

--1. Mostrar todos os clientes com saldo entre 10000 e 20000.

```
SELECT c.cliente_cod, c.cliente_nome, ct.conta_numero, ct.saldo
FROM cliente c
JOIN conta ct ON c.cliente_cod = ct.cliente_cliente_cod
WHERE ct.saldo BETWEEN 10000 AND 20000;
```

--2. Mostrar todos os clientes com empréstimos.

```
SELECT c.cliente_cod, c.cliente_nome, e.emprestimo_numero, e.quantia
FROM cliente c
JOIN emprestimo e ON c.cliente_cod = e.cliente_cliente_cod;
```

--3. Mostrar todos os clientes que moram em Niterói.

```
SELECT cliente_cod, cliente_nome, rua, cidade
FROM cliente
WHERE cidade = 'Niterói';
```

--4. Encontre os nomes e cidades de clientes que possuam empréstimos em alguma agência

```
SELECT cliente_nome, cidade
FROM cliente
WHERE cliente_cod IN (
    SELECT cliente_cliente_cod
    FROM emprestimo
);
```

--5. Mostrar o nome dos clientes que possuem conta, empréstimo ou ambos na agência de código 1

```
SELECT DISTINCT cliente.cliente_nome
FROM cliente
WHERE cliente.cliente_cod IN (
    SELECT conta.cliente_cliente_cod
    FROM conta
    WHERE conta.agencia_agencia_cod = 1
    UNION
    SELECT emprestimo.cliente_cliente_cod
    FROM emprestimo
    WHERE emprestimo.agencia_agencia_cod = 1
);
```

--6. Achar todos os clientes que possuam uma conta e um empréstimo na agência de código 1

```
SELECT cliente.cliente_nome
FROM cliente
WHERE cliente.cliente_cod IN (
    SELECT conta.cliente_cliente_cod
    FROM conta
    WHERE conta.agencia_agencia_cod = 1
)
AND cliente.cliente_cod IN (
    SELECT emprestimo.cliente_cliente_cod
```

```
FROM emprestimo
WHERE emprestimo.agencia_agencia_cod = 1
);
```

--7. Achar todos os clientes que possuem uma conta mas não possuem um empréstimo na agência de código 1

```
SELECT cliente.cliente_nome
FROM cliente
WHERE cliente.cliente_cod IN (
    SELECT conta.cliente_cliente_cod
    FROM conta
    WHERE conta.agencia_agencia_cod = 1
)
AND cliente.cliente_cod NOT IN (
    SELECT emprestimo.cliente_cliente_cod
    FROM emprestimo
    WHERE emprestimo.agencia_agencia_cod = 1
);
```

--8. Liste todas as cidades onde há agências

```
SELECT DISTINCT agencia_cidade
FROM agencia;
```

--9. Liste todas as agências existentes em 'Rio de Janeiro'

```
SELECT agencia_cod, agencia_nome, fundos
FROM agencia
WHERE agencia_cidade = 'Rio de Janeiro';
```

--10. Liste todas as agências que possuem conta com saldo maior que 100.000,00

```
SELECT DISTINCT a.agencia_cod, a.agencia_nome, a.agencia_cidade, a.fundos
FROM agencia a
JOIN conta c ON a.agencia_cod = c.agencia_agencia_cod
WHERE c.saldo > 100000;
```

--11. Mostre os dados de todos os clientes da agência 'Niterói' da cidade de 'Niterói' que têm saldo negativo

```
SELECT c.cliente_cod, c.cliente_nome, c.rua, c.cidade
FROM cliente c
JOIN conta ct ON c.cliente_cod = ct.cliente_cliente_cod
JOIN agencia a ON ct.agencia_agencia_cod = a.agencia_cod
WHERE a.agencia_nome LIKE '%Niterói%' AND a.agencia_cidade = 'Niterói' AND ct.saldo < 0;
```

--12. Liste os dados dos clientes que possuem conta mas não possuem empréstimo na agência de código 5

```
SELECT c.cliente_cod, c.cliente_nome, c.rua, c.cidade
FROM cliente c
WHERE c.cliente_cod IN (
    SELECT ct.cliente_cliente_cod
```

```

FROM conta ct
WHERE ct.agencia_agencia_cod = 5
)
AND c.cliente_cod NOT IN (
    SELECT e.cliente_cliente_cod
    FROM emprestimo e
    WHERE e.agencia_agencia_cod = 5
);

```

--13. Selecione todas as agências com código 1, 2 ou 3

```

SELECT agencia_cod, agencia_nome, agencia_cidade, fundos
FROM agencia
WHERE agencia_cod IN (1, 2, 3);

```

--14. Encontre todos os clientes que possuem uma conta e um empréstimo na agência 'Nova Iguaçu'

```

SELECT c.cliente_nome
FROM cliente c
WHERE c.cliente_cod IN (
    SELECT ct.cliente_cliente_cod
    FROM conta ct
    JOIN agencia a ON ct.agencia_agencia_cod = a.agencia_cod
    WHERE a.agencia_nome LIKE '%Nova Iguaçu%'
)
AND c.cliente_cod IN (
    SELECT e.cliente_cliente_cod
    FROM emprestimo e
    JOIN agencia a ON e.agencia_agencia_cod = a.agencia_cod
    WHERE a.agencia_nome LIKE '%Nova Iguaçu%'
);

```

--15. Encontre todas as agências que possuem ativos maiores que alguma agência de Angra dos Reis

```

SELECT agencia_cod, agencia_nome, agencia_cidade, fundos
FROM agencia
WHERE fundos > (
    SELECT MAX(fundos)
    FROM agencia
    WHERE agencia_cidade = 'Angra dos Reis'
);

```

--16. Encontrar as agências que possuem ativos maiores do que todas as agências de Resende

```

SELECT agencia_cod, agencia_nome, agencia_cidade, fundos
FROM agencia
WHERE fundos > (
    SELECT MAX(fundos)
    FROM agencia
    WHERE agencia_cidade = 'Resende'
);

```

--17. Usando a construção EXISTS, encontre todos os clientes que possuem uma conta e um empréstimo na agência 'Petrópolis'

```

SELECT c.cliente_nome
FROM cliente c
WHERE EXISTS (
    SELECT 1
    FROM conta ct
    JOIN agencia a ON ct.agencia_agencia_cod = a.agencia_cod
    WHERE ct.cliente_cliente_cod = c.cliente_cod AND a.agencia_nome LIKE '%Petrópolis%'
)
AND EXISTS (
    SELECT 1
    FROM emprestimo e
    JOIN agencia a ON e.agencia_agencia_cod = a.agencia_cod
    WHERE e.cliente_cliente_cod = c.cliente_cod AND a.agencia_nome LIKE '%Petrópolis%'
);

```

--18. Encontre o saldo médio de conta em cada agência

```

SELECT agencia_agencia_cod, AVG(saldo) AS saldo_medio
FROM conta
GROUP BY agencia_agencia_cod;

```

--19. Encontre o número de depositantes de cada agência

```

SELECT agencia_agencia_cod, COUNT(DISTINCT cliente_cliente_cod) AS numero_depositantes
FROM conta
GROUP BY agencia_agencia_cod;

```

--20. Encontre o maior saldo de cada agência

```

SELECT agencia_agencia_cod, MAX(saldo) AS maior_saldo
FROM conta
GROUP BY agencia_agencia_cod;

```

--21. Encontre a média de saldos de todas as contas

```

SELECT AVG(saldo) AS media_saldos
FROM conta;

```

--22. Remover todos os registros da conta de 'João'

```

DELETE FROM conta
WHERE cliente_cliente_cod IN (
    SELECT cliente_cod
    FROM cliente
    WHERE cliente_nome = 'João'
);

```

--23. Remova todas as contas de agências localizadas em 'Vitória'

```

DELETE FROM conta
WHERE agencia_agencia_cod IN (
    SELECT agencia_cod
    FROM agencia
    WHERE agencia_cidade = 'Vitória'
);

```

--24. Inserir uma nova conta para o cliente (código = 1), número 9000, na agência de código=2 cujo valor seja 1200

```
INSERT INTO conta (conta_numero, saldo, cliente_cliente_cod, agencia_agencia_cod)
VALUES (9000, 1200, 1, 2);
```

--25. Inserir todos os clientes que possuam empréstimos na agência 'Macaé' na relação CONTA com um saldo de 200. O número da nova conta é o número do empréstimo \* 3

```
INSERT INTO conta (conta_numero, saldo, cliente_cliente_cod, agencia_agencia_cod)
SELECT emprestimo_numero * 3, 200, cliente_cliente_cod, agencia_agencia_cod
FROM emprestimo
WHERE agencia_agencia_cod = (
    SELECT agencia_cod
    FROM agencia
    WHERE agencia_nome LIKE '%Macaé%'
);
```

--26. Suponha que esteja sendo feito o pagamento de juros, e que em todos saldos sejam acrescentados em 5%

```
UPDATE conta
SET saldo = saldo * 1.05;
```

--27. Suponha que todas as contas com saldo superiores a 10000 recebam aumento de 6% e as demais de 5%

```
UPDATE conta
SET saldo = CASE
    WHEN saldo > 10000 THEN saldo * 1.06
    ELSE saldo * 1.05
END;
```

--28. Considere, por exemplo, que todas as contas de pessoas que possuem empréstimos no banco terão acréscimo de 1%

```
UPDATE conta
SET saldo = saldo * 1.01
WHERE cliente_cliente_cod IN (
    SELECT cliente_cliente_cod
    FROM emprestimo
);
```

--29. Atualize o valor dos ativos. Os ativos são os valores dos saldos das contas da agência

```
UPDATE agencia a
SET fundos = (
    SELECT SUM(c.saldo)
    FROM conta c
    WHERE c.agencia_agencia_cod = a.agencia_cod
);
```

--30. Considere a visão consistindo em nomes de agências e de clientes

```
CREATE VIEW agencia_cliente_view AS
SELECT a.agencia_nome, c.cliente_nome
FROM agencia a
JOIN conta ct ON a.agencia_cod = ct.agencia_agencia_cod
JOIN cliente c ON ct.cliente_cliente_cod = c.cliente_cod;
```

--31. Usando a visão agencia\_cliente\_view, podemos achar todos os clientes da agência 'Rio de Janeiro'

```
SELECT cliente_nome
FROM agencia_cliente_view
WHERE agencia_nome LIKE '%Rio de Janeiro%';
```

--32. Selecione todos os clientes que possuem contas em agência(s) que possui(m) o maior ativo

```
SELECT DISTINCT c.cliente_nome
FROM cliente c
JOIN conta ct ON c.cliente_cod = ct.cliente_cliente_cod
JOIN agencia a ON ct.agencia_agencia_cod = a.agencia_cod
WHERE a.fundos = (
    SELECT MAX(fundos)
    FROM agencia
);
```

--33. Selecione o total de agências por cidade, classificado por cidade

```
SELECT agencia_cidade, COUNT(*) AS total_agencias
FROM agencia
GROUP BY agencia_cidade
ORDER BY agencia_cidade;
```

--34. Selecione o valor médio de empréstimos efetuados por cada agência em ordem crescente das cidades onde estas agências se situam

```
SELECT a.agencia_cidade, AVG(e.quantia) AS media_emprestimos
FROM emprestimo e
JOIN agencia a ON e.agencia_agencia_cod = a.agencia_cod
GROUP BY a.agencia_cidade
ORDER BY a.agencia_cidade;
```

--35. Selecione a(s) agência(s) que possui(m) a maior média de quantia emprestada

```
SELECT agencia_cod, agencia_nome, AVG(quantia) AS media_emprestada
FROM emprestimo
GROUP BY agencia_cod, agencia_nome
HAVING AVG(quantia) = (
    SELECT MAX(AVG(quantia))
    FROM emprestimo
    GROUP BY agencia_agencia_cod
);
```

--36. Selecione todas as agências situadas fora de Rio de Janeiro que possuem a média de depósitos maior do que alguma agência localizada em Duque de Caxias

```
SELECT a.agencia_cod, a.agencia_nome, a.agencia_cidade
FROM agencia a
```

```

WHERE a.agencia_cidade <> 'Rio de Janeiro'
AND (
    SELECT AVG(ct.saldo)
    FROM conta ct
    WHERE ct.agencia_agencia_cod = a.agencia_cod
) > (
    SELECT MIN(AVG(ct.saldo))
    FROM conta ct
    JOIN agencia ag ON ct.agencia_agencia_cod = ag.agencia_cod
    WHERE ag.agencia_cidade = 'Duque de Caxias'
    GROUP BY ag.agencia_cod
);

```

--37. Selecione o menor saldo de clientes, por agências

```

SELECT agencia_agencia_cod, MIN(saldo) AS menor_saldo
FROM conta
GROUP BY agencia_agencia_cod;

```

--38. Selecione o saldo de cada cliente, caso ele possua mais de uma conta no banco

```

SELECT cliente_cliente_cod, SUM(saldo) AS total_saldo
FROM conta
GROUP BY cliente_cliente_cod
HAVING COUNT(*) > 1;

```