**Exercise on WITH ROLLUP**

**Exercise 1:** Write a query to calculate the total number of orders in the "orders" table, including a grand total.

**Solution 1:**

SELECT

IFNULL(CustomerID, 'Grand Total') AS CustomerID,

COUNT(\*) AS TotalOrders

FROM orders

GROUP BY CustomerID WITH ROLLUP;

**Exercise 2:** Calculate the total revenue generated by all orders in the "order\_details" table, including subtotals for each order and a grand total.

**Solution 2:**

SELECT

IFNULL(OrderID, 'Grand Total') AS OrderID,

SUM(products.UnitPrice \* Quantity) AS Subtotal

FROM order\_details

GROUP BY OrderID WITH ROLLUP;

**Exercise 3:** Find the total number of products in the "products" table, including a grand total.

**Solution 3:**

SELECT

IFNULL(CategoryID, 'Grand Total') AS CategoryID,

COUNT(\*) AS TotalProducts

FROM products

GROUP BY CategoryID WITH ROLLUP;

**Exercise 4:** Calculate the total number of customers in the "customers" table, including a grand total.

**Solution 4:**

SELECT

IFNULL(CustomerID, 'Grand Total') AS CustomerID,

COUNT(\*) AS TotalCustomers

FROM customers

GROUP BY CustomerID WITH ROLLUP;

**Exercise 5:** Write a query to calculate the subtotal of all orders (total price for each order) in the "orders" table, including a grand total.

**Solution 5:**

SELECT

IFNULL(OrderID, 'Grand Total') AS OrderID,

SUM(products.UnitPrice \* Quantity) AS Subtotal

FROM order\_details

GROUP BY OrderID WITH ROLLUP;

**Exercise 6:** Find the total number of orders placed by each customer in the "customers" table, including a grand total.

**Solution 6:**

SELECT

IFNULL(CustomerID, 'Grand Total') AS CustomerID,

COUNT(\*) AS TotalOrders

FROM orders

GROUP BY CustomerID WITH ROLLUP;

**Exercise 7:** Calculate the total revenue generated by each category in the "categories" table, including a grand total.

**Solution 7:**

SELECT

IFNULL(CategoryName, 'Grand Total') AS CategoryName,

SUM(products.UnitPrice \* Quantity) AS CategoryRevenue

FROM order\_details

JOIN products ON order\_details.ProductID = products.ProductID

JOIN categories ON products.CategoryID = categories.CategoryID

GROUP BY CategoryName WITH ROLLUP;

**Exercise 8:** Calculate the grand total revenue for all orders, including subtotals for each order.

**Solution 8:**

SELECT

IFNULL(OrderID, 'Grand Total') AS OrderID,

SUM(products.UnitPrice \* Quantity) AS Subtotal

FROM order\_details

GROUP BY OrderID WITH ROLLUP;

**Exercise 9:** Calculate the grand total revenue for all orders placed by each customer in the "customers" table, including subtotals for each customer.

**Solution 9:**

SELECT

IFNULL(CustomerID, 'Grand Total') AS CustomerID,

SUM(products.UnitPrice \* Quantity) AS Subtotal

FROM orders

JOIN order\_details ON orders.OrderID = order\_details.OrderID

GROUP BY CustomerID WITH ROLLUP;

**Exercise 10:** Calculate the grand total revenue for each product in the "products" table, including subtotals for each product.

**Solution 10:**

SELECT

IFNULL(ProductName, 'Grand Total') AS ProductName,

SUM(products.UnitPrice \* Quantity) AS Subtotal

FROM order\_details

JOIN products ON order\_details.ProductID = products.ProductID

GROUP BY ProductName WITH ROLLUP;

**Exercise 11:** Calculate the total revenue generated by each customer, including subtotals for each order and a grand total. Use the "customers," "orders," and "order\_details" tables.

**Solution 11:**

SELECT IFNULL(Customers.CustomerID, 'Grand Total') AS CustomerID,

IFNULL(Orders.OrderID, 'Subtotal') AS OrderID,

SUM(UnitPrice \* Quantity) AS Subtotal

FROM Customers

LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID

LEFT JOIN Order\_Details ON Orders.OrderID = Order\_Details.OrderID

GROUP BY Customers.CustomerID, Orders.OrderID WITH ROLLUP;

**Exercise 12:** Calculate the total number of products ordered by each customer, including subtotals for each order and a grand total. Use the "customers," "orders," and "order\_details" tables.

**Solution 12:**

SELECT

IFNULL(Customers.CustomerID, 'Grand Total') AS CustomerID,

IFNULL(Orders.OrderID, 'Subtotal') AS OrderID,

SUM(Quantity) AS TotalProductsOrdered

FROM Customers

LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID

LEFT JOIN Order\_Details ON Orders.OrderID = Order\_Details.OrderID

GROUP BY Customers.CustomerID, Orders.OrderID WITH ROLLUP;

**Exercise 13:** Calculate the total revenue generated by each employee, including subtotals for each order and a grand total. Use the "employees," "orders," and "order\_details" tables.

**Solution 13:**

SELECT

IFNULL(Employees.EmployeeID, 'Grand Total') AS EmployeeID,

IFNULL(Orders.OrderID, 'Subtotal') AS OrderID,

SUM(UnitPrice \* Quantity) AS Subtotal

FROM Employees

LEFT JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID

LEFT JOIN Order\_Details ON Orders.OrderID = Order\_Details.OrderID

GROUP BY Employees.EmployeeID, Orders.OrderID WITH ROLLUP;

**Exercise 14:** Calculate the total number of products ordered by each employee, including subtotals for each order and a grand total. Use the "employees," "orders," and "order\_details" tables.

**Solution 14:**

SELECT

IFNULL(Employees.EmployeeID, 'Grand Total') AS EmployeeID,

IFNULL(Orders.OrderID, 'Subtotal') AS OrderID,

SUM(Quantity) AS TotalProductsOrdered

FROM Employees

LEFT JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID

LEFT JOIN Order\_Details ON Orders.OrderID = Order\_Details.OrderID

GROUP BY Employees.EmployeeID, Orders.OrderID WITH ROLLUP;

**Exercise 15:** Calculate the total revenue generated by each category, including subtotals for each product and a grand total. Use the "categories," "products," and "order\_details" tables.

**Solution 15:**

SELECT

IFNULL(Categories.CategoryName, 'Grand Total') AS CategoryName,

IFNULL(Products.ProductName, 'Subtotal') AS ProductName,

SUM(UnitPrice \* Quantity) AS Subtotal

FROM Categories

LEFT JOIN Products ON Categories.CategoryID = Products.CategoryID

LEFT JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY Categories.CategoryName, Products.ProductName WITH ROLLUP;

**Exercise 16:** Calculate the total number of products ordered from each category, including subtotals for each product and a grand total. Use the "categories," "products," and "order\_details" tables.

**Solution 16:**

SELECT

IFNULL(Categories.CategoryName, 'Grand Total') AS CategoryName,

IFNULL(Products.ProductName, 'Subtotal') AS ProductName,

SUM(Quantity) AS TotalProductsOrdered

FROM Categories

LEFT JOIN Products ON Categories.CategoryID = Products.CategoryID

LEFT JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY Categories.CategoryName, Products.ProductName WITH ROLLUP;

**Exercise 17:** Calculate the total revenue generated by each supplier, including subtotals for each product and a grand total. Use the "suppliers," "products," and "order\_details" tables.

**Solution 17:**

SELECT

IFNULL(Suppliers.SupplierName, 'Grand Total') AS SupplierName,

IFNULL(Products.ProductName, 'Subtotal') AS ProductName,

SUM(UnitPrice \* Quantity) AS Subtotal

FROM Suppliers

LEFT JOIN Products ON Suppliers.SupplierID = Products.SupplierID

LEFT JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY Suppliers.SupplierName, Products.ProductName WITH ROLLUP;

**Exercise 18:** Calculate the total number of products ordered from each supplier, including subtotals for each product and a grand total. Use the "suppliers," "products," and "order\_details" tables.

**Solution1 8:**

SELECT

IFNULL(Suppliers.SupplierName, 'Grand Total') AS SupplierName,

IFNULL(Products.ProductName, 'Subtotal') AS ProductName,

SUM(Quantity) AS TotalProductsOrdered

FROM Suppliers

LEFT JOIN Products ON Suppliers.SupplierID = Products.SupplierID

LEFT JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY Suppliers.SupplierName, Products.ProductName WITH ROLLUP;

**Exercise 19:** Calculate the total revenue generated by each employee for orders placed by each customer, including subtotals for each order, and a grand total. Use the "employees," "customers," "orders," and "order\_details" tables.

**Solution 19:**

SELECT

IFNULL(Employees.EmployeeID, 'Grand Total') AS EmployeeID,

IFNULL(Customers.CustomerID, 'Grand Total') AS CustomerID,

IFNULL(Orders.OrderID, 'Subtotal') AS OrderID,

SUM(UnitPrice \* Quantity) AS Subtotal

FROM Employees

LEFT JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID

LEFT JOIN Customers ON Orders.CustomerID = Customers.CustomerID

LEFT JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

GROUP BY Employees.EmployeeID, Customers.CustomerID, Orders.OrderID WITH ROLLUP;

**Exercise 20:** Calculate the total number of products ordered by each employee for orders placed by each customer, including subtotals for each order, and a grand total. Use the "employees," "customers," "orders," and "order\_details" tables.

**Solution 20:**

SELECT

IFNULL(Employees.EmployeeID, 'Grand Total') AS EmployeeID,

IFNULL(Customers.CustomerID, 'Grand Total') AS CustomerID,

IFNULL(Orders.OrderID, 'Subtotal') AS OrderID,

SUM(Quantity) AS TotalProductsOrdered

FROM Employees

LEFT JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID

LEFT JOIN Customers ON Orders.CustomerID = Customers.CustomerID

LEFT JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

GROUP BY Employees.EmployeeID, Customers.CustomerID, Orders.OrderID WITH ROLLUP;