→ What is RDBMS

a. RDBMS stands for Relational Database Management System
b. Relational Database Management System is a database management system based on relational model
c. In this model, data is stored in tables(relations), and is represented in the form of tuples(rows)
d. RDBMS is used to manage relational database
e. A relational database is a collection of organized set of tables, that are related to each other
f. Relational database helps us to access the data easily as all tables are related to each other
g. RDBMS helps us to do basic manipulations like Create, Read, Update & Delete, and these operations are collectively called as CRUD
h. Advantages of RDBMS:
    i. This support very large databases
    ii. RDBMS uses SQL(Structured Query Language)
    iii. RDBMS defines how data is organized in table and the relationship among these
    iv. New data can be added without modifying existing records, which in turn provides scalability
i. Disadvantages of RDBMS:
    i. RDBMS does not support complex objects such as documents, videos and images

→ What is table

a. A Table is a collection of data entries and contains row & columns to store data
b. Each table represents some real-world entities like a person(Employee, Student, etc.), any event about which we collect information
c. Properties of a table:
    i. Each table is identified by a unique name in the database
    ii. Relations/tables does not contain duplicate tuples/rows
    iii. Tuples/rows of table have no specific order. All attributes in a table are atomic

Student

Student_ID	Student_Name	Course_ID
100	ABC	CS
101	XYZ	MECH

    iv. ‘Student_ID’, ‘Student_Name’ and ‘Course_ID’ are the attributes of the ‘Student’ table

→ Record / Row / Tuple

a. A single entry in a table is called as tuple/record/row

Student

Student_ID	Student_Name	Course_ID
100	ABC	CS
101	XYZ	MECH

b. The one highlighted in red color is called tuple/row/record. Each record/row in a table is considered as a tuple
c. The order of records is irrelevant, they are identified by their content and not by the position at which data is stored

Student

Student_ID	Student_Name	Course_ID
100	ABC	CS
100	ABC	CS

d. No two rows can have identical /duplicate entries, so above two duplicate entries are not allowed

→ Columns/Attributes

a. A column is a vertical entity in a table, which contains information related to a table

Student

Student_ID	Student_Name	Course_ID
100	ABC	CS
101	XYZ	MECH

b. The above marked in red color viz., ‘Student_ID’, ‘Student_Name’ and ‘Course_ID’ are attributes for ‘Student’ table

Student

		Course_ID
100	ABC	CS
101	XYZ	MECH

c. Every column should have an attribute name, so above table is not allowed

Student

Student_ID	Student_Name	Course_ID
100	ABC	
101	XYZ	MECH

d. A null value is permitted for an attribute.e.g. For a student with ‘Student_ID’ as 100 has NULL (blank) value for ‘Course_ID’

→ Attribute Domain

a. Attribute domain refers to possible value for each attribute, that it may contain
b. It can be specified using data types

Student

Student_ID	Student_Name	Course_ID
CFG	ABC	CS
101	XYZ	MECH
102	BCD	E&TC

c. e.g. : an attribute ‘Student_name’ cannot contain an integer or float value

→ Degree

a. Total number of attributes that comprises in a table, is known as degree of that table

Student

Student_ID	Student_Name	Python_Marks	DB_Marks
100	ABC	12	14
101	XYZ	15	12

b. e.g. : in the above table, we have 4 attributes, so the degree of Table ‘Student’ is 4

→ Cardinality

a. The total number of records in a table is known as cardinality

Student

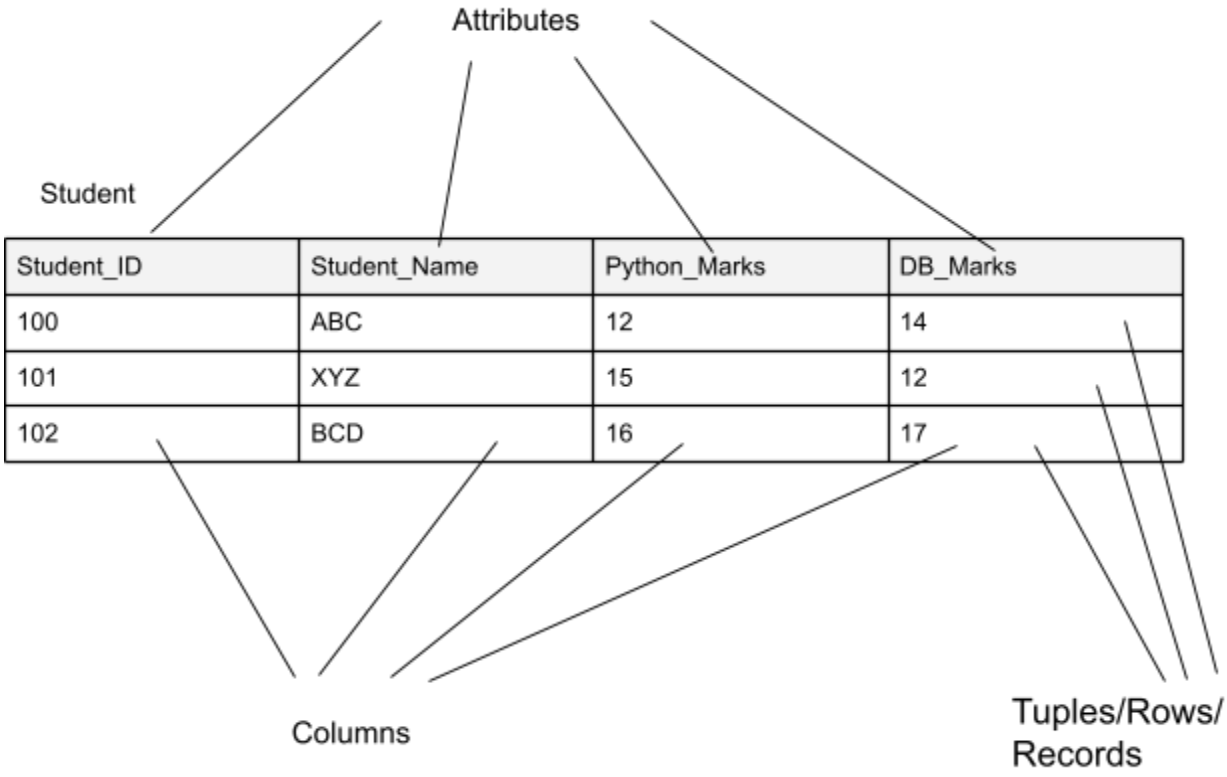
Student_ID	Student_Name	Python_Marks	DB_Marks
100	ABC	12	14
101	XYZ	15	12
102	BCD	16	17

b. e.g.: in the above table , we have 3 records/rows, so the cardinality of the table ‘Student’ is 3

→ NULL values

a. A null value of a table specify the field that is left blank

→ Parts of a table



→ Codd Rules

a. Invented by EF Codd in 1970
b. Rule 0 (The Foundation Rule):
    i. This rule states that an application or a system, in order to qualify as an RDBMS, it must be able to manage the database entirely through relational capability
    ii. e.g. : suppose, we are designing a library management system, and we need to create tables like book_details, issuer_details, etc. In this system, we will have all the tables/relations related to library only (we cannot add tables of other entities like employee details)
c. Rule 1 (Information Rule):
    i. All information is to be presented or stored in table format (rows & columns)
d. Rule 2 (Guaranteed access Rule):
    i. Each & every table/relation must be logically accessible using table name/ primary key and column name.

    ii. This means that each table will have a unique name, a table must have a primary key, and to access any data through attribute name should be proper and atomic

e. Rule 3 (Systematic treatment of NULL values):
    i. NULL values are fully supported in RDBMS
    ii. NULL is a term used to represent missing value(s)
    iii. A field with NULL value should be treated properly, so that accessibility of any kind of data is not restricted
    iv. We may assign few dummy values to handle NULL values
f. Rule 4 (Active/Dynamic Online Catalog based on Relational Model):
    i. Data dictionary is a structure description of complete database and it must be stored
    ii. A metadata about the table contains constraint, keys, data types, etc and these data should be stored in a table format so that privileged user can access these tables
g. Rule 5 (Comprehensive data sublanguage tool):
    i. Relational system supports many languages but there should be one language using which we can do Data Definition, Data Manipulation, Data Integrity, providing the constraints, etc.
h. Rule 6 (View Updating Rule):
    i. All the views should be updated as soon as table is updated
i. Rule 7 (High Level Insert, Update and Delete):
    i. There must be Insert, Update and Delete operations at each level
    ii. Operations like union, intersection should also be supported
j. Rule 8 (Physical Data Dependency):
    i. The Physical storage of data should not matter if any file/table is renamed or its location is changed, it should not affect the application
k. Rule 9 (Logical Data Independency):
    i. If there is a change in logical structure (table structure of database), the user view of the database must not change
l. Rule 10 (Integrity Independency):
    i. If there is a change in logical structure, the user view should not be changed as data integrity should be maintained throughout the manipulation process
m. Rule 11 (Distribution Independency):
    i. A database should work properly regardless of its distribution across a network
    ii. The end user should not see the data distributed over many locations, they should get an impression that data is located at a single location
n. Rule 12 (Non-subversion rule)
    i. If low level access is allowed to a system, it should not be able to bypass the integrity rule to change the data

→ DBMS vs. RDBMS

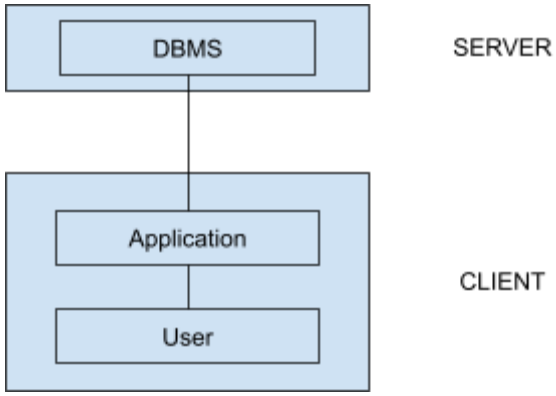
DBMS	RDBMS
There is no concept of relations	Whole RDBMS is based on relations
The speed of any operation in DBMS is very slow	The speed of operation is very fast as it deals on basis of relations
DBMS uses the concept of files	RDBMS uses the concept of tables
e.g. : FoxBase	e.g. : MySQL, OracleSQL
Normalization is not present	Normalization is there
DBMS does not support distributed database	RDBMS supports distributed database

→ DBMS Architecture

a. This is based on client-server architecture
b. Client is our system or workstation through which we make the request and servers are the machines that give us the responses and store all the data
c. DBMS architecture depends on how a user is connected to the database to get its response

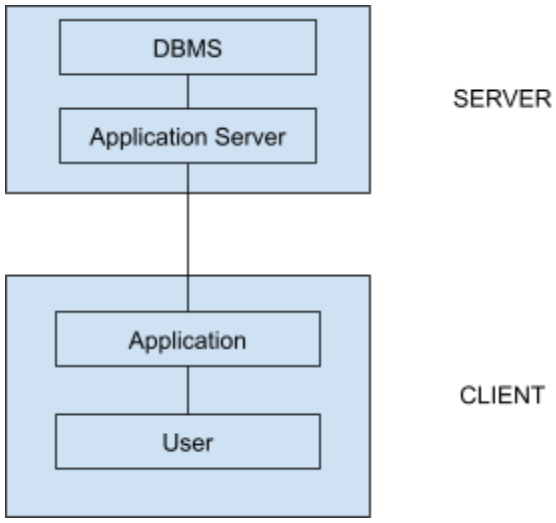
→ Types of DBMS Architecture

a. There are three types of DBMS architectures:
   i. <u>Tier 1 architecture:</u>
      1. DBMS architecture in which client server and database all reside at one system comes under this category.
      2. e.g.: MySQL which we have installed on our system
      3. In this architecture, a database is directly available to the user.
      4. Here any changes if performed are applied directly on the database
      5. These architecture is used to develop any local application, where programmers can directly communicate with the database
   ii. <u>Tier 2 architecture:</u>
      1. It is same as client-server architecture
      2. Application on client can directly communicate to the database at server end using API's like JDBC, ODBC, etc.



iii. Tier 3 architecture:

1. This type of architecture is used for large websites like google.com, amazon.com, etc.
2. In this architecture, client cannot directly connect to server
3. The application at client end will communicate to application server which further communicates to database system



→ DBMS Schemas

a. In DBMS, schema is the logical structure of the database or the blueprint of the database
b. It defines data and their relationship
c. These schema are used to provide data abstraction to hide the data complexity from the user
d. These three levels of architectures enable multiple users to access the same data with personalized view while storing the data
e. Different users needs different views on the same data
f. The user should not be worried about physical implementation and internal working of database
g. All users should be able to access same data according to their requirements

h. DBA(DataBase Administrator) should be able to change the conceptual structure of database without affecting the user

→ Levels of Data Abstraction

a. There are three levels of data abstraction

i. <u>Internal abstraction:</u>

1. This internal level has an internal schema which describes the physical storage & structure of the database.
2. This is also referred as physical schema
3. It is used to define how data will be stored in a block
4. This is the lowest level of abstraction
5. It helps the user to keep information about actual representation of the entire database
6. The internal level is generally concerned with storage space allocation like Binary Tree, Hashing, the Access Path (using indexes or keys or pointers or sequence)
7. *Example – Internal View:*

Stored Item	Length
Emp_Id:	Type=Byte(6)
Emp_Name:	Type=Byte(36), offset=10
Emp_sal:	Type=Byte(6), offset=8

ii. <u>Logical / Conceptual Abstraction:</u>

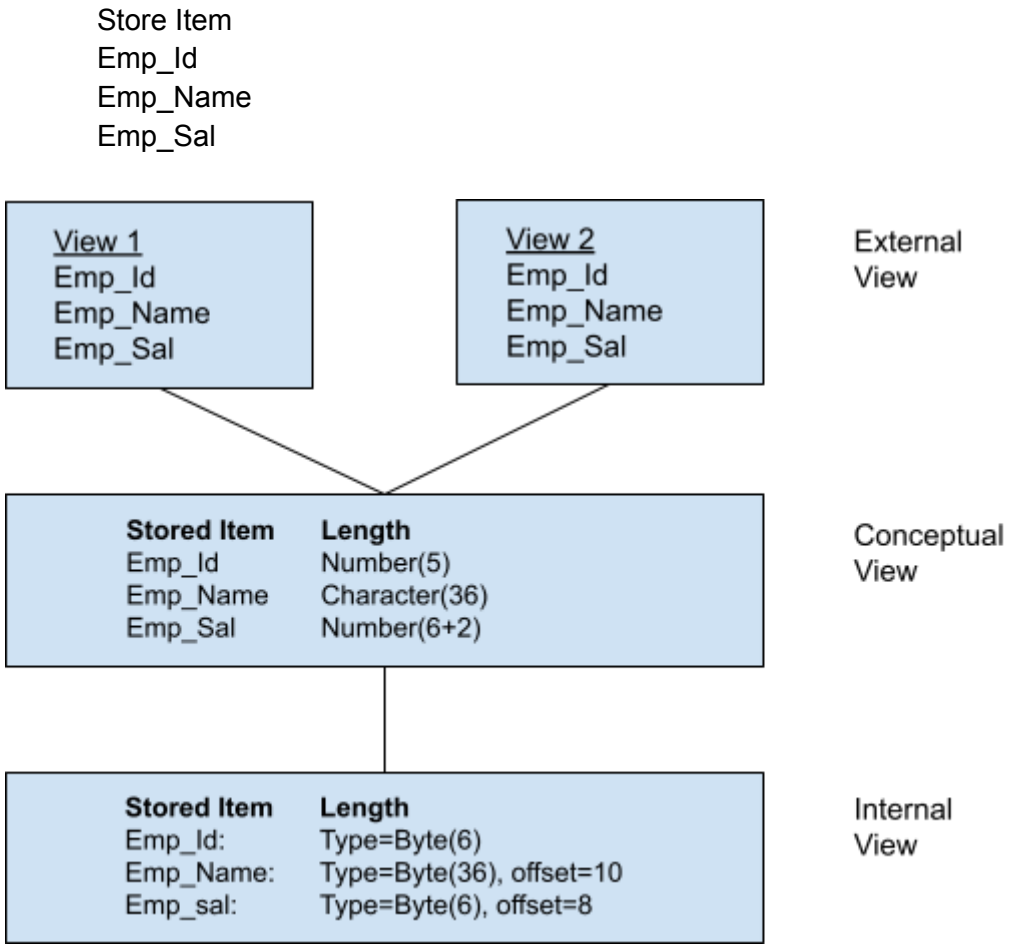
1. In conceptual level/schema, we focus on describing, data types, entities, relations, constraints, etc.
2. This level comes in-between user level/view level and internal/physical storage
3. There is only one conceptual view of single database
4. In conceptual schema, we describe the structure of whole database
5. Programmers & DBA's (DataBase Administrators) work at this level
6. *Example – Conceptual View:*

Stored Item	Length
Emp_Id	Number(5)
Emp_Name	Character(36)
Emp_Sal	Number(6+2)

iii. <u>External / View abstraction</u>

1. At external level, database contains several schema, which is referred as sub-schema
2. These sub-schema are used to describe different view of the database
3. This is also referees as view schema
4. This schema refers to end user interaction with database system
5. An external level is only related to data which is used by a specific user

6. *Example – External View:*



→ Key

a. In DBMS, an attribute or a set of attributes helps us to identify a record/row in a table
b. These keys help us to find relationship between two tables
c. Keys are used to uniquely identify a row in a table like combination of two or more attributes in a table

Example

s_id	s_name	address	mobile_no	email_id	DOB

→ Types of keys

a. There are many types of keys:
    i. <u>Super key</u>
        1. Super key is a group of single or multiple keys, which identify a row in a table

2. In super key, we may have attributes which are not needed for unique identification

Can be turned into super key

s_id			mobile_no	email_id	

ii. Primary key

1. An attribute that helps you to uniquely identify a row in a table, that can be a primary key
2. A primary key cannot be duplicate, which means an attribute like student_Id cannot hold duplicate data
3. *Rules for primary key:*
    a. In a table, we cannot, we cannot have two primary keys
    b. A primary key cannot be NULL
    c. A primary key cannot hold duplicate data
    d. The value of primary key column cannot be changed or modified if any foreign key refers to that primary key
    e. In the above case, we can make student_Id as primary key

Can be turned into primary key

s_id					

Cannot be turned into primary key

	s_name	address			DOB

    f. In the above case, we can make ‘s_id’ as primary key
    g. One student can have multiple ‘mobile_no’ and multiple ‘email_id’, hence, they cannot be turned into primary key

iii. Candidate key

1. A super key, with no repeated attributes is called as candidate key
2. A primary key must be chosen / selected from candidate keys

Can be turned into candidate key

s_id			mobile_no	email_id	

3. *Rules for candidate key:*
    a. It must contain unique values
    b. candidate key can have multiple attributes
    c. It should contain values that provide uniqueness
    d. A candidate key must not contain NULL values

iv. Alternate key

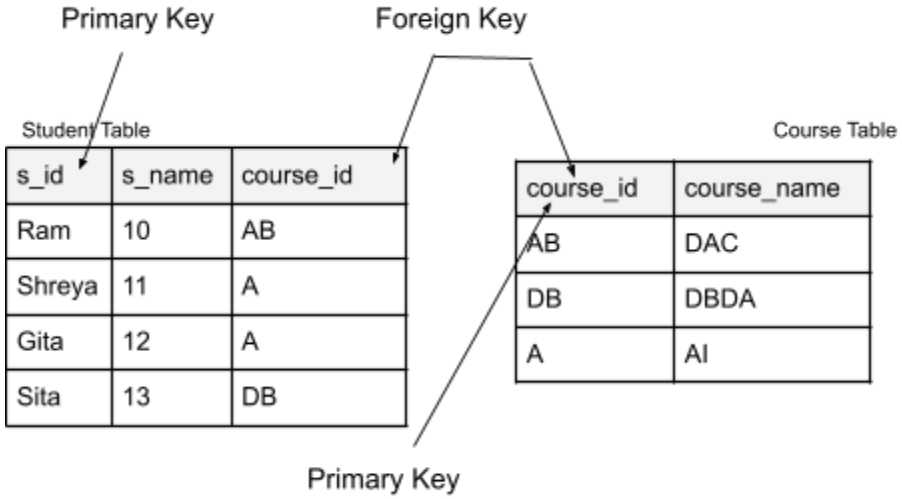
1. All the keys from the set of candidate keys, which are not primary keys are referred as alternate keys

Can be turned into alternate key

			mobile_no	email_id	

v. Foreign key

1. Foreign key is a column which is added to create a relationship with another table
2. One table's primary key acts as foreign key in other table



s_id	s_name	course_id
Ram	10	AB
Shreya	11	A
Gita	12	A
Sita	13	DB

course_id	course_name
AB	DAC
DB	DBDA
A	AI

vi. Composite key

1. Composite key is a set of primary keys that consist of two or more columns
2. It is used when one single column cannot be used to uniquely identify each row

vii. Surrogate key

1. <research it>

→ Users

a. User in a database are an individual or an application that interacts with database to perform various tasks like retrieving data, manipulating data
b. Users are broadly categorized in three sections, based on their roles, responsibilities and the level of access to the database

→ Types of User

a. There are three types of users

i. Administrator (DBA- DataBase Administrator):
    1. Administrator maintains & is responsible for administering a database
    2. DBA looks after the usage of database, creates access profiles for end-users and applies limitations , isolations and enforce security

ii. Designer:
    1. These are the group of people who work on the designing of a database
    2. These designers design the whole database/schema by maintaining relationships, constraints among the different entities

iii. End-User:
    1. These are the users who interact with database to retrieve data or get a view of a database