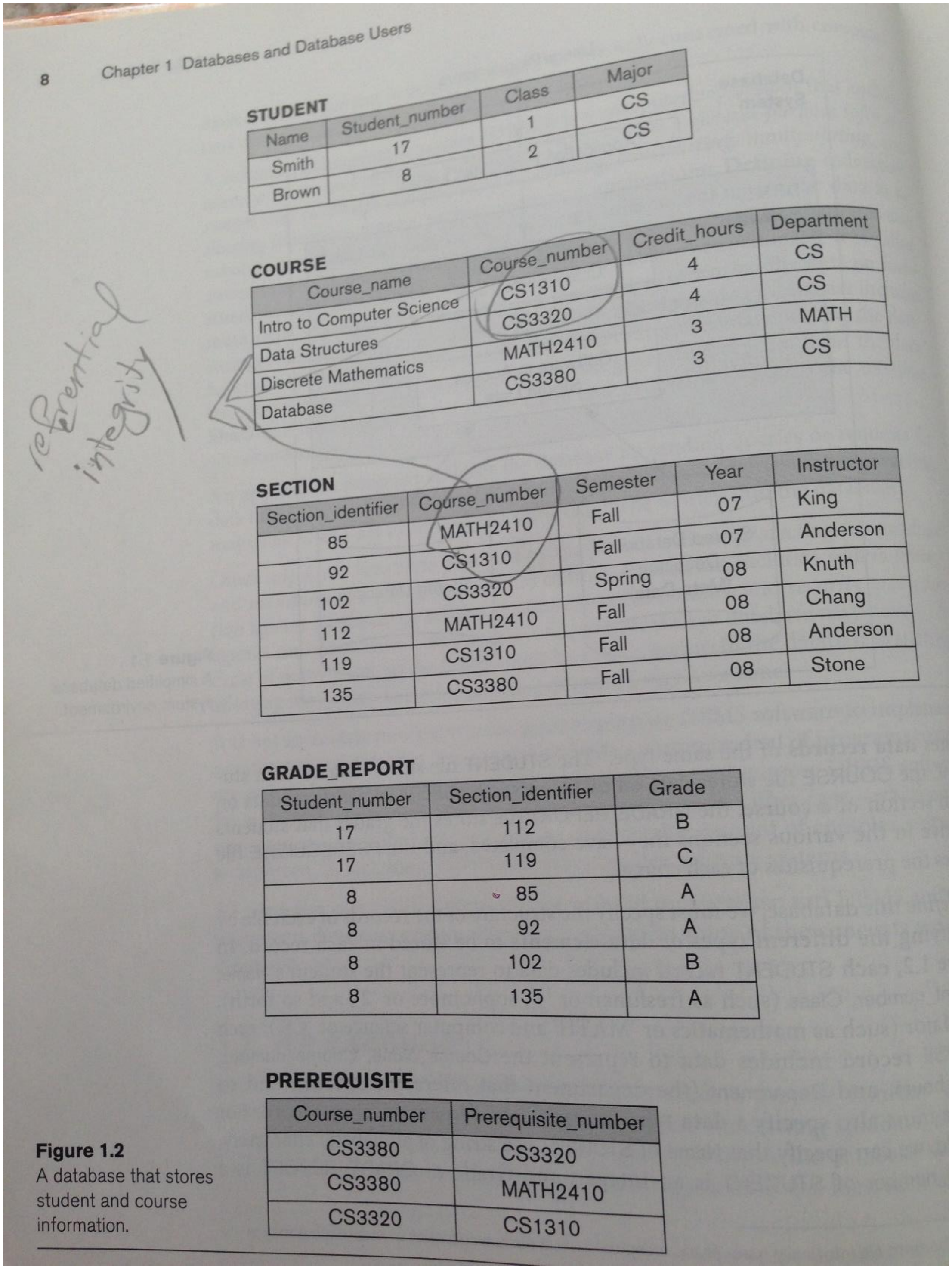
Exercise from Chapters 1 and 2 of the textbook helping selection of the project title and writing its description similar to the examples provided in the project information.
For now, consider your DB as just several tables of student DB shown in Figure 1.2 as follows:



8 Chapter 1 Databases and Database Users

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

referential integrity

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	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

Figure 1.2
A database that stores student and course information.

1.8 - Identify some informal queries and update operations that you would expect to apply to the database shown in Figure 1.2.

Answer:
(a) (Query) List the names of all students majoring in Computer Science.

(b) (Query) What are the prerequisites of the Database course?

(c) (Query) Retrieve the transcript of Smith. This is a list of <CourseName, SectionIdentifier, Semester, Year, Grade> for each course section that Smith has completed.

(d) (Update) Insert a new student in the database whose Name=Jackson, StudentNumber=23, Class=1 (freshman), and Major=MATH.

(e) (Update) Change the grade that Smith received in Intro to Computer Science section 119 to B.

1.10 - Specify all the relationships among the records of the database shown in Figure 1.2.

Answer:
(a) Each SECTION record is related to a COURSE record.

(b) Each GRADE_REPORT record is related to one STUDENT record and one SECTION record.

(c) Each PREREQUISITE record relates two COURSE records: one in the role of a course and the other in the role of a prerequisite to that course.

1.12 – Cite some examples of integrity constraints that you think can apply to the database shown in Figure 1.2.

Answer:
We give a few constraints expressed in English. Following each constraint, we give its type in the relational database terminology that will be covered in Chapter 6, for reference purposes.

(a) The StudentNumber should be unique for each STUDENT record (key constraint).

(b) The CourseNumber should be unique for each COURSE record (key constraint).

(c) A value of CourseNumber in a SECTION record must also exist in some COURSE record (referential integrity constraint).

(d) A value of StudentNumber in a GRADE_REPORT record must also exist in some STUDENT record (referential integrity constraint).

(e) The value of Grade in a GRADE_REPORT record must be one of the values in the set {A, B, C, D, F, I, U, S} (domain constraint).

(f) Every record in COURSE must have a value for CourseNumber (entity integrity constraint).

(g) A STUDENT record cannot have a value of Class=2 (sophomore) unless the student has completed a number of sections whose total course CreditHours is greater than 24 credits (general semantic integrity constraint).

2.12 Think of different users for the database of Figure 1.2. What type of applications would each user need? To which user category would each belong and what type of interface would they need?

Answer:

(a) Registration Office User: They can enter data that reflect the registration of students in sections of courses, and later enter the grades of the students. Applications can include:

- Register a student in a section of a course
- Check whether a student who is registered in a course has the appropriate prerequisite courses
- Drop a student from a section of a course
- Add a student to a section of a course
- Enter the student grades for a section

Application programmers can write a number of canned transactions for the registration office end-users, providing them with either forms and menus, or with a parametric interface.

(b) Admissions Office User: The main application is to enter newly accepted students into the database. Can use the same type of interfaces as (a).

(c) Transcripts Office User: The main application is to print student transcripts.
Application programmers can write a canned transaction using a report generator utility to print the transcript of a student in a prescribed format. The particular student can be identified by name or social security number. Another application would be to generate grade slips at the end of each semester for all students who have completed courses during that semester. Again, this application could be programmed using a report generator utility.