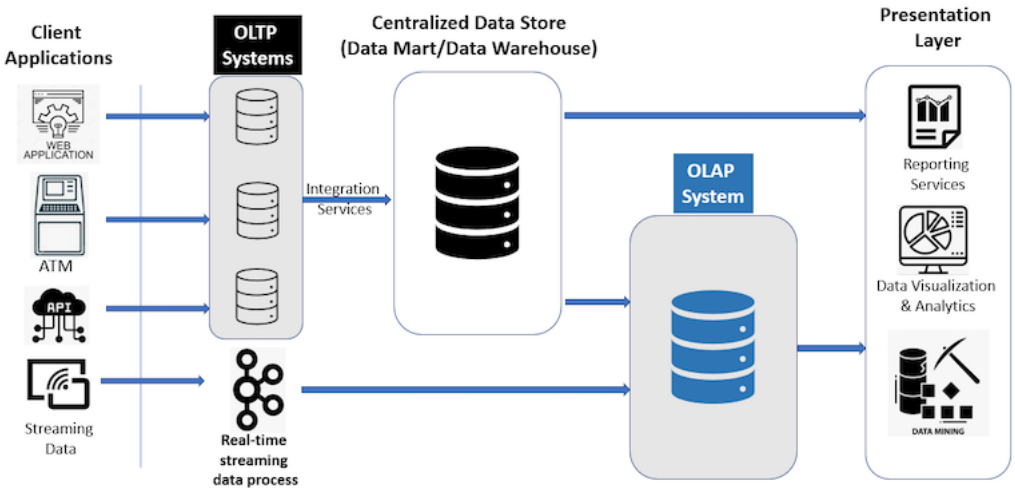
OLAP vs. OLTP

Online Transaction Processing (OLTP) and **Online Analytical Processing (OLAP)** are two fundamental types of data processing systems used in databases and data warehouses.

OLTP focuses on managing real-time operational transactions, while **OLAP** is designed for complex analysis and decision-making based on historical data.



Note: *In some cases, data architecture designs consider OLAP engine is part of Data Warehouse, in some cases one of Data Warehouse layer serves the need of OLAP engine.*

[1]

To provide a clear comparison, below is a table outlining the key differences across various aspects, based on established definitions and use cases from industry sources.

Aspect	OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)
Purpose	Handles day-to-day operational tasks, such as processing real-time transactions (e.g., bank withdrawals, e-commerce orders).[2][3]	Supports strategic decision-making through data analysis, reporting, and insights (e.g., sales trends over years).[4][5]
Data Focus	Deals with current, real-time data that is constantly updated; emphasizes accuracy and integrity for individual records. [4][6]	Works with historical, aggregated data from multiple sources; focuses on long-term patterns and summaries.[7][8]

[1] OLAP vs. OLTP | Key Differences | Blog | ByteHouse
[2] OLAP vs. OLTP | Compare | aws.amazon
[3] OLAP vs. OLTP | Key Differences | Fundamentals | Snowflake
[4] OLAP vs. OLTP | Key Differences in DBMS | DBMS | GeeksForGeeks
[5] OLAP vs. OLTP | Key Differences | Topics | Think | IBM

Query Type	Simple, standardized queries (e.g., INSERT, UPDATE, DELETE) involving few records; optimized for speed in transactions. [6][9]	Complex, ad-hoc queries (e.g., aggregations, joins across large datasets) involving many records; may include multidimensional analysis. [2][5]
Database Design	Typically uses normalized relational databases (e.g., row-oriented storage) to minimize redundancy and ensure data consistency. [2][10]	Often employs denormalized structures like data warehouses or cubes (e.g., columnar storage, star/snowflake schemas) for efficient querying. [4][10]
Performance	High throughput for many small transactions; response times in milliseconds; prioritizes write operations and concurrency. [10][11]	Optimized for read-heavy operations; queries can take seconds to minutes due to large-scale aggregations. [5][10]
Users	Front-line employees, customers, or applications (e.g., cashiers, online shoppers); many concurrent users performing routine tasks. [9]	Analysts, managers, or executives (e.g., data scientists, business intelligence teams); fewer users but with advanced querying needs. [3]
Data Volume	Processes small amounts of data per transaction but handles high volumes of transactions overall. [9][11]	Deals with very large datasets (terabytes to petabytes) for comprehensive analysis. [7][11]
Operations	Frequent writes (inserts, updates, deletes) with some reads; requires strong ACID compliance for transaction integrity. [6][10]	Primarily read-only with batch updates; focuses on summarization and multidimensional views. [2][4]
Examples	Banking systems, retail POS, airline reservations. [2]	Business intelligence tools, sales forecasting, market research reports. [8]

[6] OLAP vs. OLTP | Key Differences | Resources | StitchData
[7] OLAP vs. OLTP | Key Differences | Blog | DataCamp
[8] OLAP vs. OLTP | Key Differences | Blog | Estuary
[9] OLAP vs. OLTP | Key Differences | Questions | StackOverflow
[10] OLAP vs. OLTP | Key Differences | Comments | DataEngineering | Reddit
[11] OLAP vs. OLTP | Comparison | Blog | InerWorks