**Database Systems: CS-329**

**Week 05 : EER Models**

**Enhanced entity-relationship (EER)** diagrams are basically an expanded upon version of ER diagrams. EER models are helpful tools for designing databases with high-level models. With their enhanced features, you can plan databases more thoroughly by delving into the properties and constraints with more precision. An EER diagram provides you with all the elements of an ER diagram while adding:

• Attribute or relationship inheritances

• Category or union types

• Specialization and generalization

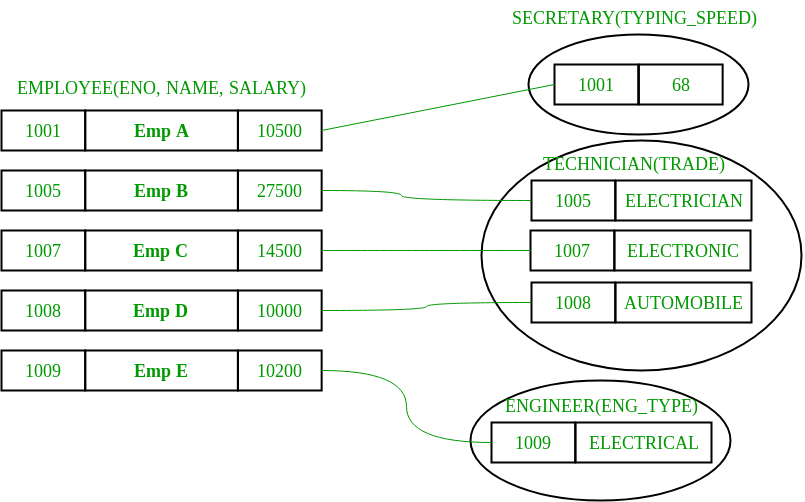
• Subclasses and superclasses

Enhanced entity-relationship diagrams are advanced database diagrams very similar to regular ER diagrams which represent requirements and complexities of complex databases.   
It is a diagrammatic technique for displaying the Sub Class and Super Class; Specialization and Generalization; Union or Category; Aggregation etc.

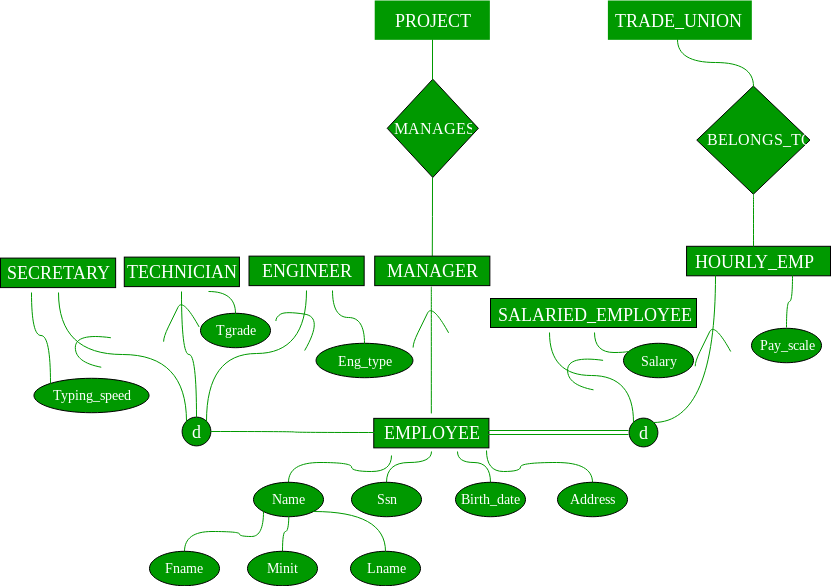
**Generalization and Specialization –**   
These are very common relationships found in real entities. However, this kind of relationship was added later as an enhanced extension to the classical ER model. **Specialized**classes are often called **subclass** while a **generalized class** is called a superclass, probably inspired by object-oriented programming. A sub-class is best understood by **“IS-A analysis”**. Following statements hopefully makes some sense to your mind “Technician IS-A Employee”, “Laptop IS-A Computer”.

An entity is a specialized type/class of another entity. For example, a Technician is a special Employee in a university system Faculty is a special class of Employee. We call this phenomenon generalization/specialization. In the example here Employee is a generalized entity class while the Technician and Faculty are specialized classes of Employee.

**Example –** This example instance of **“sub-class”** relationships. Here we have four sets of employees: Secretary, Technician, and Engineer. The employee is super-class of the rest three sets of individual sub-class is a subset of Employee set.



* An entity belonging to a sub-class is related to some super-class entity. For instance emp, no 1001 is a secretary, and his typing speed is 68. Emp no 1009 is an engineer (sub-class) and her trade is “Electrical”, so forth.
* Sub-class entity “inherits” all attributes of super-class; for example, employee 1001 will have attributes eno, name, salary, and typing speed.



* **Constraints** – There are two types of constraints on the “Sub-class” relationship.
* **Total or Partial** – A sub-classing relationship is total if every super-class entity is to be associated with some sub-class entity, otherwise partial. Sub-class “job type based employee category” is partial sub-classing – not necessary every employee is one of (secretary, engineer, and technician), i.e. union of these three types is a proper subset of all employees. Whereas other sub-classing “Salaried Employee AND Hourly Employee” is total; the union of entities from sub-classes is equal to the total employee set, i.e. every employee necessarily has to be one of them.
* **Overlapped or Disjoint** – If an entity from super-set can be related (can occur) in multiple sub-class sets, then it is overlapped sub-classing, otherwise disjoint. Both the examples: job-type based and salaries/hourly employee sub-classing are disjoint.
* Note – These constraints are independent of each other: can be “overlapped and total or partial” or “disjoint and total or partial”. Also, sub-classing has transitive property.
* **Multiple Inheritance** (sub-class of multiple superclasses) –   
  An entity can be a sub-class of multiple entity types; such entities are sub-class of multiple entities and have multiple super-classes; Teaching Assistant can subclass of Employee and Student both. A faculty in a university system can be a subclass of Employee and Alumnus. In multiple inheritances, attributes of sub-class are the union of attributes of all super-classes.