Department of Computer Science Engineering

SRM IST, Kattankulathur – 603 203

18CSC206J – SOFTWARE ENGINEERING AND PROJECT MANAGEMENT

Experiment No	07
Title of Experiment	Design a Entity relationship diagram
Name of the Candidate	Sai Rohit P
Team Members	Sai Rohit (RA2111003010806) Pavan Sagar (RA2111003010809)
Date of Experiment	

Mark Split Up			
S.No	**Description**	**Maximum Mark**	**Mark Obtained**
1	Exercise	5	
2	Viva	5	
Total		**10**	

Staff Signature with date

Aim:

To create the Entity Relationship Diagram

Team Members:

S. No.	Register Number	Name	Role
1	RA2111003010806	Sai Rohit	Rep
2	RA2111003010809	Pavan Sagar	Member

Project Title: TimetableSOS

What is ER Diagram?

- ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

- ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

- At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

What is ER Model?

- ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well-designed database.
- ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.
- ER Modelling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modelling before implementing your database.

Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

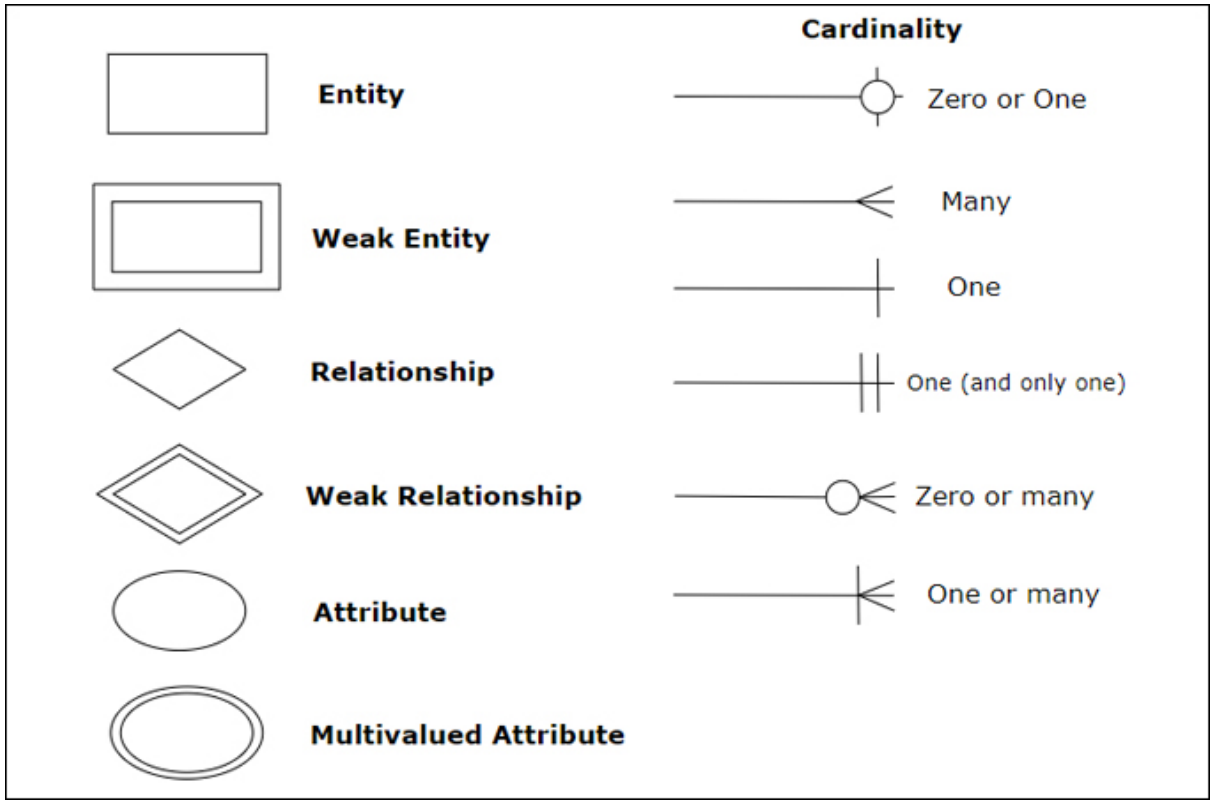
- Helps you to define terms related to entity relationship modelling
- Provide a preview of how all your tables should connect, what fields are going to be on each table
- Helps to describe entities, attributes, relationships
- ER diagrams are translatable into relational tables which allows you to build databases quickly
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
- The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
- ERD Diagram allows you to communicate with the logical structure of the database to users

Components of the ER Diagram

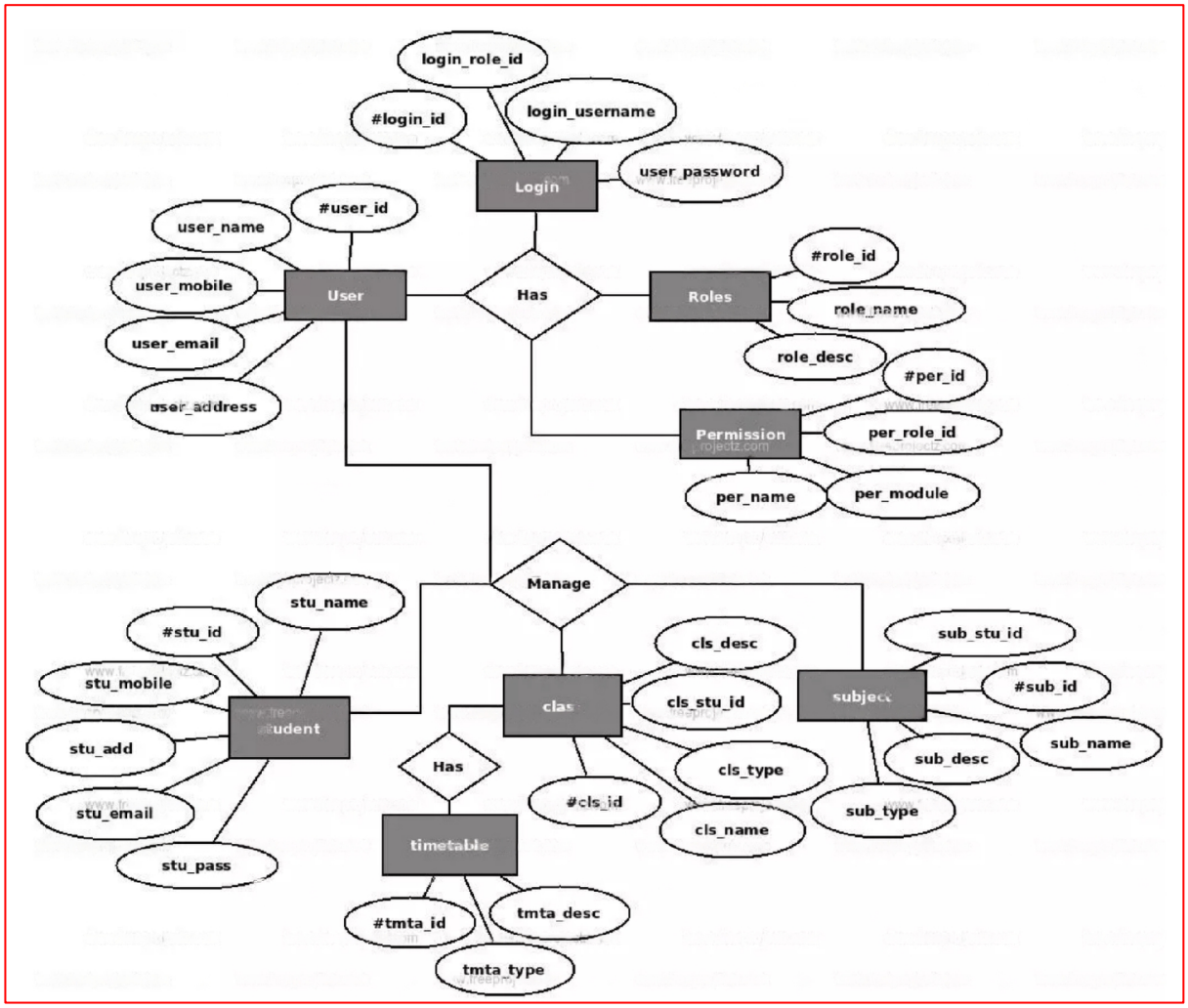
- This model is based on three basic concepts: Entities, Attributes, Relationships

ER Diagram – Notations

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
- Double ellipses represent multivalued attributes.
- Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes



ER Diagram – Timetable SOS



Additional Notes

- A database can be modelled as a collection of entities, relationship among entities.
- An entity is an object that exists and is distinguishable from other objects.
  Example: specific person, company, event, plant
- Entities have attributes.
  Example: people have names and addresses
- An entity set is a set of entities of the same type that share the same properties.
  Example: set of all persons, companies, trees, holidays
- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- We express cardinality constraints by drawing either a directed line (->), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.
- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.
  Example: customer = (customer-id, customer-name, customer-street, customer-city) loan = (loan-number, amount)
- Domain – the set of permitted values for each attribute

Attribute types:

1. Simple and composite attributes.
2. Single-valued and multi-valued attributes
   E.g. multivalued attribute: phone-numbers
3. Derived attributes-Can be computed from other attributes
   E.g. age, given date of birth

Cardinality

For a binary relationship set the mapping cardinality must be one of the following types:

1. One to one

A customer is associated with at most one loan via the relationship borrower. A loan is associated with at most one customer via borrower

2. One to many

A loan is associated with at most one customer via borrower, a customer is associated with several (including 0) loans via borrower

3. Many to one

A loan is associated with several (including 0) customers via borrower, a customer is associated with at most one loan via borrower

4. Many to many

A loan is associated with several (including 0) customers via borrower, a customer is associated with several loans (including 0) via borrower

Weak Entity Set

- An entity set that does not have a primary key is referred to as a weak entity set and represented by double outlined box in E-R diagram.
  Example : Consider the entity set payment which got three attributes : payment_number, payment_date and payment_amount. Payment numbers are sequential starting from 1 generally separately for each loan. Although each payment entity is distinct, payments for different loans may share the same payment number. Thus this entity set does not have a primary key.

Discriminator

- The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set
  Example: discriminator of weak entity set payment is the attribute payment_number since for each loan a payment number uniquely identifies one single payment for that loan.

Specialization-Generalization-ISA

- E-R model provides means of representing these distinctive entity groupings
- Process of designating subgroupings within an entity set is called specialization depicted by triangle component labelled ISA (“is a”)
- Bottom up design process in which multiple entity sets are synthesized into higher level entity set - Generalization
- ISA relationship may also be referred to as superclass-subclass relationship
- Higher and lower level entity sets are designated by the terms superclass and subclass.
- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.

Total & Partial Participation

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
  Example: participation of loan in borrower is total, every loan must have a customer associated to it via borrower
- Partial participation: some entities may not participate in any relationship in the relationship set
  Example: participation of customer in borrower is partial

Cardinality limits

- Cardinality limits can also express participation constraints
- Minimum and maximum cardinality is expressed as l..h where l is the minimum and h is the maximum cardinality
- Minimum value of 1 indicates total participation of entity set in relationship set
- Maximum value of 1 indicates entity participates in atmost one relationship set.
- Maximum value of * indicates no limit

Role indicator

- Entity sets of a relationship need not be distinct
- The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works-for relationship set.
- Roles are indicated in E-R diagrams by labelling the lines that connect diamonds to rectangles.
- Role labels are optional, and are used to clarify semantics of the relationship

Disjoint Generalization

- Disjointness constraint requires that an entity belong to more than one lower level entity set.

  Example: account entity can satisfy only one condition for account_type attribute ; entity can either be savings or chequing account but not both.

Result: Thus, the entity relationship diagram was created successfully.