

Untitled

```
CREATE TABLE suppliers (  
    id INT PRIMARY KEY,  
    supplier_Name VARCHAR(255)  
);
```

```
INSERT INTO suppliers (id, supplier_Name)  
VALUES (1, 'Supplier A'),  
      (2, 'Supplier B'),  
      (3, 'Supplier C'),  
      (4, 'Supplier E'),  
      (5, 'Supplier G'),  
      (6, 'Supplier R');
```

```
CREATE TABLE products (  
    id INT PRIMARY KEY,  
    ProductName VARCHAR(255),  
    PartNumber VARCHAR(50),  
    ProductLabel VARCHAR(50),  
    StartingInventory INT,  
    InventoryReceived INT,  
    InventoryShipped INT,  
    InventoryOnHand INT,  
    MinimumRequired INT  
);  
drop table products;
```

```
INSERT INTO products (id, ProductName, PartNumber, ProductLabel,  
StartingInventory, InventoryReceived, InventoryShipped, InventoryOnHand,  
MinimumRequired)  
VALUES  
(101,'iphone','xx1000','iphone_xx1000 ',51000,1000,500,30,10),  
(202,'MacBook','aa120','MacBook_aa120',97000,6000,500,62,10),
```

(303,'Tata Group','xxx120','Tata Group_xxx120',365,5100,52,10,3),
(404,'Infosys','kk520','Infosys_kk520',8800,6000,700,50,50),
(505,'Toyota','ax520','Toyota_ax520',1500000,625,600,50,50),
(606,'Samsung_galaxy','khj520','Samsung_galaxy_khj520',58000,5000,2000,633,50),
(707,'Amazon','ss1200','Amazon_ss1200',7000,2000,620,30,20),
(808,'Sony_playStation','lk472','Sony_playStation_lk472',36000,6300,6000,6000,6000),
(909,'Levi_jeans','ko245','Levi_jeans_ko245',3600,600,600,600,500),
(100,'HP_Envy_Laptop','xx564','HP_Envy_Laptop_xx564',58000,2000,200,20,20),
(110,'DELL_XPS','sc875','DELL_XPS_sc875',80000,96200,999,70,62),
(120,'NotePad','sxs555','NotePad_sxs555',11500,5000,552,63,60),
(130,'Canon_camera','uk545','Canon_camera_uk545',51000,700,632,63,5),
(140,'Google_pixel','bbb40','Google_pixel_bbb40',51000,5600,5230,55,63),
(150,'StarBucks_coffee','bxx999','StarBucks_coffee_bxx999',215,6400,400,362,52),
(160,'Fitbit_versa','lk574','Fitbit_versa_lk574',14000,965,652,52,63),
(170,'Boat_headphones','huk520','Boat_headphones_huk520',24000,685,202,10,14),
(180,'Rolex_Submariner','jj222','Rolex_Submariner_jj222',560000,965,252,144,55),
(190,'Sony_bravia_TV','hhh303','Sony_bravia_TV_hhh303',79000,4744,1000,152,20),
(200,'vacuum_cleaner','gj222','vacuum_cleaner_gj222',29000,7700,4520,6000,1520),
(210,'Adidas_Superstar_sneakers','jd555','Adidas_Superstar_sneakers_jd555',6400,620,320,100,250),
(220,'Honda_Civic','ksss41','Honda_Civic_ksss41',1700000,8000,6652,5555,3321),
(230,'LG_OLED_TV','kl5945','LG_OLED_TV_kl5945',97000,9854,655,554,33),
(240,'GoPro_hero_camera','kj200','GoPro_hero_camera_kj200',24000,9854,965,554,22),
(250,'Kellogg_corn_flakes','pol303','Kellogg_corn_flakes_pol303',290,6652,3333,22,646),
(260,'Philips_Sonicare_Toothbrush','llp584','Philips_Sonicare_Toothbrush_llp584',51000,710,14,22,5),
(270,'Canon_PIXMA_Printer','jij620','Canon_PIXMA_Printer_jij620',11500,800,50,66,22),
(280,'Gucci_Guilty_Perfume','kip520','Gucci_Guilty_Perfume_kip520',5000,8952,66,112,22),

(290,'microphone','ff100','microphone_ff100',58000,6666,5599,20,32),
 (300,'nikon','kkl754','nikon_kkl754',230000,8880,22,22,2),
 (310,'Toyota','pu520','Toyota_pu520',2000000,7000,7000,520,21),
 (320,'Nest_thermostat','pok54','Nest_thermostat_pok54',18500,6000,500,210,21),
 (330,'Play_station','jmn800','Play_station_jmn800',36000,8000,620,62,21),
 (340,'Samsung_TV','jmn933','Samsung_TV_jmn933',97000,951,212,51,21),
 (350,'Nike_Zoom_pegasus','kmn822','Nike_Zoom_pegasus_kmn822',8800,9854,5
 21,42,23),
 (360,'Apple_watch','kg400','Apple_watch_g400',36000,8000,520,332,110),
 (370,'Bajaj Auto','gk50','Bajaj Auto_gk50',58000,8754,254,21,2),
 (380,'Larsen & Toubro','kho444','Larsen &
 Toubro_kho444',18500,6522,400,21,30),
 (390,'Mahindra & Mahindra','gk230','Mahindra &
 Mahindra_gk230',365,8541,212,325,211),
 (400,'Sony','pl630','Sony_pl630',24000,8541,321,251,585),
 (410,'Google_Nest_hub','hdb','Google_Nest_hub_hdb',7000,5,5,2,2),
 (420,'Tesla_model3','klg','Tesla_model3_klg',129000,60,30,21,5),
 (430,'Dyson_Air_purifier','pokl','Dyson_Air_purifier_pokl',2900000,95,65,12,32),
 (440,'Under_Armour_Shirt','csbb','Under_Armour_Shirt_csbb',36000,75,65,52,45),
 (450,'Chanel_perfume','kg61','Chanel_perfume_kg61',2200,652,121,32,20),
 (460,'Bose_Speaker','uhg85','Bose_Speaker_uhg85',7000,751,456,89,35),
 (470,'HDFC Bank', 'N/A', 'HDFC', 6754, 500, 65, null, 456),
 (480,'ICICI Bank', 'N/A', 'ICICI', 15000, 453, 132, 13, 43),
 (490,'State Bank of India (SBI)', 'N/A', 'SBI', 3454, 0, 143, 987, 13),
 (500,'Hindustan Unilever Limited', 'N/A', 'HUL', 0, 10000, 569, 3400, 154),
 (510,'Wipro', 'N/A', 'Wipro', 15000, 56, 40, 34, 90),
 (520,'Adani Ports & SEZ', 'N/A', 'Adani', 45800, 498, 1304, 235, 59),
 (530,'Adani Power', 'N/A', 'Adani', 0, 6900, 5690, 4590, 134),
 (540,'Bharti Airtel', 'N/A', 'Airtel', 5, 670, 453, 09, 12),
 (550,'Tata Motors', 'N/A', 'Tata', 0, 0, 3498, 5980, 60);

select * from products;

```
CREATE TABLE orders (  
    OrderId INT PRIMARY KEY,  
    Title VARCHAR(100),  
    FirstName VARCHAR(100),  
    MiddleName VARCHAR(100),  
    LastName VARCHAR(100),  
    ProductId INT,  
    Number_Shipped INT,  
    Order_Date DATE  
);
```

```
drop table orders;
```

```
INSERT INTO orders (OrderId, Title, FirstName, MiddleName, LastName, ProductId,  
Number_Shipped, Order_Date)
```

```
VALUES
```

```
(301, 'Indian Pencil Set', 'Amit', 'Sanjay', 'Sharma', 5, 22, '2023-01-01'),  
(302, 'Indian Phone Accessories', 'Meenakshi', 'Harish', 'Gupta', 32, 11, '2023-10-  
10'),  
(303, 'Spices and Herbs India', 'Rajiv', 'Neha', 'Srinivasan', 10, 35, '2022-10-01'),  
(304, 'Indian Home Office Essentials', 'Pooja', 'Rakesh', 'Reddy', 41, 8, '2022-05-  
12'),  
(305, 'Indian Camping Gear', 'Siddharth', 'Priya', 'Kumar', 46, 18, '2010-01-03'),  
(306, 'Indian Kitchen Appliances', 'Swati', 'Suresh', 'Nair', 21, 52, '2001-01-06'),  
(307, 'Indian Party Supplies', 'Amita', 'Sarvesh', 'Saxena', 37, 35, '2003-06-06'),  
(308, 'Festival Celebrations India', 'Ritika', 'Hemant', 'Kapoor', 10, 6, '2004-05-12'),  
(309, 'Indian Cooking Spices', 'Suman', 'Raghav', 'Shah', 14, 22, '2010-04-05'),  
(310, 'Pet Care India', 'Arjun', 'Mansi', 'Mehta', 27, 49, '1998-12-06'),  
(311, 'Indian Clothing Collection', 'Vani', 'Vivek', 'Chopra', 12, 31, '1999-12-10'),  
(312, 'Indian Home Automation', 'Rahul', 'Kirti', 'Sharma', 47, 37, '2006-10-02'),  
(313, 'Books Collection India', 'Akanksha', 'Raman', 'Khanna', 32, 23, '2005-11-06'),  
(314, 'Car Maintenance India', 'Ankush', 'Manoj', 'Bhat', 45, 15, '2009-03-01'),  
(315, 'Indian Travel Essentials', 'Sakshi', 'Priyanka', 'Gupta', 50, 24, '2004-03-12'),  
(316, 'Indian Art Supplies', 'Mohan', 'Aditi', 'Malhotra', 17, 42, '2009-01-12'),  
(317, 'Nightlife in India', 'Nikhil', 'Sameer', 'Thakur', 61, 40, '2000-06-24'),  
(318, 'Indian Music Concert', 'Sangeeta', 'Rajesh', 'Chawla', 32, 30, '2020-01-15'),  
(319, 'Baby Care India', 'Priyanshi', 'Rahul', 'Mishra', 19, 41, '2008-05-07'),
```

(320, 'Indian Winter Fashion', 'Rahul', 'Seema', 'Verma', 25, 8, '2009-12-05'),
(321, 'Indian Gardening Tools', 'Kusum', 'Abhinav', 'Chawla', 40, 28, '2007-05-12'),
(322, 'Indian Entertainment Systems', 'Ravi', 'Sarika', 'Rana', 23, 50, '2004-12-12'),
(323, 'Indian Hiking Gear', 'Samarth', 'Simran', 'Shah', 36, 31, '2003-10-12'),
(324, 'Indian Jewelry Collection', 'Anika', 'Kiran', 'Sharma', 48, 17, '1999-01-12'),
(325, 'DIY Electronics India', 'Surya', 'Mihir', 'Iyer', 53, 33, '2003-07-06'),
(326, 'Indian Furniture Delivery', 'Karan', 'Ritu', 'Gupta', 21, 46, '2006-04-10'),
(327, 'Indian Poolside Relaxation', 'Kanika', 'Prateek', 'Saxena', 8, 44, '2007-08-03'),
(328, 'Indian Home Theater Setup', 'Rahul', 'Akshara', 'Thakur', 13, 25, '2002-04-05'),
(329, 'Indian Computer Accessories', 'Shivangi', 'Vikram', 'Sharma', 27, 10, '1998-03-11'),
(330, 'Indian Baking Ingredients', 'Divya', 'Naman', 'Kapoor', 15, 8, '2005-04-04'),
(331, 'DIY Home Decor India', 'Riya', 'Shreya', 'Verma', 29, 37, '2001-05-06'),
(332, 'Indian Fitness Equipment', 'Vivek', 'Aarushi', 'Joshi', 34, 15, '2000-09-02'),
(333, 'Educational Toys India', 'Aanya', 'Rudra', 'Patel', 1, 7, '2000-08-02'),
(334, 'Indian Pet Grooming', 'Mehak', 'Ananya', 'Yadav', 11, 12, '2001-05-07'),
(335, 'Indian Footwear Collection', 'Amit', 'Manju', 'Sharma', 3, 24, '2000-06-06'),
(336, 'Smartphone Accessories India', 'Sarika', 'Ashish', 'Verma', 13, 33, '2004-07-30'),
(337, 'Indian Books Collection', 'Vijay', 'Preeti', 'Singh', 6, 16, '2005-04-03'),
(338, 'Auto Care India', 'Neha', 'Aarohi', 'Mehra', 29, 19, '2008-07-06'),
(339, 'Indian Beach Essentials', 'Aarav', 'Pratik', 'Rao', 37, 14, '2003-05-17'),
(340, 'Professional Art Supplies India', 'Vedika', 'Sameer', 'Kapoor', 50, 22, '2020-07-01'),
(341, 'Indian Party Decorations', 'Prisha', 'Abhinav', 'Choudhary', 12, 9, '2020-09-02'),
(342, 'Woodworking Tools India', 'Vivaan', 'Naina', 'Pandey', 4, 5, '2010-01-01'),
(343, 'Indian Jewelry Finds', 'Yug', 'Nidhi', 'Malhotra', 9, 22, '2005-06-02'),
(344, 'DIY Robotics India', 'Aashna', 'Rahul', 'Verma', 18, 31, '1998-06-02'),
(345, 'Home Office Furniture India', 'Arya', 'Dhruv', 'Singh', 7, 28, '2007-07-16'),
(346, 'Indian Picnic Gear', 'Shivansh', 'Samaira', 'Rajput', 8, 14, '2011-09-02'),
(347, 'Indian Home Security', 'Mihika', 'Himanshu', 'Kapoor', 20, 20, '2003-02-18'),
(348, 'Indian Grooming Device', 'Anay', 'Pranjal', 'Vishwakarma', 15, 18, '2009-02-05'),
(349, 'Home Comfort India', 'Ishita', 'Gaurav', 'Tiwari', 40, 6, '2001-05-29'),
(350, 'Indian Home Ceremony', 'Aryan', 'Pihu', 'Sharma', 3, 45, '2003-06-23');

```
select * from orders;
```

```
CREATE TABLE purchases (  
    PurchaseId INT PRIMARY KEY,  
    SupplierId INT,  
    ProductId INT,  
    Number_Received INT,  
    Purchase_Date DATE  
);
```

```
INSERT INTO purchases VALUES  
(401, 211, 12, 50, '2023-11-11'),  
(402, 212, 15, 23, '2023-11-12'),  
(403, 213, 11, 14, '2023-11-13'),  
(404, 214, 13, 22, '2023-11-14'),  
(405, 215, 16, 28, '2023-11-15'),  
(406, 216, 18, 25, '2023-11-16'),  
(407, 217, 20, 22, '2023-11-17'),  
(408, 218, 19, 12, '2023-11-18'),  
(409, 219, 21, 19, '2023-11-19'),  
(410, 220, 17, 21, '2023-11-20'),  
(411, 221, 22, 15, '2023-11-21'),  
(412, 222, 23, 30, '2023-11-22'),  
(413, 223, 25, 14, '2023-11-23'),  
(414, 224, 28, 8, '2023-11-24'),  
(415, 225, 27, 35, '2023-11-25'),  
(416, 226, 30, 17, '2023-11-26'),  
(417, 227, 33, 22, '2023-11-27'),  
(418, 228, 31, 13, '2023-11-28'),  
(419, 229, 32, 20, '2023-11-29'),  
(420, 230, 34, 19, '2023-11-30'),  
(421, 226, 34, 9, '2023-11-21'),  
(422, 333, 28, 11, '2023-11-22'),  
(423, 555, 36, 14, '2023-11-23'),  
(424, 222, 49, 16, '2023-11-24'),  
(425, 111, 25, 12, '2023-11-25'),
```

(426, 444, 17, 6, '2023-11-26'),
(427, 555, 26, 8, '2023-11-27'),
(428, 333, 14, 10, '2023-11-28'),
(429, 222, 37, 9, '2023-11-29'),
(430, 111, 30, 7, '2023-11-30'),
(431, 555, 29, 8, '2023-12-01'),
(432, 333, 18, 11, '2023-12-02'),
(433, 444, 31, 9, '2023-12-03'),
(434, 222, 23, 14, '2023-12-04'),
(435, 111, 16, 6, '2023-12-05'),
(436, 555, 19, 10, '2023-12-06'),
(437, 226, 12, 9, '2023-12-07'),
(438, 333, 47, 8, '2023-12-08'),
(439, 555, 33, 7, '2023-12-09'),
(440, 444, 15, 5, '2023-12-10'),
(441, 226, 20, 12, '2023-12-11'),
(442, 444, 37, 14, '2023-12-12'),
(443, 222, 28, 9, '2023-12-13'),
(444, 111, 49, 10, '2023-12-14'),
(445, 555, 25, 8, '2023-12-15'),
(446, 226, 34, 13, '2023-12-16'),
(447, 333, 36, 7, '2023-12-17'),
(448, 222, 17, 6, '2023-12-18'),
(449, 555, 29, 10, '2023-12-19'),
(450, 333, 31, 12, '2023-12-20'),
(451, 444, 11, 15, '2023-12-21'),
(452, 222, 24, 17, '2023-12-22'),
(453, 555, 15, 10, '2023-12-23'),
(454, 333, 10, 11, '2023-12-24'),
(455, 111, 8, 9, '2023-12-25'),
(456, 226, 3, 14, '2023-12-26'),
(457, 444, 20, 16, '2023-12-27'),
(458, 222, 22, 12, '2023-12-28'),
(459, 555, 26, 18, '2023-12-29'),
(460, 333, 34, 15, '2023-12-30');

select *from purchases;

```
desc purchases;
desc supplier;
desc orders;
desc product;
```

```
select * from orders
where first name like "a%" and last name like "%r";
```

```
UPDATE products
SET StartingInventory = 1550000,
    MinimumRequired = 45
WHERE id = 5;
```

```
UPDATE products
SET StartingInventory = 1550000, MinimumRequired = 45
WHERE id = 5;
```

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;
```

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;
```


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;
```

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;
```

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;
```

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;
```

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 ;
```

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;
```

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;
```

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;
```

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;
```

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;
```

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;

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;
```

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;
```

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;
```

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;
```

21) Retrieve the 'Title' and 'FirstName' of customers who have placed orders in the year 2023.

```
SELECT Title, FirstName
FROM orders
WHERE YEAR(Order_Date) BETWEEN 2010 AND 2023
GROUP BY Title, FirstName;
```