



MYSQL PPT

Kalaimani Muthu

Top 10 Customers by Credit Limit

- Query:
- `SELECT customerName, creditLimit`
- `FROM customers`
- `ORDER BY creditLimit DESC`
- `LIMIT 10;`
- Interpretation:
- This data shows the top 10 customers by credit limit.

Average Credit Limit by Country

- Query:
- `SELECT country, AVG(creditLimit) AS averageCreditLimit`
- `FROM customers`
- `GROUP BY country;`
- Interpretation:
- This data shows the average credit limit for customers in each country.

Number of Customers by State

- Query:
- `SELECT state, COUNT(*) AS numberOfCustomers`
- `FROM customers`
- `GROUP BY state;`
- Interpretation:
- This data shows the number of customers in each state.

Customers Without Orders

- Query:
- `SELECT customerName`
- `FROM customers`
- `WHERE customerNumber NOT IN (SELECT customerNumber FROM orders);`
- Interpretation:
- This data shows the customers who haven't placed any orders.

Total Sales for Each Customer

- Query:
- `SELECT customers.customerName,
SUM(orderdetails.quantityOrdered * orderdetails.priceEach) AS
totalSales`
- `FROM customers`
- `JOIN orders ON customers.customerNumber =
orders.customerNumber`
- `JOIN orderdetails ON orders.orderNumber =
orderdetails.orderNumber`
- `GROUP BY customers.customerName;`
- Interpretation:
- This data shows the total sales for each customer.

Customers and Their Sales Representatives

- Query:
- `SELECT customers.customerName, employees.firstName AS salesRepFirstName, employees.lastName AS salesRepLastName`
- `FROM customers`
- `JOIN employees ON customers.salesRepEmployeeNumber = employees.employeeNumber;`
- Interpretation:
- This data shows the list of customers with their assigned sales representatives.

Most Recent Payment Details

- Query:
- `SELECT customers.customerName, payments.paymentDate,`
`payments.amount`
- `FROM customers`
- `JOIN payments ON customers.customerNumber =`
`payments.customerNumber`
- `WHERE (payments.customerNumber, payments.paymentDate) IN (`
- `SELECT customerNumber, MAX(paymentDate)`
- `FROM payments`
- `GROUP BY customerNumber`
- `);`
- Interpretation:
- This data provides the most recent payment details for each customer.

Product Lines Summary

- Query:
- `SELECT productLine, COUNT(*) AS
numberOfProducts`
- `FROM products`
- `GROUP BY productLine;`
- Interpretation:
- This data shows the number of products in each product line.

Employees with Their Offices

- Query:
- `SELECT employees.firstName,
employees.lastName, offices.city, offices.country`
- `FROM employees`
- `JOIN offices ON employees.officeCode =
offices.officeCode;`
- Interpretation:
- This data provides information on employees and the offices they work in.

Monthly Sales Revenue

- Query:
- `SELECT DATE_FORMAT(orderDate, '%Y-%m') AS month,
SUM(orderdetails.quantityOrdered *
orderdetails.priceEach) AS totalRevenue`
- `FROM orders`
- `JOIN orderdetails ON orders.orderNumber =
orderdetails.orderNumber`
- `GROUP BY month;`
- Interpretation:
- This data shows the trend of monthly sales revenue.

Customers and Orders

- Query:
- `SELECT customers.customerName,
orders.orderNumber, orders.orderDate, orders.status`
- `FROM customers`
- `JOIN orders ON customers.customerNumber =
orders.customerNumber;`
- Interpretation:
- This data provides a detailed view of customer orders.

Product Sales Summary

- Query:
- `SELECT products.productName,
SUM(orderdetails.quantityOrdered) AS totalQuantitySold`
- `FROM products`
- `JOIN orderdetails ON products.productCode =
orderdetails.productCode`
- `GROUP BY products.productName;`
- Interpretation:
- This data shows the sales performance of individual products.

Orders with Delayed Shipping

- Query:
- `SELECT * FROM orders`
- `WHERE shippedDate > requiredDate;`
- Interpretation:
- This data lists orders that were shipped later than their required date.

Popular Product Combinations

- Query:
- `SELECT od1.productCode AS product1, od2.productCode AS product2, COUNT(*) AS combinationCount`
- `FROM orderdetails od1`
- `JOIN orderdetails od2 ON od1.orderNumber = od2.orderNumber AND od1.productCode < od2.productCode`
- `GROUP BY product1, product2`
- `ORDER BY combinationCount DESC;`
- Interpretation:
- This data shows the most frequently ordered product combinations.

Top 10 Most Profitable Orders

- Query:
- `SELECT o.orderNumber, SUM(od.quantityOrdered * od.priceEach) AS totalRevenue`
- `FROM orders o`
- `JOIN orderdetails od ON o.orderNumber = od.orderNumber`
- `GROUP BY o.orderNumber`
- `ORDER BY totalRevenue DESC`
- `LIMIT 10;`
- Interpretation:
- This data identifies the top 10 most profitable orders based on total revenue.

Trigger for Updating Customer Credit Limit

- Trigger Creation:
- CREATE TRIGGER update_credit_limit AFTER INSERT ON orders
- FOR EACH ROW
- BEGIN
- UPDATE customers
- SET creditLimit = creditLimit - NEW.orderTotal
- WHERE customerNumber = NEW.customerNumber;
- END;
- Interpretation:
- Automatically reduces a customer's credit limit by the total order amount whenever a new order is placed.

Trigger for Logging Product Quantity Changes

- Trigger Creation:
- CREATE TRIGGER log_quantity_changes AFTER INSERT OR UPDATE ON orderdetails
- FOR EACH ROW
- BEGIN
- INSERT INTO product_quantity_log (orderNumber, productCode, oldQuantity, newQuantity, changeDate)
- VALUES (NEW.orderNumber, NEW.productCode, OLD.quantityOrdered, NEW.quantityOrdered, NOW());
- END;
- Interpretation:
- Logs changes in product quantities whenever an order detail is inserted or updated.