Brian Dones

Professor Labouseur

Database Management

17 March 2015

Relational “Rules”

First Normalization Form:

- The values in each column are atomic
- There are no repeated groups or columns

Example without 1NF		
Student	Year	Major
Alex	Freshman	Biology
Bobby	Junior	Math, Computer Science
Chris	Sophomore	Philosophy, Legal Law
Dan	Senior	Psychology

Example with 1NF		
Student	Year	Major
Alex	Freshman	Biology
Bobby	Junior	Computer Science
Bobby	Junior	Math
Chris	Sophomore	Philosophy, Legal Law
Dan	Senior	Psychology

Using the First Normalization Form guarantees that each row will be unique which is crucially important in the world of databases; however, there is a potential for redundancy. For example, in the above table with 1NF, both the second and third row refers to student Bobby.

Second Normalization Form:

- All aspects of 1NF
- There must not be any partial dependencies of any column on the primary key

Partial Dependency			
Course ID	Semester ID	# of seats	Course Name
CMSC101	F14	30	Programming
CMSC101	S15	30	Programming
CMSC203	F14	24	Operating Systems
CMSC203	F15	24	Operating Systems
CMSC305	F14	15	System Design

By looking at this table, we can see that the Course Name is not dependent on the entire key but only a part of the key. The Course name is dependent on the Course ID but the semester it is taken in is completely independent of the Course Name. By having this partial dependency there is redundant information. With the use of 2NF, we ensure that the columns fully depend on the primary key.

2NF	
Course ID	Course Name
CMSC101	Programming
CMSC203	Operating Systems
CMSC305	System Design

By using 2NF, we prevent duplicate data and guarantee no redundancy.

Third Normalization Form:

- All aspects of 2NF
- All the attributes in a table are determined only by the candidate keys of that table and not by any non-prime attributes.

Non-3NF				
Course_ID	Semester	# of Seats	Teacher_ID	Teacher_Name
CMSC230	F14	20	312	Sorenson
CMSC230	S15	20	312	Sorenson
CMSC330	F14	25	546	Martin
CMSC330	F15	25	546	Martin
CMSC415	S15	15	718	Gates

In this table, we can see that the Teacher_Name column is dependent on the Teacher_ID column so the table is not in 3NF. To make this table 3NF we will need to take the Teacher_Name field out of this table, and create another table of its own where the Teacher_ID is the key. The following is the resulting table:

3NF	
Teacher_ID	Teacher_Name
312	Sorenson
546	Martin
718	Gates

As we can see from the table, there is no redundancy. Just like 2NF, 3NF also helps with eliminating redundancy but there are two important advantages of removing transitive dependencies. First, the amount of data duplication is reduced and therefore our database becomes smaller. The second advantage is data integrity. When duplicate data is changed, there is a huge risk of updating only some of the data or having inaccurate data. The less tables we have, or the smaller our database is, the less we have to worry about this risk.