

Lab 09 – Normalization

(UNF, 1NF)

Lab 09 – Normalization(UNF, 1NF)

Name: Naiba Shabanova

ID :164598229

Email: nshabanova@myseneca.ca

Section: DBS211

Objective:

- Students create the **Un-normalized Form** (UNF) relation from a user view.
- Students identify a **Multi-valued Dependency** (a.k.a. Repeating Group)
- Students create the **First Normal Form** (1NF) relation from the UNF.

Submission:

Only submit the last section, however it is extremely important that you you're your way through it first.

Save your lab file as a PDF file. You need to submit a single **PDF** file for this lab.

YOUR NAME AND STUDENT ID MUST BE IN THE PDF document or you will receive a mark of ZERO.

The name of the file must be as follow:

DBS211_L09_LASTNAME.pdf

Definitions:

Definition: **Normalization** is the process of assigning attributes to relations in such a way that data redundancies are reduced or eliminated.

Definition: **User Views** can be individual descriptions, reports, forms, or lists of data that are required to support the operations of a particular database user.

Definition: **Unnormalized form (UNF)** is a relation that contains one or more **Multi-valued Dependencies**.

Definition: A **Multi-valued Dependency** is an attribute or collection of attributes within a relation that has multiple values for a single value of the primary key for that relation.

Definition: A relation is in **1NF** if it does not contain any multi-valued dependencies.

Instructions:

Step 1: Create UNF Relation from a User View. The goal here is to create a single relation for the data found in the user view. The method used is:

- **Examine the user view**
- **Identify all attributes**
- **Describe the user view using DBDL**

Here is an example of a user view. This reports lists the customers of the Premiere Corporation.

Premiere Corporation Customer List

CustNo	Name	Street	City	State	ZipCode	SalesRepNo
124	Sally Adams	482 Oak	Lansing	MI	49224	03
256	Ann Samuels	215 Pete	Grant	MI	49219	06
311	Don Charles	48 College	Ira	MI	49034	12
315	Tom Daniels	914 Cherry	Kent	MI	48391	06
405	Al Williams	519 Watson	Grant	MI	49219	12
412	Sally Adams	16 Elm	Lansin	MI	49224	03
522	Mary Nelson	108 Pine	Ada	MI	49441	12

- Examine the user view:** As you examine this report, you can see that it contains a title, a line of column headings and the body of the report. Each line in the body of the report contains data about a particular customer. For example we can see that in the first line of the body of the report, there is data about Customer 124. This customer's name is Sally Adams and she lives at 482 Oak in Lansing, Michigan (MI). The Sales Representative who calls on Sally Adams is Sales Rep Number 03.
- Identify all attributes:** The attributes (or characteristics) of a User View can often be found by simply looking at the column headings. In our Customer Report we see that we have the following attributes: Customer Number, Customer Name, Street, City, State, Zip Code and Sales Representative Number.
- Describe the relation using DBDL:** Database Design Language (DBDL) is a standardized way of describing relations of a relational database. You describe a relation by:
 - Choose an appropriate name for the relation. We chose to name our relation **CUSTOMER** because each line in the report describes customer data.
 - List the attributes you found in the user view inside square brackets, giving each attribute a suitable attribute name. Note: *calculated fields or derived fields* should not be included in the DBDL
 - Determine which attribute would be suitable as a primary key and underline that attribute.

The DBDL for the relation resulting from our Customer user view would look as follows:

CUSTOMER [CustNo, CustName, CustStreet, CustCity, CustSt, CustZip, CustRep]

Now you try it. Examine the following report:

Premiere Corporation Parts List

Part Number	Part Description	Qty On Hand	Class	Warehouse On Hand	Price
AX12	Iron	104	HW	3	24.95
AZ52	Dartboard	20	SG	2	12.95
BA74	Basketball	40	SG	1	29.95
BH22	Cornpopper	95	HW	3	24.95
BT04	GasGrill	11	AP	2	149.99
BZ66	Washer	52	AP	3	399.99
CA14	Griddle	78	HW	3	39.99
CB03	Bike	44	SG	1	299.99
CX11	Blender	112	HW	3	22.95
CZ81	Treadmill	68	SG	2	349.95

What type of data does each line in the report represent?

ANS: Each line in the body of the report contains data about a particular part. For example, the first line of the report gives information about the part number AX12. It contains the part description (Iron), amount on hand (104), class (HW), warehouse (3), and price (\$24.95). Each line contains information on a specific part supplied by Premiere Corporation.

What attributes can you identify from the user view?

- **Part Number**
- **Class**
- **Part Description**
- **Warehouse On hand**
- **Quantity on Hand**
- **Price**

What would be a suitable name for the UNF relation?

_____Parts_____

Which attribute would be suitable as a primary key?

_____Part Number_____

Describe the UNF relation using DBDL:

ANSWER : PARTS [PartsNum, PartsDesc, PartsQty, PartsClass, PartsWarehouse, PartsPrice]

Step 2: Recognize Multi-valued Dependencies.

For example, looking at the following User View, we see that for each Customer number, it is possible to have **multiple values** for the order number and order date attributes. Therefore the order number and date are a multi-valued dependency.

Premiere Corporation Customer Orders

Customer Number	Name	Order Number	Order Date
124	Sally Adam	12489	2016-09-02
		12500	2016-09-05
256	Ann Samuels	12495	2016-09-04
311	Don Charles	12491	2016-09-02
315	Tom Daniels	12494	2016-09-04
522	Mary Nelson	12498	2016-09-05
		12504	2016-09-05

Identify multi-valued dependencies in DBDL by placing brackets around them. For example the DBDL for this User View would look like this:

CUSTOMER [CustNo, CustName, (OrderNum, OrderDate)]

Notice the brackets around the OrderNum and Orderdate attributes. This quickly and easily identifies a multi-valued dependency to someone who is reading the DBDL.

Common Mistake: A common mistake is to incorrectly identify repeating data as a multi-valued dependency. For example, the previous report could also have been printed in the following way:

Premiere Corporation Customer Orders

Customer Number	Name	Order Number	Order Date
124	Sally Adam	12489	2016-09-02
124	Sally Adam	12500	2016-09-05
256	Ann Samuels	12495	2016-09-04
311	Don Charles	12491	2016-09-02
315	Tom Daniels	12494	2016-09-04
522	Mary Nelson	12498	2016-09-05
522	Mary Nelson	12504	2016-09-05

The fact that the Customer Number and Name for Sally Adams and Mary Nelson have been repeated on multiple lines does **not** make Customer Number and Name a multi-valued dependency! You should still identify that for one customer number, there are multiple values for the order number and date. Note that this does not mean that EVERY customer number will have multiple order numbers, just that this user view shows that it is possible for some customer

numbers to have multiple values for Order Number and Date. The Multi-valued Dependency must be identified in the DBDL. Note also that it is possible to have more than 1 multi-valued dependency.

Step 3: Create 1NF relations from UNF.

Therefore, the process of taking a relation from UNF to 1NF, involves resolving the multi-valued dependencies.

Method:

- **Choose a primary key for the multi-valued dependency.**
- **Identify the primary key of the multi-valued dependency by underlining it or writing (PK) .**
- **Rewrite the DBDL by removing the paranthesis and concatenating the original primary key with the primary key of the multi-valued dependency.**
- **Rewrite the DBDL with the two-part Primary Key and include all the non-key attributes.**

For example, using our previous report from part B, we had the following:

UNF: Customer [CustNo, CustName, (OrderNum, OrderDate)]

- Choose a primary key for the repeating group:** **OrderNum** would make a suitable primary key for the repeating group as it uniquely identifies the data in the repeating group.
- Rewrite the DBDL by removing the paranthesis and concatenating the original entity name with the entity name of the multi-valued dependency.**

CustOrder [CustNo, OrderNum, CustName, OrderDate]

NOTE: If we start with a relation that does not contain any multi-valued dependencies, it is already in 1NF.

Lab 09 Submission:

For the following User View, determine the UNF and the 1NF and hand in this page to your instructor.

Premiere Corporation Order Detail Report

Order Number	Order Date	Cust number	Part Number	Part Desc	Number Ordered	Quoted Price	Total
12489	2016-09-02	124	AX12	Iron	11	14.95	164.45
12491	2016-09-02	311	BT04	GasGrill	1	149.99	149.99
			BZ66	Washer	2	399.99	799.98
12494	2016-09-04	315	CB03	Bike	4	279.99	1,119.96
12495	2016-09-04	256	CX11	Blender	2	22.95	45.90
12498	2016-09-05	522	AZ52	Dartboard	2	12.95	25.90
			BA74	Basketball	4	24.95	99.80
12500	2016-09-05	124	BT04	GasGrill	3	149.99	449.97
12504	2016-09-05	522	CZ81	Treadmill	2	325.99	651.98

UNF:

CK = candidate key; these are two keys that TOGETHER serve as primary key.

OrderDetails: [CK orderNumber, custNumber (orderDate, orderNumber), numberOrdered (partNumber, partDesc, quotedPrice)]

ANSWER:

OrderDetails [OrderNum, custNum, partNum, orderDate, partDesc, quotePrice, numOrd]

1NF:

CustOrder: [custNumber, orderNumber, partNumber, orderDate, partDesc, numberOrdered, quotedPrice]

1. OrderDetail[CK orderNum, orderDate, FK custNum, FK partNum, numOrd]
2. Customer[CK custNum]
3. Parts[FK CK OrderNum, PartNum, partDesc, quotePrice]