

I. Basic Queries (1–10)**

1. List all customers with their phone numbers and email addresses.

```
SELECT name, phone, email FROM customers;
```

2. Show all products under the “Beverages” category.

```
SELECT p.productName, p.brand, p.sellingPrice  
FROM products p  
JOIN categories c ON p.categoryID = c.categoryID  
WHERE c.categoryName = 'Beverages';
```

3. List all staff in the “Mirpur Branch, Dhaka.”

```
SELECT name, designation, phone  
FROM staff  
WHERE address LIKE '%Mirpur%';
```

4. Show all suppliers whose payment status is “Advance.”

```
SELECT supplierName, contactPerson, totalAmount, paymentStatus  
FROM suppliers  
WHERE paymentStatus = 'Advance';
```

5. Find all active users with role “Manager.”

```
SELECT username, email  
FROM users  
WHERE roleID = 2 AND isActive = TRUE;
```

6. Display products whose selling price is greater than 200.

```
SELECT productName, sellingPrice  
FROM products  
WHERE sellingPrice > 200;
```

7. List all categories available in the store.

```
SELECT categoryID, categoryName  
FROM categories;
```

8. Show last 5 sales in the system.

```
SELECT *  
FROM sales  
ORDER BY saleDate DESC  
LIMIT 5;
```

9. List customers who are VIP.

```
SELECT name, customerType, customerStatus  
FROM customers  
WHERE customerStatus = 'VIP';
```

10. Display all discounts of type “Product.”

```
SELECT *  
FROM discounts
```

WHERE discountType = 'Product';

##II. Intermediate Queries (11–25)**

11. Calculate total sales for each customer.

```
SELECT c.name, SUM(s.finalAmount) AS totalSales
FROM sales s
JOIN customers c ON s.customerID = c.customerID
GROUP BY c.customerID;
```

12. Find products with stock less than 50.

```
SELECT productName, stock
FROM products
WHERE stock < 50;
```

13. Show purchase items for supplier “Rahman Traders.”

```
SELECT p.productName, pi.quantity, pi.unitPrice
FROM purchaseItems pi
JOIN purchases pu ON pi.purchaseID = pu.purchaseID
JOIN suppliers s ON pu.supplierID = s.supplierID
JOIN products p ON pi.productID = p.productID
WHERE s.supplierName = 'Rahman Traders';
```

14. Display staff and their total salary received this year.

```
SELECT st.name, SUM(sp.totalAmount) AS totalPaid
```

```
FROM staff st
JOIN salaryPayments sp ON st.staffID = sp.staffID
WHERE YEAR(sp.paymentDate) = YEAR(CURRENT_DATE())
GROUP BY st.staffID;
```

15. Show sales items for saleID = 5.

```
SELECT p.productName, si.quantity, si.unitPrice
FROM saleItems si
JOIN products p ON si.productID = p.productID
WHERE si.saleID = 5;
```

16. List all sales returns in the last month.

```
SELECT sr.returnID, p.productName, sr.quantity, sr.returnDate
FROM salesReturns sr
JOIN products p ON sr.productID = p.productID
WHERE sr.returnDate >= DATE_SUB(CURRENT_DATE(), INTERVAL 1 MONTH);
```

17. Show customers with more than 1000 points.

```
SELECT c.name, cp.points
FROM customerPoints cp
JOIN customers c ON cp.customerID = c.customerID
WHERE cp.points > 1000;
```

18. List purchases with a discount greater than 500.

```
SELECT purchaseID, supplierID, discount, totalAmount
FROM purchases
```

WHERE discount > 500;

19. Display staff working flexible shifts.

```
SELECT name, shift, designation
FROM staff
WHERE shift = 'Flexible';
```

20. Show payments made using Bank Transfer.

```
SELECT paymentID, relatedType, amount, paymentDate
FROM payments
JOIN paymentMethods pm ON payments.methodID = pm.methodID
WHERE pm.methodName = 'Bank Transfer';
```

21. Find all inventory changes marked as “OUT.”

```
SELECT im.inventoryID, p.productName, im.quantity, im.changeDate
FROM inventoryManagement im
JOIN products p ON im.productID = p.productID
WHERE im.changeType = 'OUT';
```

22. List top 5 products by sales quantity.

```
SELECT p.productName, SUM(si.quantity) AS totalSold
FROM saleItems si
JOIN products p ON si.productID = p.productID
GROUP BY si.productID
ORDER BY totalSold DESC
LIMIT 5;
```

23. Show staff whose salary is greater than 50,000.

```
SELECT name, salary, designation
FROM staff
WHERE salary > 50000;
```

24. Display all customers with sales pending.

```
SELECT DISTINCT c.name, s.paymentStatus
FROM sales s
JOIN customers c ON s.customerID = c.customerID
WHERE s.paymentStatus = 'PENDING';
```

25. Show purchase items with unitPrice > 1000.

```
SELECT pi.purchaseItemID, p.productName, pi.unitPrice
FROM purchaseItems pi
JOIN products p ON pi.productID = p.productID
WHERE pi.unitPrice > 1000;
```

III. Advanced Queries (26–40)**

26. Find top 3 customers by total sales in 2025.

```
SELECT c.name, SUM(s.finalAmount) AS totalSales
FROM sales s
JOIN customers c ON s.customerID = c.customerID
WHERE YEAR(s.saleDate) = 2025
```

```
GROUP BY c.customerID
ORDER BY totalSales DESC
LIMIT 3;
```

27. Get products with highest returns.

```
SELECT p.productName, SUM(sr.quantity) AS totalReturned
FROM salesReturns sr
JOIN products p ON sr.productID = p.productID
GROUP BY p.productID
ORDER BY totalReturned DESC;
```

28. Total expenses paid via “Card” this month.

```
SELECT SUM(e.amount) AS totalExpenses
FROM expenses e
JOIN paymentMethods pm ON e.paymentMethodID = pm.methodID
WHERE pm.methodName = 'Card' AND MONTH(e.expenseDate) =
MONTH(CURRENT_DATE());
```

29. Staff who haven’t received salary payments in last 2 months.

```
SELECT s.name, s.staffID
FROM staff s
LEFT JOIN salaryPayments sp ON s.staffID = sp.staffID AND sp.paymentDate >=
DATE_SUB(CURRENT_DATE(), INTERVAL 2 MONTH)
WHERE sp.salaryID IS NULL;
```

30. Active product discounts currently valid.

```
SELECT p.productName, d.percentage, d.startDate, d.endDate
```

```
FROM discounts d
JOIN products p ON d.productID = p.productID
WHERE CURRENT_DATE() BETWEEN d.startDate AND d.endDate;
```

31. Total sales and number of items sold per product.

```
SELECT p.productName, SUM(si.quantity) AS totalQuantity,
SUM(si.unitPrice*si.quantity) AS totalSales
FROM saleItems si
JOIN products p ON si.productID = p.productID
GROUP BY si.productID;
```

32. Total purchases per supplier this year.

```
SELECT s.supplierName, SUM(pu.totalAmount) AS totalPurchases
FROM purchases pu
JOIN suppliers s ON pu.supplierID = s.supplierID
WHERE YEAR(pu.purchaseDate) = YEAR(CURRENT_DATE())
GROUP BY s.supplierID;
```

33. Staff with total salary paid above 100,000 in a year.

```
SELECT st.name, SUM(sp.totalAmount) AS totalPaid
FROM staff st
JOIN salaryPayments sp ON st.staffID = sp.staffID
WHERE YEAR(sp.paymentDate) = 2025
GROUP BY st.staffID
HAVING totalPaid > 100000;
```

34. Customers who bought both “Fruits” and “Dairy” products.


```
SELECT DISTINCT c.name
FROM sales s
JOIN saleItems si ON s.saleID = si.saleID
JOIN products p ON si.productID = p.productID
JOIN categories c1 ON p.categoryID = c1.categoryID
JOIN customers c ON s.customerID = c.customerID
WHERE c1.categoryName IN ('Fruits','Dairy')
GROUP BY c.customerID
HAVING COUNT(DISTINCT c1.categoryName) = 2;
```

35. Average selling price per category.

```
SELECT c.categoryName, AVG(p.sellingPrice) AS avgPrice
FROM products p
JOIN categories c ON p.categoryID = c.categoryID
GROUP BY c.categoryID;
```

36. List inventory changes per product in last 30 days.

```
SELECT p.productName, im.changeType, SUM(im.quantity) AS totalChanged
FROM inventoryManagement im
JOIN products p ON im.productID = p.productID
WHERE im.changeDate >= DATE_SUB(CURRENT_DATE(), INTERVAL 30 DAY)
GROUP BY im.productID, im.changeType;
```

37. Total discount applied per invoice.

```
SELECT s.saleID, SUM(d.percentage * si.unitPrice * si.quantity / 100) AS totalDiscount
FROM saleItems si
JOIN sales s ON si.saleID = s.saleID
JOIN discounts d ON si.productID = d.productID
WHERE CURRENT_DATE() BETWEEN d.startDate AND d.endDate
GROUP BY s.saleID;
```

38. Customers who have not purchased anything in 2025.

```
SELECT c.name
FROM customers c
LEFT JOIN sales s ON c.customerID = s.customerID AND YEAR(s.saleDate) = 2025
WHERE s.saleID IS NULL;
```

39. Products with maximum stock available.

```
SELECT productName, stock
FROM products
ORDER BY stock DESC
LIMIT 5;
```

40. Total amount collected by payment method this month.

```
SELECT pm.methodName, SUM(p.amount) AS totalCollected
FROM payments p
JOIN paymentMethods pm ON p.methodID = pm.methodID
WHERE MONTH(p.paymentDate) = MONTH(CURRENT_DATE())
GROUP BY pm.methodID;
```