


# Ecommerce Marketplace Sales Analysis — PostgreSQL

## 1. Number of Products

	count bigint 
1	1000

SQL Query:

```
select
    count(productid)
from
    sales;
```

## 2. Total Unique Products

	count bigint 
1	100

SQL Query:

```
select
    count(distinct productname)
from
    sales;
```

## 3. Top 10 Most Expensive Products

	productname character varying (100) 	price numeric (10,2) 
1	Canvas	499.74
2	Cleanser	499.71
3	Earbuds	499.68
4	Spark Plugs	498.86
5	Thongs	496.70
6	Water Bottle	496.30
7	Tea Cup	495.75
8	Knee-High Socks	495.43
9	Board Game	495.17
10	Wool Socks	494.83

SQL Query:

```
select
    productname,
    price
from
    sales
order by
    price desc
limit 10;
```

#### 4. Number of Product Categories

	count bigint 
1	25

SQL Query:

```
select
    count(distinct category)
from
    sales;
```

## 5. List of Product Categories

	category character varying (100) 
1	Toys & Games
2	Bedsheets
3	Mobile phone accessori...
4	Vitamins and suppleme...
5	Underwear
6	Posters and artwork
7	Electronics
8	Home & Kitchen
9	Books
10	Clothing
11	Candles
12	Wine
13	Bicycles
14	Coats and jackets
15	Nail care
16	Cookware
17	Bath and body
18	Drinkware
19	Motor vehicle parts
20	Perfume and cologne
21	Art and crafting materials
22	Blankets
23	Socks
24	Makeup
25	Skin care

SQL Query:

```
select
    distinct category
from
    sales;
```

## 6. Number of Products in Every Category

	category character varying (100) 🔒	total_product bigint 🔒
1	Art and crafting materials	40
2	Bath and body	40
3	Bedsheets	40
4	Bicycles	40
5	Blankets	40
6	Books	40
7	Candles	40
8	Clothing	40
9	Coats and jackets	40
10	Cookware	40
11	Drinkware	40
12	Electronics	40
13	Home & Kitchen	40
14	Makeup	40
15	Mobile phone accessori...	40
16	Motor vehicle parts	40
17	Nail care	40
18	Perfume and cologne	40
19	Posters and artwork	40
20	Skin care	40
21	Socks	40
22	Toys & Games	40
23	Underwear	40
24	Vitamins and suppleme...	40
25	Wine	40

SQL Query:

```
select
    category,
    count(distinct productid) as total_product
from
    sales
group by
    1;
```

## 7. Average Rating and Total Items Sold in Every Category

	category character varying (100) 🔒	avg_rating numeric 🔒	total_sales bigint 🔒
1	Toys & Games	3.13	42295
2	Bedsheets	3.03	46631
3	Mobile phone accessori...	2.82	43660
4	Vitamins and suppleme...	2.98	44402
5	Underwear	3.20	34384
6	Posters and artwork	2.93	36588
7	Electronics	2.85	42844
8	Home & Kitchen	3.22	35777
9	Books	3.07	41215
10	Clothing	2.95	37672
11	Candles	2.81	34204
12	Wine	3.15	36352
13	Bicycles	3.00	42750
14	Coats and jackets	3.43	35640
15	Nail care	3.03	41545
16	Cookware	3.01	40268
17	Bath and body	2.95	42454
18	Drinkware	3.01	39870
19	Motor vehicle parts	2.88	45116
20	Perfume and cologne	2.83	35940
21	Art and crafting materials	2.97	45912
22	Blankets	3.22	40480
23	Socks	3.51	41812
24	Makeup	2.76	42337
25	Skin care	2.93	40889

SQL Query:

```
select
    category,
    round(avg(rating), 2) as avg_rating,
    sum(sales) as total_sales
from
    sales
group by
    1;
```

## 8. Top 5 Highest Rated Categories

	category character varying (100) 🔒	avg_rating numeric 🔒	total_sales bigint 🔒
1	Socks	3.51	41812
2	Coats and jackets	3.43	35640
3	Blankets	3.22	40480
4	Home & Kitchen	3.22	35777
5	Underwear	3.20	34384

SQL Query:

```
select
    category,
    round(avg(rating), 2) as avg_rating,
    sum(sales) as total_sales
from
    sales
group by
    1
order by
    avg_rating desc
limit 5;
```

## 9. Top 5 Categories by Products Sold

	category character varying (100) 🔒	avg_rating numeric 🔒	total_sales bigint 🔒
1	Bedsheets	3.03	46631
2	Art and crafting materials	2.97	45912
3	Motor vehicle parts	2.88	45116
4	Vitamins and suppleme...	2.98	44402
5	Mobile phone accessori...	2.82	43660

SQL Query:

```
select
    category,
    round(avg(rating), 2) as avg_rating,
    sum(sales) as total_sales
from
    sales
group by
    1
order by
    total_sales desc
```

```
limit 5;
```

### 10. Top Product by Items Sold in Each Category

	category character varying (100) 🔒	productname character varying (100) 🔒	total_sold bigint 🔒
2	Bicycles	Mountain Bike	20069
3	Skin care	Cleanser	15886
4	Motor vehicle parts	Spark Plugs	15745
5	Blankets	Throw Blanket	15315
6	Vitamins and suppleme...	Vitamin D	15239
7	Electronics	Headphones	15116
8	Perfume and cologne	Eau de Toilette	14820
9	Coats and jackets	Denim Jacket	14472
10	Art and crafting materials	Paint Set	14455
11	Drinkware	Water Bottle	14167
12	Wine	Rose Wine	14113
13	Nail care	Nail Clippers	13854
14	Mobile phone accessori...	Earbuds	13788
15	Bath and body	Body Scrub	13534
16	Cookware	Frying Pan	13501
17	Bedsheets	Linen Sheets	13411
18	Toys & Games	Puzzle	13253
19	Makeup	Foundation	13220
20	Socks	Wool Socks	12560
21	Clothing	T-Shirt	12542
22	Posters and artwork	Canvas Print	12480
23	Home & Kitchen	Toaster	11838
24	Candles	Scented Candle	11664
25	Underwear	Thongs	9862

SQL Query:

```
with top_product as(  
select  
category,  
    productname,  
    sum(sales) as total_sold,  
    rank() over (partition by category order by sum(sales) desc) as rank  
from sales
```

```

group by 1, 2
)
select
    category,
    productname,
    total_sold
from
    top_product
where
    rank = 1
order by
    total_sold desc;

```

## 11. Total Cities

	count bigint 
1	101

SQL Query:

```

select
    count(distinct city)
from
    sales;

```

## 12. List of Cities

	city character varying (100) 
1	Las Vegas
2	Milwaukee
3	Port St. Lucie
4	Columbia
5	New York
6	Knoxville
7	Palm Bay
8	Memphis
9	Springfield
10	Phoenix
11	Birmingham
12	San Francisco
13	St. Louis
14	Dallas
Total rows: 101    Query completed	



SQL Query:

```
select
    distinct city
from
    sales;
```

### 13. Average Discount in Every City

	city character varying (100) 🔒	avg_discount numeric 🔒
1	Las Vegas	0.23
2	Milwaukee	0.22
3	Port St. Lucie	0.15
4	Columbia	0.23
5	New York	0.30
6	Knoxville	0.23
7	Palm Bay	0.28
8	Memphis	0.29
9	Springfield	0.14
10	Phoenix	0.29
11	Birmingham	0.25
12	San Francisco	0.27
13	St. Louis	0.31
14	Dallas	0.35
15	Austin	0.24
16	Fort Worth	0.16
17	Chicago	0.32
18	Harrisburg	0.34
19	Denton	0.31
20	Jacksonville	0.23
Total rows: 101		Query complete 00:00:00.07

SQL Query:

```
select
    city,
    round(avg(discount), 2) as avg_discount
from
    sales
group by
    1;
```

#### 14. List of Products from Newest to Oldest

	productname character varying (100) 🔒	dateadded date 🔒
1	Microwave	2024-06-11
2	Vitamin D	2024-06-11
3	Cotton Socks	2024-06-10
4	Fish Oil	2024-06-10
5	Eau de Parfum	2024-06-10
6	Phone Case	2024-06-10
7	Nail Polish	2024-06-10
8	Screen Protector	2024-06-10
9	Science Book	2024-06-10
10	Puzzle	2024-06-08
11	Boxers	2024-06-08
12	Paint Set	2024-06-07
13	Boxers	2024-06-07
14	Body Scrub	2024-06-06
15	Blender	2024-06-06
Total rows: 1000		Query complete 00:00:00

SQL Query:

```
select
    productname,
    dateadded
from
    sales
order by
    dateadded desc;
```

### 15. List of Products having an Above Average Rating

	category character varying (100) 🔒	productname character varying (100) 🔒	rating numeric (5,2) 🔒
1	Makeup	Mascara	5.00
2	Makeup	Lipstick	5.00
3	Socks	Ankle Socks	5.00
4	Books	Biography	5.00
5	Home & Kitchen	Coffee Maker	5.00
6	Socks	Ankle Socks	5.00
7	Blankets	Weighted Blanket	5.00
8	Bedsheets	Bamboo Sheets	5.00
9	Bicycles	Road Bike	5.00
10	Coats and jackets	Denim Jacket	5.00
11	Skin care	Moisturizer	5.00
12	Clothing	Jeans	5.00
13	Underwear	Briefs	4.90
14	Wine	Sparkling Wine	4.90
15	Socks	Ankle Socks	4.90
16	Posters and artwork	Art Print	4.90
17	Home & Kitchen	Coffee Maker	4.90
18	Home & Kitchen	Toaster	4.90
19	Makeup	Lipstick	4.90
20	Vitamins and supplements	Fish Oil	4.90
Total rows: 504		Query complete 00:00:00.091	

SQL Query:

```
select
    category,
    productname,
    rating
from
    sales
where
    rating > (
        select
            avg(rating)
        from
            sales)
order by
    rating desc;
```

## 16. Create View: Product Overview

	productid integer	productname character varying (100)	category character varying (100)	price numeric (10,2)	discount numeric (5,2)	discounted_price numeric	rating numeric (5,2)	sales integer	stock_status text	dateadded date	city character varying (100)	revenue numeric
1	1	Headphones	Electronics	400.31	0.08	368.2852	1.70	466	In Stock	2023-11-10	Albuquerque	171620.9032
2	2	Headphones	Electronics	235.03	0.33	157.4701	2.30	1332	In Stock	2023-09-28	Queens	209750.1732
3	3	Smartwatch	Electronics	417.90	0.31	288.3510	1.80	252	In Stock	2023-10-20	Austin	72664.4520
4	4	Smartphone	Electronics	152.70	0.49	77.8770	3.40	1806	In Stock	2023-12-05	Dallas	140645.8620
5	5	Laptop	Electronics	394.74	0.23	303.9498	1.80	1508	In Stock	2023-08-10	Tucson	458356.2984
6	6	Headphones	Electronics	93.56	0.48	48.6512	1.30	241	In Stock	2023-09-21	Louisville	11724.9392
7	7	Laptop	Electronics	57.86	0.25	43.3950	3.70	1966	In Stock	2024-05-08	Cleveland	85314.5700
8	8	Laptop	Electronics	201.62	0.10	181.4580	1.70	1795	In Stock	2023-08-07	Houston	325717.1100
9	9	Smartphone	Electronics	485.10	0.45	266.8050	4.10	269	In Stock	2023-06-27	Cape Coral	71770.5450
10	10	Smartwatch	Electronics	170.00	0.16	142.8000	3.30	719	In Stock	2023-10-19	Minneapolis	102673.2000
11	11	Laptop	Electronics	274.45	0.27	200.3485	3.30	1064	In Stock	2024-01-07	Memphis	213170.8040
12	12	Headphones	Electronics	90.98	0.20	72.7840	1.10	647	In Stock	2024-04-10	Mesa	47091.2480
13	13	Headphones	Electronics	107.43	0.30	75.2010	3.80	1513	In Stock	2023-10-25	Palm Bay	113779.1130
14	14	Laptop	Electronics	66.78	0.05	63.4410	4.50	502	In Stock	2024-04-11	Columbia	31847.3820
15	15	Smartwatch	Electronics	336.30	0.28	242.1360	3.40	1762	In Stock	2024-03-03	Seattle	426643.6320
16	16	Headphones	Electronics	359.49	0.13	312.7563	4.00	794	In Stock	2023-11-18	Hartford	248328.5022
17	17	Headphones	Electronics	64.34	0.24	48.8984	2.80	1020	In Stock	2024-03-08	Palm Bay	49876.3680
18	18	Smartwatch	Electronics	454.71	0.10	409.2390	2.00	524	In Stock	2024-01-04	Charlotte	214441.2360
19	19	Headphones	Electronics	151.98	0.40	91.1880	1.60	1976	In Stock	2024-01-30	Charlotte	180187.4880
20	20	Smartwatch	Electronics	402.80	0.16	338.1056	1.70	1371	In Stock	2023-10-25	Alaska	445030.3000
Total rows: 1000    Query complete 00:00:00.997												

### SQL Query:

```
create view vw_product_overview as
select
    productid,
    productname,
    category,
    price,
    discount,
    price * (1-discount) as discounted_price,
    rating,
    sales,
    case
        when stockquantity > 0 then 'In Stock'
        else 'Out Of Stock'
    end as stock_status,
    dateadded,
    city,
    sales * (price * (1-discount)) as revenue
from
    sales;



select
    *
from
    vw_product_overview;
```

## 17. Create a Summary Table

SQL Query:

```
create table DailyProductSummary(  
    SummaryDate date,  
    Category text,  
    TotalSales int,  
    AvgPrice numeric(10, 2),  
    AvgRating numeric(10, 2),  
    TopProduct text
```

## 18. Create a Procedure to Automate the Process for Creating Daily Product Summary Table.

	name character varying (50) 	total_movies bigint 
1	English	21
2	Hindi	13
3	Telugu	3
4	Kannada	1
5	Bengali	1

SQL Query:

```
CREATE OR REPLACE PROCEDURE run_daily_product_summary()  
LANGUAGE plpgsql  
AS $$  
BEGIN  
    DELETE FROM DailyProductSummary  
    WHERE SummaryDate = CURRENT_DATE;  
END;  
$$;
```

```
insert  
    into  
        DailyProductSummary  
select  
    current_date as SummaryDate,  
    Category,  
    sum(sales) as TotalSales,  
    avg(price) as AvgPrice,  
    avg(rating) as AvgRating,  
    (  
        select
```

```
        productname
    from
        sales p2
    where
        p1.category = p2.category
    order by
        sales desc
    limit 1
) as TotalProduct
from
    sales p1
group by
    category;
end;
```

## 19. Call the Function

SQL Query:

```
call run_daily_product_summary();

select
    *
from
    DailyProductSummary;
```