

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_*

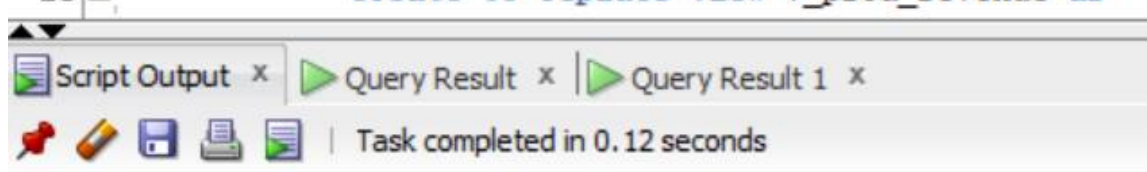
3. Business Task – Performance of STAR Scheme

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
```



The screenshot shows the SQL Developer interface. The script window contains the following SQL code:
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
Below the script window, there are tabs for 'Script Output', 'Query Result', and 'Query Result 1'. The status bar at the bottom indicates 'Task completed in 0.12 seconds'.

View V_PROD_REVENUE created.

SQL Worksheet | History

Worksheet | Query Builder

```

5 select fct.customer_id, p.product_name, c.address_customer,
6 to_char(c.CUSTOMER_SALE_DATE, 'month') as month, sum(p.price) as Revenue
7 from sal_cl.sal_fact_sales fct
8 inner join sal_cl.sal_dim_products p
9 on (fct.product_id = p.product_id)
10
11 left join sal_cl.sal_dim_customers c
12 on (fct.customer_id = c.customer_id)
13
14 where c.CUSTOMER_SALE_DATE between to_date ('01.01.20', 'DD/MM/YY') and to_date ('30.01.20', 'DD/MM/YY')
15 group by grouping sets
16 (
17 (fct.customer_id, p.product_name, c.address_customer, to_char(c.CUSTOMER_SALE_DATE, 'month')),
18 (p.product_name, c.address_customer, to_char(c.CUSTOMER_SALE_DATE, 'month')),
19 (p.product_name, c.address_customer),
20 (p.product_name)
21 )
22 order by 1, 2, 3, 4;

```

Script Output x | Query Result x | Query Result 1 x

SQL | All Rows Fetched: 56 in 0.123 seconds

CUSTOMER_ID	PRODUCT_NAME	ADDRESS_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

```

1 alter session set current_schema = sal_cl;
2 create or replace view v_revenue as
3 select c.CUSTOMER_SALE_DATE, to_char(c.CUSTOMER_SALE_DATE, 'month') as month,
4 t.quarter_number as quarter,
5 to_char(c.CUSTOMER_SALE_DATE, 'YYYY') as year,
6 sum(p.price) as Revenue
7 from sal_cl.sal_fact_sales fct
8 inner join sal_cl.sal_dim_products p
9 on (fct.product_id = p.product_id)
10 left join sal_cl.sal_dim_time t
11 on (fct.date_id = t.date_id)
12 left join sal_cl.sal_dim_customers c
13 on (fct.customer_id = c.customer_id)
14 where c.CUSTOMER_SALE_DATE between to_date ('01.01.20', 'DD/MM/YY') and to_date ('01.01.21', 'DD/MM/YY')
15 group by rollup
16 (c.CUSTOMER_SALE_DATE, to_char(c.CUSTOMER_SALE_DATE, 'month'),
17 t.quarter_number, to_char(c.CUSTOMER_SALE_DATE, 'YYYY'))
18

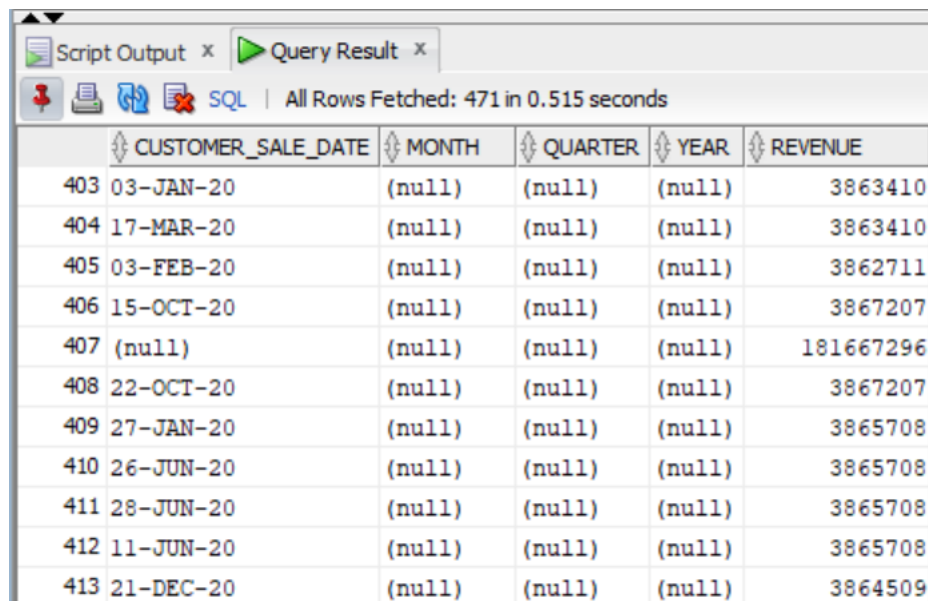
```

Script Output x | Query Result x

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



	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:

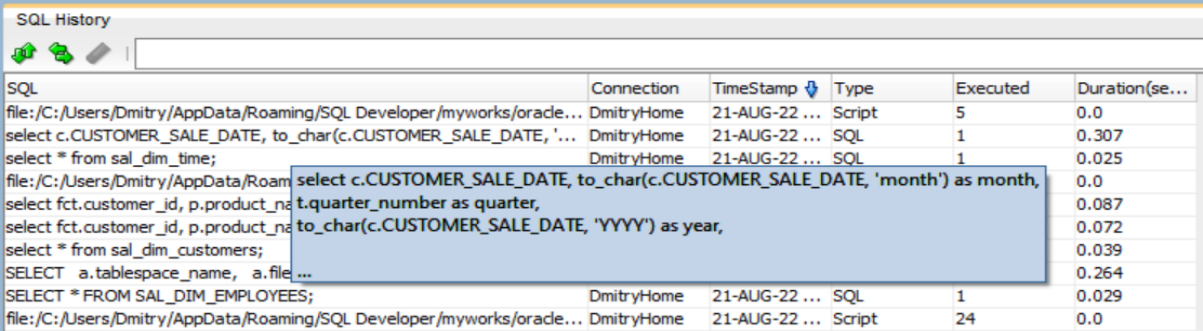
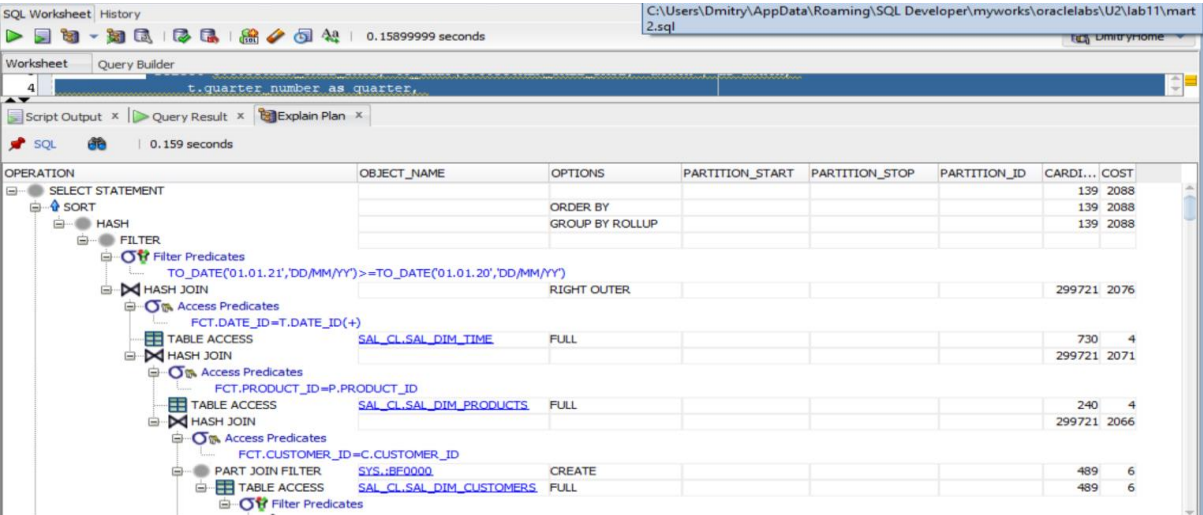
- Advancing Grouping (GROUP BY GROUPING SETs – LabWork 02)
- Model Clause (LabWork 05)
- Star Schema (LabWork 11)

Task Results:

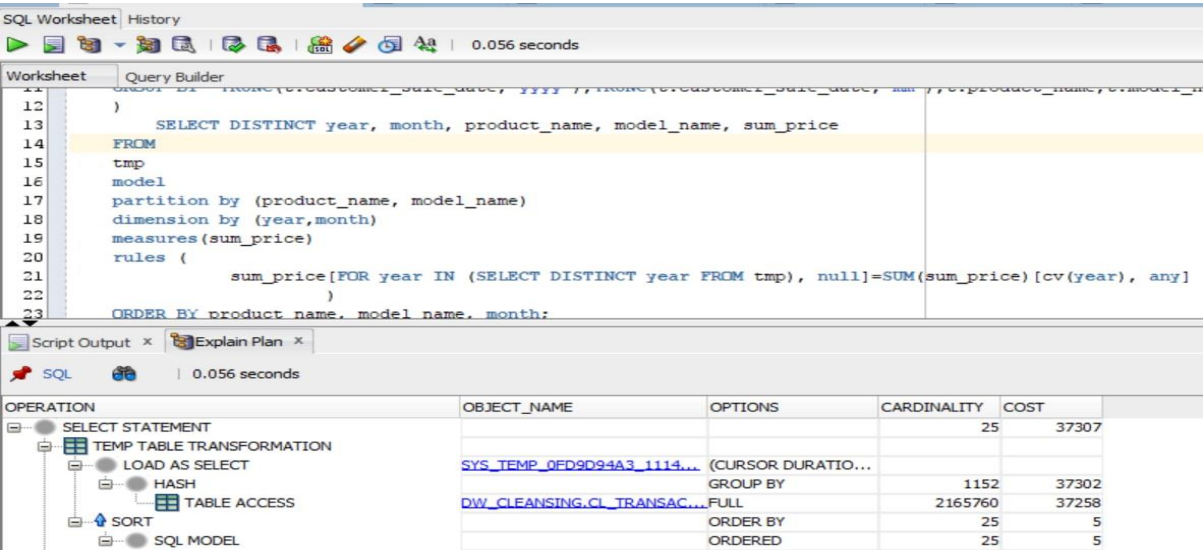
Create required objects:

- Prepare Document with Summarize table

Star Schema



Model Clause



SQL History					
SQL	Connection	TimeStamp	Type	Executed	Duration(se...
SELECT * FROM DW_CLEANSING.mv_monthly_model where product_n...	DmitryOffice	17-AUG-22 ...	SQL	128	0.003
file:/C:/Users/Dmitry/AppData/Roaming/SQL Developer/myworks/orade...	DmitryOffice	17-AUG-22 ...	Script	10	0.0
with tmpas (select t.product_name,t.model_name, TRUNC(t.custo...	DmitryOffice	17-AUG-22 ...	SQL	1	0.705
with tmpas (select t.product_name,t.model_name, TRUNC(t.custo...					
select * from DW_CLEANSING.CL_TRANSACTION as					
SELECT * FROM mv_daily_report;					
SELECT * FROM mv_mounth_report;					
SELECT * FROM sa_transactions where customer_sa					
SELECT * FROM mv_monthly_model					

Advanced Grouping

Worksheet

Query Builder

9

10

select product_name, address_customer, to_char(CUSTOMER SALE DATE, 'month') as month, sum(price) as Revenue

from sa_transactions

Script Output

Explain Plan

SQL

0.053 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			88	10129
HASH		GROUP BY ROLLUP	88	10129
FILTER				
Filter Predicates				
TO_DATE('30.01.20','DD/MM/YY')>=TO_DATE('01.01.20','DD/MM/YY')				

SQL History

SQL

Connection

TimeStamp

Type

Executed

Duration(se...

file:/C:/Users/Dmitry/AppData/Roaming/SQL Developer/myworks/orade...

DmitryHome

21-AUG-22 ...

Script

1

0.0

file:/C:/Users/Dmitry/AppData/Roaming/SQL Developer/myworks/orade...

DmitryHome

21-AUG-22 ...

Script

6

0.0

file:/C:/Users/Dmitry/AppData/Roaming/SQL Developer/myworks/orade...

DmitryHome

21-AUG-22 ...

Script

5

0.0

select c.CUSTOMER_SALE_DATE, to_char(c.CUSTOMER_SALE_DATE, '...

DmitryHome

21-AUG-22 ...

SQL

1

0.307

select * from sal_dim_time;

DmitryHome

21-AUG-22 ...

SQL

1

0.025

file:/C:/Users/Dmitry/AppData/Roaming/SQL Developer/myworks/orade...

DmitryHome

21-AUG-22 ...

Script

8

0.0

select fct.customer_id, p.product_name, c.adress_customer, ...

DmitryHome

21-AUG-22 ...

SQL

1

0.087

select fct.customer_id, p.product_name, c.adress_customer, ...

DmitryHome

21-AUG-22 ...

SQL

1

0.072

select * from sal_dim_customers;

DmitryHome

21-AUG-22 ...

SQL

2

0.039

SELECT a.tablespace_name, a.file_name, a.bytes allocated_bytes, ...

DmitryHome

21-AUG-22 ...

SQL

1

0.264

Worksheet

Query Builder

10

11

12

13

14

15

16

17

from sa_transactions

where 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

(

(product_name, address_customer,to_char(CUSTOMER_SALE_DATE, 'month')),

(product_name, address_customer),

(product_name)

);

Script Output

Explain Plan

SQL

0.053 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			88	10129
HASH		GROUP BY ROLLUP	88	10129
FILTER				
Filter Predicates				
TO_DATE('30.01.20','DD/MM/YY')>=TO_DATE('01.01.20','DD/MM/YY')				
TABLE ACCESS	SA_CUSTOMERS.SA_TRANSA...	FULL	116161	10126
Filter Predicates				
AND				
CUSTOMER_SALE_DATE<=TO_DATE('30.01.20','DD/MM/YY')				
CUSTOMER_SALE_DATE>=TO_DATE('01.01.20','DD/MM/YY')				

Nº	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.