COMPSCI 351

Fundamentals of Database Systems

Strategic Exercise 1 - Model answers

The Relational Model of Data

Exercise 1.

Consider the following relational database.

Student

Name	Student_number	Major
Smith	17	CS
Brown	8	CS

Course

Course_name	Course_number	Credits	Department
Intro to Computer Science	CS1310	15	CS
Data Structures	CS3320	15	CS
Discrete Mathematics	MATH2410	15	MATH
Database	CS3380	30	CS

Section

Section_id	Course_number	Semester	Year	Instructor
85	MATH2410	1	12	King
92	CS1310	1	12	Anderson
102	CS3320	2	13	Knuth
112	MATH2410	1	13	Chang
119	CS1310	1	13	Anderson
135	CS3380	1	13	Ullman

Grade_Report

Student_number	Section_id	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

Prerequisite

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

a. Write down the database schema, and each relation schema.

b. For each relation schema, specify a key that you consider meaningful.

c. Specify all foreign keys you consider meaningful.

- STUDENT(Name, Student_number, Major) with
  key {Student_number} and no foreign key
- COURSE(Course_name, Course_number, Credits, Department) with
  key {Course_number} and no foreign key
- SECTION(Section_id, Course_number, Semester, Year, Instructor) with
  key {Section_id} and foreign key
  [Course_number] $\subseteq$ COURSE[Course_number]
- GRADE_REPORT(Student_number, Section_id, Grade) with
  key {Student_number, Section_id} and foreign key
  [Student_number] $\subseteq$ STUDENT[Student_number] and
  [Section_id] $\subseteq$ SECTION[Section_id]
- PREREQUISITE(Course_number, Prerequisite_number) with
  key {Course_number, Prerequisite_number} and foreign keys
  [Course_number] $\subseteq$ COURSE[Course_number] and
  [Prerequisite_number] $\subseteq$ COURSE[Course_number]

Exercise 2.

Consider the relation schema SCHEDULE={c_id, l_name, time, room} which captures information on the schedule of courses (identified by their c_id) given by a lecturer (identified by their l_name) at a time (weekday and time) in a room. A single tuple over SCHEDULE is, for instance, (c_id: *351*, l_name: *Edgar Codd*, time: *Wednesday - 2pm*, room: *Eng157*).

a. Write down all keys you would enforce on SCHEDULE. Explain in English what each key means.

- {c_id, time}: the same course cannot be taught by different lectures at the same time, and the same course cannot be taught in different rooms at the same time
- {l_name, time}: the same lecturer cannot teach different courses at the same time, and the same lecturer cannot teach in different rooms at the same time
- {room, time}: different courses cannot be taught at the same time in the same room, and different lecturers cannot be in the same room and the same time

b. Write down a single relation over SCHEDULE that satisfies all of the following:

- satisfies every key you would enforce,
- violates every superkey that does not contain any key you would enforce,
- has as few tuples as possible.
- the relation

c_id	*l_name*	*time*	*room*
SOFTENG351	Edgar Codd	Wednesday-2pm	Eng157
SOFTENG351	Edgar Codd	Friday-10am	Eng157
COMPSCI211	Alan Turing	Friday-10am	Phy234

- satisfies the keys {c_id, time}, {l_name, time}, and {room, time}
- violates the superkeys {c_id, l_name, room} and {time}, and thereby violates every other superkey which does not contain any of the three keys above
- relation with only two tuples cannot violate {c_id, l_name, room} and {time} simultaneously without also violating any of the three keys above

c. Inspect your relation carefully. Does it violate any key that should be satisfied in practice?

it should be possible at any university that the same lecturer can teach the same course in the same room at different times; and it should also be possible that different courses are taught at the same time by different lecturers in different rooms, but

depending on the university some of the three keys we specified above may not be meaningful, e.g. a lecturer may teach different courses at the same time (what about COMPSCI351 and SOFTENG351 for example?)