

# TABLE 1

1) Write a query to Retrieve the Top 20 Most Expensive Products

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 use capstone;
2 show tables;
3 select * FROM cloth ORDER BY Price DESC LIMIT 20;
4
```

The result grid displays the top 20 most expensive products from the 'cloth' table. The columns are Brandname, Category, Producttype, Productname, and Price.

Brandname	Category	Producttype	Productname	Price
Manyavar	Women	Lehenga, Blouse, Dupatta, Belt, Laitkan	Light Orange Imperial Patterned Bridal Lehenga	44999
Manyavar	Women	Lehenga, Blouse, Dupatta, Belt, Laitkan	Geranium Pink Bel Buti Patterned Bridal Lehenga	44999
Manyavar	Women	Lehenga, Blouse, Dupatta, Belt, Laitkan	Orange Maroon Embroidered Bridal Lehenga	44999
Manyavar	Women	Lehenga, Blouse, Dupatta, Belt, Laitkan	Rust Red Paisley Patterned Bridal Lehenga	44999
Manyavar	Women	Lehenga, Blouse, Dupatta, Belt, Laitkan	Coral Blue Floral Art Nouveau Patterned Bridal L...	44999
Manyavar	Women	Lehenga, Blouse, Dupatta, Belt, Laitkan	Cherry Red And Sunrise Orange Rhinestone Wo...	39999

The output pane shows the execution of the query, with a message indicating that 20 rows were returned.

2) Write a query Calculate the Total Price for Each Category:

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 use clothing;
2 show tables;
3 SELECT Category, SUM(price) AS total_price
4 FROM table1
5 GROUP BY Category
6
```

The result grid displays the total price for each category from the 'table1' table. The columns are Category and total\_price.

Category	total_price
Men	3040811
Women	5210820
Kid	868234

The output pane shows the execution of the query, with a message indicating that 3 rows were returned.

### 3) Write a query to Retrieve Products with a Price Range 2000 to 5000

MySQL Workbench interface showing a query to retrieve products with a price range of 2000 to 5000. The query is executed, and the result grid shows 700 rows of data.

**Query:**

```
SELECT *
FROM table1
WHERE price BETWEEN 2000 AND 5000;
```

**Result Grid:**

Brandname	Category	Producttype	Productname	Price
Superdry	Men	Men'S Tahirts	Team Sports Cali Regular Fit Polo T-Shirt	2300
Superdry	Men	Men'S Tahirts	Running Regular Fit Round-Neck T-Shirt	2000
Superdry	Men	Men'S Tahirts	Men Regular Fit Polo T-Shirt	2159
Superdry	Men	Men'S Tahirts	Vintage Tipped Relaxed Fit Polo T-Shirt	2399
Superdry	Men	Men'S Tahirts	Classic Superstate Relaxed Fit Polo T-Shirt	2500
Arman Exchange	Men	Men'S Tahirts	Crew-Neck T-Shirt With Logo Print	3599

**Action Output:**

#	Time	Action	Message	Duration / Fetch
86	19:25:08	SELECT max(Price) as Highestprice from table1 LIMIT 0, 5000	1 row(s) returned	0.000 sec / 0.000 sec
87	20:38:49	alter table table1 rename column Price to price	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec
88	20:39:05	SELECT * FROM clothing.table1 LIMIT 0, 5000	4312 row(s) returned	0.000 sec / 0.000 sec
89	20:39:15	alter table table1 rename column price to Price	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec
90	20:39:20	SELECT * FROM clothing.table1 LIMIT 0, 5000	4312 row(s) returned	0.000 sec / 0.016 sec
91	22:30:42	SELECT * FROM table1 WHERE price BETWEEN 2000 AND 5000 LIMIT 0, 5000	700 row(s) returned	0.000 sec / 0.000 sec

### 4) Write a query to Calculate the Average Price for Each Brand.

MySQL Workbench interface showing a query to calculate the average price for each brand. The query is executed, and the result grid shows 6 rows of data.

**Query:**

```
SELECT Brandname, AVG(price) AS average_price
FROM table1
GROUP BY Brandname;
```

**Result Grid:**

Brandname	average_price
Netplay	802.5759
Teamspirit	394.3438
Wuxi	487.2000
Fort Collins	517.0000
Snitch	1239.7500
Bombay Begum	443.3333

**Action Output:**

#	Time	Action	Message	Duration / Fetch
88	20:39:05	SELECT * FROM clothing.table1 LIMIT 0, 5000	4312 row(s) returned	0.000 sec / 0.000 sec
89	20:39:15	alter table table1 rename column price to Price	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec
90	20:39:20	SELECT * FROM clothing.table1 LIMIT 0, 5000	4312 row(s) returned	0.000 sec / 0.016 sec
91	22:30:42	SELECT * FROM table1 WHERE price BETWEEN 2000 AND 5000 LIMIT 0, 5000	700 row(s) returned	0.000 sec / 0.000 sec
92	22:35:11	select avg(Price) group by Brandname	Error Code: 1054. Unknown column 'Price' in field list	0.000 sec
93	22:36:26	SELECT Brandname, AVG(price) AS average_price FROM table1 GROUP BY Brandname	223 row(s) returned	0.016 sec / 0.000 sec

5) Write a query to find Products with Unique Brand-Category Combinations

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1
2 SELECT DISTINCT Brandname, Category, Productname
3 FROM table1;
4
```

The Results Grid displays the following data:

Brandname	Category	Productname
Netplay	Men	Cotton Shirt With Patch Pocket
Teamspirit	Men	Relaxed Fit Ombre-Dyed Crew-Neck T-Shirt
Wuxi	Men	Cuban Collar Shirt With Short Sleeves
Wuxi	Men	Men Knitted Loose Fit Shirt With Mandarin Collar
Fort Collins	Men	Polo T-Shirt With Patch Pocket
Snitch	Men	Dekline Slim Fit Shirt

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
96	22:39:07	SELECT DISTINCT count(Brandname) Category, Productname FROM table1 LIMIT 0, 5000	Error Code: 1140. In aggregated query without GROUP BY, expression #2 of SELE...	0.000 sec
97	22:39:18	SELECT DISTINCT count(Brandname) FROM table1 LIMIT 0, 5000	1 row(s) returned	0.000 sec / 0.000 sec
98	22:39:32	SELECT DISTINCT count(Netplay) FROM table1 LIMIT 0, 5000	Error Code: 1054. Unknown column 'Netplay' in field list	0.000 sec
99	22:40:17	SELECT DISTINCT Brandname, Category, Productname FROM table1 LIMIT 0, 50...	3117 row(s) returned	0.031 sec / 0.000 sec
100	22:40:29	SELECT DISTINCT Brandname, Category, Productname.price FROM table1 LIMIT ...	3734 row(s) returned	0.047 sec / 0.000 sec
101	22:40:37	SELECT DISTINCT Brandname, Category, Productname FROM table1 LIMIT 0, 50...	3117 row(s) returned	0.016 sec / 0.000 sec

## TABLE 2

1) Write a query Order Rows by Material Type in Ascending Order .

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

clothing

Tables

table1

table2

Views

Stored Procedures

Functions

db

mydb

new\_schema

subQuery

sys

testdb

world

Administration Schemas

Information

No object selected

Object Info Session

Query Completed

SQL File 4

table1

table2

SQL File 5

table2

table2

table2

Limit to 5000 rows

1 SELECT \*

2 FROM table2

3 ORDER BY Materialtype ASC;

Result Grid

sno	color1	color2	color3	color4	WearType	Materialtype
4179	Pink	Gold	Orange	Violet	top wear	Art Silk
4204	Pink	Gold	Orange	Violet	top wear	Art Silk
4087	Pink	Green	White	Grey	top wear	Art Silk
4081	Pink	Green	White	Grey	top wear	Art Silk
4201	Pink	Gold	Orange	Violet	top wear	Art Silk

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:57:01	SELECT * FROM clothing table2 LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.015 sec
2	15:57:35	SELECT * FROM table2 ORDER BY Materialtype ASC LIMIT 0, 1000	1000 row(s) returned	0.078 sec / 0.016 sec
3	15:57:42	SELECT * FROM table2 ORDER BY Materialtype ASC LIMIT 0, 5000	4316 row(s) returned	0.047 sec / 0.031 sec

2) Write a query to Calculate the Average Number of Reviews

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

clothing

Tables

table1

table2

Views

Stored Procedures

Functions

db

mydb

new\_schema

subQuery

sys

testdb

world

Administration Schemas

Information

Table: table2

Columns:

sno int

color1 text

color2 text

color3 text

color4 text

WearType text

Materialtype text

reviews int

Object Info Session

Query Completed

SQL File 4

table1

table2

SQL File 5

table2

table2

table2

table2

table2

table2

table2

Limit to 5000 rows

1 select avg(reviews) as avg\_review from table2

Result Grid

avg_review
156.4076

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
48	16:56:00	SELECT * FROM clothing table2 LIMIT 0, 5000	4316 row(s) returned	0.000 sec / 0.031 sec
49	16:59:42	SELECT * FROM clothing table2 LIMIT 0, 5000	4316 row(s) returned	0.000 sec / 0.015 sec
50	16:59:42	select avg(reviews) as average_reviews LIMIT 0, 5000	Error Code: 1054. Unknown column 'reviews' in field list	0.000 sec
51	16:59:51	select avg(reviews) as average_reviews LIMIT 0, 5000	Error Code: 1054. Unknown column 'reviews' in field list	0.000 sec
52	16:59:58	SELECT * FROM clothing table2 LIMIT 0, 5000	4316 row(s) returned	0.016 sec / 0.015 sec
53	17:01:42	select avg(reviews) as avg_review from table2 LIMIT 0, 5000	1 row(s) returned	0.016 sec / 0.000 sec

3) Write a query to Retrieve the Top 30 Most Expensive Products by Wear type:

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

1 SELECT table2.WearType, table1.Productname, table1.Price
2 FROM table2
3 JOIN table1
4 ON table1.sno = table2.sno
5 ORDER BY table1.Price DESC
6 LIMIT 30;

```

The Result Grid displays the following data:

WearType	Productname	Price
top wear	Rust Red Paisley Patterned Bridal Lehenga	44999
top wear	Geranium Pink Bel Buti Patterned Bridal Lehenga	44999
top wear	Coral Blue Floral Art Nouveau Patterned Bridal L...	44999
top wear	Light Orange Imperial Patterned Bridal Lehenga	44999
top wear	Orange Maroon Embroidered Bridal Lehenga	44999

The Information panel shows the table structure for table2:

```

Columns:
sno          int
color1       text
color2       text
color3       text
color4       text
WearType     text
MaterialType text
reviews      int

```

The Output panel shows the execution message: "SELECT table2.WearType, table1.Productname, table1.Price FROM table2 JOIN table1... 30 row(s) returned".

5) Write a query to Calculate the Total Price for Each Material Type.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

1 SELECT table2.MaterialType, SUM(table1.Price) AS Total_Price
2 FROM table1
3 JOIN table2 ON table1.sno = table2.sno
4 GROUP BY table2.MaterialType;

```

The Result Grid displays the following data:

MaterialType	Total_Price
Cotton	3509539
Viscose	160356
Viscose Blend	50664
Cotton Blend	273402
Linen Blend	25856

The Information panel shows the table structure for table2:

```

Columns:
sno          int
color1       text
color2       text
color3       text
color4       text
WearType     text
MaterialType text
reviews      int

```

The Output panel shows the execution message: "SELECT table2.MaterialType, SUM(table1.Price) AS Total\_Price FROM table1 JOIN table2... 71 row(s) returned".



## TABLE 3

1) Write a query to Order Rows by Return Time in Ascending Order.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'clothing' database selected. The 'Tables' list includes 'table1', 'table2', 'table3', 'Views', 'Stored Procedures', and 'Functions'. The 'Information' tab is active, showing details for 'Table: table1' with columns: 'sno' (int), 'Brandname' (text), 'Category' (text), 'Producttype' (text), 'Productname' (text), and 'Price' (int). The main editor window shows a SQL query:   

```
-- Write a query to Order Rows by Return Time in Ascending Order
select Return_time from table3
order by Return_time asc
```

The 'Result Grid' is visible, showing the 'Return\_time' column with values: 'Assured Quality', 'Assured Quality', 'Easy 10 days return and exchange', 'Easy 10 days return and exchange', and 'Easy 10 days return and exchange'. The 'Output' tab shows the 'Action Output' with a table of results:

#	Time	Action	Message	Duration / Fetch
1	18:10:24	SELECT t1.Producttype,t3.Country_of_origin,t3.Return_time,t3.Occasion FROM table...	4311 row(s) returned	0.047 sec / 0.000 sec
2	18:38:14	SELECT * FROM clothing table2 LIMIT 0, 5000	4316 row(s) returned	0.000 sec / 0.016 sec
3	18:38:18	SELECT * FROM clothing table3 LIMIT 0, 5000	4316 row(s) returned	0.000 sec / 0.031 sec
4	18:39:10	select Return_time from table3 order by Return_time asc LIMIT 0, 5000	4316 row(s) returned	0.015 sec / 0.000 sec

2) Write a query to find distinct country of origin.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'clothing' database selected. The 'Tables' list includes 'table1', 'table2', 'table3', 'Views', 'Stored Procedures', and 'Functions'. The 'Information' tab is active, showing details for 'Table: table1' with columns: 'sno' (int), 'Brandname' (text), 'Category' (text), 'Producttype' (text), 'Productname' (text), and 'Price' (int). The main editor window shows a SQL query:   

```
-- Write a query to find distinct country of origin .
SELECT distinct(Country_of_origin) FROM table3
```

The 'Result Grid' is visible, showing the 'Country\_of\_origin' column with the value 'India'. The 'Output' tab shows the 'Action Output' with a table of results:

#	Time	Action	Message	Duration / Fetch
1	12:18:20	SELECT * FROM table3 order by Return_time asc LIMIT 0, 5000	4316 row(s) returned	0.031 sec / 0.000 sec
2	12:19:14	SELECT distinct(Country_of_origin) FROM table3 LIMIT 0, 5000	1 row(s) returned	0.015 sec / 0.000 sec

### 3) Write a query to Calculate the Average Return Time by Country of Origin

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'clothing' selected. The main editor shows a SQL query:

```
1 SELECT
2   country_of_origin,
3   AVG(LENGTH(return_time)) AS average_return_time_length
4 FROM
5   table3
6 GROUP BY
7   country_of_origin;
```

The 'Result Grid' at the bottom shows the following data:

country_of_origin	average_return_time_length
India	25.3800

### 4) Write a query to Find Occasions with High Return Times.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'clothing' selected. The main editor shows a SQL query:

```
1 -- Write a query to Find Occasions with High Return Times.
2 SELECT
3   Occasion,
4   Return_time
5 FROM
6   table3
7 WHERE
8   return_time > ('Easy 10 days return and exchange');
```

The 'Result Grid' at the bottom shows the following data:

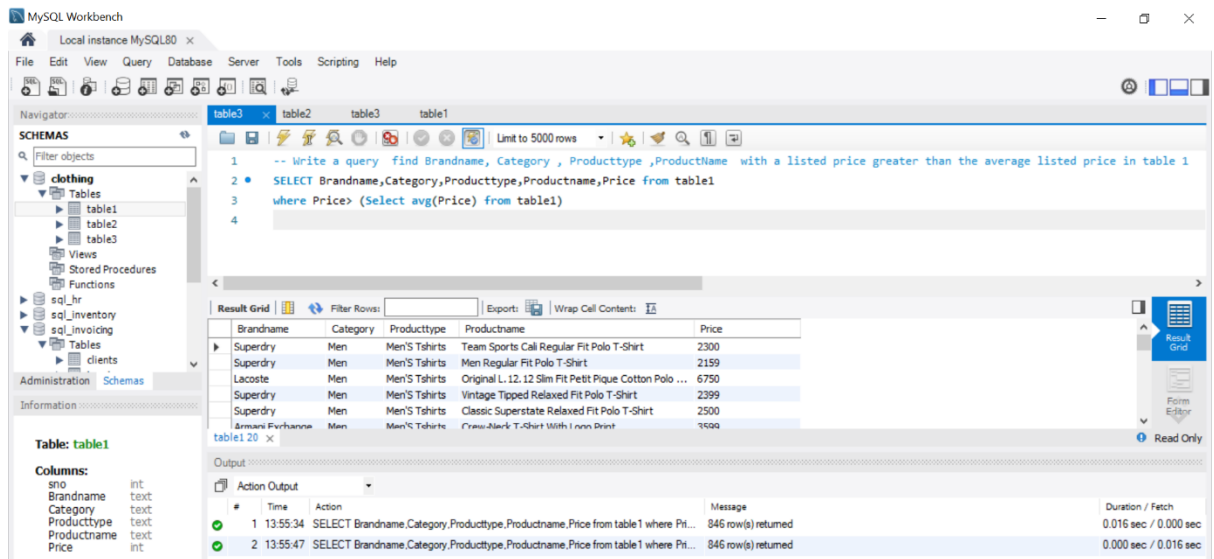
Occasion	Return_time
Regular	Easy 30 days returns
Work	Easy 30 days returns
Regular	Easy 30 days returns
Holiday	Easy 30 days returns
Regular	Easy 30 days returns
Regular	Easy 30 days returns
Regular	Easy 30 days returns
Everyday	Easy 30 days returns
Everyday	Easy 30 days returns
Regular	Easy 30 days returns
Everyday	Easy 30 days returns

[illegible]



## JOIN QUERY using table1,table2 and table3

- 1) Write a query find Brandname, Category , Producttype ,ProductName with a listed price greater than the average listed price in table 1.



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
-- Write a query find Brandname, Category , Producttype ,ProductName with a listed price greater than the average listed price in table 1
SELECT Brandname,Category,Producttype,Productname,Price from table1
where Price> (Select avg(Price) from table1)
```

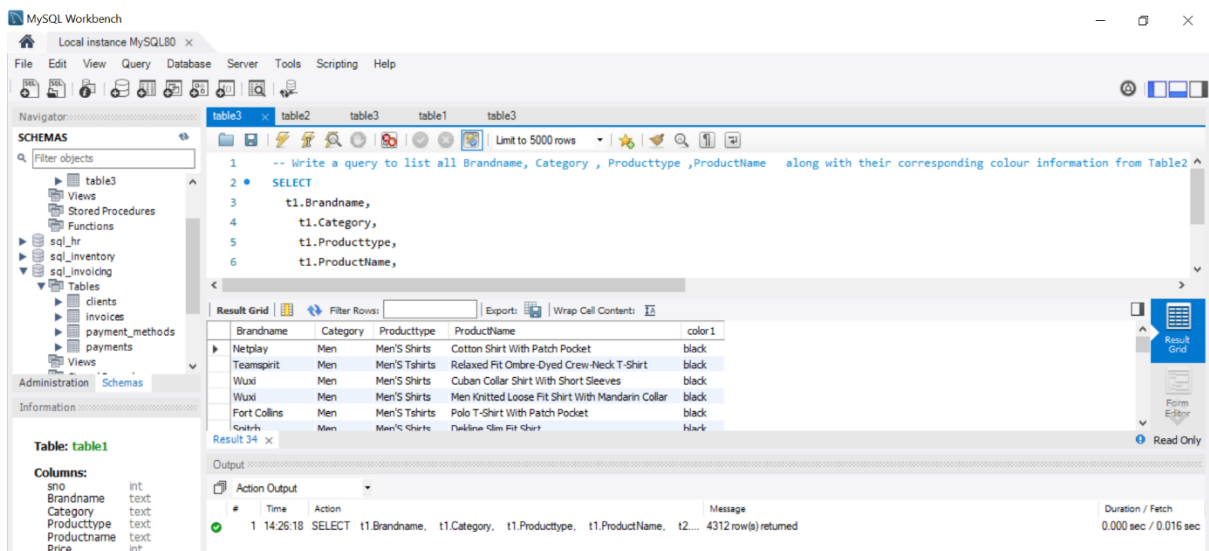
The result grid displays the following data:

Brandname	Category	Producttype	Productname	Price
Superdry	Men	Men'S Tshirts	Team Sports Call Regular Fit Polo T-Shirt	2300
Superdry	Men	Men'S Tshirts	Men Regular Fit Polo T-Shirt	2199
Lacoste	Men	Men'S Tshirts	Original L. 12.12 Slim Fit Petit Pique Cotton Polo ...	6750
Superdry	Men	Men'S Tshirts	Vintage Tipped Relaxed Fit Polo T-Shirt	2399
Superdry	Men	Men'S Tshirts	Classic Superstate Relaxed Fit Polo T-Shirt	2500
Armani Exchange	Men	Men'S Tshirts	Crew-Neck T-Shirt With Iron Print	2000

The output section shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	13:55:34	SELECT Brandname,Category,Producttype,Productname,Price from table1 where Pri...	846 row(s) returned	0.016 sec / 0.000 sec
2	13:55:47	SELECT Brandname,Category,Producttype,Productname,Price from table1 where Pri...	846 row(s) returned	0.000 sec / 0.016 sec

- 2) Write a query to list all Brandname, Category , Producttype ,ProductName along with their corresponding colour information from Table2



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
-- Write a query to list all Brandname, Category , Producttype ,ProductName along with their corresponding colour information from Table2
SELECT
t1.Brandname,
t1.Category,
t1.Producttype,
t1.ProductName,
```

The result grid displays the following data:

Brandname	Category	Producttype	ProductName	color1
Netplay	Men	Men'S Shirts	Cotton Shirt With Patch Pocket	black
Teamspirit	Men	Men'S Tshirts	Relaxed Fit Ombre-Dyed Crew-Neck T-Shirt	black
Wuxi	Men	Men'S Shirts	Cuban Collar Shirt With Short Sleeves	black
Wuxi	Men	Men'S Shirts	Men Knitted Loose Fit Shirt With Mandarin Collar	black
Fort Collins	Men	Men'S Tshirts	Polo T-Shirt With Patch Pocket	black
Grubbs	Men	Men'S Shirts	Dark Blue Slim Fit Shirt	black

The output section shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	14:26:18	SELECT t1.Brandname, t1.Category, t1.Producttype, t1.ProductName, t2...	4312 row(s) returned	0.000 sec / 0.016 sec

3) Write a query to find the average reviews for each Product type, ProductName using table 1 and table 2

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```

1  -- Write a query to find the average reviews for each Product type, ProductName using table 1 and table 2
2  • SELECT t1.Producttype,t1.Productname,t2.reviews
3  FROM table1 t1
4  JOIN table2 t2
5  ON t1.sno = t2.sno
6  WHERE t2.reviews >= (SELECT AVG(reviews) FROM table2);
7

```

The result grid shows the following data:

Producttype	Productname	reviews
Men'S Shirts	Cotton Shirt With Patch Pocket	268
Men'S Shirts	Cuban Collar Shirt With Short Sleeves	441
Men'S Shirts	Men Knitted Loose Fit Shirt With Mandarin Collar	321
Men'S Tshirts	Polo T-Shirt With Patch Pocket	320
Men'S Shirts	Dekline Slim Fit Shirt	463

The output shows 44 rows returned. The duration of the query was 0.031 sec / 0.000 sec.

4)Write a query to find products with a Product name, material type that matches the most common material type in (use table1 and table 2)

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```

1  -- Write a query to find products with a Product name, material type that matches the most common material type in (use table1 and table 2)
2  • SELECT t1.Productname,t2.Materialtype
3  FROM table1 t1
4  JOIN table2 t2
5  ON t1.sno = t2.sno
6  WHERE Materialtype = (
7      SELECT Materialtype
8      FROM table2
9      GROUP BY Materialtype
10     ORDER BY COUNT(*) DESC
11 );
12

```

The result grid shows the following data:

Productname	Materialtype
Cotton Shirt With Patch Pocket	Cotton
Relaxed Fit Ombre-Dyed Crew-Neck T-Shirt	Cotton
Cuban Collar Shirt With Short Sleeves	Cotton
Men Knitted Loose Fit Shirt With Mandarin Collar	Cotton
Polo T-Shirt With Patch Pocket	Cotton
Dekline Slim Fit Shirt	Cotton
Regular Fit Printed Polo T-Shirt	Cotton
Regular Fit Printed Polo T-Shirt	Cotton
Spread Collar Shirt With Patch Pocket	Cotton
Checked Shirt With Patch Pocket	Cotton

The output shows 47 rows returned. The duration of the query was 0.031 sec / 0.000 sec.

[illegible][illegible]

7) Write a query to find Country of origin, return time, occasion of each product type. (use table 1 and table 3)

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 -- Write a query to find Country of origin, return time, occasion of each product type.(use table 1 and table 3)
2 SELECT t1.Producttype,t3.Country_of_origin,t3.Return_time,t3.Occasion
3 FROM table1 t1
4 JOIN table3 t3
5 ON t1.sno = t3.sno
6 ORDER BY t1.Producttype
```

The query results are displayed in the Result Grid, showing 4 rows of data:

Producttype	Country_of_origin	Return_time	Occasion
Dresses And Jumpsuits	India	Easy 30 days returns	Everyday
Dresses And Jumpsuits	India	Easy 30 days returns	Regular
Dresses And Jumpsuits	India	Easy 30 days returns	Regular
Dresses And Jumpsuits	India	Easy 30 days returns	Regular

The left sidebar shows the Schemas tree with the 'clothing' database selected. The 'Table: table1' information panel is visible, listing columns: sno (int), Brandname (text), Category (text), Producttype (text), and Price (int).

The bottom panel shows the Action Output, indicating that 4311 row(s) were returned by the query.