**Atividade de Banco de Dados: Projeto Completo e Detalhado**

**Objetivo:** Desenvolver um sistema de gerenciamento de dados utilizando o MySQL, abordando desde a criação do banco de dados e tabelas até a implementação de operações CRUD (Create, Read, Update, Delete), normalização de dados, diferentes tipos de JOINs, agregações e documentação técnica completa.

**Parte 1: Criação e Configuração do Banco de Dados**

**1. Criação do Banco de Dados**

* Crie um banco de dados chamado sistema\_gestao.

create database sistema\_gestao;

use sistema\_gestao;

**2. Criação de Tabelas**

* Crie as seguintes tabelas no banco de dados sistema\_gestao:
  + **clientes**: id, nome, telefone, data\_nascimento, estado\_civil, sexo, cidade, idade.

CREATE TABLE `clientes` (

`id` int NOT NULL AUTO\_INCREMENT,

`nome` varchar(200) NOT NULL,

`telefone` varchar(15) NOT NULL,

`data\_nascimento` date NOT NULL,

`estado\_civil` varchar(10) NOT NULL,

`sexo` varchar(10) NOT NULL,

`idade` int NOT NULL,

`cidade` varchar(100) NOT NULL,

PRIMARY KEY (`id`)

* + **pedidos**: id, cliente\_id (chave estrangeira para a tabela clientes), produto, quantidade.

CREATE TABLE `sistema\_gestao`.`pedidos` (

`id` INT NOT NULL AUTO\_INCREMENT,

`cliente\_id` INT NOT NULL,

`produto` VARCHAR(100) NOT NULL,

`quant` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_pedido\_cliente\_idx` (`cliente\_id` ASC) VISIBLE,

CONSTRAINT `fk\_pedido\_cliente`

FOREIGN KEY (`cliente\_id`)

REFERENCES `sistema\_gestao`.`clientes` (`id`)

ON DELETE RESTRICT

ON UPDATE RESTRICT);

* + **produtos**: id, nome, preco.

CREATE TABLE `sistema\_gestao`.`produtos` (

`id` INT NOT NULL AUTO\_INCREMENT,

`nome` VARCHAR(100) NOT NULL,

`preço` DECIMAL(10,2) NOT NULL,

PRIMARY KEY (`id`));

* + **estoque**: produto\_id (chave estrangeira para a tabela produtos), quantidade\_disponivel.

CREATE TABLE `sistema\_gestao`.`estoque` (

`produto\_id` INT NOT NULL,

`qntd\_dispon` INT NOT NULL,

INDEX `fk\_estoque\_produto\_idx` (`produto\_id` ASC) VISIBLE,

CONSTRAINT `fk\_estoque\_produto`

FOREIGN KEY (`produto\_id`)

REFERENCES `sistema\_gestao`.`produtos` (`id`)

ON DELETE RESTRICT

ON UPDATE RESTRICT);

**3. Inserção de Dados**

* Insira dados nas tabelas conforme os exemplos fornecidos no documento.

insert into `clientes` values

(1, 'Ana Cláudia Santos', '86998987474', 19881102, 'solteiro', 'feminino', 36, 'Parnaíba'),

(2, 'Bernardo da Silva', '86965658484', 19980408, 'casado', 'masculino', 26, 'Parnaíba'),

(3, 'Flávia Lima Pereira', '31945484542', 19780102, 'casado', 'feminino', 46, 'Uberlândia'),

(4, 'Igor Azevedo Silva', '21985522558', 19790612, 'solteiro', 'masculino', 45, 'Rio de Janeiro'),

(5, 'Mário Constancio Júnior', '21974757671', 19631005, 'solteiro', 'masculino', 61, 'Rio de Janeiro'),

(6, 'Regis Nobreira Nobre', '11986858689', 19900102, 'solteiro', 'masculino', 34, 'São Paulo');

insert into `produtos` values

(1, 'Case Redmi 11', 11.90),

(2, 'Case Redmi 12', 11.90),

(3, 'Case Redmi 13 Note', 11.90),

(4, 'Case iPhone 12', 14.90),

(5, 'Case iPhone 13', 14.90),

(6, 'Case iPhone 13 Pro', 14.90),

(7, 'Case Samsung S21 Ultra', 12.90),

(8, 'Case Samsung S23', 12.90),

(9, 'Case Samsung S23 Note', 12.90),

(10, 'Case Moto G54', 11.90);

insert into `estoque` values

(1, 20),

(2, 17),

(3, 18),

(4, 18),

(5, 11),

(6, 12),

(7, 20),

(8, 19),

(9, 17),

(10, 20);

insert into `pedidos` values

(1, 3, 'Case Redmi 13 Note', 1),

(2, 6, 'Case Samsung S23', 1),

(3, 3, 'Case Samsung S21 Ultra', 1),

(4, 2, 'Case iPhone 12', 1),

(5, 1, 'Case Redmi 13 Note', 2);

**Parte 2: Operações CRUD e Consultas SQL**

**4. Operações CRUD**

* Realize operações de inserção, leitura, atualização e exclusão de dados em cada uma das tabelas.

-- inserção

alter table clientes add column endereco varchar(255);

update clientes set endereco = 'Rua das Estrelas, 1234' where id = 1;

update clientes set endereco = 'Rua das Pimentas, 1235' where id = 2;

update clientes set endereco = 'Rua das Palmeiras, 1236' where id = 3;

update clientes set endereco = 'Rua dos Buracos, 1231' where id = 4;

update clientes set endereco = 'Rua das Flores, 1238' where id = 5;

update clientes set endereco = 'Rua dos Elefantes, 1234' where id = 6;

-- leitura

select \* from clientes;

select \* from pedidos;

select \* from produtos;

select \* from estoque;

-- atualização

update clientes set nome = 'Ana Cláudia dos Santos Silva' where id = 1;

update estoque set qntd\_dispon = 18 where produto\_id = 10;

update pedidos set quant = 2 where id = 2;

update produtos set preço = 10.9 where id = 10;

-- exclusão

delete from produtos where id = 9;

delete from clientes where id = 4;

**5. Consultas SQL Avançadas**

* Realize as seguintes consultas:
  + Selecione todos os clientes que têm mais de 30 anos.

select \* from clientes where idade > 30;

* + Atualize o endereço de um cliente específico.

update clientes set endereco = 'Rua das Embarcações' where id = 6;

* + Delete todos os pedidos feitos por um cliente específico.

select \* from pedidos;

delete from pedidos where cliente\_id = 3;

select \* from pedidos;

* + Insira múltiplos produtos no estoque de uma só vez.

update estoque set qntd\_dispon = qntd\_dispon + 5 where produto\_id;

* + Atualize o preço de todos os produtos com um aumento de 10%.

update produtos set preço = preço + (preço \* 0.10);

precisou do cód: set sql\_safe\_updates = 0;

**6. Consultas Específicas**

* Utilize filtros e ordenações para:
  + Encontrar clientes cujo nome começa com 'A'.

select \* from clientes where nome like 'A%' order by idade asc;

* + Listar pedidos feitos em um intervalo de datas.

select \* from pedidos where data\_venda between date(20241201) and date(20241231);

* + Ordenar produtos pelo preço, do mais caro ao mais barato.

select \* from produtos order by preço desc;

* + Buscar produtos com quantidade disponível menor que 10.

select \* from estoque where qntd\_dispon < 10;

**Parte 3: Normalização de Dados**

**7. Normalização**

* Normalize as tabelas de acordo com as três primeiras Formas Normais (1FN, 2FN e 3FN):
  + Reestruturação para Primeira Forma Normal (1FN)
  + Reestruturação para Segunda Forma Normal (2FN)
  + Reestruturação para Terceira Forma Normal (3FN)

**Parte 4: Join e Relacionamentos**

**8. Utilização de JOINs**

* Utilize diferentes tipos de JOINs para combinar dados das tabelas:
  + **INNER JOIN**: Selecione todos os clientes que fizeram pedidos, juntamente com os detalhes dos pedidos.
  + **LEFT JOIN**: Liste todos os clientes, independentemente de terem feito pedidos ou não.
  + **RIGHT JOIN**: Encontre todos os produtos em estoque, juntamente com os detalhes dos produtos.
  + **FULL JOIN**: Combine os resultados de clientes e pedidos, mesmo aqueles sem correspondências (simulado com UNION).

**Parte 5: Agregações e Funções**

**9. Funções de Agregação**

* Utilize funções de agregação para:
  + Contar o número total de clientes.

select count(\*) as total\_clientes from clientes;

* + Calcular a soma total de todas as vendas.

select count(\*) as 'soma total de vendas' from pedidos;

* + Encontrar a média de idade dos clientes.

select avg(idade) as media\_idades from clientes;

* + Determinar a idade mínima e máxima dos clientes.

select min(idade) as 'idade mínima' from clientes;

select max(idade) as 'idade máxima' from clientes;

* + Contar quantas cidades distintas existem na tabela de clientes.

select count(distinct cidade) as qnt\_cidades from clientes;

**Parte 6: Documentação Técnica Completa**

**10. Documentação Técnica**

* Crie a documentação completa do sistema, incluindo:
  + **Requisitos do Sistema**: Descrição das funcionalidades esperadas.
  + **Diagrama Entidade-Relacionamento (ERD)**: Representação visual das tabelas e seus relacionamentos.
  + **Especificações de Tabelas**: Detalhamento de cada tabela com seus campos, tipos de dados, chaves primárias e estrangeiras, e quaisquer restrições.
  + **Plano de Backup e Recuperação de Dados**: Procedimentos para garantir a segurança dos dados.(pesquisar)
  + **Controle de Acesso e Segurança**: Definição de permissões e segurança para os usuários do banco de dados.(Pesquisar)

**Parte 7: Exercícios Práticos**

**11. Exercícios Práticos**

* Responda as seguintes perguntas:
  1. Explique, com suas palavras, o que é um banco de dados e por que ele é importante.
  2. Compare bancos de dados relacionais e não-relacionais, destacando suas principais diferenças e exemplos de uso.
  3. Descreva as especificações de uma tabela Produto com as colunas id, nome, preco e quantidade.
  4. Pesquise sobre as diferenças entre MySQL e PostgreSQL e liste suas vantagens e desvantagens.
  5. Crie uma documentação básica para um sistema de gerenciamento de estoque, incluindo requisitos do sistema e diagrama ERD.
  6. Explique a função de um SGBD e cite três exemplos de SGBDs populares.
  7. Descreva como implementar um controle de acesso básico no MySQL.(Pesquisar)
  8. Explique a importância da normalização de dados e dê exemplos práticos.

**Entrega**

* **Código SQL**: Incluindo a criação do banco de dados, tabelas, inserção de dados, operações CRUD, normalização, JOINs e agregações.
* **Documentação**: Arquivo contendo os requisitos do sistema, diagrama ERD, especificações das tabelas, plano de backup e recuperação de dados, e controle de acesso e segurança.
* **Respostas dos Exercícios**: com as respostas às perguntas práticas.