Domain-wise Data Analysis Query Sequences

CPG (Consumer Packaged Goods)

```
-- 1. Total sales overview
SELECT SUM("POS $ Sales") AS total_sales, SUM("POS Unit Sales") AS total_units FROM sales;

-- 2. Monthly/quarterly trend
SELECT Month, SUM("POS $ Sales") AS monthly_sales FROM sales GROUP BY Month ORDER BY Month;

-- 3. Category-level performance
SELECT Category, SUM("POS $ Sales") AS category_sales FROM sales GROUP BY Category ORDER BY category_sales DESC;

-- 4. Top Brands
SELECT Brand, SUM("POS $ Sales") AS brand_sales FROM sales GROUP BY Brand ORDER BY brand_sales DESC LIMIT 10;

-- 5. Price vs Units
SELECT Brand, AVG("Shelf Price") AS avg_price, SUM("POS Unit Sales") AS total_units FROM sales GROUP BY Brand;

-- 6. Region-wise sales
SELECT Region, SUM("POS $ Sales") AS regional_sales FROM sales GROUP BY Region;

-- 7. YOY Growth
SELECT Year, SUM("POS $ Sales") AS sales FROM sales GROUP BY Year ORDER BY Year;

-- 8. Promotion vs Regular Sales
SELECT IsPromoted, SUM("POS $ Sales") FROM sales GROUP BY IsPromoted;
```

Healthcare

```
-- 1. Total patient visits
SELECT COUNT(*) AS total_visits FROM patient_visits;

-- 2. Monthly patient trend
SELECT MONTH(visit_date) AS month, COUNT(*) AS patient_count FROM patient_visits GROUP BY month;

-- 3. Diagnosis distribution
SELECT diagnosis_code, COUNT(*) AS case_count FROM patient_visits GROUP BY diagnosis_code ORDER BY case_count DESC;

-- 4. Doctor performance
SELECT doctor_id, COUNT(*) AS patients_seen FROM patient_visits GROUP BY doctor_id;
```

Domain-wise Data Analysis Query Sequences

```
-- 5. Revenue by department
SELECT department, SUM(bill_amount) AS dept_revenue FROM billing GROUP BY department;

-- 6. Average cost per treatment
SELECT treatment_type, AVG(bill_amount) AS avg_cost FROM billing GROUP BY
treatment_type;

-- 7. Insurance vs Cash
SELECT payment_type, COUNT(*) AS count FROM billing GROUP BY payment_type;
```

Automobile

```
-- 1. Total car sales
SELECT COUNT(*) AS total_sales, SUM(price) AS total_revenue FROM sales;

-- 2. Sales trend by month
SELECT MONTH(sale_date) AS month, COUNT(*) AS cars_sold FROM sales GROUP BY month;

-- 3. Top selling models
SELECT model, COUNT(*) AS units_sold FROM sales GROUP BY model ORDER BY units_sold DESC;

-- 4. Sales by region
SELECT region, COUNT(*) AS regional_sales FROM sales GROUP BY region;

-- 5. Price segmentation
SELECT model, AVG(price) AS avg_price FROM sales GROUP BY model ORDER BY avg_price DESC;

-- 6. Defect reports by model
SELECT model, COUNT(*) AS defect_count FROM defects GROUP BY model;

-- 7. Time to service (after sale)
SELECT model, AVG(DATEDIFF(service_date, sale_date)) AS avg_days_to_service FROM
service_records GROUP BY model;
```

Retail

```
-- 1. Total sales and orders
SELECT COUNT(order_id) AS total_orders, SUM(total_amount) AS total_sales FROM orders;

-- 2. Sales by product category
SELECT category, SUM(total_amount) AS category_sales FROM orders GROUP BY category;

-- 3. Monthly sales trend
SELECT MONTH(order_date) AS month, SUM(total_amount) AS monthly_sales FROM orders GROUP
BY month;
```

Domain-wise Data Analysis Query Sequences

```
-- 4. Top customers by revenue
SELECT customer_id, SUM(total_amount) AS revenue FROM orders GROUP BY customer_id ORDER BY revenue DESC LIMIT 10;

-- 5. Sales by region
SELECT region, SUM(total_amount) AS regional_sales FROM orders GROUP BY region;

-- 6. Payment method split
SELECT payment_method, COUNT(*) AS count FROM orders GROUP BY payment_method;
```

E-Commerce

```
-- 1. Total orders and revenue
SELECT COUNT(order_id) AS total_orders, SUM(order_value) AS total_revenue FROM ecommerce_orders;

-- 2. Conversion rate
SELECT (COUNT(paid_orders)/COUNT(total_visits)) * 100 AS conversion_rate FROM site_metrics;

-- 3. Revenue by product category
SELECT category, SUM(order_value) AS revenue FROM ecommerce_orders GROUP BY category;

-- 4. Cart abandonment rate
SELECT (COUNT(abandoned_carts)/COUNT(started_carts)) * 100 AS abandonment_rate FROM cart_data;

-- 5. Top selling SKUs
SELECT sku, COUNT(*) AS sales_volume FROM ecommerce_orders GROUP BY sku ORDER BY sales_volume DESC LIMIT 10;
```

Finance / Banking

```
-- 1. Total transactions and volume
SELECT COUNT(*) AS transaction_count, SUM(amount) AS total_volume FROM transactions;

-- 2. Monthly transaction trend
SELECT MONTH(transaction_date) AS month, SUM(amount) AS monthly_volume FROM transactions GROUP BY month;

-- 3. Average account balance
SELECT account_type, AVG(balance) AS avg_balance FROM accounts GROUP BY account_type;

-- 4. Loan approval rate
SELECT (COUNT(CASE WHEN status='Approved' THEN 1 END) / COUNT(*)) * 100 AS approval_rate
```

Domain-wise Data Analysis Query Sequences

```
FROM loan_applications;

-- 5. Fraudulent transaction rate
SELECT (COUNT(CASE WHEN is_fraud=1 THEN 1 END) / COUNT(*)) * 100 AS fraud_rate FROM transactions;
```

Education / EdTech

```
-- 1. Total enrolled students
SELECT COUNT(*) AS total_students FROM students;

-- 2. Course popularity
SELECT course_name, COUNT(*) AS enrollment_count FROM enrollments GROUP BY course_name;

-- 3. Completion rate by course
SELECT course_name, (COUNT(CASE WHEN status='Completed' THEN 1 END)/COUNT(*)) * 100 AS completion_rate FROM enrollments GROUP BY course_name;

-- 4. Average grades
SELECT course_name, AVG(grade) AS avg_grade FROM grades GROUP BY course_name;

-- 5. Monthly sign-ups
SELECT MONTH(signup_date) AS month, COUNT(*) AS signups FROM students GROUP BY month;
```

Logistics / Supply Chain

```
-- 1. Total shipments and delivery success rate
SELECT COUNT(*) AS total_shipments,
       (COUNT(CASE WHEN status='Delivered' THEN 1 END)/COUNT(*)) * 100 AS success_rate
FROM shipments;

-- 2. Average delivery time
SELECT AVG(DATEDIFF(delivery_date, shipment_date)) AS avg_delivery_days FROM shipments WHERE status='Delivered';

-- 3. Top routes by volume
SELECT origin, destination, COUNT(*) AS shipment_count FROM shipments GROUP BY origin, destination ORDER BY shipment_count DESC;

-- 4. Shipment delays by carrier
SELECT carrier, COUNT(*) AS delayed_shipments
FROM shipments
WHERE DATEDIFF(delivery_date, expected_date) > 0
GROUP BY carrier;
```