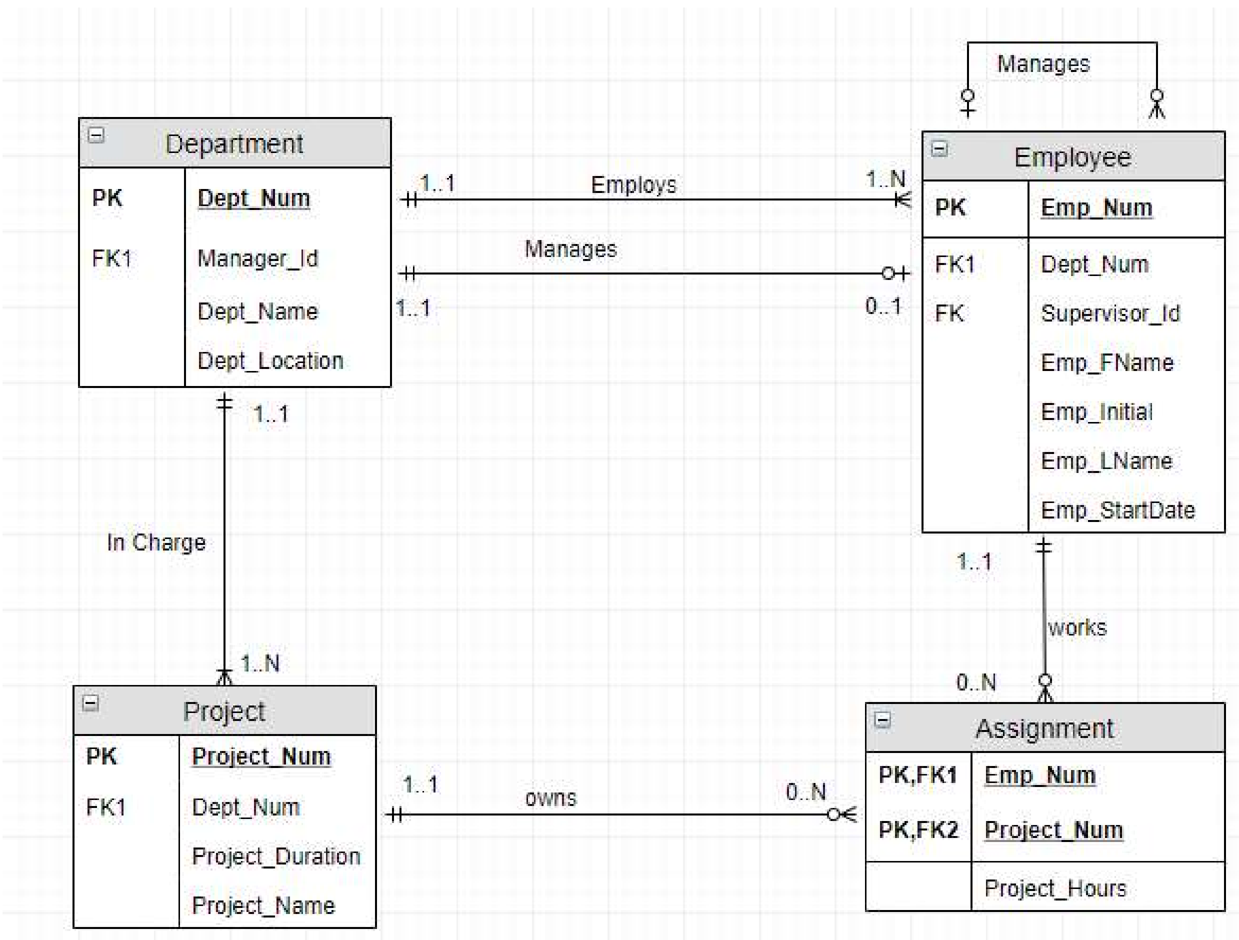
HW1: (E)ER diagrams
Total points: 6 [plus a 1 point bonus]
In this homework (consisting of 4(+1) problems), you'll be working with (E)ER diagrams. The questions are easy, straightforward. You can draw the diagrams on paper, and take legible screenshots with your phone, for submission (to include in a .doc and generate a .pdf from it). In all diagrams, indicate PK, FK, and connectivity. You can use Chen notation, or Crow's Foot.

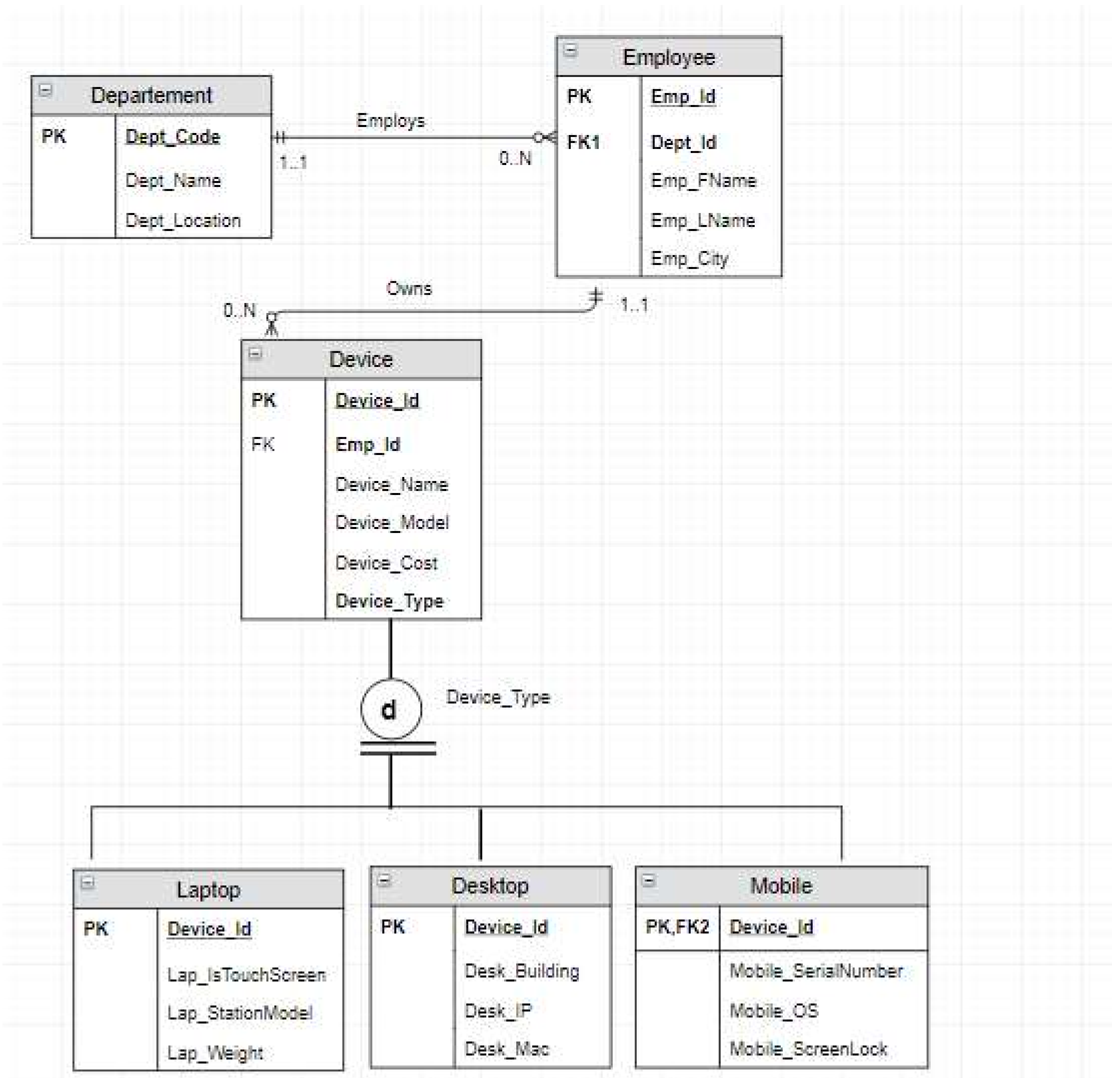
Q1 [2 points]. Draw an ER diagram to model the following scenario. A biotech company contains several departments (eg. R&D, Licensing, Sales...) each of which is in-charge of several projects. There are no projects that are shared between departments. Each employee of the company belongs to a single department, and can possibly work on multiple projects. Each project could have multiple employees. A department has one or more employees (and is in-charge of one or more projects as noted earlier). Some employees are managers, who supervise multiple employees; each employee is supervised by atmost one manager. Some employees are also department managers (they don't manage people), each department has a single manager.
We need to store employees' names and start dates, departments' names and locations, and projects' names and durations. Also, we need to store the hours that employees spend on projects they're on.
In the diagram, indicate connectivity of the form a..b [assume that 'multiple' above is 0..N].

Answer 1 :



Q2 [2 points]. Draw an EER diagram, to model a domain of your choice – ANY domain! The diagram needs to show at least one supertype/subtype hierarchy, with appropriate notations for disjointness constraint and completeness constraint. There needs to be at least two other (non subtype/supertype) relationships in the diagram. Overall, there needs to be at least 6 entities in your diagram. Be sure to indicate connectivities.

Answer 2:



Q3 [0.5 + 0.25 + 0.25 = 1 point]. Consider a ternary relationship between Supplier, Project and Product [with semantics similar to the 'doctor, patient, drug' example from the lecture].

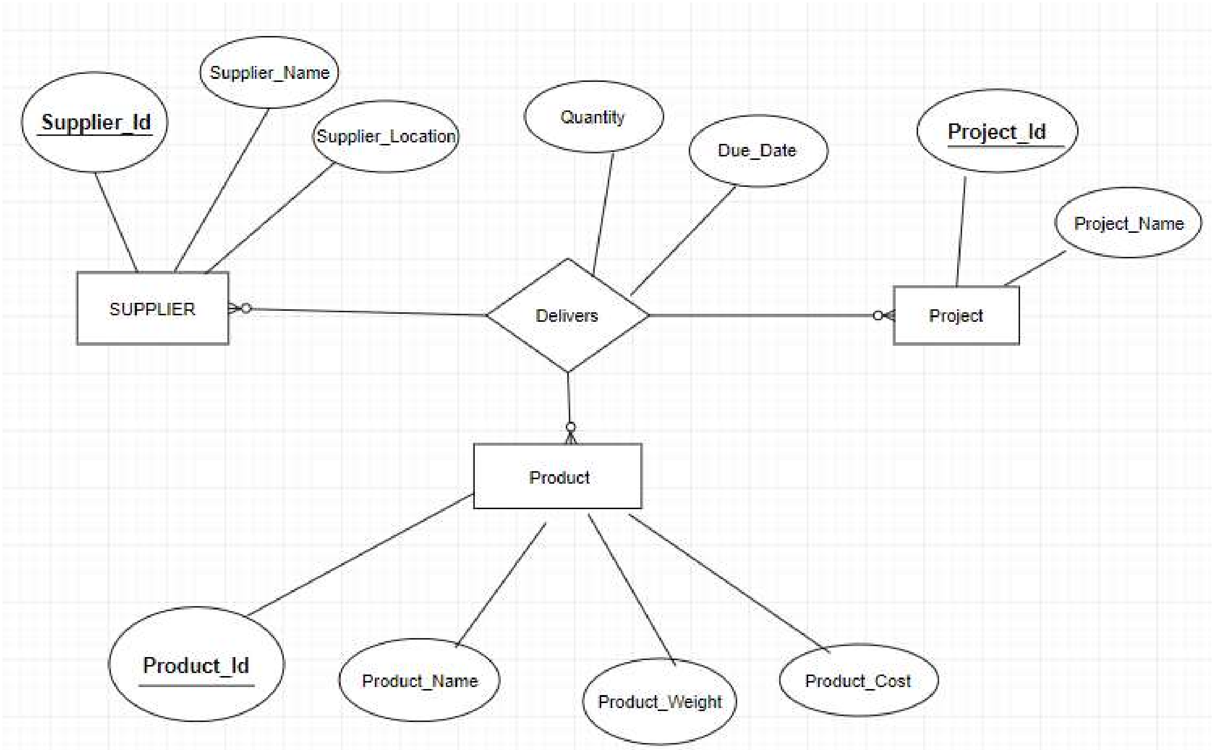
a. Draw a ternary ER diagram to model the relationship, taking into account, the need to store 'Quantity' and 'Due date' for products supplied to projects by suppliers.

b. Model (with an ER diagram) the same scenario, using binary relationships inside of a ternary one.

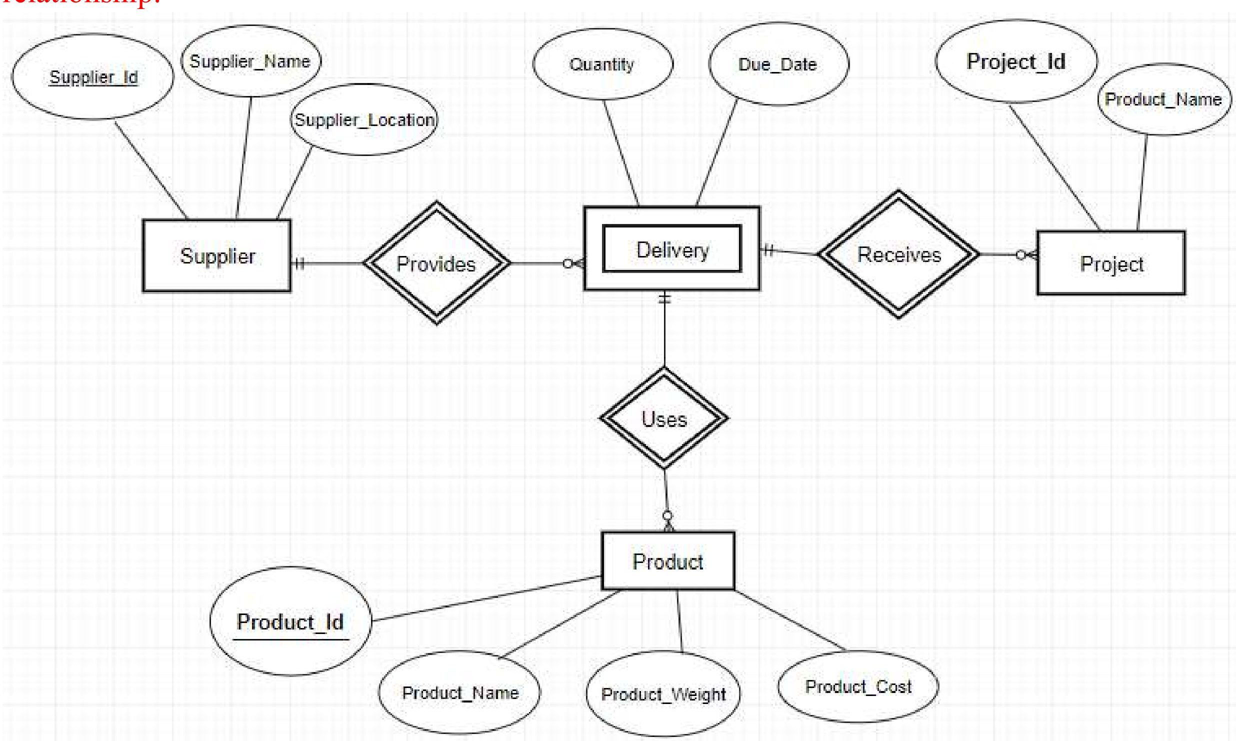
c. What is lost, in doing b.? Think carefully - create sample tables if you need to!

Answer 3:

3a.



3b. Converting the above relationship into weak-relationship, would result into binary relationship.



3c.

In 3a, can only store one relationship between an instance of Supplier , Product and Project due to relation "Delivers"

- (Supplier1, Project1, Product1)

In 3b, Can create many relationships between various entities of Supplier , Product and Project , due to the entity-set "Delivery" like

- (Supplier1, Delivery1) , (Delivery1, Product1), (Project1, Delivery1)
- (Supplier1, Delivery2) , (Delivery2, Product1), (Project2, Delivery1)
- (Supplier1, Delivery3) , (Delivery3, Product1), (Project3, Delivery1)

The Relationship Delivers(Supplier, Project, Product) Supplier, Project -> product
Is decomposed into :

- Provides(Supplier, Delivery)
- Receives(Delivery, Project)
- Has(Delivery, Product)

Using Ternary Relation , this (ref below table) can be stored but there is no way to achieve this functional dependency using Binary Relationship. It's a lossy decomposition.

Supplier	Project	Product
s	p	f1
s	p2	f
s3	p	f

Q4 [1 point]. There exists a ternary relationship between Instructor, CourseSection and Term, when it comes to courses being offered. Also, not all instructors can teach all courses - they need to be qualified. Not all courses are taught all terms. Model this using an ER diagram, again indicating connectivities that can include 0 as a lower bound (like in Q1).

Answer 4: