
SQL
To ~~be~~ or not to ~~be~~
SQL
That is the question
BeSA
Become a Solutions Architect
Week 6 (11-June)
Learn about
DATABASES ON AWS

Agenda

1. Databases on AWS

2. SQL Vs NoSQL data modelling by example

   - James

3. Tips to get your resume shortlisted at Amazon

   - Prasad

BeSA
Become a Solutions Architect

Databases on AWS

Amazon RDS

Amazon DynamoDB







How storage evolved?

- In early days of computing Storage was very costly

Image Source:
https://www.reddit.com/r/computerscience/comments/ak27u0/ibm_5mb_hard_drive_1956/

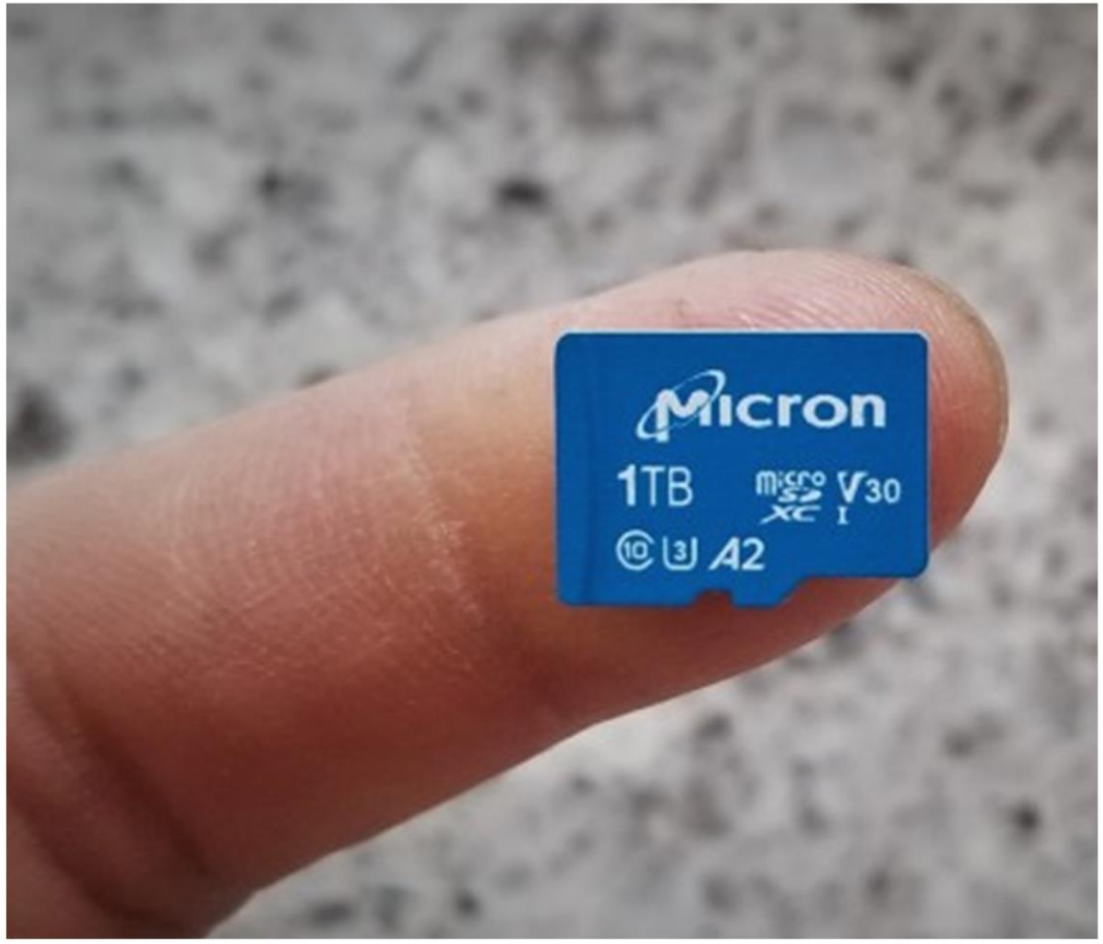
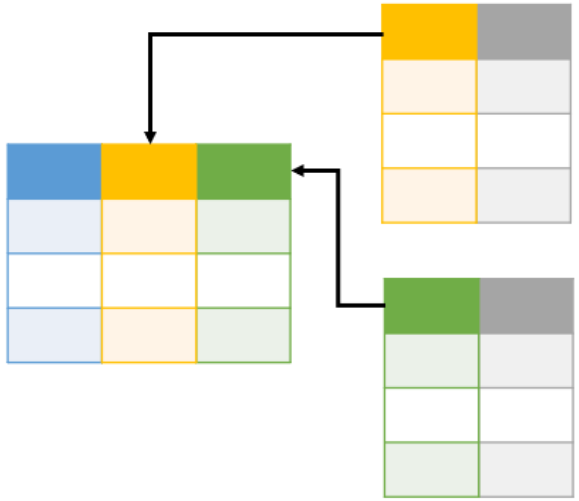


Image Source:
https://www.thessdreview.com/featured/micron-c200-microsd-card-review-1tb-as-high-capacity-becomes-the-norm-in-microsd/

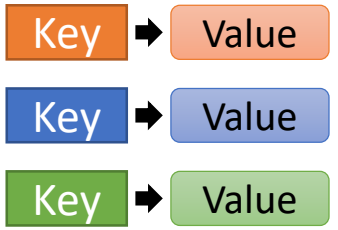
SQL vs. NoSQL Databases

	SQL (Optimized for Storage)	NoSQL (Optimized for performance)
Data Storage	Rows and Columns	Key-value, Document, Wide-column, Graph

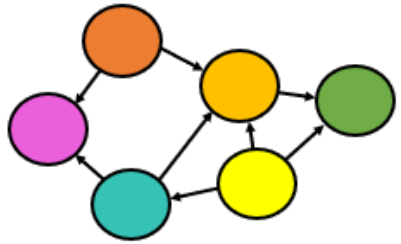
Rows and Columns



Key Value



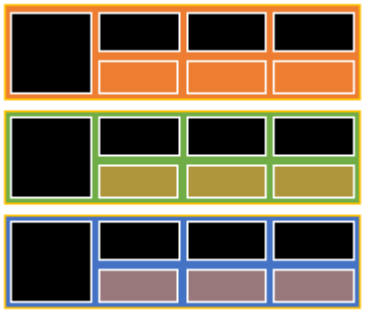
Graph



Document

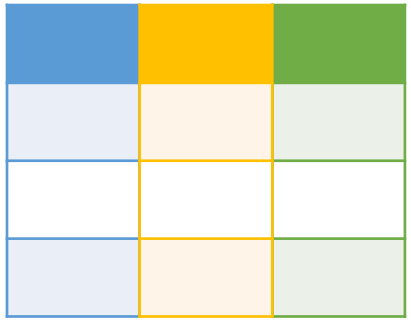
```
{
  “Id”: “1”,
  “FullName”:
   {
     “first”: “Jane”,
     “last”: “Doe”
   }
  “Year”: “2022”,
}
```

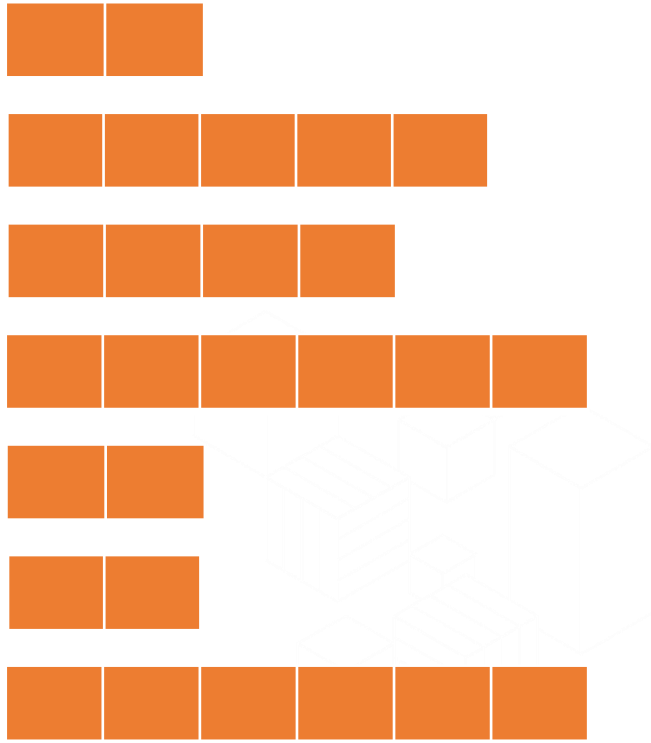
Wide Column



SQL vs. NoSQL Databases

	SQL (Optimized for Storage)	NoSQL (Optimized for performance)
Data Storage	Rows and Columns	Key-value, Document, Wide-column, Graph
Schema	Fixed	Dynamic





SQL vs. NoSQL Databases

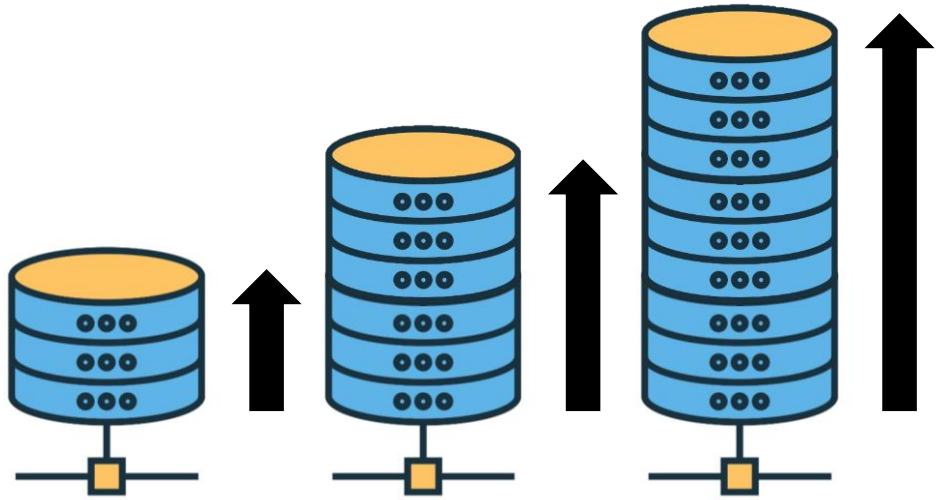
	SQL (Optimized for Storage)	NoSQL (Optimized for performance)
Data Storage	Rows and Columns	Key-value, Document, Wide-column, Graph
Schema	Fixed	Dynamic
Querying	Using SQL	Focused on collection of documents

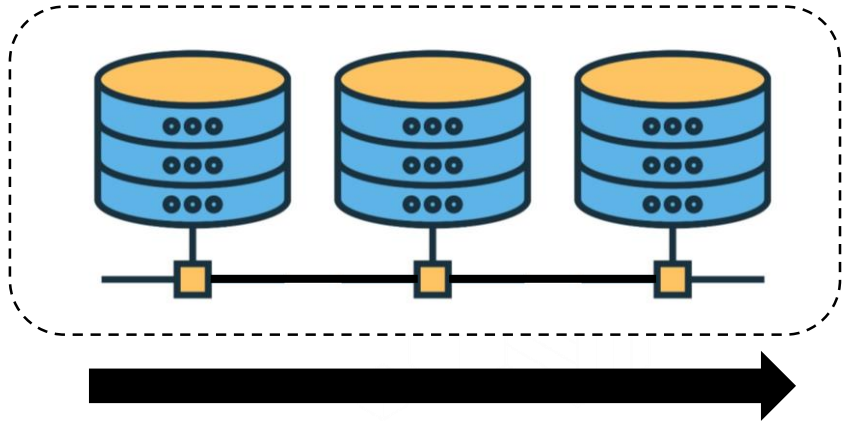
```
/*  Return all of the songs by an artist Elvis  */
SELECT * FROM Music
WHERE Artist= 'Elvis';
```

```
/*  Return all of the songs by an artist Elvis  */
{
   TableName: "Music",
   KeyConditionExpression: "Artist =
:a",
   ExpressionAttributeValues:
   {
    ":a": "Elvis"
    }
}
```

SQL vs. NoSQL Databases

	SQL (Optimized for Storage)	NoSQL (Optimized for performance)
Data Storage	Rows and Columns	Key-value, Document, Wide-column, Graph
Schema	Fixed	Dynamic
Querying	Using SQL	Focused on collection of documents
Scaling	Vertical	Horizontal





SQL vs. NoSQL Databases

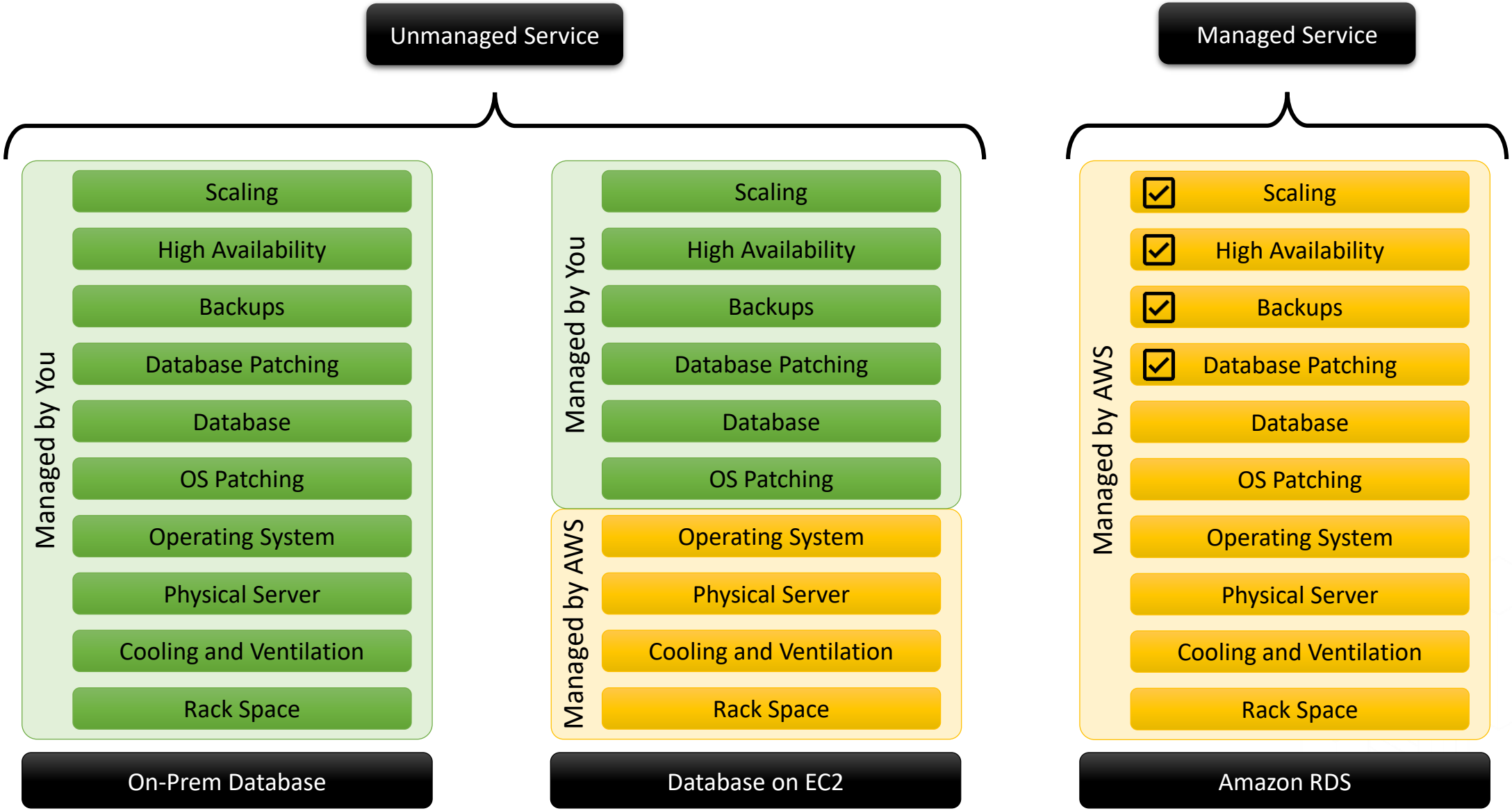
	SQL (Optimized for Storage)	NoSQL (Optimized for performance)
Data Storage	Rows and Columns	Key-value, Document, Wide-column, Graph
Schema	Fixed	Dynamic
Querying	Using SQL	Focused on collection of documents
Scaling	Vertical	Horizontal
Transactions	Supported	Support varies

A	C	I	D
Atomicity Transitions are all or nothing	**Consistency** Only valid data is saved	**Isolation** Transactions do not affect each other	**Durability** Written data won't be lost

BA	S	E
Basically Available System does guarantee availability	**Soft state** System may change over time	**Eventual consistency** system will become consistent over time

Amazon RDS

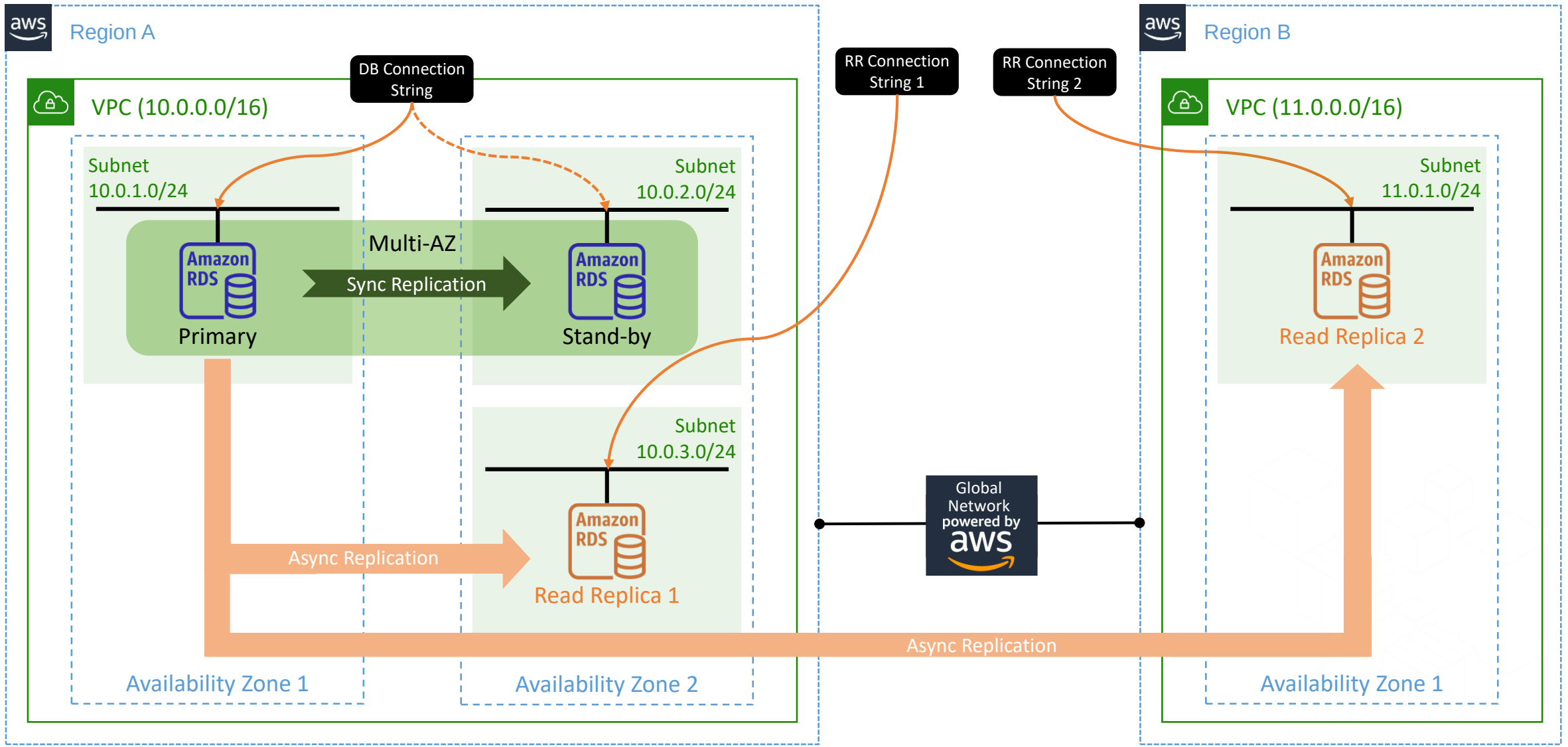
Running and Maintaining Databases



Amazon RDS

- Choice of different database engine
  - MySQL / PostgreSQL / Maria DB
  - MS SQL Server / Oracle
  - Amazon Aurora
- Supports High Availability and Read-Replica
- Snapshots can be copied across region
- Can migrate databases using Database Migration Service (DMS)
- Pricing
  - On-demand or Reserved Instance

RDS – Multi-AZ and Read Replica



Amazon Aurora

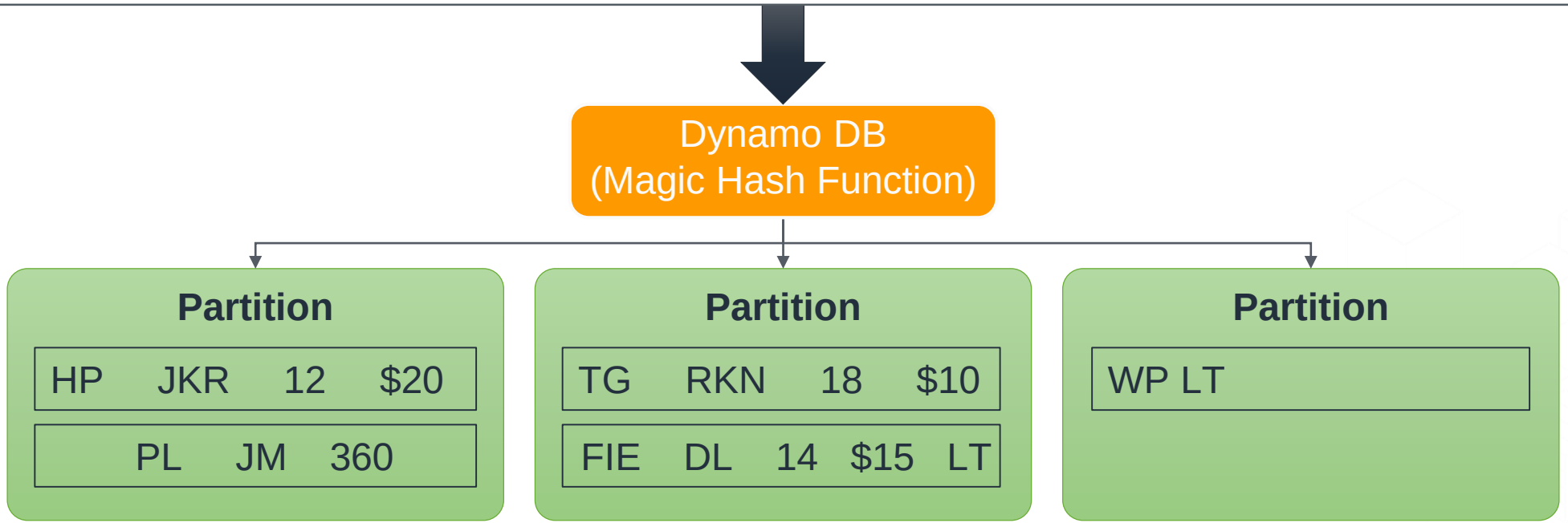
- Enterprise Databases – Oracle, MS SQL Server
  - Features/Performance ++++
  - Cost $$$$
- Open Source Databases – MySQL, MariaDB, PostgreSQL
  - Features/Performance ++
  - Cost $$
- Amazon Aurora
  - Features/Performance ++++
  - Cost $$$

Amazon Aurora

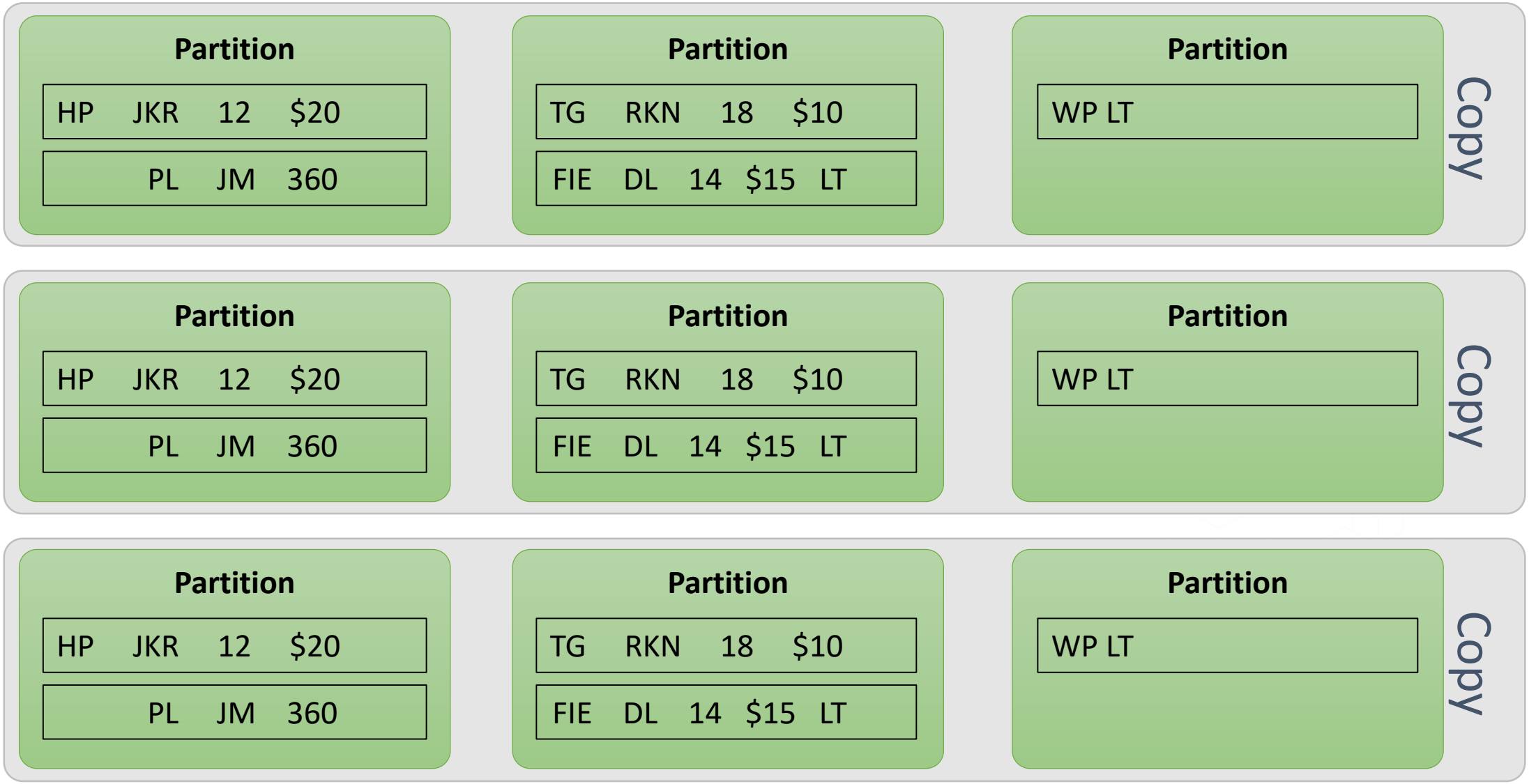
Amazon Dynamo DB

Dynamo DB

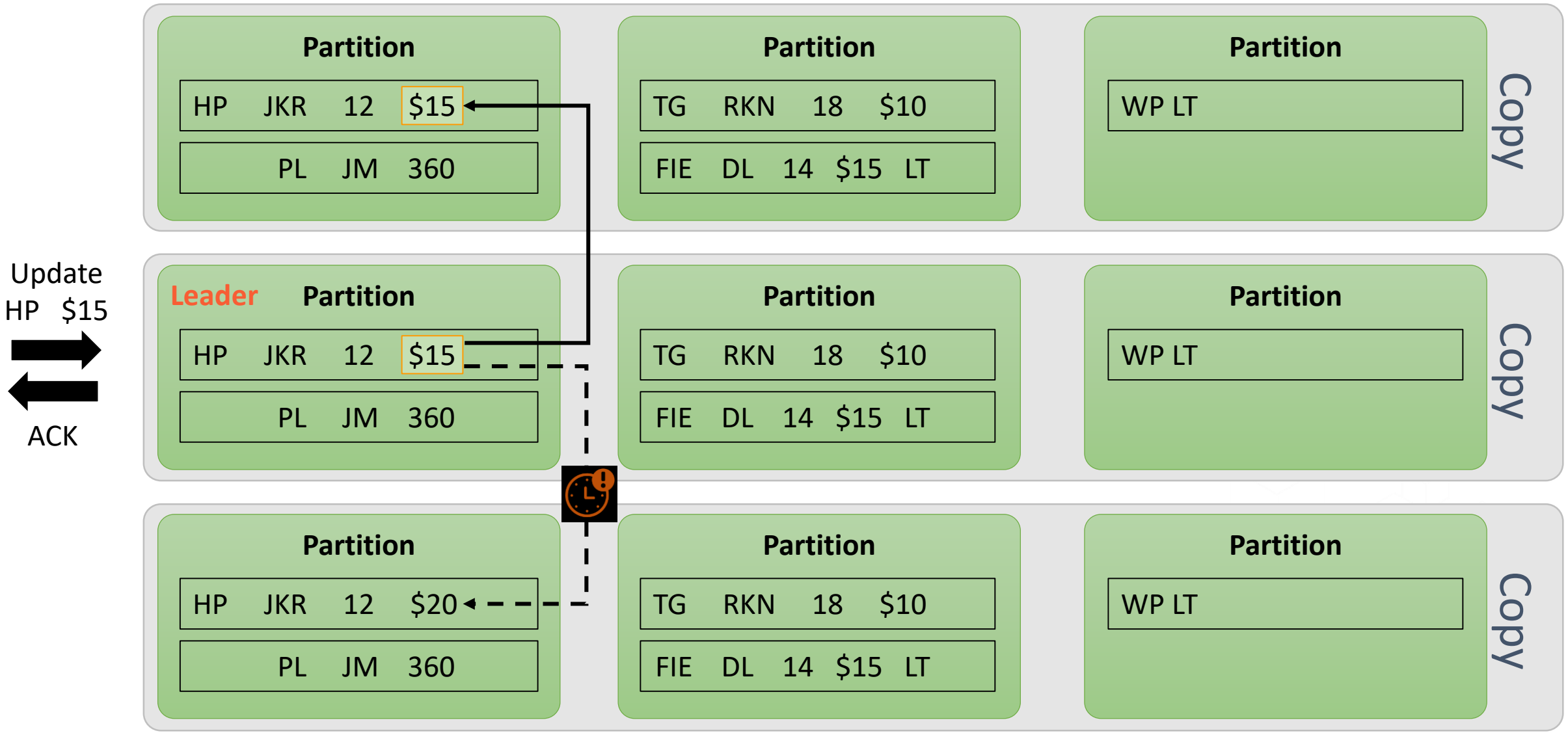
Harry Potter	J K Rowling	12	$ 20	
The Guide	R K Narayan	18	$ 10	
War and Peace	Leo Tolstoy			
Freedom in Exile	14th Dalai Lama	14	$ 15	Lhamo Thondup
Paradise Lost	John Milton	360 Pages		



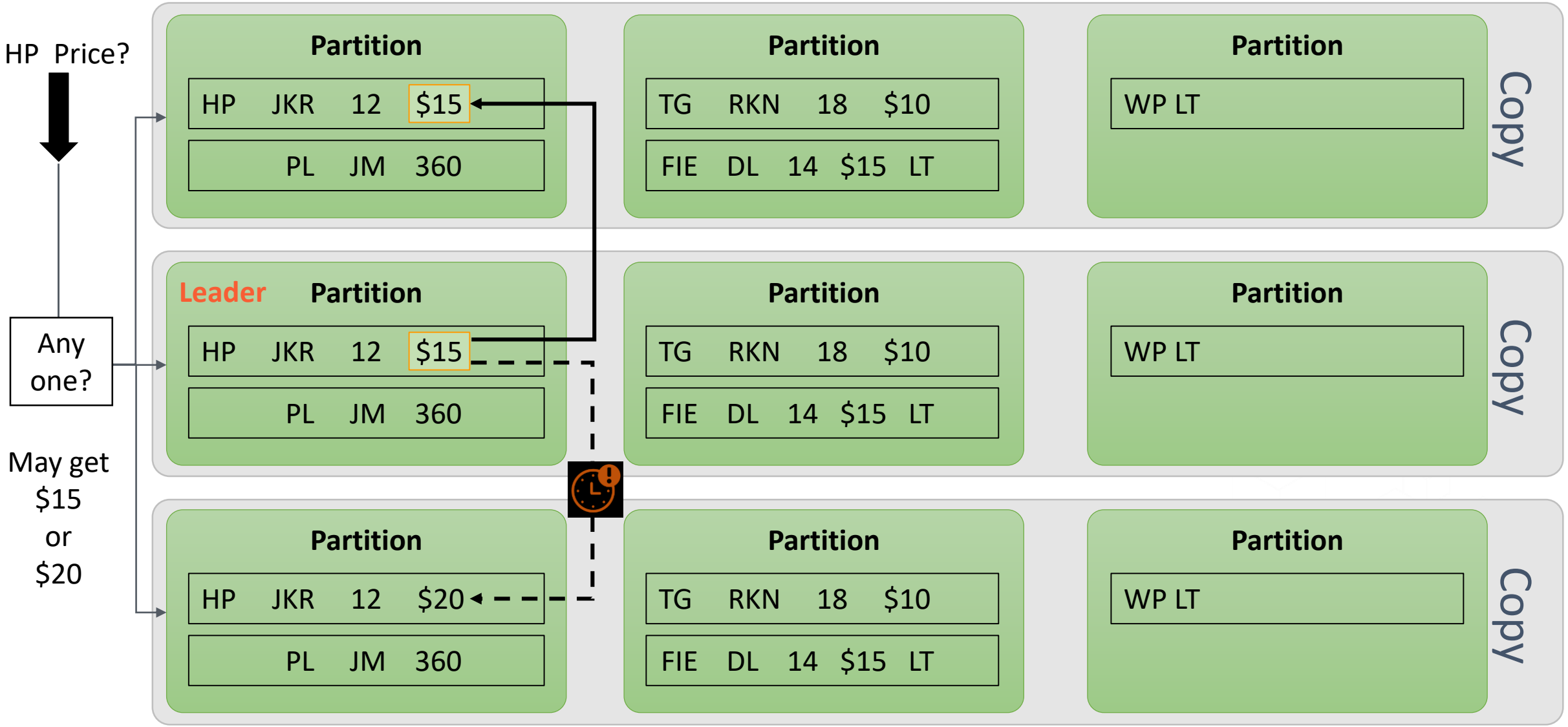
Dynamo DB Partitions



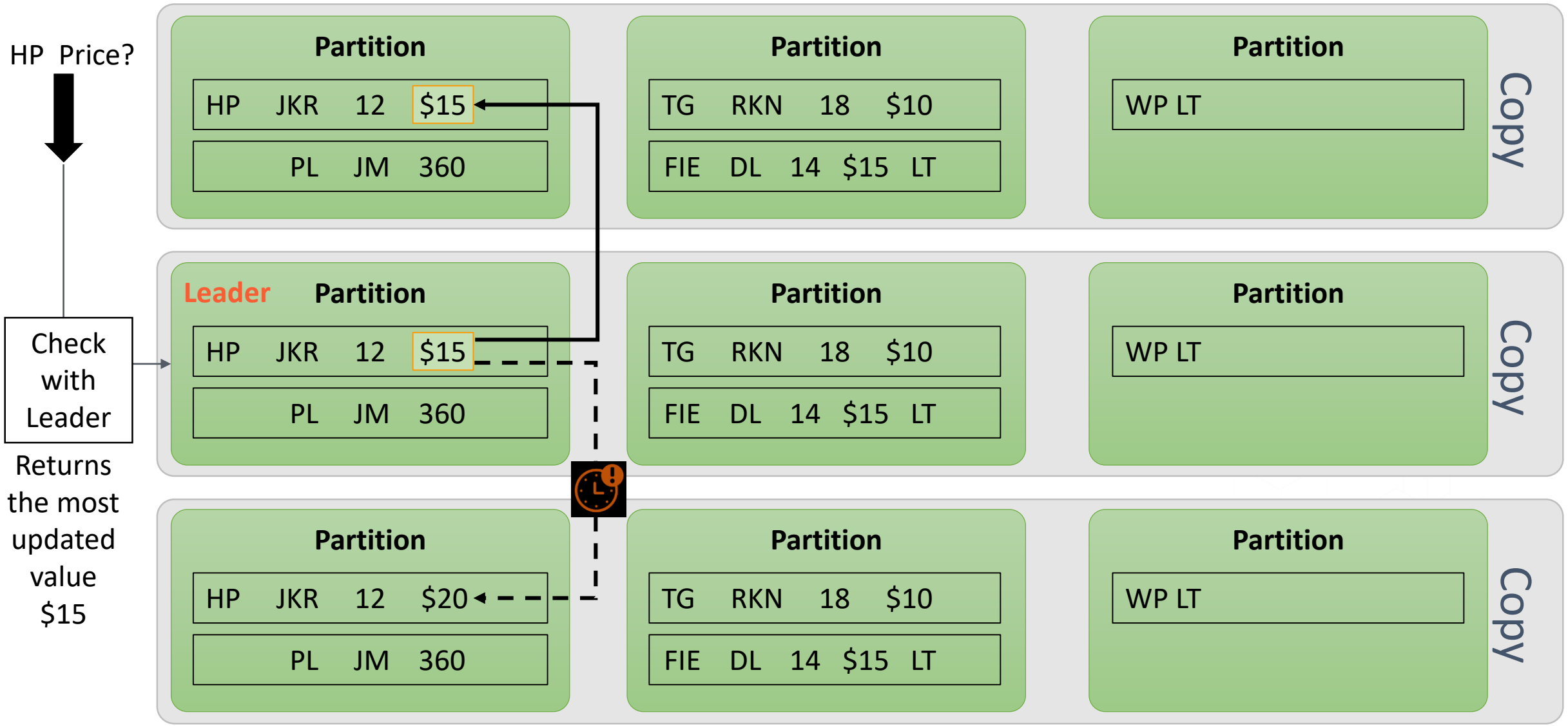
Write/Update Operation



Read - Eventual Consistency



Read - Strong Consistency



Partition Key and Sort Key

Partition Key – Title

Primary Key = Partition Key

Harry Potter	J K Rowling	12	$ 20	
The Guide	R K Narayan	18	$ 10	
War and Peace	Leo Tolstoy			
Freedom in Exile	14th Dalai Lama	14	$ 15	Lhamo Thondup
Paradise Lost	John Milton	360 Pages		
Harry Potter	J K Rowling	15	$ 25	

2nd Edition

Title – Not Unique, so can't be used as Partition Key

Harry Potter	1st Edition	J K Rowling	12	$ 20
Harry Potter	2nd Edition	J K Rowling	15	$ 25

Partition Key + Sort Key

Primary Key = (Partition Key + Sort Key)

Reference:

FAQs

Category:

Database

Amazon RDS

What?

- Amazon Relational Database Service (Amazon RDS) is a managed service that makes it easy to set up, operate, and scale a relational database in the cloud.

Why?

- You want to focus on your applications and business instead of managing time-consuming database administration tasks.
- Once your database is up and running, Amazon RDS automates common administrative tasks, such as performing backups and patching the software that powers your database.

When?

- You need the capabilities of a familiar MySQL, MariaDB, Oracle, SQL Server, PostgreSQL or Amazon Aurora database.
- You want the flexibility of being able to easily scale the compute resources or storage capacity associated with your relational database instance.

Where?

- Amazon RDS can be deployed in a Single AZ or Multi-AZ. When you provision a Multi-AZ database instance, Amazon RDS synchronously replicates your data to a standby instance in a different Availability Zone (AZ).
- Read Replica – In the same or different AWS Region than the Amazon RDS Instance.

Who?

- Amazon RDS manages the work involved in setting up a relational database from provisioning the infrastructure capacity you request to installing the database software.
- You are responsible for managing the database settings that are specific to your application.

How?

- The basic building block of Amazon RDS is the DB instance. Your DB instance can contain one or more user-created databases.
- You can access your DB instance by using the same tools and applications that you use with a standalone database instance.

Created by:

Ashish Prajapati

How much?

- You pay only for what you use, and there are no minimum or setup fees. You are billed based on: DB instance hours, Storage (per GB per month), Provisioned IOPS per month, Backup Storage and Data transfer.
- Unless you purchase reserved instances in a Region, all DB instances will be billed at on-demand hourly rates.

Reference:

FAQs

Category:

Database

Amazon Aurora

What?

- Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases.
- It features a distributed, fault-tolerant, and self-healing storage system that is decoupled from compute resources.

Why?

- Aurora automates time-consuming administration tasks like hardware provisioning, database setup, patching, and backups while providing the security, availability, and reliability of commercial databases at 1/10th the cost.

When?

- Amazon Aurora is a great option for any enterprise application that can use a relational database.
- You need high performance and availability with up to 15 low-latency read replicas, point-in-time recovery, continuous backup to Amazon S3, and replication across three AZs.

Where?

- Amazon Aurora is a regional service, it automatically maintains six copies of your data across three AZs.
- Cross-region Aurora replicas can be setup using either physical or logical replication. Physical replication uses Amazon Aurora Global Database, logical replication uses binlog for MySQL and PostgreSQL replication slots for PostgreSQL

Who?

- Amazon Aurora is fully managed by RDS and it automatically and continuously monitors and backs up your database to Amazon S3, enabling granular point-in-time recovery.
- Customer can scale the compute resources allocated to your DB Instance by changing your DB Instance class.

How?

- You choose Aurora as the DB engine option when setting up new database servers through Amazon RDS.
- After launching an Aurora instance, you can connect to it using any database client that supports MySQL or PostgreSQL.

Created by:

Ashish Prajapati

How much?

- For provisioned Aurora, you can choose On-Demand Instances and pay for your database by the hour with no long-term commitments or upfront fees, or choose Reserved Instances for additional savings.
- Aurora storage is billed in per GB-month increments, while I/Os consumed are billed in per million request increments.

Reference:

FAQs

Category:

Database

Amazon DynamoDB

What?

- Amazon DynamoDB is a fully managed, serverless, NoSQL database designed to support key-value and document data models.
- DynamoDB has a flexible schema, to easily adapt the tables as your business requirements change, without having to redefine the table schema as you would in relational databases.

Why?

- DynamoDB offers built-in security, continuous backups, automated multi-Region replication, in-memory caching, and data export tools.
- You can scale up or scale down your tables' throughput capacity without downtime or performance degradation

When?

- Build internet-scale applications supporting user-content metadata and caches that require high concurrency and connections for millions of users, and millions of requests per second.
- You want to support high-traffic, extreme-scaled events, encryption at rest with no operational overhead.

Where?

- DynamoDB is a regional service. All of your data is stored on SSDs and is automatically replicated across multiple Availability Zones in an AWS Region
- You can use global tables to keep DynamoDB tables in sync across AWS Regions.

Who?

- With DynamoDB, there are no servers to provision, patch, or manage, and no software to install, maintain, or operate. It automatically scales tables to adjust for capacity and maintains performance with zero administration.
- Availability and fault tolerance are built in and it also provides on-demand backup capability.

How?

- In DynamoDB, tables, items, and attributes are the core components. A table is a collection of items, and each item is a collection of attributes. An attribute is a fundamental data element, which does not need to be broken down further.
- It uses primary keys to uniquely identify each item in a table and secondary indexes to provide more querying flexibility.

Created by:

Ashish Prajapati

How much?

- DynamoDB charges for reading, writing, and storing data in your DynamoDB tables, along with any optional features you choose to enable. DynamoDB has two capacity modes, which come with specific billing options for processing reads and writes on your tables: on-demand and provisioned.

SQL Vs NoSQL data modelling by example

- James

Tips to get your resume shortlisted at Amazon

- Prasad

Thank you for attending. See you next Saturday (18-Jun-2022)

BeSA

Become a Solutions Architect

For content check **Resources Link** on BeSA Home Page