

Департамент образования и науки города Москвы
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»
Институт цифрового образования
Департамент информатики управления и технологий

Нургалеева Гузель Рустэмовна БД-241м

**Лабораторная работа 2. Моделирование данных и SQL для Data
Engineering**

Вариант задания: 18

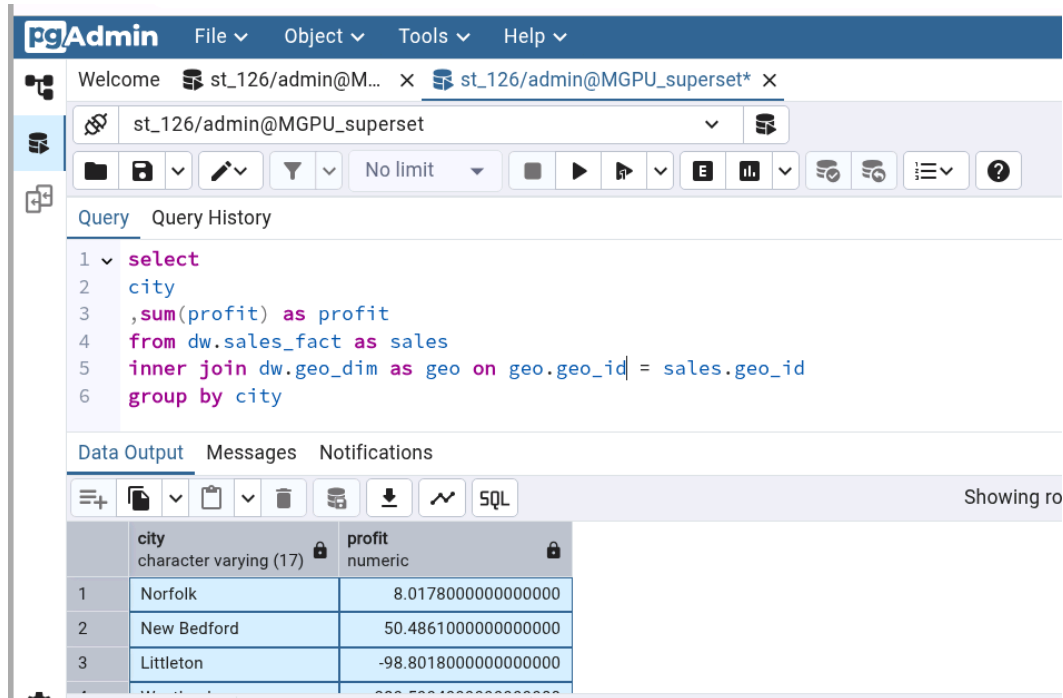
Направление подготовки/специальность
38.04.05 - Бизнес-информатика
Бизнес-аналитика и большие данные
(очная форма обучения)

Москва

2024

Задание 1. Определить прибыль по городам

```
select
city
,sum(profit) as profit
from dw.sales_fact as sales
inner join dw.geo_dim as geo on geo.geo_id = sales.geo_id
group by city
```



The screenshot shows the pgAdmin interface with a SQL query entered in the query editor. The query is the same as the one in the previous block. Below the query editor, the 'Data Output' tab is active, displaying the results of the query in a table format. The table has two columns: 'city' (character varying (17)) and 'profit' (numeric). The results show three rows of data.

	city	profit
1	Norfolk	8.017800000000000
2	New Bedford	50.486100000000000
3	Littleton	-98.801800000000000

Комментарии к запросам:

в схеме dw есть фактовая таблица с данными по прибыли – dw.sales_fact. При этом в ней нет данных по наименованию города, но есть связующее поле со справочником geo_dim, из которого можно получить наименование города.

Проверка:

- 1) Можно проверить выгрузив несколько строк и сложив значения в эксель.
Например, взять только Нью Йорк и сравнить суммы в двух приведенных ниже запросах:

```
select
geo_id,
profit
from dw.sales_fact as sales
where geo_id in (select geo_id from dw.geo_dim where city = 'New York City')

select
city
,sum(profit) as profit
from dw.sales_fact as sales
inner join dw.geo_dim as geo on geo.geo_id = sales.geo_id
where sales.geo_id in (select geo_id from dw.geo_dim where city = 'New York City')
group by city
```

Задание 2. Создать таблицу по выручке менеджеров

Вариант 1

```
drop table if exists dw.sales_by_managers;
create table dw.sales_by_managers
(
    "id"            serial NOT NULL,
    manager         varchar(17) NOT NULL,
    order_date      date NOT NULL,
    order_id        varchar(25) NOT NULL,
    geo_id          integer NOT NULL,
    prod_id         integer NOT NULL,
    sales           numeric(9,4) NOT NULL,
    profit          numeric(21,16) NOT NULL,
    quantity        int4 NOT NULL,
    discount        numeric(4,2) NOT NULL,
    CONSTRAINT PK_sales_by_managers PRIMARY KEY ("id")
);
truncate table dw.sales_by_managers;
insert into dw.sales_by_managers
select
100 + row_number() over() as "id"
,ppl.person as manager
,o.order_date
,o.order_id
,g.geo_id
,p.prod_id
,o.sales
,o.profit
,o.quantity
,o.discount
from stg.orders as o
inner join public.people as ppl on ppl.region = o.region
inner join dw.product_dim p on o.product_name = p.product_name and
o.segment=p.segment and o.subcategory=p.sub_category and o.category=p.category and
o.product_id=p.product_id
inner join dw.geo_dim g on o.postal_code = g.postal_code and g.country=o.country and
g.city = o.city and o.state = g.state
```

Query Query History

```

1 SELECT id, manager, order_date, order_id, geo_id, prod_id, sales, profit, quantity, discount
2 FROM dw.sales_by_managers;

```

Data Output Messages Notifications

	id [PK] integer	manager character varying (17)	order_date date	order_id character varying (25)	geo_id integer	prod_id integer	sales numeric (9,4)	profit numeric (21,16)
1	101	Chuck Magee	2018-12-26	CA-2018-155166	225	101	212.9400	25.5528000000000000
2	102	Anna Andreadi	2019-12-28	CA-2019-101322	313	101	340.7040	-34.0704000000000000
3	103	Chuck Magee	2017-04-16	CA-2017-142734	503	101	127.7640	2.8392000000000000
4	104	Cassandra Brandow	2018-04-15	US-2018-123750	257	102	189.5880	-145.3508000000000000

Вариант 2.

```
drop table if exists dw.managers_dim;
create table dw.managers_dim
(
manager_id    serial NOT NULL,
manager       varchar(17) NOT NULL,
region        varchar(25) NOT NULL,
CONSTRAINT PK_managers_dim PRIMARY KEY (manager_id)
```

```
);
truncate table dw.managers_dim;
insert into dw.managers_dim
select
100+row_number() over() as mng_id
, person as manager
, region
from (select distinct person, region from public.people) a;

select * from dw.managers_dim;
```

Welcome st_126/admin@MGPU_superset x st_126/admin@M... x st_126/admin@M... x

st_126/admin@MGPU_superset

Query Query History

```
14 ,region
15 from (select distinct person, region from public.people) a;
16
17 select * from dw.managers_dim;
18
```

Data Output Messages Notifications

Showing row:

	manager_id [PK] integer	manager character varying (17)	region character varying (25)
1	101	Kelly Williams	Central
2	102	Chuck Magee	East
3	103	Cassandra Brandow	South
4	104	Anna Andreadi	West

```
drop table if exists dw.profit_by_managers;
create table dw.profit_by_managers
(
profit_id serial NOT NULL,
manager_id integer NOT NULL,
order_date date NOT NULL,
profit NUMERIC(21,16) NOT NULL,
CONSTRAINT PK_profit_by_managers PRIMARY KEY (profit_id)
);
truncate table dw.profit_by_managers;
insert into dw.profit_by_managers
(select
100+row_number() over() as profit_id
,manager_id
,order_date
,sum(profit) as profit
from stg.orders as o
inner join dw.managers_dim as mng on mng.region = o.region
group by 2,3
);
select * from dw.profit_by_managers;
```

The screenshot shows a SQL query editor interface. The query is as follows:

```

16 ,sum(profit) as profit
17 from stg.orders as o
18 inner join dw.managers_dim as mng on mng.region = o.region
19 group by 2,3
20 );
21 select * from dw.profit_by_managers;
22

```

Below the query, the 'Data Output' tab is active, showing a table with 5 columns: profit_id, manager_id, order_date, and profit. The table contains 3 rows of data.

	profit_id [PK] integer	manager_id integer	order_date date	profit numeric (21,16)
1	101	102	2018-09-06	101.44499999999999000
2	102	104	2019-08-17	902.0741999999999900
3	103	103	2018-12-18	-349.3705000000000000

Комментарии к запросам:

в схеме dw есть фактовая таблица с данными по прибыли – dw.sales_fact, но в ней нет данных по менеджерам, также в схеме dw нет справочника по менеджерам, который можно было бы связать с таблицей dw.sales_fact .

Поэтому для формирования таблицы обращалась к уровню staging (stg.orders) для получения данных основных данных по заказам и прибыли, и к уровню

При этом в ней нет данных по наименованию города, но raw (public.people) для получения данных по менеджерам.

В первом варианте создана таблица, которая содержит имена менеджеров. Также в этой таблице помимо Profit есть и другие поля, которые могут потребоваться для расчетов показателей по менеджерам.

Во втором варианте создан справочник по менеджерам и фактовая таблица с прибылью по менеджерам, которая содержит manager_id и связывается со справочником по менеджерам по manager_id. В таблицу также внесла поле с датой, чтобы можно было оценивать показатели по менеджерам за год, месяц, т.д.

Задание 3. Найти среднее количество товаров в заказе

```

select
avg(sum_) as avg_
from
  (SELECT order_id,
   sum(quantity) as sum_
   FROM dw.sales_fact
   group by 1) step1;

```

18	select
19	avg(sum_) as avg_
20	from
21	(SELECT order_id,
22	sum(quantity) as sum_
23	FROM dw.sales_fact
24	group by 1) step1;

Data Output	Messages	Notifications
-------------	----------	---------------

+	📄	▼	📋	▼	🗑️	📦	⬇️	📈	SQL
---	---	---	---	---	----	---	----	---	-----

	avg_	numeric	🔒
1	7.5609902176083051		

Комментарии к запросам:

По логике запроса: сначала подсчитывается сумма товаров в одном заказе, потом считается средняя сумма по всем заказам.

Проверка:

Проверить вручную, выгрузив данные по нескольким заказам

1 шаг – сумма товаров в нескольких заказах

```
SELECT
order_id,
sum(quantity) as sum_
FROM dw.sales_fact
group by 1
order by 1 ASC
limit 10;
```

2 шаг – посчитать среднее в экселе и сравнить с результатом этого запроса:

```
select
avg(sum_)
from
(SELECT
order_id,
sum(quantity) as sum_
FROM dw.sales_fact
group by 1
order by 1 ASC
limit 10) abc;
```