Cycle 08 AWS Homework Task 1: OLAP vs. OLTP – Deep Dive Comparison

Category	OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)
Primary Purpose	Handles real-time transactional processing (e.g., insert/update/delete)	Supports complex queries for data analysis and decision making
Data Structure	Highly normalized schema (3NF or higher)	Denormalized/star or snowflake schema for faster read performance
Operation Type	Write-heavy (frequent inserts/updates)	Read-heavy (complex aggregations, queries)
Query Complexity	Simple and fast queries (CRUD operations)	Complex queries with aggregations, joins, and historical data
Response Time	Milliseconds to seconds (for user-facing applications)	Seconds to minutes (depends on query size and complexity)
Users	Frontline employees, customers	Data analysts, business intelligence teams
Data Volume	Handles small transactions frequently	Works with large volumes of historical data
Backup/Recovery	Critical, must be fast and consistent	Periodic backups, less frequent write operations
Examples	- Bank transaction system - E-commerce checkout system	- Sales performance dashboard - Marketing campaign analysis

Task 2: Cost-Benefit Analysis of Storage Options

Scenario A: When Redshift Managed Storage is Justified

Business Context:

A large retail chain with multiple physical stores across the country needs to run **daily complex analytics** on massive volumes of data — including sales, inventory, customer footfall, and real-time POS (Point of Sale) data — to make pricing and stocking decisions.

Why Redshift Managed Storage?

- **Performance:** Redshift offers high-speed performance with columnar storage, result caching, and query optimization techniques.
- **Concurrency Scaling:** Supports multiple users running simultaneous complex queries.
- **Complex Joins:** Optimized for multi-table joins and large aggregations which are common in retail analytics.
- **Data Size:** Daily data load in terabytes — making Redshift's scalability crucial.
- **Justification:** Though more expensive, the performance gains and ability to handle large analytical workloads justify the cost.

Scenario B: When Querying from S3 is More Efficient

Business Context:

A media company stores terabytes of archived video metadata in Amazon S3. Analysts occasionally need to run ad-hoc queries to generate monthly reports or identify trends.

Why Querying from S3 (using Athena or Spectrum)?

- **Infrequent Access:** Data is accessed occasionally, making fully managed storage overkill.
- **Cost Efficiency:** No need to maintain persistent infrastructure. Pay-per-query model suits low-frequency access.
- **Simplicity:** Easy integration with existing S3-based data lake.
- **Elasticity:** Automatically scales based on query size without provisioning.
- **Justification:** For rarely queried, archived data — querying directly from S3 saves both time and infrastructure cost.

Learning Outcomes

- Understood the fundamental differences between **transactional** and **analytical** processing systems.
- Gained clarity on choosing between **Redshift managed storage vs S3 querying**, based on access patterns and data volume.
- Learned how **cost-performance tradeoffs** impact architecture decisions in data engineering.