

# What we should consider when we design a database:

---

**Schemas:** How should my data be logically organized?

**Normalization:** Should my data have minimal dependency and redundancy?

**Views:** What joins will be done most often?

**Access control:** Should all users of the data have the same level of access

**DBMS:** How do I pick between all the SQL and noSQL options?

and more!

## Approaches to processing data

---

**OLTP:** Stand for the Online Transaction Processing. They are application oriented. We have a snapshot of data, with the gigabyte size. This is used by thousands of people like consumers. Examples: find a price of a book, customer transaction, employees' hours

**OLAP:** Stand for Online Analytical Processing. Oriented around a certain subject that's under analysis like average sale. OLAP is used in analysts and data science companies

**OLAP** contains the historical data, archive at the order of terabyte Calculate books with best profit, Find the average transaction, ..

**Both of them working together. OLTP is stored in an operational database and provide data and clean them for the OLAP warehouse. Before implementing anything we should figure out business requirements**

## Sorting data: data can be saved in 3 different levels such as:

---

1. Structured data

It follows a schema and the data types and tables are defined and relationships between them should be defined e.g., SQL ,tables in a relational database

## 2. Unstructured data

Schemaless like most of data in the world e.g., photos, chat logs, MP3

## 3. Semi-structured data

Does not follow larger schema and it has self-describing structure e.g., NoSQL, XML, JSON

# Beyond traditional databases

---

### **Traditional databases:**

For storing real-time relational structured data such as OLTP

**Data warehouses:** For analyzing archived structured data such as OLAP

Organized for reading & aggregating data. They are optimized for read-only analytics. They contain data from multiple sources and use the massively Parallel Processing (MPP) for faster queries. In designing their database they typically use a denormalized schema and dimensional modeling. Amazon, Google and Microsoft provide data warehouse solutions. A subset of data warehouses is called data mart and dedicated to a specific topic. They are Schema-on-write (the schema is predefined)

**Data lakes:** For storing data of all structures e.g., raw, operational databases, IoT device logs, real-time, relational and non-relational. They retain all data and can take up petabytes. Schema-on-read (the schema is created as the data is read) as opposed to schema-on-write. Data lakes need to catalog data otherwise become a data swamp. Run big data analytics using services such as Apache Spark and Hadoop.

Data lakes are useful for deep learning and data discovery because activities require so much data. When we think about where to store the data we should think about how the data get there and in what form:

**ETL:** Extract transform load: Traditional for smaller scale

**ELT** Extract load transform: Newer of bigger data

Database design is defined as how data is logically stored. And how is data going to be read and updated? Two important concepts for design: **Data base model, Schema.**

---

## Data base model

Data base models defines the database structure such as relational model. This is the first step toward the database design.

Some other options: NoSQL models, object-oriented model, network model

Process of creating a data model for the data to be stored

1. Conceptual data model: describes entities, relationships, and attributes

Tools: data structure diagrams, e.g., entity-relational diagrams and UML diagrams

2. Logical data model: defines tables, columns, relationships

Tools: database models and schemas, e.g., relational model and star schema

3. Physical data model: describes physical storage

Tools: partitions, CPUs, indexes, backup systems and tablespaces

## Schema

Schemas is the implementation of the database model. When we insert data in rational databases the schema should be followed. Defines tables, fields, relationships, indexes, and views

## Dimensional modeling

---

This is the adaptation of the relational model for data warehouse design. It is optimized for OLAP queries: aggregate data, not updating (OLTP). This is Built using the star schema and Easy to interpret and extend schema.

Two types of tables dimensional modeling:

1. Fact table: Decided by business use-case. Holds records of a metric .Changes regularly  
Connects to dimensions via foreign keys
2. Dimensional table: Holds descriptions of attributes Does not change as often

