Faculdade IMPACTA

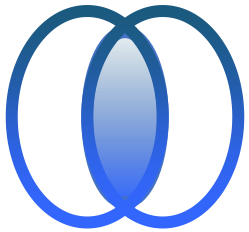
FERREIRA

Faculdade IMPACTA

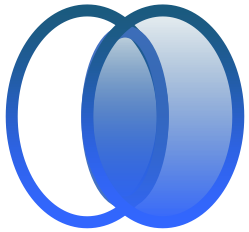
LINGUAGEM SQL

DML - JOIN

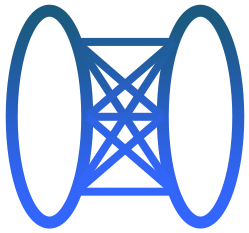
Linguagem SQL – DML - JOIN



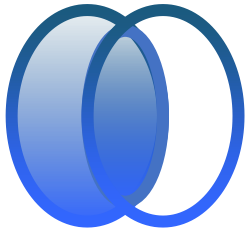
INNER JOIN



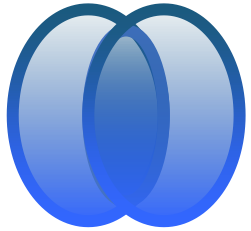
RIGHT JOIN



CROSS JOIN



LEFT JOIN



FULL JOIN

AGENDA


Faculdade
IMPACTA
antes de começar...

SELECT

```
SELECT <lista de seleção>          -- define quais colunas serão retornadas
FROM <tabela de origem>            -- define a(s) tabela(s) envolvida(s) na consulta
[INNER JOIN | LEFT JOIN | RIGHT JOIN | FULL JOIN | ICROSS JOIN <tabela2>
        ON <expressaoJoin>    --  condição de junção das tabelas
[...]]
[WHERE] <condição de pesquisa>   -- filtra as linhas requeridas
[ORDER BY] <ordem da lista>      -- ordena o retorno da lista
```

INNER JOIN

Faculdade IMPACTA

só a intersecção...

INNER JOIN

```
SELECT
        <lista_de_colunas>
FROM  <tabela1 >
[INNER JOIN <tabela2>
        ON <expressaoJoin>
[...]]
```

```
USE DB_VENDAS

SELECT CODFUN, NOME, COD_DEPTO, COD_CARGO FROM TB_EMPREGADO

SELECT * FROM TB_DEPARTAMENTO

-- CONSULTANDO 2 TABELAS NO MESMO SELECT
SELECT
  TB_EMPREGADO.CODFUN,
  TB_EMPREGADO.NOME,
  TB_DEPARTAMENTO.DEPTO
FROM TB_EMPREGADO
 INNER JOIN TB_DEPARTAMENTO
ON TB_EMPREGADO.COD_DEPTO = TB_DEPARTAMENTO.COD_DEPTO
```

INNER JOIN

```
SELECT
        <lista_de_colunas>
FROM  <tabela1 >
[INNER JOIN <tabela2>
        ON <expressaoJoin>
[...]]
```

```
-- EMPREGADOS E TABELACAR (CARGOS)
SELECT E.CODFUN, E.NOME, C.CARGO
FROM TB_EMPREGADO E
 INNER JOIN TB_CARGO C
ON E.COD_CARGO = C.COD_CARGO

-- CONSUTANDO 3 TABELAS
SELECT
  E.CODFUN, E.NOME, E.COD_DEPTO, E.COD_CARGO,
  D.DEPTO, C.CARGO
FROM TB_EMPREGADO E
  JOIN TB_DEPARTAMENTO D ON E.COD_DEPTO = D.COD_DEPTO
  JOIN TB_CARGO C ON E.COD_CARGO = C.COD_CARGO
```

INNER JOIN

```
SELECT
        <lista_de_colunas>
FROM  <tabela1 >
[INNER JOIN <tabela2>
        ON <expressaoJoin>
[...]]
```

```
/*
   JOIN COM 6 TABELAS.  FILTRANDO PEDIDOS EMITIDOS EM JANEIRO DE 2017
*/

SELECT
   I.NUM_PEDIDO, I.NUM_ITEM, I.COD_PRODUTO, PR.DESCRICAO,
   I.QUANTIDADE, I.PR_UNITARIO, T.TIPO, U.UNIDADE, CR.COR,
   PE.DATA_EMISSAO
FROM TB_ITENSPEDIDO I
   JOIN TB_PRODUTO PR ON I.ID_PRODUTO = PR.ID_PRODUTO
   JOIN TB_COR CR ON I.CODCOR = CR.CODCOR
   JOIN TB_TIPOPRODUTO T ON PR.COD_TIPO = T.COD_TIPO
   JOIN TB_UNIDADE U ON PR.COD_UNIDADE = U.COD_UNIDADE
   JOIN TB_PEDIDO PE ON I.NUM_PEDIDO = PE.NUM_PEDIDO
WHERE PE.DATA_EMISSAO BETWEEN '2017.1.1' AND '2017.1.31'
```

INNER JOIN

```
SELECT
        <lista_de_colunas>
FROM  <tabela1 >
[INNER JOIN <tabela2>
        ON <expressaoJoin>
[...]]
```

```
-- SELF-JOIN: OCORRE QUANDO UMA TABELA FAZ JOIN COM ELA PRÓPRIA
SELECT
        E.CODFUN,
        E.NOME AS FUNCIONARIO,
        S.CODFUN AS COD_SUPERVISOR,
        S.NOME AS SUPERVISOR
FROM TB_EMPREGADO E JOIN TB_EMPREGADO S ON E.COD_SUPERVISOR = S.CODFUN
```


LEFT JOIN
Faculdade
IMPACTA
tudo da
ESQUERDA

LEFT JOIN

```
SELECT
        <lista_de_colunas>
FROM  <tabela1 >
[LEFT JOIN <tabela2>
        ON <expressaoJoin>
[...]]
```

```
-- INNER JOIN = JOIN
SELECT * FROM TB_EMPREGADO -- RETORNA 61 LINHAS

SELECT E.CODFUN, E.NOME, E.COD_DEPTO, E.COD_CARGO, C.CARGO -- RETORNA 58 LINHAS
FROM TB_EMPREGADO E  INNER JOIN TB_CARGO C ON E.COD_CARGO = C.COD_CARGO

-- OUTER JOIN
SELECT  E.CODFUN, E.NOME, E.COD_DEPTO, E.COD_CARGO, C.CARGO -- RETORNA 61 LINHAS
FROM TB_EMPREGADO E
  LEFT OUTER JOIN TB_CARGO C ON E.COD_CARGO = C.COD_CARGO --LEFT JOIN

-- FILTRANDO SOMENTE OS REGISTROS NÃO CORRESPONDENTES
SELECT E.CODFUN, E.NOME, E.COD_DEPTO, E.COD_CARGO, C.CARGO -- RETORNA 3 LINHAS
FROM TB_EMPREGADO E
  LEFT JOIN TB_CARGO C ON E.COD_CARGO = C.COD_CARGO
WHERE C.COD_CARGO IS NULL
```

RIGHT JOIN

Faculdade IMPACTA

tudo da DIREITA

RIGHT JOIN

```
SELECT
        <lista_de_colunas>
FROM  <tabela1 >
[RIGHT JOIN <tabela2>
        ON <expressaoJoin>
[...]]
```

```
-- TODOS DEPARTAMENTOS COM OU SEM EMPREGADOS
SELECT
  E.CODFUN, E.NOME, E.COD_DEPTO, E.COD_CARGO, D.DEPTO
FROM TB_EMPREGADO E RIGHT JOIN TB_DEPARTAMENTO D ON E.COD_DEPTO = D.COD_DEPTO

-- FILTRANDO SOMENTE OS REGISTROS NÃO CORRESPONDENTES
SELECT
  E.CODFUN, E.NOME, E.COD_DEPTO, E.COD_CARGO, D.DEPTO
FROM TB_EMPREGADO E RIGHT JOIN TB_DEPARTAMENTO D ON E.COD_DEPTO = D.COD_DEPTO
WHERE E.COD_DEPTO IS NULL
```


FULL JOIN
Faculdade IMPACTA
TODOS
REGISTROS

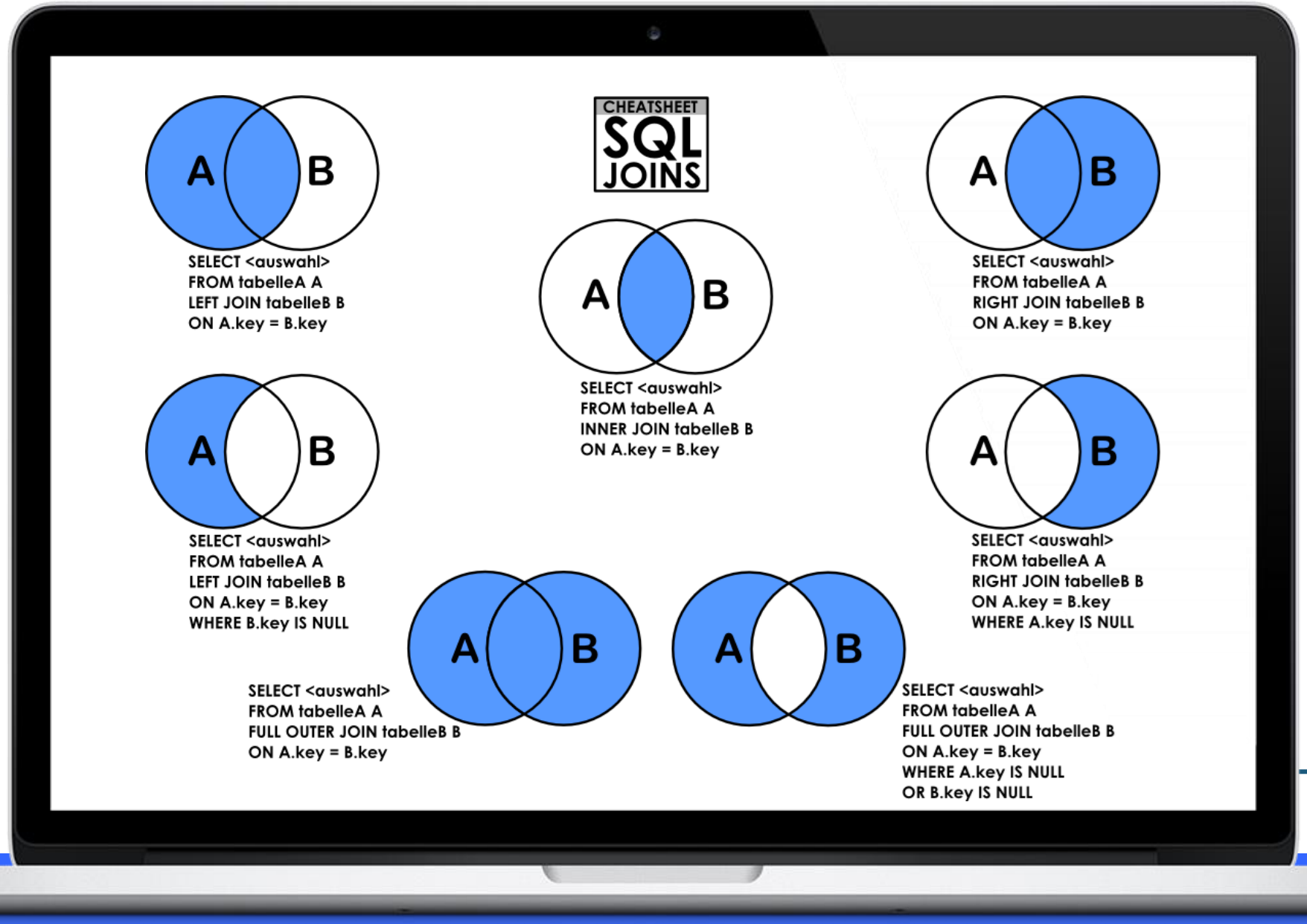
FULL JOIN

```
SELECT
        <lista_de_colunas>
FROM  <tabela1 >
[FULL JOIN <tabela2>
        ON <expressaoJoin>
[...]]
```

```
-- FULL JOIN (LEFT + RIGHT JOIN)
SELECT
  E.CODFUN, E.NOME, E.COD_DEPTO, E.COD_CARGO, C.CARGO
FROM TB_EMPREGADO E FULL JOIN TB_CARGO C ON E.COD_CARGO = C.COD_CARGO

-- FILTRANDO SOMENTE OS REGISTROS NÃO CORRESPONDENTES
SELECT
  E.CODFUN, E.NOME, E.COD_DEPTO, E.COD_CARGO, C.CARGO
FROM TB_EMPREGADO E FULL JOIN TB_CARGO C ON E.COD_CARGO = C.COD_CARGO
WHERE E.COD_CARGO IS NULL OR C.COD_CARGO IS NULL
```


Faculdade
IMPACTA
RESUMINHO !!!


CHEATSHEET
SQL JOINS
A
B
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
SELECT <auswahl>
FROM tabelleA A
INNER JOIN tabelleB B
ON A.key = B.key
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
WHERE B.key IS NULL
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL

CROSS JOIN

Faculdade IMPACTA

produto cartesiano
(lembra no Ensino Médio?)

CROSS JOIN

```
SELECT
        <lista_de_colunas>
FROM  <tabela1 >
[CROSS JOIN <tabela2>
[...]]
```

```
-- CROSS JOIN
SELECT  -- retorna 854 linhas
  E.CODFUN, E.NOME, E.COD_DEPTO, E.COD_CARGO, D.DEPTO
FROM EMPREGADOS E CROSS JOIN TABELADEP D
```

FERREIRA

Faculdade IMPACTA

BOA NOITE !
MUITO OBRIGADO

gustavo.ferreira@faculdadeimpacta.com.br www.linkedin.com/in/gustavomferreira