9/2/24

EF codd talked about RDBMS in 1970 first.
For a database to be a RDBMS he gave 12 rules

Codd's Rule =>

Rule 1 => Information representation -> Should be able to represent all types of information.

Rule 2 -> Guranteed access -> You should be able to see/get data stored in DB at all time.

Rule 3 -> Systematic Treatment of Null Values.

Rule 4 -> Database Description Rule => data/info/ description of database should be there

Rule 5 -> Comprehensive Sublanguage

Rule 6 -> View / Updating

Rule -7 => High level insert, update, delete

Rule -8 => Physical data Independence

Rule - 9 => Logical data independence

(10) The distribution Rule => If necessory db should be able to be distributed over.

Rule 11 -> Non - Subversion

Rule 12 => Integrety Rules

12/2/24

SQL

In 1970 ⇒ Edgar F. Codd gave Relational model of Database., in IBM

Father of RDBMS = E.F. Codd.

In 1974 ⇒ ~~S~~ SQL appeared

In 1978 ⇒ IBM made System/R first RDBMS.

In 1986 ⇒ First RDBMS was released, Later its was named "Oracle"

SQL = Structured Query Language.

- 1986 = first version = SQL86

1989 =	SQL89
1992 =	SQL92.
1999 =	SQL99
2003	= SQL:2003
2006	= SQL:2006
2008	= ~~2~~SQL:2008
2011	= SQL:2011
2016	= SQL:2016
2019	= SQL:2019

Components of SQL ⇒

⇒ DDL = Data Definition Language

⇒ DML = Data Manipulation Language

⇒ Embedded SQL

⇒ Transaction processing. (TCL) {Transaction Control Language}

⇒ Authorization.

Database Languages ⇒

⇒ Procedural Language → What & How

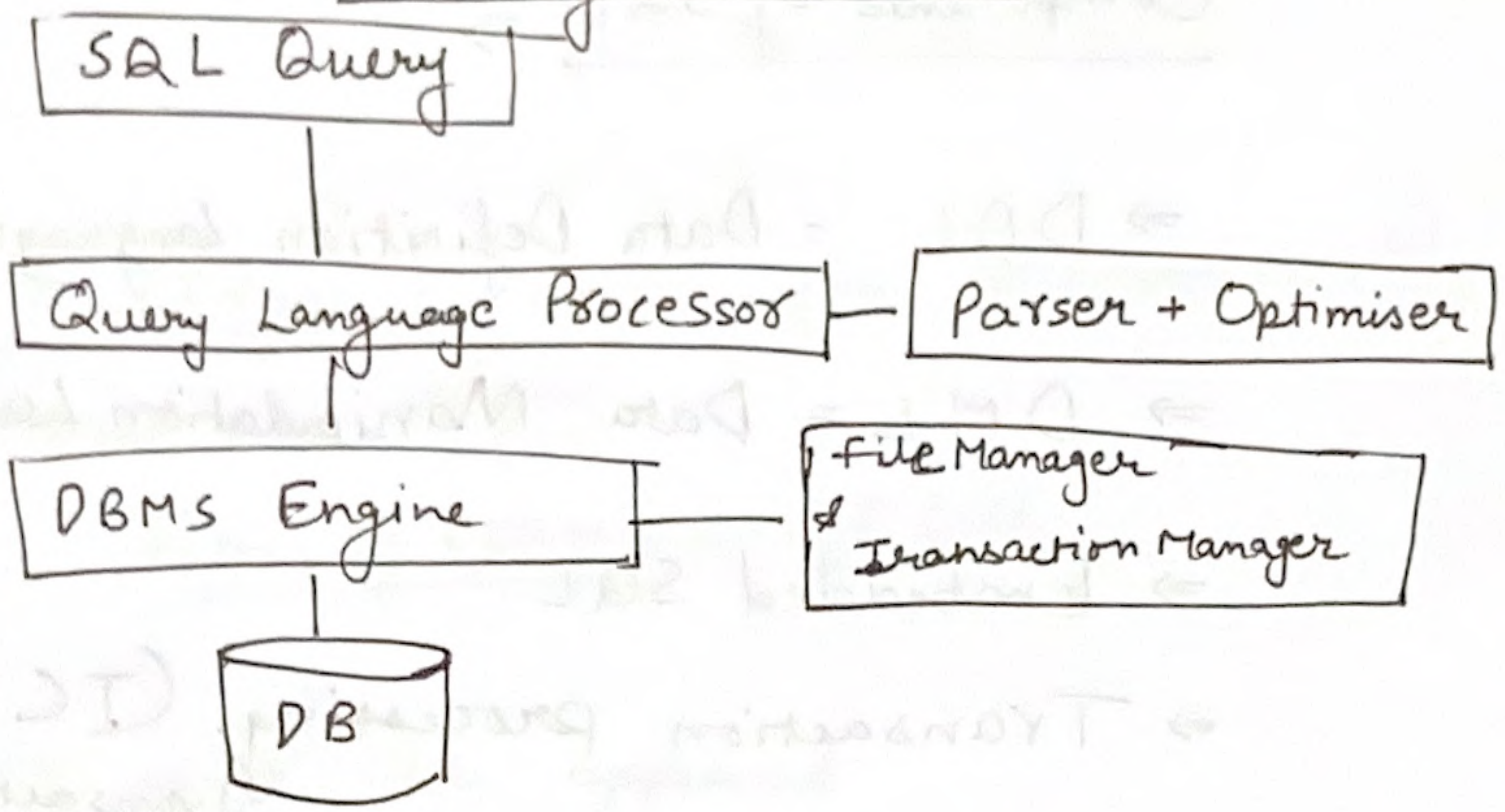
eg ⇒ Relational Algebra

⇒ Non-Procedural Language ⇒ What

eg ⇒ - SQL

- Tuple relational Calculus.
- Domain Relational Calculus.

Query Process



Rule for Data Integrity :->

① Entity Integrity ⇒ No duplicate Rows in a Table.

⇑

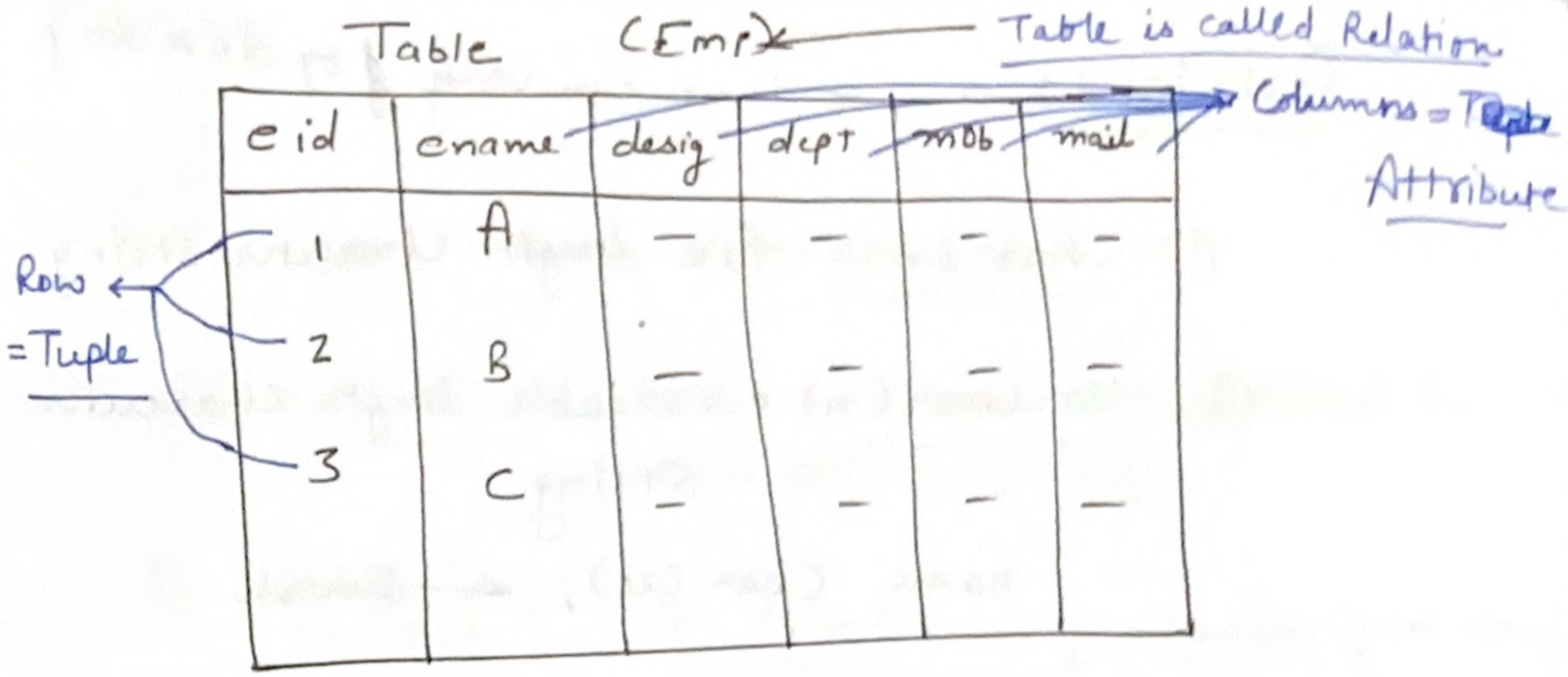
{Unique primary key & has no null values}

② Domain Integrity ⇒ Valid values should be enforced.

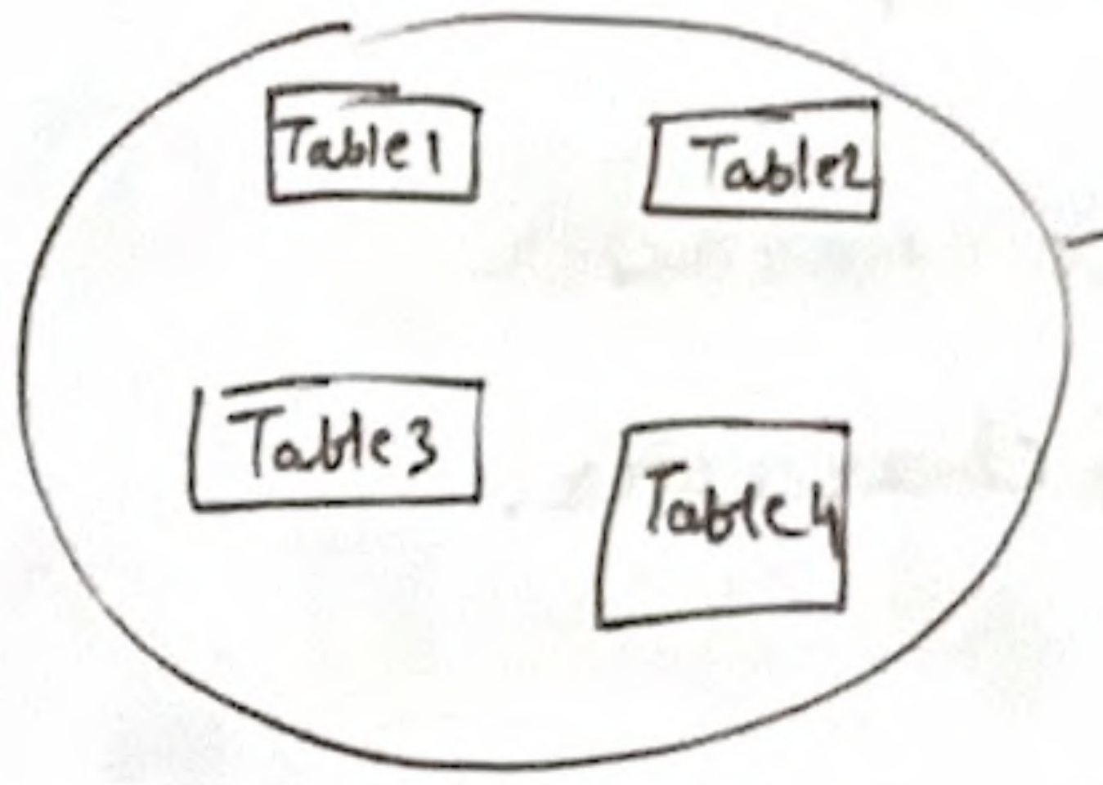
③ Referential Integrity → There should be valid value to the table referenced.

④ User Defined Integrity ⇒ rules that user will define according to their requirement.

Table (Emp) — Table is called Relation



eid	ename	desig	dept	mob	mail
1	A	—	—	—	—
2	B	—	—	—	—
3	C	—	—	—	—



Collection of related table is our database

Commands for Database

(1) ⇒ CREATE database d_name ;

(2.) ⇒ USE database d_name;

(3) ⇒ show databases;

(4) ⇒ Drop database d_name;

Datatypes => {These can vary ~~f~~ by db to db}

① Char (n) = fix length Character string

② VarChar (n) = Variable length character String.

name Char (20); ← Suresh

f_name VarChar (20); Ramesh

name => size 20 character
f_name = size 6 character.

③ INT

④ SMALLINT

⑤ Boolean

⑥ REAL

⑦ DOUBLE

⑧ FLOAT

⑨ NUMERIC(P, d)

⑩ DATE

⑪ TIME

⑫ TIMESTAMP

14/2/24

SQL Constraints {Sort of Rules}

① NOT NULL => can be assigned to the attributes that we don't want Null.

② DEFAULT => for default value of attribute. if ~~we~~ nothing is specified.

③ UNIQUE => Ensures values of attribute is different.

④ PRIMARY Key

⑤ fOREIGN KEY

⑥ CHECK => ~~Te~~ ensures all values in a Column satisfy certain Conditions.

⑦ INDEX => used to create & retrieve data from the database very quickly

Commands for Table

① Creation => eg =>

```
CREATE  TABLE  student
   RollNo     Char(8)       NOT NULL,
   Name       VarChar(20)   NOT NULL,
   Sex        Char(1)       default = 'M',
    DeptNo.     Small INT
     PRIMARY Key  (RollNo)
     fOREIN key  ( DeptNo) Refrences Department
```

⇒ (DeptId) ON DELETE SET NULL, ON UPDATE CASCADE;

Emp

eid	ename	desig	dept.no.
1	A	IA	101
2	B	IA	102
3	C	AP	103
4	D	Prog	101
5	E	Prog	102
6	F	ACP	101

Dept

did	dname
101	DoIT
102	CMO
103	Agri
104	Home.

Q = In what table we will face any Problem on addition of new Record

Ans ⇒ Employee, if we add new record in employee where dept no. is not in Dept eg (105)

Q ⇒ On deletion of a record in which table we will face problem.

Ans ⇒ Dept, we might have values referenced in Emp.

Q ⇒ If we remove record (102, CMO) from Dept what will happen.

Ans ⇒ One ~~Conditi~~ solution can be that we can Seat dept.no. a default value, so that in Case of any deleation in dept, we will have no affect on Emp.

Or we can set it Null.

Or it can Cascade

↓ meaning deletion in dept will also delete all rows of emp getting referenced from that record

i.e. eid = 2, 5 record will be deleated.

Q ⇒ If we modify did in Dept which is referenced in Emp table as well.

Ans ⇒ ① In Emp table respective values can be set to default

② or set to Null

③ or Cascade

↓ change done in Dept should also be done in Emp.

CREATE table => (Syntax)

CREATE table r (atr1 definition, attr 2 definition ---- attrn definition n, [integrity constraint 1], [integrity constraint 2], --- [integrity constraint m])

=> Attribute definition =>

=> Attribute name domain type [integrity constraints]

16/2/24

② INSERT

=> insert into table_name Values (V1, V2, V3..);

Emp

eid	ename	desig
1	A	IA
2	B	Prg

=> insert into Emp values (1, 'A', 'IA');

=> insert into Emp (ename, eid, desig) Values ('B', 2, 'Prg');

③ ALTER

ALTER TABLE {To change Schema of the table}

(i) addition of column

=> ALTER TABLE Emp add column location varchar(20)

⇓

Emp

eid	ename	desig	location

(ii) drop of a column

=> ALTER TABLE Emp drop column location;

(iii) Change constraints of a column =>

① eg=> Let's say desig have a default value in emp and we want to remove

=> ALTER TABLE Emp alter column desig drop default

② eg=> Let's say we want to add new ~~ch~~ constraints to desig.

=> ALTER TABLE Emp alter column desig set default 'IA';

4 DROP

=> To Remove structure of the table from database.

=> DROP TABLE Emp;

{ If anothere table is refering to Emp then, we Cannot ~~delete~~ drop Emp directly }

⇓

In this case we can do two things :->

1. Cascade => If we delete Emp releated table is also get deleted.
2. restrict => we would prevent the deletion of Emp. as other table is refering to it.

Q => we want to increase Basic of every employee by 5000

Ans-> Update table Emp set salary = salary + 5000;

Emp

desig	eid	ename	salary	da
IA	1	A	25000	50
AP	2	B	30000	40
Prg	3	C	28000	50
IA	4	D	35000	35
AP	5	E	32000	32

Q => Basic + 5000 & dA + 5%

Ans-> Update table Emp set Salary = Salary + 5000, dA = dA + 5;

Q => ~~update~~ Basic + 5000 of IA only.

=> Update table Emp set salary = salary + 5000 where desig = 'IA';

Q => Basic + 5000, DA + 5% of IA & AP.

=> Update Table Emp set Salary = Salary + 5000, ~~where~~ da = da + 5 where desig = 'IA' OR design = 'AP';

5 Delete

=> delete from emp;

↳ will all records of emp;

=> delete from emp where desig = 'TD';

↳ will delete all records with designation TD.

difference b/w delete & truncate

⑥ SELECT { To get data }

⇒ Select < column names > ← Projection

from < table name >

where < condition ← Selection

⇒ select * from Emp

↳ get all records from Emp table.

⇒ Select ename,design from emp

↳ list of all ename, design from emp.

⇒ Select distinct ename, desig from emp

↳ duplicate records will be removed from result.

⇒ select * from emp where dept = 'Health';

↳ get all employee from emp where department is health.