SQL Queries with Aggregate Functions and Case Statement

Stay Tuned, Rakesh Men | LinkedIn

Example 1: Conditional Sum

Suppose you have a table sales with columns sale_id, amount, and region. You want to calculate the total sales amount for two different regions (e.g., "North" and "South") in a single query.

```
SELECT
  SUM(CASE WHEN region = 'North' THEN amount ELSE 0 END) AS north_sales,
  SUM(CASE WHEN region = 'South' THEN amount ELSE 0 END) AS south_sales
FROM
  sales;
```

Example 2: Conditional Count

Suppose you have a table employees with columns employee_id, department, and salary. You want to count the number of employees in each department who earn more than $50,000.

```
SELECT
  department,
  COUNT(CASE WHEN salary > 50000 THEN 1 END) AS high_earners_count
FROM
  employees
GROUP BY
  department;
```

Example 3: Conditional Average

Suppose you have a table orders with columns order_id, customer_id, order_amount, and order_date. You want to calculate the average order amount for orders placed in the current year versus the previous year.

```
SELECT
  AVG(CASE WHEN YEAR(order_date) = YEAR(CURDATE()) THEN order_amount END) AS avg_current_year,
  AVG(CASE WHEN YEAR(order_date) = YEAR(CURDATE()) - 1 THEN order_amount END) AS avg_previous_year
FROM
  orders;
```

Example 4: Conditional Min and Max

Suppose you have a table products with columns product_id, category, and price. You want to find the minimum and maximum prices for products in two different categories (e.g., "Electronics" and "Clothing").

```
SELECT
  MIN(CASE WHEN category = 'Electronics' THEN price END) AS min_electronics_price,
  MAX(CASE WHEN category = 'Electronics' THEN price END) AS max_electronics_price,
  MIN(CASE WHEN category = 'Clothing' THEN price END) AS min_clothing_price,
  MAX(CASE WHEN category = 'Clothing' THEN price END) AS max_clothing_price
FROM
  products;
```

Example 5: Conditional Aggregation with Group By

Suppose you have a table transactions with columns transaction_id, account_id, transaction_type, and amount. You want to calculate the total deposits and withdrawals for each account.

```
SELECT
  account_id,
  SUM(CASE WHEN transaction_type = 'Deposit' THEN amount ELSE 0 END) AS total_deposits,
  SUM(CASE WHEN transaction_type = 'Withdrawal' THEN amount ELSE 0 END) AS total_withdrawals
FROM
  transactions
GROUP BY
  account_id;
```

Example 6: Conditional Count with Multiple Conditions

Suppose you have a table students with columns student_id, grade, and attendance. You want to count the number of students who have an "A" grade and attendance greater than 90%.

```
SELECT
  COUNT(CASE WHEN grade = 'A' AND attendance > 90 THEN 1 END) AS high_achievers_count
FROM
  students;
```

Example 7: Conditional Sum with Nested Case

Suppose you have a table sales with columns sale_id, amount, and region. You want to calculate the total sales amount for different regions, but with a special condition for one region.

```
SELECT
  SUM(CASE
        WHEN region = 'North' THEN amount
        WHEN region = 'South' THEN amount * 1.1  -- Add 10% for South region
        ELSE 0
      END) AS total_sales
FROM
  sales;
```

Example 8: Conditional Count with Multiple Categories

Suppose you have a table products with columns product_id, category, and price. You want to count the number of products in each category that are priced above and below a certain threshold.

```
SELECT
  category,
  COUNT(CASE WHEN price > 100 THEN 1 END) AS high_price_count,
  COUNT(CASE WHEN price <= 100 THEN 1 END) AS low_price_count
FROM
  products GROUP BY category;
```

Example 9: Conditional Sum with Group By and Filtering

Suppose you have a table orders with columns order_id, customer_id, order_amount, and order_date. You want to calculate the total order amount for each customer, but only for orders placed in the current year.

```
SELECT
  customer_id,
  SUM(CASE WHEN YEAR(order_date) = YEAR(CURDATE()) THEN order_amount ELSE 0 END) AS total_amount_current_ye
ar
FROM
  orders
GROUP BY
  customer_id;
```

Example 10: Conditional Average with Multiple Conditions

Suppose you have a table employees with columns employee_id, department, salary, and performance_rating. You want to calculate the average salary for employees with a performance rating of 4 or 5, grouped by department.

```
SELECT
  department,
  AVG(CASE WHEN performance_rating >= 4 THEN salary END) AS avg_salary_high_performers
FROM
  employees
GROUP BY
  department;
```

Example 11: Conditional Min and Max with Group By

Suppose you have a table sales with columns sale_id, region, amount, and sale_date. You want to find the minimum and maximum sale amounts for each region, but only for sales made in the last 30 days.

```
SELECT
  region,
  MIN(CASE WHEN sale_date >= DATEADD(DAY, -30, GETDATE()) THEN amount END) AS min_last_30_days,
  MAX(CASE WHEN sale_date >= DATEADD(DAY, -30, GETDATE()) THEN amount END) AS max_last_30_days
FROM
  sales
GROUP BY
  region;
```

Example 12: Conditional Count with Null Handling

Suppose you have a table students with columns student_id, name, grade, and attendance. You want to count the number of students who have a grade of "A" or "B" and attendance greater than 80%, while handling null values in the attendance column.

```
SELECT
  COUNT(CASE WHEN grade IN ('A', 'B') AND (attendance > 80 OR attendance IS NULL) THEN 1 END) AS high_achievers_co
unt
FROM
  students;
```

Example 13: Conditional Sum with Nested Case and Group By

Suppose you have a table transactions with columns transaction_id, account_id, transaction_type, and amount. You want to calculate the total deposits and withdrawals for each account, but apply a 10% fee for withdrawals.

```
SELECT
  account_id,
  SUM(CASE WHEN transaction_type = 'Deposit' THEN amount ELSE 0 END) AS total_deposits,
  SUM(CASE WHEN transaction_type = 'Withdrawal' THEN amount * 0.9 ELSE 0 END) AS total_withdrawals
FROM
  transactions
GROUP BY
  account_id;
```

Example 14: Conditional Aggregation with Date Ranges

Suppose you have a table events with columns event_id, event_name, event_date, and attendees. You want to count the number of attendees for events that occurred in the first half of the year versus the second half of the year.

```
SELECT
  SUM(CASE WHEN MONTH(event_date) BETWEEN 1 AND 6 THEN attendees ELSE 0 END) AS attendees_first_half,
  SUM(CASE WHEN MONTH(event_date) BETWEEN 7 AND 12 THEN attendees ELSE 0 END) AS attendees_second_half
FROM
  events;
```

Example 15: Conditional Count with Multiple Columns

Suppose you have a table orders with columns order_id, customer_id, order_amount, and order_status. You want to count the number of orders that are either "Completed" or "Pending" for each customer.

```
SELECT
  customer_id,
  COUNT(CASE WHEN order_status = 'Completed' THEN 1 END) AS completed_orders,
  COUNT(CASE WHEN order_status = 'Pending' THEN 1 END) AS pending_orders
FROM
  orders
GROUP BY
  customer_id;
```

Example 16: Conditional Sum with Multiple Conditions and Group By

Suppose you have a table sales with columns sale_id, region, amount, and sale_date. You want to calculate the total sales amount for each region, but only for sales made on weekends.

```
SELECT
  region,
  SUM(CASE WHEN DATENAME(WEEKDAY, sale_date) IN ('Saturday', 'Sunday') THEN amount ELSE 0 END) AS weekend_sal
es
FROM
  sales
GROUP BY region;
```

Example 17: Conditional Aggregation with String Matching

Suppose you have a table products with columns product_id, product_name, and price. You want to calculate the total price of products whose names contain the word "Premium".

```
SELECT
  SUM(CASE WHEN product_name LIKE '%Premium%' THEN price ELSE 0 END) AS total_premium_sales
FROM
  products;
```

Example 18: Conditional Count with Multiple Groupings

Suppose you have a table employees with columns employee_id, department, salary, and gender. You want to count the number of male and female employees in each department who earn more than $60,000.

```
SELECT
  department,
  COUNT(CASE WHEN gender = 'Male' AND salary > 60000 THEN 1 END) AS high_earning_males,
  COUNT(CASE WHEN gender = 'Female' AND salary > 60000 THEN 1 END) AS high_earning_females
FROM
  employees
GROUP BY
  department;
```

Example 19: Conditional Sum with Date Comparison

Suppose you have a table subscriptions with columns subscription_id, start_date, end_date, and price. You want to calculate the total revenue from active subscriptions (where the current date is between start_date and end_date).

```
SELECT
  SUM(CASE WHEN GETDATE() BETWEEN start_date AND end_date THEN price ELSE 0 END) AS total_active_revenue
FROM
  subscriptions;
```

Example 20: Conditional Aggregation with Multiple Aggregates

Suppose you have a table transactions with columns transaction_id, account_id, transaction_type, and amount. You want to calculate the total deposits, withdrawals, and net balance (deposits - withdrawals) for each account.

```
SELECT
  account_id,
  SUM(CASE WHEN transaction_type = 'Deposit' THEN amount ELSE 0 END) AS total_deposits,
  SUM(CASE WHEN transaction_type = 'Withdrawal' THEN amount ELSE 0 END) AS total_withdrawals,
  SUM(CASE WHEN transaction_type = 'Deposit' THEN amount ELSE -amount END) AS net_balance
FROM
  transactions
GROUP BY
  account_id;
```