

INSTITUTO FEDERAL

Goiano

Campus Morrinhos

BANCO DE DADOS

SQL SELECT

PROFESSOR MARCEL MELO

MARCEL.MELO@IFGOIANO.EDU.BR

O comando SELECT recupera os dados de uma ou mais tabelas, sendo um dos comandos mais simples e, ao mesmo tempo, mais extenso da SQL devido as suas funções, operandos, comandos, subcomandos e cláusulas não obrigatórias.

```
SELECT <lista-colunas-seleção>  
FROM <lista-tabelas>  
WHERE <condição>
```

SINTAXE COMPLETA DO COMANDO SELECT

```
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]  
      * | expression [ AS output_name ] [, ...]  
      [ FROM from_item [, ...] ]  
      [ WHERE condition ]  
      [ GROUP BY expression [, ...] ]  
      [ HAVING condition [, ...] ]  
      [ { UNION | INTERSECT | EXCEPT } [ ALL ] select ]  
      [ ORDER BY expression [ ASC | DESC | USING operator ] [, ...] ]  
      [ LIMIT { count | ALL } ]  
      [ OFFSET start ]  
      [ FOR { UPDATE | SHARE } [ OF table_name [, ...] ] [ NOWAIT ] [...] ]
```

UNION, INTERSECT E EXCEPT

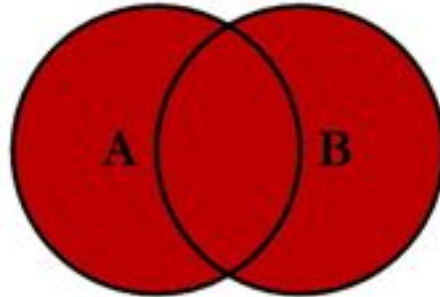
A SQL fornece operadores padrões para trabalhar com conjuntos de dados. Tais operadores permitem estender o formato de consulta básico.

Os operadores de conjunto **UNION**, **INTERSECT** e **EXCEPT** são usados para comparar e mesclar os resultados de duas expressões de consulta separadas.

As duas consultas realizadas devem retornar o mesmo número de colunas de saída com tipos de dados compatíveis. Caso contrário, os dois conjuntos de resultados não poderão ser comparados e mesclados.

UNION

A operação de União retorna as linhas de duas consulta distintas, independentemente se as linhas se derivarem de uma ou ambas as consultas realizadas.



Encontre todos os usuários que visitaram o país Brasil ou país Argentina

```
SELECT *  
FROM    Usuarios AS u  
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )  
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )  
WHERE   p.nome = 'Brasil'  
  
UNION  
  
SELECT *  
FROM    Usuarios AS u  
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )  
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )  
WHERE   p.nome = 'Argentina'
```

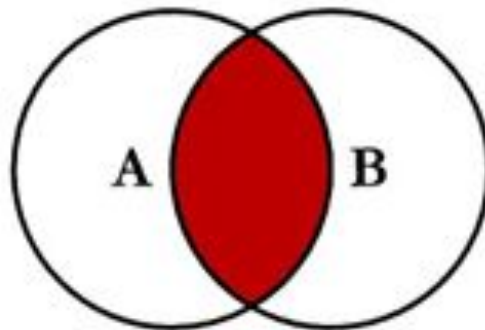
Encontre todos os usuários que visitaram o país Brasil ou país Argentina;

```
SELECT *  
FROM Usuarios AS u  
    INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )  
    INNER JOIN Países AS p ON ( p.idpais = v.idpais )  
WHERE (p.nome = 'Brasil' OR p.nome = 'Argentina')
```


INTERSECT

A Operação de intersecção retorna os registros que estão no resultados de ambas as consultas. As linhas que não forem retornadas por ambas as consultas são descartadas.

O MySQL não possui um operador **INTERSECT**. Contudo, esta operação pode ser realizada com o operador **IN**.



Selecione todos os usuários que visitaram o país Brasil e o país Argentina.

```
SELECT DISTINCT u.*  
FROM    Usuarios AS u  
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )  
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )  
WHERE   p.nome = 'Brasil' AND p.nome = 'Argentina'
```

Não retornará nenhum registro pois a consulta está logicamente incorreta!

Selecione todos os usuários que visitaram o país Brasil e o país Argentina.

```
SELECT DISTINCT u.*
FROM    Usuarios AS u
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )
WHERE   p.nome = 'Brasil' AND
        u.idUsuario IN (
            SELECT u.idUsuario
            FROM    Usuarios AS u
                    INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )
                    INNER JOIN Países AS p ON ( p.idpais = v.idpais )
            WHERE   p.nome = 'Argentina')
```

EXEMPLO INTERSECT - ALTERNATIVA

Selecione todos os usuários que visitaram o país Brasil e o país Argentina.

```
SELECT DISTINCT u.*  
FROM    Usuarios AS u  
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )  
        INNER JOIN Visitou AS v2 ON ( u.idusuario = v2.idusuario )  
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )  
        INNER JOIN Países AS p2 ON ( p2.idpais = v2.idpais )  
WHERE   p.nome = 'Brasil' AND p2.nome = 'Argentina'
```

EXEMPLO INTERSECT (NÃO SUPORTADO PELO MYSQL)

Selecione todos os usuários que visitaram o país Brasil e o país Argentina.

```
SELECT DISTINCT u.*
FROM    Usuarios AS u
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )
WHERE   p.nome = 'Brasil'

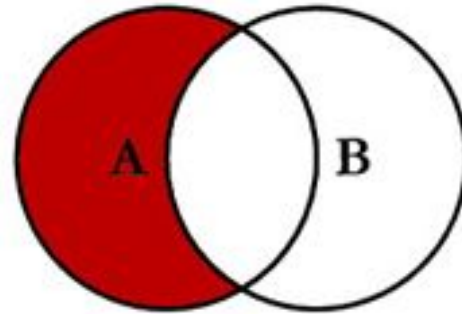
INTERSECT

SELECT u.idUsuario
FROM    Usuarios AS u
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )
WHERE   p.nome = 'Argentina')
```

EXCEPT / MINUS

A operação de exceção (**menos**) serve para retornar somente os registros que existam no resultado da primeira consulta, mas não existam no resultado da segunda consulta. Ou seja, é executada a operação de subtração de um conjunto pelo outro.

O MySQL não possui um operador **EXCEPT**. Contudo, esta operação pode ser realizada com o operador **NOT IN**





EXEMPLO EXCEPT

Selecione todos os usuários que visitaram o país Argentina mas que não visitaram o país Chile.

```
SELECT DISTINCT u.*
FROM    Usuarios AS u
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )
WHERE   p.nome = 'Argentina' AND
        u.idUsuario NOT IN (
            SELECT u.idUsuario
            FROM    Usuarios AS u
                    INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )
                    INNER JOIN Países AS p ON ( p.idpais = v.idpais )
            WHERE   p.nome = 'Chile')
```



EXEMPLO EXCEPT

Selecione todos os usuários que visitaram o país Argentina mas que não visitaram o país Chile.

```
SELECT DISTINCT u.*
FROM    Usuarios AS u
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )
        INNER JOIN Visitou AS v2 ON ( u.idusuario = v2.idusuario )
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )
        INNER JOIN Países AS p2 ON ( p2.idpais = v2.idpais )
WHERE   p.nome = 'Argentina' AND p2.nome != 'Chile'
```

EXEMPLO EXCEPT (NÃO SUPORTADO PELO MYSQL)

Selecione todos os usuários que visitaram o país Argentina mas que não visitaram o país Chile.

```
SELECT *  
FROM    Usuarios AS u  
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )  
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )  
WHERE   p.nome = 'Argentina'
```

EXCEPT

```
SELECT *  
FROM    Usuarios AS u  
        INNER JOIN Visitou AS v ON ( u.idusuario = v.idusuario )  
        INNER JOIN Países AS p ON ( p.idpais = v.idpais )  
WHERE   p.nome = 'Chile'
```

ORDER BY

A cláusula **ORDER BY** é utilizada para classificar o conjunto de resultados antes de ele ser gerado para o usuário.

A classificação pode ser realizada de forma ascendente, utilizando a palavra-chave **ASC**, que é a forma padrão, ou de forma descendente, utilizando a palavra-chave **DESC**.

```
SELECT *  
FROM Países AS p  
ORDER BY p.nome ASC
```

```
SELECT *  
FROM Países AS p  
ORDER BY p.nome DESC
```

GROUP BY

A cláusula GROUP BY é utilizada para agrupar um conjunto de resultados antes de ele ser gerado para o usuário.

O agrupamento das linhas é realizado com base nas expressões de agrupamento utilizadas na instrução select (sum, avg ou count) e retorna uma única linha para o grupo definido.

Retorne o nome do país e a média de avaliação das visitas

```
SELECT p.nome, AVG(avaliacao_visita) as media  
FROM Visitou AS v INNER JOIN Países AS p ON ( p.idpais = v.idpais )  
GROUP BY p.nome
```

MYSQL FUNCTIONS

As principais funções de agregação do MySQL

- **AVG** - Retorna a média de valores.
- **COUNT** - Retorna o número de linhas retornadas por uma consulta.
- **MIN** - Retorna o valor mínimo da expressão.
- **MAX** - Retorna o valor máximo da expressão.
- **SUM** - Retorna a soma de valores

Documentação Oficial

- <https://dev.mysql.com/doc/refman/8.0/en/functions.html>

W3Schools

- https://www.w3schools.com/sql/sql_ref_mysql.asp

HAVING

A cláusula HAVING determina uma condição de busca para um grupo ou um conjunto de registros, definindo critérios para limitar os resultados obtidos a partir do agrupamento de registros. É importante lembrar que essa cláusula só pode ser usada em parceria com GROUP BY.

Selecione o nome dos países que média de nota de avaliação maior ou igual a 6.

```
SELECT p.nome  
FROM Visitou AS v INNER JOIN Países AS p ON (p.idpais = v.idpais)  
WHERE AVG(v.avaliacao_visita) >= 6
```

Selecione o nome dos países que média de nota de avaliação maior ou igual a 6.

```
SELECT p.nome, AVG(avaliacao_visita) as média  
FROM Visitou AS v INNER JOIN Países AS p ON (p.idpais = v.idpais)  
GROUP BY p.nome  
HAVING AVG(v.avaliacao_visita) >= 6
```

A cláusula GROUP BY pode ser empregada, entre outras finalidades, para agrupar os registros de acordo com o nome de país visitado. Dentro de cada um dos grupos (formado pelos nomes dos países visitados) a cláusula HAVING pode ser usada para restringir apenas os registros que possuem uma média de avaliação de visita maior ou igual que 6, por exemplo.



O HAVING é diferente do WHERE. O WHERE restringe os resultados obtidos sempre após o uso da cláusula FROM, ao passo que a cláusula HAVING filtra o retorno do agrupamento.

LIMIT / OFFSET



LIMIT e **OFFSET** permite o usuário determinar somente um subconjunto de registros que gerados pela consulta.

- **LIMIT** – Limita o número de registros retornados ao final da consulta.
- **OFFSET** - Pula X primeiras linhas da consulta e retorna as próximas, respeitando a quantidade máxima definida no LIMIT, caso exista.

```
SELECT *  
FROM Países AS p  
ORDER BY p.nome ASC  
LIMIT 1  
OFFSET 3
```

SUB-QUERY



TRABALHANDO COM SUB-QUERY

Selecione o(s) nome(s) dos país que possuem a maior nota de avaliação já registrada.

```
SELECT p.nome  
FROM Visitou AS v INNER JOIN Países AS p ON ( p.idpais = v.idpais)  
WHERE v.avaliacao_visita = MAX(avaliacao_visita)
```

Selecione o(s) nome(s) do(s) país(es) que possuem a maior nota de avaliação já registrada.

```
SELECT *  
FROM Visitou AS v INNER JOIN Países AS p ON ( p.idpais = v.idpais)  
WHERE v.avaliacao_visita = (  
    SELECT MAX(avaliacao_visita)  
    FROM Visitou  
)
```