OLAP Implementation Techniques

Demand for Online Analytical Processing (OLAP)

- Data warehouses support strategic decisions by providing substantial data for analysis.
- Dimensional modeling helps cast data for meaningful analysis, supporting multidimensional analysis across business dimensions.

Need for Multidimensional Analysis

- Business models are inherently multidimensional (e.g., sales data linked to dates, products, stores, promotions).
- Complex queries allow comparisons (e.g., sales vs. targets, breakdowns by products, stores, or regions).
- Decision-makers need to drill down and roll up through dimensions (e.g., time, geography) for effective analysis.

Example: Users don't just ask, "How many units of Product A were sold?" but rather, "How much revenue did Product X generate in the last three months, broken down by territory, store, and promotion?"

Limitations of Traditional Methods

- **Reports**:
    - Provide SQL-based retrieval and formatting.
    - Lack support for multidimensional analysis.
- **Spreadsheets**:
    - While capable of "what if" analysis, they cannot handle large datasets or complex, multidimensional queries effectively.

OLAP Definition

OLAP is a category of software that allows analysts and managers to gain insights through fast, interactive access to data in multiple dimensions. The key elements of OLAP include speed, consistency, multidimensionality, and interactive data analysis.

Fundamental OLAP Guidelines:

1. **Multidimensional View**: Show data in a way that matches how business users think. This makes it easier for them to analyze and understand.
2. **Transparency**: Make the system and data sources easy to use and understand, so users don’t need to worry about the technical side.
3. **Accessibility**: Allow access only to the data needed for analysis, showing it in a clear, organized way.

4. **Consistent Performance**: Reports should load quickly, no matter how big the database is or how many different types of data are used.
5. **Client/Server Setup**: Use a client/server model to make the system flexible and efficient.
6. **Equal Treatment of Dimensions**: Treat all parts (dimensions) of data the same way without favoring one over another.
7. **Efficient Storage**: Automatically adjust how data is stored to handle large gaps (sparse data) efficiently.
8. **Support Multiple Users**: Let multiple people work on different or the same data models at once while keeping data secure.
9. **Cross-Dimensional Analysis**: Allow calculations and comparisons across different types of data without limits.
10. **Easy Data Manipulation**: Make actions like pivoting, drilling down, or rolling up data simple through clicks rather than complicated menus.
11. **Flexible Reporting**: Allow users to easily arrange data in different formats, such as rows and columns.
12. **Many Dimensions and Levels**: Support models with 15-20 dimensions and limitless levels for data grouping.

Additional OLAP Requirements

1. **Detailed Views**: Allow switching from summary data to detailed information in the data warehouse.
2. **Analysis Models**: Support various analysis methods, like explaining or comparing data.
3. **Safe Data Handling**: Ensure calculations do not alter the source data.
4. **Save Results Safely**: Avoid using tools that can change data on systems that handle transactions.
5. **Ignore Missing Values**: Skip over any missing data during analysis.
6. **Update Data Incrementally**: Allow for gradual updates to data, rather than replacing everything at once.
7. **SQL Compatibility**: Work smoothly with other existing systems that use SQL.

Characteristics of OLAP in Plain Language

- Provides a multidimensional and logical view of data in the warehouse.
- Facilitates interactive querying and complex analyses for users.
- Allows drill-down for detailed views and roll-up for aggregated metrics across dimensions.
- Enables intricate calculations and comparisons.
- Presents data results in various formats, including charts and graphs.

General Features of OLAP

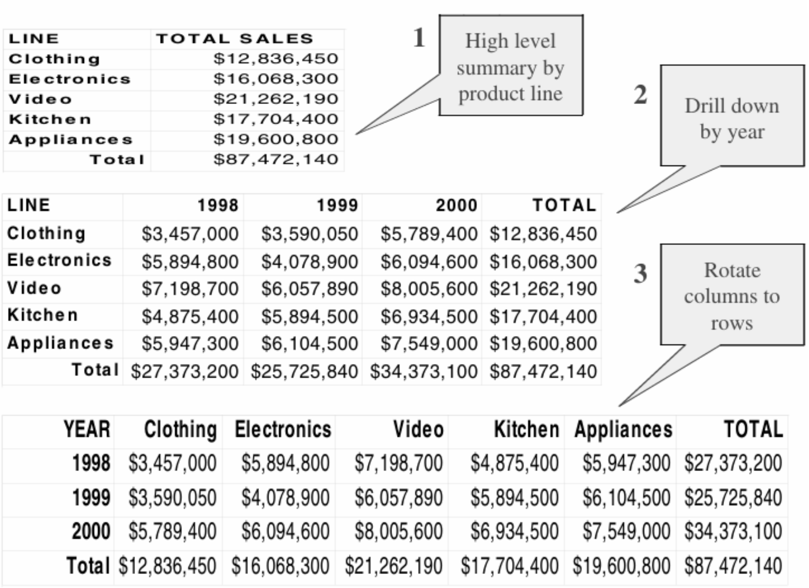
OLAP systems provide a set of tools for interactive and multidimensional analysis, focusing on high performance, ease of use, and flexible data representation. Major features can be categorized into basic and advanced types:

Basic Features:

1. **Multidimensional Analysis:** A fundamental feature allowing users to analyze data across various dimensions (e.g., time, product, location).
2. **Drill-down and Roll-up:** These functions enable users to explore data at different levels of detail. Drill-down allows looking into finer details, while roll-up consolidates data into broader categories.
3. **Multiple View Modes:** Ability to view data in various formats, such as tables, charts, or pivot tables.
4. **Powerful Calculations:** Includes the capability to perform advanced calculations across dimensions.
5. **Slice-and-Dice or Rotation:** Users can extract and analyze a subset of data by slicing (focusing on one dimension) or dicing (creating a smaller, specific dataset).
6. **Fast Response Times for Interactive Queries:** Provides immediate feedback to user queries, allowing for interactive analysis.

Advanced Features:

1. **Drill-through Across Dimensions or Details:** Allows users to access detailed data from multiple dimensions simultaneously.
2. **Consistent Performance:** Ensures that query performance remains consistent, even as data volume grows.
3. **Time Intelligence:** Enables analysis based on time-based attributes like year-to-date or fiscal periods.
4. **Cross-Dimensional Calculations:** Supports complex calculations that involve multiple dimensions.
5. **Sophisticated Presentations and Displays:** Provides advanced visualization tools to display analysis results.
6. **Application of Alert Technology:** Uses alerts to notify users about specific conditions in the data.
7. **Pre-calculation or Pre-consolidation:** Involves pre-computing data summaries to speed up queries.



LINE	TOTAL SALES
Clothing	$12,836,450
Electronics	$16,068,300
Video	$21,262,190
Kitchen	$17,704,400
Appliances	$19,600,800
Total	$87,472,140

1 High level summary by product line

2 Drill down by year

LINE	1998	1999	2000	TOTAL
Clothing	$3,457,000	$3,590,050	$5,789,400	$12,836,450
Electronics	$5,894,800	$4,078,900	$6,094,600	$16,068,300
Video	$7,198,700	$6,057,890	$8,005,600	$21,262,190
Kitchen	$4,875,400	$5,894,500	$6,934,500	$17,704,400
Appliances	$5,947,300	$6,104,500	$7,549,000	$19,600,800
Total	$27,373,200	$25,725,840	$34,373,100	$87,472,140

3 Rotate columns to rows

YEAR	Clothing	Electronics	Video	Kitchen	Appliances	TOTAL
1998	$3,457,000	$5,894,800	$7,198,700	$4,875,400	$5,947,300	$27,373,200
1999	$3,590,050	$4,078,900	$6,057,890	$5,894,500	$6,104,500	$25,725,840
2000	$5,789,400	$6,094,600	$8,005,600	$6,934,500	$7,549,000	$34,373,100
Total	$12,836,450	$16,068,300	$21,262,190	$17,704,400	$19,600,800	$87,472,140

Dimensional Analysis

Dimensional analysis is central to OLAP, where data is organized into multiple dimensions (e.g., time, product, store). The analysis involves looking at data in a multidimensional space, often visualized using cubes. A simple three-dimensional cube can represent data with axes like product (X-axis), time (Y-axis), and store (Z-axis).

For example, sales data can be represented in a cube where you can "slice" a specific view, such as total sales for "coats" in January at the "New York" store. "Slicing and dicing" helps analysts focus on specific aspects of the data for deeper insights.

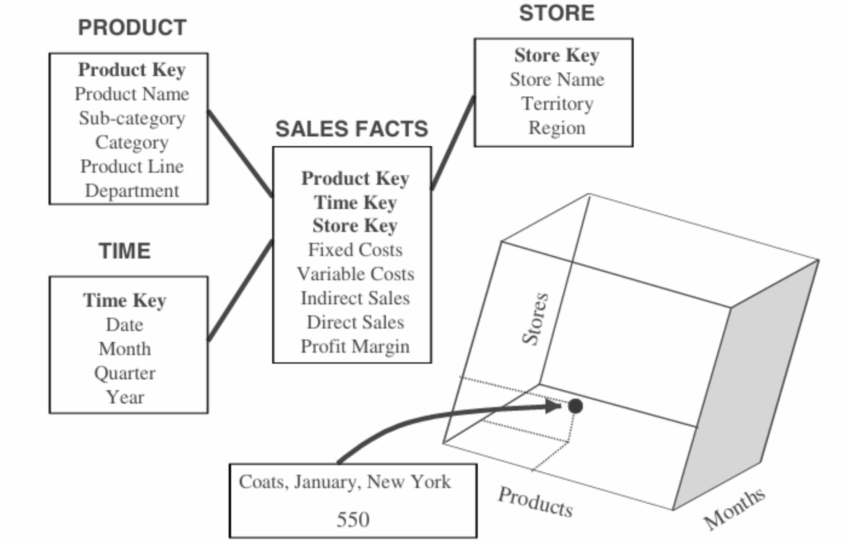
Queries in Multidimensional Analysis

Different queries can be executed on the data cube to get meaningful insights:

- **Total Sales:** Comparing total sales for products across years.
- **Yearly Comparisons:** Displaying sales differences between years, along with the percentage of increase or decrease.
- **Store/Product Analysis:** Showing product sales comparisons across different stores.
- **Rotating Data Views:** Switching rows, columns, and pages to view data from various angles.

Handling More Dimensions

With more than three dimensions, OLAP represents data as "hypercubes," expanding the multidimensional analysis capabilities beyond simple three-dimensional cubes. This allows analysts to explore data across numerous dimensions seamlessly.



What are Hypercubes?

Hypercubes, in the context of multidimensional data modeling, represent a way to visualize and organize data along multiple business dimensions. Essentially, they provide a metaphor for organizing complex data, allowing for the analysis of metrics across various dimensions.

Visualizing Data with Hypercubes

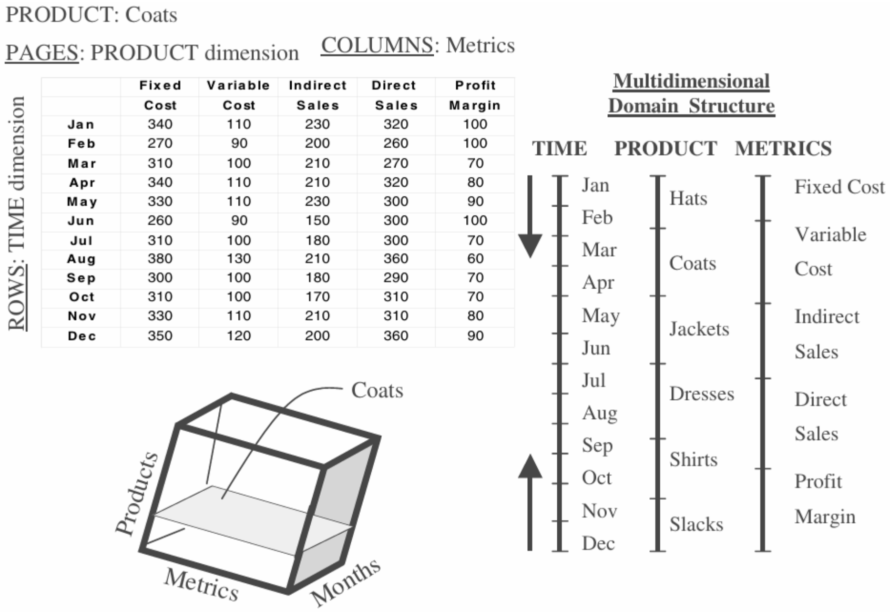
To understand hypercubes, let’s start with a simple case: a data model with three dimensions. For instance, consider the dimensions:

1. **Product**
2. **Time**
3. **Metrics** (e.g., fixed cost, variable cost, indirect sales, direct sales, profit margin)

This data can be displayed in a spreadsheet format where:

- Rows represent the **time** dimension (e.g., months of the year).
- Columns represent different **metrics** (e.g., costs, sales, margins).
- Each "page" of the spreadsheet represents a single product (e.g., coats, hats).

In this example, you can think of the data as being along the edges of a three-dimensional cube. However, as the number of dimensions increases (e.g., adding "Store" as another dimension), the simple cube metaphor becomes harder to visualize.



PRODUCT: Coats

PAGES: PRODUCT dimension COLUMNS: Metrics

ROWS: TIME dimension

	Fixed Cost	Variable Cost	Indirect Sales	Direct Sales	Profit Margin
Jan	340	110	230	320	100
Feb	270	90	200	260	100
Mar	310	100	210	270	70
Apr	340	110	210	320	80
May	330	110	230	300	90
Jun	260	90	150	300	100
Jul	310	100	180	300	70
Aug	380	130	210	360	60
Sep	300	100	180	290	70
Oct	310	100	170	310	70
Nov	330	110	210	310	80
Dec	350	120	200	360	90

Extending to More Dimensions

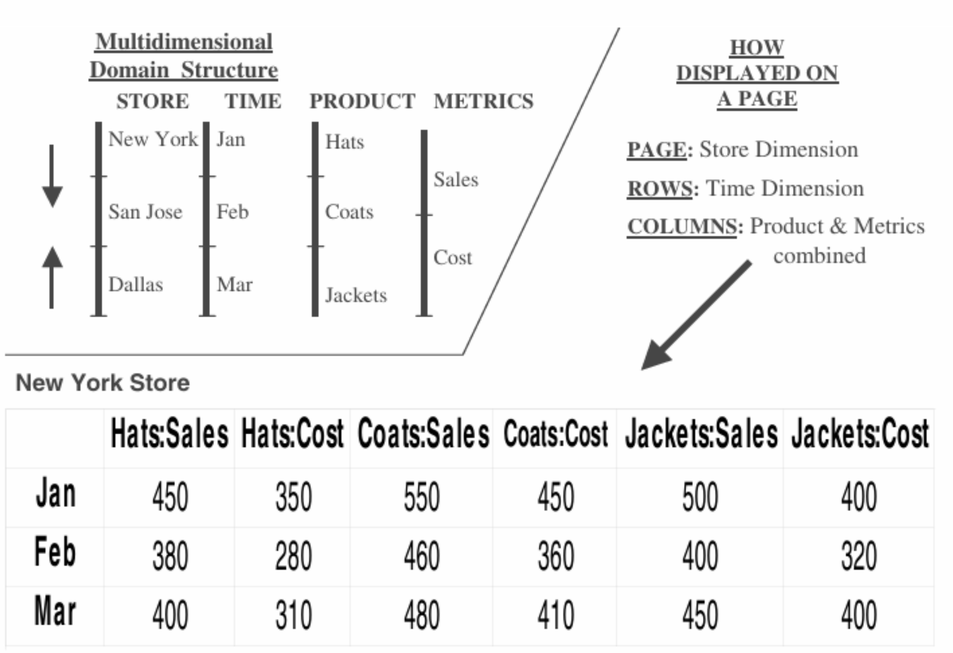
When more than three dimensions are involved (e.g., Product, Time, Metrics, Store), it becomes difficult to represent these in the form of a physical cube. Instead, the concept of a **hypercube** is used. A hypercube can accommodate data with four or more dimensions in a structured way. To represent four dimensions, we use the **Multidimensional Domain Structure (MDS)**, which uses straight lines to depict each dimension.

Example of a Four-Dimensional Model

When a fourth dimension (e.g., "Store") is added to our model:

- The physical cube metaphor breaks down because a cube can only represent up to three dimensions.
- Instead, the MDS approach allows us to represent this model by drawing four lines, each corresponding to one of the dimensions: Product, Time, Store, and Metrics.

A hypercube is essentially a generalization of a cube to more than three dimensions. This structure provides a way to accommodate and analyze data across multiple dimensions, such as time, products, store locations, promotions, etc.



New York Store

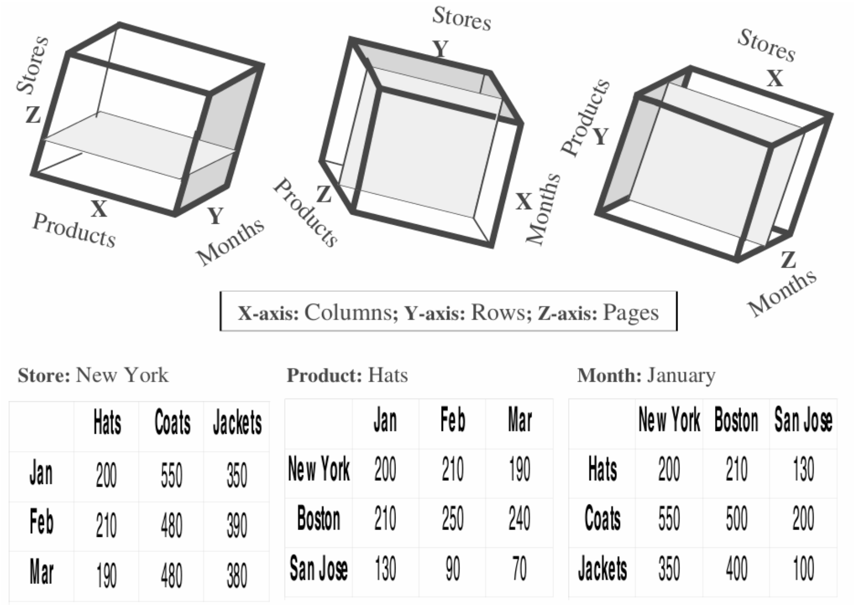
	Hats:Sales	Hats:Cost	Coats:Sales	Coats:Cost	Jackets:Sales	Jackets:Cost
Jan	450	350	550	450	500	400
Feb	380	280	460	360	400	320
Mar	400	310	480	410	450	400

Drill-Down and Roll-Up

- In OLAP systems, hierarchies in dimensions (like the product dimension) allow users to move between different levels of data aggregation.
- **Example Hierarchy**: Product → Subcategory → Category → Product Line → Department.
  - A **department** consists of product lines, which consist of categories, which further break down into subcategories, and finally, products with individual names.
- **Roll-Up**: Moving up the hierarchy to view more aggregated data (e.g., from products to subcategories).
- **Drill-Down**: Moving down the hierarchy to see more detailed data (e.g., from department to individual products).

Slice-and-Dice (Rotation)

- **Slice**: Selecting a specific subset of data from the OLAP cube (e.g., sales data for a single store in one month).
- **Dice**: Viewing data from different perspectives by rotating the data cube.



Store: New York

	Hats	Coats	Jackets
Jan	200	550	350
Feb	210	480	390
Mar	190	480	380

Product: Hats

	Jan	Feb	Mar
New York	200	210	190
Boston	210	250	240
San Jose	130	90	70

Month: January

	New York	Boston	San Jose
Hats	200	210	130
Coats	550	500	200
Jackets	350	400	100

Drill-Across:

- Drill-across is a technique used in OLAP (Online Analytical Processing) that allows you to analyze data across multiple related cubes.
- **Example:** If you have separate cubes for "sales" and "inventory," you can use drill-across to compare sales data with available stock without switching between different reports. This provides a broader view of the data by pulling from multiple cubes at once.

OLAP Models Overview

OLAP (Online Analytical Processing) models are used in data warehousing to support complex analysis and query performance. The different OLAP models—ROLAP, MOLAP, HOLAP, and DOLAP—differ primarily in how they store and process data.

1. MOLAP (Multidimensional OLAP):

MOLAP stores data in multidimensional cubes, which allows fast data retrieval since the results are pre-calculated and stored.

Characteristics:

- **Cubes:** MOLAP stores data in cubes that represent different dimensions, such as geography, product categories, and time (e.g., day, week, month). This enables a multi-dimensional view of data, making it easy to analyze various business aspects.
- **High Performance:** Since data is pre-aggregated and stored in cubes, querying is extremely fast. For example, if you want to know the total sales for a specific product category in a specific region for the last quarter, the system can quickly retrieve this data from the pre-calculated cube.
- **API Access:** MOLAP typically offers an API for front-end tools to access the cube's data, allowing seamless integration with other business intelligence tools.

Example:

Suppose a retail company uses MOLAP for sales analysis. They might have the following dimensions in their sales cube:

- **Product:** Product ID, Category, Subcategory.
- **Time:** Day, Month, Quarter, Year.
- **Geography:** Store, City, State.

This cube will store sales figures at different granularities (e.g., daily, monthly). With the cube already having pre-aggregated data, queries like "total sales of electronics in the last year in California" are answered instantly.

Maintenance Considerations:

- **Data Aggregation:** Each time new data is added (e.g., daily sales), it must be aggregated into the cube. For instance, if there are "year-to-date" summaries in the cube, every new sale entry needs to be rolled up to update this summary.
- **Storage:** The storage required for MOLAP can grow significantly as the number of dimensions and levels of granularity increase. For example, if you have hundreds of products and you want daily, monthly, and yearly aggregates, the number of possible combinations (and hence the storage needed) can become very large.

Scalability Issues:

- MOLAP struggles with high-cardinality dimensions (e.g., products with millions of SKUs) because storing a large number of pre-aggregated results leads to sparse cubes, consuming excessive storage.

Virtual Cubes:

- Virtual cubes are used when you need to analyze data from two or more related cubes that have common dimensions.
- **Example:** Imagine you have a "sales cube" that contains sales data and a "list price cube" that holds product price data. A virtual cube lets you calculate things like discounts across all stores, without needing to store the list price data again.

Partitioned Cubes:

- A partitioned cube is when one logical cube is split into several physical cubes, which are stored across different servers. This helps handle large datasets more efficiently by spreading the data.
- **Example:** A company might divide its sales data by year or region, creating separate partitions and storing them on different servers to improve performance and manage large data volumes better.

2. ROLAP (Relational OLAP):

ROLAP uses relational databases (like SQL-based databases) to store and manage data. It allows for more flexibility and can handle larger datasets compared to MOLAP.

Characteristics:

- **Star or Snowflake Schema:** ROLAP often uses star or snowflake schemas for data organization. In a star schema, a central fact table (e.g., sales) is linked to dimension tables (e.g., product, geography, time).
- **Summary Tables:** ROLAP relies heavily on summary tables to speed up query processing. These tables store pre-aggregated data to answer typical analytical questions.

Example:

For the same retail company, a ROLAP implementation might have:

- A **fact table**: Stores individual sales transactions (product ID, store ID, time ID, quantity, amount).
- **Dimension tables**: Product (product ID, category), Geography (store ID, city, state), Time (time ID, date, month, year).

If the business needs to analyze "total sales of electronics in California for 2023," ROLAP queries the fact table and dimension tables using SQL, often retrieving data from summary tables for faster results.

Maintenance and Storage:

- **Summary Tables:** The number of summary tables can grow if not managed carefully. For example, if you summarize data by week, month, and year across different geographic regions, the number of required tables can increase rapidly.
- **Storage Efficiency:** While summary tables get smaller as dimensions become less detailed (e.g., yearly summary vs. daily summary), ROLAP generally requires double the size of unsummarized data to maintain these summaries.

Scalability:

- ROLAP scales better than MOLAP due to its ability to handle large dimension tables. However, if not managed properly, maintaining summary tables becomes complex.

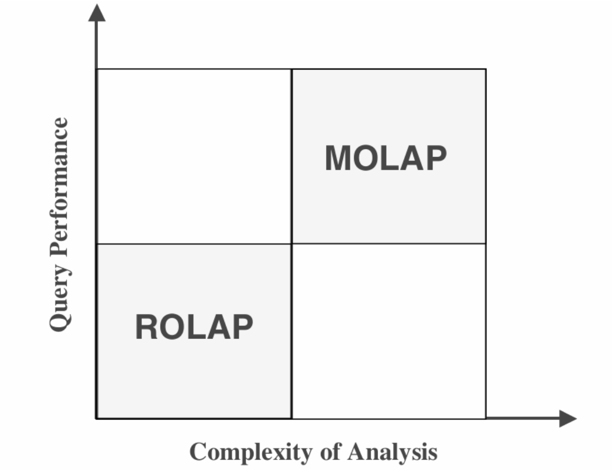
Optimization Techniques:

- **On-the-Fly Summaries:** Some ROLAP tools generate summaries dynamically based on user queries, reducing the need to store all possible pre-aggregates.
- **Aggregate Wizards:** Tools that help DBAs select the most beneficial aggregates to pre-build.

ROLAP vs. MOLAP Summary

- **Data Storage**:
  - **ROLAP**: Stores data in relational tables; handles very large data volumes.
  - **MOLAP**: Uses proprietary databases for storing summary data; suitable for moderate data volumes.
- **Technologies**:
  - **ROLAP**: Uses complex SQL to fetch data; creates data cubes dynamically.

    - **MOLAP**: Relies on specialized databases (MDDBs) and proprietary technology.
- **Query Performance**:
    - **ROLAP**: Slower but more flexible for complex, detailed queries.
    - **MOLAP**: Faster due to pre-fabricated data cubes, ideal for intensive queries.
- **Best Use Case**:
    - **ROLAP**: When flexibility and handling large volumes of detailed data is needed.
    - **MOLAP**: When fast query performance is a priority and data complexity is moderate.



3. HOLAP (Hybrid OLAP):

HOLAP combines the best of both MOLAP and ROLAP, allowing data to be stored in both multidimensional cubes and relational databases.

Characteristics:

- **Flexible Storage:** Stores detailed data in a relational database (ROLAP) while storing summarized data in a cube (MOLAP).
- **Efficient Querying:** Queries that require detailed data can access the relational database, while those needing summarized information can quickly retrieve it from the MOLAP cubes.

4. DOLAP (Desktop OLAP)

- **Purpose**: A variation of ROLAP, designed to provide portability. Datasets are created on the server and then transferred to the user's desktop for analysis.
- **Storage**: Involves using DOLAP software on the desktop, accessing pre-created multidimensional datasets.

OLAP Implementation Considerations

When implementing OLAP (Online Analytical Processing) in your data warehouse, it's important to understand several key aspects:

1. **Types of Data in OLAP**:
   - **Static Summary Data**: Most OLAP summary data is static. This means it consists of data that has been summarized from the information retrieved from the data warehouse. It does not change frequently.
   - **Dynamic Summary Data**: This type of summary data is rare in OLAP environments. It occurs when new business rules require updates or changes to the summary data, making it more fluid.
   - **Permanent Detailed Data**: This is the detailed data that is retrieved from the data warehouse and stored in the OLAP system. It provides a comprehensive view of the data.
   - **Transient Detailed Data**: This data is also detailed but is brought in from the data warehouse on a temporary basis, usually for special analysis or reporting needs.
2. **Limitations**: Complex analysis functions may be limited. While it is easier to drill down to detailed data, drilling across different datasets can be challenging.
3. **Access Speed**: OLAP systems allow for fast access and come with a wide range of functions for complex calculations. They enable easy analysis, regardless of the number of dimensions.
4. **Data Models**: Before starting with OLAP, two main issues must be addressed:
   - **Lack of Standardization**: Each vendor has its own tools and interfaces, which can complicate integration.
   - **Scalability**: OLAP handles summary data effectively but may struggle with large volumes of detailed data.
5. **Data Preparation**: The data warehouse sends data to the OLAP system. In the MOLAP model, data is stored in multidimensional cubes, while in the ROLAP model, it is generated dynamically.
6. **Data Flow**: Avoid building OLAP directly on operational systems. OLAP needs transformed, integrated, and historical data, which is better managed through the data warehouse.
7. **Customizing OLAP Data**: OLAP data is tailored for specific departments. Key steps in preparing OLAP data include:
   - **Subset Selection**: Identify data relevant to a specific department, like marketing.
   - **Summarization**: Create summaries based on departmental needs.
   - **Denormalization**: Merge tables as required by the department.
   - **Calculations**: Include specific metrics needed for that department.
   - **Indexing**: Choose the right attributes for indexing.
8. **Data Modeling**: When modeling OLAP structures, consider the different types of data mentioned earlier (static, dynamic, permanent, and transient).
9. **Administration**: Managing the OLAP environment involves:
   - Understanding user access patterns.
   - Properly selecting business dimensions and filters.

- Efficient methods for moving data into the OLAP system.
- Managing the size of the multidimensional database and ensuring data security.

10. **Performance**: The use of OLAP shifts the workload from the data warehouse, improving query performance. OLAP is designed for complex queries, allowing them to run faster by pre-calculating and storing summaries. However, this necessitates longer intervals between data updates, typically once a month.