



## Unidade 3

# Bancos de Dados Relacionais e Linguagem SQL

---

## 3.2 Junções e Funções de Grupo

---

# Junções

- Feliz seríamos, porém nada funcionais, se os SGBDs gravassem os dados em uma única tabela.
- Assim sendo, tudo que vimos até agora seria pertinente para operarmos e fazermos tudo o que for necessário.
- Mas o modelo não é este! Precisamos então associar os dados das tabelas para produzir o resultado desejado.
- O SQL nos fornece condições de junção que permitem que as informações sejam combinadas a partir das tabelas.

# Histórico SQL - Resumo

- ANSI: é o acrônimo de *American National Standards Institute* que foi fundada em 1918, como uma organização privada sem fins lucrativos que administra e coordena o sistema voluntário de avaliação de padronização e conformidade nos USA.
- Esta padronização tem como objetivo aumentar tanto a competitividade global dos negócios nos USA assim como a qualidade de vida dos cidadãos norte-americanos promovendo e facilitando os mecanismos de avaliação de padrões e conformidade a estes padrões.

# Histórico SQL - Resumo

- SQL é a linguagem de processamento de informações padrão do setor dos SGBD-R que foi desenvolvida pela IBM, em meados da década de 70 e passou a ser usada mais amplamente no início da década de 80.
- Tornou-se um padrão do setor em 1986, quando foi adotada pelo ANSI.
- O ANSI promoveu até agora, três padronizações da linguagem SQL, que são evolução das anteriores e são nomeadas com o ano em que foram publicadas: ANSI-86, ANSI-92 e ANSI-99.

# Junções

- O padrão ANSI/ISO SQL99, nos traz opções que são comuns a todos os bancos de dados relacionais que estão neste padrão (quero dizer os maiores do mercado).
- Porém há diferenças proprietárias, podemos assim dizer onde, os fabricantes dos SGBDs, irão ofertar alguns recursos que melhoram a operação de seus comandos.
- **Iremos trabalhar apenas com o padrão ANSI/ISO neste primeiro momento, ok?**

# Junções

- Uma cláusula de junção SQL combina campos de duas(ou mais) tabelas em um banco de dados relacional e podemos conceituá-las em dois tipos:
- **Junção natural (*NATURAL JOIN*):** baseia-se em todas as colunas em duas tabelas que tenham o mesmo nome e seleciona linhas de ambas que tenham valores iguais em todas as colunas relacionadas.
- **Junção cruzada (*CROSS JOIN*):** no padrão ANSI/ISO SQL-99, junta cada linha de uma tabela a cada linha de outra tabela e o resultado representa todas as combinações de linhas possíveis das duas tabelas. **Imagine uma combinação de duas tabelas com 200 e 100 linhas!**

# Junções - Natural Join

- Utilizando-se a cláusula **NATURAL JOIN**, é possível unir as tabelas sem precisar especificar as colunas na tabela correspondente.
- No entanto, os nomes e tipos de dados de ambas as colunas devem ser os mesmos.

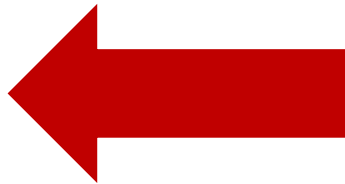


# Junções - Natural Join

```
use employees;
```

```
desc departments;
```

```
desc dept_emp;
```



Assim descobriremos qual o campo une as tabelas

```
SELECT DISTINCT(departments.dept_name), dept_emp.dept_no  
FROM departments NATURAL JOIN dept_emp  
ORDER BY dept_emp.dept_no
```

# Junções - Cross Join

- Utilizando-se a cláusula **CROSS JOIN**, todas as linhas das tabelas serão apresentadas no resultado, sem nenhum tipo de cláusula **WHERE** para aplicarmos o filtro é uma operação que pode fazer um banco de dados parar de funcionar enquanto está sendo executada.

```
use sakila;
```

```
desc address;
```

```
desc customer;
```

```
SELECT *
```

```
FROM address CROSS JOIN customer;
```

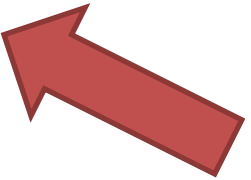
# Junções - Natural Join - USING

- Utilizando-se o **NATURAL JOIN**, se as tabelas tiverem colunas com o mesmo nome, mas tipos de dados diferentes, a junção causará um erro.
- A cláusula de junção pode ser modificada com a cláusula **USING** especificando a colunas que devem ser usadas para a junção.
- **O campo informado não deve ser vinculado a nenhuma tabela**

# Junções - Natural Join - USING

```
use employees;
```

```
SELECT employees.first_name, employees.gender  
FROM   employees JOIN dept_emp USING (emp_no)  
WHERE  dept_emp.dept_no like '%2%'
```




# Junções - Natural Join - ON



- Caso as colunas a serem unidas tiverem nomes diferentes (**isso acontece muito!**) ou se a junção usar operadores de comparação de não igualdade, como  $<$ ,  $>$  ou BETWEEN, não é possível a aplicação do **USING**.
- Utilizaremos então um outro recurso que é a cláusula **ON** que permite especificar uma variedade maior de condições de junção.

# Junções - Natural Join - ON

```
SELECT e.emp_no,e.first_name, e.last_name, e.gender, d.dept_name  
FROM   departments d JOIN dept_emp de ON (d.dept_no=de.dept_no ) JOIN employees e USING (emp_no)  
WHERE de.dept_no like '%2%'  
ORDER BY first_name;
```



```
SELECT e.emp_no,e.first_name, e.last_name, e.gender, d.dept_name  
FROM   departments d JOIN dept_emp de ON (d.dept_no=de.dept_no ) NATURAL JOIN employees e  
WHERE de.dept_no like '%2%'  
ORDER BY first_name;
```



---

*I need to see your  
Projects!  
Let's go!?*

---

# Funções de data

```
SELECT CURDATE() as "Data Corrente do SGBD" FROM DUAL;
```

```
SELECT SYSDATE() as "Data do Sistema" FROM DUAL;
```

```
USE employees;
```

```
SELECT last_name, hire_date, DATE_ADD(hire_date, INTERVAL 180 DAY) as "Daqui a 6 meses (convertidos em dias)"  
FROM employees limit 10;
```

```
SELECT last_name, hire_date, DATE_ADD(hire_date, INTERVAL 5 YEAR) as "Daqui a 5 anos"  
FROM employees limit 10;
```

```
SELECT last_name, hire_date, (DATEDIFF(CURDATE(),hire_date)/365) "Quanto tempo o empregado possui de empresa"  
FROM employees limit 10;
```

```
SELECT dayofweek(hire_date)  
FROM employees limit 10;
```

Atenção ao usar isto!



```
SELECT NOW(), SLEEP(3), NOW() FROM DUAL;
```

```
SELECT SYSDATE(), SLEEP(3), SYSDATE() FROM DUAL;
```



# Referências bibliográficas

**Neto**, Olibário, 2020. Álgebra Relacional em Bancos de Dados - Operações Unárias (Seleção, Projeção e Renomeação). Disponível em:  
<<https://www.youtube.com/watch?v=E-tIwv8jDhE>>. Acesso em: 24 out. 2021.

**Borges**, Vanessa, 2021. Linguagens relacionais - Álgebra relacional. Disponível em:  
<<https://www.youtube.com/watch?v=e39T5H8gwRY>>. Acesso em: 24 out. 2021.

**Macoratti**, José Carlos. SQL - Álgebra Relacional - Operações Fundamentais - Conceitos básicos. Disponível em:  
<[http://www.macoratti.net/13/06/sql\\_arcb.htm](http://www.macoratti.net/13/06/sql_arcb.htm)>. Acesso em: 24 out. 2021.