**Report**

**on the**

**U1M6.LW.Star Schema Basics**

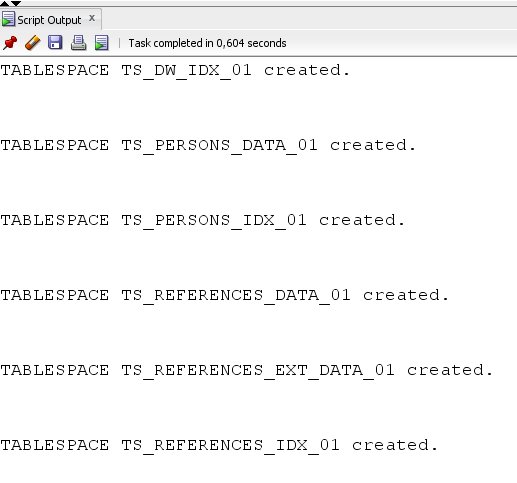
**Alina Sadovskaya**

2. OLTP – Load External References – Normalization of Data

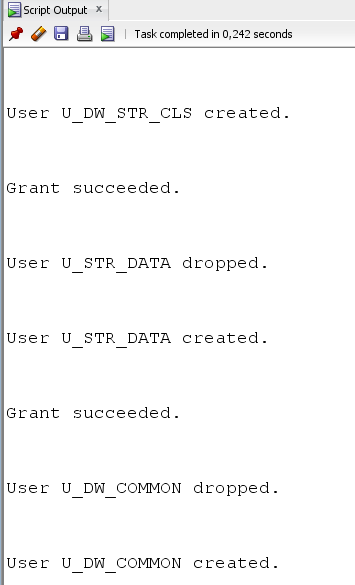
2.1. Task 01 – Install and expand load of external references T\_Languages

Run the necessary scripts:

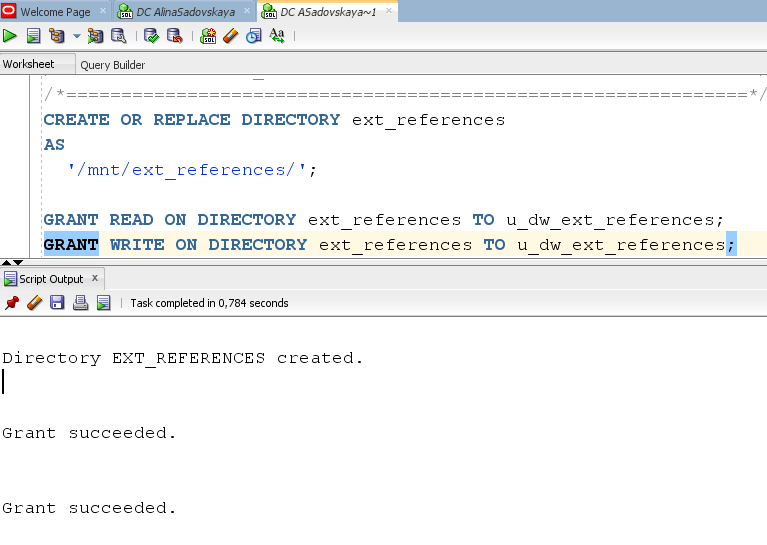
1)Сreate tablespace:



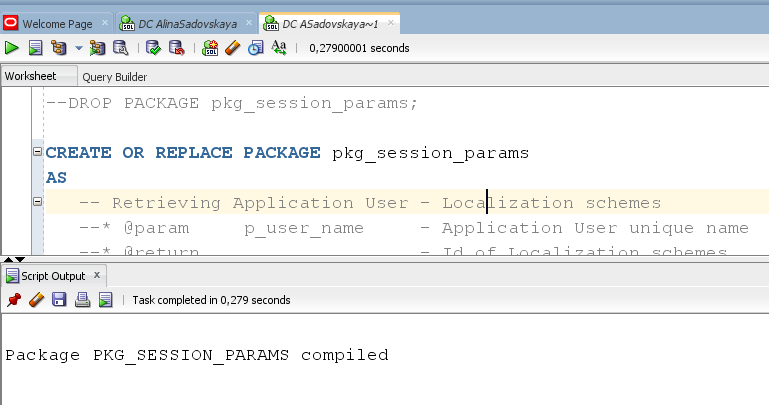
2)Сreate users:



3)Init directories:



4)Create package pkg\_session\_params:



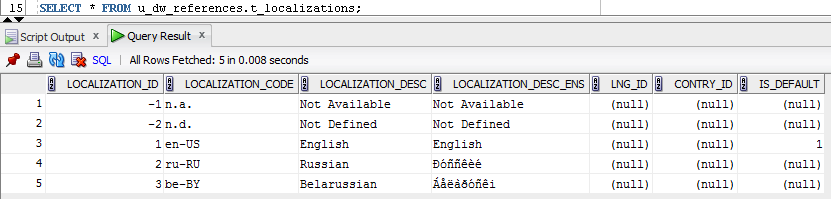
5) Run the script from the following folders in a certain order:

1. /u\_dw\_references
2. /u\_dw\_ext\_references
3. /u\_dw\_common

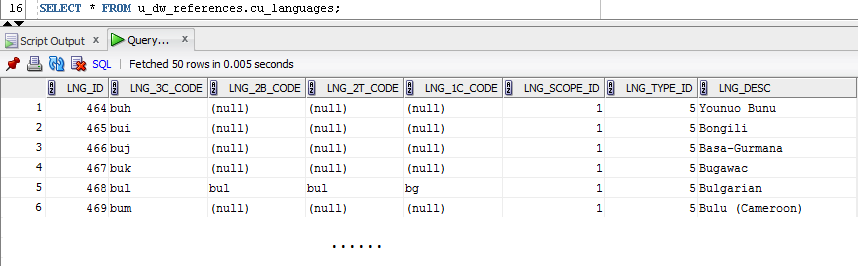
**As a result, the necessary tables were created!**

Showing result of data on next objects:

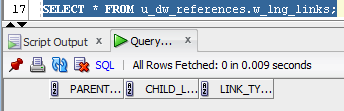
* t\_localizations



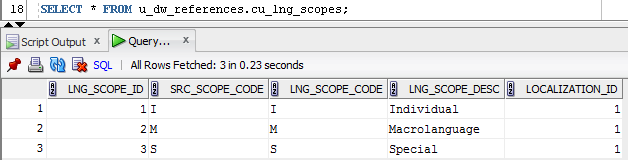
* cu\_languages



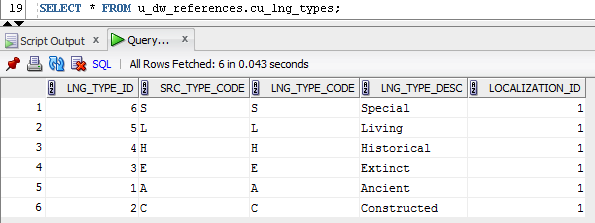
* w\_lng\_links



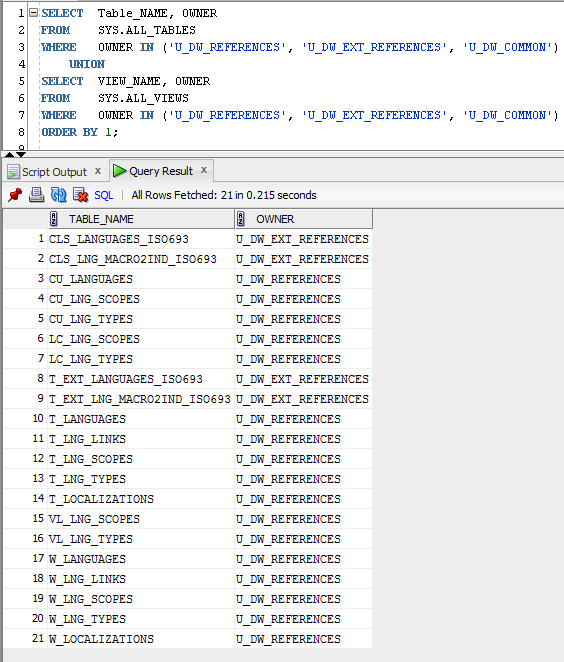
* cu\_lng\_scopes



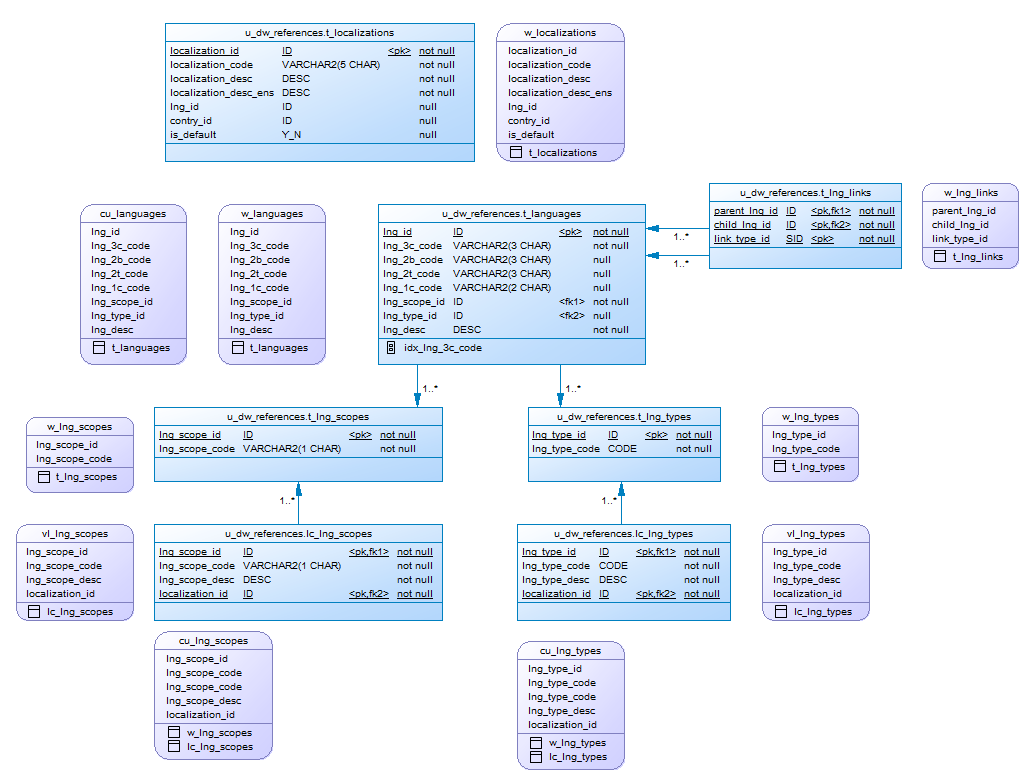
* cu\_lng\_types

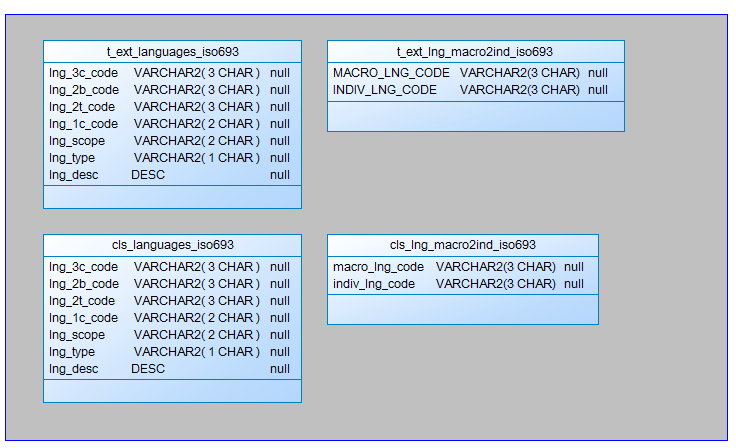


We nead to create sql scripts to show All created Tables and Views:

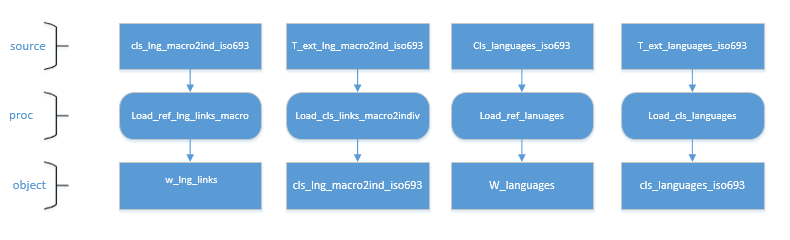


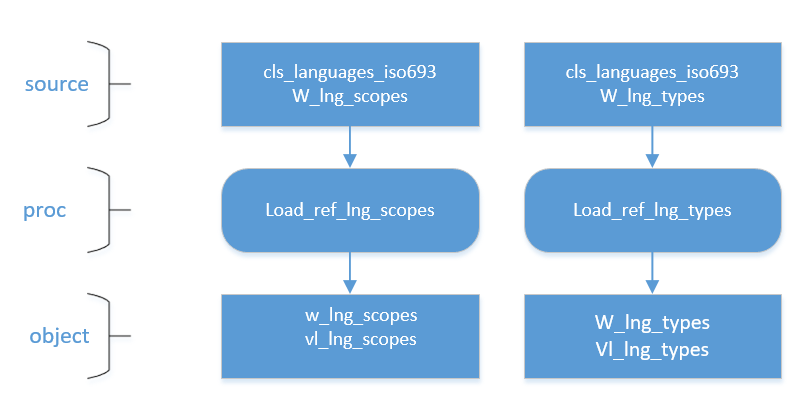
The Physical Diagram of T\_Languages below:





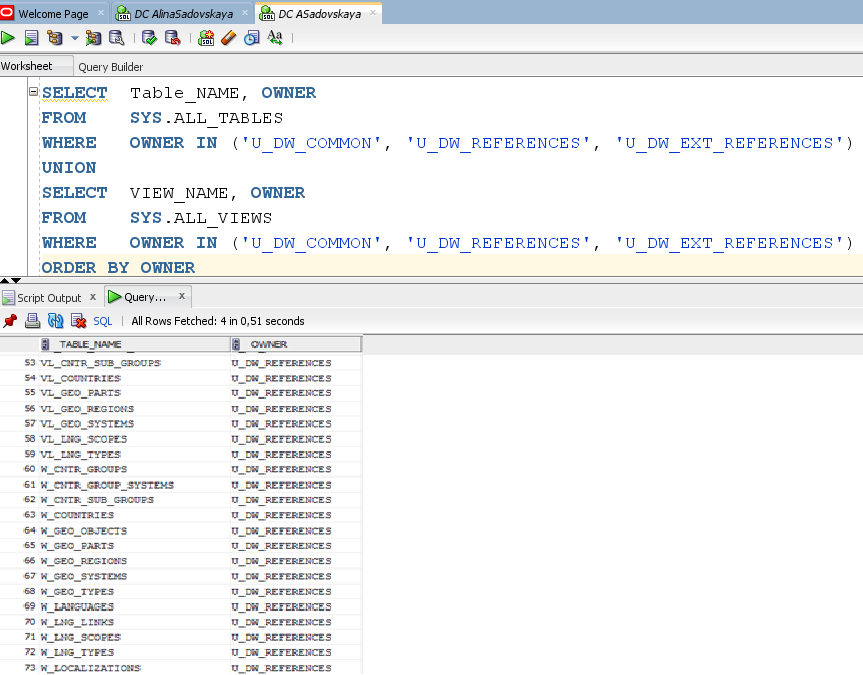
**DataFlow**





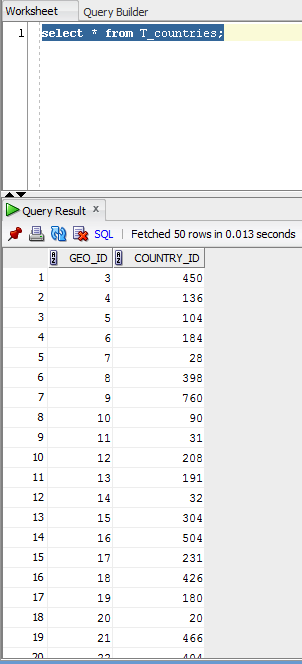
## 2.2. Task 02 – Create load process for External references T\_Countries

After running all the necessary scripts, the following tables and views were created:

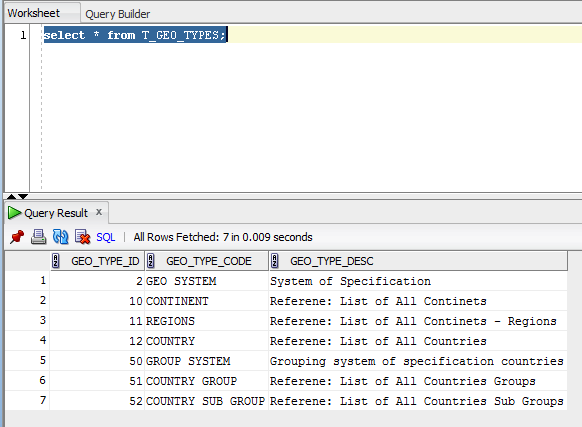


Showing result of data on next objects:

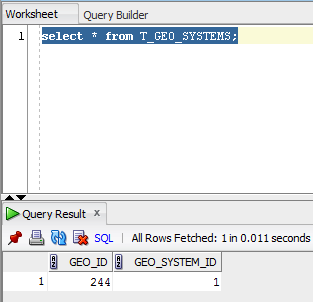
* T\_Countries



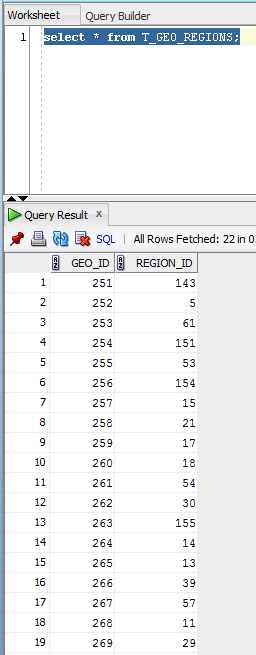
* T\_Geo\_Types



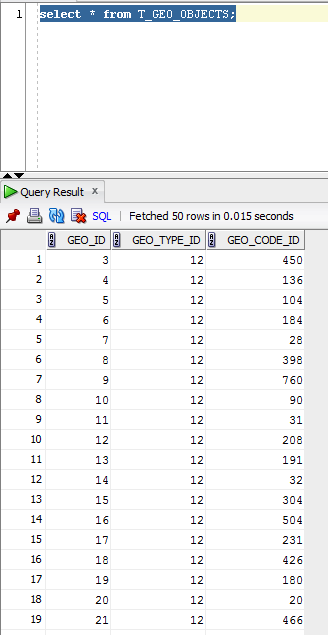
* T\_Geo\_Systems



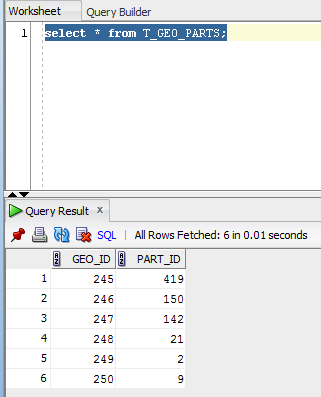
* T\_Geo\_Regions



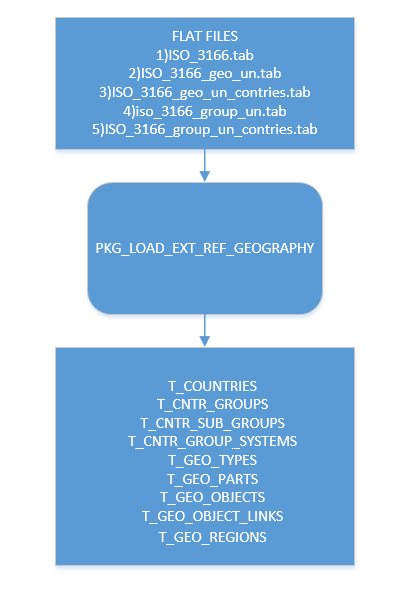
* T\_Geo\_Objects



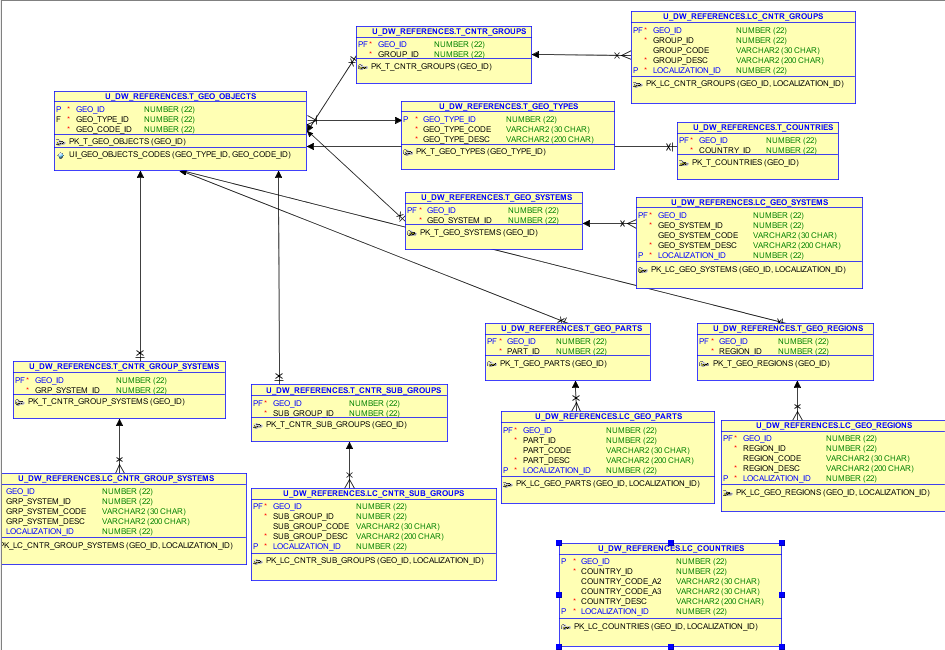
* T\_Geo\_Parts



Create DataFlow: Sketch Diagram of loading external References



Prepare The Physical Diagram of T\_Countries:



# 3. OLAP – Business analyses task

As an example of a business idea, consider the work of the Belarusian clothing brand Mark Formelle.

Brief overview of the brand Mark Formelle:

For 10 years, MARK FORMELLE has been producing clothing for home, life and sports, helping customers lift their spirits and giving them ease and self-confidence. The company's products are successfully sold not only on the territory of Belarus, but also far beyond its borders. The production process involves both traditional cotton, viscose, and synthetic materials, but also uses the latest developments of fabrics with an admixture of polyester, as well as bamboo and modal. The company's professionals participate in the development and design. They also come up with their own prints and drawings. All masters have specialized education and have been trained in the best institutions. The main vectors of activity are: clothing for everyday life, work and leisure called Life, options for active athletes line Sport, insulated underwear line Zima, as well as underwear, socks, stockings, tights.

Let this clothing brand face certain difficulties related to incorrect data handling. Some of the possible problems:

1. lack of business intelligence from multiple sources;

2. reduced query and system performance;

3. lack of timely access to data;

4. the lack of historical intelligence.

We offer the customer to use DWH(we will consider 2 types of storage schemes: star scheme and snowflake scheme) for the following reasons:

1. It is better to make decisions. Corporate decision makers will no longer have to make important business decisions based on limited data and guesswork. The data warehouse will store reliable facts and statistics, and decision makers will be able to extract this information from the data warehouse based on their personal needs.

2. Quick and easy access to data. Speed is an important factor that puts the company above its competitors. Business users can quickly access data from multiple sources from the data warehouse, which means that precious time will not be wasted extracting data from multiple sources. This allows the company to make fast and accurate decisions, with little or no support from the it Department.

3. The quality and consistency of data. As data warehouses collect information from various sources and convert it into a single and widely used format, departments will produce results that are consistent and consistent with each other. When data is standardized, a company can be confident in its accuracy, and accurate data is what makes it possible to make strong business decisions.

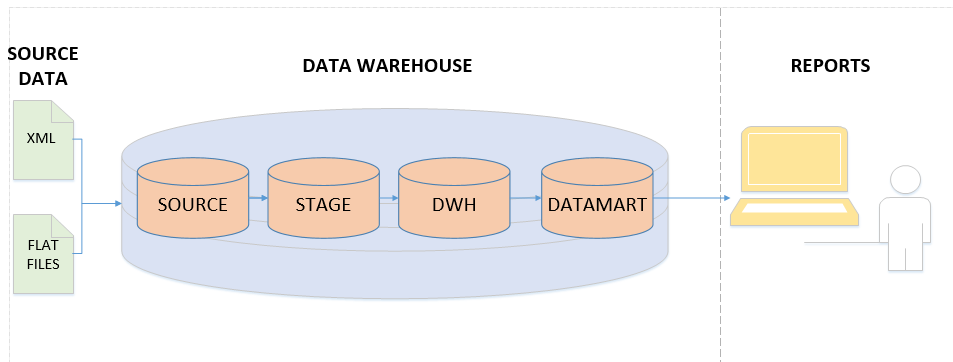
## Business Requirements:

| **ID** | **Description** |
| --- | --- |
| B01 | Сalculation of sales information (Volume, Quantity) on a monthly basis(you can collect information for large time periods due to the time hierarchy) for each product |
| B02 | Сalculation of information about the benefits of used shares for products on a monthly basis |
| B03 | - Control information about total sales for each product category for each country of concern if it is less than 3% of the region's sales for that product |
| B04 | Our brand has several stores in different countries. Each office has its own geographical location. The warehouse must take this fact into account |
| B05 | Storing information about products in the warehouse, for quick search of the selected product |

## Technical Requirements:

| **ID** | **Description** |
| --- | --- |
| T01 | Keep info from the beginning of business |
| T02 | Clean products that have been discontinued |
| Т03 | Тhe access time to the storage– 24/7/365 |
| Т04 | High performance and high availability |
| T05 | All the information must be protected according to the company’s security Policy |
| T06 | Ability to process large amounts of information per day (1 million rows) |

## DWH Solution Sketch



In paragraphs 4.1 – 4.2, we will consider 2 main types of schemes, identify their advantages and disadvantages, and select the most appropriate scheme for our case.

# 4. OLAP – Develop Star-Scheme and SnowFlake Scheme.

4.1. Task 04 – Develop Star-Scheme physical diagram

As a Business process for analysis, let's take the company's sales by various indicators. Fact granularity is Daily Sales Amount per certain Customer and Employee in a specific location.

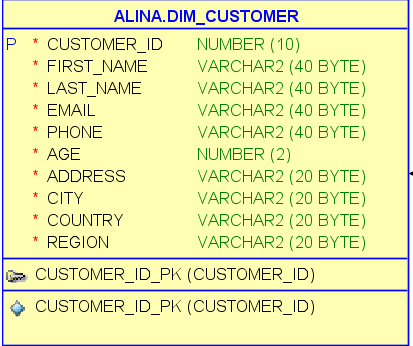
To complete the task we need to complete several steps:

1. Aggregate sales data by calculating for each product the total number of products sold and the total amount of money paid per customer. This result requires creating a nested table.
2. Attach aggregated data to customer data.
3. Create an output table of aggregated data to improve build performance. This table serves as a source for classification models.
4. Build classification models based on aggregated data.

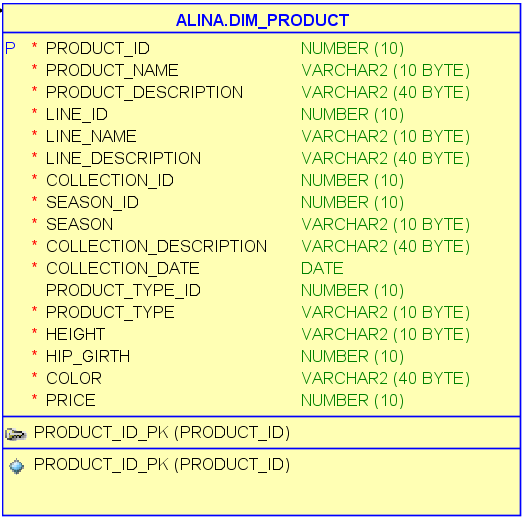
Lets identify the Dimensions:

Schema should contains next dimensions:

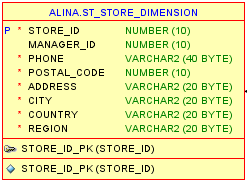
1. Customer dimension (Primary Key is Customer\_ID)



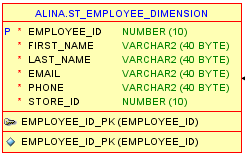
2. Product dimension (Primary Key is Product\_ID)



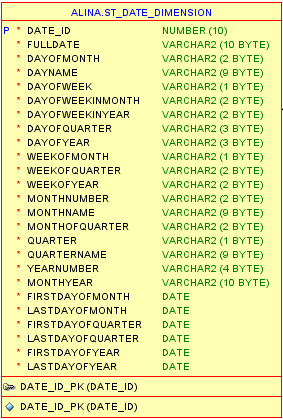
3. Store dimension (Primary Key is Store\_ID)



4. Employee dimension (Primary Key is Employee\_ID)



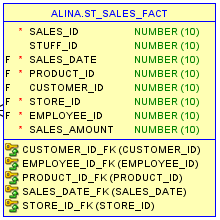
5. Date dimension (Primary Key is Date\_ID)



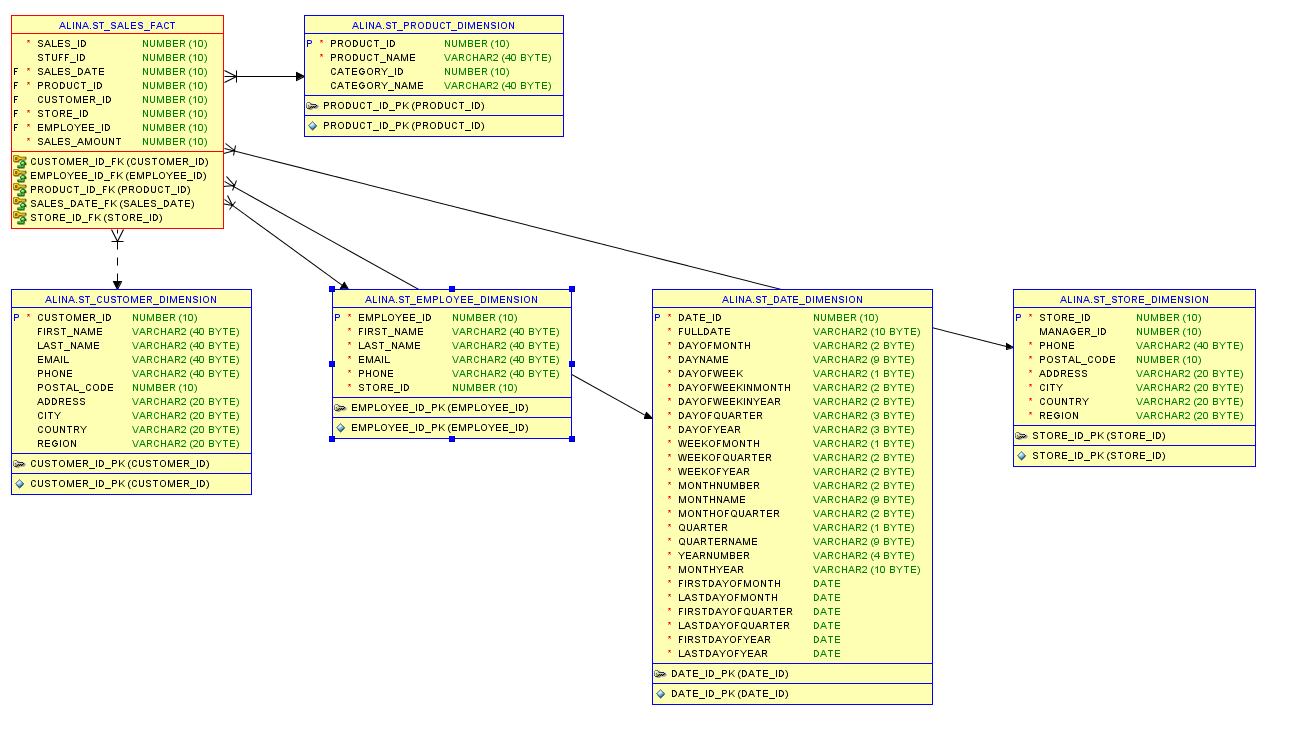
Lets identify the Facts:

Schema should contains the fact table “Sales” with next data:

1. Sales\_ID;
2. Stuff\_ID;
3. Sales\_Date as Foreign Key from Date\_Dimension;
4. Product\_ID as Foreign Key from Product\_Dimension;
5. Customer\_ID as Foreign Key from Customer\_Dimension;
6. Store\_ID as Foreign Key from Store\_Dimension;
7. Employee\_ID as Foreign Key from Employee\_Dimension;
8. Sales\_Amount as measure of Sales.



**Star Scheme diagram:**



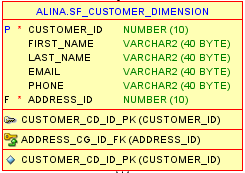
## 4.2. Task 05 – Develop SnowFlake physical diagram

As a Business process for analysis, let's take the company's sales by various indicators. Fact granularity is Daily Sales Amount per certain Customer and Employee in a specific location.

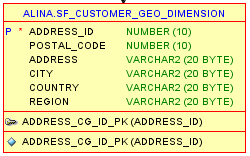
## **Lets identify the Dimensions:**

Schema should contains next dimensions:

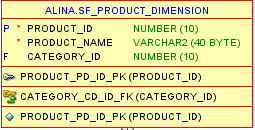
1. Customer dimension (Primary Key is Customer\_ID)



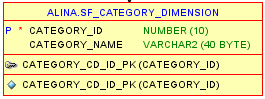
2. Customer geo dimension (Parent table is Customer Dimension; Primary Key is Address\_ID)



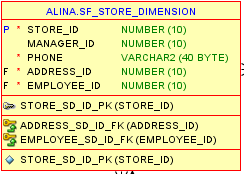
3. Product dimension (Primary Key is Product\_ID)



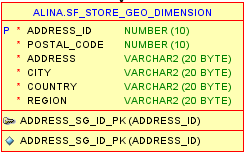
4. Category dimension (Parent table is Product Dimension; Primary Key is Category\_ID)

****

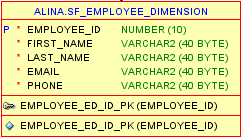
5. Store dimension (Primary Key is Store\_ID)



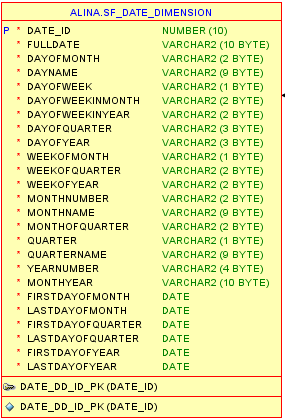
6. Store geo dimension (Parent table is Store Dimension; Primary Key is Address\_ID)



7. Employee dimension (Parent table is Store Dimension; Primary Key is Employee\_ID)



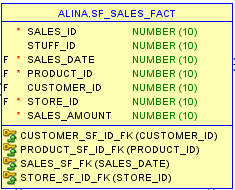
8. Date dimension



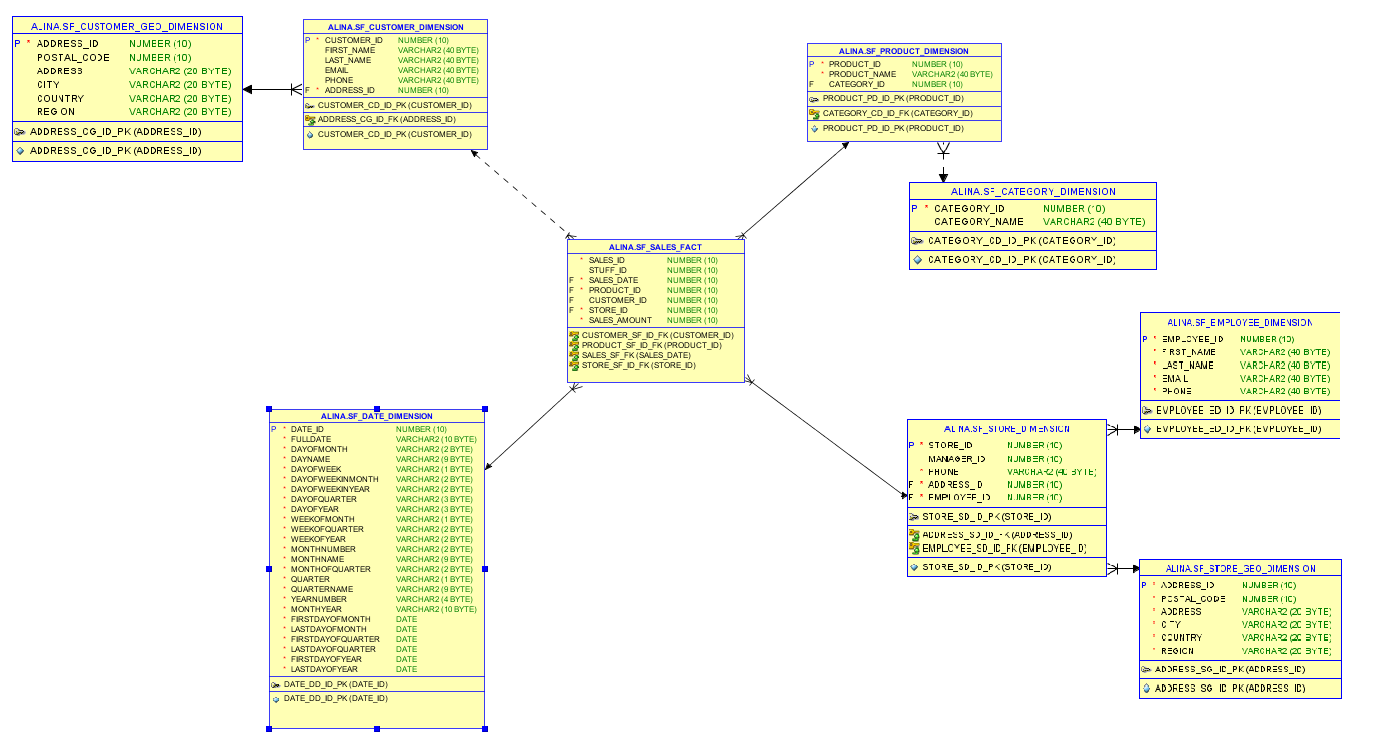
## Lets identify the Facts:

Schema should contains the fact table “Sales” with next data:

1. Sales\_ID;
2. Stuff\_ID;
3. Sales\_Date as Foreign Key from Date\_Dimension;
4. Product\_ID as Foreign Key from Product\_Dimension;
5. Customer\_ID as Foreign Key from Customer\_Dimension;
6. Store\_ID as Foreign Key from Store\_Dimension;
7. Employee\_ID as Foreign Key from Employee\_Dimension;
8. Sales\_Amount as measure of Sales.



**SnowFlake diagram:**



Let's compare the considered schemes and choose the most suitable one in our case. The most obvious characteristic of the star schema is that dimension tables are not normalized. In the model above, the pink **fact\_sales** table stores aggregated data created from our operational database(s). The snowflake schema stores exactly the same data as the star schema. The fact table has the same dimensions as it does in the star schema example. The most important difference is that the dimension tables in the snowflake schema are normalized.

**The First Difference: Normalization**

As mentioned, normalization is a key difference between star and snowflake schemas. Regarding this, there are a couple of things to know:

* Snowflake schemas will use less space to store dimension tables. This is because as a rule any normalized database produces far fewer redundant records .
* Denormalized data models increase the chances of data integrity problems. These issues will complicate future modifications and maintenance as well.

### **The Second Difference: Query Complexity**

Obviously, the snowflake schema query is more complex. Because the dimension tables are normalized, we need to dig deeper to get the name of the product type and the city. We have to add another JOIN for every new level inside the same dimension.

In the star schema, we only join the fact table with those dimension tables we need. At most, we’ll have only one JOIN per dimension table. And if we’re not using a dimension table, we don’t even need to bother with it. In the snowflake schema query, we don’t know how deep we’ll have to go to get the right dimension level, so that complicates the process of writing queries.

Joining two tables takes time because the DMBS takes longer to process the request.  Basically, a query ran against a snowflake schema data mart will execute more slowly. But in most cases this won’t present a problem: it doesn’t matter much if we get the result in one millisecond or one second.

To speed up reporting, we can:

* Aggregate data to the level we need in reports. This will compress the data significantly. We’ll need to create procedures that will transform our live data to fit into the reporting schema structure (the ETL process).
* Build a central storage area for **all** the company’s aggregated data, not just the sales data.
* Only give users the data they need for analysis and reports.

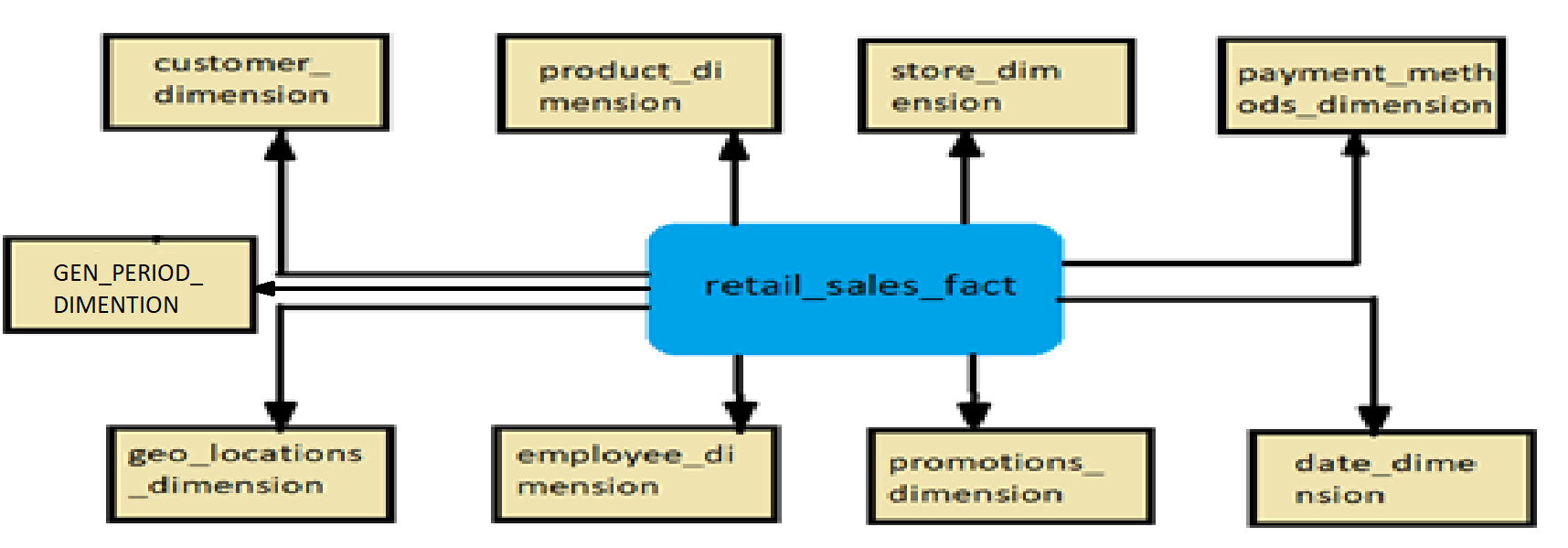
Both the star schema and the snowflake schema are relational models used for organizing data stores and/or data storefronts. No matter how similar they are, they demonstrate two different approaches and have their advantages and disadvantages. In our case, the most appropriate solution is to choose the star scheme for data storefronts (to make life easier for business users). The main reasons:

* Easily understood: a star schema is easy to understand and navigate, with dimensions joined only through the fact table
* Query performance: because a star schema database has a small number of tables and clear join paths, queries run faster
* Widely support by a large number of business intelligence tools

In our scheme, these business rules must be followed for this business task:

* Each retail sale must be associated with a valid customer.
* Retail sales are always linked to the payment method.
* Retail sales may have one or more products.
* Retail sales are always associated with a specific employee.

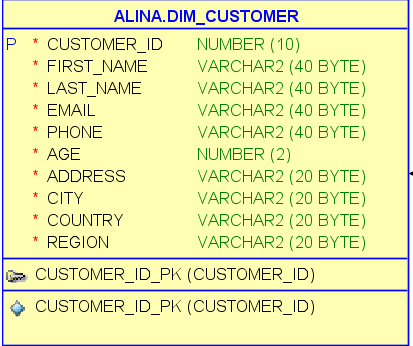
We have this logical diagram:



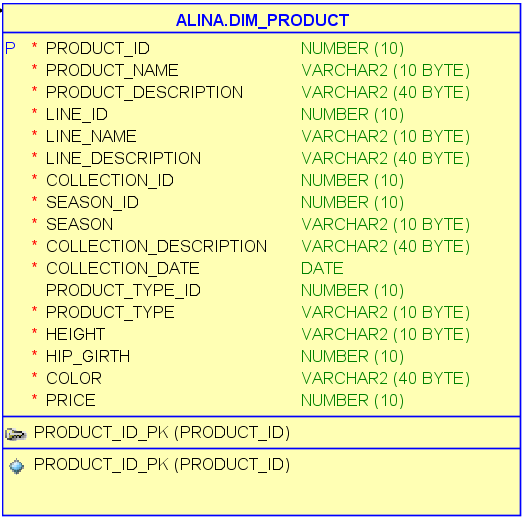
## Lets identifying the Dimensions

Schema should contains next dimensions:

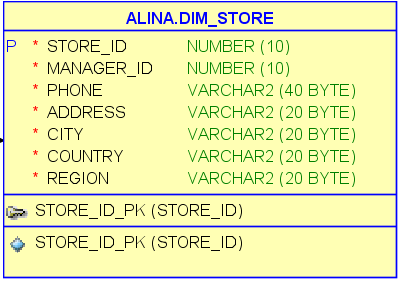
1. Customer dimension (Primary Key is Customer\_ID)



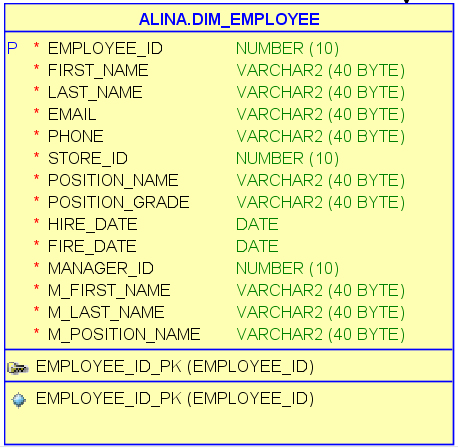
2. Product dimension (Primary Key is Product\_ID)



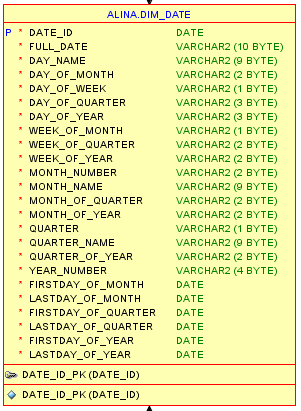
3. Store dimension (Primary Key is Store\_ID)



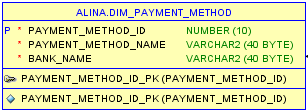
4. Employee dimension (Primary Key is Employee\_ID)



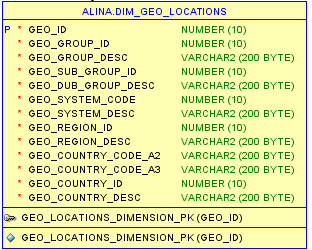
5. Date dimension (Primary Key is Date\_ID)



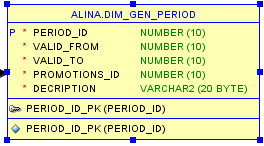
6. PAYMENT METHOD DIMENTION



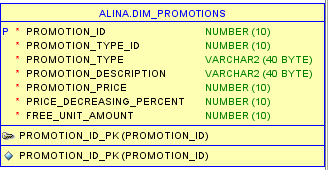
7. GEO LOCATION DIMENTION



8. GEN PERIOD DIMENTION

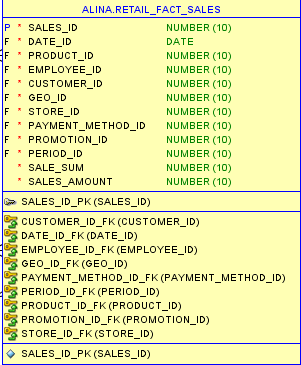


9. PROMOTION DIMENTION

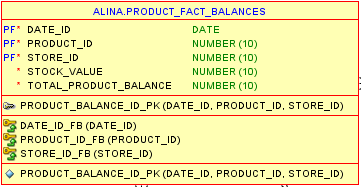


## Lets identify the Facts

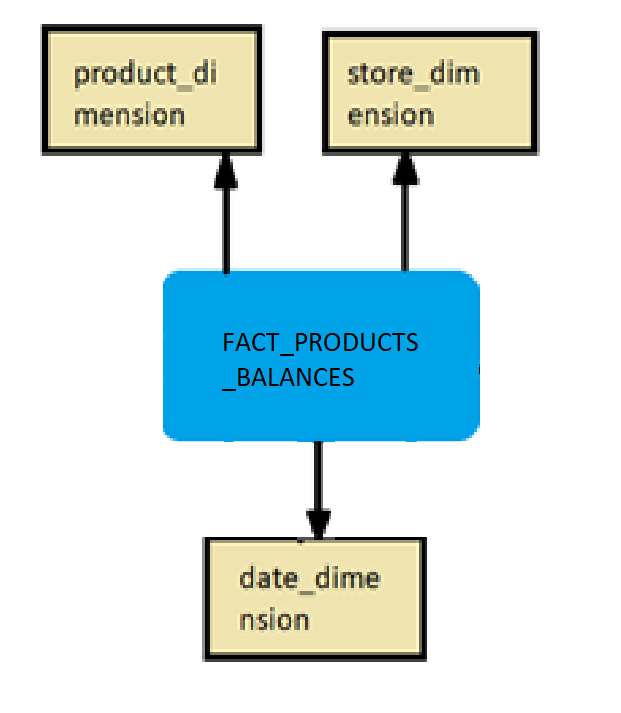
## 1) Retail\_fact\_ sales



2) FACT\_PRODUCT\_BALANCES

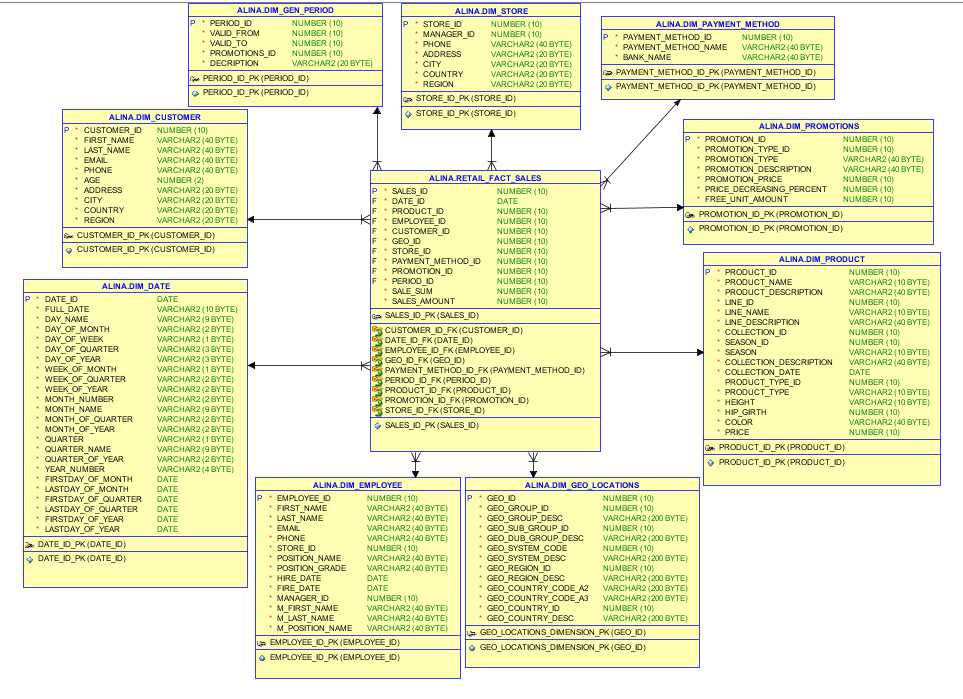


I think it is necessary to add another fact table that stores information about the remaining products in stock.



As a result we have the following **STAR SCHEME**:

1)fact table – RETAIL\_SAILES\_FACT



2)fact table –FACT\_PRODUCT\_BALANCE

