# Universidade Federal Do Piauí – UFPI

Campus Senador Helvídio Nunes De Barros – Picos Bacharelado Em Sistemas De Informação Disciplina: Banco De Dados II

# Professor: Glauber Dias Gonçalves Lista de atividades 01 – revisão BD-I

***Alunos: João dos Santos Neto e Jamile Jovita da Silva***

Requisitos funcionais para banco de dados de uma loja de telefonia móvel

Uma loja de telefonia móvel, deseja armazenar o cadastro de seus aparelhos e seus respectivos planos. Cada aparelho é vendido por um vendedor, podendo este último vender vários aparelhos, o que influenciará diretamente em suas comissões, um aparelho poderá ou não estar vinculado a um plano. A loja deseja armazenar ainda, o cadastro de seus clientes, cada cliente pode possuir até cinco planos, podendo esses serem de voz ou de internet. A loja deseja possuir um controle sobre as necessidades acima relatadas.

Baseando-se nessas informações, execute as seguintes etapas:

1. Implemente o projeto lógico de banco de dados via diagrama de entidades e relacionamentos (DER) que contemple um banco de dados relacional para o cenário acima descrito
2. De posse do DER, determine o projeto físico do banco de dados utilizando linguagem de definição de dados (DDL), especificamente, as declarações de criação (*create*), remoção (*drop*) ou alteração (*alter*).
3. Responda os questionamentos abaixo, transformando-os para linguagem de manipulação de dados, especificamente, consultas SQL.
4. Quais clientes compraram em um determinado período?
5. Quais aparelhos estão disponíveis à venda?
6. Quantos aparelhos foram vendidos em um determinado tempo?
7. Qual vendedor vendeu mais no mês?
8. Quais aparelhos estão vinculados a quais planos?
9. Quantos planos cada cliente possui?

# Inclua instâncias de dados no projeto para testar as consultas acima descritas, utilizando linguagens de manipulação de dados (DML), especificamente, inserção (*insert*), remoção (*delete*) ou atualização (*update*).

# **Respostas abaixo:**

# **1)Implemente o projeto lógico de banco de dados via diagrama de entidades e relacionamentos (DER) que contemple um banco de dados relacional para o cenário acima descrito.**

# 

# **2) De posse do DER, determine o projeto físico do banco de dados utilizando linguagem de definição de dados (DDL), especificamente, as declarações de criação (*create*), remoção (*drop*) ou alteração (*alter*).**

# SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

# SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

# SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

# DROP SCHEMA IF EXISTS `mydb` ;

# CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

# USE `mydb` ;

# CREATE TABLE IF NOT EXISTS `mydb`.`Clientes` (

# `idCliente` INT NOT NULL AUTO\_INCREMENT,

# `nomeCliente` VARCHAR(100) NOT NULL,

# `nascCliente` DATE NOT NULL,

# `cpfCliente` VARCHAR(11) NOT NULL,

# PRIMARY KEY (`idCliente`),

# UNIQUE INDEX `cpfCliente\_UNIQUE` (`cpfCliente` ASC) VISIBLE)

# ENGINE = InnoDB;

# CREATE TABLE IF NOT EXISTS `mydb`.`Planos` (

# `idPlano` INT NOT NULL AUTO\_INCREMENT,

# `descricaoPlano` VARCHAR(100) NOT NULL,

# `nomePlano` VARCHAR(100) NOT NULL,

# PRIMARY KEY (`idPlano`))

# ENGINE = InnoDB;

# CREATE TABLE IF NOT EXISTS `mydb`.`Aparelhos` (

# `idAparelho` INT NOT NULL AUTO\_INCREMENT,

# `numeroAparelho` INT NOT NULL,

# `precoAparelho` FLOAT NOT NULL,

# `Planos\_idPlano` INT NOT NULL,

# PRIMARY KEY (`idAparelho`),

# UNIQUE INDEX `numeroAparelho\_UNIQUE` (`numeroAparelho` ASC) VISIBLE,

# INDEX `fk\_Aparelhos\_Planos1\_idx` (`Planos\_idPlano` ASC) VISIBLE,

# CONSTRAINT `fk\_Aparelhos\_Planos1`

# FOREIGN KEY (`Planos\_idPlano`)

# REFERENCES `mydb`.`Planos` (`idPlano`)

# ON DELETE NO ACTION

# ON UPDATE NO ACTION)

# ENGINE = InnoDB;

# CREATE TABLE IF NOT EXISTS `mydb`.`Vendedor` (

# `idVendedor` INT NOT NULL AUTO\_INCREMENT,

# `nomeVendedor` VARCHAR(100) NOT NULL,

# `comissaoVendedor` FLOAT NOT NULL DEFAULT 0,

# PRIMARY KEY (`idVendedor`))

# ENGINE = InnoDB;

# CREATE TABLE IF NOT EXISTS `mydb`.`Vendas` (

# `idVenda` INT NOT NULL AUTO\_INCREMENT,

# `idVendedor` INT NOT NULL,

# `idAparelho` INT NOT NULL,

# `idCliente` INT NOT NULL,

# `qtdVenda` INT NOT NULL,

# `dataVenda` DATE NOT NULL,

# PRIMARY KEY (`idVenda`),

# INDEX `fk\_Vendas\_Vendedor\_idx` (`idVendedor` ASC) VISIBLE,

# INDEX `fk\_Vendas\_Aparelhos\_idx` (`idAparelho` ASC) VISIBLE,

# INDEX `fk\_Vendas\_Clientes\_idx` (`idCliente` ASC) VISIBLE,

# CONSTRAINT `fk\_Vendas\_Vendedor`

# FOREIGN KEY (`idVendedor`)

# REFERENCES `mydb`.`Vendedor` (`idVendedor`)

# ON DELETE NO ACTION

# ON UPDATE NO ACTION,

# CONSTRAINT `fk\_Vendas\_Aparelhos`

# FOREIGN KEY (`idAparelho`)

# REFERENCES `mydb`.`Aparelhos` (`idAparelho`)

# ON DELETE NO ACTION

# ON UPDATE NO ACTION,

# CONSTRAINT `fk\_Vendas\_Clientes`

# FOREIGN KEY (`idCliente`)

# REFERENCES `mydb`.`Clientes` (`idCliente`)

# ON DELETE NO ACTION

# ON UPDATE NO ACTION)

# ENGINE = InnoDB;

# SET SQL\_MODE=@OLD\_SQL\_MODE;

# SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

# SET [UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS](mailto:UNIQUE_CHECKS%3D@OLD_UNIQUE_CHECKS);

# **3) Responda os questionamentos abaixo, transformando-os para linguagem de manipulação de dados, especificamente, consultas SQL.**

# **a) Quais clientes compraram em um determinado período?**

# SELECT Clientes.nomeCliente FROM Cliente JOIN PlanoCliente ON Cliente.idCliente = PlanoCliente.Cliente\_idCliente JOIN Aparelhos ON PlanoCliente.Plano\_idPlano = Aparelho.idAparelho WHERE Aparelho.DataCompra BETWEEN '2010-01-01' AND '2023-01-01';

# **b) Quais aparelhos estão disponíveis à venda?**

# SELECT \* FROM Aparelho;

# **c) Quantos aparelhos foram vendidos em um determinado tempo?**

# SELECT COUNT(\*) AS Quantidade FROM Aparelho WHERE DataCompra BETWEEN '2010-01-01' AND '2023-01-01;

# **d) Qual vendedor vendeu mais no mês?**

# SELECT Vendedor.nomeVendedor, COUNT(\*) AS TotalVendas FROM Vendedor JOIN Aparelhos ON Vendedor.idVendedor = Aparelho.Vendedor\_idVendedor WHERE MONTH(Aparelho.DataCompra) = 5 AND YEAR(Aparelho.DataCompra) = 2023 GROUP BY Vendedor.nomeVendedor ORDER BY TotalVendas DESC LIMIT 1;

# **e) Quais aparelhos estão vinculados a quais planos?**

# SELECT Aparelhos.numeroAparelho, Plano.descricaoPlano FROM Aparelhos LEFT JOIN Planos ON Aparelho.Plano\_idPlano = Plano.idPlano;

# **f) Quantos planos cada cliente possui?**

# SELECT Clientes.nomeCliente, COUNT(\*) AS TotalPlanos

# FROM Clientes

# JOIN PlanoClientes ON Cliente.idCliente = PlanoCliente.Cliente\_idCliente

# GROUP BY Clientes.nomeCliente;

# **4) Inclua instâncias de dados no projeto para testar as consultas acima descritas, utilizando linguagens de manipulação de dados (DML), especificamente, inserção (*insert*), remoção (*delete*) ou atualização (*update*).**

# - SCRIPT DE INSERÇÃO DOS PLANOS -

# insert into Planos (descricaoPlano,nomePlano) values ('100GB de internet por 1 mes','Plano Premium');

# insert into Planos (descricaoPlano,nomePlano) values ('10GB de internet por 1 mes','Plano Platinum');

# insert into Planos (descricaoPlano,nomePlano) values ('10MB de internet e Chamadas Gratuitas por 1 mes','Plano Pré');

# - SCRIPT DE INSERÇÃO DE VENDEDORES -

# insert into Vendedor (nomeVendedor,comissaoVendedor) values ('Joao dos Santos Neto',0.0);

# insert into Vendedor (nomeVendedor,comissaoVendedor) values ('Jamile Jovita da Silva',0.0);

# insert into Vendedor (nomeVendedor,comissaoVendedor) values ('Marcos Soares Silva',0.0);

# - SCRIPT DE INSERÇÃO DOS CLIENTES -

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values ('Arthur da Silva','2001-07-10','08475886380');

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values ('Guilherme Ribeiro ','2002-11-10','06394535823');

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values ('Ismael Rodrigues','2000-05-8','12345678909');

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values ('Francisco Jose ','1999-09-1','98765428100');

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values ('Fernanda Machado ','1989-12-4','11122233344');

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values ('Atlas Corigan ', '2003-05-03','55566677788');

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values ('Icaro Borges ', ' 2005-07-1','99988877766');

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values (' Iorana Nunes', ' 1999-01-9','98309843676');

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values ('Carolina Freitas ','2007-11-5','55394590823');

# insert into Clientes (nomeCliente,nascCliente,cpfCliente) values ('Jamile Silva ','2004-1-7','57223915373');

# - SCRIPT DE INSERÇÃO DOS APARELHOS -

# insert into Aparelhos (numeroAparelho,precoAparelho,Planos\_idPlano) values (100,1750.78,1);

# insert into Aparelhos (numeroAparelho,precoAparelho,Planos\_idPlano) values (101,2750.38,2);

# insert into Aparelhos (numeroAparelho,precoAparelho,Planos\_idPlano) values (102,1500.00,3);

# insert into Aparelhos (numeroAparelho,precoAparelho,Planos\_idPlano) values (103,1679.60,2);

# insert into Aparelhos (numeroAparelho,precoAparelho,Planos\_idPlano) values (104,6789.90,1);

# insert into Aparelhos (numeroAparelho,precoAparelho,Planos\_idPlano) values (105,8000.00,1);

# insert into Aparelhos (numeroAparelho,precoAparelho,Planos\_idPlano) values (110,1750.78,2);

# insert into Aparelhos (numeroAparelho,precoAparelho,Planos\_idPlano) values (190,2356.00,2);

# insert into Aparelhos (numeroAparelho,precoAparelho,Planos\_idPlano) values (198,1500.55,3);

# - SCRIPT DE INSERÇÃO DAS VENDAS -

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (1,1,1,1,'2020-07-10');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (1,1,1,2,'2020-07-10');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (1,2,2,1,'2021-10-09');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (2,3,3,4,'2010-11-01');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (2,4,4,6,'2014-12-07');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (1,5,5,1,'2020-08-26');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (3,1,6,2,'2020-09-29');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (3,1,7,3,'2021-02-22');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (2,8,8,5,'2022-02-19');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (3,9,9,4,'2023-06-17');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (2,9,10,1,'2017-09-15');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (3,6,2,1,'2019-03-22');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (2,5,3,1,'2018-04-21');

# insert into Vendas (idVendedor, idAparelho, idCliente, qtdVenda, dataVenda) values (1,7,9,1,'20218-01-01');