Chapter 13 – Data Modeling Techniques

“A good head and a good heart are always a formidable combination.”

– Nelson Mandela

The Four Stages of Modeling for Teradata

Four core stages are identified as being relevant to any database design task. They are:

1. Requirement Analysis involves figuring out what information the users need and their processing requirements. The output of this is the Business Information Model (BIM) which shows major entities and their relationships. This is sometimes referred to as the Business Model or the Business Impact Model .

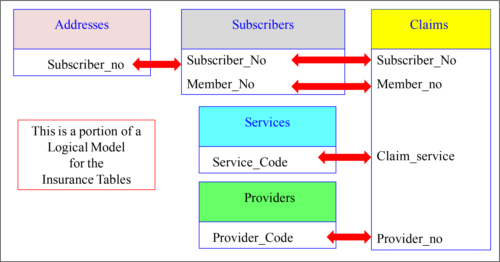
2. Logical Modeling determines defining the tables and their relationships, and this should be independent of a particular database or indexing strategy. Teradata works well with a Third Normal Form (3NF) model, and the output defines all tables and their columns and the Primary Key Foreign Key relationships between tables plus all constraints.

3. Activity Modeling involves working with the users to determine the volume, usage, frequency, and integrity analysis of a database. This process also consists of placing any constraints on domains and entities. This is called the Extended Logical Data Model (ELDM).

4. Physical Modeling Using the Activity Model and transforming the logical model into a definition of the physical model that will accelerate queries with the best Primary and Secondary Indexes that best suits a specific hardware and software configuration.

The Logical Model

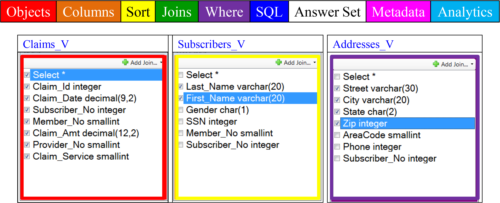
This simple logical model shows the Primary Key and Foreign Key relationships between tables.



There are five tables above that have a relationship. That means they can be joined together with SQL to produce one report.

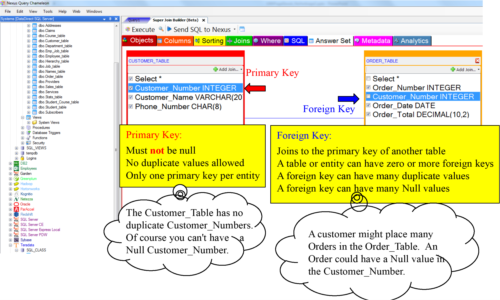
The Logical Model can be loaded inside Nexus

The Nexus Query Chameleon takes the ERwin logical model and loads it inside Nexus. Now, users can see table relationships, click on what they want, and Nexus writes the SQL.



We understand that data in a data warehouse has tables, rows, and columns. In the old days, these were called files, records, and fields. In logical modeling terms, they are called relations, tuples, and attributes. The Nexus uses the logical model to allow users to see tables and all other tables in the relationship. Then, the Nexus builds the SQL with each click of the user’s mouse.

PrimaryKeys Compared to Foreign Keys



Remember, a **Primary Key** is a **Unique Identifier** where a Primary Index is a **physical path** to the data. The **Primary Key is logical** and the **Primary Index is physical**.

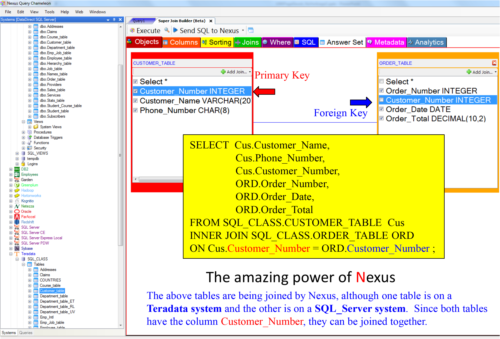
First,Second and Third Normal Form

The process of placing columns in the correct tables is called nomalization. There are three major forms of normalization, and they are first, second, and third normal form.

A **Primary Key** is usually the first column in the table and it is **always unique** and **not null** . The other columns in the table relate to the Primary Key. If for example, the Primary Key of the Customer\_Table is Customer\_Number. Once you know the Customer\_Number, you can find a great deal of information about that customer, because all other columns in the table are associated with that particular Customer\_Number, and it can be joined to other tables, such as the Order\_Table because both have corresponding rows with Customer\_Numbers that match. They can be joined because they have a Primary Key/Foreign Key relationship!

A **Foreign Key** is a normal column in the table that can be joined back to another table’s Primary Key. One table will have the Customer\_Number column as its Primary Key (such as the Customer\_Table) and the other will have Customer\_Number as a normal column in its table. Since they both have the same column Customer\_Number that represents their customers the tables can be joined (and are designed to do that. Thank you data modelers!

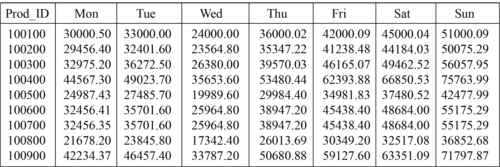
Take a good look at the picture on the next page. This will be the first time you or anyone else has ever seen this. I am joining two tables together, but the amazing part is that the tables come from two different systems. The Customer\_Table is on a Teradata system and the Order\_Table is on a SQL\_Server system. The Nexus allows you multiple ways to tell it the Primary Key/Foreign Key relationship between a series of tables. Nexus will remember the next time and guide you to what tables can be joined together. As you click on the columns you want on your report, Nexus automatically builds the SQL. Nexus then joins the data together and the user gets the report.



FirstNormal Form

First Normal Form (1NF) means that columns (attributes) must not repeat within a table. No repeating groups such as Monday\_Sales, Tuesday\_Sales, Wednesday\_Sales, etc. The table below is **NOT** in **First Normal Form**.

Sales\_Table at one of the largest data warehouses in the world showing Daily\_Sales

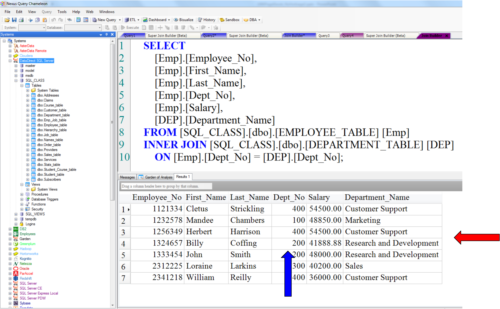


The above table actually saves space because, if the table did not use repeating groups, there would be 7 times the rows and 14 extra bytes of row overhead per row. Instead of 150,000 rows, this table would have over 1 million rows.

The above table is not about saving space. The customer needs to compare how Monday did versus Tuesday, etc. This table allows users to get the information about a product during a specific week by reading only one row and not seven. Although this violates first normal form, it can be a great technique for known queries.

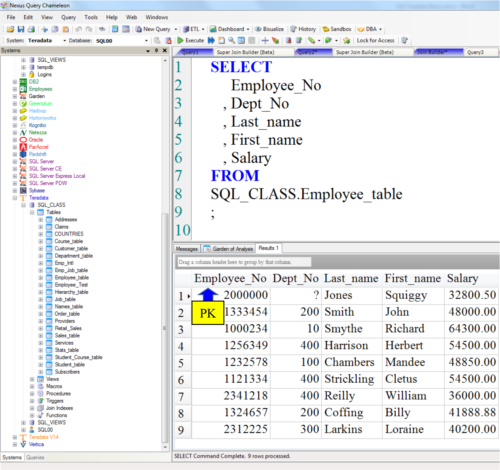
SecondNormal Form

Second Normal Form (2NF) means that all columns relate to the entire Primary Key and not just a portion of the Primary Key. Columns can also relate to each other, such as Dept\_No and Department\_Name, by both belonging to the same table. A table that has a single column Primary Key is always said to be in at least 2NF. If the below answer set was actually a table it would be in Second Normal Form.



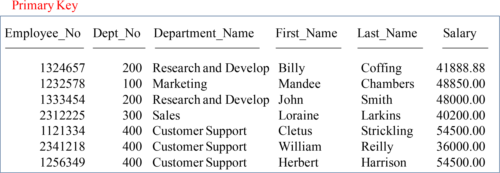
ThirdNormal Form

Third Normal Form (3NF) means all coumns must relate to the Primary Key and not to each other. This is often termed that all coumns must relate to the key, the whole key, and nothing but the key so help you Codd!



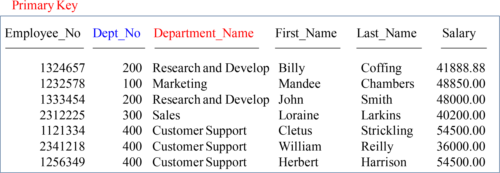
Quiz– Choose that Normalization Technique

Below is a table with rows and columns. What form of normalization is this example?



Answerto Quiz – Choose that Normalization Technique

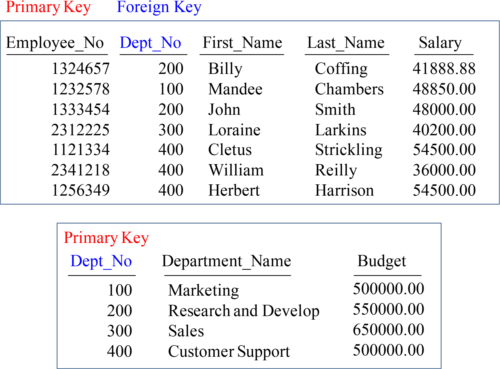
Below is a table with rows and columns. What form of normalization is this example?



This is second normal form (2NF)

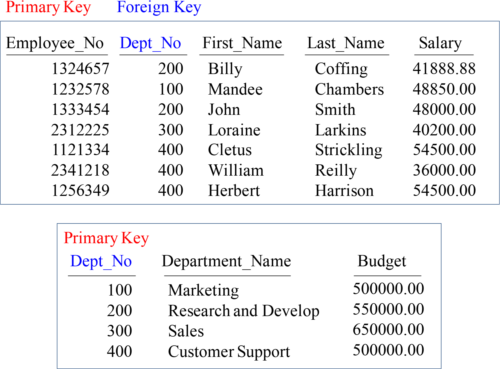
The reason is that although all columns relate to the Employee\_No, the Department\_Name is also a direct relation of the Dept\_No.

Quiz– What Normalization Is It Now?



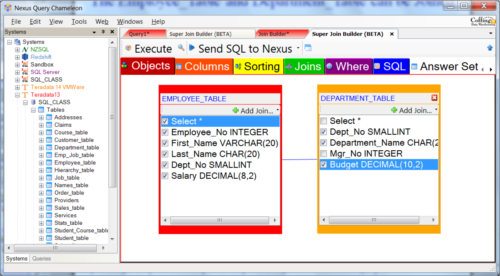
Above are tables with rows and columns. What form of normalization is this example?

Answerto Quiz – What Normalization Is It Now?



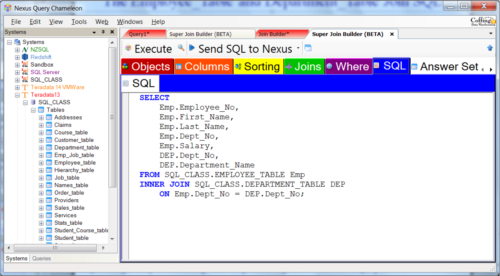
Both table are in third normal form (3NF). All columns relate to the Primary Key, the whole key, and nothing but the key. Tables are joined via the Dept\_No column that resides in both tables.

TheEmployee\_Table and Department\_Table Can Be Joined



The Nexus knows the join relationships between tables. Above, you can see that the user sees both tables, and they have clicked on the columns they want on the report. Turn the page and see the SQL that Nexus has built for them. Tables with a Primary Key-Foreign Key relationship can be joined together with SQL.

TheEmployee\_Table and Department\_Table Join SQL



Tables with a Primary Key-Foreign Key relationship can be joined together with the SQL you see above in the SQL tab of Nexus. Tools like Nexus can help users dramatically because having Nexus navigate the table relationships allows users to visually see the tables and their relationships and then let Nexus build the SQL.

DimensionalModeling

Dimensional modeling contains two types of tables. They are fact and dimension tables. A fact table will have fewer columns, but millions of rows potentially. There is only one fact table because it is the main table that contains the good information such as the products a company sells, sales totals, quantities and the dates, times and places they were sold. There will be many dimension tables. Dimension tables generally have more columns, but less rows. Each dimension table that joins back to the main fact table provides detail about the actual place or product or date it was sold. Benefits include easy querying , flexibility and improved performance .

