

Selected Exercises from Fundamentals of Database Systems

Andrew Kowalczyk

Chapter 1

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

Queries

1. Find the names of all students majoring in Mathematics.
2. What are the prerequisites of the Database course?
3. Find the transcript of the student named Brown. We would find <Course_name, Section_identifier, Semester, Year, Grade> for each course section that Brown has taken.

Updates

1. Insert a new student in the database whose Name=Kowalczyk, Student_number=25, Class=4, and Major=CS.
2. Change the grade that Brown received in Discrete Mathematics to a D.

1.9 What is the difference between controlled and uncontrolled redundancy? Illustrate with examples.

Redundancy is the term given when the same data is stored multiple times in several places in a database. If you look at Figure 1.5(a) in the text, you can see that the name of the student with Student_number=8 is Brown is stored multiple times. Redundancy is controlled when the database management system (DBMS) ensures that multiple copies of the same data are consistent. To illustrate this, let's say we are adding a new record with Student_number=8 to be stored in the database of Figure 1.5(a). If we were to have uncontrolled redundancy, the DBMS would have no control over this. If we were to have controlled redundancy, the DBMS would ensure that Student_name=Brown in that record.

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

1. Every GRADE_REPORT record is related to one STUDENT record and one SECTION record.
2. Every SECTION record is related to a COURSE record.
3. Every PREREQUISITE record relates two COURSE records. One being a course and the other being a prerequisite to that course.

1.11 Give some additional views that may be needed by other user groups for the database shown in Figure 1.2.

1. A view of each class section that groups all the students who took that section and their respective grade.
2. A view that gives the number of courses taken and the grade point average for each student.

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

Key constraints

1. Student_number must be unique for each STUDENT record.
2. Course_number must be unique for each COURSE record.

Referential integrity constraints

1. The Course_number in a SECTION record must also exist in some COURSE record.
2. The Student_number in a GRADE_REPORT record must also exist in some STUDENT record.

Domain constraints

1. Grades in a given GRADE_REPORT record must be one of these values: A, B, C, D, F, I, W .

Chapter 2

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

1. Students add and drop classes. Actions that they can do are as listed:
 - (a) Register themselves in a section of a course
 - (b) Drop themselves from a section of a course
2. Registrar. They enter data of registration of students in sections of courses, and later enter the grades of the students. Actions that they can do are as listed:
 - (a) Check whether a student who is registered in a course has the appropriate prerequisite courses
 - (b) Add a student to a section of a course
 - (c) Enter the student grades for a section
3. Admissions. Their main application would be to enter newly accepted students into the database. Actions that they can do are as listed:
 - (a) Add students to the school's records

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Chapter 10