Slowly Changing Dimensions Exercise

Knowledge Management IV – Changing Dimensions Practical Assignment 1

This assignment focuses on a few of the changing dimension techniques discussed in Chapter 5 of your textbooks. For each of the scenarios below you will be required to write all SQL statements needed to implement the needed change(s).

Instructions:

You will be using the same database as for assignment I (Reporting for BI purposes). You may continue working on your existing copy, or download a new one.

Please note: **SQL that was generated by MS Access is not allowed**. You have to write the SQL in the SQL design view of Access and format it so that it is "clean" and human readable (NO INNER JOINS ALLOWED. Do not use a HAVING clause where a WHERE clause would work!).

For each scenario you have to write the Scenario number, which Changing Dimension Technique you used, why you chose this technique, and all SQL statements needed to implement the needed changes. Submit your answers as a single neatly formatted word document similar to the template below:

> ### Scenario:
> Scenario x
>
> ### Technique Used:
> Type X
>
> ### Motivation:
> These are my reasons
>
> ### SQL:
> SQL syntax here (If there are multiple statements needed list them separately in the correct sequence)

Scenarios

1. One of our product descriptions contain a mistake. The unit of measurement was misspelled as "garms" instead of "grams".

This is a MISTAKE. We can thus simply overwrite it.

```
UPDATE Product

SET Product.[Product Unit of Measure] = REPLACE (Product.[Product Unit of Measure], 'garms', 'grams');
```

2. Our No Name brand White Bread (product 7) recipe changed. This resulted in its diet type changing from “Not Diet” to “Diet”. Unfortunately, this was only picked up during an audit so incorrect sales data entries were already made into the database. The change happened from (including) the 4th of December 1997.

In this case we need to do several things. First we need to create a new record reflecting the changed dimension data (Type 2 change) but allowing us to retain the history. In addition to the new record we need to add columns to keep track of the effective date of the new one and the expiry date of the old one. Since no one indicated that they want to see the new and old data on the same line there is no need to add a column for that (NOT type 3). **Remember we DO NOT USE SQL DATES we use a DATE KEY from our DATE DIMENSION**

```
ALTER TABLE Product ADD [Row Effective Date] Integer;

ALTER TABLE Product ADD [Row Expiration Date] Integer;

ALTER TABLE Product ADD [Current Row Indicator] Integer;
```

```
UPDATE Product SET Product.[Row Effective Date] = 671,

Product.[Row Expiration Date] = 703,

 Product.[Current Row Indicator]= "Expired"

WHERE [Product Key] = 7;
```

```
INSERT INTO Product ([Product Key], [SKU], [Product Description], [Brand],[Diet Type],[Product Unit

of Measure],[Product size],[Row Effective Date],[Row Expiration Date],[Current Row Indicator])

VALUES (11, 100200, "White Bread", "No Name","Diet","grams",800,704,NULL,"Current");
```

FINALLY, we need to fix the mistake we made in the fact table!

```
UPDATE [Sales Fact]

SET [Product Key] = (the value of the new key generated above)

WHERE [Product Key] = 7 AND [Sales Date] >= 704
```

3. Our PE store was rebranded as the Nelson Mandela Bay store from the 14th of December 1997. Our managers have indicated that they would like to see comparisons of the data for both the old and new store brand name on a single report.

The need to see both entries on one row implies a Type 3 change. IF this happens again in future we'll deal with subsequent changed using a Type 2.

ALTER TABLE Stores ADD [Prior Store Name] Short text;

UPDATE [Stores]

SET [Store Name] = "Nelson Mandela Bay store", [Prior Store Name] = "PE Hypermarket"

WHERE [Store Key] =1;

4. The store recently decided to launch a customer loyalty program. This program will become **active on the 1st of January 2000** (I know this is in the past, pretend it is not). We want to track what our customers buy in order to create targeted marketing campaigns in future. We will mail reward vouchers to customers, so we are pretty sure they will keep their details updated in our database. The following data is gathered when a customer signs up for the program.
    a. Title
    b. Name
    c. Surname
    d. Gender
    e. Marital Status
    f. Date of Birth
    g. Age
    h. Address
    i. Postal Code

    Our management would like the loyal membership details linked to our retail sales schema.

    The answer to this depends on whether or not we keep track of any data already. I'll assume we do not. We therefore need a table for this. Note Date of Birth is set as an INT

    CREATE TABLE Customers([Customer Key] int primary key,[Title] varchar(10),[Name] varchar(30), [Surname] varchar(30), [Gender] char(1), [Marital Status] varchar(30), [Date of Birth] int, [Age] varchar(50),[Address] varchar(50), [Postal Code]varchar(50));

    We will also need to add a column to the Fact Table and then have a customer row in the Customer dimension to indicate PRIOR to the scheme