2019UCS0073
Pulkit Mahajan

# Question 1

A.)

ans.

Weak entity sets :
When the attributes associated with an entity set are not sufficient to uniquely identify an entity, then this is called a weak entity.

A weak entity set doesn't have any primary key for discriminating the entities in a set, the weak entity set is dependant on a particular strong entity set

The weak entity set gets its primary key by combining the discriminating attributes of weak entity and the primary key of 'identifying' strong entity set.

Representation in ER Diagram.

⊙ The weak entity set is shown by a double rectangle

⊙ Attributes of weak entity set are underlined with a dashed line.

⊙ Relationship of a weak entity set with a strong entity set is shown with a double diamond

Strong Entity Set
- Primary key
- att1
- att2

Relation

Weak Entity Set
- name   place
- etc
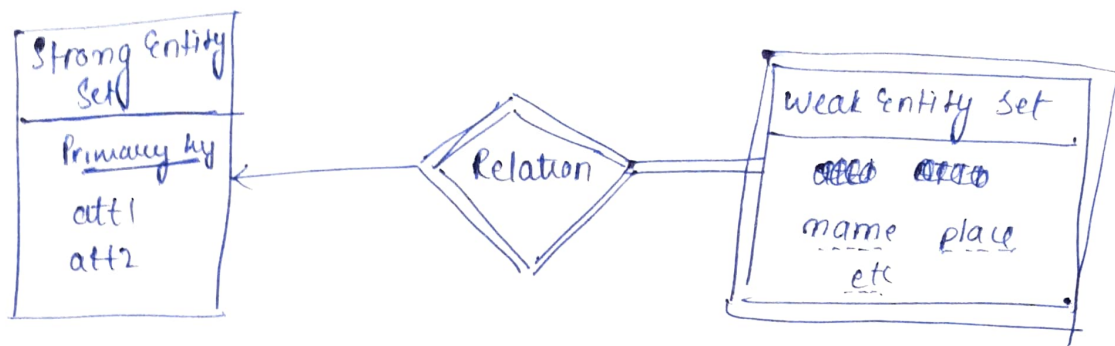
Ques 1

B:)

Armstrong axioms have to be sound and complete

Axioms

Reflexivity ⇒ if $\beta \subseteq \alpha$ then $\alpha \rightarrow \beta$

Augmentation ⇒ if $\alpha \rightarrow \beta$ holds and γ is another attribute set
then $\alpha\gamma \rightarrow \gamma\beta$ also holds

transitivity ⇒ If $\alpha \rightarrow \beta$ holds
and $\beta \rightarrow \gamma$ also holds
the implication that $\alpha \rightarrow \gamma$
also holds true

Sound: Given a set of functional dependencies F on a Relational schema R, any dependency that we can infer from F using the primary rules of armstrong axioms holds in every relation state of R that satisfies the dependencies in F

Complete: using Primary rules repeatedly to infer dependencies until no more dependencies can be inferred.
↳ Hence complete set of Dependencies possible are included.

Class 2

1.
ans   ENTITIES :

       1) Departments   Enterprise:
       2) Projects   AR Construction
       3) Employees   House

2)

Entity 1	Entity 2	relationship	Participation Type mentioned in brackets
Employee (Full)	Department (Partial)	'belongs to'	
Department (partial)	Projects (Full)	'manages'	
Employee (Full)	Project (Partial)	'assigned'	

3)

Department

Department_id
dept-name
head_of_dept
building-id
floor_id

1——1   manages   many

Projects

project-id
proj-domain
proj-manager
date-started
budget
tentative-time

1——1

1——1

Employee

employee-id
name
age
salary
address
experience

belongs to   many
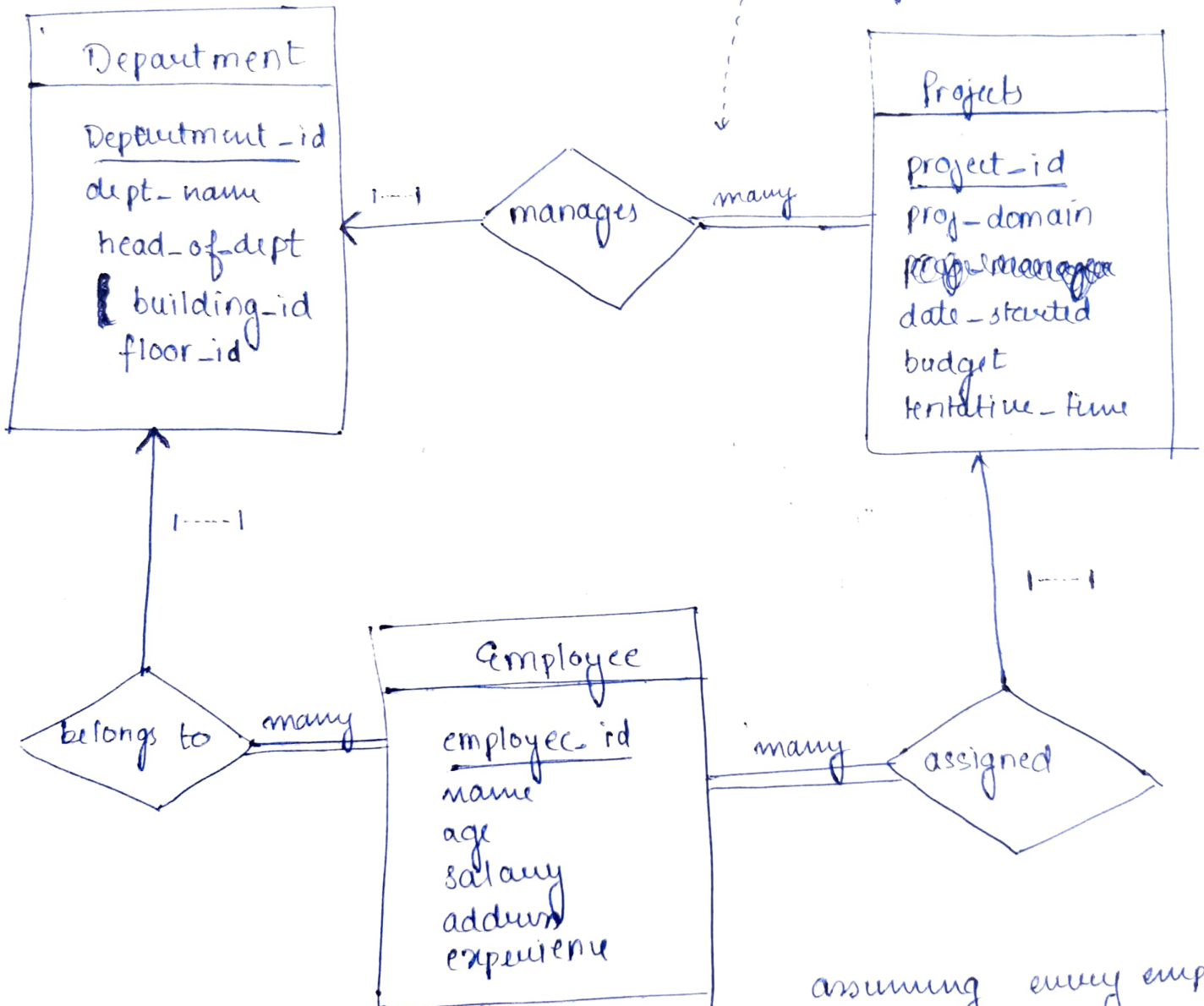
many   assigned

assuming every employee is necessarily assigned to a project.

4)

Initially we create a tentative.
Department (Department_id , dept-name, head-of-dept,
building-id , floor-id)

Projects (Project_id , project_domain , date-started , budget , tentative-time)

Employee (employee-id, name, age , salary , address, experience)

Relationships { These are not schemas only explanations}

manages (participation is full on project side (many)
         "   "   partial on Dept-side (one)
     add primary key of Department to Attributes of Projects)

belongs to (Employee has per' full participation (many side)
         Department "   partial participation (one side)
     add primary key of Department to relational schema
     of Employee)

assigned (Employee side - many - full participation
         Project   - one - partial participation
     so add primary key of projects to Employee schema)

— x — x — x — x — x — x — x — x —

FINALLY → Final Relational Schemas

                    * pk
Department < Department_id , dept-name , head-of-dept, building-id
                    floor-id

           * pk   * fk
Projects < Project-id, Department-id , proj-domain, date-started
                    budget , tentative-time

           * pk   * fk   * fk
Employee < employee-id , project-id , department_id , name, age, salary
                    address, experience