Module 3: Normalization

Lab: Normalizing Data

Exercise 1: Normalizing to First Normal Form

Task 1: Review the Data Set

1. Ensure that the 10985C-MIA-DC and 10985C-MIA-SQL virtual machines are both running, and then log on to 10985C-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. In the **D:\Labfiles\Lab03\Starter** folder, double-click **Normalize to 1NF.xlsx**.
3. Click the **Raw Data** tab, and then review the data set, noting any aspects of the data that cause it to violate first normal form.

Task 2: Normalize the Data Set to First Normal Form

1. Decompose the table in the **Raw Data** worksheet to create a new version of the data that is in first normal form. Document your solution in the **Normalize to 1NF** worksheet. (Hint: to do this, you should decompose the raw data to create three tables.)
2. Click the **1NF Solution** tab, and then review the suggested solution, comparing it to your own.
3. Close **Normalize to 1NF.xlsx**, saving any changes if prompted.

Exercise 2: Normalizing to Second Normal Form

Task 1: Review the Data Set in First Normal Form

1. In the **D:\Labfiles\Lab03\Starter** folder, double-click **Normalize to 2NF.xlsx**.
2. Click the **Normalize to 2NF** tab, and then review the table to identify any features of the data that might cause the table to violate second normal form.

Task 2: Normalize the Data Set to Second Normal Form

1. Decompose the table in the **Normalize to 2NF** worksheet to create a new version of the data that is in second normal form. Document your solution under the **Orders** table in the **Normalize to 2NF** worksheet.
2. Click the **2NF Solution** tab, and then review the suggested solution, comparing it to your own.
3. Close **Normalize to 2NF.xlsx**, saving any changes if prompted.

Exercise 3: Normalizing to Third Normal Form

Task 1: Review the Data Set in Second Normal Form

1. In the **D:\Labfiles\Lab03\Starter** folder, double-click **Normalize to 3NF.xlsx**.
2. Click the **Normalize to 3NF** tab, and then review the table to identify any features of the data that might cause the table to violate third normal form.

Task 2: Normalize the Data Set to Third Normal Form

1. Decompose the table in the **Normalize to 3NF** worksheet to create a new version of the data that is in third normal form. Document your solution under the **Customers** table in the **Normalize to 3NF** worksheet.

2. Click the **3NF Solution** tab, and then review the suggested solution, comparing it to your own.

3. Close **Normalize to 3NF.xlsx**, saving any changes if prompted.

Exercise 4: Denormalizing Data

Task 1: Review the Data Set in Third Normal Form

1. In the **D:\Labfiles\Lab03\Starter** folder, double-click **Denormalize.xlsx**.

2. Click the **Denormalize** tab, and note that there is one row in the **Orders** table for each order, and one or more rows for each order in the **OrderDetails** table.

Task 2: Denormalize the Orders Table

1. On the **Denormalize** tab, denormalize the **Orders** table to meet the criteria outlined in the scenario for this exercise. Document your solution under the **OrderDetails** table.

2. Click the **Denormalized Solution** tab, and then review the suggested solution, comparing it to your own.

3. Close **Denormalize.xlsx**, saving any changes if prompted.

Lab Review

Question: In the suggested solution, the Orders table has a surrogate key as its primary key. Why do you think this key was chosen instead of a composite candidate key, such as the OrderDate and Customer columns?

Answer There are two good reasons for this choice:

- The primary key in Orders is referenced from the OrderDetails table. Using the composite candidate key would mean that these larger key values would have to be repeated multiple times in the OrderDetails table, which is inefficient. Using the surrogate key instead means that the OrderDetails table references the Orders table by using a much smaller key.

- It is logically possible for a customer to place more than one order in a day, so the combination of OrderDate and Customer is not actually a valid candidate key. This is another example of the importance of understanding the business processes in addition to the data.

Question: In the suggested solution, in the Customers table, are there any other columns that you could break down even further? If so, why might you do this?

Answer You could break down the following columns:

- Customer could be split into multiple columns, including Title, FirstName, MiddleName, and LastName.

- Address could be split into multiple columns, including Number, Street, City, PostalCode, State, and Country.

You might break down these columns if it made sense for business processes. For example, updating a customer’s name if it is misspelt, or if they change their name because of marriage, is easier and less vulnerable to error if you only need to update the last name, instead of the full name. Alternatively, it might be important to be able to query customers based on a part of their address, such as City or Postal Code. In this case, breaking down the address makes these types of queries easier to write and will be more efficient.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.