Faculdade
IMPACTA

FERREIRA

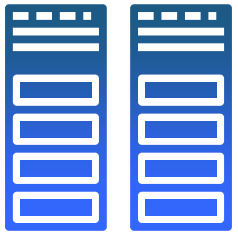
FERREIRA

Faculdade IMPACTA

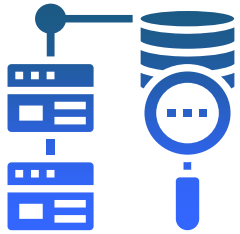
LINGUAGEM SQL

Funções de Agregação

Linguagem SQL – AGREGAÇÃO E EXTRAS



AGREGAÇÃO



SUBQUERY

AGENDA

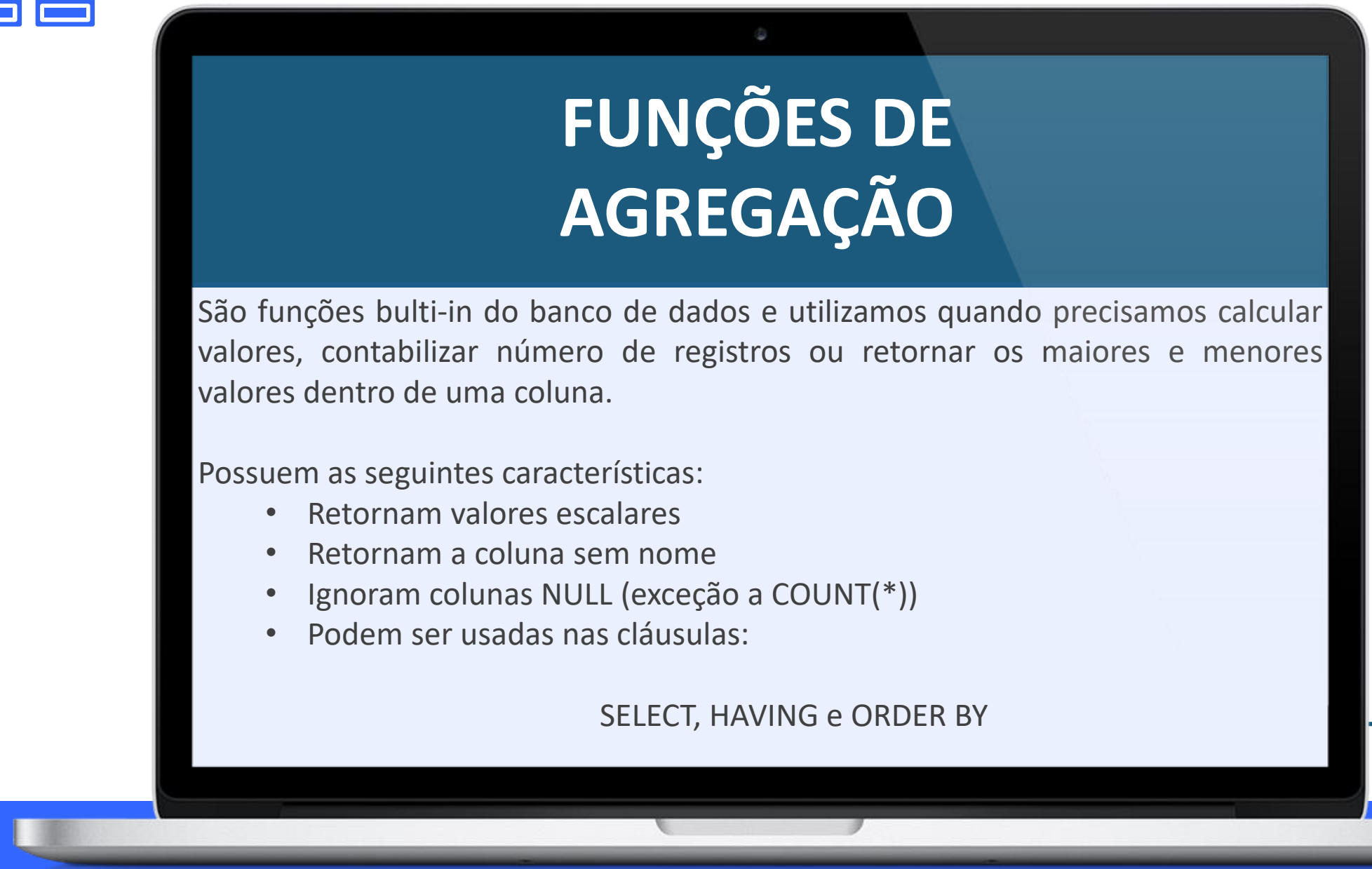
VIEW

AGREGAÇÃO

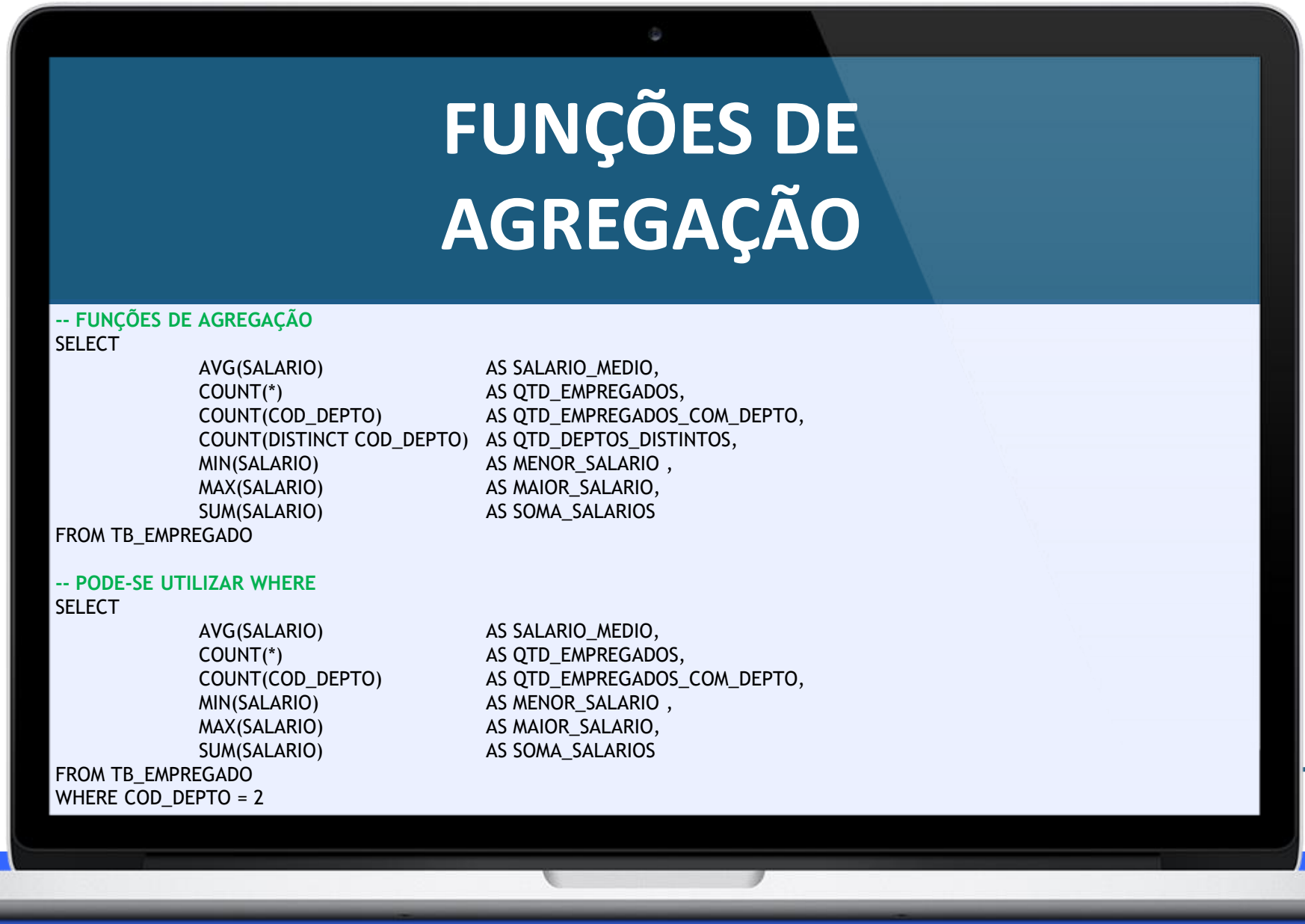
Faculdade IMPACTA

Transformando dado em informação...

FUNÇÕES DE AGREGAÇÃO



FUNÇÕES DE AGREGAÇÃO



FUNÇÕES DE AGREGAÇÃO

FUNÇÕES DE AGREGAÇÃO

GROUP BY

Agrupa o retorno das linhas de acordo com a combinação da(s) **coluna(s)** escritas na cláusula **GROUP BY.** Retira os "detalhes" das linhas, fazendo um cálculo com a função de agregação selecionada para a coluna escolhida.

```
-- TOTAL DE SALÁRIO DE CADA DEPARTAMENTO
SELECT COD_DEPTO, SUM( SALARIO ) AS TOT_SAL
FROM TB_EMPREGADO
GROUP BY COD_DEPTO
ORDER BY TOT_SAL

-- GROUP BY + JOIN
SELECT V.NOME AS VENDEDOR, C.NOME AS CLIENTE, COUNT(*) AS QTD_PEDIDOS
FROM TB_PEDIDO PE
JOIN TB_CLIENTE C ON PE.CODCLI = C.CODCLI
JOIN TB_VENDEDOR V ON PE.CODVEN = V.CODVEN
WHERE PE.DATA_EMISSAO BETWEEN '2006.1.1' AND '2006.6.30'
GROUP BY V.NOME , C.NOME;
```

FUNÇÕES DE AGREGAÇÃO

HAVING

- **Filtra** os dados obtidos através do **GROUP BY**.
- Fornece condição de pesquisa que precisa ser satisfeita para cada grupo.
- É **processado após** a execução do **GROUP BY**.

```
-- WHERE (FILTRO) SALÁRIO POR DEPARTAMENTO
-- SOMENTE EMPREGADOS QUE GANHAM MAIS DE 5000

SELECT
            E.COD_DEPTO,
            D.DEPTO,
            SUM( E.SALARIO ) AS TOT_SAL
FROM TB_EMPREGADO E
JOIN TB_DEPARTAMENTO D
ON E.COD_DEPTO = D.COD_DEPTO
WHERE E.SALARIO > 5000
GROUP BY E.COD_DEPTO, D.DEPTO
ORDER BY TOT_SAL
```

```
--HAVING (FILTRO PARA GROUP BY)
-- DEPARTAMENTO QUE GASTAM MAIS DE 5000

SELECT
            E.COD_DEPTO,
            D.DEPTO,
            SUM( E.SALARIO ) AS TOT_SAL
FROM TB_EMPREGADO E
JOIN TB_DEPARTAMENTO D
ON E.COD_DEPTO = D.COD_DEPTO
GROUP BY E.COD_DEPTO, D.DEPTO
HAVING SUM(E.SALARIO) > 5000
ORDER BY TOT_SAL
```

FUNÇÕES DE AGREGAÇÃO

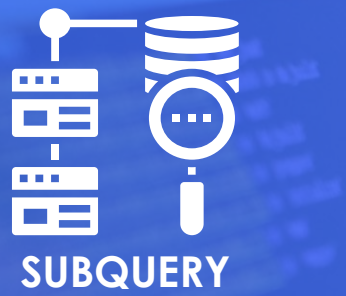
WHERE x HAVING

WHERE

- Filtra linhas **ANTES** dos grupos serem criados.
- Controla quais **LINHAS** serão passadas para o GROUP BY.

HAVING

- Filtra grupos **APÓS** eles serem criados.
- Controla quais **GRUPOS** serão retornados.



Faculdade IMPACTA

Consulta dentro de consulta

SUBQUERY

Instrução SELECT que está condicionada dentro de outra instrução SQL. Como resultado desta operação, podemos fazer uso de subconsultas para criarmos consultas que seriam difíceis ou impossíveis de serem feitas utilizando outras maneiras. Desde que saibamos codificar nossas instruções SELECT, saberemos como codificar uma subconsulta, já que ela é apenas uma instrução SQL no interior de outra instrução SQL.

```
-- Pedidos da vendedora chamada LEIA vendidos para clientes de SP
-- que não compraram em Janeiro de 2007, mas compraram em Dezembro de 2006
SELECT * FROM PEDIDOS
WHERE
        CODVEN IN ( SELECT CODVEN FROM VENDEDORES  WHERE NOME = 'LEIA' )
        AND
        CODCLI IN( SELECT CODCLI FROM CLIENTES
                WHERE ESTADO = 'SP'
                AND CODCLI NOT IN (SELECT CODCLI FROM PEDIDOS
                        WHERE DATA_EMISSAO BETWEEN '20070101' AND '20070131')
                AND CODCLI IN (SELECT CODCLI FROM PEDIDOS
                        WHERE DATA_EMISSAO BETWEEN '20061201' AND '20061231')
                )
```

Faculdade IMPACTA

VISUALizando

VIEWS

Objeto virtual composto por linhas e colunas provenientes de tabelas referenciadas em uma query. Este objeto oferece uma **VISUALIZAÇÃO LÓGICA** dos dados, podendo ser compartilhado. Os dados são gerados de forma dinâmica no momento em que a view é referenciada (utilizada).

CREATE VIEW <NOME DA VIEW>
AS SELECT ...

```
CREATE VIEW VW_ITENSPEDIDO AS
SELECT
        I.NUM_PEDIDO, I.NUM_ITEM, I.ID_PRODUTO,I.QUANTIDADE, I.PR_UNITARIO, I.DESCONTO,
        I.QUANTIDADE * I.PR_UNITARIO *( 1 - I.DESCONTO / 100 ) AS VALOR,
        P.DESCRICAO, T.TIPO, U.UNIDADE, C.COR
FROM TB_ITENSPEDIDO I
        JOIN TB_PRODUTO P               ON I.ID_PRODUTO = P.ID_PRODUTO
        JOIN TB_TIPOPRODUTO T           ON P.COD_TIPO = T.COD_TIPO
        JOIN TB_UNIDADE U               ON P.COD_UNIDADE = U.COD_UNIDADE
        JOIN TB_COR C                   ON I.CODCOR = C.CODCOR
        JOIN TB_PEDIDO PE               ON I.NUM_PEDIDO=PE.NUM_PEDIDO;

SELECT * FROM VW_ITENSPEDIDO;

SELECT NUM_PEDIDO, QUANTIDADE, VALOR, DESCRICAO FROM VW_ITENSPEDIDO;
```

FERREIRA

Faculdade IMPACTA

BOA NOITE !
MUITO OBRIGADO

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