

Atividade Prática – SQL com Oracle APEX

Seções 7 a 9 – Oracle Academy: Database Programming with SQL

Tema: Junções, funções de agregação, agrupamentos e operadores de conjunto. Ferramenta: Oracle APEX – SQL Workshop

Base de dados utilizada: Sistema bancário com tabelas agencia, cliente, conta, emprestimo.

Parte 1 – Junções e Produto Cartesiano (Seção 7)

1. Usando a sintaxe proprietária da Oracle, exiba o nome de cada cliente junto com o número de sua conta.

```
SELECT C.CLIENTE_NOME, CT.CONTA_NUMERO  
  
FROM CLIENTE C, CONTA CT  
  
WHERE C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD
```

2. Mostre todas as combinações possíveis de clientes e agências (produto cartesiano).

```
SELECT C.CLIENTE_NOME, AG.AGENCIA_NOME  
  
FROM CLIENTE C  
  
CROSS JOIN AGENCIA AG
```

3. Usando aliases de tabela, exiba o nome dos clientes e a cidade da agência onde mantêm conta.

```
SELECT C.CLIENTE_NOME, AG.AGENCIA_NOME  
  
FROM CLIENTE C  
  
JOIN CONTA CT ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD  
  
JOIN AGENCIA AG ON CT.AGENCIA_AGENCIA_COD = AG.AGENCIA_COD
```

Parte 2 – Funções de Grupo, COUNT, DISTINCT e NVL (Seção 8)

4. Exiba o saldo total de todas as contas cadastradas.

```
SELECT SUM(CT.SALDO)  
FROM CONTA CT
```

5. Mostre o maior saldo e a média de saldo entre todas as contas.

```
SELECT MAX(CT.SALDO), AVG(CT.SALDO)  
FROM CONTA CT
```

6. Apresente a quantidade total de contas cadastradas.

```
SELECT COUNT(CT.CONTA_NUMERO)  
FROM CONTA CT
```

7. Liste o número de cidades distintas onde os clientes residem.

```
SELECT COUNT(DISTINCT C.CIDADE)  
FROM CLIENTE C
```

8. Exiba o número da conta e o saldo, substituindo valores nulos por zero.

```
SELECT CT.CONTA_NUMERO, NVL(CT.SALDO,0) AS SALDO  
FROM CONTA CT
```

Parte 3 – GROUP BY, HAVING, ROLLUP e Operadores de Conjunto

(Seção 9) 9. Exiba a média de saldo por cidade dos clientes.

```
SELECT C.CIDADE, AVG(CT.SALDO)
```

```
FROM CONTA CT
```

JOIN CLIENTE C ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD

GROUP BY C.CIDADE

10. Liste apenas as cidades com mais de 3 contas associadas a seus moradores.

```
SELECT C.CIDADE, COUNT(*)  
FROM CLIENTE C  
GROUP BY C.CIDADE  
HAVING COUNT(C.CIDADE) > 3
```

11. Utilize a cláusula ROLLUP para exibir o total de saldos por cidade da agência e o total geral.

```
SELECT AG.AGENCIA_CIDADE, SUM(CT.SALDO)  
FROM CONTA CT  
JOIN AGENCIA AG ON CT.AGENCIA_AGENCIA_COD = AG.AGENCIA_COD  
GROUP BY ROLLUP (AG.AGENCIA_CIDADE, CT.SALDO);
```

12. Faça uma consulta com UNION que combine os nomes de cidades dos clientes e das agências, sem repetições.

```
SELECT AG.AGENCIA_CIDADE  
FROM AGENCIA AG  
UNION  
SELECT C.CIDADE  
FROM CLIENTE C
```

Atividade Prática – SQL com Oracle APEX

Seção 10 – Subconsultas

Tema: Subconsultas de linha única, multilinha, correlacionadas, com EXISTS, NOT EXISTS e a cláusula WITH.

Ferramenta: Oracle APEX – SQL Workshop

Base de dados utilizada: Sistema bancário (agencia, cliente, conta, emprestimo)

Parte 1 – Subconsultas de Linha Única

1. Liste os nomes dos clientes cujas contas possuem saldo **acima da média geral** de todas as contas registradas.

```
SELECT C.CLIENTE_NOME  
  
FROM CLIENTE C  
  
JOIN CONTA CT ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD  
  
WHERE CT.SALDO >  
  
      (SELECT AVG(CT.SALDO)  
  
       FROM CONTA CT)
```

2. Exiba os nomes dos clientes cujos saldos são **iguais ao maior saldo** encontrado no banco.

```
SELECT C.CLIENTE_NOME  
  
FROM CLIENTE C  
  
JOIN CONTA CT ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD  
  
WHERE CT.SALDO =  
  
      (SELECT MAX(CT.SALDO)  
  
       FROM CONTA CT)
```

3. Liste as cidades onde a **quantidade de clientes** é maior que a quantidade **média de clientes por cidade**.

```
SELECT C.CIDADE, COUNT(*)
```

```

FROM CLIENTE C

GROUP BY C.CIDADE

HAVING COUNT(*) > (

    SELECT AVG(CIDADE)

    FROM (

        SELECT COUNT(*) AS CIDADE

        FROM CLIENTE C

        GROUP BY C.CIDADE

    )

);

```

Parte 2 – Subconsultas Multilinha

4. Liste os nomes dos clientes com saldo igual a **qualquer** um dos dez maiores saldos registrados.

```

SELECT C.CLIENTE_NOME, CT.SALDO
FROM CLIENTE C
JOIN CONTA CT ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD
WHERE CT.SALDO IN (

    SELECT DISTINCT SALDO

    FROM CONTA

    ORDER BY SALDO DESC

    FETCH FIRST 10 ROWS ONLY

);

```

5. Liste os clientes que possuem saldo **menor que todos** os saldos dos clientes da cidade de **Niterói**.

```

SELECT C.CLIENTE_NOME, CT.SALDO

```

```
FROM CLIENTE C
JOIN CONTA CT ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD
WHERE CT.SALDO < ALL (
    SELECT CT2.SALDO
    FROM CLIENTE C2
    JOIN CONTA CT2 ON C2.CLIENTE_COD = CT2.CLIENTE_CLIENTE_COD
    WHERE C2.CIDADE = 'NITERÓI'
);
```

6. Liste os clientes cujos saldos estão **entre os saldos de clientes de Volta Redonda**.

```
SELECT C.CLIENTE_NOME, CT.SALDO
FROM CLIENTE C
JOIN CONTA CT ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD
WHERE CT.SALDO BETWEEN
(
    SELECT MIN(CT2.SALDO)
    FROM CLIENTE C2
    JOIN CONTA CT2 ON C2.CLIENTE_COD = CT2.CLIENTE_CLIENTE_COD
    WHERE C2.CIDADE = 'VOLTA REDONDA'
)
AND
(
    SELECT MAX(CT2.SALDO)
    FROM CLIENTE C2
    JOIN CONTA CT2 ON C2.CLIENTE_COD = CT2.CLIENTE_CLIENTE_COD
    WHERE C2.CIDADE = 'VOLTA REDONDA'
);
```

Parte 3 – Subconsultas Correlacionadas

7. Exiba os nomes dos clientes cujos saldos são maiores que a **média de saldo das contas da mesma agência**.

```
SELECT C.CLIENTE_NOME, CT.SALDO, A.AGENCIA_COD
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 CT.SALDO > (
    SELECT AVG(CT2.SALDO)
    FROM CONTA CT2
    WHERE CT2.AGENCIA_AGENCIA_COD = CT.AGENCIA_AGENCIA_COD
);
```

8. Liste os nomes e cidades dos clientes que têm saldo **inferior à média de sua própria cidade**.

```
SELECT C.CLIENTE_NOME, C.CIDADE, CT.SALDO
FROM CLIENTE C
JOIN CONTA CT ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD
WHERE CT.SALDO < (
    SELECT AVG(CT2.SALDO)
    FROM CLIENTE C2
    JOIN CONTA CT2 ON C2.CLIENTE_COD = CT2.CLIENTE_CLIENTE_COD
    WHERE C2.CIDADE = C.CIDADE
);
```

Parte 4 – Subconsultas com EXISTS e NOT EXISTS

9. Liste os nomes dos clientes que **possuem pelo menos uma conta** registrada no banco.

```
SELECT C.CLIENTE_NOME
```

```
FROM CLIENTE C
```

```
WHERE EXISTS (
```

```
    SELECT 1
```

```
    FROM CONTA CT
```

```
    WHERE CT.CLIENTE_CLIENTE_COD = C.CLIENTE_COD
```

```
);
```

10. Liste os nomes dos clientes que **ainda não possuem conta** registrada no banco.

```
SELECT C.CLIENTE_NOME
```

```
FROM CLIENTE C
```

```
WHERE NOT EXISTS (
```

```
    SELECT 1
```

```
    FROM CONTA CT
```

```
    WHERE CT.CLIENTE_CLIENTE_COD = C.CLIENTE_COD
```

```
);
```


Parte 5 – Subconsulta Nomeada com WITH

11. Usando a cláusula WITH, calcule a **média de saldo por cidade** e exiba os clientes que possuem saldo **acima da média de sua cidade**.

```
WITH MEDIA_POR_CIDADE AS (  
    SELECT C.CIDADE, AVG(CT.SALDO) AS MEDIA_SALDO  
    FROM CLIENTE C  
    JOIN CONTA CT ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD  
    GROUP BY C.CIDADE  
)  
SELECT C.CLIENTE_NOME, C.CIDADE, CT.SALDO  
FROM CLIENTE C  
JOIN CONTA CT ON C.CLIENTE_COD = CT.CLIENTE_CLIENTE_COD  
JOIN MEDIA_POR_CIDADE M ON C.CIDADE = M.CIDADE  
WHERE CT.SALDO > M.MEDIA_SALDO;
```