CASE 1 : Employee Case

1. Answer the following question:
    a. No, because one record cannot store multiple data. There are problems with attributes Proj-id, Proj-start-date, Location, Weeks-on-project. It should make a new table to store multiple record and connect either table using foreign key, the cardinality is one to many
    b. Yes, the attributes are Proj-id, Proj-start-date, Location, Weeks-on-project
    c. Emp-No:
        - Emp-Name
        - Dept
        - Manager

       Dept:
        - Manager

       Proj-id, Weeks-on-Project :
        - Proj-Start-Date
        - Location

       Proj-id:
        - Proj-Start-Date

    d. Decompose the tables

       Employee

Emp-No	Emp-Name	Dept	Manager
5	Smith	Marketing	Jones
7	Bond	Accounts	Bloggs
9	King	Info Systems	Hurne
10	Holt	Accounts	Bloggs

       Project Start Date

Proj-id	Proj-Start-Date
A	12-93
B	06-94
C	06-94
D	06-94

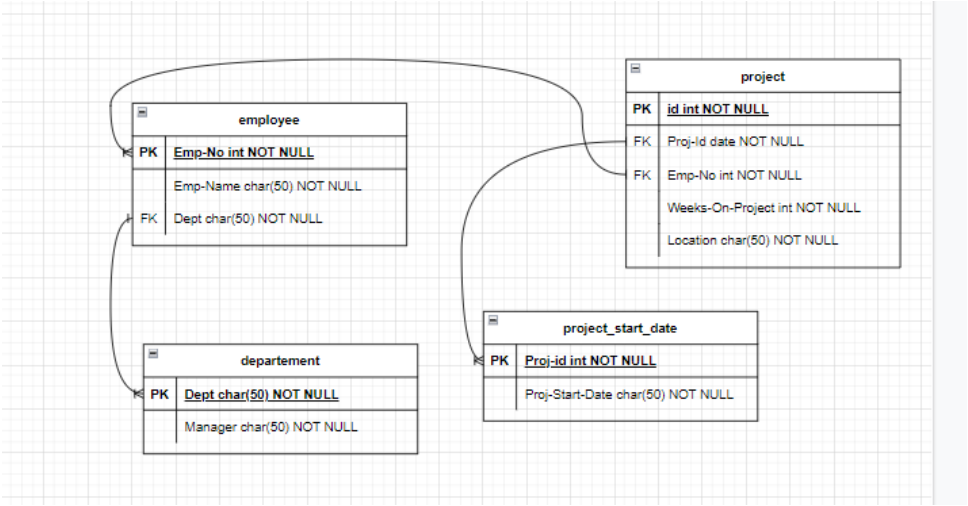
Departemen

Dept	Manager
Marketing	Jones
Accounts	Bloggs
Info Systems	Hurne

Project

Emp-No	Proj-ID	Weeks on Project	Location
5	A	11	Poole
5	B	15	Playmouth
5	C	6	Portsmouth
7	B	3	playMouth
7	D	9	Berlin
9	C	10	Portsmouth
10	A	21	Poole
10	B	10	Belfast
10	D	12	Hamburg

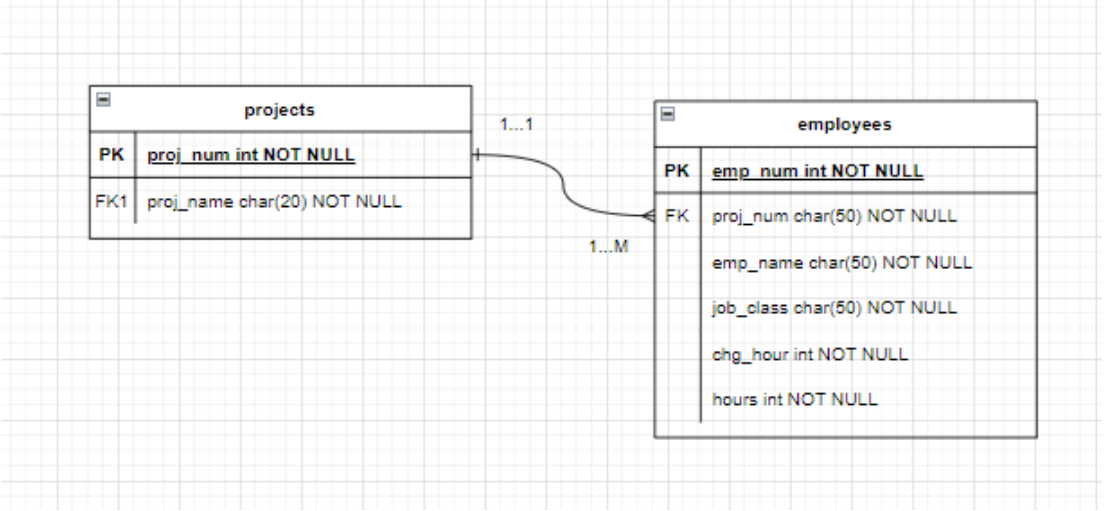
2. Study the new table you've created (as the result from table decomposition). Are they having a good design? Give your reason.
   It should create one table for project information, like this:



3. Apply 1st Normal Form if possible. To do so, remove the repeating group/ multivalued attributes.

4. Apply 2nd Normal Form if possible. Remember that the 2nd Normal Form removes partial dependencies, i.e all non-primary key attribute fully dependent on the primary key (not partially)
5. Apply 3rd Normal Form if possible. Remember that the 3rd Normal Form removes transitive dependencies, i.e remove dependencies on the nonprimary key attribute
6. Now, i have 4 tables

CASE 2 : Project Case



CASE 3 : Library Case