non-Relational Databases offer flexible schema designs that can adapt to changing data needs, storing information as key-value pairs, documents, graphs, or wide columns. NoSQL databases such as DynamoDB and MongoDB scale horizontally across multiple servers and are ideal for unstructured or rapidly evolving data.

The key difference lies in how they manage and store data - relational databases prioritize consistency and structure, while non-relational databases emphasize flexibility and scalability.

Operational vs. Analytical Databases

Operational Databases (OLTP - Online Transaction Processing) are optimized for day-to-day business operations like processing customer orders or inventory updates. Services like Amazon RDS and DynamoDB handle high volumes of simple, fast transactions with frequent writes and reads.

Analytical Databases (OLAP - Online Analytical Processing) like Amazon Redshift specialize in complex queries across large historical datasets for business intelligence and reporting. While operational databases power live applications, analytical databases derive insights from aggregated data, often sourced from multiple operational systems.

AWS Database Services

AWS provides several managed database services to meet different needs:

- **EC2 Databases** - For those requiring full control, databases can be installed directly on EC2 instances, allowing customization of both the database software and underlying operating system.
- **Amazon RDS** - Offers managed relational databases supporting popular engines like PostgreSQL, MySQL and Oracle, handling maintenance tasks like backups and patching.
- **DynamoDB** - Delivers a fully managed NoSQL service with automatic scaling to accommodate unpredictable workloads.
- **Amazon Redshift** - Enables petabyte-scale analytics with columnar storage and massively parallel processing for data warehousing.
- **ElastiCache** - Improves application performance by providing managed Redis or Memcached in-memory caching to reduce database load.

Amazon RDS Deep Dive

Amazon RDS is a fully managed relational database service designed for online transaction processing (OLTP) workloads. As a managed service, RDS simplifies database administration tasks while providing scalable, high-performance database solutions in the cloud. The service runs on Amazon EC2 instances, requiring users to select appropriate instance types during setup that align with their performance requirements.

A single RDS instance can host multiple user-created databases, allowing for efficient resource utilization. For storage, RDS leverages Amazon EBS volumes, providing durable block storage with configurable performance characteristics. The service includes comprehensive backup capabilities, supporting both manual snapshots and automated backup systems to protect critical data.

Supported Database Engines

RDS offers support for multiple popular database engines:

- **Amazon Aurora** - AWS's proprietary engine, delivering high scalability and cost efficiency while maintaining compatibility with MySQL and PostgreSQL
- **MySQL** - Widely-used open-source relational database
- **PostgreSQL** - Open-source database with additional support for non-relational querying capabilities
- **Oracle Database** - Enterprise solution available with either bring-your-own-license or AWS-provided licensing
- **Microsoft SQL Server** - Available in various editions for enterprise needs
- **MariaDB** - Community-developed fork of MySQL under the GNU GPL license

Scaling and Availability

Vertical Scaling involves upgrading to larger instance types with more CPU, memory, and storage capacity, though this requires a database restart and results in temporary downtime.

Horizontal Scaling is achieved through read replicas, which allow distribution of read traffic across multiple instances while maintaining a single primary instance for write operations. These read replicas can be deployed in the same Availability Zone, different AZs, or even different regions.

Multi-AZ Deployment maintains a synchronous standby replica in a different Availability Zone, enabling automatic failover in case of primary instance failure with DNS endpoint redirection handled seamlessly.

Cross-Region Read Replicas provide disaster recovery across regions and can be promoted to primary status if needed for comprehensive business continuity.

RDS Backup & Maintenance

RDS offers automated daily backups (retained 0-35 days) and manual snapshots (kept indefinitely). Automated backups enable point-in-time recovery, while manual snapshots provide long-term retention. Backups occur during configurable windows - you can set specific times or let AWS choose.

For Single-AZ deployments, backups cause brief I/O suspension. Multi-AZ configurations are more resilient:

- **SQL Server**: Brief primary I/O pause
- **MySQL/MariaDB/PostgreSQL/Oracle**: Backups run on standby with no production impact

Maintenance (patches/updates) occurs during weekly windows you can customize. Multi-AZ minimizes downtime by updating standbys first before failover. Both backup and maintenance windows can be set to "no preference" for AWS-optimized scheduling.

Amazon RDS Security

Network Security

Amazon RDS databases operate within a Virtual Private Cloud (VPC) and are assigned private IP addresses for secure network isolation. While the service offers a "Publicly Accessible" option that enables internet connectivity, security best practices strongly recommend maintaining databases in private subnets and restricting access exclusively to authorized applications. Security groups serve as virtual firewalls, with a typical deployment utilizing two distinct groups: a dedicated RDS security group (e.g., "RDS-SG") that limits inbound traffic to specific database ports (such as 3306 for MySQL), and a separate application security group (e.g., "App-SG") assigned to EC2 instances that need database access. This layered approach ensures only permitted application servers can establish connections to the database tier.

Encryption in Transit & at Rest

For secure data transmission, RDS provides SSL/TLS encryption between applications and databases, which is enabled by default for most database engines and can be configured to enforce mandatory encrypted connections. Data at rest protection employs robust AES-256 encryption that safeguards not just the primary database storage but also extends to automated backups, manual snapshots, and read replicas. The AWS Key Management Service (KMS) centrally manages all encryption keys, with critical implementation details to note: encryption must be configured during the initial database creation and cannot be modified afterward, whether to enable it on an existing unencrypted instance or disable it on an encrypted one. For

Oracle and SQL Server deployments, Transparent Data Encryption (TDE) offers an additional storage-level encryption option, though with potential performance considerations.

Key Encryption Considerations

Several crucial architectural decisions surround RDS encryption. The system maintains strict consistency between primary databases and their read replicas - an encrypted primary necessitates encrypted replicas, and vice versa, with cross-region replicas requiring separate KMS keys in the destination region. The platform enforces encryption continuity for backups, preventing restoration of unencrypted snapshots to encrypted instances or encrypted snapshots to unencrypted instances. For organizations needing to transition existing unencrypted databases to encrypted ones, the process involves creating an unencrypted snapshot, copying it with encryption enabled, then restoring as a completely new database instance. This migration requires application updates to point to the new encrypted endpoint, as the encryption status of an existing database instance cannot be modified in-place. These design choices reflect AWS's security-first approach while providing clear migration paths for evolving compliance requirements.

Additional Important Points:

- **Read replicas**: Provide read-only copies of your primary database to offload read traffic
- **Cross-Region Replication**: Can be used for disaster recovery or lower-latency reads in different regions
- **Promotion Process**: Converts a replica into a standalone database when needed
- **RDS Operations**: Most modifications in RDS take several minutes to complete

Encrypting an Existing RDS Database

To encrypt an existing unencrypted RDS database:

1. Create a snapshot of your current database (which remains unencrypted)
2. Copy the snapshot and enable encryption during the copy process
3. Restore the encrypted snapshot to create a new database instance

Key points:

- Encryption can't be added to or removed from an existing database directly
- The restored encrypted database will have a new endpoint (applications must update their connection strings)
- Encryption status is permanent for snapshots and restored instances
- Use AWS KMS to manage your encryption keys

This process creates a secure copy of your data while maintaining the original unencrypted database until you're ready to switch over. Remember to clean up old snapshots and databases after migration to avoid unnecessary costs.

Amazon Aurora

Amazon Aurora is a high-performance relational database service within Amazon RDS, compatible with MySQL and PostgreSQL. Designed for speed and resilience, Aurora delivers up to 5x faster performance than MySQL and 3x faster than standard PostgreSQL, making it ideal for demanding workloads.

Architecture & Resilience

Aurora uses a distributed, fault-tolerant storage system that automatically scales up to 128TB per database instance. Data is replicated across multiple Availability Zones (AZs) with six copies (two per AZ) for redundancy. The architecture includes:

- A primary instance handling all write operations
- Aurora Replicas (up to 15 per region) for read scaling
- Self-healing storage that repairs disk failures automatically

Replication & High Availability

Aurora offers two types of replicas:

Aurora Replicas (In-Region)

- Provide low-latency reads (<10ms lag)
- Can be automatically promoted for failover
- Support auto-scaling to adjust replica count dynamically

MySQL Read Replicas (Cross-Region)

- Used for disaster recovery (DR) and global read scaling
- Have higher latency than Aurora Replicas
- Require manual promotion in case of failover

Global Database & Serverless Option

- **Global Database** enables cross-region clusters with fast replication for low-latency reads and failover capabilities.
- **Aurora Serverless** provides automatic scaling based on demand, eliminating the need for manual capacity management.

Aurora provides:

- **High Performance** – Optimized for fast transactions and analytics
- **Automated Backups** – Continuous backups to Amazon S3
- **Multi-AZ Resilience** – Automatic failover with minimal downtime
- **Cost Efficiency** – Pay only for the storage and compute used

Amazon Aurora Deployment Options

Amazon Aurora provides multiple deployment options designed to maximize availability, scalability, and performance across different workloads. These options span from regional high-availability configurations to global replication and serverless scalability.

1. In-Region Deployment with Aurora Replicas

Within a single AWS Region, Aurora ensures fault tolerance and high availability through multi-AZ replication and read scalability.

- Aurora uses a single logical volume with multiple data copies replicated across three Availability Zones (AZs).
- The primary instance handles all write operations and is accessed through the primary endpoint.
- You can create up to 15 Aurora Replicas for read scalability, each with its own independent endpoint.
- Replica lag is typically under 10 milliseconds, allowing near real-time reads.
- Replicas can be promoted to primary automatically or manually in case of a failure.
- Priority tiers can be set to control the order of failover, e.g., to prefer promotion in a different AZ than the failed primary.
- Aurora also supports auto-scaling of replicas based on CloudWatch metrics to handle fluctuating read workloads.

2. Cross-Region Replication (MySQL-style)

Aurora supports asynchronous cross-region replication using MySQL-based replication mechanisms.

- The primary region handles all writes.
- You can deploy read replicas in other AWS regions.
- These MySQL replicas provide global read access and disaster recovery options.
- This replication is asynchronous, suitable for eventual consistency needs.

3. Aurora Global Database

Aurora Global Database is optimized for applications requiring low-latency global reads and fast recovery from region failures.

- Write operations occur only in the primary region.

- Read operations can be performed from secondary regions using reader endpoints.
- Uses the Aurora storage layer for fast, efficient cross-region replication.
- Secondary region clusters can be promoted to full read-write capability in under one minute.
- Write forwarding is supported—write requests sent to a secondary region can be forwarded to the primary region automatically.

4. Aurora Serverless

Aurora Serverless offers on-demand automatic scaling for database capacity, ideal for unpredictable or intermittent workloads.

- Auto-scales based on application demand using Aurora Capacity Units (ACUs).
- Each ACU includes 2 GB of memory and proportional CPU.
- AWS maintains a warm pool of capacity for rapid scaling.
- Applications connect via a router fleet that manages access to the right database capacity.
- Ideal use cases include:
  - Infrequently used applications
  - New or unknown workloads
  - Variable or unpredictable usage patterns
  - Development and testing environments
  - Multi-tenant applications

Amazon RDS Proxy

Amazon RDS Proxy acts as an intermediary layer between your applications and Amazon RDS/Aurora databases, serving as a fully managed database proxy service. Its primary purpose is to efficiently manage database connections, particularly in serverless architectures where applications like AWS Lambda functions may experience rapid and unpredictable scaling. When numerous Lambda functions attempt to connect directly to a database simultaneously, they can overwhelm the database with connection requests, consuming valuable CPU and memory resources. RDS Proxy solves this by maintaining a pool of established database connections that multiple application instances can share, preventing connection storms and improving overall database performance.

The service provides several key benefits beyond connection pooling. It enhances availability by automatically handling failovers between database instances without dropping application connections. From a security perspective, RDS Proxy integrates with AWS Identity and Access Management (IAM) and AWS Secrets Manager to centralize and secure database credentials. This setup eliminates the need to embed credentials in application code while enabling seamless credential rotation. The proxy also reduces the operational overhead of managing database connections, as it automatically scales to handle connection requests and optimizes connection reuse.

RDS Proxy is particularly valuable for serverless applications that experience variable workloads, microservices architectures with many concurrent connections, and applications requiring high availability. By sitting between your applications and database, it essentially acts as a traffic controller - managing, optimizing, and securing the flow of database connections while insulating your database from the impact of application scaling patterns. This results in more efficient resource utilization, improved application responsiveness, and greater database stability.

AWS Database Practice Quizzes

Relational vs. Non-Relational Quiz

Q1: Which database type uses tables with strict schemas?

- A) NoSQL
- B) SQL

Answer: B (SQL)

Q2: DynamoDB is an example of a:

- A) Relational database
- B) Key-value store

Answer: B (Key-value store)

Q3: Which database type scales horizontally by adding more servers?

- A) SQL
- B) NoSQL

Answer: B (NoSQL)

Q4: What is the primary query language for relational databases?

- A) GraphQL
- B) SQL

Answer: B (SQL)

Q5: Which scaling approach do relational databases typically use?

- A) Horizontal scaling

- B) Vertical scaling

Answer: B (Vertical scaling)

Q6: NoSQL databases are best for:

- A) Structured data with fixed schema
- B) Unstructured or rapidly changing data

Answer: B (Unstructured or rapidly changing data)

Operational vs. Analytical Quiz

Q1: Redshift is optimized for:

- A) OLTP
- B) OLAP

Answer: B (OLAP)

Q2: Which service handles eCommerce transactions?

- A) RDS
- B) Redshift

Answer: A (RDS)

Q3: OLTP databases are best suited for:

- A) Complex analytics
- B) Day-to-day transactions

Answer: B (Day-to-day transactions)

Q4: Where would you store aggregated historical data for reporting?

- A) DynamoDB
- B) Redshift

Answer: B (Redshift)

Q5: What does OLTP stand for?

- A) Online Transaction Processing

- B) Offline Transaction Processing

Answer: A (Online Transaction Processing)

Q6: Which system would you use for real-time inventory management?

- A) OLAP system
- B) OLTP system

Answer: B (OLTP system)

AWS Services Quiz

Q1: Which is serverless?

- A) RDS
- B) DynamoDB

Answer: B (DynamoDB)

Q2: ElastiCache improves:

- A) Data warehousing
- B) Query performance

Answer: B (Query performance)

Q3: Which service would you use for a managed PostgreSQL database?

- A) Amazon RDS
- B) Amazon Redshift

Answer: A (Amazon RDS)

Q4: What is the primary benefit of running a database on EC2?

- A) Automatic scaling
- B) Full control over configuration

Answer: B (Full control over configuration)

Q5: Which AWS service provides in-memory caching?

- A) ElastiCache

- B) Aurora

Answer: A (ElastiCache)

Q6: What type of storage does Amazon Redshift use for analytics?

- A) Row-based
- B) Columnar

Answer: B (Columnar)

RDS Practice Quiz

Q1: What type of workloads is RDS primarily designed for?

- A) OLAP (Analytical Processing)
- B) OLTP (Transaction Processing)
- C) Batch Processing
- D) Real-time Streaming

Answer: B) OLTP (Transaction Processing)

Q2: Which storage system does RDS use for its databases?

- A) Amazon S3
- B) Amazon EBS
- C) Amazon EFS
- D) Instance Store

Answer: B) Amazon EBS

Database Engines

Q3: Which RDS database engine is AWS-proprietary but MySQL/PostgreSQL compatible?

- A) Oracle
- B) Aurora
- C) MariaDB
- D) SQL Server

Answer: B) Aurora

Q4: True or False: PostgreSQL in RDS can handle both relational and JSON/document queries.

Answer: True

Scaling & Availability

Q5: What type of scaling requires a database restart in RDS?

- A) Adding read replicas
- B) Vertical scaling (instance type upgrade)
- C) Enabling Multi-AZ
- D) Storage auto-scaling

Answer: B) Vertical scaling (instance type upgrade)

Q6: How does Multi-AZ improve availability?

- A) By distributing reads across replicas
- B) By maintaining a sync standby in another AZ
- C) By caching frequent queries
- D) By reducing storage costs

Answer: B) By maintaining a sync standby in another AZ

Backups & Recovery

Q7: Which backup method in RDS is automatic?

- A) Manual snapshots
- B) Point-in-time recovery
- C) Automated daily backups
- D) Export to S3

Answer: C) Automated daily backups

Q8: For cross-region disaster recovery, what RDS feature would you use?

- A) Multi-AZ
- B) Read replicas
- C) Elastic Cache
- D) Storage gateway

Answer: B) Read replicas

Scenario Questions

Q9: Your e-commerce application needs to handle sudden spikes in read traffic. Which RDS feature helps most?

- A) Vertical scaling
- B) Read replicas
- C) Multi-AZ
- D) Larger EBS volumes

Answer: B) Read replicas

Q10: You need to minimize downtime during database maintenance. Which two features should you enable?

- A) Read replicas + Larger instance type
- B) Multi-AZ + Automated backups
- C) Cross-region replicas + Manual snapshots
- D) None of the above

Answer: B) Multi-AZ + Automated backups

Backups & Recovery Quiz

Q1: What is the maximum retention period for RDS automated backups?

- A) 7 days
- B) 15 days
- C) 35 days
- D) 60 days

Answer: C) 35 days

Q2: Which RDS feature enables restoration to any second within your backup window?

- A) Manual snapshots
- B) Point-in-time recovery
- C) Read replicas
- D) Multi-AZ failover

Answer: B) Point-in-time recovery

Q3: What happens to manual snapshots if you don't delete them?

- A) Auto-delete after 35 days
- B) Persist indefinitely
- C) Convert to automated backups
- D) Archive to S3 Glacier

Answer: B) Persist indefinitely

Q4: Which deployment type avoids production I/O suspension during backups for PostgreSQL?

- A) Single-AZ
- B) Multi-AZ
- C) Cross-region replica
- D) Aurora Serverless

Answer: B) Multi-AZ

Q5: Where are RDS automated backups stored?

- A) EBS volumes
- B) S3
- C) EFS
- D) Instance store

Answer: B) S3

Read Replicas & Operations Quiz

Q1: What is the primary purpose of an RDS read replica?

- A) Increase write capacity
- B) Offload read traffic from the primary database
- C) Replace automated backups
- D) Reduce storage costs

Answer: B) Offload read traffic from the primary database

Q2: Can a read replica be promoted to a standalone database?

- A) No, read replicas are permanently read-only
- B) Yes, but it requires downtime
- C) Yes, and it becomes an independent database
- D) Only if it's in the same region as the primary

Answer: C) Yes, and it becomes an independent database

Q3: How does cross-region replication benefit an RDS deployment?

- A) Provides automatic failover during outages
- B) Reduces read latency for geographically distributed users

- C) Eliminates the need for backups
- D) Allows changing encryption status

Answer: B) Reduces read latency for geographically distributed users

Q4: True or False: Most RDS modifications (like scaling) happen instantly without downtime.

- A) True
- B) False

Answer: B) False

Encrypting an Existing RDS Database

Q5: What is the first step to encrypt an unencrypted RDS database?

- A) Enable encryption in the RDS console
- B) Take an unencrypted snapshot
- C) Restore from an existing encrypted backup
- D) Contact AWS Support

Answer: B) Take an unencrypted snapshot

Q6: Can you disable encryption on an encrypted RDS snapshot?

- A) Yes, via the AWS Management Console
- B) No, encryption is permanent once applied
- C) Only if the snapshot is less than 30 days old
- D) Only for MySQL databases

Answer: B) No, encryption is permanent once applied

Q7: What must applications do after restoring an encrypted RDS database?

- A) Nothing, the endpoint remains the same
- B) Update to use the new database endpoint
- C) Manually decrypt the data
- D) Reconfigure the security group

Answer: B) Update to use the new database endpoint

Q8: Which AWS service manages encryption keys for RDS?

- A) AWS Secrets Manager
- B) AWS KMS (Key Management Service)
- C) AWS Certificate Manager

- D) AWS CloudHSM

Answer: B) AWS KMS (Key Management Service)

Amazon Aurora Practice Quiz

Core Concepts

Q1: Which database engines is Amazon Aurora compatible with?

- A) MySQL and PostgreSQL
- B) Oracle and SQL Server
- C) MongoDB and DynamoDB
- D) Only PostgreSQL

Answer: A) MySQL and PostgreSQL

Q2: How does Aurora's performance compare to standard MySQL?

- A) 2x slower
- B) Equal performance
- C) Up to 5x faster
- D) Only optimized for analytics

Answer: C) Up to 5x faster

Architecture & Resilience

Q3: How many copies of data does Aurora maintain across AZs?

- A) 2
- B) 3
- C) 6 (two per AZ)
- D) 15

Answer: C) 6 (two per AZ)

Q4: What is the maximum storage capacity per Aurora database instance?

- A) 64TB
- B) 128TB
- C) 256TB
- D) Unlimited

Answer: B) 128TB

Q5: What feature allows Aurora to automatically recover from disk failures?

- A) Multi-AZ deployment
- B) Self-healing storage
- C) Read replicas
- D) Global Database

Answer: B) Self-healing storage

Replication & High Availability

Q6: How many Aurora Replicas can be created within a single region?

- A) 5
- B) 10
- C) 15
- D) Unlimited

Answer: C) 15

Q7: What is the typical replication lag for Aurora Replicas?

- A) <10 milliseconds
- B) 1-5 seconds
- C) 10-30 seconds
- D) Variable

Answer: A) <10 milliseconds

Q8: Which type of replica supports cross-region replication?

- A) Aurora Replicas
- B) MySQL Read Replicas
- C) PostgreSQL Replicas
- D) Both A and B

Answer: B) MySQL Read Replicas

Q9: What is required to promote a MySQL Read Replica to primary?

- A) Automatic failover
- B) Manual intervention
- C) AWS Support ticket
- D) It cannot be promoted

Answer: B) Manual intervention

Global Database & Serverless

Q10: What does Aurora Global Database provide?

- A) Cross-region replication with low-latency reads
- B) Only disaster recovery (no reads)
- C) Manual scaling
- D) Reduced storage costs

Answer: A) Cross-region replication with low-latency reads

Q11: Which Aurora option automatically adjusts capacity based on demand?

- A) Multi-AZ
- B) Aurora Serverless
- C) Global Database
- D) Reserved Instances

Answer: B) Aurora Serverless

Features & Benefits

Q12: Where are Aurora's automated backups stored?

- A) Amazon EBS
- B) Amazon S3
- C) Local instance storage
- D) AWS Backup Vault

Answer: B) Amazon S3

Q13: How does Aurora ensure high availability during AZ failures?

- A) Manual replica promotion
- B) Automatic failover to standby
- C) Pausing the database
- D) Restoring from S3

Answer: B) Automatic failover to standby

Q14: True or False: Aurora charges for idle compute capacity.

- A) True
- B) False (pay only for used resources)

Answer: B) False

Amazon Aurora Deployment Options Quiz

Q1: How many Aurora Replicas can you create within a single AWS Region?

- A) 5
- B) 10
- C) 15
- D) Unlimited

Answer: C) 15

Q2: What is the typical replica lag for Aurora Replicas?

- A) <1 second
- B) <10 milliseconds
- C) 1-5 seconds
- D) 5-10 seconds

Answer: B) <10 milliseconds

Q3: Which Aurora feature allows automatic scaling of read replicas based on workload?

- A) Global Database
- B) Priority tiers
- C) Auto-scaling of replicas
- D) Write forwarding

Answer: C) Auto-scaling of replicas

Q4: What type of replication does Aurora use for cross-region MySQL replicas?

- A) Synchronous
- B) Asynchronous
- C) Semi-synchronous
- D) Bidirectional

Answer: B) Asynchronous

Q5: How quickly can an Aurora Global Database secondary region be promoted to primary?

- A) Instantly
- B) Under one minute

- C) 5-10 minutes
- D) 15-30 minutes

Answer: B) Under one minute

Q6: What does Aurora Serverless use to measure capacity?

- A) Compute Units
- B) Aurora Capacity Units (ACUs)
- C) vCPUs
- D) Memory Units

Answer: B) Aurora Capacity Units (ACUs)

Q7: Which Aurora feature allows write requests to be sent to a secondary region?

- A) Multi-master
- B) Write forwarding
- C) Cross-region replication
- D) Priority promotion

Answer: B) Write forwarding

Q8: How many copies of data does Aurora maintain across Availability Zones?

- A) 2
- B) 3
- C) 6
- D) 15

Answer: C) 6 (two per AZ)

Q9: What is the primary benefit of Aurora Serverless?

- A) Fixed pricing
- B) Manual scaling control
- C) Automatic capacity scaling
- D) Cross-region replication

Answer: C) Automatic capacity scaling

Q10: Which deployment option is best for applications with unpredictable workloads?

- A) In-Region with Aurora Replicas
- B) Cross-Region Replication
- C) Global Database

- D) Aurora Serverless

Answer: D) Aurora Serverless

Q11: What controls the order of replica promotion during failover?

- A) Auto-scaling rules
- B) Priority tiers
- C) ACU allocation
- D) Replication lag

Answer: B) Priority tiers

Q12: Where do all write operations occur in a Global Database setup?

- A) Primary region only
- B) Any region
- C) Secondary regions only
- D) Rotates between regions

Answer: A) Primary region only

Q13: What component manages connections in Aurora Serverless?

- A) Primary endpoint
- B) Reader endpoint
- C) Router fleet
- D) Storage layer

Answer: C) Router fleet

Q14: How many Availability Zones does Aurora's storage span by default?

- A) 1
- B) 2
- C) 3
- D) 4

Answer: C) 3

Q15: Which feature is NOT available with Aurora Serverless?

- A) Automatic scaling
- B) Cross-region replication
- C) Pay-per-use billing
- D) Warm pool of capacity

Answer: B) Cross-region replication

Amazon RDS Proxy Quiz

Q1: What is the primary purpose of Amazon RDS Proxy?

- A) To replace Amazon RDS entirely
- B) To cache database query results
- C) To manage and pool database connections
- D) To migrate databases between regions

Answer: C) To manage and pool database connections

Q2: Which AWS service benefits MOST from using RDS Proxy?

- A) Amazon EC2 (static workloads)
- B) AWS Lambda (serverless applications)
- C) Amazon S3 (object storage)
- D) AWS Batch (batch processing)

Answer: B) AWS Lambda (serverless applications)

Q3: How does RDS Proxy improve database performance?

- A) By compressing data before storage
- B) By maintaining a pool of reusable connections
- C) By automatically scaling database storage
- D) By creating read replicas

Answer: B) By maintaining a pool of reusable connections

Q4: Which feature does RDS Proxy provide for high availability?

- A) Automatic read replica creation
- B) Seamless failover without dropping connections
- C) Cross-region replication
- D) Automated database patching

Answer: B) Seamless failover without dropping connections

Q5: How does RDS Proxy enhance security?

- A) By encrypting data at rest automatically
- B) By integrating with IAM and Secrets Manager

- C) By blocking all public internet access
- D) By requiring VPN connections

Answer: B) By integrating with IAM and Secrets Manager

Q6: What problem does RDS Proxy solve for serverless applications?

- A) Cold start delays
- B) Connection storms from rapid scaling
- C) Lack of persistent storage
- D) High latency to databases

Answer: B) Connection storms from rapid scaling

Q7: Which database engines work with RDS Proxy? (Select TWO)

- A) Amazon Aurora (MySQL/PostgreSQL)
- B) Amazon DynamoDB
- C) Amazon RDS for MySQL/PostgreSQL
- D) Amazon Redshift

Answers: A) Amazon Aurora and C) Amazon RDS for MySQL/PostgreSQL

Q8: True or False: RDS Proxy eliminates the need for database credentials in application code.

- A) True
- B) False

Answer: A) True