

# Untitled

1) Find the products with the highest number of units received

```
SELECT ProductId, MAX(Number_Received) AS Max_Received  
FROM purchases  
GROUP BY ProductId  
ORDER BY Max_Received DESC;
```

The screenshot shows a database query results interface. At the top, there are several lines of code with line numbers 49 through 56. Lines 51 through 54 are highlighted in blue, indicating they are the executed query. Below the code is a progress bar at 10% completion and a timestamp of 1:246. The main area displays a 'Result Grid' with two columns: 'ProductId' and 'Max\_Received'. The data is as follows:

ProductId	Max_Received
12	50
27	35
23	30
16	28
18	25
15	23
13	22
20	22
33	22
17	21
32	20

Below the grid, it says 'Result 1'. To the right of the grid, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Types'. At the bottom right of the interface, there is a 'Read Only' button.

2) Find the total quantity of products ordered in each order, and the associated product and order.

```
SELECT o.OrderId, o.Order_Date, p.ProductName, SUM(pu.Number_Received) AS TotalOrdered  
FROM orders o  
JOIN purchases pu ON o.ProductId = pu.ProductId  
JOIN products p ON o.ProductId = p.id  
GROUP BY o.OrderId, o.Order_Date, p.ProductName  
ORDER BY o.OrderId;
```

```
259 GROUP BY Purchase_date;
260
261
262
263 • SELECT o.OrderId, o.Order_Date, p.ProductName, SUM(pu.Number_Received) AS TotalOrdered
264 FROM orders o
265 JOIN purchases pu ON o.ProductId = pu.ProductId
266 JOIN products p ON o.ProductId = p.id
267 GROUP BY o.OrderId, o.Order_Date, p.ProductName
268 ORDER BY o.OrderId;
269
270
```

3) To know the top spender of products?

```
SELECT p.ProductName, SUM(o.Number_Shipped) AS TotalOrders
FROM products p
LEFT JOIN orders o ON p.id = o.ProductId
GROUP BY p.ProductName
ORDER BY TotalOrders DESC
LIMIT 1;
```

```
347
348
349 •   SELECT p.ProductName, SUM(o.Number_Shipped) AS TotalOrders
350     FROM products p
351     LEFT JOIN orders o ON p.id = o.ProductId
352     GROUP BY p.ProductName
353     ORDER BY TotalOrders DESC
354     LIMIT 1;
355
356
357
358
```

100% 1:357

Result Grid Filter Rows:  Search Export: Fetch rows:

... Tot...  
Hi NULL

Result 1 Read Only

4)Calculate the average starting inventory of all products.

```
SELECT AVG(p.StartingInventory) AS AverageStartingInventory
FROM products p
JOIN (SELECT DISTINCT StartingInventory FROM products) AS subquery
ON p.StartingInventory = subquery.StartingInventory;
```

The screenshot shows a database interface with the following details:

- Code area:

```
362
363 ✘ SELECT AVG(p.StartingInventory) AS AverageStartingInventory
364   FROM products p
365   JOIN (SELECT DISTINCT StartingInventory FROM products) AS subquery
366     ON p.StartingInventory = subquery.StartingInventory;
367
368
369
```
- Status bar: 100% | 67:365 | 1 error found
- Result Grid: Shows a single row with the value 188491.7818.
- Toolbar: Result Grid, Filter Rows, Search, Export, Read Only
- Right sidebar: Result Grid, Form Editor

5)List the purchases made on or after 2023-12-15.Find the latest purchase date in the table.

```
SELECT p.Purchase_Date
FROM purchases p
WHERE p.Purchase_Date >= '2023-12-15'
UNION
SELECT MAX(Purchase_Date) AS LatestPurchaseDate
FROM purchases;
```

```
426
427 •  SELECT p.Purchase_Date
428   FROM purchases p
429   WHERE p.Purchase_Date >= '2023-12-15'
430   UNION
431   SELECT MAX(Purchase_Date) AS LatestPurchaseDate
432   FROM purchases;
433
434
435
```

100% 6:430 | 1 error found

Result GridFilter Rows:  SearchExport:

Purchase_Date
▶ 2023-12-15
2023-12-16
2023-12-17
2023-12-18
2023-12-19
2023-12-20
2023-12-21
2023-12-22
2023-12-23
2023-12-24

Result 1

Read Only

Action Output ▼

6) Calculate the total quantity received for ProductId 20.

```
SELECT p.ProductName, SUM(pr.Number_Received) AS TotalQuantityReceived  
FROM products p  
LEFT JOIN purchases pr ON p.id = pr.ProductId  
WHERE p.id = 20  
GROUP BY p.ProductName;
```

```
477  
478  
479  
480 • SELECT p.ProductName, SUM(pr.Number_Received) AS TotalQuantityReceived  
481 FROM products p  
482 LEFT JOIN purchases pr ON p.id = pr.ProductId  
483 WHERE p.id = 20  
484 GROUP BY p.ProductName;  
485
```

100% 1:478 1 error found

Result Grid Filter Rows: Search Export:

... TotalQuantityReceiv...

Result 1 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
1	22:01:46	SELECT p.ProductName, SUM(pr.Numb...	0 row(s) returned	0.0011 sec / 0.00002...

7) How many products have a minimum required quantity of less than 5.

```

SELECT COUNT(p.id) AS ProductCount
FROM products p
LEFT JOIN purchases pu ON p.id = pu.ProductId
WHERE p.MinimumRequired < 5
GROUP BY p.id ;

```

343  
344 • **SELECT COUNT(p.id) AS ProductCount**  
345 **FROM products p**  
346 **LEFT JOIN purchases pu ON p.id = pu.ProductId**  
347 **WHERE p.MinimumRequired < 5|**  
348 **GROUP BY p.id;**  
349  
350  
351

100% 28:347 1 error found

Result Grid Filter Rows: Search Export:

ProductCount
1
1
1
1

Result 1 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
1	22:07:39	SELECT COUNT(p.id) AS ProductCount...	4 row(s) returned	0.0011 sec / 0.00001...

8) Find products with inventory on hand greater than 100 and ordered more than 500.

```

SELECT p.ProductName, SUM(o.Number_Shipped) AS TotalShipped
FROM products p
RIGHT JOIN orders o ON p.id = o.ProductId
WHERE p.InventoryOnHand > 100 AND o.Number_Shipped > 500
GROUP BY p.ProductName;

```

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a code editor with the following SQL query:

```

337
338 • SELECT p.ProductName, SUM(o.Number_Shipped) AS TotalShipped
339   FROM products p
340   RIGHT JOIN orders o ON p.id = o.ProductId
341 WHERE p.InventoryOnHand > 100 AND o.Number_Shipped > 500
342 GROUP BY p.ProductName;
343
344 • SELECT COUNT(p.id) AS ProductCount
345   FROM products p

```

The status bar at the bottom left indicates "100%" completion, "1:343" execution time, and "1 error found".

In the bottom-right pane, there is a "Result Grid" viewer showing a single row of data:

... TotalShipped

A tooltip "Result 1" is visible below the grid. To the right of the grid is a vertical toolbar with three items: "Result Grid" (selected), "Form Editor", and "Field Types".

At the bottom of the interface, there is an "Action Output" section with the following table:

Action	Time	Response	Duration / Fetch Time
SELECT p.ProductName, SUM(o.Number_Shipped) AS TotalShipped FROM products p RIGHT JOIN orders o ON p.id = o.ProductId WHERE p.InventoryOnHand > 100 AND o.Number_Shipped > 500 GROUP BY p.ProductName;	22:11:07	0 row(s) returned	0.00068 sec / 0.000...

A "Read Only" status indicator is also present.

9)Find the products that have not been ordered yet.

```

SELECT p.ProductName
FROM products p
LEFT JOIN orders o ON p.id = o.ProductId
GROUP BY p.ProductName
HAVING COUNT(o.ProductId) = 0;

```

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the following SQL code:

```

382
383
384 •  SELECT p.ProductName
385   FROM products p
386   LEFT JOIN orders o ON p.id = o.ProductId
387   GROUP BY p.ProductName
388   HAVING COUNT(o.ProductId) = 0;
389
390

```

The results grid displays the following data:

ProductName
Wipro
vacuum_cleaner
Under_Armour_Shirt
Toyota
Tesla_model3
Tata Motors
Tata Group
State Bank of India (SBI)
StarBucks_coffee
Sony_playStation
Sony_bravia_TV

Action Output

Time	Action	Response	Duration / Fetch Time
22:16:32	SELECT p.ProductName FROM product...	54 row(s) returned	0.0011 sec / 0.00003...

Result Grid

10) The LIMIT 1 ensures that we retrieve only the product with the highest total number of orders.

```

SELECT p.ProductName, SUM(o.Number_Shipped) AS TotalOrders
FROM products p
LEFT JOIN orders o ON p.id = o.ProductId
GROUP BY p.ProductName
ORDER BY TotalOrders DESC
LIMIT 5;

```

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```

478
479 • SELECT p.ProductName, SUM(o.Number_Shipped) AS TotalOrders
480   FROM products p
481   LEFT JOIN orders o ON p.id = o.ProductId
482   GROUP BY p.ProductName
483   ORDER BY TotalOrders DESC
484   LIMIT 5;
485

```

The results grid displays the following data:

ProductName	Tot...
HP_Envy_Laptop	NULL
iphone	NULL
DELL XPS	NULL
NotePad	NULL
Canon_camera	NULL

Action Output details:

	Time	Action	Response	Duration / Fetch Time
1	22:21:14	SELECT p.ProductName, SUM(o.Num... 5 row(s) returned		0.0012 sec / 0.00002...

11) Retrieve the products that were ordered on 2023-10-10.

```

SELECT p.ProductName, SUM(o.Number_Shipped) AS TotalOrders
FROM products p
INNER JOIN orders o ON p.id = o.ProductId
WHERE o.Order_Date = '2023-10-10'
GROUP BY p.ProductName;

```

12) List all products with names that contain "apple".

```
SELECT p.ProductName, COUNT(o.ProductId) AS TotalOrders
```

FROM products p

LEFT JOIN orders o ON p.id = o.ProductId

```
WHERE p.ProductName LIKE '%apple%'
```

GROUP BY p.ProductName;

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a query editor window titled "Query 3" containing the following SQL code:

```

404 FROM products p
405 LEFT JOIN orders o ON p.id = o.ProductId
406 WHERE p.ProductName LIKE '%apple%'
407 GROUP BY p.ProductName;
408
409
410
411 • SELECT p.ProductName, p.StartingInventory
412 FROM products p

```

The status bar at the bottom indicates "100% 1:411 | 1 error found".

In the bottom-right pane, there is a "Result Grid" showing the output of the query:

ProductName	Tot...
Apple_watch	0

The status bar at the bottom right of the result grid says "Read Only".

On the far right, there is a vertical sidebar with several icons and text:

- SELECT Syntax**
- SELECT** retrieves select more to include statements
- union, subquery**
- SELECT** can start with define expressions
- SELECT** access is used in the query
- Result Grid**
- Form Editor**
- Field Types**

13) Retrieve the orders placed on or before '2022-05-12'.

```

SELECT o.OrderId, o.Title, o.FirstName, o.MiddleName, o.LastName, o.ProductId,
p.ProductName, COUNT(*) AS TotalOrders
FROM orders o
JOIN products p ON o.ProductId = p.id
WHERE o.Order_Date <= '2022-05-12'
GROUP BY o.OrderId, o.Title, o.FirstName, o.MiddleName, o.LastName,
o.ProductId, p.ProductName
ORDER BY o.OrderId;

```

The screenshot shows the MySQL Workbench interface. The top navigation bar includes tabs for 'admin...', 'Schemas' (selected), 'Query 3', and 'Context...'. Below the navigation is a toolbar with various icons for database management. The main area displays a query editor with the following SQL code:

```
495
496 •  SELECT o.OrderId, o.Title, o.FirstName, o.MiddleName, o.LastName, o.ProductId, p.ProductName,
497   FROM orders o
498   JOIN products p ON o.ProductId = p.id
499   WHERE o.Order_Date <= '2022-05-12'
500   GROUP BY o.OrderId, o.Title, o.FirstName, o.MiddleName, o.LastName, o.ProductId, p.ProductName
501   ORDER BY o.OrderId;
502
```

Below the code, status information shows '100%' completion, '1:502' duration, and '1 error found'. The 'Result Grid' tab is selected, showing a table with columns: OrderId, Title, FirstName, MiddleName, LastName, ProductId, and ProductName. A tooltip indicates 'Tot...' for the last column. On the right, a sidebar offers options: 'Result Grid' (selected), 'Form Editor', and 'Field Types'. At the bottom, a 'Read Only' button is visible.

14) Retrieve the orders placed by customers with MiddleName 'Sanjay.'

SELECT \*

FROM orders

```
WHERE MiddleName = 'Sanjay';
```

15)

Calculate the total number of orders for each product label (grouped by ProductLabel).

```
SELECT p.ProductLabel, COUNT(o.OrderId) AS TotalOrders
FROM products p
LEFT JOIN orders o ON p.id = o.ProductId
GROUP BY p.ProductLabel;
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a text area displaying the SQL query. The results are shown in a grid below, with a vertical sidebar on the right containing tabs for 'Result Grid', 'Form Editor', and 'Field Types'. The bottom section shows the execution details.

Query results:

ProductLabel	Tot...
HP_Envy_Laptop_xx564	0
iphone_xx1000	0
DELL_XPS_sc875	0
NotePad_sxs555	0
Canon_camera_uk545	0
Google_pixel_bbb40	0
StarBucks_coffee_bxx999	0
Fitbit_versa_lk574	0
Boat_headphones_huk520	0
Rolex_Submariner_jj222	0
Sony_bravia_TV_hhh303	0

Action Output:

Time	Action	Response	Duration / Fetch Time
23:03:47	SELECT p.ProductLabel, COUNT(o.Ord...	54 row(s) returned	0.0012 sec / 0.00003...

16) Calculate the total number of orders placed by each customer (grouped by FirstName and LastName).

```
SELECT FirstName, LastName, COUNT(OrderId) AS TotalOrders
FROM orders
GROUP BY FirstName, LastName
ORDER BY TotalOrders DESC;
```

```

515
516
517
518 •  SELECT FirstName, LastName, COUNT(OrderId) AS TotalOrders
519   FROM orders
520   GROUP BY FirstName, LastName
521   ORDER BY TotalOrders DESC;
522
523
100% 12:519 | 1 error found

```

**Result Grid** Filter Rows: Search Export:

FirstName	LastName	Tot...
Amit	Sharma	2
Meenakshi	Gupta	1
Rajiv	Srinivasan	1
Pooja	Reddy	1
Siddharth	Kumar	1
Swati	Nair	1
Amita	Saxena	1
Ritika	Kapoor	1
Suman	Shah	1
Arjun	Mehta	1
Vani	Chopra	1

Result 1 Read Only

Action Output

Action	Time	Response	Duration / Fetch Time
SELECT FirstName, LastName, COUNT...	23:07:14	49 row(s) returned	0.00076 sec / 0.0000...

17) Find the products with the lowest minimum required quantity.

```

SELECT p.ProductName, p.MinimumRequired
FROM products p
JOIN (SELECT MIN(MinimumRequired) AS MinMinReq
      FROM products)
      ) min_req ON p.MinimumRequired = min_req.MinMinReq;
    
```

```

534
535 •  SELECT p.ProductName, p.MinimumRequired
536   FROM products p
537   JOIN (SELECT MIN(MinimumRequired) AS MinMinReq
538     FROM products
539   ) min_req ON p.MinimumRequired = min_req.MinMinReq;
540
541
542
100% 52:539 | 1 error found

```

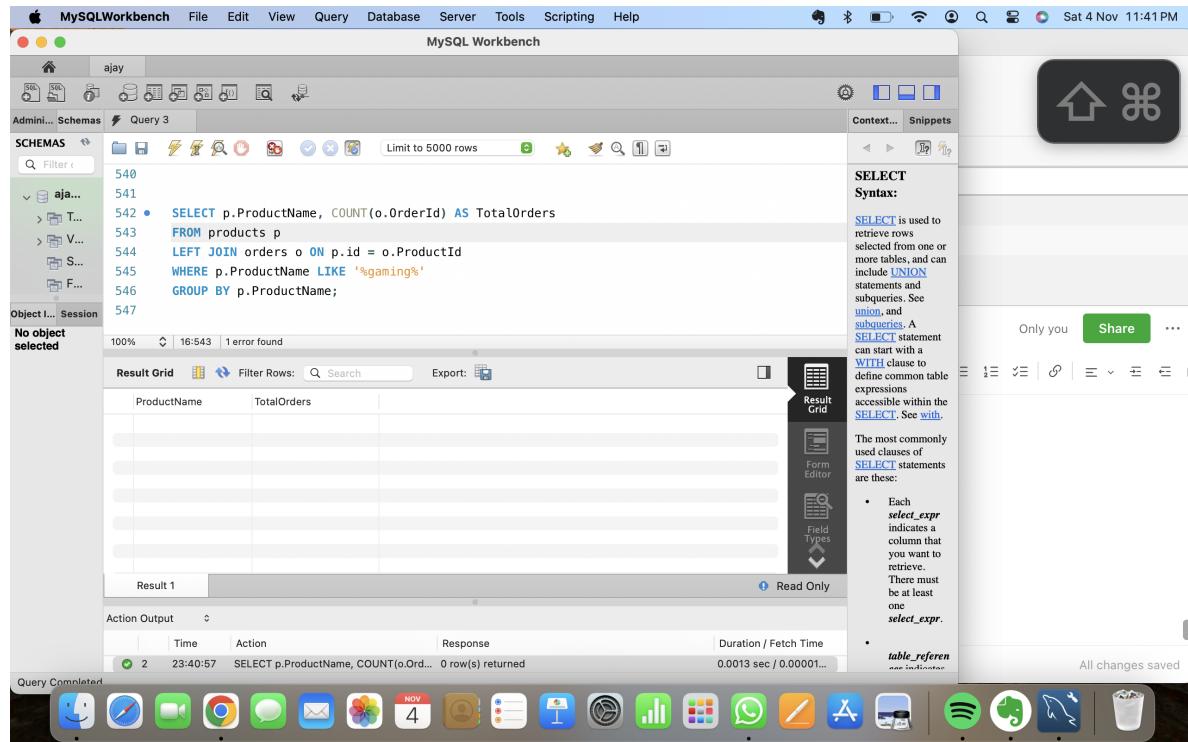
**Result Grid** Filter Rows: Search Export:

ProductName	MinimumRequired
nikon	2
Bajaj Auto	2
Google_Nest_hub	2

Result 1 Read Only

18) List all products with names containing "gaming".

```
SELECT p.ProductName, COUNT(o.OrderId) AS TotalOrders
FROM products p
LEFT JOIN orders o ON p.id = o.ProductId
WHERE p.ProductName LIKE '%gaming%'
GROUP BY p.ProductName;
```



19) Retrieve a list of products that have a 'StartingInventory' greater than 100,000 and a 'MinimumRequired' quantity less than 20."

```
SELECT ProductName, StartingInventory, MinimumRequired
FROM products
WHERE StartingInventory > 100000 AND MinimumRequired < 20;
```

The screenshot shows the MySQL Workbench interface. In the top pane, a SQL query is being run:

```
562 FROM products
563 WHERE ProductName LIKE '%watch%' AND StartingInventory > 50000;
564
565
566
567 • SELECT ProductName, StartingInventory, MinimumRequired
568   FROM products
569   WHERE StartingInventory > 100000 AND MinimumRequired < 20;
571
```

The results grid below shows two rows of data:

ProductName	StartingInventory	MinimumRequired
nikon	230000	2
Tesla_model3	129000	5

The status bar at the bottom indicates "100%" completion, "1:565" time, and "1 error found". On the right, a sidebar shows "Result Grid" selected.

20) To find out how many orders were made for purchases.

```
SELECT p.ProductId, COUNT(DISTINCT o.OrderId) AS TotalOrders
FROM purchases p
LEFT JOIN orders o ON p.ProductId = o.ProductId
GROUP BY p.ProductId
ORDER BY p.ProductId ASC;
```



```

592     GROUP BY Title, FirstName;
593
594
595 •  SELECT Title, FirstName
596   FROM orders
597   WHERE YEAR(Order_Date) BETWEEN 2010 AND 2023
598   GROUP BY Title, FirstName;
599
600

```

100% 12:596 | 1 error found

**Result Grid** Filter Rows: Search Export:

Title	FirstName
► Indian Pencil Set	Amit
Indian Phone Accessories	Meenakshi
Spices and Herbs India	Rajiv
Indian Home Office Essentials	Pooja
Indian Camping Gear	Siddharth
Indian Cooking Spices	Suman
Indian Music Concert	Sangeeta
Professional Art Supplies India	Vedika
Indian Party Decorations	Prisha
Woodworking Tools India	Vivaan
Indian Picnic Gear	Shivansh

orders 1 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
1	12:46:22	SELECT Title, FirstName FROM orders...	11 row(s) returned	0.00073 sec / 0.0000...

