CS 4347 — Database Systems, Homework 4

https://iniyanjoseph.com | iniyanijoseph@gmail.com

Initial Setup:

```
CREATE TABLE STUDENT (
  snum int,
  sname varchar(255),
  major varchar(255),
  standing varchar(255),
  age int,
  PRIMARY KEY (snum)
);

CREATE TABLE FACULTY (
  fid int,
  fname varchar(255),
  deptid int,
  PRIMARY KEY (fid)
);

CREATE TABLE COURSE (
  cname varchar(255),
  meets_at varchar(255),
  room int,
  fid int,
  PRIMARY KEY (cname)
);
```

Assume that the Student, Faculty, and Course tables have been created, write a SQL create statement to create the Enrolled relation, and choose the appropriate action (reject, cascade, set to NULL, set to default) for each referential integrity constraint on this relation. Justify your action selections. Attached is a sample snapshot with Student, Faculty, and Course tables.

COMMAND:

```
CREATE TABLE Enrolled (
  snum int,
  cname varchar(255),
  PRIMARY KEY (snum, cname),
  FOREIGN KEY (snum)
    REFERENCES STUDENT(snum)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (cname)
    REFERENCES COURSE(cname)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

JUSTIFICATION:

Every student is uniquely identified by snum, and every course is uniquely identified by its cname. Since enrollment is a relationship, a new table can be used to represent the enrollment into courses, with the snum and cname as the primary key. If a student is removed, they cannot be enrolled in courses, so the constraint behaviour is to CASCADE. Similarly, if the course is

removed, no student can be enrolled in that course, thus the constraint behaviour is also to CASCADE. Similar logic holds for the constraint behaviour for UPDATE.

Write an SQL command that would remove every class that has no students enrolled in it

```
DELETE FROM COURSE AS c
WHERE NOT EXISTS(
  SELECT 1
  FROM Enrolled AS e
  WHERE e.cname = c.cname
);
```

Write each of the following queries in SQL

List the name of each class and the name of its instructor

```
SELECT DISTINCT c.cname, f.fname
FROM COURSE AS c, FACULTY as f
WHERE c.fid = f.fid;
```

List the names of classes in which they have CS majors and who are of 'JR' standing

```
SELECT DISTINCT c.cname
FROM COURSE AS c
WHERE EXISTS (
  SELECT 1
  FROM Enrolled AS e
  WHERE e.cname = c.cname AND EXISTS (
    SELECT 1
    FROM STUDENT as s
    WHERE s.snum = e.snum AND s.standing = 'JR' AND s.major = 'CS'
  )
);
```

List the names and majors of students who have not enrolled in any class with instructors working in department 32

```
SELECT DISTINCT s.sname, s.major
FROM STUDENT as s
WHERE NOT EXISTS (
  SELECT 1
  FROM COURSE as c, FACULTY as f
  WHERE f.deptid = 32 AND c.fid = f.fid AND EXISTS (
    SELECT 1
    FROM ENROLLED as e
    WHERE e.snum = s.snum AND e.cname = c.cname
  )
);
```