
Add ideas to this sticky note
Add ideas to this sticky note
Add ideas to this sticky note
Github : pradeep00000
SQL
Retail Supply Chain Sales Analysis
SHOP SMALL
Add ideas to this shape
Add ideas to this shape

```sql
create schema retail;
select * from retail.retail_sales;

#1. Retrieve the first 10 rows of the dataset.
select * from retail.retail_sales
limit 10;

#2. List all unique ship modes available.
select distinct `Ship Mode`
from retail.retail_sales;

#3. Count the total number of orders.
select count(distinct `Order ID`) as Total_Orders
from retail.retail_sales;

```

```sql
#4. Find the total sales amount.
select sum(Sales) as Total_Sales
from retail.retail_sales;

#5. Find the top 5 cities by number of orders.
select City , count(*) as Order_Count
from retail.retail_sales
group by City
order by Order_Count desc
limit 5;

#6. Find the average discount applied.
select avg(Discount) as Avg_Descount
from retail.retail_sales;
```

```sql
#7. Calculate the total profit for each product category.
select Category , sum(Profit) as Total_Profit
from retail.retail_sales
group by Category;

#8. List the top 3 customers by sales.
select `Customer Name` as Top_Customers , sum(sales) as Total_sales
from retail.retail_sales
group by Top_Customers
order by Total_sales desc
limit 3;

```

```sql
#9. Identify the most frequently returned product category.
select Category , count(*) as Total_Return
from retail.retail_sales
where Returned = 'Yes'
group by Category
order by Total_Return desc;

#10. Find the average profit by region.
select Region , avg(Profit) as Avg_Profit
from retail.retail_sales
group by Region;
```

```sql
#11. List products that resulted in a negative profit.
select `Product Name` , Profit
from retail.retail_sales
where Profit < 0;

#12. Identify the customer with the highest number of returns.
select `Customer Name`,count(`Customer Name`) as Total_Return
from retail.retail_sales
where Returned = 'Yes'
group by `Customer Name`
order by Total_Return desc;
```

```sql
#13. Calculate the profit margin (Profit/Sales) for each product.
select `Product Name` , (Profit/Sales) as Profit_Margin
from retail.retail_sales
order by Profit_Margin;

#14. Determine the percentage of orders returned.
select (count(case when Returned = 'Yes' then 1 end)
* 100.0 / count(*)) as Return_Percentage
from retail.retail_sales;
```

```sql
#15. Find the top-performing salesperson.
select `Retail Sales People` as Sales_Person , sum(sales) as Total_Sales
from retail.retail_sales
group by Sales_Person
order by Total_Sales desc
limit 1;

#16. Compare average sales in each segment.
select Segment , sum(Sales) as Total_Sales
from retail.retail_sales
group by Segment;
```

```sql
#17. Find customers who spent more than $10,000 in total.
select `Customer Name` , sum(Sales) as Total_Spend
from retail.retail_sales
group by `Customer Name`
having Total_Spend > 1000;
```