My SQL :- Structured Query language.

Data :- it is distinct pieces of Information or Collection of facts related to any entity.

Database :- Simply a container where all the data will be Stored.

Database Management System :- A Collection of programs which enables User to access database, manipulate data and represent data.

Types of Database Management Systems (DBMS) :-

In Order to handle the data. we use this DBMS there are Several kinds - are ① Hierarchical DBMS ② Network DBMS ③ Relational DBMS ④ Object-Oriented DBMS.

Hierarchical DBMS :- it is a parent-child relationship & Storing Data. and Structure like a tree with nodes representing record and branches representing fields.

Network DBMS :- it Supports Many-to-Many relations. this results in Complex data base Structure.

Relational DBMS :- it defines database relationship in-form of Tables, also known as relations, row and Column representation & data known as table where Vertical representation known as.

field and horizantal representation is record. Common fields in table represent relation between two tables.

Object-Oriented DBMS :- This type supports storage of new data types, the data stored in form of objects.

Structured Query language (SQL) Tasks :-

i) it is a standard programming language, which is used for managing relational Databases.

ii) with SQL we can modify databases, add, update or delete rows of data, retrive subsets of information from a data base and, any more

iii) Relational Databases like oracle, MySQL, Amazon red sheet etc are used SQL.

iv) Queries and other Sql operations are written as statements

ex: Select, insert, add, update, delete, Create, alter, truncate.

My SQL :- Is a relational Database Management System

* it is an Open-Source Software, which provides multi User access and its supports multi User Engines. and works on many platforms

My SQL Work bench :- it is a visual database designing and modelling access tool for MySQL Server relational database. where we do modeling of database, SQL Development.

SQL Command Categories :-

1) Data Definition Language (DDL)
2) Data Manipulation Language (DML)
3) Data Control Language (DCL)
4) Transaction Control Language (TCL)

1) DDL :- Consist of Commands that can be used to define the database Schema.

Ex : Create, Drop, ALTER, TRUNCATE, COMMENT, RENAME

2) DML :- The Sql Commands that deals with the manipulation of data present in database.

Ex : SELECT, INSERT, UPDATE, DELETE

3) D.C.L :- Include Commands which mainly ~~data~~ deals with the rights Permissions and other Controls of the database System.

Ex : GRANT, INVOKE

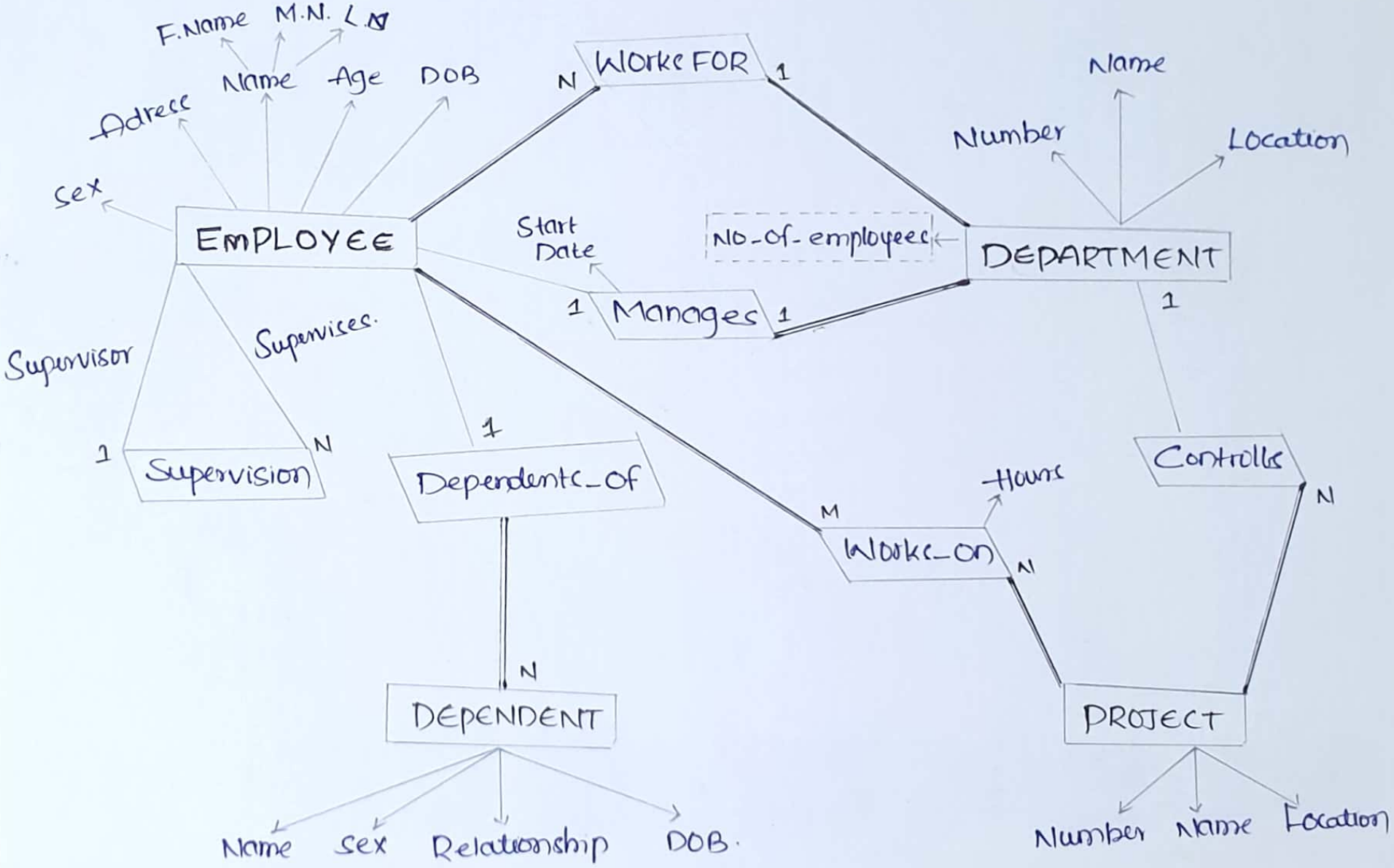
4) T.C.L :- This Commands which mainly deal with the transaction of Database.

Ex :- COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION.

Data Modeling :- it is nothing but designing of Data. where we design by ① Normalize Database

② Entity - Relationship Diagram.

Entity – Relationship (E-R) Diagram :-



i) From the diagram we can say Entities are - Employee, Department, project, Dependendent

ii) There are relationships built in between entity if we observe. in between employee and department, there works-for is one relation and manages is another relation. Similarly in between Department and project controle is relation, inbetween employee and project works-on is relation, finally in between employee and dependent Dependents-of is relationships.

iii) There dependent entity is weak entity because it depends on employee entity (when ever any entity is depends on other entity is called as weak entity) Similarly project entity is also weak entity.

iv) where employee entity is Strong entity the relationship between Strong entity to weak entity is always "Weak Relationship"

v) Similarly if Relationship between two Strong entities or more like employee to Dependtment (or) employee to project is always "Strong relationship"

v) where "Self Relationship" occures when an employee him self employee like, Bose and worker relationship known as Supervision.

vi) finally department Controle project is a Control-type Relationship.

Entity & Its Attributes :-

[E-R] Diagram Consist of many attributes as shown in figure

They are - 1) Composite attributes 2) Simple Attributes

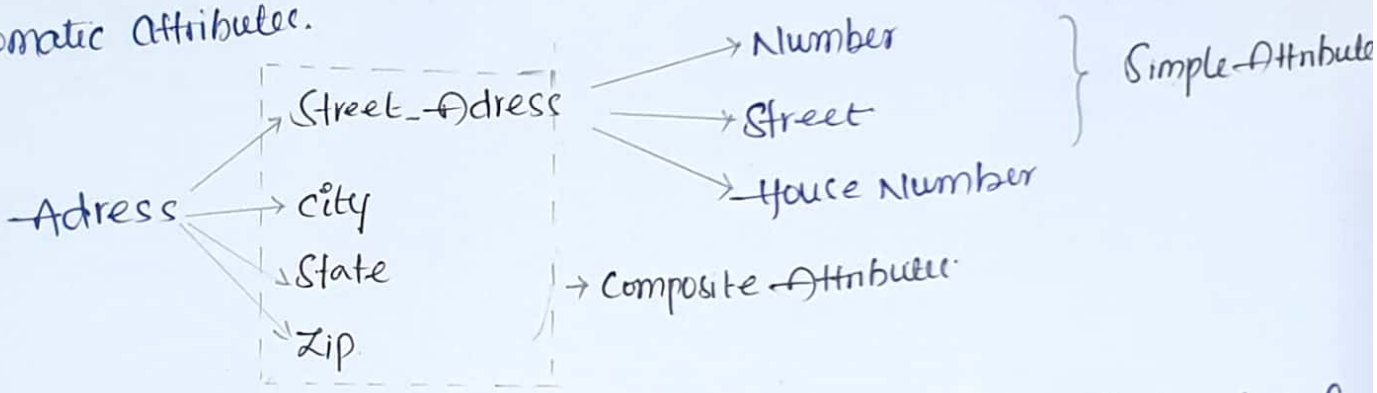
3) Single. 4) multiple valued 5) Stored 6) Derived

7) Complex Attributes.

1) Composite vs Simple Attributes :- A composite attribute can be devided into Smaller Subparts, these Subparts represents individual basic attributes with their own meaning.

Where attributes which is not divisible are Simply known as Simple or automatic Attributes.

Ex



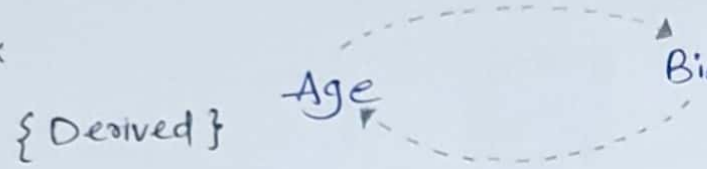
2) Single vs multivalued Attributes :- Attributes having Single value for a particular entity is known as Single-valued attributes.

In multivalued attribute may have lower and upper bounds to Constrain the number of values allowed for each individual entity.

3) Stored vs derived Attributes :- The Stored attributes which are already Stored in the database and from which the value of another attribute is derived is called Stored attribute.

Attributes which are derived from the real entities are known as derived attributes.

Ex



In {E-R} Diagram NO of employees {Represent in dot lines} is one type of Derived attribute.

Complex Attributes :- These attributes represent by Grouping Composite attributes between (), separating the Components with Commas, and by displaying the multivalued attributes between [] are known as Complex Att

Ex : {Adress_phone ({phone (Area_code, phone_Number)},
Adress (Street_adress {Number, Street, flat_Number), City, State, Zip))}

Entity Types, Sets, keys & Value Sets :-

Enty →	Employee	Company
Entity type →	Name, Age, Salary	Name, Head Quaters, Owner
Entity Sets →	e1 Ameer, 25, 50K e2 Sameer, 24, 45K	c1 Amazon, Newyork, Abdul c2 microsoft, India, kanav

* Values entered in entity sets are called as value sets in every record there is diffrent values are entered.

Different Types of Keys in Relational Database :-

1) Candidate key
2) Super key
3) Primary key
4) Alternate key
5) Foreign key

Candidate key :- is basically a minimal set of attributes which can uniquely identify tuple and not null value present in it. Sometimes Candidate key also a primary key.

Super key :- Super key also a one kind of candidate key where a minimal set of attributes which can uniquely identify a tuple together.

Primary key :- it is also a set of attributes which can be used to uniquely identify every tuple.

There can be more than one candidate key in a relation out of which one can be chosen as primary key.

Alternate key :- the candidate key other than primary key is called as alternate key.

Foreign key :- if an attribute can only take the values which are present as values of some other attribute, it will basically be the foreign key to the attribute which it refers.

Refrenced attribute & refrencing attribute should be a primary key.

EMPLOYEE TABLE (T-1)

EMP-ID	EMP_NAME	PHONE	STATE	COUNTRY	AGE
1	AMEER	8319054	Telangana	India	28
2	AMEER	7654120	Andhra	India	25
3	SAMEER	8934651	Delhi	India	26
4	SAMEENA	9848501	Maharastra	India	21

EMPLOYEE DEPT (T-2)

EMP-ID	DEPT-NO	DEPT-NAME
1	D1	TECH
2	D2	MARKETING
1	D2	MARKETING

from Above two tables (T1, T2) We can Say (EMP-ID), EMP-NAME, PHONE NO having Uniquely devided whole tuple So Individually this columns we can Say Candidate key.

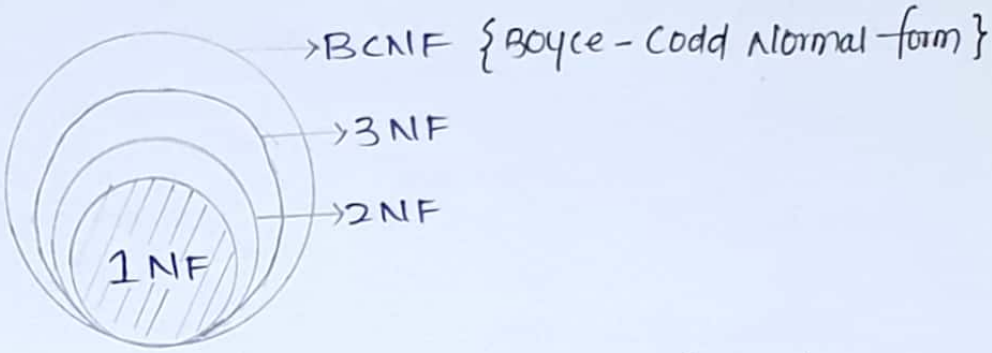
from above mentioned Columns of table-1 (Candidate key) represent together Columns which devide whole tuple is called as Super key.

Similarly the Columns which we mentioned as Candidate key except that the Column which have no null values present and Uniquely tells the whole tuple is called as primary key.

Other than this keys for Suppose if we choose EMP-ID as primary key, EMP-NAME as Candidate key, PHONE-NO remaining represent alternate key which is Uniquely identified and Similarly devide whole tuple.

finally if we make relation in between Table 1 and 2 by a key is called as foreign key and that should be a primary key, from the above tables (Emp_ID)'s are primary keys in both tables and make relation between them by refrencing numbers and this represent whole record in table 1.

Normalization :- It is a technique that Organizes tables in such a way that redundency and dependency of data is reduced.



Here we just modeling the data by normalization forms there are four types of forms called as (1NF, 2NF, 3NF, BCNF) where data will be normalized at any one of form Initially at 1NF whole data will be normalized. if not goes to final form BCNF. Here we just Increasing the level of normal forms. Such that Organize a table in a well manner.

SQL Attribute Data Types :- classified as -

1) Numeric - INT, FLOAT, REAL, DECIMAL(i,j)

2) Character - String - CHAR(n), VARCHAR(n), CLOB

3) BIT-STRING – BIT(n), VARYING(n)

4) Boolean – TRUE, FALSE, Unknown

5) Date-format – Date & Time

6) Timestamp.

Constraints :– In Simple terms this are the rules that Can be applied on the type of data in a table.

NOT NULL – null values Can't be Stored in columns.

UNIQUE – All the values in a Column are different

CHECK – All the values in a Column Satisfy a Special Condition

DEFAULT – Set of default values for Column when no value is Specified

INDEX – this Constrain Used to Create and retrive data from the Database very quickly.

SQL – Commands :–

CREATE INDEX : Create an Index for a table

ALTER TABLE : Modify a Table (add, modifies, or deletes attributes)

DROP TABLE : permanently delete a Table

CREATE SCHEME : Create Database Schema.

AUTHORIZATION

CREATE TABLE : Create a new table in User's Database Schema.

NOT NULL : Column will not have a null values.

UNIQUE : Column will not have a duplicate values

PRIMARY KEY : Defines a primary key for a Table

FOREIN KEY : Defines a foreign key for a Table

DEFAULT : Defines a default value for a Column

CHECK : Constraint used to validate data in a attribute.

DROP VIEW : Perminently deletes a view

DROP INDEX : Perminently deletes a Index

CREATE VIEW : Create dynamic Subset of rows/columns from one or more Tables.

CREATE TABLE AS : Create new table based on query in the users data-base Schema.

SQL - Data Manipulation Commands :

UPDATE : Modify an attributes values in One or more tables rows

COMMIT : Perminently saves data changes.

INSERT : Insert row(s) in to table

SELECT : Select attributes from rows in One or more tables.

WHERE : Restricts the selction of rows based on a Conditional Expression.

Group By : Groups the Selected rows based on One or more attributes

HAVING : Restrict the selection of grouped rows based on a. Condition.

ORDER BY : Orders the Selected rows based on One or more attributes

DELETE : Delete One or more rows from a table

ROLLBACK : Restore data to their Original values.

Logical Operators : AND | OR | NOT

Comparision Operators : >, <, =, >=, <=, <>

Aggregate Functions : COUNT, MINT, MAX, SUM, AVG

Special Operators : BETWEEN, ISNULL, LIKE, IN, EXISTS, DISTINCT

BETWEEN : checks whether an attribute value is with in a range.

LIKE : Checks whether an attribute value maches given string pattern

IN : Checks " " " " any value with in a value list.

EXISTS : Check whether a Subquery returns any row.

DISTINCT : Limits values to unique values.

Relational Algebra & Relational Calculus :

Unary Relational Operators :-

1) Select Operation 2) Project operation 3) Rename operation

Select operation : it is used to choose a subset of the tuples from a relation that satisfy a selection condition.

Project Operation : Select Certain Column from the table and discards the other columns.

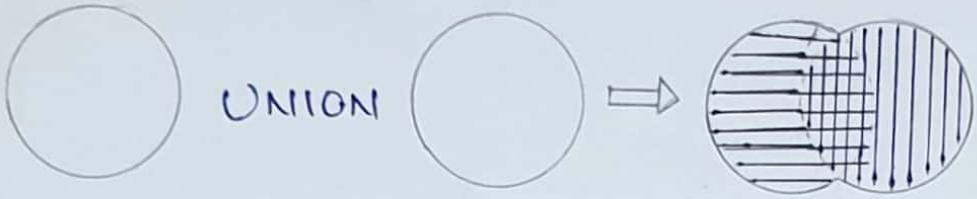
Rename operation : is a Unary operation used for renaming attributes of a relation.

Set Operations :- We have different kind of Setoperations are presents. They are — ① Union Operation

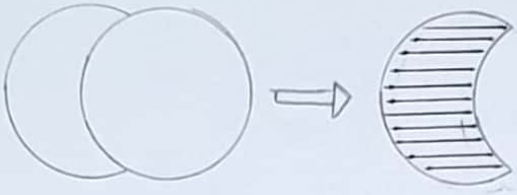
② Intersect Operation ③ Minus (or) Except Operation.

④ Cattesian (Cross product) Operation.

Left Query Right Query Final Result

1) UNION  Combine rows from both the Queries

2) Intersect  Keeps only those rows which are common in both the Queries

3) MINUS  Keeps rows from the left Query which are not included in the Right Query.

Cross product Operation :-

Cross product between 2 Relations (A & B) will result all the attributes of 'A' followed by each attribute of 'B'. each record of 'A' will be pair with every record of 'B'.

Ex:

A

Name	Age	Sex	ID	Dept
Ameer	26	M	1	Tech
Raju	24	M	2	Education
Sunny	27	F		

B

⇒

A × B

Name	Age	Sex	ID	Dept
Ameer	26	M	1	Tech
Ameer	26	M	2	Edu
Raju	24	M	1	Tech
Raju	24	M	2	Edu
Sunny	27	F	1	Tech
Sunny	27	F	2	Edu