



SQL

DEVinHouse

Parcerias para desenvolver a sua carreira

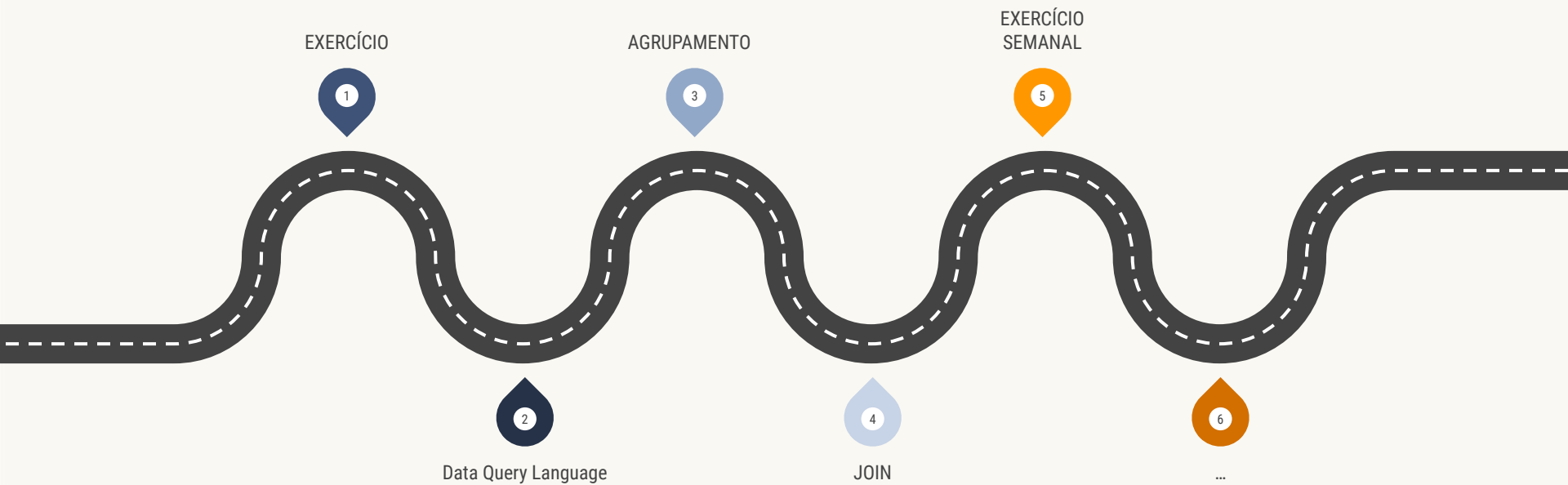
SENAI

<LAB365>

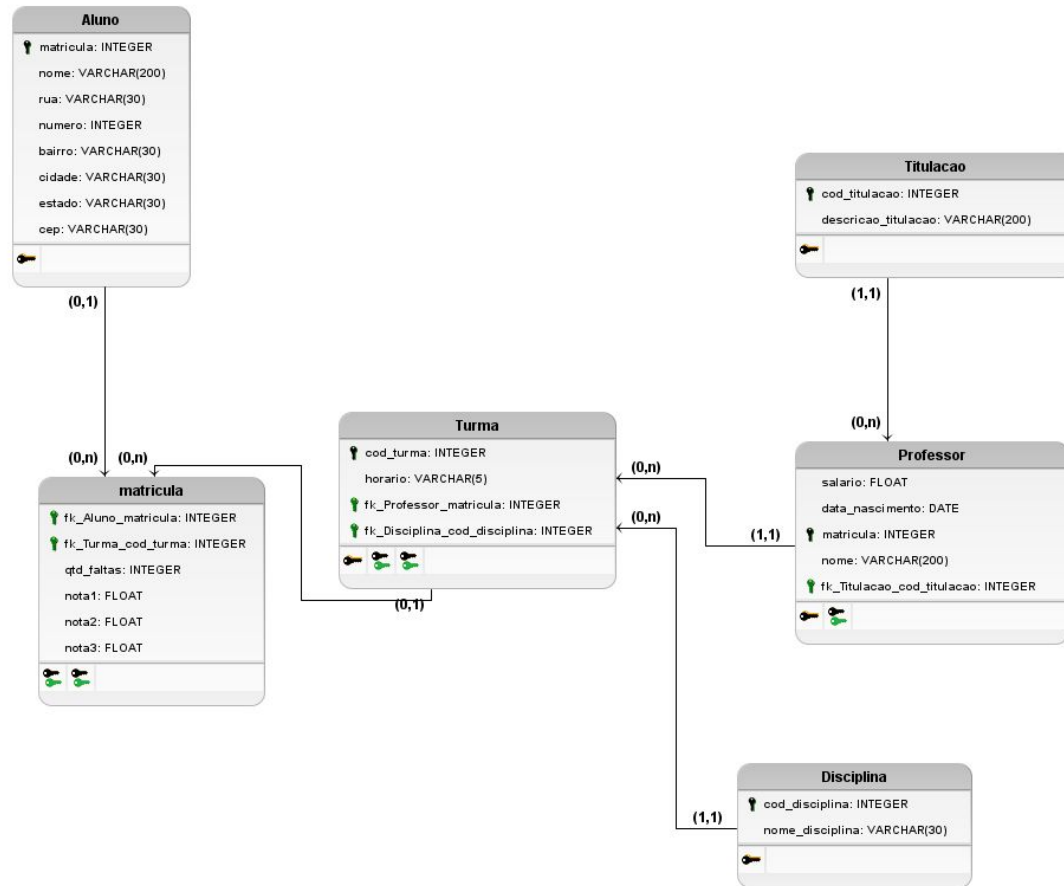
OBSERVAÇÕES

- A interação é crucial para um melhor entendimento;
- Qualquer dúvida, favor levantar a mão ou enviar no chat;
- Caso queira, pode enviar sua dúvida via Slack;
- Utilize os materiais complementares;

AGENDA



EXERCÍCIO 10



EXERCÍCIO 10

- A partir do modelo lógico apresentado anteriormente, realize:
 - Criação de schema;
 - Criação das tabelas contendo os tipos de dados informados no modelo.
 - Insira dados nas tabelas;
 - Atualize pelo menos três linhas de dados em cada uma das tabelas;
 - Remova pelo menos um dado em cada uma das tabelas;
 - Obs:.. Nessa atividade é necessário a definição das chaves estrangeiras entre as tabelas

DATA QUERY LANGUAGE



DEVinHouse

Parcerias para desenvolver a sua carreira

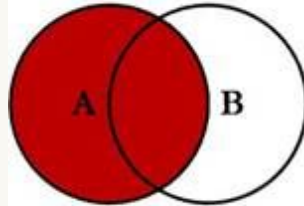
SENAI

<LAB365>

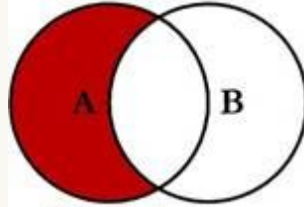
FUNÇÕES DE AGRUPAMENTO

- A cláusula agrupada pode ser combinada com a cláusula WHERE, sendo que os dados só são agrupados depois de filtrados;
- Quando o critério de agrupamento possui mais de um item de grupo, a classificação dos dados nos vários grupos levará em consideração a combinação dos valores das colunas que compõem a cláusula GROUP BY;
- A cláusula HAVING só pode ser definida combinada com a cláusula GROUP BY
- Somente itens de grupo e funções de agregação podem ser utilizadas nas expressões condicionais do HAVING;
- As funções de agregação utilizadas na cláusula HAVING não precisam ser as mesmas utilizadas na projeção;

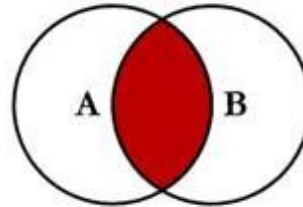
SQL JOINS



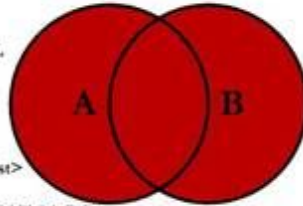
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



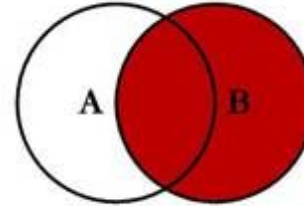
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



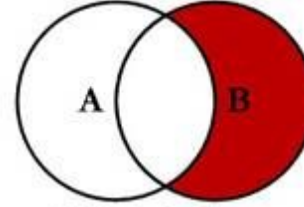
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



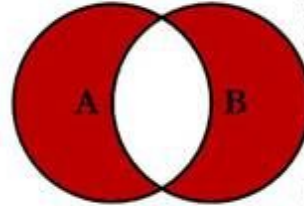
```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

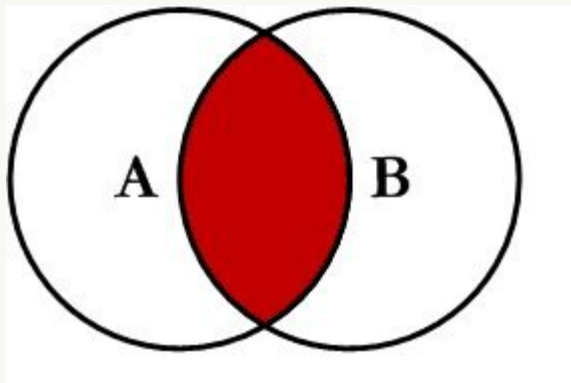


```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```


INNER JOIN

- Retorna registros comuns às duas tabelas.

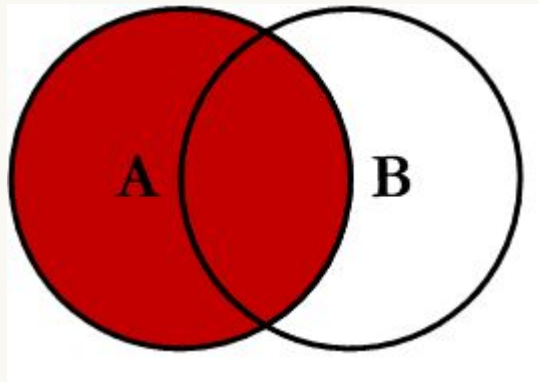
```
SELECT * FROM cliente as A  
INNER JOIN endereco as B ON cliente.codigo = endereco.codigo
```



LEFT JOIN

- Retorna registros que estão na tabela A (cliente) e os registros comuns de B (endereço) e A.

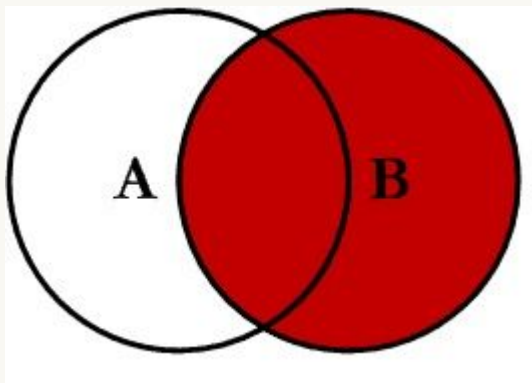
```
SELECT * FROM cliente as A  
LEFT JOIN endereco as B ON cliente.codigo = endereco.codigo
```



RIGHT JOIN

- Retorna registros que estão na tabela B (endereço) e os registros comuns de A (cliente) e B.

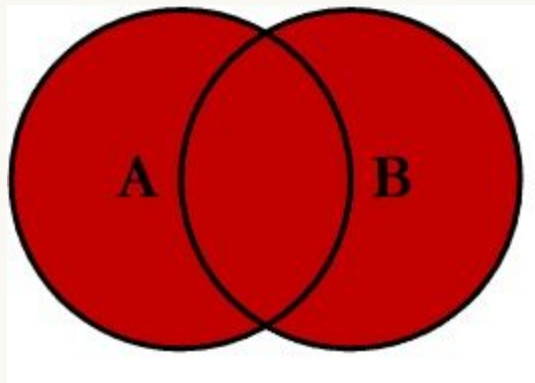
```
SELECT * FROM cliente as A  
RIGHT JOIN endereco as B ON cliente.codigo = endereco.codigo
```



OUTER JOIN

- Retorna todos registros que estão na tabela A (cliente) e todos os registros na tabela B (endereço).

```
SELECT * FROM cliente as A  
FULL OUTER JOIN endereco as B ON cliente.codigo = endereco.codigo
```



SUBSELECT

- É um recurso oferecido pelo SQL que permite a um comando SELECT obter e utilizar dinamicamente os dados de um outro para composição do resultado final;
- Consulta Principal:
 - Quais são os alunos que possuem a MGP maior que a do Aluno X (matrícula 123)?
 - Subconsulta
 - Qual a média do aluno X?



DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>