Report

Laboratory Work 2

Dmitry Ladutsko

August 09, 2022

# 1. Prerequisites

## 1.1. Passwords Index

|  |  |  |
| --- | --- | --- |
| Password Group | Login Name | Password |
| Operation System | root | “rootadmin” |
|  | oracle | “oracleadmin” |
|  |  |  |
| Oracle System | sys | “sysadmin” |
|  | system | “sysadmin” |
|  |  |  |
| Oracle Users | All DB users | “%PWD%” |
|  |  |  |
|  |  |  |

## 1.2. Folder Paths Index

|  |  |  |
| --- | --- | --- |
| Path Group | Path Description | Path |
| Operation System | Oracle RDBMS – BIN | /oracle/app/oracle |
|  | Oracle Inventory | /oracle/app/oraInventory |
|  | Oracle Database Storage | /oracle/oradata |
|  | Oracle Install Directory | /oracle/install |
| Oracle | ORACLE\_BASE | /oracle/app/oracle |
|  | ORACLE\_HOME | $ORACLE\_BASE/product/11.2 |
|  |  |  |
| FTP | ftp Incoming Folder | /ftp/incoming |
|  |  |  |
|  |  |  |

2. Business analyses tasks – Reports

2.1. Task 01: CREATE Daily Reports Layouts

**The Main Task** is to create Reports Layouts according your Business Solution Proposal, which was developed on Exit Task Module 6.Introduction to DWH.

**Task Results:**

Create report layouts:

* Create excel report layouts
* Put report layouts on Git – Folder BI Tasks – Product Name (author)  - Repots

Table

Description automatically generated

Picture 1 - Daily Reports Layout

2.2. Task 02: CREATE Monthly Reports Layouts

**The Main Task** is to create Reports Layouts according your Business Solution Proposal, which was developed on Exit Task Module 6.Introduction to DWH.

**Task Results:**

Create report layouts:

* Create excel report layouts
* Put report layouts on Git – Folder BI Tasks – Product Name (author)  - Repots

Table

Description automatically generated

Picture 2 - Monthly Reports Layout

3. Advanced Grouping tasks – Reports

3.1. Task 03: CREATE Test AdHoc SQL - Daily Reports (CUBE)

**The Main Task** is to create adhoc SQL script, which will calculate Daily Reports (According report layouts on task 01).

**Requirements:**

* USE: CUBE Extension

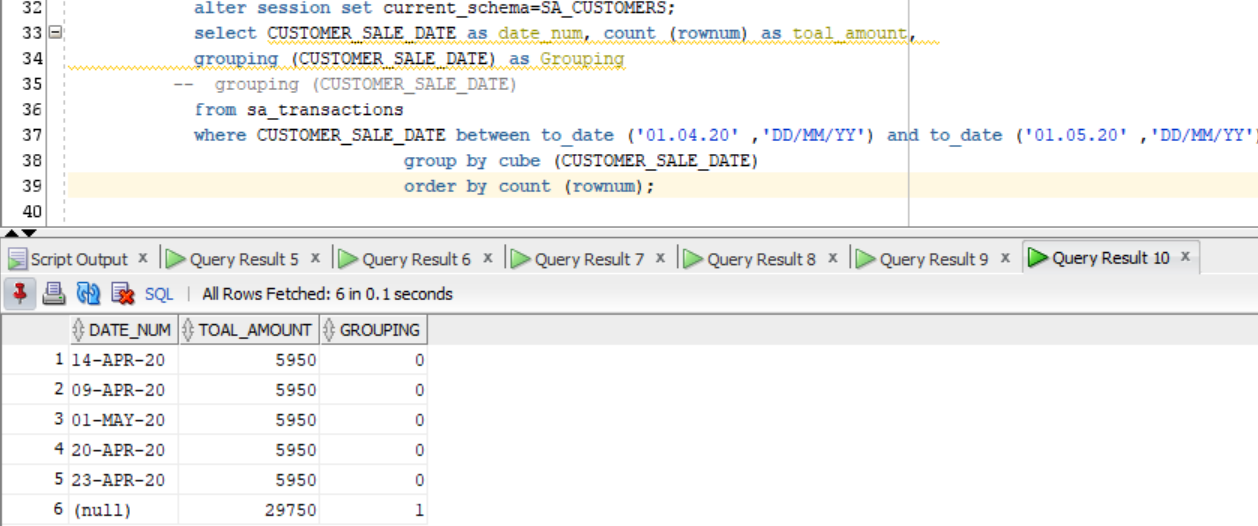
Graphical user interface, application

Description automatically generated

Picture 3 - Group by cube

**CUBE** extension will generate subtotals for all combinations of the dimensions specified. If "n" is the number of columns listed in the CUBE, there will be 2n subtotal combinations.

* USE: Grouping() function



Picture 4 – Grouping

It can be quite easy to visually identify subtotals generated by rollups and cubes, but to do it programatically you really need something more accurate than the presence of null values in the grouping columns. This is where the **GROUPING** function comes in. It accepts a single column as a parameter and returns "1" if the column contains a null value generated as part of a subtotal by a ROLLUP or CUBE operation or "0" for any other value, including stored null values.

* USE: Grouping\_ID function

Graphical user interface, application

Description automatically generated

Picture 5 - Grouping\_id

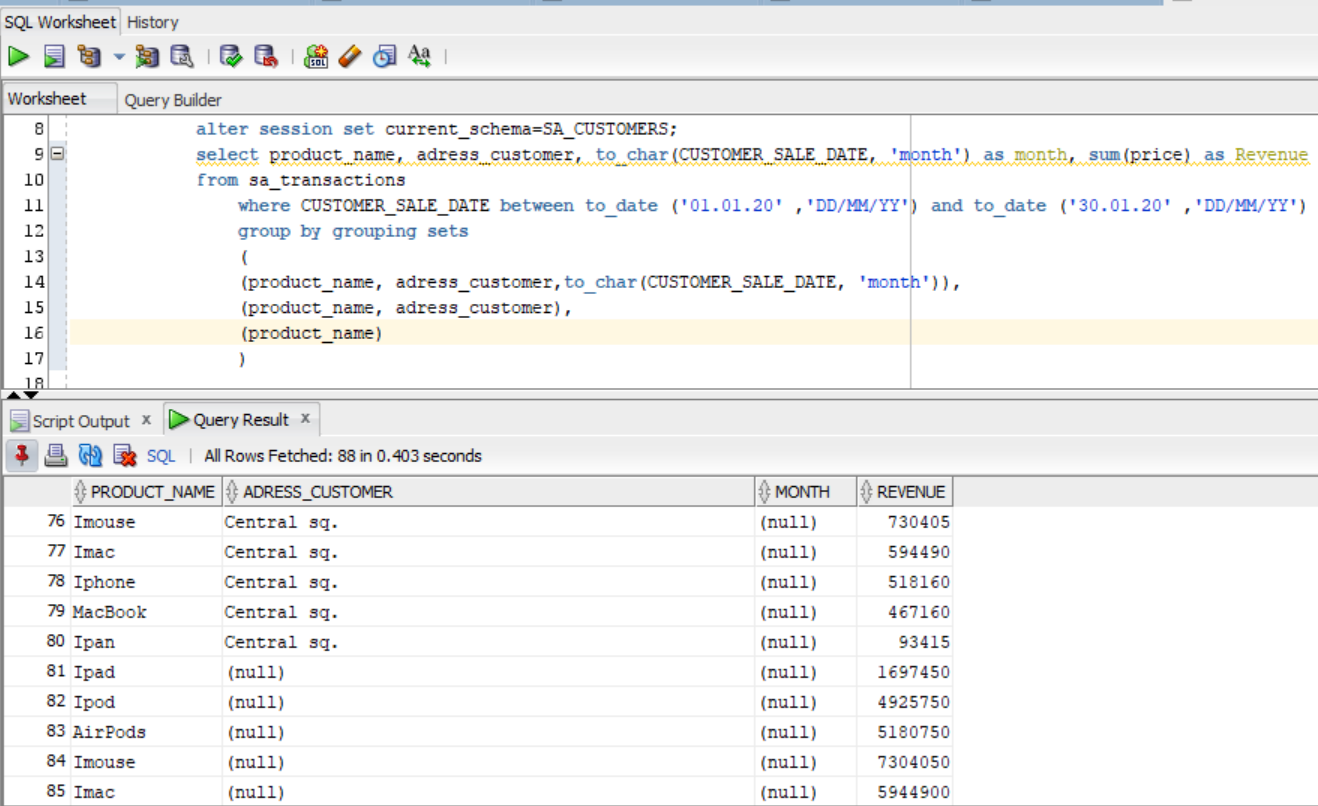
The **GROUPING\_ID**function provides an alternate and more compact way to identify subtotal rows. Passing the dimension columns as arguments, it returns a number indicating the GROUP BY level.

3.2. Task 04: CREATE Test AdHoc SQL - Monthly Reports (ROLLUP & GROUPING SETS)

**The Main Task** is to create adhoc SQL script, which will calculate Monthly Reports (According report layouts on task 01).

**Requirements:**

* USE: ROLLUP or GROUPING SETS Extension



Picture 6 - Group by grouping sets

**GROUPING SETS**  calculating all possible subtotals in a cube, especially those with many dimensions, can be quite an intensive process. If you don't need all the subtotals, this can represent a considerable amount of wasted effort. The following cube with three dimensions gives 8 levels of subtotals (GROUPING\_ID: 0-7)

Text

Description automatically generated with medium confidence

Picture 7 - Group by rollup

*In addition to the regular aggregation* results we expect from the **GROUP BY**clause, the **ROLLUP** extension produces group subtotals from right to left and a grand total. If "n" is the number of columns listed in the **ROLLUP**, there will be n+1 levels of subtotals.

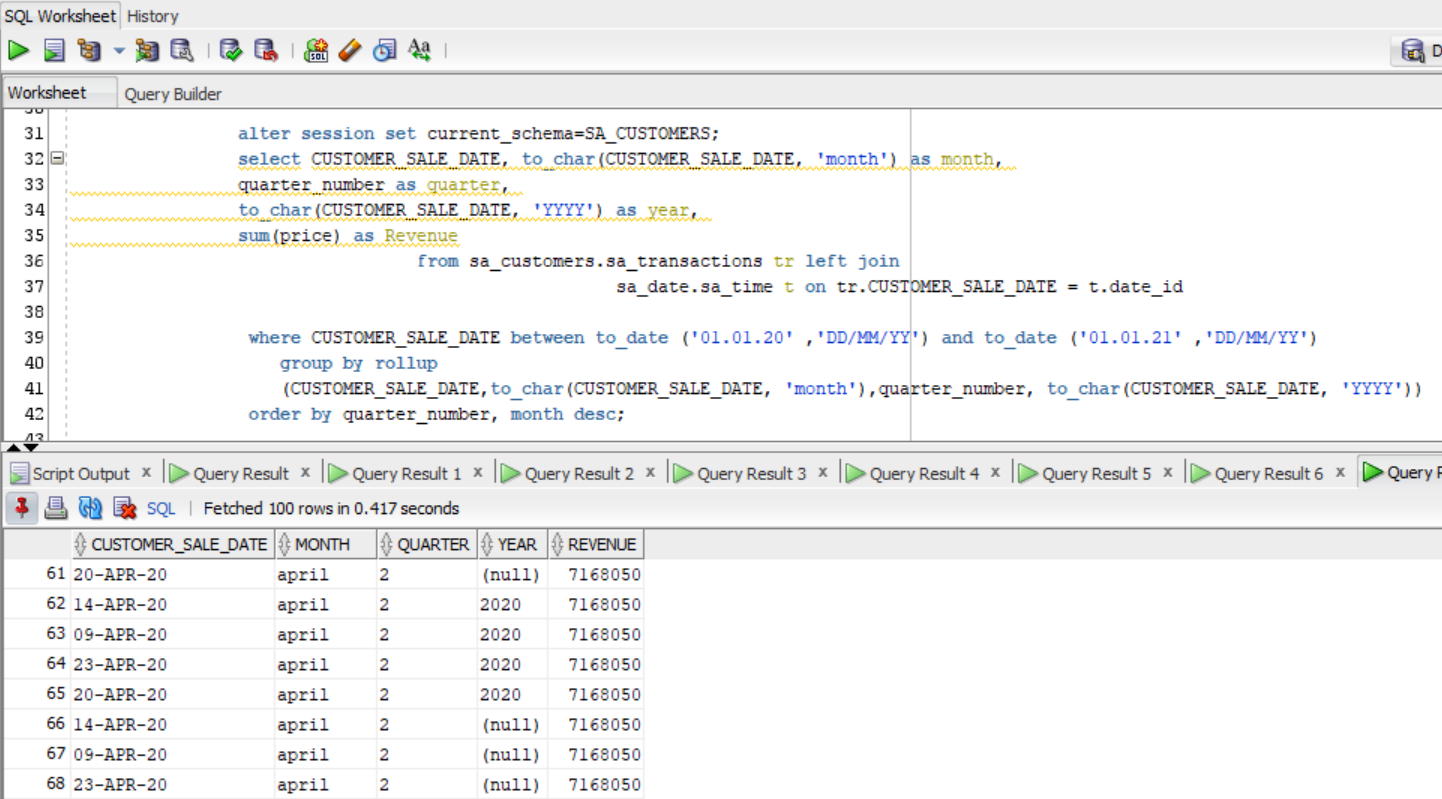
3.3. Task 05: CREATE Test AdHoc SQL – ROLLUP by Time

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

Calculate measurements by next levels:

* DAY
* MONTH
* QUARTER
* YEAR

**Sum Revenue**



Picture 8 - Group by rollup day - month - quarter – year

**Average Revenue**

Graphical user interface, text, application

Description automatically generated

Picture 9 - Group by rollup day - month - quarter – year

**Laboratory work summary:**

**At this lab** we have learned how we can use different group statements to group data in useful Reports. We used following group statements:

* Group by
* Group by cube
* Grouping
* Grouping\_id
* Group by rollup
* Group by grouping sets

**Now we have** much more understanding about grouping data in decision – making reports.