Module No.3

Relational Model and relational Algebra

Introduction to the Relational Model, relational schema and concept of keys. Mapping the ER and EER Model to the Relational Model, Relational Algebra-operators, Relational Algebra Queries.

Relational Model:

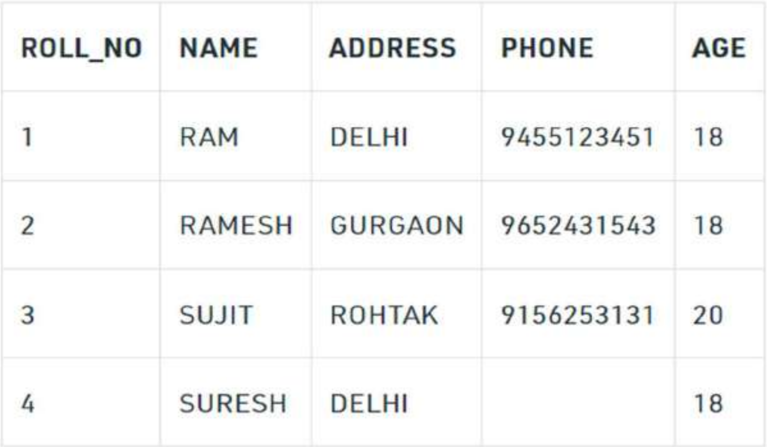
The relational Model was proposed by E.F. Codd to model data in the form of relations or tables. After designing the conceptual model of the Database using ER diagram, we need to convert the conceptual model into a relational model which can be implemented using any RDBMS language like Oracle SQL, MySQL, etc. So we will see what the Relational Model is. Relational Model (RM) represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.

What is the Relational Model?

The relational model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables). Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE, and AGE shown in Table 1.

STUDENT



ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

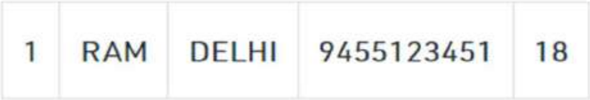
Relational Model Concepts in DBMS

Attribute: Attributes are the properties that define a relation. e.g.; ROLL_NO, NAME

- Each attribute of a relation has a name
- The set of allowed values for each attribute is called the domain of the attribute
- Attribute values are (normally) required to be atomic; that is, indivisible
- Note: multivalued attribute values are not atomic
- Note: composite attribute values are not atomic
- The special value null is a member of every domain
- The null value causes complications in the definition of many operations

Relation Schema: A relation schema represents the name of the relation with its attributes. e.g.; STUDENT (ROLL_NO, NAME, ADDRESS, PHONE, and AGE) is the relation schema for STUDENT. If a schema has more than 1 relation, it is called Relational Schema.

Tuple: Each row in the relation is known as a tuple. The above relation contains 4 tuples, one of which is shown as:



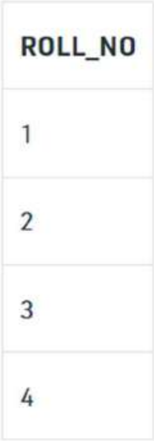
1	RAM	DELHI	9455123451	18

Relation Instance: The set of tuples of a relation at a particular instance of time is called a relation instance. Table 1 shows the relation instance of STUDENT at a particular time. It can change whenever there is an insertion, deletion, or update in the database.

Degree: The number of attributes in the relation is known as the degree of the relation. The STUDENT relation defined above has degree 5.

Cardinality: The number of tuples in a relation is known as cardinality. The STUDENT relation defined above has cardinality 4.

Column: The column represents the set of values for a particular attribute. The column ROLL_NO is extracted from the relation STUDENT.



ROLL_NO
1
2
3
4

NULL Values: The value which is not known or unavailable is called a NULL value. It is represented by blank space. e.g.; PHONE of STUDENT having ROLL_NO 4 is NULL.

Relation key – Every row has one, two or multiple attributes, which is called relation key.

Attribute domain – Every attribute has some pre-defined value and scope which is known as attribute domain

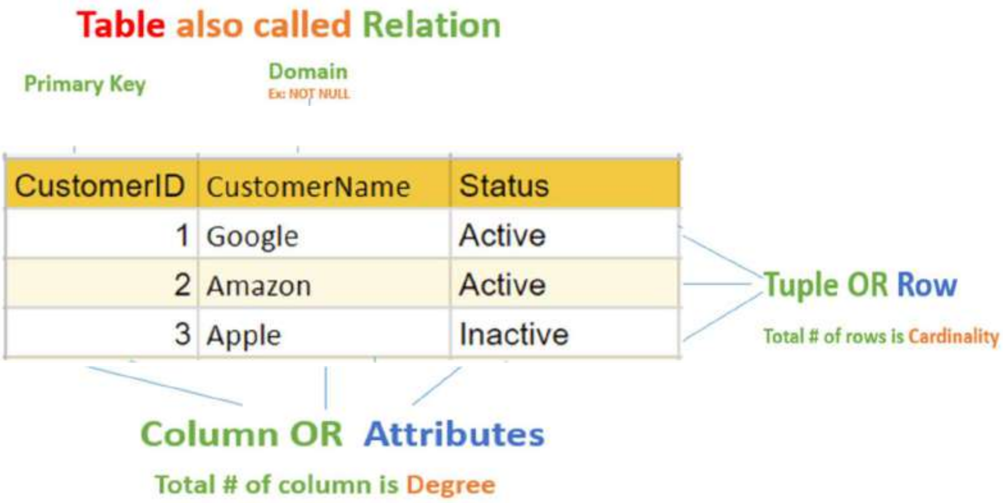


Table also called Relation
Primary Key
Domain
Ex: NOT NULL
Tuple OR Row
Total # of rows is Cardinality
Column OR Attributes
Total # of column is Degree

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Properties of Relations

- Name of the relation is distinct from all other relations.
- Each relation cell contains exactly one atomic (single) value
- Each attribute contains a distinct name
- Attribute domain has no significance
- tuple has no duplicate value

- Order of tuple can have a different sequence

Advantages of Relational Database Model

- Simplicity: A Relational data model in DBMS is simpler than the hierarchical and network model.
- Structural Independence: The relational database is only concerned with data and not with a structure. This can improve the performance of the model.
- Easy to use: The Relational model in DBMS is easy as tables consisting of rows and columns are quite natural and simple to understand
- Query capability: It makes possible for a high-level query language like SQL to avoid complex database navigation.
- Data independence: The Structure of Relational database can be changed without having to change any application.
- Scalable: Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability.

Disadvantages of Relational Model

- Few relational databases have limits on field lengths which can’t be exceeded.
- Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.
- Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.

Query Language:

Query is a question or requesting information. Query language is a language which is used to retrieve information from a database.

Query language is divided into two types −

- Procedural language
- Non-procedural language

Procedural language

Information is retrieved from the database by specifying the sequence of operations to be performed.

For Example − Relational algebra.

Structure Query language (SQL) is based on relational algebra.

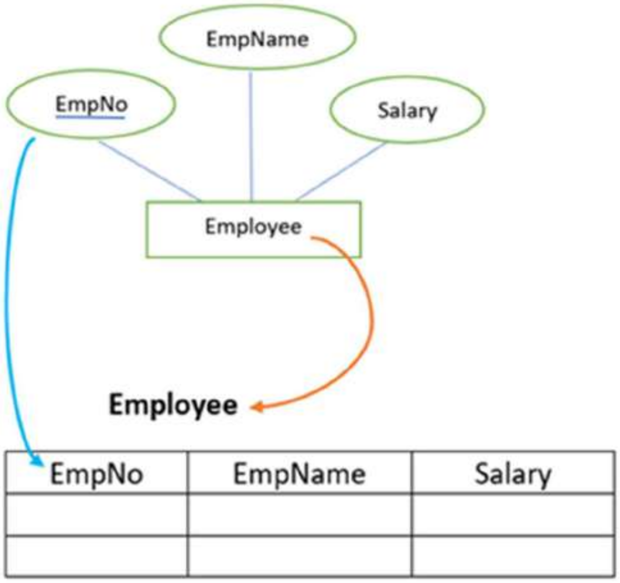
Relational algebra consists of a set of operations that take one or two relations as an input and produces a new relation as output.

Mapping of E-R Diagram into Relational Model
conversion of E-R diagram into a relational model involves the following:

1. Mapping of an entity set into relation (tables) of the database.
2. The attributes of a table include the attributes of an entity
3. The key attribute of an entity becomes the primary key of the relation

For example,

Consider the following E-R diagram in the figure below. The E-R diagram consists of *Employee* as an entity set and *EmpNo*, *EmpName*, and *Salary* as its attributes. Here we map entity set into a relation *Employee* and attributes of an entity set will become the attributes inside the table. The key attribute will become the primary key of the table.



Employee

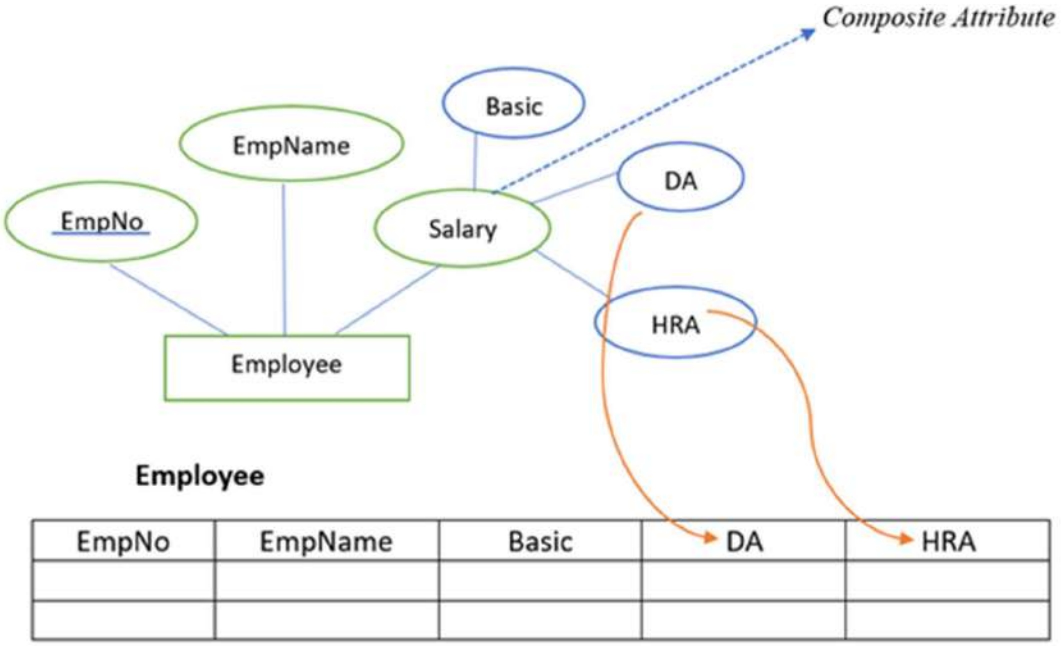
EmpNo	EmpName	Salary

Conversion of a simple E-R diagram into a Table

The conversion of an E-R diagram into the relational model also depends on the type of components used in the E-R diagram. The following section describes the different cases of the E-R diagram and their conversion into their corresponding tables.

Entity set with a composite attribute

While converting an E-R diagram consisting of a composite attribute we do not include the composite attribute in the relational model. The compositions of the composite attribute will become the attributes in the relational model. For example in the figure given below, *Salary* is the composite attribute, and *Basic*, *DA*, and *HRA* are its compositions.



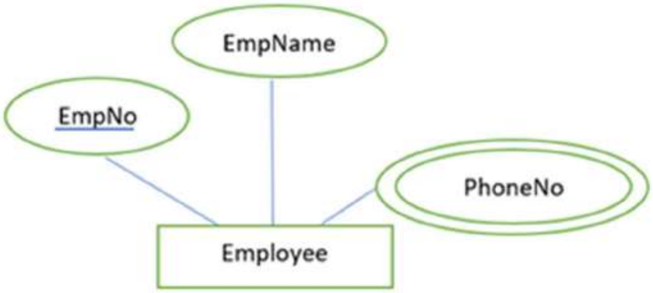
Employee

EmpNo	EmpName	Basic	DA	HRA

Conversion of an E-R diagram containing composite attribute

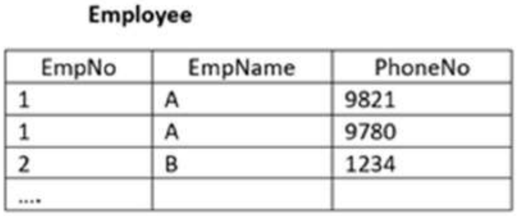
Entity set with multivalued attributes

If an entity contains a multivalued attribute, we split the attributes into two relations in the relational model. One with key attribute and all simple attributes and other with key attribute and all multivalued attributes. For example, in the figure given below, *PhoneNo* is the multivalued attribute.



E-R diagram containing multivalued attribute

If we include the PhoneNo in the table with all other attributes, then for a single-valued tuple we may have multiple entries as shown in the table below

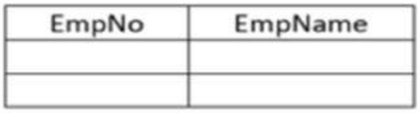


Employee

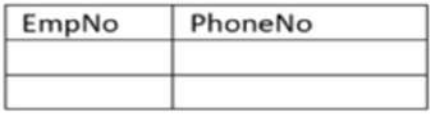
EmpNo	EmpName	PhoneNo
1	A	9821
1	A	9780
2	B	1234
....		

Duplicate values with multivalued attribute

However, to avoid duplicate values in the table, we split the attributes into two different relations as shown in the figure below



EmpNo	EmpName



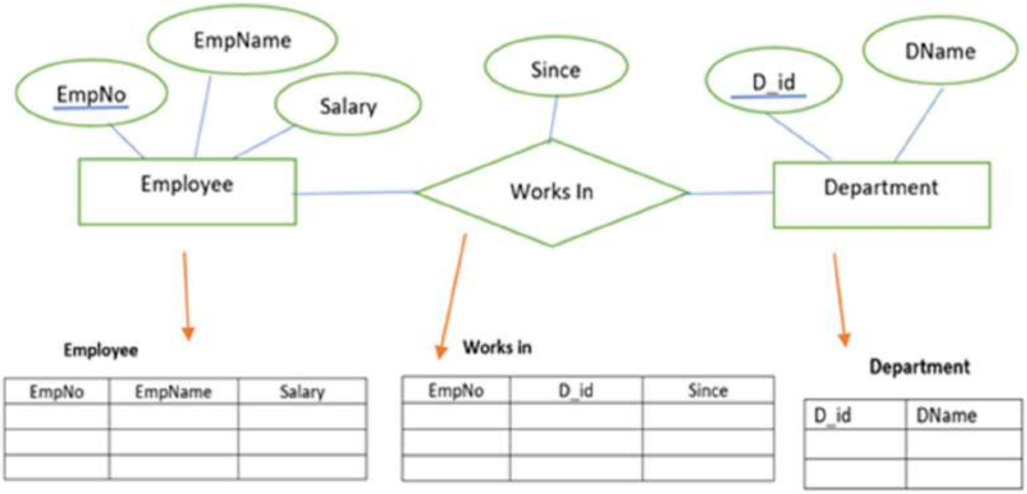
EmpNo	PhoneNo

Conversion of multivalued attributes into relation

Translation of a relationship into a relation

Likewise, we map the entity set into the relation in a relational model, we can also map a relationship set into a relation. The attribute of such a relation includes key attributes of the participating relations. The attributes are will become a foreign key.

For example, in the figure given below, there are two entity sets *Employee* and *Department*. These entity sets are participating in a relationship *works in.* The relationship set is converted into relation with attributes EmpNo from *Employee* relation, D_id from *Department* relation and *Since*, the attribute of the relationship set itself.



Employee

EmpNo	EmpName	Salary

Works in

EmpNo	D_id	Since

Department

D_id	DName

Translation of a relationship into a relation