

# Bacharelado em Sistemas de Informação

# BANCOS DE DADOS

## Aula 18

## SQL Junção de tabelas Funções de Grupo



## ✓ **JUNÇÃO DE TABELAS**

- Usado quando uma *query* necessita de dados de mais de uma tabela.
- As linhas de uma tabela podem ser juntadas às linhas de outra tabela.
- *junção de tabelas* é especificada na cláusula **WHERE**.
- Sua forma mais simples é quando os valores em uma tabela são iguais aos desejados da outra tabela. Corresponde geralmente à igualdade entre chaves do Modelo Relacional.

## ✓ **Junção de tabelas**

- Se a condição da junção é omitida ou inválida, o resultado é um produto, que gera um grande número de linhas, normalmente inúteis. Corresponde ao resultado do Produto Cartesiano da Álgebra Relacional.

# ✓ Junção de tabelas

- Exemplos: Tabelas EMP e DEPT

```
SQL> DESCRIBE DEPT;
```

Name	Null?	Type
-----	-----	----
DEPTNO	NOT NULL	NUMBER (2)
DNAME		VARCHAR2 (14)
LOC		VARCHAR2 (13)

```
SQL> SELECT * FROM DEPT;
```

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



# ✓ Junção de tabelas

- Exemplos: Tabelas EMP e DEPT

```
SELECT ENAME, JOB, DNAME  
FROM EMP, DEPT  
WHERE EMP.DEPTNO = DEPT.DEPTNO;
```

ENAME	JOB	DNAME
SMITH	CLERK	RESEARCH
ALLEN	SALESMAN	SALES
WARD	SALESMAN	SALES
JONES	MANAGER	RESEARCH
MARTIN	SALESMAN	SALES
BLAKE	MANAGER	SALES
CLARK	MANAGER	ACCOUNTING
SCOTT	ANALYST	RESEARCH
KING	PRESIDENT	ACCOUNTING
TURNER	SALESMAN	SALES
ADAMS	CLERK	RESEARCH
JAMES	CLERK	SALES
FORD	ANALYST	RESEARCH
MILLER	CLERK	ACCOUNTING



# ✓ Junção de tabelas

## • Exemplos: Tabelas EMP e DEPT

```
SELECT DEPT.DEPTNO, ENAME, JOB, DNAME
FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO
ORDER BY DEPT.DEPTNO;
```

DEPTNO	ENAME	JOB	DNAME
10	CLARK	MANAGER	ACCOUNTING
10	KING	PRESIDENT	ACCOUNTING
10	MILLER	CLERK	ACCOUNTING
20	SMITH	CLERK	RESEARCH
20	ADAMS	CLERK	RESEARCH
20	FORD	ANALYST	RESEARCH
20	SCOTT	ANALYST	RESEARCH
20	JONES	MANAGER	RESEARCH
30	ALLEN	SALESMAN	SALES
30	BLAKE	MANAGER	SALES
30	MARTIN	SALESMAN	SALES
30	JAMES	CLERK	SALES
30	TURNER	SALESMAN	SALES
30	WARD	SALESMAN	SALES



## ✓ **FUNÇÕES DE GRUPO**

- Operam em um conjunto de linhas.
- Retornam resultados baseados em grupos de linhas, ao invés de um resultado por linha.



# ✓ FUNÇÕES DE GRUPO

## Função

## Valor Retornado

**AVG (DISTINCT | ALL | n)**

**Média de  $n$ , ignorando *nulls***

**COUNT (DISTINCT | ALL |  $expr$ )**

**Número de vezes que  $expr$  não retorna nulo**

**MAX (DISTINCT | ALL |  $expr$ )**

**Máximo valor de  $expr$**

**MIN (DISTINCT | ALL |  $expr$ )**

**Mínimo valor de  $expr$**

**STDDEV (DISTINCT | ALL |  $n$ )**

**Desvio padrão de  $n$ , ignorando *nulls*.**

**SUM (DISTINCT | ALL |  $n$ )**

**Soma valores de  $n$ , ignorando *nulls*.**

**VARIANCE (DISTINCT | ALL |  $n$ )**

**Variância de  $n$ , ignorando *nulls*.**





# ✓ FUNÇÕES DE GRUPO

## ► Observações:

- Cláusula **DISTINCT**: faz a função considerar somente valores distintos.
- Cláusula **ALL**: opção *default*, inclui os valores duplicados.
- Tipos de dados de *expr* podem ser **CHAR**, **NUMBER** ou **DATE**
- Todas as funções, exceto **COUNT(\*)** ignoram valores nulos.



# ✓ FUNÇÕES DE GRUPO

- Exemplos:

```
SQL> SELECT AVG (SAL) FROM EMP;
```

```
AVG (SAL)
```

```
-----
```

```
2073,2143
```

```
SQL> SELECT MIN (SAL) FROM EMP;
```

```
MIN (SAL)
```

```
-----
```

```
800
```

# ✓ FUNÇÕES DE GRUPO

- Exemplos:

```
SQL>  SELECT COUNT (*)  
      2  FROM EMP  
      3  WHERE DEPTNO = 20;
```

COUNT (\*)

-----

5

# ✓ FUNÇÕES DE GRUPO

## ► Cláusula GROUP BY

- Pode ser usada para dividir um conjunto de linhas em grupos menores, enquanto que as funções de grupo podem ser usadas para retornar informação sobre cada grupo.

# ✓ FUNÇÕES DE GRUPO

## ► Cláusula GROUP BY

- Exemplo:

```
SQL> SELECT JOB, AVG (SAL)
      2 FROM EMP
      3 GROUP BY JOB;
```

JOB	AVG (SAL)
-----	-----
ANALYST	3000
CLERK	1037,5
MANAGER	2758,3333
PRESIDENT	5000
SALESMAN	1400

# ✓ FUNÇÕES DE GRUPO

## ► Exclusão de linhas da seleção

- Se alguma linha não deve ser incluída na seleção, ela deve ser excluída com a cláusula **WHERE**.

- **Exemplo:**

```
SQL> SELECT JOB, AVG(SAL)
2 FROM EMP
3 WHERE JOB <> 'MANAGER'
4 GROUP BY JOB;
```

JOB	AVG(SAL)
-----	-----
ANALYST	3000
CLERK	1037,5
PRESIDENT	5000
SALESMAN	1400



# ✓ FUNÇÕES DE GRUPO

## ▶ Grupos dentro de grupos

- A cláusula GROUP BY também pode ser usada para fornecer resultados de grupos dentro dos grupos.
- Exemplo:

# ✓ FUNÇÕES DE GRUPO

## ► Grupos dentro de grupos

- Exemplo:

```
SQL> SELECT DEPTNO, JOB, AVG(SAL)
      2 FROM EMP
      3 GROUP BY DEPTNO, JOB;
```

DEPTNO	JOB	AVG (SAL)
-----	-----	-----
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	3000
20	CLERK	950
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	1400





# ✓ FUNÇÕES DE GRUPO

## ▶ Cláusula HAVING

- Especifica quais grupos devem ser mostrados.
- Observação: a cláusula WHERE é usada somente para restringir as linhas que serão selecionadas para a formação dos grupos, enquanto a cláusula HAVING é usada para *restringir* os grupos já formados.

# ✓ FUNÇÕES DE GRUPO

## ► Cláusula HAVING

- Exemplo:

```
SQL> SELECT DEPTNO, AVG (SAL)
2   FROM EMP
3   GROUP BY DEPTNO
4   HAVING COUNT (*) > 3;
```

DEPTNO	AVG (SAL)
20	2175
30	1566,6667

## ✓ SUBQUERIES

- Comando **SELECT** aninhado dentro de outro comando **SELECT**.

- **Formato geral:**

**SELECT** coluna1, coluna2,..

**FROM** tabela 1

**WHERE** colunaN = (**SELECT** colunaM

**FROM** tabela 2

**WHERE** condição)



# ✓ SUBQUERIES

- Exemplos:

```
SQL> SELECT ENAME, JOB, SAL  
      FROM EMP  
      WHERE SAL = (SELECT MIN(SAL) FROM EMP);
```

ENAME	JOB	SAL
SMITH	CLERK	800

# ✓ SUBQUERIES

- Exemplos:

```
SQL>  SELECT ENAME, JOB
        FROM    EMP
        WHERE   JOB = (SELECT JOB
                        FROM EMP
                        WHERE ENAME = 'BLAKE');
```

ENAME	JOB
JONES	MANAGER
BLAKE	MANAGER
CLARK	MANAGER

# ✓ SUBQUERIES

- Retorno de mais de uma linha
- Observação: uso do operador IN no exemplo.

- **Exemplo:**

```
SQL> SELECT ENAME, SAL, DEPTNO
      FROM EMP
      WHERE SAL IN (SELECT MIN(SAL)
                    FROM EMP
                    GROUP BY DEPTNO;
```

ENAME	SAL	DEPTNO
SMITH	800	20
JAMES	950	30
MILLER	1300	10

## ✓ SUBQUERIES

- Comparação de mais de um valor
- Observação: ordem das colunas
  - Exemplo:

```
SQL> SELECT ENAME, SAL
      FROM EMP
      WHERE (SAL, DEPTNO) IN (SELECT MIN(SAL), DEPTNO
                             FROM EMP
                             GROUP BY DEPTNO);
```

ENAME	SAL
SMITH	800
JAMES	950
MILLER	1300

# ✓ SUBQUERIES

## – Operadores ANY e ALL

- Usados em *subqueries* que retornam mais de uma linha.
- Usados nas cláusulas WHERE ou HAVING em conjunto com operadores lógicos.
- ANY compara um valor a cada valor retornado pela *subquery*.





# ✓ SUBQUERIES

## – Operadores ANY e ALL

- Exemplo:

```
SQL> SELECT ENAME, SAL, JOB, DEPTNO
      FROM EMP
      WHERE SAL > ANY (SELECT DISTINCT SAL
                        FROM EMP
                        WHERE DEPTNO=30);
```

ENAME	SAL	JOB	DEPTNO
ALLEN	1600	SALESMAN	30
WARD	1250	SALESMAN	30
JONES	2975	MANAGER	20
MARTIN	1250	SALESMAN	30
BLAKE	2850	MANAGER	30
CLARK	2450	MANAGER	10
SCOTT	3000	ANALYST	20
KING	5000	PRESIDENT	10
TURNER	1500	SALESMAN	30
ADAMS	1100	CLERK	20
FORD	3000	ANALYST	20
MILLER	1300	CLERK	10



# ✓ SUBQUERIES

## – Operadores ANY e ALL

- ALL compara um valor a todos os valores retornados pela *subquery*.
- Exemplo:

```
SQL> SELECT ENAME, SAL, JOB, DEPTNO
      FROM EMP
      WHERE SAL > ALL (SELECT DISTINCT SAL
                       FROM EMP
                       WHERE DEPTNO=30) ;
```

ENAME	SAL	JOB	DEPTNO
JONES	2975	MANAGER	20
SCOTT	3000	ANALYST	20
KING	5000	PRESIDENT	10
FORD	3000	ANALYST	20

## Exercícios

1. Encontre o menor, o maior e a média dos salários dos empregados.
2. Mostre o menor e maior salário para cada cargo.
3. Encontre a quantidade de gerentes (MANAGER) que há na tabela EMP.
4. Descubra a média salarial e a média de remuneração total (salário + comissão) para cada cargo.
5. Mostre a diferença entre o maior e o menor salário.
6. Mostre os códigos dos departamentos com mais de 3 empregados.
7. Verifique se existe algum funcionário com código duplicado.
8. Mostre quantos funcionários têm nome começando com a letra "A".
9. Mostre os menores salários subordinados a cada gerente. Exclua os grupos com os salários menores que 1000. Classifique os dados por salário.
10. Mostre o funcionário (nome, cargo e salário) que recebe o maior salário em cada cargo, classificando-os em ordem decrescente de salário.
11. Mostre os funcionários (nome, cargo e salário) que recebem o menor salário de seu cargo. Mostre os resultados em ordem crescente de salários.
12. Mostre os contratados (nome, depto e data de admissão) mais recentes de cada departamento, classificados em ordem de data de admissão.
13. Mostre os dados dos funcionários (nome, salário e departamento) que recebem salário maior que a média salarial da empresa.
14. Mostre os dados dos funcionários (nome, salário e departamento) que recebem salário menor que a média salarial do seu departamento.
15. Mostre os dados dos funcionários (nome, comissão e departamento) que recebem comissão maior que a média de comissões do seu departamento.



# Bacharelado em Sistemas de Informação

# BANCOS DE DADOS

## Aula 18

## SQL Junção de tabelas Funções de Grupo

