1.
```
SELECT account_id, balance,
  CASE
    WHEN balance < 50000 THEN 'Low'
    WHEN balance BETWEEN 50000 AND 200000 THEN 'Medium'
    ELSE 'High'
  END AS balance_category
FROM accounts;
```

2.
```
SELECT customer_id, SUM(balance) AS total_balance,
      RANK() OVER (ORDER BY SUM(balance) DESC) AS rank
FROM accounts
GROUP BY customer_id
LIMIT 5;
```

3.
```
SELECT account_id, balance,
  CASE
    WHEN balance > AVG(balance) OVER () THEN 'Above Average'
    WHEN balance < AVG(balance) OVER () THEN 'Below Average'
    ELSE 'Average'
  END AS balance_position
FROM accounts;
```

4.
```
SELECT account_id, transaction_type, amount, timestamp
FROM (
  SELECT *,
       ROW_NUMBER() OVER (PARTITION BY account_id ORDER BY timestamp ASC) AS
rn_asc,
       ROW_NUMBER() OVER (PARTITION BY account_id ORDER BY timestamp DESC) AS
rn_desc
  FROM transactions
) t
WHERE rn_asc = 1 OR rn_desc = 1
ORDER BY account_id, timestamp;
```

5.
```
WITH recent_txns AS (
  SELECT * FROM transactions
  WHERE timestamp >= CURRENT_DATE - INTERVAL '30 days'
)
SELECT account_id,
     SUM(CASE WHEN transaction_type = 'Credit' THEN amount ELSE 0 END) AS total_credit,
     SUM(CASE WHEN transaction_type = 'Debit' THEN amount ELSE 0 END) AS total_debit
FROM recent_txns
GROUP BY account_id;
```

6.
```
SELECT loan_id, customer_id, loan_type, status,
  CASE
    WHEN status = 'Defaulted' THEN 'High Risk'
    WHEN status = 'Active' THEN 'Medium Risk'
    WHEN status = 'Closed' THEN 'Low Risk'
    ELSE 'Unknown'
  END AS risk_score
FROM loans;
```
7.
```
SELECT *
FROM (
  SELECT *, ROW_NUMBER() OVER (PARTITION BY account_id ORDER BY timestamp DESC) AS
txn_rank
  FROM transactions
) sub
WHERE txn_rank <= 3;
```
8.
```
SELECT a.account_id, a.customer_id, a.balance
FROM accounts a
WHERE NOT EXISTS (
  SELECT 1 FROM transactions t WHERE t.account_id = a.account_id
);
```

9.
```
SELECT loan_id, loan_type, loan_amount,
      (SELECT AVG(l2.loan_amount)
       FROM loans l2
       WHERE l2.loan_type = l1.loan_type) AS avg_by_type,
      loan_amount -
      (SELECT AVG(l2.loan_amount)
       FROM loans l2
       WHERE l2.loan_type = l1.loan_type) AS diff_from_avg
FROM loans l1;
```

10.
```
SELECT DATE(timestamp) AS txn_date,
      SUM(CASE WHEN transaction_type = 'Credit' THEN amount ELSE 0 END) AS total_credit,
      SUM(CASE WHEN transaction_type = 'Debit' THEN amount ELSE 0 END) AS total_debit,
      SUM(CASE WHEN transaction_type = 'Credit' THEN amount ELSE 0 END) -
      SUM(CASE WHEN transaction_type = 'Debit' THEN amount ELSE 0 END) AS net_flow
FROM transactions
WHERE timestamp >= CURRENT_DATE - INTERVAL '7 days'
GROUP BY DATE(timestamp)
ORDER BY txn_date;
```