

Universidade da Amazônia
Curso: Análise e Des. de Sistemas
Turma: ALC 040 01 03 NMA
Disciplina: Banco de Dados
Professor: Alan Souza

EXERCÍCIO PRÁTICO

Suponha que você foi contratado(a) para participar da construção de um sistema que realiza a venda de ingressos para o Theatro do Paz. Levando em consideração os requisitos abaixo, construa o que se pede em seguida.

Requisito #1: um cliente precisa informar nome, CPF, email e senha para realizar o cadastro no site e pode comprar vários ingressos.

Requisito #2: o pagamento do ingresso só pode ser realizado pelo site (crédito, débito ou boleto) e, quando ele for impresso, as seguintes informações precisam aparecer no ingresso: nome do espetáculo, data e hora do espetáculo, nome de quem comprou, número da cadeira da plateia e se é meia entrada ou não.

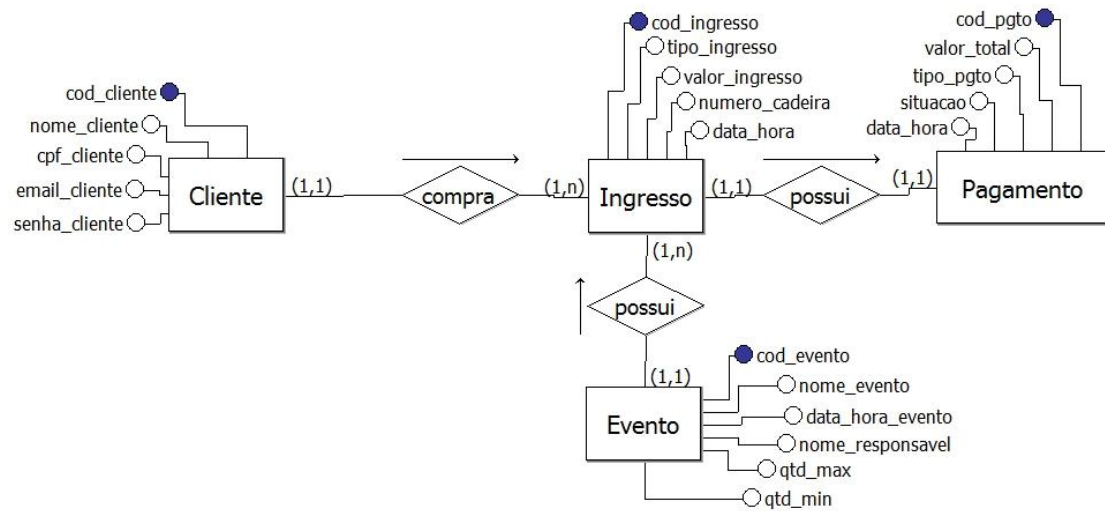
Requisito #3: um espetáculo deve possuir um nome, a data de cadastro, a data e hora que ele vai acontecer, o nome e telefone do responsável pelo espetáculo, a quantidade mínima e máxima de pessoas que poderão comparecer.

Tarefas:

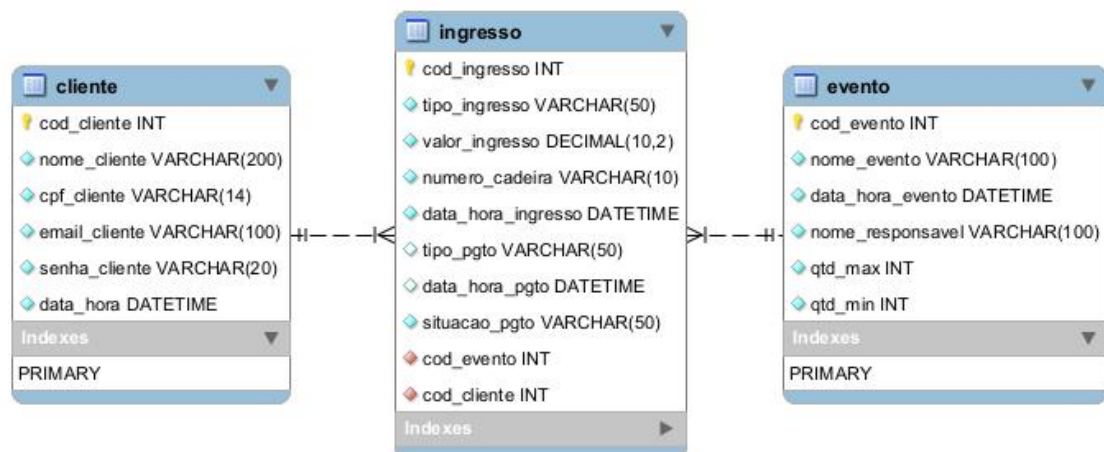
1. Criar o modelo conceitual de acordo com os requisitos e o que mais você achar coerente em relação ao minimundo (gerenciamento de ingressos de teatro). Esse modelo deve conter, no mínimo, quatro entidades relacionadas e cinco atributos em cada entidade. Você pode usar o BR Modelo ou papel e caneta pra fazer o modelo conceitual;
2. Converter o modelo conceitual em modelo lógico, usando o MySQL Workbench ou papel e caneta para criar o modelo lógico;
3. Converter o modelo lógico em modelo físico, usando o MySQL Workbench e comandos DDL do SQL;
4. Usar comandos DML do SQL para inserir, atualizar, consultar e remover dados do banco de dados. Descreva cada comando DML utilizado com um comentário antes dele.

RESPOSTAS:

1) Modelo conceitual:



2) Modelo lógico:



3) Modelo físico (SQL - DDL):

-- 1) Criar e usar o banco de dados teatro_bd:

```
CREATE DATABASE `teatro_bd` DEFAULT CHARACTER SET utf8 ;
USE `teatro_bd` ;
```

-- 2) Criar a tabela cliente:

```
CREATE TABLE IF NOT EXISTS `cliente` (
  `cod_cliente` INT NOT NULL,
  `nome_cliente` VARCHAR(200) NOT NULL,
  `cpf_cliente` VARCHAR(14) NOT NULL,
  `email_cliente` VARCHAR(100) NOT NULL,
  `senha_cliente` VARCHAR(50) NOT NULL,
  `data_hora` DATETIME NOT NULL,
  PRIMARY KEY (`cod_cliente`))
ENGINE = InnoDB;
```

-- 3) Criar a tabela evento:

```
CREATE TABLE IF NOT EXISTS `evento` (  
  `cod_evento` INT NOT NULL,  
  `nome_evento` VARCHAR(100) NOT NULL,  
  `data_hora_evento` DATETIME NOT NULL,  
  `nome_responsavel` VARCHAR(100) NOT NULL,  
  `qtd_max` INT NOT NULL,  
  `qtd_min` INT NOT NULL,  
  PRIMARY KEY (`cod_evento`))  
ENGINE = InnoDB;
```

-- 3) Criar a tabela ingresso:

```
CREATE TABLE IF NOT EXISTS `ingresso` (  
  `cod_ingresso` INT NOT NULL,  
  `tipo_ingresso` VARCHAR(50) NOT NULL,  
  `valor_ingresso` DECIMAL(10,2) NOT NULL,  
  `numero_cadeira` VARCHAR(10) NOT NULL,  
  `data_hora_ingresso` DATETIME NOT NULL,  
  `tipo_pgto` VARCHAR(50) NULL,  
  `data_hora_pgto` DATETIME NULL,  
  `situacao_pgto` VARCHAR(50) NOT NULL,  
  `cod_evento` INT NOT NULL,  
  `cod_cliente` INT NOT NULL,  
  PRIMARY KEY (`cod_ingresso`),  
  INDEX `fk_evento` (`cod_evento` ASC),  
  INDEX `fk_cliente` (`cod_cliente` ASC),  
  CONSTRAINT `fk_evento`  
    FOREIGN KEY (`cod_evento`)  
    REFERENCES `evento` (`cod_evento`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_cliente`  
    FOREIGN KEY (`cod_cliente`)  
    REFERENCES `cliente` (`cod_cliente`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

SQL - DML:

-- 1) Inserir dados na tabela cliente:

```
INSERT INTO cliente  
(cod_cliente, nome_cliente, cpf_cliente, email_cliente, senha_cliente, data_hora)  
VALUES  
(1, 'Maria Silva', '882.754.902-90', 'maria@email.com', md5('123456'), now()),  
(2, 'José Souza', '321.543.766-01', 'jose@email.com', md5('jose123'), now()),  
(3, 'João Oliveira', '123.312.542-31', 'joao@email.com', md5('joli321'), now());
```

-- 2) Inserir dados na tabela evento:

```
INSERT INTO evento
(cod_evento, nome_evento, data_hora_evento, nome_responsavel, qtd_max, qtd_min)
VALUES
(1, 'Orquestra', now(), 'Mahler', 1000, 100),
(2, 'Standup Paraense', now(), 'Carla Braga', 1000, 50);
```

-- 3) Inserir dados na tabela ingresso:

```
INSERT INTO ingresso
(cod_ingresso, tipo_ingresso, valor_ingresso, numero_cadeira, data_hora_ingresso,
tipo_pgto, data_hora_pgto, situacao_pgto, cod_evento, cod_cliente)
VALUES
(1, 'Inteira', 80.00, 'A1', now(), 'Boleto', null, 'Aguardando', 1, 1),
(2, 'Inteira', 80.00, 'A2', now(), 'Boleto', null, 'Aguardando', 1, 2),
(3, 'Meia', 40.00, 'A7', '2020-03-03 19:00', 'Crédito', now(), 'Pago', 1, 3),
(4, 'Inteira', 35.00, 'A10', now(), 'Débito', now(), 'Pago', 2, 1),
(5, 'Meia', 17.50, 'B2', now(), 'Débito', now(), 'Pago', 2, 3);
```

-- 4) Consulta para selecionar todos os dados de todos os clientes ordenados por nome em ordem alfabética:

```
select * from cliente
order by nome_cliente;
```

-- 5) Consulta para selecionar todos os dados dos eventos:

```
select * from evento;
```

-- 6) Consulta para selecionar todos os dados dos ingressos pagos:

```
select * from ingresso
where situacao_pgto = 'Pago';
```

-- 7) Alterar a situação de pagamento dos ingressos de códigos 1 e 2 para Pago e a data e hora do pagamento para a data e hora atual:

```
update ingresso
set situacao_pgto = 'Pago', data_hora_pgto = now()
where cod_ingresso = 1 OR cod_ingresso = 2;
```