

Normalization example

Symbol	Attribute Name	comment
I_S	student_id	uniquely identifies a student
N_S	student_name	a student's name
M	major	4-character string uniquely identifying a major e.g. COMP, MATH
I_A	advisor_id	uniquely identifies an advisor
N_A	advisor_name	an advisor's name

Figure 1: Symbols and names of attributes for student/advisor example.

I_S	N_S	M	I_A	N_A
931	Minh	COMP	46	MacCormick
416	Tayyaba	MATH	53	Schaefer
842	Minh	ENGL	21	Seiler
416	Tayyaba	COMP	46	MacCormick
729	Harold	AMST	73	Seiler

Figure 2: Some example tuples of a possible advising relation.

Suppose that a fictional college keeps a database of which professors are advisors for which students. Relevant attributes of entities in the schema are shown in figure 1. As stated in the figure, the student ID uniquely identifies a student and the adviser ID uniquely identifies an advisor. The only additional constraint is: every major is associated with exactly one advisor, who acts as the advisor for all students in that major. There are no other constraints.

Suppose that at present the college stores advising data in a single database table described by the advising relation schema:

$$\text{advising}(\underline{I_S}, N_S, \underline{M}, I_A, N_A). \quad (1)$$

Some examples of possible data for five student-advisor combinations are shown in figure 2.

The database described above is not in 3NF or BCNF. Your task for this question is to design a new set of relations, representing the same data in BCNF if possible. If BCNF is not possible, use 3NF where necessary.

(a) (25 points) The database design described above does not employ good normalization techniques. Redesign the database, representing the same information and constraints in a set of relations that are in BCNF (or 3NF wherever BCNF is not possible). Give your answer as a set of relations using the same notation as in equation (1) above, with additional annotations for any foreign keys. Reasoning and explanation are not required.

(b) (10 points) Describe the reasoning behind your design, explaining why you believe the new formulation is in BCNF or 3NF. Rigorous proof is not required, but your answer should demonstrate your understanding of BCNF and 3NF.

Solution to (a)

$\text{students}(\underline{I_S}, N_S)$
 $\text{advisors}(\underline{I_A}, N_A)$
 $\text{major_advisors}(\underline{M}, I_A); I_A \text{ is foreign key referencing advisors}$
 $\text{student_majors}(\underline{I_S}, \underline{M}); I_S \text{ is foreign key referencing students,}$
 $M \text{ is foreign key referencing major_advisors}$

Solution to (b)

[Note: The question states that this answer is worth only 10 points. Therefore you should spend only about 10 minutes on it. I have tried to provide a very detailed explanation below. Under exam conditions you would need to summarize the following explanation briefly, keeping only highlights so that you can complete your answer in 10 minutes.]

In the original advising relation schema, the only key is $\{I_S, M\}$. This can be determined by exhaustively checking every subset of columns and noting that only subsets containing I_S, M yield uniqueness in the remaining columns. (Additional explanation: the only other subset that looks like it might be a key is $\{I_S, I_A\}$. But this is not a key because the assumptions do not state that a given professor advises only one major.)

Again by checking all possible subsets of columns, we find that the following is a complete list of minimal functional dependencies (*minimal* means we can't delete any columns from the left hand side):

$$\begin{aligned}
 I_S &\rightarrow N_S \\
 I_A &\rightarrow N_A \\
 M &\rightarrow I_A, N_A \\
 I_S, M &\rightarrow N_S, I_A, N_A
 \end{aligned}$$

(By the way, there are some other functional dependencies such as $M \rightarrow I_A, M \rightarrow N_A$ —but these follow immediately from the above and we don't bother to list them separately. I generally try to list the functional dependencies with the smallest possible left hand side and the largest possible right hand side. This will capture all of the meaningful dependencies.)

The first three functional dependencies above violate BCNF, because in each case the left hand side is not a superkey. Therefore we should create a new relation schema for each of the violating functional dependencies, removing fully determined columns as we go. We haven't studied a formal algorithm for this, but informally we can proceed as follows. The first violating dependency above suggests creating $\text{students}(\underline{I_S}, N_S)$. This is in BCNF. It also fully determines the column N_S , so we can delete N_S everywhere else. The next violating dependency works in a similar fashion: we create $\text{advisors}(\underline{I_A}, N_A)$, check that it is in BCNF, then delete N_A everywhere else. The next violating dependency suggests $\text{major_advisors}(\underline{M}, I_A)$, which is in BCNF. Column N_A was already deleted and now we also delete I_A . Finally, the last functional dependency above yields a

relation schema $\text{student_majors}(\underline{I_S}, \underline{M})$ —all the other columns were deleted. This last relation schema is in BCNF, so we are done.