1)To retrieve the refund status for a specific customer's orders.

```
SELECT
   Order.OrderID,
   Refund.RefundStatus
FROM
   Order
LEFT JOIN
   Refund ON Order.OrderID = Refund.OrderID
WHERE
   Order.CustomerID = <customer_id>;
```

2)To retrieve the list of orders to be dispatched for a specific supplier.

```
SELECT
   Order.OrderID,
   Order.OrderDate,
   Order.Status AS OrderStatus,
   Product.ProductName,
   Product.AvailableQuantity
FROM
   Order
JOIN
   Product ON Order.OrderID = Product.OrderID
WHERE
   Product.SupplierID = <supplier_id>
   AND Order.Status = 'Pending';
```

3)To view customer information, including their purchase history, ratings, and feedback.

```
SELECT
   Customer.CustomerID,
   Customer.Fname AS CustomerFirstName,
   Customer.Lname AS CustomerLastName,
   Order.OrderID,
   Order.OrderDate,
   Order.Status AS OrderStatus,
   Product.ProductName,
   Feedback.Rating,
   Feedback.Comment
FROM
   Customer
LEFT JOIN
   Order ON Customer.CustomerID = Order.CustomerID
LEFT JOIN
   Product ON Order.OrderID = Product.OrderID
LEFT JOIN
   Feedback ON Customer.CustomerID = Feedback.CustomerID
WHERE
   Product.SupplierID = <supplier_id>;
```

4)To Retrieve all customers with their total purchase amount

```
SELECT
   Customer.CustomerID,
   Customer.Fname,
   Customer.Lname,
    SUM(Product.Price) AS TotalPurchaseAmount
FROM
    Customer
LEFT JOIN
    Order ON Customer.CustomerID = Order.CustomerID
LEFT JOIN
    Product ON Order.OrderID = Product.OrderID
GROUP BY
    Customer.CustomerID
ORDER BY
    Customer.CustomerID ASC;
```

5)To show the top 5 products based on customer ratings

```
SELECT
    Product.ProductID,
    Product.ProductName,
    AVG(Feedback.Rating) AS AverageRating
FROM
    Product
LEFT JOIN
    Feedback ON Product.ProductID = Feedback.ProductID
GROUP BY
    Product.ProductID
ORDER BY
    AverageRating DESC
LIMIT 5;
```

6)To Retrieve orders with their respective products and associated warehouses.

```
SELECT
    Order.OrderID,
    Order.OrderDate,
    Product.ProductName,
    Warehouse.WarehouseName
FROM
    Order
JOIN
    Product ON Order.OrderID = Product.OrderID
JOIN
    Warehouse ON Product.WarehouseID = Warehouse.WarehouseID;
```

7)To track the order status for a specific customer.

```
    SELECT
        Order.OrderID,
        Order.OrderDate,
        Order.Status AS OrderStatus,
        Product.ProductName
    FROM
        Order
    JOIN
        Product ON Order.OrderID = Product.OrderID
   WHERE
        Order.CustomerID = 1001
   ORDER BY
        Order.OrderID ASC;
```

8)To filter products by attributes such as price range, quantity in stock, or location.

```
SELECT
   P.ProductID,
   P.ProductName,
   P.Category,
   P.Price,
   P.AvaiableQuantity,
   W.Location AS WarehouseLocation
FROM
   Product P
JOIN
   Warehouse W ON P.WarehouseID = W.WarehouseID
WHERE
   P.Price BETWEEN <Min_Price> AND <Max_Price>
   AND P.AvailableQuantity >= <Min_Quantity>
   AND W.Location = '<Warehouse_Location>';
```

9)To search for products based on its category.

```
SELECT
   P.ProductID,
   P.ProductName,
   P.Price
FROM
   Product P
WHERE
   P.Category LIKE <Category>;
```

10)To know the sales for a particular month.

```
SELECT
   TO_CHAR(OrderDate, 'YYYY-MM') AS YearMonth,
   SUM(Price) AS TotalSalesAmount
FROM
   Order
JOIN
   Product ON Order.OrderID = Product.OrderID
WHERE
   TO_CHAR(OrderDate, 'YYYY-MM') = '2024-04' -- Change '2024-04' to the desired month
GROUP BY
   TO_CHAR(OrderDate, 'YYYY-MM');
```

11)To know the sales from Month X to Month Y.

```
SELECT
   TO_CHAR(OrderDate, 'YYYY-MM') AS YearMonth,
      SUM(Price) AS TotalSalesAmount
FROM
   Order
INNER JOIN
   Product ON Order.OrderID = Product.OrderID
WHERE
   TO_CHAR(OrderDate, 'YYYY-MM') >= '2024-01' -- Change '2024-01' to the starting month
   AND TO_CHAR(OrderDate, 'YYYY-MM') <= '2024-03' -- Change '2024-03' to the ending month
GROUP BY
```

```
    TO_CHAR(OrderDate, 'YYYY-MM');
```

12)For supplier to access a sales register to track sales transactions by date, including order details and customer information.

```
SELECT
    O.OrderID,
    O.OrderDate,
    C.CustomerID,
    CONCAT(C.Fname,' ', C.Mname ,' ', C.Lname) AS CustomerName,
    C.State,
    C.City,
    C.Pincode,
    C.Street,
    P.ProductID,
    P.ProductName,
    P.Price
FROM
    Order O
JOIN
    Customer C ON O.CustomerID = C.CustomerID
JOIN
    Product P ON O.OrderID = P.OrderID
WHERE
    O.OrderDate = '2023-05-12'; -- Change the date as needed
```

13)For manager To facilitate stock allocation.

```
SELECT
    W.WarehouseID,
    W.WarehouseName,
    W.Location,
    W.Capacity - COALESCE(SUM(P.AvailableQuantity), 0) AS RemainingCapacity
FROM
    Warehouse W
LEFT JOIN
    Product P ON W.WarehouseID = P.WarehouseID
GROUP BY
    W.WarehouseID,W.Location, W.WarehouseName, W.Capacity
ORDER BY
    W.WarehouseID ASC;
```

14)List of Suppliers with Their Total Products and Available Quantity.

```
SELECT
    S.SupplierID,
    CONCAT(S.Fname, ' ',S.Mname, ' ', S.Lname) AS SupplierName,
    COUNT(P.ProductID) AS TotalProducts,
    SUM(P.AvailableQuantity) AS Total_Quantity
FROM
    Supplier S
LEFT JOIN
    Product P ON S.SupplierID = P.SupplierID
GROUP BY
    S.SupplierID, SupplierName
ORDER BY
    S.SupplierID ASC;
```

15)List of Warehouses with Total Capacity and Used Capacity.

```
SELECT
   W.WarehouseID,
   W.WarehouseName,
   W.Location,
   W.Capacity AS TotalCapacity,
   SUM(P.AvailableQuantity) AS UsedCapacity
FROM
   Warehouse W
LEFT JOIN
   Product P ON W.WarehouseID = P.WarehouseID
GROUP BY
   W.WarehouseID, W.WarehouseName, W.Capacity, W.Location
ORDER BY
   W.WarehouseID ASC;
```

16)List of Orders with Order Details (Product Name,Price,Date).

```
SELECT
   O.OrderID,
   O.OrderDate,
   P.ProductName,
   P.AvailableQuantity,
   P.Price
FROM
   Order O
JOIN
   Product P ON O.OrderID = P.OrderID
ORDER BY
   O.OrderDate DESC;
```

17)List of Products with Low Stock (Available Quantity < Threshold).

```
SELECT
   ProductID,
   ProductName,
   Category,
   AvailableQuantity
FROM
   Product
WHERE
   AvailableQuantity < 50; -- Adjust threshold as needed
```

18)To get Total Number of Orders Placed Each Month.

```
SELECT
   TO_CHAR(OrderDate, 'YYYY-MM') AS YearMonth,
   COUNT(OrderID) AS TotalOrders
FROM
   Order
GROUP BY
   TO_CHAR(OrderDate, 'YYYY-MM')
ORDER BY
   YearMonth;
```

19)List of Customers Who Haven't Placed Orders.

```
SELECT
   C.CustomerID,
   CONCAT(C.Fname, ' ', C.Mname,' ', C.Lname) AS CustomerName
FROM
   Customer C
LEFT JOIN
   Order O ON C.CustomerID = O.CustomerID
WHERE
   O.CustomerID IS NULL;
```

20)List of Customers with Highest Total Spending (Top 10 Customers).

```
SELECT
   C.CustomerID,
   CONCAT(C.Fname, ' ', C.Mname,' ', C.Lname) AS CustomerName,
   SUM(P.Price) AS TotalSpending
FROM
   Customer C
JOIN
   Order O ON C.CustomerID = O.CustomerID
JOIN
   Product P ON O.OrderID = P.OrderID
GROUP BY
   C.CustomerID, CustomerName
ORDER BY
   TotalSpending DESC
LIMIT 10; -- Adjust the limit as needed
```

21)Retrieve the Top 10 most Expensive Products

```
SELECT ProductName, Price
FROM Product
ORDER BY Price DESC
LIMIT 10; -- Adjust the limit as needed
```

22)Retrieve the customers who have placed orders but not provided any feedback.

```
SELECT C.CustomerID, C.Fname, C.Lname
FROM Customer C
WHERE C.CustomerID NOT IN (
  SELECT CustomerID
  FROM Feedback
)
AND C.CustomerID IN (
  SELECT CustomerID
  FROM Order
ORDER BY CustomerID ASC
);
```

23)Retrieve the products that have no associated feedback.

```
SELECT
   P.ProductID,
   P.ProductName
```

```
FROM
    Product P
WHERE P.ProductID NOT IN (
  SELECT ProductID
  FROM Feedback
);
```

24)Retrieve the customers who have placed orders for products from a specific supplier.

```
SELECT DISTINCT C.CustomerID, C.Fname, C.Lname
FROM
    Customer C
JOIN
    Order O ON C.CustomerID = O.CustomerID
JOIN
    Product P ON O.OrderID = P.OrderID
WHERE
    P.SupplierID = <supplier_id>
```

25)Retrieve the suppliers who have supplied products that have received the highest average rating.

```
SELECT S.SupplierID, S.Fname, S.Lname
FROM Supplier S
JOIN Product P ON S.SupplierID = P.SupplierID
JOIN (
  SELECT ProductID, AVG(Rating) AS AvgRating
  FROM Feedback
  GROUP BY ProductID
  ORDER BY AvgRating DESC
  LIMIT 3
) T ON P.ProductID = T.ProductID
GROUP BY S.SupplierID, S.Fname, S.Lname;
```

26)List the total amount associated with a particular order id.

```
SELECT OrderID, SUM(Price) AS TotalPrice
FROM Product
WHERE OrderID = <order_id>
GROUP BY OrderID
ORDER BY OrderID ASC;
```

27)To Retrieve the employee details who have prepared a specific order.

```
SELECT E.EmployeeID, CONCAT(E.Fname, ' ', COALESCE(E.Mname, ''), ' ', E.Lname) AS EmployeeName
FROM Order O
JOIN Employee E ON O.EmployeeID = E.EmployeeID
WHERE O.OrderID = <order_id>;
```

28)To retrieve the invoice associated with a specific order.

```
SELECT I.InvoiceID, I.TotalAmount, I.InvoiceDate
FROM Invoice I
JOIN Order O ON I.OrderID = O.OrderID
WHERE O.OrderID = <order_id>;
```