

The Age of Data



Outline

- What is "Big Data" Reminder
- Big Data Foundational Technologies
- The structure of data projects
- Data visualization



“BIG DATA” IS DATA WHOSE SCALE, DIVERSITY, AND COMPLEXITY REQUIRE NEW ARCHITECTURE, TECHNIQUES, ALGORITHMS, AND ANALYTICS TO MANAGE IT AND EXTRACT VALUE AND HIDDEN KNOWLEDGE FROM IT...

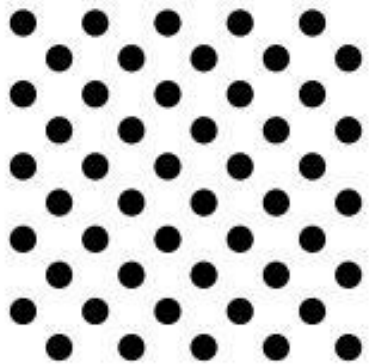
THE BIG DEAL ABOUT BIG DATA

Goldman
Sachs

Asset

The Challenges in Big Data: The four V's

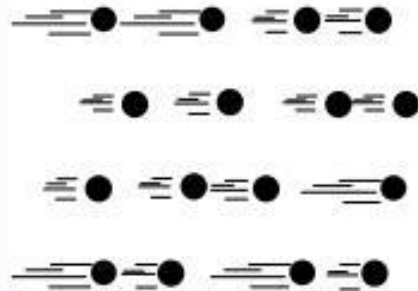
Volume



Data at Rest

Terabytes to exabytes of existing data to process

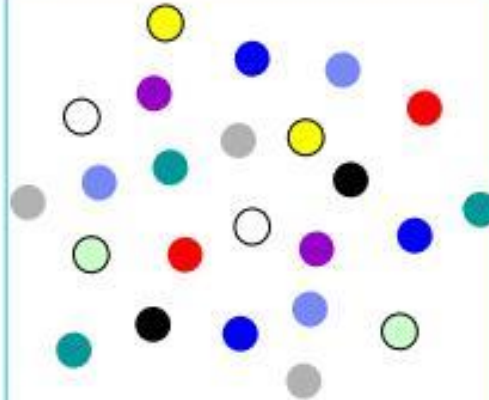
Velocity



Data in Motion

Streaming data, milliseconds to seconds to respond

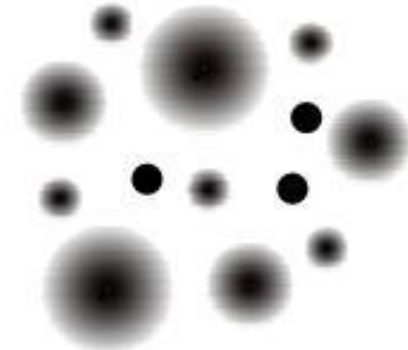
Variety



Data in Many Forms

Structured, unstructured, text, multimedia

Veracity*



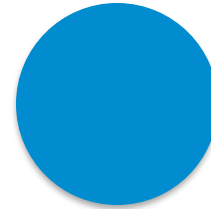
Data in Doubt

Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations

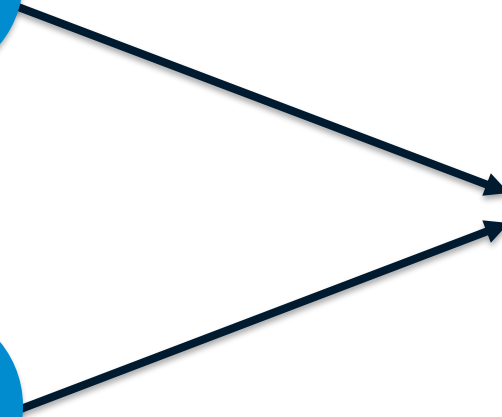
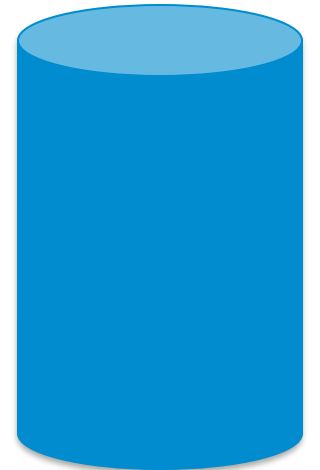
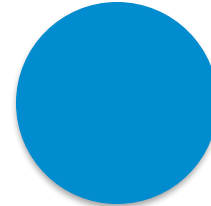
Sources of Data

- Paper
- Spreadsheet
- Proprietary format
- Open format
- Available through API
- Available export from application
- Data feeds
- Communications
- Other

INTERNAL DATA



EXTERNAL DATA



Data Lake

- Repository for analyzing large quantities of disparate sources of data in its native or raw format
- Reduce up-front effort by ingesting data in any format without requiring a schema initially
- Make acquiring new data easy, so it can be available for data science & analysis quickly
- Store large volume of multi-structured data in its native format
- <https://www.mongodb.com/databases/data-lake-vs-data-warehouse-vs-database>



Data Lake

- Typically, the primary purpose of a data lake is to analyze the data to gain insights.
- Organizations sometimes use data lakes simply for their cheap storage with the idea that the data may be used for analytics in the future.
- <https://www.mongodb.com/databases/data-lake-vs-data-warehouse-vs-database>
- <https://www.mongodb.com/blog/post/unlocking-operational-intelligence-from-the-data-lake-part-1-the-rise-of-the-data-lake>
- <https://www.youtube.com/watch?v=LxCH6z8TFpl>



Data Warehousing

- Data is integrated from multiple systems.
- For example provide a full view of a customer:
 - Sales activity
 - Delinquent invoices
 - Support/help requests
- Focus is on reading the information and creating analysis
- Data modelling and ETL process consume most of the time and effort in setting up a data warehouse



Data Lake

- Agility
- Flexibility
- Rapid Delivery
- Easy exploration
- Data acquisition is easier
- Data retrieval requires more effort

Data Warehouse

- Governance
- Reliability
- Standardization
- Security
- Data acquisition requires more effort
- Data retrieval is easier

Database

- A database is a collection of data or information.
- Databases are typically accessed electronically and are used to support Online Transaction Processing (OLTP).
- Database Management Systems (DBMS) store data in the database and enable users and applications to interact with the data.
- The term “database” is commonly used to reference both the database itself as well as the DBMS



OLTP

- We often speak about two types of data processing OLTP Vs OLAP
- Online Transaction Processing (OLTP) enables the real-time execution of large numbers of database transactions by large numbers of people, typically over the Internet.
- Process a large number of relatively simple transactions — usually insertions, updates and deletions to data.
- Enable multi-user access to the same data, while ensuring data integrity.
- Support very rapid processing, with response times measured in milliseconds.
- Provide indexed data sets for rapid searching, retrieval and querying.
- Be available 24/7/365, with constant incremental backups.

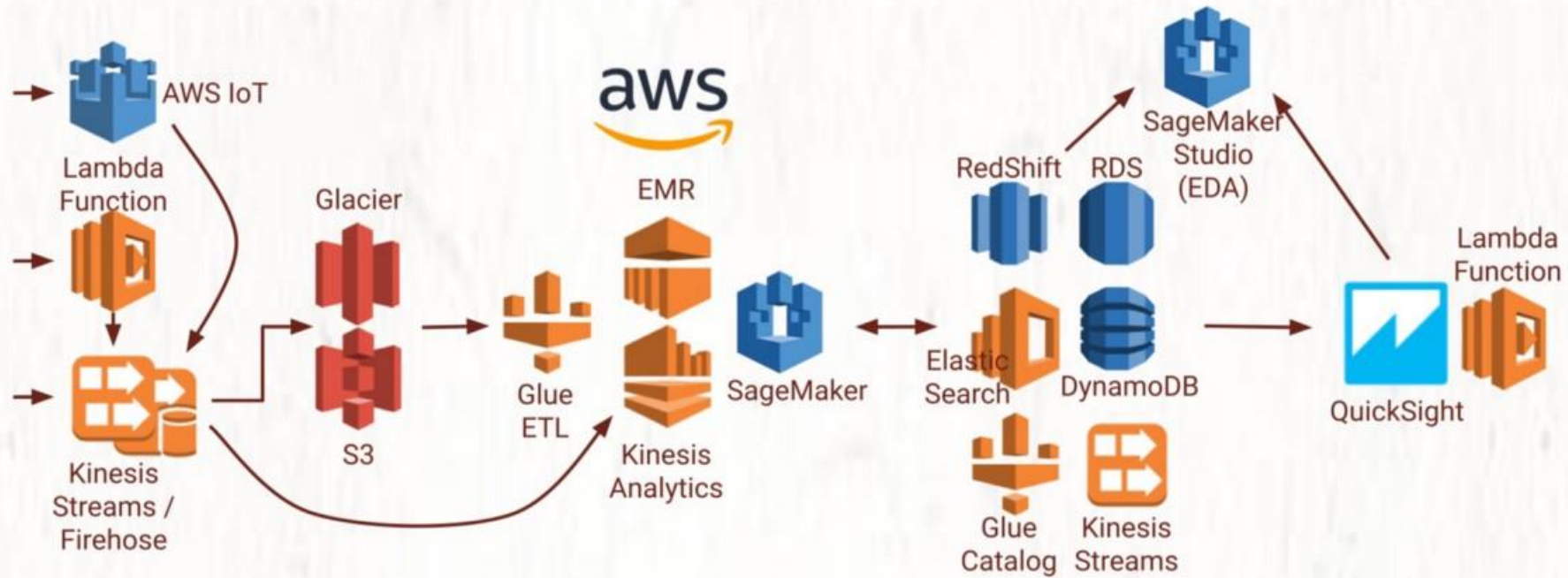
OLAP

- Online analytical processing (OLAP) is the processing of data for analysis and analytics purposes.
- OLAP systems allow data to be extracted for complex analysis. To drive business decisions, the queries often involve large numbers of records.
- In OLAP, response times are orders of magnitude slower than OLTP. Workloads are read-intensive, involving enormous data sets.
- Since they don't modify current data, OLAP systems can be backed up less frequently. However, OLTP systems modify data frequently, since this is the nature of transactional processing. They require frequent or concurrent backups to help maintain data integrity.

A Cloud-Based Data Pipeline Example

Big Data Pipelines on AWS, Azure, and Google Cloud

scgupta.link/big-data-pipeline



Ingestion

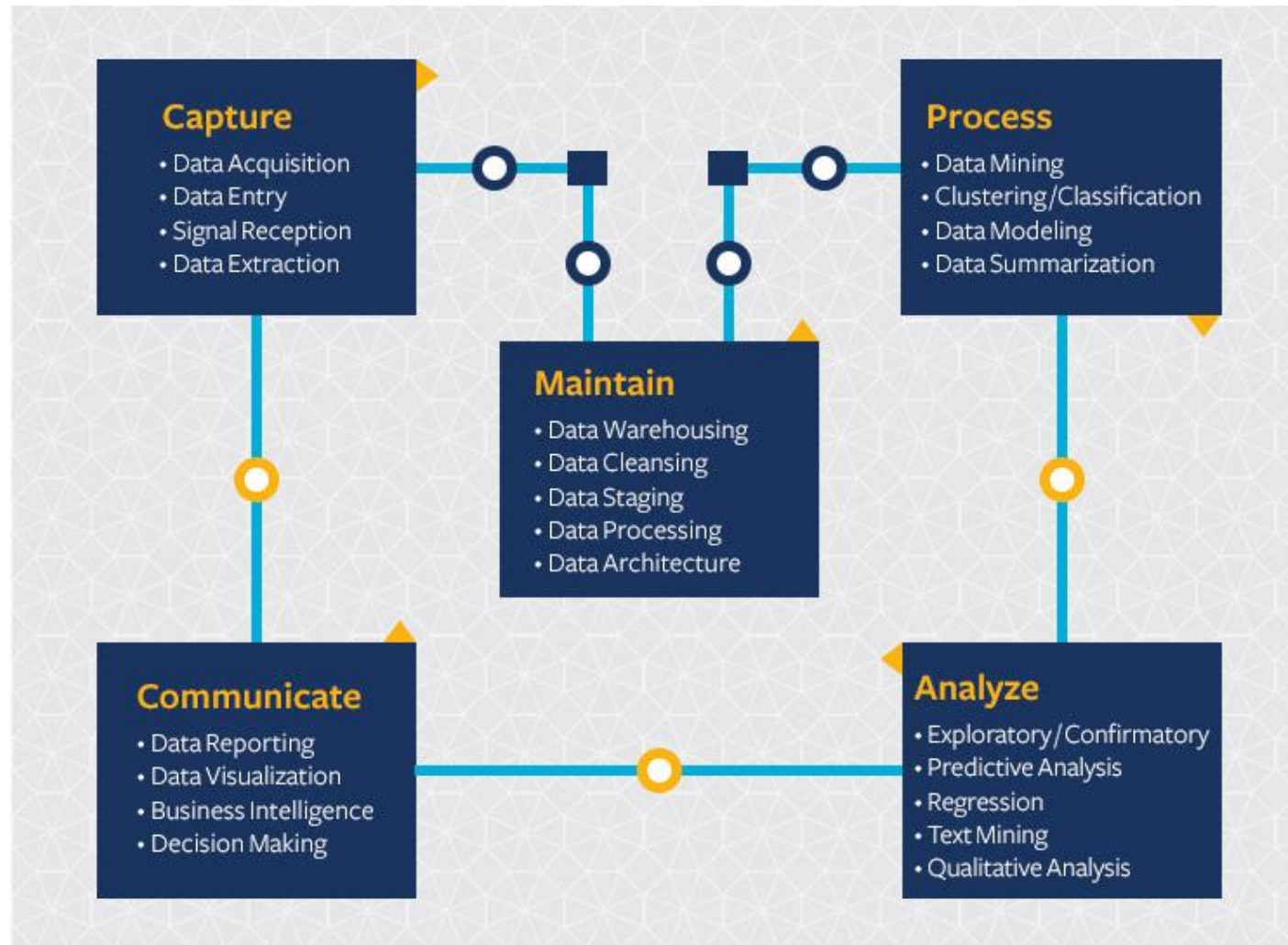
Data Lake

Preparation & Computation

Data Warehouse

Presentation

A Data Science View of the Process



ETL – Extract, Transform, Load

- (**ETL**) is the general procedure of copying data from one or more sources into a destination system which represents the data differently from the source(s) or in a different context than the source(s)
- Traditionally, ETL has been used to move data between elements in a data pipeline
 - Online Transaction Processing Database => Data Lake => Data Warehouse
- More recently, unified data platforms provide a "one-stop-shop" thus reducing the need for ETL

The Evolution of Databases

- Relational Databases & SQL
- The rise of NoSQL
- Popular NoSQL Databases

Relational Databases - SQL



A **relational database** is a digital database whose organization is based on the relational model of data, as proposed by E. F. Codd in 1970.

The **relational model** organizes data into one or more tables of columns and rows, with a unique key identifying each row.

Relationships are a logical connection between different tables, established on the basis of interaction among these tables.

Virtually all relational database systems use **SQL** as the language for querying and maintaining the database.

Relational Databases - SQL

- Relational Database Management System (RDBMS)
- Developed at IBM (early 1970s)
- First commercial version was by Relational Software (now Oracle) in 1979.
- SQL is an ISO standard, but most vendors add their own extensions.
- Popular implementations of SQL include
 - Oracle Database
 - MySQL – a free and open-source implementation now owned by Oracle
 - MS SQL Server
 - PostgreSQL

Relational Databases - Strengths

- Extremely well proven and widely used in the industry
 - E.g. Oracle, SQL Server, MySQL
- Quality of service guarantees
 - Highly efficient, e.g. via indexes, load balancing, etc.
 - Highly available, e.g. via replication, fail-over, etc.
 - Highly secure
 - Transactional

Relational Databases – Limitations

- Not good at storing unstructured or heterogeneous data
 - This kind of data doesn't fit nicely into the structured world of rectangular tables and fixed relationships
- Not ideal for ingress of data at high velocity
 - It takes time to break the data down into rectangular chunks, so that it can be inserted into table(s) in an RDBMS
- Not good for rapidly evolving (agile) requirements
 - You can't keep changing the database schema all the time!
- Not ideal for scale-out architectures
- RDBMS aren't really designed for the cloud / commodity storage

The Rise of NoSQL

- The term "NoSQL" gained popularity around 2009
- NoSQL is a general term to represent non-relational database management systems
 - Encompasses a wide variety of database technologies
- NoSQL databases are designed to address the demands of building applications dealing with “Big Data”
 - Unstructured data
 - Handling big data
 - Data modelling agility
 - Scale-out architecture via auto-sharding, i.e. natively and automatically spread data across any number of servers

The Rise of NoSQL

● MongoDB
Topic

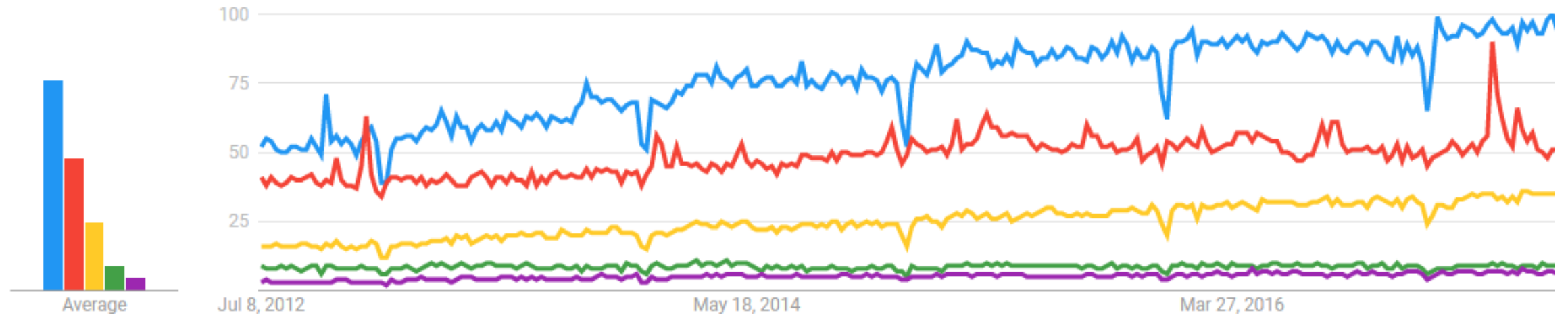
● Apache Cassandra
Topic

● Redis
Software

● Apache HBase
Software

● Neo4j
Search term

Interest over time ?



CAP Theorem

Consistency

Consistency means that all clients see the same data at the same time, no matter which node they connect to. For this to happen, whenever data is written to one node, it must be instantly forwarded or replicated to all the other nodes in the system before the write is deemed 'successful.'

Availability

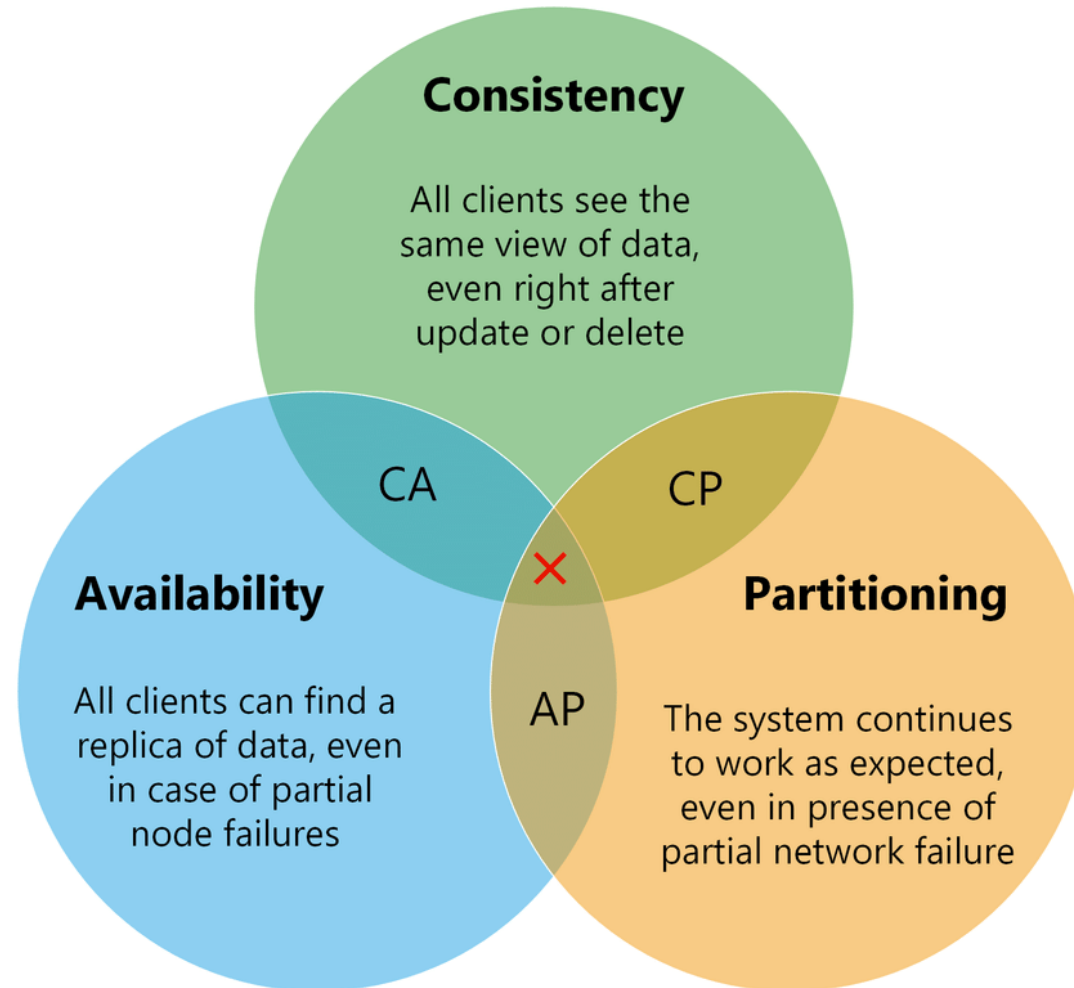
Availability means that any client making a request for data gets a response, even if one or more nodes are down. Another way to state this—all working nodes in the distributed system return a valid response for any request, without exception.

Partition tolerance

A *partition* is a communications break within a distributed system—a lost or temporarily delayed connection between two nodes. Partition tolerance means that the cluster must continue to work despite any number of communication breakdowns between nodes in the system.

<https://www.ibm.com/cloud/learn/cap-theorem>

CAP Theorem



Relational Data Modelling

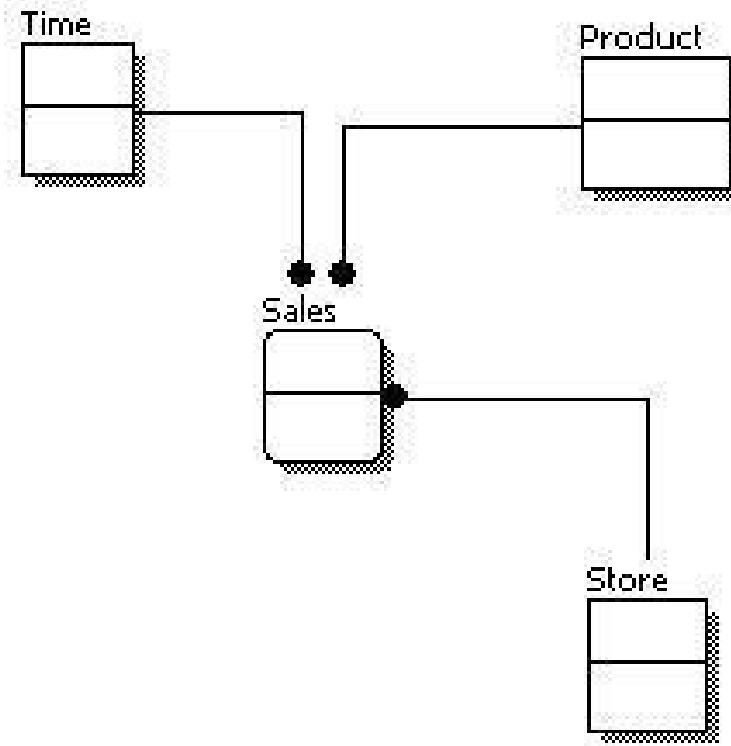
What is Data Modelling?

- Data modelling is the analysis of data objects and their relationships to other data objects.
- Often the first step in database design and object-oriented programming as the designers first create a conceptual model of how data items relate to each other.
- On approach to data modelling involves a progression from a conceptual model to a logical model to a physical schema.

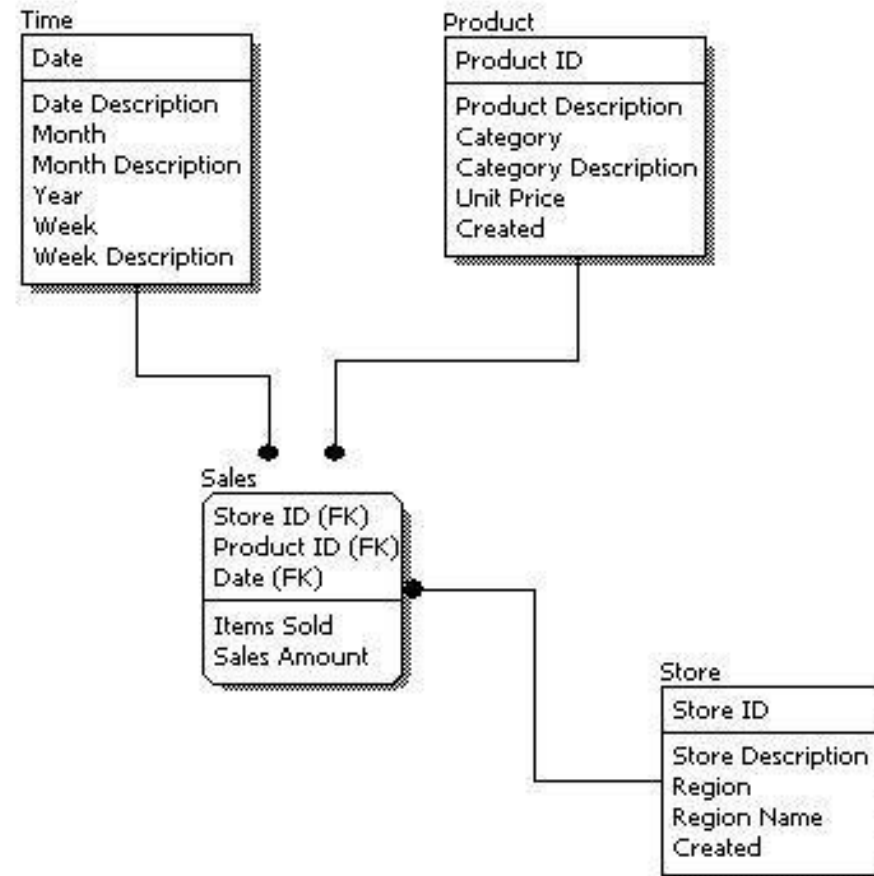
Levels of Modelling

- Conceptual
 - Describes WHAT the system contains - This data model includes all major entities and relationships.
 - Will not contain much detail about the attributes.
- Logical
 - Describes HOW the system will be implemented regardless of the DBMS you will use.
 - This is the actual implementation and extension of a conceptual data model into a logical data model.
- Physical
 - Describe HOW the system will be implemented using a specific DBMS – e.g. an Oracle 11g database.
 - Complete model that includes all required tables, columns, relationship, database properties

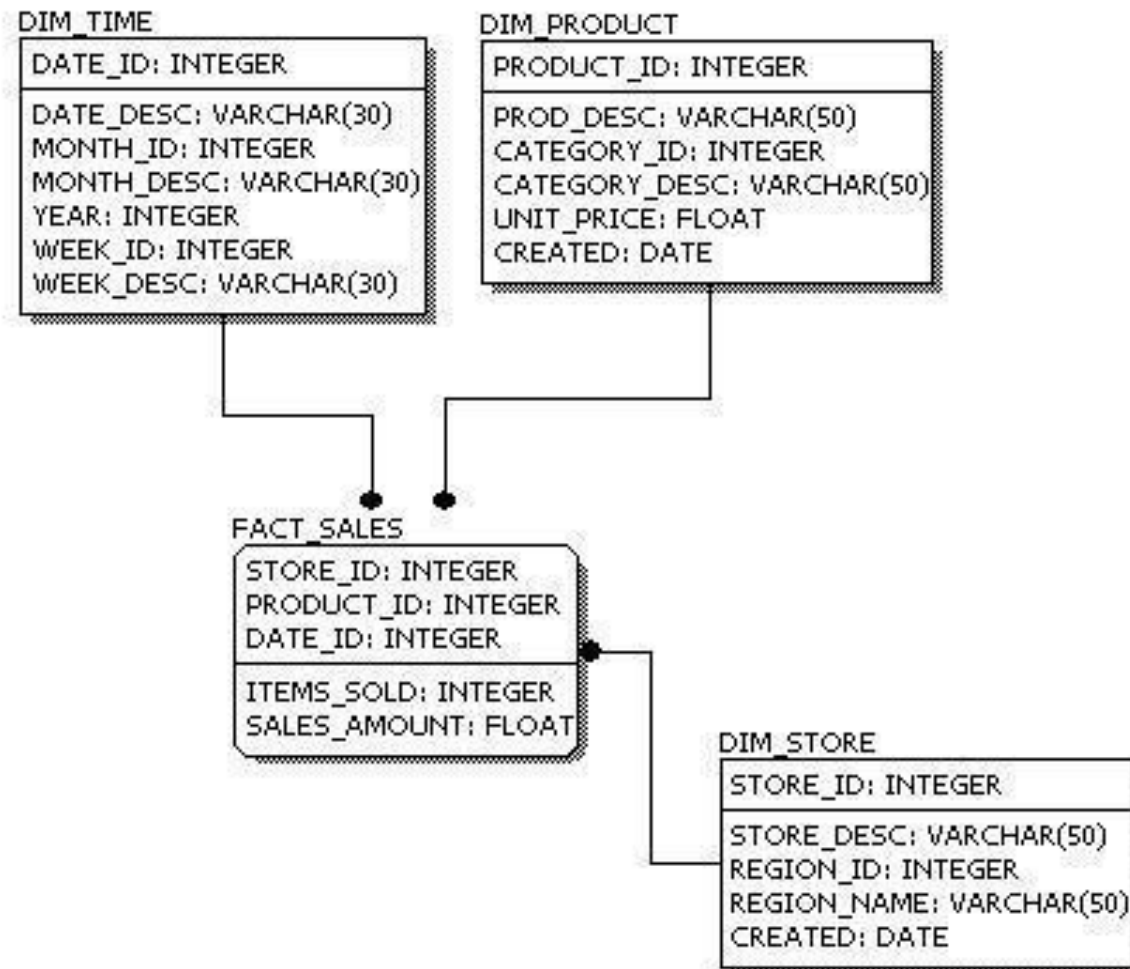
Conceptual Model



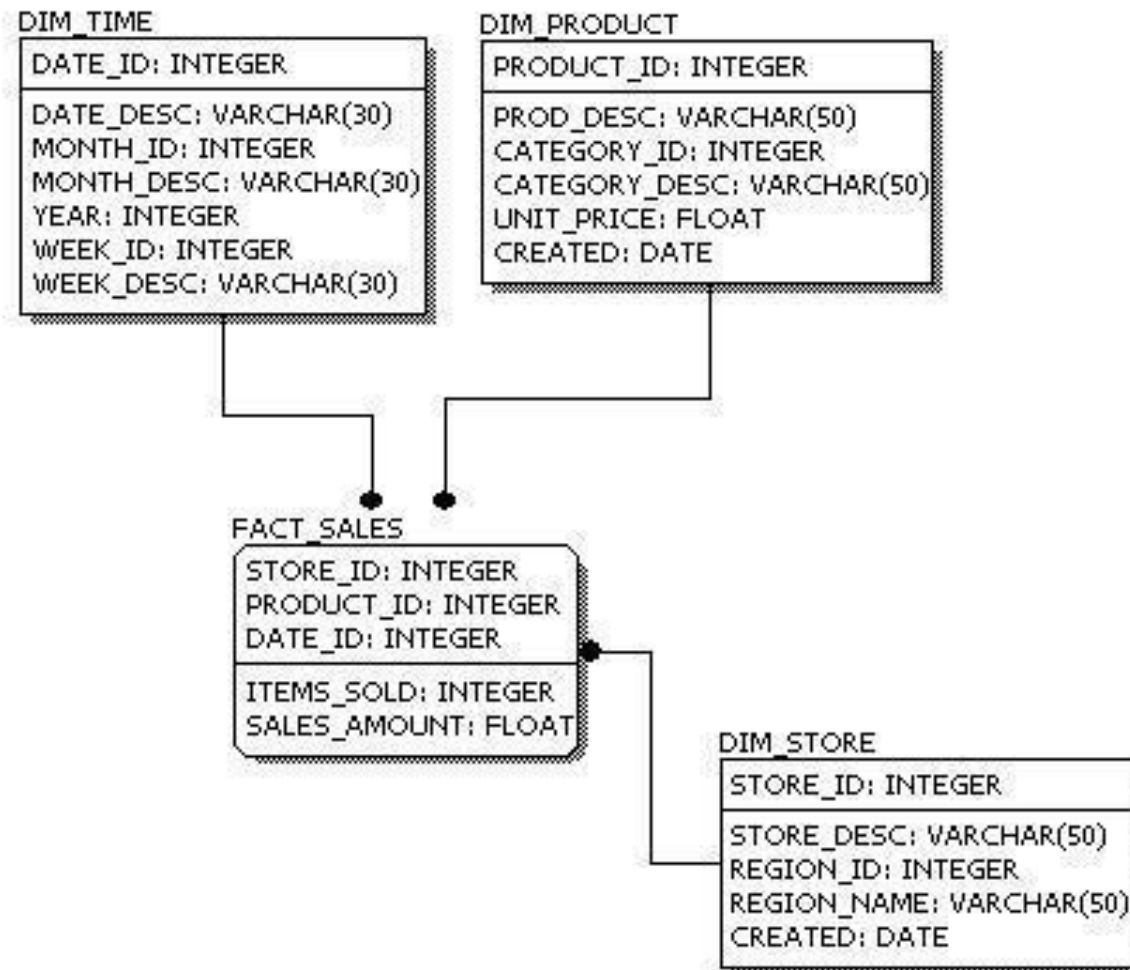
Logical Model



Physical Model



Physical Model



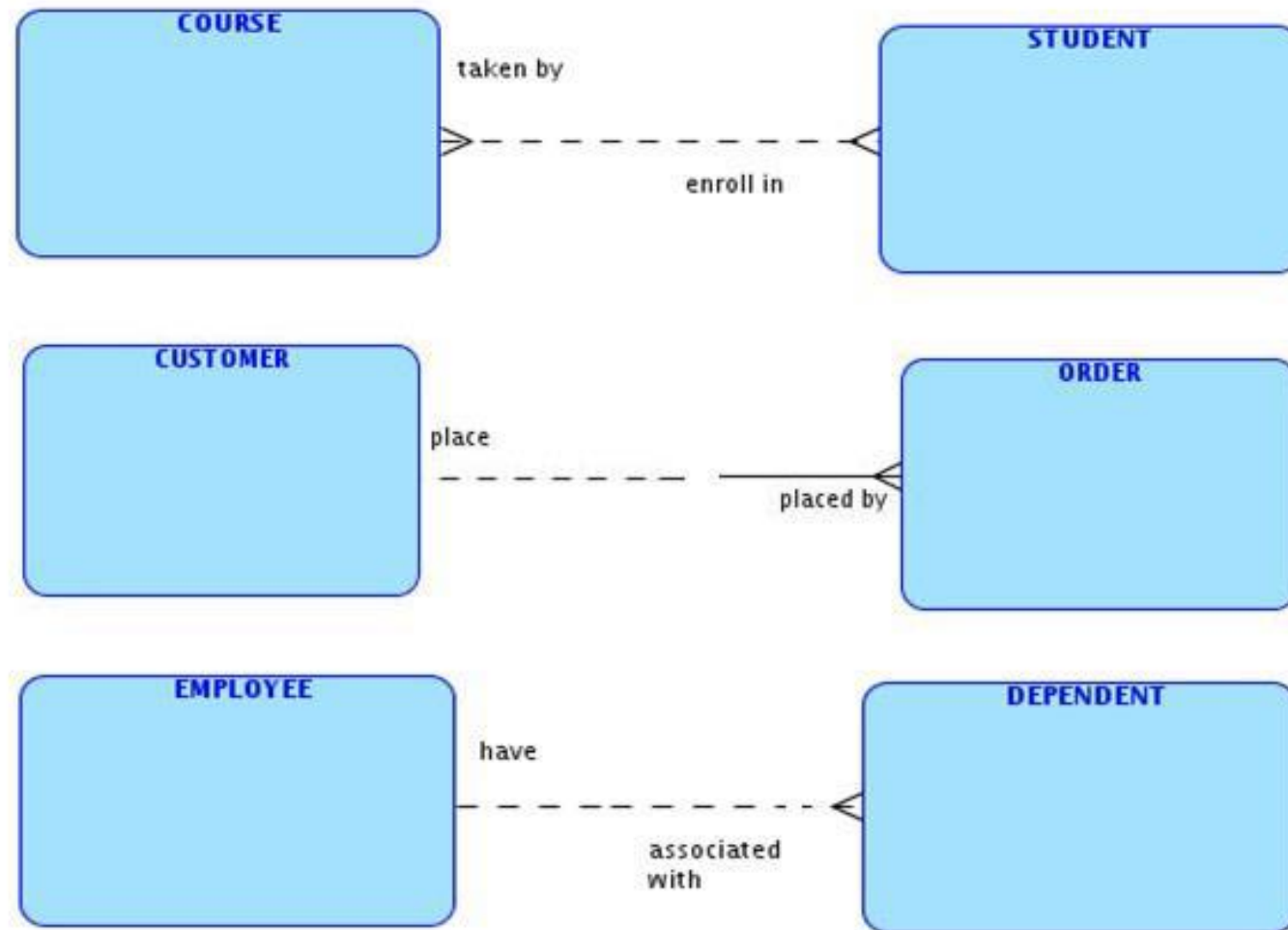
Entities, Attributes, Relationships

- Entities
 - Entities are categories of things that are important for a business and about which information must be kept.
- Attributes
 - Attributes are information about an entity that must be known or held.
- Relationships
 - A relationship represents the business rules that link entities.
 - Each side of a relationship has:
 - A name (for example, 'contain one' or 'assigned to')
 - An optionality (for example, either 'must be' or 'may be')
 - A degree (for example, either 'one and only one' or 'one or more')

Unique Identifiers

- A unique identifier (UID) is a special attribute (or group of attributes) that uniquely identifies a particular instance of an entity.
- Often designated in diagrams with a # symbol.
- Each component of a unique identifier must be mandatory.
- A unique identifier (UID) is a special attribute (or group of attributes
- An entity can have more than one unique identifier.
 - When this situation occurs, select one candidate unique identifier to be the primary unique identifier, and the others to be secondary unique identifiers.

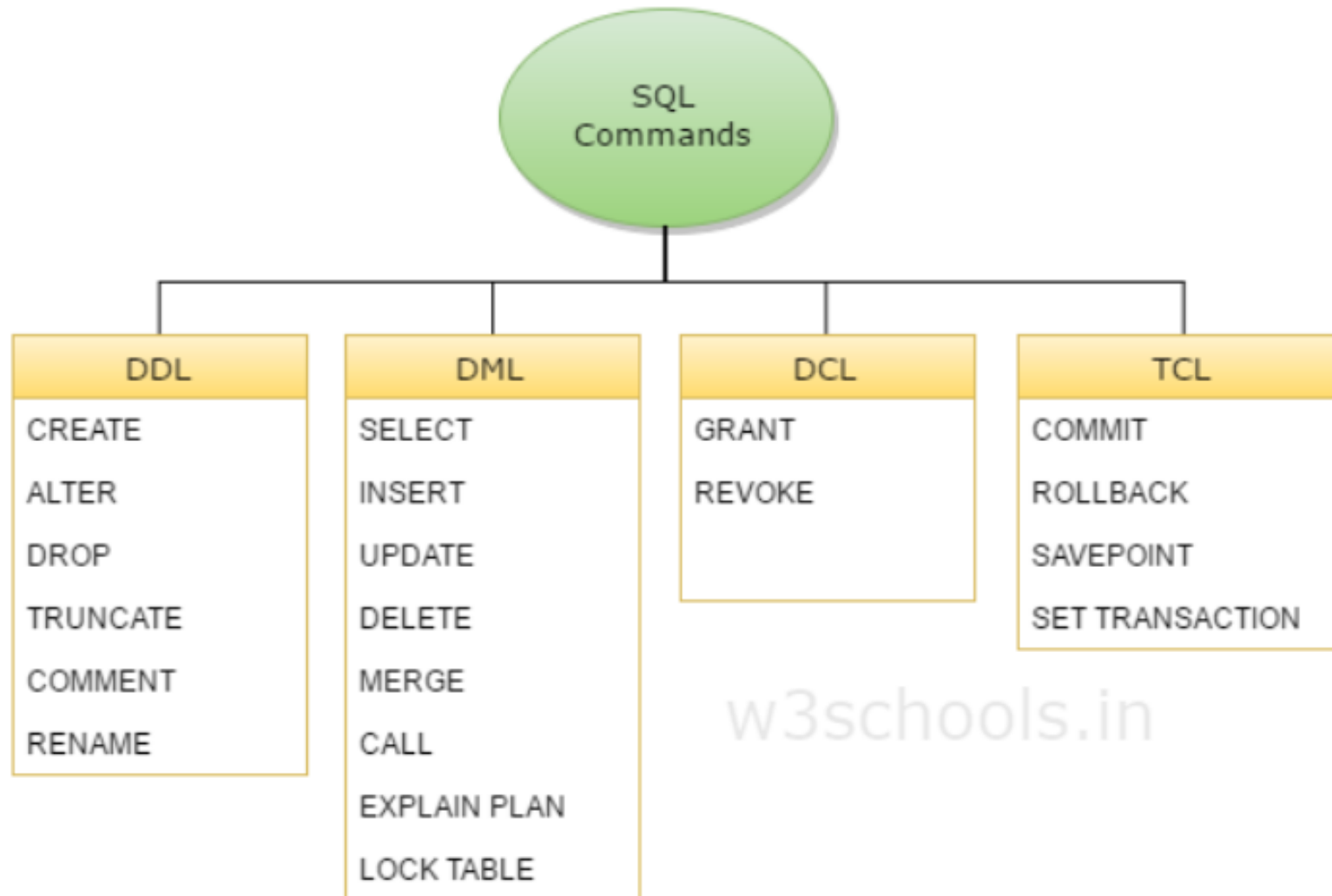
Describing Relationships



Describing Relationships

- **Many-to-one (M:1) or one-to-many (1:M):**
 - There are crow's feet on one side of the relationship.
 - The direction of the crow's feet determines whether the relationship is M:1 or 1:M.
- **Many-to-many (M:M):**
 - There are crow's feet on both sides of this relationship.
 - In relational models these relationships are commonly implemented through a new table to implement the relationship
- **One-to-one (1:1):**
 - This type of relationship is a line without crow's feet on either end
- In relational databases these relationships are described in SQL

Types of SQL Statement

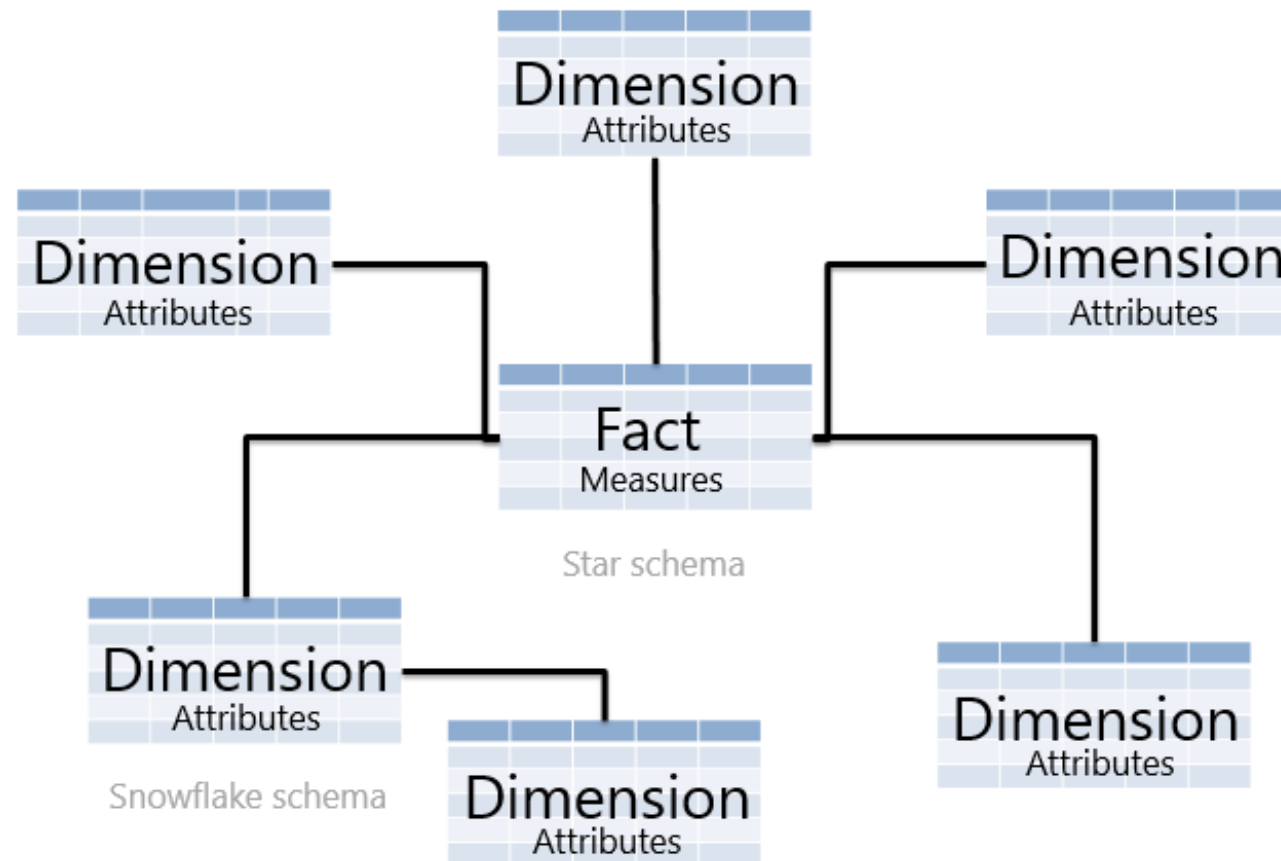


Dimensional Modelling

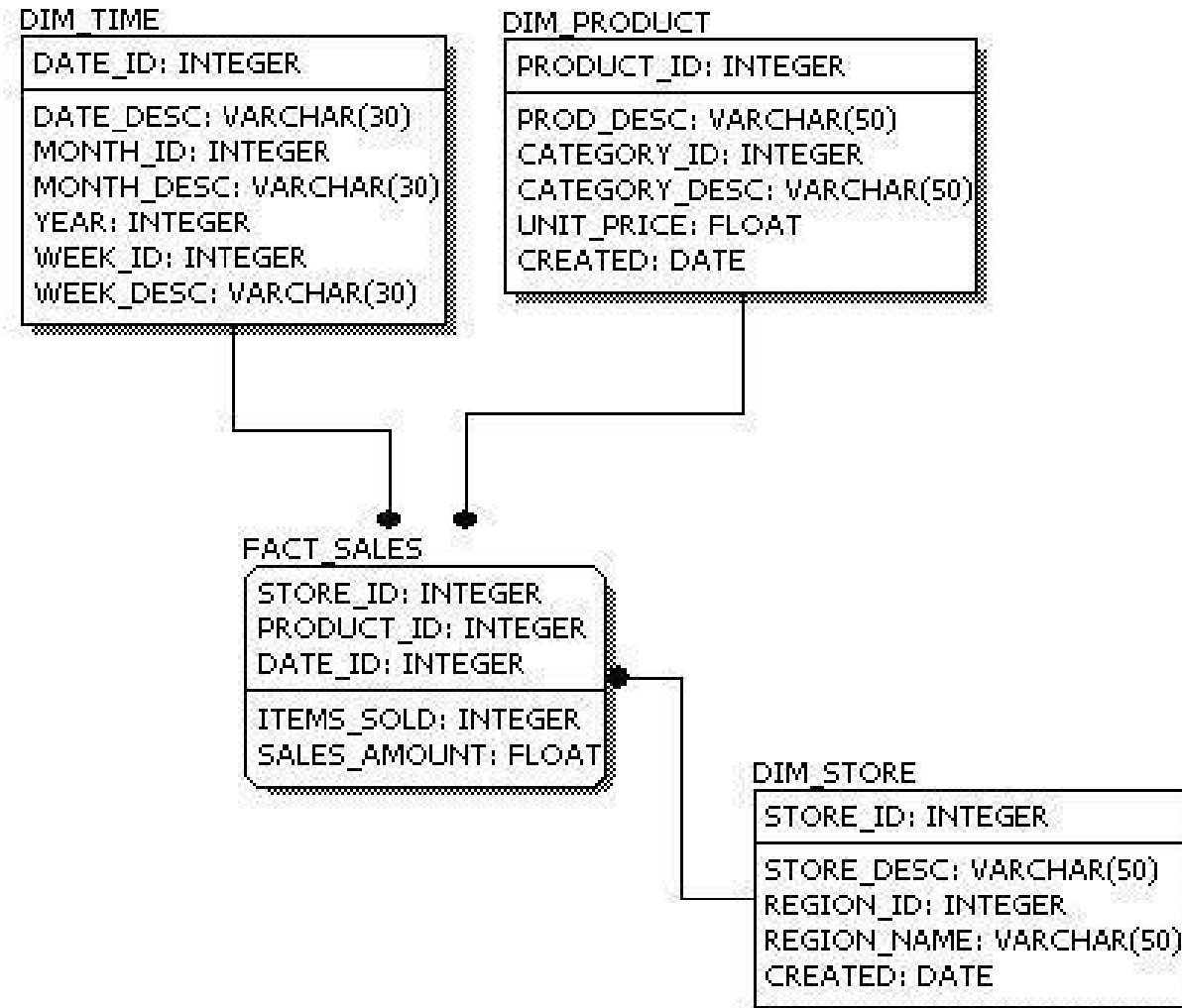
- Data warehouses can be implemented as normalized, relational database schemas,
- However many designs are based on the dimensional model advocated by Ralph Kimball.
- The numeric business measures that are analyzed and reported are stored in fact tables, which are related to multiple dimension tables, in which the attributes by which the measures can be aggregated are stored.
- E.g. A fact table might store sales order measures, such as revenue and profit, and be related to dimension tables representing business entities such as product and customer.

Dimensional Modelling

The Dimensional Model



Dimensional Modelling



Steps to Dimensional Modelling

1. Determine analytical and reporting requirements
2. Identify the business processes that generate the required data
3. Examine the source data for those business processes
4. Conform dimensions across business processes
5. Prioritize processes and create a dimensional model for each
6. Document and refine the models to determine the database logical schema
7. Design the physical data structures for the database

Dimensional Modelling

Business Processes	Conformed Dimensions								
	Time	Product	Customer	Salesperson	Factory Line	Shipper	Account	Department	Warehouse
Manufacturing	x	x			x				
Order Processing	x	x	x	x					
Order Fulfilment	x		x			x			
Financial Accounting	x						x	x	
Inventory Management	x	x							x

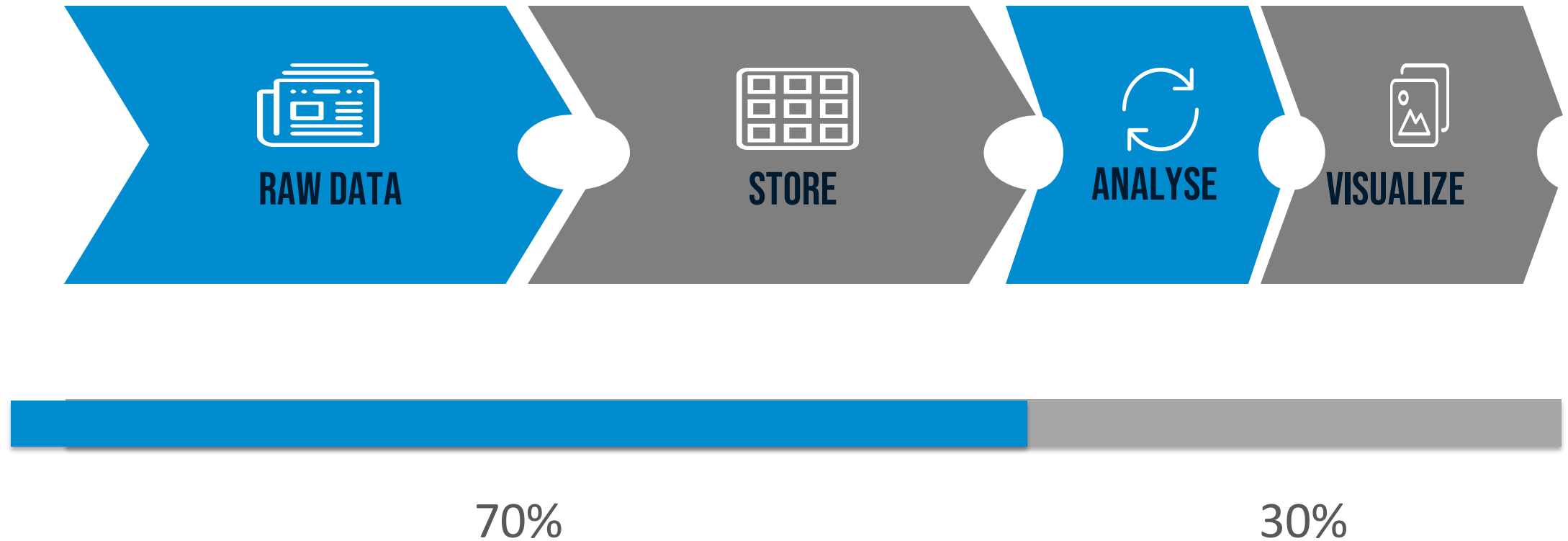
- **Grain:** 1 row per order item
- **Dimensions:** Time (order date and ship date), Product, Customer, Salesperson
- **Facts:** Item Quantity, Unit Cost, Total Cost, Unit Price, Sales Amount, Shipping Cost

Benefits of Dimensional Modelling

- Improved data retrieval
 - Dimensional models are optimized for SELECT operations.
- Simplified business reporting logic
- Fast aggregations
 - The simpler queries dimension model can result in improved performance for aggregation operations
- Better understanding
 - Everything in dimensional model falls into two tables, Fact and dimensions
- Extensibility
 - Dimensional models are scalable and can easily accommodate unexpected new data

Data Projects

The Data Science Process - Revisited



Data in the real world is dirty

- **Incomplete:**
 - lacking attribute values
 - lacking certain attributes of interest
- **Noisy:**
 - containing errors or outliers (spelling, phonetic and typing errors, word transpositions, multiple values in a single free-form field)
- **Inconsistent:**
 - containing discrepancies in codes or names (synonyms and nicknames)
 - prefix and suffix variations
 - abbreviations, truncation and initials
- **Lack of Currency**
 - Out of date
 - No longer relevant

Why is Data Dirty?

- Incomplete data comes from:
 - non available data value when collected
 - different criteria between the time when the data was collected and when it is analyzed.
 - human/hardware/software problems
- Noisy data comes from:
 - data collection: faulty instruments
 - data entry: human or computer errors
 - data transmission
- Inconsistent (and redundant) data comes from:
 - Different data sources, so non uniform naming conventions/data codes
 - Functional dependency and/or referential integrity violation

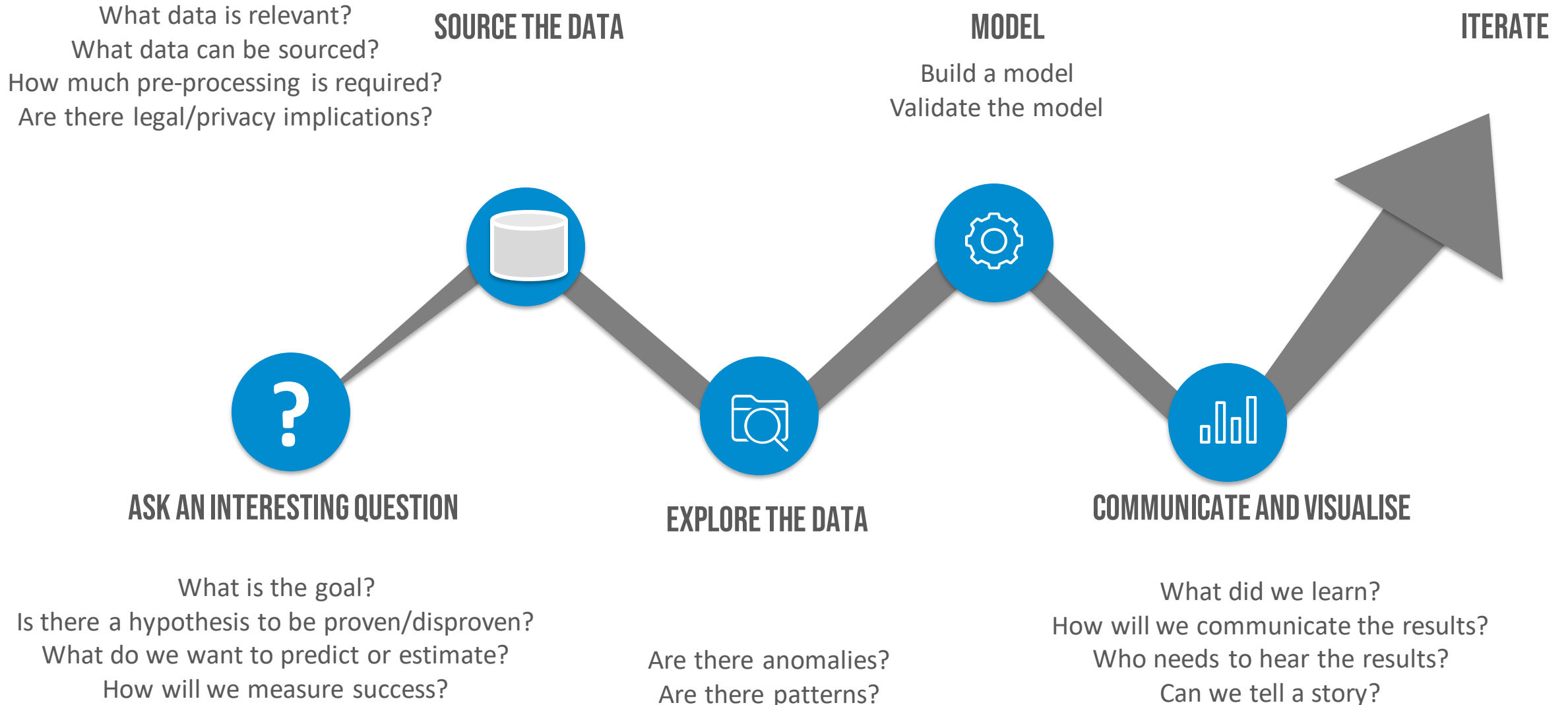
What do we need to ask about our data?

- Can we interpret the data?
 - What do the fields mean?
- Are there data glitches?
 - Typos, multiple formats, missing / default values
- Do you need metadata and/or domain expertise
 - Is the revenue field in cents, euros, or 1000s euros?

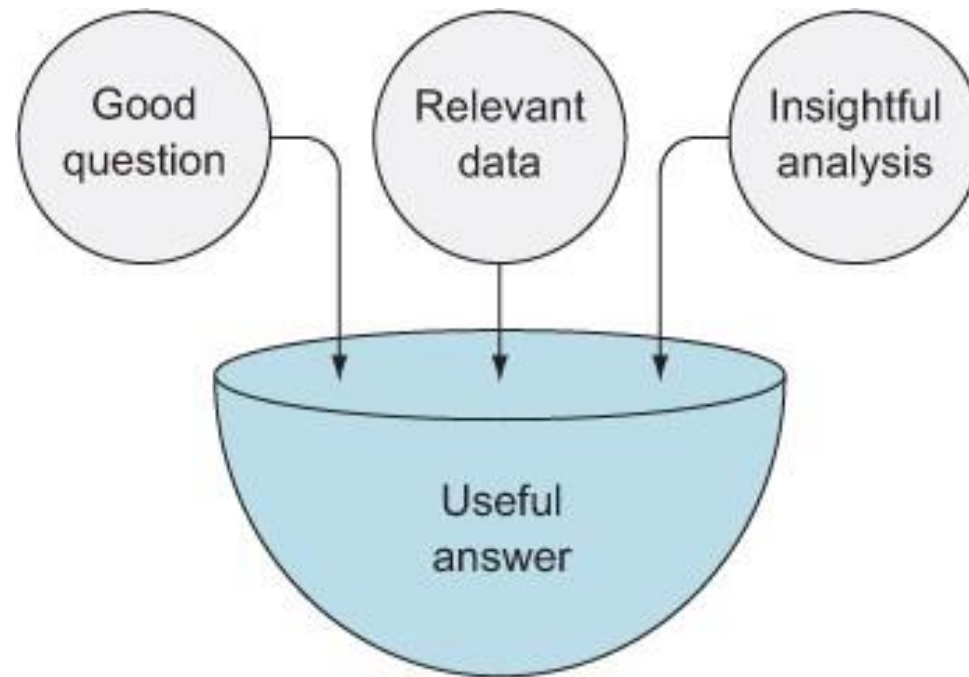
Cleaning Tools

- Most data analytics engines will include specific features for cleaning data
- "No Code" Data Scrubbing Tools:
 - help to clean and correct lists and databases by identifying incomplete, incorrect, inaccurate, irrelevant, etc. parts of the data and then replacing, modifying, or deleting this dirty data
 - E.g. XPlenty, Tibco Clarity
- Scripting Languages:
 - Often scripts in Python and R are used to do bulk clean-up.
 - There are libraries available to help e.g. treat all empty cells in the same way (eg fill with mean, media, or specific values, or be removed entirely)
 - There are libraries to enable processing of distributed data (e.g. Spark)

Data Project Process



Data Analysis



Getting Started on Projects

- Start small and iterate
- Be question or hypothesis led
- Solve real problems that address a business challenge and are actionable
- Involve all stakeholders
- Build partnerships between functions (IT, finance, business)
- Consider collecting data from new sources such as short surveys or other qualitative information
- Tell a story from a data

Don't Confuse Reporting with Analysis

Reporting



Analysis

- Asking questions and finding answers in the data
- Forming hypothesis, testing and measuring results in the data.
- Understanding the meaning behind the numbers and lines and taking action based on that new understanding

Failure Rate of Data Projects

- July 2019: VentureBeat AI reports **87% of data science projects never make it into production**
- Jan 2019: New Vantage survey reports 77% of businesses report that "business adoption" of big data and AI initiatives continues to represent a big challenge for business
- Jan 2019: Gartner says 80% of analytics insights will not deliver business outcomes through 2022

Reasons Projects Fail

- The question is not relevant to the business
- The timeline is too ambitious
 - Look for short term wins and build the analysis as you go
- Poor quality data
 - Fields are mislabeled or abbreviated in different ways across the company
- Failure to address legal, compliance, privacy and ownership issues
- Not possible to translate insights into action
- The data is not presented in a way that is meaningful to the consumer of the insights

Questions to Ask Prior to Starting A Data Project

The Problem

1. What is my business question?
2. What is the current solution to the problem? Will the proposed approach be i) better and ii) worth the cost of investment ?
3. Who are key stakeholders and what are their most important questions?
4. Who is the audience?
5. Who can access the information?
6. How will the results be used?
7. What reports will be produced? Detail the data and charts on each report.
8. What is the simplest and most effective solution to the problem that can be created quickly? i) one-off answer to support a strategic decision, ii) standalone light-weight app for stakeholders to use iii) a real-time data product that integrates into other systems?

Questions to Ask Prior to Starting A Data Project

The Data

1. What data sources are available to work with?
2. Will this data answer my business question?
3. What is size of each data set and how much data will you need to use from each one?
4. For each individual source – i) Is it complete? ii) Accurate? iii) Up to date?
5. Do you have the required permissions or credentials to access the data necessary for analysis?
6. What transformation needs to be done on the data?
7. What is the frequency of updates required for the data?
8. Who will maintain the data?
9. Have you addressed all legal and compliance issues related to this data?

Questions to Ask Prior to Starting A Data Project

Predictive Models

1. How will you validate your model?
2. How will you gather feedback on the solution?
3. How will you monitor your model?
4. How will you maintain your model?

Questions to Ask Prior to Starting A Data Project

The Project

1. How will success be measured?
2. What is the ROI?

Data Policies

- Create clear, consistent policies that are readily available and easy to understand
 - Transparency
 - Consent
 - Privacy
 - Anonymization
 - Usage and aggregation
- Encrypt and secure all data
- Aggregate data and report on aggregated results

Thankyou
