

QUESTIONS

Tabel 1:

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

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including 'clothing_brands' and 'data_trained'. The main query editor displays the following SQL query:

```
1 • SELECT * FROM table1 ORDER BY price DESC LIMIT 20;
```

The 'Result Grid' pane shows the results of the query, displaying columns: SL no, Brand name, category, product type, product name, product code, and price. The results are sorted by price in descending order, showing the top 20 most expensive products.

SL no	Brand name	category	product type	product name	product code	price
1136	ANDAMEN	WOMEN	women	Regular Fit Button-Down Shirt with Embroidery	467209000000	4990
1170	ANDAMEN	WOMEN	women	Men Regular Fit Polo T-Shirt	467037000000	3590
1132	GUESS	WOMEN	women	Brand Print Regular Fit T- Shirt	469546000000	3490
1352	ANDAMEN	WOMEN	women	Men Logo Embroidered Regular Fit Shirt with Pa...	467024000000	2990
1329	ANDAMEN	WOMEN	women	Men Logo Embroidered Regular Fit Shirt with Pa...	467024000000	2990
1560	U.S. POLO ASSN.	WOMEN	women	Men Brand Embroidered Slim Polo T-Shirt	442401000000	2699
1561	U.S. POLO ASSN.	WOMEN	women	Men Brand Embroidered Slim Polo T-Shirt	442401000000	2699
1100	LEE	WOMEN	women	Men Checked Regular Fit Shirt with Button-Closure	467138000000	2599
488	DENNISLINGO PREMIUM ATTIRE	WOMEN	women	Striped Slim Fit Shirt	462153000000	2499
7	DENNISLINGO PREMIUM ATTIRE	MEN	shirt	Striped Slim Fit Shirt	462153000000	2499

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

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including 'clothing_brands' and 'data_trained'. The main query editor displays the following SQL query:

```
1 SELECT Category, SUM(price) AS TotalPrice
2 FROM table1
3 GROUP BY Category;
```

The 'Result Grid' pane shows the results of the query, displaying columns: Category and TotalPrice. The results show the total price for each category: MEN (985922) and WOMEN (1185061).

Category	TotalPrice
MEN	985922
WOMEN	1185061

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

Query 1 | table1 x table1

Limit to 1000 rows

```

1 • SELECT * FROM table1
2 WHERE price BETWEEN 2000 AND 5000;
3

```

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

	SL no	Brand name	category	product type	product name	product code	price
▶	7	DENNISLINGO PREMIUM ATTIRE	MEN	shirt	Striped Slim Fit Shirt	462153000000	2499
	14	ZIKARAA	MEN	shirt	Waffles Pattern Loose Fit Shirt with Patch Pocket	467094000000	2299
	487	GODFREY	WOMEN	women	Regular Fit Striped Polo T-Shirt	466332000000	2499
	488	DENNISLINGO PREMIUM ATTIRE	WOMEN	women	Striped Slim Fit Shirt	462153000000	2499
	494	THE HOLLANDER	WOMEN	women	Men Graphic Print Crew-Neck Oversized T-Shirt	466218000000	2299
	495	ZIKARAA	WOMEN	women	Waffles Pattern Loose Fit Shirt with Patch Pocket	467094000000	2299
	1100	LEE	WOMEN	women	Men Checked Regular Fit Shirt with Button-Closure	467138000000	2599
	1132	GUESS	WOMEN	women	Brand Print Regular Fit T- Shirt	469546000000	3490
	1136	ANDAMEN	WOMEN	women	Regular Fit Button-Down Shirt with Embroidery	467209000000	4990
	1170	ANDAMEN	WOMEN	women	Men Regular Fit Polo T-Shirt	467037000000	2500

table14 x

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

Navigator: | Query 1 | table1 x table1 | table1 | clothing_brands.table1 | table1 | table1

Limit to 1000 rows

```

1 • SELECT Brand_name , AVG(price) AS Avg_Price FROM table1
2 GROUP BY Brand_name ;
3
4
5

```

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

	Brand_name	Avg_Price
▶	BOMBAY BEGUM	448.8235
	BULLMER	405.3333
	GODFREY	1149.0000
	DENNISLINGO PREMIUM ATTIRE	804.4783
	PUMA	845.6250
	EYEBGLER	267.7143
	WOXEN	448.3333
	THE HOLLANDER	452.6486
	ZIKARAA	975.6667
	TOM HIDDLE	322.3750

Result 5 x

SCHEMAS

Filter objects

- clothing_brands
 - Tables
 - table1
 - table2
 - table3
 - Views
 - Stored Procedures
 - Functions
- data_trained
 - Tables
 - employee
 - rm_mapping_master
 - rm_master
 - sales
 - sales_master
 - Views
 - Stored Procedures
 - Functions
- sakila
- sys
- world

Administration | Schemas

Information

Schema: clothing_brands

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

Query 1

table1 x table1 table1

Limit to 1000 rows

```

1 • SELECT Brand_name, category, COUNT(*) AS Count FROM table1
2   GROUP BY Brand_name, Category HAVING Count = 1;
3
4
5
6

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Brand_name	category	Count
VERTUSY	MEN	1
TEE TOWN	MEN	1
7SHORES	MEN	1
PAUL STREET	MEN	1
YOVISH	MEN	1
A23 LIFESTYLE	MEN	1
TRENDS TOWER	MEN	1
POIPET	MEN	1
NOBLE MONK	MEN	1
JAINISH	MEN	1

Result 6 x

Output

Table2

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

Query 1

table1 table2 x table2

Limit to 1000 rows

```

1 • SELECT * FROM table2 ORDER BY `material type` ASC;
2

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

SL no	wear type	material type	colour 1	colour 2	colour 3	colour 4	reviews
5785	bottom wear	Cambric	blue	Grey	White	white	7525
5848	bottom wear	Cambric	blue	Grey	White	white	1807
5817	bottom wear	Cambric	blue	Grey	White	white	9137
5862	bottom wear	Cambric	blue	Grey	White	white	7369
5879	bottom wear	Cambric	blue	Grey	White	white	1948
5160	bottom wear	Chiffon	blue	Grey	White	white	7484
4823	bottom wear	Chiffon	blue	Grey	White	white	3183
4939	bottom wear	Chiffon	blue	Grey	White	white	9130
5012	bottom wear	Chiffon	blue	Grey	White	white	1032
5180	bottom wear	Chiffon	blue	Grey	White	white	0077

table2 2 x

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

Query 1 | table1 | table2 | table2

Limit to 1000 rows

```
1 • SELECT AVG(reviews) AS AverageReviews FROM table2;
2
3
```

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

AverageReviews
2334.6077

Result 3 x

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

Query 1 | table1 | table2 | table2 | table1

Limit to 1000 rows

```
1 • SELECT table2.`Material type`, SUM(table1.price) AS TotalPrice
2 FROM table1, table2
3 WHERE table1.`SL no` = table2.`SL no`
4 GROUP BY table2.`material type`;
5
6
```

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

Material type	TotalPrice
cotton	1006429
Regular Fit	46776
polycotton	44120
polyester	224931
cottob blend	374813
Satin	4426
Poly lycra	1843
rayon	10256
Pure cotton	264146
lycra	1078

Result 2 x

Output

Table 3.csv

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

The screenshot shows a database query editor with a query window and a result grid. The query window contains the following SQL query:

```
1 • SELECT * FROM table3 ORDER BY 'return_time' ASC;
```

The result grid displays the following data:

	SL no	occasion	return_time	country_origin	size1	size2	size3	size4
	10	casual	10	India	S	M	XS	XL
	11	casual	10	India	XS	M	XL	XXL
	12	casual	10	India	XS	M	XL	XXL
	13	casual	10	India	XS	M	XL	XXL
	14	casual	10	India	XS	M	XL	XXL
	15	casual	10	India	XS	M	XL	XXL
	16	casual	10	India	S	M	L	XL
	17	casual	10	India	XS	M	XL	XXL
	18	casual	10	India	S	M	L	XL
	19	casual	10	India	S	M	L	XL

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

The screenshot shows a database query editor with a query window and a result grid. The query window contains the following SQL query:

```
1 • SELECT distinct country_origin FROM table3;
```

The result grid displays the following data:

	country_origin
▶	India
	china
	pennsylvania
	america
	europa
	united kingdom
	italy
	germany
	argentina
	singapore

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

The screenshot shows a SQL query editor with the following query:

```
1 • SELECT country_origin, AVG(return_time) FROM table3 GROUP BY country_origin;
```

Below the query editor is a result grid showing the output of the query:

country_origin	AVG(return_time)
India	13.8778
china	10.0000
pennsylvania	10.0000
america	10.0000
europa	10.0000
united kingdom	10.0000
italy	10.0000
germany	10.0000
argentina	10.0000
singapore	10.0000

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

The screenshot shows a SQL query editor with the following query:

```
1 • SELECT occasion, return_time FROM table3 WHERE return_time > 10;
```

Below the query editor is a result grid showing the output of the query:

occasion	return_time
Everyday	30
Work	30
Everyday	30
Holiday	30
Everyday	30
Everyday	30
Celebration	30
Everyday	30
Everyday	30
Everyday	30

5) Write a query to Retrieve Products with Multiple Sizes (Size1, Size2, Size3, Size4):

The screenshot shows a database query editor with a query window and a result grid. The query window contains the following SQL code:

```

1 • SELECT *
2 FROM table3
3 WHERE Size1 != '-----'
4 AND Size2 != '-----'
5 AND Size3 != '-----'
6 AND Size4 != '-----';

```

The result grid displays the following data:

SL_no	occasion	return_time	country_origin	size1	size2	size3	size4
1	casual	10	India	S	M	L	XL
2	casual	10	India	S	M	L	XL
3	casual	10	India	XS	M	XL	XXL
4	casual	10	India	S	M	L	XL
5	casual	10	India	S	M	L	XL
6	regular	10	India	XS	M	XL	XXL
7	casual	10	India	XS	M	XL	XXL
8	casual	10	India	S	M	XS	XL
9	casual	10	china	XS	M	XL	XXL
10	casual	10	India	S	M	XS	XL

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 a database query editor with a query window and a result grid. The query window contains the following SQL code:

```

1 • SELECT table1.Brand_name, table1.category, table1.product_type, table1.product_name
2 from table1 join ( SELECT AVG(price) as avgprice from table1) avg_prices ON
3 table1.price > avg_prices.avgprice ;

```

The result grid displays the following data:

Brand_name	category	product_type	product_name
GODFREY	MEN	T- shirts	Regular Fit Striped Polo T-Shirt
DENNISLINGO PREMIUM ATTIRE	MEN	shirt	Striped Slim Fit Shirt
PUMA	MEN	T- shirts	Men Ess Slim Fit Polo T-Shirt
THE HOLLANDER	MEN	T- shirts	Men Graphic Print Crew-Neck Oversized T-Shirt
ZIKARAA	MEN	shirt	Waffles Pattern Loose Fit Shirt with Patch Pocket
DENNISLINGO PREMIUM ATTIRE	MEN	shirt	Checked Slim Fit Shirt
VERTUSY	MEN	shirt	Men Regular Fit Shirt with Front Patch Pocket
BADMAASH	MEN	shirt	Men Slim Fit Shirt with Spread Collar
BADMAASH	MEN	shirt	Slim Fit Shirt with Spread Collar
CLAFOUTIS	MEN	T- shirts	Men Relaxed Fit Polo T-Shirt
DENNISLINGO PREMIUM ATTIRE	MEN	shirt	Spread Collar Slim Fit Shirt

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

table3 table1 x table2 table1

Limit to 10000 rows

```

1 • SELECT t1.Brand_name, t1.category, t1.product_type, t1.product_name,
2     t2.colour_1, t2.colour_2, t2.colour_3, t2.colour_4
3 FROM Table1 t1
4 JOIN Table2 t2 ON t1.SL_no= t2.SL_no;
5

```

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

	Brand_name	category	product_type	product_name	colour_1	colour_2	colour_3	colour_4
▶	BOMBAY BEGUM	MEN	shirt	Men Regular Fit Shirt with Cutaway Collar	black	brown	White	Grey
	BULLMER	MEN	T- shirts	Loose Fit Graphic Print Round-Neck T-Shirt	black	brown	White	Grey
	BOMBAY BEGUM	MEN	shirt	Spread Collar Shirt with Patch Pocket	black	brown	White	Grey
	BOMBAY BEGUM	MEN	shirt	Men Regular Fit Shirt with Short Sleeves	black	brown	White	Grey
	BULLMER	MEN	T- shirts	Loose Fit Graphic Print Crew-Neck T-Shirt	black	brown	White	Grey
	GODFREY	MEN	T- shirts	Regular Fit Striped Polo T-Shirt	black	brown	White	Grey
	DENNISLINGO PREMIUM ATTIRE	MEN	shirt	Striped Slim Fit Shirt	black	brown	White	Grey
	BULLMER	MEN	T- shirts	Graphic Oversized Fit T-shirt	black	brown	White	Grey
	PUMA	MEN	T- shirts	Men Ess Slim Fit Polo T-Shirt	black	brown	White	Grey
	EYEBGLER	MEN	T- shirts	Regular Fit Checked Print Polo T-Shirt	black	brown	White	Grey
	WOXEN	MEN	shirt	Spread Collared Checked Shirt	black	brown	White	Grey

Result 4 x

Output

- 3) Write a query to find the **average reviews** for each Producttype ,ProductName using table 1 and table 2

table3 table1 x table2 table1

Limit to 10000 rows

```

1 • SELECT t1.product_type, t1.product_name, AVG(t2.reviews) AS AverageReviews
2 FROM Table1 t1
3 JOIN Table2 t2 ON t1.SL_no = t2.SL_no
4 GROUP BY t1.product_type, t1.product_name;
5

```

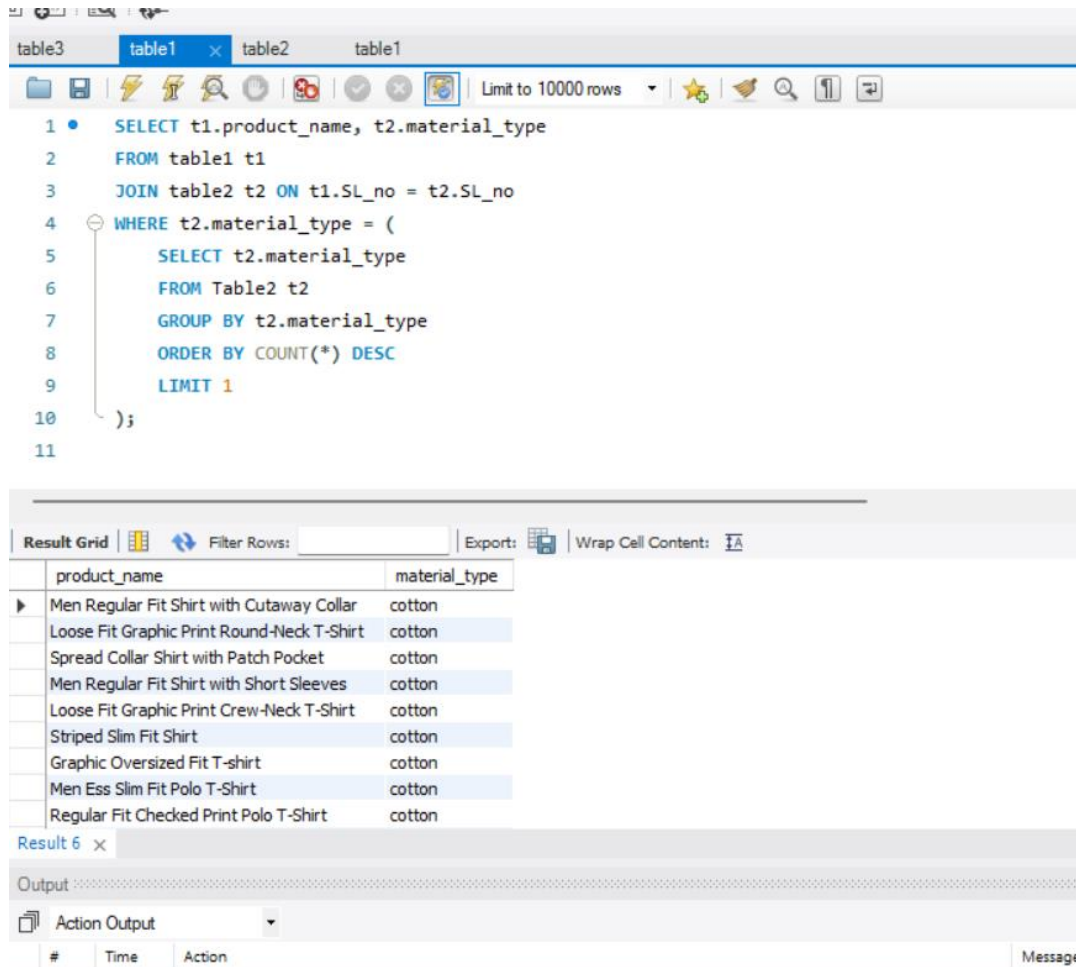
Result Grid | Filter Rows: | Exports: | Wrap Cell Content: |

	product_type	product_name	AverageReviews
▶	shirt	Men Regular Fit Shirt with Cutaway Collar	113.0000
	T- shirts	Loose Fit Graphic Print Round-Neck T-Shirt	267.0000
	shirt	Spread Collar Shirt with Patch Pocket	109.2500
	shirt	Men Regular Fit Shirt with Short Sleeves	219.0000
	T- shirts	Loose Fit Graphic Print Crew-Neck T-Shirt	222.0000
	T- shirts	Regular Fit Striped Polo T-Shirt	298.8000
	shirt	Striped Slim Fit Shirt	368.0000
	T- shirts	Graphic Oversized Fit T-shirt	459.0000
	T- shirts	Men Ess Slim Fit Polo T-Shirt	319.0000
	T- shirts	Regular Fit Checked Print Polo T-Shirt	144.0000
	shirt	Spread Collared Checked Shirt	480.0000

Result 5 x

Output

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 a SQL IDE with a query editor and a result grid. The query is as follows:

```

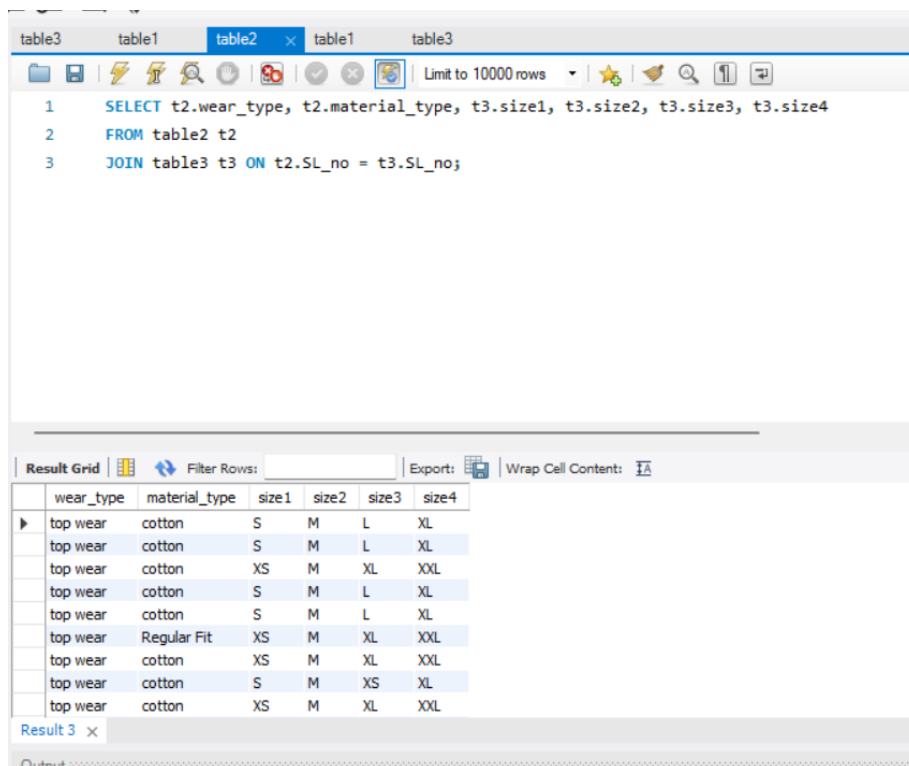
1 SELECT t1.product_name, t2.material_type
2 FROM table1 t1
3 JOIN table2 t2 ON t1.SL_no = t2.SL_no
4 WHERE t2.material_type = (
5     SELECT t2.material_type
6     FROM Table2 t2
7     GROUP BY t2.material_type
8     ORDER BY COUNT(*) DESC
9     LIMIT 1
10 );
11

```

The result grid shows the following data:

product_name	material_type
Men Regular Fit Shirt with Cutaway Collar	cotton
Loose Fit Graphic Print Round-Neck T-Shirt	cotton
Spread Collar Shirt with Patch Pocket	cotton
Men Regular Fit Shirt with Short Sleeves	cotton
Loose Fit Graphic Print Crew-Neck T-Shirt	cotton
Striped Slim Fit Shirt	cotton
Graphic Oversized Fit T-shirt	cotton
Men Ess Slim Fit Polo T-Shirt	cotton
Regular Fit Checked Print Polo T-Shirt	cotton

- 5) Write a query to list all products Weartype , Material type along with their corresponding **sizes** from (use table 2 and table 3)



The screenshot shows a SQL IDE with a query editor and a result grid. The query is as follows:

```

1 SELECT t2.wear_type, t2.material_type, t3.size1, t3.size2, t3.size3, t3.size4
2 FROM table2 t2
3 JOIN table3 t3 ON t2.SL_no = t3.SL_no;

```

The result grid shows the following data:

wear_type	material_type	size1	size2	size3	size4
top wear	cotton	S	M	L	XL
top wear	cotton	S	M	L	XL
top wear	cotton	XS	M	XL	XXL
top wear	cotton	S	M	L	XL
top wear	cotton	S	M	L	XL
top wear	Regular Fit	XS	M	XL	XXL
top wear	cotton	XS	M	XL	XXL
top wear	cotton	S	M	XS	XL
top wear	cotton	XS	M	XL	XXL

6) Write a query to find the **average return time** for each product type (use table1 and

table 3)

The screenshot shows a database query editor with a tab labeled 'table1' selected. The query is as follows:

```
1 • SELECT t1.product_type, AVG(t3.return_time) AS Average_Return_Time
2 FROM table1 t1
3 JOIN table3 t3 ON t1.SL_no= t3.SL_no
4 GROUP BY t1.product_type;
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query:

product_type	Average_Return_Time
shirt	10.8427
T- shirts	10.0000
women	10.0000
kurta	30.0000
trouser	30.0000
churidar	11.0127
shorts	30.0000
dress	10.1563
salwar	30.0000

The result grid is labeled 'Result 7'.

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 a database query editor with a tab labeled 'table1' selected. The query is as follows:

```
1 • SELECT t1.product_type, t3.country_origin, t3.return_time, t3.occasion
2 FROM table1 t1
3 JOIN table3 t3 ON t1.SL_no = t3.SL_no;
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query:

product_type	country_origin	return_time	occasion
shirt	India	10	casual
T- shirts	India	10	casual
shirt	India	10	casual
shirt	India	10	casual
T- shirts	India	10	casual
T- shirts	India	10	regular
shirt	India	10	casual
T- shirts	India	10	casual
T- shirts	china	10	casual

The result grid is labeled 'Result 10'.