

need to know

03|2015



Databases

Database engines are a critical component of modern game technology stacks. The explosion of “NoSQL” (non-relational) data stores and expectations around performance and availability of worldwide, cloud hosted solutions makes the evaluation of these different options a time consuming and sometimes confusing process. This guide aims to outline some of the main categories of datastores, why you might want to use them and what considerations you need to be asking of your vendors.

Databases

Terminology

- + **Vertical scaling:** Increase the resources allocated to each database server.
- + **Horizontal scaling:** Increase the number of database servers.
- + **Redundancy:** Data is redundant if it is held in two or more locations.
- + **Durability:** A database is said to have strong durability if it guarantees that data is stored on disk. Weak durability implies that data is stored in RAM (so would be lost if the server restarts).
- + **Consistency:** A database is consistent if all servers see the same data at the same time. There is a trade off between consistency and availability (if data is spread across multiple servers, what should happen when communication between those servers fails?).

Questions to ask

- + What types of questions do you need to ask your database and how long can you wait for answers?
- + How “big” will your data get and how many concurrent connections?
- + Will you need to scale horizontally?
- + Does all your data fit in RAM? Is data durability a must-have?
- + What are your consistency and transactional requirements?
- + What are your availability, replication, and geo-location requirements?
- + Do I need a flexible schema for rapid development? Will your data model change over time?
- + Do I want to host it myself, use cloud infrastructure, or a fully managed service?

Types of database

Type	Data format	Use cases	Example Products
Relational (RDBMS)	Tables, rows and relationships between them defined in a schema	General purpose workloads in a single location where query flexibility, durability and strong consistency are required	PostgreSQL, MySQL, IBM DB2, Oracle, Microsoft SQL Server
Key/value store	Unstructured records identified by a unique key	Constant read/write workloads e.g. caches, session storage	Redis, Riak, Amazon DynamoDB, IBM Websphere eXtreme Scale
Document database	JSON or XML documents with a flexible schema	Mixed workloads for web and mobile applications. Different products trade durability, availability and consistency	MongoDB, Apache CouchDB, IBM Cloudant, Azure DocumentDB, MarkLogic
Big Table	Semi-structured records stored in rows	Write-heavy workloads over semi-structured, time-series data	Apache Cassandra, Apache HBase, IBM DB2 BLU

Other considerations

- + **Licensing:** does the software incur a licensing cost per server (may influence a design around vertical vs horizontal scaling)?
- + **DIY vs hosted vs managed:** DIY implies running the database yourself (bring your own infrastructure). Hosted services are run in the cloud, typically on shared infrastructure and with a self-service philosophy. Managed services take hosting one step further and offer private infrastructure options, pro-active support and account management.

What should I do next?

If you'd like further information, visit cloudant.com/history/contact-us to submit an inquiry and a member of the team will get back to you.

@cloudant 
Cloudant on facebook 
Cloudant on LinkedIn 
cloudant.com 

@uk.ie 
Ukie on facebook 
Ukie on LinkedIn 
ukie.org.uk 