Topic 1 Discussion Topic: The Purpose of Normalization

Describe the purpose of normalization and the normalization process in your own words. Provide an example of the application of Normalization. Identify the importance of normalization in your own words.

This is it Edward Jackson ✉ 5/31/2014 5:05:35 PM

In my own words, normalization is about organizing a database so it operates efficiently; it is not necessarily about speed (which I have seen some responses about). It can speed up a database, but the main focus of normalization is removing redundancies and making sure the database itself makes sense. I actually watched the Lynda.com series on programming databases, which explained the normal forms and why each is important.

There are three normal forms (actually, there are more, but we are only focusing on these three), they are first normal form (1NF), second normal form (2NF), and third normal form (3NF). In a 1NF, the design idea is to remove repeating groups and to make sure single values are in each attribute. At first, I still was not quite sure what that meant. But, after watching the videos, it made perfect sense.

The repeating groups would look something like this in a computer inventory example: Computer1, Computer2, Computer3 (see the increments?)….those would be listed across the top of a table when collecting serial numbers. A better way to handle this in a database is to not have the repeating group, but to pull out “Computer,” make it a separate table, and link the “Computer” table with the original table with a one-to-many relationship from the original table to the Computer table. Notice how an extra table had to be created…this seems to be the case in many scenarios of normalization.

Now, about single values being in attributes. This was also made clear in the videos. Let’s say you have a ComputerSerial attribute in an Employee table. If an employee could have one and only one ComputerSerial number assigned to them, there wouldn’t be an issue. But let’s say an employee tests applications, and they have 3 computers, thus having 3 serial numbers. The incorrect way to store these serials would be under the ComputerSerial attribute you list something like this: 34673647, 37467346, 232232. You have just violated the 1NF rule. There should only be one piece of data per attribute. (Allardice, n.d.)

Now, in the 2NF you have to look at the primary key. If the primary key is a composite key, then you need to look at the other attributes to make sure they are dependent on the “whole” key---emphasis on whole key. For example, let’s say our composite key is Course_Name AND Course_Start_Date….and our other attributes are Course_Title and Room_Number. Room_Number is different per term…per course, thus Course_Name and Course_Start_Date are unique…and work. However, with the attribute of Course_Title, it can be extracted from the first part of our primary key Course_Name, so this is a violation of 2NF. To resolve this, we would pull out the Course_Name, and make a Courses table…linking the Courses table to the original table (that would be 1:M from the original table to the new Courses table). (Allardice, n.d.)

Now, on to 3NF. In the 3NF normalization, we are comparing each non-key to other non-keys, meaning comparing attributes to attributes other than the primary keys. The objective here to make sure non-keys are not dependent on other non-keys. For example, staying with our college examples, let's say we have a Classroom table with Room_Number, Total_Capacity, and Available_Seats non-key attributes. Let me put some numbers in there so you see how it looks:

* This is just the non-keys

Room_Number	Total_Capacity	Available_Seats
1A	10	5
2A	20	10
2A	20	15
3A	10	10
3A	10	7

Now, by looking at this table…we notice some dependent data…room numbers and total capacity are dependent on one another. This is a violation of 3NF, and once again can be resolved by creating a separate table called Room_Number, and linking the tables together in a one-to-many relationship. (Allardice, n.d.)

References

Allardice, Simon. (n.d.). Foundations of Programming: Databases. Retrieved from http://www.lynda.com/Programming-tutorials/Foundations-Programming-Databases/112585-

2.html?srchtrk=index:1%0Alinktypeid:2%0Aq:Foundations%2Bof%2BProgramming:%2BDatabases%2B%0Apage:1%0As:relevance%0Asa:true%0Aproducttypeid:2

Chapple, Mike. (n.d.). Database Normalization Basics. Retrieved from http://databases.about.com/od/specificproducts/a/normalization.htm

Coronel, C., Rob, P., & Morris, S. (2012). *Database systems: desi, implementation, and management (10e. ed.)*. Boston, MA: Cengage Learning.

Topic 2 Discussion Topic: Applying Normalization

As part of your Assignment you will find a form on the web and follow the bottom up database design approach. Please identify the website URL and provide the image of this form and provide the first two steps of the Assignment.

1. Find all the attributes on the form.
2. Establish the dependencies (determinants).

University Registration Example	Edward Jackson	5/31/2014 5:43:03 PM

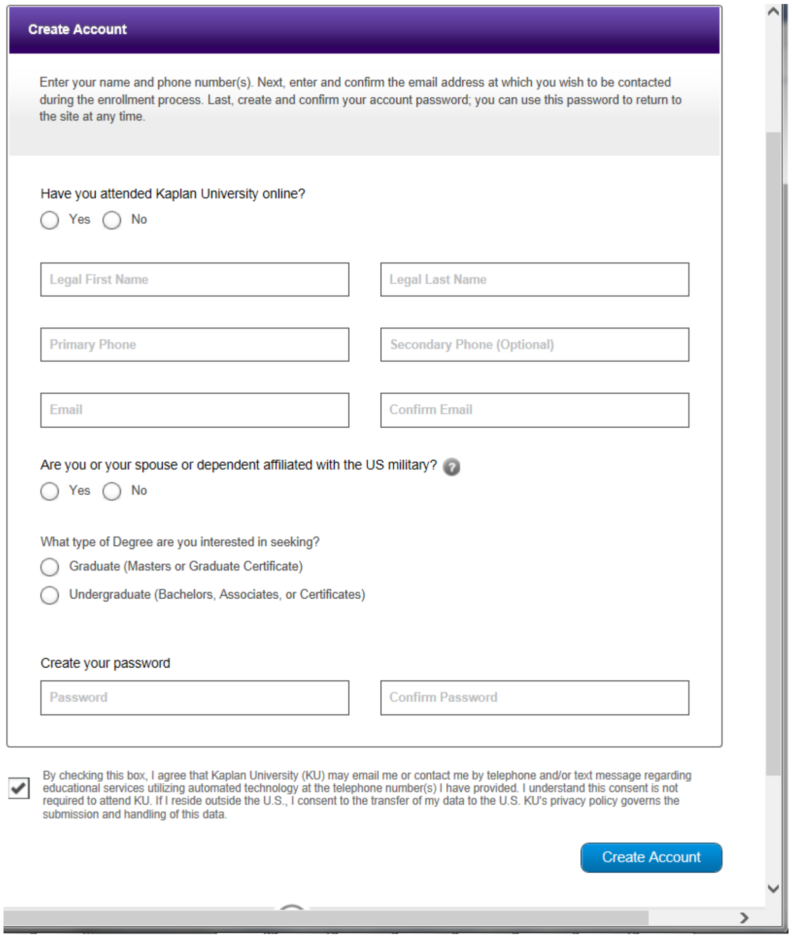
The form I have selected is from Kaplan University, and is the student enrollment form: https://enroll.kaplanuniversity.edu/Account/Register

Attributes

The attributes would be Online_Status, Legal_First_Name, Legal_Last_Name, Primary_Phone, Secondary_Phone, Email, Military_Status, Graduate_Status, Password.

Dependencies

The dependencies, if I were creating a database, would be Legal_First_Name and Legal_Last_Name, which I would use to create a unique key of stuID. For example, using the first and last name, I would create a stuID that looked something like this EddieJackson21.



Create Account

Enter your name and phone number(s). Next, enter and confirm the email address at which you wish to be contacted during the enrollment process. Last, create and confirm your account password; you can use this password to return to the site at any time.

Have you attended Kaplan University online?

○ Yes ○ No

Legal First Name	Legal Last Name
Primary Phone	Secondary Phone (Optional)
Email	Confirm Email

Are you or your spouse or dependent affiliated with the US military?

○ Yes ○ No

What type of Degree are you interested in seeking?

○ Graduate (Masters or Graduate Certificate)

○ Undergraduate (Bachelors, Associates, or Certificates)

Create your password

Password	Confirm Password

☑ By checking this box, I agree that Kaplan University (KU) may email me or contact me by telephone and/or text message regarding educational services utilizing automated technology at the telephone number(s) I have provided. I understand this consent is not required to attend KU. If I reside outside the U.S., I consent to the transfer of my data to the U.S. KU's privacy policy governs the submission and handling of this data.

http://eddiejackson.net/web_images/myformclass.png

Normalization

Normalization is the process of removing redundancy so that the table is easier to modify and insertions and/or modifications do not result in unexpected side effects called Modification Anomalies. Normal forms are the state of a relation that results from applying simple rules regarding functional dependencies (or relationships between attributes) to that relation.

The unit will review 1st, 2nd, and 3rd Normal Form.

Outcomes

After completing this unit, you should be able to:

- Present the concept of Normalization.
- Identify 1st, 2nd, and 3rd Normal Forms.
- Develop a database in 3rd Normal Form.

Course outcome(s) practiced in this unit:

IT525-4: Construct relations in first, second and third normal form.

What do you have to do in this unit?

- Complete the assigned Reading.
- Participate in the Seminar or complete the Alternative Assignment.
- Participate in the Discussion Board.
- Complete the unit Assignment.

- **Normalization:** The process of decomposing relations with anomalies to produce smaller, well-structured relations.
- **Determinant:** The attribute on the left-hand side of the arrow in a functional dependency.
- **Functionaldependency:** A constraint between two attributes or two sets of attributes.

Transitivedependency: A functional dependency between two (or more) non-key attributes. From Wikipedia: In Database Management System, a **transitive dependency** is a functional dependency which holds by virtue of transitivity. A transitive dependency can occur only in a relation that has three or more attributes. Let A, B, and C designate three distinct attributes (or distinct collections of attributes) in the relation. Suppose all three of the following conditions hold:

1. A → B
2. It is not the case that B → A
3. B → C

Then the functional dependency A → C (which follows from 1 and 3 by the axiom of transitivity) is a transitive dependency.

In database normalization, one of the important features of third normal form is that it excludes certain types of transitive dependencies. E.F. Codd, the inventor of the relational model, introduced the concepts of transitive dependence and third normal form in 1971.[1]

Example: The functional dependency {Book} → {Author Nationality} applies; that is, if we know the book, we know the author's nationality.

- **Normal form:** A state of a relation that results from applying simple rules regarding functional dependencies (or relationships between attributes) to that relation.
- **Partialfunctionaldependency:** A functional dependency in which one or more non-key attributes (such as Name) are functionally dependent on part (but not all) of the primary key.
- **1NF:** No multi-valued attributes; no repeating groups that have assumed values rather than real values; every attribute value is atomic; all relations are in 1NF.
- **2NF:** Every non-key column depends on an entire primary key, not part of a key; no partial functional dependencies; every relationship/table is in both 2NF and 1NF.
- **3NF:** Every non-key column depends only on a key not on non-key columns; no transitive dependencies; every relationship/table is in 3NF, 2NF, and 1NF.

Question 1

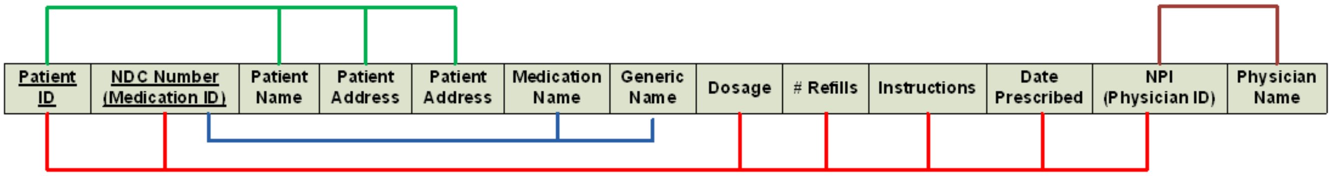
Find a form on the Web and follow the bottom up database design approach. Please include the website URL and an image of this form.

1. Find all the attributes on the form.

2. Establish the dependencies (determinants).
3. Group attributes that have a common determinant into an entity type; name it.
4. Find directly-related entity type pairs.
5. Determine the connectivity for each pair.
6. Draw the ERD.
7. Review the ERD and update to be in 3NF if ERD from step 6 is not in 3NF.

Question 2 Dependency Diagram

1. Based on the dependency diagram below create an ERD in 2NF but not 3NF showing the dependency diagram.
2. Based on A, create an ERD in 3NF.



See the grading rubric on the next page.

Review the grading rubric below before beginning this activity.

100 point project grading rubric

Project Requirements/criterion	**Points Possible**	**Points earned by student**
1 . Design shows steps 1-8 of the Bottom Up Database Design Process.	0-50	
2A. ERD demonstrates that the student was able to provide a Database on 2nd NF.	0-30	
2B. ERD demonstrates that the student was able to provide a Database on 3rd NF.	0-20	
Total (Sum of all points)		
Points deducted for spelling, grammar, and/or APA errors.		
Adjusted total points		

Attending live Seminars is important to your academic success, and attendance is highly recommended. The Seminar allows you to review the important concepts presented in each unit, discuss work issues in your lives that pertain to these concepts, ask your instructor questions, and allow you to come together in real time with your fellow classmates. There will be a graded Seminar in Units 1 through 5 in this course. You must either attend the live Seminar or you must complete the Seminar alternative assignment in order to earn points for this part of the class.

Topics reviewed in the Seminar will include:

- Review the concept of Normalization.
- Review 1st, 2nd, and 3rd Normal Forms.
- Develop a Normalized a Database Design.

Option 2- Alternative Assignment:

You will benefit most from attending the graded Seminar as an active participant. However, if you are unable to attend you have the opportunity to make up the points by completing the alternative assignment.

Please review the Seminar. Provide a 200 word summary of the Seminar. Follow APA format. Include at least two references and two citations.

Format:

- One inch margins (top, bottom, sides), Times New Roman or Arial 12 point font.
- Double spaced.
- Running header with title, name, and page numbers.
- References and citations follow APA Format. Do not use more than 5 words directly from a source without quotation marks to avoid plagiarism.

Rubric:

1. Two hundred words. 5 points.
2. Compliance with APA format. At least two reference and citations. 5 points
3. Writing ability (Grammar, Spelling, Flow). 5 points
4. Mastery of database design concepts. 10 points

Your paper should be in APA format and cite all references used. Submit to the Seminar Dropbox.