SQLQuery1.sql - L...0UQ3AT3\Manu (56))* | SQLQuery2.sql - L...0UQ3AT3\Manu (72))

```sql
1. Total Sales Per Product

SELECT product_id, SUM(total_price) AS total_sales
FROM retail_price
GROUP BY product_id
ORDER BY total_sales DESC
```

147 %

Results | Messages

	product_id	total_sales
1	health2	63885
2	health5	54730.2
3	computers4	46916.51
4	bed2	42938.66
5	computers6	41082.6
6	watches6	37683.42
7	furniture2	37608.9
8	watches1	31786.82
9	watches7	31623.81
10	watches5	30467.5
11	bed5	29997.36
12	garden6	26577.22
13	watches4	23943
14	health8	22627.23
15	garden10	21390.69
16	garden4	21056.8

```sql
2. Monthly Sales Trend
SELECT month_year, SUM(total_price) AS monthly_sales
FROM retail_price
GROUP BY month_year
ORDER BY month_year
```

147 %

Results | Messages

	month_year	monthly_sales
1	2017-01-01 00:00:00.000	2864.19
2	2017-02-01 00:00:00.000	3584.11
3	2017-03-01 00:00:00.000	10204.38
4	2017-04-01 00:00:00.000	11524.62
5	2017-05-01 00:00:00.000	21843.33
6	2017-06-01 00:00:00.000	23245.24
7	2017-07-01 00:00:00.000	41049.89
8	2017-08-01 00:00:00.000	49550.41
9	2017-09-01 00:00:00.000	44826.6
10	2017-10-01 00:00:00.000	48569.38
11	2017-11-01 00:00:00.000	87009.77
12	2017-12-01 00:00:00.000	65311.36
13	2018-01-01 00:00:00.000	77569.62
14	2018-02-01 00:00:00.000	88275.95
15	2018-03-01 00:00:00.000	85049.85
16	2018-04-01 00:00:00.000	75977.18
17	2018-05-01 00:00:00.000	91232.46

```sql
3. Average Price and Freight by Category

SELECT product_category_name,
       AVG(unit_price) AS avg_unit_price,
       AVG(freight_price) AS avg_freight
FROM retail_price
GROUP BY product_category_name
```

147 %

Results | Messages

	product_category_name	avg_unit_price	avg_freight
1	bed_bath_table	78.6292780363935	16.1397175761148
2	computers_accessories	119.482322892029	25.1037411318841
3	consoles_games	27.0337662336364	14.8094149281818
4	cool_stuff	107.85751177614	18.9750957305263
5	furniture_decor	60.1542624745834	16.9446169483333
6	garden_tools	80.0946991227499	28.4583100731875
7	health_beauty	132.309869986615	18.6074475930538
8	perfumery	89.348812576923	14.3363105792308
9	watches_gifts	164.880006987087	16.4928399131359

```sql
4. Top Products by Quantity Sold

SELECT Top 10 product_id, SUM(qty) AS total_qty
FROM retail_price
GROUP BY product_id
ORDER BY total_qty DESC
```

147 %

Results | Messages

	product_id	total_qty
1	furniture2	527
2	bed2	487
3	garden6	484
4	garden10	391
5	garden4	388
6	garden9	373
7	computers4	341
8	watches6	323
9	health9	281
10	computers6	274

```sql
5.  Compare Own Price vs Competitor (comp_1)

SELECT product_id,
       AVG(unit_price) AS avg_own_price,
       AVG(comp_1) AS avg_comp1_price
FROM retail_price
GROUP BY product_id
```

147 %

Results | Messages

	product_id	avg_own_price	avg_comp1_price
1	bed1	42.21198863625	86.1885115
2	bed2	86.774536	86.774536
3	bed3	92.1013636363636	84.5014712727273
4	bed4	46.725444444	85.5516184
5	bed5	198.38170272	89.9
6	computers1	92.4821486546667	137.619907153333
7	computers2	87.234536041	133.60284392
8	computers3	143.47296429	153.59186624
9	computers4	141.576033738889	141.576033738889
10	computers5	100.2437367025	137.7180402
11	computers6	149.9568181875	129.5060549
12	consoles1	28.2416666666667	28.2416666666667
13	consoles2	25.584285714	25.733333333
14	cool1	98.362507936	98.362507936
15	cool2	128.215961538462	99.2110256407692
16	cool3	42	79.2619047614286

```sql
6.  Impact of Holidays on Sales

SELECT holiday, SUM(total_price) AS total_sales
FROM retail_price
GROUP BY holiday
```

147 %

Results | Messages

	holiday	total_sales
1	0	50438.94
2	1	503555.25
3	2	229523.68
4	3	91223.46
5	4	87009.77

```sql
7. Average Freight by Weight Range
SELECT
    CASE
        WHEN product_weight_g < 500 THEN 'Light'
        WHEN product_weight_g BETWEEN 500 AND 2000 THEN 'Medium'
        ELSE 'Heavy'
    END AS weight_category,
    AVG(freight_price) AS avg_freight
FROM retail_price
GROUP BY
    CASE
        WHEN product_weight_g < 500 THEN 'Light'
        WHEN product_weight_g BETWEEN 500 AND 2000 THEN 'Medium'
        ELSE 'Heavy'
    END
```

147 %

Results | Messages

	weight_category	avg_freight
1	Heavy	30.9938133677852
2	Light	15.581407285151
3	Medium	19.6655727556064

```sql
8. Price Change Over Time for a Product

SELECT product_id, month_year, unit_price, lag_price,
       (unit_price - lag_price) AS price_change
FROM retail_price
WHERE product_id = 'bed1'
ORDER BY month_year
```

147 %

Results | Messages

	product_id	month_year	unit_price	lag_price	price_change
1	bed1	2017-05-01 00:00:00.000	45.95	45.9	0.0500000000000043
2	bed1	2017-06-01 00:00:00.000	45.95	45.95	0
3	bed1	2017-07-01 00:00:00.000	45.95	45.95	0
4	bed1	2017-08-01 00:00:00.000	45.95	45.95	0
5	bed1	2017-09-01 00:00:00.000	45.95	45.95	0
6	bed1	2017-10-01 00:00:00.000	45.95	45.95	0
7	bed1	2017-11-01 00:00:00.000	40.53181818	45.95	-5.41818182
8	bed1	2017-12-01 00:00:00.000	39.99	40.53181818	-0.54181818
9	bed1	2018-01-01 00:00:00.000	39.99	39.99	0
10	bed1	2018-02-01 00:00:00.000	39.99	39.99	0
11	bed1	2018-03-01 00:00:00.000	39.99	39.99	0
12	bed1	2018-04-01 00:00:00.000	39.99	39.99	0
13	bed1	2018-05-01 00:00:00.000	39.99	39.99	0
14	bed1	2018-06-01 00:00:00.000	39.99	39.99	0

```
9. Customer Purchase Patterns by Weekday

SELECT weekday, COUNT(DISTINCT customers) AS unique_customers,
       SUM(total_price) AS total_sales
FROM retail_price
GROUP BY weekday
ORDER BY weekday
```

47 %

Results | Messages

	weekday	unique_customers	total_sales
1	20	18	103384.68
2	21	36	277603.97
3	22	37	292791.58
4	23	40	287970.87

```
10.  Conversion Rate by Product Score

SELECT product_score,
       COUNT(*) AS num_orders,
       SUM(qty) AS total_units,
       SUM(total_price) AS total_revenue
FROM retail_price
GROUP BY product_score
ORDER BY product_score DESC
```

147 %

Results | Messages

	product_score	num_orders	total_units	total_revenue
1	4.5	11	137	4096.3
2	4.4	35	475	55936.96
3	4.3	120	1340	156803.8
4	4.2	155	2359	264701.7
5	4.1	120	2328	201487.93
6	4	63	801	59119
7	3.9	71	1125	81526.85
8	3.8	47	521	63307.47
9	3.7	25	292	37785.92
10	3.5	18	310	26680.21
11	3.3	11	111	10304.96

```sql
11. Product Volume Contribution

SELECT product_id,
       SUM(volume) AS total_volume,
       SUM(total_price) AS total_sales,
       SUM(volume) / SUM(total_price) AS volume_per_dollar
FROM retail_price
GROUP BY product_id
ORDER BY volume_per_dollar DESC
```

147 %

Results | Messages

	product_id	total_volume	total_sales	volume_per_dollar
1	garden7	520960	6220.9	83.7435097815429
2	garden2	553520	7333.7	75.4762261886906
3	furniture3	168000	3507.95	47.8912185179378
4	cool5	425568	9627.65	44.202687052396
5	garden5	275184	10759.3	25.5763850808138
6	garden10	520960	21390.69	24.3545205881624
7	bed3	230384	10304.96	22.3566127379437
8	furniture1	130000	6162.11	21.0966698095295
9	cool1	236250	11868.42	19.9057667322188
10	garden9	336600	20387.2	16.510359441218
11	cool3	54432	3656.5	14.8863667441542
12	cool2	204750	15159.81	13.5061059472381
13	bed4	80000	6011.77	13.3072289858062
14	garden4	277200	21056.8	13.1643934500969

```sql
12. Category Performance on Weekends vs Weekdays

SELECT product_category_name,
       SUM(CASE WHEN weekend = 1 THEN total_price ELSE 0 END) AS weekend_sales,
       SUM(CASE WHEN weekend = 0 THEN total_price ELSE 0 END) AS weekday_sales
FROM retail_price
GROUP BY product_category_name
```

147 %

Results | Messages

	product_category_name	weekend_sales	weekday_sales
1	bed_bath_table	0	0
2	computers_accessories	0	0
3	consoles_games	0	0
4	cool_stuff	0	0
5	furniture_decor	0	0
6	garden_tools	0	0
7	health_beauty	0	0
8	perfumery	0	0
9	watches_gifts	0	0

```sql
13. Freight as a Percentage of Total Price

SELECT product_id,
       AVG(freight_price / total_price) * 100 AS avg_freight_percentage
FROM retail_price
GROUP BY product_id
ORDER BY avg_freight_percentage DESC
```

147 %

Results | Messages

	product_id	avg_freight_percentage
1	cool3	23.1747797798382
2	garden2	18.3941110646624
3	garden5	16.4287727522605
4	consoles2	14.1027928594705
5	furniture3	13.9981471071357
6	consoles1	12.5238685540341
7	garden1	8.93378954388571
8	garden3	8.92430797418882
9	garden7	7.96516041791302
10	health9	7.95596561796345
11	watches3	7.79129827025506
12	bed1	7.61658473624236
13	health3	7.41353334113668

```sql
14. Product Description & Name Length vs Sales

SELECT product_id,
       AVG(product_name_lenght) AS avg_name_length,
       AVG(product_description_lenght) AS avg_description_length,
       SUM(total_price) AS total_sales
FROM retail_price
GROUP BY product_id
```

147 %

Results | Messages

	product_id	avg_name_length	avg_description_length	total_sales
1	bed1	39	161	5831.77
2	bed2	54	245	42938.66
3	bed3	55	312	10304.96
4	bed4	38	735	6011.77
5	bed5	56	162	29997.36
6	computers1	45	236	12306.81
7	computers2	33	256	15439.25
8	computers3	42	363	15111.82
9	computers4	59	1893	46916.51
10	computers5	33	300	11240.96
11	computers6	48	894	41082.6
12	consoles1	49	100	3416.7
13	consoles2	56	237	2384
14	cool1	54	1012	11868.42

```sql
15. Correlation Proxy: Product Score vs Unit Price

SELECT product_id,
       AVG(product_score) AS avg_score,
       AVG(unit_price) AS avg_price
FROM retail_price
GROUP BY product_id
ORDER BY avg_score DESC
```

147 %

Results | Messages

	product_id	avg_score	avg_price
1	health4	4.5	29.9
2	bed5	4.4	198.38170272
3	cool3	4.4	42
4	furniture4	4.4	96.156904762
5	perfumery2	4.4	123.823009769231
6	computers6	4.3	149.9568181875
7	cool5	4.3	98.6565
8	garden1	4.3	105.764478112778
9	garden2	4.3	54.8343137252941
10	health5	4.3	350.74431818
11	cool2	4.3	128.215961538462
12	health9	4.3	22.6237448561111
13	perfumery1	4.3	54.8746153846154
14	watches4	4.2	106.26428571