Unit:III
Lecture: 1
Relational Model

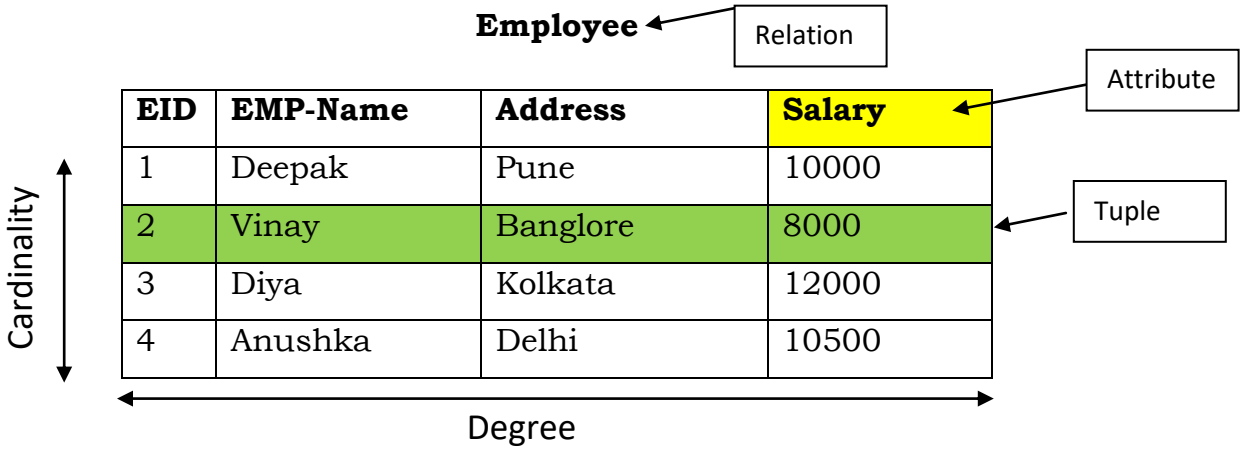
Relational Model was proposed by E.F. Codd to model data in the form of relations or tables.
After designing the conceptual model of Database using ER diagram, we need to convert the conceptual model in the relational model which can be implemented using any RDBMS languages like Oracle SQL, MySQL etc.

The relational model is an abstract theory of data that is based on the mathematical concept of relations. In relational database, information is stored in tabular form or in tables. So, it consists of collection of tables.

The relational model is concerned with three aspects of data: data structure, data integrity and data manipulation.

- Data structure: tables
- Data Integrity: primary key rule, foreign key rule
- Data manipulation: Relation algebra and Relational Calculus

Relational Model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables).



EID	EMP-Name	Address	Salary
1	Deepak	Pune	10000
2	Vinay	Banglore	8000
3	Diya	Kolkata	12000
4	Anushka	Delhi	10500

Fig. Basic data structures of relational database

A table is a matrix of a series of row and column intersections.

Terminology used in Relational Model:

- **Relation:** In relational model name of table is known as relation. It consists of all possible tuples.
- **Tuple:** Single row of any relation is known as tuple.
- **Attributes:** These are the characteristics of any relation e.g. EID, EMP-Name, Address, Salary.

- **Domain:** is the set of all permitted values or information about any attribute e.g. in relation Employee domain for attribute Emp-Name is: (Deepak, Vinay, Diya, Anushka).
- **Degree:** Number of columns in any relation is known as its degree.
- **Cardinality:** Number of rows in any relation is known as its cardinality.

Relational Model Notation:

- A relation schema R of degree n is denoted by $R(A_1, A_2, \ldots, A_n)$.
- An n-tuple t in a relation $r(R)$ is denoted by $t = \langle v_1, v_2, \ldots, v_n \rangle$, where v_i is the value corresponding to attribute A_i. The following notation refers to **component values** of tuples:
  - Both $t[A_i]$ and $t.A_i$ (and sometimes $t[i]$) refer to the value v_i in t for attribute A_i.
  - Both $t[A_u, A_w, \ldots, A_z]$ and $t.(A_u, A_w, \ldots, A_z)$, where $A_u, A_w, \ldots, A_z$ is a list of attributes from R, refer to the subtuple of values $\langle v_u, v_w, \ldots, v_z \rangle$ from t corresponding to the attributes specified in the list.
- The letters Q, R, S denote relation names.
- The letters q, r, s denote relation states.
- The letters t, u, v denote tuples.
- In general, the name of a relation schema such as STUDENT *also indicates* the current set of tuples in that relation—the *current relation state*—whereas STUDENT(Name, SSN, . . .) refers *only* to the relation schema.
- An attribute A can be qualified with the relation name R to which it belongs by using the *dot notation* $R.A$—for example, STUDENT.Name or STUDENT.Age. This is because the same name may be used for two attributes in different relations. However, all attribute names *in a particular relation* must be distinct.

Advantages of Relational Model:

- Simplicity: The main advantage of relational model is the simplicity in representing data than network and hierarchical model.
- Easy to design, implementation, maintenance and use: Relational model provides physical, logical, integrity and distribution data independence which makes the whole database design cycle easier.
- It offers flexible and powerful query capability.
- It is structural independent. Any changes in database structure do not affect the data access.

Disadvantages:

- It needs more powerful computing hardware and data storage devices that increase the cost and hardware overhead.
- Ease to design can lead to a bad design.