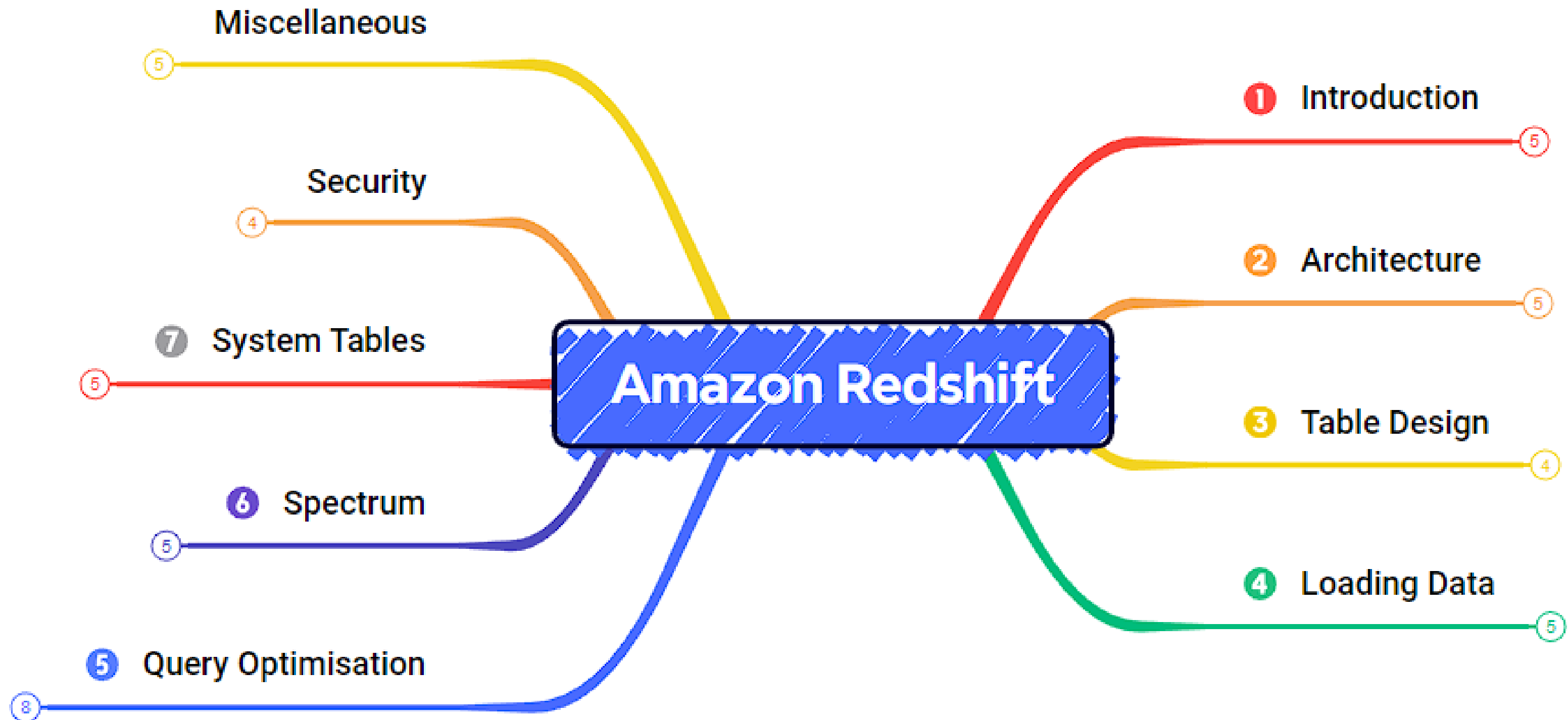






Why Amazon Redshift?

Tens of thousands of customers use Amazon Redshift for modern data analytics at scale, delivering up to 3x better price performance and 7x better throughput than other cloud data warehouses. Amazon Redshift seamlessly integrates with Amazon SageMaker Lakehouse, allowing you to use its powerful SQL analytic capabilities across your data lakes and Amazon S3 data warehouses and Amazon S3) data lakes. Enable your team's decision-making with data insights from the applications which connect data

databases, and third-party enterprise applications without building complex data pipelines. Amazon Redshift Serverless makes scaling your analytics effortless, allowing you to analyze petabytes of data without the burden of infrastructure management. Boost your team's productivity with Amazon Q in Amazon Redshift, which simplifies SQL authoring through natural language. Maximize the value of your data by using Amazon Redshift as a structured knowledge base for generative AI

insights in Amazon Bedrock, leading to more relevant and accurate outputs for your applications.

Amazon Redshift Documentation

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence tools. It is optimized for datasets ranging from a few hundred gigabytes to a petabyte or more and costs less than \$1,000 per terabyte per year, a tenth the cost of most traditional data warehousing solutions.



Amazon Redshift is a fully **managed**, petabyte-scale **data warehouse** service in AWS.

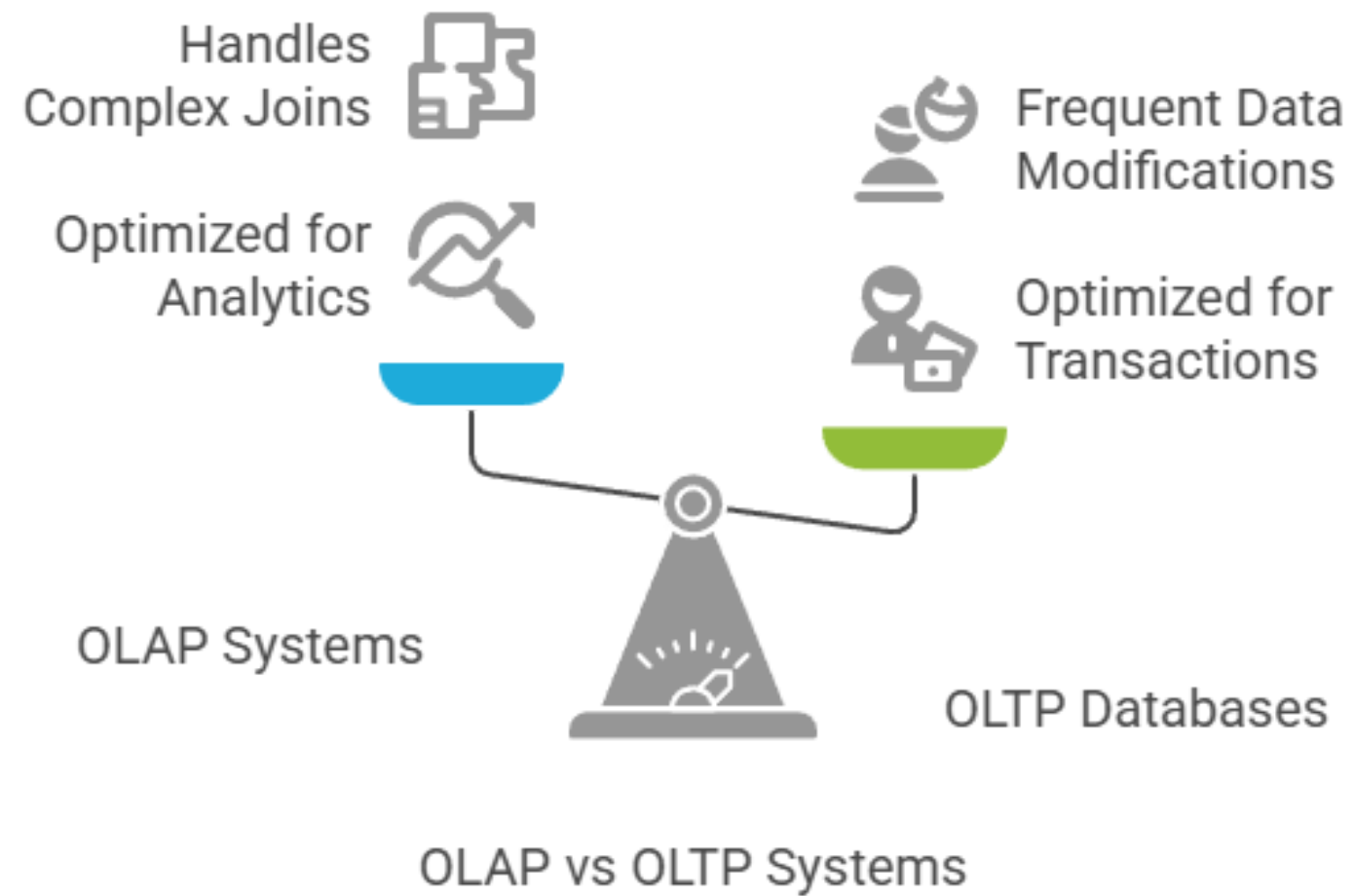
It allows organizations to store structured data and perform fast queries using **SQL**.

But how is Redshift different from a traditional databases?

Let's compare OLAP vs. OLTP first
& then with traditional datawarehouses



OLAP vs OLTP



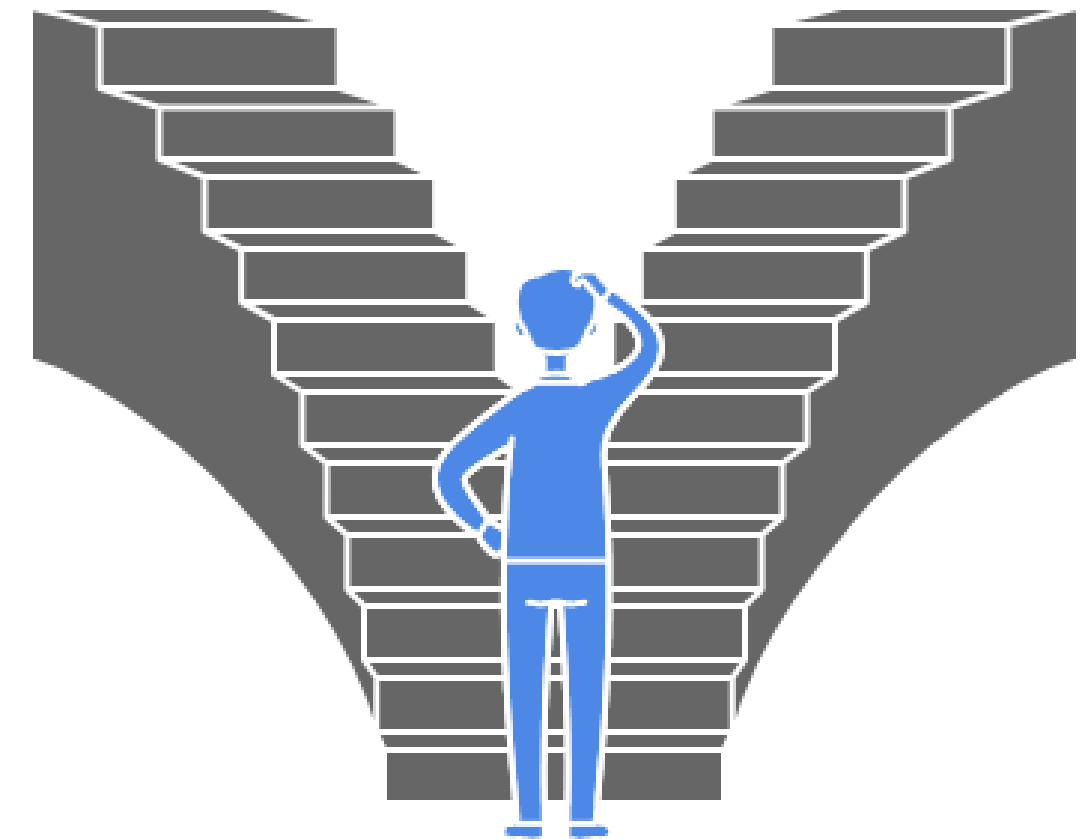
Which database system to use for a project?

Use Redshift

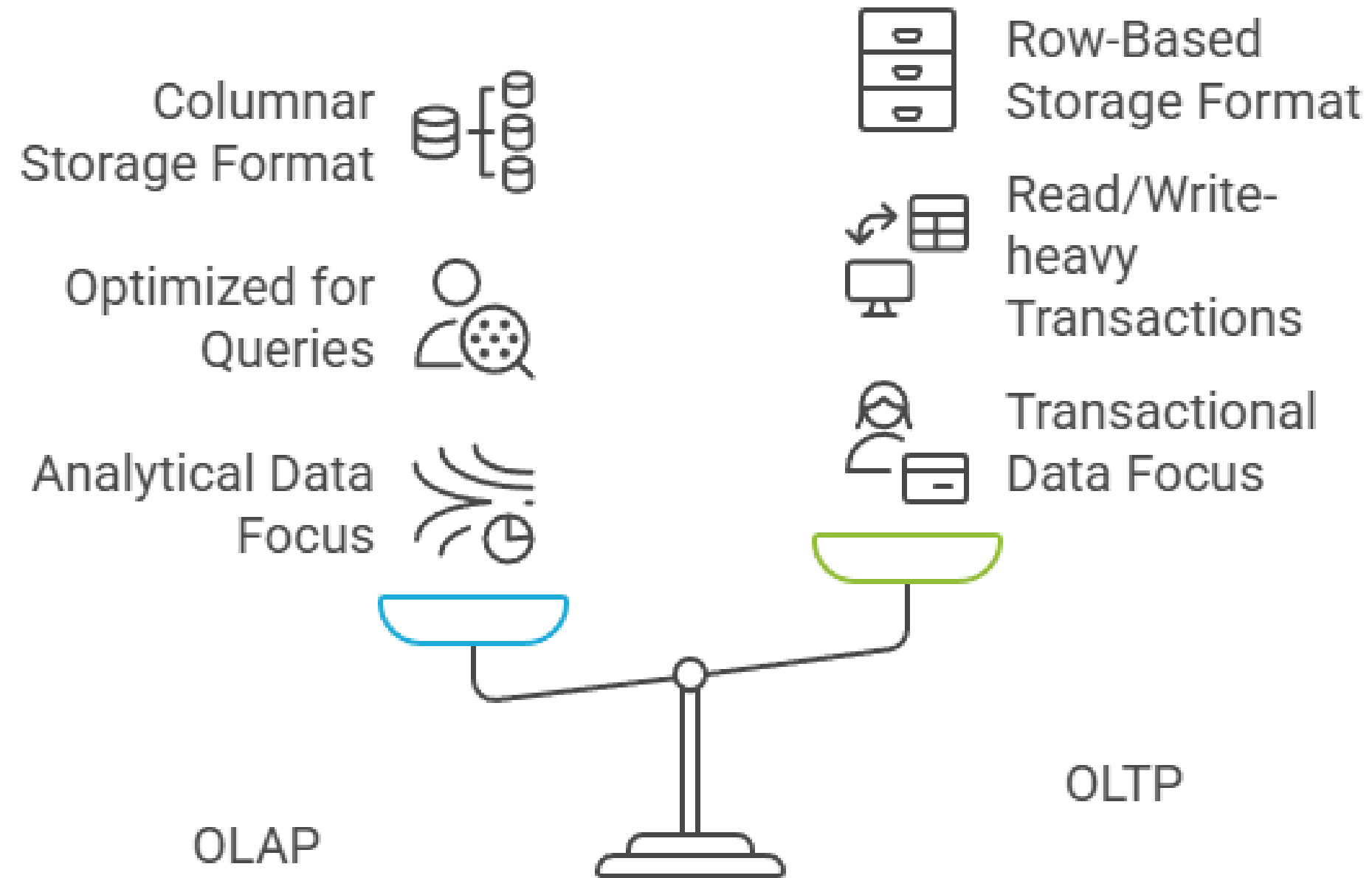
Ideal for large-scale analytics and reporting with complex aggregations.

Use OLTP System

Suitable for projects with frequent transactional updates.



OLAP vs OLTP



Choose the right database for your data needs.

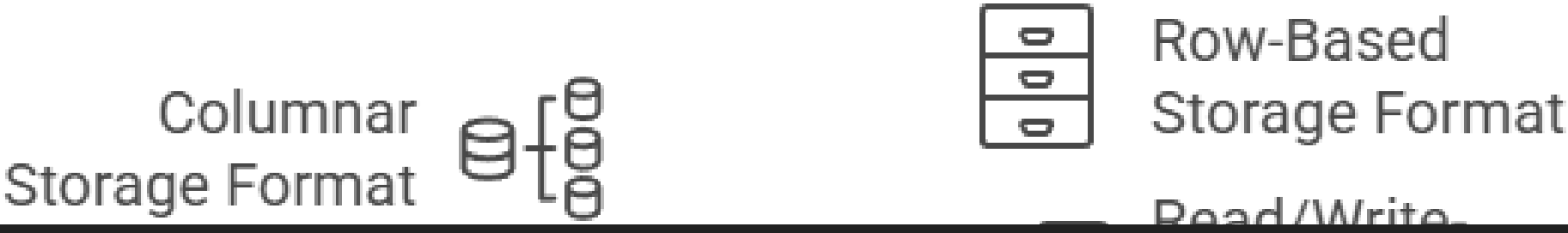


 Table Comparison:

Feature	OLAP (Redshift)	OLTP (MySQL, PostgreSQL)
Data Type	Analytical Data	Transactional Data
Query Pattern	Aggregations, Joins, Reports	Read/Write-heavy Transactions
Storage Format	Columnar Storage	Row-Based Storage
Performance	Optimized for Queries	Optimized for Transactions

Choose the right database for your data needs.

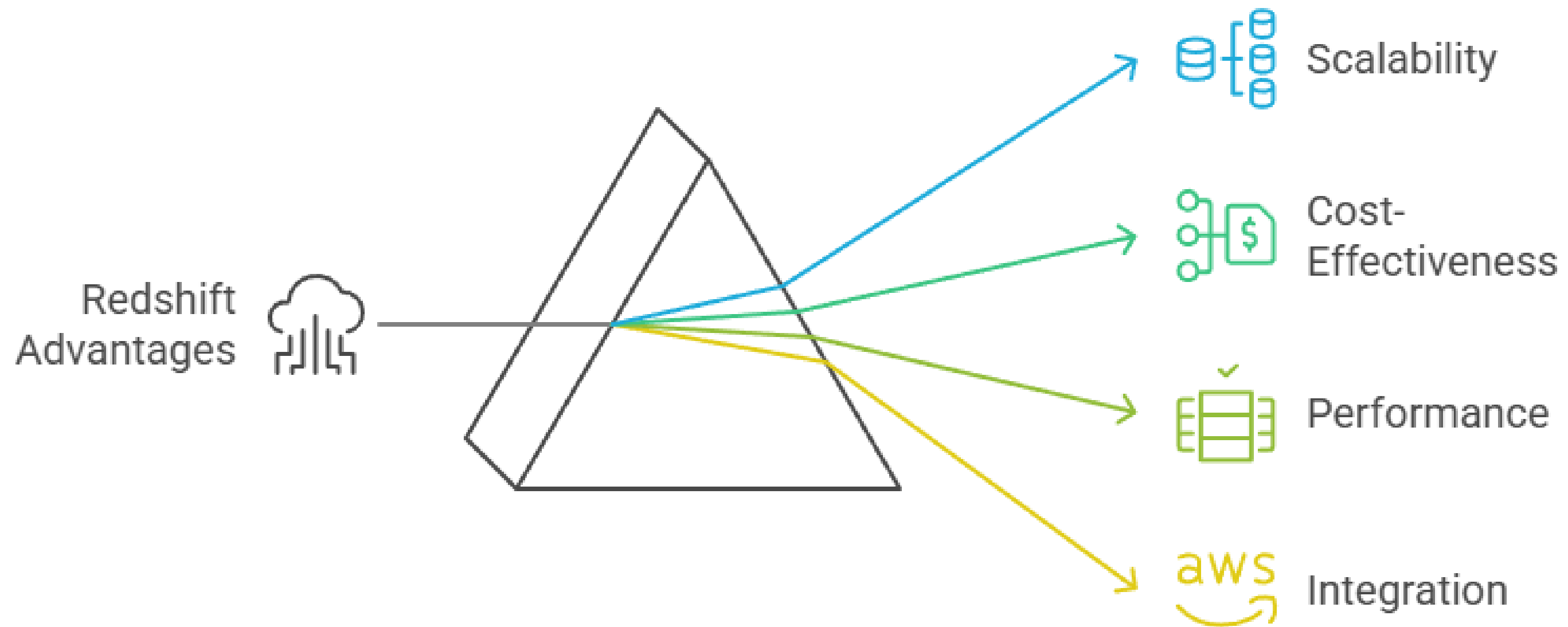
Before Redshift, companies used **expensive on-premise data warehouses** like Teradata, Oracle Exadata, and Netezza.

These solutions required heavy infrastructure investments and maintenance.

Redshift changes the game by offering a fully managed cloud-based data warehouse, eliminating hardware costs and providing seamless scalability.



Redshift's Advantages Over Traditional Data Warehouses



Redshift's Advantages Over Traditional Data Warehouses

- ✓ Scalability - Scale clusters up or down as needed
- ✓ Cost-Effective - Pay-as-you-go model with reserved instances
- ✓ Performance - Uses MPP (Massively Parallel Processing) for fast execution
- ✓ Integration - Connects with AWS services like S3, Glue, and Quicksight



Which Redshift pricing model should I choose?

On-Demand

Flexible, pay-per-hour pricing suitable for short-term use.

Reserved Instances

Long-term commitment with significant cost savings for consistent workloads.

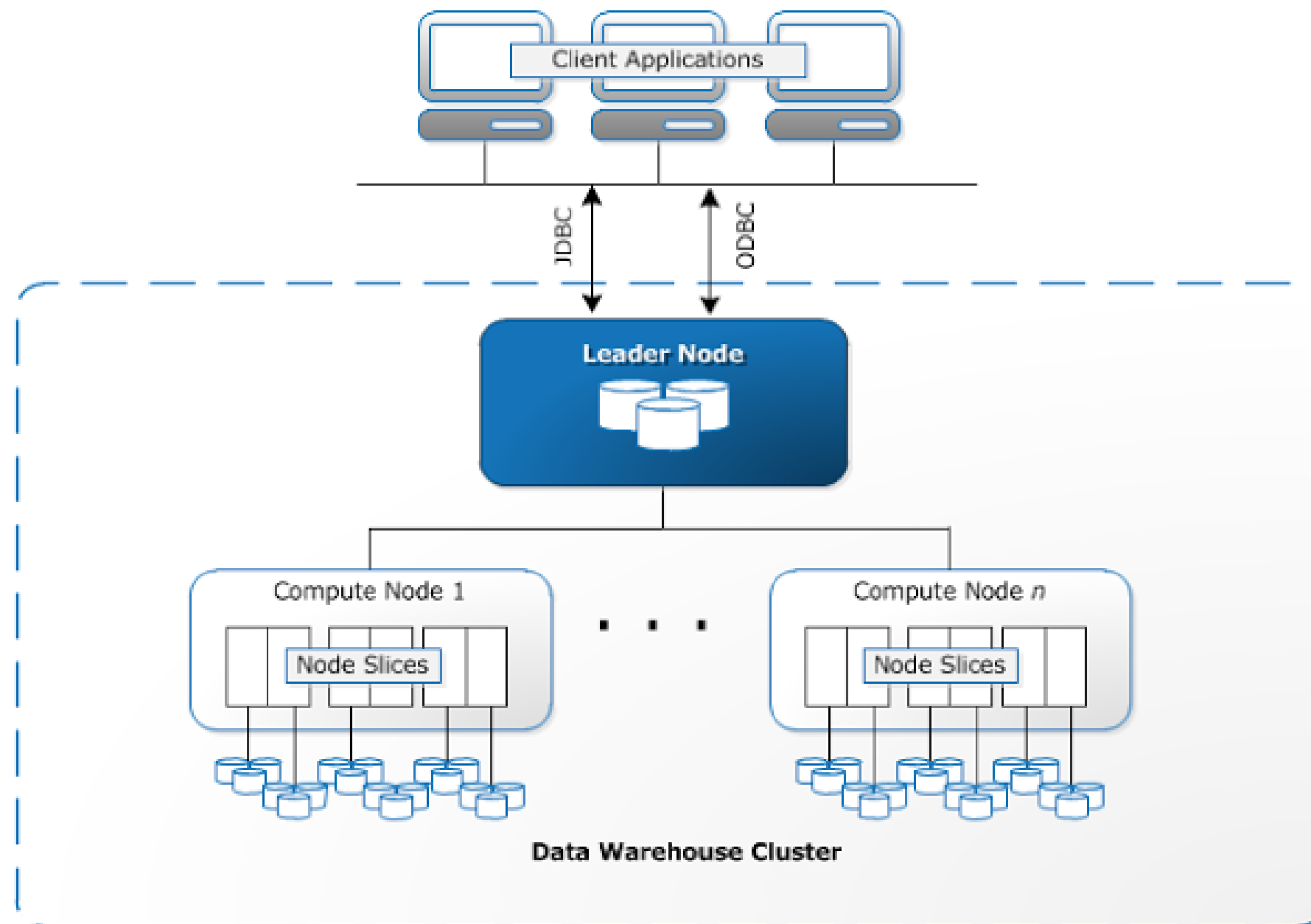
Redshift Serverless

Automatically scales with usage, ideal for variable workloads without cluster management.



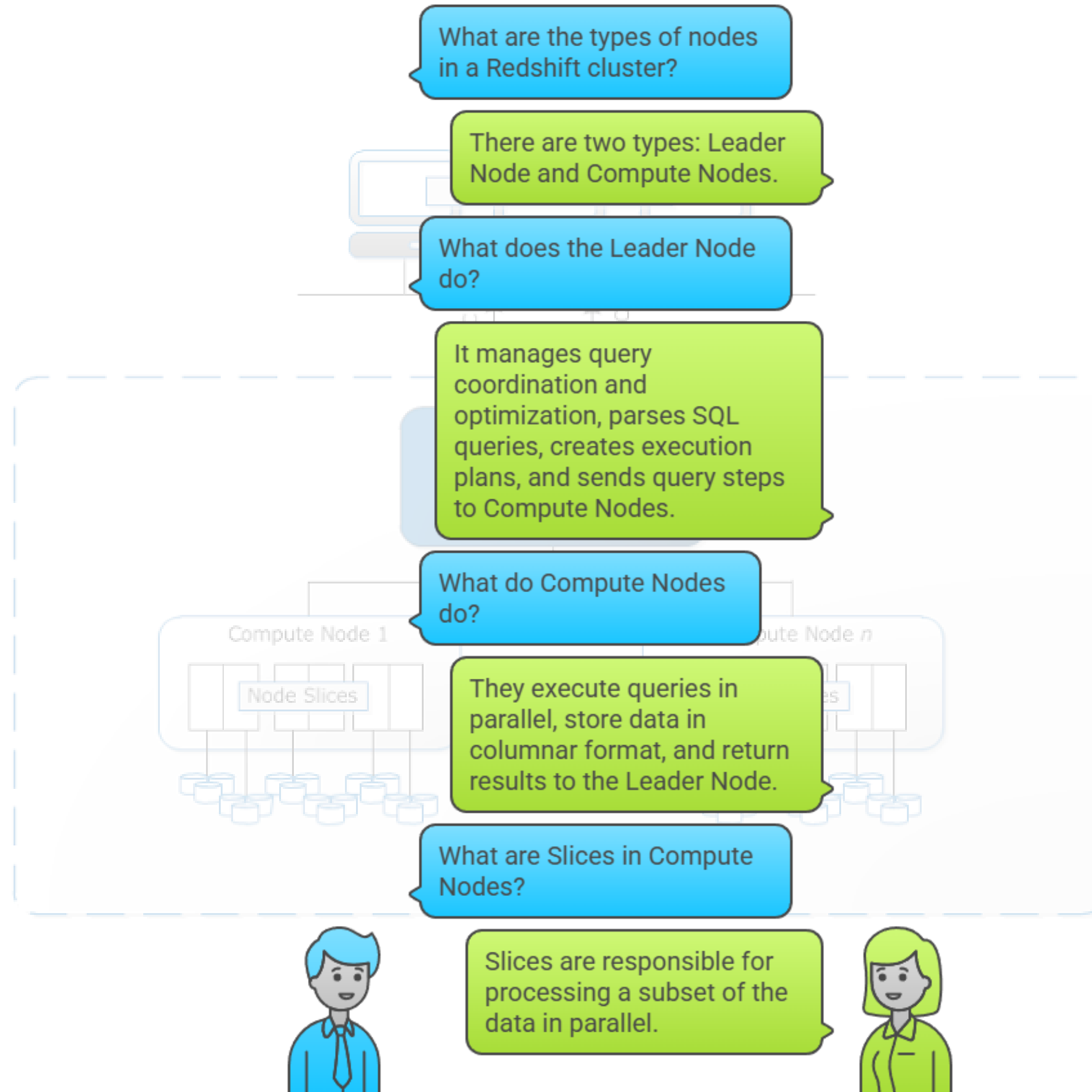
Hands-on exercise

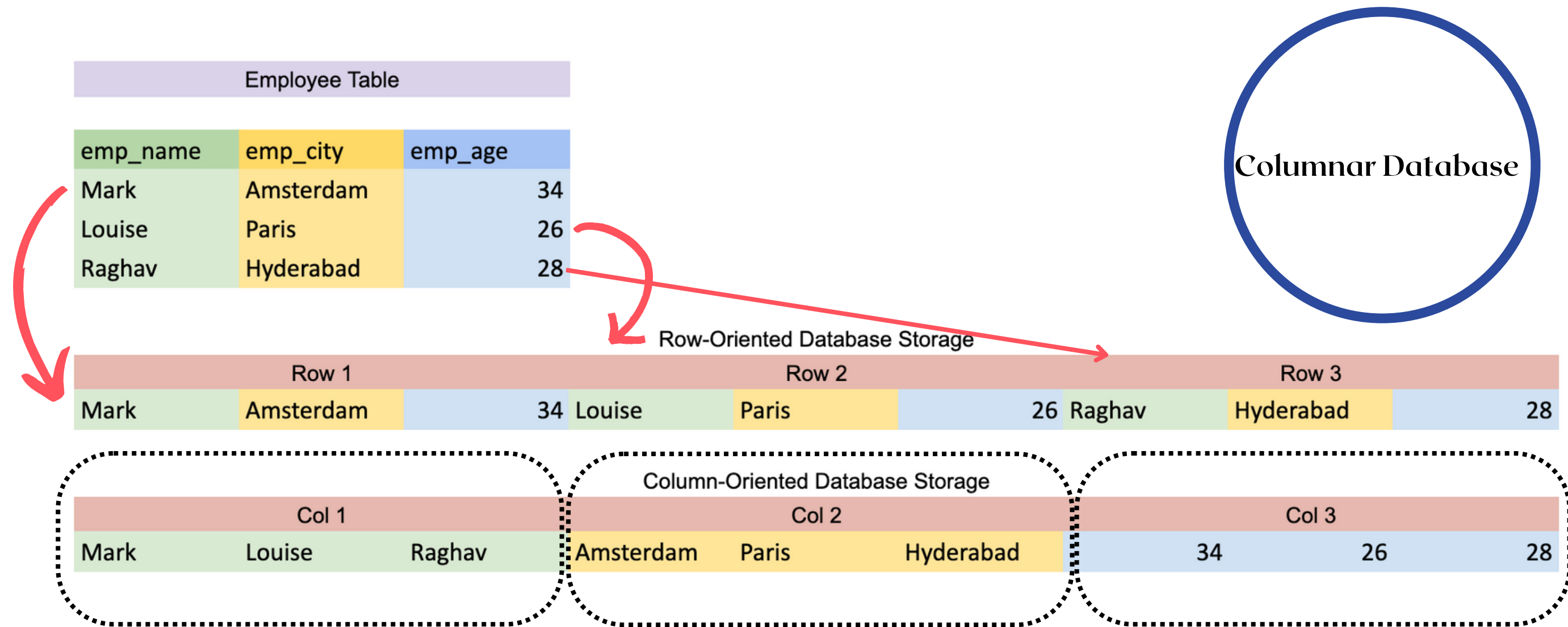
- Create Amazon Redshift cluster
- Run your first SQL query



Source: AWS Docs

Types of Nodes in a Redshift Cluster

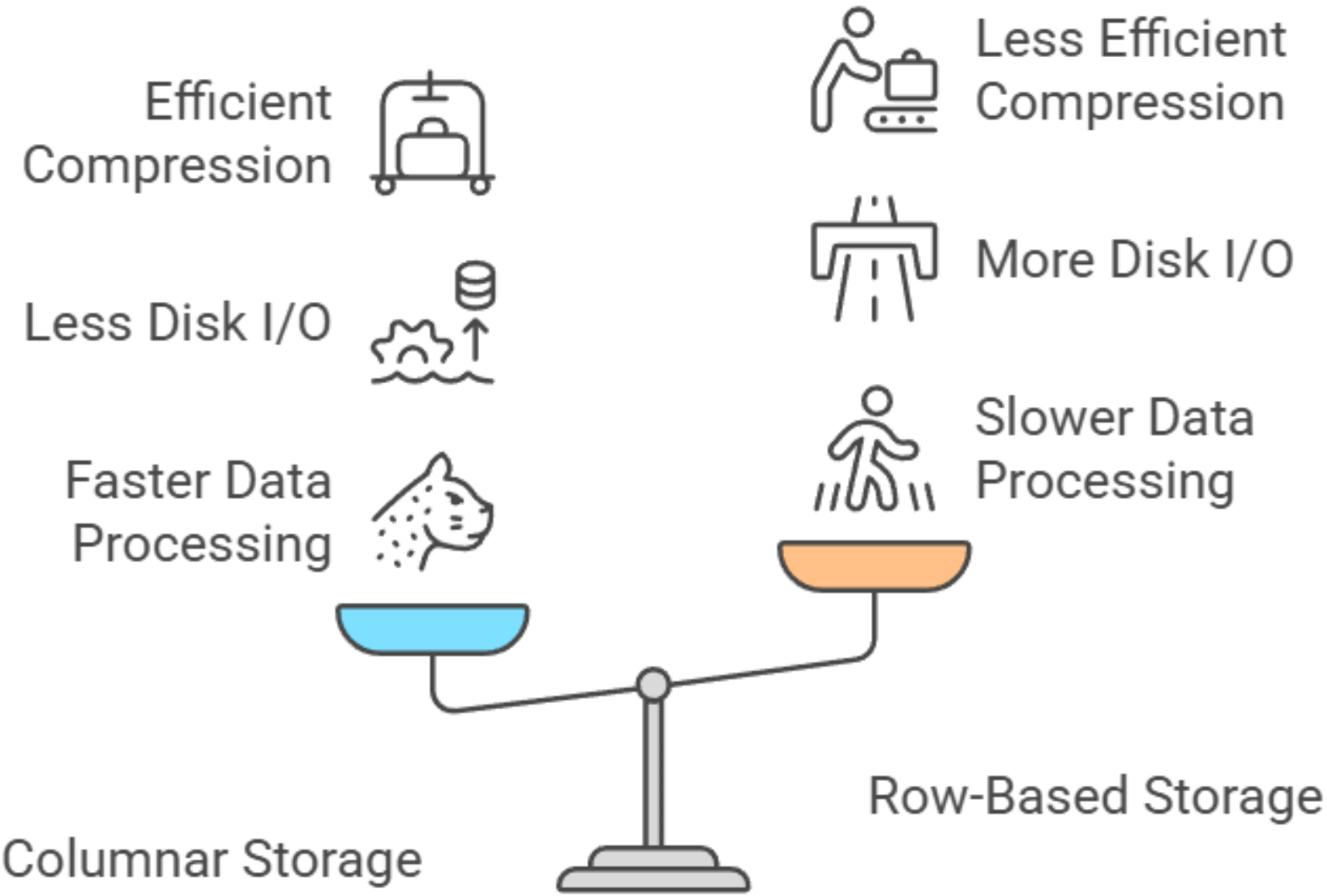




columns are stored
in same/adjacent
block

efficient read when few
columns are required

better compression at
column level



Columnar storage enhances data processing efficiency.

Employee Table		
emp_name	emp_city	emp_age
Mark	Amsterdam	34
Louise	Paris	26
Raghav	Hyderabad	28
Emma	Sydney	27

KEY

Louise

Slice - 1

Raghav
Emma

Slice - 2

Mark

Slice - 3

ALL

Mark
Louise
Raghav
Emma

Slice - 1

Mark
Louise
Raghav
Emma

Slice - 2

Mark
Louise
Raghav
Emma

Slice - 3

EVEN

Mark
Emma

Slice - 1

Louise

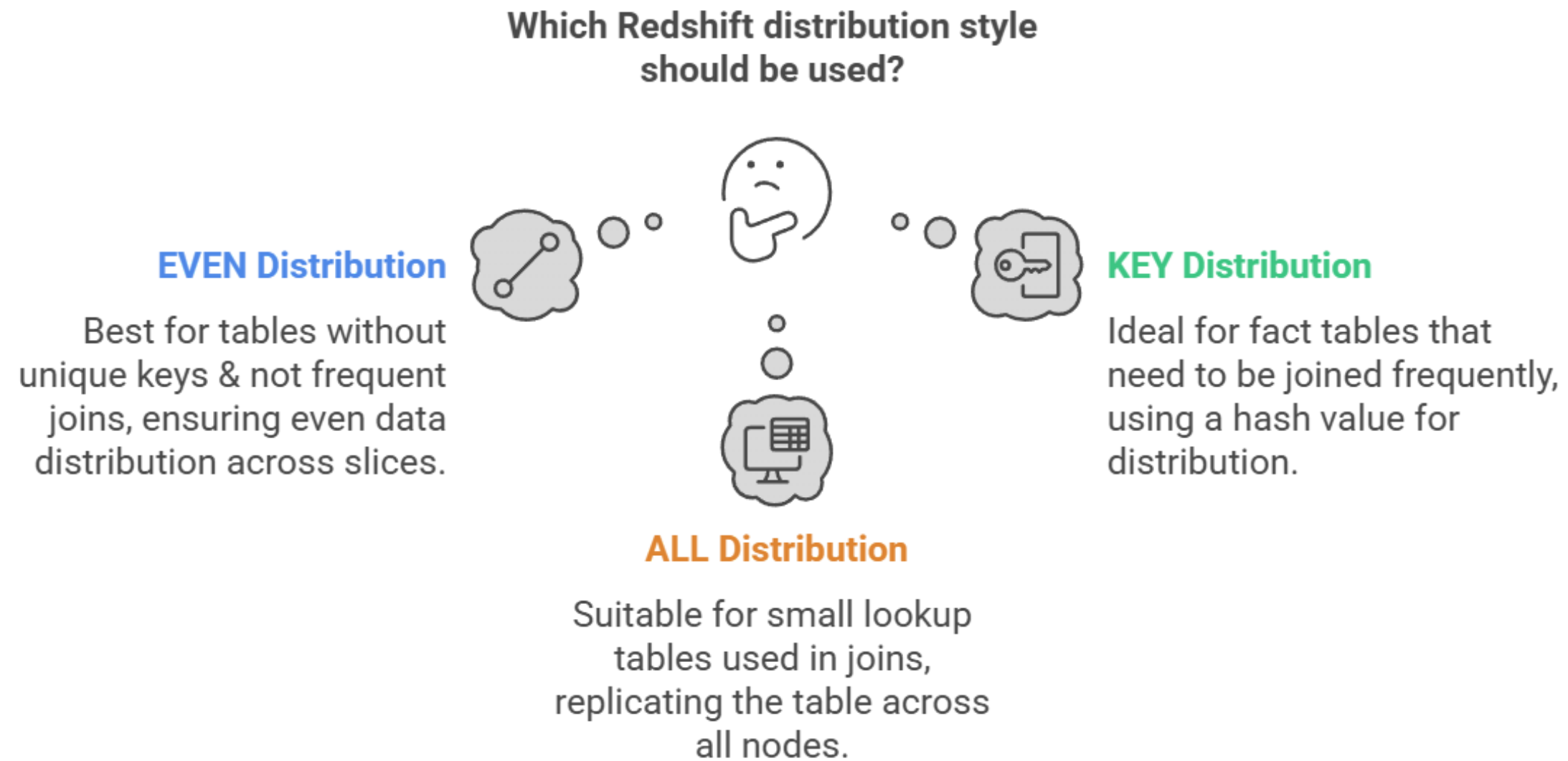
Slice - 2

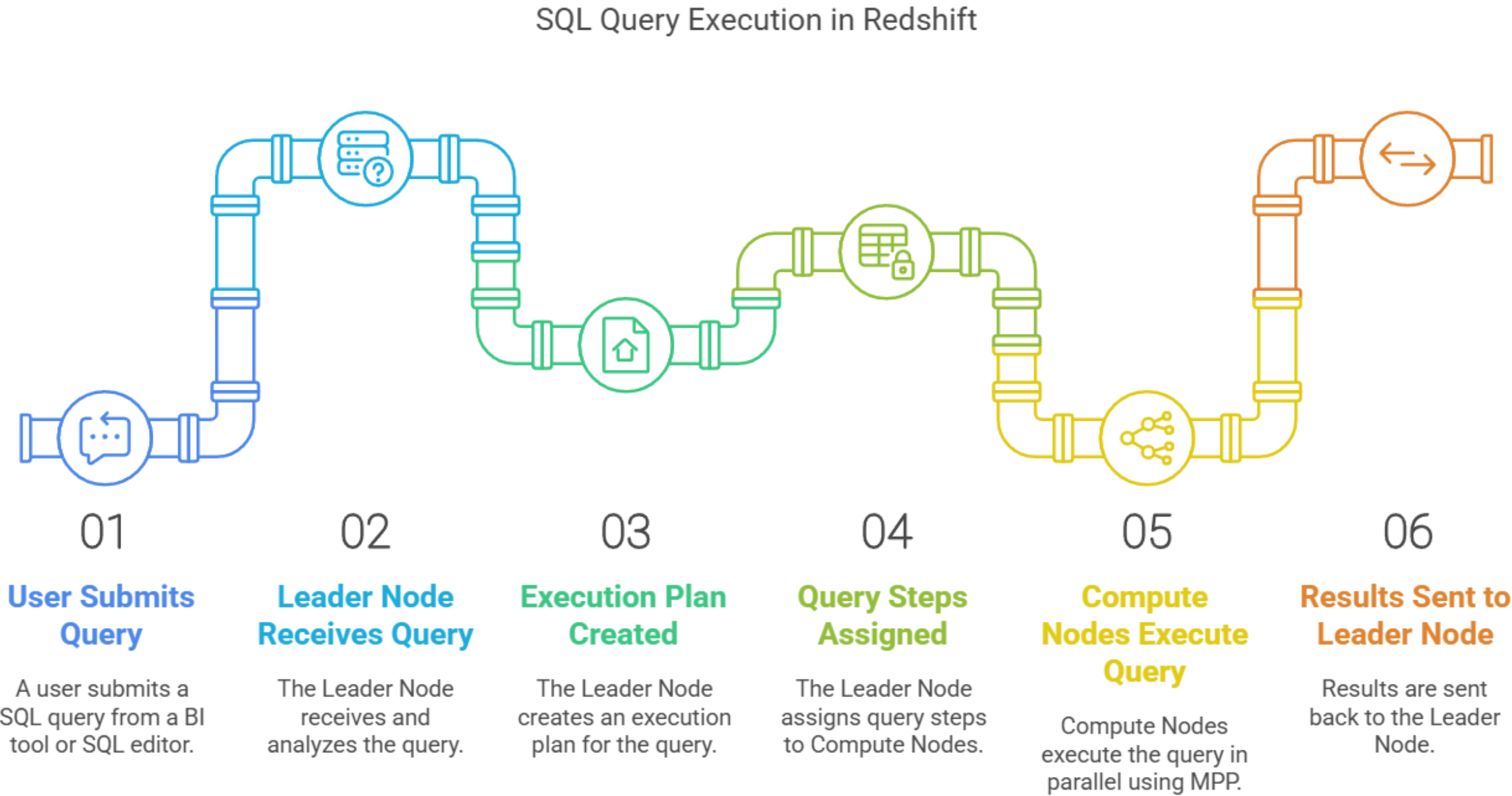
Raghav

Slice - 3

DISTRIBUTION
STYLE



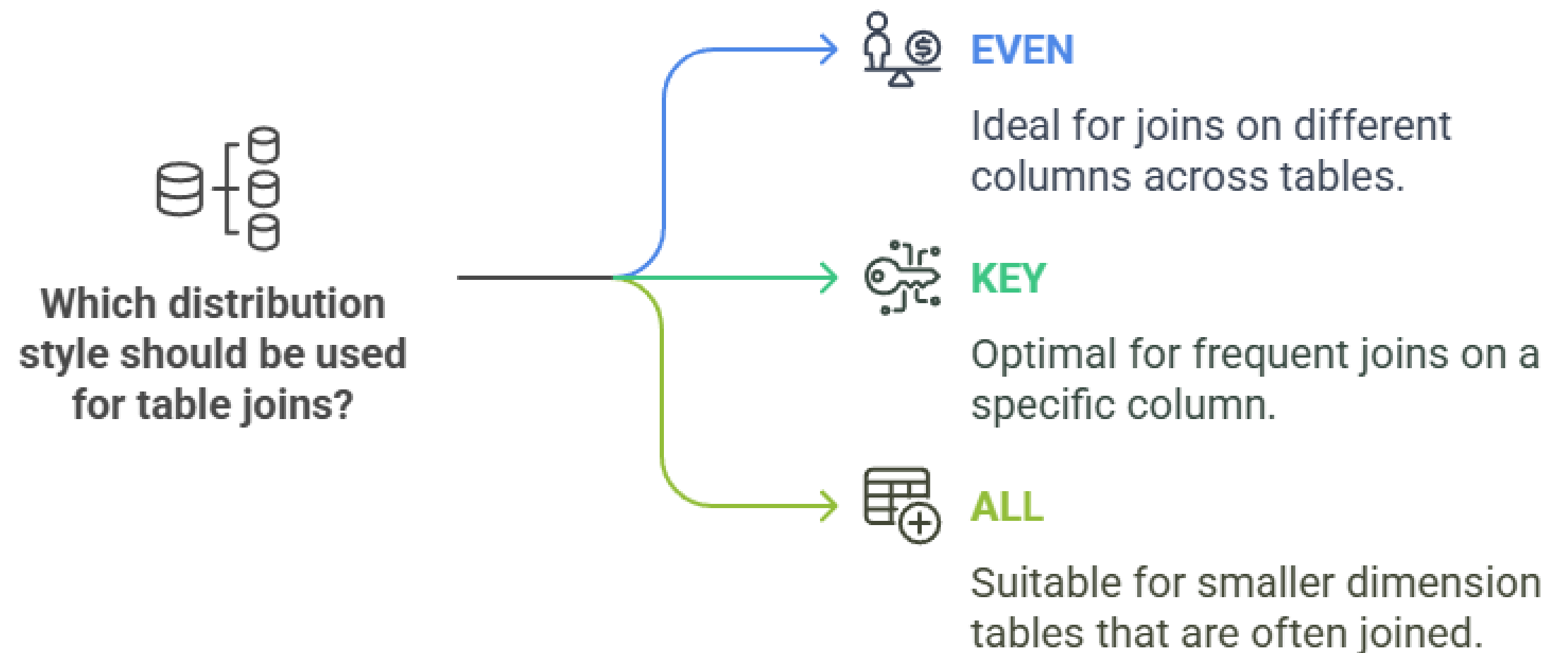




Key insights on table design importance:

Distribution Style Impact:

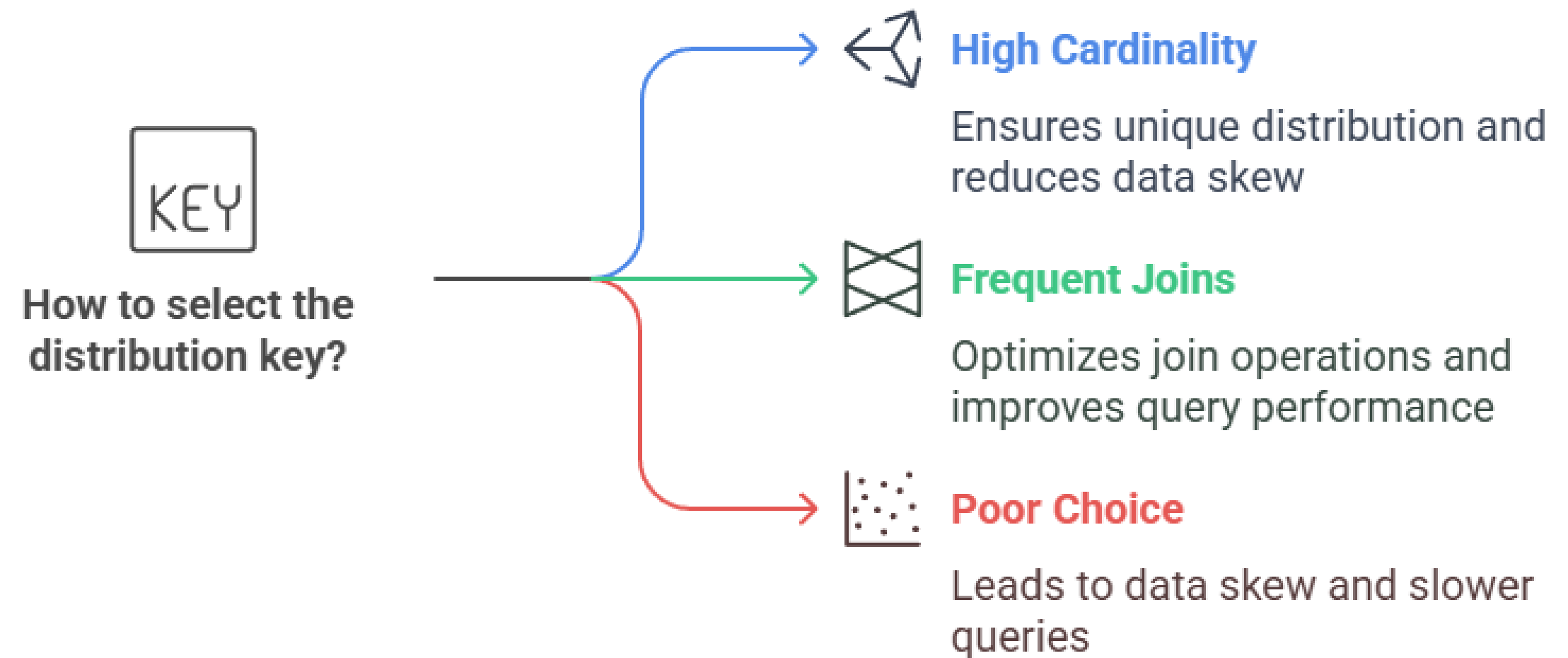
- **EVEN:** Good for tables where joins are done on different columns
- **KEY:** Best when you frequently join tables on a specific column
- **ALL:** Great for smaller dimension tables that are frequently joined



Key insights on table design importance:

Distribution Key Selection:

- Choose columns with high cardinality (many unique values)
- Choose columns frequently used in joins
- Poor distkey choice can lead to data skew and slower queries
-



Made with Napkin

Key insights on table design importance:

Sort Key Benefits:

- Speeds up range-restricted scans
- Improves performance of ORDER BY and GROUP BY
- Compound sort keys work best when columns are used together
- Interleaved sort keys balance the weight of each column

Optimizing Sort Key Strategies



Key insights on table design importance:

Performance Indicators to Watch:

- Data skew across slices
- Disk-based operations (shows insufficient memory)
- Data redistribution during query execution
- Query elapsed time

Performance Indicators

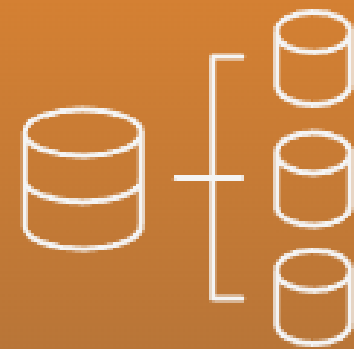


AWS Learning Objectives



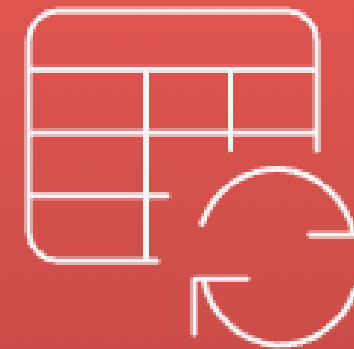
Querying S3 Files

Learn how to query files in S3 without loading.



AWS Glue Benefits

Discover how AWS Glue simplifies data management tasks.



Redshift Data Sharing

Explore methods for sharing data across Redshift clusters.



Problem Identification

Learn to identify and resolve issues using system tables.

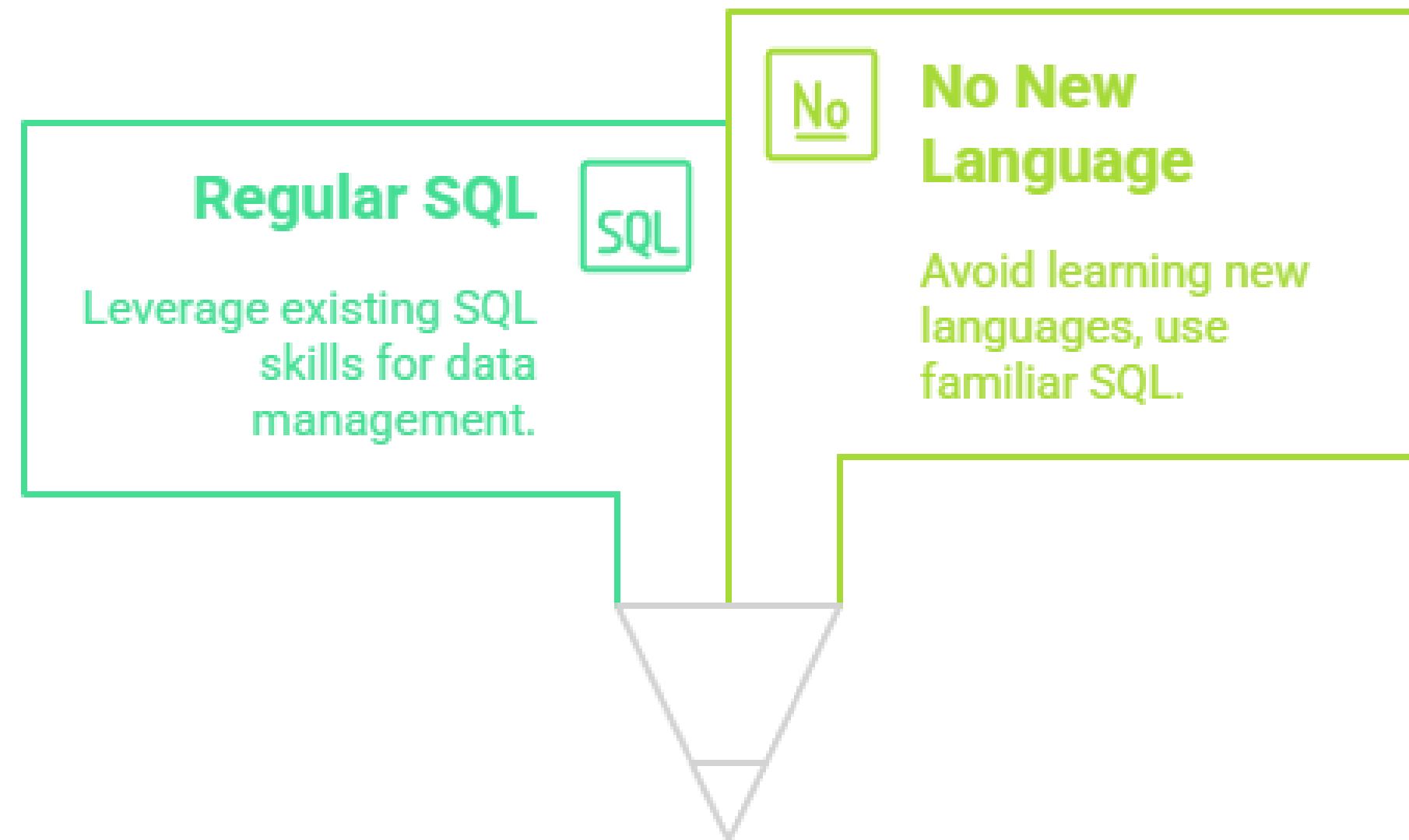
Choose the best method for querying data in S3



Choose the best storage approach for cost savings and efficiency.



SQL Simplicity



Harmonizing Database Elements

