Example: Summarize the Normal Forms by FDs

CS 4750
Database Systems

[C.M. Ricardo and S.D. Urban, "Database Illuminated," Ch.6]

Example

Let's consider the `work_project` table that stores information about projects, assignment, and budget and hour allocation

`work_project`

prjName	prjMgr	empId	hours	eName	budget	startDate	salary	empMgr	empDept	rating
Jupiter	Smith	E101	25	Jones	100000	01/15/19	60000	Levine	10	9
Jupiter	Smith	E105	40	Adams	100000	01/15/19	55000	Jones	12	
Jupiter	Smith	E110	10	Rivera	100000	01/15/19	48000	Levine	10	8
Maxima	Lee	E101	15	Jones	200000	03/15/18	60000	Levine	10	
Maxima	Lee	E110	30	Rivera	200000	03/15/18	48000	Levine	10	
Maxima	Lee	E120	15	Tanaka	200000	03/15/18	45000	Jones	15	

Notice that the table consists of many attributes, some data are redundant.

Assumptions We Made

- Each project has a unique name
- Names of employees are managers are not unique
- Each project has one manager, whose name is stored in `prjMgr`
- Many employees can be assigned to work on each project
- An employee can be assigned to more than one project
- The attribute `hours` tell the number of hours per week a particular employee is assigned to work on a particular project
- `budget` stores the amount budgeted for a project
- `startDate` gives the starting date for a project
- `salary` gives the annual salary of an employee

Assumptions We Made (2)

- `empMgr` gives the name of the employee's manager, who might not be the same as the project manager
- `empDept` gives the employee's department
- Department names are unique
- Each department has only one manager
- The employee's manager is the manager of the employee's department
- `rating` gives the employee's rating for a particular project
- The project manager assigns the rating at the end of the employee's work on that project

List of FDs

Based on the assumptions, we can list the following FDs

```
prjName → prjMrg, budget, startDate
empId → empName, salary, empMgr, empDept
prjName, empId → hours, rating
empDept → empMgr
```

Since we assumed people's names were not unique, `empMgr` does not functionally determine `empDept`.

`prjMgr` does not determine `prjName`.

Is `work_project` Table in 1NF?

`work_project`

prjName	prjMgr	empId	hours	eName	budget	startDate	salary	empMgr	empDept	rating
Jupiter	Smith	E101	25	Jones	100000	01/15/19	60000	Levine	10	9
Jupiter	Smith	E105	40	Adams	100000	01/15/19	55000	Jones	12	
Jupiter	Smith	E110	10	Rivera	100000	01/15/19	48000	Levine	10	8
Maxima	Lee	E101	15	Jones	200000	03/15/18	60000	Levine	10	
Maxima	Lee	E110	30	Rivera	200000	03/15/18	48000	Levine	10	
Maxima	Lee	E120	15	Tanaka	200000	03/15/18	45000	Jones	15	

```
prjName → prjMrg, budget, startDate
empId → empName, salary, empMgr, empDept
prjName, empId → hours, rating
empDept → empMgr
```

Given FDs

Check if there is a column that is not atomic. Fix as needed.

Is `work_project` Table in 2NF?

`work_project`

prjName	prjMgr	empId	hours	eName	budget	startDate	salary	empMgr	empDept	rating
Jupiter	Smith	E101	25	Jones	100000	01/15/19	60000	Levine	10	9
Jupiter	Smith	E105	40	Adams	100000	01/15/19	55000	Jones	12	
Jupiter	Smith	E110	10	Rivera	100000	01/15/19	48000	Levine	10	8
Maxima	Lee	E101	15	Jones	200000	03/15/18	60000	Levine	10	
Maxima	Lee	E110	30	Rivera	200000	03/15/18	48000	Levine	10	
Maxima	Lee	E120	15	Tanaka	200000	03/15/18	45000	Jones	15	

```
prjName → prjMrg, budget, startDate
empId → empName, salary, empMgr, empDept
prjName, empId → hours, rating
empDept → empMgr
```

Given FDs

Check if there is a partial dependency. Fix as needed.

Is `work_project` Table in 3NF?

`work_project`

prjName	prjMgr	empId	hours	eName	budget	startDate	salary	empMgr	empDept	rating
Jupiter	Smith	E101	25	Jones	100000	01/15/19	60000	Levine	10	9
Jupiter	Smith	E105	40	Adams	100000	01/15/19	55000	Jones	12	
Jupiter	Smith	E110	10	Rivera	100000	01/15/19	48000	Levine	10	8
Maxima	Lee	E101	15	Jones	200000	03/15/18	60000	Levine	10	
Maxima	Lee	E110	30	Rivera	200000	03/15/18	48000	Levine	10	
Maxima	Lee	E120	15	Tanaka	200000	03/15/18	45000	Jones	15	

2NF

```
Project(prjName, prjMgr, budget, startDate)
Emp(empId, empName, salary, empMgr, empDept)
Work(prjName, empID, hours, rating)
```

Use the set of tables `Project`, `Emp`, and `Work`. Check if there is a non-key attribute or combination of attribute that functionally determines another non-key attribute.

Check if there is a transitive dependency. Fix as needed.

Sample Data

`Emp`

empId	eName	salary	empDept
E101	Jones	60000	10
E105	Adams	55000	12
E110	Rivera	48000	10
E120	Tanaka	45000	15

`Project`

prjName	prjMgr	budget	startDate
Jupiter	Smith	100000	01/15/19
Maxima	Lee	200000	03/15/18

`Work`

prjName	empId	hours	rating
Jupiter	E101	25	9
Jupiter	E105	40	
Jupiter	E110	10	8
Maxima	E101	15	
Maxima	E110	30	
Maxima	E120	15	

`Department`

empDept	empMgr
10	Levine
12	Jones
15	Jones

Is `work_project` Table in BCNF?

`work_project`

prjName	prjMgr	empId	hours	eName	budget	startDate	salary	empMgr	empDept	rating
Jupiter	Smith	E101	25	Jones	100000	01/15/19	60000	Levine	10	9
Jupiter	Smith	E105	40	Adams	100000	01/15/19	55000	Jones	12	
Jupiter	Smith	E110	10	Rivera	100000	01/15/19	48000	Levine	10	8
Maxima	Lee	E101	15	Jones	200000	03/15/18	60000	Levine	10	
Maxima										
Maxima										

Given FDs

```
prjName → prjMrg, budget, startDate
empId → empName, salary, empMgr, empDept
prjName, empId → hours, rating
empDept → empMgr
```

Check if there is a determinant that is not a superkey. Fix as needed.