

**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

Lecture 01

Introduction to DW & BI

# Content

- Module overview
  - Course content
  - Learning outcomes
  - Assessment criteria
  - Prerequisites
- Introduction to DW & BI
- OLTP vs DW (OLAP)
- Implementation steps

# Module Overview



# Course Content

- Introduction to DW/BI
- OLTP vs DW (OLAP)
- Data Warehousing Architectures
- Data Warehouse Designs & Concepts
  - Dimensional Data Modelling
- ETL/ELT Process (Data Ingestion Flows)
- OLAP Cubes and Related Concepts
- Business Intelligence
- Testing & Tuning
- Emerging Trends in Data Engineering

# Learning Outcomes

- **LO 01:** Describe the role of BW & BI in today's marketplace.
- **LO 02:** Develop familiarity with the data warehouse modelling concepts, and various technologies and tools required to implement a data warehouse.
- **LO 03:** Apply existing methods and tools for extracting, transforming, and loading data.
- **LO 04:** Design and implement BI solutions for real world problems.
- **LO 05:** Design and implement testing and tuning processes for DW & BI solutions.

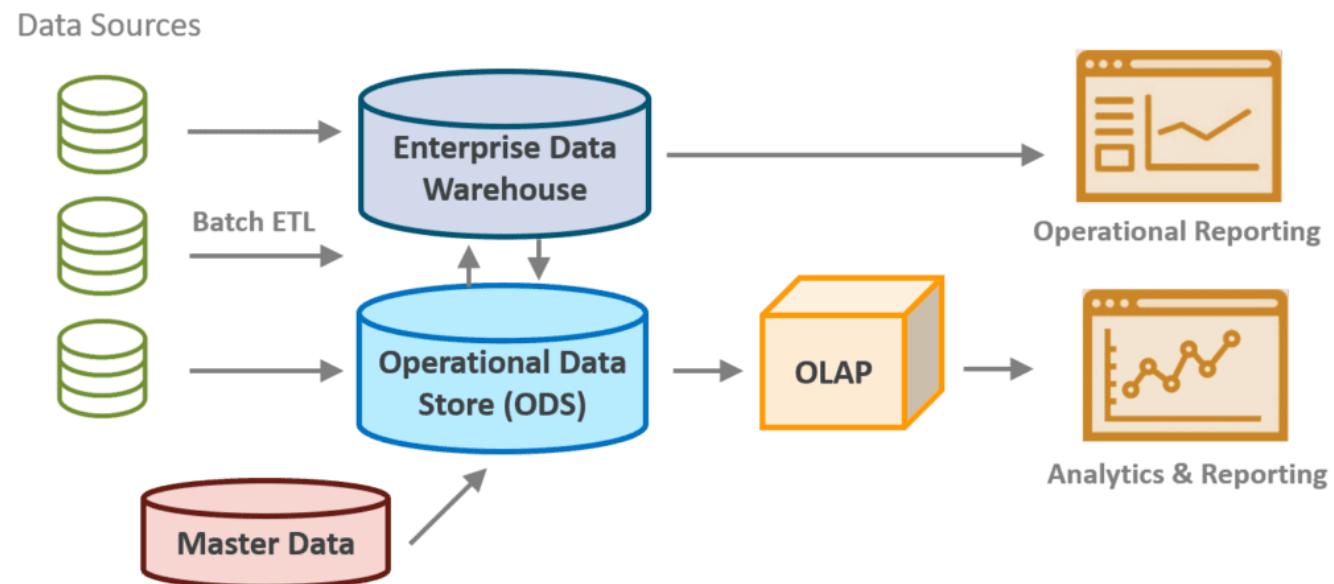
# Assessment Criteria

- Continues assessment
  - Practical Assignment 1: 20% (**L01 – L03**)
  - Practical Assignment 2: 20% (**L04 – L05**)
- End semester assessment
  - Final Examination - 60% (**L01 – L05**)

# Prerequisites

- Basic concepts of database systems
- Database design process and E-R model (ERD)
- Normalization/De-normalization
- Relational database design
- Structured Query Language (SQL)

# Introduction to DW & BI



# Data

- One of the most important assets of any organization
- Purposes of data:
  - Operational record keeping (OLTP)
  - Analytical decision making (OLAP)
- Why different solutions for multiple purposes?
  - Different types of users
  - Different requirements
  - Performance reasons

# What is DW & BI?

- Data Warehousing (DW)
  - It is a set of processes, architectures and technologies for collecting and managing data from various sources to support deriving meaningful business insights from raw data
  - Data collection involves data gathering, transforming and storing
  - It also includes database creation and data integration process development along with 'data profiling' and business validation rules
  - High level tasks include data acquisition, metadata management, data cleansing, data transformation, data distribution and data recovery/backup planning

# What is DW & BI?

- Business Intelligence (BI)
  - It is a set of processes, architectures, and technologies for converting raw data into meaningful information and knowledge that supports profitable business actions
  - BI helps finding insights which portray business's current picture (as-is) and historical picture (as-was)
  - Enable interested parties (end-users and down stream systems) to consume organization's data by providing access to a consolidated data store (DW)
  - Deals with OLAP, data visualization, and data mining and query/reporting tools

# History of Data Warehousing

- **1960** - Dartmouth and General Mills in a joint research project, develop the terms dimensions and facts
- **1970** - Nielsen and IRI introduces dimensional data marts for retail sales
- **1983** - Tera Data Corporation introduces a database management system which is specifically designed for decision support
- **Late 1980s** - IBM worker Paul Murphy and Barry Devlin developed a Business Data Warehouse which is considered as the start of Data Warehousing
- **1990** - the real concept was given by **Inmon Bill** (father of data warehouse). He had written about a variety of topics for building, usage, and maintenance of the warehouse & the Corporate Information Factory

# Definition of Data Warehouse

- The term "Data Warehouse" was first coined by Bill Inmon in 1990
- According to Inmon:

‘a data warehouse is a **subject oriented, integrated, time-variant, and non-volatile** collection of data. This data helps analysts to take informed decisions in an organization’

# Characteristics of Data Warehouse

- **Subject-Oriented**
  - Offers information regarding a theme instead of organization's ongoing operations. These subjects can be sales, marketing, distributions, etc. Subjects contain their unique and also common set of entities.
  - Emphasis on modelling and analysis of data for decision making.
- **Integrated**
  - A data warehouse is developed by integrating data from varied sources like a mainframe, relational databases, flat files, etc. Moreover, it must keep consistent naming conventions, format, and coding.
  - Establishment of a common unit of measure for all similar data from dissimilar databases. The data also needs to be stored in the Data Warehouse in common and universally acceptable manner.
  - This integration helps in effective analysis of data. Consistency in naming conventions, attribute measures, encoding structure etc. have to be ensured.

# Characteristics of Data Warehouse

- **Time-Variant**
  - The time horizon for data warehouse is quite extensive compared with operational systems. The data collected in a data warehouse is recognized with a particular period and offers information from the historical point of view. It contains an element of time.
  - In general, once data is inserted in the warehouse, it can't be updated or changed.
- **Non-volatile**
  - Previous data is not erased when new data is entered in it.
  - Data is periodically refreshed. This also helps to analyse historical data and understand what & when happened. It does not require transaction process, recovery and concurrency control mechanisms.
  - Activities like delete, update, and insert which are performed in an operational application environment are omitted in Data Warehouse environment.

# Goals of DW & BI

- Make information accessible easily
  - Understandability
  - Obviousness
  - Users' vocabulary
  - Easy to use tools
- Present information consistently and timely
  - Credible data
  - Quality assured
  - Based on business requirements
  - Weekly, daily, hourly, near-real-time, etc.

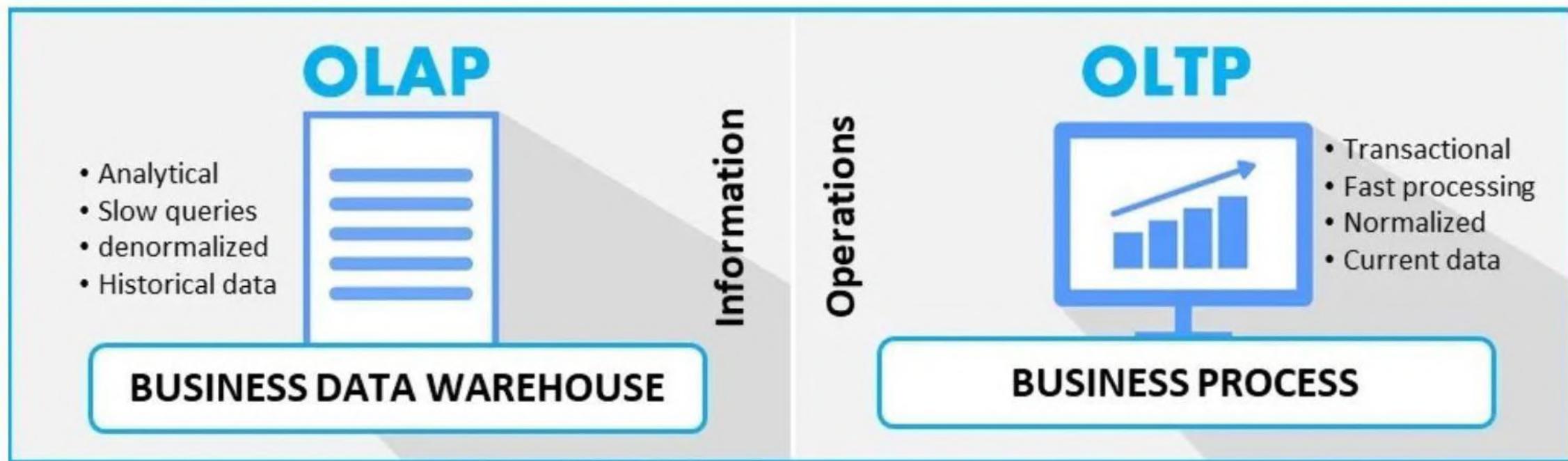
# Goals of DW & BI

- Provides security
  - Access control
  - Data masking
- Acceptability of the solution
  - Active users
- Adapt to change
  - Requirements
  - Business logics
  - Technology

# Who Needs Data Warehouses?

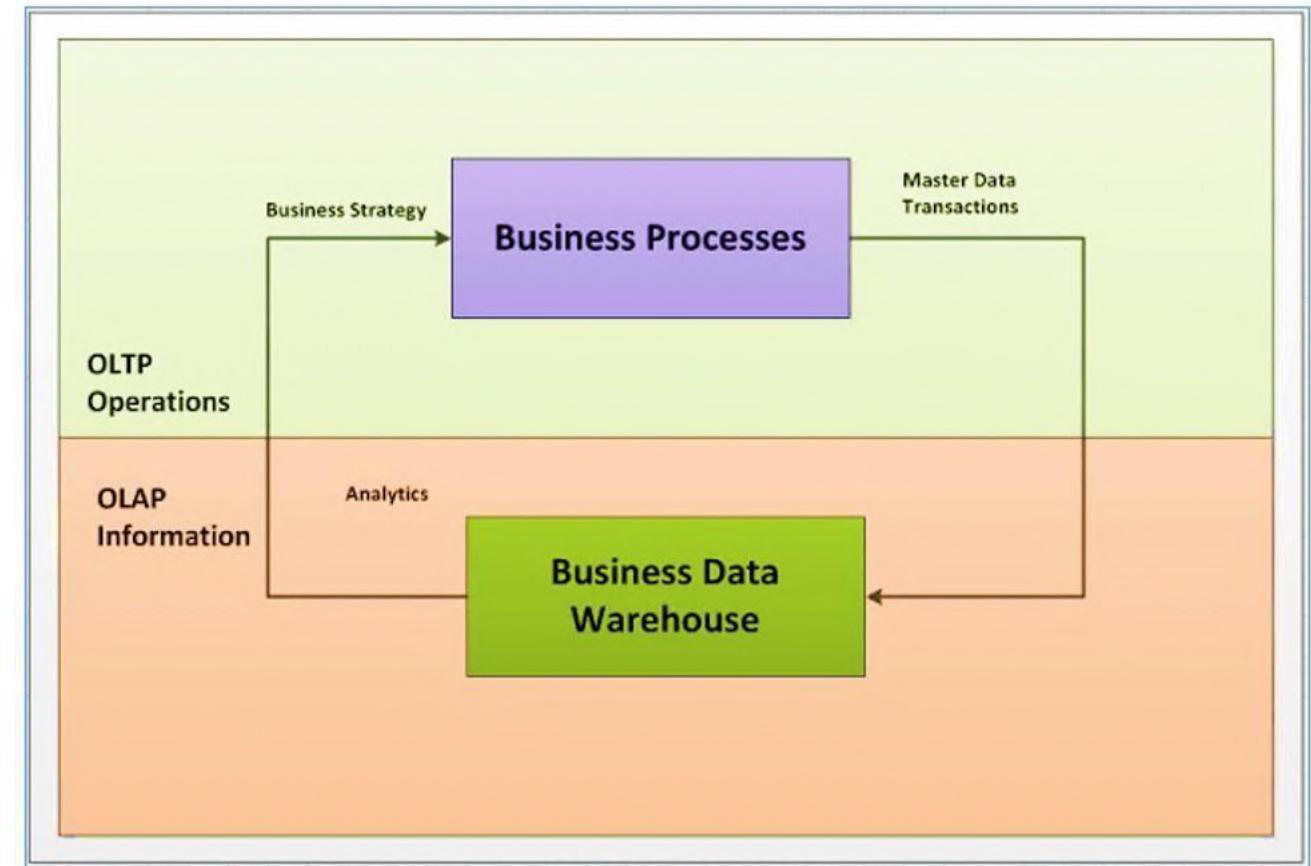
- People who wants a systematic approach for decisions making and relies on mass amount of data
- People who uses customized, complex processes to obtain information from multiple data sources
- People who relies on simple technologies to access the data
- People who wants fast performance on a huge amount of data, which is a necessity for visualization
- Who wants to discover ‘hidden patterns’ of data-flows and groupings

# OLTP vs. DW (OLAP)



# OLTP vs. DW (OLAP) Comparison

- **OLTP:** primary objective is data processing and not data analysis
- **OLAP:** primary objective is data analysis and not data processing



# OLTP vs. DW (OLAP) Comparison

Parameter	OLTP	DW (OLAP)
Design	Application oriented	Subject oriented
Purpose	Operational and real-time/Transactional	Analytical
Data processing	Optimised for mostly updates and for required reads to support the operation	Optimised for reads. Rarely writes
Volume of data	Data required to support current operations of business	Vast amount of data to support historical analysis. Includes historical snapshots
Level of data	Elemental data required for day-to-day transactions	Raw data preserving history and summarised data
Data Integrity	Maintains PKs	Business PKs are not must. Integrity managed using different mechanisms (SK)

# OLTP vs. DW (OLAP) Comparison

Parameter	OLTP	DW (OLAP)
Data model	Normalized relational model	De-normalized dimensional model and multidimensional views

**Normalized**

Normalized – Data is broken into multiple tables

Product		Color		Product-Color	
ProductID	Desc	ColorID	Desc	ProductID	ColorID
1	Mtn Bike #778	1	Red	1	1
2	Road Bike #123	2	Black	1	2
3	Touring Bike #222	3	Silver	2	1
		4	Mauve	2	2
				3	3
				3	1
				3	3
				3	4

**Denormalized**

Denormalized – Data combined

Product (denormalized)				
ProductSK	ProductID	ColorID	Desc	Color
1	1	1	Mtn Bike #778	Red
2	1	1	Mtn Bike #778	Black
3	2	1	Road Bike #123	Red
4	2	2	Road Bike #123	Black
5	2	2	Road Bike #123	Silver
6	3	1	Touring Bike #222	Red
7	3	3	Touring Bike #222	Silver
8	3	4	Touring Bike #222	Mauve

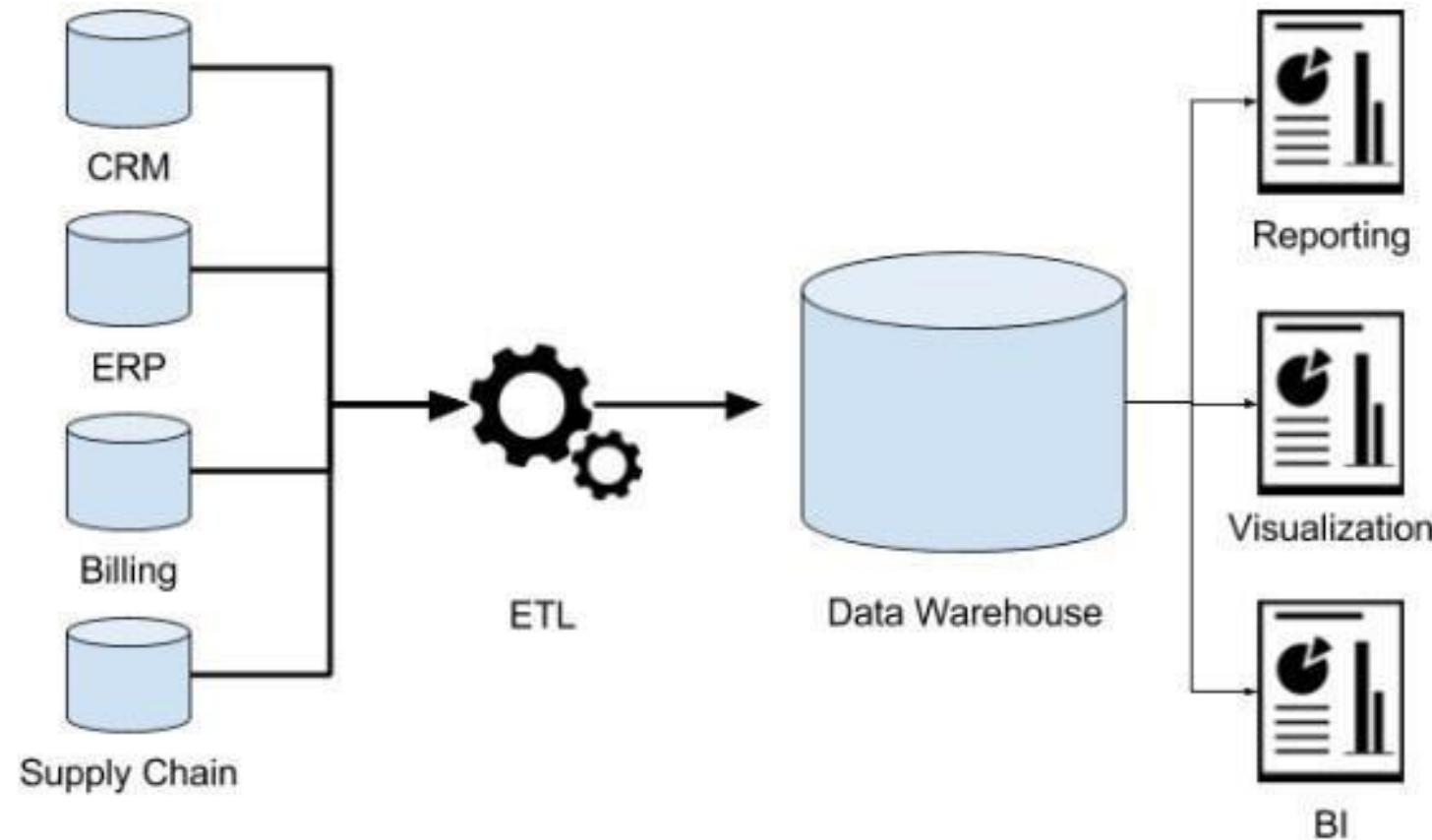
# OLTP vs. DW (OLAP) Comparison

Parameter	OLTP	DW (OLAP)
Response times	Miliseconds	Seconds to minutes
Usefulness	To support and control intended business operations	To support business decisions
User audience	People who runs the business operations	Decision makers and analysts
Number of users	High. Anyone who is involved in business operations	Low. Executive level decision makers, top management, analysts
Back-up	Regular backups.	Not a must compared to OLTP. But time-to-time backups are taken
Tools	Traditional DBMS	DBMS for data warehouses (sometimes specialized appliances), OLAP engines, ETL tools, BI tools
Performance metric	Transactional throughput	Query throughput

# Why Not OLTP for Analytics?

- Frequent updates
  - Great deal of locking
- Highly normalized data
  - Many table joins
- Too complex to support ad-hoc queries
  - Many tables to work with
- Slowness & impact on the transaction system

# High-Level Architecture



# Implementation Steps

Step	Tasks	Deliverable
1	Determine business objective and define scope	Scope definition
2	Collect and analyze requirements (business & technical), and identification of required architectural components	Architectural documents
3	Analyse source systems to understand data (this will help us understanding data quality requirements too!)	Data profile and analysis report
4	Data model building for data layer components (data warehouse, staging databases, operational data store, semantic layer including OLAP cubes)	Conceptual models Logical models <b>*discussed in lecture 02</b>
5	Identify required tools and technologies to implement the solution	Implementation plan
6	Detailed data model for the data warheouse and other data layer components	Physical model <b>*discussed in lecture 02</b>
7	Install/configure necessary tools and softwear (this could be cloud based tools or services!)	Documented implementation details and readied environment

# Implementation Steps

Step	Tasks	Deliverable
8	Implement the data warehouse and other data layer components	Physically developed data models
9	Design and develop ETL process flow in each layer as applicable	Developed, ready to deploy set of process flows (ETLs)
10	Design and develop OLAP layer	Developed, ready to deploy set of OLAP layer
11	Design and develop BI layer components (required visualizations: reports/dashboards, self-service BI platform, front end BI application, etc.)	Developed, ready to deploy set of BI components
12	Initial data loading from sources to data layer	Components with data ingested
13	Process scheduling automation	Automated processes
14	Monitor, tuning, and enhancements	

Please note, testing is not mentioned in above tasks list, and should take place in most of the steps as required!

**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

Lecture 02

Architecture & Components

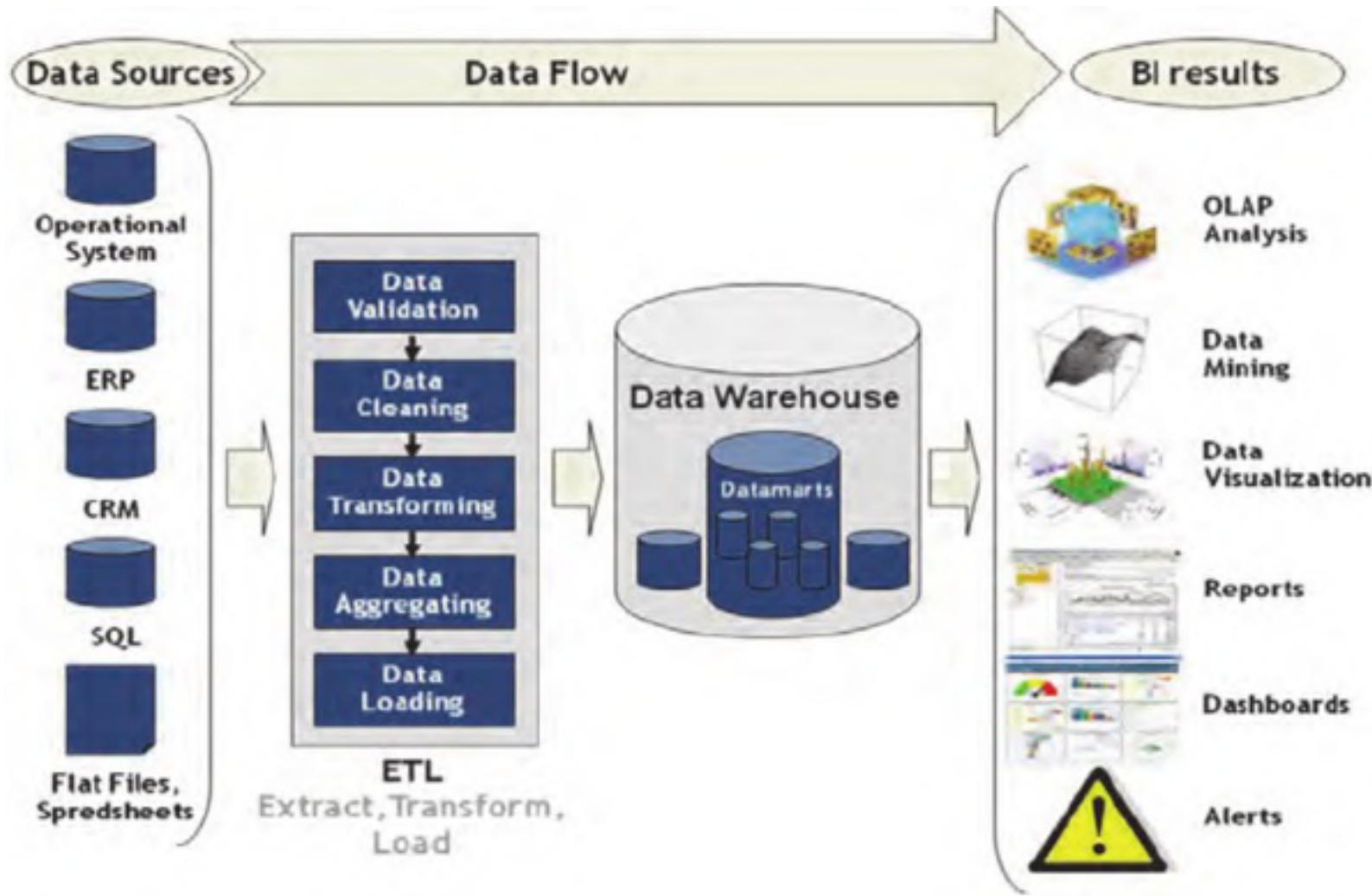
# Content

- Main components of DW/BI
- Architectures
- Component details
- Required tools
- MS SQL Server components

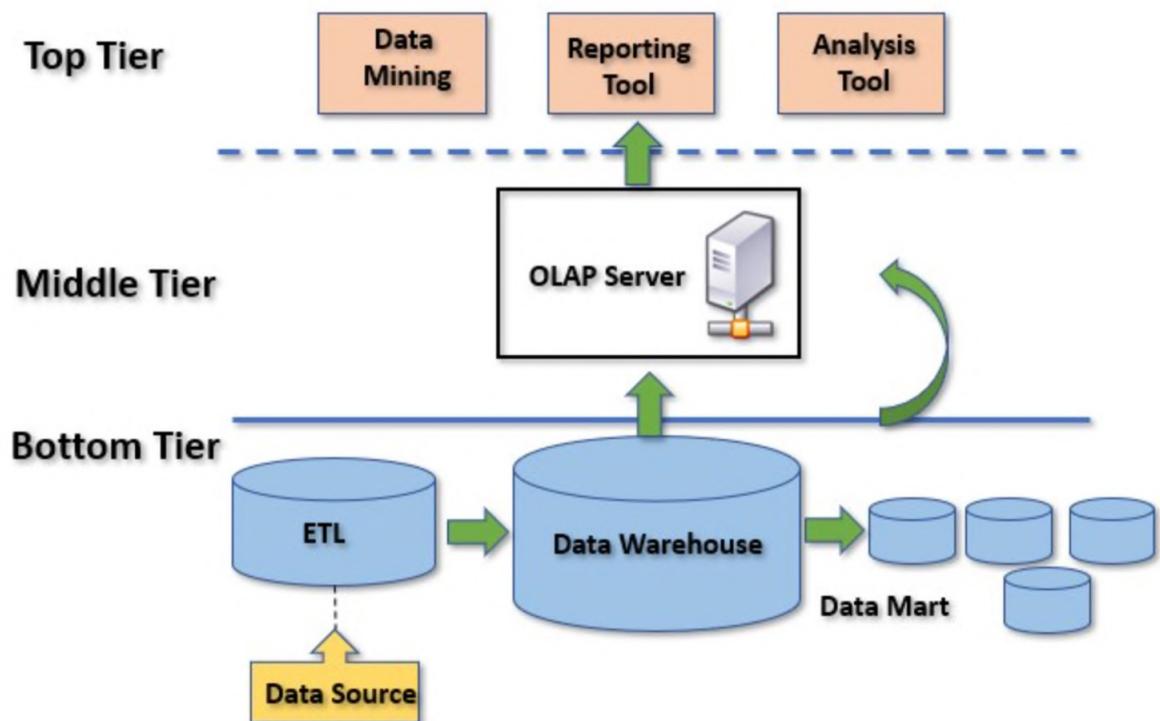
# Main Components of a DW & BI Solution

- Data sources
- ETL/ELT
- Storage layer components
- Data consumption: BI/Advanced Analytics

# A Simple Architecture



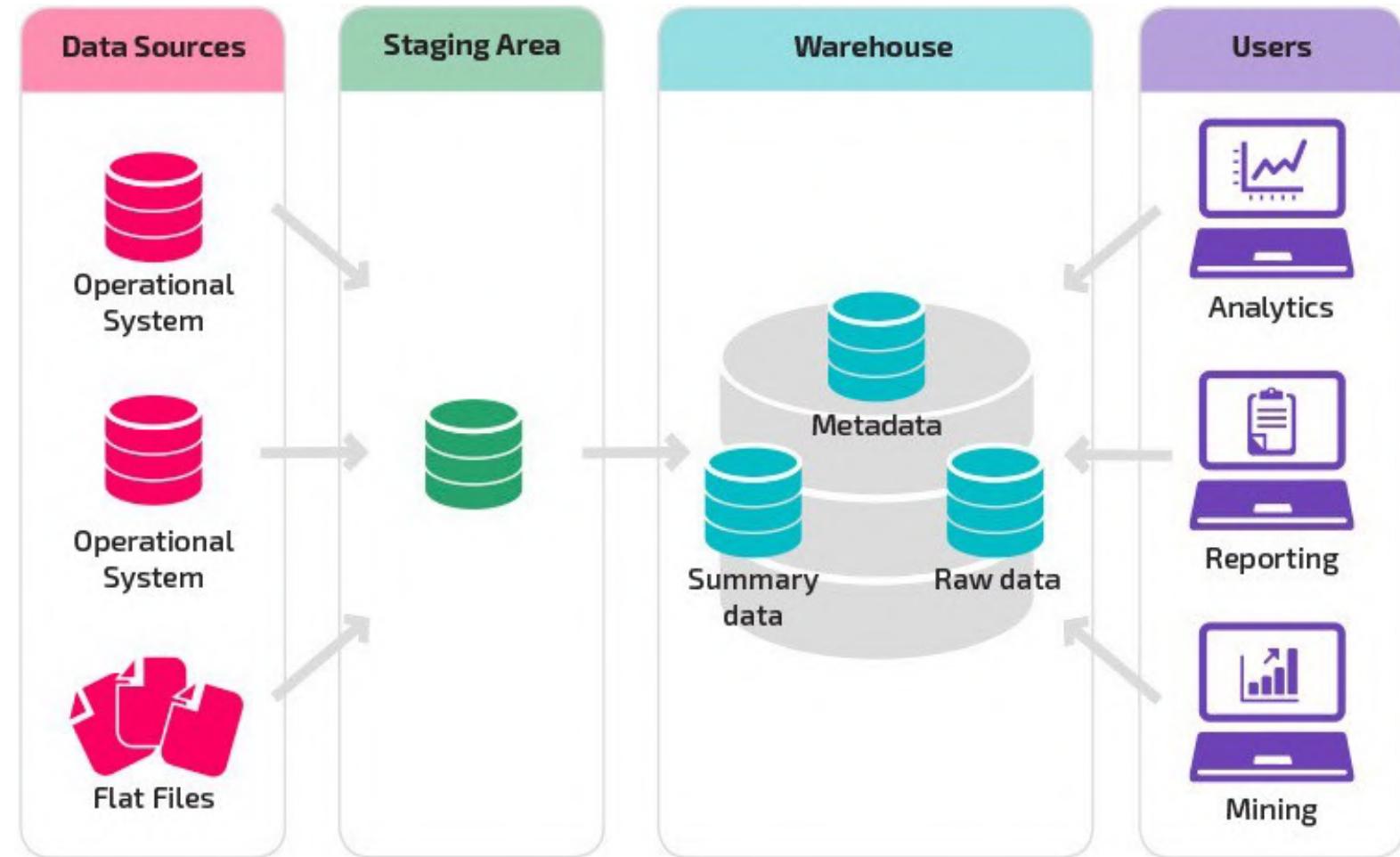
# DW Architecture: 3 Tiered



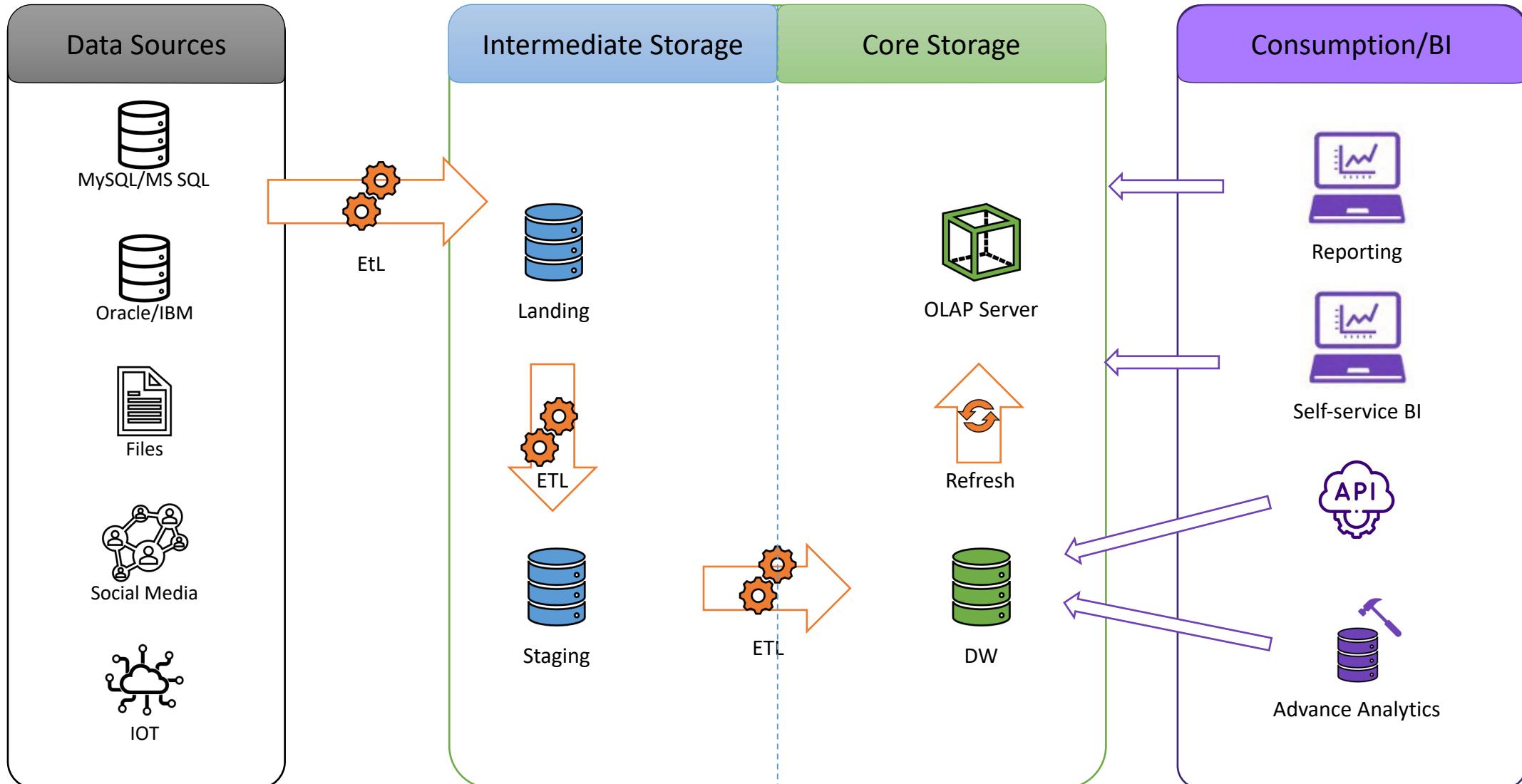
- **Top-Tier:**
  - The top tier is a front-end client layer.
  - Tools and APIs that you connect and get data out from the data warehouse.
  - It could be query tools, reporting tools, managed query tools, analysis tools and data mining tools.
- **Middle Tier:**
  - The middle tier is an OLAP server.
  - For a user, this application tier presents an abstracted view of the database.
  - Acts as a mediator between the end-user and the database.
- **Bottom Tier:**
  - The database of the data warehouse servers as the bottom tier.
  - It is usually a relational database system.
  - Data is cleansed, transformed, and loaded into this layer using back-end tools.

# DW Architectures: with Staging Area

- Purpose:
  - Minimize dependency
  - Minimize impact



# A Modern Architecture



# Architectural Components: Data Sources

- Data sources can be:
  - Structured, semi-structured or unstructured
  - Disintegrated data
  - Internal or external data
  - Sometime, non-digitized
  - Examples:
    - Backend databases of operational systems (OLTP)
    - Files (text, csv, tsv, xml, etc.)
    - APIs
    - IOT devices
    - Web data (logs, click streams, etc.)
    - Social media data

# Architectural Components: ETL

- Extraction: reading source data
  - Full vs. incremental (CDC)
  - Online vs. offline
  - Push vs. pull
- Transformation: preparing data to be inserted to the target model
  - Cleansing, integrating (matching & combining), de-duplicating, enriching, calculations, aggregations, etc.
- Loading: insertion of data to the target
  - Different types of tables (facts, dimensions, etc.)
  - Order of loading matters
  - Surrogate key generation
  - Hierarchy management

# Architectural Components: Storage Layer

- Storage layer can be implemented in many different forms (next slide)
- Intermediate stages
  - Landing area
  - Staging area
- Data Modelling
  - Dimensional data model: facts and dimension tables
  - Multiple schema options: star, snowflake, etc.
  - Different levels of aggregations: row vs aggregate

# Implementations of Storage Layer

- Operational Data Store:
  - ODS is a data store used when neither data warehouse nor OLTP systems support organizations reporting needs
  - ODS is refreshed in real time and hence used for real-time BI/analytical requirements
  - Often landing area is converted to ODS
- Enterprise Data Warehouse:
  - Centralized data warehouse which provides decision support services across the enterprise
  - It offers a unified approach for organizing and representing data
  - It also provides the ability to classify data according to the subject and give access according to those divisions
- Data Mart:
  - A subset of the data warehouse specially designed for a particular line of business, such as sales, distribution or finance
- Data Lake:
  - It is a centralized repository that allows you to store all your structured and unstructured data at any scale

# Architectural Components: OLAP Engine

- Sometimes called ‘Semantic Layer’: instead of an OLAP engine, aggregated tables can be used
- Storage modes:
  - ROLAP
  - MOLAP
  - HOLAP
- OLAP operations:
  - Slice
  - Dice
  - Roll-up
  - Drill-down

# Architectural Components: Consumption/BI

- Operational reports
- Dashboards
- Ad-hoc analysis
- Self-service BI
- Advanced analytics
  - Data mining
  - Machine learning and predictive modelling
- Data for other operational applications
  - API

# Architectural Components: Consumption/BI

- BI Applications
  - Web applications
  - Mobile applications
  - Self-service BI tools
  - BI Accessibility
    - Desktop/Laptop
    - Mobile devices
- Other
  - Data mining/modelling tools
  - APIs
  - Direct data access (a source for other systems)

# Need for DW/BI Tools

- Implementation of storage layer components
  - ODS/EDW/data mart/data lake
  - Staging/landing areas
- Implementation and automation of ETL processes
  - Data cleansing tools
  - Data validation tools
- Implementation of analytical storage layer
- Implementation of BI components and other methods of consumption

# MS SQL Server Components

- MS SQL Server
  - Database Engine
  - SSIS
  - SSAS
  - SSRS
- SQL Server Management Studio (client access tool)
- SSDT (development IDE)
- Report Builder (development IDE)
- DQS
- Power BI
- MS Excel

# Use of MS BI Components in DW

- Implementation storage layer components:
  - Database engine
  - SQL Server Management Studio – client access tool
- Implementation and automation of ETL processes:
  - SSDT – IDE for ETL flow developments
  - SSIS – ETL engine
  - SQL Server Agent – Process automation
  - SQL Server Management Studio – SSIS repository access

# Use of MS BI Components in DW

- Implementation of analytical storage layer:
  - SSDT – IDE for Cube developments
  - SSAS – Analytical processing and data mining engine
  - SQL Server Management Studio – SSAS repository access
- Implementation of BI components:
  - SSDT – IDE for report developments
  - Report Builder - IDE for static report developments
  - Mobile Report Publisher – IDE for mobile report developments
  - Report Server Configuration Manager – Integration and administration of SSRS web application
  - SSRS – Server based report generation, administration and hosting platform
  - SQL Server Management Studio – SSRS repository/meta data access
  - Power BI – Cloud based visualization platform/Self-service BI
  - MS Excel

**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

Lecture 03

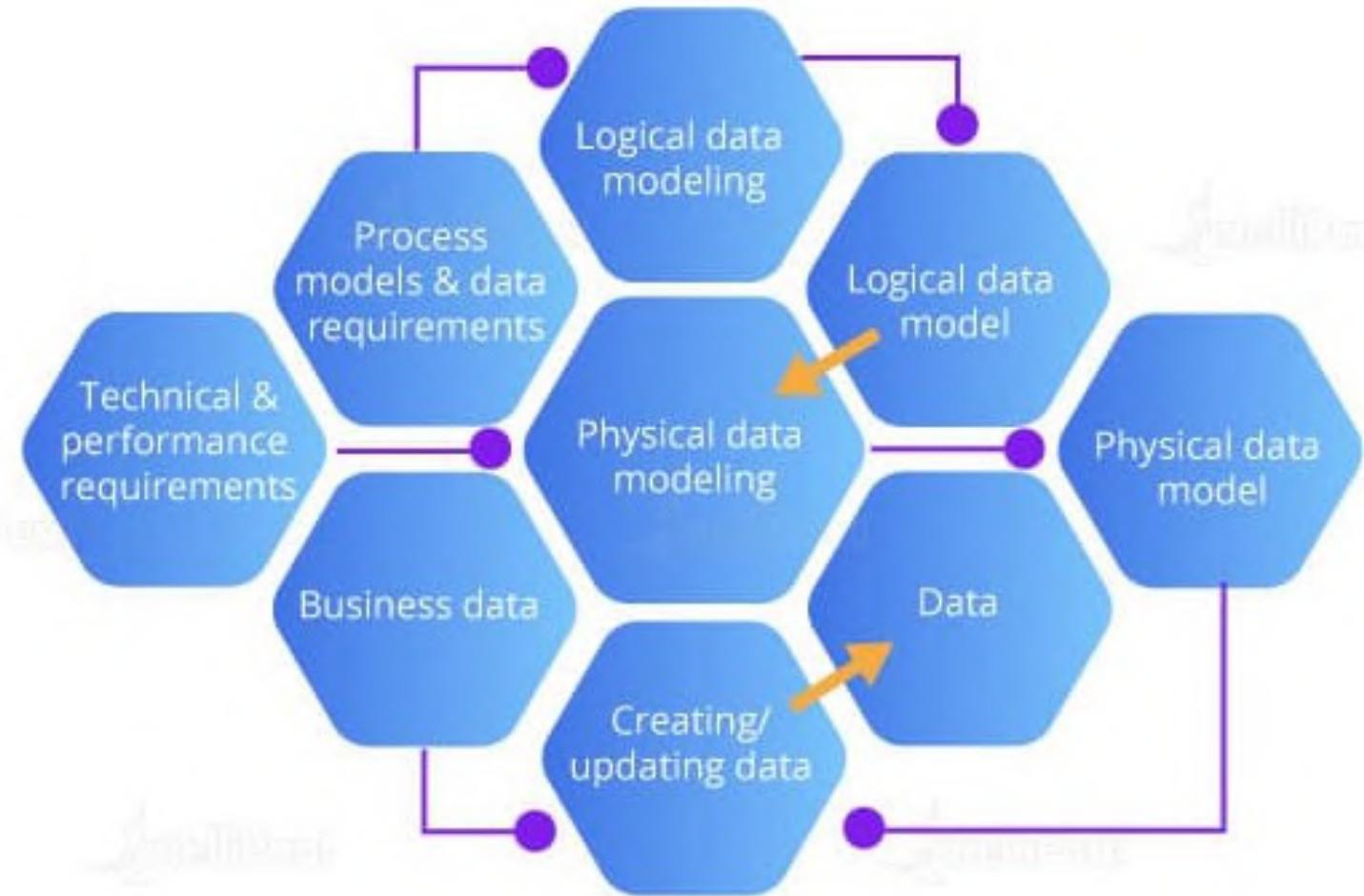
Dimensional Modelling

Part 01

# Content

- Data modelling
- Dimensional modelling
  - Dimension tables
  - Facts and fact tables
  - Schemas
    - Star
    - Snowflake
    - Galaxy
  - Surrogate keys
- Dimensional modelling steps

# Data Modelling



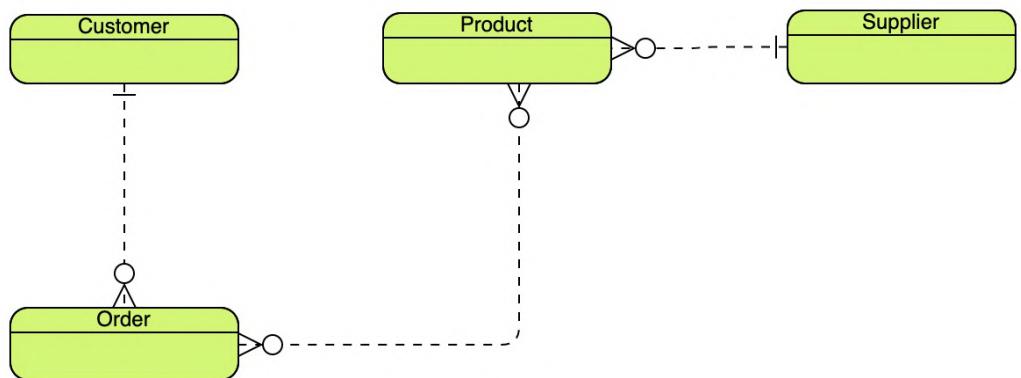
# Data Modelling: What is it?

- Data modelling is the process of creating a data model for the data to be stored in a database
- It helps to visually represent data and enforces business rules, regulatory compliances, and government policies on the data
- It is like architect's building plan
- Emphasizes on what data is needed and how it should be organized instead of what operations need to be performed on the data
- Ensure consistency in naming conventions, default values, semantics, security, while ensuring quality of the data

# Data Modelling: Why?

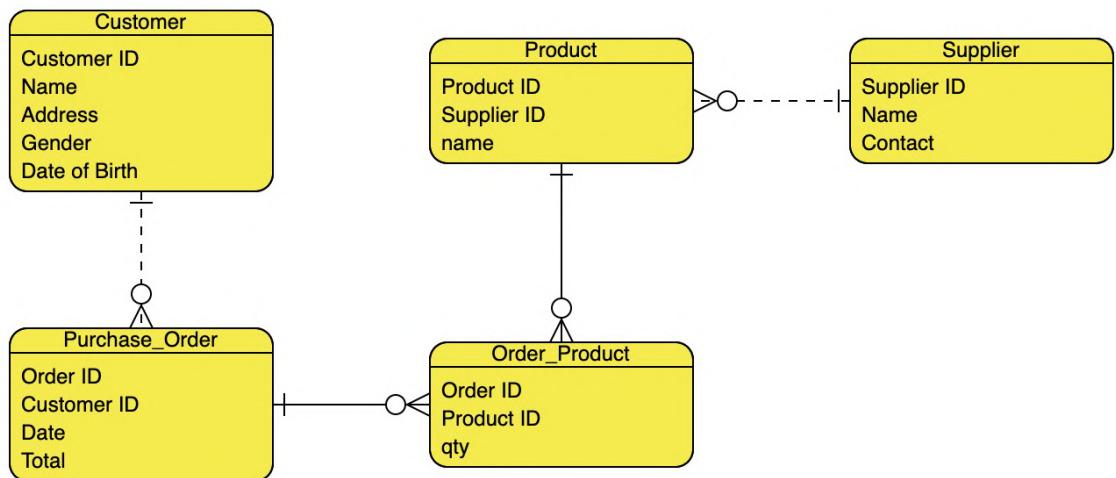
- All required data objects are accurately represented
- It provides a clear picture of the base data and can be used by database developers to create physical objects in a database
  - Start with a conceptual diagram, then a logical diagram, followed by a physical diagram
  - Finally provides you with relational tables, primary and foreign keys and stored procedures
- It is also helpful to identify missing and redundant data
- Though the initial creation of data model is labour and time consuming, in the long run, it makes your IT infrastructure upgrade and maintenance cheaper and faster.

# Conceptual Model



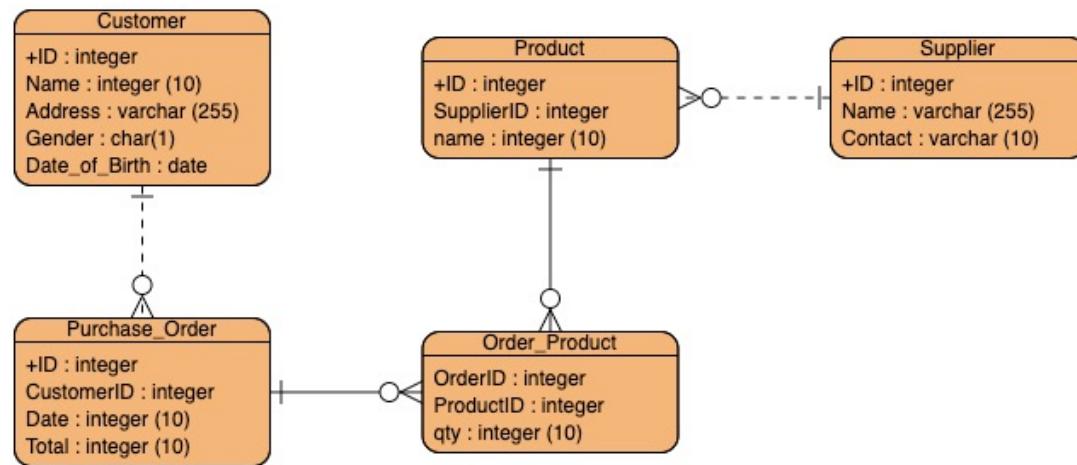
- Models the business objects that should exist in a system and the relationships between them
- A conceptual model is developed to present an overall picture of the system by recognizing the business objects involved
- It defines what entities and relationships exist, NOT tables
- Highly abstract

# Logical Model



- Detailed version of a conceptual ERD: more of a relational model
- Attributes (including PKs and FKs) in each entity are defined and operational and transactional entities are introduced
  - User friendly attribute names
- Normalized and entity relationships are mapped into additional relations as required (e.g., M:N)
- Although a logical data model is still independent of the actual database system in which the database will be created, you can still consider that it affects the design

# Physical Model



- Developed for a specific version of a DBMS, location, data storage or technology to be used in the project
  - Follow DBMS compatible names and data types
- Columns should have exact data types, lengths assigned and default values
- Constraints are defined

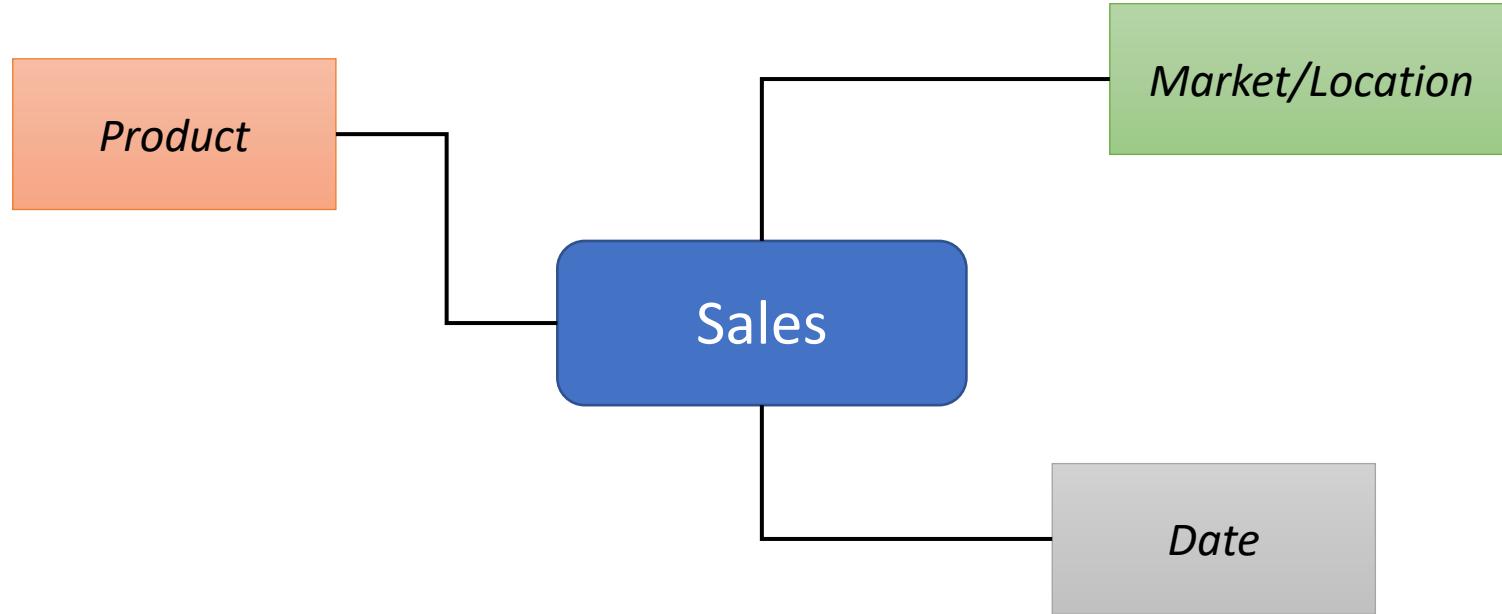
# Dimensional Modelling



# Dimensional Data Models

- The concept of Dimensional Modelling was developed by Ralph Kimball and is comprised of “*fact*” and “*dimension*” tables.
- A Dimensional model is designed to read, summarize, analyse numeric information like values, balances, counts, weights, etc. in a data warehouse.
- In contrast,
  - relational models are optimized for adding, updating and deleting of data in a real-time online transactional system (OLTP) and,
  - in the relational models, normalization reduce redundancy in data.
- On the contrary, dimensional model arranges data in such a way that it is easier to retrieve information and generate reports.

# Dimensional Data Models for Analysis



*“We sell products in various markets (store locations). Thus, how to measure the performance of the organization in terms of sales, product-wise, location-wise, over a period of time?”*

# Dimension Tables

- Dimensions provide the context surrounding a business process event, answering who, what, where, when, how & why like questions.
  - e.g., to understand sales performances, related dimensions would be:
    - *Who* – customer
    - *Where* – locations
    - *What* – products
- In other words, a dimension is a window to view measures/transactional values, providing meaning to analytical capabilities such as grouping, filtering, drilling down, etc.
- Dimensions encapsulates attributes (textual/descriptive context) associated with measures/values by separating and categorising these attributes into logical, distinct groups.
- Each dimension table is defined by a single primary key.
  - Surrogate Key: an integer value

# Dimension Tables

Product Dimension
Product Key (PK)
SKU Number (Natural Key)
Product Description
Brand Name
Category Name
Department Name
Package Type
Package Size
Weight
Weight Unit of Measure
Storage Type
...

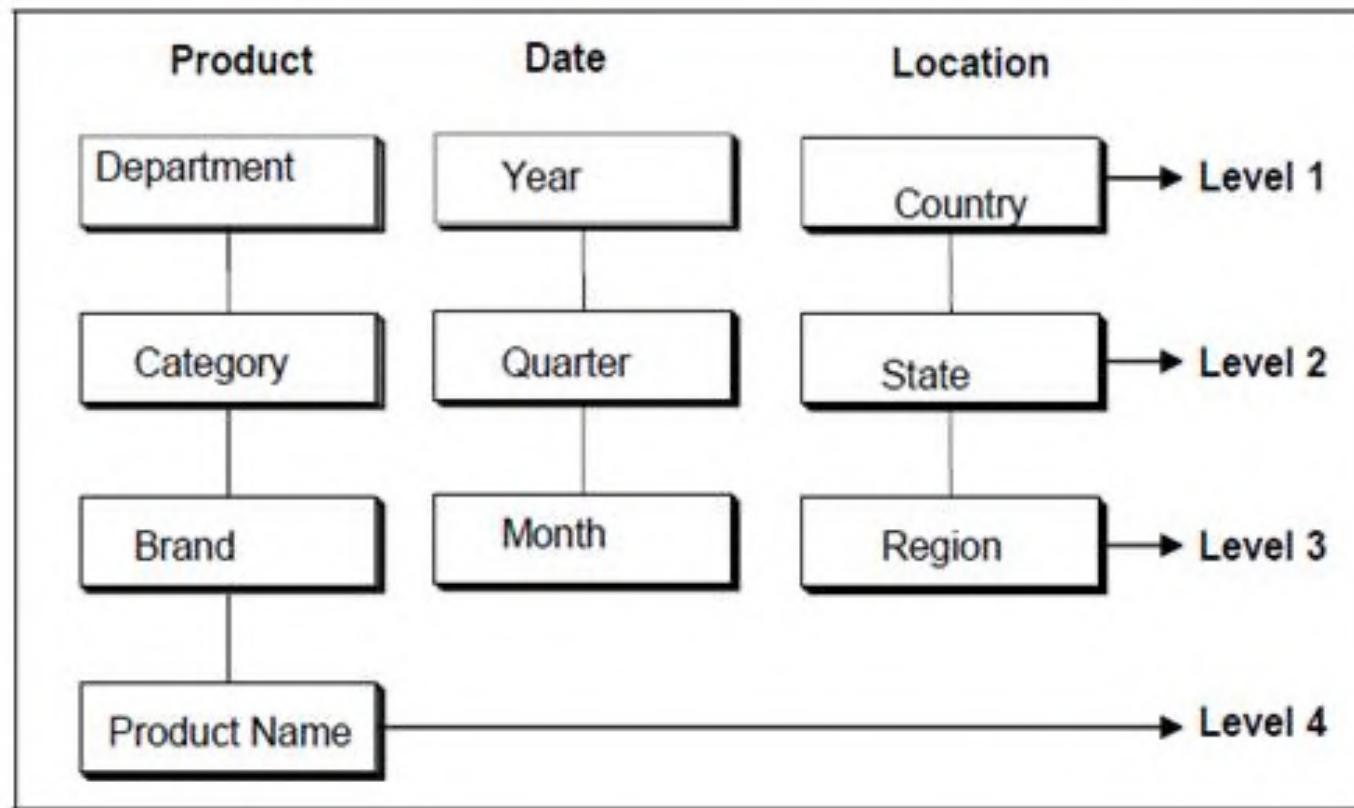
# Attributes

- Attributes are the various characteristics of the dimension
- In a “*Location*” dimension, attributes could be:
  - State
  - Country
  - Zip code
- Attributes are used to search, filter, or classify measures/transactional data
- Dimension tables contain *attributes* in columns

# Hierarchies

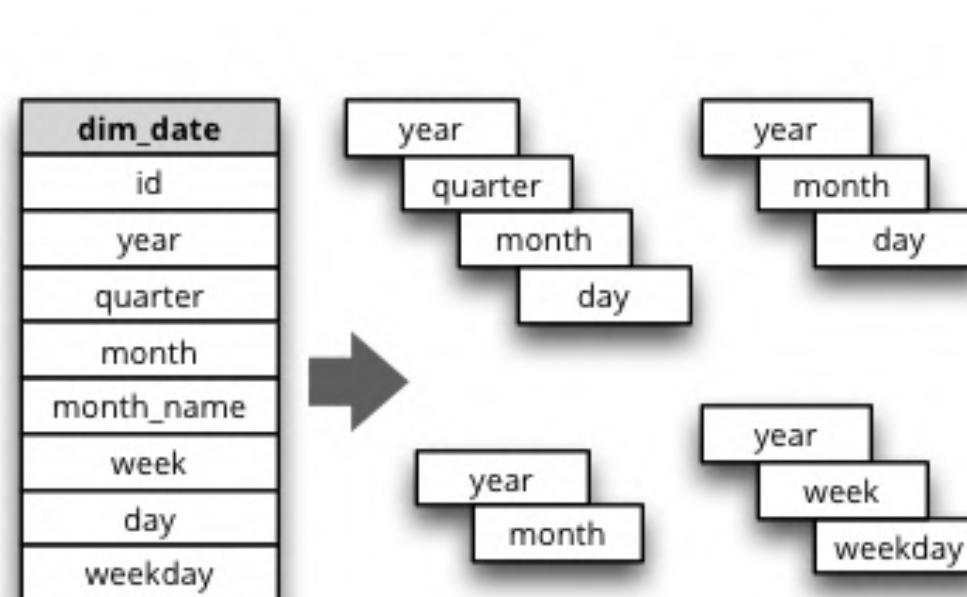
- It is the specification of levels that represents relationships between different attributes within an entity (could be one or more dimensions)
- In other words, it is a series of parent-child relationships where parent members can be further aggregated as children of another parent
  - Hierarchies in *Date* dimension: **Year → Quarter → Month → Day**
  - Hierarchies in *Product* dimension: **Category → Brand → Product**
  - Hierarchies in *Region* dimension : **City → Region → Country**
- They provide a navigation path to drill down (deeper analysis)

# Hierarchies



# Hierarchies

- It is possible for a dimension to contain more than one hierarchy



# Hierarchies

- What hierarchies can we derive?

ProductKey	ProductAlternateKey	ProductName	Color	ProductSubcategoryName	ProductCategoryName	ProductLine
222	HL-U509-B	Sport-100 Helmet, Blue	Blue	Helmets	Accessories	S
487	HY-1023-70	Hydration Pack - 70 oz.	Silver	Hydration Packs	Accessories	S
528	TT-M928	Mountain Tire Tube	NA	Tires and Tubes	Accessories	M
529	TT-R982	Road Tire Tube	NA	Tires and Tubes	Accessories	R
341	BK-R50B-48	Road-650 Black, 48	Black	Road Bikes	Bikes	R
345	BK-M82S-42	Mountain-100 Silver, 42	Silver	Mountain Bikes	Bikes	M
563	BK-T79Y-54	Touring-1000 Yellow, 54	Yellow	Touring Bikes	Bikes	T
237	LJ-0192-X	Long-Sleeve Logo Jersey, XL	Multi	Jerseys	Clothing	S
445	SH-M897-S	Men's Sports Shorts, S	Black	Shorts	Clothing	S
463	GL-H102-S	Half-Finger Gloves, S	Black	Gloves	Clothing	S
424	RW-R820	HL Road Rear Wheel	Black	Wheels	Components	R
427	FR-M63B-44	ML Mountain Frame - Black, 44	Black	Mountain Frames	Components	M
430	FR-R72Y-40	ML Road Frame-W - Yellow, 40	Yellow	Road Frames	Components	R

# Facts and Fact Tables

## Facts/Measures

- Facts are the measurements/metrics/transaction values generated from the business process
  - e.g., sales numbers

## Fact Tables

- Stores,
  - the measures of interest (facts) and,
  - foreign keys to dimension tables.
- These measures are stored in the fact table with the appropriate granularity, or level of detail: **“Grain”**
  - e.g., one record in a *Sales* fact table per line item sold in each transaction
- Fact table primary key is a composite key made up of all or a subset of foreign keys

# Fact Tables

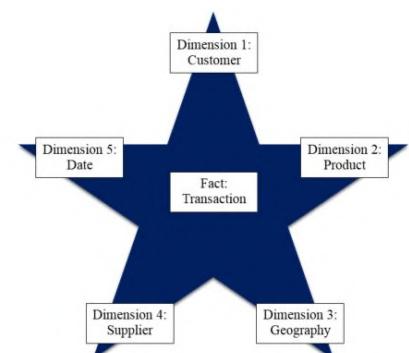
Retail Sales Facts
Date Key (FK)
Product Key (FK)
Store Key (FK)
Promotion Key (FK)
Customer Key (FK)
Clerk Key (FK)
Transaction #
Sales Dollars
Sales Units

# Schemas for Dimensional Modelling

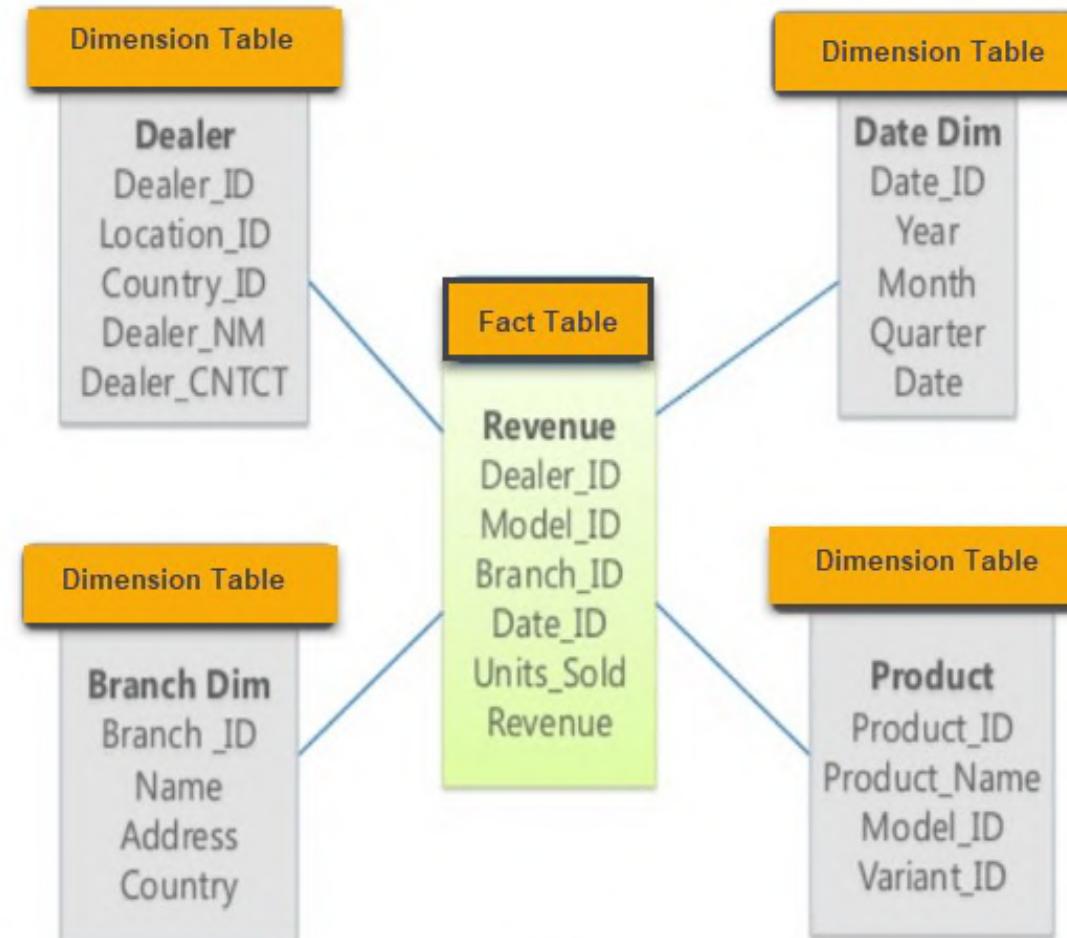
- Fact tables and dimension tables are joined to define a dimensional model
- Schemas provide techniques to join facts and dimensions
- Types of schemas:
  - Star
  - Snowflake
  - Galaxy

# Star Schema

- Single object (fact table) sits in the middle and connected to other surrounding objects (dimension tables) like a star
- A simple star consists of one fact table; a complex star can have more than one fact table
- **Each and every dimensional entity in a star schema is represented with only one dimension table**
- Dimension tables are not normalized
- Dimension table is joined to the fact table
- Dimension table are not joined to each other



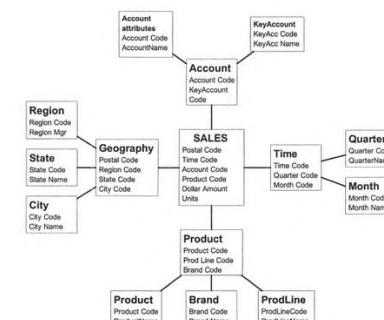
# Star Schema



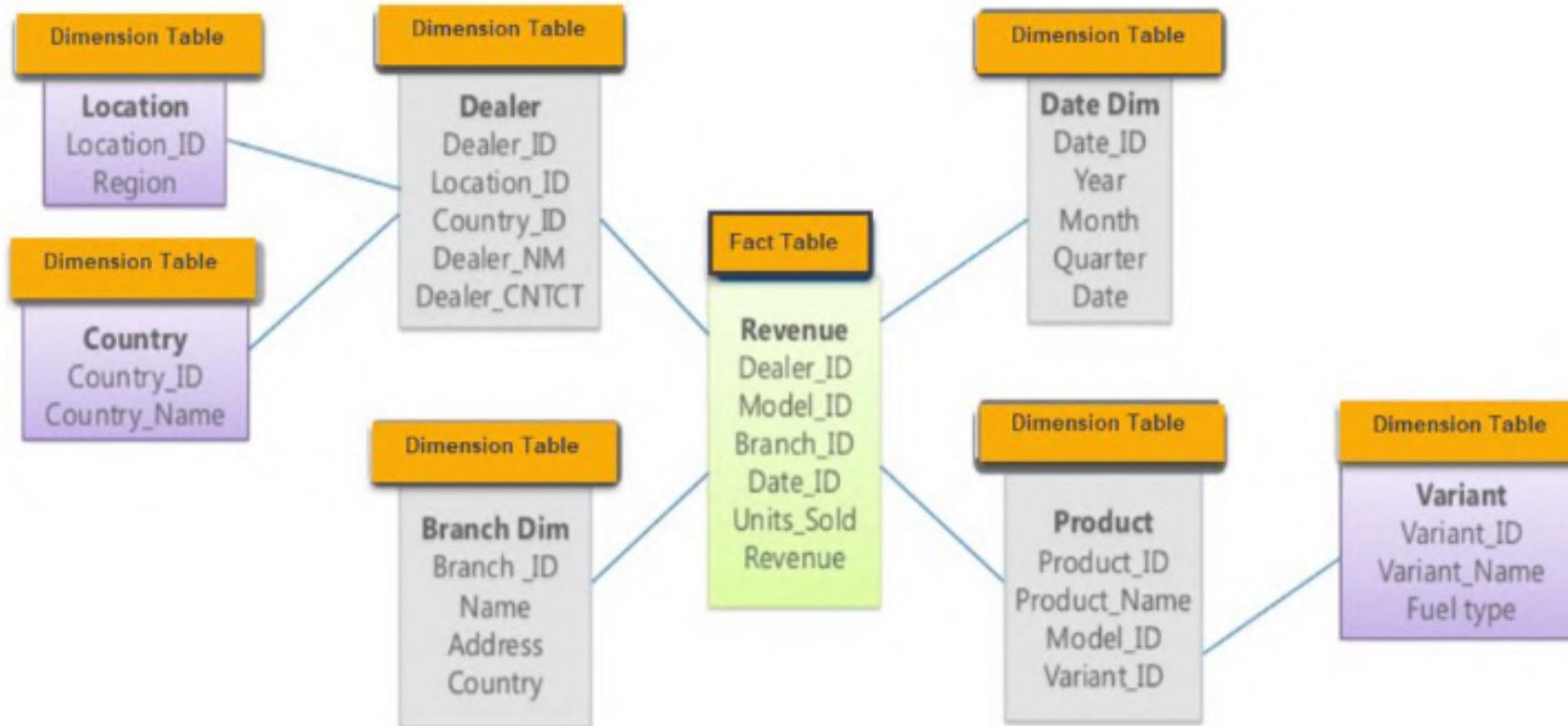
Example of Star Schema

# Snowflake Schema

- An extension of the star schema, where one or more surrounding point of the star expands into more points
- Some dimension tables in the snowflake schema are **normalized**, each representing one or more levels in the dimensional hierarchy
- The main benefit of the snowflake schema is it uses smaller disk space
- Easier to expand a dimension: add a new table to the schema
- Due to multiple tables, query performance is reduced
- More maintenance efforts, because of the more lookup tables



# Snowflake Schema



# Star vs. Snowflake

- By design:

Star	Snowflake
Contains a fact table surrounded by dimension tables	Contains a fact table surrounded by dimension tables, which are in turn surrounded by more dimension tables
A single join with the fact table creates the relationship between the fact and the dimensional entity	Multiple joins are required to create the relationship between the fact and the dimensional entity
Hierarchies for the dimension are stored in one dimension table	Hierarchies for the dimension are divided into separate table
Simple DB design	Complex DB design
Denormalized data structures	Normalized data structures
High level of data redundancy	Low level of data redundancy

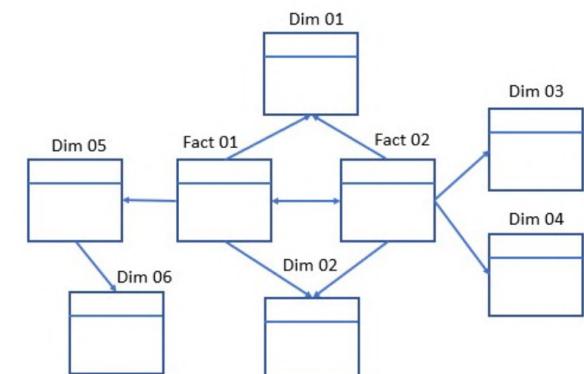
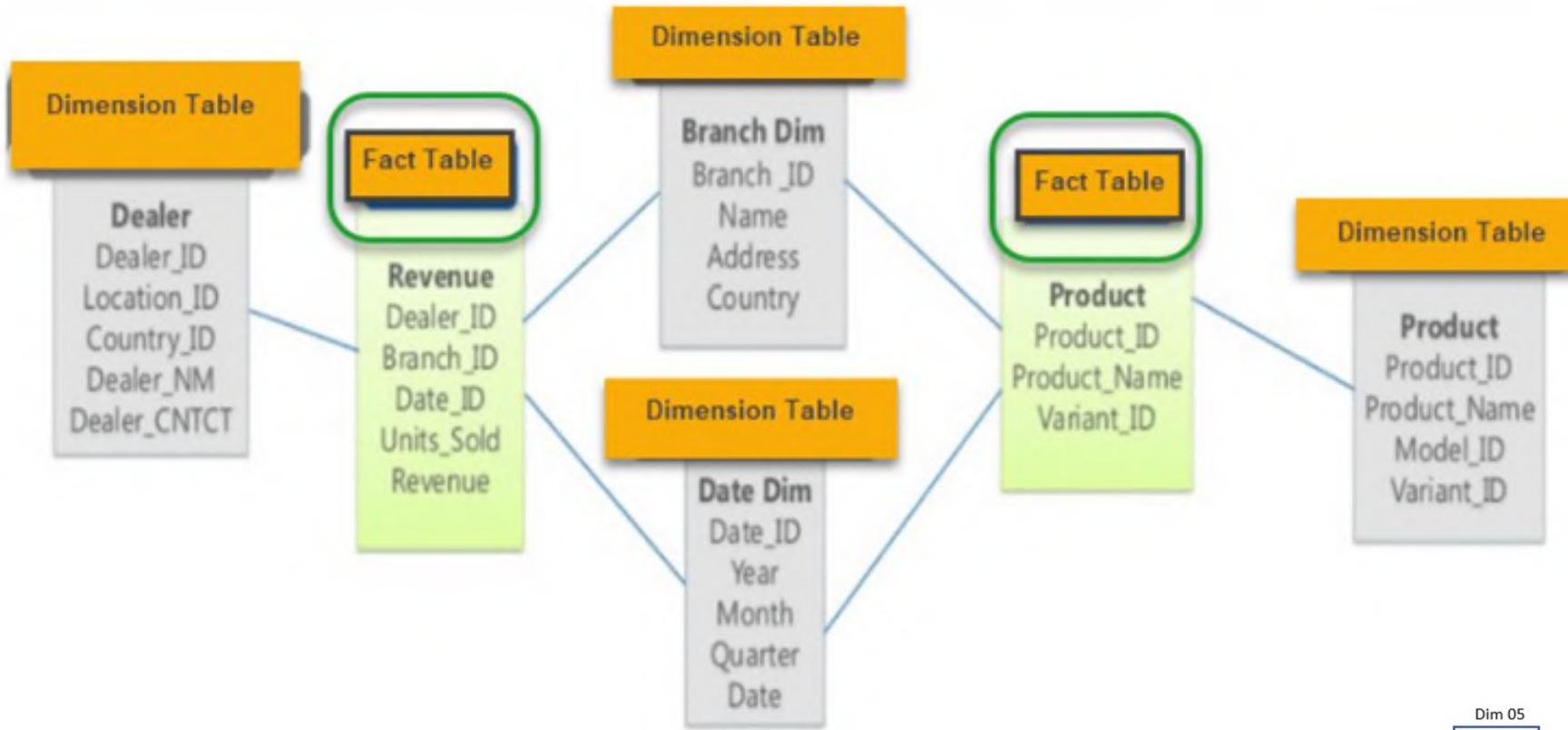
# Star vs. Snowflake

- Model simplicity (Star: one-dimension table to work with)
- Storage efficiency (Snowflake: no redundancy)
- ETL complexity (Star: less lookups)
- Query performance:
  - SQL table joins (Star) vs. I/O performance (Snowflake)
  - In general, the simplified joins of a star design give the best performance for smaller tables (few columns and thousands of rows in dimension), whereas the snowflake design has better performance for larger tables (many columns and millions of rows in dimension)
- Analytical requirements:
  - Metrics analysis: “what is the revenue for a given customer” (Star)
  - Dimension analysis: “how many customers from a given region” (Snowflake)

# Galaxy Schema

- A galaxy schema contains two or more fact tables that share dimension tables
- Galaxy schema can be derived as a collection of star schemas interlinked and completely normalized, to avoid redundancy
  - Since it is viewed as a collection of star schemas, it is also called as a **Fact Constellation Schema** (same reason to be called as galaxy too)
- It can also be a meaningful association of a Snowflake schema with a Star schema, where the fact tables of both schemas can be linked
- The dimensions in this schema are normalized into separate dimensions based on the various levels of hierarchy
  - e.g., If geography has four levels of hierarchy like region, country, state, and city then galaxy schema should have four dimensions
- Moreover, it is possible to build this type of schema by splitting the one-star schema into more Star schemes

# Galaxy Schema



# Database Keys

- **Key:** is one or more data attributes that uniquely identify an occurrence in an entity. In a physical database, a key would be formed of one or more table columns whose value(s) uniquely identifies a row within a relational table
- **Natural key:** A key that is formed of attributes that already exist in the real world
  - e.g., NIC, PP Number
- **Candidate key:** Minimal set of attributes that uniquely identifies each occurrence of an entity
- **Primary key:** Candidate key selected to uniquely identify each occurrence of an entity
- **Composite key:** A candidate key that consists of two or more attributes
- **Foreign key:** One or more attributes in an entity that represents a key, either primary or secondary, in another entity
- **Alternate key:** Also known as a secondary key, is another unique identifier from the remaining candidate keys, when the primary key is selected
- **Surrogate key:** A key with no business meaning

# Surrogate Keys

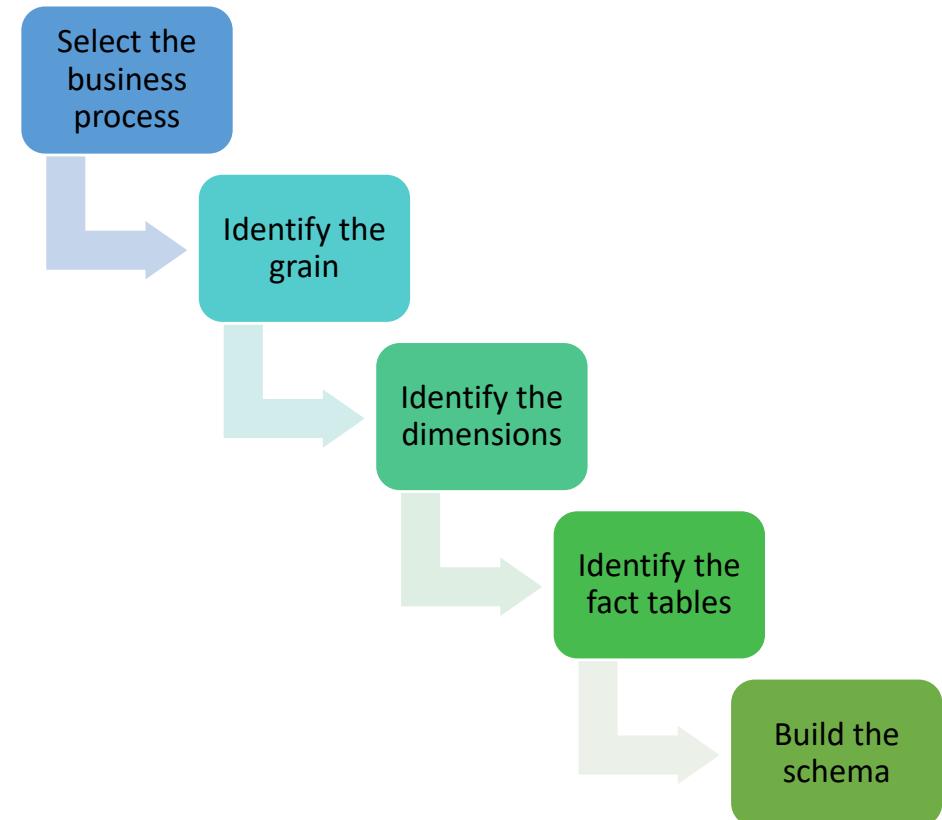
- A unique primary key for dimension tables
- Cannot be the operational system's natural key
  - Duplicates when track changes over time
  - Created by more than one source system
  - Performance issues – text fields
- Simple integers, assigned in sequence
- The date dimension is a special case

# Surrogate Keys: Advantages

- Protects the DW system from unexpected administrative changes in sources
- Allow the system to integrate the same data
- Provide the means for tracking changes in dimension attributes over time
- Integer surrogate keys can improve query and processing performance compared to larger character (text) keys

# Steps of Dimensional Modelling

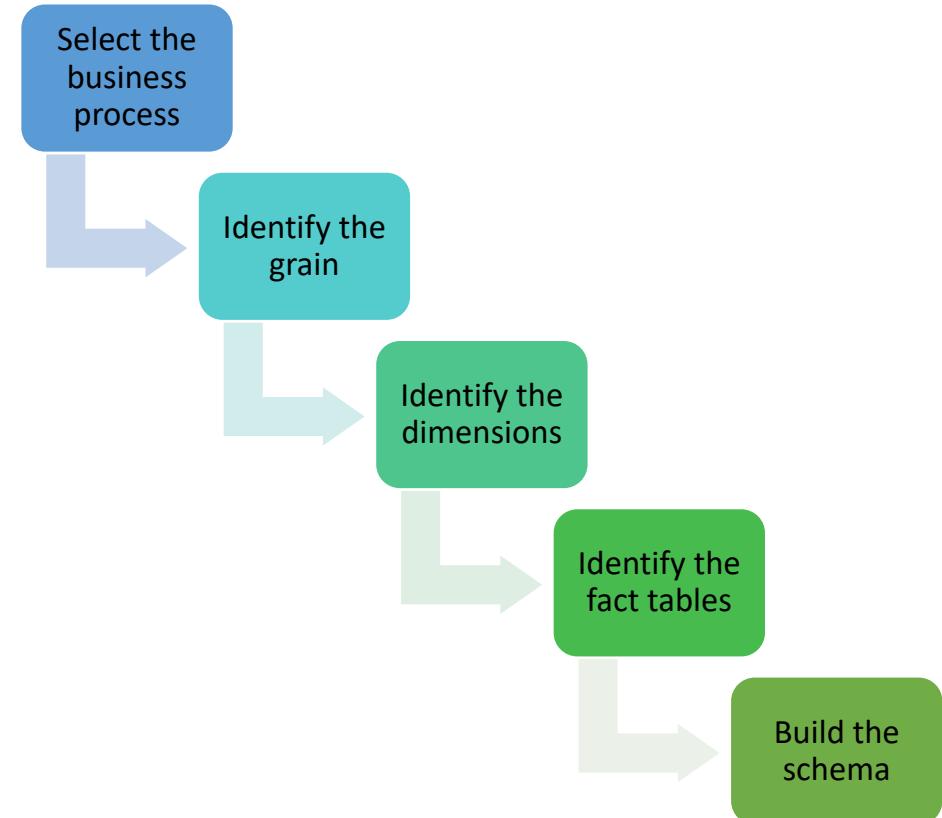
- Step 1 - Identify the business process:
  - Identifying the actual business process the data warehousing solution should address
    - This could be marketing, sales, hr, etc. as per the analytical needs of the organization
  - The selection of the business process also depends on the availability and the quality of data for that process
  - A failure in this process would impact in cascading and irreparable defects
  - To describe the business process, plain text or basic Business Process Modelling Notation (BPMN) or Unified Modelling Language (UML) can be used



# Steps of Dimensional Modelling

- Step 2 - Identify the grain:

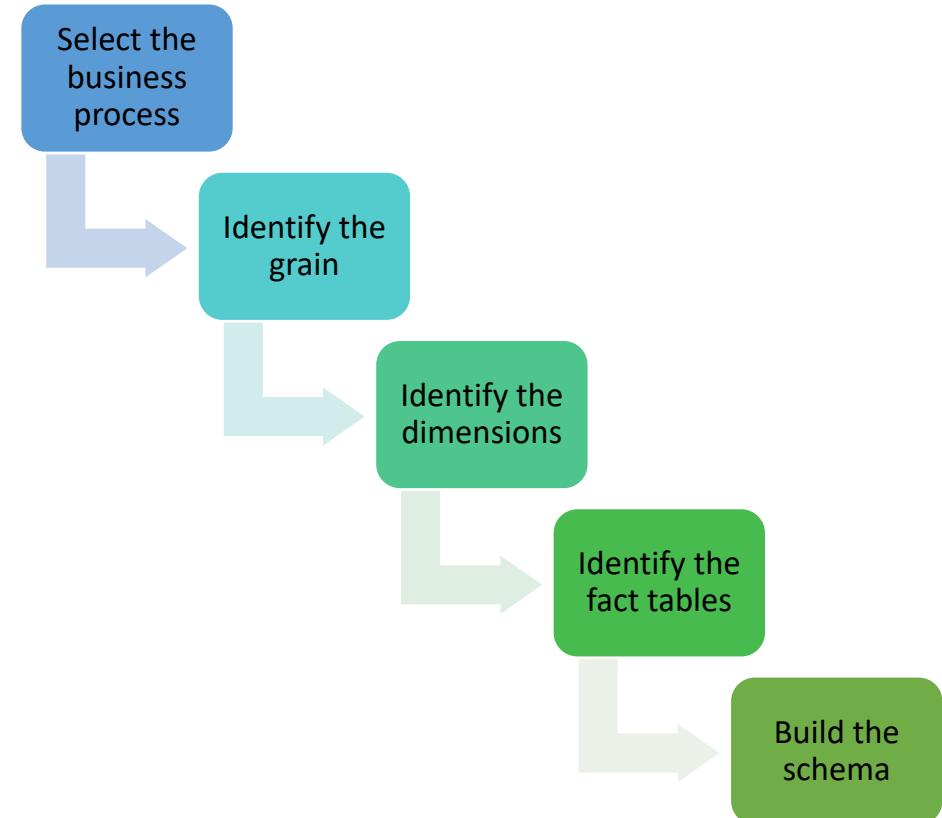
- It is the process of identifying the lowest level of information for tables (including aggregate tables) in your data warehouse model
  - e.g., Sales data – line item wise for each order
- Store aggregate data on a monthly, weekly, daily or hourly basis, based on the nature of reports and frequent analysis
  - e.g., The CEO wants to find the sales for specific products in different locations on a daily basis
- In general, data warehouse fact tables store data in the lowest possible granular level



# Steps of Dimensional Modelling

- Step 3 - Identify the dimensions:

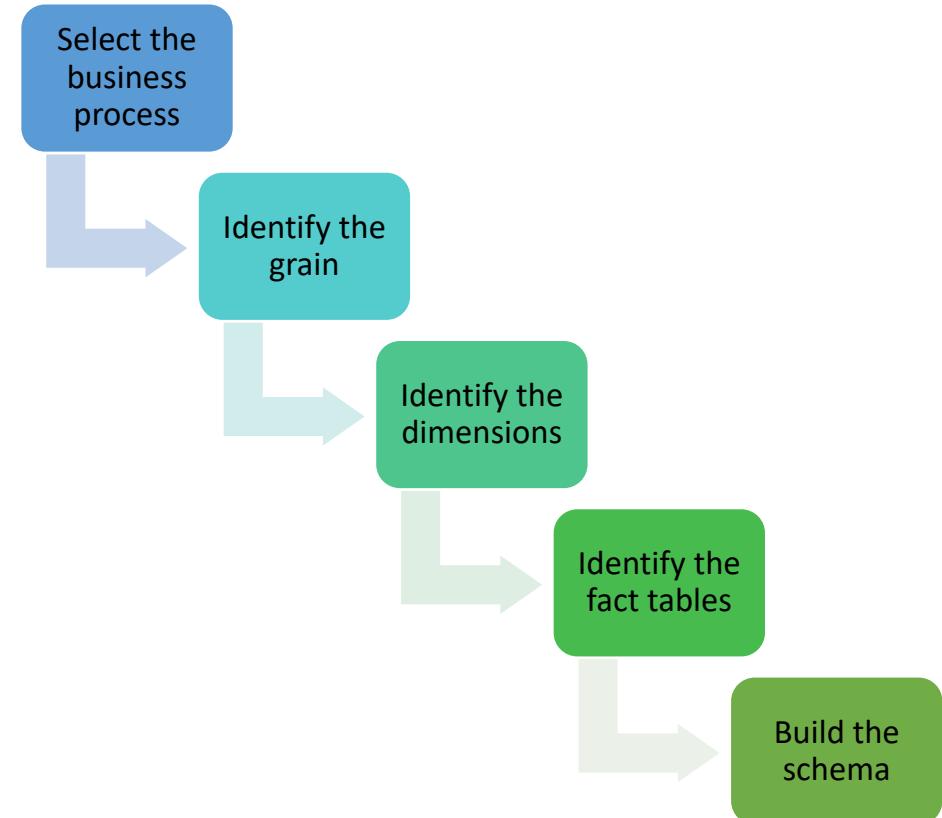
- Dimensions are nouns like date, store, inventory, etc.
- Dimensions are where all the descriptive (non-measurable data) should be stored
  - For example, the *Date* dimension may contain data like a year, month and weekday
- Example of Dimensions:
  - Dimensions: Product, Location and Time
  - Attributes: for *Product*: Product key (SK), SKU, Name, Type, Specifications
  - Hierarchies: for *Location*: Country, State, City, Street Address, Name



# Steps of Dimensional Modelling

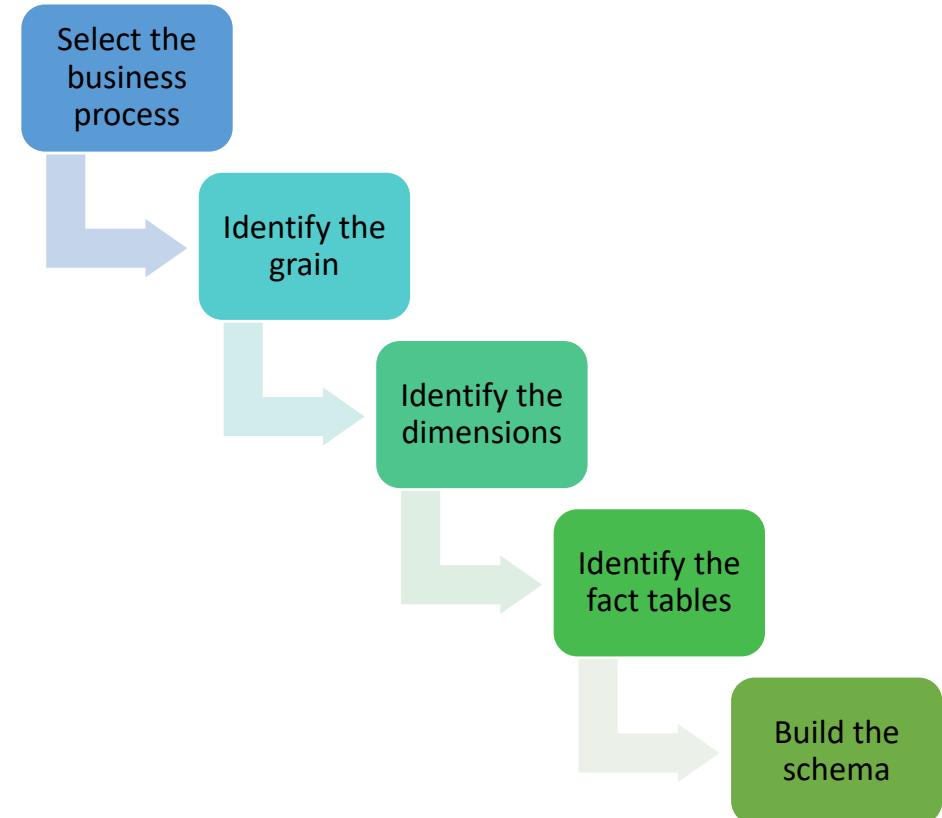
- Step 4 - Identify fact tables:

- This step is co-associated with the business users of the system because this is where they get access to data stored in the data warehouse
- Most of the fact table rows are numerical values like price or cost per unit, etc.
- Example of Facts:
  - Sales
  - Deliveries
  - Payments



# Steps of Dimensional Modelling

- Step 5 - Build the Schema:
  - A schema is nothing but the database structure (arrangement of tables)
  - Identify the schema required as per the business/analytical requirements
  - Design the schema:
    - Star Schema
    - Snowflake Schema



# General Rules for Dimensional Modelling

- Load atomic data into dimensional structures
- Build dimensional models around business processes
- Need to ensure that every fact table has an associated date dimension table
- Ensure that all facts in a single fact table are at the same grain or level of detail
- It's essential to store report labels and filter domain values in dimension tables
- Need to ensure that dimension tables use a surrogate key
- Continuously balance requirements and realities to deliver business solution to support their decision-making

**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

Lecture 04

Dimensional Modelling

Part 02

# Content

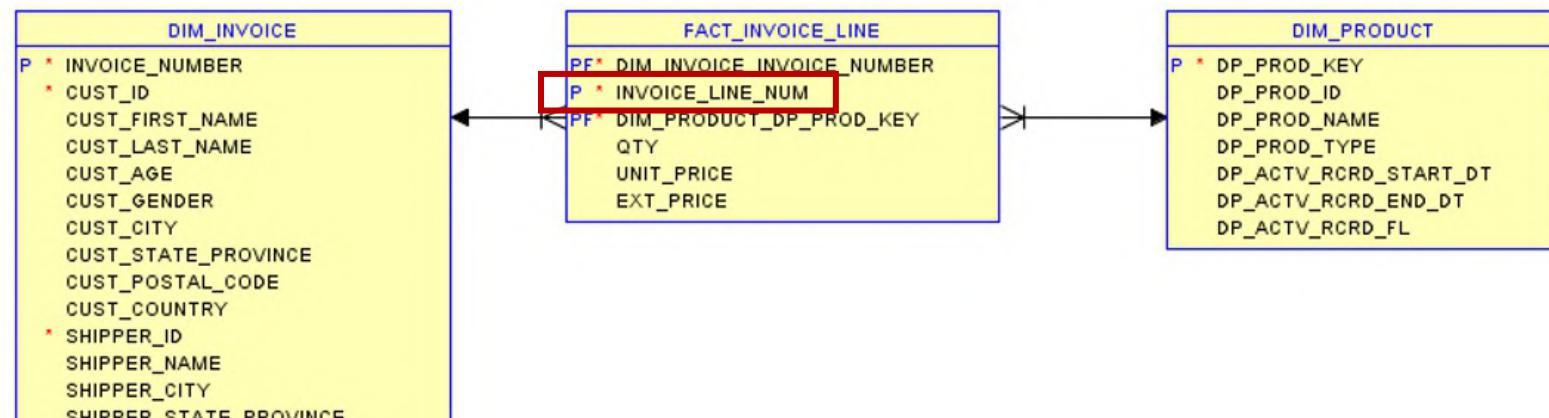
- Types of dimensions
  - Different types of dimensions
  - Slowly changing dimension types
- Types of fact tables
  - Types of measures
  - Different types of fact tables

# Types of Dimensions

---

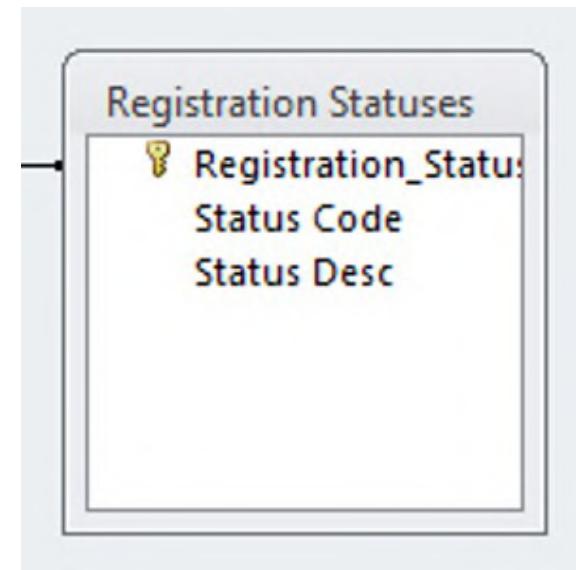
# Degenerated Dimensions

- A dimension that has no content except for its primary key.
- Thus, dimension attribute is stored in fact table (no separate dimension table)
- e.g.,
  - *Order\_No*
  - *Invoice\_No*



# Static Dimensions

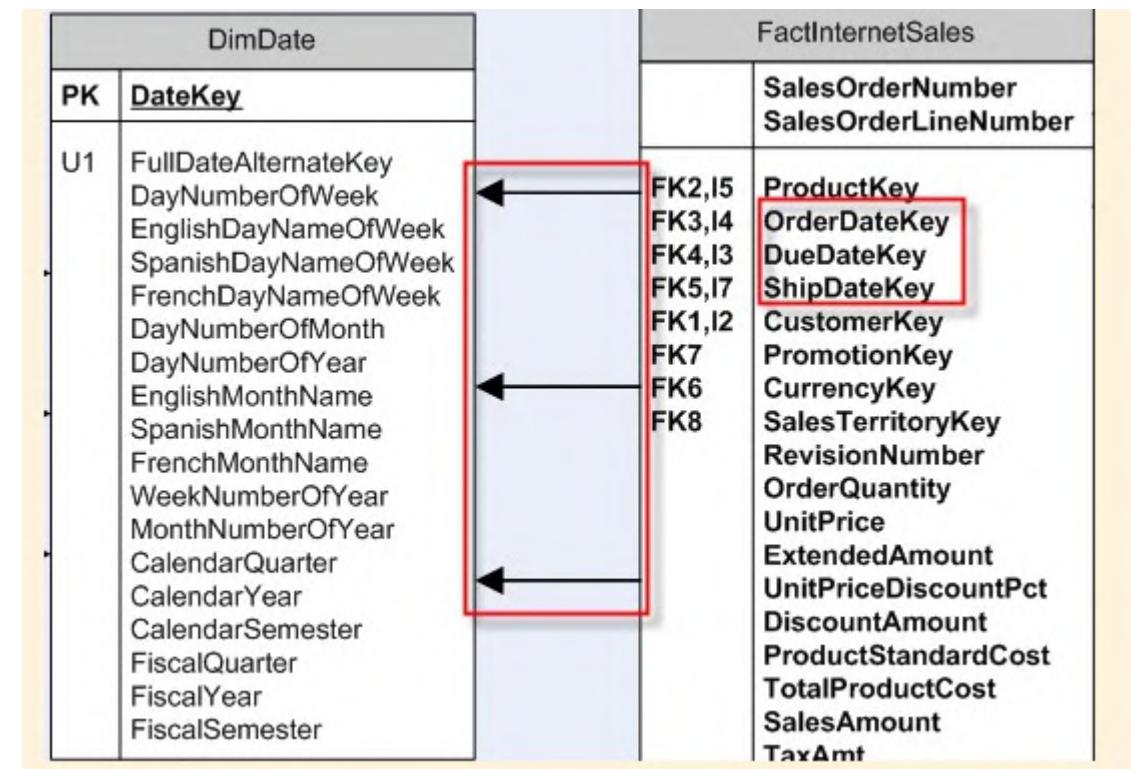
- Loaded manually
- Incremental data loading (scheduled) is not required
- Rarely change
- Examples:
  - *Status\_Code\_Dimension*
  - Dimensions for flags/indicators



# Role-Playing Dimensions

- A single physical dimension can be referenced multiple times in a fact table for multiple purposes
- e.g., *Date* dimension

*FactInternetSales* refers to  
multiple columns *DimDate*.



# Date Dimension

- Attached to virtually every fact table to allow navigation of the fact table through familiar dates, months, fiscal periods, and special days on the calendar (fiscal/calendar)
- Remove the requirement to compute any date related functions, but rather want to look it up in the date dimension
- Falls into different types of dimensions static, role-playing

	DateKey	FullDateUK	FullDateUSA	DayOfMonth	DayName	DayOfWeekUK	DayOfWeekUSA	DayOfQuarter	WeekOfMonth	WeekOfQuarter	MonthName	MONTH	Quarter	YEAR	MonthYear	IsHolidayUK	HolidayUK	IsHolidayUSA	HolidayUSA
1	20130101	01/01/2013	01/01/2013	1	Tuesday	2	3	1	1	1	January	1	1	2013	Jan-2013	1	New Year's Day	1	New Year's Day
2	20130102	02/01/2013	01/02/2013	2	Wednesday	3	4	1	1	1	January	1	1	2013	Jan-2013	0	NULL	0	NULL
3	20130103	03/01/2013	01/03/2013	3	Thursday	4	5	1	1	1	January	1	1	2013	Jan-2013	0	NULL	0	NULL
4	20130104	04/01/2013	01/04/2013	4	Friday	5	6	1	1	1	January	1	1	2013	Jan-2013	0	NULL	0	NULL
5	20130105	05/01/2013	01/05/2013	5	Saturday	6	7	1	1	1	January	1	1	2013	Jan-2013	0	NULL	0	NULL
6	20130106	06/01/2013	01/06/2013	6	Sunday	7	1	1	2	1	January	1	1	2013	Jan-2013	0	NULL	0	NULL
7	20130107	07/01/2013	01/07/2013	7	Monday	1	2	1	2	1	January	1	1	2013	Jan-2013	0	NULL	0	NULL
8	20130108	08/01/2013	01/08/2013	8	Tuesday	2	3	2	2	2	January	1	1	2013	Jan-2013	0	NULL	0	NULL
9	20130109	09/01/2013	01/09/2013	9	Wednesday	3	4	2	2	2	January	1	1	2013	Jan-2013	0	NULL	0	NULL
10	20130110	10/01/2013	01/10/2013	10	Thursday	4	5	2	2	2	January	1	1	2013	Jan-2013	0	NULL	0	NULL
11	20130111	11/01/2013	01/11/2013	11	Friday	5	6	2	2	2	January	1	1	2013	Jan-2013	0	NULL	0	NULL
12	20130112	12/01/2013	01/12/2013	12	Saturday	6	7	2	2	2	January	1	1	2013	Jan-2013	0	NULL	0	NULL
13	20130113	13/01/2013	01/13/2013	13	Sunday	7	1	2	3	2	January	1	1	2013	Jan-2013	0	NULL	0	NULL
14	20130114	14/01/2013	01/14/2013	14	Monday	1	2	2	3	2	January	1	1	2013	Jan-2013	0	NULL	0	NULL
15	20130115	15/01/2013	01/15/2013	15	Tuesday	2	3	3	3	3	January	1	1	2013	Jan-2013	0	NULL	0	NULL

# Range Dimensions

- Transform continues values into categorical attributes
- Grouping of measures in a dimension
- Required for data mining algorithms where categorical values are expected
- e.g., customer age

<b>Key</b>	<b>Age</b>	<b>Income</b>	<b>Purchase Count</b>	<b>Rating</b>	<b>Status</b>	<b>Credit Score</b>
1	20 – 29	Less than \$20,000	0 -25	Poor	Collect	Below 500
:	:	:	:	:	:	:
27	30 – 39	\$30,000 – \$39,999	26- -50	Good	Current	750 - 774
:	:	:	:	:	:	:
74	50 – 59	Over \$100,000	150+	Excellent	Current	800+

# Parent-Child Dimensions

- Based on two dimension table columns that together define the lineage relationships among the members of the dimension
- e.g., *Employee* dimension
  - *manager\_ID* attribute (refers to *employee\_ID*)

Table: Employees

	EmplID	EmplName	ManagerID
▶	1	James Smith	3
	2	Amy Jones	3
	3	Paul West	Null
	4	Jill Kelley	3
	5	Jon Grande	1
	6	Jo Brown	1

# Junk Dimensions

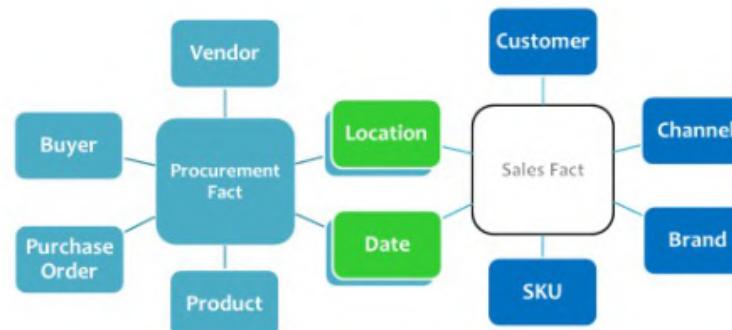
- Transactional business processes typically produce several miscellaneous flags and indicators
- Rather than making separate dimensions for each flag and attribute, a single junk dimension can be created by combining all the flags and values together
- Need not be the cartesian product of all the attributes' possible values, but only contain the combination of values that actually occur in the fact tables
- This dimension, frequently labeled as a "*transaction profile dimension*" in a schema

# Junk Dimensions

Order Indicator Key	Payment Type Description	Payment Type Group	Order Type	Commission Credit Indicator
1	Cash	Cash	Inbound	Commissionable
2	Cash	Cash	Inbound	Non-Commissionable
3	Cash	Cash	Outbound	Commissionable
4	Cash	Cash	Outbound	Non-Commissionable
5	Visa	Credit	Inbound	Commissionable
6	Visa	Credit	Inbound	Non-Commissionable
7	Visa	Credit	Outbound	Commissionable
8	Visa	Credit	Outbound	Non-Commissionable
9	MasterCard	Credit	Inbound	Commissionable
10	MasterCard	Credit	Inbound	Non-Commissionable
11	MasterCard	Credit	Outbound	Non-Commissionable
12	MasterCard	Credit	Outbound	Commissionable

# Conformed Dimensions

- A dimension that has exactly the same meaning and content when being referred from different fact tables
  - Thus, information from separate fact tables can be combined in a single report by using conformed dimension attributes that are associated with each fact table
- For two dimension tables to be considered as conformed, they must either be identical or one must be a subset of another
  - Two dimension tables that are exactly the same except for the primary key are not considered conformed dimensions
- Conformed dimension are important in making the data warehouse being “*integrated*”
- If a particular entity had different meanings and different attributes in different source systems, there must be a single version of this entity once the data flows into the data warehouse
- Examples:
  - *Location* dimension
  - *Date* dimension

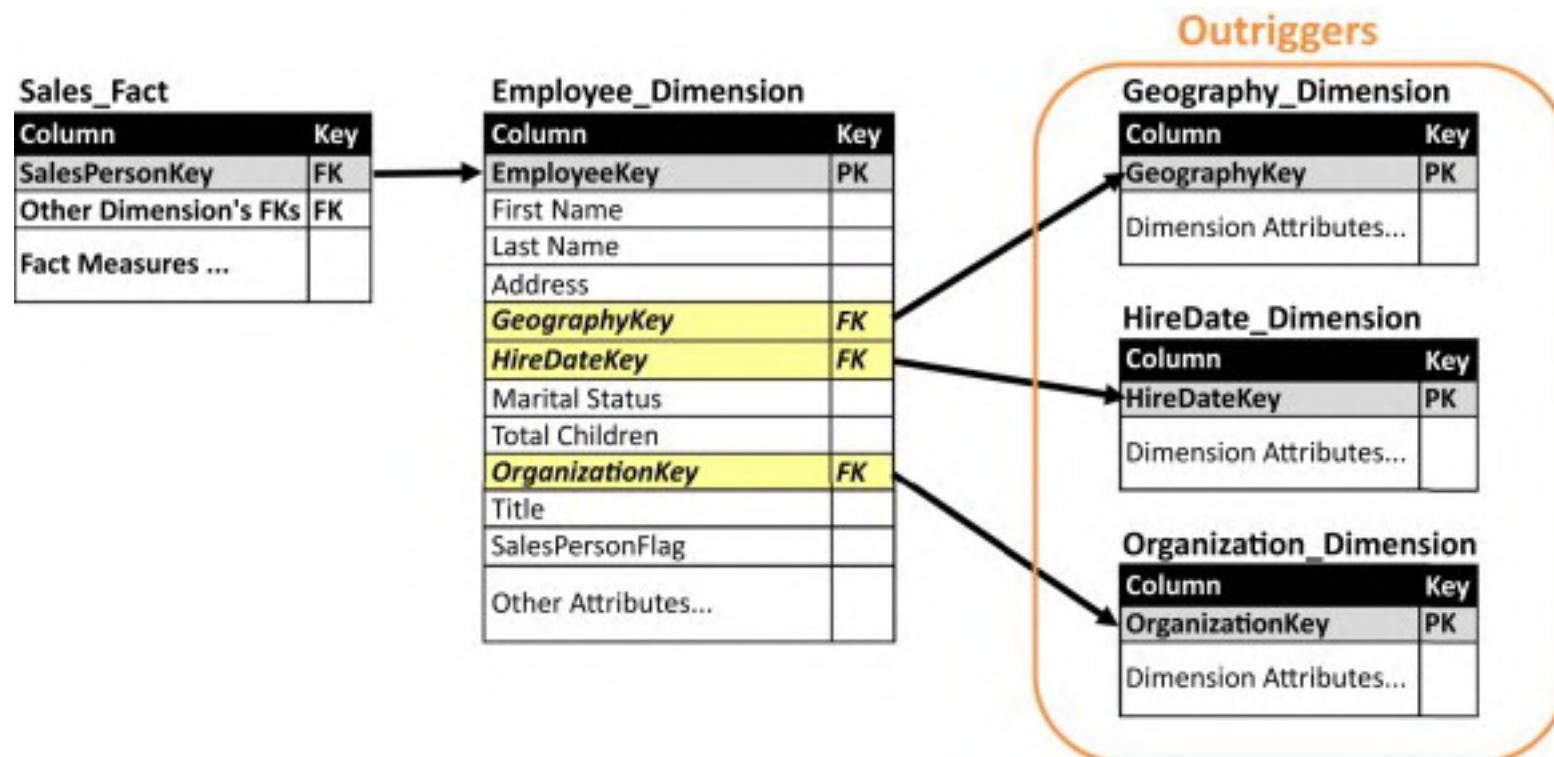


# Outrigger Dimensions

- A dimension can contain a reference to another dimension table
  - A bank account dimension can reference a separate dimension representing the date the account was opened
- These secondary dimension references are called outrigger dimensions
- Outrigger dimensions are permissible, but not recommended
- In most cases, the correlations between dimensions should be demoted to a fact table, where both dimensions are represented as separate foreign keys

# Outrigger Dimensions

- *HireDate\_Dimension* is as same as *Date\_Dimension*???



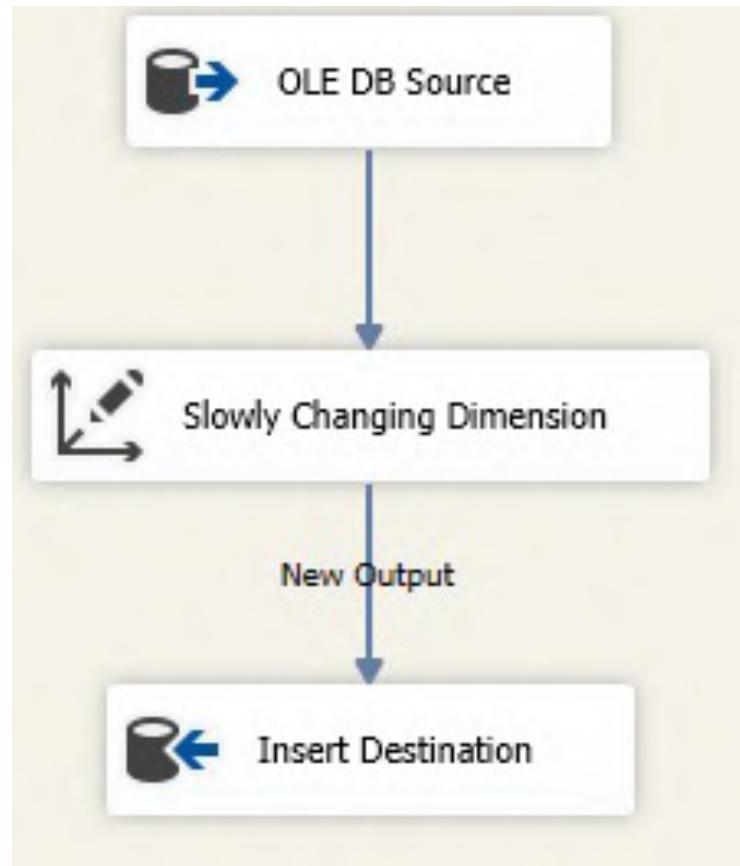
# Slowly Changing Dimensions (SCD)

- Dimensions which contains *slowly changing dimension attributes*
- *Slowly changing dimension attributes*: an attribute of a record in a dimension table that varies over time, unpredictably
- e.g., *Customer* dimension, *Employee* dimension
- Approaches to deal with SCD attributes:
  - Type 0 – Retain original
  - Type 1 – Overwrite
  - Type 2 – Add new record
  - Type 3 – Add new attribute
  - Type 4 – Add mini-dimension
  - Type 5 – Add mini-dimension & Type 1 outrigger (hybrid)
  - Type 6 – Combine approaches of types 1,2,3 (hybrid)

# Type 0 – Retain Original

- No special action is performed upon dimensional attribute value changes
- Some dimension data can remain the same as it was first time inserted, other data may be overwritten
- Appropriate for any attribute labelled “*original*”
- e.g.,
  - *customer\_original\_credit\_score*
  - *created\_date*
  - *created\_by*

# Type 0 – Implementation



# Type 1 – Overwrite

- Old attribute value in the dimension row is overwritten with the new value
- No history of dimension changes are tracked. Thus, is easy to maintain
- Aggregate fact tables and OLAP cubes affected by this change are recomputed
- Often use for data which changes are caused by processing corrections or for data which does not require history
- Examples:
  - Removal special characters
  - Correcting spelling errors

# Type 1 – Overwrite

OLTP

CustomerID	Name	CustomerType
C0001	Cust_1	Corporate



DW

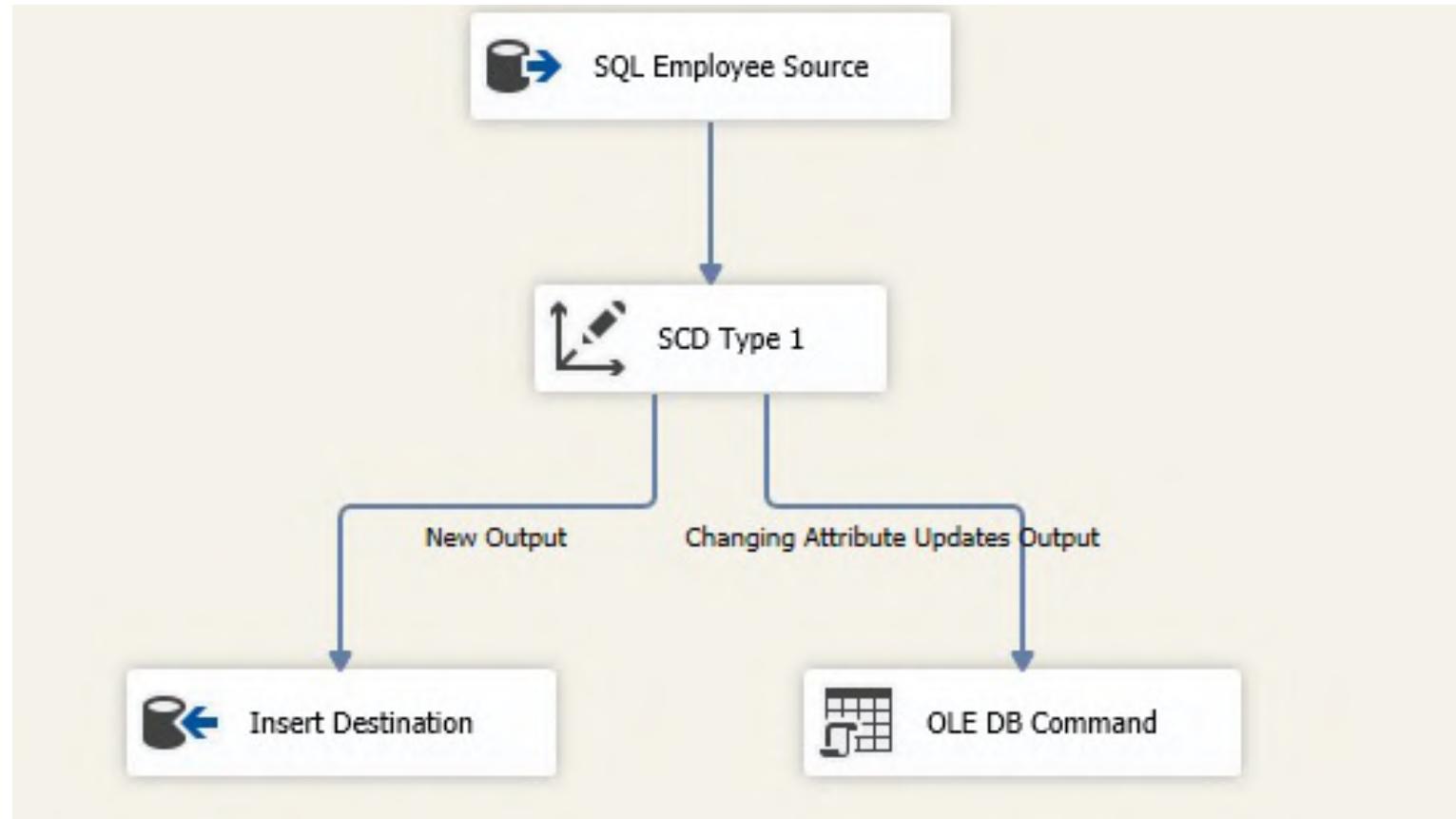
CustomerSK	CustomerID	Name	CustomerType
1	C0001	Cust_1	Corporate



CustomerSK	CustomerID	Name	CustomerType
1	C0001	Cust_1	Retail

# Type 1 – Implementation

If Exists  
    Update  
Else  
    Insert



# Type 2 – Add New Record

- Creates a new record for a given NK with a new SK
- All history of dimension changes is kept in the database
- Also *effective\_date*, *expiry\_date* and *current\_record\_indicator* columns are used in this method
- There could be only one record with current indicator set to ‘Y’
- e.g., customer address

# Type 2 – Add New Record

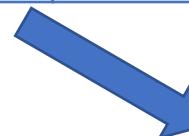
OLTP

CustomerID	Name	Address
C0001	Cust_1	Colombo



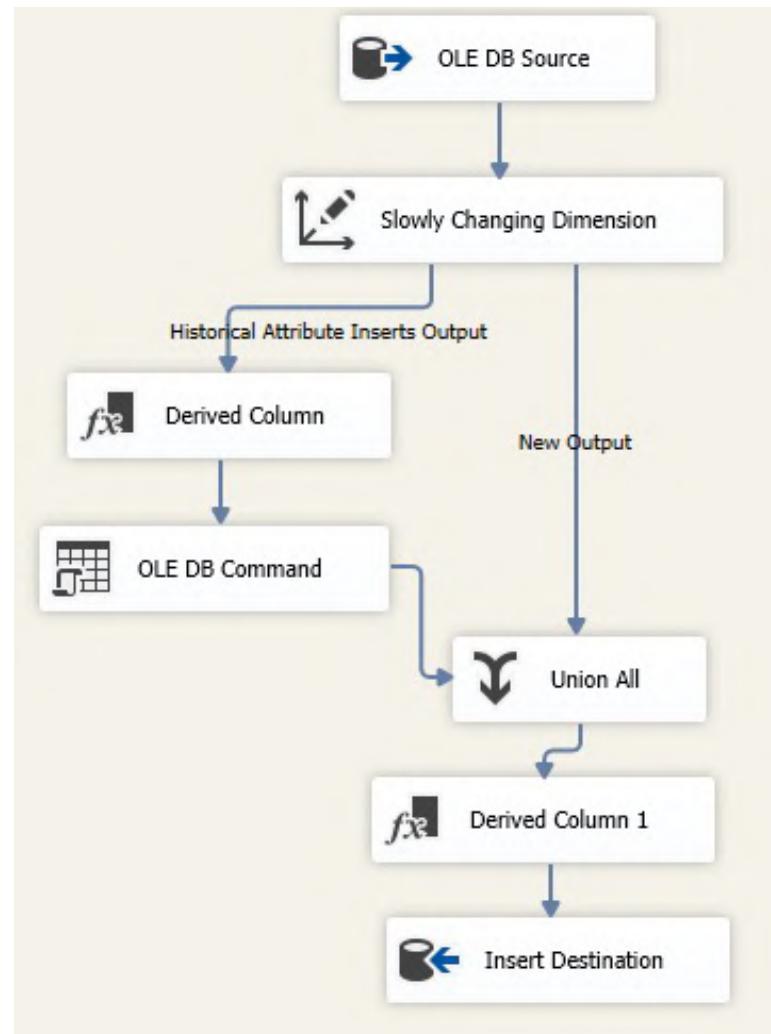
DW

CustomerSK	CustomerID	Name	Address	StartDate	EndDate	CurrentFlag
1	C0001	Cust_1	Colombo	2001-01-30	9999-12-31	Y



CustomerSK	CustomerID	Name	Address	StartDate	EndDate	CurrentFlag
1	C0001	Cust_1	Colombo	2001-01-30	2015-01-09	N
2	C0001	Cust_1	Kandy	2015-01-10	9999-12-31	Y

# Type 2 – Implementation



# Type 3 – Add New Attribute

- Tracks changes using separate columns
- Preserves limited history as it is limited to the number of columns designated for storing historical data
- Only the current and previous value of dimension is kept
- Used relatively infrequently

# Type 3 – Add New Attribute

OLTP

CustomerID	Name	CustomerType
C0001	Cust_1	Corporate



CustomerID	Name	CustomerType
C0001	Cust_1	Retail

DW

CustomerSK	CustomerID	Name	Current_CusType	EffectiveDate	Previous_CusType
1	C0001	Cust_1	Corporate	2001-03-31	Corporate



CustomerSK	CustomerID	Name	Current_CusType	EffectiveDate	Previous_CusType
1	C0001	Cust_1	Retail	2011-03-31	Corporate

# Type 4 – Add Mini-Dimension

- A separate historical table is used to track all historical changes
- The ‘main’ dimension table keeps only the current data
- Mini-dimension (history table) requires its own unique primary key
- Both surrogate keys are referenced in the fact table to enhance query performance
- e.g., *customer* and *customer\_history* tables

# Type 4 – Add Mini-Dimension

OLTP

CustomerID	Name	CustomerType
C0001	Cust_1	Corporate



DW

*Customer*

CustomerSK	CustomerID	Name	Current_CusType
1	C0001	Cust_1	Corporate

*Customer\_History*

CustomerSK	CustomerID	Name	CusType	StartDate	EndDate
1001	C0001	Cust_1	Corporate	2001-03-31	9999-12-31



*Customer*

CustomerSK	CustomerID	Name	Current_CusType
2	C0001	Cust_1	Retail

*Customer\_History*

CustomerSK	CustomerID	Name	CusType	StartDate	EndDate
1001	C0001	Cust_1	Corporate	2001-03-31	2011-03-30
1002	C0001	Cust_1	Retail	2011-03-31	9999-12-31

# Type 6 – Combine Approaches of Types 1,2,3

- Add a new record as in type 2. The *current\_type* information is overwritten with the new one as in type 1. We store the history in a *historical\_column* as in type 3.
- Dimension table contains following additional columns:
  - *current\_type*: for keeping current value of the attribute. All history records for given item of attribute have the same current value
  - *historical\_type*: for keeping historical value of the attribute. All history records for given item of attribute could have different values
  - *start\_date*: for keeping start date of ‘*effective date*’ of attribute’s history
  - *end\_date*: for keeping end date of ‘*effective date*’ of attribute’s history
  - *current\_flag*: for keeping information about the most recent record

# Type 6 – Combine Approaches of Types 1,2,3

OLTP

CustomerID	Name	Address
C0001	Cust_1	Colombo
CustomerID	Name	Address
C0001	Cust_1	Kandy
CustomerID	Name	Address
C0001	Cust_1	Galle

DW

CustomerSK	CustomerID	Name	Curr_Address	Hist_Address	StartDate	EndDate	CurrentFlag
1	C0001	Cust_1	Colombo	Colombo	2001-01-30	9999-12-31	Y



CustomerSK	CustomerID	Name	Curr_Address	Hist_Address	StartDate	EndDate	CurrentFlag
1	C0001	Cust_1	Kandy	Colombo	2001-01-30	2015-01-09	N
2	C0001	Cust_1	Kandy	Kandy	2015-01-10	9999-12-31	Y



CustomerSK	CustomerID	Name	Curr_Address	Hist_Address	StartDate	EndDate	CurrentFlag
1	C0001	Cust_1	Galle	Colombo	2001-01-30	2015-01-09	N
2	C0001	Cust_1	Galle	Kandy	2015-01-10	2018-03-31	N
3	C0001	Cust_1	Galle	Galle	2018-04-01	9999-12-31	Y

# Types of Fact Tables

---

# Fact Table Structure

- At the lowest grain, a fact table row corresponds to a measurement event and vice versa
  - Design of a fact table is entirely based on a physical activity and is not influenced by the eventual reports that may be produced
- Fact table contains:
  - Numeric measures
  - Foreign keys for each of its associated dimensions
  - Optional degenerate dimension keys
  - Date/time stamps

# Types of Measures

- **Fully additive:** measures that can be summed across any of the dimensions associated with the fact table



# Types of Measures

- **Semi-additive:** measures that can be summed across some dimensions, but not all
  - Balance amounts are common semi-additive facts: they are additive across all dimensions except date/time
  - Another example is *inventory\_level*
- **Non-additive:** measures that can not be summed across any dimensions
  - e.g., ratios

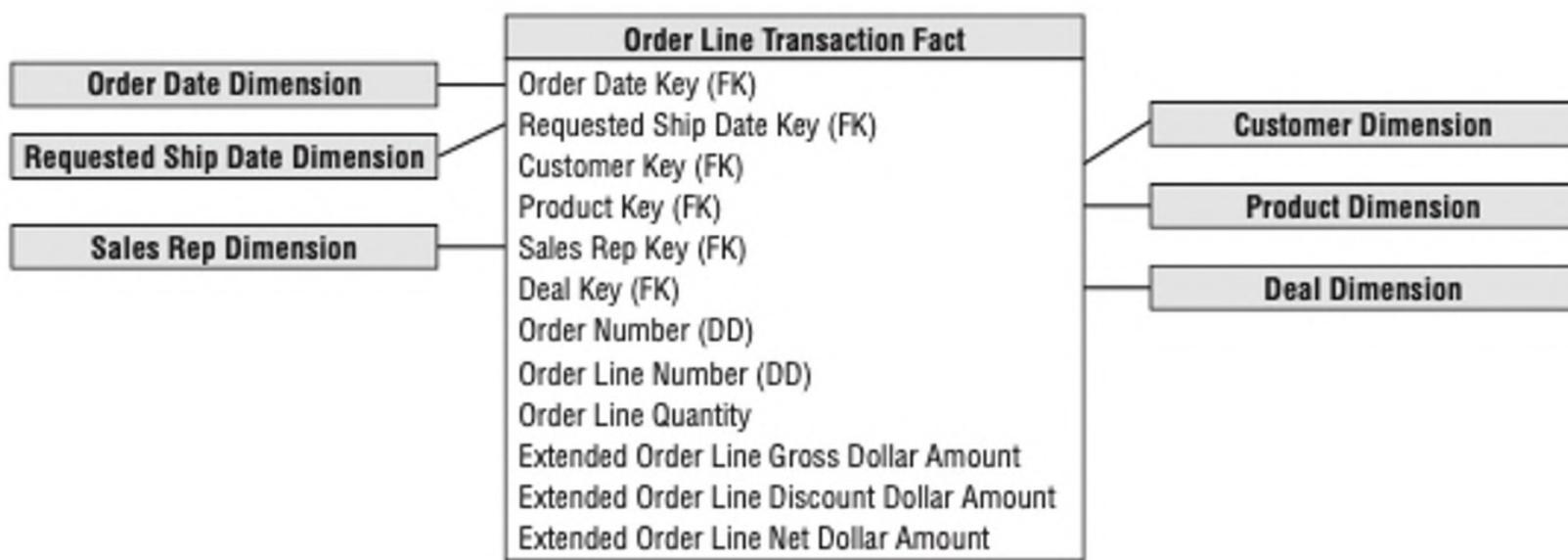
Date
Account
Current_Balance
Profit_Margin

# Handling Non-Additives

- Whenever possible, store the fully additive components of the non-additive measure and sum these components into the final answer set before calculating the final non-additive fact.
- This final calculation is often done in the BI layer or OLAP cube
  - e.g., instead of storing the *Profit\_Margin*, store *Profit* and calculate sum of *Profit\_Margin* using sums of *Balances* and *Profits*

# Transaction Fact Tables

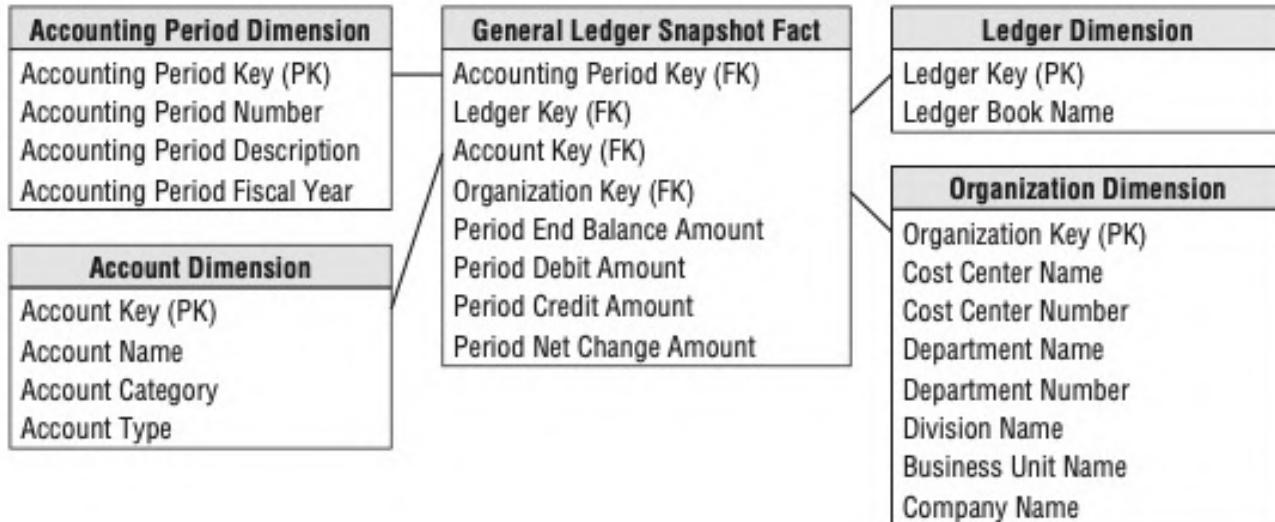
- A row in a transaction fact table corresponds to a *measurement event* at a point in space and time
  - Measures are additive



Role	Item
Foreign Key	Datetime
Foreign Key	Product
Foreign Key	Promotion
Foreign Key	Location
Degenerate	ItemID
Measure	Quantity
Measure	Tax Amount
Measure	Sale Amount
Measure	Item Total

# Periodic Snapshot Fact Tables

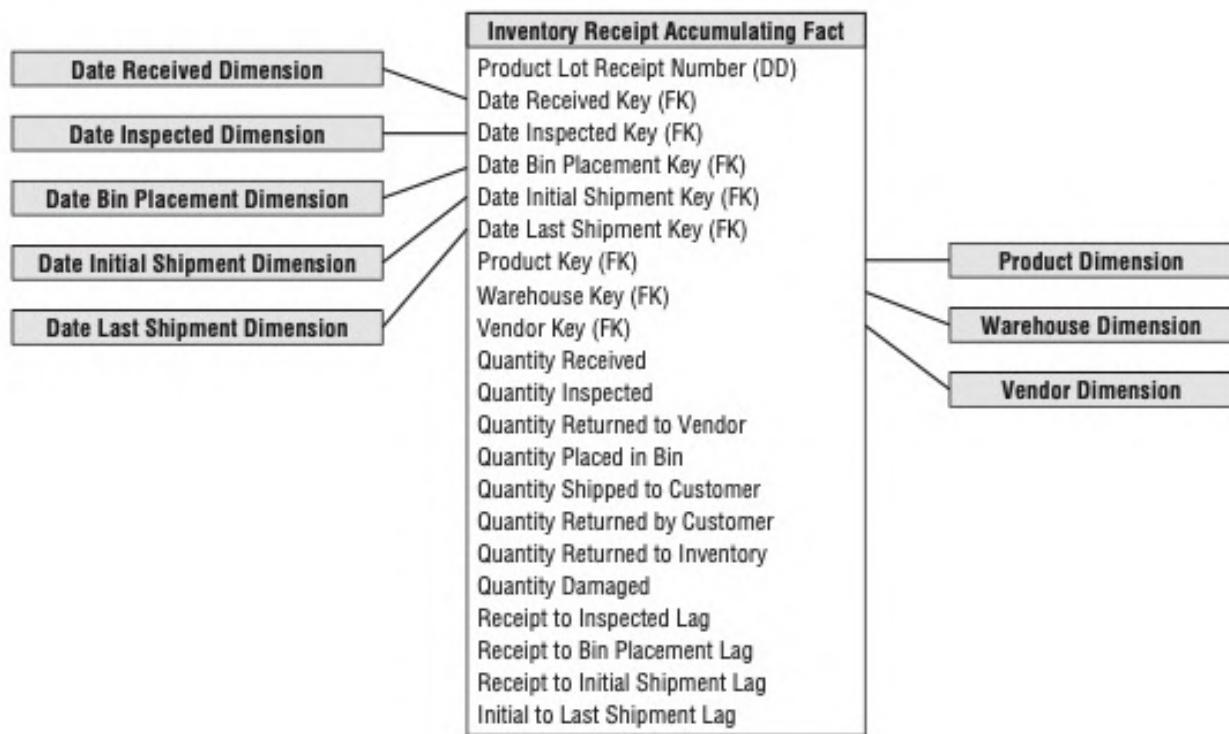
- A row in a periodic snapshot fact table summarizes many *measurement events occurring over a standard period*, such as a day, a week, or a month. The grain is the period, not the individual transaction.
  - Contains semi-additive measures



Role	Item
Foreign Key	Month
Foreign Key	Account
Foreign Key	Department
Foreign Key	Scenario
Measure	Count
Measure	Amount

# Accumulating Snapshot Fact Tables

- A row in an accumulating snapshot fact table summarizes the *measurement events occurring at predictable steps between the beginning and the end of a process.*



Role	Item
Foreign Key	ClaimDate
Foreign Key	PaidDate
Foreign Key	Claim
Foreign Key	Insurer
Foreign Key	Location
Measure	Reserved
Measure	Paid
Measure	Recovered
Measure	Balance

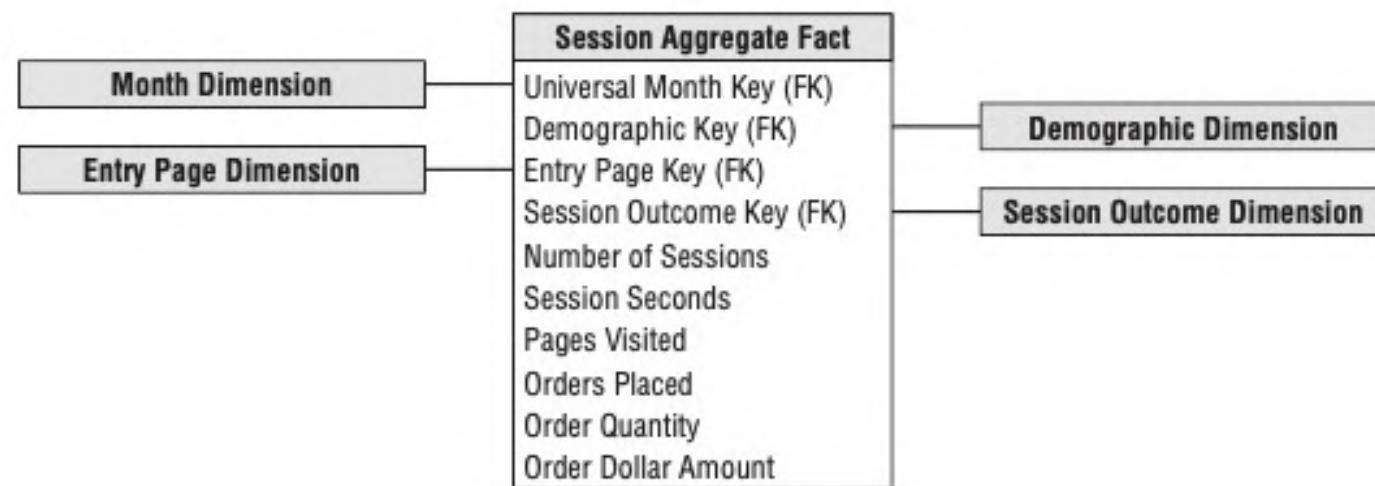
# Factless Fact Tables

- It is possible that a business event merely records a set of dimensional entities coming together at a moment in time.
  - e.g., a student attending a class on a given day may not have a recorded numeric fact, but a fact row with foreign keys for calendar day, student, teacher, location, and class is well-defined.



# Aggregate Fact Tables

- Aggregate fact tables are simple numeric rollups of atomic fact table data built solely to accelerate query performance.
  - Semantic layer aggregate tables
  - OLAP Cubes
  - e.g., aggregate clickstream fact table



**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

Lecture 05

Data Ingestion Flows

# Data Integration

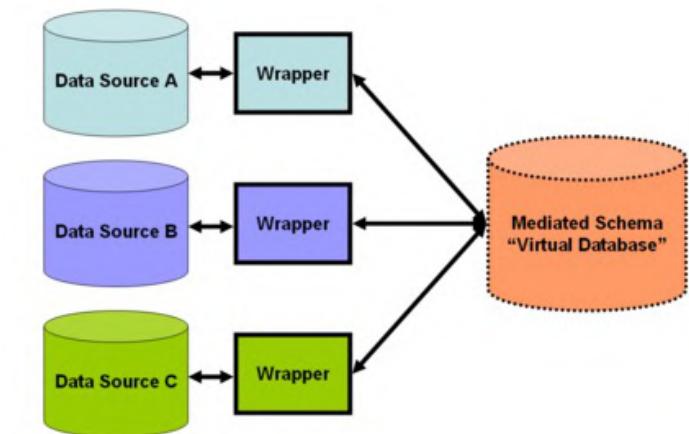
# What is Data Integration?

- Data integration is consolidation of business information or data sets from **various sources**, maybe in **various formats**, and providing users with a **unified view**.

# Data Integration Techniques

- Data Federation

- Provides a single virtual view of one or more data source
- No physical data movement
- Pulls data from source systems on an on-demand basis
  - A query against the virtual view, retrieves data from source systems, integrates the result and sends to the requesting application/user
    - Hence, real time!
    - Required data transformations are done dynamically
    - These details abstracted from the application/user
  - Application/user sees only the virtual view (mediated schema)
  - Implementation approach:
    - Enterprise Information Integration (EII)

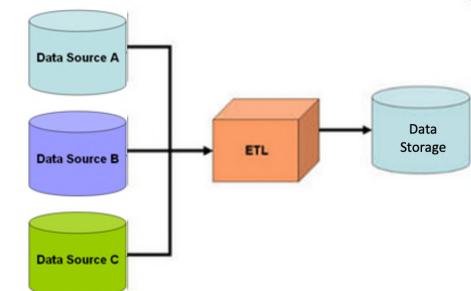


# Data Integration Techniques

- Data Propagation
  - Usually operate online and push data to the target location
    - Event-driven
  - Updates in a source system may be propagated asynchronously or synchronously to the target system
    - Synchronous propagation requires that updates to both source and target systems occur in the same physical transaction
  - Regardless of the type of synchronization, data propagation guarantees the delivery of the data to the target
    - This guarantee is a key distinguishing feature of data propagation
  - Implementation approaches:
    - Enterprise Application Integration (EAI): applications copy data from one location to another
    - Enterprise Data Replication (EDR): different mechanisms available

# Data Integration Techniques

- Data Consolidation
  - Data from multiple source systems are integrated into a single persistent data store
    - Used for reporting, analysis
    - Act as a data source for downstream applications
    - e.g., Data Warehouse, Operational Data Store, Data Mart, Data Lake
  - Usually there is a delay/latency for the updates in sources to appear in the target store
    - Latency depends on business needs; seconds, hours, days, etc.
    - Data with zero latency is known as real-time data
      - Difficult to achieve using data consolidation (big data tools & technologies used)
  - Implementation approach:
    - Extract-Transform-Load (ETL)
    - Extract-Load-Transform (ELT)



# ETL Process

Source Systems



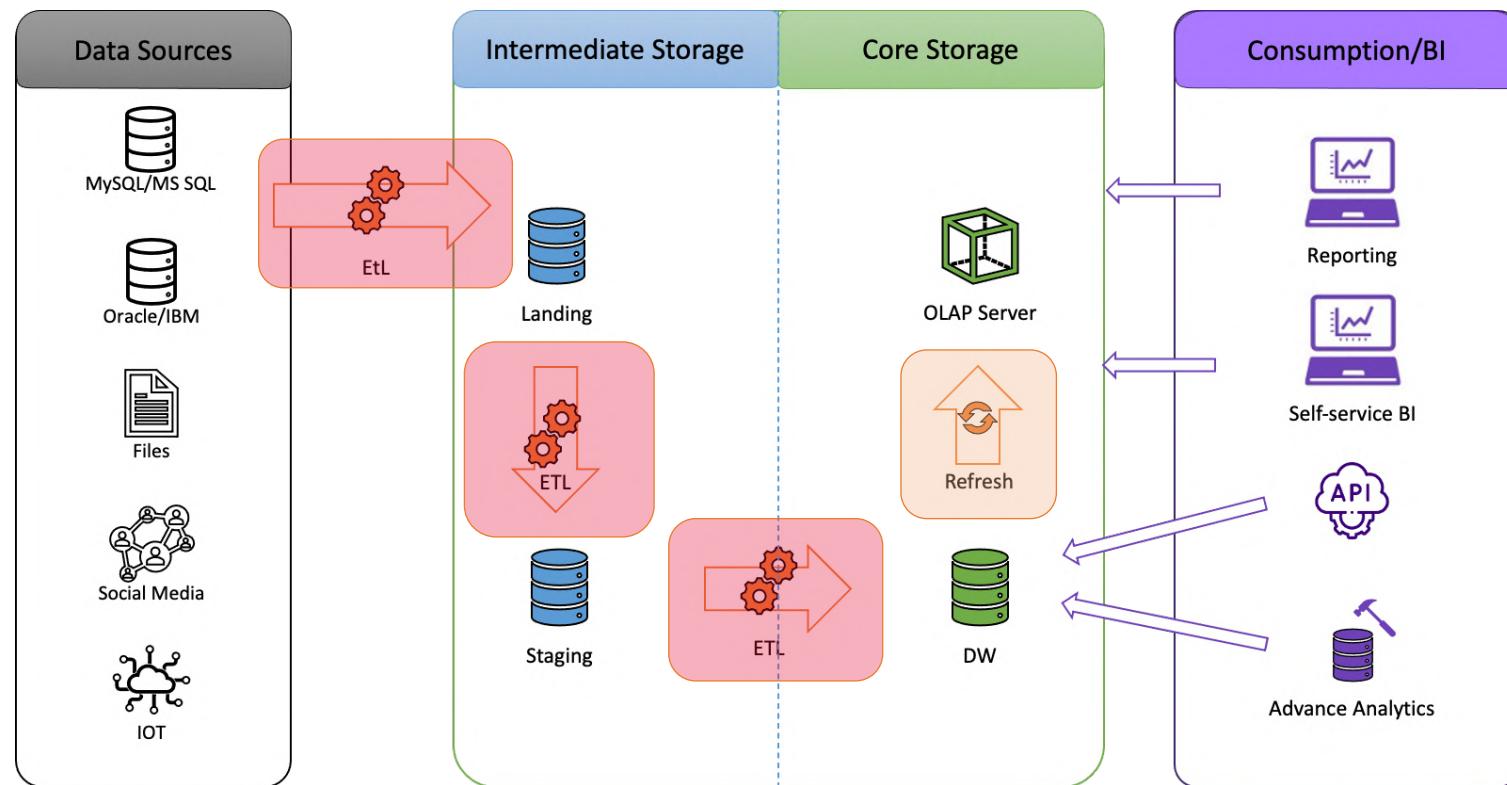
Extract, Transform, Load

Destination  
(Data Warehouse)



# ETL at Several Levels

- Many ETL layers due to intermediate storage layers
  - Intermediate storage layers are used to minimize dependency and minimize impact on source systems



# ETL Considerations

- Different types of sources:
  - Database systems
  - Files (.csv, .txt, .xlsx, etc.)
  - Documents (JSON, XML, etc.)
  - Legacy systems
  - APIs
- Data source connectivity and security concerns
- Different frequency requirements:
  - Near real-time, hourly, daily, weekly, etc.

# ETL Considerations

- Extraction window limitations:
  - Small time frames to cater near real time requirements
  - Probably off-peak times for scheduled batch extractions
- Source data realities
  - Data quality issues
  - Extraction window limitations
- Capabilities of tools used
  - Support of traditional ETL tools for real-time extractions

# First, Understanding Data!!!

- Before data extraction, need to understand source data
- **Data Profiling** to understand data:
  - Profiling is the technical analysis of data: describe its content, consistency, and structure
  - Strategic decisions: whether to include certain data or not in the data warehouse
  - Tactical decisions: find data issues as much as possible, and try to resolve
  - Issues found will be:
    - Notify/sent back to the originator of the source for correction
    - Resolve as a part of the data quality process

# Data Extraction

---

# Logical Extraction Methods

- **Full extraction**
  - Extract all the data, every time!
  - Some dimensions can be flushed and loaded based on the complexity and requirements
- **Incremental extraction**
  - Extract only the changes/updates
  - Most dimensions will be loaded with latest changes allowing the maintenance of historical information
  - Fact table data

# Physical Extraction Methods

- **Online extraction**
  - Data extracted directly from the source systems itself
- **Offline extraction**
  - Data staged outside the original source system
  - Source system push the data into a sperate storage in some format
    - Flat files
    - Dump files
  - Extraction process access this storage

# Push versus Pull

- **Push**

- A process in the source system creates a snapshot of changes delivers to downstream systems
- Mostly occurs in incremental extraction and offline extraction

- **Pull**

- Downstream system connects to the source, request to get data from the source
- Mostly occurs in online extraction
- Could be used with both full and incremental extraction

# Incremental Extraction: Capturing Changes

- Change Data Capture (CDC): the process of capturing changes occurred at the source systems in order applying them throughout the enterprise
- CDC minimizes the resources required for ETL processes
  - Because it only deals with data changes

Source table				
Client_ID	FName	LName	DoB	MStatus
135001	John	Doe	1/12/1960	married
135002	John	Doe Junior	12/31/1985	single
135003	John Q.	Public	5/6/1985	divorced
135004	Judy	Doe	10/22/1980	single
135005	Jane	Roe	8/16/1982	married

Source table				
Client_ID	FName	LName	DoB	MStatus
135001	John	Doe	1/12/1960	married
135003	John Q.	Public	5/6/1985	divorced
135004	Judy	Smith	10/22/1980	married
135005	Jane	Roe	8/16/1982	married
135006	Joe	Bloggs	7/14/1970	single

Target table							
ClientID	FirstName	LastName	DoB	MStatus	DateBegin	DateEnd	St_Code
135001	John	Doe	1/12/1960	married	2/1/2011		I
135002	John	Doe Junior	12/31/1985	single	2/1/2011		I
135003	John Q.	Public	5/6/1985	divorced	2/1/2011		I
135004	Judy	Doe	10/22/1980	single	2/1/2011		I
135025	Jane	Roe	8/16/1982	married	2/1/2011		I
135004	Judy	Smith	10/22/1980	married	2/2/2011		U
135006	Joe	Bloggs	25763	single	2/2/2011		I

Figure 1. Initial Load

Target table								
ClientID	FirstName	LastName	DoB	MStatus	DateBegin	DateEnd	St_Code	
135001	John	Doe	1/12/1960	married	2/1/2011		I	
135002	John	Doe Junior	12/31/1985	single	2/1/2011		D	
135003	John Q.	Public	5/6/1985	divorced	2/1/2011		I	
135004	Judy	Doe	10/22/1980	single	2/1/2011		I	
135025	Jane	Roe	8/16/1982	married	2/1/2011		I	
135004	Judy	Smith	10/22/1980	married	2/2/2011		U	
135006	Joe	Bloggs	25763	single	2/2/2011		I	

Figure 2. Incremental Load

# Capturing Changes with CDC

- Goals of CDC:
  - Capture all changes made to source data
  - Isolate the changes to allow selective processing
  - Tag changes with reason codes to distinguish error corrections from true updates
  - Support compliance tracking
- **Note:** CDC is not required for the initial data load

# Change Data Capture: Approaches

- **Full-diff compare**
  - Uses a snapshot of yesterday's data and compares with today's snapshot to find changes on a record by record and field by field basis
  - A thorough approach but resource intensive
- **CRC-code (Cyclic Redundancy Checksum)**
  - Identifies the changes using the CRC code
  - Recognizes the changes more efficient than the field by field compare method

# Change Data Capture: Approaches

- **Database log scrapping**
  - Takes a snapshot of the database log and extracts modified records
- **Delta file**
  - Contains all the data modifications that were made by applications during business operational activities
  - Source systems should generate this files
  - CDC process become very easy and efficient
  - A dependency on the source system!

# Change Data Capture: Approaches

- **Trigger method**

- Adding triggers to source tables whose changes should be controlled
- Triggers are turned on for insert, update or delete operations and record these modifications in a specific tables or files; then read the modification from these tables or files

- **Message queue monitoring**

- Used in a message-based operational/transactional systems
- Low overhead
- Monitors the message queue for all the operations/transactions

# Change Data Capture: Approaches

- **Timed extracts**
  - Typically selects all rows where the source record *create* or *modified* date-time values are greater than or equal to last extracted date-time
    - *SYSDATE – 1* for a daily ETL; but need to ensure failed jobs are retriggered
  - Manual intervention is required if the process fails to perform clean-ups
  - Tend to have duplicates if the process fails
    - Must be addressed before loading them into the target

# Transformations

---

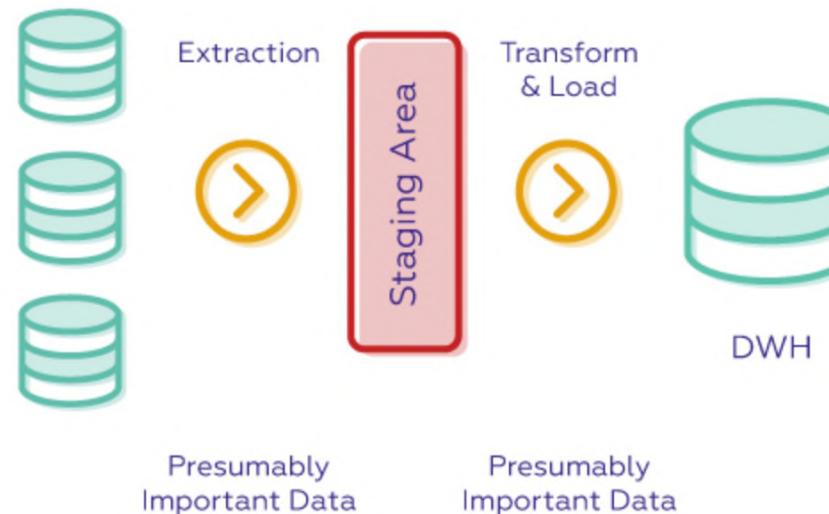
# Purpose

- Transform data stored in different data structures to a uniform data structure
- Apply data cleansing approaches to deal with data issues
- Apply business logic to generate expected data set that needs to be stored in data warehouse
- Add value to data by enhancing raw data
- Can be architected to generate metadata that can be used to diagnose what's wrong with source systems
  - Thus, provide suggestions to source systems to correct erroneous data within the source

# Transformation Techniques

- **Multistage data transformation (ETL)**

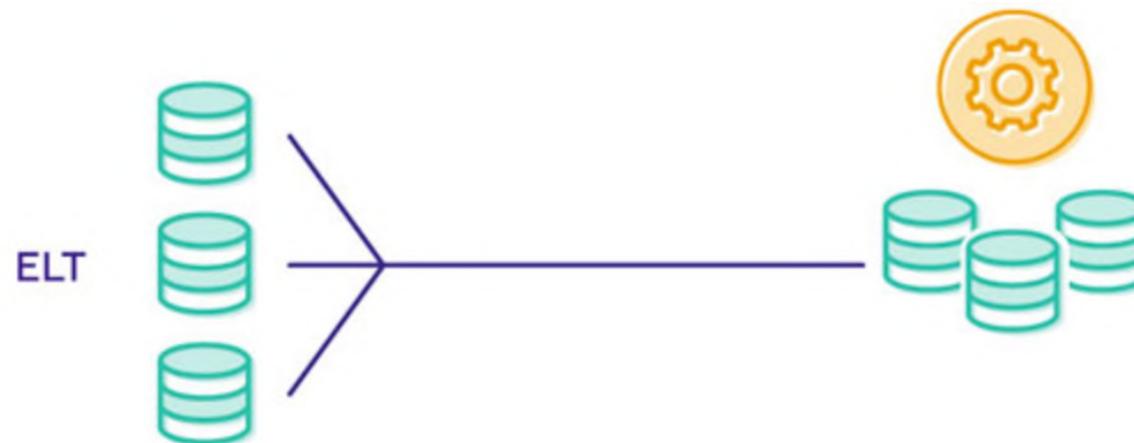
- This is the classic extract, transform, load process
- Extracted data is moved to a staging area where transformations occur prior to loading the data into the warehouse



# Transformation Techniques

- **In-warehouse data transformation (ELT)**

- Data is extracted and loaded into the analytics warehouse, and transformations are done there
- Utilize the increased performance and scalability of the modern analytics databases
- ELT tools provide ability to write in-database transformations to be written in SQL



# ETL vs. ELT

Parameter	ETL	ELT
Process	Transformation of data happens at the staging server and then loaded to the data warehouse	Data is loaded into the data warehouse and then transformation takes place
Data size vs. complexity of transformations	ETL is best suited for dealing with smaller data sets that require complex transformations	ELT is best when dealing with massive amounts of structured and unstructured data
Added calculations	Calculations will either replace existing columns, or append the dataset to push them to the target data system	ELT adds calculated columns directly to the existing dataset
Implementation complexity	Highly evolved ETL tools are available and experts are easy to find	Tools are still evolving and experts with the right knowledge and skills can be difficult to find

# ETL vs. ELT

Parameter	ETL	ELT
Waiting time to load information	<p>ETL load times are longer than ELT because it's a multi-stage process:</p> <ol style="list-style-type: none"><li>1. data loads into the staging area</li><li>2. transformations take place</li><li>3. data loads into the data warehouse</li></ol> <p>Once the data is loaded, analysis of the information is faster than ELT</p>	<p>Data loading happens faster because there's no waiting for transformations and the data only loads one time into the target data system. However, analysis of the information can be slower than ETL if not pre-prepared</p>
Availability of data in the system	<p>Transforms and loads the data that you decide as necessary during the design. Thus, only this information will be available</p>	<p>ELT can load all data immediately, and users can determine later which data to transform and analyse</p>
Adoption of the technology and availability of tools and experts	<p>Well-developed process used for over 20 years and experts are readily available</p>	<p>As this is a new technology it can be difficult to locate experts and more challenging to develop</p>

# ETL vs. ELT

Parameter	ETL	ELT
Costs	On-prem tools are mostly commercial tools and most of them are expensive. Cloud based tools are available with pay-per-usage models	Open source tools are also available, Can load and save data without transforming, then apply transformations as needed which will save money if pay-per-usage cloud services are used
Hardware requirements	Legacy, onsite ETL processes have extensive and costly hardware requirements, but they are not as popular today	ELT processes don't require special hardware and there are many cloud-based solutions too
Compliance	ETL can remove sensitive information before putting it into the data warehouse. Hence, it is easier to satisfy compliance standards. It also protects data from hacks and inadvertent exposure	ELT requires data to be uploaded to the target before removing sensitive information. This could violate compliance standards. Sensitive information will be more vulnerable to hacks and inadvertent exposure

# ETL vs. ELT

Parameter	ETL	ELT
Unstructured data support	ETL can be used to structure unstructured data, but it can't be used to pass unstructured data into the target system	ELT is a solution for uploading any type of data (unstructured, semi-structured and structured) into a data lake and make them data available to business intelligence systems
Compatibility with data lakes	Generally transforms data for integration with a structured relational data warehouse system	ELT offers a pipeline for data lakes to ingest unstructured data and then transforms the data on an as-needed basis for analysis

# Transformations

- Application of business logic
  - Calculation
  - Derived columns (e.g.,  $sale\_amount = qty \times unit\_price$ )
  - Joining and splitting (e.g., linking data from multiple sources, splitting comma separated values)
- Data cleansing
  - Deduplication
  - Handling NULL values
  - Format unit of measure
  - Format texts
    - Change text to lower, upper or proper case
    - Get rid of extra spaces
  - Data conversion such as converting numbers stored as text into numbers
  - Spell check

# Transformations

- Data filtering
  - Selection of required columns
  - Filter unwanted outliers
  - Filter records based on quality or requirement
- Data enrichment
  - Value added data
    - Full name to first, middle and last name
    - Adding social data with available data
    - Adding up all purchases a customer has made to build a customer lifetime value (CLV) metric
- Data validation
  - Simple or complex data validation
    - For example, if the first three columns in a row are empty then reject the row from processing

# Transformations

- Generating surrogate-key values
- Look-ups
- Transposing or pivoting
- Aggregation
  - Data elements are aggregated from multiple data sources and databases
  - for example, rollup; summarizing multiple rows of data — total sales for each store, and for each region, etc.

# Data Quality

- Handling missing data:
  - Dropping records that have missing values is one approach
    - However, this is sub-optimal because when you drop records with missing data, you drop potentially valuable information
    - The fact that the values were missing may be informative in itself
    - In the real world, you often need to make predictions on new data even if some of the features are missing!
  - Generating missing values based on other observations whenever possible is another approach

# Data Quality

- Handling missing **categorical** data
  - Generate missing values using a business logic if possible
  - Else, the best way to handle missing data for categorical features is to simply label them as '*missing*'!
  - This is essentially adding a new class (category) for the feature (attribute)
  - This tells the algorithm that the value was missing
  - This also gets around the technical requirement to handle missing values

# Data Quality

- Handling missing numeric data
  - Generate missing values using a business logic if possible
  - Else, for missing numeric data, flag and fill the values
    - Flag the records with an indicator
    - Then, fill the original missing value with a suitable numeric value (which is not used as actual data) just to meet the technical requirement to handle missing values
  - By using this technique of flagging and filling, you are essentially allowing algorithms to estimate the optimal constant for missingness (e.g., in ML), instead of just filling it in with the mean value based on available data

# Data Quality Systems

- There are dedicated data quality tools to cleanse data
  - Goal is to offer a comprehensive architecture for cleansing data, capturing quality events as well as ultimately controlling data quality
  - Not only limited for data warehousing solutions
- Should start by profiling data to understand the depth, we should address
- These systems can provide specific descriptions of data errors, expected to be found in ETL processes
- These systems can be integrated with ETL process to cleanse data are a separate task

# Data Quality Validation

- Quality validation tasks ensure data provided through ETL/ELT processes are in acceptable level
  - Can be considered as a testing task within ETL process
- Two main tasks
  - **Quality screens**
    - Column screen: checks data within a single column (e.g., data type)
    - Structure screen: test relationships of data across columns (e.g., constraints such as keys)
    - Business rule screen: checks data by relating them to business rules
  - **Respond to quality events**
    - Halting the process
    - Sending the error records to a suspense file for later processing
    - Tagging the record with an error code and passing it to the next step

# Data Loading

---

# Data Loading Concerns

- Order of data loading
  - Dimensions vs. facts
- Surrogate key generation
- Hierarchy management
- Loading data into special dimensions
  - Slowly changing dimensions
- Fact table loading
  - Transaction vs. periodic snapshot vs. accumulating fact tables
- Loading aggregate tables

# Loading into Dimension Tables

- Dimensions provide different aspects to analyse measures in fact tables
- They are usually smaller in size compared to fact tables
- Dimensions are the first to be loaded with latest data before updating the fact tables
  - Fact tables contains references to dimension tables using surrogate keys
- Dimension tables could be loaded in different ways
  - Fresh load
  - delta load
  - Follow different SCD approaches

# Loading Slowly Changing Dimensions

- Based on the way slowly changing dimensions are handled, data could be loaded in different ways
  - Type 1: update existing records
  - Type 2: add new record and update existing records to expire it
  - Type 3: update existing record with new value and old value
  - Type 4: update main dimension mini-dimension appropriately
- Most commercial mature ETL tools handle data loading to SCD tables
  - We just have to configure the ETL flow as we want!

# Handling Surrogate Key

- Primary key of dimension tables are called which is a separate integer based auto incrementing column
  - Separated from business/natural keys being received from source systems
  - the keys in the transaction source systems
- When loading data to dimension tables, surrogate key must be,
  - auto populated by the data warehouse, or
  - inserted by the ETL flow
- When loading data to fact tables correct surrogate key in referencing dimension tables must be looked up and inserted in to the fact tables' reference columns

# Loading into Fact Tables

- Fact tables are usually larger in size compared to dimension tables and thus, only changes (modifications of the source systems) are loaded
- Based on the type of the fact table, different loading approaches will be used
  - Transaction: mostly insert new transactional records
  - Periodic snapshot: mostly insert new aggregated records based on some date period
    - Daily snapshot, weekly snapshot
  - Accumulating fact tables: insert new events and update existing events
- Based on the behaviour of source systems, different loading approaches may have to be used
  - If source system updates old transactional records, a similar approach may have to be followed when loading transactional data to fact tables
- Foreign keys must be properly dealt with using surrogate keys
  - Correct surrogate key in relevant dimension table must be referenced (looked up) and inserted to the fact table

# Loading into Aggregations

- Aggregate tables improve the performance of the data warehouse (specially when the data volume is high) when data is being consumed by BI layer components such as reports, dashboards
  - Rather than performing required aggregation calculation dynamically to generate the report/dashboard, these aggregate (summarized) values are pre-calculated and stored in the data warehouse
- ETL tool populates and maintains aggregate fact tables
  - Once the base fact tables are loaded, aggregate tables can be populated
- Fastest way is to incrementally load the aggregate tables
  - May require a delete-insert approach for some records
- Aggregates must always be consistent with base data (main fact table)

# Process Flow Automation

---

# Ochestrator and Automation Tools

- Every enterprise DW should have a robust ETL orchestration and automation tool(s) that manage the entire ETL process through a metadata-driven job control environment
- It should be aware of and control the relationships and dependencies between ETL jobs/tasks
- Should support a fully automated process
  - This includes managing ETL processes during failures
- In general, orchestration and automation tools should support following features:
  - Task orchestration: define relationships and dependencies between ETL jobs/tasks
  - Triggering of ETL jobs: executing ETL jobs based on a set date-time or based on some other trigger such as a file
  - Metadata capture
  - Logging
  - Notifications

**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

Lecture 06

QA in Data Warehousing

# Content

- Importance of testing
- ETL/ELT testing
  - ETL/ELT testing types
  - ETL bugs
  - ETL test automation
  - Challenges in ETL testing
- DW performance testing
- DW security testing
- BI testing
- Testing process

# Importance of Testing

- Data drives critical business decisions; thus, testing the data warehouse data integration process is essential
- Data comes from numerous sources; hence, data sources affect data quality.
- Source data history, business rules, or audit information may change; testing must be an ongoing process
- In the ETL process, data flows through a pipeline before reaching the data warehouse. Thus, entire ETL pipeline must be tested to ensure each type of data is transformed or copied as expected
- Data must be available on a timely manner and be consistent
- Only relevant people/applications must have access to data in the data warehouse and also only the minimum required level of access should be given
- BI layer must ensure the correctness of data presented; this is the front end of data warehousing layer for business users

# Types of Testing in Data Warehousing

- ETL/ELT testing
- DW performance testing
- DW security testing
- BI testing

# ETL/ELT Testing

- Ensures ETLs run as intended
- ETL in different stages must be tested
- Data completeness/quality testing
  - Data at every entry/exit point must be tested
    - If not, the data that is harmed, altered inadequately or lost on the way, will result in misleading insights and cause faulty business decisions

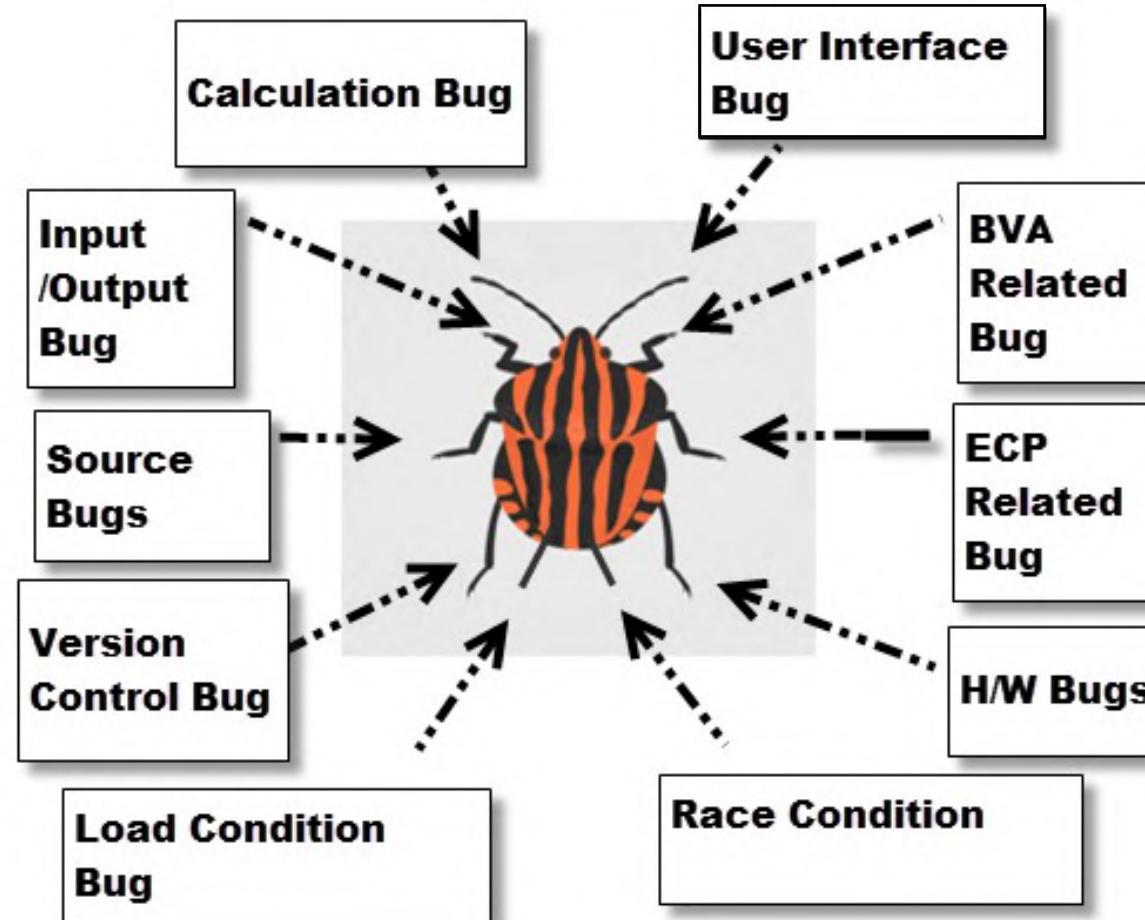
# ETL/ELT Testing Types

- Metadata testing: ensures that the table definitions conform to the data model and application design specifications
  - Tests: table structure check, data type check, data length check, and index/constraint check, etc.
- Data completeness testing: validates that all the expected source data has been successfully loaded to the target
  - Tests: compare and validate counts, aggregates (min, max, sum, average), and actual data between the source and destination, rejected record report
- Data quality (correctness) testing: ensures data quality issues are handled as intended
  - Tests: NULL replacements checks, duplicate check, etc.
- Data transformation testing:
  - White box testing: review the transformation logic from the mapping design document and the ETL code. This is used to create test cases. (data enrichment, business logic, record create/modify dates, SCD records, etc.)
  - Black box testing: examines the functionality of an application without looking at internal structures for transformation testing; this involves reviewing output of ETL process against defined test cases

# ETL/ELT Testing Types

- ETL regression testing: validates that the ETL produces the same output for a specific input before and after the change
- Incremental ETL testing: verifies that updates on the sources are getting loaded into the target system correctly
- ETL integration testing: end-to-end testing of data flows through the ETL process from all of the source systems to target application (dw/semantic layer)
- ETL performance testing: end-to-end testing to ensure that the all steps in the ETL process are working with expected data volumes
  - One pitfall of this testing method is the lack of actual data to emulate appropriate volumes

# Types of ETL Bugs



# Types of ETL Bugs

Types of Bug	Description
User interface bugs/cosmetic bugs	Related to GUI of application Font style, font size, colours, alignment, spelling mistakes, navigation and so on
Boundary Value Analysis (BVA) related bug	Minimum and maximum values
Equivalence Class Partitioning (ECP) related bug	Valid and invalid type
Input/Output bugs	Valid values not accepted Invalid values accepted
Calculation bugs	Mathematical errors Final output is wrong

# Types of ETL Bugs

Types of Bug	Description
Load Condition bugs	Does not allows multiple users Does not allows customer expected load
Race Condition bugs	System crash & hang System cannot run client platforms
Version control bugs	No logo matching No version information available This occurs usually in regression testing
H/W bugs	Device is not responding to the application
Help Source bugs	Mistakes in help documents

# Automation of ETL Testing

- General methodology of ETL testing:
  - SQL scripting
  - Do “eyeballing” of data.
- These approaches to ETL testing are time-consuming, error-prone and hardly provide complete test coverage
- To accelerate, improve coverage, reduce costs, improve defect detection ratio of ETL testing in production and development environments, automation is the needed
- Further, Some testing needs to be done iteratively with each ETL execution.

# Challenges in ETL Testing

- Different types of formatted data
- Data of Heterogeneous sources
- Huge volume of data
- Lack of testing tools
- Lack of testing knowledge/ and experience
- Lack of requirements clarity
- Missing values in large volume of data
- Outdated data warehousing tools
- DW systems could have different architectures
- Complex data warehousing architectures with many ETL tools and components
- Automation of testing will impact on ETL execution performance

# Data Warehouse Performance Testing

- Detects possible bottlenecks
- Ensures data warehouse handles the increasing number of user requests
  - Performing various operations simultaneously
- Ensures data warehouse handles the growing data volume from data sources

# Data Warehouse Security Testing

- Ensures the correctness of role-based access
- Ensures the correct work of data encryption and decryption implementations
- Ensures the validity of data backup operations (if applicable)

# Testing in Business Intelligence Layer

- Ensures OLAP operations work correctly
  - Roll-up
  - Drill-down
  - Slicing
  - Dicing
  - Pivot
- Ensures data presented in reports and dashboards are accurate
- Ensures rendering time of reports and dashboards are in agreement with requirements
- Ensures the permissions set in reports and dashboards are in line with the requirements

# Testing Process

- Business requirements gathering
  - Helps to test the accuracy of the data model, define business flow requirements and assess reporting needs based on client expectations
  - It's important to start here so the scope of the project is clearly defined, documented, and understood fully by testers
- Validate data sources
  - Perform a data count check and verify that the table and column data type meets specifications of the data model
  - If not done correctly, the data in data warehouse, semantic layer, reports could be inaccurate or misleading
- Design test cases
  - Design ETL mapping scenarios, create SQL scripts, and define transformational rules to be executed in parallel with ETL jobs
  - It is important to validate the mapping document as well, to ensure it contains all of the information
- Extract data from source systems
  - Execute ETL tests per business requirement; identify types of bugs or defects encountered during testing and make a report
  - It is important to detect and reproduce any defects, report, fix the bug, resolve, and close bug report before continuing to next step

# Testing Process

- Apply transformation logic (ETL flow testing)
  - Ensure data is transformed to match schema of target data warehouse
  - Check data threshold, alignment, and validate data flow
  - This ensures the data type matches the mapping document for each column and table
- Load data into target data warehouse
  - Perform a record count check before and after data is moved from staging to the data warehouse
  - Confirm that invalid data is rejected and that the default values are accepted
- BI report/dashboard testing
  - Verify layout, options, filters and export functionality of BI reports/dashboards
- Performance testing
- Security testing
- Test Closure
- File test closure

**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

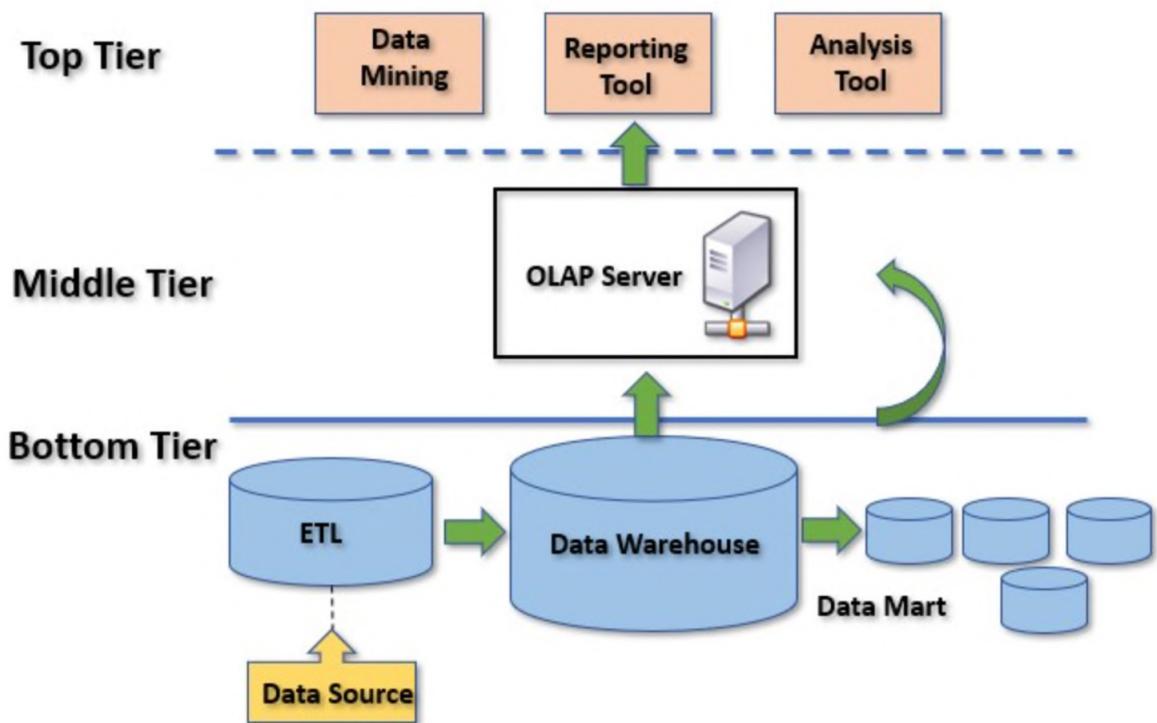
Lecture 07

OLAP Cubes

# Content

- What are OLAP cubes?
- OLAP operations
- Aggregations in OLAP cubes
- Storage types
- Introduction to SSAS

# Role of OLAP in Data Warehousing



- **Top-Tier:**
  - The top tier is a front-end client layer
- **Middle Tier:**
  - The middle tier is an OLAP server
  - For a user, this application tier presents an abstracted view of the database
  - Acts as a mediator between the end-user and the database
- **Bottom Tier:**
  - The database of the data warehouse servers as the bottom tier

# OLAP Cubes

- A method for storing data in a multi-dimensional form
- Enables users to analyse multidimensional data (called facts) interactively (by browsing and querying) from multiple perspectives (called dimensions)
- Pre-calculates most of the queries that are typically time taking to execute over tabular databases such as aggregation, joining and grouping
  - A data structure optimized for very quick data analysis
- Specially optimized for OLAP operations:
  - Roll-up
  - Drill-down
  - Slice
  - Dice
  - Pivot

# Representing Multi-dimensional Data

- Table vs. matrix

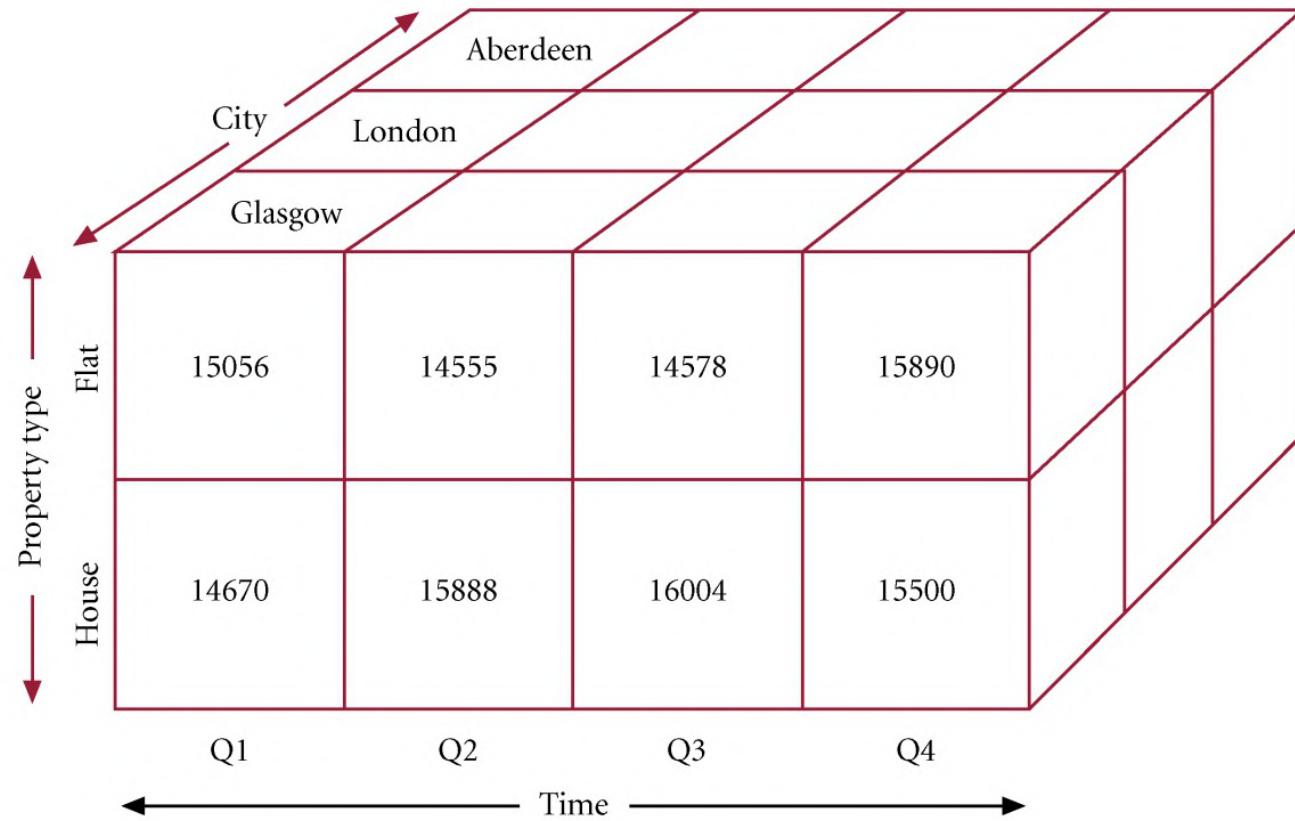
City	Time	Total Revenue
Glasgow	Q1	29726
Glasgow	Q2	30443
Glasgow	Q3	30582
Glasgow	Q4	31390
London	Q1	43555
London	Q2	48244
London	Q3	56222
London	Q4	45632
Aberdeen	Q1	53210
Aberdeen	Q2	34567
Aberdeen	Q3	45677
Aberdeen	Q4	50056
.....	.....	.....
.....	.....	.....

City	Glasgow	London	Aberdeen	.....
Quarter				
Q1	29726	43555	53210	.....
Q2	30443	48244	34567	.....
Q3	30582	56222	45677	.....
Q4	31390	45632	50056	.....

# Representing Multi-dimensional Data

- Table vs. cube

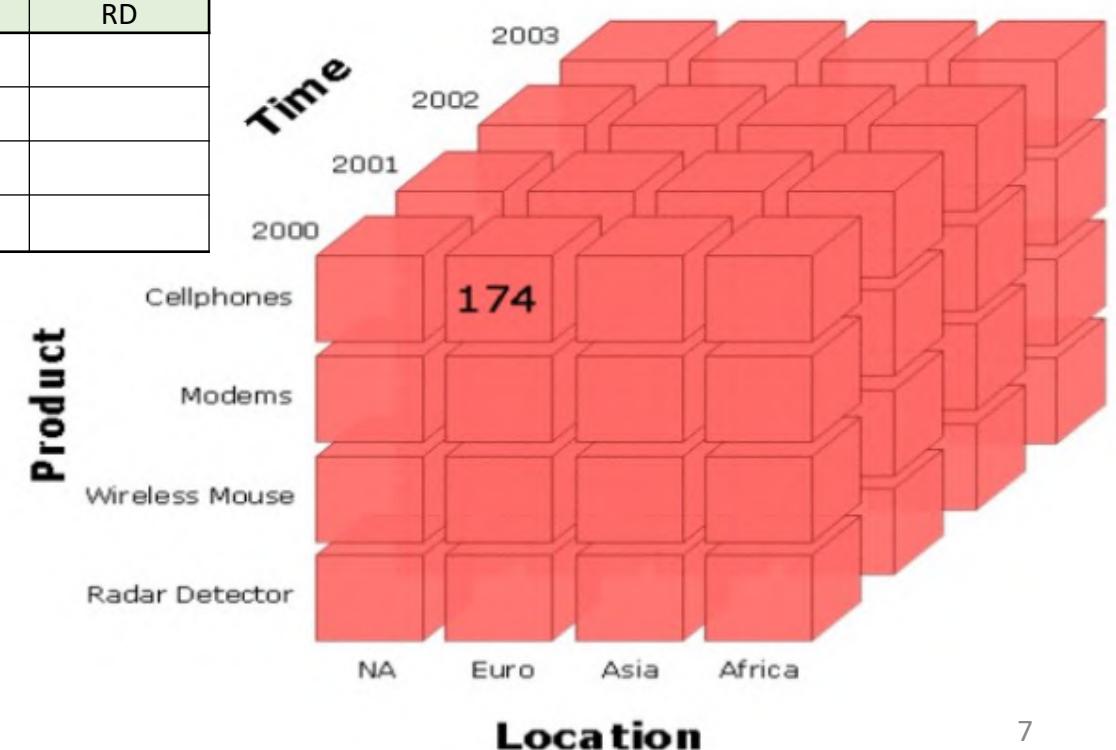
Property Type	City	Time	Total Revenue
Flat	Glasgow	Q1	15056
House	Glasgow	Q1	14670
Flat	Glasgow	Q2	14555
House	Glasgow	Q2	15888
Flat	Glasgow	Q3	14578
House	Glasgow	Q3	16004
Flat	Glasgow	Q4	15890
House	Glasgow	Q4	15500
Flat	London	Q1	19678
House	London	Q1	23877
Flat	London	Q2	19567
House	London	Q2	28677
.....	.....	.....	.....
.....	.....	.....	.....



# Representing Multi-dimensional Data

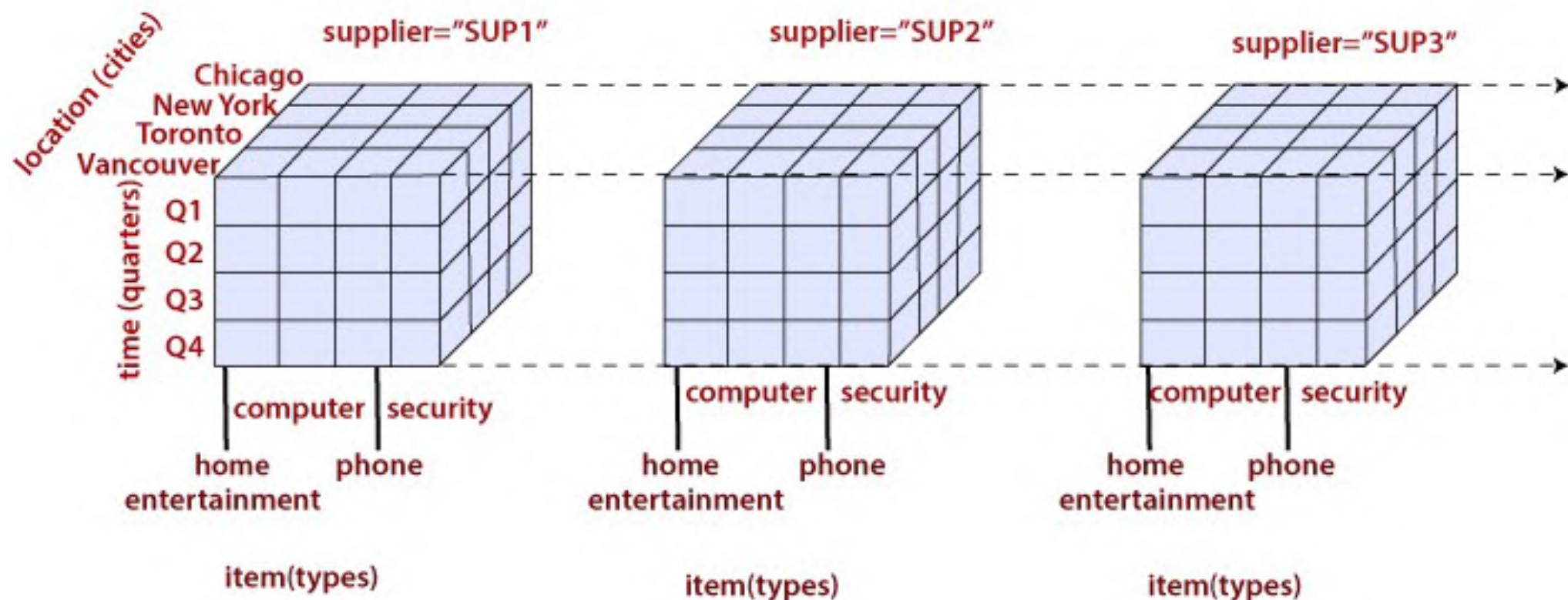
- Matrix vs. cube

Time	Location="Asia"				Location="Euro"			
	Product				Product			
	Phones	Modems	Mouse	RD	Phones	Modems	Mouse	RD
2000					174			
2001								
2002								
2003								



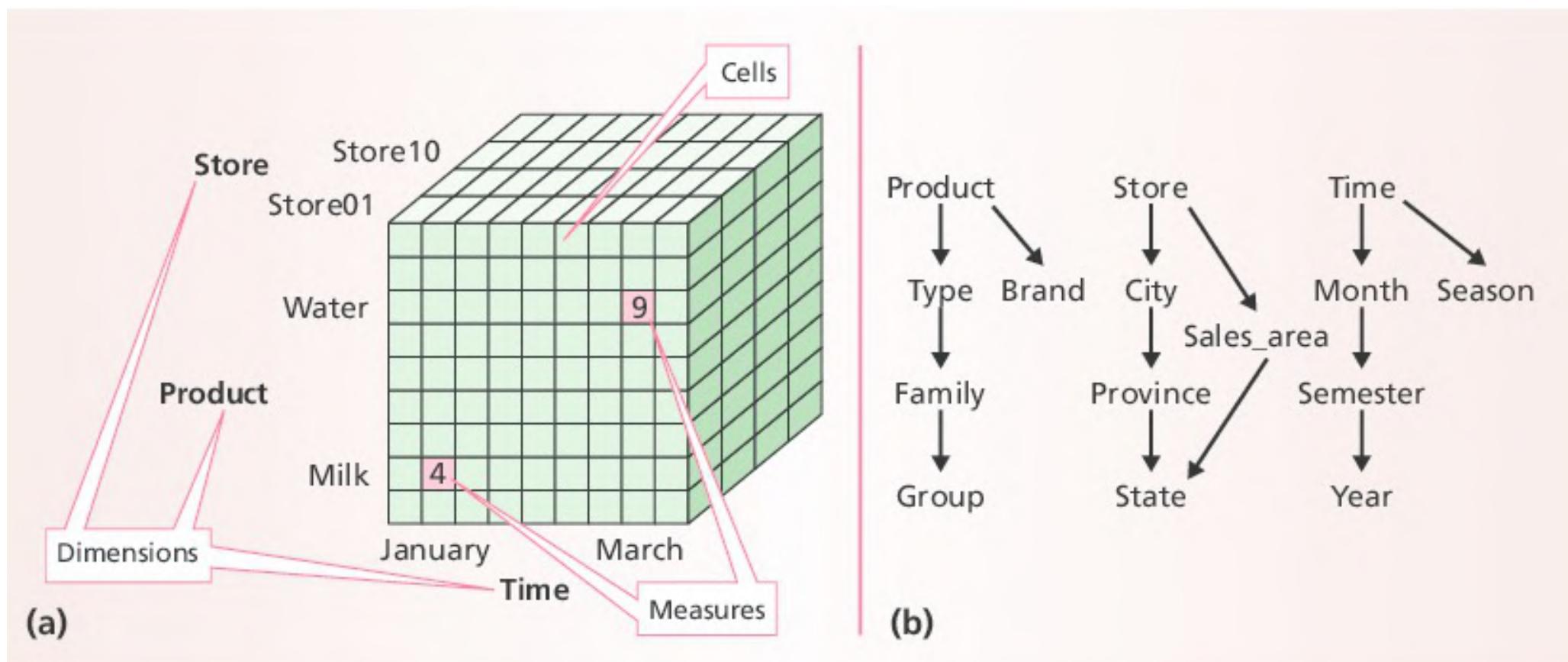
# Representing Multi-dimensional Data

- Series of cubes



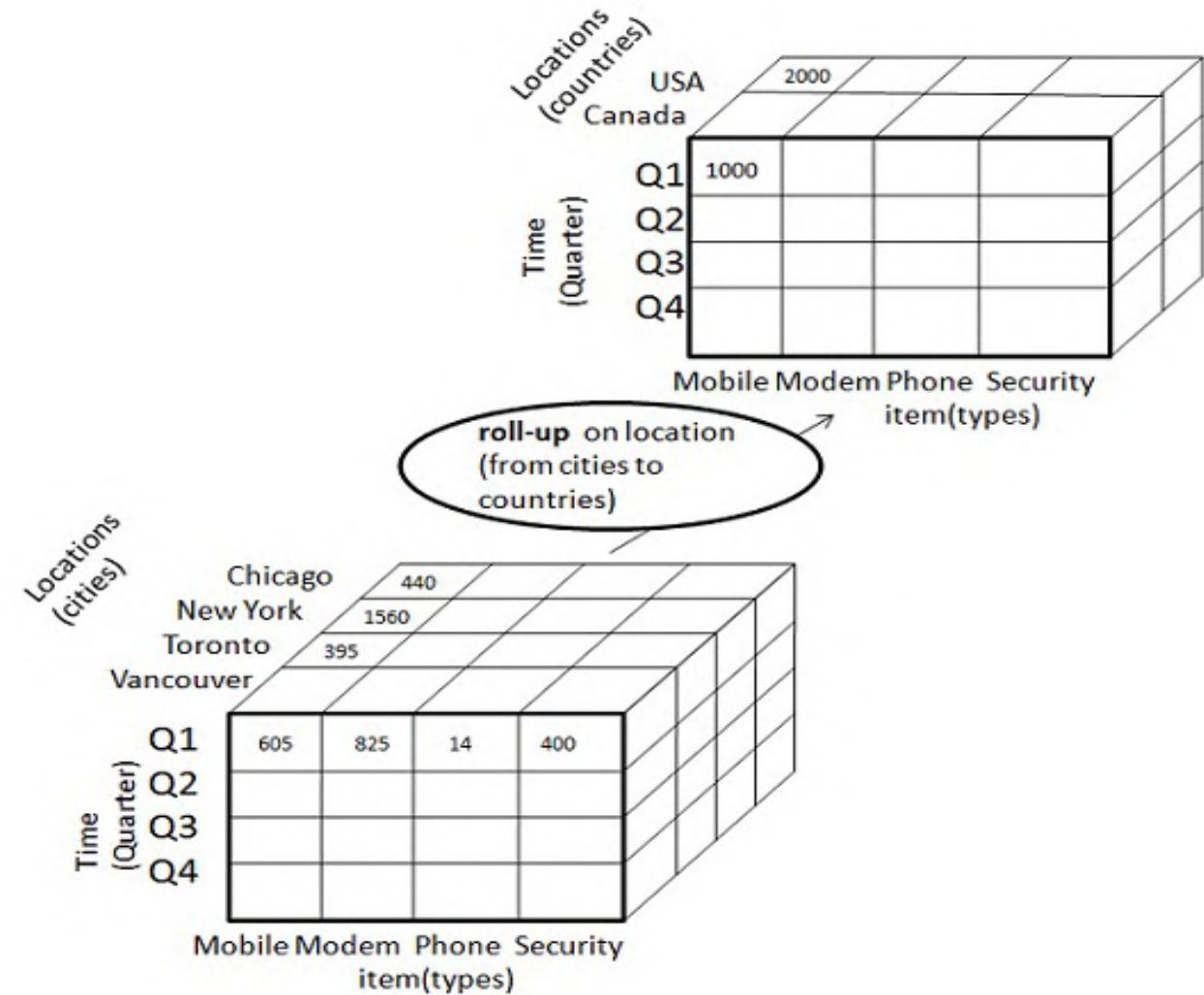
# Dimensional Hierarchy

- A dimensional hierarchy defines mappings from a set of lower-level concepts to higher level concepts



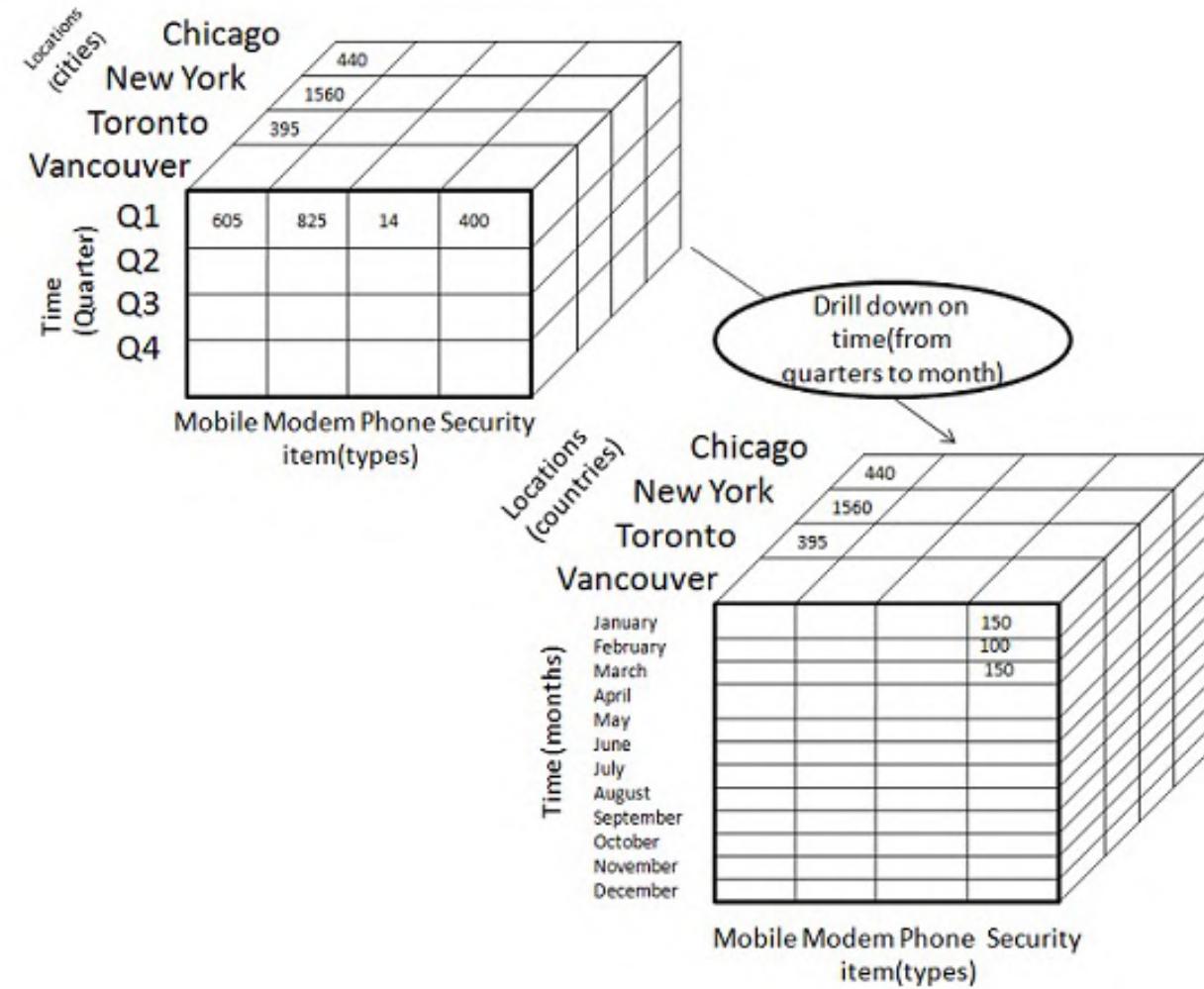
# OLAP Operations: Roll-up

- Climbing up a hierarchy of a dimension to aggregate data
- Also known as consolidation or aggregation
- e.g., roll-up through country-city hierarchy in *Location* dimension



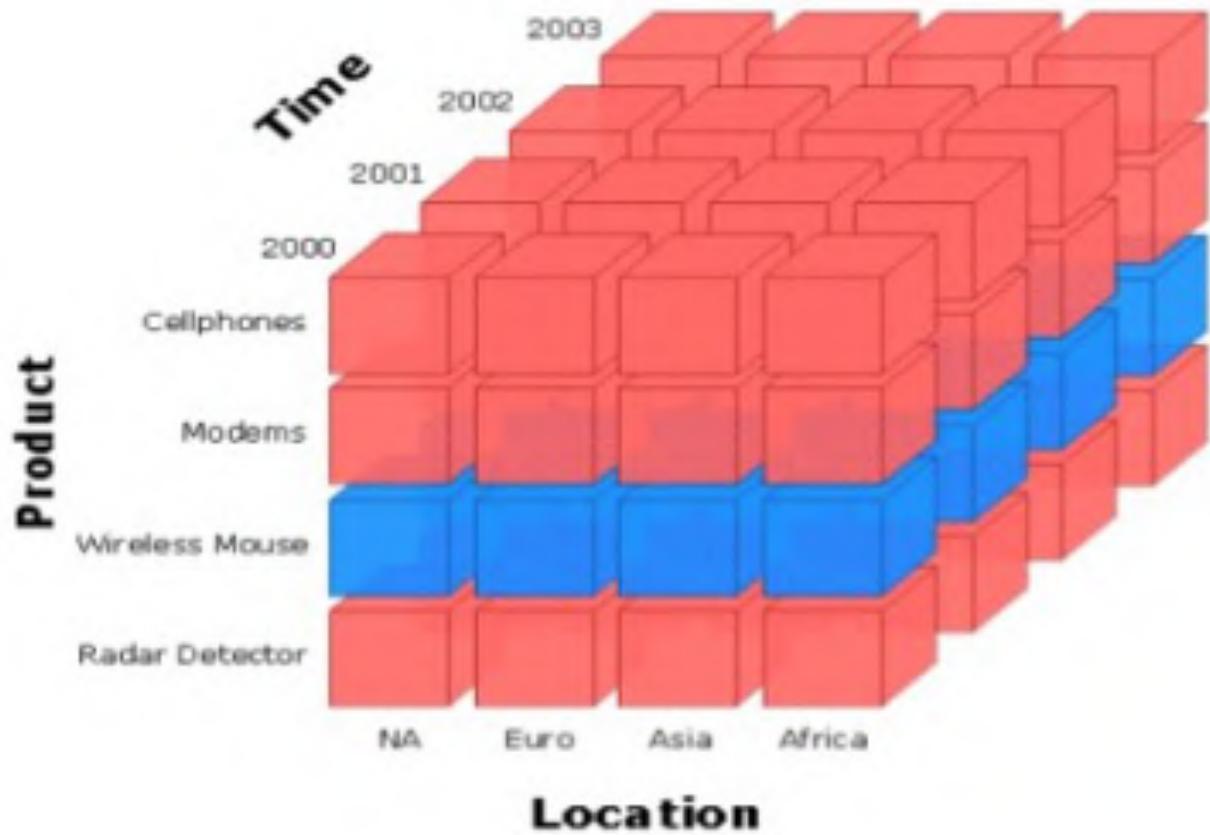
# OLAP Operations: Drill-down

- Stepping down a hierarchy of a dimension allowing navigation through details
- Data is fragmented into smaller parts
- It is the opposite of the rollup process
- e.g., drill-down through quarter-month hierarchy in *Time* dimension



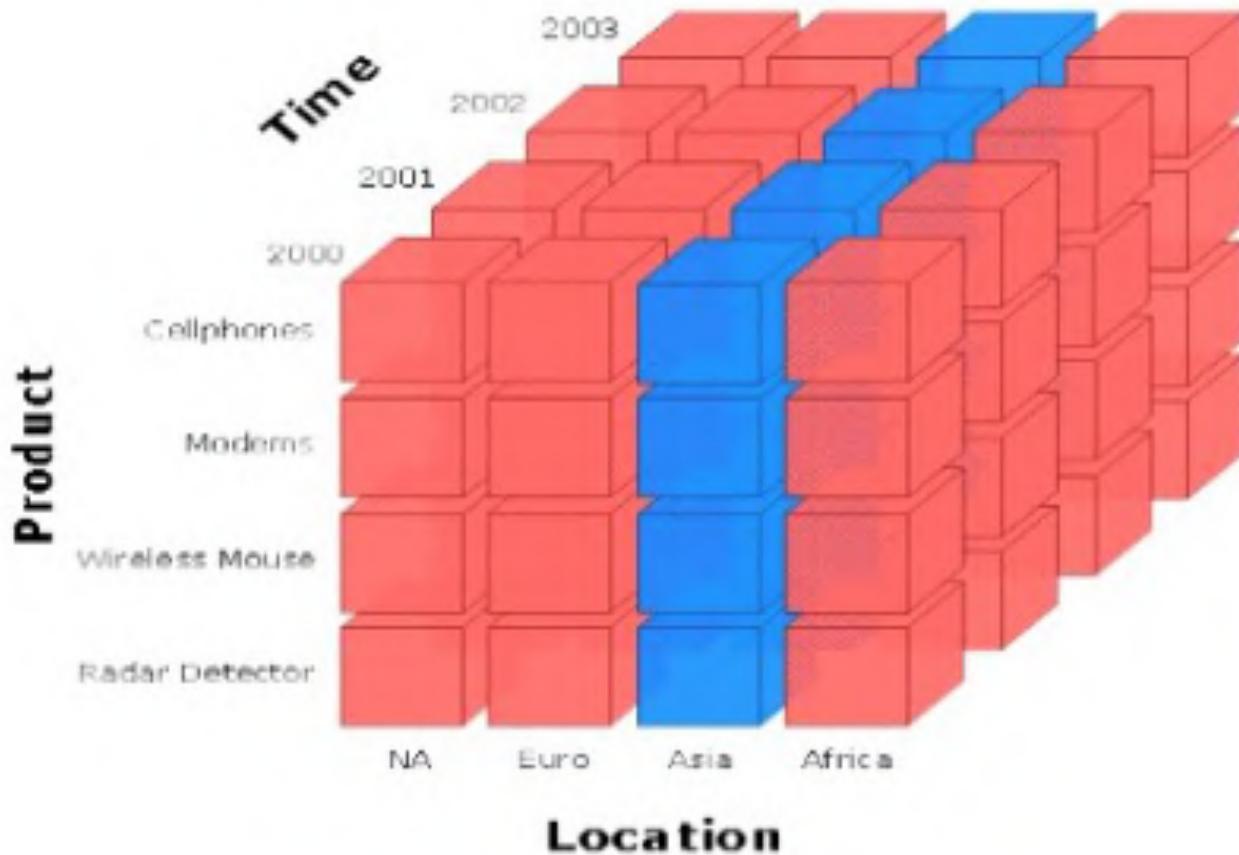
# OLAP Operations: Slice

- A rectangular subset of a cube, by choosing a single value for one of its dimensions
- e.g., dimension *Product* is sliced with “Wireless Mouse” as the filter



# OLAP Operations: Slice

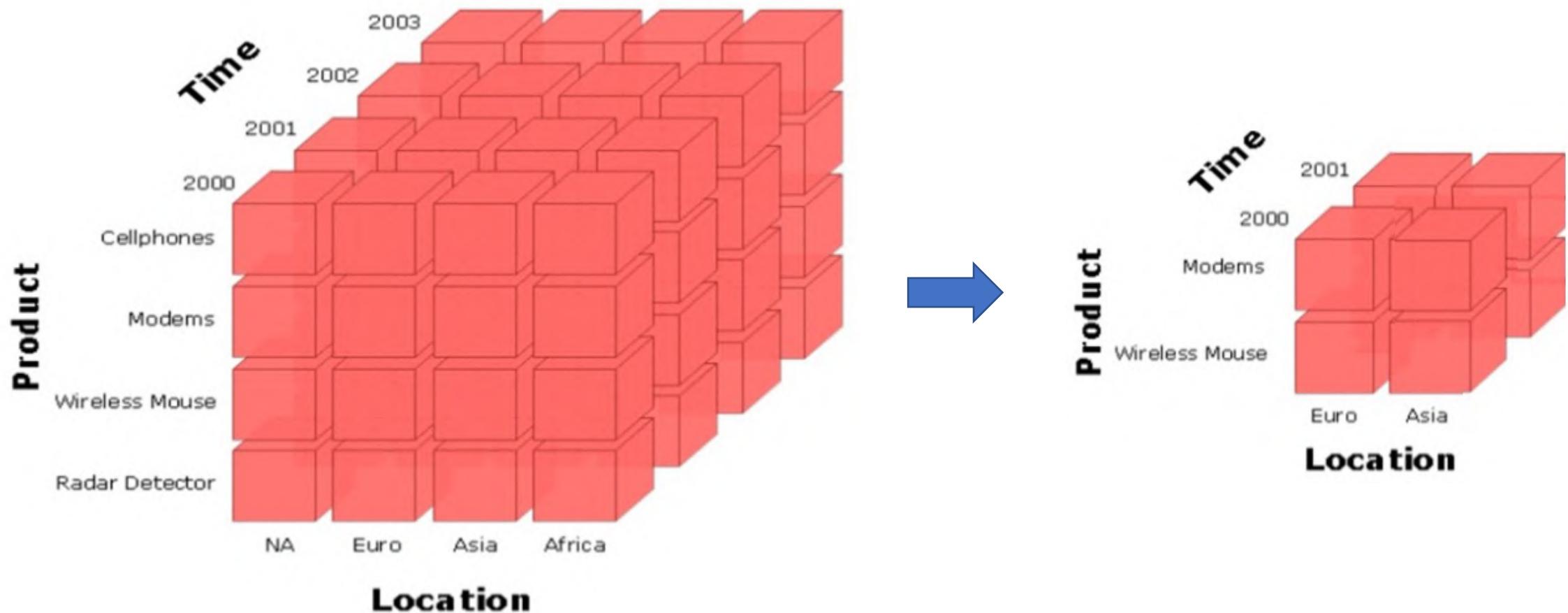
- e.g., dimension Location is sliced with “Asia” as the filter



# OLAP Operations: Dice

- Selects two or more dimensions from a given cube and provides a new sub-cube by selecting specific values on those selected dimensions
- e.g., (Product = “Modems” or “Wireless Mouse”) and  
(Time = “2000” or “2001”) and  
(Location = “Euro” or “Asia”)

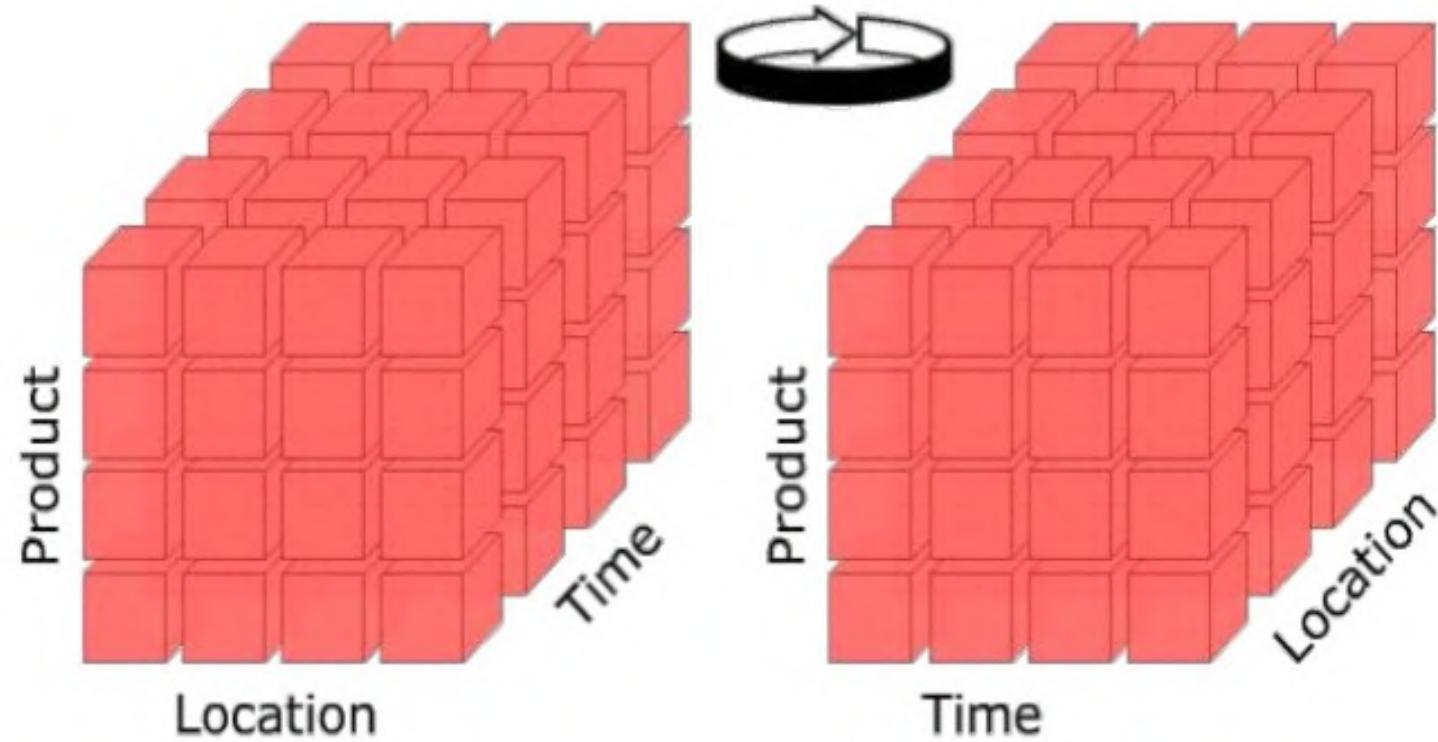
# OLAP Operations: Dice



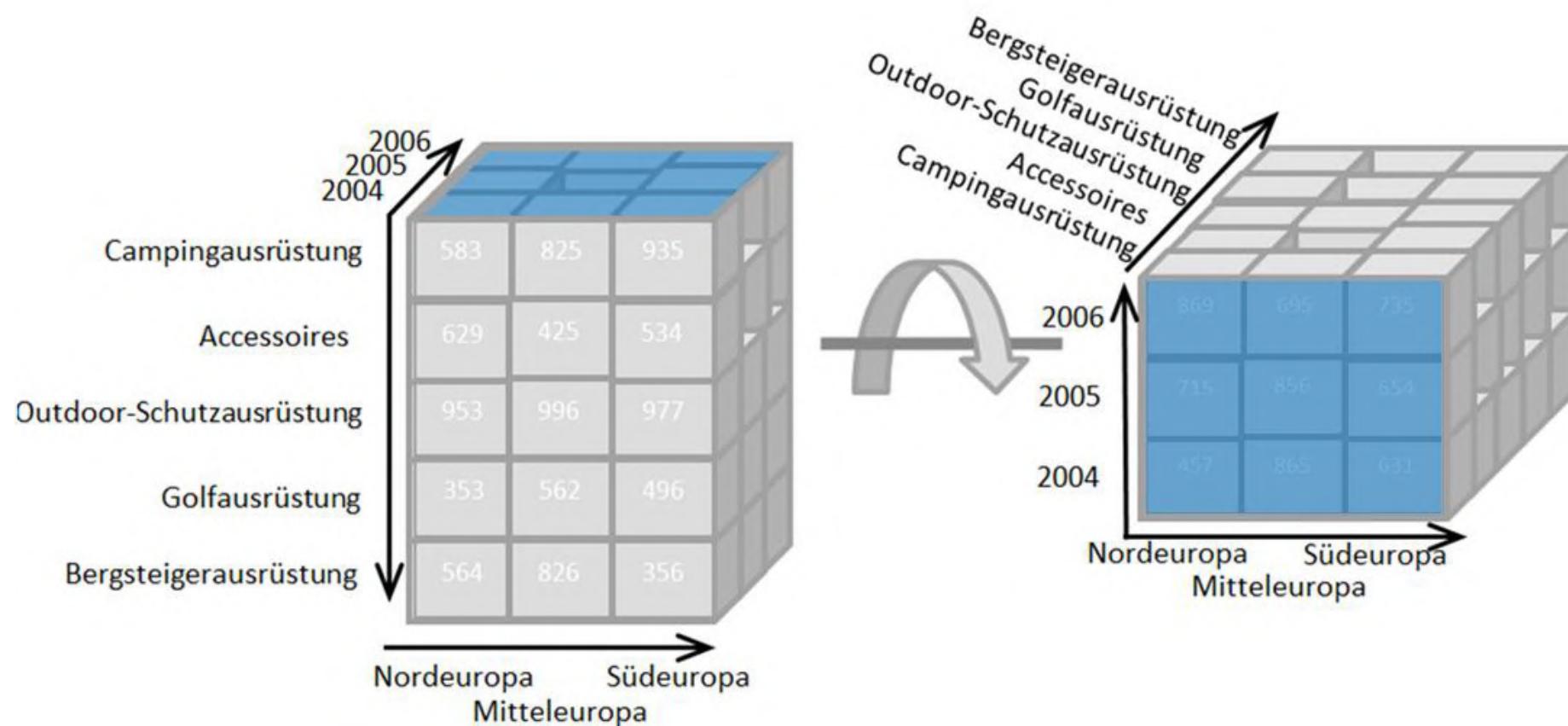
# OLAP Operations: Pivot

- Pivot is a visualization operations which rotates the data axes to provide an alternative presentation of the data
  - Rotate the cube to see its various faces
- It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions

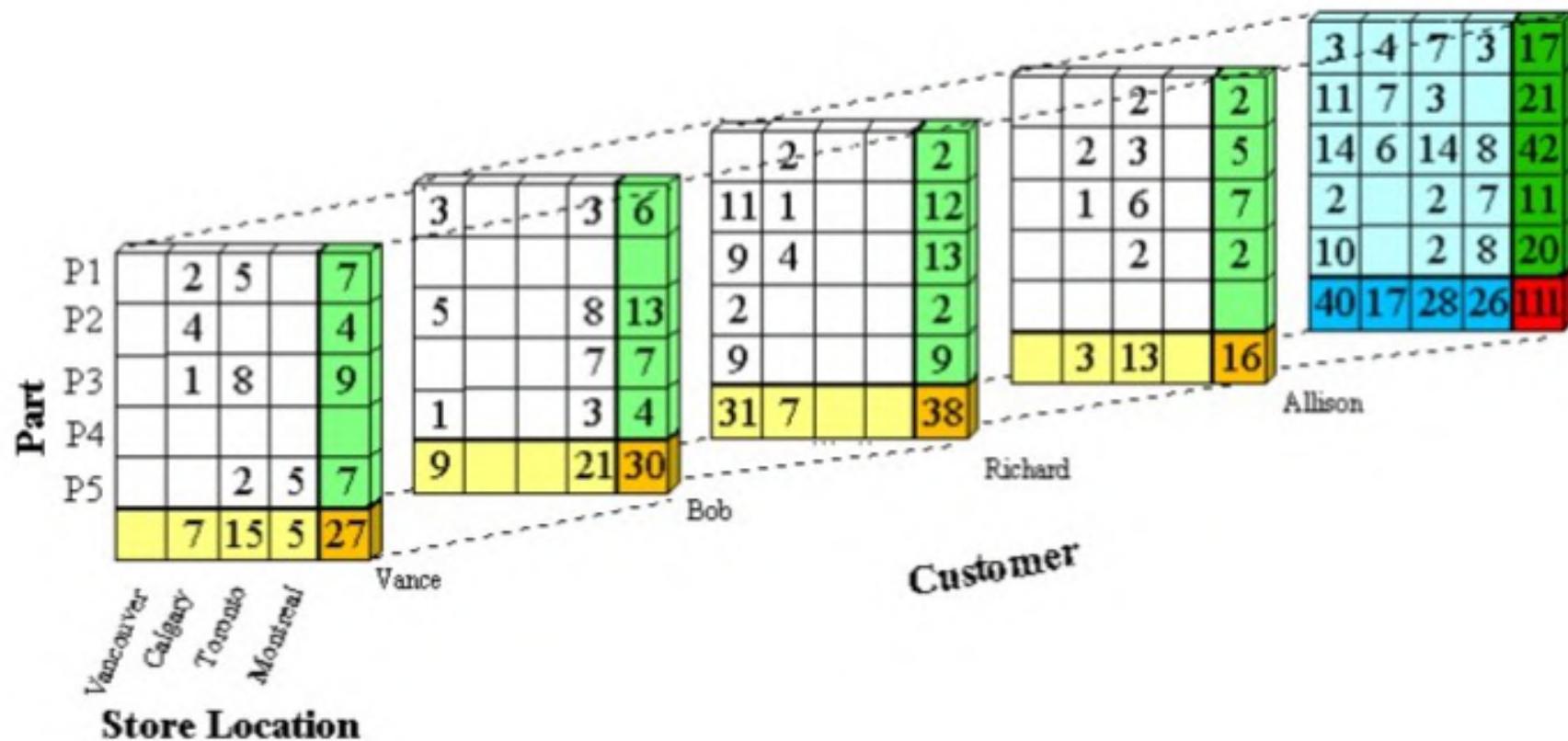
# OLAP Operations: Pivot



# OLAP Operations: Pivot



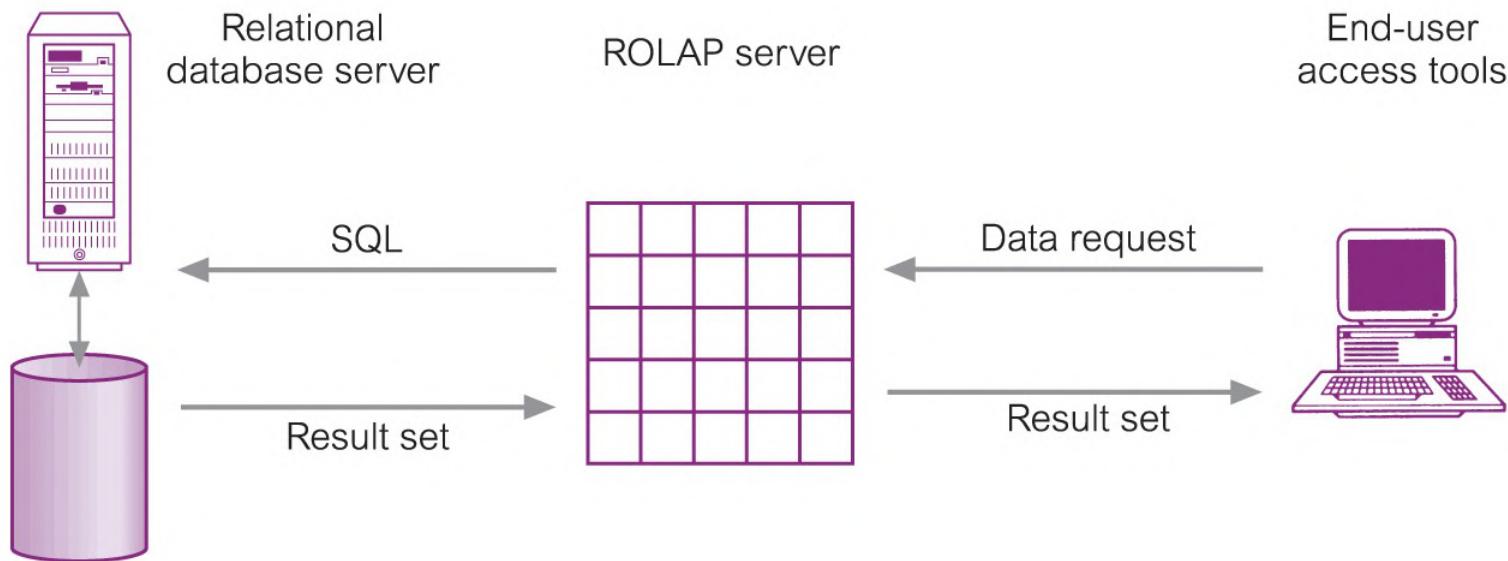
# Aggregations in OLAP Cubes



# OLAP Storage Types

- **ROLAP:** Relational OLAP
- **MOLAP:** Multidimensional OLAP (OLAP Cubes)
- **HOLAP:** Hybrid OLAP
- However, irrespective of the storage type, all of the above leverage same dimensional concepts

# Relational OLAP

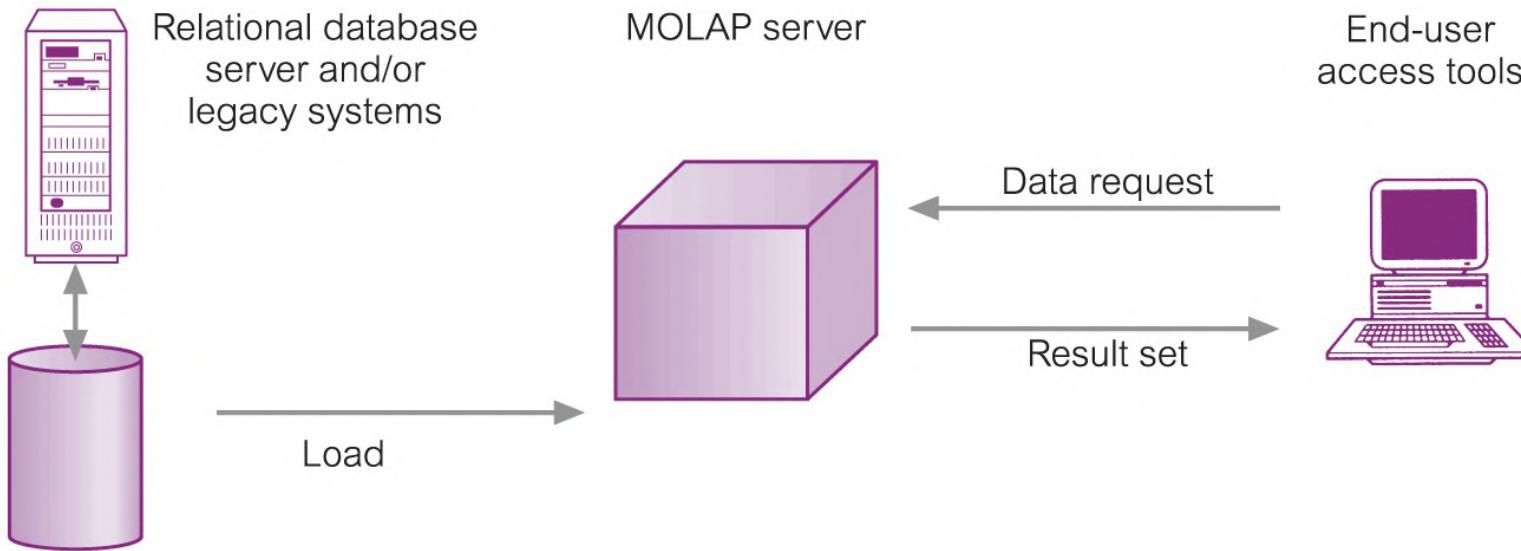


- Works primarily from the data that resides in a relational database, where the base data and dimension tables are stored as relational tables
- This model permits the multidimensional analysis of data
- Manipulates the data stored in the relational database to give the presence of traditional OLAP functionality
  - Maps functions on multidimensional data to standard relational operations
  - Data manipulation/filtering through “WHERE” clauses

# Relational OLAP

- PROS:
  - Can handle large amounts of data (no limitation)
  - Leverage relational database functionalities
  - Detailed data is available
  - Highly scalable; comparatively poor query performances
- CONS:
  - Slow performance (essentially everything is a query)
  - Limited by SQL functionalities (However, vendor may provide specific out-of-the-box features)

# Multidimensional OLAP

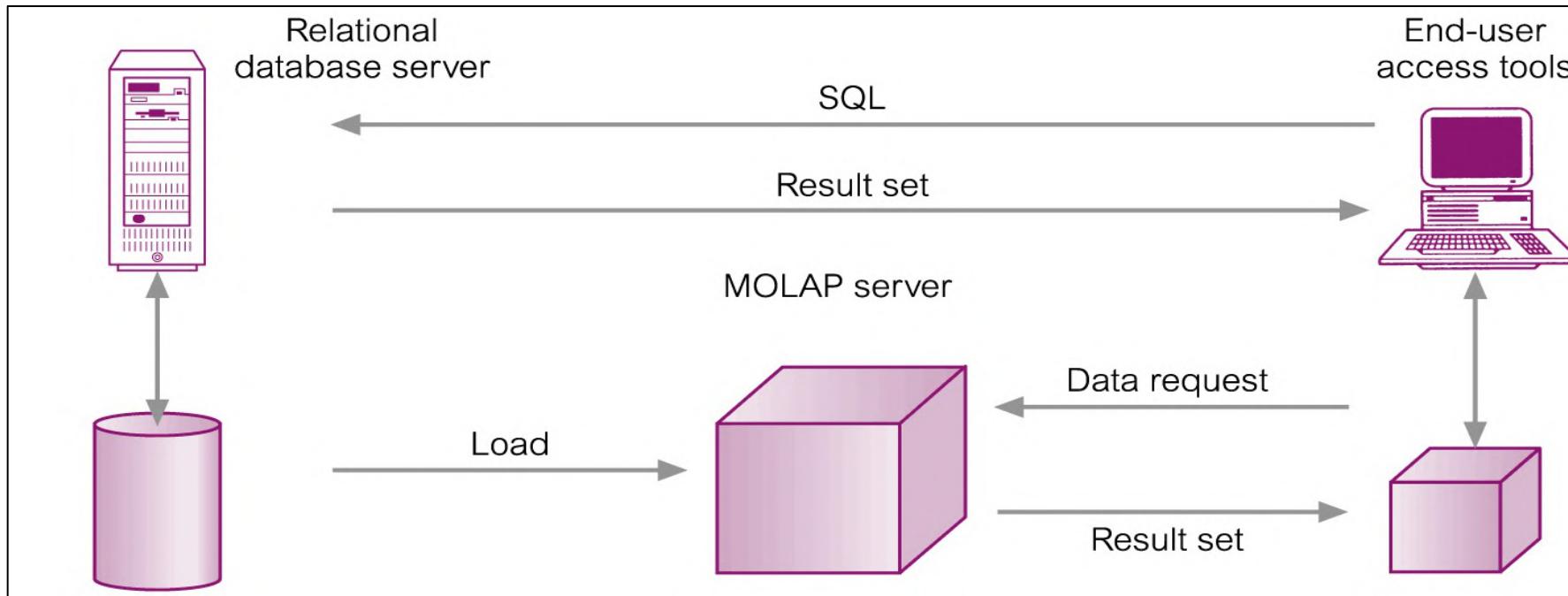


- Based on a native logical model that directly supports multidimensional data and operations
- Uses multidimensional storage engines (arrays) to store data
  - Proprietary formats
- One of the significant distinctions of MOLAP against a ROLAP is that data are summarized and are stored in an optimized format in a multidimensional cube, instead of in a relational database

# Multidimensional OLAP

- PROS:
  - Excellent performance
  - Pre-calculated aggregations (fast data retrieval)
  - Can perform complex calculations and return results quickly
  - Optimal for OLAP operations (slicing, dicing, roll-up, drill-down)
- CONS:
  - Limited in amount of data (maintains a copy)
  - Additional investment (proprietary technologies)
  - Latency is high (refresh data only when processed)

# Hybrid OLAP

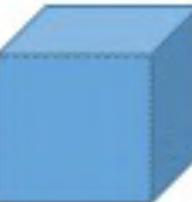
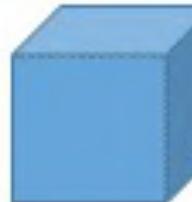
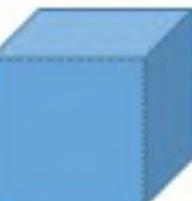
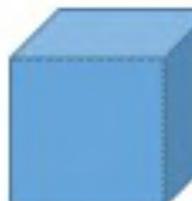
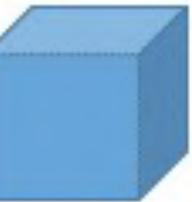
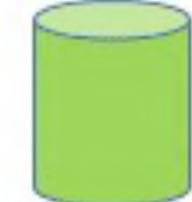
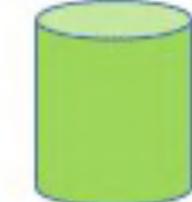


- A combination of advantages of both ROLAP and MOLAP

# Hybrid OLAP

- PROS:
  - Data stored in relational format (ROLAP): minimum latency
  - Aggregations are stored in cubes: better query performance
  - It offers higher scalability of ROLAP and faster computation of MOLAP
  - HOLAP servers allows to store large data volumes of detailed information
- CONS:
  - HOLAP architecture is very complicated because it supports both MOLAP and ROLAP servers

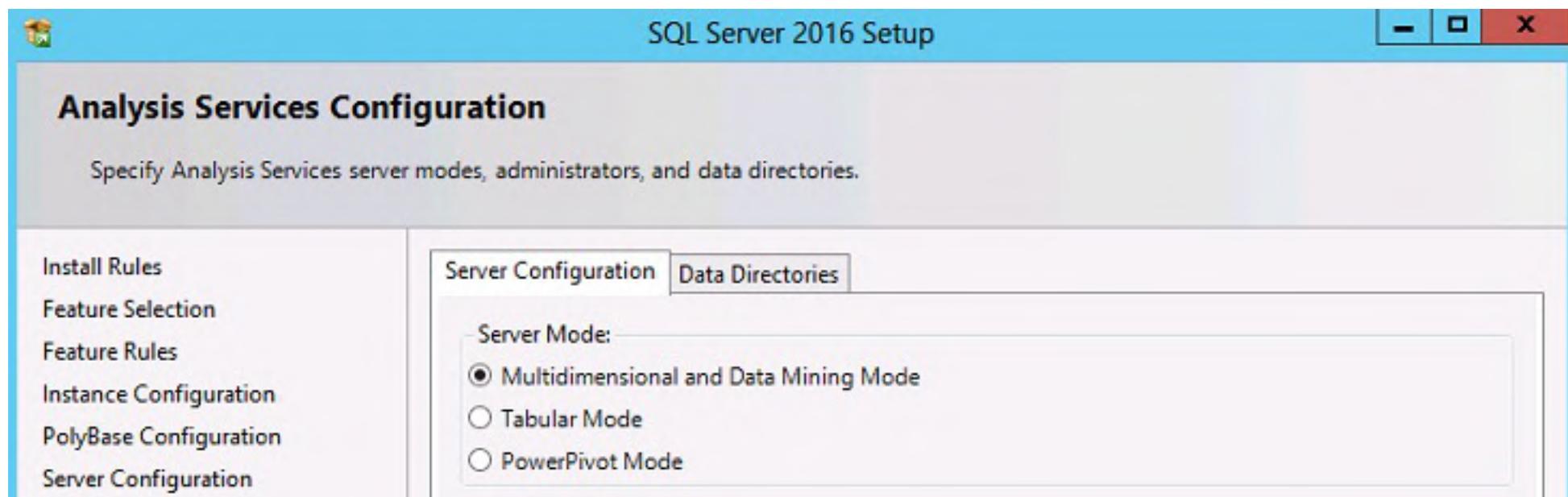
# ROLAP vs. MOLAP vs. HOLAP

	MOLAP	HOLAP	ROLAP
Cube Structure			
Preprocessed Aggregates			
Detail-Level Values			

 Multidimensional Storage     Relational Storage

# SQL Server Analysis Services

- 3 Server Modes



<https://docs.microsoft.com/en-us/sql/analysis-services/instances/determine-the-server-mode-of-an-analysis-services-instance>

<https://docs.microsoft.com/en-us/sql/analysis-services/comparing-tabular-and-multidimensional-solutions-ssas>

# SQL Server Analysis Services

- SSAS Multi-dimensional mode has several storage modes
  - Real Time ROLAP
  - Real Time HOLAP
  - Low Latency MOLAP
  - Medium Latency MOLAP
  - Automatic MOLAP
  - Scheduled MOLAP
  - MOLAP

# SQL Server Analysis Services

- Querying SSAS
  - **DAX:** Data Analysis Expressions
  - **MDX:** Multi-Dimensional Expressions
    - Retrieving data from SSAS cubes
  - **DMX:** Data Mining Extensions
    - Used for data mining structures (creating, processing, copying, browsing of mining structures and models)
  - **XMLA:** XML for Analysis
    - Commonly used in SSAS administration tasks such as backup or restore database, copy and move database or learning meta data information
  - Native SSAS Browser can also be used to analyse data

# SQL Server Analysis Services

- Data mining with SSAS
  - Association rule
  - Clustering
  - Decision Trees
  - Regression
  - Naïve Bayes
  - Neural Network
  - Sequence Clustering
  - Time Series

**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

Lecture 08

Data Consumption &  
Business Intelligence

# Content

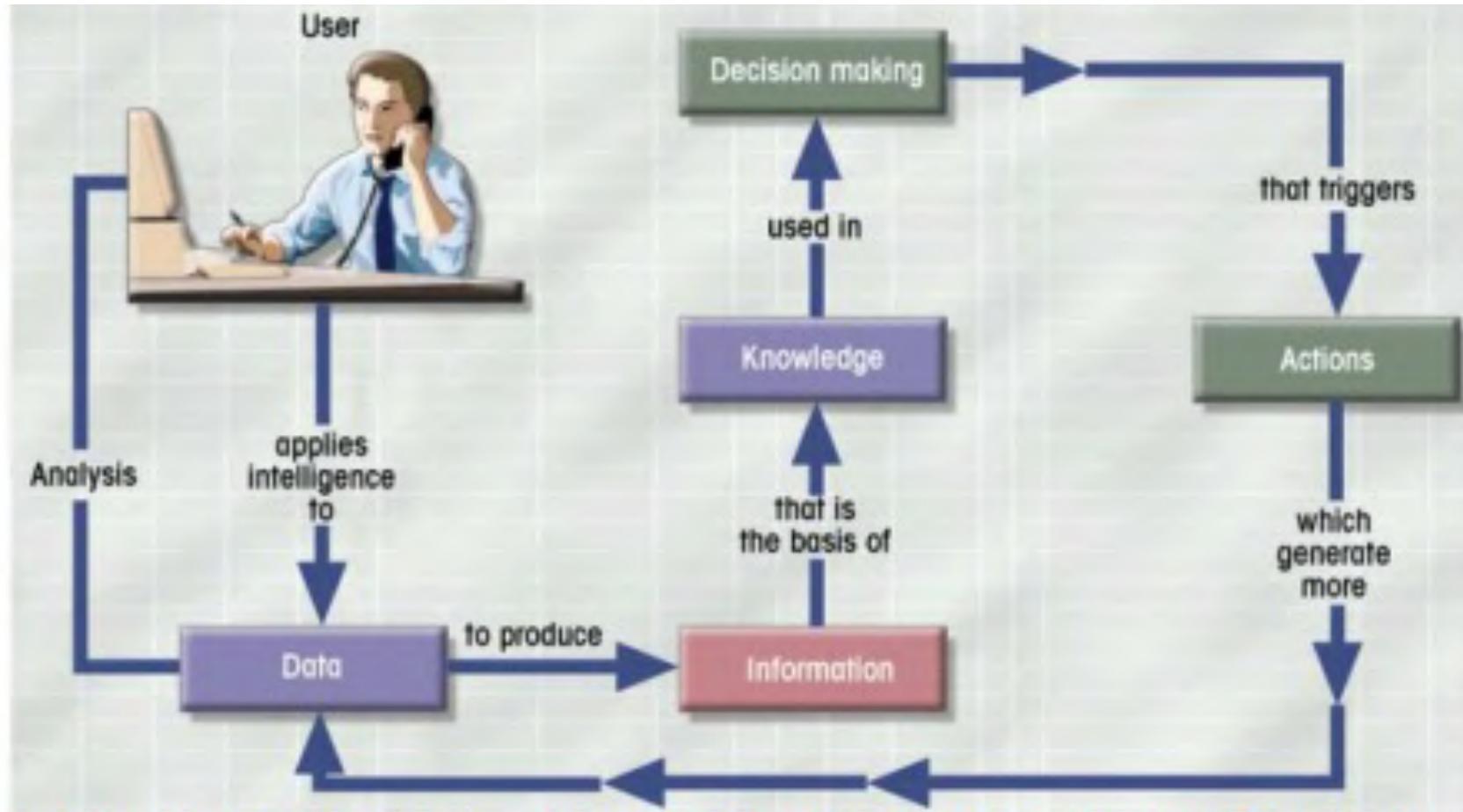
- Business Intelligence
  - What is business intelligence?
  - Business intelligence process
- Data Visualization
  - Types of data visualizations
  - Ways of presenting data
    - Operational and analytical reports, dashboards, scorecards, self-service BI
- Well-known visualization features
- Challenges in visualizations

# Business Intelligence

---



# Data-Information-Decision Cycle



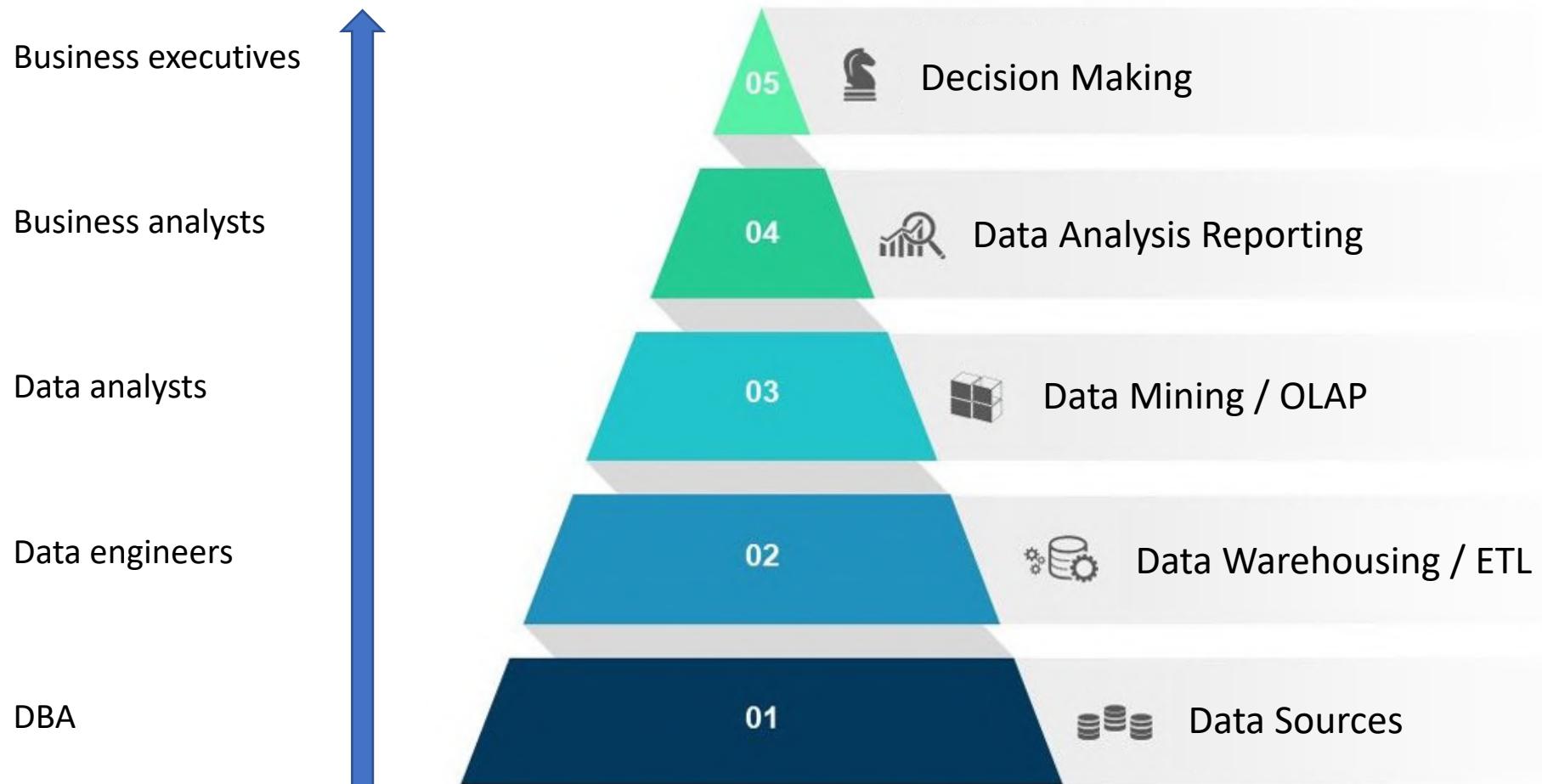
# Decision Making

- Decision making at different levels
  - Operational
    - Related to daily activities with short-term effects
    - Structured decisions taken by lower management
  - Tactical
    - Semi-structured decisions taken by middle management
  - Strategic
    - Long-term effects
    - Unstructured decisions taken by top management
- Decision making steps:
  - Problem identification
  - Finding alternative solutions
  - Making a choice

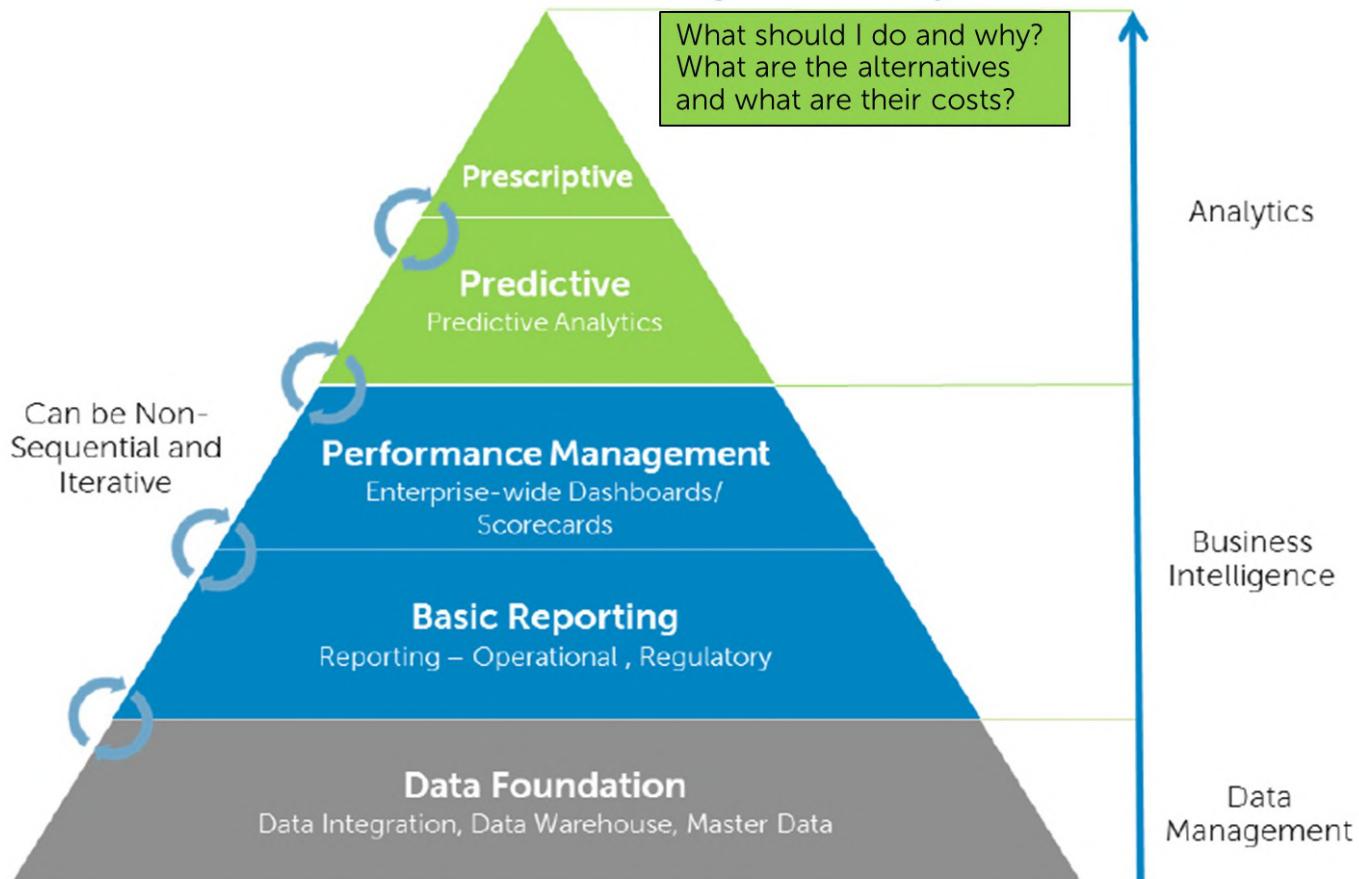
# What is Business Intelligence?

- BI leverages software and services to transform data into actionable insights, used for organization's strategic and tactical business decisions
  - BI tools access and analyse data sets and present analytical findings in reports, summaries, dashboards, graphs, charts and maps to provide users with detailed intelligence about the state of the business
- The term business intelligence often also refers to a range of tools that provide quick, easy-to-digest access to insights about an organization's current state, based on available data
- **New trends** in BI includes AI, collaborative BI, embedded BI, cloud analytics.

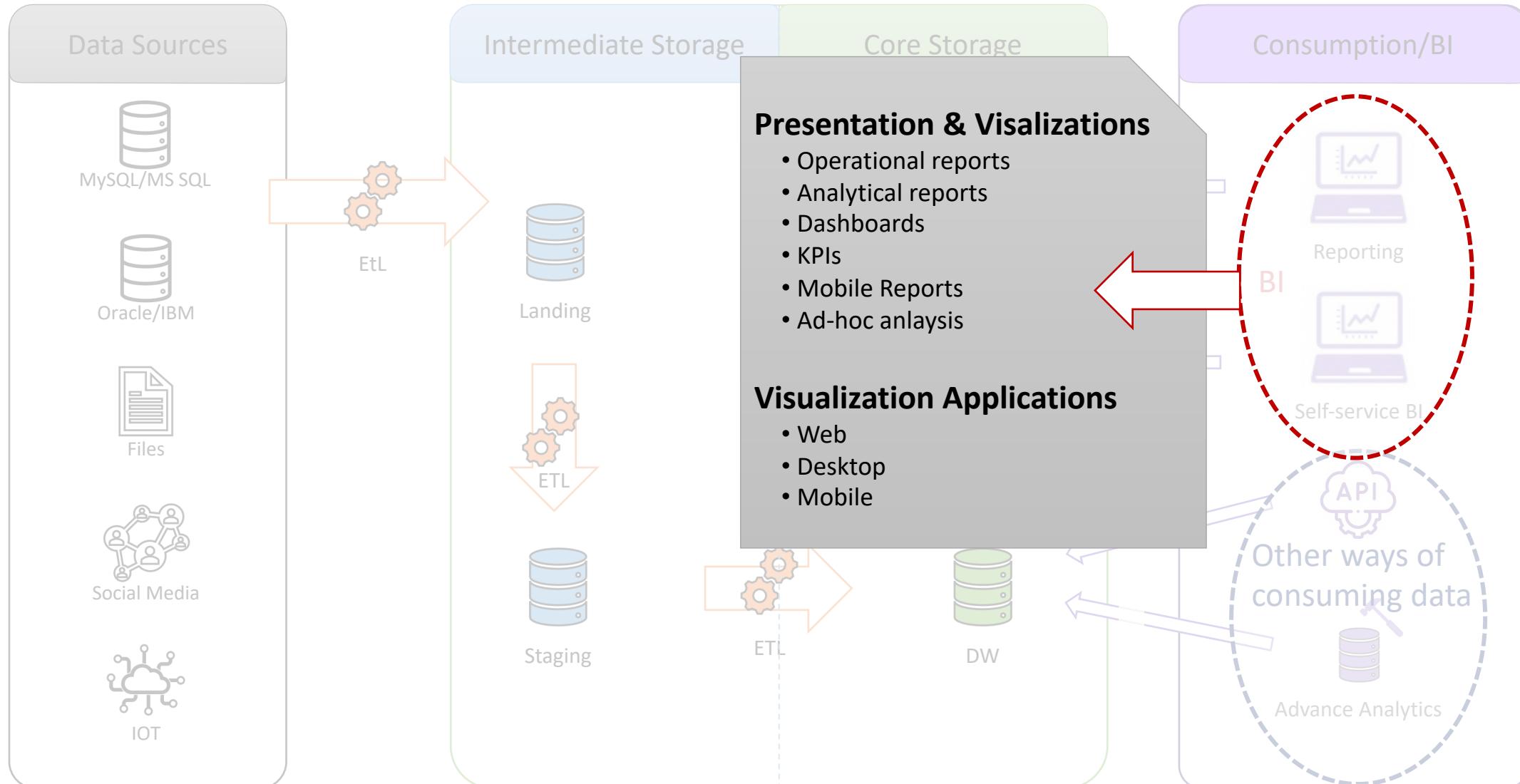
# Business Intelligence Process



# BI Process with Predictive Analytics



# Business Intelligence Layers



# Visualization

---



# Data Visualization

- Data visualization is the graphical representation of information and data
- By using visual elements like charts, graphs, and maps, data visualization tools provide an easier way to see and understand trends, outliers, and patterns in data
- Often data visualization tools turn patterns which are invisible in raw data format into visible patterns, that people can understand intuitively
- With the growth of the volume of data, visualization tools and technologies are essential to analyse massive amounts of information and make data-driven decisions

# Common Types of Data Visualization

- Charts
- Tables
- Graphs
- Maps
- Infographics

# More Specific Visualisation Types

- Area Chart
- Bar Chart
- Box-and-whisker Plots
- Bubble Cloud
- Bullet Graph
- Cartogram
- Circle View
- Dot Distribution Map
- Gantt Chart
- Heat Map
- Highlight Table
- Histogram
- Matrix
- Network
- Polar Area
- Radial Tree
- Scatter Plot (2D or 3D)
- Streamgraph
- Text Tables
- Timeline
- Treemap
- Wedge Stack Graph
- Word Cloud

# When to Use Which Type?

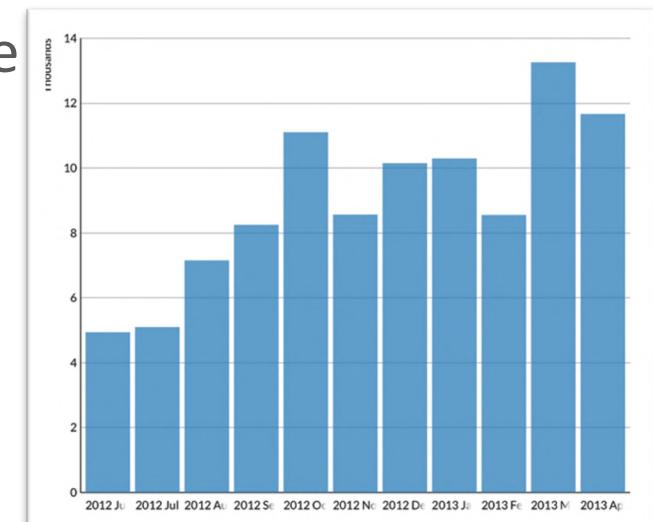
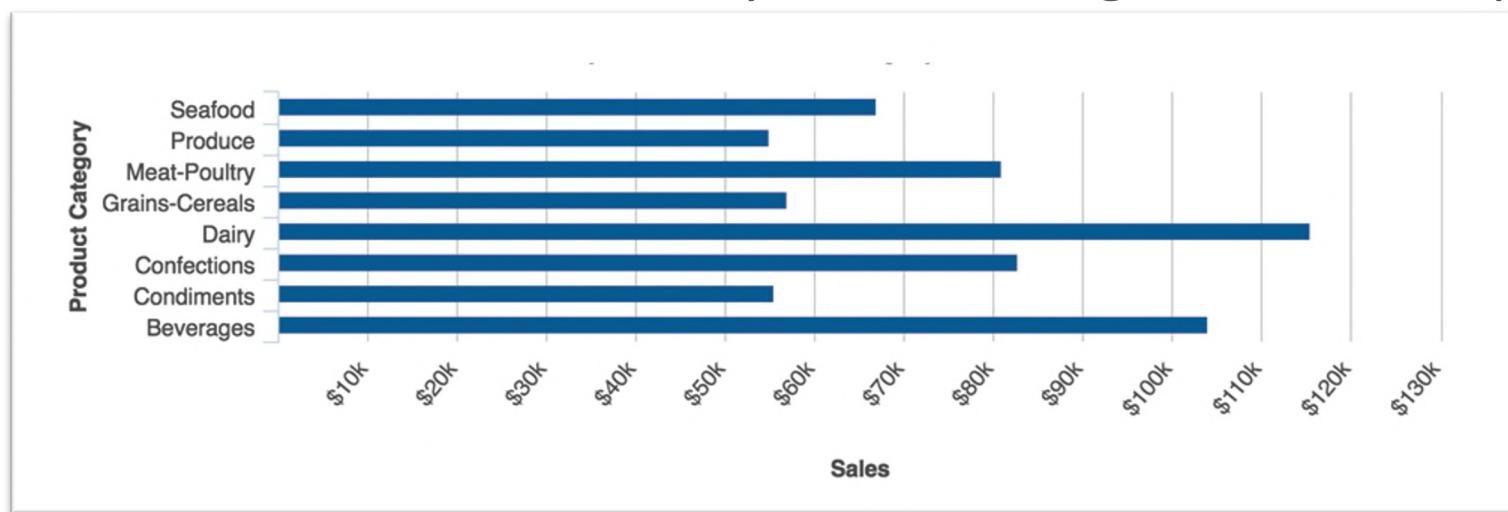
- Tabular format: best used when exact quantities of numbers must be known
  - Numbers are presented in rows and columns, and may contain summary information, as in PivotTables
  - Not suitable for finding trends and comparing sets of data
- Line charts: best used when trying to visualize continuous data over time and are ideal for showing trends in data

Interaction by Day of Week		
Day of week	Data Hub Activities	Pages / Visit
1	20	3.30
2	14	3.22
0	8	3.26
3	2	3.48
5	2	2.39



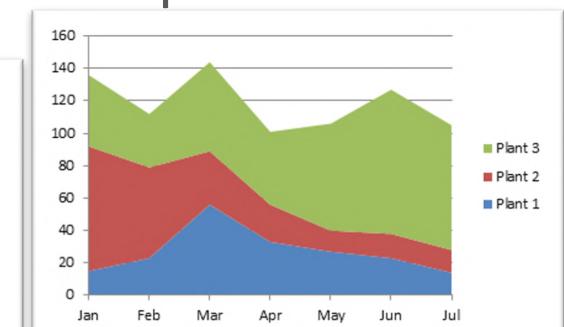
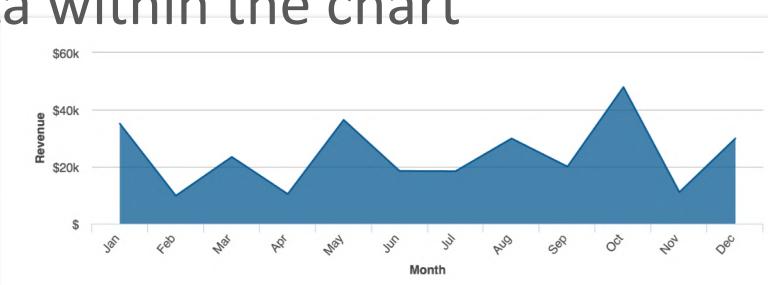
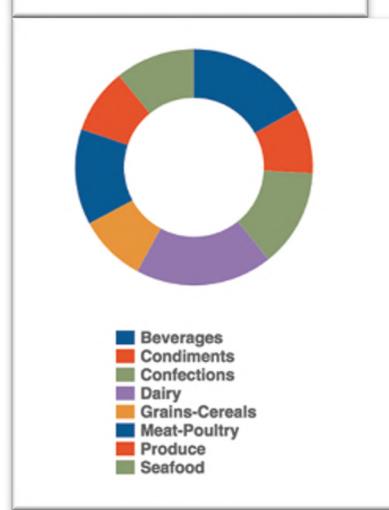
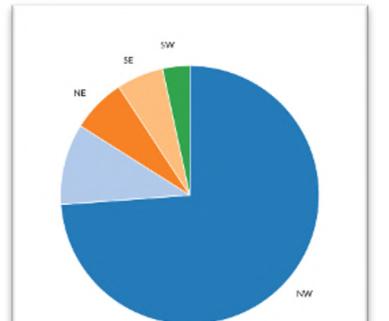
# When to Use Which Type?

- Bar charts: best used when showing comparisons between categories
  - Typically, the bars are proportional to the values they represent and can be plotted either horizontally or vertically
  - One axis of the chart shows the specific categories being compared, and the other axis represents discrete values
  - Bar charts are ideal when you're working with limited space



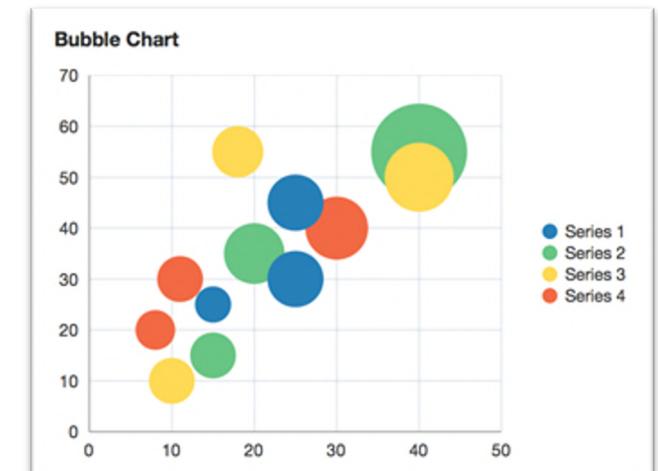
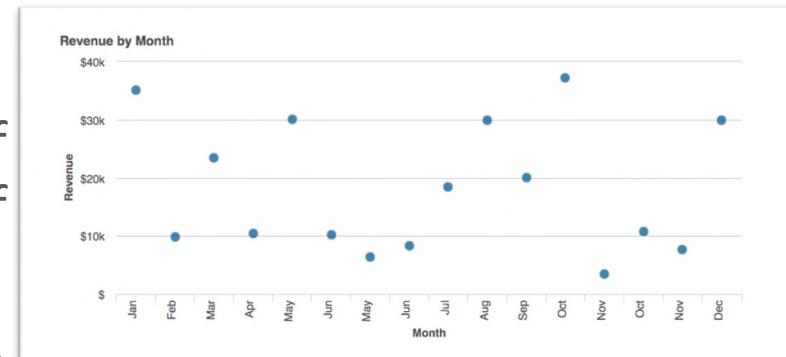
# When to Use Which Type?

- Pie charts: best used to compare parts to the whole
  - Pie charts make it easy for an audience to understand the relative importance of values
  - Alternate visual styles include the exploded pie wedge chart and the donut pie chart
- Area charts: best used for showing cumulated totals over time via numbers or percentages
  - These are basically line charts that are filled in to provide a deeper view of multiple series of data within the chart



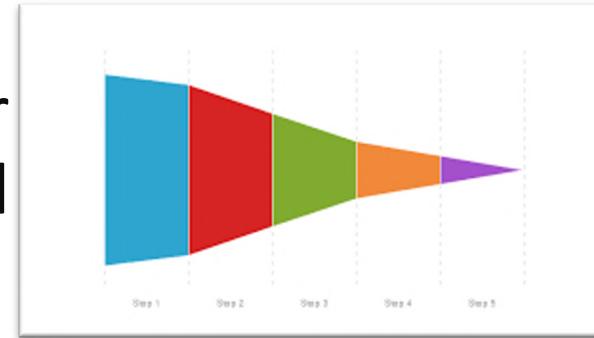
# When to Use Which Type?

- Scatter plot: best used to display relationships between 2 variables
  - The data is displayed as a collection of points; the value of one variable determines x-axis position, while the value of the other variable determines the y-axis position
  - Scatter charts work best when you have an integer value on both the Y- and X-axis; otherwise, your scatter chart will look like a line chart without the line
- Bubble charts: used to show three dimensions of data—comparing entities in terms of their relative values, positions, and **sizes**
  - Bubble charts are similar to scatter plots, where the data



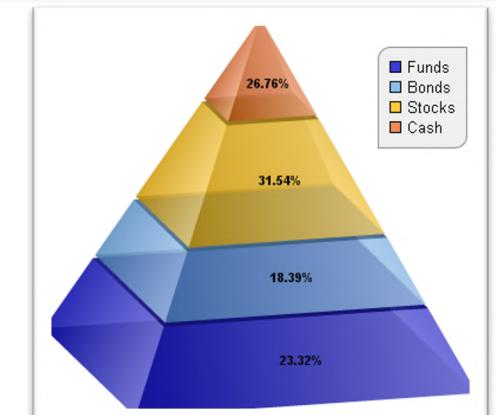
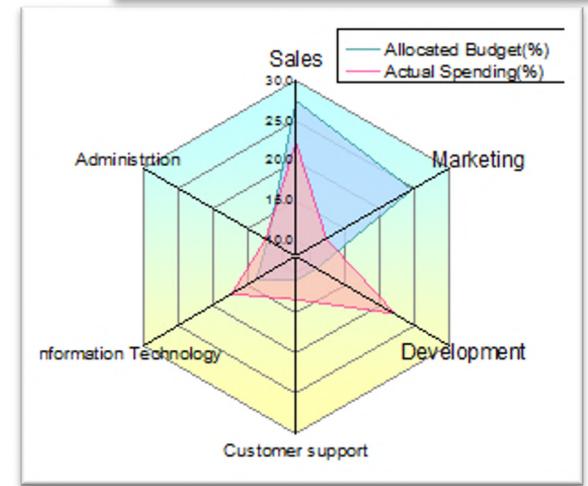
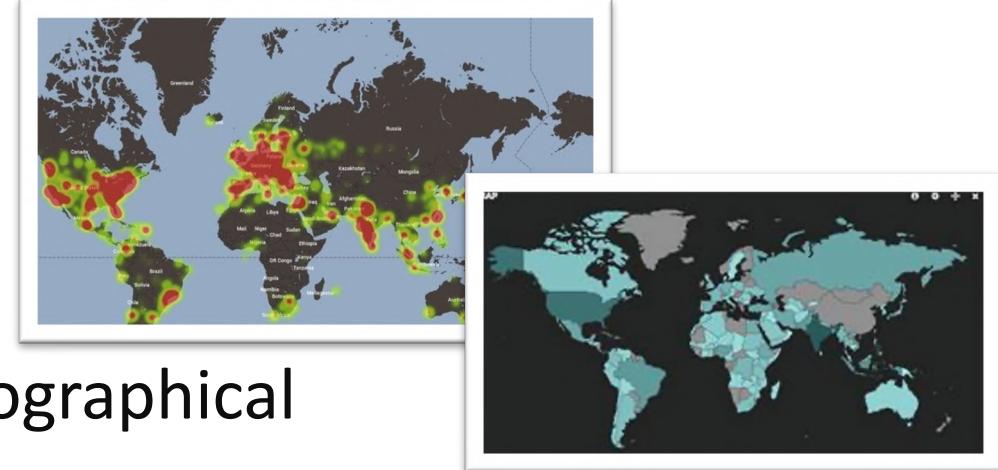
# When to Use Which Type?

- Funnel charts: ideal for showing stages in a particular process (e.g., sales process) or identifying potential problem areas within an organization's process
- Gauges: best used to show a range
  - Ideal when you have an absolute floor value and absolute ceiling value and you want to show where the value lies within that range
  - However, gauges take up valuable space but provide limited information since they present data on a single dimension
  - They tell you whether something is on target, above target,



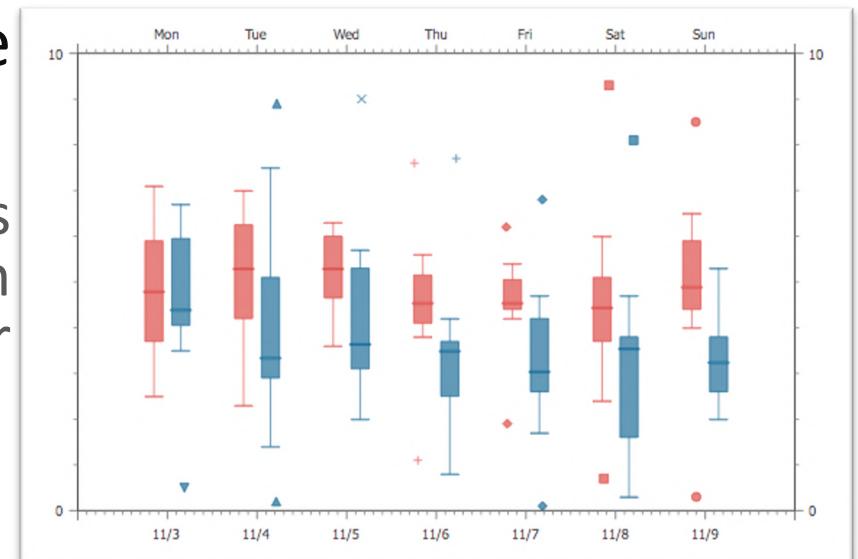
# When to Use Which Type?

- Heat maps: best for showing a geographical representation of data
  - Individual values can be shown as colours
- Polar charts: best for displaying multivariate observations with an arbitrary number of variables in the form of a two-dimensional chart
  - Alternative names include radar chart, web chart, spider chart, and star chart
- Pyramid charts: ideal for showing comparisons of data, using the thickness of layers to denote relative values



# When to Use Which Type?

- Sparkline charts: best for showing many trends at once
  - A prime example of a sparkline chart is the market summary
- Whisker charts or Box plots: best for statistical analysis and showing the distribution of a dataset
  - The lines that extend vertically from the boxes in these charts are the “whiskers,” which denote variability outside the upper and lower quartiles
- For more types:  
<https://datavizcatalogue.com/>



# Selection of Visualization Type

- In choosing the type of visualization, make sure you clearly understand the following points:
  - Specifics of your data set: domain knowledge of the data
  - Audience: people you want to present the information to
  - Connection logic: comparison of objects, distribution, relationship, process description, etc.
  - Output: simply, the reason for showing this information to somebody and which type to be used for that reason

# Ways to Present Data

- Operational reports
- Analytical reports
- Dashboards
  - KPIs
- Scorecards
  - KPIs
- Self Service BI
- Above could be presented via web tools, mobile tools or desktop tools.

# Operational Reports

- Typically, a multi-column, document-style, static report that contain data in text and table form
  - List of transactions that took place between a certain date range, within a specific location or region, or by sales rep
  - It is a listing, or in many cases an Excel export, potentially with groups and totals
- Usually delivered to various stakeholders periodically
- Generally presented without the ability to manipulate data
  - Good for generating reports with fixed-layout and optimized for printing or save as PDF or Excel formats
- Contains much more detailed information and much longer
- Tables and charts that live within a report can take up many pages of a printed medium
  - In the electronic mediums, a report will likely require the reader to scroll through many screens or click from page to page

# Operational Reports

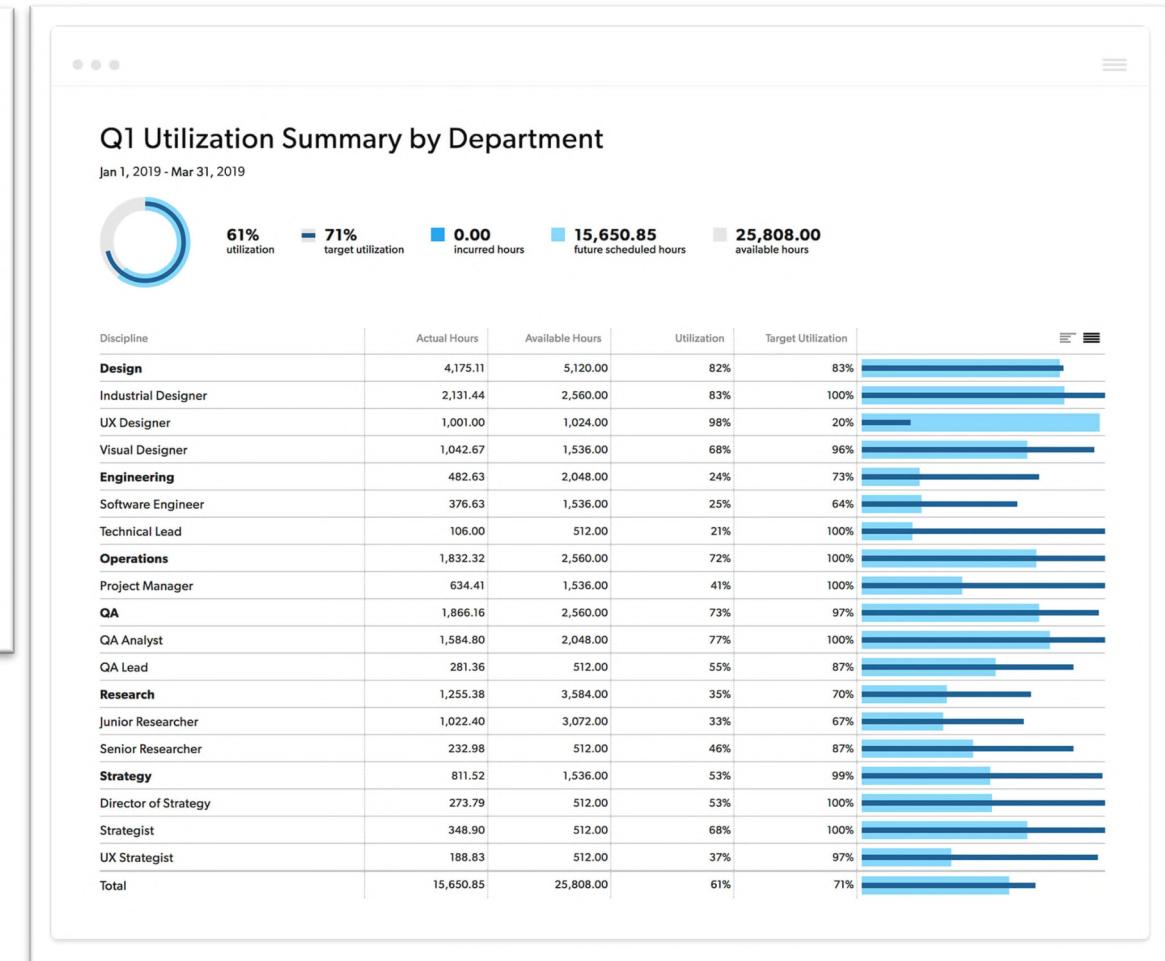
View Menu for Last Month All Hours for All Employees More Update

Product Mix Top Groups Top Items Top Modifiers Item Details Modifier Details 86 Report

**Menu Items ordered in the current time period**

25 items per page Showing 1 to 25 of 75 Show / hide columns

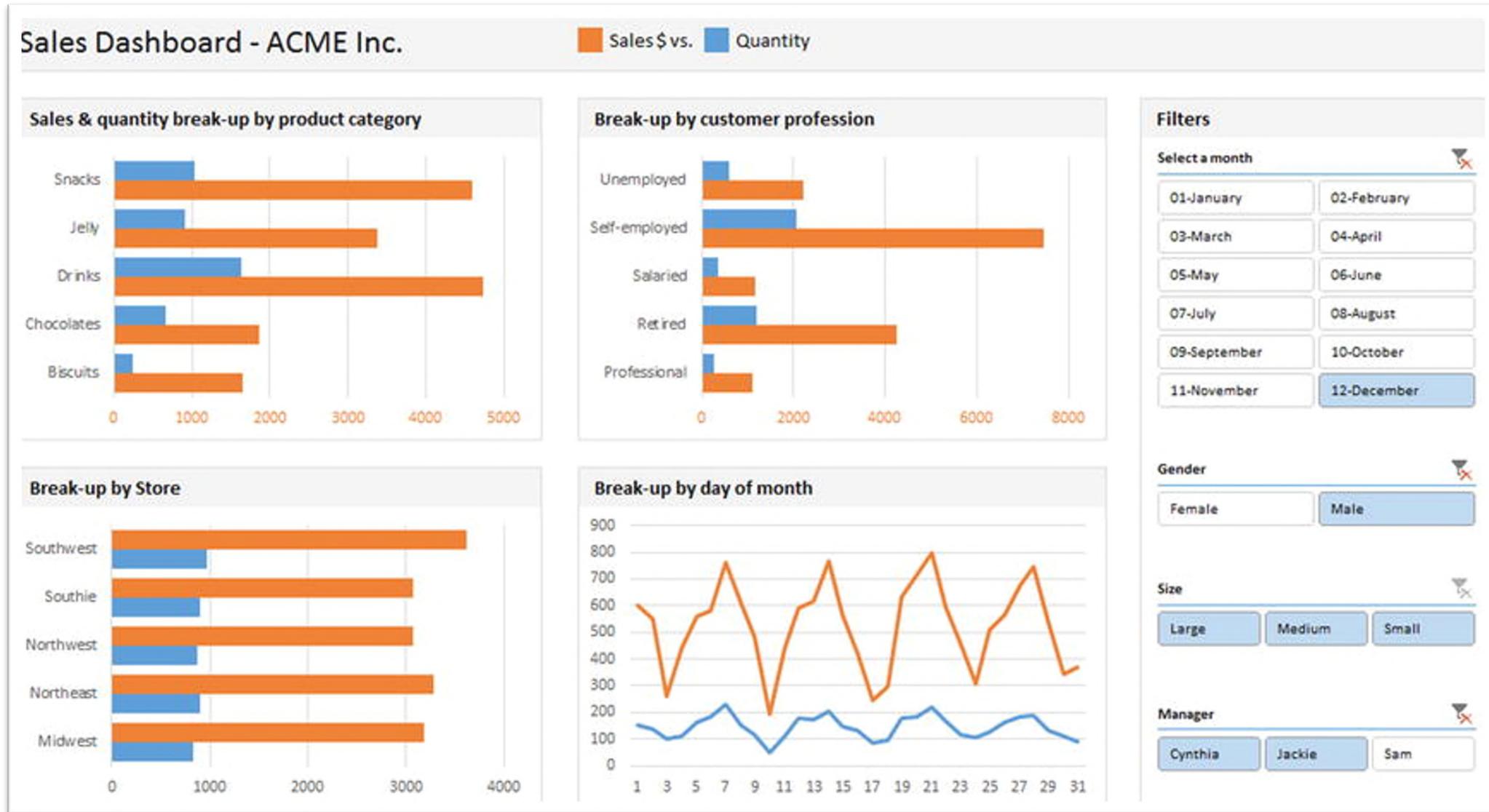
Menu Item	Menu	Item Qty	Gross Amount	Net Amount
Cheese Pizza	PIZZA	49	\$637.00	\$637.00
Burger	Event Menu	12	\$120.00	\$120.00
BBQ Chicken	PIZZA	6	\$90.00	\$75.00
\$33 MENU	Dinner	2	\$66.00	\$66.00
SMALL VEGETARIAN	PIZZA	5	\$56.10	\$56.10



# Analytical Reports

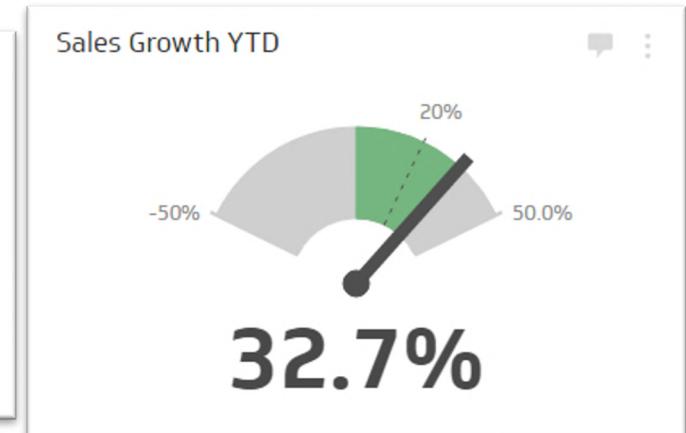
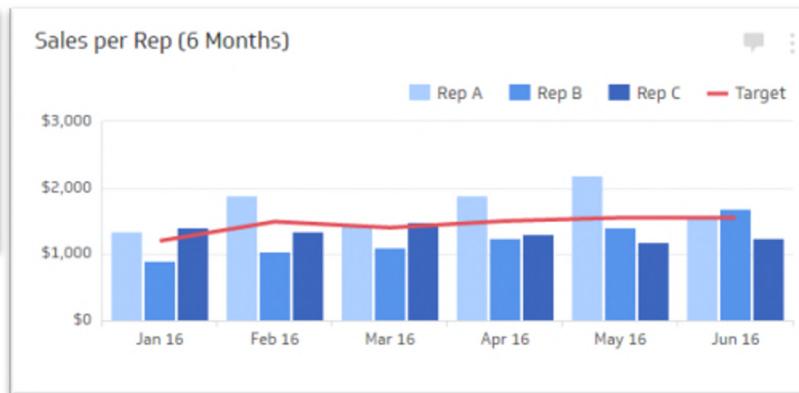
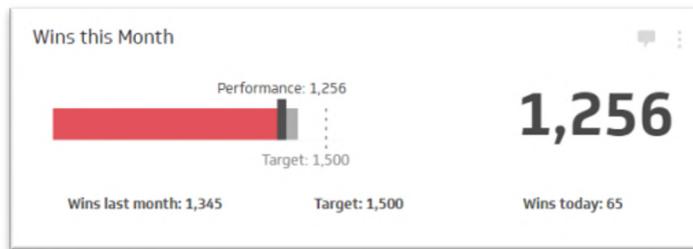
- Provides an overall understanding of business and operational activities
- Provides capabilities to view, understand, and summarize a large amount of information about your business through data visualization
- Letting the end-user view multidimensional charts and interact with data using data visualization tools and features such as drill-up/down, slice, dice, pivot
- Can be enabled to give the end user recommendations instead of just plain numbers
  - Analytical reports are based on historical data, statistics and provide predictive analysis such as forecasting, for a specific issue

# Analytical Reports



# Key Performance Indicators

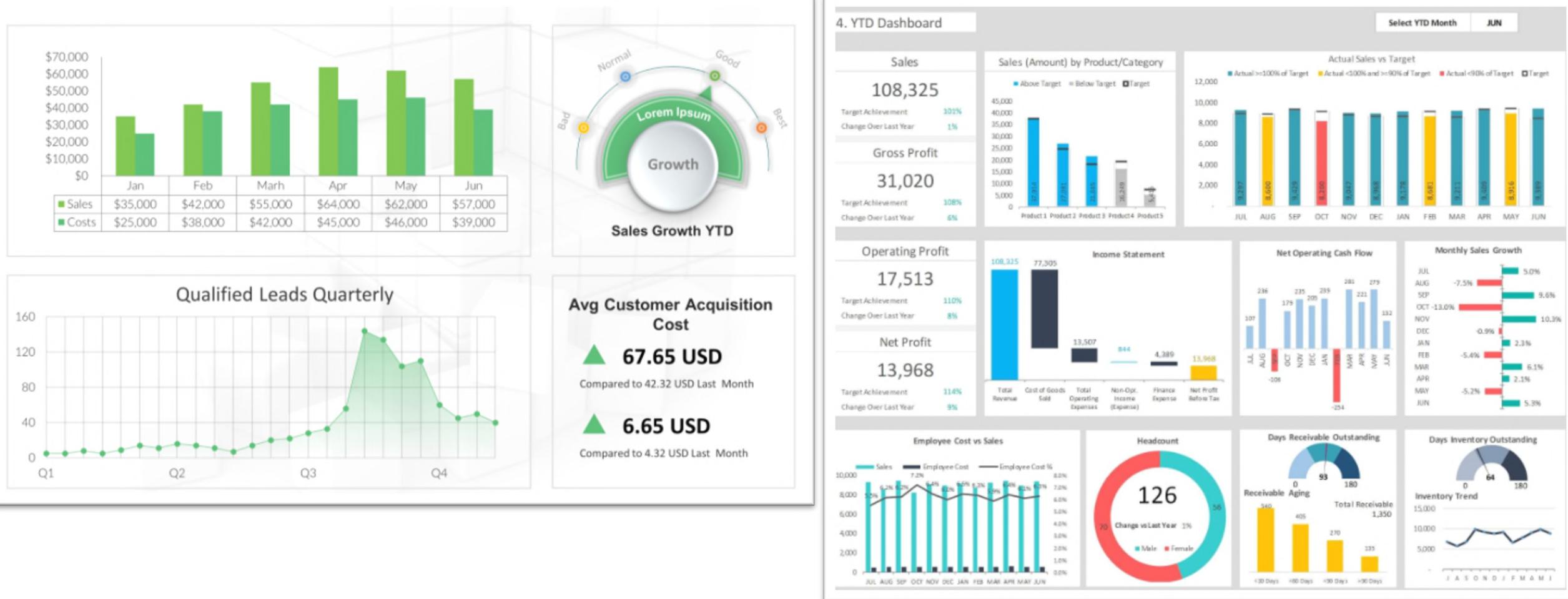
- A measurable value that demonstrates how effectively a company is achieving key business objectives
- Organizations use KPIs at multiple levels to evaluate their success at reaching targets
- High-level KPIs may focus on the overall performance of the business (used in scorecards), while low-level KPIs (used in dashboards) may focus on processes in departments such as sales, marketing, HR, support and others
- Start with the basics and understand what the organizational objectives are, what is the plan to achieve them, how to measure the success, and then define KPIs based on that



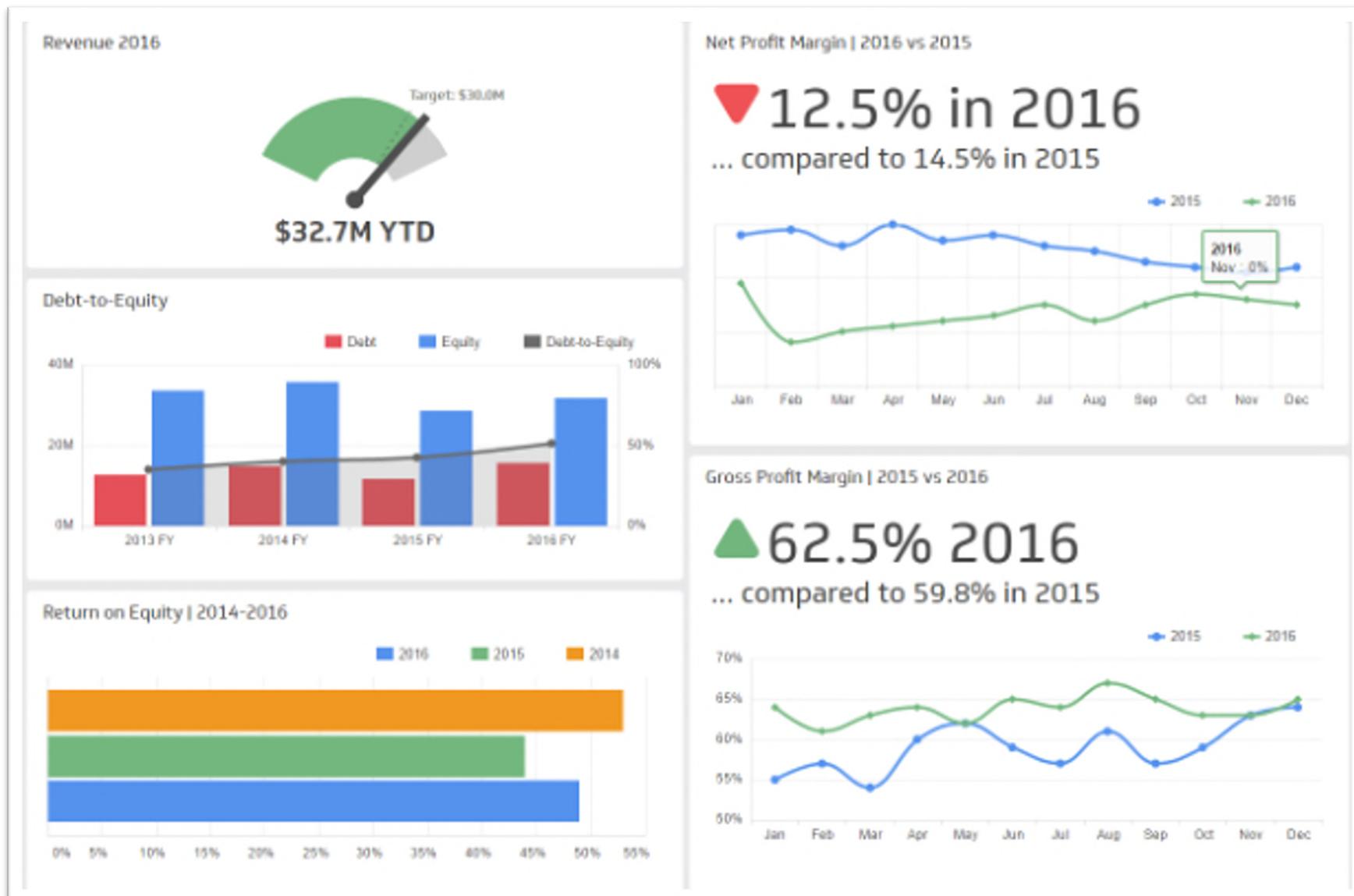
# Dashboards

- It should be a single page of key metrics represented by tables, charts, gauges, colours, and numbers, and arranged and consolidated in such a way that the consumer can identify and focus on areas requiring immediate attention or more investigation
  - A dashboard should confine its display to a single screen with no need for scrolling or switching among multiple screens
  - Does not go into detailed information
    - Focus may lead to additional dashboards or other reports
- A business dashboard offers at-a-glance insights based on key performance indicators (KPIs) and is an intuitive and visually pleasing way to consume data

# Dashboards



# Dashboards



# Dashboards



# Scorecards

- Scorecards offer organizations a snapshot of their current performance when compared to their goals
- They are useful tools for organizations which need to **manage performance** and **make strategic decisions** better based on the distance between current performance and the goal
  - As such, scorecards present a more static view of an organization at a point in time rather than a dynamic hub with live data to monitor success
- Scorecards serve to monitor **strategic goals relative to KPIs** and to make **decisions on a larger scale**
  - These decisions can include tracking the progress of a set strategy, measuring the efficiency of particular teams or departments towards meeting goals or even identifying problems and how they can be resolved
- Scorecards are generally periodic measures, usually updated at set intervals such as weekly or monthly

# Scorecards vs. Dashboards

- Unlike scorecards, dashboards are used as a monitoring tool in real-time
  - Data is constantly updated, giving organizations an opportunity to track their operational performance in real time
- As opposed to progress, dashboards measure performance, tracking metrics without comparing them to target values
- Dashboards are used daily in organizations as they offer a more operational view of success than scorecards' focus on strategic goals
- Data available in dashboards is used to provide a foundation for better decision making and more efficient day-to-day management of teams, resources, and expenses
- More importantly, dashboards help organizations view their historic data as a function of current performance. For example, companies can see their revenues over the past 12 months or measure their month-to-month sales growth on an ongoing basis
- For example, sales performance scorecard may show performance related to sales, revenue and profit against target or budget estimates

# Scorecards vs. Dashboards

Comparison based on	Dashboard	Scorecard
Purpose	Performance Monitoring	Performance Management
Parameters	Performance Metric	KPI (Metric + Target)
Measures	Performance	Progress ( Current value versus the target )
Updates information	Real Time Basis	Periodically ( Weekly/Monthly/Quarterly )
Focused On	Short Term Goal	Long Term Goal
Decision Influences	Daily Operations	Companies Policies
Nature of Decisions	Tactical	Strategic
Supported By	Individual Managers	Top Management
Provides	Snapshot of Business Performance	Trends and changes in business activity over period of time.
Nature of Data	Real Time data obtained	Summarized/ Consolidated

# Developing Dashboards: Some Guidelines

1. Define your dashboard audience and objective (requirements)
  1. Who is your audience?
  2. What do they do on a daily basis?
    - Differences in daily tasks will result in different goals and KPIs
      - Daily life of a sales agent who has to get all of their own leads is quite different from the daily work of a sales agent who has all of their leads supplied to them
      - Daily life of the sales manager who is in charge of all the sales agents is more different still
  3. What goals are they trying to reach?
    - e.g., achieving more sales, getting more leads, completion of projects, etc.

# Developing Dashboards: Some Guidelines

1. Define your dashboard audience and objective (cont.)
  4. What KPIs, if measured, will help them reach their goals?
    - KPIs must be set against goals: sales growth, customer acquisition, project completion %, etc.
  5. How are they currently viewing these KPIs?
    - What are their pain points in current process?
  6. How can I use storytelling to put my KPIs into context?
    - To get results, storytelling should become a primary focus.
    - Avoid providing all the information you got, but the best information to aid in getting the actionable insights they need
    - Interactive visualizations are especially relevant when you have a broad target audience

# Developing Dashboards: Some Guidelines

## 2. Select the right chart type for your data

### 1. What story do you want to tell?

- Data-driven storytelling is a powerful force as it takes stats and metrics and puts them into context through a narrative
- Knowing what kind of story or message you want to convey will help to choose the right data visualization types
  - Analyse trends (line charts, column charts, area charts)
  - Demonstrate a composition (pie charts, waterfall charts, stacked charts, map-based graphs)
  - Compare sets of values (bubble charts, spider charts, bar charts, columned visualizations, scatter plots)

### 2. Who do you want to tell it to?

- Based on the audience, which data visualization types will make the most tangible connection with the people will defer

### 3. How do you want to show your KPIs?

- Comparing data or demonstrating a relationship or demonstrating a trend

# Developing Dashboards: Some Guidelines

## 3. Don't forget about colour theory

- Avoid bad colour combinations: roughly 8% of men and 0.5% of women are colour-blind
- Avoid overlapping shades of colours that have similar brightness value
  - Use patterns and texture to show contrast.
  - The rule of thumb is to select 2, maximum 3, and stick to them across the board
  - Use both colours and as well as symbols

## 4. Build a balanced perspective

- Present a mix of past, real-time and predictive data to communicate your message
- Adding filters and other functionalities to your dashboard will make it easy for users to analyse, arrange, and view

## 5. Make sure your dashboard is mobile-optimized

- Small screen design greatly differentiates from large screen designs as you have much smaller space and different screen dimensions
- Professional dashboard tools will help you to adjust your desktop designs quickly and seamlessly

# Developing Dashboards: DO's and DON'Ts

- DO'S
  - DO focus on the needs of your audience
  - DO keep your dashboards as simple, clean, and minimalist as you can while including most important KPIs as necessary
  - DO make sure that your final dashboard is better than your audience's previous method of viewing their KPIs
  - DO tell a story, as stories are easily understood by the human mind
- DON'Ts
  - DON'T clutter your dashboard with too much data. This is the number one rule to follow!
    - Too much data → too hard to use → waste of time
  - DON'T use colours that are very similar in brightness as your main colours
    - Colour blind people won't be able to use your dashboard
  - DON'T make a "one size fits all" dashboard; make it with specific people and needs in mind
  - DON'T use pie charts except in cases where you are showing parts of a whole

# Self Service BI

- Self-service business intelligence (SSBI) empowers teams such as product developers, sales, finance, marketing, operations, and more to answer data questions with minimal technical support from IT
  - However, IT should ensure proper data governance is in place
- Traditional BI relied on IT departments to create data analysis processes for business goals
  - IT query the data to generate performance reports, forecast trends, and more; IT has much more control over data quality
  - Often meant bottlenecks and resource constraints that would throttle a business's speed to insight
- SSBI focuses on supporting the end user, allowing business users and analysts to be more involved in their own data analysis
  - Data teams are still involved; focus on how data is ingested and governed within the organization
  - Data engineers/scientists can use their expertise to dig deep into data mining or modelling projects rather than answering ad hoc reporting requests
  - Business users are given the ability to explore their data quickly within the well-considered boundaries that IT or data teams set up

# Top BI Tools

- Microsoft Power BI (good for SSBI too)
- Tableau (good for SSBI too)
- Qlik Sense and QlikView (good for SSBI too)
- SAP BusinessObjects BI Suite
- Sisense (great for SSBI too)
- SQL Server Reporting Services (SSRS)
- Microstrategy
- TIBCO Spotfire
- IBM Cognos Analytics
- Oracle Business Intelligence Enterprise Edition
- Amazon QuickSight
- SAS Enterprise Guide

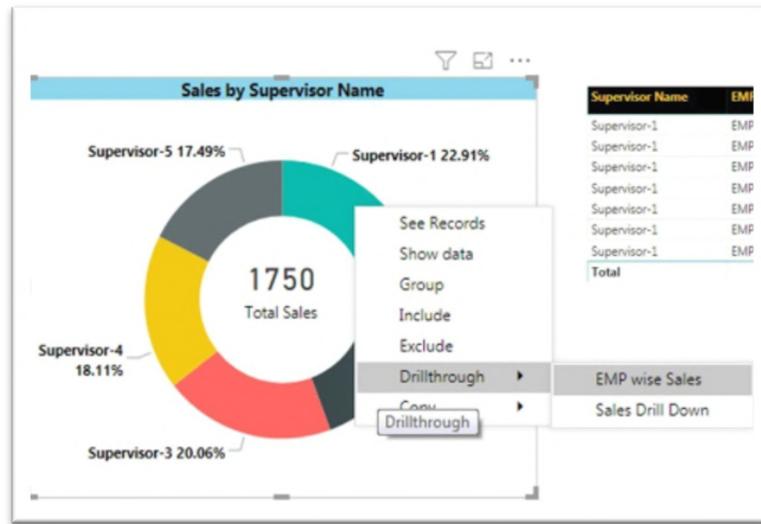
# Some Well-Known Features in Visualizations

- Drilldown: take users from a more general view of the data to a more specific one, enabling them to dig deeper into the layers in a hierarchy.
  - For example, in a report that shows sales revenue by country, the user can select a country and then drill down to see sales revenue by province, state, or city

Country	State Province	City	Postal Code	Internet Sales Amount
⊕				29358677.2207
⊕ Australia				2524846240.9802
⊕ Canada	⊕			29358677.2207
	⊕ Alberta			146793386.1035
	⊕ British Columbia	⊕		29358677.2207
		⊕ Burnaby		205510740.5449
		⊕ Cliffside		29358677.2207
			V8Y 1L1	29358677.2207
		⊕ Haney		58717354.4414
		⊕ Langford		58717354.4414
		⊕ Langley		58717354.4414

# Some Well-Known Features in Visualizations

- Drill-through: allow users to pass from one report to another while still analysing the same set of data
  - For example, in a tabular report that shows sales revenue by state, the user can drill through to reveal an analysis grid of the same data, or perhaps a heat map representing the data in visual form



A table titled "2009 Product Category Sales: Online and Reseller". The columns are:

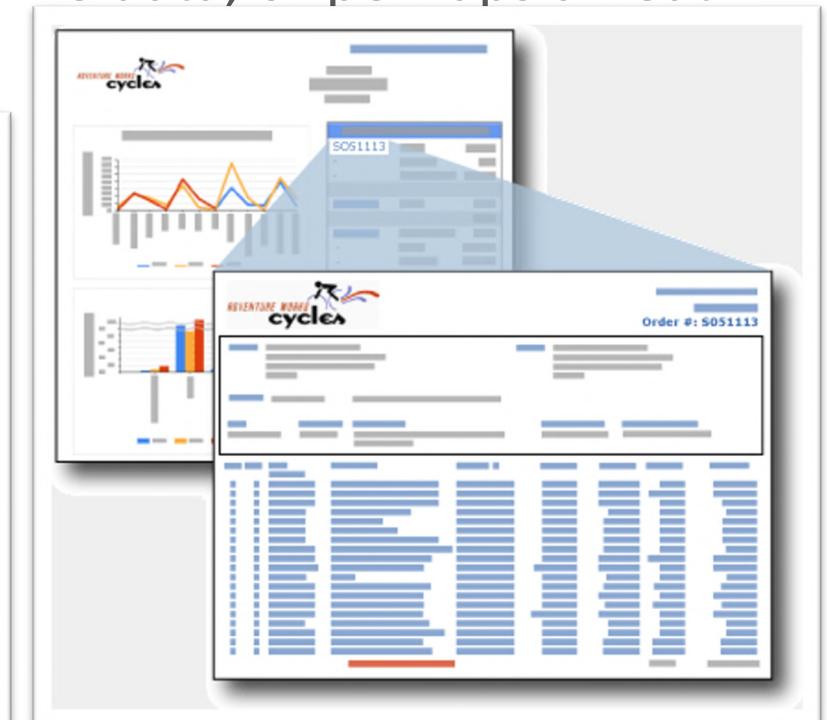
Product Category Name	Online		Reseller		Total	
	Net QTY	Net Sales	Net QTY	Net Sales	Net QTY	Net Sales
Audio	◆	◆	◆	◆	◆	◆
Cameras and camcorders	◆	◆	◆	◆	◆	◆
Cell phones	●	◆	◆	◆	●	◆
Computers	◆	●	◆	◆	◆	●
Games and Toys	●	◆	◆	◆	●	◆
Home Appliances	◆	●	◆	◆	◆	●
Music, Movies and Audio Books	◆	◆	◆	◆	◆	◆
TV and Video	◆	◆	◆	◆	◆	◆
Total						

Below the table is a timestamp: 5/12/2010 10:38:45 AM.

A second table titled "Sales and Returns for Category: Games and Toys" is shown below:

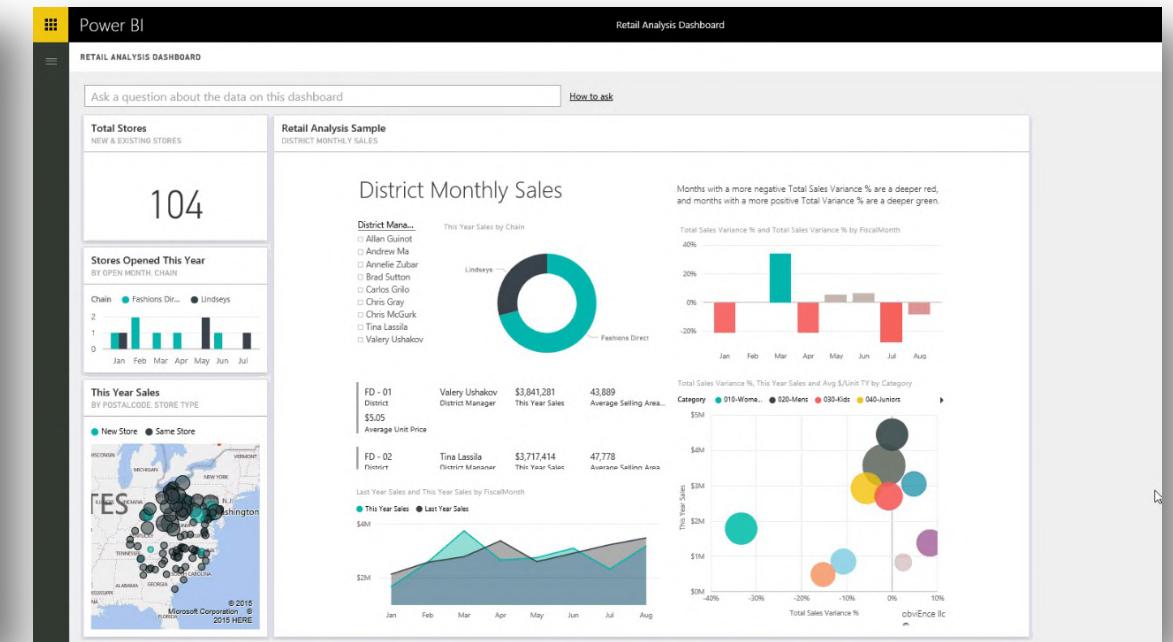
Subcategory	Online		Reseller		Total	
	Sales	Returns	Sales	Returns	Sales	Returns
Boxed Games	\$2,405,279.28	6,001.89	\$1,572,574.12	3,765.25	\$3,977,353.40	9,767.14
Download Games	\$41,673,923.69	637,964.74	\$5,747,775.69	65,771.32	\$47,421,599.39	703,736.06
Total	\$44,079,202.97	643,966.63	\$7,320,349.81	69,636.57	\$51,399,552.79	713,503.20

Below the table is a timestamp: 4/29/2010 4:19:22 PM.



# Some Well-Known Features in Visualizations

- Slice/Dice: data visualized in the report can be sliced and diced with slicers



# Some Well-Known Features in Visualizations

- Interactivity: report is interactive for user and it can be filtered, sliced/diced, drill-down, etc.
- Multi-pages: an have a number of pages in a report. Switching between pages is simply possible with the help of navigation pane at the bottom of the report
- Publish to web or a portal: report can be published to web or a locally hosted portal for other to use it
- Automatic refresh: base data set can be scheduled to be refreshed automatically or when the visualization is being opened
- “Ask a Question”: some tools provide the ability to query the data set using natural English language (one of the top features of the Power BI)
- Alerts: can define alerts

# Alerts

- Alerts are typically single metrics or phrases that are sent to users.
- Based on a key metric, a warning or notification can be provided to a user
- Usually an automated message or notification sent via email, SMS, etc., which indicates that a predefined event or error condition has occurred and that some action is needed
- Developers can easily set up automatic alerts and notifications to be sent to certain users when specific data values or conditions occur in a report
- For example, a store manager can be automatically informed when in-stock levels of a critical items fall below or rise above a certain level

# Challenges Related to Visualization Process

- Defining visualization is somewhat less technology driven
- While there are semi-AI driven business intelligence tools, the user is still the one who decides what format of visualization will be placed on a canvas, and what will be the data properties
- Thus, we sometimes tend to use unsuitable visualizations to tell a specific story!
- Pitfall 1: using the wrong visualization format
  - It's very easy to get lost in the forest of graphs, charts, and maps, so it will take some time to study the required or most suitable visualization for the business/audience
  - For example, using a spider chart when the object has only one characteristic to compare will make everyone scratch their heads or a line graph applied to compare multidimensional units, like seasonal sales across 3 countries, each with 10 provinces, is doomed to failure!
- Pitfall 2: using the wrong type of data
  - A very similar issue, but it takes a couple of times to understand what type of data can be applied to your tried-and-tested visualizations

# Challenges Related to Visualization Process

- Pitfall 3: visualization tools don't generate reports, you do!
  - Only a few really expensive tools can interpret some part of the information for you
- Pitfall 4: wrong tooling choice
  - If a decision is made to use a free tool or decide to mess with libraries in the tool, perhaps it can't be wrong for you.
  - But when we talk about the choice of vendor, things get more serious
    - Vendors of data visualization offer the whole service to make your life easier
    - But it is important to understand whether the service is scalable, so it covers the amount of data, frequency of updates, number of users (total/concurrent)
    - Visualization capabilities should also be considered, because industry-specific analytics may include exotic forms of visualizations and other features such as support for mobile devices

**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

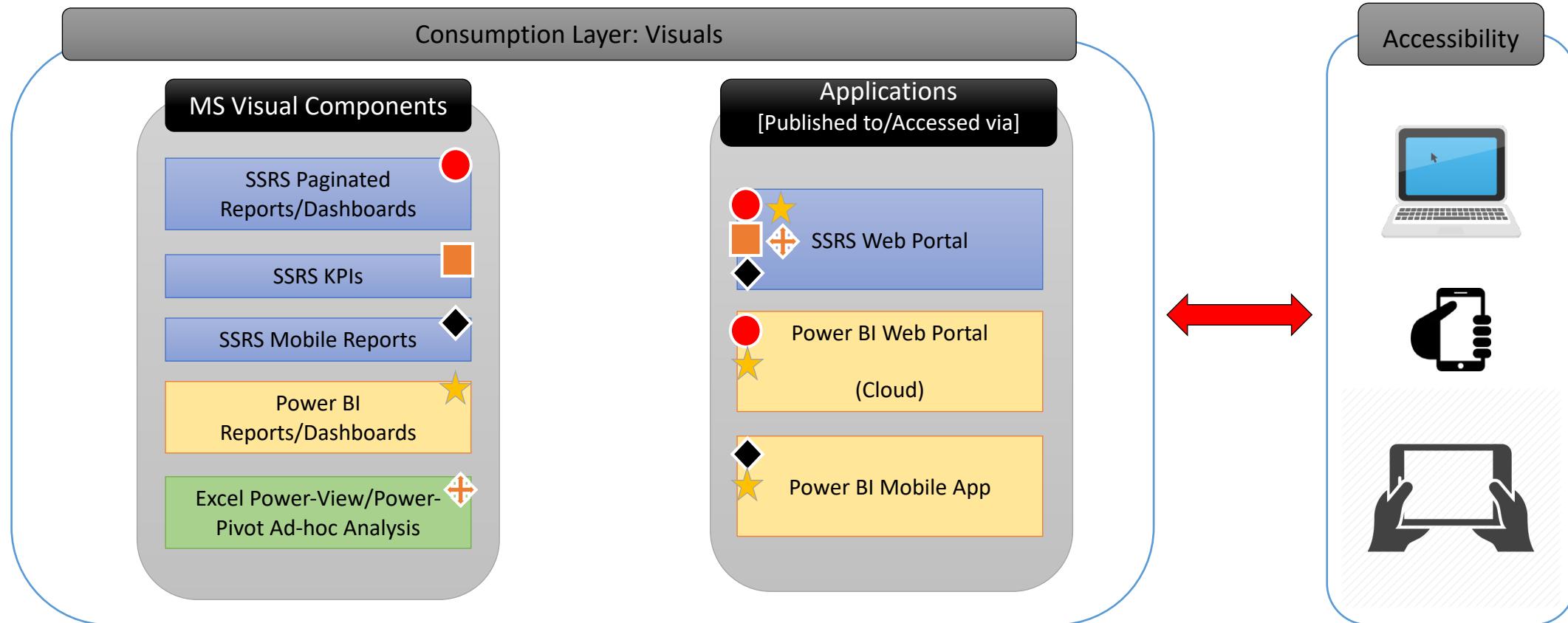
Lecture 09

Visualizations with SQL  
Server Reporting Services

# Content

- Overview of MS visualization tools
- SSRS components
- SSRS report types
  - Paginated reports
  - Mobile reports
  - KPIs
- SSRS specific concepts

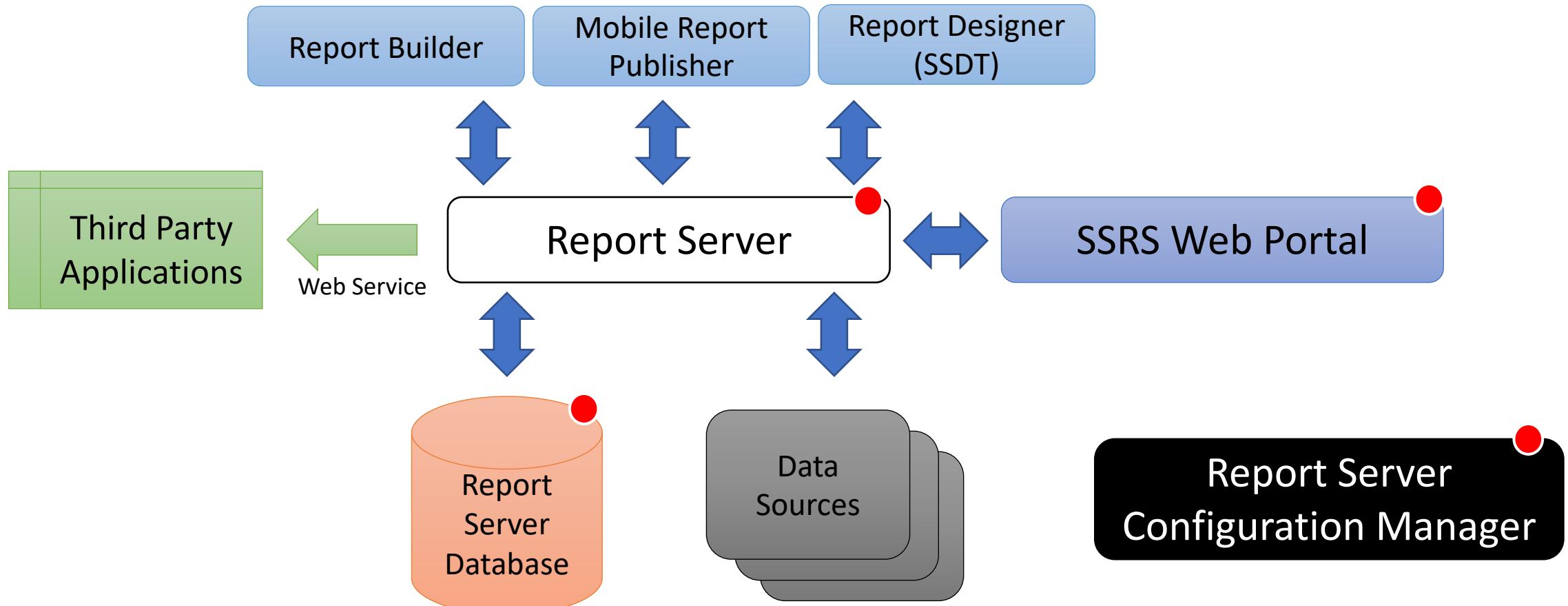
# MS Tools for Visualization: Overview



# SQL Server Reporting Services

SQL Server Reporting Services (SSRS) is a platform for **creating, publishing, and managing** mobile and paginated **reports/dashboards**, then **delivering** them to the right **users** in different ways, such as via a **web browser**, on their **mobile device**, or via **email**.

# SSRS Components



# SSRS Components

- Report Server
  - A computer that has an instance of Reporting Services installed
  - Internally stores items such as paginated and mobile reports, report-related items and resources, schedules, and subscriptions
  - User interface to interact with the Report Server content is ***SSRS web portal***
  - Provides a web service known as ***Reporting Services web service***, which is an API to interact with Report Server items programmatically through scripts or third party applications

# SSRS Components

- Report Designer (SSDT):
  - A feature of SSDT, used to design paginated reports
  - Once designed, paginated reports can be deployed to a report server
  - Mostly used by report designers/developers
- Report Builder:
  - A stand-alone application for creating paginated reports
  - Once designed, paginated reports can be deployed to a report server
  - Can be used as an ad-hoc report development tool as this is used by most tech-savvy business users who prefer to work in a stand-alone environment instead of using Report Designer in Visual Studio

# SSRS Components

- Mobile Report Publisher:
  - Allows you to create and publish SSRS mobile reports to the SSRS web portal
  - Can display SSRS mobile reports in Power BI mobile application
  - Power BI mobile application and Report Server needs to be configured
- SSRS Web Portal:
  - This portal comes with SQL Server which can be used to publish KPIs, mobile reports, paginated reports, Excel workbooks and Power BI desktop files (.pbix)
  - A web based user interface where users log-in and interact with published reports and SSRS features such as subscriptions
  - Authorized users can launch Mobile Report Publisher and Report Builder from the web portal for report development

# SSRS Components: SSRS Web Portal (2016)

SSRS Web Portal:  
Landing Page

The screenshot shows the SSRS Web Portal (2016) landing page. At the top, there's a navigation bar with 'Favorites' and 'Browse' buttons, followed by a toolbar with 'Manage Folder', 'New', 'Upload', 'View', 'Search', and a user icon. Below the toolbar, there are four main categories: 'Data Sources', 'Datasets', 'Mobile Reports', and 'Paginated Reports'. The main content area is divided into several sections:

- KPIs (10):** A grid of ten KPI cards with metrics like Annual Sales (\$1.15M), Annual Spending (949,607), Customer Retention (62%), Customer Satisfaction (88%), IT Spending (\$250K), Monthly Sales (\$126K), New Customers (162), Quarterly Sales (\$353K), Sales MTD (\$505K), and Service Downtime (0).
- Mobile Reports:** A grid of five mobile report cards, including Marketing Scorecard, New Mobile Report, Parameters, RS Demo, Sales vs. Previous Year, Sales vs. Targets, and Sample Parameter Report.
- Paginated Reports:** A grid of five paginated report cards, including Customers\_Near\_Stores, Daily Store Report, Daily Store Report 2, Employee\_Sales\_Summary, Sales\_by\_Region, Sales\_Order\_Detail, and Store\_Contacts.

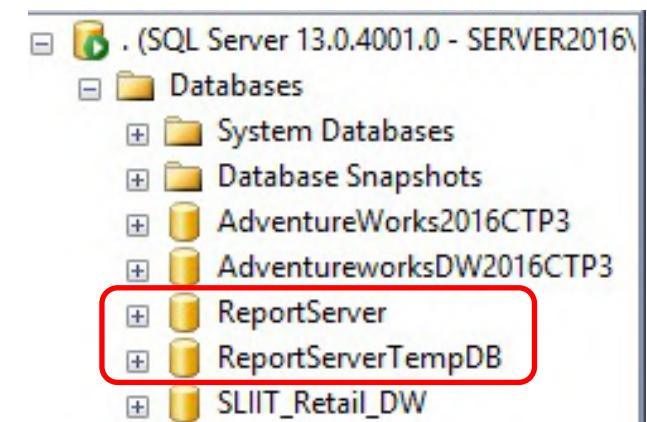
Red dashed boxes highlight specific areas of the interface, each labeled with its corresponding component name:

- Options:** The top toolbar area.
- Folder structure:** The area below the toolbar, containing 'Data Sources', 'Datasets', 'Mobile Reports', and 'Paginated Reports' buttons.
- KPIs:** The section containing the ten KPI cards.
- Mobile Reports:** The section containing the five mobile report cards.
- Paginated Reports:** The section containing the five paginated report cards.

# SSRS Components

- Report Server Database:

- Report server related properties, objects, and metadata are stored in an SQL Server database. Default name of the database is “ReportServer”
- Report Server database contains:
  - Published reports and related meta data
  - Shared data sources, and related meta data
  - Security settings that are associated with items
  - Subscription definitions
  - Report snapshots (which include query results) and snapshot history
  - System properties and system-level security settings
  - Report execution log data

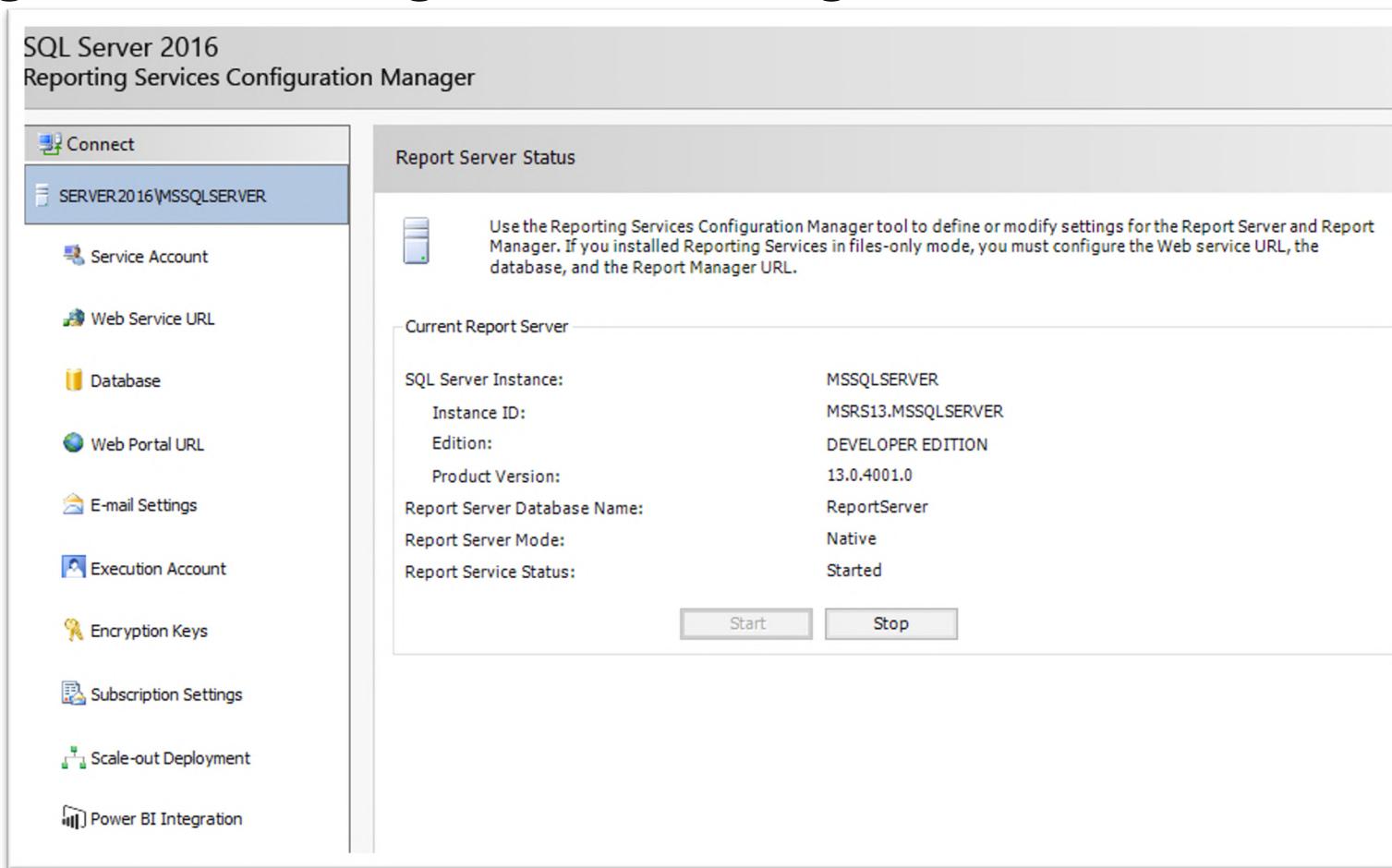


# SSRS Components

- Report Server Configuration Manager:
  - A tool to manage Reporting Services instance related configurations
  - Some of the tasks this tool used for is:
    - Configure the Report Server service account
    - Create and configure one or more Reporting Services web service URL
    - Configure the Report Manager (SSRS web portal) URL
    - Create and configure the report server database
    - Configure an SMTP server for e-mail delivery
    - Integration of Power BI

# SSRS Components

- Reporting Service Configuration Manager interface



# SSRS Report Types

- Paginated reports
- Mobile reports
- KPIs

# Paginated Reports

- Document-style, operational reports
- Good for generating reports with fixed-layout and optimized for print-friendly formats such as PDF
- Development tools:
  - SSDT
  - Report Builder
- User accessibility:
  - Paginated reports are published to SSRS web portal (accessed via a web browser)
  - Certain report items can be “pinned” to Power BI portal (accessed via a web browser)

# Paginated Reports

- Types of visualizations include table format, matrix, list, charts (area charts, bar charts, column charts, line charts, pie charts, etc.), gauge and map
- Reports can be exported in different file formats such as .pdf, .doc(x), .xls(x), .ppt(x), .csv
- Features:
  - Drill-down: allows expand or collapse a section of a report to show or hide detail data.
  - Drill-through: user clicks on a link or an area in a chart with summarized data, which then opens a separate, related report to show detailed data

# Paginated Reports

- Drill-down report

The diagram shows two overlapping reports from the Adventure Works Cycles database. The top report displays a hierarchy of geographical regions: Australia, Canada, Central, France, Germany, Northeast, Northwest, Southeast, Southwest, and United Kingdom. The bottom report is a detailed view of the United Kingdom, showing sales records for two employees: Shu Ito and Linda Mitchell. The report includes a table with columns for Employee Name, Order ID, and Total Sales.

Employee Name	Order ID	Total Sales
Shu Ito	S045784	\$5,001.87
Shu Ito	S045559	\$1668.61
Shu Ito	S045557	\$43,167.84
Shu Ito	S045554	\$33,797.43
Shu Ito	S045555	\$87,266.26
Shu Ito	S045664	\$83,297.69
Shu Ito	S045566	\$1,159.98
Shu Ito	S045572	\$14,190.49
Linda Mitchell	S045531	\$5,657.17
Linda Mitchell	S045523	\$1,948.05
Linda Mitchell	S045339	\$36,821.21
Linda Mitchell	S045343	\$27,940.33
Linda Mitchell	S045311	\$24,259.49

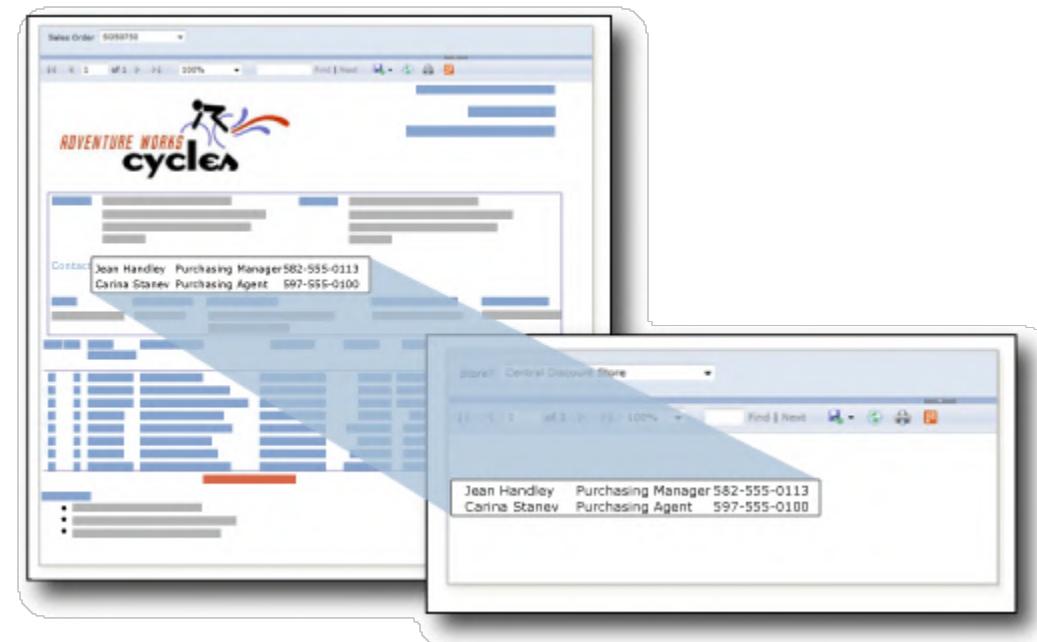
- Drill-through report

The diagram shows a complex reporting interface for the Adventure Works Cycles database. It features a main dashboard with a line chart and a detailed report for Order # S051113. The detailed report includes a table of sales items and a large, zoomed-in bar chart at the bottom. The main dashboard also displays a smaller version of the detailed report.

Order ID	Product Name	Quantity	Unit Price	Total
S051113	Mountain Bike	1	\$1,299.00	\$1,299.00
S051113	Urban Bike	1	\$799.00	\$799.00
S051113	Hybrid Bike	1	\$599.00	\$599.00
S051113	Electric Bike	1	\$1,999.00	\$1,999.00
S051113	Commuter Bike	1	\$499.00	\$499.00
S051113	Road Bike	1	\$999.00	\$999.00
S051113	Cross Bike	1	\$699.00	\$699.00
S051113	MTB Components	1	\$1,199.00	\$1,199.00
S051113	Urban Components	1	\$799.00	\$799.00
S051113	Hybrid Components	1	\$599.00	\$599.00
S051113	EBike Components	1	\$1,999.00	\$1,999.00
S051113	Commuter Components	1	\$499.00	\$499.00
S051113	Road Components	1	\$999.00	\$999.00
S051113	Cross Components	1	\$699.00	\$699.00
S051113	MTB Parts	1	\$1,199.00	\$1,199.00
S051113	Urban Parts	1	\$799.00	\$799.00
S051113	Hybrid Parts	1	\$599.00	\$599.00
S051113	EBike Parts	1	\$1,999.00	\$1,999.00
S051113	Commuter Parts	1	\$499.00	\$499.00
S051113	Road Parts	1	\$999.00	\$999.00
S051113	Cross Parts	1	\$699.00	\$699.00
S051113	MTB Tools	1	\$1,199.00	\$1,199.00
S051113	Urban Tools	1	\$799.00	\$799.00
S051113	Hybrid Tools	1	\$599.00	\$599.00
S051113	EBike Tools	1	\$1,999.00	\$1,999.00
S051113	Commuter Tools	1	\$499.00	\$499.00
S051113	Road Tools	1	\$999.00	\$999.00
S051113	Cross Tools	1	\$699.00	\$699.00
S051113	MTB Maintenance	1	\$1,199.00	\$1,199.00
S051113	Urban Maintenance	1	\$799.00	\$799.00
S051113	Hybrid Maintenance	1	\$599.00	\$599.00
S051113	EBike Maintenance	1	\$1,999.00	\$1,999.00
S051113	Commuter Maintenance	1	\$499.00	\$499.00
S051113	Road Maintenance	1	\$999.00	\$999.00
S051113	Cross Maintenance	1	\$699.00	\$699.00
S051113	MTB Accessories	1	\$1,199.00	\$1,199.00
S051113	Urban Accessories	1	\$799.00	\$799.00
S051113	Hybrid Accessories	1	\$599.00	\$599.00
S051113	EBike Accessories	1	\$1,999.00	\$1,999.00
S051113	Commuter Accessories	1	\$499.00	\$499.00
S051113	Road Accessories	1	\$999.00	\$999.00
S051113	Cross Accessories	1	\$699.00	\$699.00
S051113	MTB Apparel	1	\$1,199.00	\$1,199.00
S051113	Urban Apparel	1	\$799.00	\$799.00
S051113	Hybrid Apparel	1	\$599.00	\$599.00
S051113	EBike Apparel	1	\$1,999.00	\$1,999.00
S051113	Commuter Apparel	1	\$499.00	\$499.00
S051113	Road Apparel	1	\$999.00	\$999.00
S051113	Cross Apparel	1	\$699.00	\$699.00
S051113	MTB Shoes	1	\$1,199.00	\$1,199.00
S051113	Urban Shoes	1	\$799.00	\$799.00
S051113	Hybrid Shoes	1	\$599.00	\$599.00
S051113	EBike Shoes	1	\$1,999.00	\$1,999.00
S051113	Commuter Shoes	1	\$499.00	\$499.00
S051113	Road Shoes	1	\$999.00	\$999.00
S051113	Cross Shoes	1	\$699.00	\$699.00
S051113	MTB Helmets	1	\$1,199.00	\$1,199.00
S051113	Urban Helmets	1	\$799.00	\$799.00
S051113	Hybrid Helmets	1	\$599.00	\$599.00
S051113	EBike Helmets	1	\$1,999.00	\$1,999.00
S051113	Commuter Helmets	1	\$499.00	\$499.00
S051113	Road Helmets	1	\$999.00	\$999.00
S051113	Cross Helmets	1	\$699.00	\$699.00
S051113	MTB Gloves	1	\$1,199.00	\$1,199.00
S051113	Urban Gloves	1	\$799.00	\$799.00
S051113	Hybrid Gloves	1	\$599.00	\$599.00
S051113	EBike Gloves	1	\$1,999.00	\$1,999.00
S051113	Commuter Gloves	1	\$499.00	\$499.00
S051113	Road Gloves	1	\$999.00	\$999.00
S051113	Cross Gloves	1	\$699.00	\$699.00
S051113	MTB Brakes	1	\$1,199.00	\$1,199.00
S051113	Urban Brakes	1	\$799.00	\$799.00
S051113	Hybrid Brakes	1	\$599.00	\$599.00
S051113	EBike Brakes	1	\$1,999.00	\$1,999.00
S051113	Commuter Brakes	1	\$499.00	\$499.00
S051113	Road Brakes	1	\$999.00	\$999.00
S051113	Cross Brakes	1	\$699.00	\$699.00
S051113	MTB Pedals	1	\$1,199.00	\$1,199.00
S051113	Urban Pedals	1	\$799.00	\$799.00
S051113	Hybrid Pedals	1	\$599.00	\$599.00
S051113	EBike Pedals	1	\$1,999.00	\$1,999.00
S051113	Commuter Pedals	1	\$499.00	\$499.00
S051113	Road Pedals	1	\$999.00	\$999.00
S051113	Cross Pedals	1	\$699.00	\$699.00
S051113	MTB Chain	1	\$1,199.00	\$1,199.00
S051113	Urban Chain	1	\$799.00	\$799.00
S051113	Hybrid Chain	1	\$599.00	\$599.00
S051113	EBike Chain	1	\$1,999.00	\$1,999.00
S051113	Commuter Chain	1	\$499.00	\$499.00
S051113	Road Chain	1	\$999.00	\$999.00
S051113	Cross Chain	1	\$699.00	\$699.00
S051113	MTB Chainring	1	\$1,199.00	\$1,199.00
S051113	Urban Chainring	1	\$799.00	\$799.00
S051113	Hybrid Chainring	1	\$599.00	\$599.00
S051113	EBike Chainring	1	\$1,999.00	\$1,999.00
S051113	Commuter Chainring	1	\$499.00	\$499.00
S051113	Road Chainring	1	\$999.00	\$999.00
S051113	Cross Chainring	1	\$699.00	\$699.00
S051113	MTB Crankset	1	\$1,199.00	\$1,199.00
S051113	Urban Crankset	1	\$799.00	\$799.00
S051113	Hybrid Crankset	1	\$599.00	\$599.00
S051113	EBike Crankset	1	\$1,999.00	\$1,999.00
S051113	Commuter Crankset	1	\$499.00	\$499.00
S051113	Road Crankset	1	\$999.00	\$999.00
S051113	Cross Crankset	1	\$699.00	\$699.00
S051113	MTB Bottom Bracket	1	\$1,199.00	\$1,199.00
S051113	Urban Bottom Bracket	1	\$799.00	\$799.00
S051113	Hybrid Bottom Bracket	1	\$599.00	\$599.00
S051113	EBike Bottom Bracket	1	\$1,999.00	\$1,999.00
S051113	Commuter Bottom Bracket	1	\$499.00	\$499.00
S051113	Road Bottom Bracket	1	\$999.00	\$999.00
S051113	Cross Bottom Bracket	1	\$699.00	\$699.00
S051113	MTB Fork	1	\$1,199.00	\$1,199.00
S051113	Urban Fork	1	\$799.00	\$799.00
S051113	Hybrid Fork	1	\$599.00	\$599.00
S051113	EBike Fork	1	\$1,999.00	\$1,999.00
S051113	Commuter Fork	1	\$499.00	\$499.00
S051113	Road Fork	1	\$999.00	\$999.00
S051113	Cross Fork	1	\$699.00	\$699.00
S051113	MTB Stem	1	\$1,199.00	\$1,199.00
S051113	Urban Stem	1	\$799.00	\$799.00
S051113	Hybrid Stem	1	\$599.00	\$599.00
S051113	EBike Stem	1	\$1,999.00	\$1,999.00
S051113	Commuter Stem	1	\$499.00	\$499.00
S051113	Road Stem	1	\$999.00	\$999.00
S051113	Cross Stem	1	\$699.00	\$699.00
S051113	MTB Handlebar	1	\$1,199.00	\$1,199.00
S051113	Urban Handlebar	1	\$799.00	\$799.00
S051113	Hybrid Handlebar	1	\$599.00	\$599.00
S051113	EBike Handlebar	1	\$1,999.00	\$1,999.00
S051113	Commuter Handlebar	1	\$499.00	\$499.00
S051113	Road Handlebar	1	\$999.00	\$999.00
S051113	Cross Handlebar	1	\$699.00	\$699.00
S051113	MTB Seatpost	1	\$1,199.00	\$1,199.00
S051113	Urban Seatpost	1	\$799.00	\$799.00
S051113	Hybrid Seatpost	1	\$599.00	\$599.00
S051113	EBike Seatpost	1	\$1,999.00	\$1,999.00
S051113	Commuter Seatpost	1	\$499.00	\$499.00
S051113	Road Seatpost	1	\$999.00	\$999.00
S051113	Cross Seatpost	1	\$699.00	\$699.00
S051113	MTB Saddle	1	\$1,199.00	\$1,199.00
S051113	Urban Saddle	1	\$799.00	\$799.00
S051113	Hybrid Saddle	1	\$599.00	\$599.00
S051113	EBike Saddle	1	\$1,999.00	\$1,999.00
S051113	Commuter Saddle	1	\$499.00	\$499.00
S051113	Road Saddle	1	\$999.00	\$999.00
S051113	Cross Saddle	1	\$699.00	\$699.00
S051113	MTB Pedal Cleat	1	\$1,199.00	\$1,199.00
S051113	Urban Pedal Cleat	1	\$799.00	\$799.00
S051113	Hybrid Pedal Cleat	1	\$599.00	\$599.00
S051113	EBike Pedal Cleat	1	\$1,999.00	\$1,999.00
S051113	Commuter Pedal Cleat	1	\$499.00	\$499.00
S051113	Road Pedal Cleat	1	\$999.00	\$999.00
S051113	Cross Pedal Cleat	1	\$699.00	\$699.00
S051113	MTB Pedal Spindle	1	\$1,199.00	\$1,199.00
S051113	Urban Pedal Spindle	1	\$799.00	\$799.00
S051113	Hybrid Pedal Spindle	1	\$599.00	\$599.00
S051113	EBike Pedal Spindle	1	\$1,999.00	\$1,999.00
S051113	Commuter Pedal Spindle	1	\$499.00	\$499.00
S051113	Road Pedal Spindle	1	\$999.00	\$999.00
S051113	Cross Pedal Spindle	1	\$699.00	\$699.00
S051113	MTB Pedal Axle	1	\$1,199.00	\$1,199.00
S051113	Urban Pedal Axle	1	\$799.00	\$799.00
S051113	Hybrid Pedal Axle	1	\$599.00	\$599.00
S051113	EBike Pedal Axle	1	\$1,999.00	\$1,999.00
S051113	Commuter Pedal Axle	1	\$499.00	\$499.00
S051113	Road Pedal Axle	1	\$999.00	\$999.00
S051113	Cross Pedal Axle	1	\$699.00	\$699.00
S051113	MTB Pedal Crank Arm	1	\$1,199.00	\$1,199.00
S051113	Urban Pedal Crank Arm	1	\$799.00	\$799.00
S051113	Hybrid Pedal Crank Arm	1	\$599.00	\$599.00
S051113	EBike Pedal Crank Arm	1	\$1,999.00	\$1,999.00
S051113	Commuter Pedal Crank Arm	1	\$499.00	\$499.00
S051113	Road Pedal Crank Arm	1	\$999.00	\$999.00
S051113	Cross Pedal Crank Arm	1	\$699.00	\$699.00
S051113	MTB Pedal Crankset	1	\$1,199.00	\$1,199.00
S051113	Urban Pedal Crankset	1	\$799.00	\$799.00
S051113	Hybrid Pedal Crankset	1	\$599.00	\$599.00
S051113	EBike Pedal Crankset	1	\$1,999.00	\$1,999.00
S051113	Commuter Pedal Crankset	1	\$499.00	\$499.00
S051113	Road Pedal Crankset	1	\$999.00	\$999.00
S051113	Cross Pedal Crankset	1	\$699.00	\$699.00
S051113	MTB Pedal Crankset	1	\$1,199.00	\$1,199.00
S051113	Urban Pedal Crankset	1	\$799.00	\$799.00
S051113	Hybrid Pedal Crankset	1	\$599.00	\$599.00
S051113	EBike Pedal Crankset	1	\$1,999.00	\$1,999.00
S051113	Commuter Pedal Crankset	1	\$499.00	\$499.00
S051113	Road Pedal Crankset	1	\$999.00	\$999.00
S051113	Cross Pedal Crankset	1	\$699.00	\$699.00
S051113	MTB Pedal Crankset	1	\$1,199.00	\$1,199.00
S051113	Urban Pedal Crankset	1	\$799.00	\$799.00
S051113	Hybrid Pedal Crankset	1	\$5	

# Paginated Reports

- Features (cont.):
  - Sub report: a report displayed within the body of another report (report within a report); similar to a frame in a web page



# Paginated Reports

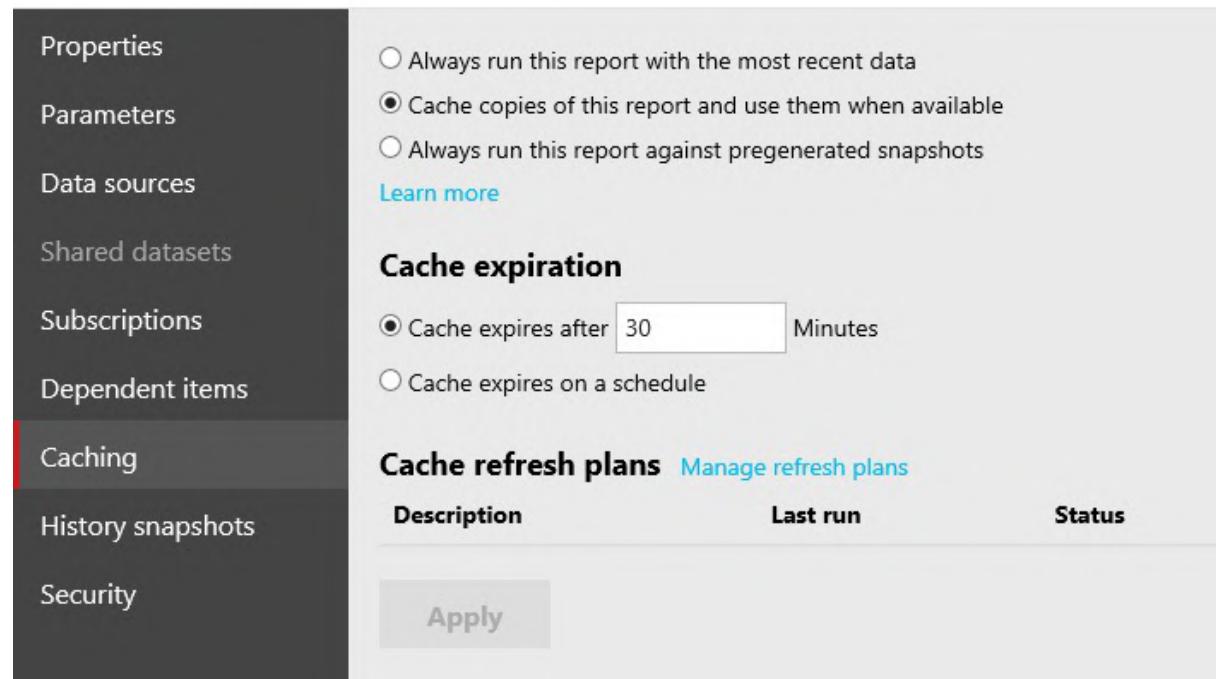
- Features (cont.):
  - Cached reports: a saved copy of a compiled (executed) report with its data (has a lifetime)
    - Used to improve performance by delivering the saved copy when the same report is executed multiple times with same parameter values
    - For example, when report “test” with parameter value ‘abc’ is executed for the first time,
      - report will connect to data source and get data and display (comparatively slow),
      - then create a copy and store for a defined time period and,
      - when the same report is executed again by passing the same parameter values (by any user), if the cached copy is not expired, second request will get the report from the cached copy (faster)

# Paginated Reports

- Features (cont.):

- Cached reports:  Manage Test Report

[Home](#) > [Test Report](#) > [Manage](#) > Caching



The screenshot shows the 'Manage Test Report' interface for a 'Test Report'. On the left, a sidebar menu lists several options: Properties, Parameters, Data sources, Shared datasets, Subscriptions, Dependent items, Caching (which is selected and highlighted in red), History snapshots, and Security. The main content area is titled 'Caching' and contains the following settings:

- Run this report:** Radio buttons for "Always run this report with the most recent data", "Cache copies of this report and use them when available" (which is selected), and "Always run this report against pregenerated snapshots". A "Learn more" link is provided.
- Cache expiration:** Radio buttons for "Cache expires after [input field set to 30] Minutes" (selected) and "Cache expires on a schedule".
- Cache refresh plans:** A link to "Manage refresh plans". Below it is a table with columns: Description, Last run, and Status. The table currently has one row with a single entry: "Apply".

# Paginated Reports

- Features (cont.):
  - Snapshots: a report that contains data which was retrieved and stored at a specific point in time (has a lifetime)
    - Snapshots can be scheduled and persisted
    - For example report “test” can be scheduled to be executed and store a snapshot with parameter value ‘abc’, every morning at 9.00 am
    - Whenever the report is executed, if the parameter value passed is ‘abc’, and if the snapshot is not expired, report request will get the report from the snapshot
    - Snapshot data will get updated based on the schedule
    - A history of snapshots is maintained with SSRS server

# Paginated Reports

- Features (cont.):

- Snapshots:



Home > Test Report > Manage > Caching

The screenshot shows the 'Manage Test Report' interface with the 'Caching' tab selected in the left sidebar. The main area displays options for managing snapshots, including a list of available snapshots and a section for creating cache snapshots on a schedule. The 'Report-specific schedule' option is selected, indicating a daily run at 9:00 AM starting from April 25, 2018.

Properties  
Parameters  
Data sources  
Shared datasets  
Subscriptions  
Dependent items  
**Caching**  
History snapshots  
Security

Manage Test Report

Home > Test Report > Manage > Caching

Always run this report with the most recent data  
Cache copies of this report and use them when available  
Always run this report against pregenerated snapshots

[Learn more](#)

**Cache snapshots**

Create cache snapshots on a schedule

Shared schedule [Select a shared schedule](#)

Report-specific schedule [Edit schedule](#)

At 9:00 AM every day, starting 4/25/2018

Create a cache snapshot when I click Apply on this page

**Apply**

# Paginated Reports

- Features (cont.):
  - Subscriptions and delivery: a request to deliver a report in a specified file format (pdf, word, etc.), at a specific time
    - Subscriptions can be used to schedule and then automate the delivery of a report
    - Delivery can be done to an email inbox or to a file share
    - An alternative to running a report on demand
    - Two types:
      - Standard subscriptions
      - Data driven subscription

# Paginated Reports

The screenshot displays a paginated report from SQL Server Reporting Services. The report includes the following components:

- Report Header:** Shows the URL <http://localhost/reports/powerbi/Sample%20>, the title "Sample Sales Report - SQL S...", and the "Home - SQL Server Reporting..." link.
- Report Navigation:** Includes "Favorites", "Browse", "Comments", and a user profile for "ccundy".
- Report Content:** A map titled "Total Sales by Country" showing sales distribution across continents. A legend indicates: United States (green circle), Canada (black dot), Australia (red dot), United Kingdom (yellow dot), France (grey dot), and Germany (light blue dot).
- Summary Metrics:** A summary box containing:
  - Total Sales:** \$109.23M
  - Total Units Sold:** 272K
  - Total Gross Profit:** \$12.49M
- Reseller Total Sales by Month and Calendar Year:** A line chart showing monthly sales for three years: 2011 (cyan), 2012 (black), and 2013 (red). The Y-axis ranges from \$0M to \$5M.
- Internet Total Sales by Month and Calendar Year:** A line chart showing monthly sales for three years: 2011 (cyan), 2012 (black), and 2013 (red). The Y-axis ranges from \$0M to \$2M.
- Table:** A detailed table showing Internet Total Sales and Reseller Total Sales by Country and State Province for Australia and Canada.

# Mobile Reports

- Something designed for a large screen will not provide an optimal experience on a small screen
- Mobile reports provide a responsive layout that dynamically adjust the content to fit the screen on different devices and based on the rotation of the mobile device
- Development tools:
  - Mobile Report Publisher
- User accessibility:
  - Mobile Reports are focused to be delivered via the Power BI mobile application
  - They also can be published to SSRS web portal (accessed via a web browser)
- Types of visualizations include gauges (number, radial, linear), progress bars, charts (pie chart, funnel chart, category charts, waterfall), time navigator, tree-map, heat-map, map

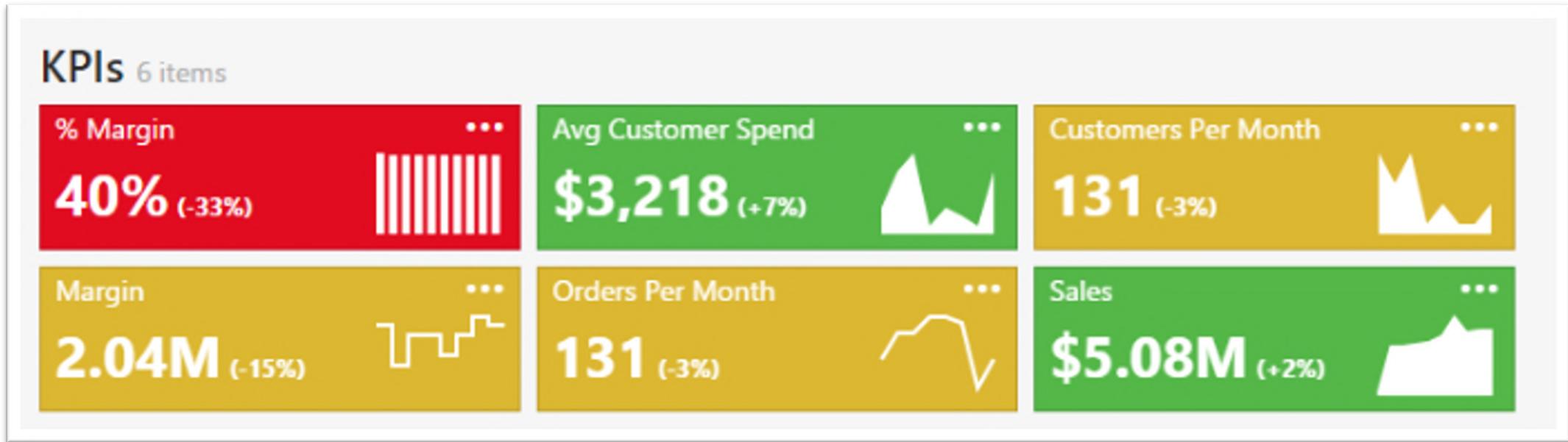
# Mobile Reports



# Key Performance Indicators

- A Key Performance Indicator (KPI) is a visual that communicates the amount of progress made toward a goal
- Key Performance Indicators are valuable for teams, managers, and businesses to evaluate quickly the progress made against measurable goals
- Development steps & tools:
  - Create a shared data source and publish to the SSRS web portal
  - Create a shared data set using SSDT or Report Builder and publish to the SSRS web portal
  - Create the KPI in the SSRS web portal
- User accessibility:
  - KPIs are created and published on SSRS web portal (accessed via a web browser)

# Key Performance Indicators



# Some SSRS Specific Concepts

- Data Source: a connection made from report to the actual data source
  - Shared Data Source: can be used by multiple reports; stored in the Report Server (accessed via the SSRS web portal)
  - Embedded Data Source: a report specific data source; stored in the report itself
- Data Set: represents actual report data retrieved using the connection specified in the Data Source
  - Shared Data Set : can be used by multiple reports; stored in the Report Server (accessed via the SSRS web portal)
  - Embedded Data Set: stored in and used by a single report

# Some SSRS Specific Concepts

- Report Parameters:
  - Use to pass values to filter data in a report
  - Can be used with paginated reports and mobile reports
  - Also used to link multiple reports
- Folder structure/hierarchy:
  - Folders in SSRS web portal provide a hierarchical navigation structure
  - Folder hierarchy and folder permissions are used control access to report server items, known as item-level security
  - By default, permissions set for a specific folder are inherited by it's child folders in the folder hierarchy
    - If a specific set of permissions are assigned to a folder, the inheritance rules no longer apply
  - Root node of the folder structure is “Home”
  - The user interface to navigate through the folder structure is the SSRS web portal

# Some SSRS Specific Concepts

- Roles and Permissions:
  - Access permissions in SSRS are attached to “Roles”
  - Based on the permissions you need to provide to a SSRS web portal users, they can be attached to one or more “Roles”
  - Each role has a set of predefined permissions
  - Predefined item-level roles:
    - Content Manger: full permission to manage content
    - Publisher: can add items to a report server
    - Browser: can execute and subscribe to reports
    - Report Builder: can create and edit reports using Report Builder
    - My Reports: can manage a personal workspace for storing items
  - Predefined system-level roles:
    - System Administrator: enable features, control overall security, manage everything
    - System Use: can view server level information

**BSc in IT: Specialising in Data Science**

IT3021: Data Warehousing  
and Business Intelligence

Lecture 10

Trending Technologies

# Content

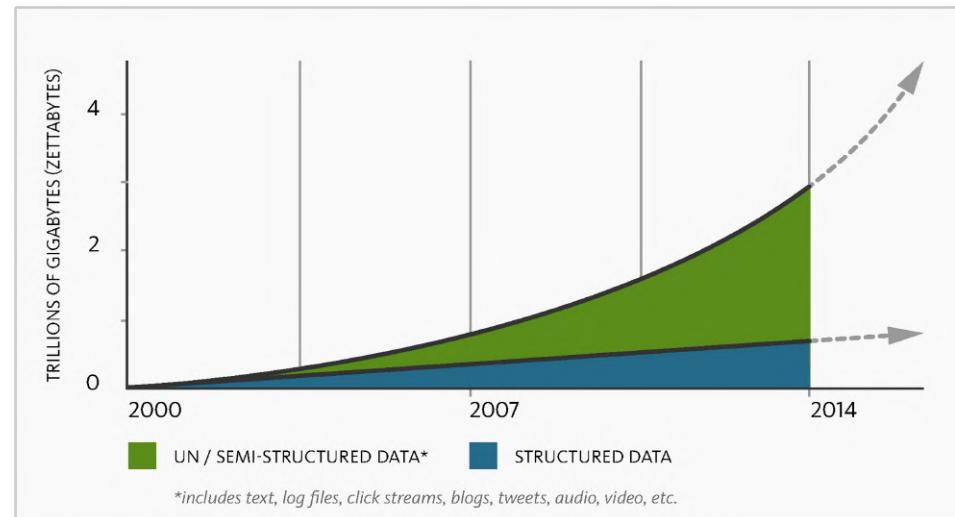
- Current trends in IT
- Challenges in current trends
- Big Data & Hadoop
- NoSQL databases
- Data lakes
- Cloud data warehouses

# Current Trends in IT

- Digital economy
  - Support large number of globally distributed and concurrent users
  - Deliver highly responsive experience and be always available
  - Handle semi and unstructured data
  - Rapidly adapt to changing requirements
- Cloud computing
  - Scaling on demand to support more customers, store more data
  - Operating applications on a global scale - customers worldwide
  - Minimizing infrastructure costs, achieving a faster time to market
- Everything on mobiles
  - Creating “offline first” apps: network connection not required
  - Synchronizing mobile data with remote databases in cloud
  - Supporting multiple mobile platforms with a single backend

# Current Trends in IT

- More users on social media
- IoT
  - Supporting many different devices with different data structures
  - Generating data from different hardware and software
  - Supporting continuous streams of real-time data
- Big Data Analytics



# Current Trends: 21<sup>st</sup> Century Technologies



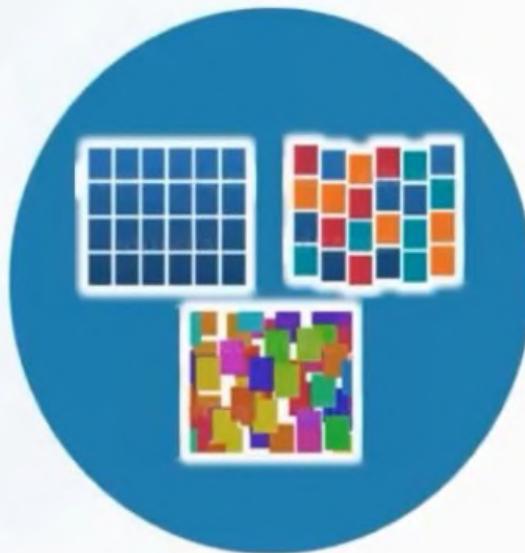
# New Requirements for New Trends

- Technical trends
    - **Volume** of data
    - **Velocity** of data
    - **Cloud** generated/hosted data
    - **Variety** of data
  - Technical requirements
    - Real **scalability**
      - Massive database distribution
      - Dynamic resource management
      - Horizontally scaling systems
    - **Frequent update** operations
    - Massive read **throughput**
    - **Flexible** database **schema**
- 
- Big Data

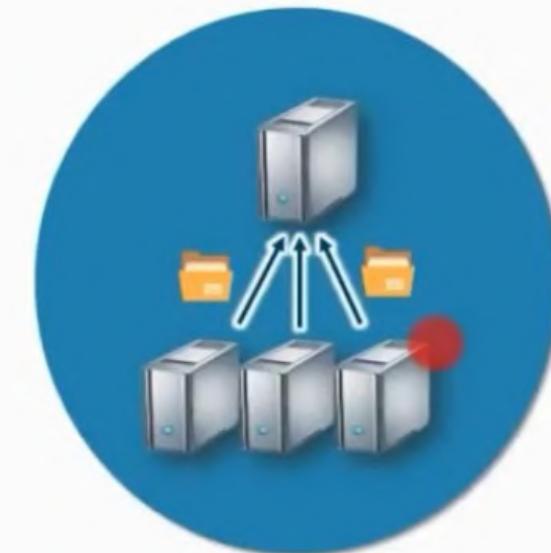
# Challenges in Addressing New Requirements



Storing huge and exponentially growing datasets



Processing data having complex structure  
(structured, un-structured, semi-  
structured)



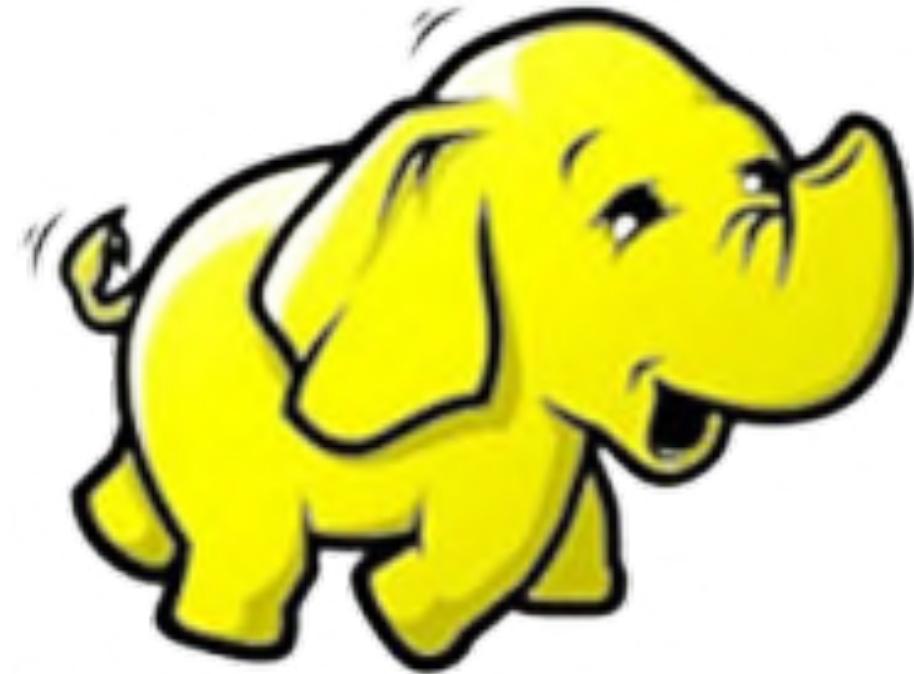
Bringing huge amount of data to  
computation unit becomes a bottleneck

# New Technologies

- Distributed file systems (GFS, HDFS, etc.)
- Distributed Processing (MapReduce, Spark, etc.)
- NoSQL databases
- Grid computing, cloud computing
- Large-scale machine learning
- etc.

# Hadoop for Big Data

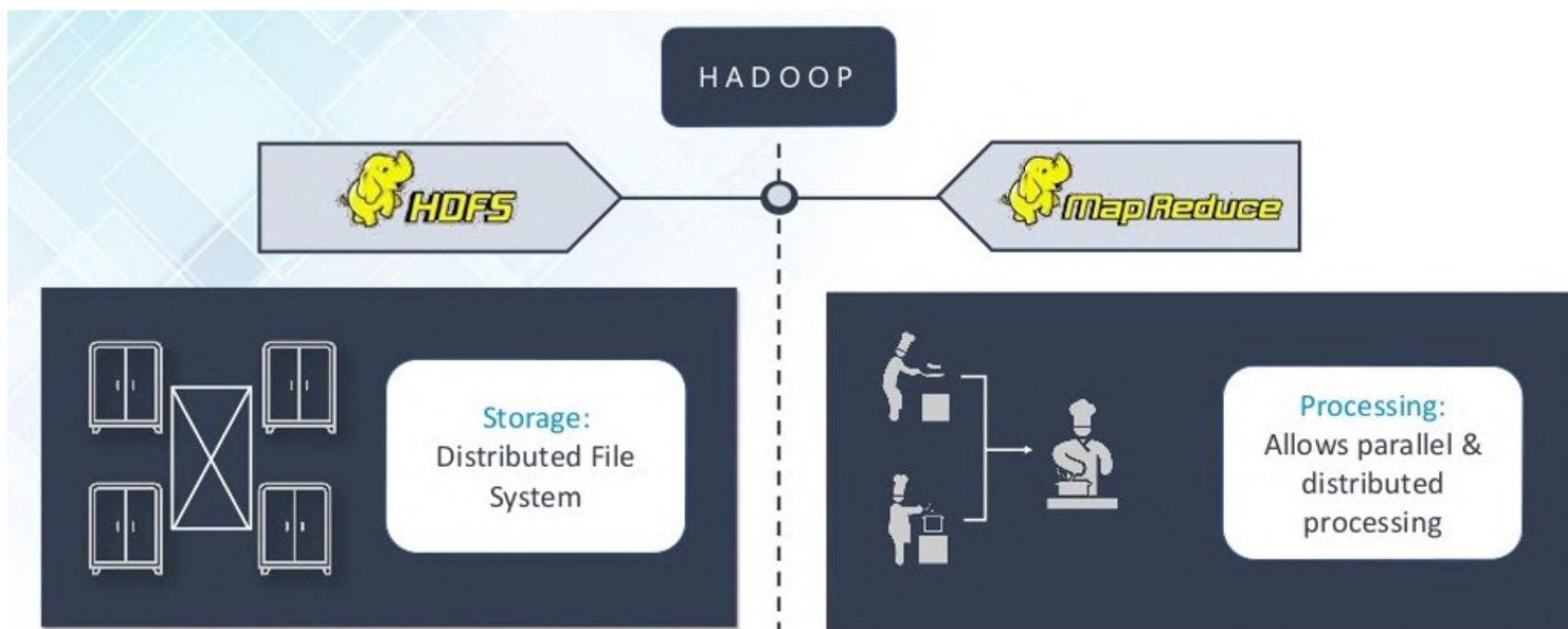
---



*hadoop*

# What is Hadoop?

- Hadoop is a **framework** that allows to **store** and **process** large data sets **in parallel and distributed fashion**

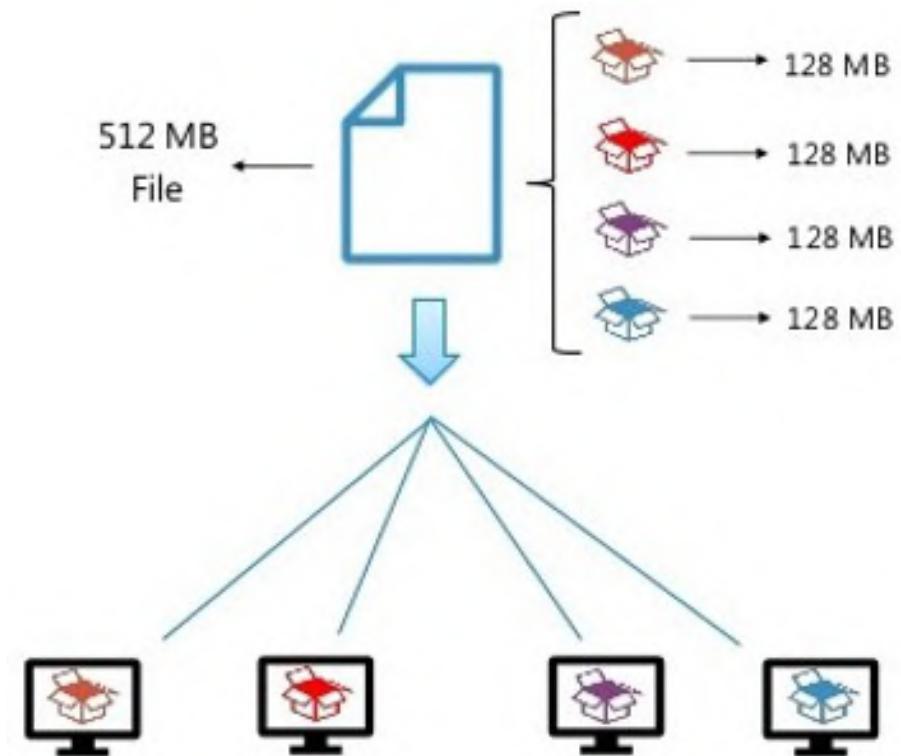


# Hadoop Solutions for Big Data Problems

## 1. Storing exponentially growing huge datasets

- Solution: HDFS

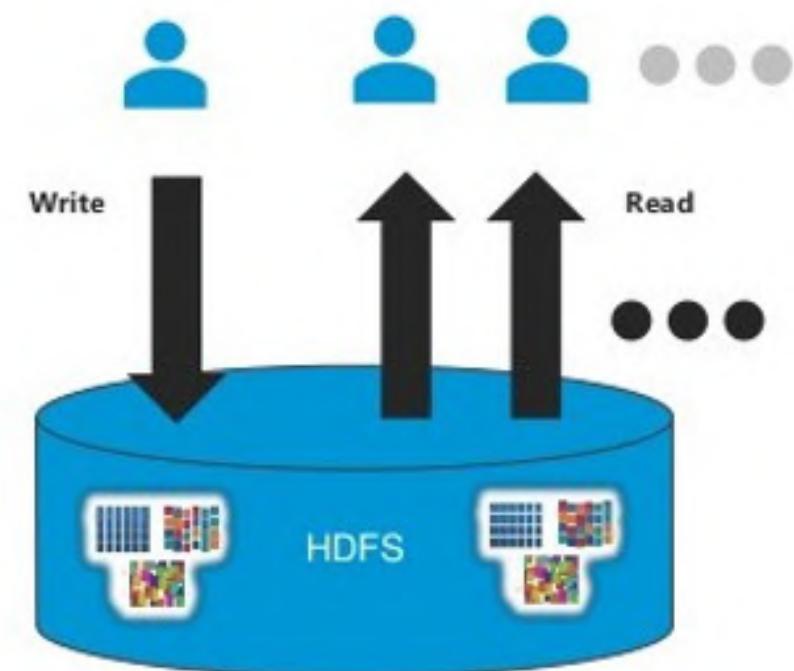
- Uses a distributed file system
- Divide files into smaller chunks
- Store them across the cluster
- Replicates chunks of data
- Scalable (horizontally)



# Hadoop Solutions for Big Data Problems

## 2. Processing data having complex, various structures

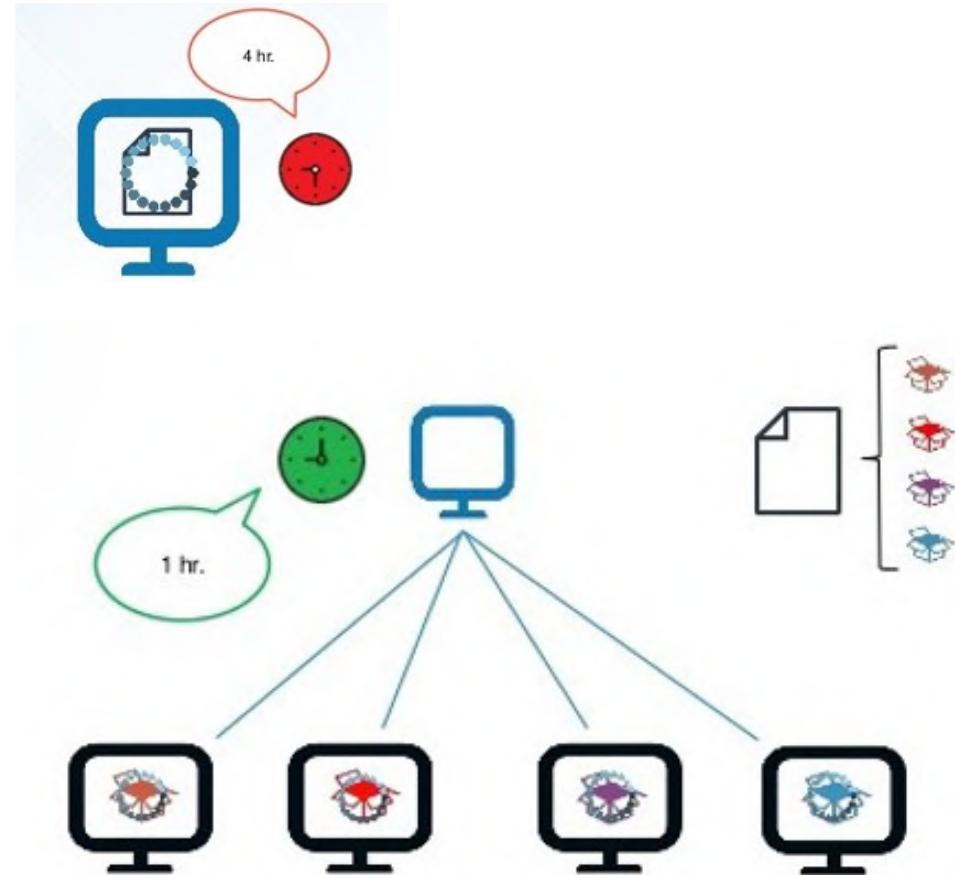
- Solution: HDFS
  - Allows storing any kind of data
  - No schema validation while dumping
  - Follows WORM (Write once Read Many)



# Hadoop Solutions for Big Data Problems

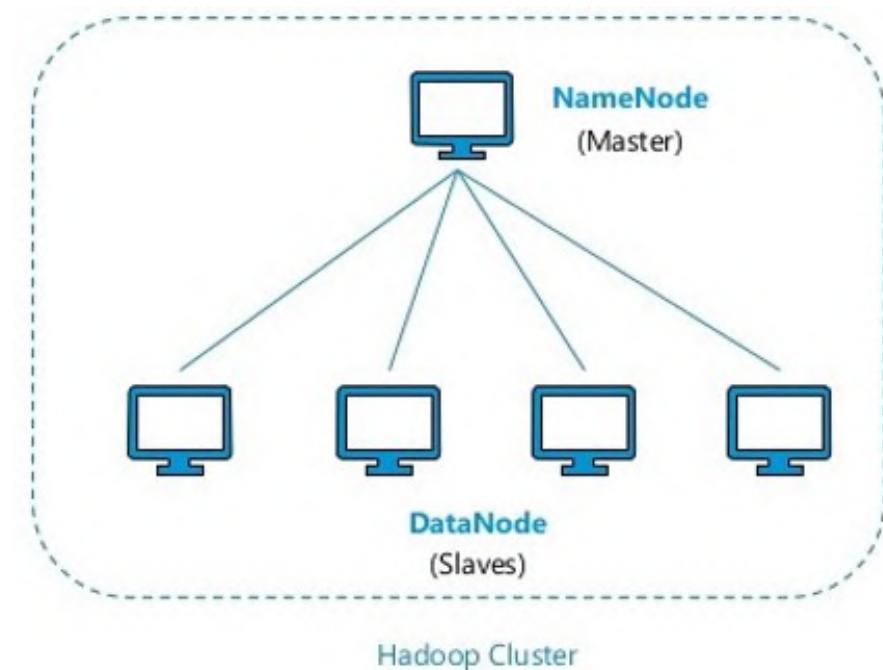
## 3. Bringing Data to Processing

- Solution: MapReduce
  - Parallel processing of data
  - Process data locally
  - Move processing to data
  - Each node processes data which is stored in



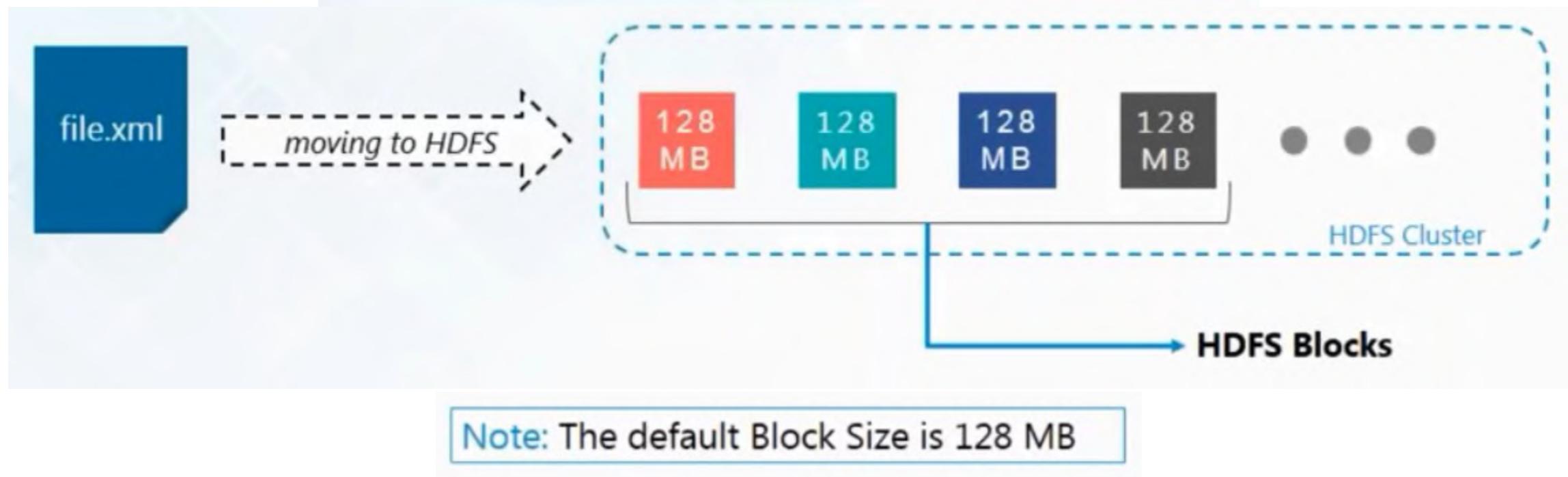
# HDFS

- HDFS creates a level of abstraction over the resources, from where the whole HDFS file system is seen as a single unit
- HDFS components:
  - **NameNode**: main node that contains meta data about the data stored
  - **DataNodes**: commodity hardware used to stored data, in the distributed architecture
  - **Secondary NameNode**



# HDFS Block

- HDFS stores the data in form of blocks
- Block size can be configured base on requirements

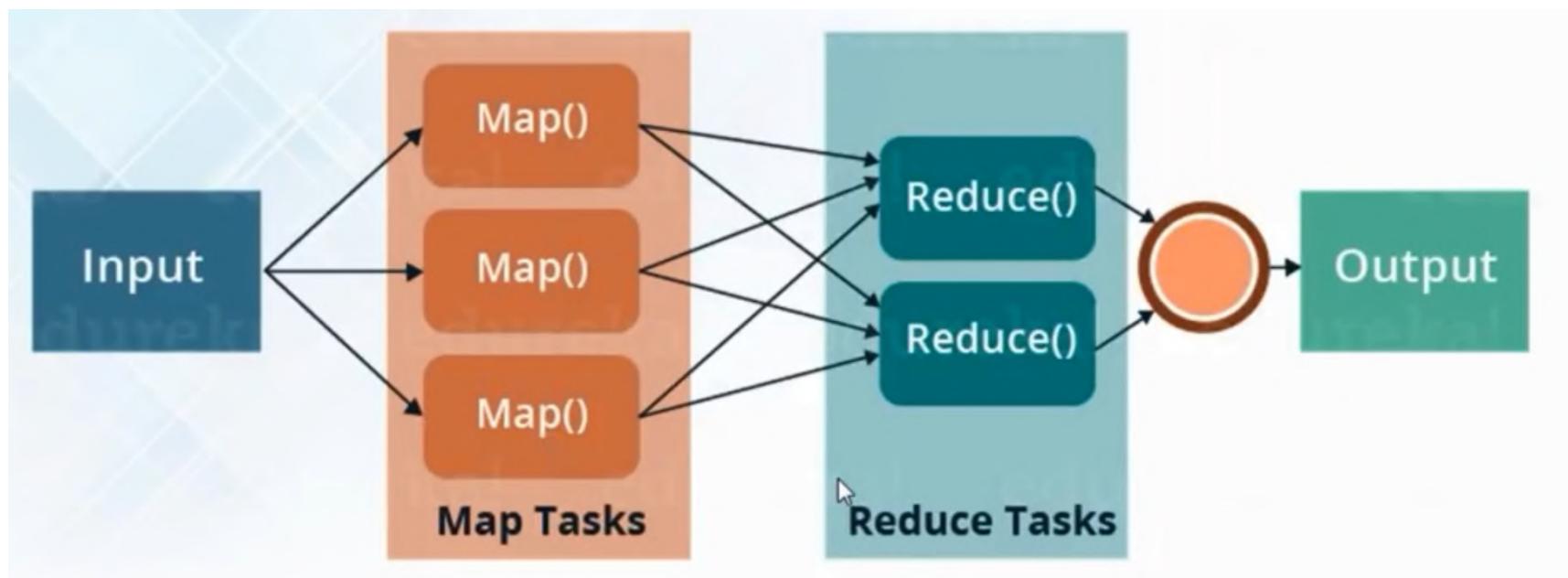


# Other HDFS Concepts

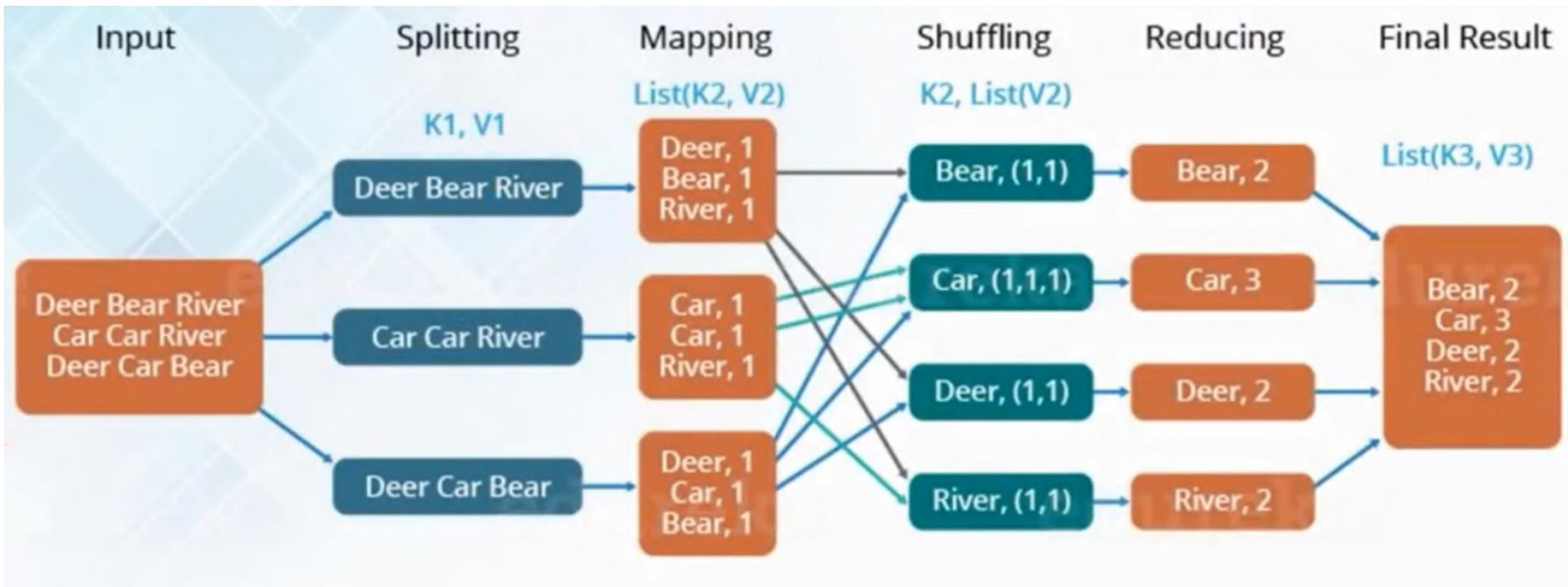
- HDFS replication
- Rack awareness
- Data write mechanism
- Data read mechanism
- Cluster fault tolerance

# Map Reduce

- MapReduce is the **programming framework** that allows to perform **distributed and parallel processing** on large data sets in distributed environment

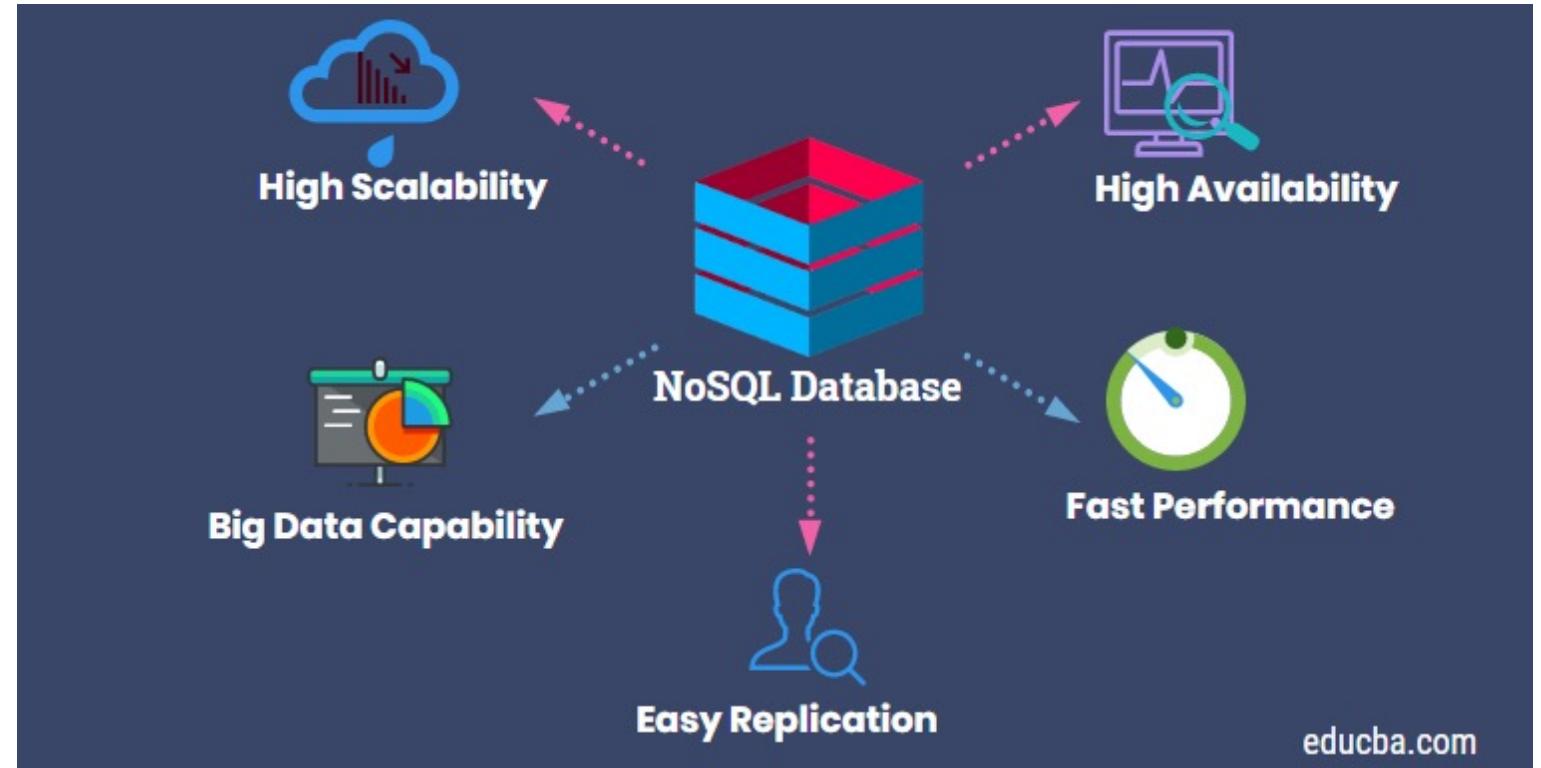


# MapReduce: Example



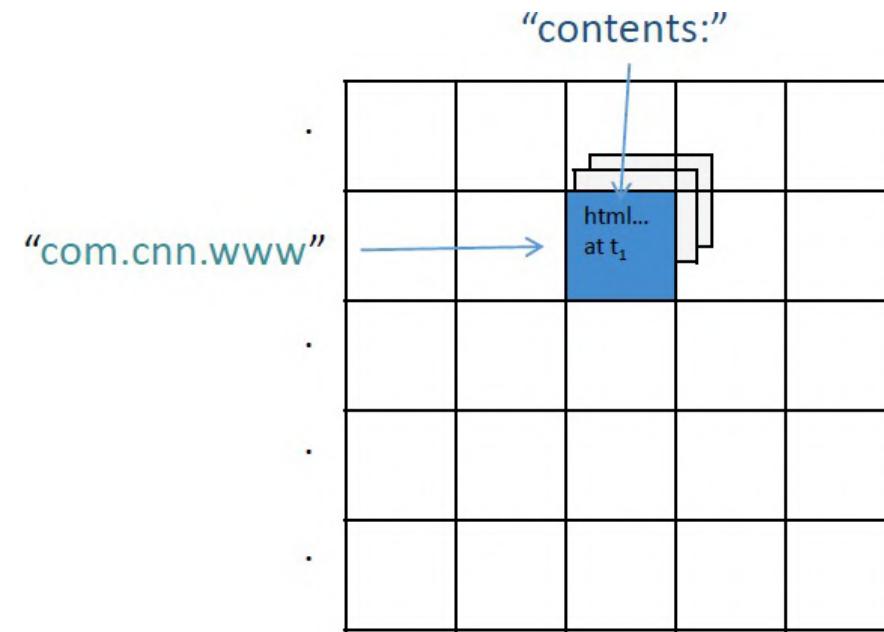
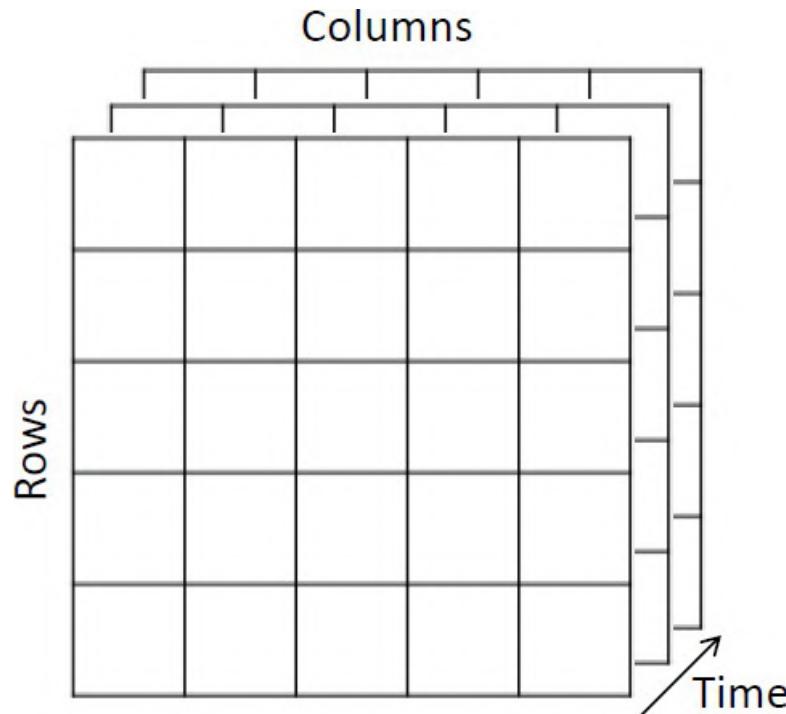
# NoSQL Databases

---



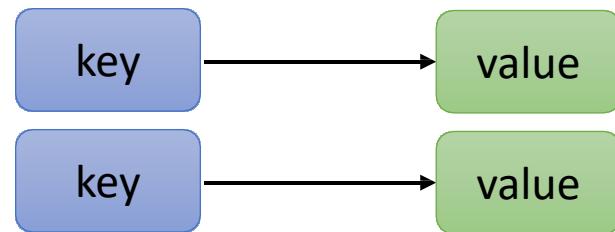
# Google BigTable (2006)

- Data model: three-dimensional indexed sorted map
  - Input(row, column, timestamp) -> output(cell contents)



# Amazon: Dynamo DB (2007)

- Data model:
  - Simple hash table (map): key-value data store



# Common Characteristics of NoSQL Databases

- Not using relational model (nor the SQL language)
- Designed to run on large clusters (horizontally scalable)
- Schema-less: fields can be freely added to any record
- Open source
- Other characteristics (often true):
  - Easy replication support (fault-tolerance, query efficiency)
  - Simple API
  - Eventually consistent (not ACID)

# RDBMS vs. NoSQL: When to Use What?

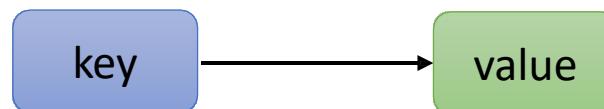
	RDBMS	NoSQL
<b>Integrity</b>	is mission-critical	OK as long as most data is correct
<b>Data format</b>	consistent, well-defined	unknown or inconsistent
<b>Data life-time</b>	is of long-term value	is expected to be replaced
<b>Growth</b>	predictable, linear growth	unpredictable growth (exponential?)
<b>Querying</b>	non-programmers writing queries	only programmers writing queries
<b>Fault tolerance</b>	regular backups	automatic data replication
<b>Distribution</b>	access through master server	data sharding (partitioning)

# Types of NoSQL Databases

- Key-Value Stores
- Document Databases
- Column-family Stores
- Graph Databases

# Key-Value Stores

- A simple hash table (map), primarily used when all accesses to the database are via primary key
  - key-value mapping
- In RDBMS world: A table with two columns:
  - ID column (key)
  - DATA column storing the value (unstructured BLOB)
- Basic operations
  - Get the value for the key
  - Put a value for a key
  - Delete a key-value



# Key-Value Stores: Representatives



LevelDB

ORACLE  
NOSQL DATABASE



ORACLE  
BERKELEY DB 12<sup>c</sup>

Infinispan



# Document Databases

- Also key-value pairs
- But value is a semi-structured text data – document
- Documents are self-describing pieces of data
- Hierarchical tree data structures
  - Nested associative arrays (maps), collections, scalars
  - XML, JSON, etc.
- Can query inside document
  - Building search indexes on various keys/fields

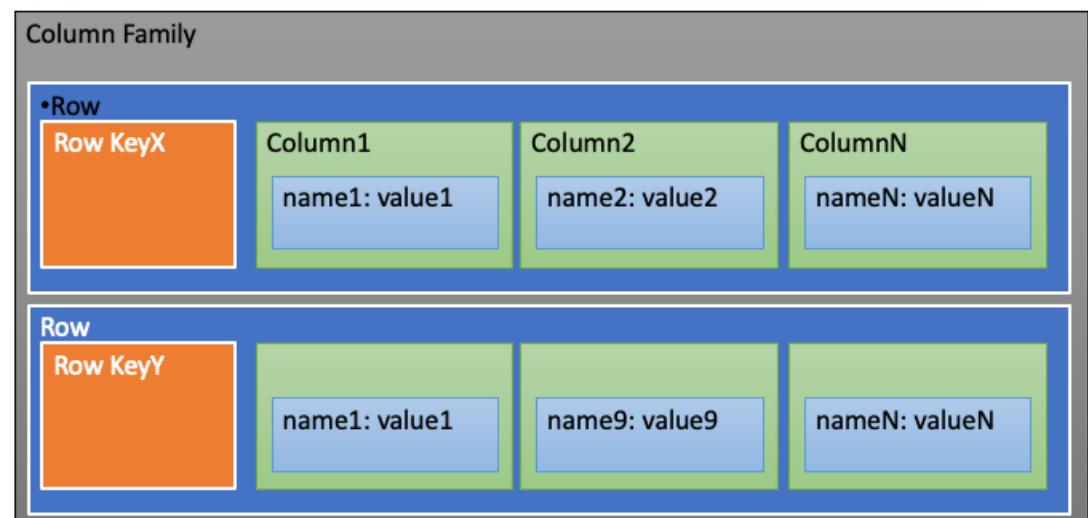
```
<key=CustomerID>
{
  "customerid": 3
  "customer":
  {
    "firstname": "Kamal",
    "lastname": "Perera",
    "likes": ["cricket", "singing"]
  }
  "address":
  {"province": "Western",
   "city": "Colombo",
   "no": 111
  }
}
```

# Document Databases: Representatives



# Column-Family Stores

- Data model: rows that have many columns associated with a row key
- Data is physically stored by column families
- Column families are groups of related data (columns) that are often accessed together
  - e.g., for a customer, we typically access all profile information at the same time, but not customer's orders

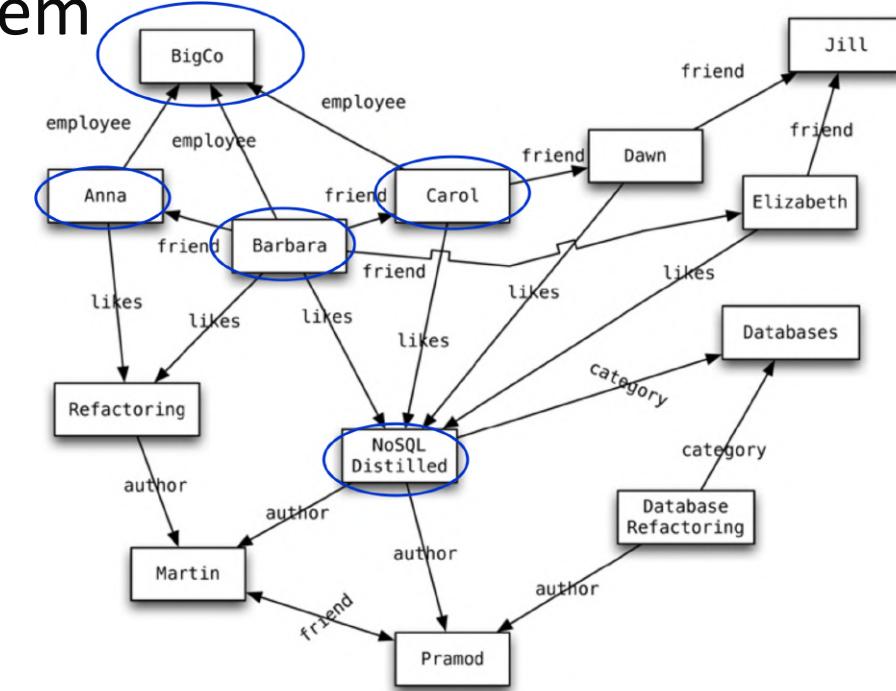


# Column-Family Stores: Representatives



# Graph Databases

- To store entities and relationships between them
  - Nodes are instances of objects
  - Nodes have properties; e.g., name
  - Edges have directional significance
  - Edges have types; e.g., likes, friend
- Nodes are organized by relationships
- Allow to find interesting patterns
  - For example, get all nodes that are “employee” of “Big Company” and that “likes” “NoSQL Distilled”



# Graph Databases: Representatives



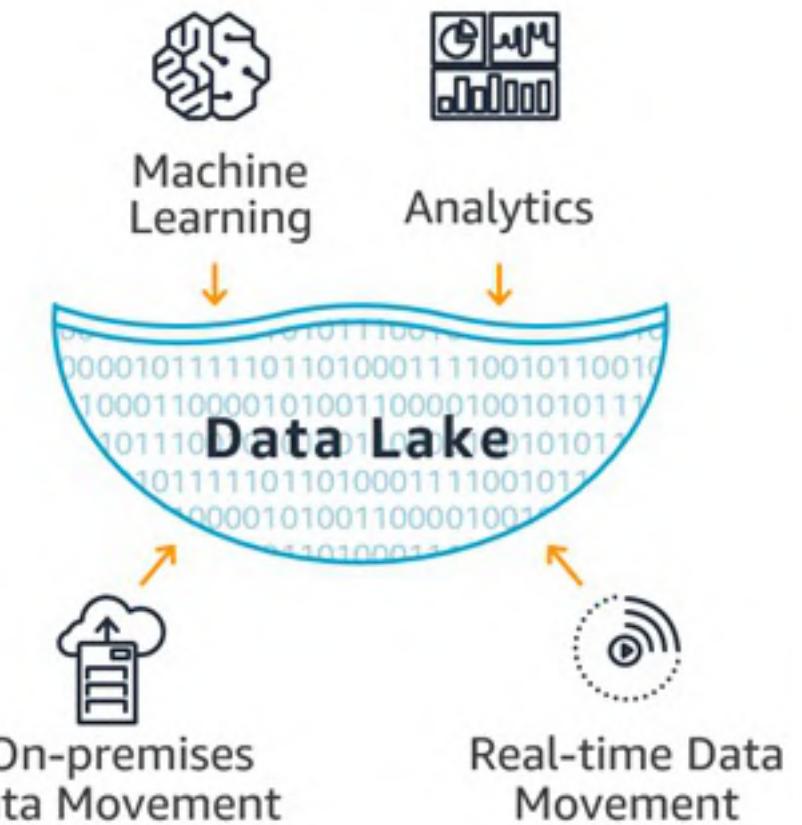
# Data Lakes

---



# What is a Data Lake?

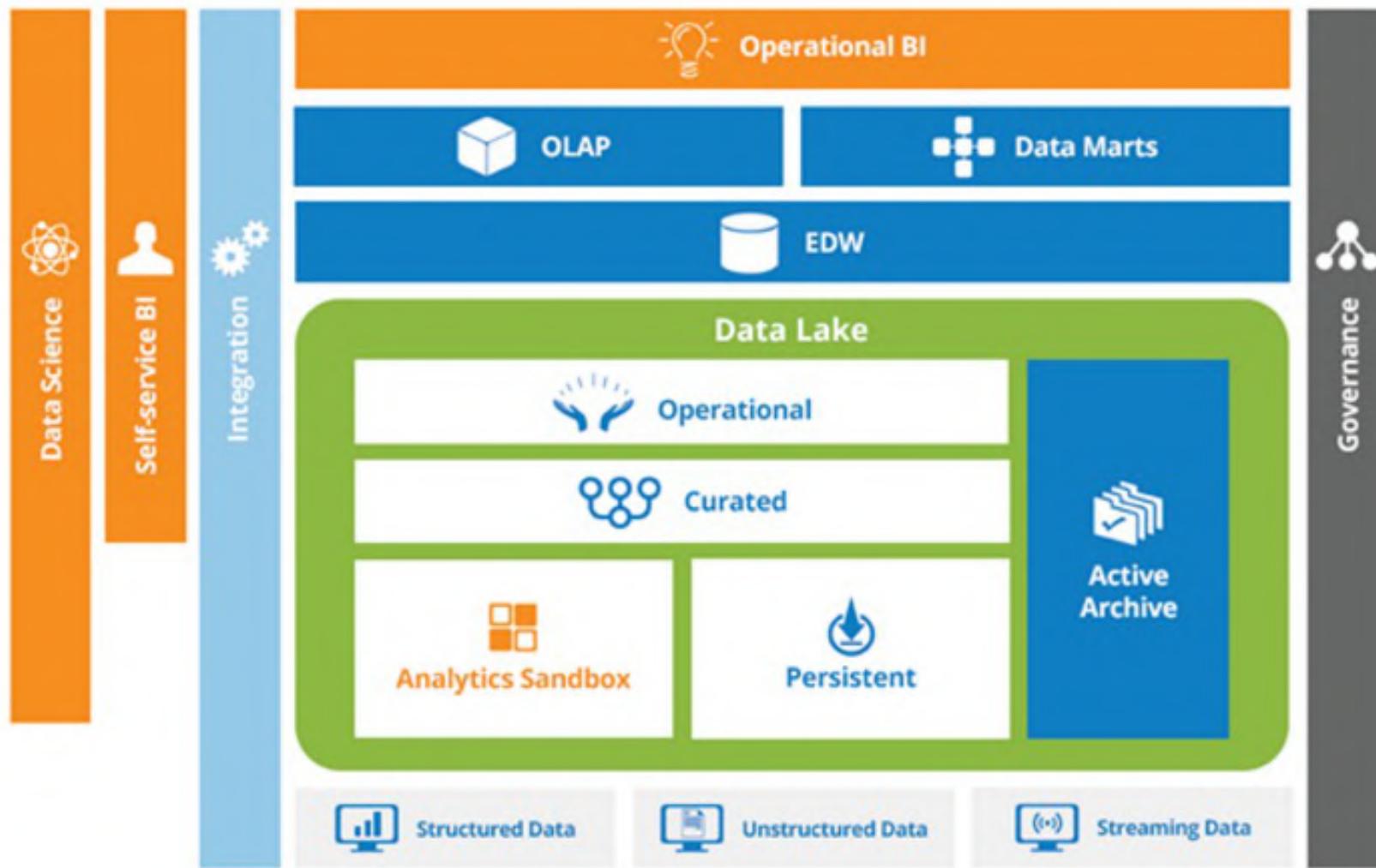
- A centralized repository that allows you to store structured, semi-structured and unstructured data as-is without having to structure the data, at any scale
- Allows you to run different types of analytics; from dashboards and visualizations to big data processing, real-time analytics, and machine learning



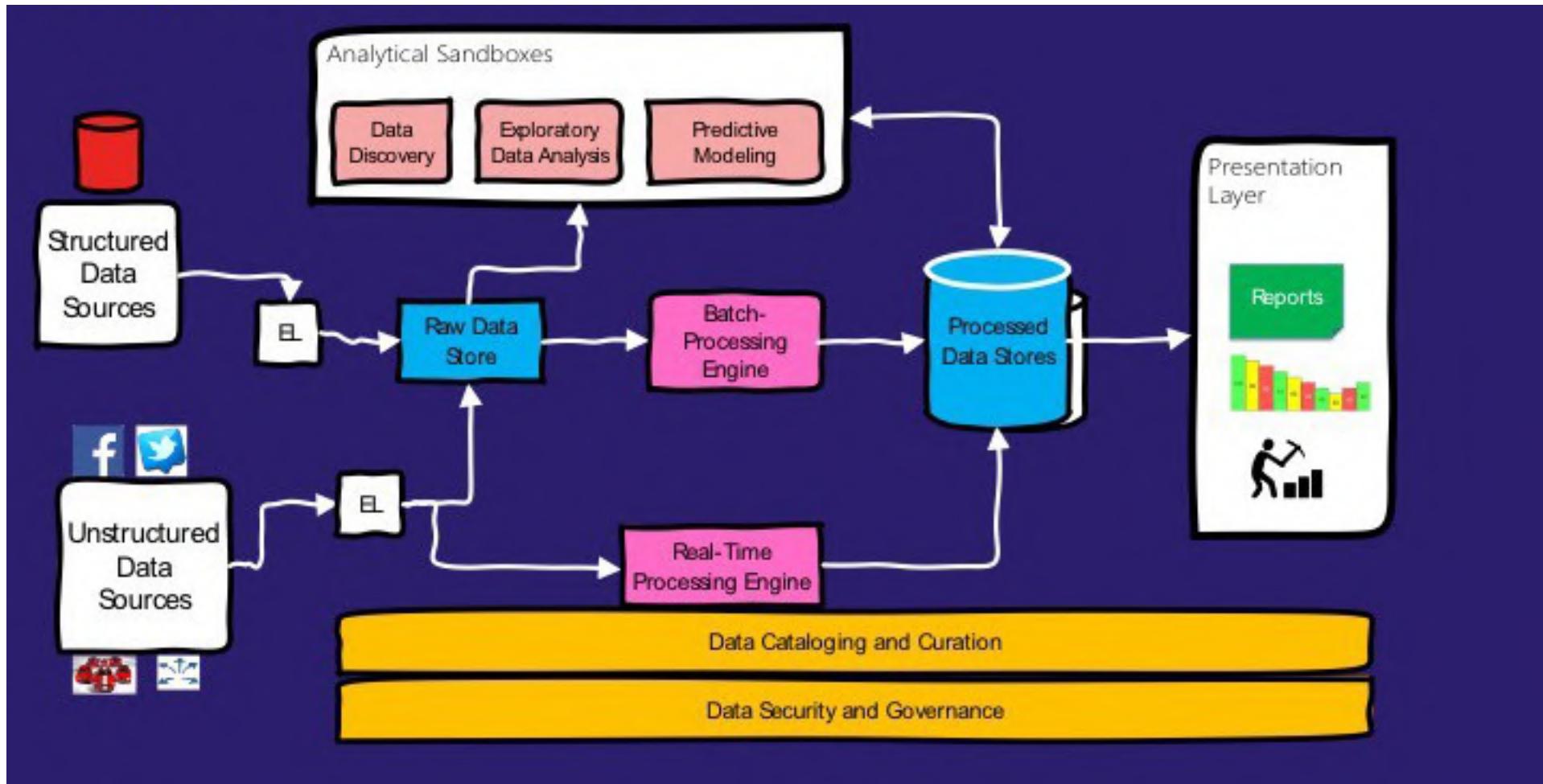
# Why do we need Data Lakes?

- Cater to large volumes of data: companies have large volumes of data that it is either costly and/or sometimes RDBMS/DW can not handle
- Seamlessly integrate diverse data source and formats: rigid structures/architectures of traditional systems cannot cater to different, changing data formats
- Process data that are being generated at fast rate: traditional tools are unable to process data at the rate of the they are being generated
- Enterprise scale analytical/BI requirements: analytical tools can consume any available data within and outside of the organization and make use of self-service analytical features
- A single data store to cater all types of data requirements

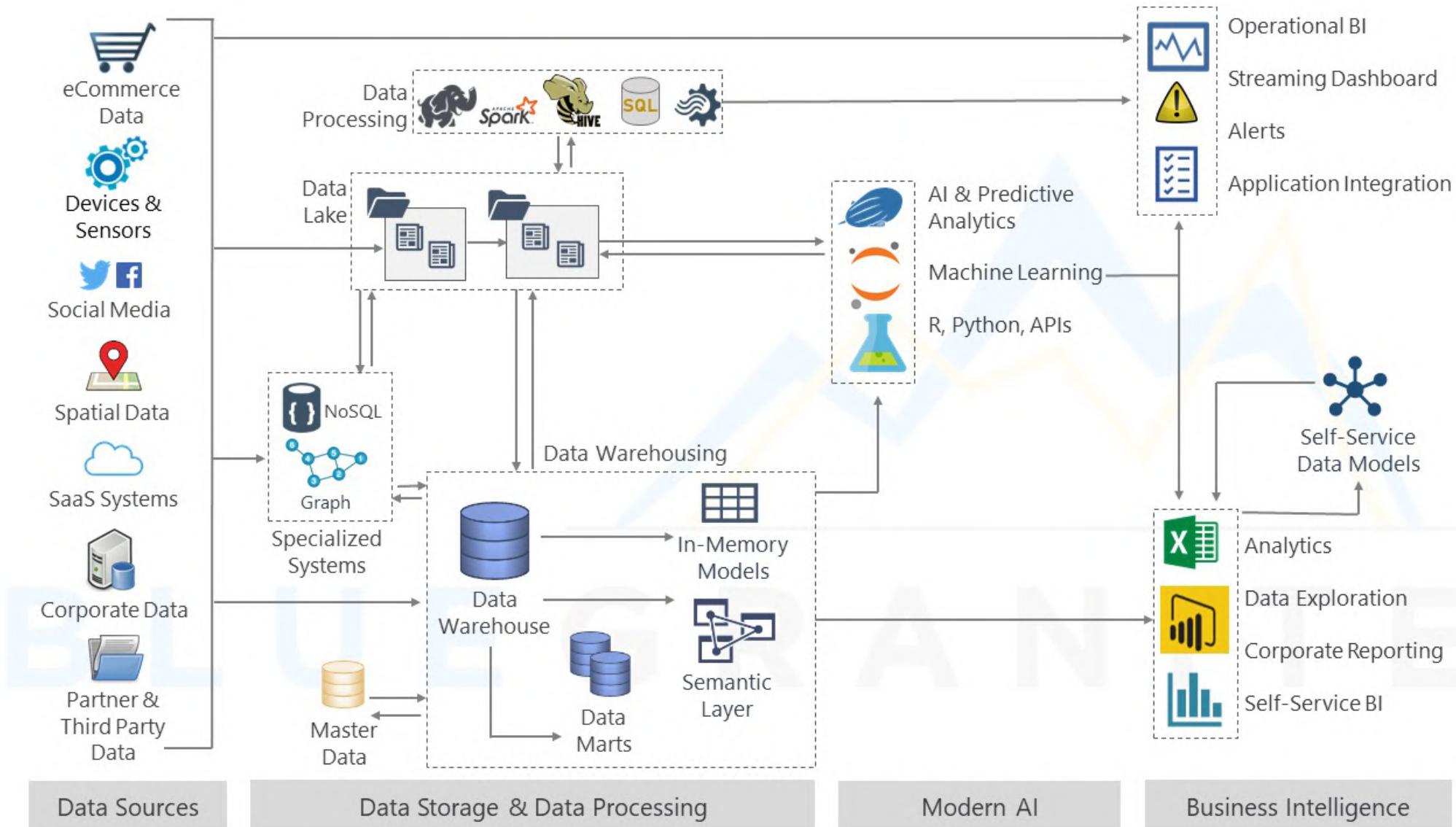
# Data Lake Architecture



# Data Lake Architecture



# Data Lake Architecture



# Challenges of Data Lakes

- Data reliability: can insert almost anything
- Implementation of data validation methods and quality enforcement
- Reprocessing data due to broken pipelines: updates are not possible
- Combining batch and streaming data: lambda architecture follows different paths
- Bulk updates, merges, and deletes
- Query performance:
  - Due to large volumes
  - Unstructured data/depends on the way data is structured
  - Small files
- Metadata management

# Traditional DW vs. Data Lakes

Parameters	Data Lakes	Data Warehouse
Data	Data lakes store everything	Data warehouse focuses on mainly business processes related data
Processing	Data are mainly unprocessed	Highly processed data
Type of Data	It can be unstructured, semi-structured and structured	It is mostly in tabular form & structure
End-users	Data lake is mostly used by data scientists/engineers/IT professionals	Business professionals widely use data warehouse
Storage	Data lakes design for low-cost storage	Expensive storage that give fast response times are used
Security	Offers lesser control	Allows better control of the data
Schema	Schema on reading (no predefined schemas)	Schema on write (predefined schemas)
Data Processing	Helps for fast ingestion of new data	Time-consuming to introduce new content
Tools	Can use open-source tools like Hadoop/Map Reduce	Mostly commercial tools

# Tools & Frameworks



- HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets
- Amazon Simple Storage Service (S3) is a cloud object storage service for structured and unstructured data to build a data lake
- Azure data lake storage is a massively scalable and secure data lake service over cloud for high-performance analytics workloads

# Tools & Frameworks



- Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data
- Apache Spark is an open source big data processing engine, enabling SQL queries and rapid distributed processing of the data
- Apache Spark Streaming is a scalable fault-tolerant streaming processing system that natively supports both batch and streaming workloads

# Tools & Frameworks

cloudera



databricks

- Hadoop platform that integrate the most popular Apache Hadoop open source software within one place
- HDInsight is a fully-managed cloud Hadoop offering that provides optimized open source analytic clusters for Spark, Hive, MapReduce, HBase, Storm, Kafka, and R Server backed by a 99.9% SLA
- A comprehensive data lake platform for data engineering, data science, machine learning and analytics

# Cloud Data Warehouses



**amazon  
REDSHIFT**

- Scalable, cluster of nodes optimized for data warehouses



- An elastic, large-scale data warehouse platform-as-a-service that leverages the broad ecosystem of SQL Server and uses distributed MPP architecture



BigQuery

- Serverless, highly scalable, and cost-effective multi-cloud data warehouse designed for business agility



- A data warehouse-as-a-service, and operates across multiple clouds, including AWS, Microsoft Azure, and Google Cloud. It separates storage, compute, and services into separate layers, allowing them to scale independently