

U2.Exit task

Sadovskaya Veronika

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

Task 1 - Test Star Data

Cleancing level:

Re-run the script with procedures that fill tables with data from sa level to dw_cl:

The screenshot shows a PL/SQL editor window with the following code:

```
1 BEGIN
2     pkg_etl_clients_cl.load_CLEAN_CLIENTS;
3     pkg_etl_dishes_cl.load_CLEAN_DISHES;
4     pkg_etl_employees_cl.load_CLEAN_EMPLOYEES;
5     pkg_etl_gen_priods_cl.load_CLEAN_GEN_PERIODS;
6     pkg_etl_payment_methods_cl.load_CLEAN_PAYMENT_METHODS;
7     pkg_etl_restaurants_cl.load_CLEAN_RESTAURANTS;
8     pkg_etl_transactions_cl.load_CLEAN_TRANSACTIONS;
9 END;
```

Below the code, the "Script Output" tab shows:

Task completed in 37,926 seconds

PL/SQL procedure successfully completed.

Selects from tables on dw_cl level:

1) dw_cl.cl_t_client

	FIRST_NAME	LAST_NAME	PHONE	EMAIL	STREET	COUNTRY	CITY	CLIENT_STATUS
1	Adriana	Karnitskaya	375294217925	AdrianaKarnitskaya@mail.ru	Yesenin	Belarus	Minsk	Y
2	Ada	Alymova	375292376362	AdaAlymova@mail.ru	Nikolskaya	Russia	Moscow	Y
3	Alice	Lysenko	375294487084	AliceLysenko@mail.ru	Fifth Avenue	USA	New York	Y
4	Ian	Astafyev	375295321165	IanAstafyev@mail.ru	Champs Elysees	France	Paris	Y
5	Pavel	Moshko	375291330060	PavelMoshko@mail.ru	Antoine Dansaert	Belgium	Brussels	Y

2) dw_cl.cl_t_dish

	DISH_NAME	DISH_CATEGORY	PRICE	COMPOSITION	WEIGHT	DISH_STATUS
1	chebupelli	hot	61	chebupelli ingredients	512	Y
2	pasta	hot	87	pasta ingredients	959	Y
3	soup	hot	15	soup ingredients	915	Y
4	pizza	hot	25	pizza ingredients	486	Y
5	greek salad	appetizer	82	greek salad ingredients	797	Y

3) dw_cl.cl_t_employee (SCD2)

	FIRST_NAME	LAST_NAME	PHONE	EMAIL	DEPARTMENT	JOB_TITLE	STREET	COUNTRY	CITY	BUILDING	APARTMENT	RESTAURANT_STATUS
1	Adriana	Karnitskaya	375295542493	AdrianaKarnitskaya@mail.ru	department_name_1	director	Yesenin	Belarus	Minsk			
2	Ada	Alyanova	375292270876	AdaAlyanova@mail.ru	department_name_2	manager	Nikolskaya	Russia	Moscow			
3	Alice	Lysenko	375292277398	AliceLysenko@mail.ru	department_name_3	employee	Fifth Avenue	USA	New York			
4	Alyssa	Malysheva	375296172078	AlyssaMalysheva@mail.ru	department_name_4	employee	Vladimirskaya	Ukraine	Kiev			
5	Anna	Buyanova	375298636808	AnnaBuyanova@mail.ru	department_name_5	employee	Gurchevskaya	Poland	Warsaw			
6	Christina	Mayorova	375291650540	ChristinaMayorova@mail.ru	department_name_1	director	Gammel Strand	Denmark	Copenhagen			
7	Clara	Zaykova	375297467232	ClaraZaykova@mail.ru	department_name_2	manager	Abby	England	London			
8	Veronika	Sadovskaya	375291959494	VeronikaSadovskaya@mail.ru	department_name_3	employee	D Dunkri	Estonia	Tallinn			
9	Eva	Moshko	375297667494	EvaMoshko@mail.ru	department_name_4	employee	Aloyas	Latvia	Riga			
10	Vera	Lebedeva	375293430946	VeraLebedeva@mail.ru	department_name_5	employee	G Galve	Lithuania	Vilnius			
11	Alexandra	Etkina	375293183125	AlexandraEtkina@mail.ru	department_name_1	director	Graben	Austria	Vienna			
12	Rita	Astafyeva	375294007451	RitaAstafyeva@mail.ru	department_name_2	manager	K Kaiser-Friedrich	Germany	Berlin			
13	Ian	Astafyev	375294996103	IanAstafyev@mail.ru	department_name_3	employee	Champs Elysees	France	Paris			
14	Pavel	Moshko	375295241317	PavelMoshko@mail.ru	department_name_4	employee	Antoine Dansaert	Belgium	Brussels			
15	Prokhor	Lebedev	375299557623	ProkhorLebedev@mail.ru	department_name_5	employee	Via del Corso	Italy	Rome			
16	Oleg	Etkin	375298010701	OlegEtkin@mail.ru	department_name_1	director	Abby	Belarus	Brest			

4) dw_cl.cl_t_gen_period

	VALID_FROM	VALID_TO	PROMOTION_NAME	PROMOTION_PERCENT	DESCRIPTION
1	01.01.21	31.01.21	promotion_name_1	25	description_1
2	01.02.21	28.02.21	promotion_name_2	15	description_2
3	01.03.21	31.03.21	promotion_name_3	10	description_3
4	01.04.21	30.04.21	promotion_name_4	5	description_4
5	01.05.21	31.05.21	promotion_name_5	20	description_5
6	01.06.21	31.12.21	promotion_name_6	20	description_6

5) dw_cl.cl_t_payment_method

	PAYMENT_METHOD_NAME	PAYMENT_METHOD_STATUS
1	bank card	Y
2	cash	Y

6) dw_cl.cl_t_restaurant

	PHONE	EMAIL	STREET	COUNTRY	CITY	BUILDING	APARTMENT	RESTAURANT_STATUS
1	375294308850	1@mail.ru	Yesenin	Belarus	Minsk	81	132	Y
2	375299013465	2@mail.ru	Nikolskaya	Russia	Moscow	32	165	Y
3	375292356572	3@mail.ru	Fifth Avenue	USA	New York	30	189	Y
4	375295176493	4@mail.ru	Vladimirskaya	Ukraine	Kiev	36	185	Y
5	375295841669	5@mail.ru	Gurchevskaya	Poland	Warsaw	46	230	Y

7) dw_cl.cl_t_transaction

	FIRST_NAME_C	LAST_NAME_C	PHONE_C	EMAIL_C	STREET_C	COUNTRY_C	CITY_C	CLIENT_STATUS	DISH_NAME	DISH_C
1	Alice	Lysenko	375294487084	AliceLysenko@mail.ru	Fifth Avenue	USA	New York	Y	cheburelli	hot
2	Adriana	Karnitskaya	375294217925	AdrianaKarnitskaya@mail.ru	Yesenin	Belarus	Minsk	Y	greek salad	appetizer
3	Ada	Alyanova	375292270876	AdaAlyanova@mail.ru	Nikolskaya	Russia	Moscow	Y	cheburelli	hot
4	Alice	Lysenko	375294487084	AliceLysenko@mail.ru	Fifth Avenue	USA	New York	Y	pasta	hot
5	Alice	Lysenko	375294487084	AliceLysenko@mail.ru	Fifth Avenue	USA	New York	Y	pasta	hot
6	Alice	Lysenko	375294487084	AliceLysenko@mail.ru	Fifth Avenue	USA	New York	Y	pizza	hot
7	Ian	Astafyev	375295321165	IanAstafyev@mail.ru	Champs Elysees	France	Paris	Y	pasta	hot
8	Ian	Astafyev	375295321165	IanAstafyev@mail.ru	Champs Elysees	France	Paris	Y	greek salad	appetizer
9	Ada	Alyanova	375292376362	AdaAlyanova@mail.ru	Nikolskaya	Russia	Moscow	Y	greek salad	appetizer
10	Ada	Alyanova	375292376362	AdaAlyanova@mail.ru	Nikolskaya	Russia	Moscow	Y	cheburelli	hot
11	Ada	Alyanova	375292376362	AdaAlyanova@mail.ru	Nikolskaya	Russia	Moscow	Y	cheburelli	hot
12	Ian	Astafyev	375295321165	IanAstafyev@mail.ru	Champs Elysees	France	Paris	Y	cheburelli	hot
13	Ian	Astafyev	375295321165	IanAstafyev@mail.ru	Champs Elysees	France	Paris	Y	pasta	hot
14	Ian	Astafyev	375295321165	IanAstafyev@mail.ru	Champs Elysees	France	Paris	Y	pasta	hot
15	Ian	Astafyev	375295321165	IanAstafyev@mail.ru	Champs Elysees	France	Paris	Y	soup	hot
16	Alice	Lysenko	375294487084	AliceLysenko@mail.ru	Fifth Avenue	USA	New York	Y	pizza	hot

DW level:

Re-run the script with procedures that fill tables with data from dw_cl level to dw:

```

1 BEGIN
2   pkg_etl_clients_dw.load_CLEAN_CLIENTS_DW;
3   pkg_etl_restaurants_dw.load_CLEAN_RESTAURANTS_DW;
4   pkg_etl_employees_dw.load_CLEAN_EMPLOYEES_DW;
5   pkg_etl_dishes_dw.load_CLEAN_DISHES_DW;
6   pkg_etl_payment_methods_dw.load_CLEAN_PAYMENT_METHODS_DW;
7   pkg_etl_gen_periods_dw.load_CLEAN_GEN_PERIODS_DW;
8   pkg_etl_transactions_dw.load_CLEAN_TRANSACTIONS_DW;
9 END;
10

```

Script Output | Task completed in 221,453 seconds

PL/SQL procedure successfully completed.

Selects from tables on dw level:

1) dw_data.client_dimension

	CLIENT_ID	FIRST_NAME	LAST_NAME	PHONE	EMAIL	STREET	COUNTRY	CITY	STATUS
1	1	Adriana	Karnitskaya	375294217925	AdrianaKarnitskaya@mail.ru	Yesenin	Belarus	Minsk	Y
2	2	Ian	Astafyev	375295321165	IanAstafyev@mail.ru	Champs Elysees	France	Paris	Y
3	3	Ada	Alyanova	375292376362	AdaAlyanova@mail.ru	Nikolskaya	Russia	Moscow	Y
4	4	Pavel	Moshko	375291330060	PavelMoshko@mail.ru	Antoine Dansaert	Belgium	Brussels	Y
5	5	Alice	Lysenko	375294487084	AliceLysenko@mail.ru	Fifth Avenue	USA	New York	Y

2) dw_data.restaurant_dimension

	RESTAURANT_ID	PHONE	EMAIL	ADDRESS	COUNTRY	CITY	BUILDING	APARTMENT	STATUS
1	1	375295841669	5@mail.ru	Gurchevskaya	Poland	Warsaw	46	230	Y
2	2	375292356572	3@mail.ru	Fifth Avenue	USA	New York	30	189	Y
3	3	375299013465	2@mail.ru	Nikolskaya	Russia	Moscow	32	165	Y
4	4	375295176493	4@mail.ru	Vladimirskaya	Ukraine	Kiev	36	185	Y
5	5	375294308850	1@mail.ru	Yesenin	Belarus	Minsk	81	132	Y

3) dw_data.employee_dimension (SCD2)

	PHONE	EMAIL	DEPARTMENT	JOB_TITLE	ADDRESS	COUNTRY	CITY	BUILDING	APARTMENT	START_DATE	END_DATE	IS_ACTIVE
1	375295542493	AdrianaKarnitskaya@mail.ru	department_name_1	director	Yesenin	Belarus	Minsk	5	143	21.08.22	01.01.99	Y
2	375296172078	AlyssaMalysheva@mail.ru	department_name_4	employee	Vladimirskaya	Ukraine	Kiev	91	189	21.08.22	01.01.99	Y
3	375294007451	RitaAstafyeva@mail.ru	department_name_2	manager	K Kaiser-Friedrich	Germany	Berlin	11	199	21.08.22	01.01.99	Y
4	375297467234	ClaraZaykova@mail.ru	department_name_2	manager	Abby	England	London	9	164	21.08.22	01.01.99	Y
5	375297667494	EvaMoshko@mail.ru	department_name_4	employee	Aloyas	Latvia	Riga	34	185	21.08.22	01.01.99	Y
6	375293676621	IvanSemin@mail.ru	department_name_4	employee	Via del Corso	Belgium	Brussels	69	229	21.08.22	01.01.99	Y
7	37529496103	IanAstafyev@mail.ru	department_name_3	employee	Champs Elysees	France	Paris	97	43	21.08.22	01.01.99	Y
8	375292270876	AdaAlyanova@mail.ru	department_name_2	manager	Nikolskaya	Russia	Moscow	31	290	21.08.22	01.01.99	Y
9	375298589471	MaximMayorov@mail.ru	department_name_2	manager	D Dunkri	England	London	92	63	21.08.22	01.01.99	Y
10	375291959498	VeronikaSadovskaya@mail.ru	department_name_3	employee	D Dunkri	Estonia	Tallinn	92	83	21.08.22	01.01.99	Y
11	375295241317	PavelMoshko@mail.ru	department_name_4	employee	Antoine Dansaert	Belgium	Brussels	79	186	21.08.22	01.01.99	Y
12	375293183125	AlexandraEtkina@mail.ru	department_name_1	director	Graben	Austria	Vienna	4	19	21.08.22	01.01.99	Y
13	375292197332	NikitaMalyshев@mail.ru	department_name_4	employee	G Galve	Latvia	Riga	32	117	21.08.22	01.01.99	Y
14	375292934093	MironParfencov@mail.ru	department_name_3	employee	Antoine Dansaert	France	Paris	36	205	21.08.22	01.01.99	Y

4) dw_data.dish_dimension

Query Result						
SQL All Rows Fetched: 5 in 0,035 seconds						
DISH_ID	DISH_NAME	DISH_CATEGORY	PRICE	COMPOSITION	WEIGHT	STATUS
1	1 soup	hot	15	soup ingredients	915 Y	
2	2 pizza	hot	25	pizza ingredients	486 Y	
3	3 pasta	hot	87	pasta ingredients	959 Y	
4	4 chebupelli	hot	61	chebupelli ingredients	512 Y	
5	5 greek salad appetizer		82	greek salad ingredients	797 Y	

5) dw_data.payment_method_dimension

Query Result		
SQL All Rows Fetched: 2 in 0,023 seconds		
PAYMENT_METHOD_ID	PAYMENT_METHOD_NAME	STATUS
1	1 bank card	Y
2	2 cash	Y

6) dw_data.dim_gen_period

Query Result					
SQL All Rows Fetched: 6 in 0,028 seconds					
PERIOD_ID	VALID_FROM	VALID_TO	PROMOTION_NAME	PROMOTION_PERCENT	DESCRIPTION
1	1 01.04.21	30.04.21	promotion_name_4	5	decription_4
2	2 01.02.21	28.02.21	promotion_name_2	15	decription_2
3	3 01.06.21	31.12.21	promotion_name_6	20	decription_6
4	4 01.03.21	31.03.21	promotion_name_3	10	decription_3
5	5 01.05.21	31.05.21	promotion_name_5	20	decription_5
6	6 01.01.21	31.01.21	promotion_name_1	25	decription_1

7) dw_data.order_fact

Query Result											
SQL Fetched 50 rows in 0,262 seconds											
ORDER_ID	CLIENT_ID	EMPLOYEE_ID	RESTAURANT_ID	DATE_ID	PERIOD_ID	PAYMENT_METHOD_ID	DISH_ID	DISH_AMOUNT	TOTAL_COST	DELIVERY	
1	1	5	13	4 06.01.21	6	1	4	6	422 Y		
2	2	1	11	1 05.01.21	6	2	5	10	834 Y		
3	3	3	21	5 05.01.21	6	2	4	22	1351 Y		
4	4	5	11	4 06.01.21	6	1	3	13	1191 Y		
5	5	5	8	1 06.01.21	6	1	3	14	1223 Y		
6	6	5	14	1 06.01.21	6	1	2	11	291 Y		
7	7	2	13	5 06.01.21	6	1	3	15	1354 Y		
8	8	2	11	5 06.01.21	6	1	5	11	903 Y		
9	9	3	11	2 05.01.21	6	2	5	5	453 Y		
10	10	3	13	4 05.01.21	6	2	4	0	33 Y		
11	11	3	14	4 05.01.21	6	2	4	15	942 Y		
12	12	2	13	2 06.01.21	6	1	4	22	1402 Y		
13	13	2	13	2 06.01.21	6	1	3	7	685 Y		
14	14	2	11	4 06.01.21	6	1	3	13	1165 Y		
15	15	2	1	4 06.01.21	6	1	1	94	1267 Y		

SAL level:

Let's create view sal_cl.employee_dim_actual_position that contains information about the duration of work of all employees in the current position:

22	CREATE OR REPLACE VIEW sal_cl.employee_dim_actual_position AS
23	SELECT first_name ' ' last_name AS employee_name,
24	phone,
25	email,
26	department,
27	job_title AS position,
28	start_date,
29	(TO_DATE(SYSDATE, 'DD/MM/YYYY') - TO_DATE(start_date, 'DD/MM/YYYY')) AS num_days_at_position
30	FROM dw_data.employee_dimension
31	WHERE is_active='Y';

View SAL_CL.EMPLOYEE_DIM_ACTUAL_POSITION created.

Select from view sal_cl.employee_dim_actual_position:

EMPLOYEE_NAME	PHONE	EMAIL	DEPARTMENT	POSITION	START_DATE	NUM_DAYA_AT_POSITION
Adriana Karnitskaya	375295542493	AdrianaKarnitskaya@mail.ru	department_name_1	director	17.08.22	1
Alyssa Malyshova	375296172078	AlyssaMalyshova@mail.ru	department_name_4	employee	17.08.22	1
Rita Astafyeva	375294007451	RitaAstafyeva@mail.ru	department_name_2	manager	17.08.22	1
Clara Zaykova	375297467234	ClaraZaykova@mail.ru	department_name_2	manager	17.08.22	1
Eva Moshko	375297667494	EvaMoshko@mail.ru	department_name_4	employee	17.08.22	1
Ivan Semin	37529367621	IvanSemin@mail.ru	department_name_4	employee	17.08.22	1
Ian Astafyev	375294966103	IanAstafyev@mail.ru	department_name_3	employee	17.08.22	1
Ada Alymova	375292270876	AdaAlymova@mail.ru	department_name_2	manager	17.08.22	1
Maxim Mayorov	375298589471	MaximMayorov@mail.ru	department_name_2	manager	17.08.22	1
Veronika Sadovskaya	375291959498	VeronikaSadovskaya@mail.ru	department_name_3	employee	17.08.22	1
Pavel Moshko	375295241317	PavelMoshko@mail.ru	department_name_4	employee	17.08.22	1

Let's create view sal_cl.employee_dim_work that contains information about the amount of work of each employee per month (number of orders and total profit):

```

35 ┌─ CREATE OR REPLACE VIEW sal_cl.employee_dim_work AS
36   SELECT DISTINCT TRUNC(date_id, 'MM') AS month,
37     first_name||' '||last_name AS employee_name,
38     job_title AS position,
39     COUNT(order_id) AS orders_count_employee,
40     SUM(total_cost) AS employee_profit
41   FROM dw_data.order_fact ord
42     RIGHT JOIN dw_data.employee_dimension emp
43       ON (ord.employee_id=emp.employee_id)
44   GROUP BY TRUNC(date_id, 'MM'), first_name||' '||last_name, job_title
45   ORDER BY 1,2,3,4,5;

```

View SAL_CL.EMPLOYEE_DIM_WORK created.

Select from view sal_cl.employee_dim_work:

MONTH	EMPLOYEE_NAME	POSITION	ORDERS_COUNT_EMPLOYEE	EMPLOYEE_PROFIT
01.01.21	Ada Alymova	manager	7750	5793895
01.01.21	Adriana Karnitskaya	director	7750	5907859
01.01.21	Miron Buyanova	employee	7750	5810239
01.01.21	Miron Parfenov	employee	7750	5783234
01.01.21	Nikita Malyshев	employee	7750	5802457
01.01.21	Pavel Moshko	employee	7750	5814850
01.02.21	Ada Alymova	manager	7000	5272705
01.02.21	Adriana Karnitskaya	director	7000	5281627
01.02.21	Miron Buyanova	employee	7000	5241723
01.02.21	Miron Parfenov	employee	7000	5286688
01.02.21	Nikita Malyshev	employee	7000	5201214
01.02.21	Pavel Moshko	employee	7000	5247193
01.03.21	Ada Alymova	manager	7750	5822997
01.03.21	Adriana Karnitskaya	director	7750	5811832
01.03.21	Miron Buyanova	employee	7750	5796356
01.03.21	Miron Parfenov	employee	7750	5832271
01.03.21	Nikita Malyshev	employee	7750	5801947

Let's create sal_cl.restaurant_profit view that contains information about the profit for the month for each restaurant, the number of dishes sold and the total number of orders:

```

49 CREATE OR REPLACE VIEW sal_cl.restaurant_profit AS
50   SELECT DISTINCT TRUNC(date_id, 'MM') AS month,
51     country,
52     city,
53     SUM (total_cost) AS total_profit,
54     SUM (dish_amount) AS total_dish_amount,
55     COUNT (order_id) AS total_order_count
56   FROM dw_data.order_fact ord
57   RIGHT JOIN dw_data.dish_dimension dsh
58   ON (ord.dish_id=dsh.dish_id)
59   RIGHT JOIN dw_data.restaurant_dimension rst
60   ON (rst.restaurant_id=ord.restaurant_id)
61   GROUP BY TRUNC(date_id, 'MM'),country,city
62   ORDER BY 1,4,5,6;

```

Script Output x | Task completed in 0,05 seconds

View SAL_CL.RESTAURANT_PROFIT created.

Select from view sal_cl.restaurant_profit:

	MONTH	COUNTRY	CITY	TOTAL_PROFIT	TOTAL_DISH_AMOUNT	TOTAL_ORDER_COUNT
1	01.01.21	Ukraine	Kiev	6949272	200707	9300
2	01.01.21	Russia	Moscow	6963031	200022	9300
3	01.01.21	USA	New York	6973329	199000	9300
4	01.01.21	Belarus	Minsk	6992781	201567	9300
5	01.01.21	Poland	Warsaw	7034121	201540	9300
6	01.02.21	Russia	Moscow	6231573	179280	8400
7	01.02.21	USA	New York	6259544	179442	8400
8	01.02.21	Ukraine	Kiev	6324705	181011	8400
9	01.02.21	Poland	Warsaw	6352629	181906	8400
10	01.02.21	Belarus	Minsk	6362699	183313	8400
11	01.03.21	Belarus	Minsk	6942576	197324	9300
12	01.03.21	Ukraine	Kiev	6960043	199358	9300
13	01.03.21	Russia	Moscow	6971340	198941	9300
14	01.03.21	USA	New York	6974337	200391	9300
15	01.03.21	Poland	Warsaw	6991878	201533	9300

Let's create view sal_cl.top_dishes that contains information about the popularity of each dish in each restaurant for a month:

```

66 CREATE OR REPLACE VIEW sal_cl.top_dishes AS
67   SELECT DISTINCT TRUNC(date_id, 'MM') AS month,
68     country,
69     city,
70     dish_name,
71     SUM(dish_amount) AS total_dish_amount
72   FROM dw_data.order_fact ord
73   RIGHT JOIN dw_data.dish_dimension dsh
74   ON (ord.dish_id=dsh.dish_id)
75   RIGHT JOIN dw_data.restaurant_dimension rst
76   ON (rst.restaurant_id=ord.restaurant_id)
77   GROUP BY TRUNC(date_id, 'MM'),country,city, dish_name
78   ORDER BY 1,2,3,5 DESC;

```

Script Output x | Task completed in 0,064 seconds

View SAL_CL.TOP_DISHES created.

Select from view sal_cl.top_dishes:

The screenshot shows a database interface with a query window and a results window. The query window contains the SQL command: `SELECT * FROM sal_cl.top_dishes;`. The results window displays a table with 11 rows, showing data for dishes in Belarus, Poland, and Russia. The columns are MONTH, COUNTRY, CITY, DISH_NAME, and TOTAL_DISH_AMOUNT.

MONTH	COUNTRY	CITY	DISH_NAME	TOTAL_DISH_AMOUNT	
1	01.01.21	Belarus	Minsk	soup	93578
2	01.01.21	Belarus	Minsk	pizza	54888
3	01.01.21	Belarus	Minsk	chebupelli	21783
4	01.01.21	Belarus	Minsk	greek salad	16322
5	01.01.21	Belarus	Minsk	pasta	14996
6	01.01.21	Poland	Warsaw	soup	92888
7	01.01.21	Poland	Warsaw	pizza	54838
8	01.01.21	Poland	Warsaw	chebupelli	22181
9	01.01.21	Poland	Warsaw	greek salad	16287
10	01.01.21	Poland	Warsaw	pasta	15346
11	01.01.21	Russia	Moscow	soup	91575

Let's create view sal_cl.restaurant_visits that contains information about the attendance of each restaurant for a month:

The screenshot shows a database interface with a script window and a results window. The script window contains the SQL command to create the view: `CREATE OR REPLACE VIEW sal_cl.restaurant_visits AS SELECT DISTINCT TRUNC(date_id, 'MM') AS month, country, city, COUNT(order_id) AS total_visits_count FROM dw_data.order_fact ord RIGHT JOIN dw_data.restaurant_dimension rst ON (rst.restaurant_id=ord.restaurant_id) WHERE delivery='N' GROUP BY TRUNC(date_id, 'MM'), country, city ORDER BY 1,4 DESC;`. The results window shows a message: "View SAL_CL.RESTAURANT_VISITS created."

Select from view sal_cl.restaurant_:

The screenshot shows a database interface with a query window and a results window. The query window contains the SQL command: `SELECT * FROM sal_cl.restaurant_;`. The results window displays a table with 11 rows, showing data for restaurants in Poland, Ukraine, Belarus, USA, and Russia. The columns are MONTH, COUNTRY, CITY, and TOTAL_VISITS_COUNT.

MONTH	COUNTRY	CITY	TOTAL_VISITS_COUNT	
1	01.01.21	Poland	Warsaw	6161
2	01.01.21	Ukraine	Kiev	6141
3	01.01.21	Belarus	Minsk	6137
4	01.01.21	USA	New York	6127
5	01.01.21	Russia	Moscow	6105
6	01.02.21	USA	New York	5549
7	01.02.21	Belarus	Minsk	5536
8	01.02.21	Poland	Warsaw	5509
9	01.02.21	Ukraine	Kiev	5487
10	01.02.21	Russia	Moscow	5437
11	01.03.21	Poland	Warsaw	6162

Let's create view sal_cl.w_client_dimension that stores all information about clients:

```

33 | CREATE OR REPLACE VIEW sal_cl.w_client_dimension
34 | AS SELECT * FROM dw_data.client_dimension;
35 |

```

Script Output X
Task completed in 0,037 seconds

Table SAL_CL.ORDER_FACT created.

View SAL_CL.W_DISH_DIMENSION created.

View SAL_CL.W_CLIENT_DIMENSION created.

Select from view sal_cl.w_client_dimension:

```

37 | SELECT * FROM sal_cl.w_client_dimension;
38 |

```

Query Result X
SQL | All Rows Fetched: 5 in 0,03 seconds

	CLIENT_ID	FIRST_NAME	LAST_NAME	PHONE	EMAIL	STREET	COUNTRY	CITY	STATUS
1	1 Adriana	Karnitskaya	375294217925	AdrianaKarnitskaya@mail.ru	Yesenin	Belarus	Minsk	Y	
2	2 Ian	Astafyev	375295321165	IanAstafyev@mail.ru	Champs Elysees	France	Paris	Y	
3	3 Ada	Alymova	375292376362	AdaAlymova@mail.ru	Nikolskaya	Russia	Moscow	Y	
4	4 Pavel	Moshko	375291330060	PavelMoshko@mail.ru	Antoine Dansaert	Belgium	Brussels	Y	
5	5 Alice	Lysenko	375294487084	AliceLysenko@mail.ru	Fifth Avenue	USA	New York	Y	

Let's create view sal_cl.w_dish_dimension that stores all information about dishes:

```

27 | CREATE OR REPLACE VIEW sal_cl.w_dish_dimension
28 | AS SELECT * FROM dw_data.dish_dimension;
29 |

```

Script Output X
Task completed in 0,05 seconds

Table SAL_CL.ORDER_FACT created.

View SAL_CL.W_DISH_DIMENSION created.

Select from view sal_cl.w_dish_dimension:

```

29 |
30 | SELECT * FROM sal_cl.w_dish_dimension;
31 |

```

Query Result X
SQL | All Rows Fetched: 5 in 0,027 seconds

	DISH_ID	DISH_NAME	DISH_CATEGORY	PRICE	COMPOSITION	WEIGHT	STATUS
1	1 soup	hot	15	soup ingredients	915	Y	
2	2 pizza	hot	25	pizza ingredients	486	Y	
3	3 pasta	hot	87	pasta ingredients	959	Y	
4	4 chebupelli	hot	61	chebupelli ingredients	512	Y	
5	5 greek salad appetizer		82	greek salad ingredients	797	Y	

Let's create view sal_cl.w_restaurant_dimension that stores all information about restaurants:

```

41 | CREATE OR REPLACE VIEW sal_cl.w_restaurant_dimension
42 | AS SELECT * FROM dw_data.restaurant_dimension;
43 |
44 |
45 | Script Output X
46 | Task completed in 0,045 seconds
47 |
48 | View SAL_CL.W_RESTAURANT_DIMENSION created.

```

Select from view sal_cl.w_restaurant_dimension:

```

45 | SELECT * FROM sal_cl.w_restaurant_dimension;
46 |
47 | Query Result X
48 | SQL | All Rows Fetched: 5 in 0,029 seconds
49 |
50 | RESTAURANT_ID | PHONE | EMAIL | ADDRESS | COUNTRY | CITY | BUILDING | APARTMENT | STATUS
51 | 1 | 375295841669 | 5@mail.ru | Gurchevskaya | Poland | Warsaw | 46 | 230 | Y
52 | 2 | 375292356572 | 3@mail.ru | Fifth Avenue | USA | New York | 30 | 189 | Y
53 | 3 | 375299013465 | 2@mail.ru | Nikolskaya | Russia | Moscow | 32 | 165 | Y
54 | 4 | 375295176493 | 4@mail.ru | Vladimirskaya | Ukraine | Kiev | 36 | 185 | Y
55 | 5 | 375294308850 | 1@mail.ru | Yesenin | Belarus | Minsk | 81 | 132 | Y

```

Let's create view sal_cl.w_employee_dimension that stores all information about employees:

```

47 | CREATE OR REPLACE VIEW sal_cl.w_employee_dimension
48 | AS SELECT * FROM dw_data.employee_dimension;
49 |
50 |
51 | Script Output X
52 | Task completed in 0,053 seconds
53 |
54 | View SAL_CL.W_EMPLOYEE_DIMENSION created.

```

Select from view sal_cl.w_employee_dimension:

```

51 | SELECT * FROM sal_cl.w_employee_dimension;
52 |
53 | Query Result X
54 | SQL | All Rows Fetched: 25 in 0,023 seconds
55 |
56 | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | PHONE | EMAIL | DEPARTMENT | JOB_TITLE | ADDRESS | COUNTRY | CR
57 | 1 | Adriana | Karnitskaya | 375295542493 | AdrianaKarnitskaya@mail.ru | department_name_1 | director | Yesenin | Belarus | Minsk
58 | 2 | Alyssa | Malyshева | 375296172078 | AlyssaMalysheva@mail.ru | department_name_4 | employee | Vladimirkaya | Ukraine | Kiev
59 | 3 | Rita | Astafyeva | 375294007451 | RitaAstafyeva@mail.ru | department_name_2 | manager | K Kaiser-Friedrich | Germany | Berlin
60 | 4 | Clara | Zaykova | 375297467234 | ClaraZaykova@mail.ru | department_name_2 | manager | Abby | England | London
61 | 5 | Eva | Moshko | 375297667494 | EvaMoshko@mail.ru | department_name_4 | employee | Aloyas | Latvia | Riga
62 | 6 | Ivan | Semin | 375293676621 | IvanSemin@mail.ru | department_name_4 | employee | Via del Corso | Belgium | Brussels
63 | 7 | Ian | Astafyev | 375294996103 | IanAstafyev@mail.ru | department_name_3 | employee | Champs Elysees | France | Paris
64 | 8 | Ada | Alymova | 375292270076 | AdaAlymova@mail.ru | department_name_2 | manager | Nikolskaya | Russia | Moscow
65 | 9 | Maxim | Mayorov | 375290589471 | MaximMayorov@mail.ru | department_name_2 | manager | D Dunkri | England | London
66 | 10 | Veronika | Sadovskaya | 3752919559498 | VeronikaSadovskaya@mail.ru | department_name_3 | employee | D Dunkri | Estonia | Tallinn
67 | 11 | Pavel | Moshko | 375295241317 | PavelMoshko@mail.ru | department_name_4 | employee | Antoine Dansaert | Belgium | Brussels
68 | 12 | Alexandra | Etkina | 375293183125 | AlexandraEtkina@mail.ru | department_name_1 | director | Graben | Austria | Vienna
69 | 13 | Nikita | Malyshев | 375292197332 | NikitaMalyshev@mail.ru | department_name_4 | employee | G Galve | Latvia | Riga

```

Let's create view sal_cl.w_payment_method_dimension that stores all information about payment methods:

```

53 | CREATE OR REPLACE VIEW sal_cl.w_payment_method_dimension
54 | AS SELECT * FROM dw_data.payment_method_dimension;
55 |
56 |
57 | Script Output X
58 | Task completed in 0,042 seconds
59 |
60 | View SAL_CL.W_PAYMENT_METHOD_DIMENSION created.

```

Select from view sal_cl.w_payment_method_dimension:

```
60 | SELECT * FROM sal_cl.w_payment_method_dimension;
61 |
62 |  
Query Result x
63 | 1 bank card
64 | 2 cash
```

PAYMENT_METHOD_ID	PAYMENT_METHOD_NAME	STATUS
1	bank card	Y
2	cash	Y

Let's create view sal_cl.w_gen_period that stores all information about gen periods:

```
59 | CREATE OR REPLACE VIEW sal_cl.w_dim_gen_period
60 | AS SELECT * FROM dw_data.dim_gen_period;
61 |
62 |  
Script Output x
63 | Task completed in 0,045 seconds
64 |
65 | View SAL_CL.W_DIM_GEN_PERIOD created.
```

Select from view sal_cl.w_gen_period:

```
68 | SELECT * FROM sal_cl.w_dim_gen_period;
69 |
70 |  
Query Result x
71 | All Rows Fetched: 6 in 0,024 seconds
72 |  
PERIOD_ID VALID_FROM VALID_TO PROMOTION_NAME PROMOTION_PERCENT DESCRIPTION
73 | 1 01.04.21 30.04.21 promotion_name_4 5 decription_4
74 | 2 01.02.21 28.02.21 promotion_name_2 15 decription_2
75 | 3 01.06.21 31.12.21 promotion_name_6 20 decription_6
76 | 4 01.03.21 31.03.21 promotion_name_3 10 decription_3
77 | 5 01.05.21 31.05.21 promotion_name_5 20 decription_5
78 | 6 01.01.21 31.01.21 promotion_name_1 25 decription_1
```

Let's create table sal_cl.order_fact that stores all information about orders:

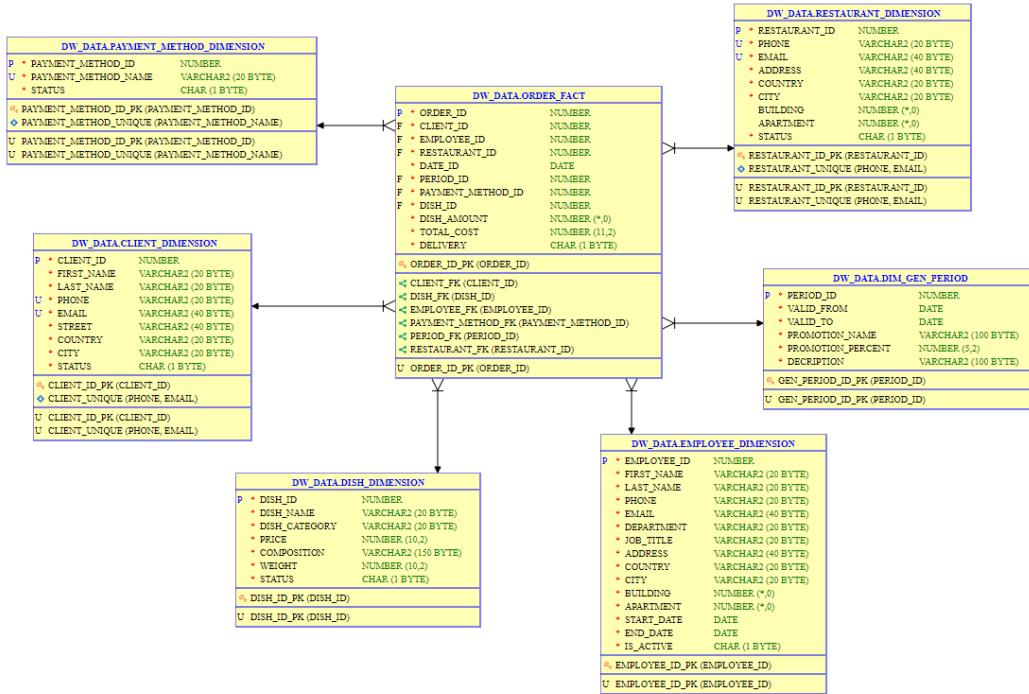
```
3 | CREATE TABLE sal_cl.order_fact(
4 |   order_id NUMBER,
5 |   client_id NUMBER NOT NULL,
6 |   employee_id NUMBER NOT NULL,
7 |   restaurant_id NUMBER NOT NULL,
8 |   date_id DATE NOT NULL,
9 |   period_id NUMBER NOT NULL,
10 |  payment_method_id NUMBER NOT NULL,
11 |  dish_id NUMBER NOT NULL,
12 |  dish_amount INT NOT NULL,
13 |  total_cost DECIMAL (11,2) NOT NULL,
14 |  delivery CHAR(1) NOT NULL CHECK (delivery IN ('N','Y'))
15 |  PARTITION BY RANGE (date_id)
16 |  --subpartition by hash(client_id) subpartitions 4
17 |  (
18 |    PARTITION quarter_1 VALUES LESS THAN(to_date('01.04.2021','DD.MM.YYYY')),
19 |    PARTITION quarter_2 VALUES LESS THAN(to_date('01.07.2021','DD.MM.YYYY')),
20 |    PARTITION quarter_3 VALUES LESS THAN(to_date('01.10.2021','DD.MM.YYYY')),
21 |    PARTITION quarter_4 VALUES LESS THAN(to_date('01.01.2022','DD.MM.YYYY'))
22 |  );
23 |
24 |  
Script Output x
25 | Task completed in 0,066 seconds
26 |
27 | Table SAL_CL.ORDER_FACT created.
```

Let's create package pkg_etl_sal_level and procedure load_sal_order_fact to load data to table sal_cl.order_fact:

Select from table sal_cl.order_fact:

	ORDER_ID	CLIENT_ID	EMPLOYEE_ID	RESTAURANT_ID	DATE_ID	PERIOD_ID	PAYMENT_METHOD_ID	DISH_ID	DISH_AMOUNT	TOTAL_COST	DELIVERY
1	306223	5	11		231.03.21	4		1	4	23	1438 N
2	306224	5	21		431.03.21	4		1	5	6	498 N
3	306286	5	1		328.01.21	6		1	2	4	100 N
4	306287	5	8		228.01.21	6		1	1	96	1447 N
5	306288	5	13		428.01.21	6		1	5	2	179 N
6	306289	5	13		128.01.21	6		1	3	12	1113 N
7	306290	5	1		330.01.21	6		2	5	4	374 N
8	306291	5	13		531.01.21	6		1	4	23	1419 N
9	306292	5	14		430.01.21	6		2	4	5	344 N
10	306293	5	21		130.01.21	6		2	5	0	35 N
11	306294	5	13		431.01.21	6		1	5	0	35 N
12	306295	5	11		529.01.21	6		2	5	10	845 N
13	306296	5	8		329.01.21	6		2	2	52	1313 N
14	306297	5	11		229.01.21	6		2	5	18	1482 N

Star scheme:



Statistics for Star Scheme:

The statistics which is required by cost based optimizer was gathered using DBMS_STATS package. Result of procedure execution for schema is presented below:

```
EXECUTE DBMS_STATS.GATHER_SCHEMA_STATS (ownname => 'DW_DATA');

Script Output | Task completed in 2,049 seconds
PL/SQL procedure successfully completed.
```

Task 2 - Performance comparison

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

1) Lab 2, sa level, advancing grouping

```

1 WITH CTE_FN AS (
2   SELECT /*+ gather_plan_statistics */
3     dish_name,
4     DECODE (GROUPING (TRUNC(order_date, 'YYYY')),1,'All years',TRUNC(order_date, 'YYYY')) AS year_ord,
5     DECODE (GROUPING (TRUNC(order_date, 'MM')),1,'All months',TRUNC(order_date, 'MM')) AS month_ord,
6     COUNT(*) AS dish_count
7   FROM sa_orders.sa_t_transaction
8   GROUP BY GROUPING SETS ((dish_name),
9     (dish_name, TRUNC(order_date, 'YYYY')),
10    (dish_name,TRUNC(order_date, 'YYYY'),TRUNC(order_date, 'MM')))
11   ORDER BY 1,2,3,4)
12  SELECT dish_name,year_ord, month_ord, dish_count FROM CTE_FN;

```

Query Result | 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


```

1 WITH CTE_FN AS (
2   SELECT /*+ gather_plan_statistics */
3     dish_name,
4     DECODE (GROUPING (TRUNC(order_date, 'YYYY')),1,'All years',TRUNC(order_date, 'YYYY')) AS year_ord,
5     DECODE (GROUPING (TRUNC(order_date, 'MM')),1,'All months',TRUNC(order_date, 'MM')) AS month_ord,
6     COUNT(*) AS dish_count
7   FROM sa_orders.sa_t_transaction
8   GROUP BY GROUPING SETS ((dish_name),
9     (dish_name, TRUNC(order_date, 'YYYY')),
10    (dish_name,TRUNC(order_date, 'YYYY'),TRUNC(order_date, 'MM')))
11   ORDER BY 1,2,3,4)
12  SELECT dish_name,year_ord, month_ord, dish_count FROM CTE_FN;

```

Query Result | Explain Plan | 0,11seconds

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:

```

14 WITH CTE_FN AS
15   ( SELECT dish_name,
16     TRUNC(order_date, 'YYYY') AS year,
17     TO_CHAR(order_date, 'MM') AS month,
18     'MONTH' period,
19     COUNT(*) AS dish_count
20   FROM sa_orders.sa_t_transaction
21   GROUP BY dish_name, TRUNC(order_date, 'YYYY'),TO_CHAR(order_date, 'MM')
22   ORDER BY dish_name, year, month)
23  SELECT dish_name, year, month, period, dish_count
24  FROM CTE_FN
25  MODEL
26  DETERMINANT BY (dish_name)

```

Query Result | 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

```

14 WITH CTE_FN AS
15   ( SELECT dish_name,
16     TRUNC(order_date, 'YYYY') AS year,
17     TO_CHAR(order_date, 'MM') AS month,
18     'MONTH' period,
19     COUNT(*) AS dish_count
20   FROM sa_orders.sa_t_transaction
21   GROUP BY dish_name, TRUNC(order_date, 'YYYY'), TO_CHAR(order_date, 'MM')
22   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_QFD9...	(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			1291	3
TABLE ACCESS	SAL_CL.null		1291	3
BUFFER	SYS.SYS_TEMP_0...	FULL	1291	3
HASH	SORT		1	4
VIEW	UNIQUE		1	4
	SAL_CL.null		1291	3
	CSC_CSC_TRANS_A	FULL	1291	2

3) Lab 11, dw level, star scheme:

```

36 WITH CTE_FN AS (
37   SELECT /*+ gather_plan_statistics */
38     dish_name,
39     DECODE (GROUPING (TRUNC(date_id, 'YYYY')),1,'All years',TRUNC(date_id, 'YYYY')) AS year_ord,
40     DECODE (GROUPING (TRUNC(date_id, 'MM')),1,'All months',TRUNC(date_id, 'MM')) AS month_ord,
41     COUNT(*) AS dish_count
42   FROM dw_data.order_fact fct
43   LEFT JOIN dw_data.dish_dimension dsh
44     ON fct.dish_id=dsh.dish_id
45   GROUP BY GROUPING SETS ((dish_name),
46     (dish_name, TRUNC(date_id, 'YYYY')),
47     (dish_name,TRUNC(date_id, 'YYYY'),TRUNC(date_id, 'MM')))
48   ORDER BY 1,2,3,4

```

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

```

36 WITH CTE_FN AS (
37   SELECT /*+ gather_plan_statistics */
38     dish_name,
39     DECODE (GROUPING (TRUNC(date_id, 'YYYY')),1,'All years',TRUNC(date_id, 'YYYY')) AS year_ord,
40     DECODE (GROUPING (TRUNC(date_id, 'MM')),1,'All months',TRUNC(date_id, 'MM')) AS month_ord,
41     COUNT(*) AS dish_count
42   FROM dw_data.order_fact fct
43   LEFT JOIN dw_data.dish_dimension dsh
44     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(+)	DW_DATA.DISH...	FULL	5	4
TABLE ACCESS			547500	1 4 6 1033
PARTITION RANGE	ALL		547500	1 4 7 1033
PARTITION HASH			547500	1 16 7 1033
TABLE ACCESS	DW_DATA.ORDE...	FULL	547500	

Nº	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