Database Systems — CSci 4380
Final Exam Data Model

In this database, we will store information about a university offering all of its courses online. The database stores detailed information about the courses including which online sites are used as a resource for different components of courses. Additionally, it stores information about exam dates, office hours etc. For each attribute, example values are provided.

Note: Each class in the database may have zero to many of each of the following: instructors, sites for resources, office hours, class meetings, exams and students. Example date formatting: 2020/11/05.

```sql
-- All courses offered in the university
CREATE TABLE courses (
      classcode       varchar(40) PRIMARY KEY -- ex: CSCI-4380
    , coursename      varchar(200) -- ex: Database Systems
    , credits         int          -- ex: 4
    , department      varchar(40)  -- ex: Computer Science
) ;


-- Classes are offerings of a specific course in a specific semester,
-- year and section. We will assume no cross listed courses for simplicity.
CREATE TABLE classes(
      , crn            int   PRIMARY KEY
    , classcode        varchar(40)
    , semester         varchar(10) -- ex: Fall, Spring, Summer I, Summer II
    , year             int         -- ex: 2020, 2021
    , sectionno        int         -- ex: 1,2
    , FOREIGN KEY (classcode) REFERENCES courses (classcode)
    , UNIQUE (classcode, semester, year, sectionno)
) ;


-- All instructors in the university
CREATE TABLE instructors(
      , id             int   PRIMARY KEY
    , name             varchar(40)    -- ex: Sibel Adali
    , email            varchar(40)  -- ex: adalis@rpi.edu
    , onlineroom       varchar(40)  -- ex: rensselaer.webex.com/meet/adalis
    , note             text
) ;


-- Who teaches which course(s)
CREATE TABLE teaches(
      crn              int
    , instructorid     int
    , PRIMARY KEY (crn, instructorid)
    , FOREIGN KEY (crn) REFERENCES classes(crn)
      ON DELETE CASCADE ON UPDATE CASCADE
    , FOREIGN KEY (instructorid) REFERENCES instructors(id)
      ON DELETE CASCADE ON UPDATE CASCADE
) ;


-- When classes meet
CREATE TABLE classmeetings(
      crn          int
    , dayofweek    varchar(10)  -- ex: Monday, Tuesday
    , starttime    time          -- ex: time '14:30'
    , duration     int           -- in minutes, ex: 150
    , note         text
    , PRIMARY KEY (crn, dayofweek, starttime)
    , FOREIGN KEY (crn) REFERENCES classes(crn) ON DELETE CASCADE ON UPDATE CASCADE
) ;
```

```sql
-- When classes have office hours
CREATE TABLE officehours(
    crn         int
    , dayofweek    varchar(10)  -- ex: Monday, Tuesday
    , starttime    time         -- ex: time '18:00'
    , endtime      time         -- ex: time '19:30'
    , PRIMARY KEY (crn, dayofweek, starttime)
    , FOREIGN KEY (crn) REFERENCES classes(crn) ON DELETE CASCADE
) ;


-- When classes have exams
CREATE TABLE exams(
    crn         int
    , examname     varchar(40) -- ex: Exam 1, Exam 2, Final Exam
    , examdate     date        -- ex: date '2020/11/02'
    , pointvalue   int         -- ex: 12, 20
    , starttime    time        -- ex: time '14:30'
    , duration     int         -- in minutes, ex: 130
    , note         text
    , PRIMARY KEY (crn, examname)
    , FOREIGN KEY (crn) REFERENCES classes(crn) ON DELETE CASCADE ON UPDATE CASCADE
) ;


-- All sites that are used for different courses.
CREATE TABLE sites(
    , sitename      varchar(40) PRIMARY KEY --ex: slack, discord, teams, submitty
    , bestbrowser   varchar(40)  --ex: firefox, chrome
    , generalurl    varchar(100) --ex: webex.com
) ;


-- Which sites are used for which courses, example resourcetypes are
-- discussions, hw, videos, exams, coursenotes, meetings, officehours
CREATE TABLE resourcesites(
    rid             int  PRIMARY KEY
    , crn           int
    , resourcetype  varchar(100) -- ex: see above.
    , sitename      varchar(40)
    , resourceurl   varchar(100)
    , FOREIGN KEY (crn) REFERENCES classes(crn) ON DELETE CASCADE
    , FOREIGN KEY (sitename) REFERENCES sites(sitename) ON UPDATE CASCADE
) ;


-- All students in the database
CREATE TABLE students (
    studentid       int   PRIMARY KEY
    , email         varchar(100)
    , firstname     varchar(100)
    , lastname      varchar(100)
) ;


-- Who is enrolled in which class.
CREATE TABLE enrollment (
    crn             int
    , studentid     int
    , PRIMARY KEY (crn, studentid)
    , FOREIGN KEY (crn) REFERENCES classes(crn) ON DELETE CASCADE
    , FOREIGN KEY (studentid) REFERENCES students(studentid) ON DELETE CASCADE

) ;
```