

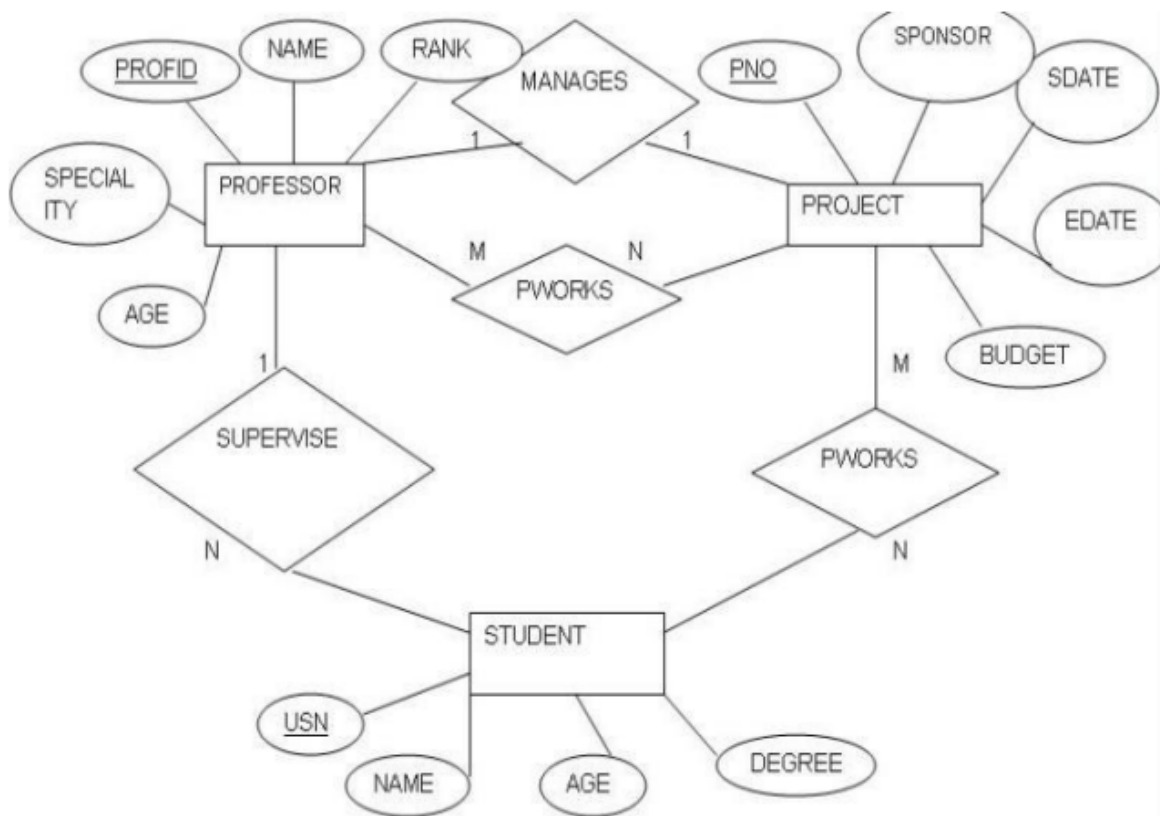
Reference:

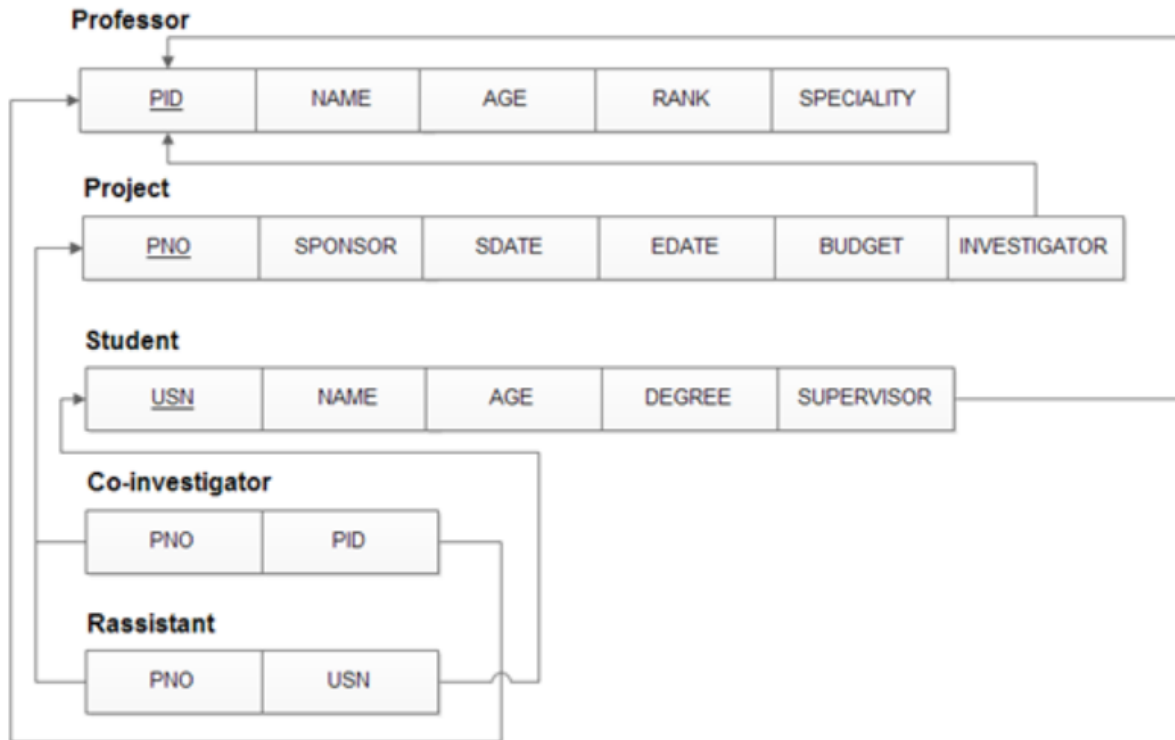
<https://studyittoday.blogspot.com/2014/07/dbms-lab2-mysql.html>

Exercise 1

6

Professors have a PROFID, a name, an age, a rank, and a research specialty. Projects have a project number, a sponsor name (e.g. UGC/AICTE/...), a starting date, an ending date, and a budget. Graduate students have an USN, a name, an age, and a degree program (e.g. MCA/MPhil/BE/ME ..). Each project is managed exactly 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/work on multiple projects. Each project is worked on by one or more graduate students (known as the project's research assistants). Graduate students can work on multiple projects. Each professor can supervise many students. A student who is working on a project can be supervised by only one professor.





1.create table professor(profid int(3) primary key ,name varchar(20),age int(3), rank\_val int(3), research varchar(20));

2) create table projects (pno int(3) primary key,sponsor varchar(10), sdate date, edate date, budget int(5), p\_investigator int(3) references professor(profid));

3) create table student(usn int(5) primary key,sname varchar(20),age int(3),degree varchar(10), profid int(3) references professor(profid));

4)create table pworkson(profid int(3) references professor(profid), pno int(3) references projects(pno), primary key(profid,pno))

5)create table sworkson(usn int(5) references student(usn),pno int(3) references projects(pno), primary key(usn,pno));

```
CREATE TABLE professor (
->  profid INT(3) PRIMARY KEY,
->  name VARCHAR(20),
->  age INT(3),
```

```
-> rank_val INT(3),  
-> research VARCHAR(20)  
-> );
```

```
mysql> CREATE TABLE projects (  
-> pno INT(3) PRIMARY KEY,  
-> sponsor VARCHAR(10),  
-> sdate DATE,  
-> edate DATE,  
-> budget INT(5),  
-> p_investigator INT(3),  
-> FOREIGN KEY (p_investigator) REFERENCES professor(profid)  
-> );
```

```
mysql> CREATE TABLE student (  
-> usn INT(5) PRIMARY KEY,  
-> sname VARCHAR(20),  
-> age INT(3),  
-> degree VARCHAR(10),  
-> profid INT(3),  
-> FOREIGN KEY (profid) REFERENCES professor(profid)  
-> );
```

Query OK, 0 rows affected, 3 warnings (1.04 sec)

```
mysql> CREATE TABLE pworkson (  
-> profid INT(3),  
-> pno INT(3),  
-> PRIMARY KEY (profid, pno),  
-> FOREIGN KEY (profid) REFERENCES professor(profid),  
-> FOREIGN KEY (pno) REFERENCES projects(pno)  
-> );
```

Query OK, 0 rows affected, 2 warnings (0.84 sec)

```
mysql> CREATE TABLE sworkson (  
-> usn INT(5),  
-> pno INT(3),
```

```
-> PRIMARY KEY (usn, pno),  
-> FOREIGN KEY (usn) REFERENCES student(usn),  
-> FOREIGN KEY (pno) REFERENCES projects(pno)  
-> );
```

Query OK, 0 rows affected, 2 warnings (1.18 sec)

-- Insert queries to populate data for professors

```
INSERT INTO professor (profid, name, age, rank_val, research) VALUES (1, 'Professor  
A', 40, 1, 'AI');  
INSERT INTO professor (profid, name, age, rank_val, research) VALUES (2, 'Professor  
B', 45, 2, 'Machine Learning');  
INSERT INTO professor (profid, name, age, rank_val, research) VALUES (3, 'Professor  
C', 38, 3, 'Data Science');
```

-- Insert queries to populate data for projects

```
INSERT INTO projects (pno, sponsor, sdate, edate, budget, p_investigator) VALUES (4,  
'UGC', '2023-01-01', '2023-12-31', 150000, 1);  
INSERT INTO projects (pno, sponsor, sdate, edate, budget, p_investigator) VALUES (5,  
'AICTE', '2023-06-01', '2024-05-31', 120000, 2);  
  
INSERT INTO projects (pno, sponsor, sdate, edate, budget, p_investigator) VALUES (1,  
'UGC', '2006-01-01', '2008-12-31', 120000, 1);  
INSERT INTO projects (pno, sponsor, sdate, edate, budget, p_investigator) VALUES (2,  
'AICTE', '2008-05-01', '2011-04-30', 150000, 2);  
INSERT INTO projects (pno, sponsor, sdate, edate, budget, p_investigator) VALUES (3,  
'UGC', '2007-03-01', '2010-02-28', 90000, 3);
```

-- Insert queries to populate data for graduate students

```
INSERT INTO student (usn, sname, age, degree, profid) VALUES (1001, 'Student X', 25,  
'MCA', 1);  
INSERT INTO student (usn, sname, age, degree, profid) VALUES (1002, 'Student Y', 24,  
'MPhil', 2);
```

```
INSERT INTO student (usn, sname, age, degree, profid) VALUES (1003, 'Student Z', 26, 'BE', 3);
```

-- Insert queries to populate data for professor working on projects

```
INSERT INTO pworkson (profid, pno) VALUES (1, 1);
INSERT INTO pworkson (profid, pno) VALUES (2, 2);
INSERT INTO pworkson (profid, pno) VALUES (3, 3);
INSERT INTO pworkson (profid, pno) VALUES (1, 4);
INSERT INTO pworkson (profid, pno) VALUES (2, 5);
INSERT INTO pworkson (profid, pno) VALUES (3, 4);
INSERT INTO pworkson (profid, pno) VALUES (3, 5);
INSERT INTO pworkson (profid, pno) VALUES (3, 1);
INSERT INTO pworkson (profid, pno) VALUES (3, 2);
```

-- Insert queries to populate data for students working on projects

```
Insert into sworkson values (1001, 1), (1002, 2), (1001, 2), (1003, 1), (1002, 1), (1002, 4),
(1003, 5), (1003, 3);
```

### Queries

a) Retrieve the names of all professors who do not have an ongoing project of more than 1 lakh.

```
SELECT DISTINCT p.name FROM professor p LEFT JOIN pworkson pw ON p.profid = pw.profid LEFT JOIN projects pj ON pw.pno = pj.pno WHERE (pj.budget > 100000 OR pj.budget IS NULL) AND (pj.edate < CURDATE() OR pj.edate IS NULL);
```

```
mysql> select p.name from professor p,project pr where pr.budget>100000 and pr.investigator=p.pid;
```

b) Retrieve the names of all graduate students along with their professors under whom they work and project sponsor.

```
SELECT DISTINCT s.sname AS student_name, p.name AS professor_name, pr.sponsor FROM student s
```

```
JOIN professor p ON s.profid = p.profid  
JOIN projects pr ON s.profid = pr.p_investigator;
```

```
mysql> select s.name,p.name,pr.sponsor from student s,professor p,project pr where  
p.pid=s.supervisor and p.pid=pr.investigator;
```

c) List the professors and sum of the budget of their projects started after 2005 but ended in 2010.

```
SELECT p.name, SUM(pr.budget)  
FROM professor p  
JOIN projects pr ON p.profid = pr.p_investigator  
WHERE pr.sdate >= '2005-01-01' AND pr.edate <= '2010-04-10'  
GROUP BY p.profid;
```

```
mysql> select p.name,sum(distinct (budget)) from professor p,project where  
sdate>="2005-01-01" and edate<="2010-04-10" and p.pid=investigator group by p.pid;
```

d) List the names of professors who has a total worth of project greater than the average budget of projects sanctioned

```
SELECT DISTINCT p.name FROM professor p JOIN projects pr ON p.profid =  
pr.p_investigator WHERE pr.budget > (SELECT AVG(budget) FROM projects);
```

```
mysql> select distinct p.name from professor p,project pr where pr.budget > (select  
avg(budget) from project) and p.pid=pr.investigator;
```

e) List the professors who work on all the projects.

```
SELECT p.name
```

```
FROM professor p
WHERE (
    SELECT COUNT(DISTINCT pno)
    FROM projects
) = (
    SELECT COUNT(DISTINCT pno)
    FROM workson pw
    WHERE pw.profid = p.profid
);
```

```
mysql> select distinct p.name from professor p, project pr where p.pid=pr.investigator;
```