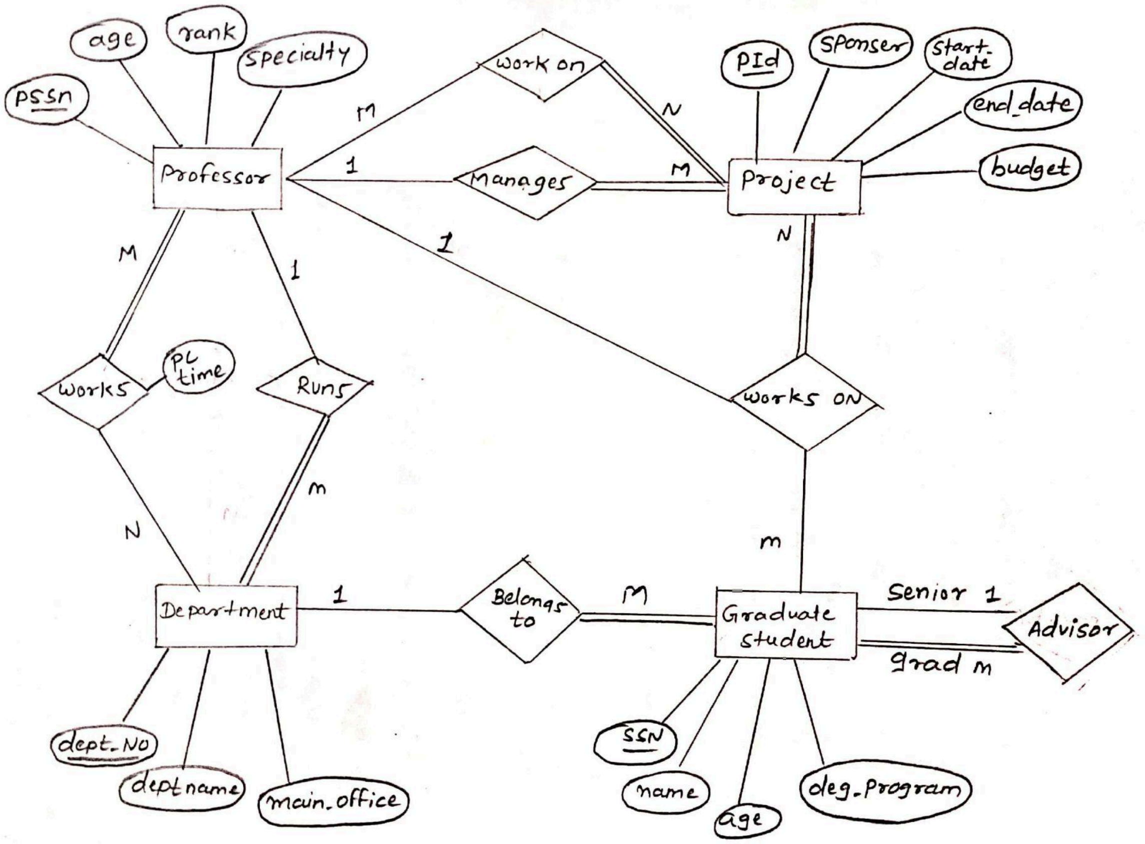
DATABASE SYSTEMS – DIGITAL ASSIGNMENT 1

Submission Due Date – 02-Sep-2024

Name :Jagtap Mahesh
Reg No.24MCS1017

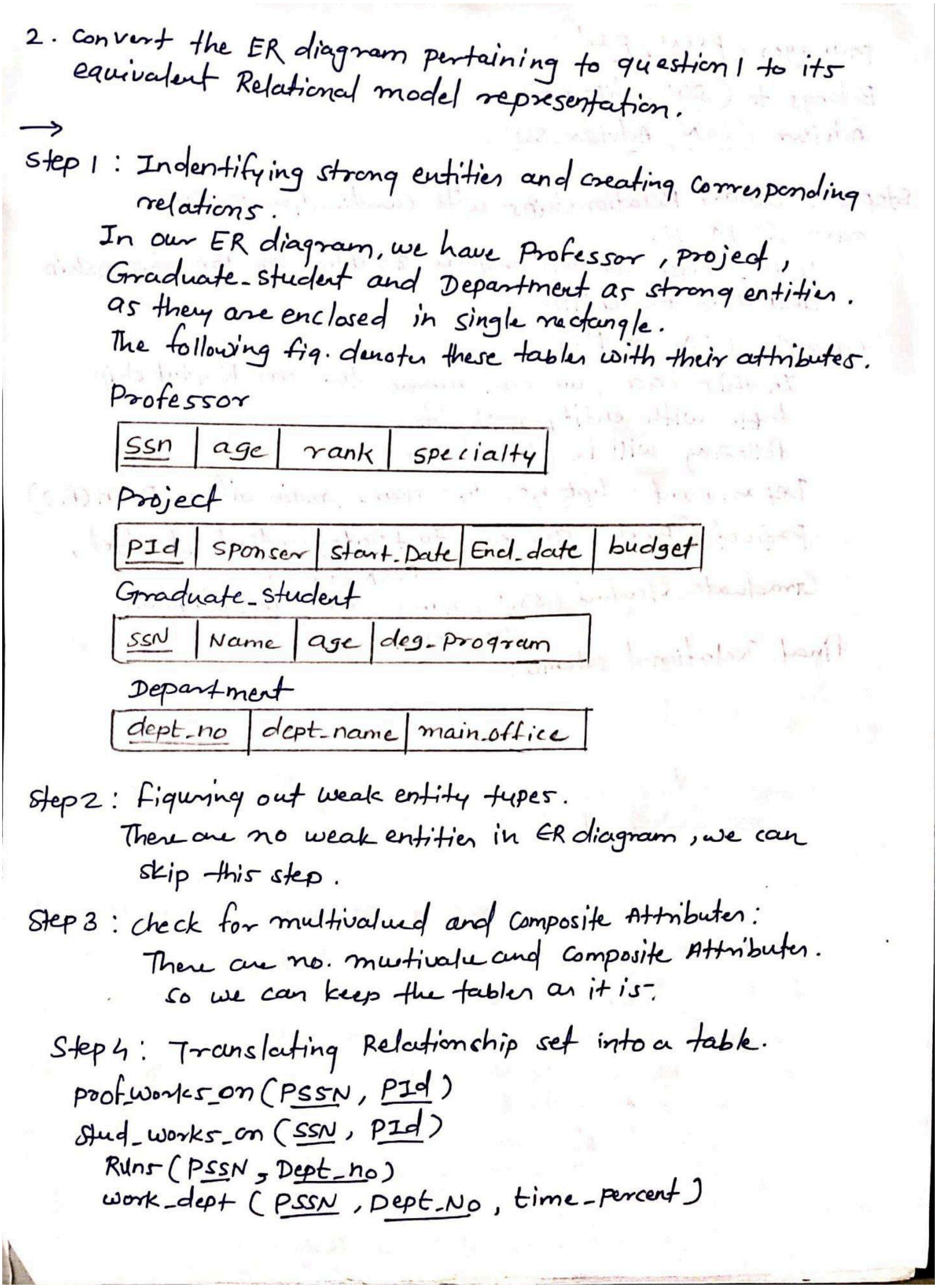
1. Consider the following information about a university database: [10 Marks]
   - Professors have an SSN, a name, an age, a rank, and a research specialty.
   - Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget.
   - Graduate students have an SSN, a name, an age, and a degree program (e.g., B.E, M.E, M.S. or Ph.D.).
   - Each project is managed by one professor (known as the project's principal investigator).
   - Each project is worked on by one or more professors (known as the project's co-investigators).
   - Professors can manage and/or work on multiple projects.
   - Each project is worked on by one or more graduate students (known as the project's research assistants).
   - When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.
   - Departments have a department number, a department name, and a main office. II Departments have a professor (known as the chairman) who runs the department.
   - Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.
   - Graduate students have one major department in which they are working on their degree
   - Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take.

Design and draw an ER diagram that captures the information about the university. Use only the basic ER model here; that is, entities, relationships, and attributes. Be sure to indicate any key and participation constraints.



University Management System ER Diagram

2. Convert the ER diagram pertaining to question 1 to its equivalent Relational model representation. [10 Marks]



2. Convert the ER diagram pertaining to question 1 to its equivalent Relational model representation.

→

step 1 : Indentifying strong entities and creating corresponding relations.

In our ER diagram, we have Professor, project, Graduate-student and Department as strong entities. as they are enclosed in single rectangle.
The following fig. denotes these tables with their attributes.

Professor

SSN	age	rank	specialty

Project

PId	sponser	start-Date	End-date	budget

Graduate_student

SSN	Name	age	deg-program

Department

dept_no	dept_name	main.office

step 2 : figuring out weak entity types.

There are no weak entities in ER diagram, we can skip this step.

Step 3 : check for multivalued and Composite Attributes:

There are no. mutivalue and composite Attributes. so we can keep the tables as it is.

Step 4 : Translating Relationship set into a table.

proof_works_on (PSSN, PId)

Stud_works_on (SSN, PId)

Runs (PSSN, Dept_no)

work_dept (PSSN, Dept_No, time-percent)

Manager (PSSN, PId)

Belongs_to (SSN, dept_no)

Adviser (SSN, Adviser_SSN)

Step 5: Binary Relationships with cardinality Ratios.

case-① M : N.

In this case we will require 3 tables. one for relationship and two for entities.

case ② 1 : N or N : 1

In this case, we can merge the relation~~tab~~ship table with entity on N side.

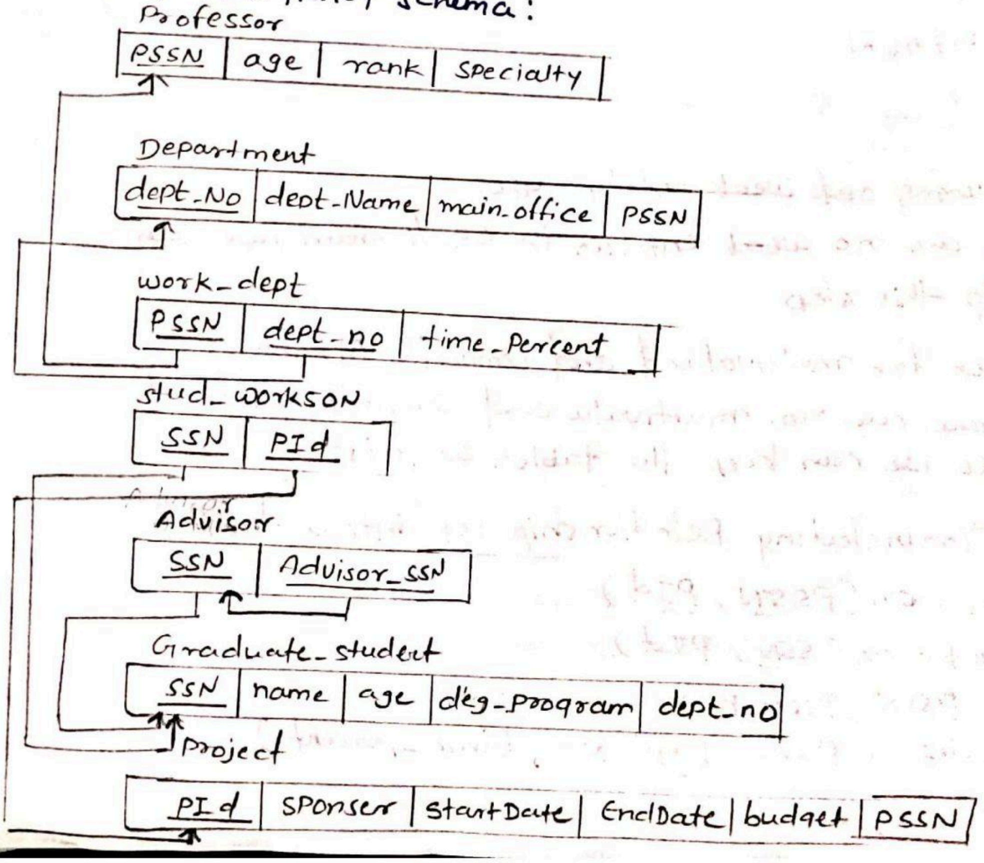
following will be the changes:

Department (dept-no, dept-name, main_office, PSSN (fk))

Project (PId, sponser, StartDate, EndDate, budget, PSSN (fk))

Graduate_student (SSN, name, age, deg-program, dept_no)

final Relational schema:



Professor

PSSN	age	rank	Specialty

Department

dept-No	dept-Name	main-office	PSSN

work_dept

PSSN	dept_no	time_Percent

stud_WORKSON

SSN	PId

Advisor

SSN	Advisor_SSN

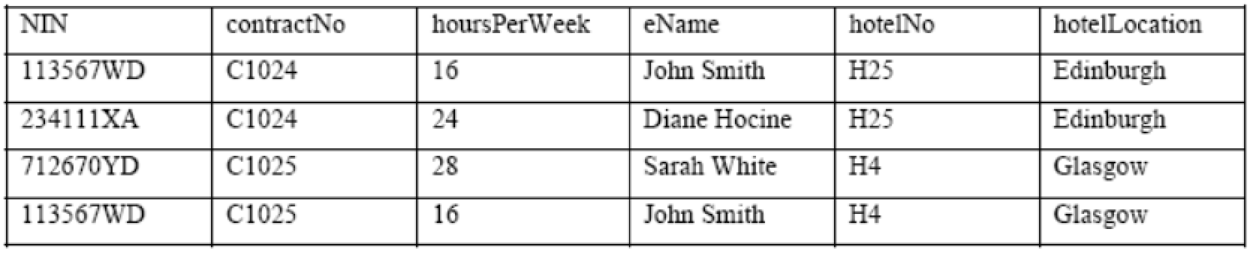
Graduate_student

SSN	name	age	deg-Program	dept_no

Project

PId	SPonser	StartDate	EndDate	budget	PSSN

3. An agency called XYZ supplies part-time/temporary staff to hotels throughout the country ABC. The table shown below lists the time spent by agency staff working at two hotels. The National Insurance Number (NIN) is unique for employee.



NIN	contractNo	hoursPerWeek	eName	hotelNo	hotelLocation
113567WD	C1024	16	John Smith	H25	Edinburgh
234111XA	C1024	24	Diane Hocine	H25	Edinburgh
712670YD	C1025	28	Sarah White	H4	Glasgow
113567WD	C1025	16	John Smith	H4	Glasgow

a. Provide examples for different types of anomalies. [3 Marks]

b. Describe and illustrate the process of normalizing the table shown in the figure up to 3NF. State any assumptions you make about the data shown in this table. [7 Marks]

Q.3

a) → The table given is susceptible to following anomalies:

i) Insertion anomalies:-

If we want to add the details of new hotel but currently have no employees assigned to that hotel, we cannot add the hotel details without an employee record as there cannot null values for primary keys. This leads to insertion anomaly.

ii) Update anomaly:

suppose the location of 'H25' is changed from 'Edinburgh' to 'Glasgow'. In this case, all rows containing hotelNo. 'H25' need to be updated. If even one row is missed, it will result in inconsistent data. Similarly, if we want to edit staff details, several records needs to be updated.

iii) Deletion anomalies:

If we want to only delete staff details, the details about contract and hotel like contractNo, hotellocation & hotelNo. will also gets deleted.

Similarly, when we delete a contract No, other staff details will also be lost. In this way deletion anomalies will make the data inconsistent.

b) → Normalizing the table upto 3NF.

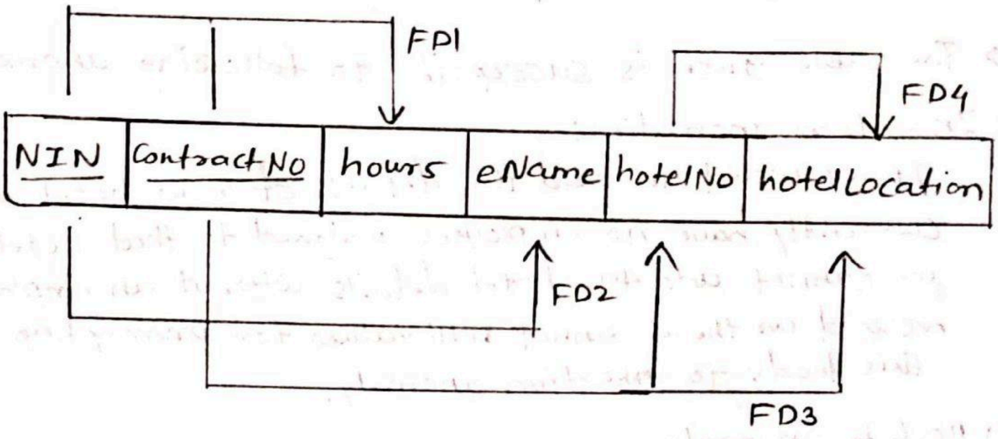
=

i) First Normal form (1NF):

1NF requires that the table only contains atomic values & each entry in the table is unique.

In the provided table, the data already satisfies 1NF.

Then we assign primary keys and figure out functional dependencies, represented, below:



ii) Second Normal form :-

To achieve 2NF, table must satisfy 1NF, and all non-key attributes must be fully functionally dependent on the Primary key. This means that there should not be any partial dependency.

FD2 & FD3 voilate the 2NF(partial dependeny).
we can remove this partial dependeny by splitting it into new tables as shown below:

NIN	ContractNO	hours

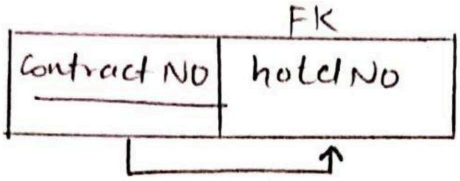
NIN	eName

ContractNo	hotelNo	hotelLocation

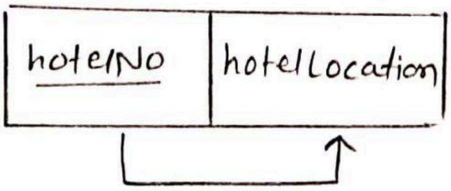
iii) Third Normal form (3NF)

To achieve 3NF, the table must satisfy 2NF and all attributes must be functionally dependent on the primary key, with no transitive dependeny.

FD4 voilates the transitive dependeny.
we will create a seperate table for hotels to remove this transitive dependeny.



	FK
Contract NO	hotelNO



hotelNO	hotelLocation

Finally, rearranging tables and assigning names:

NIN	eName

staff (NIN, eName)

hotelNO	hotelLocation

Hotel (hotelNO, hotelLocation)

	FK
Contract NO	hotelNO

Contract (contractNo, hotelNo)

FK	FK	
NIN	ContractNO	hours

workhours (NIN, contractNO, hours)