


```
115 #3)Number of customers by marital status:
116 • SELECT marital_status, COUNT(customer_id) AS customer_count
117 FROM customer
118 GROUP BY marital_status;
119
120 #4)Total revenue:
121 • SELECT SUM(t.quantity * p.product_retail_price) AS total_revenue
122 FROM transactions t
123 JOIN products p ON t.product_id = p.product_id;
```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

marital_status	customer_count
M	5110
S	5099



Result  
Grid



Form  
Editor



Field  
Types



Query  
Stats



```
1      #Maven Markets
2
3      #step 1 creating tables and adding neccessary constraints
4
5      • CREATE DATABASE MAVEN_MARKET;
6
7      • use maven_market;
8
9      • CREATE TABLE customer (
10         customer_id int primary key,
11         customer_address varchar(200),
12         customer_city varchar(200),
13         customer_state_province varchar(200),
14         birthdate date,
15         marital_status varchar(150),
16         yearly_income varchar(200),
17         gender varchar(2),
18         member_card varchar(200)
19     );
20
21
22      • CREATE TABLE regions (
23         region_id int primary key,
24         sales_district varchar(200),
25         sales_region varchar(200)
26     );
```

```
27 ● ○ CREATE TABLE products (  
28     product_id int primary key,  
29     product_brand varchar(200),  
30     product_name varchar(200),  
31     product_retail_price float,  
32     product_cost float,  
33     product_weight float,  
34     recyclable varchar(20),  
35     low_fat varchar(20)  
36 );  
37  
38  
39 ● ○ CREATE TABLE store (  
40     store_id int primary key,  
41     region_id int, foreign key(region_id)REFERENCES regions(region_id),  
42     store_type varchar(255),  
43     store_name varchar(255),  
44     store_street_address varchar(255),  
45     store_city varchar(255),  
46     store_state varchar(255)  
47 );  
48  
49 ● ○ CREATE TABLE returns (  
50     return_date date,  
51     product_id int,  
52     store_id int, foreign key(store_id)references store(store_id),  
53     quantity int  
54 );  
55
```

04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14

#1)Average cost per product:

```
SELECT AVG(product_cost) AS average_cost_per_product  
FROM products;
```

#2)Store-wise sales:

```
SELECT t.store_id, SUM(t.quantity * p.product_retail_price) AS total_sales  
FROM transactions t  
INNER JOIN products p ON t.product_id = p.product_id  
GROUP BY t.store_id;
```

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

average_cost_per_product
0.8616738188612103



Result  
Grid



Form  
Editor



Field  
Types



Query  
Stats



```

109 #2)Store-wise sales:
110 • SELECT t.store_id, SUM(t.quantity * p.product_retail_price) AS total_sales
111 FROM transactions t
112 INNER JOIN products p ON t.product_id = p.product_id
113 GROUP BY t.store_id;
114

```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

store_id	total_sales
7	3152.780003130436
13	5964.860008120537
11	3478.560002863407
3	3215.5800048708916
23	1981.1700007915497
24	3592.680001080036
2	286.2500013113022
17	4837.9400117993355
6	2584.330002486706
16	1428.5600016117096
15	3109.2000052928925
22	270.3199999332428
14	256.610000371933



Result  
Grid



Form  
Editor



Field  
Types



Query  
Stats





82

```
83 • select year(stock_date) from maven_market_backup.mavenmarket_transactions;  
84 • set sql_safe_updates = 0;  
85 • update mavenmarket_transactions set stock_date = str_to_date(stock_date, '%d-%m-%Y');  
86 • ALTER TABLE mavenmarket_transactions modify stock_date date;
```

87

```
88 • select year(return_date) from maven_market_backup.mavenmarket_returns;  
89 • set sql_safe_updates = 0;  
90 • update mavenmarket_returns set return_date = str_to_date(return_date, '%d-%m-%Y');  
91 • ALTER TABLE mavenmarket_returns modify return_date date;
```

92

93 #Step 2 Inserting Data into maven\_market database

```
94 • INSERT INTO maven_market.customer select * from maven_market_backup.mavenmarket_customers;  
95 • INSERT INTO maven_market.products select * from maven_market_backup.mavenmarket_products;  
96 • INSERT INTO maven_market.regions select * from maven_market_backup.mavenmarket_regions;  
97 • INSERT INTO maven_market.store select * from maven_market_backup.mavenmarket_stores;  
98 • INSERT INTO maven_market.returns select * from maven_market_backup.mavenmarket_returns;  
99 • INSERT INTO maven_market.transactions select * from maven_market_backup.mavenmarket_transactions;
```

00

```
01 • use maven_market;
```


02


```
57 • CREATE TABLE transactions(  
58     transaction_date date,  
59     stock_date date,  
60     product_id int,foreign key (product_id) references products (product_id),  
61     customer_id int,foreign key (customer_id) references customer(customer_id),  
62     store_id int,foreign key (store_id) references store(store_id),  
63     quantity int  
64 );  
65  
66 #Creating Backup database and importing data  
67  
68 • create database MAVEN_MARKET_BACKUP;  
69  
70 # Transformation  
71 • select year(birthdate) from maven_market_backup.mavenmarket_customers;  
72 • set sql_safe_updates = 0;  
73 • update mavenmarket_customers set birthdate = str_to_date(birthdate, '%d-%m-%Y');  
74 • ALTER TABLE mavenmarket_customers modify birthdate date;  
75  
76  
77 • select year(transaction_date) from maven_market_backup.mavenmarket_transactions;  
78 • set sql_safe_updates = 0;  
79 • update mavenmarket_transactions set transaction_date = str_to_date(transaction_date, '%d-%m-%Y');  
80 • ALTER TABLE mavenmarket_transactions modify transaction_date date;
```


```
20 #4)Total revenue:
21 • SELECT SUM(t.quantity * p.product_retail_price) AS total_revenue
22 FROM transactions t
23 JOIN products p ON t.product_id = p.product_id;
24
25 #5)Average weight per product:
26 • SELECT AVG(product_weight) AS average_weight_per_product
```


Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

total_revenue
34158.84004366398

  
Result  
Grid

  
Form  
Editor

  
Field  
Types

  
Query  
Stats



42

43

#8)Revenue by customer income:

44

• `SELECT c.yearly_income, SUM(t.quantity * p.product_retail_price) AS total_revenue`

45

`FROM transactions t`

46

`RIGHT JOIN products p ON t.product_id = p.product_id`




47

`RIGHT JOIN customer c ON t.customer_id = c.customer_id`

48

`GROUP BY c.yearly_income;`

49

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

yearly_income	total_revenue
\$30K - \$50K	11452.980021059513
\$70K - \$90K	3962.760001063347
\$50K - \$70K	6085.179994285107
\$10K - \$30K	6941.360012888908
\$90K - \$110K	1644.0500051379204
\$110K - \$130K	1507.2500020861626
\$130K - \$150K	1903.5200073122978
\$150K +	661.7399998307228

Result  
GridForm  
EditorField  
TypesQuery  
Stats



Result 7 x

! Read Only

```

150 #9)Total number of transactions:
151 • SELECT COUNT(product_id) AS total_transactions
152 FROM transactions;
153
154 #10)Revenue by customer membership:
155 • SELECT c.member_card, SUM(t.quantity * p.product_retail_price) AS total_revenue
156 FROM transactions t
157 JOIN customer c ON t.customer_id = c.customer_id
158 JOIN products p ON t.product_id = p.product_id
159 GROUP BY c.member_card;
160 #inner join

```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

member_card	total_revenue
Bronze	19315.920019328594
Silver	3017.100003838539
Normal	7668.47001427412
Golden	4157.350006222725



Result Grid



Form Editor



Field Types





Query Stats



```

125 #5)Average weight per product:
126 • SELECT AVG(product_weight) AS average_weight_per_product
127 FROM products;
128
129 #6)Total revenue by region:
130 • SELECT s.region_id, SUM(t.quantity * p.product_retail_price) AS total_revenue
131 FROM transactions t
132 JOIN products p ON t.product_id = p.product_id
133 JOIN store s ON t.store_id = s.store_id
134 GROUP BY s.region_id;
135

```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

region_id	total_revenue
3	3152.780003130436
23	5964.860008120537
22	3478.560002863407
76	3215.5800048708916
89	1981.1700007915497
7	3592.680001080036
78	286.2500013113022
84	4837.9400117993355
47	2584.330002486706
87	1428.5600016117096
18	3109.2000052928925
88	270.3199999332428
1	256.610000371933



Result Grid



Form Editor



Field Types



Query Stats