Report

Laboratory Work 11

Dmitry Ladutsko

August 20, 2022

1. Prerequisites Task Information

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. ETL Advanced Refresh Scenarios – Refactoring Load to SAL

Task 01 is common for LabWork 10 (Task 02), 11(Task 01).

2.1. Task 01: Loading to SAL Layer Data

The Main Task is to load dimension to SAL layer

Required points:

- Create new package for Load FCT_* and DIM_* to SAL Layer
- Load Dimension
- Load SCD Dimension
- Load FCT_*

3. Business Task – Performance of STAR Scheme

3.1. Task 02: Prepare Report Layout

The Main Task is to create Ad Hoc SQL for Report Layout Monthly that was developed on LabWork 2 (Use STAR schema objects for source of Data).

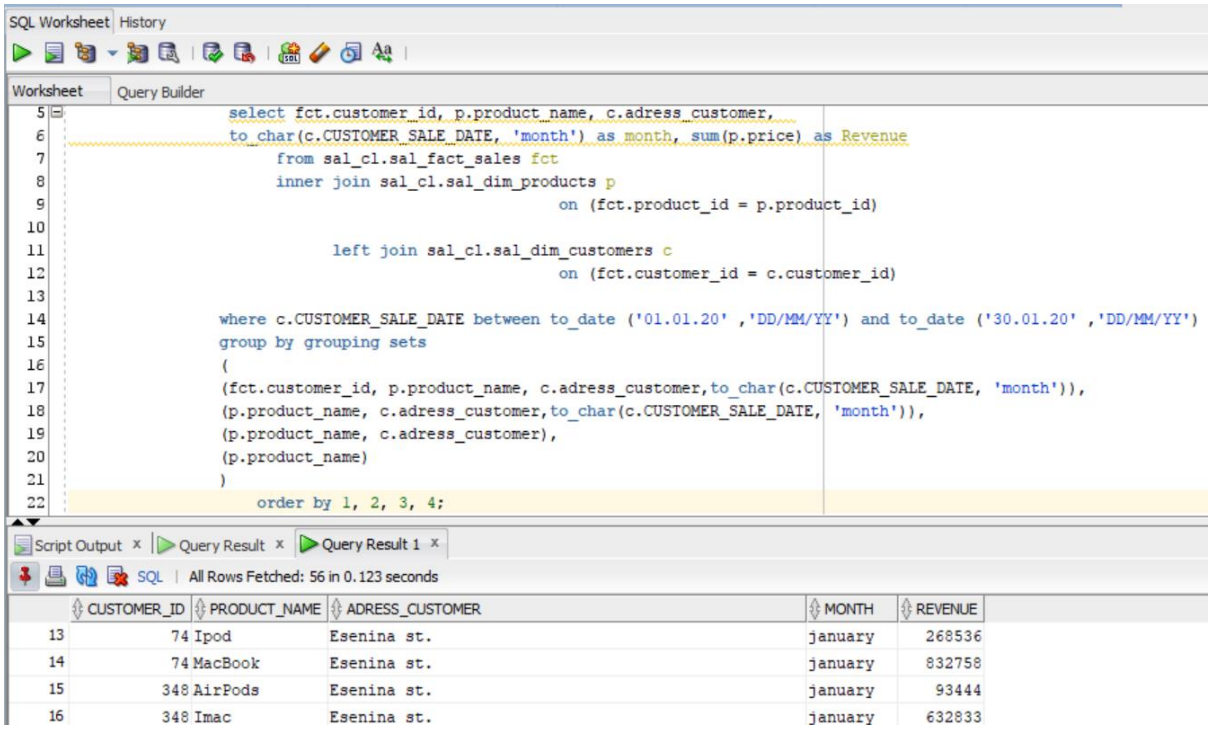
Note. I took **Monthly report group by grouping sets** from Laboratory Work 2, to create a **view as select.** I used **star schema objects** instead of using **transaction table** (as in lab work 2)

```
14  alter session set current_schema = sal_cl;
15
16
17              alter session set current_schema = sal_cl;
18              create or replace view v_prod_revenue as
```

Script Output | Query Result | Query Result 1

Task completed in 0.12 seconds

```
View V_PROD_REVENUE created.
```

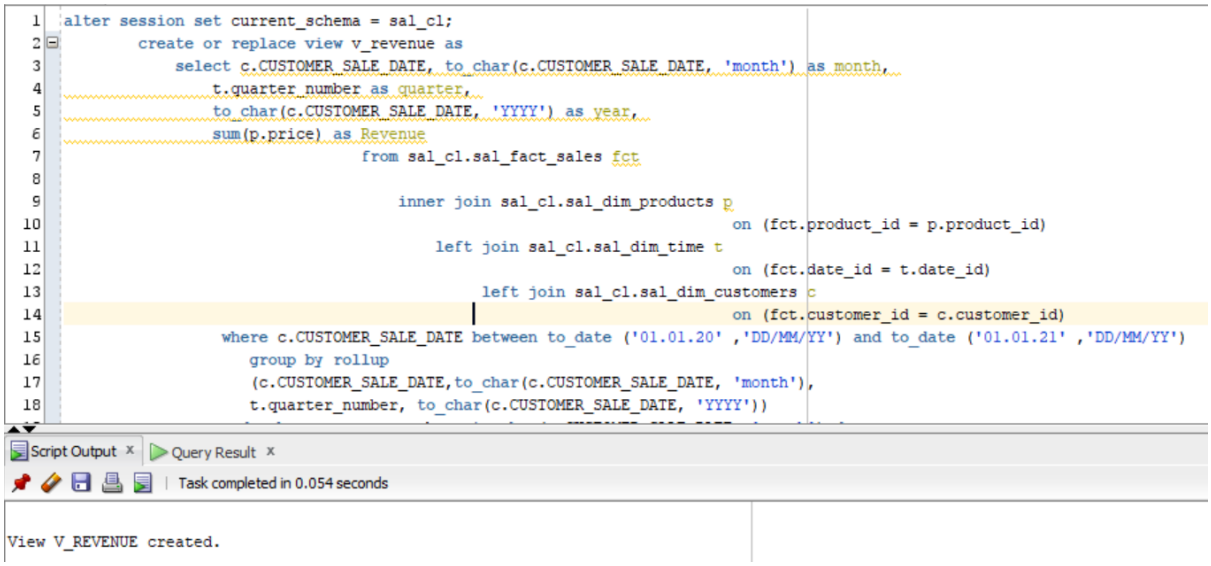


```
select fct.customer_id, p.product_name, c.adress_customer,
to_char(c.CUSTOMER_SALE_DATE, 'month') as month, sum(p.price) as Revenue
    from sal_cl.sal_fact_sales fct
    inner join sal_cl.sal_dim_products p
                                    on (fct.product_id = p.product_id)

        left join sal_cl.sal_dim_customers c
                                    on (fct.customer_id = c.customer_id)

where c.CUSTOMER_SALE_DATE between to_date ('01.01.20' ,'DD/MM/YY') and to_date ('30.01.20' ,'DD/MM/YY')
group by grouping sets
(
(fct.customer_id, p.product_name, c.adress_customer,to_char(c.CUSTOMER_SALE_DATE, 'month')),
(p.product_name, c.adress_customer,to_char(c.CUSTOMER_SALE_DATE, 'month')),
(p.product_name, c.adress_customer),
(p.product_name)
)
    order by 1, 2, 3, 4;
```

	CUSTOMER_ID	PRODUCT_NAME	ADRESS_CUSTOMER	MONTH	REVENUE
13	74	Ipod	Esenina st.	january	268536
14	74	MacBook	Esenina st.	january	832758
15	348	AirPods	Esenina st.	january	93444
16	348	Imac	Esenina st.	january	632833



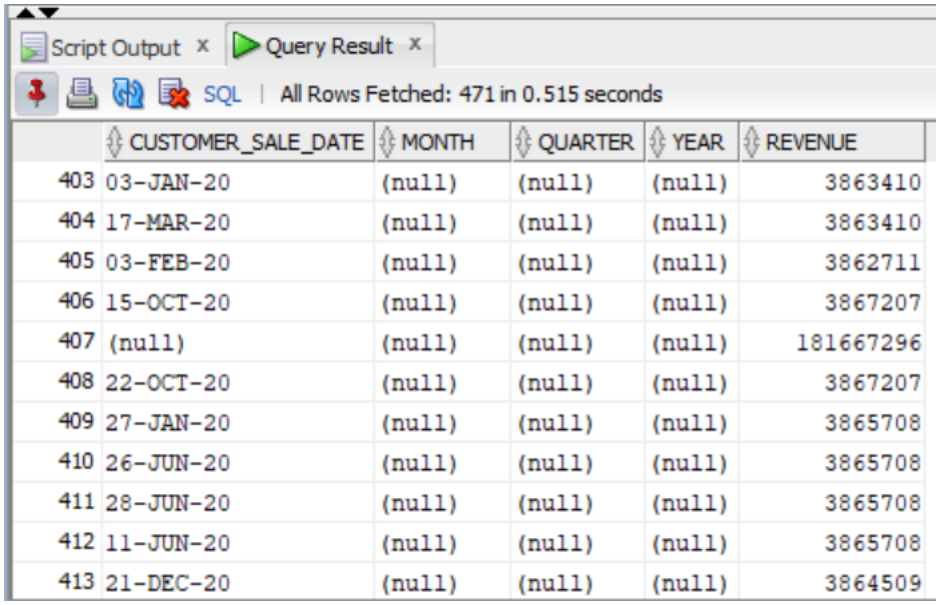
```
alter session set current_schema = sal_cl;
        create or replace view v_revenue as
            select c.CUSTOMER_SALE_DATE, to_char(c.CUSTOMER_SALE_DATE, 'month') as month,
                t.quarter_number as quarter,
                to_char(c.CUSTOMER_SALE_DATE, 'YYYY') as year,
                sum(p.price) as Revenue
                                from sal_cl.sal_fact_sales fct

                                    inner join sal_cl.sal_dim_products p
                                                                    on (fct.product_id = p.product_id)
                                        left join sal_cl.sal_dim_time t
                                                                    on (fct.date_id = t.date_id)
                                            left join sal_cl.sal_dim_customers c
                                                                    on (fct.customer_id = c.customer_id)
            where c.CUSTOMER_SALE_DATE between to_date ('01.01.20' ,'DD/MM/YY') and to_date ('01.01.21' ,'DD/MM/YY')
               group by rollup
               (c.CUSTOMER_SALE_DATE,to_char(c.CUSTOMER_SALE_DATE, 'month'),
               t.quarter_number, to_char(c.CUSTOMER_SALE_DATE, 'YYYY'))
```

Task completed in 0.054 seconds

```
View V_REVENUE created.
```

Note. As you can see I have also created a view to see the **summarized revenue** grouped by **day – month – quarter – year** hierarchy using **Star Schema objects**



Script Output | Query Result

SQL | All Rows Fetched: 471 in 0.515 seconds

	CUSTOMER_SALE_DATE	MONTH	QUARTER	YEAR	REVENUE
403	03-JAN-20	(null)	(null)	(null)	3863410
404	17-MAR-20	(null)	(null)	(null)	3863410
405	03-FEB-20	(null)	(null)	(null)	3862711
406	15-OCT-20	(null)	(null)	(null)	3867207
407	(null)	(null)	(null)	(null)	181667296
408	22-OCT-20	(null)	(null)	(null)	3867207
409	27-JAN-20	(null)	(null)	(null)	3865708
410	26-JUN-20	(null)	(null)	(null)	3865708
411	28-JUN-20	(null)	(null)	(null)	3865708
412	11-JUN-20	(null)	(null)	(null)	3865708
413	21-DEC-20	(null)	(null)	(null)	3864509

3.2. Task 03: Compare Report Layout Performance

The Main Task is to create summarize table with comparison Performance of next Report Layout:

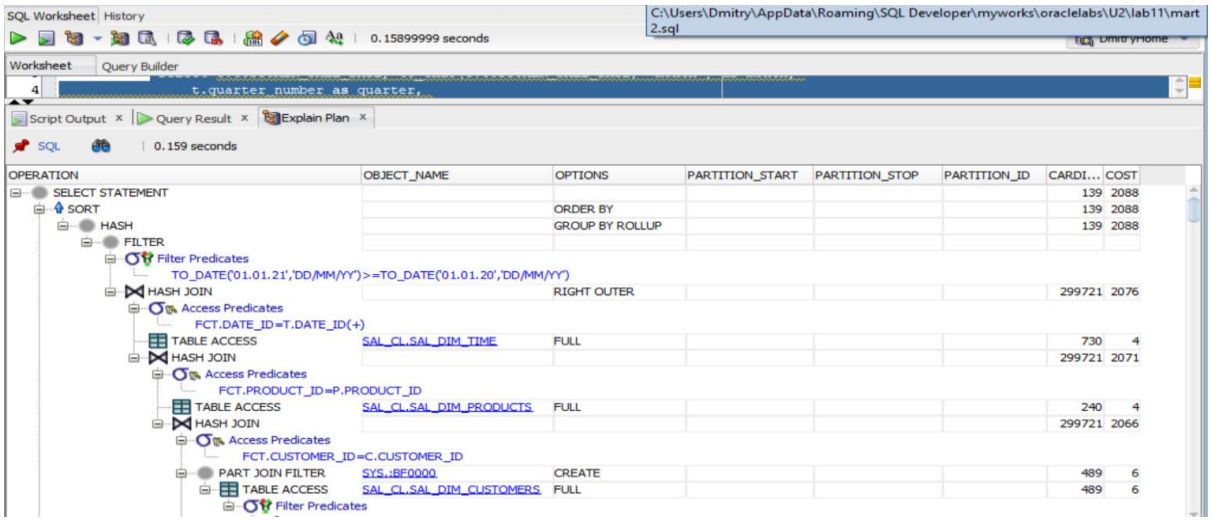
- Advancing Grouping (GROUP BY GROUPING SETs – LabWork 02)
- Model Clause (LabWork 05)
- Star Schema (LabWork 11)

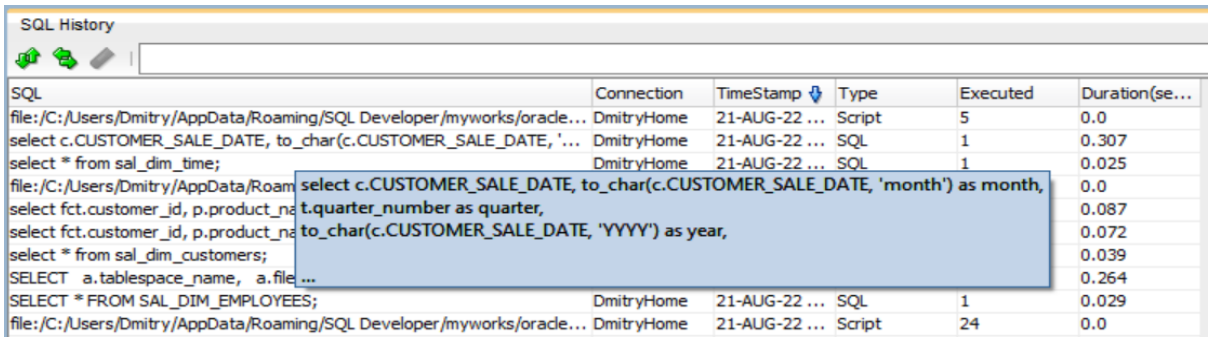
Task Results:

Create required objects:

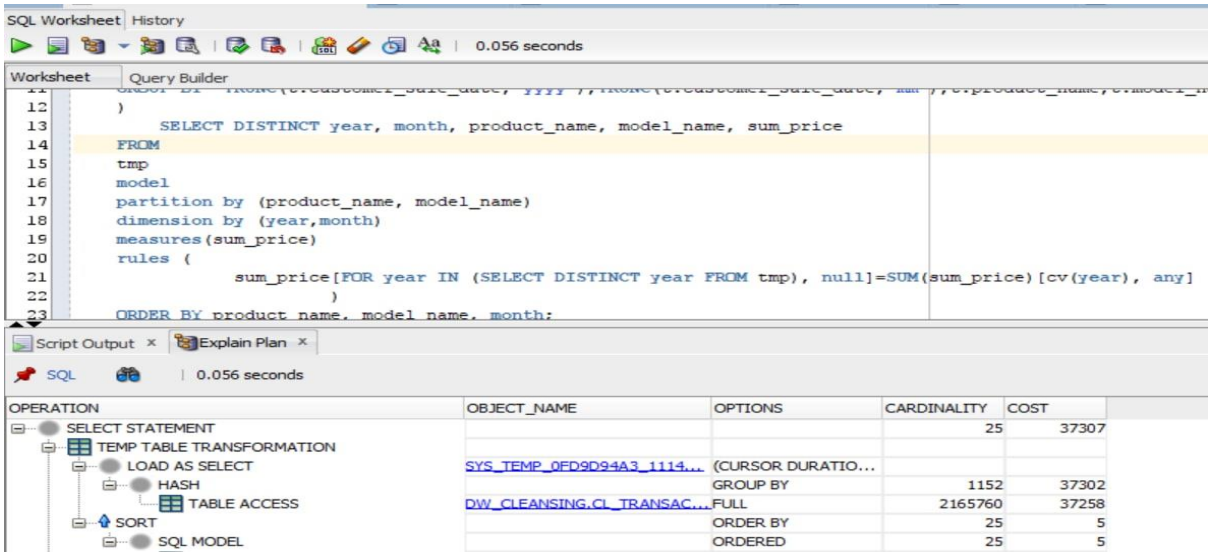
- Prepare Document with Summarize table

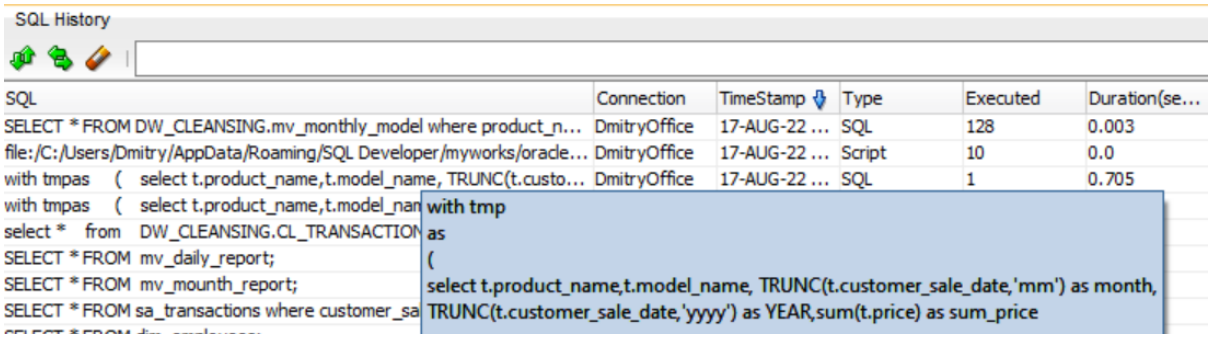
Star Schema



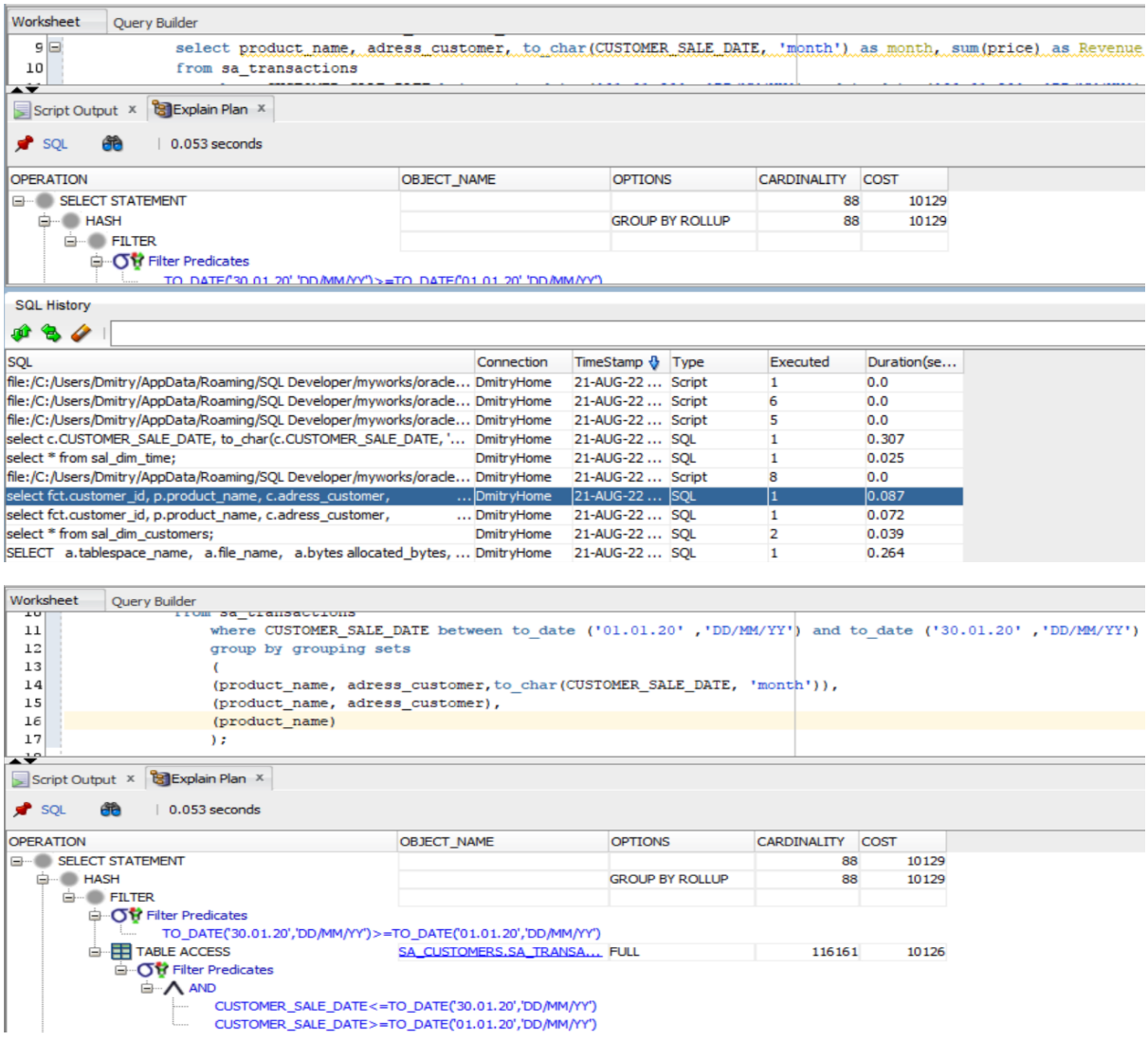


Model Clause





Advancing Grouping



№	Source Type	Explain Plan - Statistics	Time, Sec.
1	Advancing Grouping (lab 2)	Cost - 10129	0.087
2	Model Clause (lab 5)	Cost - 37307	0.705
3	Star Schema (lab 11)	Cost - 2088	0.307

Laboratory Work Summary

At this laboratory work we created views with different group statements as select from our created star schema. We recalled the explain plan statistic to show how oracle's optimizer estimates performance of different ways of report outputs.