

Databases

RDS · MULTI-AZ & READ REPLICAS · DYNAMODB · REDSHIFT · AURORA · CHOOSING A DATABASE

STUDY & REVIEW

01 MUST KNOW

if you remember five things, remember these

- 01 Managed vs. unmanaged:** with an **unmanaged** service *you* handle scaling, fault tolerance, and availability; a **managed** service builds them in. Amazon RDS is the managed relational database — AWS runs the OS, database software, backups, high availability, and scaling; you keep only **application optimization**.
- 02 Multi-AZ ≠ read replicas:** Multi-AZ is **availability** — a **synchronous** standby in another AZ, promoted **automatically** on failure (same DNS endpoint, no app change). Read replicas are **read scaling** — **asynchronous** copies that offload read queries; promotion is **manual**, and a replica can live in another Region.
- 03 DynamoDB:** fast, flexible **NoSQL** (key-value + document) with consistent **single-digit-millisecond** latency at any scale. SSD-only, virtually unlimited table size and throughput; items in the same table can have **different attributes**.
- 04 Redshift:** fully managed **data warehouse** for analytics — standard **SQL** and your existing **BI tools**, **columnar storage** + **massively parallel processing** over petabytes; **Spectrum** queries exabytes directly in S3.
- 05 Aurora:** enterprise-class relational engine, **MySQL- and PostgreSQL-compatible**, fully managed by RDS — multiple copies of data across multiple AZs, continuous backup to S3, up to **15 read replicas**, crash recovery in under 60 seconds.

02 KEY TERMS

TERM	DEFINITION
Managed service	Scaling, fault tolerance, and availability are built in; unmanaged = you provide them
Amazon RDS	Managed service that sets up, operates, and scales relational databases in the cloud
DB instance	Isolated database environment holding multiple user-created databases; sized by instance class + storage type
Multi-AZ deployment	Synchronous standby copy in another AZ; automatic failover via the RDS DNS endpoint
Read replica	Asynchronous copy that serves read traffic; manual promotion; can be cross-Region
NoSQL (non-relational) DB	No relational model; scales out horizontally; handles unstructured and semistructured data
Amazon DynamoDB	Fast, flexible NoSQL database service — single-digit-ms latency at any scale
Partition key	Simple DynamoDB primary key made of one attribute
Composite primary key	Partition key + sort key — two attributes identify each item
Amazon Redshift	Fast, fully managed data warehouse — analytic SQL over petabytes of structured data
Redshift Spectrum	Runs queries against exabytes of data directly in Amazon S3
Amazon Aurora	MySQL/PostgreSQL-compatible relational DB built for the cloud; managed by RDS

03 THE DATABASE CHOOSER

the exam's favorite question: which database for which job

SERVICE	TYPE	BUILT FOR	REMEMBER
Amazon RDS	Relational (SQL)	Complex transactions and complex queries — web/mobile apps, ecommerce, games	Up to 30,000 IOPS; one node/shard; high durability
Amazon Aurora	Relational (SQL)	Enterprise-grade MySQL/PostgreSQL workloads that need top availability	RDS engine · 15 read replicas · <60 s crash recovery

SERVICE	TYPE	BUILT FOR	REMEMBER
DynamoDB	NoSQL key-value + document	Simple GET/PUT at massive scale — mobile, web, gaming, adtech, IoT	Single-digit ms; no size or throughput limits
Redshift	Data warehouse	Analytic queries over petabytes of structured data with SQL + BI tools	Columnar + parallel; Spectrum reads S3

04 UNMANAGED VS. MANAGED DATABASE

DB ON EC2 – UNMANAGED <i>AWS takes the building; you take the software</i> — AWS: server maintenance, rack & stack, power/HVAC/network — You: OS installation and patches — You: DB software installs and patches — You: backups, high availability, scaling — You: application optimization	DB ON RDS / AURORA – MANAGED <i>you tune the app; AWS runs the rest</i> — AWS: OS + DB software installs and patches — AWS: automatic backups, high availability — AWS: scaling, power, racking, server maintenance — You: application optimization — that's it
---	---

05 MULTI-AZ VS. READ REPLICAS

availability vs. read scaling – never confuse them

	MULTI-AZ DEPLOYMENT	READ REPLICAS
Purpose	High availability and durability	Scale out read-heavy workloads
Replication	Synchronous to a standby in another AZ (same VPC)	Asynchronous to replica instances
Failover	Automatic — standby becomes the new main; same DNS endpoint, no app change	Manual promotion to primary
Extras	Protects during maintenance, instance failure, AZ disruption	Cross-Region OK (DR, lower read latency); MySQL, MariaDB, PostgreSQL, Aurora

06 SQL VS. NOSQL

RELATIONAL (SQL) <i>RDS · Aurora · Redshift</i> — Structured data in tables, records, columns — Well-defined relationships between tables; SQL interface — Difficult to scale out horizontally — Semistructured data is awkward — many joins for normalized data	NON-RELATIONAL (NOSQL) <i>DynamoDB</i> — No relational model — items can differ in attributes — Scales out horizontally — Works with unstructured and semistructured data — Add attributes as the app evolves — no schema migration
---	---

07 DYNAMODB IN ONE BOX

core facts the knowledge check pulls from

FACT	DETAIL
Core components	Tables (collections of data) → items (uniquely identifiable groups of attributes) → attributes (fundamental data elements)
Primary keys	Partition key (simple — one attribute) or partition key + sort key (composite — two attributes)
Query vs. scan	Query locates items by primary key via partitioning; scan matches non-key attributes but reads every item — less efficient
Performance	Runs exclusively on SSDs; consistent single-digit-ms latency at any scale
Throughput	Provision read/write capacity manually, or let auto scaling watch the load and adjust it
Extras	Global tables (auto replication across chosen Regions), encryption at rest, item TTL

- ✗ “Multi-AZ speeds up my database and scales reads.” — Multi-AZ is **availability**: a synchronous standby for failover. Read scaling comes from **read replicas**.
- ✗ “Read replicas fail over automatically.” — Promotion of a read replica is a **manual** action (replication is asynchronous). Automatic failover belongs to **Multi-AZ**.
- ✗ “With RDS, AWS manages everything.” — You still own **application optimization**; AWS takes the OS, database software, backups, HA, and scaling.
- ✗ “150,000 writes per second? Use RDS.” — RDS tops out around **30,000 IOPS**. Massive rates, sharding, or simple GET/PUT ⇒ **DynamoDB**; deep RDBMS customization ⇒ run the engine on **EC2**.
- ✗ “Aurora is AWS's NoSQL database.” — Aurora is **relational** (MySQL/PostgreSQL-compatible). The fully managed NoSQL service is **DynamoDB**.
- ✗ “DynamoDB items must share one schema.” — Items in the same table can have **different attributes**; new formats sit beside old ones with no schema migration.
- ✗ “A DynamoDB scan is just another query.” — A scan reads **every item** and filters on non-key attributes; a query uses the primary key and partitioning to go straight to the data.
- ✗ “Redshift is a general-purpose app database.” — Redshift is a **data warehouse** for analytic queries; transactional application workloads belong on RDS, Aurora, or DynamoDB.

09 SELF-QUIZ

answers are upside-down below

Q1 Sample CCP exam: which of these is a fully managed NoSQL database service — Amazon RDS, Amazon DynamoDB, Amazon Aurora, or Amazon Redshift?

Q2 How does data reach the standby in a Multi-AZ deployment, and how does it reach a read replica?

Q3 Name the six database engines Amazon RDS supports.

Q4 An application needs 150,000 writes per second. Is RDS the right home — and if not, what is?

Q5 List DynamoDB's three core components.

Q6 What two kinds of primary keys can a DynamoDB table have?

Q7 Which Redshift feature runs queries against exabytes of data directly in Amazon S3?

Q8 In a Redshift cluster, which node communicates with client programs and builds the query plans?

Q9 Which two open-source database engines is Aurora drop-in compatible with?

Q10 How many read replicas can Aurora use, and where does it continuously back up data?

ANSWER KEY (rotate the page)

Q1 Amazon DynamoDB (keyword: NoSQL) **Q2** Multi-AZ: synchronously; read replica: asynchronously **Q3** Aurora, PostgreSQL, MySQL, MariaDB, Oracle, Microsoft SQL Server **Q4** no — a NoSQL service like DynamoDB (or a DB on EC2) **Q5** tables, items, attributes **Q6** partition key; partition key + sort key (composite) **Q7** Redshift Spectrum **Q8** the leader node (compute nodes run the compiled code) **Q9** MySQL and PostgreSQL **Q10** up to 15; Amazon S3