

Comparativa técnica entre servicios gestionados de bases de datos

Comparación entre Amazon RDS (relacional), Amazon DynamoDB (NoSQL) y Amazon Redshift (analítico). Se destacan sus características principales, ventajas, limitaciones y casos de uso recomendados, finalizando con una conclusión práctica sobre cuándo usar cada servicio.

Criterio	Amazon RDS (Relacional)	Amazon DynamoDB (NoSQL)	Amazon Redshift (Analítico)
Modelo de datos	Relacional (SQL, ACID)	NoSQL (clave-valor, documentos)	Data warehouse columnar (OLAP)
Escalabilidad	Vertical y horizontal (réplicas de lectura)	Automática e ilimitada	Escalado horizontal (nodos)
Velocidad	Alta, pero mayor latencia que DynamoDB	Milisegundos, ideal para apps con tráfico alto	Alta en consultas analíticas complejas
Costo	Pago por instancia + almacenamiento	Pago por capacidad provisionada o por demanda	Pago por almacenamiento y consultas
Flexibilidad	Compatible con MySQL, PostgreSQL, Oracle, etc.	Flexible en estructura, sin esquema rígido	Optimizado para SQL estándar y BI
Ventajas	Consistencia transaccional, soporte a sistemas críticos	Escalabilidad masiva, baja latencia, serverless-friendly	Análisis masivo, integración con BI y S3
Limitaciones	Costos altos en cargas intensivas	No apto para relaciones complejas ni SQL ACID	No diseñado para OLTP ni transacciones
Casos ideales	ERP, CRM, banca, apps web con SQL	E-commerce, IoT, microservicios, apps serverless	Big data, reporting, marketing analytics

Conclusión

- Amazon RDS: adecuado cuando se requiere consistencia, integridad transaccional y compatibilidad SQL (ejemplo: banca, salud, sistemas de gestión).
- Amazon DynamoDB: ideal para aplicaciones con alta escalabilidad y baja latencia, especialmente en tráfico variable como e-commerce, IoT y apps móviles.
- Amazon Redshift: recomendado para análisis de grandes volúmenes de datos, generación de informes y visualización con BI.