ECS165 HW4

aelkman, aleshchuk

December 2016

1 README

use python3
run 'python3 __main__.py' to build the schema
run problem3.py for problem 3 set

2 Schema

The schema was designed to break up as many redundancies as possible while maintaining a small number of tables. Students information (surname, prefname, surname) is stores in student. Student status (major, status, level, class) for every term is stores in student$_s tatus.Coursesinformation(units_minandunits_max)isstoredinCourses.Th$

We can add meeting held by professors in the meeting table. We are able to add or remove students and courses in the future updates that will be necessary.

3 Functional Dependencies

Students(sid, term, email)
(sid, term) → email

student$_s tatus(sid, term, level, class, major, status)$
$(sid, term)\rightarrow$ level, class, major, status

Courses(subj, crse, units)
(subj, crse) → units

scheduled$_c ourses(cid, sec, term, crse, subj)$
$(cid, sec, term)\rightarrow$ crse, subj

Enrolled(cid, sec, term, sid, seat, units, grade)
(cid, sec, term, sid) → seat
(cid, sec, term, sid) → units
(cid, sec, term, sid) → grade
(cid, sec, term, sid) → (seat, units)
(cid, sec, term, sid) → (seat, grade)
(cid, sec, term, sid) → (units, grade)
(cid, sec, term, sid) → (seat, units, grade)

Meeting(cid, sec, term, type, days, time, build, room, professor)
(cid, sec, term, type, days, time, build, room) → professor

Rooms(<u>room</u>, <u>building</u>, capacity)
(room, building) $\rightarrow$ capacity

4 ER Diagram

5 Problem 3