****

**EPAM Training Center**

**DWH**

**Tasks 6**

**Report**

**Minsk, 2017**

Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 08.11.2017 | 0.1 | Description of workflow (Tasks 6) | Olga Hilko |
|  |  |  |  |

Contests

[Introduction 4](#_Toc497925869)

[Purpose 4](#_Toc497925870)

[1 Business Model 4](#_Toc497925871)

[1.1 Business Process Description 4](#_Toc497925872)

[1.2 Grain 4](#_Toc497925873)

[1.3 Dimensions 4](#_Toc497925874)

[1.4 Fact 4](#_Toc497925875)

[2 Star scheme 5](#_Toc497925876)

[2.1 SQL code 5](#_Toc497925877)

[3 Snowflake sheme 7](#_Toc497925878)

# Introduction

## Purpose

This document includes the results of completed task 6.

# Business Model

## Business Process Description

In order to identify the business process

The present logic data model describes the process of making orders and holding payments for the goods. The main data describe the sales with their details by stores in different contexts.

It is useful to notice several peculiarities of the present logical model:

* Only entity Payment has a numeric measurement (amount),
* There is no currency codes and other countable values in spite of amount in Payment entity,
* All entities have update\_date attribute,
* All entities are divided into logical groups (Products, Customer data, Geo data, Business).

So, in order to the top description we have data on warehouse-shop. One order (Sale) can be made by one Customer and served by an Employee, but payed in several steps (Payments) by the same or another Customer. Besides this the Payment can be taken by the Stuff, which differs from the Employee, who severed the order. In one Order only one Product can present. The order (Sale) is made in one Store.

## Grain

The grain of the business process is Payment event, which deals with the Order (Sale). We have 1:M relation between Sales and Payments, so Payments have the less granularity. Moreover, only Payment has a numeric measurement (amount), Sales do not have anyone. So our grain describes an atomic fact of Payment-event.

## Dimensions

This Payment-event is described in different contexts (linked with Customers, Products, Stores, Stuff or Employees) in time. Geo data is not a specific characteristic of the Sale, it is linked with Customers, Stores and Employees.

So we have the next dimensions (for attributes – see the code description in the paragraph 2.1):

* Customers (with their Address, but not in detailed form),
* Products (with Categories),
* Stores (identified the stores, their Manager’s info and their detailed Location: Address, City, Country, Region, EconRegion),
* Employees (describes Employees and Staff, includes their addresses),
* Date (each dimension has date-field, but they do not link with the DimDate, only Sales\_date references to this dimension).

## Fact

There is no currency codes and other countable values in spite of **amount** in Payment entity. That is why it was taken as a measurement.

Moreover, it illustrates the main fact of payment for some amount on the Product.

The fact is characterized by:

Customer\_id\_sales – person who made the order,

Customer\_id\_payment – person who payed for the order,

Employee\_id – stuff who server the order,

Staff\_id – stuff who took the payment,

Sales\_date\_id – date of the order was made,

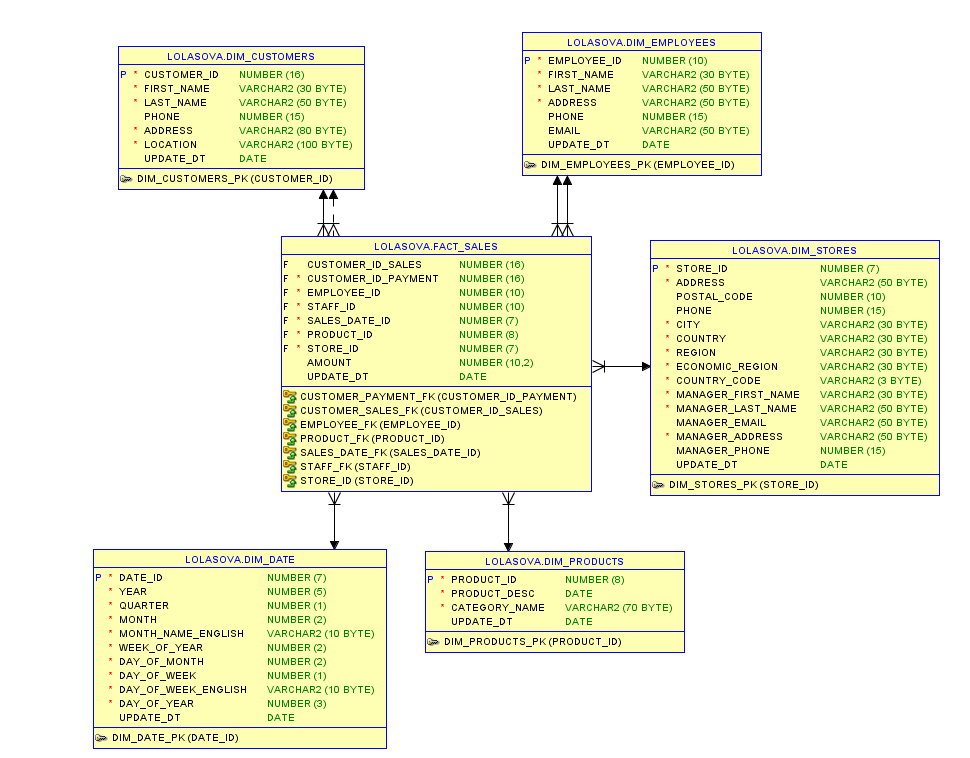
Product\_id – ordered product,

Store\_id – store where the order was made,

***Amount*** – the payed quantity of the product,

Update\_dt – served attribute, which illustrates when last changes in the database record were made.

# Star scheme



## SQL code

CREATE TABLE Dim\_Products (

Product\_id NUMBER(8) PRIMARY KEY,

Product\_desc DATE NOT NULL,

Category\_name VARCHAR2(70) NOT NULL,

Update\_dt DATE

);

CREATE TABLE Dim\_Customers (

Customer\_id NUMBER(16) PRIMARY KEY,

First\_name VARCHAR2(30) NOT NULL,

Last\_name VARCHAR2(50) NOT NULL,

Phone NUMBER(15),

Address VARCHAR2(80) NOT NULL,

Location VARCHAR2(100) NOT NULL,

Update\_dt DATE

);

CREATE TABLE Dim\_Stores (

Store\_id NUMBER(7) PRIMARY KEY,

Address VARCHAR2(50) NOT NULL,

Postal\_code NUMBER(10),

Phone NUMBER(15),

City VARCHAR2(30) NOT NULL,

Country VARCHAR2(30) NOT NULL,

Region VARCHAR2(30) NOT NULL,

Economic\_region VARCHAR2(30) NOT NULL,

Country\_code VARCHAR2(3) NOT NULL,

Manager\_first\_name VARCHAR2(30) NOT NULL,

Manager\_last\_name VARCHAR2(50) NOT NULL,

Manager\_email VARCHAR2(50),

Manager\_Address VARCHAR2(50) NOT NULL,

Manager\_phone NUMBER(15),

Update\_dt DATE

);

CREATE TABLE Dim\_Employees (

Employee\_id NUMBER(10) PRIMARY KEY,

First\_name VARCHAR2(30) NOT NULL,

Last\_name VARCHAR2(50) NOT NULL,

Address VARCHAR2(50) NOT NULL,

Phone NUMBER(15),

Email VARCHAR2(50),

Update\_dt DATE

);

CREATE TABLE Dim\_Date (

Date\_id NUMBER(7) PRIMARY KEY,

Year NUMBER(5) NOT NULL,

Quarter NUMBER(1) NOT NULL,

Month NUMBER(2) NOT NULL,

Month\_name\_English VARCHAR(10) NOT NULL,

Week\_of\_year NUMBER(2) NOT NULL,

Day\_of\_month NUMBER(2) NOT NULL,

Day\_of\_week NUMBER(1) NOT NULL,

Day\_of\_week\_English VARCHAR2(10) NOT NULL,

Day\_of\_year NUMBER(3) NOT NULL,

Update\_dt DATE

);

CREATE TABLE **Fact\_Sales** (

Customer\_id\_sales NUMBER(16),

Customer\_id\_payment NUMBER(16) NOT NULL,

Employee\_id NUMBER(10) NOT NULL,

Staff\_id NUMBER(10) NOT NULL,

Sales\_date\_id NUMBER(7) NOT NULL,

Product\_id NUMBER(8) NOT NULL,

Store\_id NUMBER(7) NOT NULL,

Amount NUMBER(10,2),

Update\_dt DATE,

CONSTRAINT customer\_sales\_fk FOREIGN KEY (Customer\_id\_sales) REFERENCES Dim\_Customers(Customer\_id),

CONSTRAINT customer\_payment\_fk FOREIGN KEY (Customer\_id\_payment) REFERENCES Dim\_Customers(Customer\_id),

CONSTRAINT employee\_fk FOREIGN KEY (Employee\_id) REFERENCES Dim\_Employees(Employee\_id),

CONSTRAINT staff\_fk FOREIGN KEY (staff\_id) REFERENCES Dim\_Employees(Employee\_id),

CONSTRAINT sales\_date\_fk FOREIGN KEY (sales\_date\_id) REFERENCES Dim\_Date(Date\_id),

CONSTRAINT product\_fk FOREIGN KEY (product\_id) REFERENCES Dim\_Products(Product\_id),

CONSTRAINT store\_id FOREIGN KEY (store\_id) REFERENCES Dim\_Stores(Store\_id)

);

# Snowflake sheme

In the snowflake model some dimensions are divided into several tables to be closer to the 3rd NF.

**Products** dimension is divided to the Products itself and its Categories.

**Customers** dimension is divided to Customer information and their proper Location (City, Country, Region, EconRegion) to have the opportunity to make filters and grouping on the Sales by their place of residence.

**Date** dimension is left as it was.

**Employees** dimension is a little be modified, too. We are not interested in the Location information of the Stuff during Sales analysis. Moreover this information is not mandatory and the attribute is big enough. That is why address is taken in a separate dimension (1:1 with the Employees).

**Stores** dimension has a proper detailing by Address (has telephone, address in the City, postal code), City, Country, Region, and EconRegion.

As it was in the star scheme each snowflake dimension has served date-update attribute.

