|  |
| --- |
| EPAM Systems, RD Dep. |
| Data Warehouse Architecture |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial status | Valeryia\_Lupanava | 11-NOV-2017 |  |  |

Contents

[1. Data Modeling Task 3](#_Toc498134338)

[2. Analytical task 6](#_Toc498134339)

[2.1. 3NF-model 9](#_Toc498134340)

# Data Modeling Task

The following script will create a date dimension table for a standard calendar. It starts by creating a table with dimension members and labels. The second part of the script fills in end date attributes. The section that creates end date can be easily adapted for completing other calendars where the members have already been filled in.

Calendar starts from **01.01.2017** and includes **10950 days ≈ 30 years**.

**DATE DIMENSION SCRIPT:**

CREATE TABLE date\_dimension AS

WITH base\_calendar AS (

SELECT

n AS date\_id,

start\_date AS full\_date,

TO\_CHAR(start\_date,'Day') AS week\_day\_full\_name,

TO\_CHAR(start\_date,'DY') AS week\_day\_short\_name,

to\_number(TRIM(LEADING '0' FROM TO\_CHAR(start\_date,'D') ) ) AS day\_number\_of\_week,

to\_number(TRIM(LEADING '0' FROM TO\_CHAR(start\_date,'DD') ) ) AS day\_number\_of\_month,

to\_number(TRIM(LEADING '0' FROM TO\_CHAR(start\_date,'DDD') ) ) AS day\_number\_of\_year,

upper(TO\_CHAR(start\_date,'Mon')

|| '-'

|| TO\_CHAR(start\_date,'YYYY') ) AS month\_year,

TO\_CHAR(start\_date,'Month') AS month\_full\_name,

TO\_CHAR(start\_date,'Mon') AS month\_short\_name,

to\_number(TRIM(LEADING '0' FROM TO\_CHAR(start\_date,'MM') ) ) AS month\_number\_of\_year,

'Q'

|| upper(TO\_CHAR(start\_date,'Q')

|| '-'

|| TO\_CHAR(start\_date,'YYYY') ) AS quarter\_year,

to\_number(TO\_CHAR(start\_date,'Q') ) AS quarter\_number\_of\_year,

CASE

WHEN to\_number(TO\_CHAR(start\_date,'Q') ) <= 2 THEN 1

ELSE 2

END

AS half\_year\_number,

CASE

WHEN to\_number(TO\_CHAR(start\_date,'Q') ) <= 2 THEN 'H'

|| 1

|| '-'

|| TO\_CHAR(start\_date,'YYYY')

ELSE 'H'

|| 2

|| '-'

|| TO\_CHAR(start\_date,'YYYY')

END

AS half\_year,

TO\_CHAR(start\_date,'YYYY') AS year

FROM

(

SELECT

level n,

-- Calendar starts at the day after 31/12/2016.

TO\_DATE('31/12/2016','DD/MM/YYYY') + numtodsinterval(level,'DAY') start\_date

FROM

dual

-- The number of days to be added to the table.

CONNECT BY

level <= 10950

)

) SELECT

date\_id,

full\_date,

week\_day\_full\_name,

week\_day\_short\_name,

day\_number\_of\_week,

day\_number\_of\_month,

day\_number\_of\_year,

MAX(

date\_id

) OVER(PARTITION BY

month\_year

) AS month\_end\_date,

month\_full\_name,

month\_short\_name,

month\_number\_of\_year,

month\_year,

quarter\_year,

quarter\_number\_of\_year,

half\_year\_number,

half\_year,

year,

MAX(

date\_id

) OVER(PARTITION BY

year

) AS year\_end\_date

FROM

base\_calendar

ORDER BY date\_id;

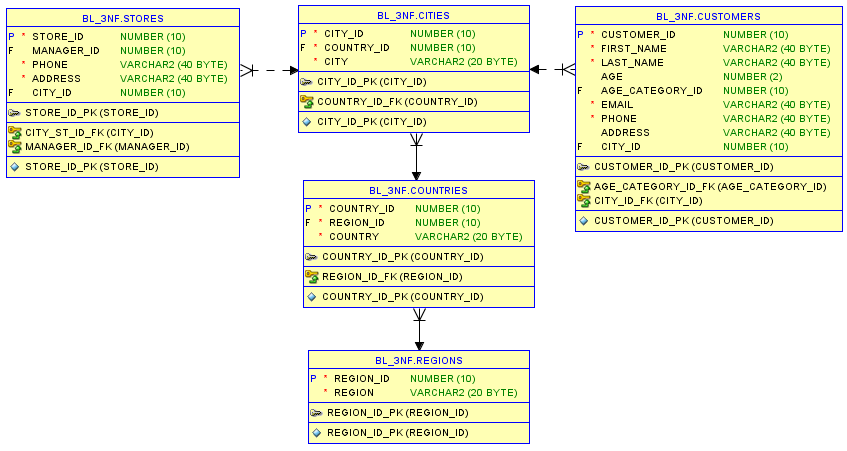
After creation I added **PRIMARY KEY** on the field **DATE\_ID**.

ALTER TABLE date\_dimension ADD CONSTRAINT date\_id\_pk PRIMARY KEY ( date\_id );

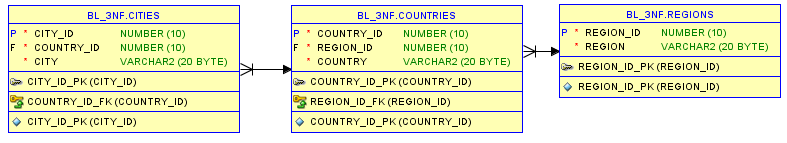
# Analytical task

In this model the normalization was made with next steps:

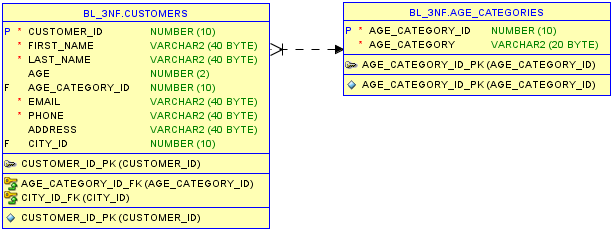
1. Geographical information from the “Stores” and “Customers” was separated on different tables. Tables were created for each geographical object: region -> country -> city. These objects were connected in series: from the lowest in granularity to the highest.



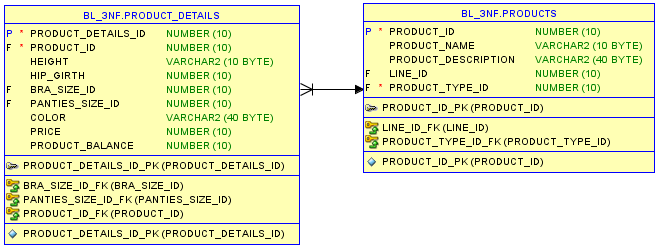
1. Information about products was organized in the same way. There were separate tables for each object: collections -> lines -> products. These objects were connected in series: from the lowest in granularity to the highest.



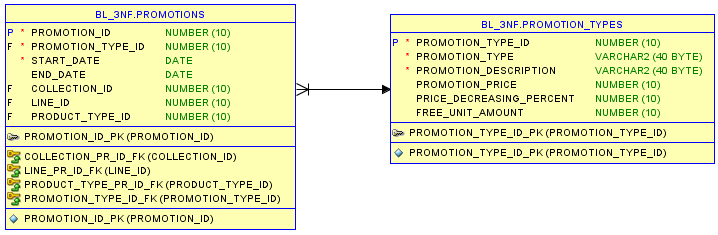
1. A separate “Age\_categories” table was created for Customers. This table describes all possible age ranges that are important for business. Subsequently, each Customer can be assigned to a specific age category.



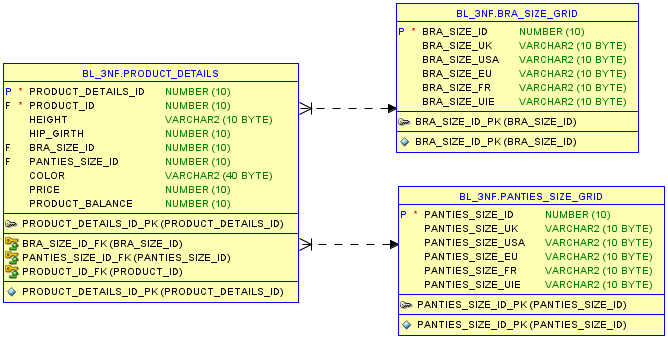
1. Information about product characteristics was separated in a specific table “Product\_details”. It contains a complete description of each commodity unit of a certain size.



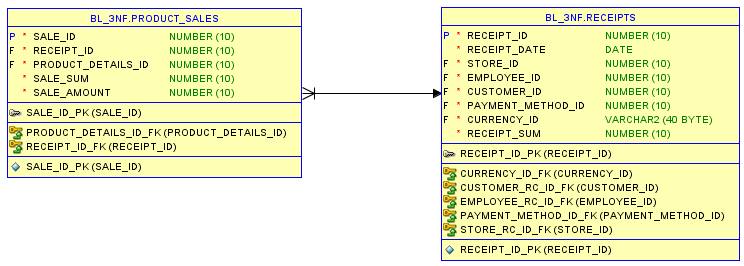
1. Also, the object “Promotions” was detailed as follows. Separately, a table was created on all possible types of promotions that were conducted on the network and was linked to a list of standard promotions.



1. The tables of the size grids “Bra\_size\_grid” and “Panities\_size\_grid”, which contain information on the international dimension matching for bra and penities, were separately created.



1. Additionally receipts information was add. It includes the calculation of their total cost.
2. The “Product\_sales” table details the check information, namely it describes a concrete product and amount of units what were purchased.



## 3NF-model

