



**RELATIONAL
DATA BASE
MANAGEMENT
SYSTEM(RDBMS)**

COMPUTER AWARENESS

EPIISODE-13



Computer Awareness

Part 13

- Funsta Team





Computer Awareness



- Part 1 Intro/Generation/ Classification of Computers
- Part 2 Computer Architecture & Memory
- Part 3 Computer Hardware
- Part 4 Computer Software and System Utilities
- Part 5 Number System
- Part 6 Computer Codes & Logic Gates





Computer Awareness



- Part 7 Introduction to Operating System
- Part 8 Operating System
- Part 9 Data Communication
- Part 10 Computer Networks & Network Topology
- Part 11 OSI Layers & Network
- Part 12 Database Management System (DBMS)

Lets move on to
Next Part



Sl. No	Topic	Page Number
1	Relational Database	5
2	Terms Related to Database	6
3	Keys	11
4	Types of Keys	12
5	Database Languages	17
6	Entry Relationship Model (E-R Model)	21



Relational Database



A **relational database** is a type of **database** that stores and provides access to data points that are related to one another.



The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.



e.g.: Oracle, IBM DB2 and Microsoft SQL Server.

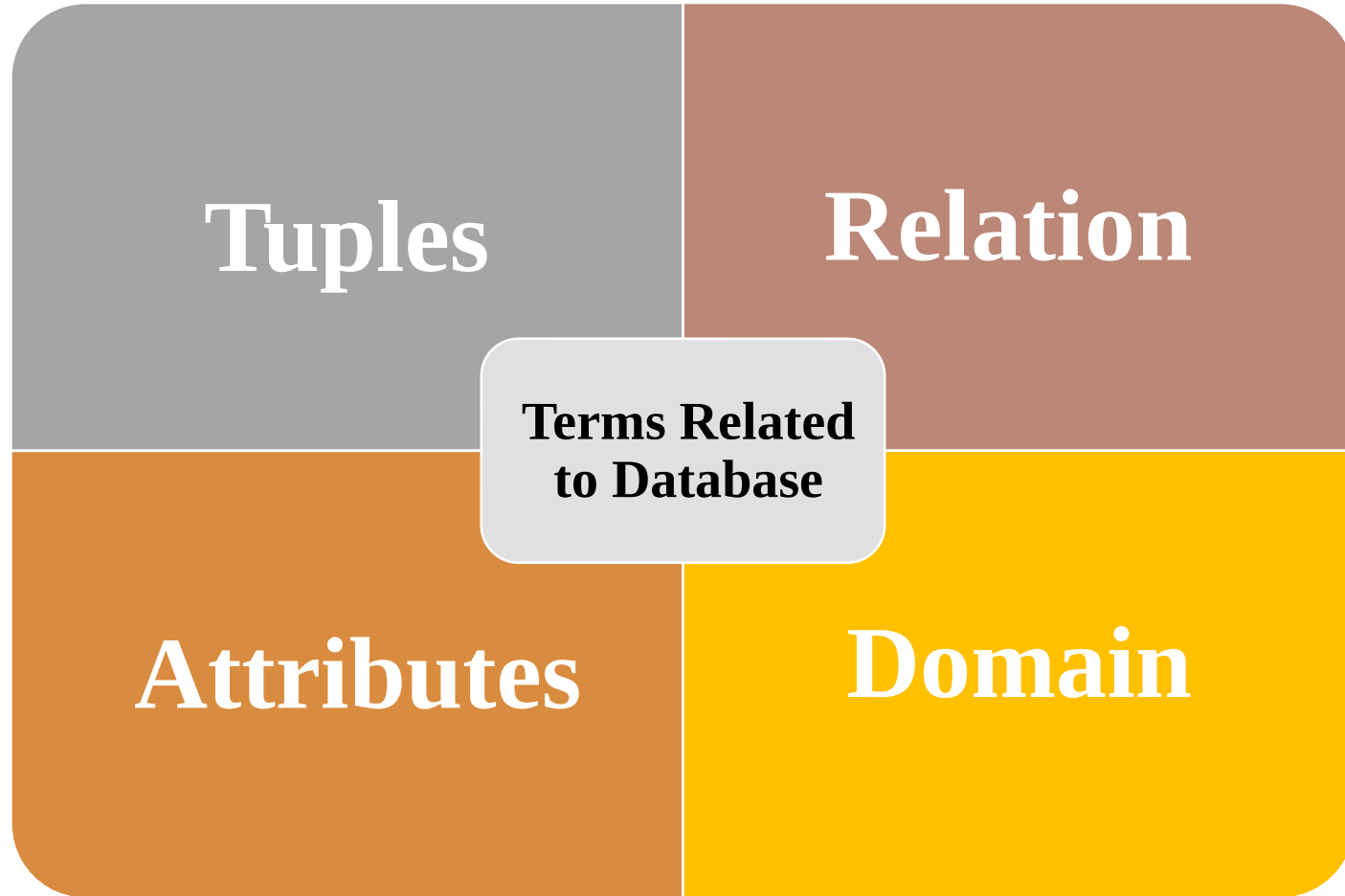


Relational Database





Terms Related to Database





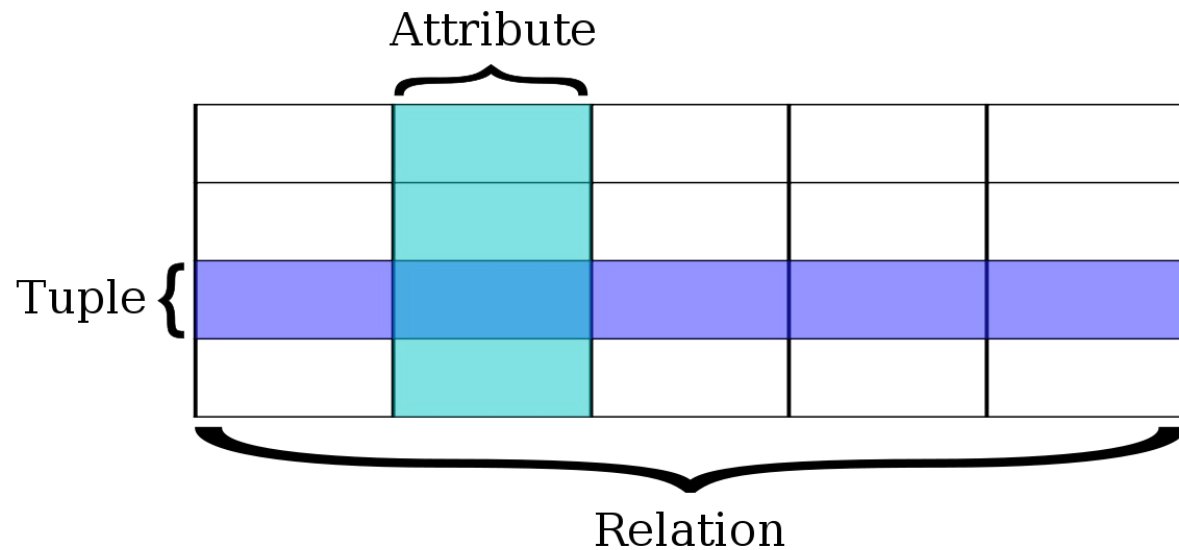
Relation



The term n-tuple refers to a tuple of degree n ($n \geq 0$).



In SQL, a **database** language for relational **databases**, **relations** are represented by tables, where each row of a table represents a single tuple, and where the values of each attribute form a column.



Back to
Terms Related to
Data Base

Domain



In data management and **database** analysis, a data **domain** is the collection of values that a data element may contain.



Reference tables are formally **related** to other tables in a **database** by the use of foreign keys.

[Back to
Terms Related to
Data Base](#)

Attributes



- ◀⋯▶ In general, an **attribute** is a characteristic.
- ◀⋯▶ In a **database** management system (DBMS), an **attribute** refers to a **database** component, such as a table.
- ◀⋯▶ It also may refer to a **database** field.
- ◀⋯▶ **Attributes** describe the instances in the column of a **database**

[Back to
Terms Related to
Data Base](#)

Tuples



Tuple – A single row of a table, which contains a single record for that relation is called a **tuple**. Relation instance – A finite set of **tuples** in the relational **database system** represents relation instance.

[Back to
Terms Related to
Data Base](#)

Keys



Keys are very important part of Relational **database** model.



They are used to establish and identify relationships between tables and also to uniquely identify any record or row of data inside a table.

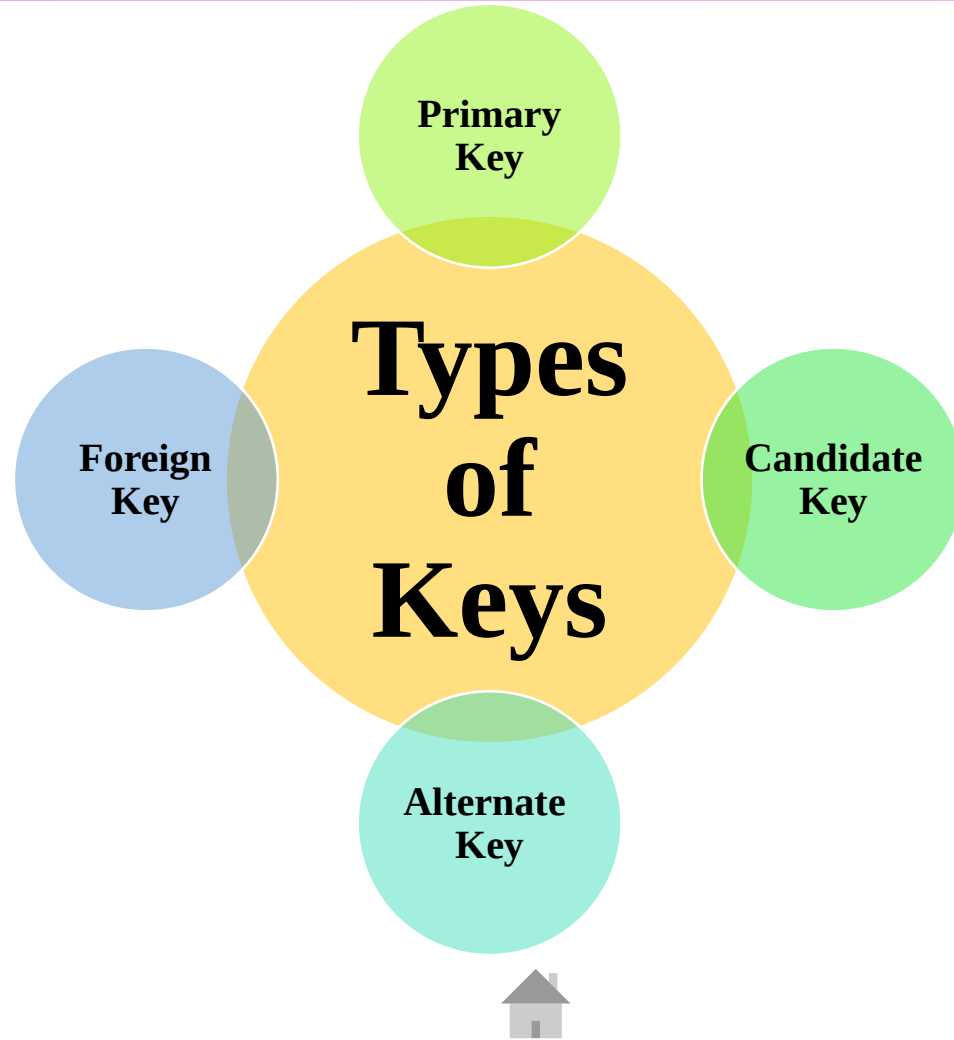


A **Key** can be a single attribute or a group of attributes, where the combination may act as a **key**.





Types of Keys



Primary Key



A **primary key** is a field in a table which uniquely identifies each row/record in a database table.



Primary keys must contain unique values.



A **primary key** column cannot have NULL values.



When multiple fields are used as a **primary key**, they are called a composite **key**.

[Back to Types of Keys](#)

Candidate Key



A **candidate key** is a set of attributes (or attribute) which uniquely identify the tuples in relation or table.



As we know that Primary **key** is a minimal super **key**, so there is one and only one primary **key** in any relationship but there is more than one **candidate key** can take place.

[Back to
Types of Keys](#)

Alternate Key



An **alternate key** is a **key** associated with one or more columns whose values uniquely identify every row in the table, but which is not the primary **key**.



For example, where the primary **key** for a table may be the employee id, the **alternate key** might combine the first, middle, and last names of the employee.

[Back to
Types of Keys](#)

Foreign Key



A **FOREIGN KEY** is a **key** used to link two tables together.



A **FOREIGN KEY** is a field (or collection of fields) in one table that refers to the **PRIMARY KEY** in another table.



For **example**, say we have two tables, a CUSTOMER table that includes all customer data, and an ORDERS table that includes all customer orders

[Back to
Types of Keys](#)

Database Languages



Database languages are used to read, update and store data in a **database**.



There are several such **languages** that can be used for this purpose; one of them is SQL

- Data Definition Language
- Data Manipulation Language
- Data Control Language



Data Definition Language



DDL stands for **Data Definition Language**.



It is used to **define database** structure or pattern.



It is used to create schema, tables, indexes, constraints, etc. in the **database**.



Using the **DDL** statements, you can **create,alter,delete,Drop** the skeleton of the **database**.

Back to
Database Languages

Data Manipulation Language



A **data manipulation language (DML)** is a computer programming **language** used for adding (inserting), deleting, and modifying (updating) **data** in a database.



A popular **data manipulation language** is that of Structured Query **Language (SQL)**, which is used to retrieve and **manipulate data** in a relational database.



Select, Insert , Update , Delete

Back to
Database Languages

Data Control Language



A **data control language** (DCL) is a syntax similar to a computer programming **language** used to **control** access to **data** stored in a database (Authorization).



In particular, it is a component of **Structured Query Language** (SQL).



Examples of DCL commands include: GRANT to allow specified users to perform specified tasks.- **User Access Control**

[Back to Database Languages](#)

Entry Relationship Model (E-R Model)



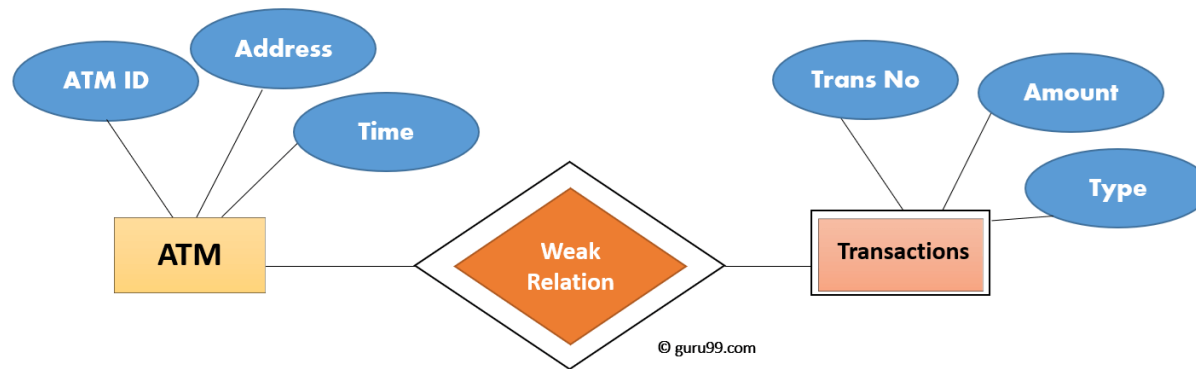
ENTITY RELATIONAL (ER) MODEL is a high-level conceptual data **model diagram**.



ER modeling helps you to analyze data requirements systematically to produce a well-designed database.



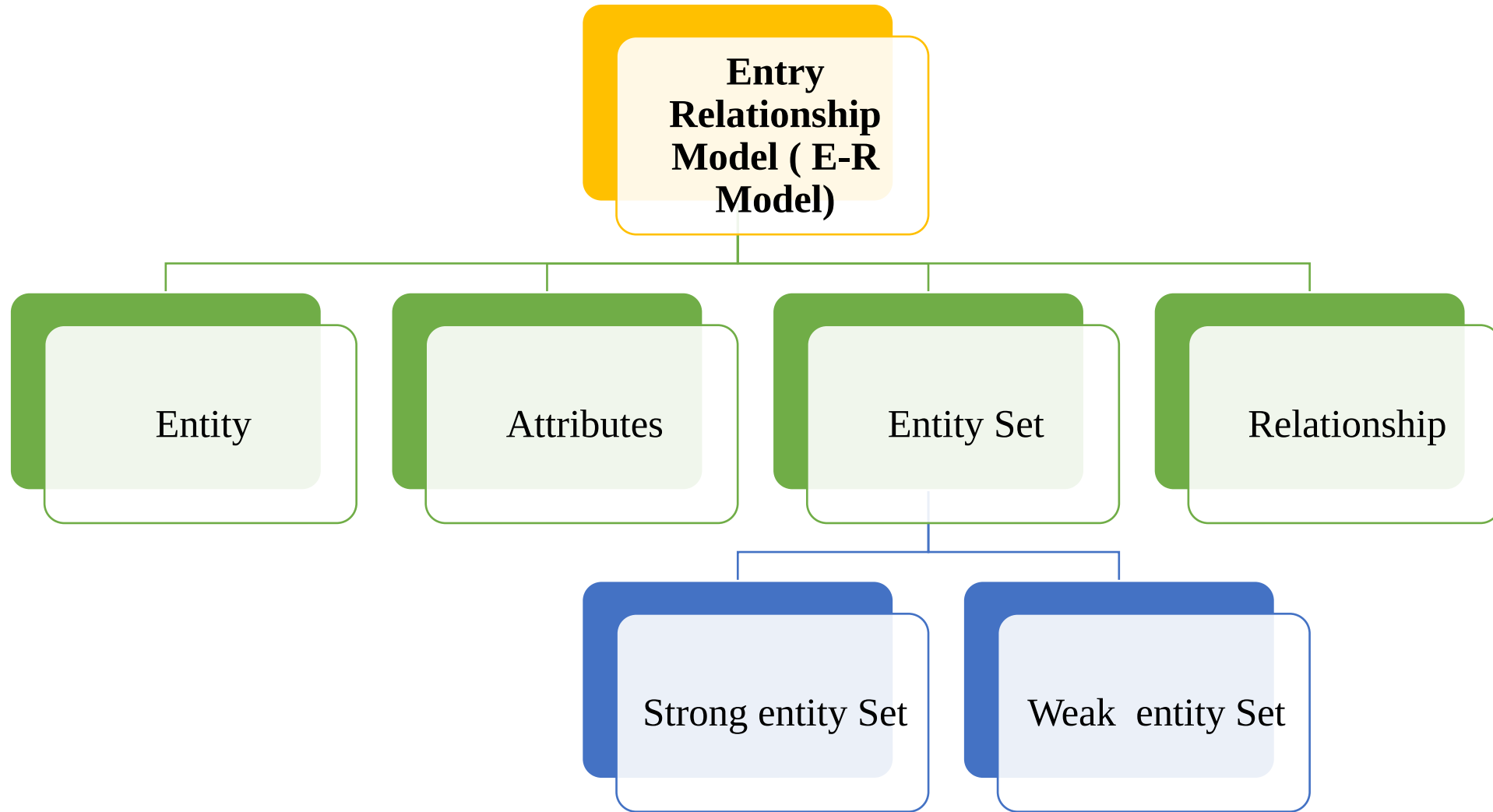
The **Entity-Relation model** represents real-world **entities** and the **relationship** between them



Types of ER Model

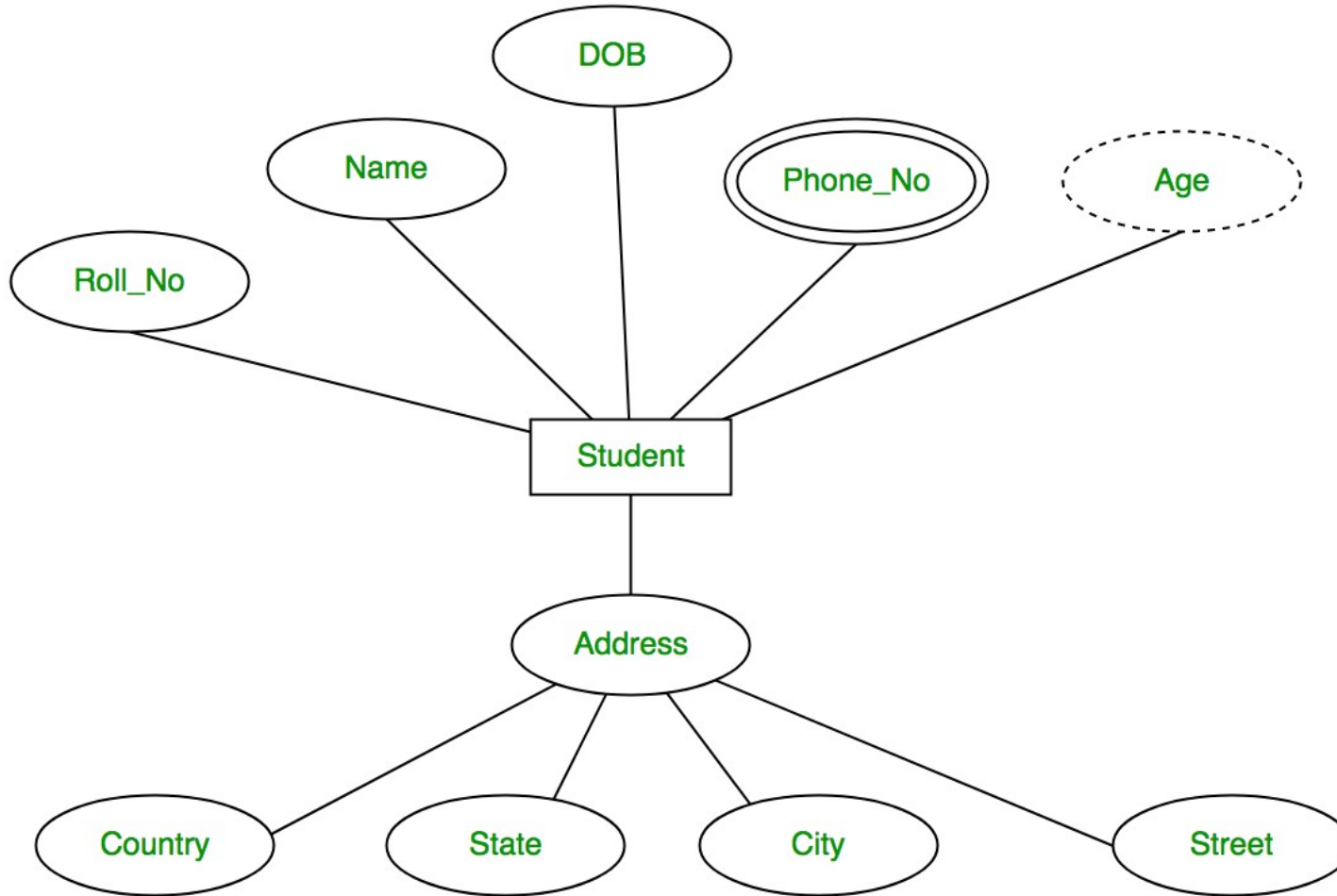


Entry Relationship Model (E-R Model)



Explanation of ER Model

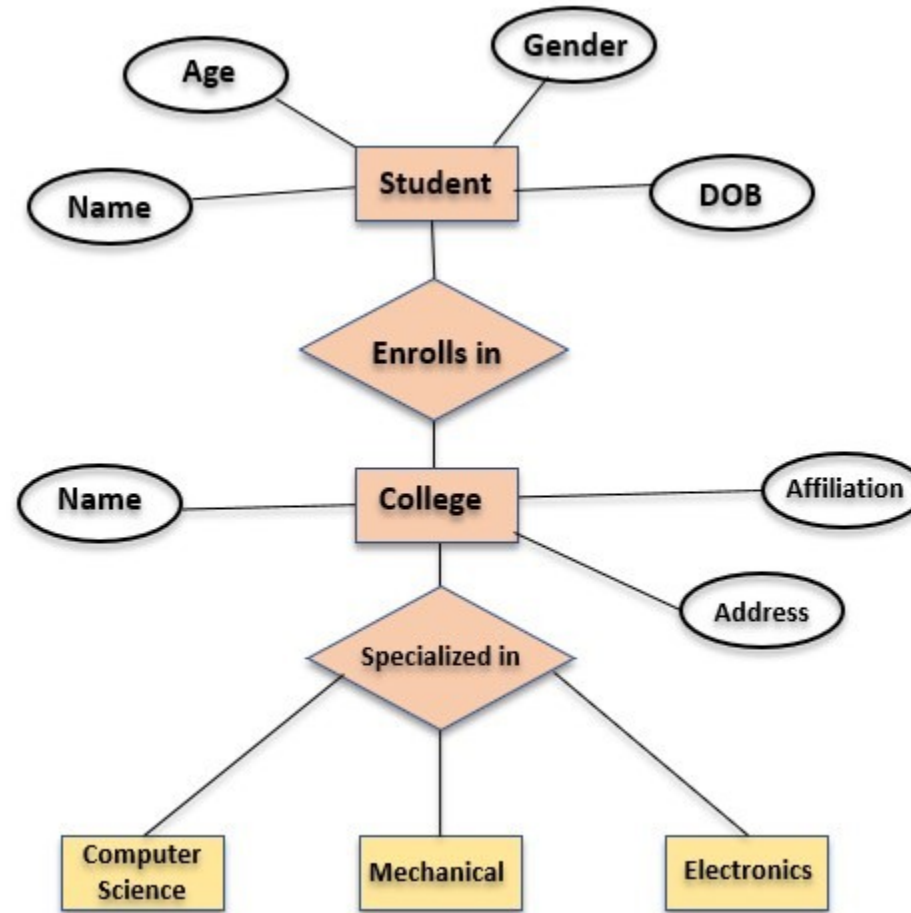
Entry Relationship Model (E-R Model)



Explanation of ER Model



Entry Relationship Model (E-R Model)



Explanation of ER Model

Entity



An **ERD entity** is a definable thing or concept within a system, such as a person/role (e.g. Student), object (e.g. Invoice), concept (e.g. Profile) or event (e.g. Transaction) note:



In **ERD**, the term "**entity**" is often used instead of "table", but they are the same

Types of ER Model

Attributes



Attributes are the properties which define the entity type. For example, Roll No, Name, DOB, Age, Address, Mobile No are the **attributes** which defines entity type Student.



In **ER diagram**, **attribute** is **represented** by an oval.



The **attribute** which uniquely identifies each entity in the entity set is called key **attribute**.

Types of ER Model



Entity Set



An **entity set** is a group of similar **entities** and these **entities** can have attributes.



In terms of **DBMS**, an **entity** is a table or attribute of a table in **database**, so by showing **relationship** among tables and their attributes, **ER diagram** shows the complete logical structure of a **database**.



There are 2 types of Entity set. They are

- Strong entity Set
- Weak entity Set

Types of ER Model

Strong entity Set



Strong entity set always has a primary key.



Primary Key is one of its attributes which helps to identify its member.

Weak entity Set



A **weak entity** is an **entity** that cannot be uniquely identified by its attributes alone; therefore, it must use a foreign key in conjunction with its attributes to create a primary key.

Back to
Entity Set

Types of ER Model



Relationship



A relationship type represents the association between entity



There are three types of relationships



One-to-one



One-to-many



Many-to-many

Types of ER Model



'Hurrah!'

We completed this section.



Next Section

Coming
Soon...

