Fuad Hassan

1.

* ProjID ---> ProjName: this means that the project ID uniquely determines the project name, so it's a functional dependency.
* EmpID --> EmpName, JobClass, HourlyRate: meaning the employee ID uniquely determines the employee’s name, job class, and hourly rate.
* JobClass --> HourlyRate: suggests that job class uniquely determines the hourly rate, although from the table this doesn't seem to hold true as multiple job classes have different hourly rates for different employees.
* ProjID, EmpID --> Hours: indicates a composite dependency, where the combination of project ID and employee ID determines the hours worked.

2. yes, it is 1NF

3. To normalize a table into the Second Normal Form (2NF), we remove partial dependencies, which means an attribute is dependent on only part of a multi-valued key. To do this, we'd separate the table into different tables where each non-prime attribute is fully functionally dependent on the primary key of its table.

We'd create tables such as Projects (ProjID\*, ProjName), Employees (EmpID\*, EmpName, JobClass, HourlyRate), and ProjectAssignments (ProjID\*, EmpID\*, Hours).

Note that in each table, the primary key is marked with an (\*).

4. To reach the Third Normal Form (3NF), we remove transitive dependencies, where non-prime attributes don't depend on other non-prime attributes. Given that there's a potential transitive dependency with JobClass --> HourlyRate, we might need to create a separate table for job classes if the hourly rate is indeed determined by job class alone. This would result in a JobClasses (JobClass\*, HourlyRate) table, and the Employees table would no longer include the HourlyRate column, which now would come from the JobClasses table.

So, the final 3NF tables might look like:

• Projects (ProjID\*, ProjName)

• Employees (EmpID\*, EmpName, JobClass)

• ProjectAssignments (ProjID\*, EmpID\*, Hours)

• JobClasses (JobClass\*, HourlyRate)