### **UNIT 4.4**

### **RELATIONAL DATABASE SYSTEM**

**TUTOR NAME: MS. IBTISAM MOGUL**

**SUBMITTED BY FATIMA EJAZ BARRI**

Contents

[**UNIT 4.4** 1](#_Toc22118071)

[**RELATIONAL DATABASE SYSTEM** 1](#_Toc22118072)

# [ENTITY 2](#_Toc22118073)

# ENTITY

* **ER Models**

An entity relationship diagram (ERD) is a data modelling technique that graphically illustrates information systems entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure. Also known as ERD, ER Diagram or ER Model, is a type of structural diagram for use in database design. However ER Diagram includes many specialized symbols and is meanings make this model unique

Here are some typical use cases.

* Database design:

Depending on the scale of change, it can be risky to alter a database structure directly in a DBMS. To avoid ruining the data in a production database it is important to plan the changes carefully. ERD is a tool that helps. By drawing ER diagrams to visualize database design ideas, you have a chance to identify the mistakes and design flaws, and to make corrections before executing the changes in the database.

* Database debugging:

To debug database issues can be challenging, especially when the database contains many tables, which require writing complex SQL in getting the information you need by visualizing a database schema with an ERD, you have a full picture of the entire database schema. You can easily locate entities, view their attributes and identify the relationships they have with others. All these allow you to analyse an existing database and to reveal database problems easier.

* Database creation and patching:

Visual paradigm an ERD tool supports a database generation tool that can automate the database creation and patching process by means of ER diagrams. Therefore, with this ER diagram tool, your ER design is no longer just a static diagram but a mirror that reflects truly the physical database structure.

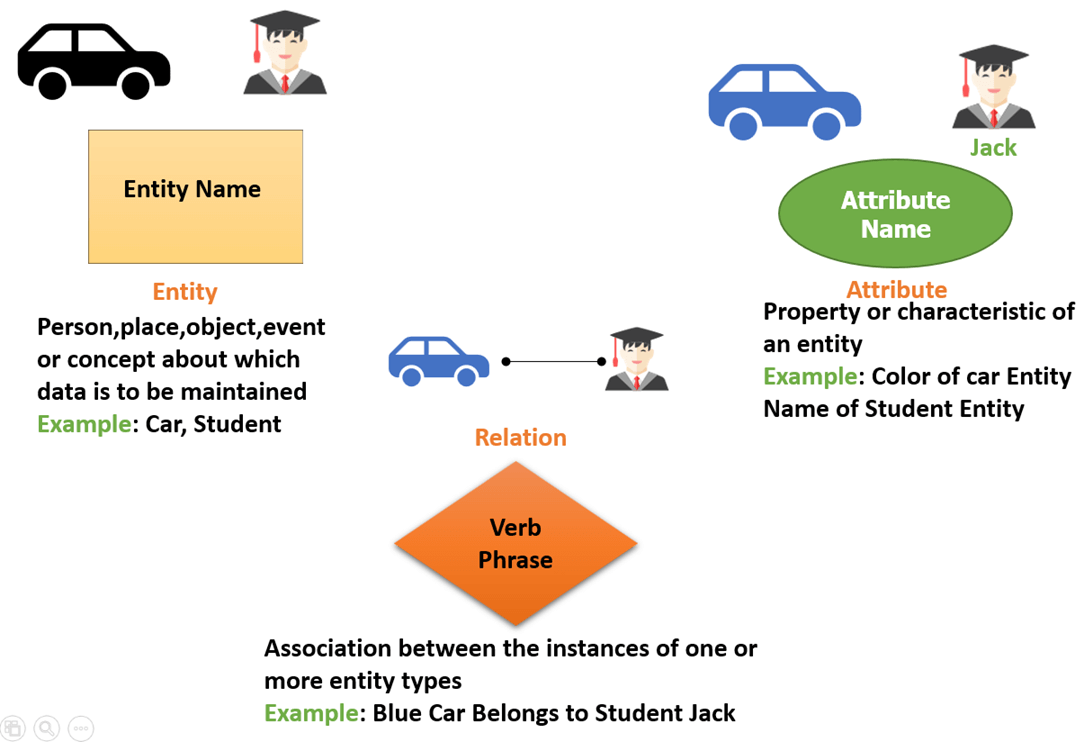
* Aid in requirements gathering:

Determine the requirements of an information system by drawing a concept ERD that depicts the high-level business objects of the system. Such an initial model can also be evolved into a physical database model that aids the creation of a relational database or aids in the creation of process maps and data flow modes.

Components of the ER Diagram

This model is based on three basic concepts:

* Entities
* Attributes
* Relationships

For example, in a university database, we might have entities for students, courses, and lecturers. Student’s entity can have attributes like roll no, name, and dept.id they might have relationships with courses and lectures.

* Entity

A real world things either living or non-living that is easily recognizable and non-recognizable. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simple a fact about the enterprise or an event that happens in the real world.an entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute and a unique key. Every entity is made up of some attributes, which represent that entity.

Examples of entities:

Person: employee, student, patient

Place: store, building

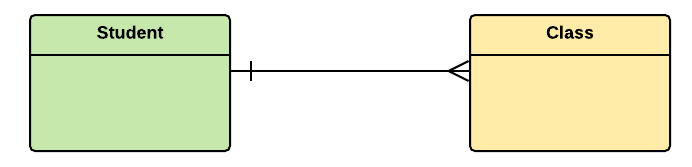
Object: machine, product, and car

Event: sale, registration, renewal

Concept: account, course

* Entity set:

Student

An entity set is a group of similar kind of entities.it may contain entities with attribute sharing similar values. Entities are represented by their properties, which also called attributes. All attributes have their separate values.

For example, a student entity may have a name, age, class as attributes.

Example of entities:

A university may have some departments. All these departments employ various lectures and offers several programs. Some course makeup each program. Students register in a particular program and enrol in various courses. A lecturer from the specific department takes each course, and each lecturer teaches a various group of students.

* 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.

**Key Attribute:**

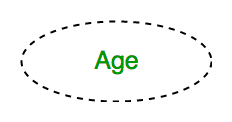
The attribute, which uniquely identifies each entity in the entity set, is called key attribute. For example, roll no will be unique for each student.in ER diagram; key attribute is represented by an oval with underlying lines.

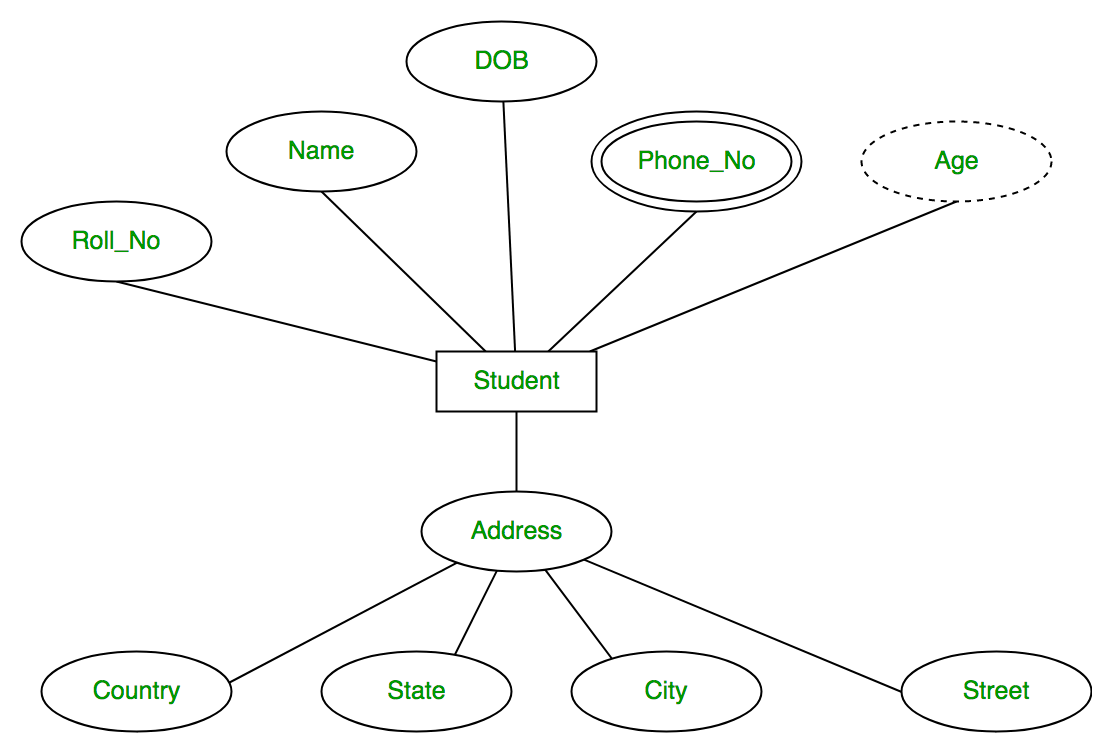
**Composite attribute:**

An attribute composed of many other attribute is called as composite attribute. For example, address attribute of student entity type consists of street, city, state, and contry.in ER diagram, composite attribute is represented by an oval comprising of ovals.

**Multivalued Attribute:**

An attribute consisting more than one value for a given entity. For example, phone no (Can be more than one for a given student). In ER diagram, multivalued attribute is represented by double oval.

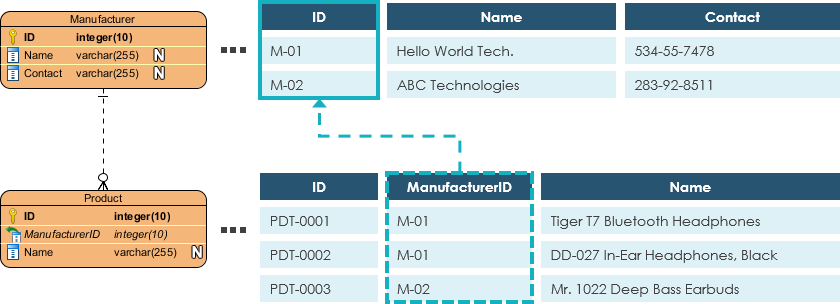


The complete entity type student with its attributes can be represented as:

**Primary Key**

Also known as, PK, a primary key is a special kind of entity attribute that uniquely defines a record in a database table.in other words there must not be two or more records that share the same value for the primary key attribute. The ERD example below shows entity products with primary key attribute ID and a preview of table records in the database. The third record is invalid because another record already uses the value of ID “PDT-0002”.

**Foreign Key**

Also known as, FK, a foreign key is a reference to a primary key in a table. It is used to identify the relationships between entities. Note that foreign keys need not be unique. Multiple records can share the same values. The ER diagram example bellow shows an entity with some columns, among which a foreign key is used in referencing another entity.

**Relationship**

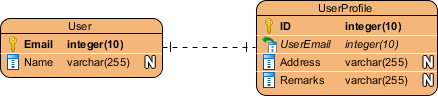
A relationship between two entities signifies that the two entities are associated with each other somehow. For example, a student might enrol in a course. The entity student is therefore related to course and a relationship is presented as a connector connecting between them.

**Cardinality**

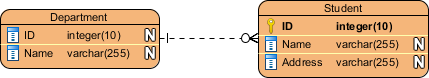
Cardinality defines the possible number of occurrences in one entity, which is associated with the number of occurrences in another. For example, one team has many players. When present in an ERD, the entity team and player are inter-connected with a one-to-many relationship.in an ERD diagram, cardinality is represented as a crows foot at the connectors ends. The three common cardinal relationships are one-to-one, one-to-many, and many-to-many.

**One-to-one cardinality example**

A one-to-one relationship is mostly used to split an entity in two to provide information concisely and make it more understandable. The figure below shows an example of a one-to-one relationship.

 **One-to-many cardinality example**

A one-to-many relationship refers to the relationship between two entities X and Y in which an instance of X maybe linked to many instances of Y but an instance of Y is linked to only one instance of X. The figure below shows an example of a one-to-many relationship.



**Many-to-many cardinality example**

A many-to-many relationship refers to the relationship between two entities X and Y in which X maybe linked to many instance of Y and vice versa. The figure below shows an example of a many-to-many relationship. Note that a many-to-many relationship is split into a pair of one-to-many relationships in a physical ERD. You will know what a physical ERD is in the next section.