Introduction to Database System – Assignment 1

Student Name: Pei-chi, Huang

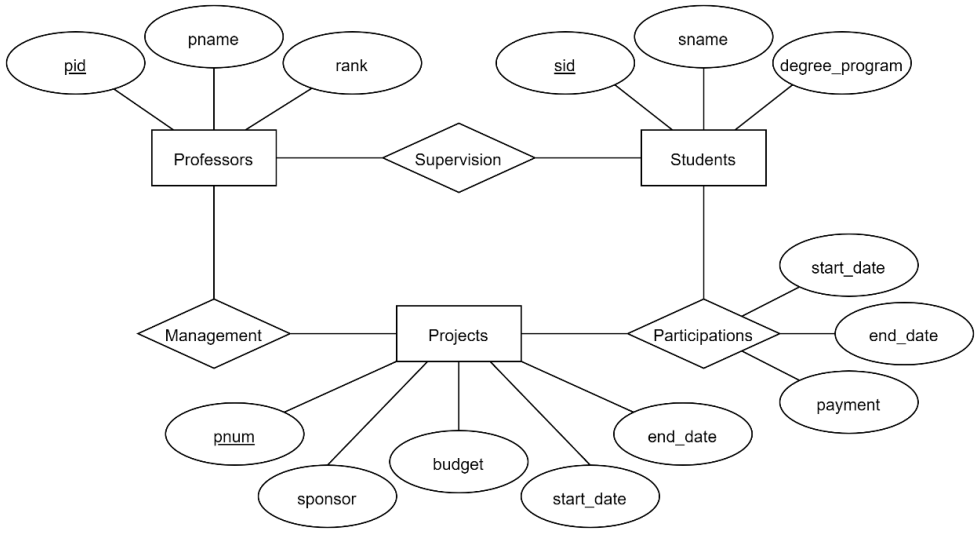
Student ID: 108020017

Department: Interdisciplinary Program of Science

---

**Problem 1**

ER Model -




Relational Model -

```
CREATE TABLE Professor (
    pid             INTEGER,        -- professor id
    pname           VARCHAR(20),    -- professor name
    rank            INTEGER,
    PRIMARY KEY (pid)
);

CREATE TABLE Student (
    sid             INTEGER,        -- student id
    pid             INTEGER,        -- id of professor who supervise the student
    sname           VARCHAR(20),    -- student name
    degree_program VARCHAR(50),
    PRIMARY KEY (sid),
    FOREIGN KEY (pid) REFERENCES Professor(pid)
);

CREATE TABLE Project (
    pnum            INTEGER,        -- project number
    sponsor         VARCHAR(20),    -- sponsor name
    budget          INTEGER,
    start_date      Date,
    end_date        Date,
    pid             VARCHAR(20),    -- id of professor who manage the project
    PRIMARY KEY (pnum),
    FOREIGN KEY (pid) REFERENCES Professor(pid)
);

CREATE TABLE Participation (
    sid             INTEGER,        -- student id
    pnum            INTEGER,        -- project num
    start_date      Date,           -- when the student starts working
    end_date        Date,           -- when the student stops working
    payment         INTEGER,        -- how much the student is paid
    PRIMARY KEY (sid, pnum),
    FOREIGN KEY (sid) REFERENCES Student(sid),
    FOREIGN KEY (pnum) REFERENCES Project(pnum)
);
```

**Problem 2 – Decomposition**

Table 1 (Primary Key: `forum_name`)

| forum_name | popularity |
|---|---|
| Gossiping | 100 |
| Joke | 23 |

Table 2 (Primary Key: {`forum_name, post_id`})

| forum name | post_id | title | article | reply |
|---|---|---|---|---|
| Gossiping | 131 | Girlfriend | How can I get girlfriend? | ["Haha", "I don't know"] |
| Gossiping | 252 | Friends | I don't have a friend... | ["Haha", "I can be", "QQ"] |
| Joke | 46 | Knock | Knock! Knock! ... | ["Then?", "What's the point ?"] |
| Joke | 151 | Santa Claus | Hold! Hold! Hold! | ["XDD"] |

It suffices to prove that for every functional dependency X -> Y, X is a super key or Y is prime attribute.

For the functional dependency "`forum_name -> popularity`" in table 1, `forum_name` uniquely identifies a tuple, so `forum_name` is a super key and can be selected to be the primary key of table 1.

For the functional dependency "{`forum_name, post_id`} -> {`title, article, reply`}" in table 2, {`forum_name, post_id`} uniquely identifies a tuple, so {`forum_name, post_id`} is a super key and can be selected to be the primary key of table 2.

Hence, table 1 and table 2 follow the 3rd normal form.