

ONLINE SHOP SALES PROJECT

Преобразование даты (Transaction_Date) в формат даты, соответствующий внутреннему формату базы данных

```
update online_shop_sales  
set Transaction_Date = STR_TO_DATE(Transaction_Date, '%d.%m.%Y');
```

Изменение типа столбца 'Transaction_Date' на формат даты

```
alter table online_shop_sales  
modify column Transaction_Date DATE;
```

Замена запятых на точки в столбце 'Avg_Price' (аналогично в столбцах 'Delivery_Charges' и 'Online_Spend'), так как для корректной обработки данных в базе данных и выполнения математических операций требуется использование точки

```
update online_shop_sales  
set Avg_Price = REPLACE(Avg_Price, ',', '.');
```

Изменение типа столбца 'Avg_Price' на DOUBLE

```
alter table online_shop_sales  
modify column Avg_Price double;
```

Типы данных для всех столбцов:

```
describe online_shop_sales
```

	Field	Type	Null	Key	Default	Extra
►	CustomerID	int	YES		NULL	
	Gender	text	YES		NULL	
	Location	text	YES		NULL	
	Tenure_Months	int	YES		NULL	
	Transaction_ID	int	YES		NULL	
	Transaction_Date	date	YES		NULL	
	Product_SKU	text	YES		NULL	
	Product_Description	text	YES		NULL	
	Product_Category	text	YES		NULL	
	Quantity	int	YES		NULL	
	Avg_Price	double	YES		NULL	
	Delivery_Charges	double	YES		NULL	
	Coupon_Status	text	YES		NULL	
	Offline_Spend	int	YES		NULL	
	Online_Spend	double	YES		NULL	
	Month	int	YES		NULL	
	Coupon_Code	text	YES		NULL	
	Discount_pct	int	YES		NULL	

Удаление дублирующего столбца 'Date'

```
alter table online_shop_sales  
drop column date;
```

Удаление столбца 'GST'

```
alter table online_shop_sales  
drop column gst;
```

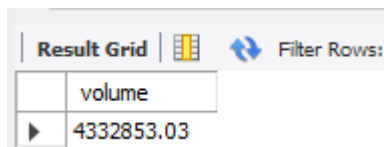
Удаление записей с отсутствующим 'CustomerID'

```
delete from online_shop_sales
where customerid is null;
```

Анализ продаж:

Расчет объема продаж за весь отчетный период с учетом скидок

```
select round(sum(
case
when discount_pct=10 and coupon_status= "used" then quantity * avg_price * (1 - 10/100)
when discount_pct=20 and coupon_status= "used" then quantity * avg_price * (1 - 20/100)
when discount_pct=30 and coupon_status= "used" then quantity * avg_price * (1 - 30/100)
ELSE quantity * avg_price
end),2) as volume
from online_shop_sales;
```



Result Grid	Filter Rows:
volume	
4332853.03	

Объяснение запроса:

- `select round(sum(...),2) as volume`: выбирает округленное до двух знаков после запятой значение общего объема продаж.
- `case when discount_pct=... and coupon_status = 'used' then then quantity * avg_price * (1 - discount_pct /100)`: проверяет, была ли использована скидка, и, если да, рассчитывает итоговую стоимость с учетом скидки.
- `ELSE quantity * avg_price`: если скидка не была использована, рассчитывает полную стоимость без скидки.
- `From online_shop_sales`: указывает таблицу, из которой берутся данные.

Расчет объема продаж с учетом скидок и процентного прироста

```
with monthly_sales as (
select month, round(sum(
case
when discount_pct=10 and coupon_status= "used" then quantity * avg_price * (1 - 10/100)
when discount_pct=20 and coupon_status= "used" then quantity * avg_price * (1 - 20/100)
when discount_pct=30 and coupon_status= "used" then quantity * avg_price * (1 - 30/100)
ELSE quantity * avg_price
end),2) as volume
from online_shop_sales
group by month
)
select month, volume,
round(((volume - lag(volume,1) over (order by month)) / lag(volume,1)
over (order by month) * 100),2) as increase_percentage
from monthly_sales
order by month
```

Result Grid			
Filter Rows:			
	month	volume	increase_percentage
▶	1	388083.01	NULL
	2	288900.32	-25.56
	3	311720.17	7.9
	4	383403.77	23
	5	285067.47	-25.65
	6	286912.58	0.65
	7	357501.31	24.6
	8	371335.86	3.87
	9	319809.19	-13.88
	10	394562.32	23.37
	11	472876.08	19.85
	12	472680.95	-0.04

Объяснение запроса:

- with monthly_sales as: создаёт временную таблицу monthly_sales, которая содержит месяцы и соответствующие объемы продаж.
- select month, round(sum(...),2) as volume: вычисляет общий объем продаж за ноябрь и декабрь, округленный до двух знаков после запятой.
- case...end: рассчитывает итоговую стоимость с учетом скидки, если купон был использован.
- select month, volume, round(((volume - lag(volume,1) over (order by month)) / lag(volume,1) over (order by month) * 100),2) as increase_percentage : рассчитывает процентный прирост объема продаж по сравнению с предыдущим месяцем, округленный до двух знаков после запятой.
- order by month: сортирует результаты по месяцам.

Анализ заказов:

Подсчет общего количества заказов за отчетный период

```
select count(distinct transaction_id) as total_orders
from online_shop_sales;
```

Result Grid	
Filter Rows:	
	total_orders
▶	25000

Подсчет заказов по месяцам с расчетом процентного прироста

```
select month, count(distinct transaction_id) as total_orders,
(count(transaction_id) - lag(count(transaction_id),1)
over (order by month)) / lag(count(transaction_id),1)
over (order by month) * 100 as increase_percentage
from online_shop_sales
group by month
order by month
```

Result Grid			
Filter Rows:			
	month	total_orders	increase_percentage
▶	1	2096	NULL
	2	1658	-20.90
	3	1982	19.54
	4	1806	-8.88
	5	2019	11.79
	6	1934	-4.21
	7	2079	7.50
	8	2411	15.97
	9	1929	-19.99
	10	2123	10.06
	11	2279	7.35
	12	2684	17.77

Объяснение запроса:

- select month, count(transaction_id) as total_orders: извлекает месяц и подсчитывает общее количество заказов (transaction_id) для каждого месяца.
- (count(transaction_id) - lag(count(transaction_id),1) over (order by month)) / lag(count(transaction_id),1) over (order by month) * 100 as increase_percentage: вычисляет процентный прирост заказов между текущим и предыдущим месяцем, используя оконную функцию LAG для получения предыдущего значения.
- from online_shop_sales: указывает таблицу, из которой берутся данные.
- group by month: группирует данные по месяцам.
- order by month: сортирует результаты по месяцам в порядке возрастания.

Анализ проданных товаров:

Подсчет общего количества проданных товаров за отчетный период

```
select sum(quantity) as items_sold
from online_shop_sales;
```

Result Grid	
Filter Rows:	
	items_sold
▶	236367

Подсчет проданных товаров по месяцам с расчетом процентного прироста

```
select month, sum(quantity) as items_sold,
(sum(quantity) - lag(sum(quantity),1)
over (order by month)) / lag(sum(quantity),1)
over (order by month) * 100 as increase_percentage
from online_shop_sales
group by month
order by month
```

Result Grid			
Filter Rows:			
	month	items_sold	increase_percentage
▶	1	19987	NULL
	2	16410	-17.8966
	3	20543	25.1859
	4	24922	21.3163
	5	18519	-25.6922
	6	21482	15.9998
	7	22613	5.2649
	8	27379	21.0764
	9	19329	-29.4021
	10	17400	-9.9798
	11	15116	-13.1264
	12	12667	-16.2014

Объяснение запроса:

- `select month, sum(quantity) as items_sold`: извлекает месяц и суммирует количество проданных единиц (`quantity`) для каждого месяца.
- `(sum(quantity) - lag(sum(quantity),1) over (order by month)) / lag(sum(quantity),1) over (order by month) * 100 as increase_percentage`: вычисляет процентный прирост количества проданных товаров между текущим и предыдущим месяцем, используя оконную функцию `LAG` для получения предыдущего значения.
- `from online_shop_sales`: указывает таблицу, из которой берутся данные.
- `group by month`: группирует данные по месяцам.
- `order by month`: сортирует результаты по месяцам в порядке возрастания.

Визуализация:

Ежедневный тренд заказов

```
select dayname(transaction_date) as order_day, count(transaction_id) as total_orders
from online_shop_sales
group by dayname(transaction_date), dayofweek(transaction_date)
order by dayofweek(transaction_date)
```

Result Grid		
Filter Rows:		
	order_day	total_orders
▶	Sunday	8448
	Monday	4429
	Tuesday	4578
	Wednesday	8830
	Thursday	8924
	Friday	9202
	Saturday	8113

Объяснение запроса:

- `select dayname(transaction_date) as order_day, count(transaction_id) as total_orders`: извлекает название дня недели из даты транзакции и подсчитывает общее количество заказов (`transaction_id`) для каждого дня недели.
- `from online_shop_sales`: указывает таблицу, из которой берутся данные.
- `group by dayname(transaction_date), dayofweek(transaction_date)`: группирует данные по названиям дней недели и порядку дней недели, чтобы обеспечить правильную сортировку.

- `order by dayofweek(transaction_date)`: сортирует результаты по порядку дней недели, начиная с воскресенья (1) до субботы (7).

Ежемесячный тренд заказов

```
select monthname(transaction_date) as month_name, count(transaction_id) as total_orders
from online_shop_sales
group by month_name, month(transaction_date)
order by month(transaction_date)
```

Result Grid			Filter Rows:
	month_name	total_orders	
▶	January	4023	
	February	3245	
	March	4290	
	April	4098	
	May	4516	
	June	4159	
	July	5196	
	August	6112	
	September	4273	
	October	4153	
	November	3957	
	December	4502	



Объяснение запроса:

- `select monthname(transaction_date) as month_name, count(transaction_id) as total_orders`: извлекает название месяца из даты транзакции и подсчитывает общее количество заказов (`transaction_id`) для каждого месяца.
- `from online_shop_sales`: указывает таблицу, из которой берутся данные.
- `group by month_name, month(transaction_date)`: группирует данные по названиям месяцев и порядку месяцев, чтобы обеспечить правильную сортировку.
- `order by month(transaction_date)`: сортирует результаты по порядку месяцев, начиная с января (1) до декабря (12).

Топ категорий товаров по объему продаж

```
select product_category, volume, round(volume/total_volume * 100,2) as percentage_ratio
from (
select product_category, round(sum(
case
when discount_pct=10 and coupon_status= "used" then quantity * avg_price * (1 - 10/100)
when discount_pct=20 and coupon_status= "used" then quantity * avg_price * (1 - 20/100)
when discount_pct=30 and coupon_status= "used" then quantity * avg_price * (1 - 30/100)
else quantity * avg_price
end),2) as volume,
(select round(sum(
case
when discount_pct=10 and coupon_status= "used" then quantity * avg_price * (1 - 10/100)
when discount_pct=20 and coupon_status= "used" then quantity * avg_price * (1 - 20/100)
when discount_pct=30 and coupon_status= "used" then quantity * avg_price * (1 - 30/100)
else quantity * avg_price
end),2) from online_shop_sales) as total_volume
from online_shop_sales
```

group by product_category
order by volume desc) as a

Result Grid		  Filter Rows:	Export:
	product_category	volume	percentage_ratio
▶	Nest-USA	2383641.54	55.01
	Apparel	551665.06	12.73
	Nest	480543.69	11.09
	Office	257724.87	5.95
	Drinkware	188554.06	4.35
	Bags	142357.26	3.29
	Notebooks & Journals	101533.48	2.34
	Lifestyle	69525.29	1.6
	Nest-Canada	66675.28	1.54
	Headgear	47998.04	1.11
	Gift Cards	18521.38	0.43
	Accessories	6738.2	0.16
	Bottles	6349.85	0.15
	Waze	5892.31	0.14
	Housewares	4451.24	0.1
	Android	681.48	0.02

Объяснение запроса:

Подзапрос:

- `select product_category, round(sum(...),2) as volume`: подсчитывает общий объем продаж для каждой категории товаров с учетом скидок и использования купонов. Результаты округлены до двух знаков после запятой.
- `case ... end`: определяет стоимость продажи с учетом различных уровней скидок и применения купонов. Если купон был использован, применяется соответствующая скидка.
- `(select round(sum(...),2) from online_shop_sales) as total_volume`: вычисляет общий объем продаж по всем полам с использованием той же логики расчета скидок и применения купонов.
- `group by product_category`: группирует данные по категориям товаров.
- `order by volume desc`: сортирует результаты по объему продаж в порядке убывания.

Основной запрос:

- `select product_category, volume, round(volume/total_volume * 100,2) as percentage_ratio`: извлекает название категории товара, объем продаж и вычисляет процентное соотношение объема продаж этой категории к общему объему продаж. Результат округлен до двух знаков после запятой.
- `from (...) as a`: использует результаты вложенного запроса для дальнейших вычислений.

Топ 5 продуктов по объему продаж

```
select product_description,  
round(sum(  
case  
when discount_pct=10 and coupon_status= "used" then quantity * avg_price * (1 - 10/100)  
when discount_pct=20 and coupon_status= "used" then quantity * avg_price * (1 - 20/100)  
when discount_pct=30 and coupon_status= "used" then quantity * avg_price * (1 - 30/100)
```

```

else quantity * avg_price
end),2) as volume
from online_shop_sales
group by product_description
order by volume desc
limit 5

```

Result Grid		Filter Rows:	Export:
	product_description	volume	
▶	Nest Learning Thermostat 3rd Gen-USA - Stainl...	642649.09	
	Nest Cam Outdoor Security Camera - USA	588453.61	
	Nest Cam Indoor Security Camera - USA	495349.17	
	Nest Protect Smoke + CO White Battery Alarm-...	199793.88	
	Nest Protect Smoke + CO White Wired Alarm-USA	197998.53	

Объяснение запроса:

- `select product_description, round(sum(...),2) as volume`: извлекает описание продукта и подсчитывает общий объем продаж для каждого продукта. Объем продаж рассчитывается с учетом скидок и использования купонов, и округляется до двух знаков после запятой.
- `case ... end`: определяет стоимость продажи с учетом различных уровней скидок и применения купонов. Если купон был использован, применяется соответствующая скидка.
- `from online_shop_sales`: указывает таблицу, из которой берутся данные.
- `group by product_description`: группирует данные по описанию продукта.
- `order by volume desc`: сортирует результаты по объему продаж в порядке убывания.
- `limit 5`: ограничивает результаты до топ 5 продуктов с наибольшим объемом продаж.

Географический анализ продаж

```

select location,
round(sum(
case
when discount_pct=10 and coupon_status= "used" then quantity * avg_price * (1 - 10/100)
when discount_pct=20 and coupon_status= "used" then quantity * avg_price * (1 - 20/100)
when discount_pct=30 and coupon_status= "used" then quantity * avg_price * (1 - 30/100)
else quantity * avg_price
end),2) as volume
from online_shop_sales
group by location
order by volume desc

```

Result Grid		Filter Rows:
	location	volume
▶	Chicago	1505565.45
	California	1340059.76
	New York	868554.01
	New Jersey	379071.25
	Washington DC	239602.56

Объяснение запроса:

- `select location, round(sum(...),2) as volume`: извлекает местоположение и вычисляет общий объем продаж для каждой локации. Объем продаж рассчитывается с учетом скидок и использования купонов и округляется до двух знаков после запятой.
- `case ... end`: определяет стоимость продажи с учетом различных уровней скидок и применения купонов. Если купон был использован, применяется соответствующая скидка.
- `from online_shop_sales`: указывает таблицу, из которой берутся данные.
- `group by location`: группирует данные по местоположению.
- `order by volume desc`: сортирует результаты по объему продаж в порядке убывания.

Анализ по полу покупателя

```
select gender, volume, round(volume/total_volume * 100,2) as percentage_ratio
from (
select gender, round(sum(
case
when discount_pct=10 and coupon_status= "used" then quantity * avg_price * (1 - 10/100)
when discount_pct=20 and coupon_status= "used" then quantity * avg_price * (1 - 20/100)
when discount_pct=30 and coupon_status= "used" then quantity * avg_price * (1 - 30/100)
else quantity * avg_price
end),2) as volume,
(select round(sum(
case
when discount_pct=10 and coupon_status= "used" then quantity * avg_price * (1 - 10/100)
when discount_pct=20 and coupon_status= "used" then quantity * avg_price * (1 - 20/100)
when discount_pct=30 and coupon_status= "used" then quantity * avg_price * (1 - 30/100)
else quantity * avg_price
end),2) from online_shop_sales) as total_volume
from online_shop_sales
group by gender
order by volume desc) as a
```

Result Grid				Filter Rows:
	gender	volume	percentage_ratio	
▶	F	2696449.24	62.23	
	M	1636403.79	37.77	

Объяснение запроса:

Подзапрос:

- `select gender, round(sum(...),2) as volume`: вычисляет объем продаж для каждого пола с учетом скидок и использования купонов. Объем округляется до двух знаков после запятой.
- `case ... end`: определяет стоимость продажи с учетом различных уровней скидок и применения купонов. Если купон был использован, применяется соответствующая скидка.
- `(select round(sum(...),2) from online_shop_sales) as total_volume`: вычисляет общий объем продаж по всем полам с использованием той же логики расчета скидок и применения купонов.
- `group by gender`: группирует данные по полу покупателей.



- order by volume desc: сортирует результаты по объему продаж в порядке убывания.

Основной запрос:

- select gender, volume, round(volume/total_volume * 100,2) as percentage_ratio: извлекает пол, объем продаж и вычисляет процентное соотношение объема продаж по отношению к общему объему продаж, округленное до двух знаков после запятой.
- from (...) as a: использует результаты вложенного запроса для дальнейших вычислений.

Анализ продаж будни/выходные

```
select
case when dayofweek(transaction_date) in (1,7) then "Weekends"
else "Weekdays"
end as day_type,
round(sum(
case
when discount_pct=10 and coupon_status= "used" then quantity * avg_price * (1 - 10/100)
when discount_pct=20 and coupon_status= "used" then quantity * avg_price * (1 - 20/100)
when discount_pct=30 and coupon_status= "used" then quantity * avg_price * (1 - 30/100)
else quantity * avg_price
end),2) as volume
from online_shop_sales
group by day_type
```

Result Grid   Filter Rows:		
	day_type	volume
▶	Weekdays	3063817.7
	Weekends	1269035.34

Объяснение запроса:

- select case when dayofweek(transaction_date) in (1,7) then "Weekends" else "Weekdays" end as day_type, round(sum(case... end),2) as volume: Определяет тип дня (будний или выходной) и вычисляет общий объем продаж для каждого типа дня. Объем продаж рассчитывается с учетом скидок и использования купонов и округляется до двух знаков после запятой.
- from online_shop_sales: Указывает таблицу, из которой берутся данные.
- group by day_type: Группирует данные по типу дня (будний или выходной).