

Experiment No.04

PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1 Aim: To study ER modeling and designing of relational model.

A.2 Prerequisite:

Basic concepts of database, data models like ER-models and relational model.

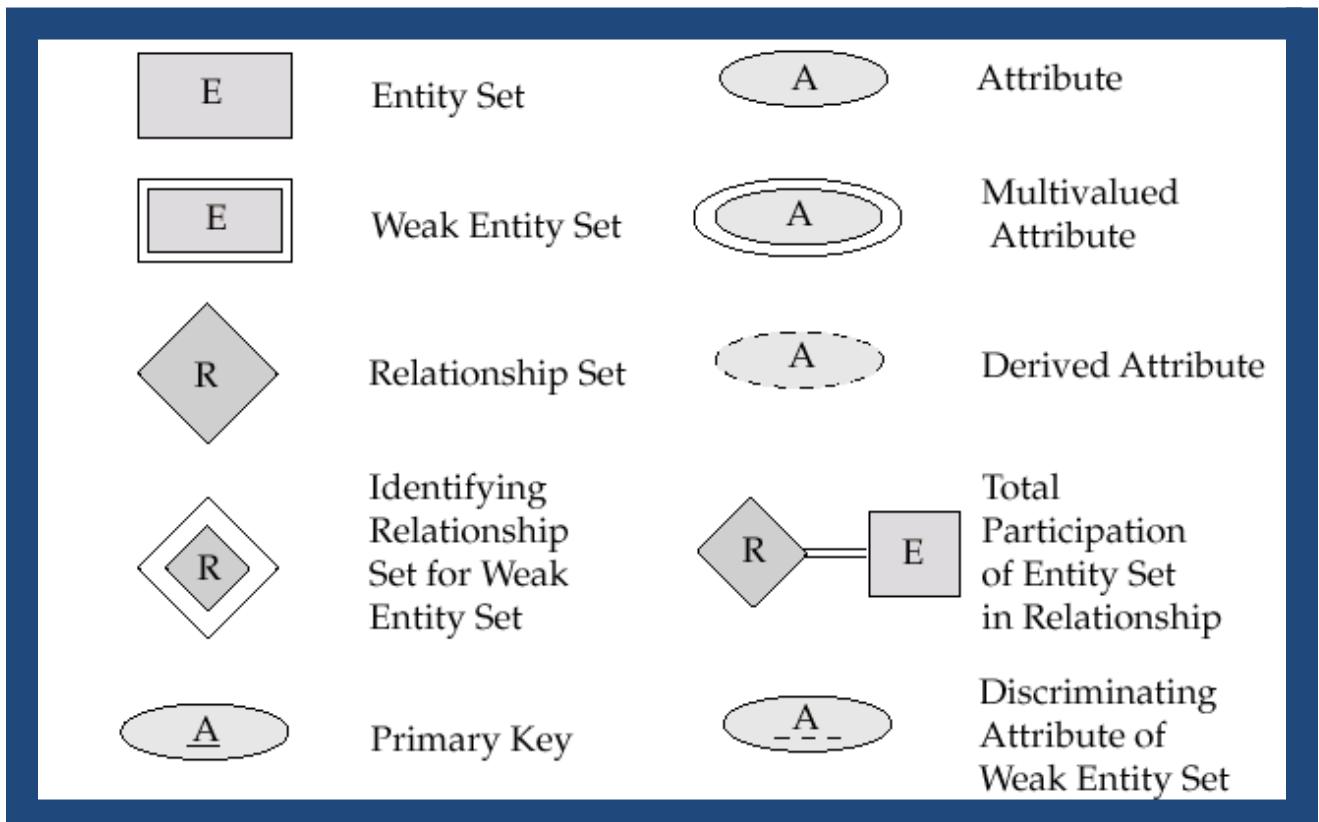
A.3 Outcome:

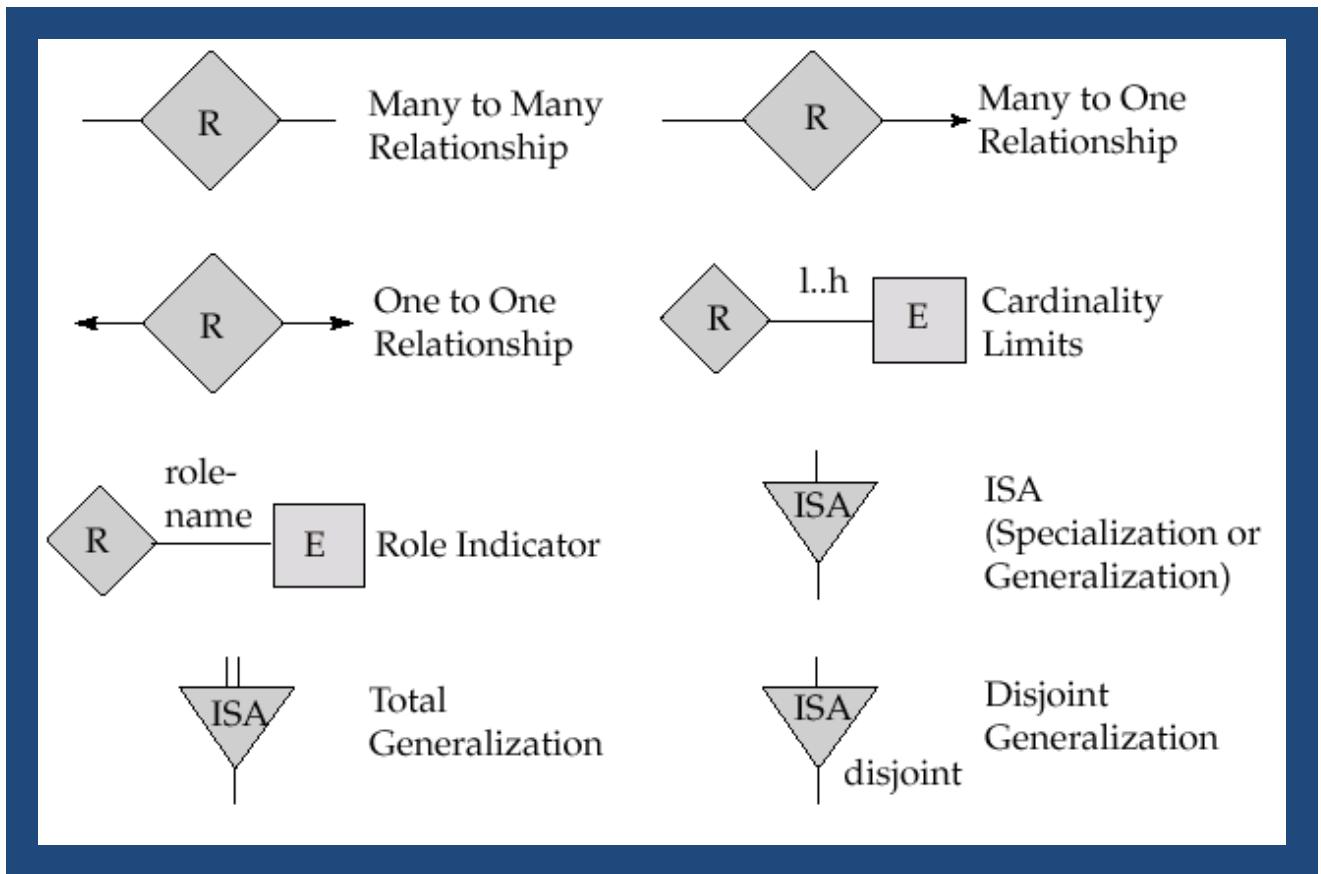
After successful completion of this experiment students will be able to

1. Design a relational database management system and develop it through ER modeling and relational models.

A.4 Theory:

Summary of symbols used in ER notation:





Reduction of ER model to relational model:

Rules:

- **Build a table for each entity set**
- **Build a table for each relationship set if necessary (depending upon cardinality)**
- **Make a column in the table for each attribute in the entity set**
- **Weak Entity Set Cannot exists alone**
 - **To build a table/schema for weak entity set**
 - **Construct a table with one column for each attribute in the weak entity set**

- Remember to include discriminator
 - Augment one extra column on the right side of the table, put in there the primary key of the Strong Entity Set (the entity set that the weak entity set is depending on)
 - Primary Key of the weak entity set = Discriminator + foreign key
-
- Representation of relationship set:
 - For one-to-one relationship
 - Build a table with two columns, one column for each participating entity set's primary key. Add successive columns, one for each descriptive attributes of the relationship set (if any).
 - For one-to-many relationship
 - For one-to-many/many-to-one relationship with one entity set having total participation on “many” side
 - Augment one extra column on the right side of the table of the entity set on the “many” side, put in there the primary key of the entity set on the “one” side as per to the relationship.
 - For many-to-many relationship
 - Primary key of this new schema is the union of the foreign keys of both entity sets.
-
- Representing composite attribute:
 - One column for each component attribute

- **NO column for the composite attribute itself**
- **Representing multi-valued attribute:**
 - **For each multivalue attribute in an entity set/relationship set**
 - **Build a new relation schema with two columns**
 - **One column for the primary keys of the entity set/relationship set that has the multivalue attribute**
- **Representing Generalization\Specialization:**
 - **Two general approaches depending on disjointness and completeness**
 - **For non-disjoint and/or non-complete class hierarchy:**
 - **create a table for each super class entity set according to normal entity set translation method.**
 - **Create a table for each subclass entity set with a column for each of the attributes of that entity set plus one for each attributes of the primary key of the super class entity set**
 - **This primary key from super class entity set is also used as the primary key for this new table**
 - **For disjoint AND complete mapping class hierarchy:**
 - **DO NOT create a table for the super class entity set**
 - **Create a table for each subclass entity set include all attributes of that subclass entity set and attributes of the superclass entity set**

A.5 Task:

Task1 :

Draw an ER diagram for given case study

We need to create a database schema design based on the following (simplified) requirements of the COMPANY Database:

1. The company is organized into DEPARTMENTS. Each department has a name, number and an employee who manages the department. We keep track of the start date of the department manager. A department may have several locations.
2. Each department controls a number of PROJECTS. Each project has a unique name, unique number and is located at a single location.
3. There are two types of employees in company- technical employee and non-technical employee (administrative staff).
4. We store each EMPLOYEE's social security number, address, salary, sex, and birthdate.
5. Each technical employee works for one department but may work on several projects.
6. We keep track of the number of hours per week that a technical employee currently works on each project.
7. The technical employee gains score based on work done. The latest score of each technical employee is stored.
8. The non-technical employees are given incentives based on their work.
9. We also keep track of the direct supervisor of each employee.
10. Each employee may have a number of DEPENDENTS.
11. For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.

Task 2: Convert it into relational model

Task 3: Explore the available software to create ER diagram

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

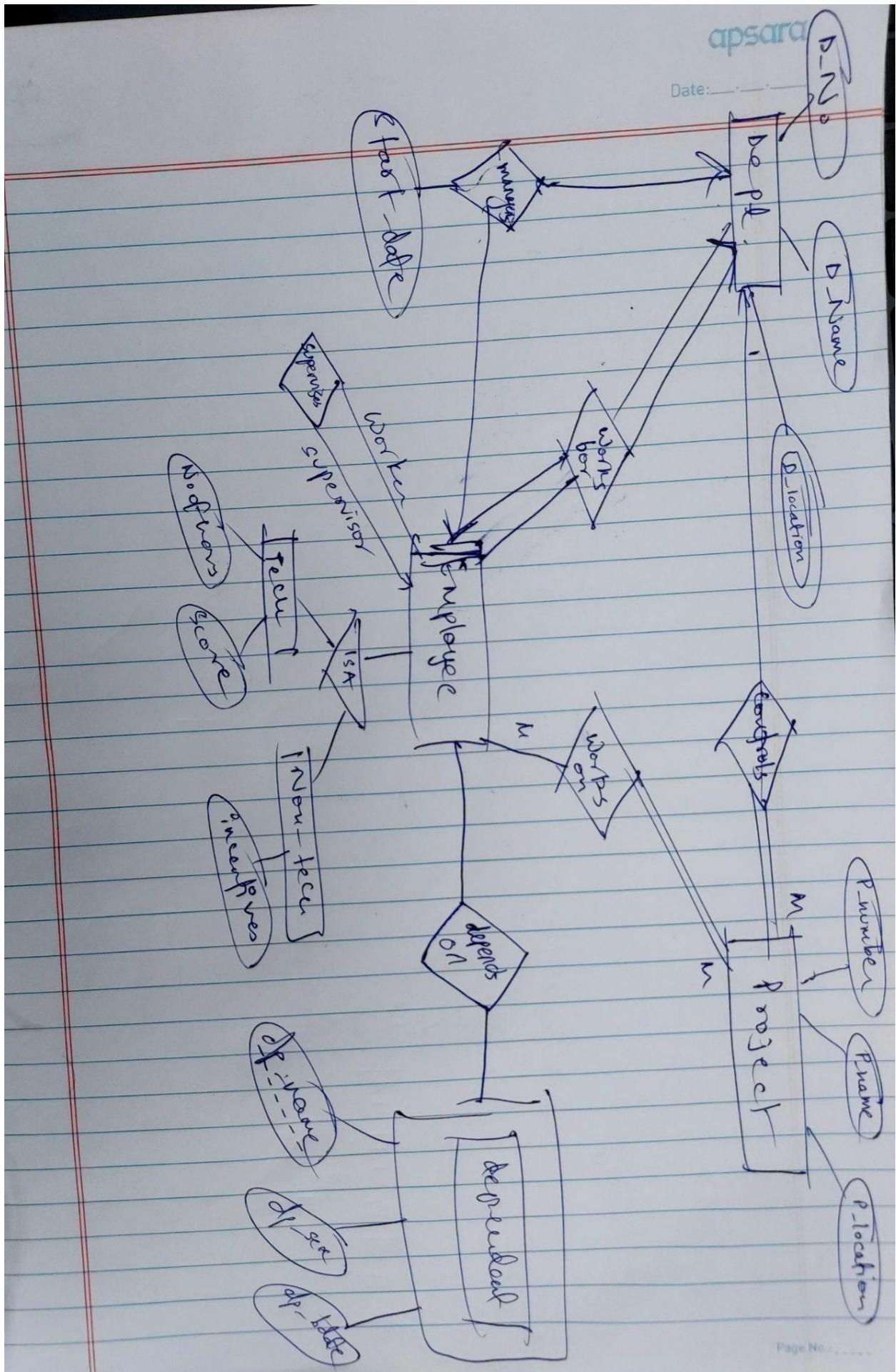
(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

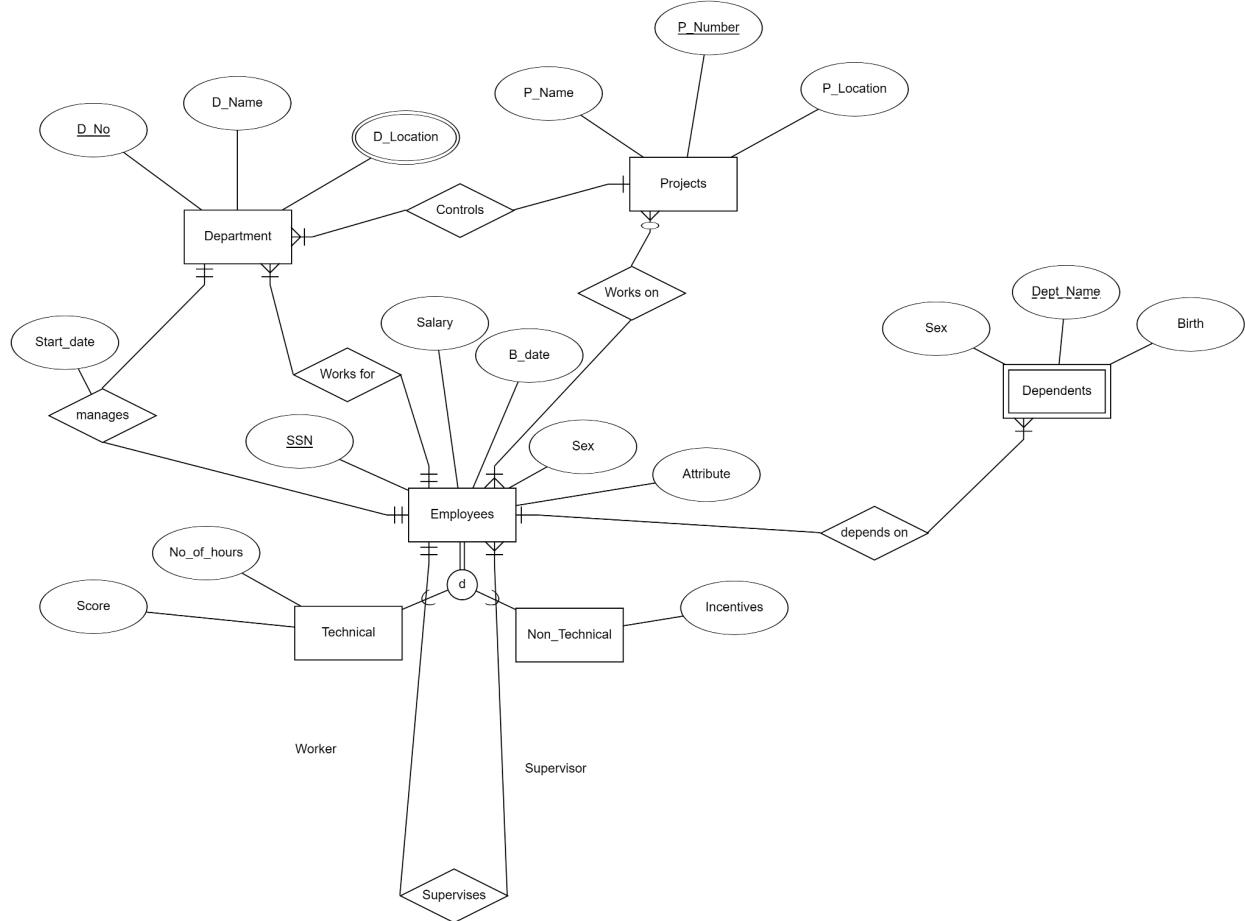
Roll No. I066	Name: Srihari Thyagarajan
Program : B Tech Artificial Intelligence	Division: I
Batch: B3	Date of Experiment: 26/08/2022
Date of Submission: 26/08/2022	Grade :

B.1 ER diagram:

Classwork ERD -

ERD+ Diagram -





B.2 Relational model- tables:

Class work Relational Model -

From
Rule 1

Step IV
Relational schema -

* Department (DNO, Dname)

* Project (PNO, Pname, Plocation, DNO)
Managerid*, startdate*

* Employee (SSN, Address, Salary, ~~bdate~~, sex,
DNO*, supervisor-id*)

* Technical (SSN, Address, Salary, b-date, sex,
no-of-hours, score)

* Non-Technical (SSN, Address, Salary,
~~bdate, sex, incentives~~)

for multi-valued
attributes, see Rule 2

* Location (DNO*, Dlocation)

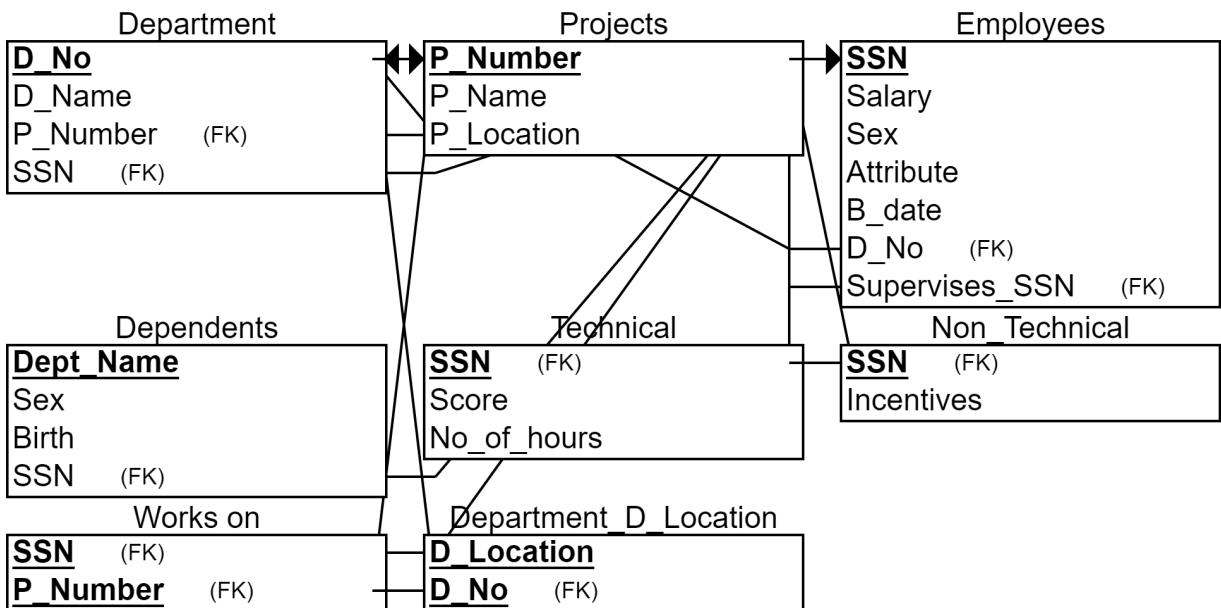
for weak
entities
see rule 3.

* Dependent (DepName, ddate, sex, SSN*)

Tables
based
on relationship's

* Works on (SSN*, PNO*)

ERD+ Relational Schema:



B.3 Curiosity Question:

Give example of weak entity set.

Example of weak entity sets include -

It can exist only when another entity exists.

- 1) Rooms cannot exist without buildings.
- 2) Payments cannot exist without customers.

B.4 Conclusion:

From the above experiment, we learnt to build Entity Relation Diagrams from the data given in problem statements. Also converting them to respective Relational schemas to help in understanding the structure of the Database.