Business Intelligence

Lecture 3

Lecture Agenda

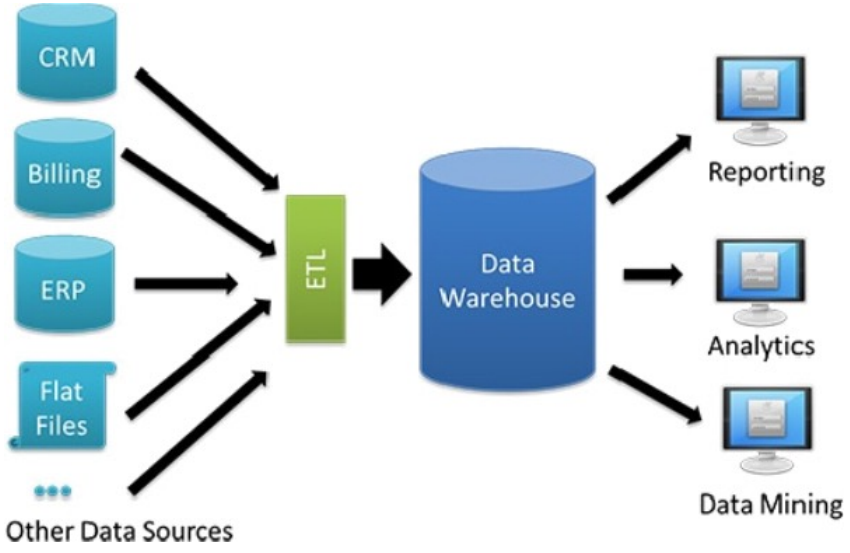
- Data Warehouse
- Multidimensional analysis
- Schemas: Star, Snowflake, Galaxy
- ETL
- ELT
- Batch Processing

Data Warehouse and Multidimensional Analysis

What is Data Warehouse?

"A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."

Data warehousing: The process of constructing and using data warehouses



Data Warehouse

Subject Oriented

- Organized around major subjects, such as customer, product, sales.
- Focusing on the modeling and analysis of data for decision makers.

Time Variant

- Operational database: current value data
- Data warehouse data: provide information from a historical perspective

(e.g., past 5-10 years)

Data Warehouse

Integrated

- Constructed by integrating multiple, heterogeneous data sources:

  relational databases, flat files, on-line transaction records

- Data cleaning and data integration techniques are applied.
- Ensure **consistency** in naming conventions, encoding structures, attribute measures, etc.

Non-Volatile

- A physically separate store of data transformed from the operational environment.
- Operational update of data does not occur in the data warehouse environment.

Why Separate Data Warehouse?

Different functions and different data:

- missing data: Decision support requires historical data which operational DBs do not typically maintain.
- data consolidation: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
- data quality: different sources typically use inconsistent data representations

Data Warehouse Back-End Tools and Utilities

- **Data extraction**: get data from multiple, heterogeneous, and external sources
- **Data cleaning**: detect errors in the data and rectify them when possible
- **Data transformation**: convert data from legacy or host format to warehouse format
- **Load**: sort, summarize, consolidate, compute views, check integrity, and build indices and partitions
- **Refresh**: propagate the updates from the data sources to the warehouse

Data Warehouse Process Architecture

Centralized Process Architecture - the data is collected into single centralized storage and processed upon completion by a single machine with a huge structure in terms of memory, processor, and storage.

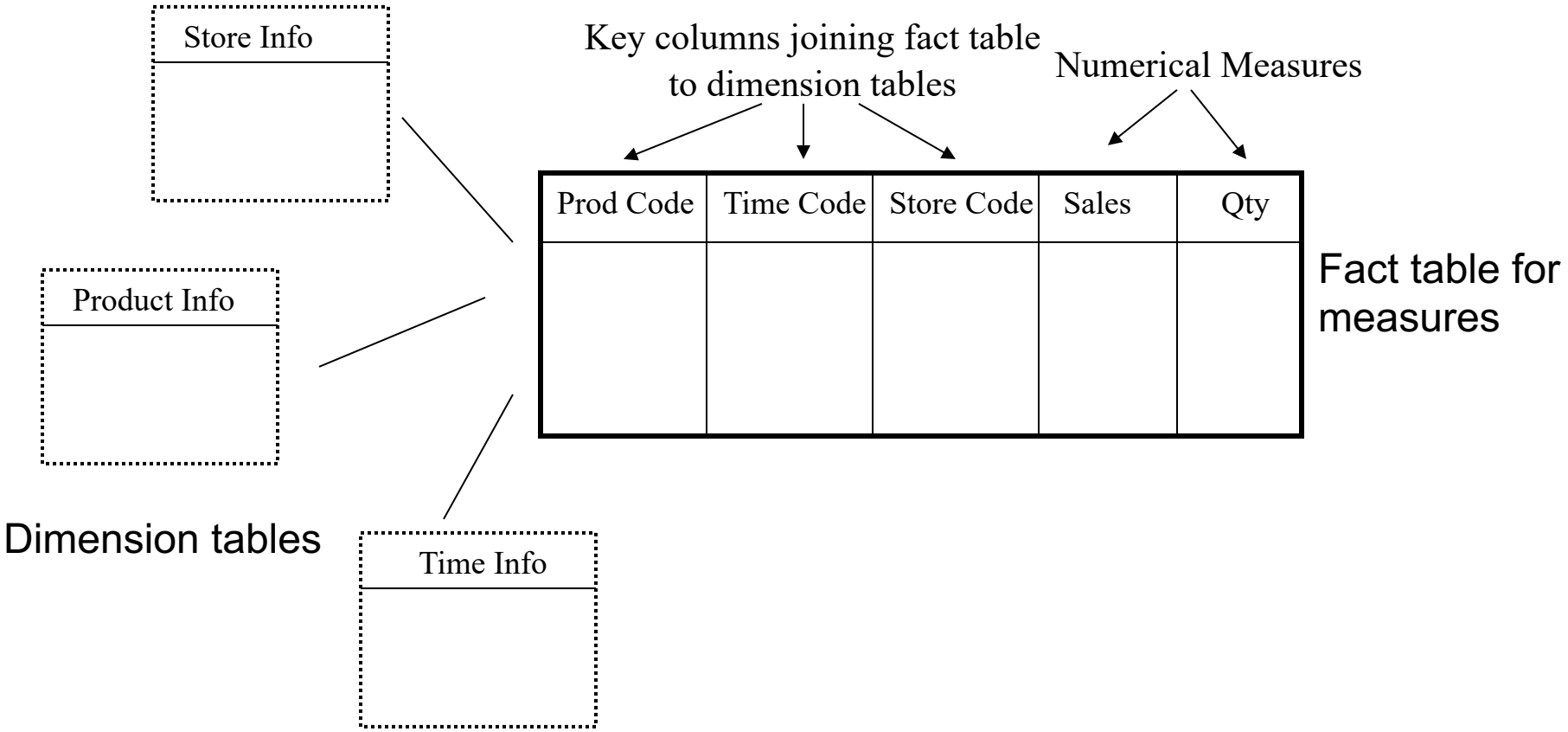
Distributed Process Architecture - information and its processing are allocated across data centers, and its processing is distributed across data centers, and processing of data is localized with the group of the results into centralized storage.

Multidimensional Modeling

A technique for structuring data around the business concepts.

- *ER* models describe **entities** and **relationships**
- *Multi-dimensional* models describe **measures** and **dimensions**
  - **Measures** - numerical data being tracked in business, can be analyzed and examined
  - **Dimensions** - business parameters that define a transaction

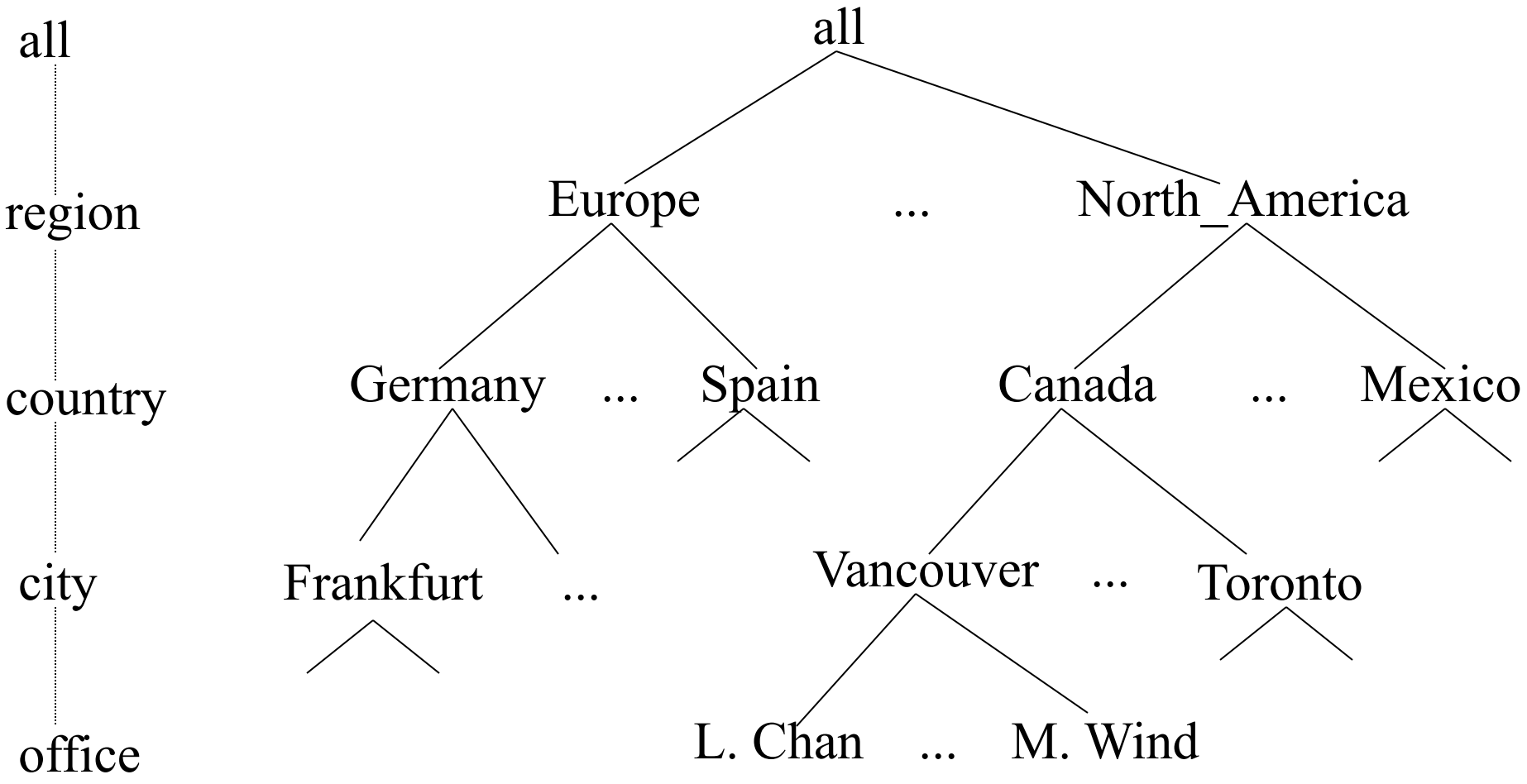
The Multi-Dimensional Model



Dimensional Modeling

- Dimensions are organized into hierarchies
  - E.g., Time dimension: days $\rightarrow$ weeks $\rightarrow$ quarters
  - E.g., Product dimension: product $\rightarrow$ product line $\rightarrow$ brand
- Dimensions have attributes
  - E.g., Time dimension: date, month, year
  - E.g., Store dimension: id, city, state, country

A Concept Hierarchy: Dimension (location)

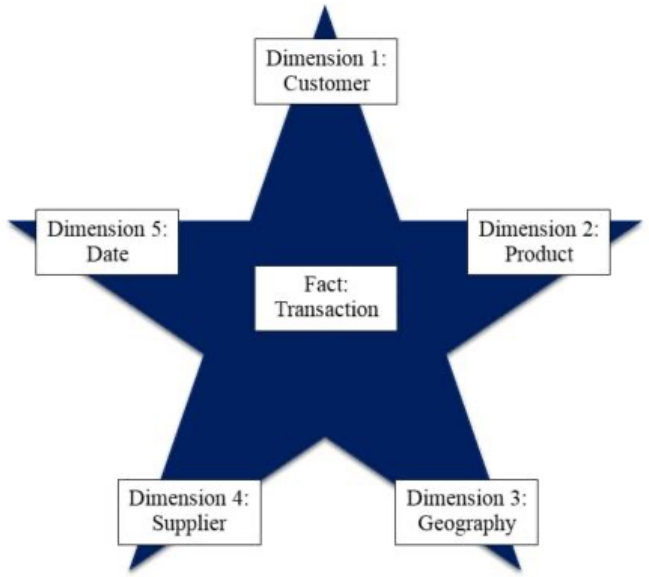


Data Warehouse Modeling

The “Classic” Star Schema

A relational model with a **one-to-many** relationship between **dimension table** and **fact table**.

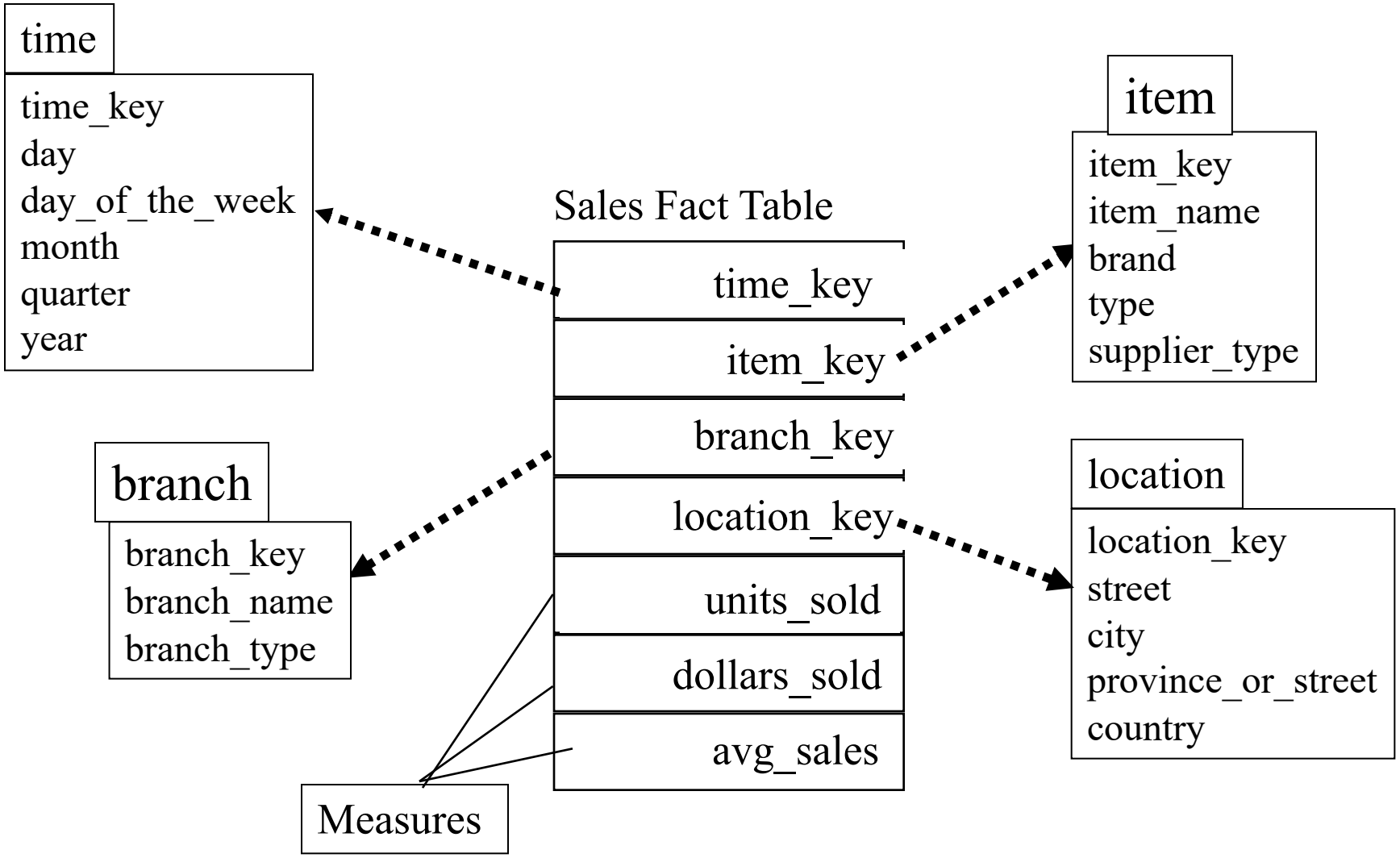
- A single **fact table**, with detail and summary data
- **Fact table primary key has only one key column per dimension**
- Each dimension is a single table, highly denormalized



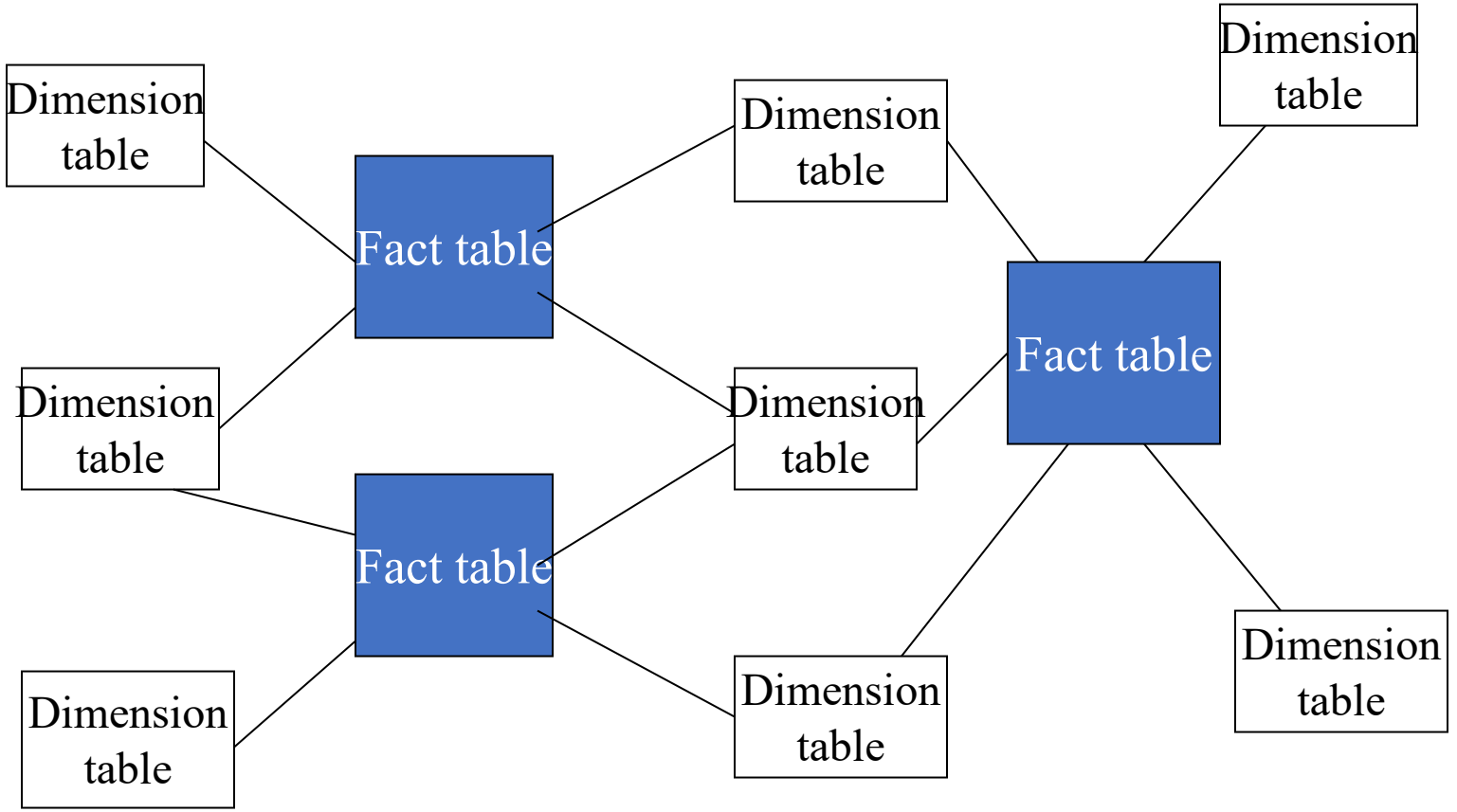
The “Classic” Star Schema

- **Benefits**
  - Easy to understand
  - Intuitive mapping between the business entities
  - Easy to define hierarchies
  - Reduces number of joins
- **Drawbacks**
  - Summary data in the fact table yields poorer performance for summary level
  - Huge dimension tables a problem

Star Schema


time
time_key
day
day_of_the_week
month
quarter
year
item
item_key
item_name
brand
type
supplier_type
Sales Fact Table
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales
branch
branch_key
branch_name
branch_type
location
location_key
street
city
province_or_street
country
Measures

Families of Stars

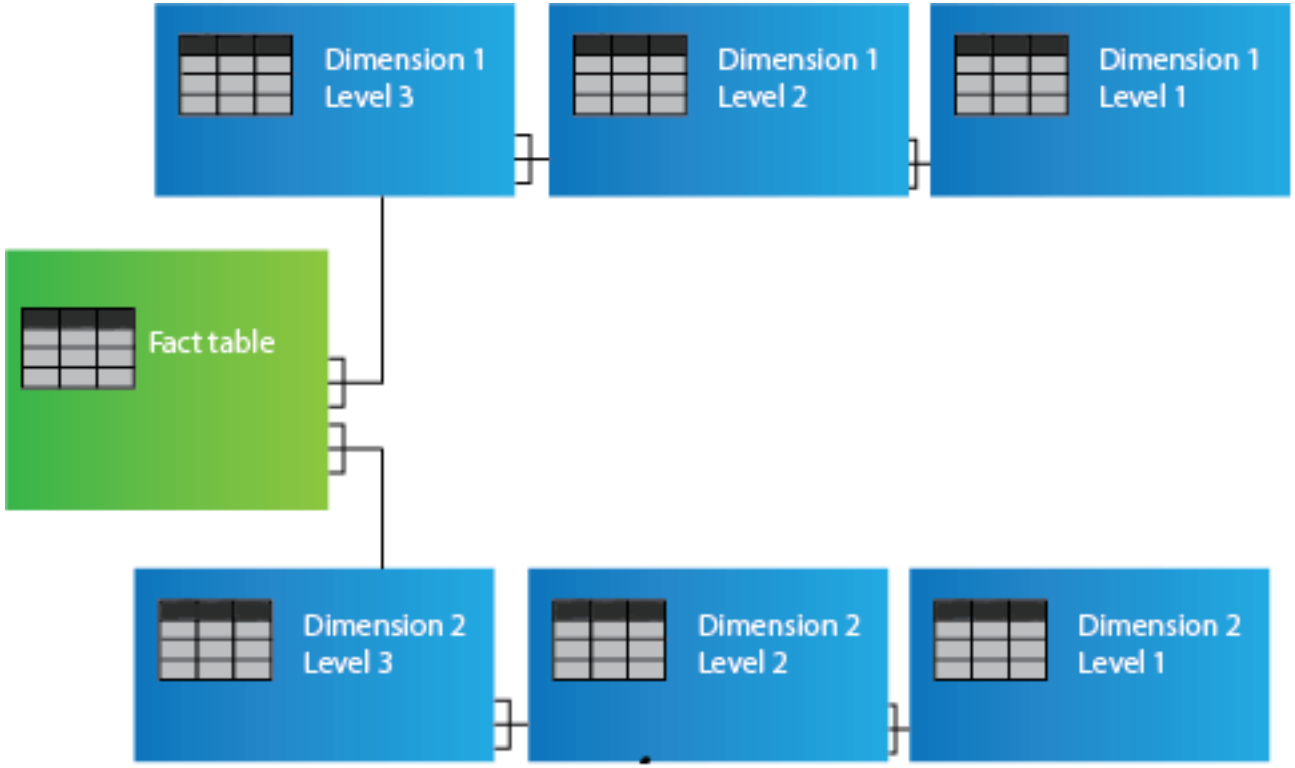


The Snowflake Schema

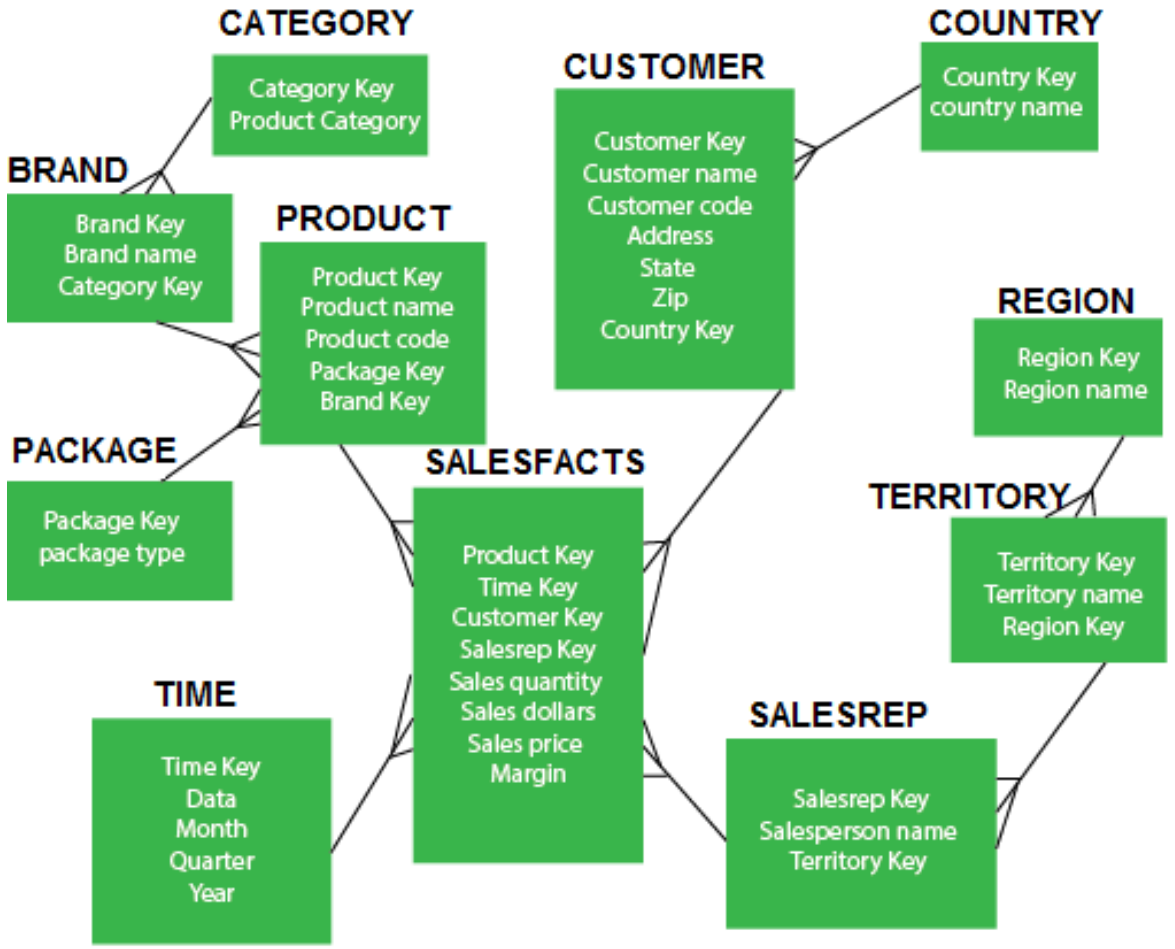
"A schema is known as a snowflake if one or more-dimension tables do not connect directly to the fact table but must join through other dimension tables."

- A snowflake schema is suitable for many to many and one to many relationships between dimension levels.
- "Snowflaking" is a method of normalizing the dimension tables in a star schema.
- The result is more complex queries and reduced query performance.

Snowflake Schema

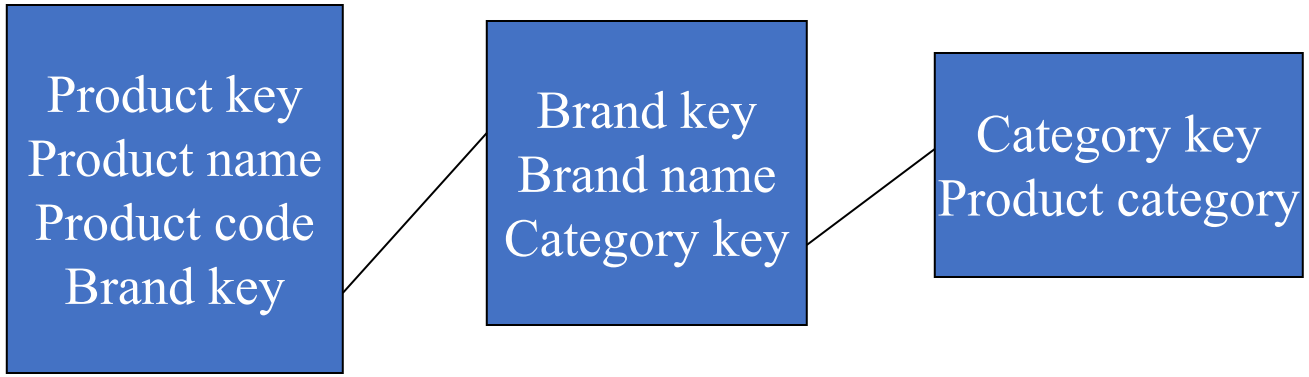


Example of Snowflake Schema



Snowflaking

The attributes with low cardinality in each original dimension table are removed to form separate tables. These new tables are linked back to the original dimension table through artificial keys.



Snowflake Schema

Advantages

- Small saving in storage space
- Normalized structures are easier to update and maintain

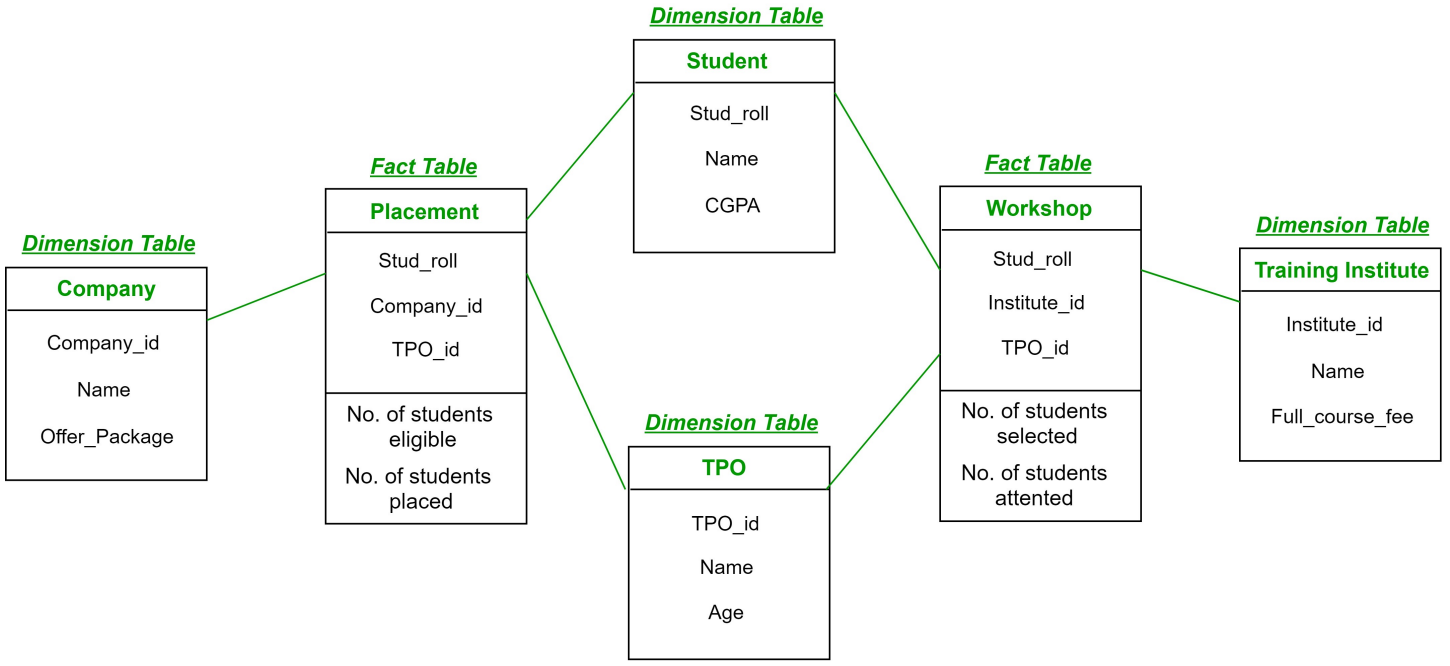
Disadvantages

- Schema less intuitive
- Difficult to browse through the contents

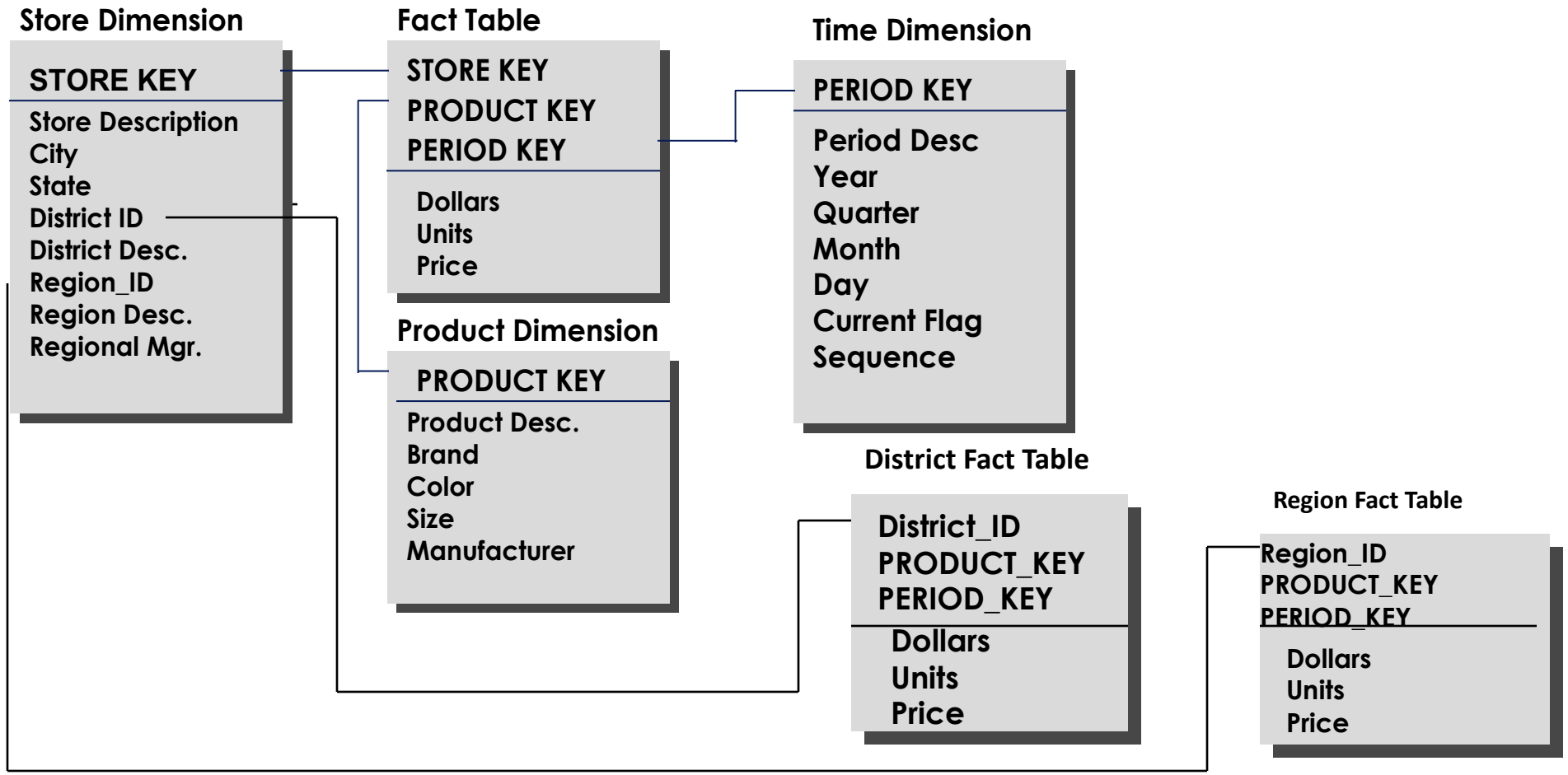
The “Fact Constellation” Schema - Galaxy

Galaxy schema - Two or more fact tables sharing one or more dimensions.

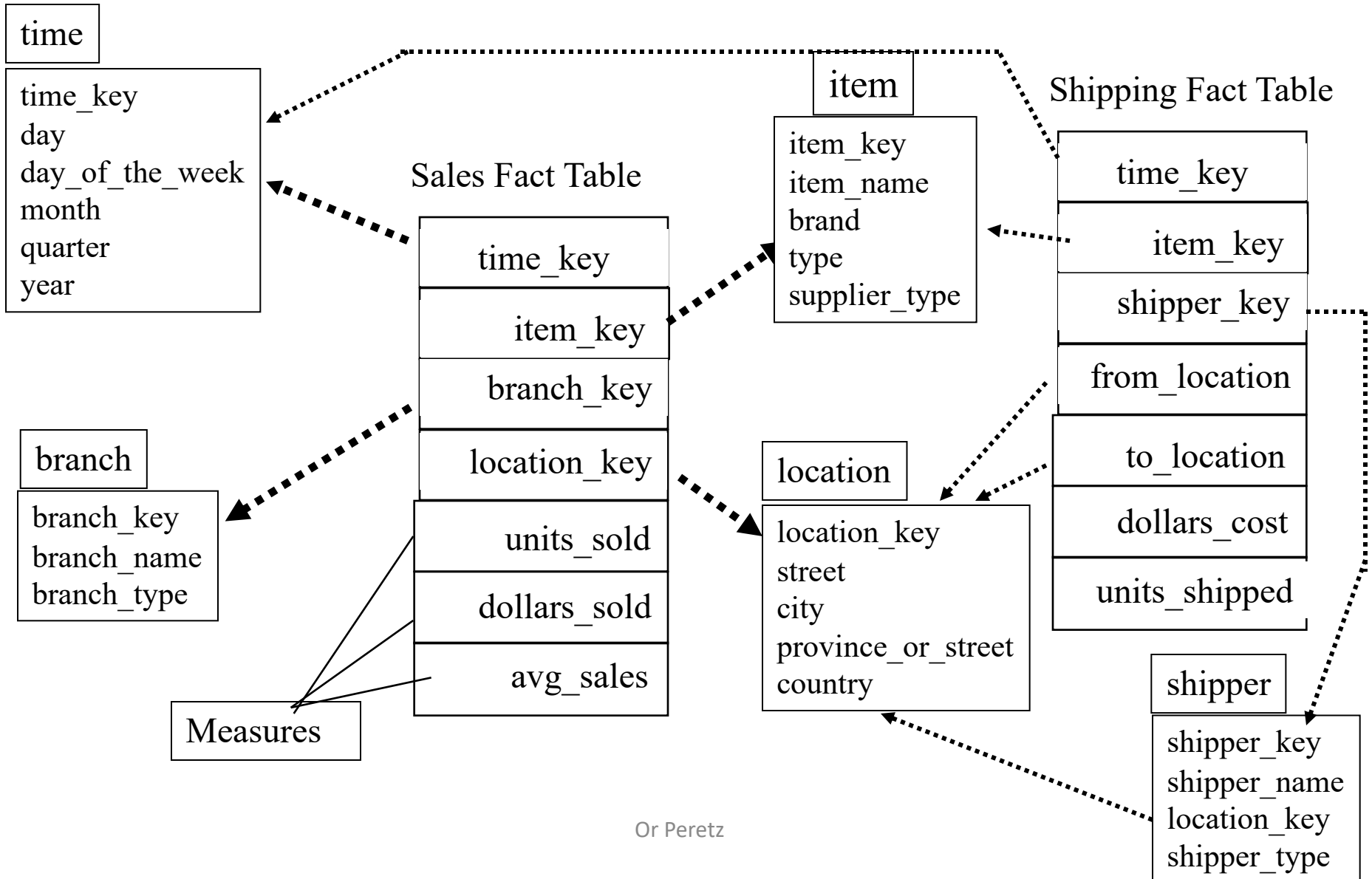
Fact Constellation Schema describes a logical structure of data warehouse or data mart.



The “Fact Constellation” Schema



Example of Fact Constellation



What is the Best Design?

- Performance benchmarking can be used to determine what is the best design.
- Snowflake schema easier to maintain dimension tables when dimension tables are very large (reduce overall space). It is not generally recommended in a data warehouse environment.
- Star schema more effective for data cube browsing (less joins): can affect performance.

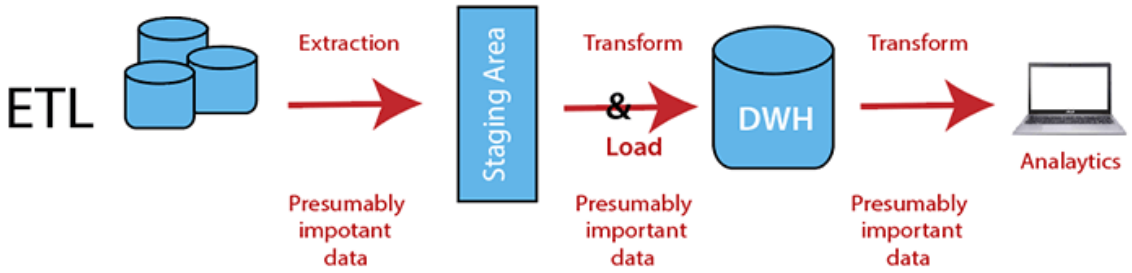
ETL vs ELT

Extract, Transform, Load - ETL

ETL process uses the pipelining concept. In this concept, as soon as the data is extracted, it can be transformed, and during the period of transformation, new data can be obtained. And when the modified data is being loaded into the data warehouse, the already extracted data can be transformed.

Extract, Transform, Load

Extract, Transform, and load is a process which involves extracting data from outside sources and transforming it to fit operational needs, then loading it into the target database or data warehouse. To use this approach is reasonable when we are using different database for our data warehouse.



Extraction

Extraction - The main goal of extraction is to collect the data from the source system as fast as possible and less convenient for these source systems. It also states that the most applicable extraction method should be chosen for source date/time stamps, database log tables, hybrid depending on the situation.

Transformation

Transformation is the second step of the ETL process, where all the collected data has been transformed into the same format. It may involve the following tasks:

- **Filtering:** Only specific attributes are loading into the data warehouse.
- **Cleaning:** Filling up the null values with specific default values.
- **Joining:** Join the multiple attributes into the one.
- **Splitting:** Splitting the single attribute into multiple attributes.
- **Sorting:** Sort the tuples based on the attributes.

Loading

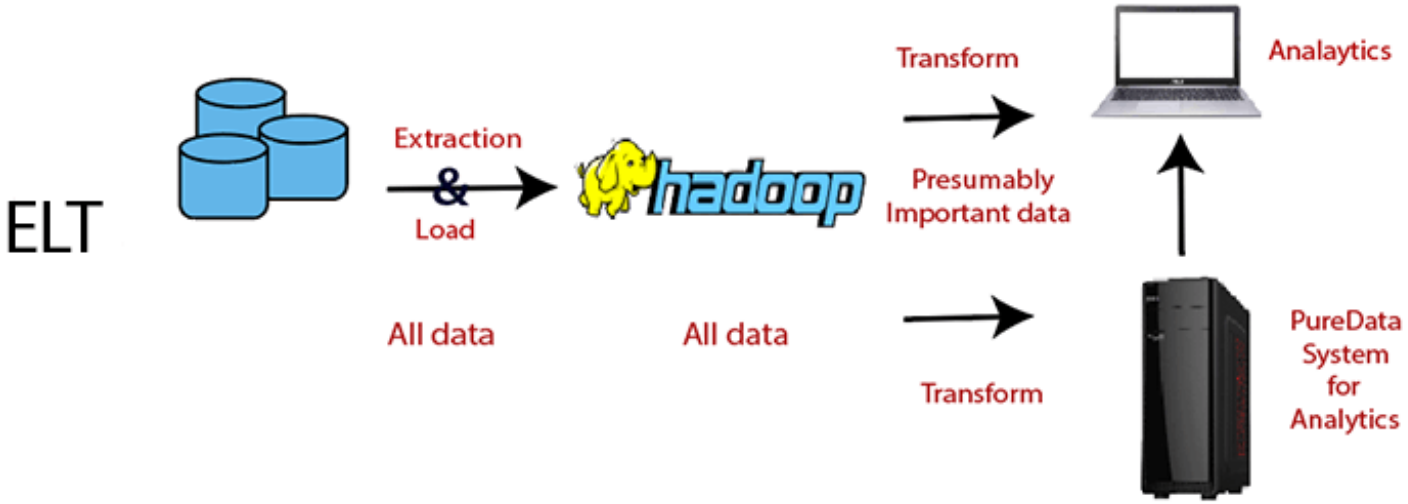
Loading is the final step of the ETL process.

The big chunk of data is collected from various sources, transformed them,

and finally loaded to the data warehouse.

Extract, Load, Transform - ELT

In the ELT approach, after extracting the data, we immediately start loading phase, moving all the data sources into a single, centralized data repository. With today's infrastructure technologies are using the cloud, and systems can now support large storage and scalable compute.

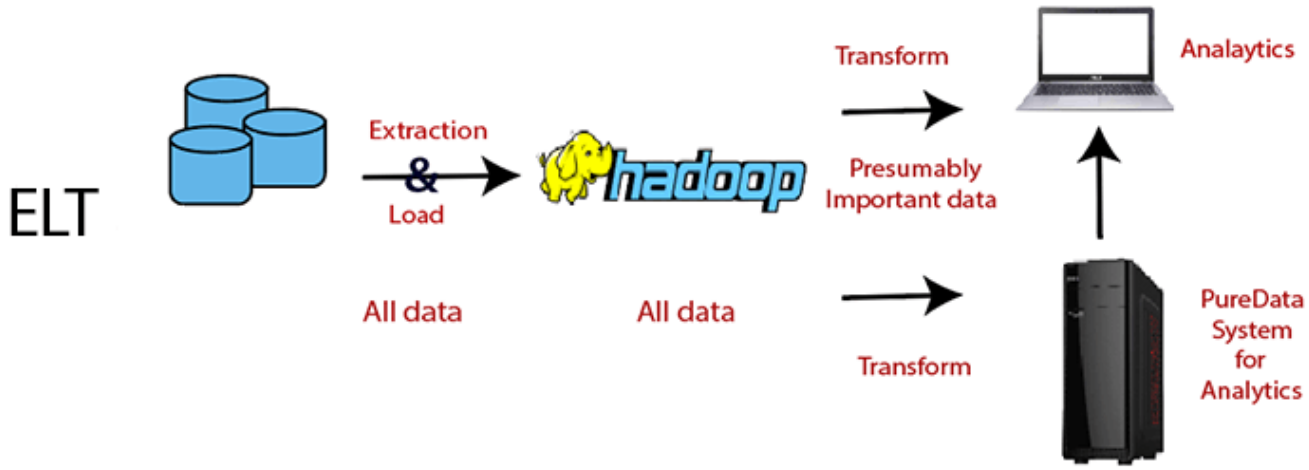


Extract, Load, Transform - ELT

The ELT approach provides a modern alternative to ETL,

but there are cases when we need to use ELT:

- When the volume of data is high.
- When the source database and target database both are same.



Working of ELT

- **Extract:** Extract the data from different data source which work similar in both data management approaches.
- **Load:** ELT delivers the whole data to the site where it will live. ELT shortens the cycle between the extraction and delivery, but there is a lot of work which should be done before the data becomes useful.
- **Transform:** Here, data warehouse and database sorts and normalize the data. The overhead for storing this data is high, but it comes with more opportunities.

Differences between ETL and ELT

Parameter	ETL	ELT
Process	Data is transferred in staging server and then moved to Data Warehouse Database.	Data remains in the DB of the data warehouse.
Code Usage	Small amount of data	Huge data
Transformation-Time	Needs time for transformation completion.	Speed never depends on the size of the data

Differences between ETL and ELT

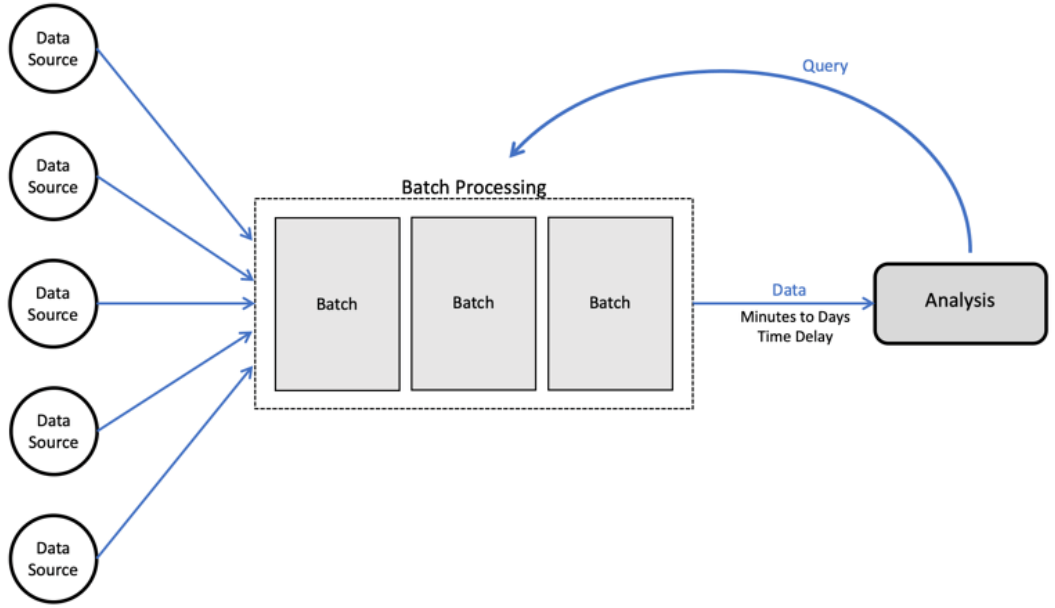
Parameter	ETL	ELT
Maintenance-Time	High maintenance - select the data to load and transform	Low maintenance - the data is available all the time
Implementation Complexity	Easier to implement it at an early stage	To implement the ELT process organization should have in-depth knowledge of expert skills and tools
Cost	High for small and medium business	Low entry costs using SaaS

Differences between ETL and ELT

Parameter	ETL	ELT
Complexity	Loads only the essential data, which is identified at design time	Online backward and loading only relevant data
LookUps	Dimension and facts need to be available in the staging area	All data will be available because extract and load occur in one single action
Calculations	Need to override the existing column by update values	Easy to add the column to the existing table
Unstructured Data	Supports relational data	Supports unstructured data

Building an ETL Pipeline – Batch Processing

Transfer and process the data in batches from the source database to the data warehouse. It is challenging to develop an enterprise ETL Pipeline.



Building an ETL Pipeline – Batch Processing

Step 1: Reference Data

Here, we will create a set of data that defines the set of permissible values and may contain the data.

Example**:** In a country data field, we can define the country codes which are allowed.

Step 2: Extract from Data Reference

Most of the ETL systems combine the data from multiple source systems, including relational database, non-relational database, XML, JSON, CSV files and after successful extraction.

Data is converted into a single format for standardizing the format.

Building an ETL Pipeline – Batch Processing

Step 3: Data Validation

An automated process confirms whether the pulled data from sources have expected values. Example:

- A data field should contain valid dates within the past 12 months in a database of financial transaction from the past year.
- The validation engine **rejects** the data if it fails the validation rules.
- We analyze the rejected records, on a regular basis, to identify what went wrong.
- Here we correct the source data or modify extracted data to resolve the problem in the next batches.

Building an ETL Pipeline – Batch Processing

Step 4: Transform Data

- Removing extraneous or erroneous data
- Applying business rules
- Checking data integrity (ensuring that data was not corrupted by ETL)
- We need to program and test a series of rules or functions that can achieve the required transformation and run them on the extracted data.

Example: if we analyze revenue, we can summarize the dollar amount of invoices into a daily or monthly total.

Building an ETL Pipeline – Batch Processing

Step 5: Stage

- Data should be entered first into a staging database
- Making it easier to roll back if something goes wrong.

Step 6: Publish to Data Warehouse

- Loading the data to the target tables.
- Some data warehouse overwrites existing information every time, the ETL pipeline loads a new batch daily, monthly or weekly.