



UNIPAC - CENTRO UNIVERSITÁRIO PRESIDENTE ANTÔNIO CARLOS  
CAMPUS BARBACENA

Bacharelado em Ciência da Computação



# Banco de Dados

## Material de Apoio

**Parte VIII – DML - Data Manipulation Language**

Prof. José Osvano da Silva, PMP, PSM I  
joseosvano@unipac.br

1º sem / 2022

# Sumário

- Introdução
- Consultas de recuperação básicas em SQL
- A estrutura SELECT-FROM-WHERE das consultas SQL básicas
- Cláusula WHERE não especificada
- Operadores aritméticos
- Instruções INSERT, DELETE e UPDATE em SQL
- Exercício de Fixação.

# Introdução

- Uma linguagem de manipulação de dados (DML) permite aos usuários acessar ou manipular dados conforme organizados pelo modelo de dados apropriado.
- Compõe uma única linguagem de consulta de banco de dados, como por exemplo, o SQL.
- Se assemelha a uma linguagem simples (do inglês) e torna fácil a interação do usuário com o sistema de banco de dados.
- DML fornece comandos para que os usuários consigam manipular dados em um banco de dados.
- A manipulação envolve inserir, recuperar, excluir e atualizar dados em tabelas de banco de dados.

# Introdução

- **SELECT:** recupera informações armazenadas no banco de dados
- **INSERT:** adiciona novas informações no banco de dados
- **UPDATE:** altera informações armazenadas no banco de dados
- **DELETE:** deleta informações do banco de dados

# Consultas de recuperação básicas em SQL

- Instrução **SELECT**
  - Uma instrução básica para recuperar informações de um banco de dados
- SQL permite que uma tabela tenha duas ou mais tuplas que são idênticas em todos os seus valores de atributo
  - Assim, em geral, uma tabela **SQL** não é um conjunto de tuplas
  - **Multiconjunto** ou comportamento bag



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

## ■ Forma básica do comando **SELECT**:

```
SELECT  <lista atributos>  
FROM    <lista tabelas>  
WHERE   <condição>;
```

onde

- <lista atributos> é uma lista de nomes de atributo cujos valores devem ser recuperados pela consulta.
- <lista 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 (cont.)

- Operadores de comparação lógicos
  - =, <, <=, >, >= e <>
- **Atributos de projeção**
  - Atributos listados na cláusula SELECT, cujos valores devem ser recuperados
- **Condição de seleção**
  - Condição booleana especificada pela cláusula WHERE que deve ser verdadeira para qualquer tupla recuperada

# SELECT Projeção

SELECT Marca, Modelo FROM Taxi



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1 9 9 9
DKL4598	Wolksvagen	Gol	2 0 0 1
DKL7878	Ford	Fiesta	2 0 0 1
JDM8776	Wolksvagen	Santana	2 0 0 2
JJM3692	Chevrolet	Corsa	1 9 9 9



# SELECT Projeção

SELECT **Marca**, **Modelo** FROM Taxi

Marca	Modelo
Ford	Fiesta
Wolkswagen	Gol
Ford	Fiesta
Wolkswagen	Santana
Chevrolet	Corsa

# SELECT Seleção

SELECT \* FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1 9 9 9
DKL4598	Wolksvagen	Gol	2 0 0 1
DKL7878	Ford	Fiesta	2 0 0 1
JDM8776	Wolksvagen	Santana	2 0 0 2
JJM3692	Chevrolet	Corsa	1 9 9 9

# SELECT Seleção

SELECT \* FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1 9 9 9
DKL4598	Wolksvagen	Gol	2 0 0 1
DKL7878	Ford	Fiesta	2 0 0 1
JDM8776	Wolksvagen	Santana	2 0 0 2
JJM3692	Chevrolet	Corsa	1 9 9 9

# SELECT Seleção

SELECT \* FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
DKL4598	Wolksvagen	Gol	2 0 0 1
DKL7878	Ford	Fiesta	2 0 0 1
JDM8776	Wolksvagen	Santana	2 0 0 2



# Banco de dados Empresa

## 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	Erito	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
Novosbeneficios	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	Janaina	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



# A estrutura SELECT-FROM-WHERE das consultas SQL básicas (cont.)

- Instruções e resultados de consultas SQL aplicados ao banco de dados EMPRESA.

**Consulta 0.** Recuperar a data de nascimento e o endereço do(s) funcionário(s) cujo nome seja 'João B. Silva'.

```
C0: SELECT  Datanasc, Endereco
        FROM    FUNCIONARIO
        WHERE   Pnome='João' AND Minicial='B' AND
                Unome='Silva';
```

(a)

<u>Datanasc</u>	<u>Endereco</u>
09-01-1965	Rua das Flores, 751, São Paulo, SP

**Consulta 1.** Recuperar o nome e o endereço de todos os funcionários que trabalham para o departamento 'Pesquisa'.

```
C1. SELECT      Pnome, Unome, Endereco
        FROM      FUNCIONARIO, DEPARTAMENTO
        WHERE      Dnome='Pesquisa' AND
                  Dnumero=Dnr;
```

(b)

<u>Pnome</u>	<u>Unome</u>	<u>Endereco</u>
João	Silva	Rua das Flores, 751, São Paulo, SP
Fernando	Wong	Rua da Lapa, 34, São Paulo, SP
Ronaldo	Lima	Rua Rebouças, 65, Piracicaba, SP
Joice	Leite	Av. Lucas Obes, 74, São Paulo, SP

## A estrutura SELECT-FROM-WHERE das consultas SQL básicas (cont.)

- Instruções e resultados de consultas SQL aplicados ao banco de dados EMPRESA.
- Consulta **seleção-projeção-junção**

**Consulta 2.** 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.

```
C2: SELECT Projnumero, Dnum, Unome, Endereco,
          Datanasc
FROM PROJETO, DEPARTAMENTO,
     FUNCIONARIO
WHERE Dnum=Dnumero AND
      Cpf_gerente=Cpf AND
      Projlocal='Mauá';
```

(c)

<u>Projnumero</u>	<u>Dnum</u>	<u>Unome</u>	<u>Endereco</u>	<u>Datanasc</u>
10	4	Souza	Av. Artur de Lima, 54, Santo André, SP	20-06-1941
30	4	Souza	Av. Artur de Lima, 54, Santo André, SP	20-06-1941

## Nomes de atributos ambíguos

- O mesmo nome pode ser usado para dois (ou mais) atributos em relações diferentes
  - É preciso **qualificar** o nome do atributo com o nome da relação, para evitar ambiguidade
  - Isso é feito *prefixando* o nome da relação ao nome do atributo e separando os dois por um ponto

**Exemplo:** Suponha que os atributos Dnr e Unome da relação FUNCIONARIO fossem chamados de Dnumero e Nome, e o atributo Dnome de DEPARTAMENTO também fosse chamado Nome.

```
C1A: SELECT Pnome, FUNCIONARIO.Nome,  
           Endereco  
FROM   FUNCIONARIO, DEPARTAMENTO  
WHERE  DEPARTAMENTO.Nome='Pesquisa' AND  
       DEPARTAMENTO.  
       Dnumero=FUNCIONARIO.Dnumero;
```

# Apelido, renomeação e variáveis de tupla

- **Apelidos ou variáveis de tupla**

- Declarar nomes de relação alternativos F e S

```
SELECT      F.Pnome, F.Unome, S.Pnome, S.Unome  
FROM        FUNCIONARIO AS F, FUNCIONARIO AS S  
WHERE       F.Cpf_supervisor=S.Cpf;
```

- **Apelidos para atributos**

```
FUNCIONARIO AS F(Pn, Mi, Un, Cpf, Dn, End, Sexo, Sal,  
                  Scpf, Dnr)
```



## Cláusula WHERE não especificada

- A falta de uma cláusula WHERE
  - Indica que não há condição sobre a seleção de tuplas
- 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.

**Consultas 9 e 10.** Selecionar todos os Cpf's de FUNCIONARIO (C9) e todas as combinações de Cpf de FUNCIONARIO e Dnome de DEPARTAMENTO (C10) no banco de dados.

**C9: SELECT** Cpf  
**FROM** FUNCIONARIO;

**C10: SELECT** Cpf, Dnome  
**FROM** FUNCIONARIO, DEPARTAMENTO;



## Uso do asterisco

- Especificar um asterisco (\*)
  - Recuperar todos os valores de atributo das tuplas selecionadas

```
C1C: SELECT *  
      FROM  FUNCIONARIO  
      WHERE Dnr=5;  
C1D: SELECT *  
      FROM  FUNCIONARIO, DEPARTAMENTO  
      WHERE Dnome='Pesquisa' AND Dnr=Dnumero;  
C10A: SELECT *  
      FROM  FUNCIONARIO, DEPARTAMENTO;
```

## Tabelas como conjuntos em SQL

- SQL não elimina automaticamente tuplas duplicadas nos resultados das consultas
- Usa-se a palavra-chave **DISTINCT** na cláusula SELECT
  - Apenas as tuplas distintas deverão permanecer no resultado

**Consulta 11.** Recuperar o salário de cada funcionário (C11) e todos os valores de salário distintos (C11A).

```
C11:  SELECT  ALL Salario
      FROM    FUNCIONARIO;
C11A: SELECT  DISTINCT Salario
      FROM    FUNCIONARIO;
```

## Tabelas como conjuntos em SQL (cont.)

- Operações de conjunto
  - A SQL incorporou algumas das operações da *teoria dos conjuntos* da matemática
  - **UNION, EXCEPT** (diferença), **INTERSECT**
  - Operações multiconjuntos correspondentes: **UNION ALL, EXCEPT ALL, INTERSECT ALL**

**Consulta 4.** Fazer uma lista de todos os números de projeto para aqueles que envolvam um funcionário cujo último nome é 'Silva', seja como um trabalhador ou como um gerente do departamento que controla o projeto.

```
C4A: ( SELECT DISTINCT Projnumero
      FROM PROJETO, DEPARTAMENTO,
      FUNCIONARIO
      WHERE Dnum=Dnumero AND
            Cpf_gerente=Cpf
            AND Unome='Silva' )
      UNION
      ( SELECT DISTINCT Projnumero
        FROM PROJETO, TRABALHA_EM,
        FUNCIONARIO
        WHERE Projnumero=Pnr AND Fcpf=Cpf
          AND Unome='Silva' );
```

## Combinação de padrão de subcadeias

- Operador de comparação **LIKE**
  - Usado para **combinação de padrão** de cadeia
  - % substitui um número qualquer de zero ou mais caracteres

```
SELECT      Pnome, Unome  
FROM        FUNCIONARIO  
WHERE        Endereco LIKE '%Dourados%';
```

- Sublinhado (\_) substitui um único caractere

```
SELECT      Pnome, Unome  
FROM        FUNCIONARIO  
WHERE        Datanasc LIKE '______ 5 _';
```



# Operadores aritméticos

- Operadores aritméticos padrão:
  - Adição (+), subtração (−), multiplicação (\*) e divisão (/)

```
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';
```



# Operadores aritméticos

- Operador de comparação BETWEEN

```
SELECT      *  
FROM        FUNCIONARIO  
WHERE       (Salario BETWEEN 30.000 AND 40.000)  
             AND Dnr = 5;
```

## Ordem dos resultados da consulta

- Use a cláusula **ORDER BY**
  - Palavra-chave **DESC** para ver o resultado em uma ordem decrescente de valores
  - Palavra-chave **ASC** para especificar a ordem crescente explicitamente
    - **ORDER BY** D.Dnome **DESC**, F.Unome **ASC**, F.Pnome **ASC**

```
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;
```

## Discussão e resumo das consultas de recuperação da SQL básica

```
SELECT      <lista atributos>  
FROM        <lista tabelas>  
[ WHERE     <condição> ]  
[ ORDER BY  <lista atributos> ];
```

# Instruções INSERT, DELETE e UPDATE em SQL

- Três comandos usados para modificar o banco de dados:
  - INSERT, DELETE e UPDATE

## O comando INSERT

- 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.

```
U1:  INSERT INTO FUNCIONARIO  
      VALUES      ( 'Ricardo', 'K', 'Marini',  
                    '65329865388', '30-12-  
                    1962', 'Rua Itapira, 44,  
                    Santos, SP', 'M', 37.000,  
                    '65329865388', 4 );
```



## O comando INSERT

- Segunda forma da instrução INSERT
  - Permite especificar nomes de atributos explícitos aos valores fornecidos no comando INSERT
  - Os valores precisam incluir todos os atributos com a especificação NOT NULL

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

## O comando DELETE

- Remove tuplas de uma relação
  - Inclui uma cláusula WHERE para selecionar as tuplas a serem excluídas

```
U4A:  DELETE FROM  FUNCIONARIO
      WHERE        Unome='Braga';
U4B:  DELETE FROM  FUNCIONARIO
      WHERE        Cpf='12345678966';
U4C:  DELETE FROM  FUNCIONARIO
      WHERE        Dnr=5;
U4D:  DELETE FROM  FUNCIONARIO;
```

## O comando UPDATE

- Modifica valores de atributo de uma ou mais tuplas selecionadas
- Cláusula **SET** adicional no comando UPDATE
  - Especifica os atributos a serem modificados e seus novos valores

```
U5: UPDATE PROJETO
    SET      Projlocal = 'Santo André', Dnum
           = 5
    WHERE    Projnumero=10;
```

```
U6: UPDATE  FUNCIONARIO
    SET      Salario = Salario * 1.1
    WHERE    Dnr = 5;
```

## Exercício de Fixação 09

- Basei-se nas Relações Abaixo:
  - Ambulatório (númeroA, andar, capacidade)
  - Médico (CRM, rg, nome, idade, cidade, especialidade, #*númeroA*)
  - Paciente (RG, nome, idade, cidade, doença)
  - Consulta (#CRM, #RG, data, hora)
  - Funcionário (RG, nome, idade, cidade, salário)
- Escreva um comando de insert para cada tabela.
- Salve o arquivo com o nome de “dml.sql” e envie no portal para a correção.



# Dúvidas



**José Osvano da Silva**  
[joseosvano@unipac.br](mailto:joseosvano@unipac.br)