|  |
| --- |
| EPAM Systems, RD Dep. |
| MTN.BI.08 Fact Table Techniques Labwork |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
|  | Report | Vitaliya Adamchuk | 27-NOV-2017 |  |  |

Contents

[1. Data 3](#_Toc498992608)

[1.1 Test\_ce\_countries 3](#_Toc498992609)

[1.2 Test\_ce\_countries 3](#_Toc498992610)

[1.3 Employees 4](#_Toc498992611)

[1.4 Category 4](#_Toc498992612)

[1.5 Products 4](#_Toc498992613)

[1.6 Products 5](#_Toc498992614)

[1.7 Date 5](#_Toc498992615)

[1.8 Result 6](#_Toc498992616)

[2. Advanced Grouping tasks – Reports 7](#_Toc498992617)

[2.1. Create Test AdHoc SQL - Daily Report (CUBE) 7](#_Toc498992618)

[2.2. Create Test AdHoc SQL – ROLLUP by Time 8](#_Toc498992619)

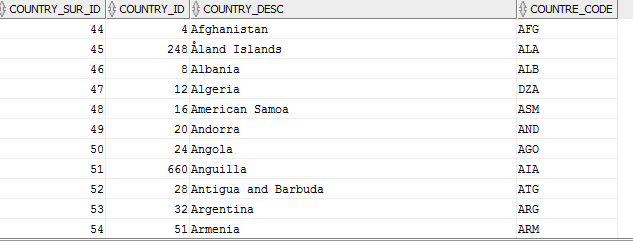
# Data

The Main Task is to generate test data on Storage layers objects that was created on task 01.

Task Results:

* Scripts put on Git
* Storage Layers Objects test data select screenshots

## Test\_ce\_countries



1. Таблица сформирована на основе данных из таблицы из предыдущей лабы.

Чтобы получить к ней доступ были выданы соответсвующие гранты, а также создан синоним.

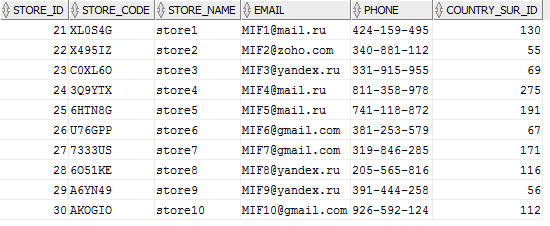
exec system.grants\_mgmt.grant\_blat ('SELECT',‘BL\_3NF','countries','test\_DWH');

CREATE PUBLIC SYNONYM countries FOR BL\_3NF.countries;

1. Country\_sur\_id сформированы на основе seq (код в папке), для создания последовательности также были выданы гранты.

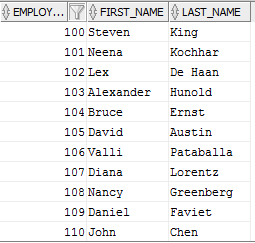
exec system.grants\_mgmt.grant\_blat ('CREATE ANY SEQUENCE', 'test\_DWH');

## Test\_ce\_countries



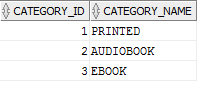
1. Store\_id создавался с помощью последвательности.
2. Store\_code: *dbms\_random.string ( 'X', 6 ) AS store\_code*, - генерировалась рандомная строка длиной в 6 символов, состоящая из цифр и букв.
3. Почтовый домен генерировался с помощью рандома и case.
4. Номер телефона также генерировался.
5. Country\_id генерировался на основе test\_ce\_countries таблиицы.

## Employees



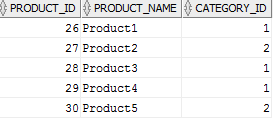
Генерировалась из таблицы hr.employees (планировалось ее дальнейшее использование, но не понадобилась)

## Category



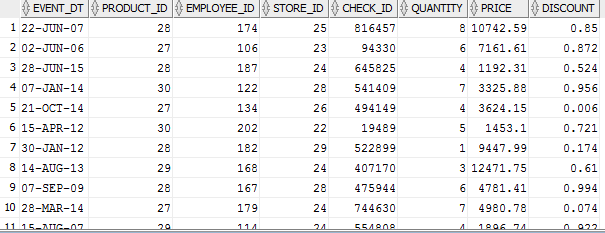
Заполнялась вручную.

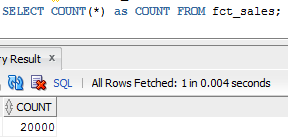
## Products



1. Product\_id – sequence.
2. Category\_id генерировалась на основе таблицы категорий.

## Products





1. Было сгенировано 20,000 пробных записей.
2. Все ключи генерились на основе существующих таблиц (через select)

Пример:

ROUND(dbms\_random.value(

(SELECT MIN(employee\_id) FROM employees

),

(SELECT MAX(employee\_id) FROM employees

))) AS employee\_id,

1. Цена генерировалась с помощью нормального распределения.

ABS(ROUND((DBMS\_RANDOM.NORMAL()\*10000),2)) AS price

Предусмотрен модуль, так как возможны отрицательные значения.

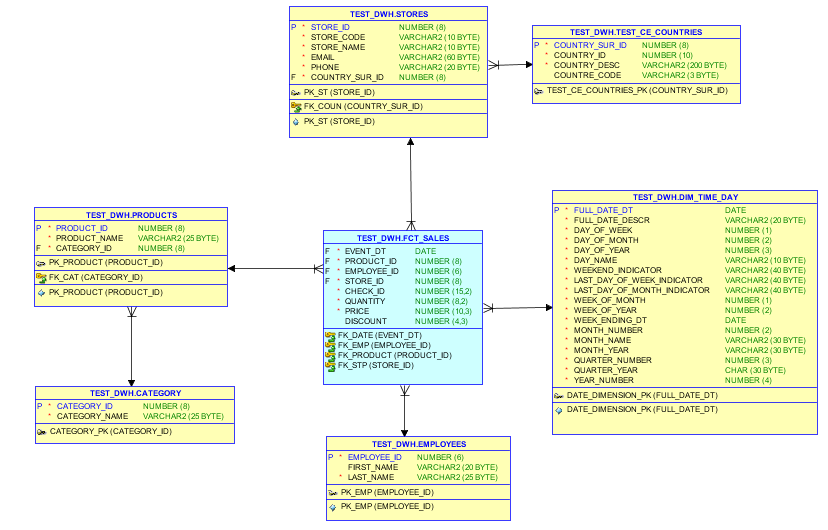
1. Количество и скидка генерировались рандомно, но я еще попытаюсь сделать разброс более продуманным (через case)

## Date

Генерировалась с помощью процедуры.

## Result

Были добавлены все необходимые ограничения и сформирована схема.



# Advanced Grouping tasks – Reports

## Create Test AdHoc SQL - Daily Report (CUBE)

The Main Task is to create adhoc SQL script, which will calculate Daily Reports (According report layouts on Module 7).

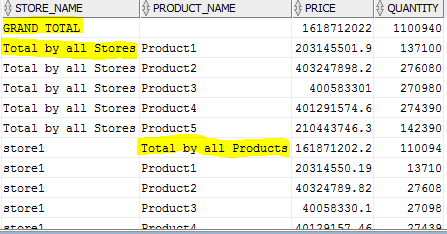
Requirements:

* Use CUBE Extension, ROLLUP, Grouping(), Grouping\_ID functions

Task Results:

* Create document that will compare script results and report layouts
* Put scripts on Git

Был проведен анализ в разрезе проданных продуктов и магазинов.



## Create Test AdHoc SQL – ROLLUP by Time

The Main Task is to create adhoc SQL script, which will calculate Time Based Reports

Calculate measurements by next levels:

* DAY
* MONTH
* QUARTER
* YEAR

Requirements:

* USE: ROLLUP
* USE: Grouping() function
* USE: Grouping\_ID function

Task Results:

* Create document that will show script results
* Put scripts on Git

Был проведен анализ в разрезе прибыли в разрезе года, квартала, месяца, дня и суммарно за все время работы.

