



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RORAIMA
Campus Ji-Paraná

BANCO DE DADOS II

PROF. JACKSON HENRIQUE

APRESENTAÇÃO DA DISCIPLINA

Curso: Técnico em
Informática
Integrado ao
Ensino Médio

Ano: 3º ano

Carga Horária:
80 horas

CH Prática: 40h
CH Teórica: 40h

PLANO DE ENSINO DA DISCIPLINA

- **Objetivos**

- ✓ Conhecer os mecanismos de seleção de dados no SGBD;
- ✓ Aplicar funções e rotinas em banco de dados;
- ✓ Implementar a segurança em banco de dados;
- ✓ Integrar o banco de dados com a aplicação;

PLANO DE ENSINO DA DISCIPLINA

- **Ementa**

- ✓ Procedimentos armazenados;
- ✓ Gatilhos;
- ✓ Linguagem de Seleção de Dados (DQL);
- ✓ Transações;
- ✓ Linguagem de Controle de Dados (DCL);
- ✓ Backup e Recuperação de Falhas;

BIBLIOGRAFIA BÁSICA

ANGELOTTI, Elaini Simoni. **Banco de dados.**
Curitiba: Editora do Livro Técnico, 2010



HEUSER. C. A. **Projeto de banco de dados.**
6.ed., São Paulo: Bookman, 2009.



BEIGHLEY, Lynn. **Use a Cabeça! SQL.** Rio de Janeiro. Alta Books, 2010.



COSTA, Rogério Luís de C. **SQL: Guia prático.**
2.ed. Rio de Janeiro: Brasport, 2007

BIBLIOGRAFIA COMPLEMENTAR

- COUGO, Paulo Sérgio. **Modelagem conceitual e projeto de banco de dados**. Rio de Janeiro: *Campus*, 2001;
- OLIVEIRA, Celso Henrique Poderoso de. **SQL: Curso prático**. São Paulo: Novatec, 2009;
- MANNINO, Michael V. **Projeto, desenvolvimento de aplicações e administração de banco de dados**. São Paulo: Mc Graw Hill, 2008;
- TAKAHASHI, Mana. **Guia Mangá de Banco de Dados**. São Paulo: Novatec, 2009;
- TEORY, Toby. **Projeto e Modelagem de Banco de Dados**. 2. ed. *Campus*: 2013;
- DATE, C. J. **Introdução a sistemas de banco de dados**. 8.ed. Rio de Janeiro: *Campus*, 2003;

AVALIAÇÃO 1ª ETAPA

- ✓ **Participação nas Atividades: 40 pontos**
- ✓ **Prova Final: 60 pontos**
- ✓ **+ Pontos Extras – Kahootit!**

A decorative graphic consisting of thin, dark blue lines that resemble a circuit board. These lines extend horizontally from the left and right edges of the central black rectangle, with several small circles at the end of the lines, suggesting connection points or components.

REVISÃO MODELAGEM

VAMOS RELEMBRAR

- Modele um diagrama de entidade-relacionamento para o sistema de uma **Farmácia**. Segue informações sobre o negócio:
 - Saiba que essa farmácia vende remédios, produtos de beleza e chocolates e bebidas;
 - O sistema deve controlar o estoque dos produtos, permitindo que o usuário saiba tudo que sai na venda e entra na compra;
 - O sistema deve controlar o financeiro, para saber tudo que entra no recebimento de uma venda e sai no pagamento de uma despesa ou compra;
- ❖ LEMBRE-SE: Analise antes de projetar...

A decorative graphic consisting of thin, dark blue lines that resemble a circuit board or a network diagram. These lines extend horizontally from the left and right sides of the central black rectangle, with several small circles at the end of the lines, suggesting connection points or nodes.

REVISÃO LINGUAGEM SQL

INTRODUÇÃO AO SQL

- Todo **Sistema Gerenciador de Banco de Dados (SGBD)** deve oferecer aos seus usuários e administradores meios de criar e manipular dados armazenados em seu banco de dados;
- A linguagem SQL, sigla em inglês para ***Structured Query Language***, ou **Linguagem de Consulta Estruturada**, é uma espécie de padrão adotado por diferentes SGBDs de modo a facilitar a comunicação e a integração dos sistemas;

INTRODUÇÃO AO SQL

- O SQL tem o objetivo de acessar, de forma estruturada e declarativa, elementos de tabelas de bancos de dados;
- Sua principal tarefa é **criar e manter objetos, manipular e recuperar** dados do banco de dados, administrar dados e manter a segurança.;

INTRODUÇÃO AO SQL

- SQL é caracterizada pela utilização de palavras-chaves que podem ser classificadas, de acordo com sua função, as três principais são:

1. DML;


2. DDL;

3. DQL;



DDL

LINGUAGEM DE DEFINIÇÃO DE DADOS

- Permite ao utilizador criar novas tabelas, alterar a sua estrutura ou excluí-las;
 - Comandos:
CREATE (Criar)
DROP (Excluir)
ALTER (Alterar)
- 

DML

LINGUAGEM DE MANIPULAÇÃO DE DADOS

- Subconjunto do SQL que permite ao utilizador inserir dados em uma tabela, atualizar os dados inseridos ou deletar os dados excluídos;
- **Comandos:**

INSERT (Inserir)

UPDATE (Atualizar)

DELETE (Deletar)

The slide features a dark blue background with white decorative circuit lines. These lines, consisting of straight segments and small circles at junctions, are positioned along the left and right edges of the slide, framing the central content.

DQL

LINGUAGEM DE CONSULTA DE DADOS

- Permite ao utilizar selecionar de diversas maneiras os dados inseridos no banco de dados. Através de filtros e junções, possibilita ao utilizador criar novas informações a partir dos dados disponíveis nas tabelas;
- **Comando:**

SELECT (Selecionar)

COMANDOS BÁSICOS

- **Criação de Banco de Dados**
- **Sintaxe:** *CREATE DATABASE*
nome_do_banco_de_dados;
- **Exemplo:** *CREATE DATABASE locadora;*

COMANDOS BÁSICOS

- **Usar o banco de dados**
- **Sintaxe:** `USE nome_do_banco_de_dados;`
- **Exemplo:** `USE locadora;`

DDL - CREATE (CRIAR)

- **Sintaxe:**

```
CREATE TABLE nome_da_tabela (  
    atributo 1 tipo1,  
    atributo 2 tipo 2,  
    ...  
    atributo N tipo N  
);
```

DDL - CREATE (CRIAR)

- **CREATE TABLE** Cliente (
- **codigo_cli** INT,
- **nome_cli** VARCHAR(30),
- **cpf_cli** VARCHAR(12),
- **data_nasc_cli** DATE,
- **sexo_cli** VARCHAR(1),
- **email_cli** VARCHAR(50)
- **);**

DDL - CREATE (CRIAR)

- IMPORTANTE:
 - ✓ Todo atributo deve possuir um sufixo ou prefixo que identifique a sua tabela de origem;
 - ✓ Devem possuir de 3 a 5 letras;
 - ✓ Devem ser separados por underline do nome do atributo;

```

create table Departamento (
    cod_Dep int not null auto_increment primary key,
    nome_Dep varchar(100)
);

create table Funcionario (
    codigo_Func int not null auto_increment,
    nome_Func varchar(200),
    função_Func varchar(50),
    salario_Func numeric(7, 2),
    cpf_Func int,
    rg_Func int,
    endereço_Func varchar(200),
    sexo_Func varchar(20),
    telefone_Func varchar(30),
    celular_Func varchar(30),
    cod_Depa int not null,
    primary key (codigo_Func),
    foreign key (cod_Depa) references Departamento (cod_Dep)
);

create table Dependente(
    codigo_Dep int not null auto_increment primary key,
    nome_Dep varchar(100) not null,
    idade_Dep int,
    dataNasc_Dep date,
    nomePai_Dep varchar(100),
    nomeMãe_Dep varchar(100),
    codTitular int not null,
    foreign key (codTitular) references Funcionario (codigo_Func)
);

```

CHAVE ESTRANGEIRA

- Observe que Funcionário recebe a Chave Primária de Departamento;
- Essa chave é chamada de Chave Estrangeira;
- É usada para vincular as duas tabelas;

DDL – DROP (APAGAR)

- Comando usado para deletar uma tabela no banco de dados;
- Síntese: **DROP TABLE nome_da_tabela;**
- Exemplo: **DROP TABLE cliente;**

DDL – ALTER (ALTERAÇÃO)

- Usando o comando ALTER é possível realizar as seguintes alterações na estrutura de uma tabela:
 - *Adicionar colunas;*
 - *Excluir colunas;*
 - *Alterar o tipo e o nome de uma coluna já existente.*

DDL – ALTER (ALTERAÇÃO)

- **Comando para adicionar uma nova coluna em uma tabela já existente;**
- **Síntese:** ALTER TABLE nome_da_tabela ADD coluna tipo;
- **Exemplo:** ALTER TABLE cliente ADD endereco_cli
VARCHAR(100);

DDL – ALTER (ALTERAÇÃO)

- As palavras chaves como FIRST (primeiro), AFTER (após), BEFORE (antes) e LAST (por último) pode ser utilizadas para posicionar a **nova coluna** na posição que desejar na tabela. Também podem ser utilizadas as palavras SECOND (segundo), THIRD (terceiro) e assim por diante.
- **ALTER TABLE cliente ADD email_cli VARCHAR(50) FIRST;**
- **ALTER TABLE cliente ADD email_cli VARCHAR(200) AFTER cpf_cli;**

DDL – ALTER (ALTERAÇÃO)

- **Comando para excluir uma coluna de uma tabela já existente;**
- **Síntese:** ALTER TABLE nome_da_tabela DROP COLUMN atributo;
- **Exemplo:**
- ALTER TABLE cliente DROP COLUMN email_cli;

DDL – ALTER (ALTERAÇÃO)

- **Comando para alterar o nome de uma coluna já existente em uma tabela;**
- **Síntese:** ALTER TABLE nome_da_tabela CHANGE COLUMN nome_da_coluna novo_nome_da_coluna tipo;
- **Exemplo:** ALTER TABLE cliente CHANGE COLUMN email_cli endereco_eletronico_cli VARCHAR(80);

DDL – ALTER (ALTERAÇÃO)

- **Comando para alterar o tipo de uma coluna já existente em uma tabela;**
- **Síntese:** ALTER TABLE nome_da_tabela MODIFY COLUMN nome_da_coluna tipo;
- **Exemplo:** ALTER TABLE cliente MODIFY COLUMN nome_cli VARCHAR(50);

DML – INSERT (INSERIR)

- **Comando para inserir um registro, que equivale a uma linha, em uma tabela já existente;**
- **Síntese:** INSERT INTO nome_da_tabela (atributo1, atributo2, ...) VALUES (valor1, valor2, ...);
- **Exemplos:**
 - ✓ INSERT INTO cliente (codigo_cli, nome_cli, cpf_cli, data_nasc_cli, sexo_cli, email_cli) VALUES (1, 'José da Silva', '123456789-10', '1980-12-10', 'M', 'joseSilva@cursoSQL.com');
 - ✓ INSERT INTO cliente VALUES (3, 'Francisco da Silva', '109876543-21', '1990-01-01', 'M', 'franciscoSilva@cursoSQL.com');
 - ✓ INSERT INTO cliente (codigo_cli, nome_cli, sexo_cli, email_cli) VALUES (4, 'Francisca da Silva', 'F', 'franciscaSilva@cursoSQL.com');

DML – INSERT (INSERIR)

- **LEMBRE-SE:**

- ✓ Registros do tipo *varchar* são inseridos entre aspas simples;
- ✓ Chaves primárias não se repetem!
- ✓ Utilize **auto increment** na criação do atributo da Chave Primária, com isso não é preciso mencionar a Chave Primária no insert;
- ✓ Atributos com *not null* são obrigatórios e sem *not null* são opcionais;
- ✓ Registros do tipo *date* são inseridos ao contrário e entre aspas simples. Ex: '1990-12-31'
- ✓ Existem 03 formas de construir um *insert*:
 1. Mencionando todos os atributos
 2. Sem mencionar os atributos
 3. Mencionando parcialmente os atributos

DML – UPDATE (ATUALIZAR)

- Comando para alterar o valor de um atributo ou vários registros de uma tabela;
- **Síntese:** UPDATE nome_da_tabela SET atributo = valor WHERE condição;
- **Exemplos:**
 - ✓ UPDATE cliente SET email_cli = 'silvaJose@gmail.com' WHERE codigo_cli = 50;
 - ✓ UPDATE funcionario SET salario_func = 5000 WHERE funcao_func = "Vendedor";

DML – UPDATE (ATUALIZAR)

- **LEMBRE-SE:** A condição (WHERE) do Update serve para selecionar quais os registros serão alterados;
- **AJUSTE NO SGBD MYSQL:**
 - ✓ Por padrão o MySQL Workbench bloqueia o update e delet em diversos registros de uma só vez;
 - ✓ Para permitir, vá em **Edit ---> Preferences ---> SQL Editor -> Desmarque a caixa de seleção** da opção “Safe Updates.”
Forbide updates...

UPDATE COM AND E OR

- Você pode usar operações de comparação **AND** e **OR** na CONDIÇÃO de um UPDATE para selecionar os registros a serem alterados;
 - **AND:** Todas as comparações devem ser verdadeiras;
 - **OR:** Somente uma das comparações precisa ser verdadeira;
- **Exemplos:**
 - ✓ UPDATE funcionario SET salario_func = 4000 WHERE (funcao_func = "Vendedor") **AND** (salario_func <= 2000);
 - ✓ UPDATE funcionario SET salario_func = 1000 WHERE (salario_func = 800) **OR** (salario_func = 900);

UPDATE COM MAIOR E MENOR

- Você pode usar o parâmetro *maior* > ou *menor* < para comparar os registros na CONDIÇÃO do UPDATE;
- **Exemplo:**
 - ✓ UPDATE cliente SET idade_cli = 31 WHERE data_nasc_cli > '1987-06-30';
 - ✓ UPDATE funcionario SET salario_func = 2000 WHERE idade_func < 60;

DML – DELETE (DELETAR)

- **Comando para excluir um registro (linha) ou vários registros (linhas) de uma tabela;**
- **Síntese:** DELETE FROM nome_da_tabela WHERE condição;
- **Exemplos:**
 - ✓ DELETE FROM cliente WHERE sexo_cli = 'Masculino';
 - ✓ DELETE FROM funcionario WHERE funcao_func = 'Entregador';

DML – DELETE (DELETAR)

- **CUIDADO:** Comandos delete sem uma CONDIÇÃO (WHERE) excluem todos os registros de uma tabela;
- **Exemplo:** DELETE FROM Clientes;
- **Observações:**
 - **Não existe** o comando delete de um **atributo específico**;
 - O comando delete apaga toda a linha do registro selecionado;
 - Para apagar o valor de apenas um atributo deve ser usado o comando **UPDATE**. Exemplo: **UPDATE** cliente **SET** nome_cli = “ where codigo_cli = 10;

VER ESTRUTURA DA TABELA

- O comando DESC mostra a estrutura física da tabela;
- **Síntese:**
- DESC nome_da_tabela;
- **Exemplo:**
- DESC cliente;



EXERCÍCIOS DE REVISÃO

- Acesse o AVA e baixe o arquivo **Lista de Exercícios I - Revisão Comandos DML e DDL – Parte 1 - 1ª Etapa;**
- Com os conhecimentos de hoje desenvolva a lista;
- Utilize o MySQL Workbench disponível no AVA ou no Laboratório de Informática;