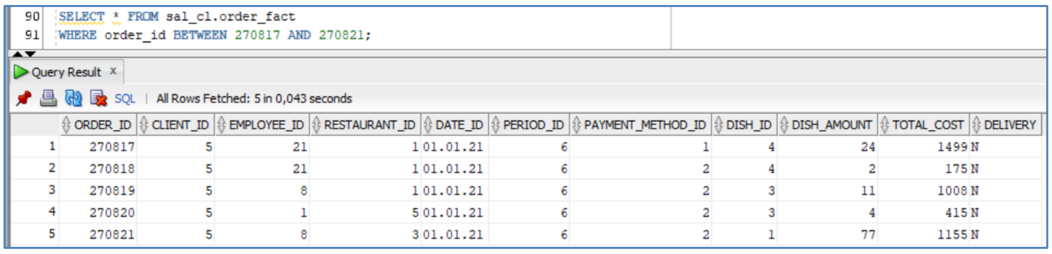
Lab report #11

Sadovskaya Veronika

GitHub: https://github.com/sdveronika/DataMola22

Task 1 – Loading to SAL Layer Data

Select from table sal_cl.order_fact where order_id between 270817 and 270821, because they are in partition quarter_1 we will change:



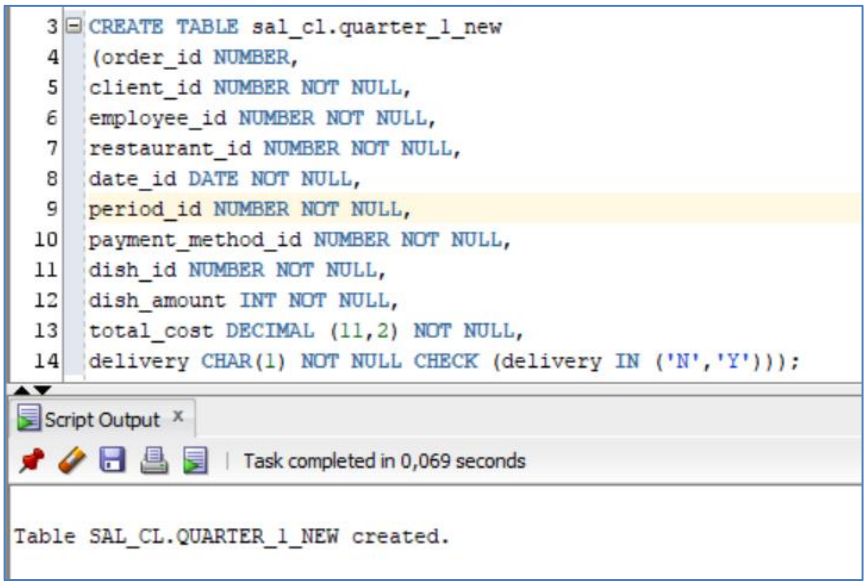
```
SELECT * FROM sal_cl.order_fact
WHERE order_id BETWEEN 270817 AND 270821;
```

Query Result

All Rows Fetched: 5 in 0,043 seconds

	ORDER_ID	CLIENT_ID	EMPLOYEE_ID	RESTAURANT_ID	DATE_ID	PERIOD_ID	PAYMENT_METHOD_ID	DISH_ID	DISH_AMOUNT	TOTAL_COST	DELIVERY
1	270817	5	21	1	01.01.21	6	1	4	24	1499	N
2	270818	5	21	1	01.01.21	6	2	4	2	175	N
3	270819	5	8	1	01.01.21	6	2	3	11	1008	N
4	270820	5	1	5	01.01.21	6	2	3	4	415	N
5	270821	5	8	3	01.01.21	6	2	1	77	1155	N

Create table that will store our entire partition, which we will change:



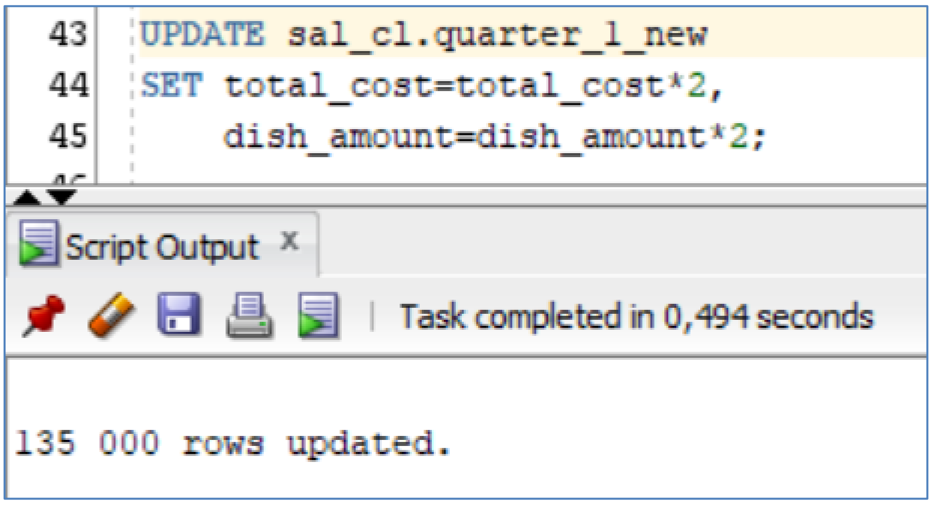
```
CREATE TABLE sal_cl.quarter_1_new
(order_id NUMBER,
client_id NUMBER NOT NULL,
employee_id NUMBER NOT NULL,
restaurant_id NUMBER NOT NULL,
date_id DATE NOT NULL,
period_id NUMBER NOT NULL,
payment_method_id NUMBER NOT NULL,
dish_id NUMBER NOT NULL,
dish_amount INT NOT NULL,
total_cost DECIMAL (11,2) NOT NULL,
delivery CHAR(1) NOT NULL CHECK (delivery IN ('N','Y')));
```

Script Output

Task completed in 0,069 seconds

```
Table SAL_CL.QUARTER_1_NEW created.
```

Let's update the data in the table, so that later we can see the result of replacing the partition:



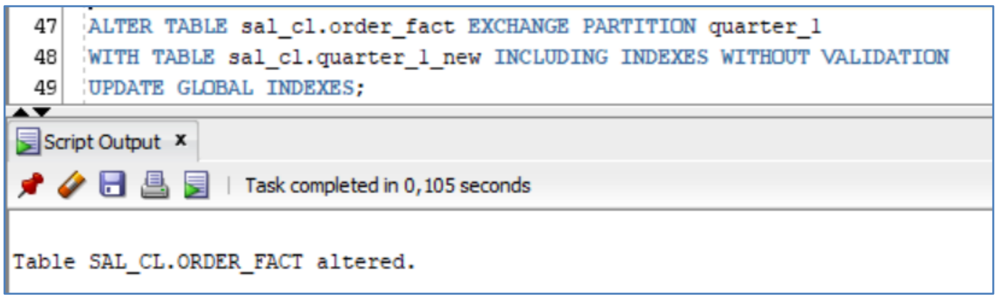
```
UPDATE sal_cl.quarter_1_new
SET total_cost=total_cost*2,
    dish_amount=dish_amount*2;
```

Script Output

Task completed in 0,494 seconds

```
135 000 rows updated.
```

Making a partition replacement in the original table:



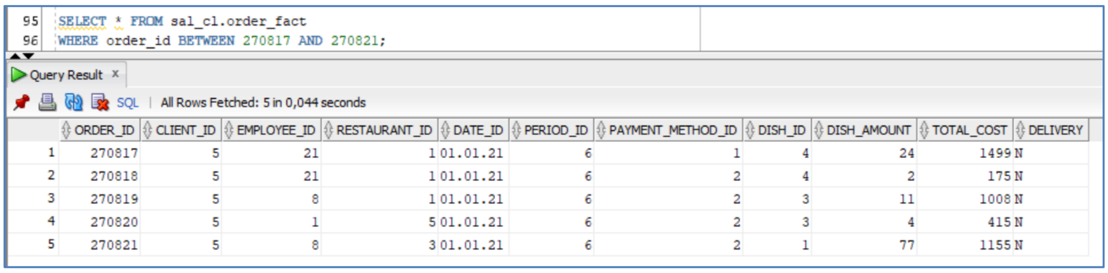
```
ALTER TABLE sal_cl.order_fact EXCHANGE PARTITION quarter_1
WITH TABLE sal_cl.quarter_1_new INCLUDING INDEXES WITHOUT VALIDATION
UPDATE GLOBAL INDEXES;
```

Script Output ×

Task completed in 0,105 seconds

```
Table SAL_CL.ORDER_FACT altered.
```

Replacement result:



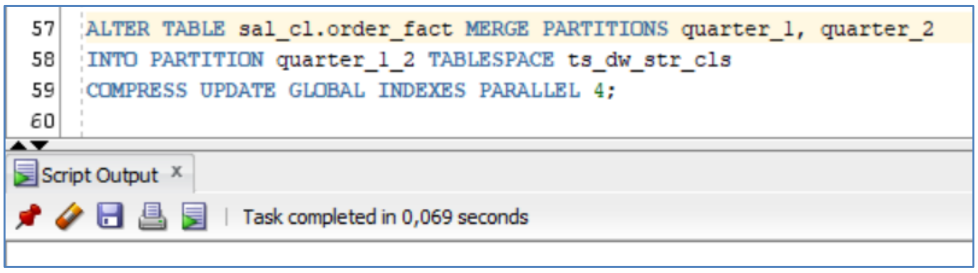
```
SELECT * FROM sal_cl.order_fact
WHERE order_id BETWEEN 270817 AND 270821;
```

Query Result ×

SQL | All Rows Fetched: 5 in 0,044 seconds

	ORDER_ID	CLIENT_ID	EMPLOYEE_ID	RESTAURANT_ID	DATE_ID	PERIOD_ID	PAYMENT_METHOD_ID	DISH_ID	DISH_AMOUNT	TOTAL_COST	DELIVERY
1	270817	5	21	1	01.01.21	6	1	4	24	1499	N
2	270818	5	21	1	01.01.21	6	2	4	2	175	N
3	270819	5	8	1	01.01.21	6	2	3	11	1008	N
4	270820	5	1	5	01.01.21	6	2	3	4	415	N
5	270821	5	8	3	01.01.21	6	2	1	77	1155	N

Let's merge 2 partitions into one:

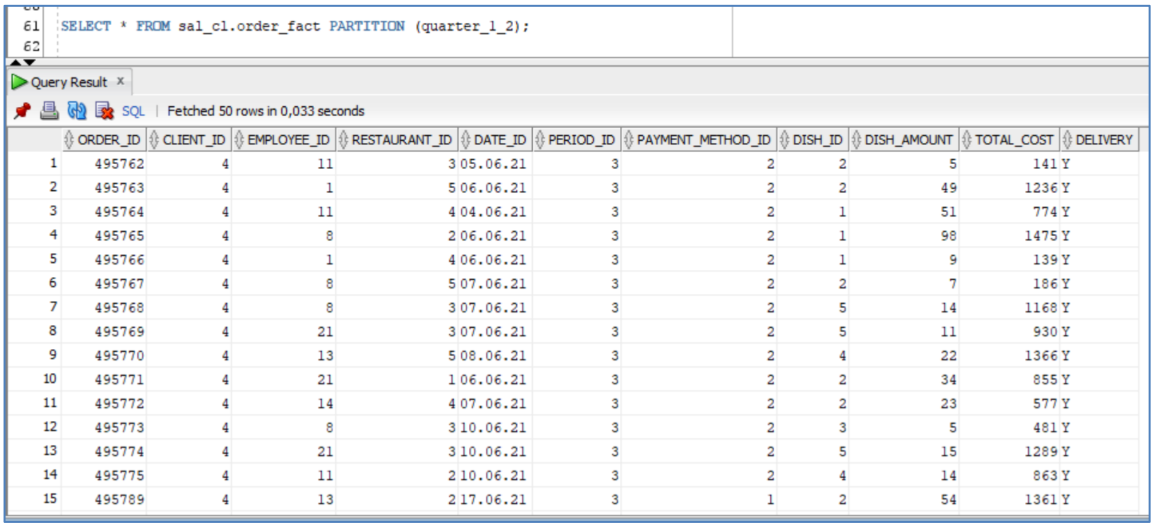


```
ALTER TABLE sal_cl.order_fact MERGE PARTITIONS quarter_1, quarter_2
INTO PARTITION quarter_1_2 TABLESPACE ts_dw_str_cls
COMPRESS UPDATE GLOBAL INDEXES PARALLEL 4;
```

Script Output ×

Task completed in 0,069 seconds

Extract data from a new partition:



```
SELECT * FROM sal_cl.order_fact PARTITION (quarter_1_2);
```

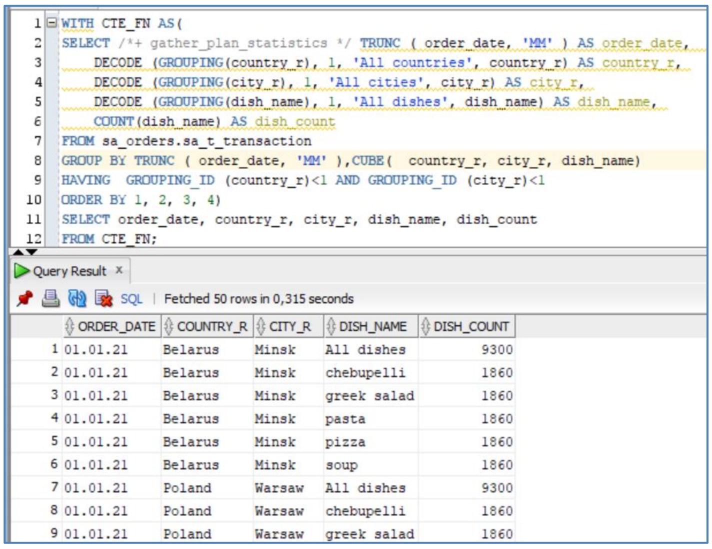
Query Result ×

SQL | Fetched 50 rows in 0,033 seconds

	ORDER_ID	CLIENT_ID	EMPLOYEE_ID	RESTAURANT_ID	DATE_ID	PERIOD_ID	PAYMENT_METHOD_ID	DISH_ID	DISH_AMOUNT	TOTAL_COST	DELIVERY
1	495762	4	11	3	05.06.21	3	2	2	5	141	Y
2	495763	4	1	5	06.06.21	3	2	2	49	1236	Y
3	495764	4	11	4	04.06.21	3	2	1	51	774	Y
4	495765	4	8	2	06.06.21	3	2	1	98	1475	Y
5	495766	4	1	4	06.06.21	3	2	1	9	139	Y
6	495767	4	8	5	07.06.21	3	2	2	7	186	Y
7	495768	4	8	3	07.06.21	3	2	5	14	1168	Y
8	495769	4	21	3	07.06.21	3	2	5	11	930	Y
9	495770	4	13	5	08.06.21	3	2	4	22	1366	Y
10	495771	4	21	1	06.06.21	3	2	2	34	855	Y
11	495772	4	14	4	07.06.21	3	2	2	23	577	Y
12	495773	4	8	3	10.06.21	3	2	3	5	481	Y
13	495774	4	21	3	10.06.21	3	2	5	15	1289	Y
14	495775	4	11	2	10.06.21	3	2	4	14	863	Y
15	495789	4	13	2	17.06.21	3	1	2	54	1361	Y

Task 2 – Prepare Report Layout

Let's execute a query from lab 2, which counts the number of times each item was bought(data is taken from the sa level):



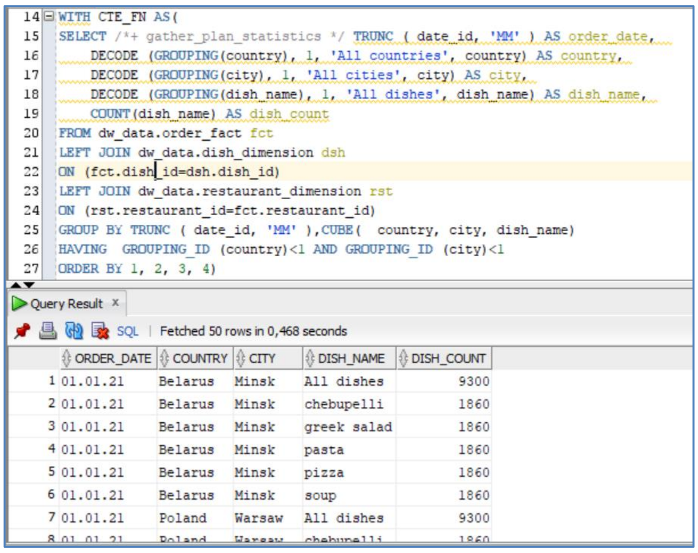
```
WITH CTE_FN AS(
SELECT /*+ gather_plan_statistics */ TRUNC ( order_date, 'MM' ) AS order_date,
    DECODE  (GROUPING(country_r), 1, 'All countries', country_r) AS country_r,
    DECODE  (GROUPING(city_r), 1, 'All cities', city_r) AS city_r,
    DECODE  (GROUPING(dish_name), 1, 'All dishes', dish_name) AS dish_name,
    COUNT(dish_name) AS dish_count
FROM sa_orders.sa_t_transaction
GROUP BY TRUNC ( order_date, 'MM' ),CUBE(  country_r, city_r, dish_name)
HAVING  GROUPING_ID (country_r)<1 AND GROUPING_ID (city_r)<1
ORDER BY 1, 2, 3, 4)
SELECT order_date, country_r, city_r, dish_name, dish_count
FROM CTE_FN;
```

Query Result

SQL | Fetched 50 rows in 0,315 seconds

```
   ORDER_DATE COUNTRY_R CITY_R  DISH_NAME   DISH_COUNT
 1 01.01.21   Belarus   Minsk   All dishes        9300
 2 01.01.21   Belarus   Minsk   chebupelli        1860
 3 01.01.21   Belarus   Minsk   greek salad       1860
 4 01.01.21   Belarus   Minsk   pasta             1860
 5 01.01.21   Belarus   Minsk   pizza             1860
 6 01.01.21   Belarus   Minsk   soup              1860
 7 01.01.21   Poland    Warsaw  All dishes        9300
 8 01.01.21   Poland    Warsaw  chebupelli        1860
 9 01.01.21   Poland    Warsaw  greek salad       1860
```

Now let's demonstrate the same query, but using tables from the dw level (star scheme). Tables are joined using left joins:



```
WITH CTE_FN AS(
SELECT /*+ gather_plan_statistics */ TRUNC ( date_id, 'MM' ) AS order_date,
    DECODE  (GROUPING(country), 1, 'All countries', country) AS country,
    DECODE  (GROUPING(city), 1, 'All cities', city) AS city,
    DECODE  (GROUPING(dish_name), 1, 'All dishes', dish_name) AS dish_name,
    COUNT(dish_name) AS dish_count
FROM dw_data.order_fact fct
LEFT JOIN dw_data.dish_dimension dsh
ON (fct.dish_id=dsh.dish_id)
LEFT JOIN dw_data.restaurant_dimension rst
ON (rst.restaurant_id=fct.restaurant_id)
GROUP BY TRUNC ( date_id, 'MM' ),CUBE(  country, city, dish_name)
HAVING  GROUPING_ID (country)<1 AND GROUPING_ID (city)<1
ORDER BY 1, 2, 3, 4)
```

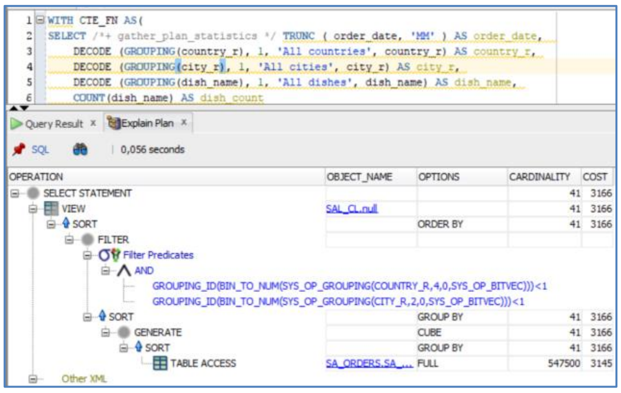
Query Result

SQL | Fetched 50 rows in 0,468 seconds

```
   ORDER_DATE COUNTRY CITY   DISH_NAME   DISH_COUNT
 1 01.01.21   Belarus Minsk  All dishes        9300
 2 01.01.21   Belarus Minsk  chebupelli        1860
 3 01.01.21   Belarus Minsk  greek salad       1860
 4 01.01.21   Belarus Minsk  pasta             1860
 5 01.01.21   Belarus Minsk  pizza             1860
 6 01.01.21   Belarus Minsk  soup              1860
 7 01.01.21   Poland  Warsaw All dishes        9300
 8 01.01.21   Poland  Warsaw chebupelli        1860
```

Let's compare the explain plans of both queries.

Explain plan of query from lab 2:



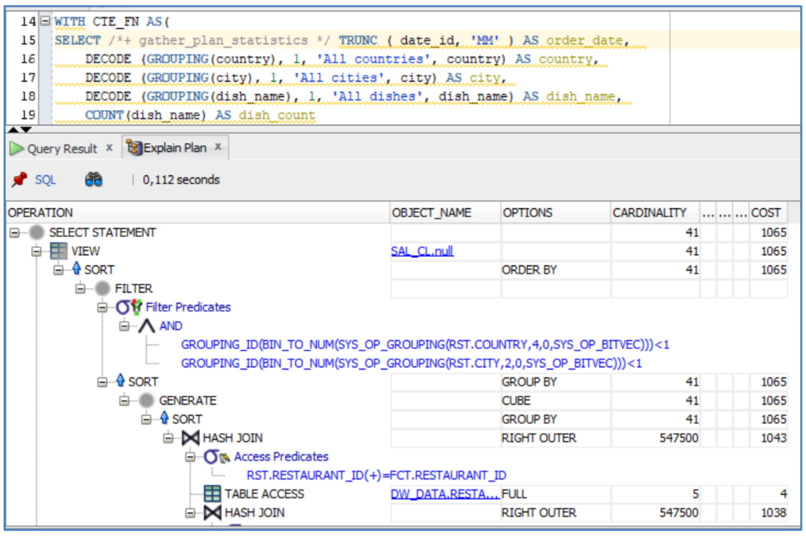
```
WITH CTE_FN AS(
SELECT /*+ gather_plan_statistics */ TRUNC ( order_date, 'MM' ) AS order_date,
    DECODE  (GROUPING(country_r), 1, 'All countries', country_r) AS country_r,
    DECODE  (GROUPING(city_r), 1, 'All cities', city_r) AS city_r,
    DECODE  (GROUPING(dish_name), 1, 'All dishes', dish_name) AS dish_name,
    COUNT(dish_name) AS dish_count
```

Query Result | Explain Plan

SQL | 0,056 seconds

```
OPERATION                                   OBJECT_NAME      OPTIONS   CARDINALITY  COST
SELECT STATEMENT                                                                41  3166
  VIEW                                      SAL_CL.null                         41  3166
    SORT                                                     ORDER BY           41  3166
      FILTER
        Filter Predicates
          AND
            GROUPING_ID(BIN_TO_NUM(SYS_OP_GROUPING(COUNTRY_R,4,0,SYS_OP_BITVEC)))<1
            GROUPING_ID(BIN_TO_NUM(SYS_OP_GROUPING(CITY_R,2,0,SYS_OP_BITVEC)))<1
        SORT                                                 GROUP BY           41  3166
          GENERATE                                           CUBE               41  3166
            SORT                                             GROUP BY           41  3166
              TABLE ACCESS                  SA_ORDERS.SA_... FULL           547500  3145
  Other XML
```

Explain plan of query from lab 11:



```sql
WITH CTE_FN AS(
SELECT /*+ gather_plan_statistics */ TRUNC ( date_id, 'MM' ) AS order_date,
    DECODE (GROUPING(country), 1, 'All countries', country) AS country,
    DECODE (GROUPING(city), 1, 'All cities', city) AS city,
    DECODE (GROUPING(dish_name), 1, 'All dishes', dish_name) AS dish_name,
    COUNT(dish_name) AS dish_count
```

Query Result × | Explain Plan ×

SQL | 0,112 seconds

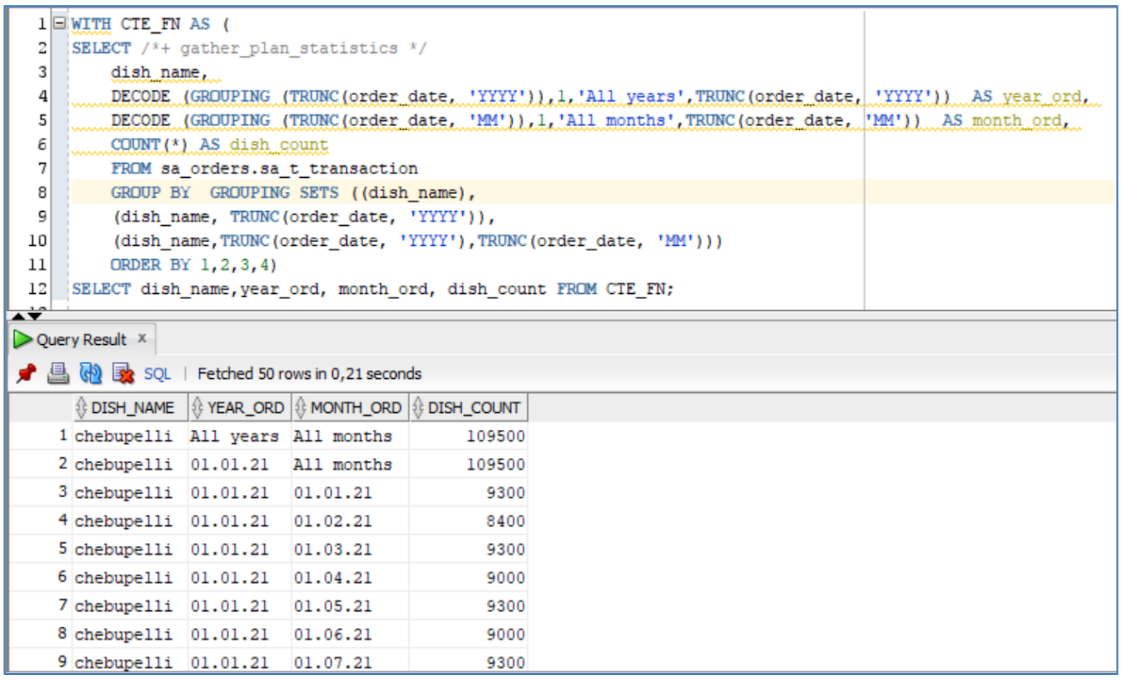
```
OPERATION                                                  OBJECT_NAME      OPTIONS       CARDINALITY  ... ... ... COST
SELECT STATEMENT                                                                                   41              1065
  VIEW                                                     SAL_CL.null                             41              1065
    SORT                                                                    ORDER BY               41              1065
      FILTER
        Filter Predicates
          AND
            GROUPING_ID(BIN_TO_NUM(SYS_OP_GROUPING(RST.COUNTRY,4,0,SYS_OP_BITVEC)))<1
            GROUPING_ID(BIN_TO_NUM(SYS_OP_GROUPING(RST.CITY,2,0,SYS_OP_BITVEC)))<1
        SORT                                                                GROUP BY               41              1065
          GENERATE                                                          CUBE                   41              1065
            SORT                                                            GROUP BY               41              1065
              HASH JOIN                                                     RIGHT OUTER        547500              1043
                Access Predicates
                  RST.RESTAURANT_ID(+)=FCT.RESTAURANT_ID
                TABLE ACCESS                               DW_DATA.RESTA... FULL                    5                 4
                HASH JOIN                                                   RIGHT OUTER        547500              1038
```

From the screenshots, it can be seen that the cost has decreased by almost 3 times, although the execution time has slightly increased.

Task 3 – Compare Report Layout Performance

Let's compare 3 queries that get the quantity of each item found in orders:

1) Lab 2, sa level, advancing grouping

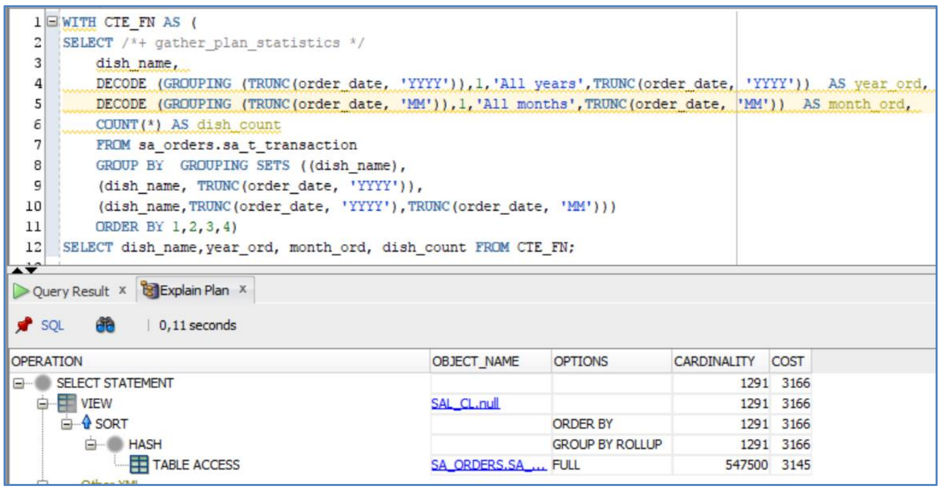


```sql
WITH CTE_FN AS (
SELECT /*+ gather_plan_statistics */
    dish_name,
    DECODE (GROUPING (TRUNC(order_date, 'YYYY')),1,'All years',TRUNC(order_date, 'YYYY'))  AS year_ord,
    DECODE (GROUPING (TRUNC(order_date, 'MM')),1,'All months',TRUNC(order_date, 'MM'))  AS month_ord,
    COUNT(*) AS dish_count
    FROM sa_orders.sa_t_transaction
    GROUP BY  GROUPING SETS ((dish_name),
    (dish_name, TRUNC(order_date, 'YYYY')),
    (dish_name,TRUNC(order_date, 'YYYY'),TRUNC(order_date, 'MM')))
    ORDER BY 1,2,3,4)
SELECT dish_name,year_ord, month_ord, dish_count FROM CTE_FN;
```

Query Result ×

SQL | Fetched 50 rows in 0,21 seconds

	DISH_NAME	YEAR_ORD	MONTH_ORD	DISH_COUNT
1	chebupelli	All years	All months	109500
2	chebupelli	01.01.21	All months	109500
3	chebupelli	01.01.21	01.01.21	9300
4	chebupelli	01.01.21	01.02.21	8400
5	chebupelli	01.01.21	01.03.21	9300
6	chebupelli	01.01.21	01.04.21	9000
7	chebupelli	01.01.21	01.05.21	9300
8	chebupelli	01.01.21	01.06.21	9000
9	chebupelli	01.01.21	01.07.21	9300



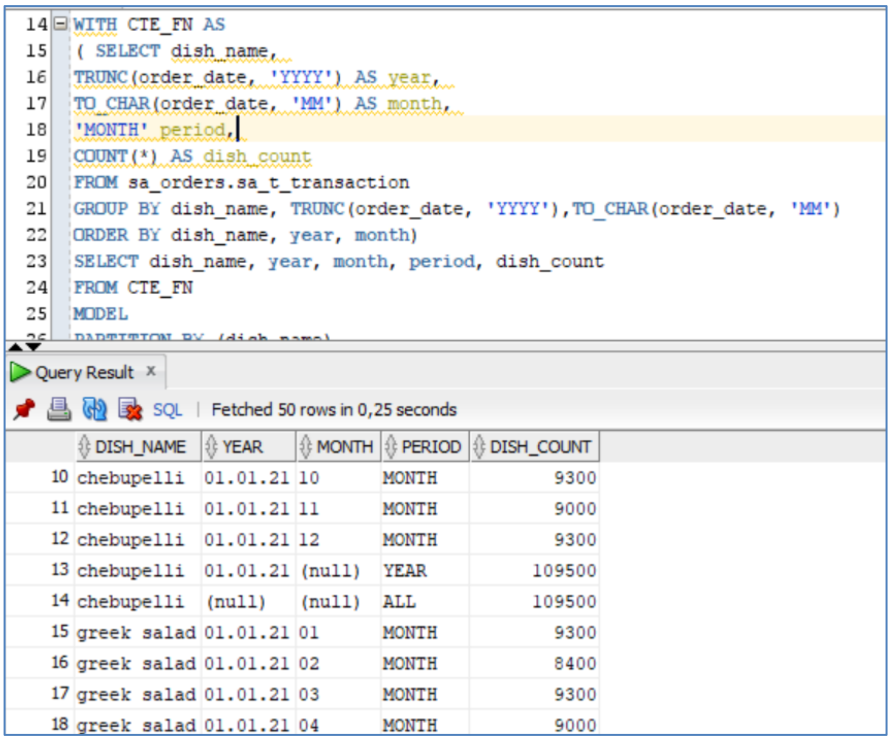
```sql
WITH CTE_FN AS (
SELECT /*+ gather_plan_statistics */
    dish_name,
    DECODE (GROUPING (TRUNC(order_date, 'YYYY')),1,'All years',TRUNC(order_date, 'YYYY'))  AS year_ord,
    DECODE (GROUPING (TRUNC(order_date, 'MM')),1,'All months',TRUNC(order_date, 'MM'))  AS month_ord,
    COUNT(*) AS dish_count
    FROM sa_orders.sa_t_transaction
    GROUP BY  GROUPING SETS ((dish_name),
    (dish_name, TRUNC(order_date, 'YYYY')),
    (dish_name,TRUNC(order_date, 'YYYY'),TRUNC(order_date, 'MM')))
    ORDER BY 1,2,3,4)
SELECT dish_name,year_ord, month_ord, dish_count FROM CTE_FN;
```

Query Result × Explain Plan ×

SQL | 0,11 seconds

```
OPERATION                      OBJECT_NAME       OPTIONS           CARDINALITY  COST
SELECT STATEMENT                                                          1291  3166
  VIEW                         SAL_CL.null                                1291  3166
    SORT                                         ORDER BY                 1291  3166
      HASH                                       GROUP BY ROLLUP          1291  3166
        TABLE ACCESS           SA_ORDERS.SA_...  FULL                   547500  3145
```

2) Lab 5, sa level, model clause:

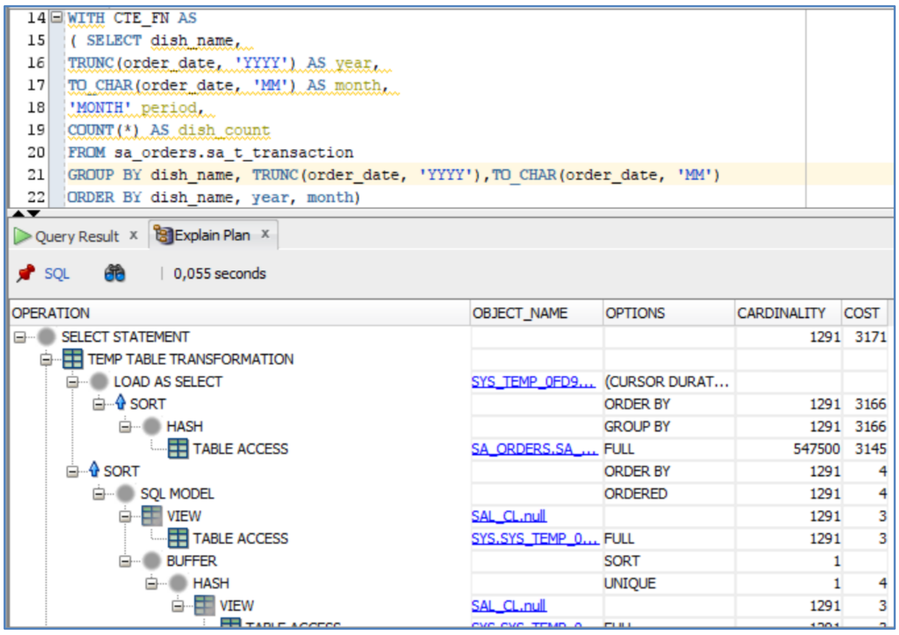


```sql
WITH CTE_FN AS
( SELECT dish_name,
TRUNC(order_date, 'YYYY') AS year,
TO_CHAR(order_date, 'MM') AS month,
'MONTH' period,
COUNT(*) AS dish_count
FROM sa_orders.sa_t_transaction
GROUP BY dish_name, TRUNC(order_date, 'YYYY'),TO_CHAR(order_date, 'MM')
ORDER BY dish_name, year, month)
SELECT dish_name, year, month, period, dish_count
FROM CTE_FN
MODEL
```

Query Result ×

SQL | Fetched 50 rows in 0,25 seconds

```
    DISH_NAME    YEAR      MONTH   PERIOD  DISH_COUNT
10  chebupelli   01.01.21  10      MONTH         9300
11  chebupelli   01.01.21  11      MONTH         9000
12  chebupelli   01.01.21  12      MONTH         9300
13  chebupelli   01.01.21  (null)  YEAR        109500
14  chebupelli   (null)    (null)  ALL         109500
15  greek salad  01.01.21  01      MONTH         9300
16  greek salad  01.01.21  02      MONTH         8400
17  greek salad  01.01.21  03      MONTH         9300
18  greek salad  01.01.21  04      MONTH         9000
```



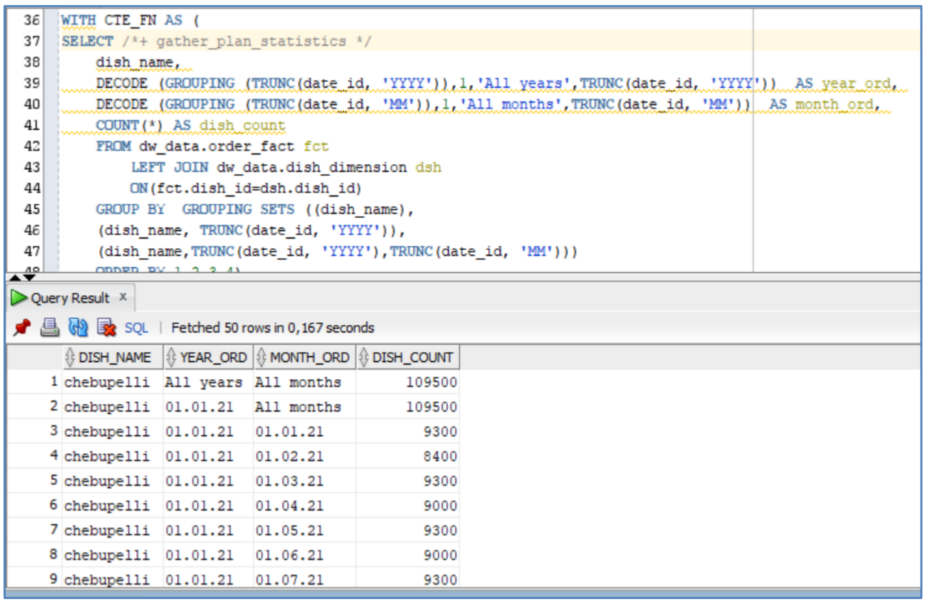
```sql
WITH CTE_FN AS
( SELECT dish_name,
TRUNC(order_date, 'YYYY') AS year,
TO_CHAR(order_date, 'MM') AS month,
'MONTH' period,
COUNT(*) AS dish_count
FROM sa_orders.sa_t_transaction
GROUP BY dish_name, TRUNC(order_date, 'YYYY'),TO_CHAR(order_date, 'MM')
ORDER BY dish_name, year, month)
```

Query Result × Explain Plan ×

SQL | 0,055 seconds

```
OPERATION                          OBJECT_NAME         OPTIONS              CARDINALITY  COST
SELECT STATEMENT                                                                   1291  3171
  TEMP TABLE TRANSFORMATION
    LOAD AS SELECT                 SYS_TEMP_0FD9...    (CURSOR DURAT...
      SORT                                             ORDER BY                    1291  3166
        HASH                                           GROUP BY                    1291  3166
          TABLE ACCESS             SA_ORDERS.SA_...    FULL                      547500  3145
    SORT                                               ORDER BY                    1291     4
      SQL MODEL                                        ORDERED                     1291     4
        VIEW                       SAL_CL.null                                     1291     3
          TABLE ACCESS             SYS.SYS_TEMP_0...   FULL                        1291     3
        BUFFER                                         SORT                           1
          HASH                                         UNIQUE                         1     4
            VIEW                   SAL_CL.null                                     1291     3
```

3) Lab 11, dw level, star scheme:

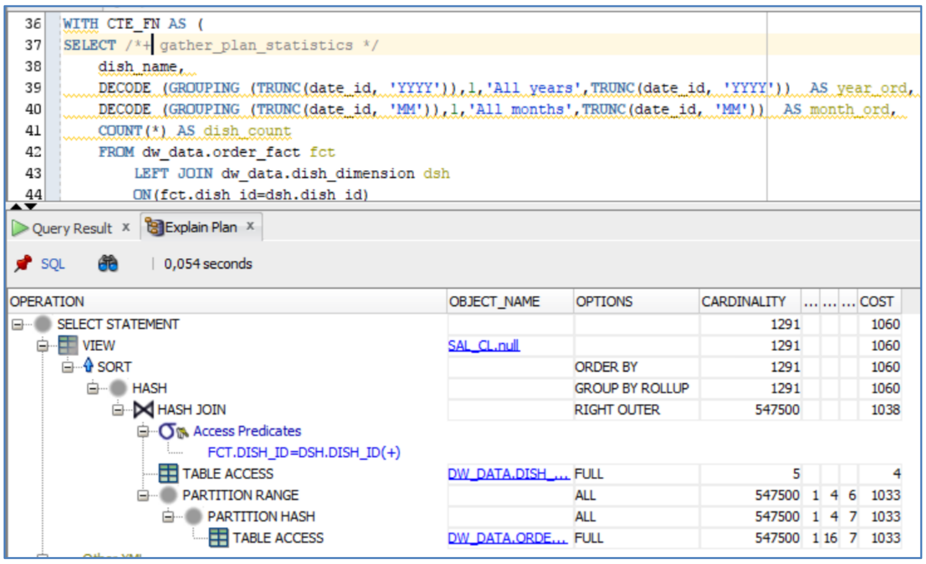


```
WITH CTE_FN AS (
SELECT /*+ gather_plan_statistics */
    dish_name,
    DECODE (GROUPING (TRUNC(date_id, 'YYYY')),1,'All years',TRUNC(date_id, 'YYYY'))  AS year_ord,
    DECODE (GROUPING (TRUNC(date_id, 'MM')),1,'All months',TRUNC(date_id, 'MM'))  AS month_ord,
    COUNT(*) AS dish_count
    FROM dw_data.order_fact fct
        LEFT JOIN dw_data.dish_dimension dsh
        ON(fct.dish_id=dsh.dish_id)
    GROUP BY  GROUPING SETS ((dish_name),
    (dish_name, TRUNC(date_id, 'YYYY')),
    (dish_name,TRUNC(date_id, 'YYYY'),TRUNC(date_id, 'MM')))
```

Query Result ×

SQL | Fetched 50 rows in 0,167 seconds

	DISH_NAME	YEAR_ORD	MONTH_ORD	DISH_COUNT
1	chebupelli	All years	All months	109500
2	chebupelli	01.01.21	All months	109500
3	chebupelli	01.01.21	01.01.21	9300
4	chebupelli	01.01.21	01.02.21	8400
5	chebupelli	01.01.21	01.03.21	9300
6	chebupelli	01.01.21	01.04.21	9000
7	chebupelli	01.01.21	01.05.21	9300
8	chebupelli	01.01.21	01.06.21	9000
9	chebupelli	01.01.21	01.07.21	9300



```
WITH CTE_FN AS (
SELECT /*+ gather_plan_statistics */
    dish_name,
    DECODE (GROUPING (TRUNC(date_id, 'YYYY')),1,'All years',TRUNC(date_id, 'YYYY'))  AS year_ord,
    DECODE (GROUPING (TRUNC(date_id, 'MM')),1,'All months',TRUNC(date_id, 'MM'))  AS month_ord,
    COUNT(*) AS dish_count
    FROM dw_data.order_fact fct
        LEFT JOIN dw_data.dish_dimension dsh
        ON(fct.dish_id=dsh.dish_id)
```

Query Result × | Explain Plan ×

SQL | 0,054 seconds

```
OPERATION                                OBJECT_NAME        OPTIONS           CARDINALITY  ... ... ... COST
SELECT STATEMENT                                                                     1291              1060
  VIEW                                   SAL_CL.null                                 1291              1060
    SORT                                                    ORDER BY                 1291              1060
      HASH                                                  GROUP BY ROLLUP          1291              1060
        HASH JOIN                                           RIGHT OUTER            547500              1038
          Access Predicates
            FCT.DISH_ID=DSH.DISH_ID(+)
          TABLE ACCESS                   DW_DATA.DISH_...   FULL                        5                 4
          PARTITION RANGE                                   ALL                    547500    1   4   6  1033
            PARTITION HASH                                  ALL                    547500    1   4   7  1033
              TABLE ACCESS               DW_DATA.ORDE...    FULL                   547500    1  16   7  1033
```

№	Source Type	Explain Plan - Statistics		Time, Sec.
		Cardinality	Cost	
1	Lab 2, Advancing Grouping	1291	3166	0,21
2	Lab 5, Model Clause	1291	3171	0,25
3	Lab 11, Star Schema	1291	1060	0,167