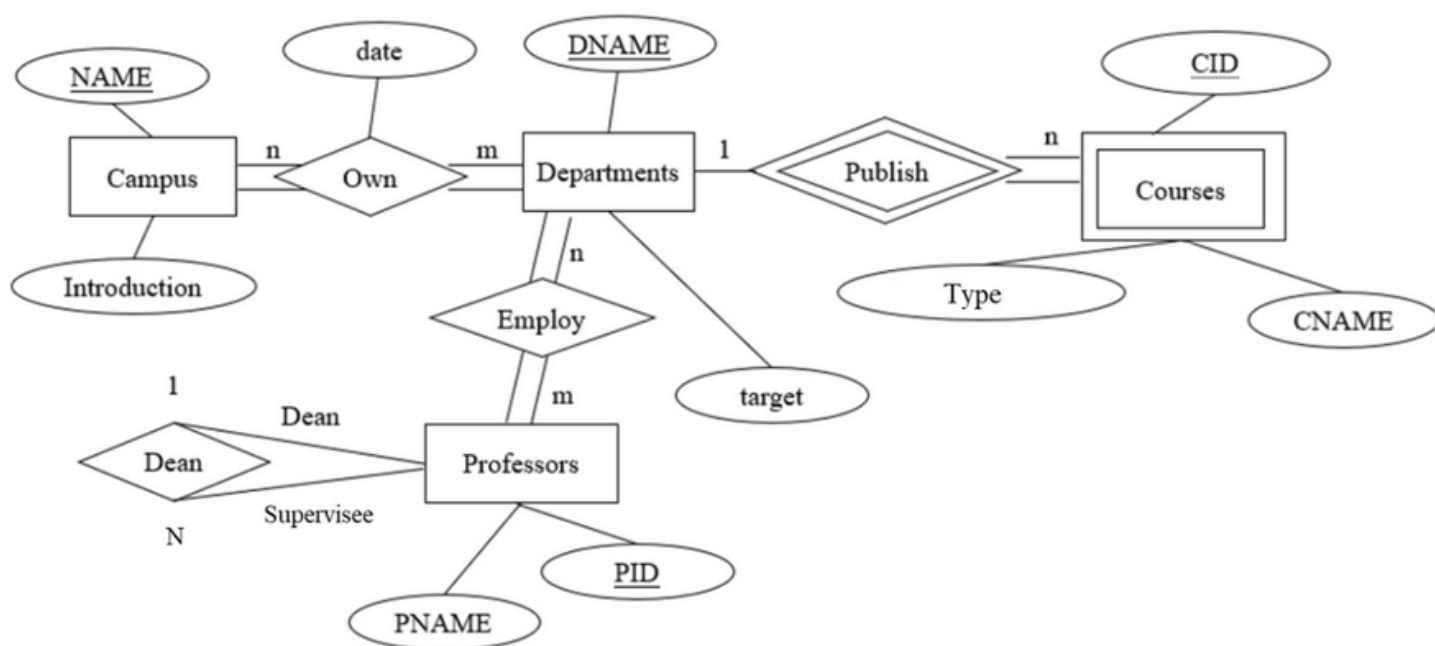
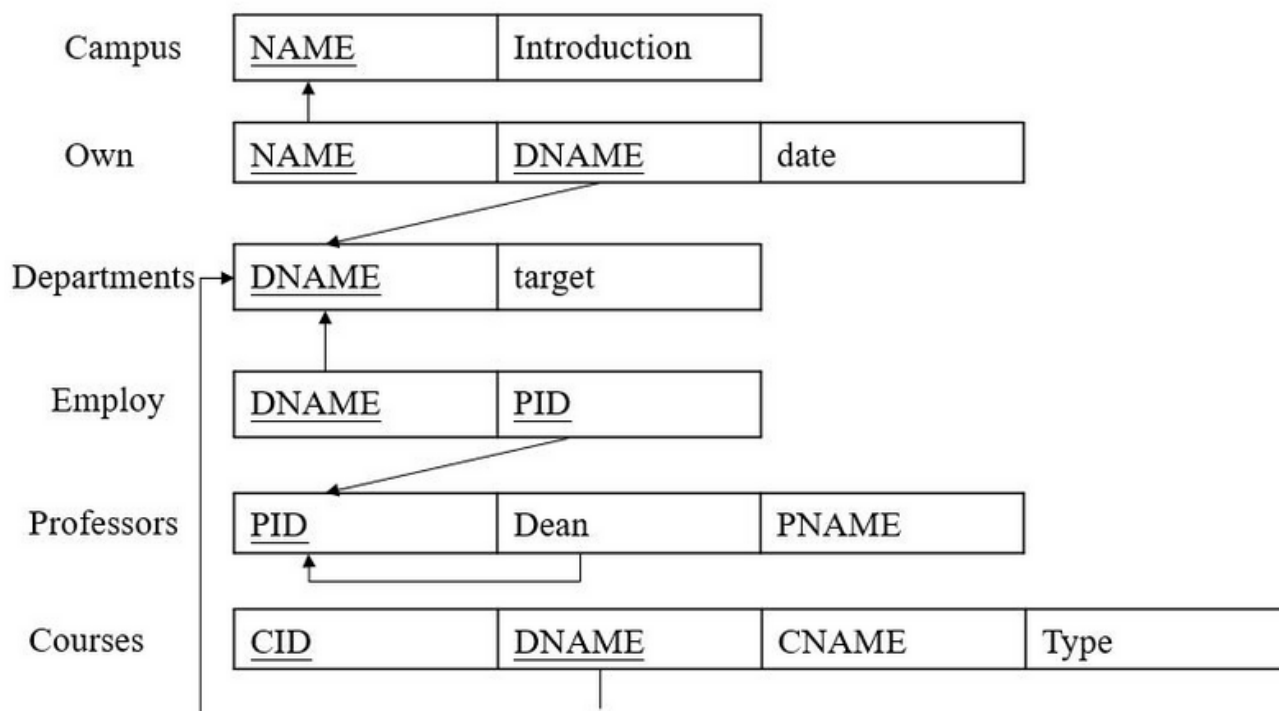


Please convert the following completed ER diagram into Relation Model. (12 points)



Your Answer:



Suppose we have a relational database containing three tables Student(SSN, Name, Email, Sex, DeptName), Department(DeptName, Address, Phone, College\_ID, DeptHead, Head\_ID) and Employee(Emp\_ID, Name, BirthData, Phone). The current state of the database is shown in the following tables. (14 points)

**Student**

SSN	Name	Sex	DeptName
004-73-2862	Otis West	M	ME
141-45-2946	Flynn Crane	F	MATH
056-38-9627	Michael Scott	M	MKT
092-99-0173	Otis West	F	MKT
654-58-6672	Kelly Kapoor	F	CS
758-40-0263	Celia Hamer	M	ME
164-20-5327	Josh Porter	F	LT
565-61-4682	Flynn Crane	M	MATH
041-17-4051	Jim Halpert	M	LT

Department

DeptName	Address	College_ID	DeptHead	Head_ID
CS	20 Joy Ridge Lane Maumee, OH	CENG	Alicja Weir	2
ME	20 Joy Ridge Lane Maumee, OH	CENG	Alicja Weir	2
MKT	33 New Saddle Court Lapeer, MI	CB	Vernon Robin	3
MATH	944 E. Ocean St. Trenton, NJ	CSCI	Beauden Lang	1
LT	156 Thorne St. Winter Springs, FL	CLASS	Khadijah Holman	4

Employee

Emp_ID	Name	BirthDate
2	Alicja Weir	1969-09-05
4	Lackawana Country	1958-02-19
5	Alicja Weir	1964-03-15
3	Vernon Robin	1958-02-19
1	Beauden Lang	1973-07-22
8	Manahil Rodriquez	1967-11-17

(1) Based on the application, list the primary key and the foreign key(s) for each relation. [4 points]

(2) For the questions below, suppose each of the following Update operations is applied directly to the database. Discuss all integrity constraints violated by each operation, if any, and the different ways of enforcing these constraints.

a. Insert <'296-67-9428','Peter Chan','M','EE' > into Student [5 points]

b. Update Emp\_ID of Employee tuple with Emp\_ID=1 to 3. [5 points]

(1)

For Student table:

Primary key: SSN

For Department table:

Primary key: DeptName

For Employee table:

Primary key: Emp\_ID

Foreign key: student.DeptName, department.Head\_ID

(2)

a.

Violates both referential integrity constraint and domain constraint. Violates referential integrity because DeptName is foreign key, and there is no tuple in the Department relation with DeptName='EE'. Violates domain constraint because the domain of Sex should not be integer. (3 points)

We may enforce the constraint by: (i) rejecting the insertion, (ii) changing the value of DeptName to an existing value in Department, or (iii) inserting a new Department tuple with DeptName='EE'. Meanwhile changing the Sex into char type. (2 points)

b.

Violates the primary key constraint and the referential integrity constraint. Violates the key constraint because there already exists an Employee tuple with Emp\_ID=3. Violates the referential integrity constraint because once the table is updated, no corresponding referenced tuple could be found for the tuple of Department with Head\_ID=1. (3 points)

We may enforce the constraint by: (i) rejecting the modifying operation, or (ii) changing the value of Emp\_ID to a value that is not null and doesn't exist in the table Employee. At the same time, we need to change the Head\_ID of Department to a value that exists in Employee table. (2 points)

Given the following relations about the information of courses in a university.

Student (**StudentID**: integer, **Name**: string, **Age**: integer, **Gender**: string)

Course (**CourseID**: integer, **Name**: string, **TeacherID**: integer)

Teacher (**TeacherID**: integer, **Name**: string)

Grade (**StudentID**: integer, **CourseID**: integer, **Score**: integer)

Suppose now we have a valid database state. Write the following queries in SQL.

1 Retrieve name of courses which are not taken by any student. (6 points)

2 Count the number of courses taught by each teacher. Order the results in the descending order of teacher's ID. (6 points)

Your Answer:

1.

select Name

from Course

where NOT EXISTS ( select \*

from Grade

where Course.CourseID= Grade.CourseID);

2.

SELECT Teacher.TeacherID, COUNT(\*)

FROM Teacher, Course,

WHERE Teacher.TeacherID= Course.TeacherID

GROUP BY Teacher.TeacherID;

ORDER BY Teacher.TeacherID DESC;

**OR**

SELECT Course.TeacherID, COUNT(\*)

FROM Course,

GROUP BY Course.TeacherID;

ORDER BY Course.TeacherID DESC;