

```

1  -- Find missing values in key columns
2  SELECT * FROM Customers WHERE name IS NULL OR email IS NULL;
3  SELECT * FROM Products WHERE name IS NULL OR price IS NULL;
4  SELECT * FROM Sales WHERE total_price IS NULL;
5
6  -- Find duplicate customers (same email)
7  SELECT email, COUNT(*)
8  FROM Customers
9  GROUP BY email
10 HAVING COUNT(*) > 1;
11
12
13 --Total Sales Revenue
14 SELECT SUM(total_price) AS total_revenue FROM Sales;
15
16
17 --Best-Selling Products
18 SELECT
19 P.name AS product_name,
20 P.category,
21 SUM(S.quantity) AS total_units_sold,
22 SUM(S.total_price) AS total_revenue
23 FROM Sales S
24 JOIN Products P ON S.product_id = P.product_id
25 GROUP BY P.name, P.category
26 ORDER BY total_units_sold DESC LIMIT 10;
27
28
29
30 --Top Customers by Spending
31 SELECT
32 C.name AS customer_name,
33 C.location,
34 SUM(S.total_price) AS total_spent
35 FROM Sales S
36 JOIN Customers C ON S.customer_id = C.customer_id
37 GROUP BY C.name, C.location
38 ORDER BY total_spent DESC
39 LIMIT 10;
40
41
42 --Customer Purchase Frequency
43 SELECT
44 C.name AS customer_name,
45 COUNT(S.sale_id) AS purchase_count
46 FROM Sales S
47 JOIN Customers C ON S.customer_id = C.customer_id
48 GROUP BY C.name
49 ORDER BY purchase_count DESC
50 LIMIT 10;
51
52
53 --Top Sales Employees
54 SELECT
55 E.name AS employee_name, E.role,
56 COUNT(S.sale_id) AS total_sales,
57 SUM(S.total_price) AS total_revenue_generated
58 FROM Sales S
59 JOIN Employees E ON S.employee_id = E.employee_id
60 GROUP BY E.name, E.role
61 ORDER BY total_revenue_generated DESC
62 LIMIT 5;
63
64
65 --Employee Salary vs. Sales Performance
66 SELECT E.name AS employee_name, E.salary,
67 COUNT(S.sale_id) AS sales_count,
68 SUM(S.total_price) AS revenue_generated
69 FROM Employees E
70 LEFT JOIN Sales S ON E.employee_id = S.employee_id
71 GROUP BY E.name, E.salary
72 ORDER BY revenue_generated DESC;

```

```
73
74 --Top-Performing Stores
75 SELECT ST.name AS store_name, ST.location,
76 SUM(S.total_price) AS total_sales
77 FROM Sales S
78 JOIN Stores ST ON S.store_id = ST.store_id
79 GROUP BY ST.name, ST.location
80 ORDER BY total_sales DESC
81 LIMIT 5;
82
83
84 --Store Revenue Contribution
85 SELECT ST.name AS store_name, ST.revenue,
86 SUM(S.total_price) AS total_sales,
87 (SUM(S.total_price) / (SELECT SUM(total_price) FROM Sales)) * 100 AS sales_percentage
88 FROM Sales S
89 JOIN Stores ST ON S.store_id = ST.store_id
90 GROUP BY ST.name, ST.revenue
91 ORDER BY sales_percentage DESC;
92
93
94 --Low Stock Products
95 SELECT
96 name AS product_name, stock_quantity
97 FROM Products
98 WHERE stock_quantity < 10
99 ORDER BY stock_quantity ASC;
100
101
102 --Product Turnover Rate (Most Frequently Sold Items)
103 SELECT P.name AS product_name,
104 P.category,
105 SUM(S.quantity) AS total_sold
106 FROM Sales S
107 JOIN Products P ON S.product_id = P.product_id
108 GROUP BY P.name, P.category
109 ORDER BY total_sold DESC
110 LIMIT 10;
111
```