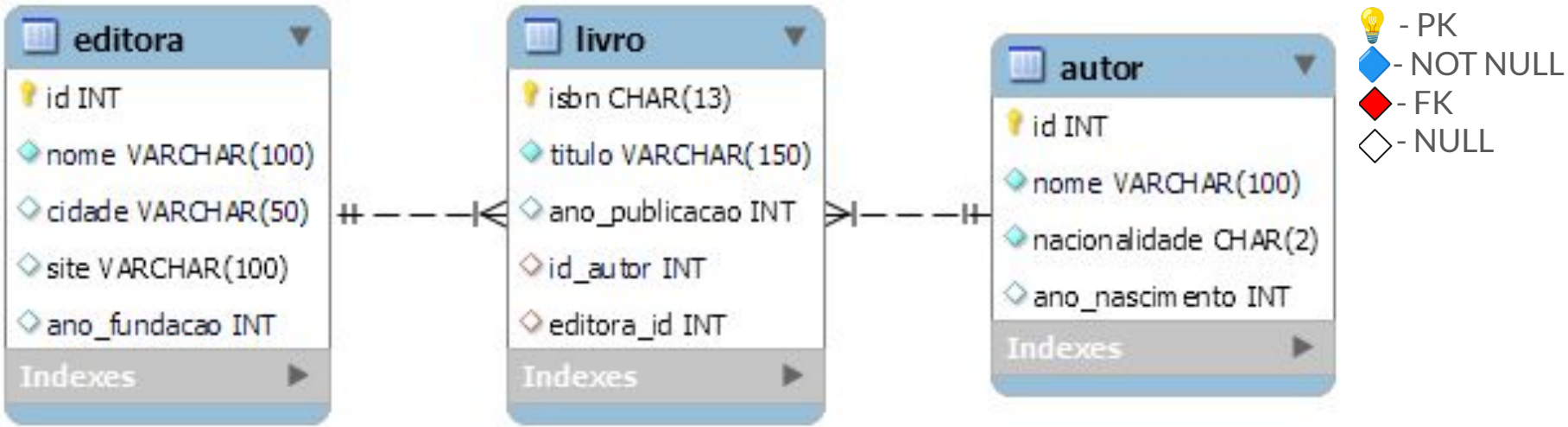
SQL
DML – *Data Manipulation Language*

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

Forward Engineering

O ***Forward Engineering*** é o processo de **gerar o código SQL automaticamente** a partir de um modelo visual (EER Diagram) no MySQL Workbench. Ele cria o script DDL completo, com CREATE DATABASE, CREATE TABLE, FOREIGN KEY, etc.

Forward Engineering



Utilizar identificadores artificiais (AUTO_INCREMENT) para autor e para editora.

DML – *Data Manipulation Language*

A **DML** é utilizada para **inserir, atualizar, deletar e consultar dados** nas tabelas.

Principais comandos DML:

- SELECT: consulta dados.
- INSERT: insere novos dados.
- UPDATE: altera dados existentes.
- DELETE: remove dados.

Instrução INSERT

O comando INSERT

O comando INSERT INTO é utilizado para adicionar novos registros a uma tabela no banco de dados. Há duas maneiras principais de utilizar este comando:

O comando INSERT

Especificando os nomes das colunas e os valores a serem inseridos:

```
INSERT INTO nome_da_tabela (coluna1, coluna2, coluna3, ...)
VALUES (valor1, valor2, valor3, ...);
```

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

Inserir na tabela autores os valores abaixo:

- Machado de Assis, brasileiro, nascido em 1839.
- George Orwell, britânico, nascido em 1903.

O comando INSERT

Inserindo valores para todas as colunas da tabela:

```
INSERT INTO nome_da_tabela
VALUES (valor1, valor2, valor3, ...);
```

O comando INSERT

Inserir na tabela editoras

- Companhia das Letras, localizada em São Paulo, site institucional: https://www.companhiadasletras.com.br, fundada em 1986.
- Penguin Books, localizada em Londres, site institucional: https://www.penguin.co.uk, fundada em 1935.

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 INSERT

```
INSERT INTO nome_da_tabela (coluna1, coluna2, coluna3, ...)
VALUES  (valor1, valor2, valor3, ...),
        (valor1, valor2, valor3, ...),
        (valor1, valor2, valor3, ...);
```

O comando INSERT

```
INSERT INTO nome_da_tabela
VALUES  (valor1, valor2, valor3, ...),
        (valor1, valor2, valor3, ...),
        (valor1, valor2, valor3, ...);
```

O comando INSERT

Inserir os registros abaixo na tabela livro em um único comando INSERT

- Livro "Dom Casmurro", ISBN 9788571644029, publicado em 1899, escrito por Machado de Assis, publicado pela Companhia das Letras.
- Livro "1984", ISBN 9780451524935, publicado em 1949, escrito por George Orwell, publicado pela Penguin Books.

O comando INSERT

Utilize os comandos SQL INSERT INTO para registrar as seguintes informações:

- Livro "A Revolta dos Algoritmos", ISBN 9781234567890, publicado em 2022, escrito por Juca da Silva, publicado pela Editora TecNova.
- Livro "A Revolta dos AlgoritmosII", ISBN 9781234567899, publicado em 2024, escrito por Juca da Silva, publicado pela Editora TecNova
- Autor: Juca da Silva, nacionalidade BR, nascido em 1985.
- Editora Editora TecNova, localizada em Porto Alegre, site institucional: https://www.tecnova.com.br, fundada em 2010.

Instrução DELETE

O comando DELETE

A cláusula DELETE é usada para remover uma ou mais linhas (registros) de uma tabela em um banco de dados. Diferente do comando DROP, que remove a estrutura da tabela, o DELETE remove apenas os dados, mantendo a tabela intacta.

O comando DELETE

```
DELETE FROM nome_da_tabela
WHERE condição;
```

O comando DELETE

Delete informações da tabela autor.

DELETE FROM nome_da_tabela;

O comando DELETE

```
DELETE FROM nome_da_tabela
WHERE condição;
```

NÃO UTILIZAR
DELETE / UPDATE
SEM A CLÁUSULA WHERE

Cláusula WHERE

A cláusula WHERE é usada para especificar **uma condição** que determina quais registros serão afetados pelo comando DELETE. Sem o WHERE, o comando remove todos os dados da tabela.

Cláusula WHERE

Exemplos de condições:

```
WHERE id = 5

WHERE nome = 'Juca da Silva'

WHERE ano_publicacao < 2000

WHERE id_editora IS NULL
```

O comando DELETE

Remova o registro abaixo do banco.

- Livro "A Revolta dos AlgoritmosII", ISBN 9781234567890, publicado em 2024, escrito por Juca da Silva, publicado pela Editora TecNova

Instrução UPDATE

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 nome_da_tabela
SET coluna1 = valor1, coluna2 = valor2, ...
WHERE condição;
```

O comando DELETE

Altera a data de publicação dos livros.

UPDATE nome_da_tabela

SET coluna1 = valor1, coluna2 = valor2, ...

O comando UPDATE

```
UPDATE nome_da_tabela

SET coluna1 = valor1, coluna2 = valor2, ...

WHERE condição;
```

NÃO UTILIZAR
DELETE / UPDATE
SEM A CLÁUSULA WHERE

O comando DELETE

Corrigir as datas de publicação dos livros

- Livro "Dom Casmurro", ISBN 9788571644029, ano original: 1899
- Livro "1984", ISBN 9780451524935, ano original: 1949
- Livro "A Revolta dos Algoritmos", ISBN 9781234567890, ano original: 2022

Prática:

Insira os valores contidos nas tabelas ao lado ao banco 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	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	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

Consultas de recuperação básicas em SQL

Consultas de recuperação básicas em SQL

O comando SELECT é utilizado para **selecionar dados** de um banco de dados. Os dados retornados são armazenados em uma tabela de resultado, chamada de conjunto de resultados (result-set).

A estrutura SELECT-FROM-WHERE

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.

A estrutura SELECT-FROM-WHERE

```
SELECT    <Lista de atributos>
FROM      <Lista de tabelas>
WHERE     <Condição>;
```

A estrutura SELECT-FROM-WHERE

- **<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.

Cláusula WHERE

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

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.

Cláusula WHERE

Selecionar todos os registros contidos na tabela funcionario

Cláusula WHERE

```
SELECT *

FROM FUNCIONARIOS.
```

A estrutura SELECT-FROM-WHERE

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 **!=.**

Operador	Descrição	Exemplo
=	Igual	WHERE coluna = valor
>	Maior que	WHERE coluna > valor
<	Menor que	WHERE coluna < valor
>=	Maior ou igual	WHERE coluna >= valor
<=	Menor ou igual	WHERE coluna <= valor
<> ou !=	Diferente.	WHERE coluna <> valor
BETWEEN	Entre um determinado intervalo	WHERE coluna BETWEEN v1 AND v2
LIKE	Buscar por um padrão	WHERE coluna LIKE 'padrão%'
AND	Combinação de duas ou mais condições (E)	WHERE coluna1 = x AND coluna2 > y
OR	Combinação de condições alternativas (OU)	WHERE coluna1 = x OR coluna2 = y

A estrutura SELECT-FROM-WHERE

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

A estrutura SELECT-FROM-WHERE

```
SELECT Datanasc, Endereco

FROM FUNCIONARIO

WHERE Pnome = 'João' AND Minicial = 'B' AND Unome = 'Silva';
```

A estrutura SELECT-FROM-WHERE

Recuperar o nome completo dos funcionários que tem um salário maior que 29 mil e menor que 41 mil.

A estrutura SELECT-FROM-WHERE

```
SELECT Pnome, Minicial, Unome
FROM FUNCIONARIO
WHERE Salario > 29000 AND Salario < 41000;
```

A estrutura SELECT-FROM-WHERE

Recuperar todas as informações do(s) funcionário(s) que tem salário maior ou igual a 40 mil.

A estrutura SELECT-FROM-WHERE

```
SELECT *

FROM FUNCIONARIO

WHERE Salario >= 40000;
```

A estrutura SELECT-FROM-WHERE

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.

A estrutura SELECT-FROM-WHERE

Recuperar informações dos funcionários e departamentos, sem utilizar a cláusula where.

A estrutura SELECT-FROM-WHERE

```
SELECT *

FROM FUNCIONARIO, DEPARTAMENTO
```

A estrutura SELECT-FROM-WHERE

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

A estrutura SELECT-FROM-WHERE

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

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

A estrutura SELECT-FROM-WHERE

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

A estrutura SELECT-FROM-WHERE

```
SELECT P.Projnumero, P.Dnum, F.Unome, F.Endereco, F.Datanasc
FROM PROJETO P, DEPARTAMENTO D, FUNCIONARIO F
WHERE P.Projlocal = 'Mauá'
    AND P.Dnum = D.Dnumero
    AND D.Cpf_gerente = F.Cpf;
```

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

Selecionar somente os salários dos funcionário.

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

O que é recuperado nas consultas acima?

Operador LIKE

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.

Operador LIKE

```
SELECT Pnome, Unome

FROM FUNCIONARIO

WHERE Endereco LIKE ‘%valor%’;
```

Operador LIKE

Recuperar todos os funcionários cujo endereço esteja em São Paulo.

Operador LIKE

```
SELECT Pnome, Unome

FROM FUNCIONARIO

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

Operador LIKE

Encontrar todos os funcionários cujo primeiro nome tem exatamente 5 letras.

Operador LIKE

```
SELECT Pnome, Unome
FROM FUNCIONARIO
WHERE Pnome LIKE '_____';
```

Operador LIKE

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

Dica: use YEAR(Datanasc) para recuperar o ano.

Combinação de padrão de subcadeias e operadores aritméticos

```
SELECT Pnome, Unome, Datanasc

FROM FUNCIONARIO

WHERE YEAR(Datanasc) LIKE '195_'

…. outra forma

WHERE YEAR(Datanasc) BETWEEN 1950 AND 1959;
```

Operadores aritméticos

Os operadores aritméticos padrão para adição (+), subtração (–), multiplicação (*) e divisão (/) podem ser a**plicados a valores ou atributos numéricos** com domínios numéricos

Operadores aritméticos

Operador	Descrição	Exemplo Genérico
+	Adição entre valores ou colunas	SELECT salario + bonus AS total_recebido
-	Subtração entre valores ou colunas	SELECT preco - desconto AS preco_final
*	Multiplicação entre valores ou colunas	SELECT quantidade * valor_unitario AS total_venda
/	Divisão entre valores ou colunas	SELECT distancia / tempo AS velocidade_media

Operadores aritméticos

Apresente os salários atualizados dos funcionários que estão alocados no projeto 'ProdutoX', considerando um reajuste de 10% sobre seus salários atuais.

A consulta deve exibir o nome do funcionário e o salário reajustado.

Operadores aritméticos

```
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';
```

Operador BETWEEN

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.

Operador BETWEEN

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

Operador BETWEEN

```
SELECT *
FROM FUNCIONARIO
WHERE (Salario BETWEEN 30000 AND 40000) AND Dnr = 5;
```

Cláusula ORDER BY

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.

Cláusula ORDER BY

Liste o nome e o salário de todos os funcionários, ordenando pelo salário.

Cláusula ORDER BY

```
SELECT Pnome, Salario
FROM FUNCIONARIO
ORDER BY Salario;
```

Cláusula ORDER BY

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.

Cláusula ORDER BY

```
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;
```

Cláusula ORDER BY

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.

Cláusula ORDER BY

Liste o nome e o salário de todos os funcionários, ordenando o resultado em ordem crescente de salário.

Cláusula ORDER BY

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

HEUSER, Carlos Alberto. Projeto de banco de dados. 4. ed. Porto Alegre: Sagra Luzzatto, 2001. 204 p.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistema de banco de dados. Rio de Janeiro, RJ: Elsevier, 2006. 781 p.

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

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