# Introduction

## What is DBMS (**D**ata **B**ase **M**anagement **S**ystem)?

Data: Information about something

Base: Store

Database: Storage of information / Collection of data

System: Organise way of doing something

Management: We are trying to manage the data here

DBMS: Software system that allow us to store, manage, fetch data

## What are the types of Databases?

1. Relational Database (SQL)

MySQL, Azure SQL

1. Non-Relational Database (NO SQL)

Mongo, Cassandra

## Relational Data Model?

Relational: Anything that is related.

Data: Information about something

Model: Anything which is describing something

Data Model: Representation of the data.

Formal Definition: Data represented as multiple relations that are related to each other.

Relation: Table

Attributes: Columns of a table

Tuple: Row of table

Degree: number of columns

Cardinality: number of records/Tuples.

Table: Collection of rows

## Important points

* When querying the data order of retrieval is never guaranteed. Because data is stored as SET in table
* Order of the column does not matter.
* Value in Each cell should be atomic (single smallest value possible). Multi valued values are not allowed that means No List, No Set, No Json
* Each row needs to be unique (data is set is also unique)

## Keys

A set of columns that help us to uniquely identify the row in a table.

If we match value of these columns which will result in one row only.

## Types of Keys

1. **Super key**

Any set of columns whose values can uniquely identify a row.

1. **Candidate key**

Key of minimum size

Minimum: Something you cannot break. If you break that it will lose its property.

If we remove any column then the resultant key is not a key anymore.

1. **Primary key**

It is a candidate key that has been assigned as primary key while creating table in DB.

Primary key mostly used for

auto incremental of DB

auto id generator of DB

used for UUID also.

1. **Foreign key**

*Constraints*

Restricted / not allowed

Cascade

NULL

1. **Composite key**

Candidate key with more than one column

## SQL (Structured Query Language)

Mechanism or Syntax to Query a relational database.

SQL is query language and MYSQL is database.

1. Data manipulation (Store | fetch | update)
2. Data Control (access control)
3. Data definition (Declaration of the table)

# Schema Design

## Note:

<https://docs.google.com/document/d/1qLOvl019ofVxvxqiBwRWI1yQuQSYdy211QTZ-yaGZTg/edit>

* Create database
* Create table
* Alter table
* According to research only 12% to 20% of the time we write code. The Rest is planning requirements design.

## What is the typical structure of design doc?

Overview / problem statement

Current state

Scope of problem

HLD with approach: Need approval by architect

LLD with approach: class diagram, DB schema: Need approval by architect

Deployment/Testing plan: Need approval by architect

Program plan

Metrics / KPIs

Effort estimation

## What is DB schema, why we need it?

How will my DB look like so that it can handle my requirement.

Blueprint of production (REAL) DB.

Pictorial representation.

To ensure DB can handle the requirements efficiently.

## DB normalization?

Set of techniques to reduce the redundancy in DB.

What is Redundancy here in the below table Aug 22 is repeating. That is same data is being duplicated, that is we are replicating value at multiple places.

Id, Name, psp, batch id, batch name

1, Nikhil , 80 , 1 , Aug22

2, Akshay , 80 , 1 , Aug22

3, Krishna , 82, 1 , Aug22

This duplication leads to anomalies (problems)

* Insertion problem

Id, Name, psp, batch id, batch name

1, Nikhil , 80 , 1 , Aug22

2, Sep 23

We can not have batch without student, but there can be a student without batch. Insertion or creation will fail. So split the table into 2 tables.

Student Batch

Id, name, psp, batch id Id, Batch name

So now we can create a batch without student

* Deletion Anomaly

Id, Name, psp, Batch id, Batch name

1, Nikhil , 80 , 1 , Aug22

If I delete Nikhil data here, I will have no trace of batch.

At the time of deletion of something it will delete some data unintentionally.

* Update anomaly

Id, Name, psp, Batch id, Batch name

1, Nikhil , 80 , 1 , Ag22

-, - , - , - , Ag22

-, - , - , - , Ag22

-, - , - , - , Ag22

To update the batch from Ag22 to Aug 22. We must write one update statement to change in every place.

Let’s say there are 1 million students in Ag 22, then the query will take lot of time for example 4 hrs. In between if DB gets restarted then there will be discrepancy in the data.

Also, just to change one data we must update lot of rows. So, if there are 2 different table

Student Batch

Id, name, psp, batch id Id, Batch name

Then we just must change the name of the batch.

## DB Normal forms?

1NF

2NF

3NF => BCNF ( Boyce Codd NF )

4NF

5NF

6NF

As we go deeper in NF It becomes stricter. In practical life scenario if we go beyond 3NF then we will get worst performance. 4NF to 6NF are not important.

Refer this YouTube link to understand normalization.

<https://www.youtube.com/watch?v=GFQaEYEc8_8>