Lab report #2

Sadovskaya Veronika

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

Task 1 - CREATE Daily Reports Layouts

Day	Profit
01.01.2022	10255
02.01.2022	12000
03.01.2022	7375
04.01.2022	13055
05.01.2022	10955
06.01.2022	5255
07.01.2022	11155
08.01.2022	9055
...	...
Total profit per month	...

Day	Orders
01.01.2022	150
02.01.2022	167
03.01.2022	118
04.01.2022	179
05.01.2022	186
06.01.2022	95
07.01.2022	163
08.01.2022	134
...	...
Total orders per month	...

*not delivery

Dish	Amount
Pasta	120
Pizza	100
Chebupelli	136
Soup	51
Greek salad	200
Total dishes per day	607

Category	Amount dishes
Hot	5000
Appetizer	359
Total dishes per day	5359

Date	Average check of the order
01.01.2022	36
02.01.2022	50
03.01.2022	26
04.01.2022	69
05.01.2022	45
06.01.2022	25
07.01.2022	51
08.01.2022	49
...	...
Average check for month	43,875

Task 2 - CREATE Monthly Reports Layouts

Day	Profit
01.01.2022	1025500
01.02.2022	1200000
01.03.2022	737500
01.04.2022	1305500
01.05.2022	1095500
01.06.2022	525500
01.07.2022	1115500
01.08.2022	905500
...	...
Total profit per year	...

Day	Orders
01.01.2022	15000
01.02.2022	16700
01.03.2022	11800
01.04.2022	17900
01.05.2022	18600
01.06.2022	9500
01.07.2022	16300
01.08.2022	13400
...	...
Total orders per year	...

*not delivery

Dish	Amount
Pasta	12000
Pizza	10000
Chebupelli	13600
Soup	5100
Greek salad	20000
Total dishes per month	60700

Category	Amount dishes
Hot	500000
Appetizer	35900
Total dishes per month	535900

Date	Average check of the order
01.01.2022	3600
01.02.2022	5000
01.03.2022	2600
01.04.2022	6900
01.05.2022	4500
01.06.2022	2500
01.07.2022	5100
01.08.2022	4900
...	...
Average check for year	4387,5

Note: tasks 1 and 2 present the simplest examples of results, to which you can add grouping by country, region, city, delivery, or order payment method.

Task 3 - CREATE Test AdHoc SQL - Daily Reports (CUBE)

```sql
WITH CTE_FN AS (
SELECT  /*+ gather_plan_statistics */ TRUNC ( order_date, 'DD' ) AS order_date, country_r, city_r, SUM(total_cost) AS profit
FROM sa_orders.sa_t_transaction
GROUP BY CUBE(order_date, country_r,  city_r)
HAVING TRUNC ( order_date, 'DD' ) IS NOT NULL AND country_r IS NOT NULL AND city_r IS NOT NULL
ORDER BY 1)
SELECT order_date, country_r, city_r, profit FROM CTE_FN;
```

Query Result

SQL | Fetched 50 rows in 0,261 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	PROFIT
1	01.01.21	Belarus	Minsk	188151
2	01.01.21	USA	New York	184192
3	01.01.21	Poland	Warsaw	186307
4	01.01.21	Ukraine	Kiev	186921
5	01.01.21	Russia	Moscow	197161
6	02.01.21	Belarus	Minsk	189782
7	02.01.21	USA	New York	194885
8	02.01.21	Ukraine	Kiev	208638
9	02.01.21	Russia	Moscow	194597

```sql
WITH CTE_FN AS (
SELECT  /*+ gather_plan_statistics */ TRUNC ( order_date, 'DD' ) AS order_date,
country_r, city_r, COUNT(email_c)  AS visit_amount
FROM sa_orders.sa_t_transaction
WHERE delivery = 'N'
GROUP BY CUBE(TRUNC ( order_date, 'DD' ), country_r, city_r)
HAVING TRUNC ( order_date, 'DD' ) IS NOT NULL AND country_r IS NOT NULL AND city_r IS NOT NULL
ORDER BY 1, 2, 4)
SELECT order_date, country_r, city_r, visit_amount FROM CTE_FN;
```

Query Result

SQL | Fetched 50 rows in 0,158 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	VISIT_AMOUNT
1	01.01.21	Belarus	Minsk	50
2	01.01.21	Poland	Warsaw	50
3	01.01.21	Russia	Moscow	50
4	01.01.21	Ukraine	Kiev	50
5	01.01.21	USA	New York	50
6	02.01.21	Belarus	Minsk	50
7	02.01.21	Poland	Warsaw	50
8	02.01.21	Russia	Moscow	50
9	02.01.21	Ukraine	Kiev	50

```sql
WITH CTE_FN AS (
SELECT  /*+ gather_plan_statistics */ TRUNC ( order_date, 'DD' ) AS order_date,
country_r, city_r,dish_name, COUNT(dish_name)  AS dish_count
FROM sa_orders.sa_t_transaction
GROUP BY CUBE(TRUNC ( order_date, 'DD' ), country_r, city_r,dish_name)
HAVING TRUNC ( order_date, 'DD' ) IS NOT NULL AND country_r IS NOT NULL AND city_r IS NOT NULL AND dish_name IS NOT NULL
ORDER BY 1,2,4,5)
SELECT order_date, country_r, city_r, dish_name,dish_count FROM CTE_FN;
```

Query Result

SQL | Fetched 50 rows in 0,291 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	DISH_NAME	DISH_COUNT
1	01.01.21	Belarus	Minsk	chebupelli	50
2	01.01.21	Belarus	Minsk	greek salad	50
3	01.01.21	Belarus	Minsk	pasta	50
4	01.01.21	Belarus	Minsk	pizza	50
5	01.01.21	Belarus	Minsk	soup	50
6	01.01.21	Poland	Warsaw	chebupelli	50
7	01.01.21	Poland	Warsaw	greek salad	50
8	01.01.21	Poland	Warsaw	pasta	50
9	01.01.21	Poland	Warsaw	pizza	50

```sql
WITH CTE_FN AS (
SELECT  /*+ gather_plan_statistics */ TRUNC ( order_date, 'DD' ) AS order_date,
country_r, city_r,dish_category, COUNT(dish_name)  AS dish_count
FROM sa_orders.sa_t_transaction
GROUP BY CUBE(TRUNC ( order_date, 'DD' ), country_r, city_r,dish_category)
HAVING TRUNC ( order_date, 'DD' ) IS NOT NULL AND country_r IS NOT NULL AND city_r IS NOT NULL AND dish_category IS NOT NULL
ORDER BY 1,2,4,5)
SELECT order_date, country_r, city_r, dish_category,dish_count FROM CTE_FN;
```

Query Result

Fetched 50 rows in 0,211 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	DISH_CATEGORY	DISH_COUNT
1	01.01.21	Belarus	Minsk	appetizer	50
2	01.01.21	Belarus	Minsk	hot	200
3	01.01.21	Poland	Warsaw	appetizer	50
4	01.01.21	Poland	Warsaw	hot	200
5	01.01.21	Russia	Moscow	appetizer	50
6	01.01.21	Russia	Moscow	hot	200
7	01.01.21	Ukraine	Kiev	appetizer	50
8	01.01.21	Ukraine	Kiev	hot	200
9	01.01.21	USA	New York	appetizer	50

```sql
WITH CTE_FN AS (
SELECT  /*+ gather_plan_statistics */ TRUNC ( order_date, 'DD' ) AS order_date, country_r,
city_r, AVG(total_cost) AS avd_chek
FROM sa_orders.sa_t_transaction
GROUP BY CUBE(order_date, country_r,  city_r)
HAVING TRUNC ( order_date, 'DD' ) IS NOT NULL AND country_r IS NOT NULL AND city_r IS NOT NULL
ORDER BY 1)
SELECT order_date, country_r, city_r, avd_chek FROM CTE_FN;
```

Query Result

Fetched 50 rows in 0,283 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	AVD_CHEK
1	01.01.21	Belarus	Minsk	752,604
2	01.01.21	USA	New York	736,768
3	01.01.21	Poland	Warsaw	745,228
4	01.01.21	Ukraine	Kiev	747,684
5	01.01.21	Russia	Moscow	788,644
6	02.01.21	Belarus	Minsk	759,128
7	02.01.21	USA	New York	779,54
8	02.01.21	Ukraine	Kiev	834,552
9	02.01.21	Russia	Moscow	778,388

Task 4 - CREATE Test AdHoc SQL - Monthly Reports (ROLLUP & GROUPING SETS)

```sql
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,
    SUM(total_cost) AS profit
FROM sa_orders.sa_t_transaction
GROUP BY TRUNC ( order_date, 'MM' ),CUBE(  country_r, city_r)
HAVING  GROUPING_ID (country_r)<1
ORDER BY 1, 2, 3, 4)
SELECT order_date, country_r, city_r, profit
FROM CTE_FN;
```

Query Result

Fetched 50 rows in 0,227 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	PROFIT
1	01.01.21	Belarus	All cities	5830956
2	01.01.21	Belarus	Minsk	5830956
3	01.01.21	Poland	All cities	5827673
4	01.01.21	Poland	Warsaw	5827673
5	01.01.21	Russia	All cities	5850846
6	01.01.21	Russia	Moscow	5850846
7	01.01.21	Ukraine	All cities	5817933
8	01.01.21	Ukraine	Kiev	5817933
9	01.01.21	USA	All cities	5869857
10	01.01.21	USA	New York	5869857

```sql
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,
    COUNT(email_c)  AS visit_amount
FROM sa_orders.sa_t_transaction
WHERE delivery = 'N'
GROUP BY TRUNC ( order_date, 'MM' ),CUBE( country_r, city_r)
HAVING  GROUPING_ID (country_r)<1
ORDER BY 1, 2, 3, 4)
SELECT order_date, country_r, city_r, visit_amount
FROM CTE_FN;
```

Query Result

SQL | Fetched 50 rows in 0,135 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	VISIT_AMOUNT
1	01.01.21	Belarus	All cities	1550
2	01.01.21	Belarus	Minsk	1550
3	01.01.21	Poland	All cities	1550
4	01.01.21	Poland	Warsaw	1550
5	01.01.21	Russia	All cities	1550
6	01.01.21	Russia	Moscow	1550
7	01.01.21	Ukraine	All cities	1550
8	01.01.21	Ukraine	Kiev	1550
9	01.01.21	USA	All cities	1550

```sql
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 0,262 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	DISH_NAME	DISH_COUNT
1	01.01.21	Belarus	Minsk	All dishes	7750
2	01.01.21	Belarus	Minsk	chebupelli	1550
3	01.01.21	Belarus	Minsk	greek salad	1550
4	01.01.21	Belarus	Minsk	pasta	1550
5	01.01.21	Belarus	Minsk	pizza	1550
6	01.01.21	Belarus	Minsk	soup	1550
7	01.01.21	Poland	Warsaw	All dishes	7750
8	01.01.21	Poland	Warsaw	chebupelli	1550
9	01.01.21	Poland	Warsaw	greek salad	1550

```sql
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_category), 1, 'All categories', dish_category) AS dish_category,
    COUNT(dish_name) AS dish_count
FROM sa_orders.sa_t_transaction
GROUP BY TRUNC ( order_date, 'MM' ), ROLLUP( country_r, city_r,dish_category)
ORDER BY 1,2,4,5)
SELECT order_date, country_r, city_r, dish_category,dish_count FROM CTE_FN;
```

Query Result

SQL | Fetched 50 rows in 0,189 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	DISH_CATEGORY	DISH_COUNT
1	01.01.21	All countries	All cities	All categories	38750
2	01.01.21	Belarus	All cities	All categories	7750
3	01.01.21	Belarus	Minsk	All categories	7750
4	01.01.21	Belarus	Minsk	appetizer	1550
5	01.01.21	Belarus	Minsk	hot	6200
6	01.01.21	Poland	Warsaw	All categories	7750
7	01.01.21	Poland	All cities	All categories	7750
8	01.01.21	Poland	Warsaw	appetizer	1550
9	01.01.21	Poland	Warsaw	hot	6200

```sql
WITH CTE_FN AS (
SELECT  /*+ gather_plan_statistics */ TRUNC ( order_date, 'DD' ) 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,
    AVG(total_cost) AS avd_chek
FROM sa_orders.sa_t_transaction
GROUP BY TRUNC ( order_date, 'DD' ),
    GROUPING SETS ((order_date), (country_r),(city_r),(order_date, country_r,city_r ))
HAVING  GROUPING_ID (country_r)<1
ORDER BY 1,2,3,4)
SELECT order_date, country_r, city_r, avd_chek FROM CTE_FN;
```

Query Result

SQL | Fetched 50 rows in 0,254 seconds

	ORDER_DATE	COUNTRY_R	CITY_R	AVD_CHEK
1	01.01.21	Belarus	All cities	752,604
2	01.01.21	Belarus	Minsk	752,604
3	01.01.21	Poland	All cities	745,228
4	01.01.21	Poland	Warsaw	745,228
5	01.01.21	Russia	All cities	788,644
6	01.01.21	Russia	Moscow	788,644
7	01.01.21	Ukraine	All cities	747,684
8	01.01.21	Ukraine	Kiev	747,684
9	01.01.21	USA	All cities	736,768

Task 5 - CREATE Test AdHoc SQL – ROLLUP by Time

```sql
SELECT DECODE ( GROUPING_ID ( TRUNC ( order_date, 'YYYY' )
                            , TRUNC ( order_date, 'Q' )
                            , TRUNC ( order_date, 'MM' )
                            , TRUNC ( order_date, 'DD' ) )
                 , 7, 'Total for year'
                 , 15, 'GRANT TOTAL'
```

Query Result

SQL | All Rows Fetched: 383 in 0,472 seconds

	YEAR	QUARTER	MONTH	DAY	TOTAL_COST
361	01.01.21	01.10.21	01.12.21	27.12.21	974398
362	01.01.21	01.10.21	01.12.21	28.12.21	949846
363	01.01.21	01.10.21	01.12.21	30.12.21	931128
364	01.01.21	01.10.21	01.12.21	31.12.21	931982
365	01.01.21	01.10.21	01.12.21	29.12.21	940218
366	01.01.21	01.01.21	Total for month	(null)	29197265
367	01.01.21	01.01.21	Total for month	(null)	26354553
368	01.01.21	01.01.21	Total for month	(null)	29083715
369	01.01.21	01.04.21	Total for month	(null)	28120665
370	01.01.21	01.04.21	Total for month	(null)	29087063
371	01.01.21	01.04.21	Total for month	(null)	28253245
372	01.01.21	01.07.21	Total for month	(null)	29316375
373	01.01.21	01.07.21	Total for month	(null)	29048233
374	01.01.21	01.07.21	Total for month	(null)	28043813

```
SELECT DECODE ( GROUPING_ID ( TRUNC ( order_date, 'YYYY' )
                            , TRUNC ( order_date, 'Q' )
                            , TRUNC ( order_date, 'MM' )
                            , TRUNC ( order_date, 'DD' ) )
                , 7, 'Total for year'
                , 15, 'GRANT TOTAL'
```

Query Result

All Rows Fetched: 383 in 0,472 seconds

	YEAR	QUARTER	MONTH	DAY	TOTAL_COST
370	01.01.21	01.04.21	Total for month	(null)	29087063
371	01.01.21	01.04.21	Total for month	(null)	28253245
372	01.01.21	01.07.21	Total for month	(null)	29316375
373	01.01.21	01.07.21	Total for month	(null)	29048233
374	01.01.21	01.07.21	Total for month	(null)	28043813
375	01.01.21	01.10.21	Total for month	(null)	29147342
376	01.01.21	01.10.21	Total for month	(null)	28202780
377	01.01.21	01.10.21	Total for month	(null)	29222714
378	01.01.21	Total for quarter	(null)	(null)	84635533
379	01.01.21	Total for quarter	(null)	(null)	85460973
380	01.01.21	Total for quarter	(null)	(null)	86408421
381	01.01.21	Total for quarter	(null)	(null)	86572836
382	Total for year	(null)	(null)	(null)	343077763
383	GRANT TOTAL	(null)	(null)	(null)	343077763