



SQL Básica

Herysson R. Figueiredo
herysson.figueiredo@ufn.edu.br



Sumário

- Revisão
- Instruções INSERT, DELETE e UPDATE em SQL.
- Consultas de recuperação básicas em SQL.



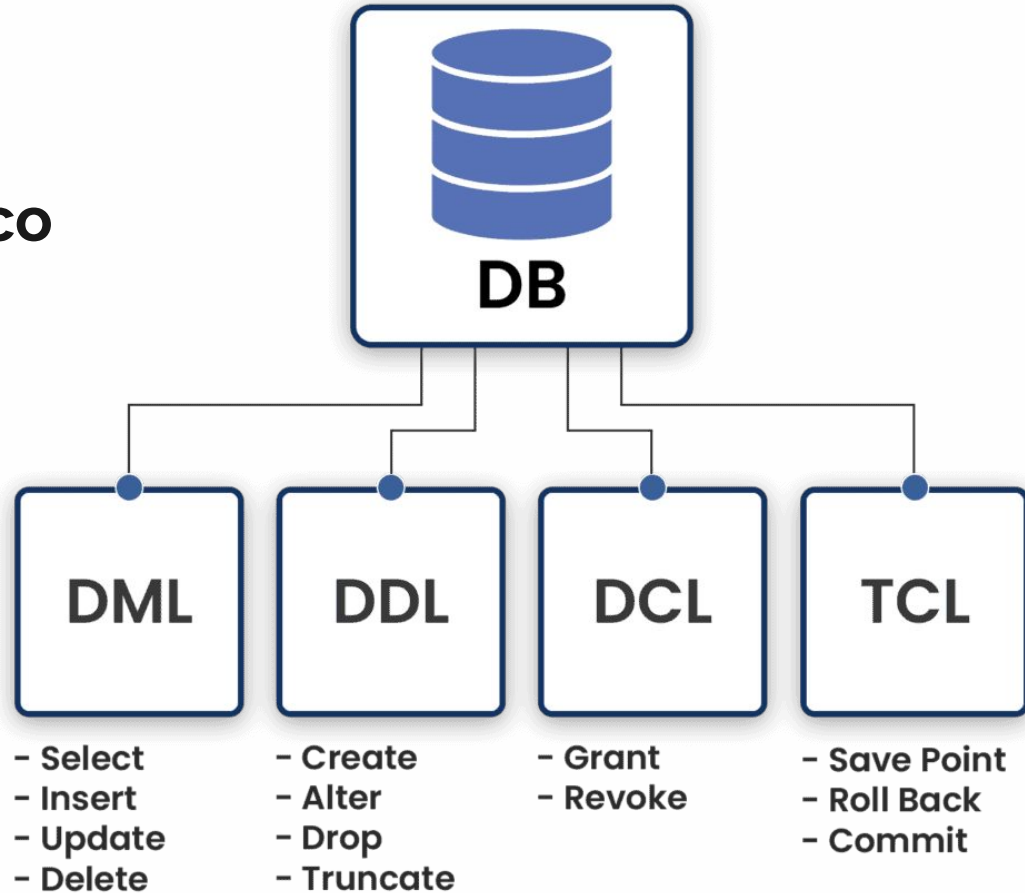
SQL Básico

SQL é uma linguagem de banco de dados abrangente: tem instruções para definição de dados, consultas e atualizações. Logo, ela é uma **DDL** - *Data definition language* (Linguagem de definição de dados) e uma **DML** - *Data Manipulation Language* (Linguagem de manipulação de dados).

DDL - CREATE - DROP - ALTER

DML - SELECT, INSERT, UPDATE E DELETE.

SQL Básico



DCL - Data Control Language | TCL - Transaction Control Language



Revisão

A SQL usa os termos **tabela**, **linha** e **coluna** para os termos do modelo relacional formal *relação*, *tupla* e *atributo*, respectivamente

Instruções DML - INSERT, DELETE e UPDATE em SQL



O comando INSERT

Em sua forma mais simples, **INSERT** é usado para acrescentar uma única tupla a uma relação. Temos de especificar o nome da relação e uma lista de valores para a tupla. Os valores devem ser listados na mesma ordem em que os atributos correspondentes foram especificados no comando **CREATE TABLE**.



O comando INSERT

```
INSERT INTO FUNCIONARIO VALUES ( 'Ricardo', 'K', 'Marini',  
'65329865388', '30-12- 1962', 'Rua Itapira, 44, Santos, SP', 'M', 37.000,  
'65329865388',4 );
```




O comando INSERT

Uma segunda forma da instrução **INSERT** permite que o usuário especifique nomes de atributo explícitos que correspondem aos valores fornecidos no comando **INSERT**



O comando INSERT

```
INSERT INTO FUNCIONARIO (Pnome, Unome, Dnr, Cpf) VALUES  
(‘Ricardo’, ‘Marini’, 4, ‘65329865388’);
```

Os atributos não especificados são definidos como seu valor DEFAULT ou NULL, e os valores são listados na mesma ordem que os atributos são listados no próprio comando INSERT.



O comando INSERT

Também é possível inserir em uma relação múltiplas tuplas separadas por vírgulas em um único comando **INSERT**. Os valores de atributo que formam cada tupla ficam entre parênteses.



O comando DELETE

O comando **DELETE** remove tuplas de uma relação. Ele inclui uma cláusula **WHERE**, semelhante a que é usada em uma consulta SQL, para selecionar as tuplas a serem excluídas. As tuplas são explicitamente excluídas de apenas uma tabela por vez.

```
DELETE FROM FUNCIONARIO WHERE Unome='Braga';
```



O comando **UPDATE**

O comando **UPDATE** é usado para modificar valores de atributo de uma ou mais tuplas selecionadas. Assim como no comando **DELETE**, uma cláusula **WHERE** no comando **UPDATE** seleciona as tuplas a serem modificadas em uma única relação.

Uma cláusula **SET** adicional no comando **UPDATE** especifica os atributos a serem modificados e seus novos valores



O comando UPDATE

UPDATE	PROJETO
SET	Projlocal = 'Santo André', Dnum = 5
WHERE	Projnumero=10;

Prática:

Insira os valores contidos nas tabelas ao lado.

FUNCIONARIO

Pnome	Minicial	Unome	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
João	B	Silva	12345678966	09-01-1965	Rua das Flores, 751, São Paulo, SP	M	30.000	33344555587	5
Fernando	T	Wong	33344555587	08-12-1955	Rua da Lapa, 34, São Paulo, SP	M	40.000	88866555576	5
Alice	J	Zelaya	99988777767	19-01-1968	Rua Souza Lima, 35, Curitiba, PR	F	25.000	98765432168	4
Jennifer	S	Souza	98765432168	20-06-1941	Av. Arthur de Lima, 54, Santo André, SP	F	43.000	88866555576	4
Ronaldo	K	Lima	66688444476	15-09-1962	Rua Rebouças, 65, Piracicaba, SP	M	38.000	33344555587	5
Joice	A	Leite	45345345376	31-07-1972	Av. Lucas Obes, 74, São Paulo, SP	F	25.000	33344555587	5
André	V	Pereira	98798798733	29-03-1969	Rua Timbira, 35, São Paulo, SP	M	25.000	98765432168	4
Jorge	E	Brito	88866555576	10-11-1937	Rua do Horto, 35, São Paulo, SP	M	55.000	NULL	1

DEPARTAMENTO

Dnome	Dnumero	Cpf_gerente	Data_inicio_gerente
Pesquisa	5	33344555587	22-05-1988
Administração	4	98765432168	01-01-1995
Matriz	1	88866555576	19-06-1981

LOCALIZACAO_DEP

Dnumero	Dlocal
1	São Paulo
4	Mauá
5	Santo André
5	Itu
5	São Paulo

TRABALHA_EM

Fcpf	Pnr	Horas
12345678966	1	32,5
12345678966	2	7,5
66688444476	3	40,0
45345345376	1	20,0
45345345376	2	20,0
33344555587	2	10,0
33344555587	3	10,0
33344555587	10	10,0
33344555587	20	10,0
99988777767	30	30,0
99988777767	10	10,0
98798798733	10	35,0
98798798733	30	5,0
98765432168	30	20,0
98765432168	20	15,0
88866555576	20	NULL

PROJETO

Projnome	Projnumero	Projlocal	Dnum
ProdutoX	1	Santo André	5
ProdutoY	2	Itu	5
ProdutoZ	3	São Paulo	5
Informatização	10	Mauá	4
Reorganização	20	São Paulo	1
Novosbenefícios	30	Mauá	4

DEPENDENTE

Fcpf	Nome_dependente	Sexo	Datanasc	Parentesco
33344555587	Alicia	F	05-04-1986	Filha
33344555587	Tiago	M	25-10-1983	Filho
33344555587	Janaína	F	03-05-1958	Esposa
98765432168	Antonio	M	28-02-1942	Marido
12345678966	Michael	M	04-01-1988	Filho
12345678966	Alicia	F	30-12-1988	Filha
12345678966	Elizabeth	F	05-05-1967	Esposa

Consultas de recuperação básicas em SQL



Consultas de recuperação básicas em SQL

A SQL tem uma instrução básica para recuperar informações de um banco de dados: a instrução **SELECT**. Existem muitas opções e tipos de instrução SELECT em SQL, nesta aula vamos utilizar os recursos da SQL para consultas de **recuperação simples**.



A estrutura **SELECT-FROM-WHERE** das consultas SQL básicas

A forma básica do comando **SELECT**, às vezes chamada de mapeamento ou bloco select-from-where, é composta pelas três cláusulas **SELECT**, **FROM** e **WHERE**, e tem a seguinte forma:

SELECT <Lista de atributos>

FROM <Lista de tabelas>

WHERE <Condição>;



A estrutura **SELECT-FROM-WHERE** das consultas SQL básicas

- <Lista de atributos> é uma lista de nomes de atributo cujos valores devem ser recuperados pela consulta.
- <Lista de Tabelas> é uma lista dos nomes de relação exigidos para processar a consulta.
- <Condição> é uma expressão condicional (booleana) que identifica as tuplas a serem recuperadas pela consulta.



A estrutura **SELECT-FROM-WHERE** das consultas SQL básicas

Em SQL, os **operadores** básicos de comparação lógicos para comparar valores de atributo entre si e com constantes literais são =, <, <=, >, >= e <>.

Estes correspondem aos operadores da álgebra relacional =, <, ≤, >, ≥ e ≠, respectivamente, e aos operadores da linguagem de programação C/C++ =, <, <=, >, >= e !=.



A estrutura **SELECT-FROM-WHERE** das consultas SQL básicas

Recuperar a data de nascimento e o endereço do(s) funcionário(s) cujo nome seja 'João B. Silva'.



A estrutura **SELECT-FROM-WHERE** das consultas SQL básicas

Recuperar o nome e endereço de todos os funcionários que trabalham para o departamento 'Pesquisa'.



A estrutura **SELECT-FROM-WHERE** das consultas SQL básicas

Para cada projeto localizado em 'Mauá', liste o número do projeto, o número do departamento que o controla e o sobrenome, endereço e data de nascimento do gerente do departamento



Nomes de atributos ambíguos, apelido, renomeação e variáveis de tupla

Em SQL, o mesmo nome pode ser usado para dois (ou mais) atributos, desde que estes estejam em **relações diferentes**.

```
SELECT F.Pnome, F.Unome, F.Endereco
```

```
FROM FUNCIONARIO F, DEPARTAMENTO D
```

```
WHERE D.DNome='Pesquisa' AND D.Dnumero=F.Dnr;
```




Cláusula WHERE não especificada e uso do asterisco

A falta de uma cláusula WHERE indica que não há condição sobre a seleção de tuplas; logo, todas as tuplas da relação especificada na cláusula FROM se qualificam e são selecionadas para o resultado da consulta.



Cláusula WHERE não especificada e uso do asterisco

Se mais de uma relação for especificada na cláusula FROM e não houver uma cláusula WHERE, então o PRODUTO CARTESIANO — **todas as combinações de tuplas possíveis** — dessas relações será selecionado.



Cláusula WHERE não especificada e uso do asterisco

Selecionar todos os Cpf's de FUNCIONARIO e todas as combinações de Cpf de FUNCIONARIO e Dnome de DEPARTAMENTO no banco de dados



Cláusula WHERE não especificada e uso do asterisco

Para recuperar todos os valores de atributo das tuplas selecionadas, não precisamos listar os nomes de atributo explicitamente em SQL; basta especificar um asterisco (*), que significa todos os atributos.



Tabelas como conjuntos em SQL

A SQL normalmente trata uma tabela não como um conjunto, mas como um **multiconjunto**; **tuplas duplicadas podem aparecer mais de uma vez em uma tabela**, e no resultado de uma consulta.



Tabelas como conjuntos em SQL

Se quisermos eliminar tuplas duplicadas do resultado de uma consulta SQL, usamos a palavra-chave **DISTINCT** na cláusula SELECT, significando que apenas as tuplas distintas deverão permanecer no resultado.

```
SELECT ALL Salario FROM FUNCIONARIO;
```

```
SELECT DISTINCT Salario FROM FUNCIONARIO
```



Tabelas como conjuntos em SQL

A SQL incorporou diretamente algumas das operações de conjunto da teoria de conjuntos da matemática, que também fazem parte da álgebra relacional.

Existem operações de união de conjunto (**UNION**), diferença de conjunto (**EXCEPT**), e interseção de conjunto (**INTERSECT**).



Combinação de padrão de subcadeias e operadores aritméticos

O operador de comparação **LIKE**. Isso pode ser usado para combinação de padrão de cadeia. Cadeias parciais são especificadas usando dois caracteres reservados: % substitui um número qualquer de zero ou mais caracteres, e o sublinhado (_) substitui um único caractere. Ex: Recuperar todos os funcionários cujo endereço esteja em São Paulo.

```
SELECT Pnome, Unome
```

```
FROM FUNCIONARIO
```

```
WHERE Endereco LIKE '%Sao Paulo, SP%';
```




Combinação de padrão de subcadeias e operadores aritméticos

Recuperar todos os funcionários cujo endereço esteja em São Paulo.

```
SELECT Pnome, Unome
```

```
FROM FUNCIONARIO
```

```
WHERE Endereco LIKE '%SaoPaulo,SP%';
```



Combinação de padrão de subcadeias e operadores aritméticos

Encontrar todos os funcionários que nasceram durante a década de 1950.

```
SELECT Pnome, Unome
```

```
FROM FUNCIONARIO
```

```
WHERE Datanasc LIKE '_______5_';
```



Combinação de padrão de subcadeias e operadores aritméticos

Outro recurso permite o uso de aritmética nas consultas. Os operadores aritméticos padrão para adição (+), subtração (-), multiplicação (*) e divisão (/) podem ser aplicados a valores ou atributos numéricos com domínios numéricos



Combinação de padrão de subcadeias e operadores aritméticos

Mostrar os salários resultantes se cada funcionário que trabalha no projeto 'ProdutoX' receber um aumento de 10 por cento.

```
SELECT F.Pnome, F.Unome, 1.1 * F.Salario AS Aumento_salario
FROM FUNCIONARIO AS F, TRABALHA_EM AS T, PROJETO AS P
WHERE F.Cpf=T.Fcpf AND T.Pnr=P.Projnumero AND
P.Projnome='ProdutoX';
```



Combinação de padrão de subcadeias e operadores aritméticos

Para os tipos de dados de cadeia, o operador de concatenação `||` pode ser usado em uma consulta para anexar dois valores de cadeia. Para tipos de dados data, hora, timestamp e intervalo, os operadores incluem incremento (+) ou decremento (-) de uma data, hora ou timestamp por um intervalo



Combinação de padrão de subcadeias e operadores aritméticos

Um valor de intervalo é o resultado da diferença entre dois valores de data, hora ou timestamp. Outro operador de comparação, que pode ser usado por conveniência, é **BETWEEN**.



Combinação de padrão de subcadeias e operadores aritméticos

Recuperar todos os funcionários no departamento 5 cujo salário esteja entre R\$ 30.000 e R\$ 40.000

```
SELECT *
```

```
FROM FUNCIONARIO
```

```
WHERE (Salario BETWEEN 30000 AND 40000) AND Dnr = 5;
```



Ordem dos resultados da consulta

A SQL permite que o usuário ordene as tuplas no resultado de uma consulta pelos valores de um ou mais dos atributos que aparecem, usando a cláusula **ORDER BY**.



Ordem dos resultados da consulta

Recuperar uma lista dos funcionários e dos projetos em que estão trabalhando, ordenada por departamento e, dentro de cada departamento, ordenada alfabeticamente pelo sobrenome, depois pelo nome.

```
SELECT D.Dnome, F.Unome, F.Pnome, P.Projnome
```

```
FROM DEPARTAMENTO D, FUNCIONARIO F, TRABALHA_EM T, PROJETO P WHERE  
D.Dnumero= F.Dnr AND F.Cpf= T.Fcpf AND T.Pnr= P.Projnumero
```

```
ORDER BY D.Dnome, F.Unome, F.Pnome;
```



Ordem dos resultados da consulta

A ordem padrão está em ordem crescente de valores. Podemos especificar a palavra-chave **DESC** se quisermos ver o resultado em uma ordem decrescente de valores. A palavra-chave **ASC** pode ser usada para especificar a ordem crescente explicitamente.



Ordem dos resultados da consulta

Por exemplo, se quisermos a ordem alfabética decrescente de Dnome e ordem crescente de Unome, Pnome, a cláusula ORDER BY da consulta anterior pode ser escrita como:

...

ORDER BY D.Dnome **DESC**, F.Unome **ASC**, F.Pnome ASC;



Exercícios

Desenvolvam os exercícios contidos na lista SQL Básica.



Referência Bibliográfica

ELMASRI, Ramez; NAVATHE, Shamkant B.. Sistemas de banco de dados, 7ª ed., 2018

PUGA, Sandra; FRANÇA, Edson e GOYA, Milton. Banco de dados: Implementação em SQL, PL/SQL e Oracle 11g, 2013.

W3SCHOOL, MySQL Database, <https://www.w3schools.com/mysql/> acessado em 10/02/2023;