Data Warehouse Concepts: Traditional vs. Cloud

Are you on-premise or cloud? Learn about traditional vs cloud data warehouse concepts used by popular data warehouse services.
A data warehouse is any system that collates data from a wide range of sources within an organization. Data warehouses are used as centralized data repositories for analytical and reporting purposes.

Over the last five years, data warehouse architecture has seen a huge shift towards cloud-based warehouses and away from traditional on-site warehouses. Here are a few of the benefits of cloud-based data warehouses, compared to on-premise solutions:

- **Scalability**: It is much easier to scale data warehouses in the cloud compared with on-site warehouses.
- **Cost**: Cloud-based warehouses are cheaper to set up because there are no hardware or upfront licensing costs.
- **Time to market**: It's quick and easy to get a data warehouse up and running in the cloud. Deploying an on-premise data warehouse takes much longer.
- **Performance**: Cloud data warehouses are optimized for analytics. They use columnar storage and massively parallel processing (MPP) which enables much better performance when running complex queries.

Most traditional data warehouse concepts are still relevant in the cloud. But with new technology comes new concepts and ideas, which you'll also need to understand as you move to the cloud.

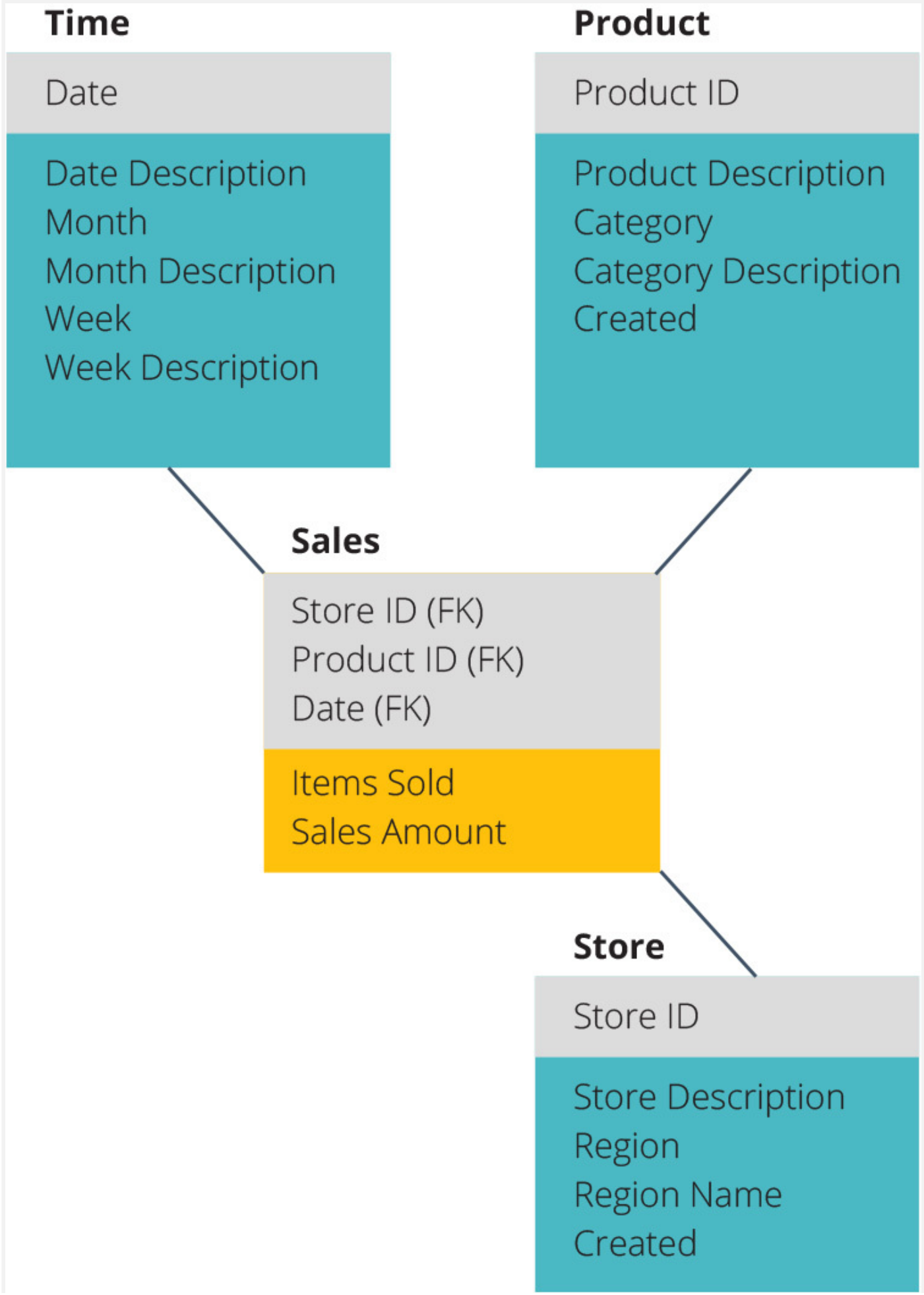
Traditional Data Warehouse Concepts

Data Models

In data warehouses, dimensions are used to categorize and describe facts and measures, in order to meaningfully answer business questions. Dimensions give structured labels to otherwise unordered measures.

A **conceptual model** for a data warehouse defines relationships between the key entities that describe the data

The **logical data model** describes that data in as much detail as possible without considering database design.

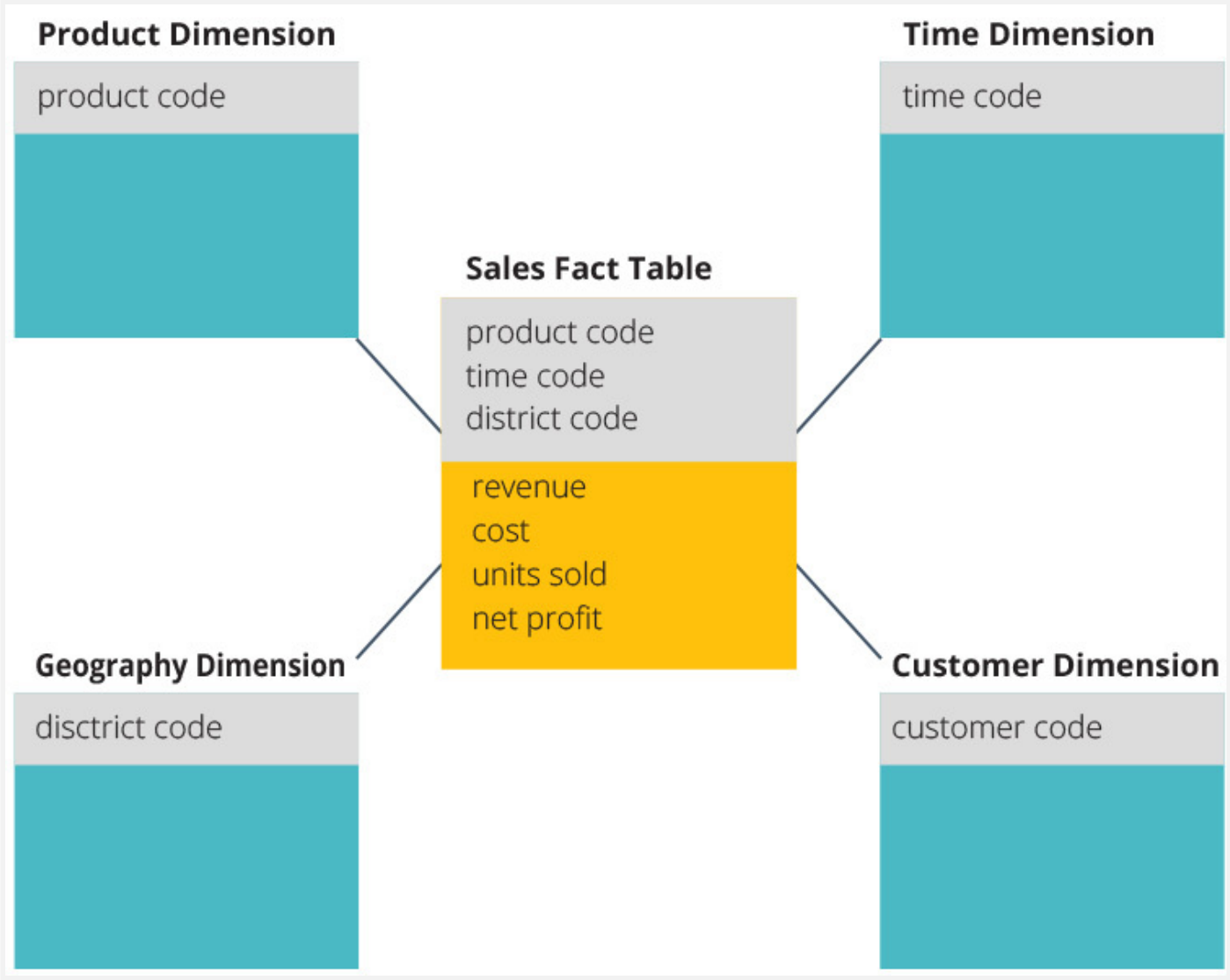

Time
Date
Date Description
Month
Month Description
Week
Week Description
Product
Product ID
Product Description
Category
Category Description
Created
Sales
Store ID (FK)
Product ID (FK)
Date (FK)
Items Sold
Sales Amount
Store
Store ID
Store Description
Region
Region Name
Created

The **physical data model** represents how the model will be incorporated into a database design. The physical data model defines tables, their structure, and the relationship between them. It also specifies data types for columns.

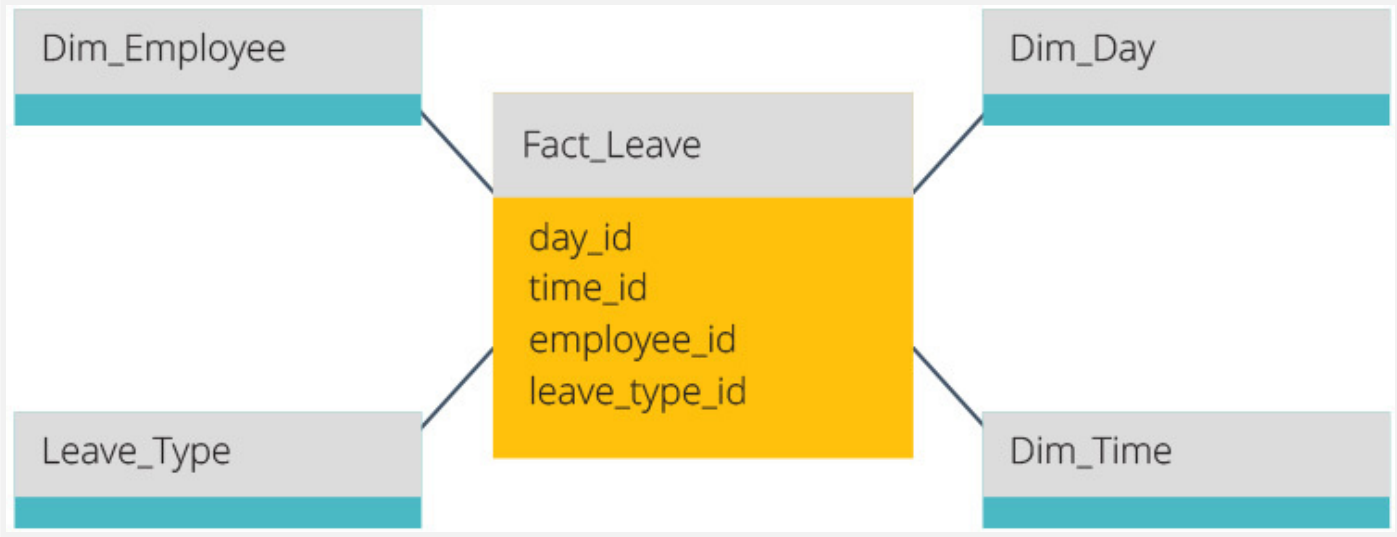
Fact Table

A **fact table** is a table consisting of the metrics, measurements, or facts for a given business process, and their associated dimensions. The fact table is often denormalized, meaning it is grouped and may contain redundancies.

Fact tables typically have two types of columns. The foreign keys columns are used for joining a fact table with a dimension table. The second type of column includes the data (facts or measures) to be analyzed.



A **factless fact table** is a special type of fact table entirely consisting of dimensions—there is no second column containing data. Such tables are useful for tracking events, such as student attendance or employee leave.



Three Tier Architecture

Traditional data warehouses are typically structured in three tiers:

- **Bottom Tier**: The bottom tier is a database server, typically a RDBMS that extracts data from different sources using a gateway. Data sources fed into the bottom tier can include operational databases, and other types of front-end data.
- **Middle Tier**: The middle tier is an OLAP server that either maps the operations on multidimensional data to standard relational operations, or directly implements the operations.
- **Top Tier**: The top tier contains the querying and reporting tools for analysis and business intelligence.

Kimball vs. Inmon

There are two important approaches to data warehouse design, proposed by Bill Inmon and Ralph Kimball. Bill Inmon is an American computer scientist who is recognized as the father of the data warehouse. Ralph Kimball is one of the original architects of data warehousing, and has written several books on the topic.

The two experts had conflicting opinions on how data warehouses should be structured, and have given rise to two schools of thought.

The Inmon approach is a top-down design. With the Inmon methodology, the data warehouse is created first and is seen as the central component of the analytic environment. Data is then summarized and distributed from the centralized warehouse to one or more dependant data marts.

The Kimball approach takes a bottom-up view of data warehouse design. In this architecture, an organization creates separate data marts, which provide views into single departments within an organization. The data warehouse is the combination of these data marts.

OLAP vs. OLTP

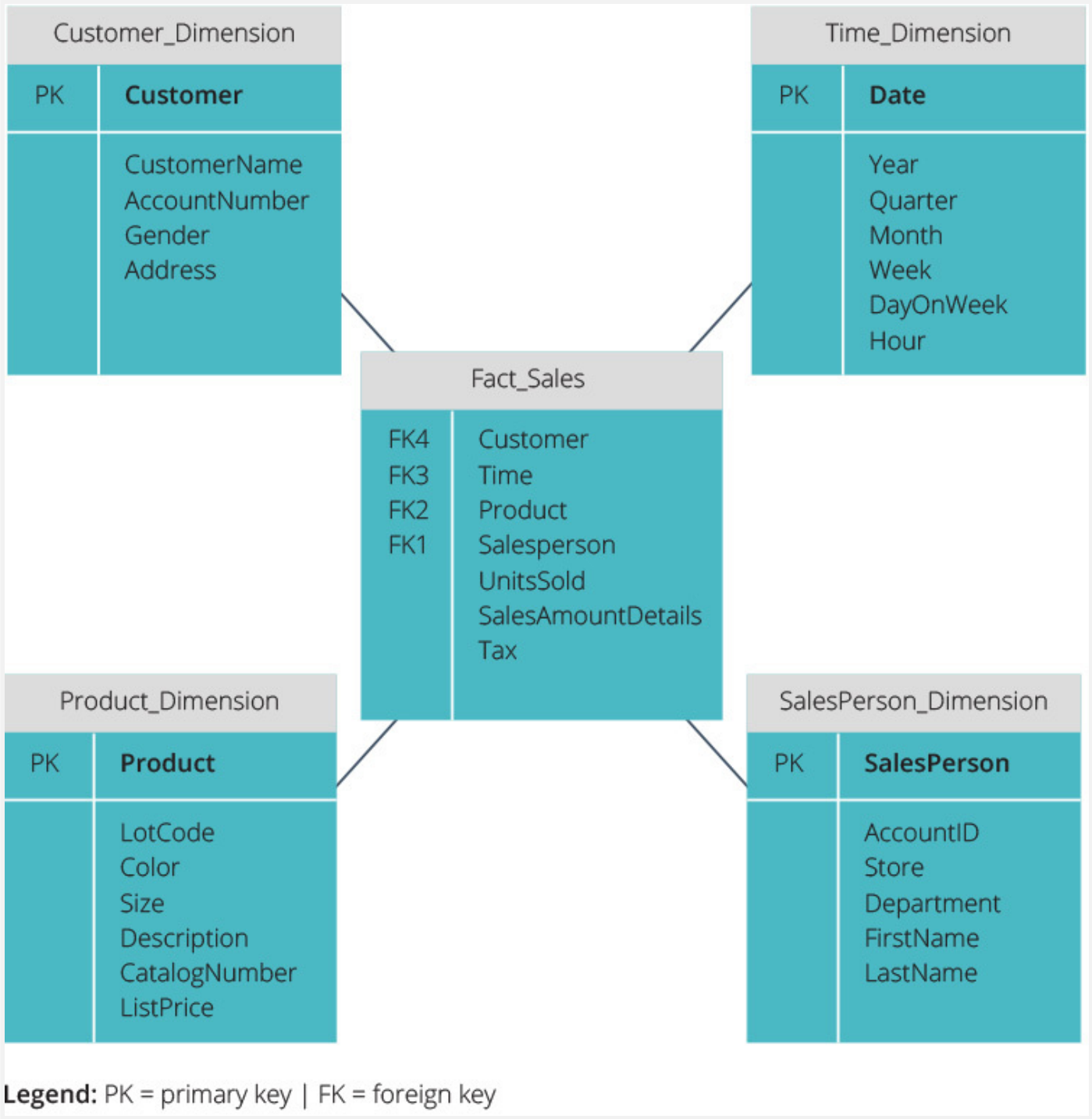
Online transaction processing (OLTP) is characterized by short write transactions that involve the front-end applications of an enterprise's data architecture. OLTP databases emphasize fast query processing and only deal with current data. Such data systems are used for capturing information for business processes, providing source data for the data warehouse.

Online analytical processing (OLAP) is characterized by complex read queries used to perform detailed analysis of historical transactional data. OLAP systems help to analyze the data in the data warehouse.

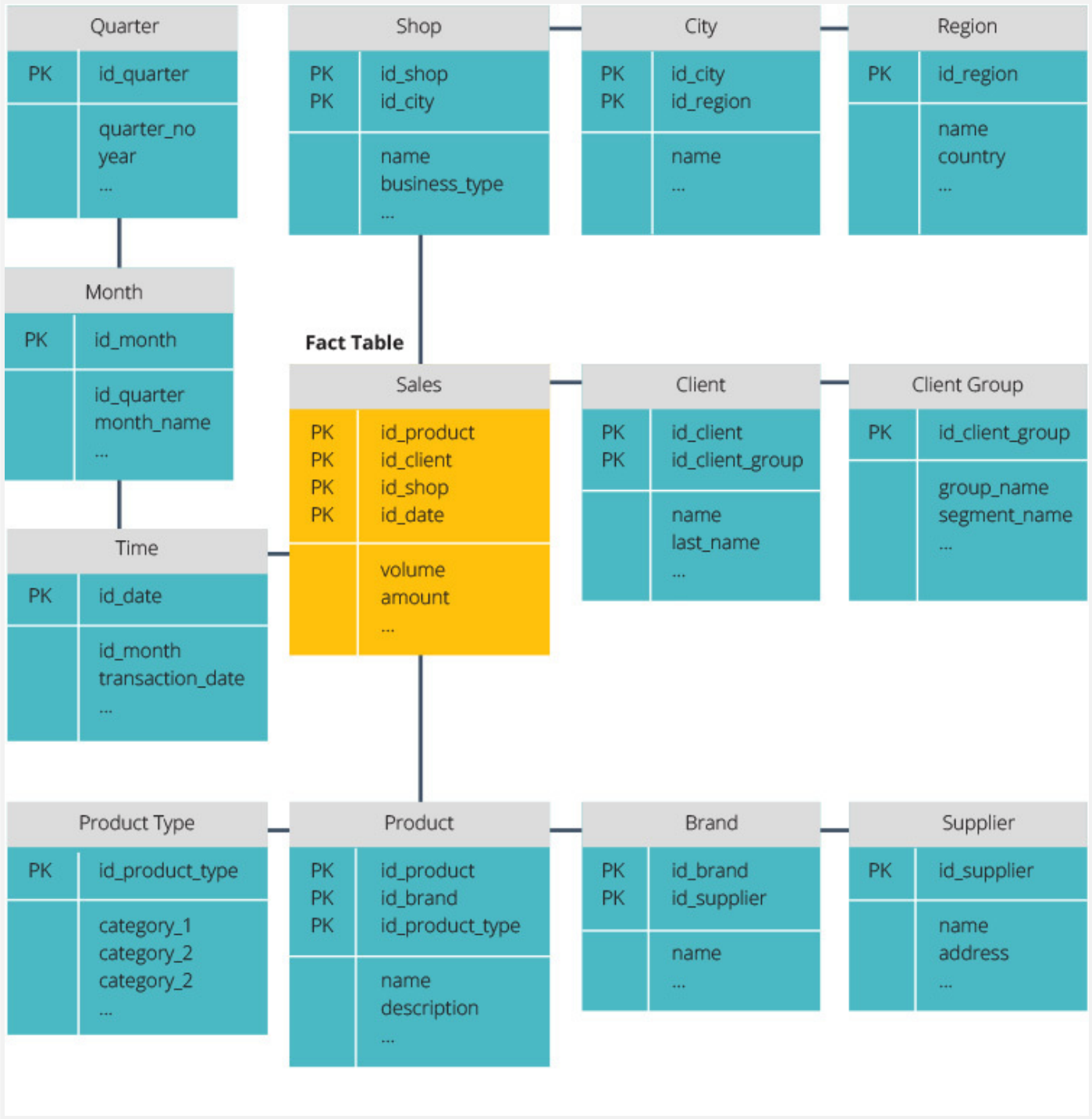
Star Schema vs. Snowflake Schema

The star schema and snowflake schema are simply two different ways of organizing data warehouses. Both schemas use dimension tables that describe the information contained within a fact table.

The star schema takes the information from the fact table and splits it into denormalized dimension tables. The emphasis for the star schema is on query speed. Only one join is needed to link fact tables to each dimension, so querying each table is easy.



The snowflake schema splits the fact table into a series of normalized dimension tables. Normalizing creates more dimension tables with multiple joins and reduces data integrity issues. However, querying is more challenging using the snowflake schema, because queries need to dig deeper to access the relevant data.



ETL vs ELT

Extract, Transform, Load (ETL) describes the process of extracting the data from source systems (typically transactional systems), converting the data to a format or structure suitable for querying and analysis, and finally loading it into the data warehouse. ETL leverages a

separate staging database and applies a series of rules or functions to the extracted data before loading.

Extract, Load, Transform (**ELT**) is a different approach to loading data. ELT takes the data from disparate sources and loads it directly into the target system, such as the data warehouse. The system then transforms the loaded data on-demand to enable analysis.

ELT offers quicker loading than ETL but it requires a powerful system to perform the data transformations on-demand.

Virtual Data Warehouse / Data Mart

Virtual data warehousing uses distributed queries on several databases, without integrating the data into one physical data warehouse.

Data marts are subsets of data warehouses oriented for specific business lines, such as sales or finance. A data warehouse typically combines information from several data marts in multiple business lines, but a data mart contains data from a set of source systems related to one business line.

Enterprise Data Warehouse

An enterprise data warehouse is intended as a unified, centralized warehouse containing all transactional information in the organization, both current and historical. An enterprise data warehouse should incorporate data from all subject areas related to the enterprise, such as marketing, sales, finance, human resources.

Cloud Data Warehouse Concepts

Cloud data warehouses are new. In order to best understand their fundamental concepts, it is best to learn about the leading cloud data warehouse solutions.

Three leading cloud data warehouse solutions are Amazon Redshift,Google BigQuery, and Panoply. Below, we explain key concepts from each of these services. We hope this will provide a general understanding of how modern data warehouses are built.

Cloud Data Warehouse Concepts - Amazon Redshift

The following concepts are used specifically in the Amazon Redshift cloud data warehouse, but may be applicable to additional data warehouse solutions in the future based on Amazon infrastructure.

Clusters

Amazon Redshift's architecture is based on clusters. A cluster is simply a group of shared computing resources, called nodes.

Nodes

Nodes are computing resources that have their own CPU, RAM, and hard disk space. A cluster containing two or more nodes is composed of a leader node and compute nodes.

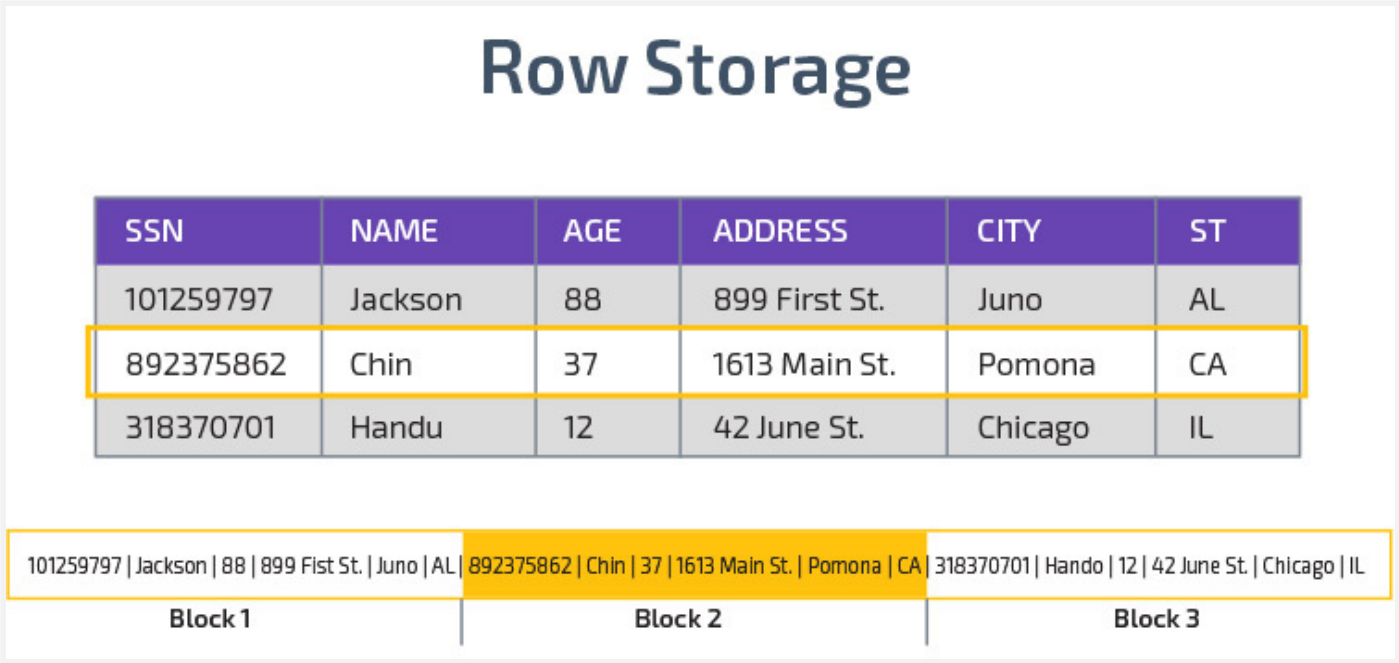
Leader nodes communicate with client programs and compile code to execute queries, assigning it to compute nodes. Compute nodes execute the queries, and returns the results to the leader node. A compute node only executes queries that reference tables stored on that node.

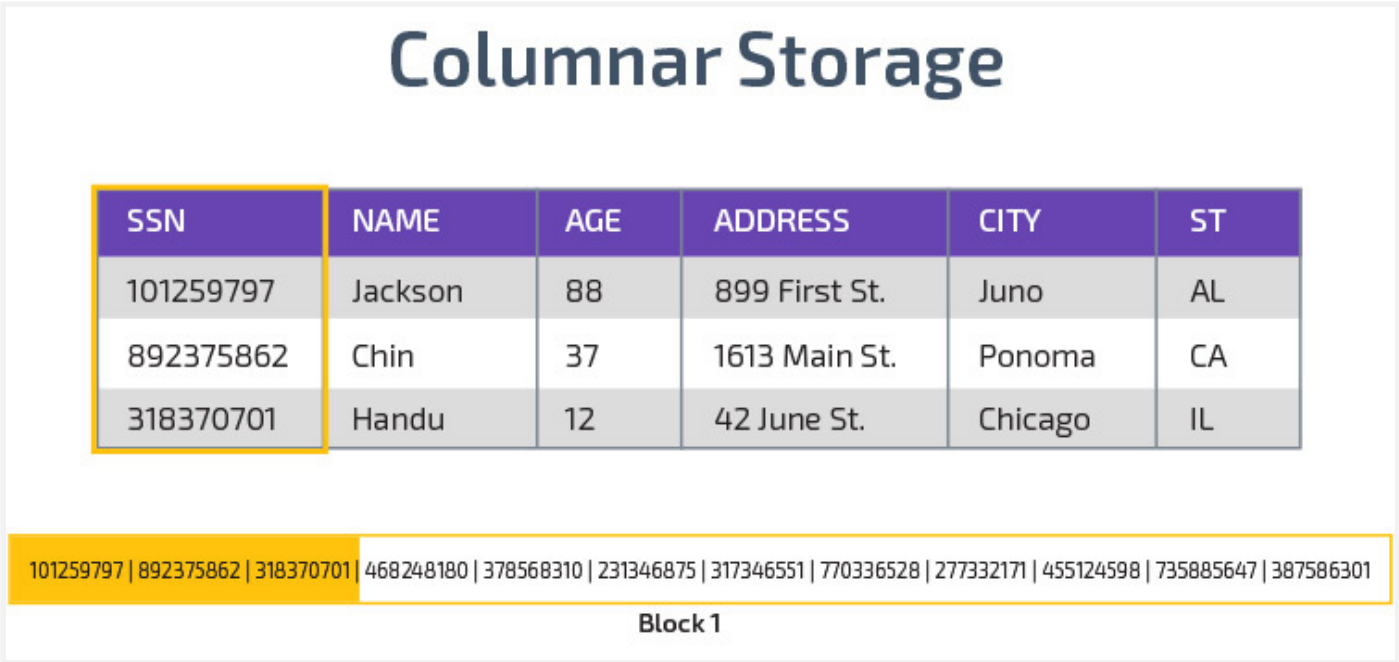
Partitions/Slices

Each compute node is partitioned into slices. A slice receives an allocation of memory and disk space on the node. Slices operate in parallel to speed up query execution time.

Columnar Storage

Redshift uses columnar storage, enabling better analytic query performance. Instead of storing records in rows, it stores values from a single column for multiple rows. The following diagrams make this clearer:





Columnar storage makes it possible to read data faster , for analytical queries that span many columns in a data set. Columnar storage also takes up less disk space, because each block contains the same type of data, meaning it can be compressed into a specific format.

Compression

Compression reduces the size of stored data. In Redshift, because of the way data is stored, compression occurs at the column level. Redshift allows you to compress data manually when creating a table, or automatically using the COPY command.

Data Loading

You can use Redshift's COPY command to load large amounts of data into the data warehouse. The COPY command leverages Redshift's MPP architecture to read and load data in parallel from files on Amazon S3, from a DynamoDB table, or from text output from one or more remote hosts.

It is also possible to stream data into Redshift, using the Amazon Kinesis Firehose service.

Cloud Database Warehouse - Google BigQuery

The following concepts are used specifically in the Google BigQuery cloud data warehouse, but may be applicable to additional solutions in the future based on Google infrastructure.

Serverless Service

BigQuery uses a serverless architecture. With BigQuery, businesses don't need to manage physical server units to run their data warehouses. Instead, BigQuery dynamically manages the allocation of its computing resources. Enterprises using the service simply pay for data storage per gigabyte and queries per terabyte.

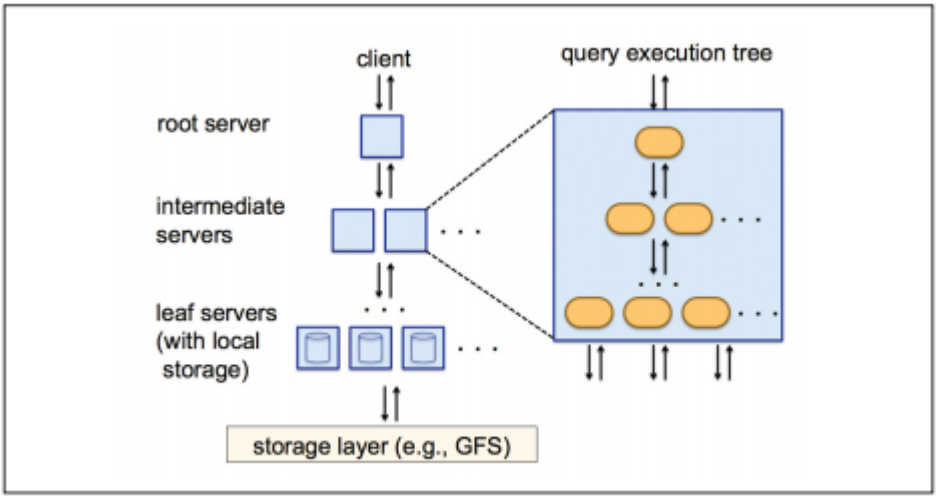
Colossus File System

BigQuery uses the latest version of Google's distributed file system, code-named Colossus. The Colossus file system uses columnar storage and compression algorithms to optimally store data for analytical purposes.

Dremel Execution Engine

The Dremel execution engine uses a columnar layout to query huge stores of data quickly. Dremel's execution engine can run ad-hoc queries on billions of rows in seconds because it uses massively parallel processing in the form of a tree architecture.

The tree architecture distributes queries among several intermediate servers from a root server. The intermediate servers push the query down to leaf servers (containing stored data), which scan the data in parallel. On the way back up the tree, query results are sent from each leaf server, and the intermediate servers perform a parallel aggregation of partial results.



Tree architecture of Dremel

Image source

This enables organizations to run queries on up to tens of thousands of servers simultaneously. According to Google, Dremel can scan 35 billion rows without an index in tens of seconds.

Data Sharing

Google BigQuery's serverless architecture allows enterprises to easily share data with other organizations without requiring those organizations to invest in their own storage.

Organizations that want to query shared data can do so, and they'll only pay for the queries. There is no need to create costly shared data silos, external to the organization's data infrastructure, and copy the data to those silos.

Streaming and Batch Ingestion

It is possible to load data to BigQuery from Google Cloud Storage, including CSV, JSON (newline-delimited), and Avro files, as well as Google Cloud Datastore backups. You can also load data directly from a readable data source.

BigQuery also offers a Streaming API to load data into the system at a speed of millions of rows per second without performing a load. The data is available for analysis almost immediately.

Cloud Data Warehouse Concepts - Panoply

Panoply is a smart data warehouse that doesn't require any ETL. Panoply delivers rapid data insights by using AI technology to eliminate the development and coding associated with transforming, integrating, and managing big data.

Below are some of the main concepts in the Panoply data warehouse related to data modeling and data protection.

Primary Keys

Primary keys ensure that all rows in your tables are unique. Each table has one or more primary keys that define what represents a single unique row in the database. All APIs have a default primary key for tables.

Incremental Keys

An incremental key is used to control attributes for incrementally loading data to the data warehouse from sources rather than reloading the entire dataset each time something changes. This feature is helpful for larger datasets which can take a long time to read mostly unchanged data. The incremental key indicates the last update point for the rows in that data source.

Nested Data

Nested data is not fully compatible with BI suites and standard SQL queries—Panoply deals with nested data by using a strongly relational model that doesn't permit nested values. Panoply transforms nested data in these ways:

- **Subtables**: By default, Panoply transforms nested data into a set of many-to-many or one-to-many relationship tables, which are flat relational tables.
- **Flattening**: With this mode enabled, Panoply flattens the nested structure onto the record that contains it.

History Tables

Sometimes you need to analyze data by keeping track of changing data over time to see exactly how the data changes (for example, people's addresses).

To perform such analyses, Panoply uses History Tables, which are time-series tables that contain historic snapshots of every row in the original static table. You can then perform straightforward querying of the original table or revisions to the table by rewinding back to any point in time.

Transformations

Panoply uses ELT, which is a variation on the original ETL data integration process. By transforming data immediately after the data is first injected from the source into your data warehouse, you get real-time data analysis with optimal performance.

String Formats

Panoply parses string formats and handles them as if they were nested objects in the original data. Supported string formats are CSV, TSV, JSON, JSON-Line, Ruby object format, URL query strings, and web distribution logs.

Data Protection

Panoply is built on top of AWS, so it the latest security patches and encryption capabilities provided by AWS, including hardware accelerated RSA encryption and Amazon Redshift's specific set of security features.

Extra protection comes from columnar encryption, which lets you use your own private keys that are not stored on Panoply's servers.

Access Control

Panoply uses two-step verification to prevent unauthorized access, and a permission system lets you restrict access to specific tables, views, or columns. Anomaly detection identifies queries coming from new computers or a different country, letting you block those queries unless they receive manual approval.

IP Whitelisting

It's recommended to block connections from unrecognized sources by using a firewall or an AWS security group, and whitelist the range of IP addresses that Panoply's data sources always use when accessing your database.

Conclusion: Traditional vs. Data Warehouse Concepts in Brief

To wrap up, we'll summarize the concepts introduced in this document.

Traditional Data Warehouse Concepts

- **Dimension**: Used to categorize and contextualize facts and measures, enabling analysis of and reporting on those measures.
- **Conceptual model**:This model defines the key data entities and the relationships between those high-level entities.
- **Logical model**: The logical model describes data relationships, entities, and attributes in as much detail as possible without considering how to physically implement a system that gathers this data.
- **Physical model**: The physical model is a representation of how to implement the data design in a specific database management system.
- **Inmon approach**: Bill Inmon's data warehouse approach defines the data warehouse as the centralized data repository for the entire enterprise. Data marts can be built from the data warehouse to serve the analytic needs of different departments.
- **Kimball approach**: Ralph Kimball describes a data warehouse as the merging of mission-critical data marts, which are first created to serve the analytic needs of different departments.
- **OLTP**: Online transaction processing systems facilitate fast, transaction-oriented processing with simple queries.
- **OLAP**: online analytical processing systems are designed to facilitate analysis of large volumes of historical transaction data using complex read-only queries and aggregations.
- **Star schema**: The star schema takes a fact table and splits its information into denormalized dimension tables.
- **Snowflake schema**:The snowflake schema splits the fact table into normalized dimension tables. Normalizing data reduces data redundancy issues and improves data integrity, but queries are more complex.

- **ETL**: Integrates data into the data warehouse by extracting it from different transactional sources, transforming the data to optimize it for analysis, and finally loading the transformed into the data warehouse.
- **ELT**: A variation on ETL that extracts raw data from an organization's data sources and loads the raw data into the data warehouse, where it can be transformed when it's needed for analytical purposes.
- **Data mart**: an archive of data focusing on a specific subject or department within an organization.
- **Enterprise data warehouse:**The EDW consolidates data from all subject areas related to the enterprise.

Cloud Data Warehouse Concepts - Amazon Redshift as Example

- **Cluster**: A group of shared computing resources based in the cloud.
- **Node**: A node is a computing resource contained within a cluster. Each node has its own CPU, RAM, and hard disk space.
- **Columnar storage**: This stores the values of a table in columns rather than rows, which optimizes the data for aggregated queries.
- **Compression**: Compression techniques reduce the size of stored data.
- **Data loading**: Getting data from sources into the cloud-based data warehouse. In Redshift, the COPY command or a data streaming service can be used.

Cloud Data Warehouse Concepts - BigQuery as Example

- **Serverless service**: In a serverless service, the cloud provider dynamically manages the allocation of machine resources based on the amount of resources the user consumes. The cloud provider hides server management and capacity planning decisions from the users of the service.

- **Colossus file system**: A distributed file system that uses columnar storage and data compression algorithms to optimize data for analysis.
- **Dremel execution engine**: This is a query engine that uses massively parallel processing and columnar storage to quickly execute queries.
- **Data sharing**: In a serverless service it is practical to query another organization's shared data without investing in data storage—you simply pay for the queries.
- **Streaming data**: Inserting data in real-time into the data warehouse without performing a load. Data can be streamed in batch requests, which are multiple API calls combined into one HTTP request.

Learn More about Data Warehouses

- Data Warehouse Architecture: Traditional vs. Cloud
- Database vs. Data Warehouse
- Data Mart vs. Data Warehouse
- Amazon Redshift Architecture

Built by Panoply