



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS VARGINHA

SISTEMAS DE INFORMAÇÃO
DEISYMAR B. TAVARES E PROF. WEIDER P. RODRIGUES

TRABALHO PRÁTICO DE BANCO DE DADOS

GERENCIAMENTO DE INGRESSOS PARA PARTIDAS DE
FUTEBOL

Christian Sales de Oliveira Assis

Gabriel Pereira de Oliveira

Pedro Lucca Caetano

Vinícius Castelhana Mantovani

VARGINHA - MG

2025

SUMÁRIO

1.	MINIMUNDO	3
2.	DIAGRAMA ENTIDADE RELACIONAMENTO (DER)	4
3.	MODELO ENTIDADE RELACIONAMENTO (MER)	5
4.	SCRIPT DE CRIAÇÃO	6
5.	SCRIPT DE INSERÇÃO	9
6.	CONSULTAS SQL	12
7.	GATILHO	15

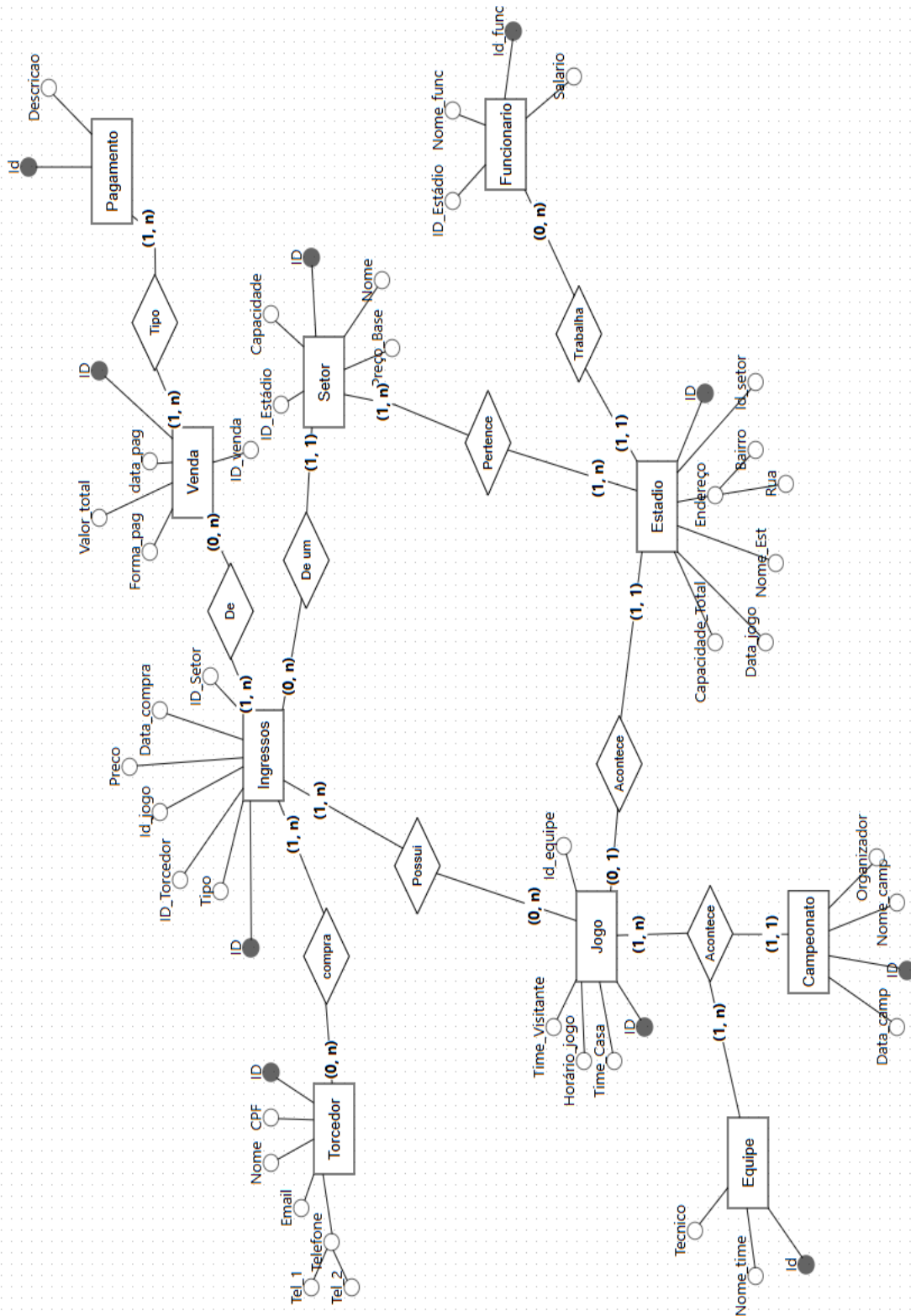
1. MINIMUNDO

O sistema gerencia a venda de ingressos para partidas de futebol realizadas em um estádio. Os torcedores podem comprar ingressos para diferentes jogos, que ocorrem em datas e horários específicos, envolvendo times distintos. Cada ingresso está vinculado a um torcedor(que possui identificação, nome, números de telefones e e-mails) e a um setor do estádio, que possui diferentes preços e capacidades. Os ingressos vendidos podem ser do tipo inteira ou meia-entrada, não sendo permitida a venda de ingressos por cortesia. As vendas são registradas com informações como forma de pagamento e valor total da compra.

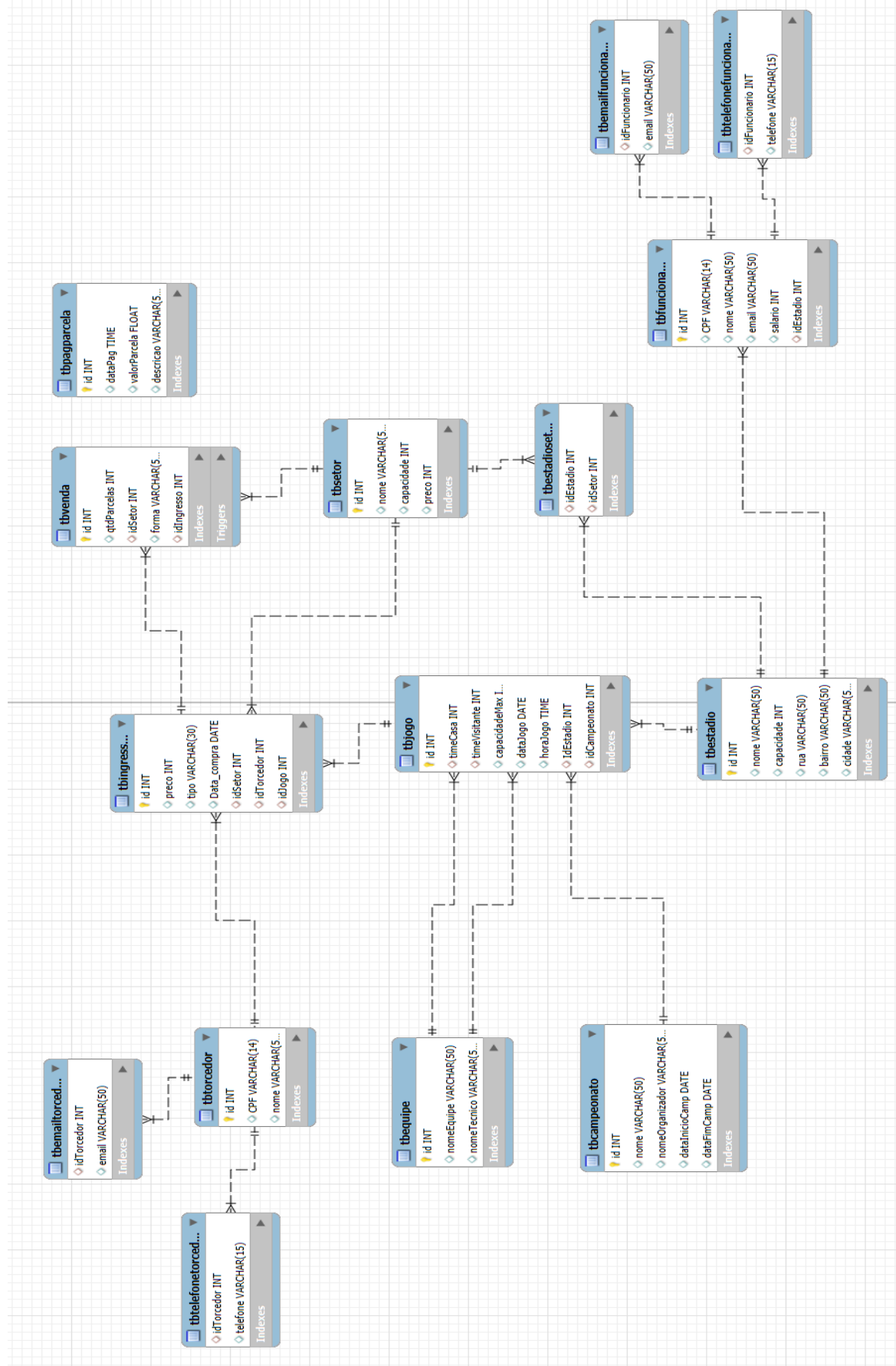
Cada jogo acontece em um estádio específico, que possui nome, endereço, capacidade total e diferentes setores. Os setores do estádio possuem um preço base e uma capacidade determinada. Os times que participam dos jogos são cadastrados no sistema com nome e técnico responsável. Os campeonatos registram os jogos que fazem parte da competição, contendo informações como nome, data e organizador.

O estádio também possui funcionários, que têm informações como nome, salário e identificação(CPF), além de números de telefone e um e-mail profissional.

2. DIAGRAMA ENTIDADE RELACIONAMENTO (DER)



3. MODELO ENTIDADE RELACIONAMENTO (MER)



4. SCRIPT DE CRIAÇÃO

Primeiramente, foram criadas as tabelas tbTorcedor, tbIngressos, tbJogo, tbCampeonato, tbEstadio, tbSetor, tbFuncionario, tbVenda e tbPagParcela de acordo com a descrição dada pelo mini-mundo e pela análise do DER. Além disso, foram criadas as tabelas necessárias para atributos multivalorados, como a tbEmailTorcedor, a tbTelefoneTorcedor, e a tbTelefoneFuncionario. Por fim, foi criada a tbEstadioSetor devido ao relacionamento muitos-para-muitos entre a tbEstadio e a tbSetor. Esta é portanto a primeira parte do script:

```
create table tbTorcedor(
    id int primary key auto_increment,
    CPF varchar(14) unique,
    nome varchar(50)
);

create table tbEmailTorcedor(
    idTorcedor int,
    email varchar(50),
    foreign key (idTorcedor) references tbTorcedor(id)
);

create table tbTelefoneTorcedor(
    idTorcedor int,
    telefone varchar(15),
    foreign key (idTorcedor) references tbTorcedor(id)
);

create table tbEquipe(
    id int primary key auto_increment,
    nomeEquipe varchar(50),
    nomeTecnico varchar(50)
);

create table tbEstadio(
    id int primary key auto_increment,
    nome varchar(50),
    capacidade int,
    rua varchar(50),
    bairro varchar(50),
    cidade varchar(50)
);

create table tbCampeonato(
    id int primary key auto_increment,
    nome varchar(50),
    nomeOrganizador varchar(50),
    dataInicioCamp date,
    dataFimCamp date
);
```

```
create table tbJogo(  
    id int primary key auto_increment,  
    timeCasa int,  
    timeVisitante int,  
    capacidadeMax int,  
    dataJogo date,  
    horaJogo time  
);
```

```
create table tbIngressos(  
    id int primary key auto_increment,  
    preco int,  
    tipo varchar(30),  
    Data_compra date  
);
```

```
create table tbSetor(  
    id int primary key auto_increment,  
    nome varchar(50),  
    capacidade int,  
    preco int  
);
```

```
create table tbVenda(  
    id int primary key auto_increment,  
    qtdParcelas int,  
    idSetor int,  
    forma varchar(50)  
);
```

```
create table tbPagParcela(  
    id int primary key auto_increment,  
    dataPag time,  
    valorParcela float,  
    descricao varchar(50)  
);
```

```
create table tbFuncionario(  
    id int primary key auto_increment,  
    CPF varchar(14) unique,  
    nome varchar(50),  
    email varchar(50),  
    salario int  
);
```

```
create table tbTelefoneFuncionario(  
    idFuncionario int,  
    telefone varchar(15),  
    foreign key (idFuncionario) references tbFuncionario(id)  
);
```

```
create table tbEstadioSetor(  
    idEstadio int,  
    idSetor int  
)ENGINE=InnoDB;
```

Após a criação das tabelas, através da operação alter table, foram adicionadas as colunas e as foreign keys necessárias para o relacionamento entre as tabelas. Eis a segunda parte do script:

```
alter table tbIngressos add column idSetor int,  
add column idTorcedor int, add column idJogo int,  
add foreign key (idSetor) references tbSetor(id),  
add foreign key (idTorcedor) references tbTorcedor(id),  
add foreign key (idJogo) references tbJogo(id);
```

```
alter table tbVenda add column idIngresso int,  
add foreign key (idIngresso) references tbIngressos(id);
```

```
alter table tbFuncionario add column idEstadio int,  
add foreign key (idEstadio) references tbEstadio(id);
```

```
alter table tbJogo add column IdEstadio int, add column idCampeonato int,  
add foreign key (idEstadio) references tbEstadio(id),  
add foreign key (idCampeonato) references tbCampeonato(id),  
add foreign key (timeCasa) references tbEquipe(id),  
add foreign key (timeVisitante) references tbEquipe(id);
```

```
alter table tbEstadioSetor add foreign key(idEstadio) references tbEstadio(id) on delete set null;  
alter table tbEstadioSetor add foreign key(idSetor) references tbSetor(id) on delete set null;
```

```
alter table tbVenda add foreign key (idSetor) references tbSetor(id);
```


5. SCRIPT DE INSERÇÃO

Após a parte de criação ser executada, são feitas as inserções nas tabelas tbEquipe, tbEstadio, tbCampeonato, tbTorcedor, tbEmailTorcedor, tbTelefoneTorcedor, tbFuncionario, tbTelefoneFuncionario, tbSetor, tbEstadioSetor, tbJogo, tbIngressos e tbVenda. Abaixo, está o script:

```
insert into tbEquipe(nomeEquipe, nomeTecnico) values
('São Paulo', 'Luis Zubeldía'),
('Corinthians', 'Ramón Díaz'),
('Santos', 'Pedro Caixinha'),
('Palmeiras', 'Abel Ferreira'),
('Botafogo', 'Artur Jorge'),
('Flamengo', 'Filipe Luís'),
('Fluminense', 'Mano Menezes'),
('Vasco da Gama', 'Felipe Maestro'),
('Grêmio', 'Gustavo Quinteros'),
('Internacional', 'Roger Machado'),
('Atlético-MG', 'Cuca'),
('Cruzeiro', 'Fernando Diniz'),
('Bahia', 'Rogerio Ceni');
```

```
insert into tbEstadio(nome, capacidade, rua, bairro, cidade) values
('Morumbi', 66795, 'Praça Roberto Gomes Pedrosa', 'Morumbi', 'São Paulo'),
('Maracanã', 78838, 'R. Prof. Eurico Rabelo', 'Maracanã', 'Rio de Janeiro'),
('Vila Belmiro', 16798, 'Rua Princesa Isabel', 'Vila Belmiro', 'Santos'),
('Nilton Santos', 44661, 'R. José dos Reis', 'Engenho de Dentro', 'Rio de Janeiro'),
('Mineirão', 61927, 'Av. Antônio Abrahão Caram', 'São José', 'Belo Horizonte'),
('Arena MRV', 46000, 'Rua Cristina Maria de Assis', 'Califórnia', 'Belo Horizonte'),
('Neo Química Arena', 47605, 'Av. Miguel Ignácio Curi', 'Vila Carmosina', 'São Paulo'),
('Beira-Rio', 50842, 'Av. Padre Cacique', 'Praia de Belas', 'Porto Alegre'),
('Arena do Grêmio', 55662, 'Av. Padre Leopoldo Brentano', 'Farrapos', 'Porto Alegre'),
('Arena Fonte Nova', 50025, 'Ladeira da Fonte das Pedras', 'Nazaré', 'Salvador');
```

```
insert into tbCampeonato(nome, nomeOrganizador, dataInicioCamp, dataFimCamp) values
('Libertadores', 'CONMEBOL', '2025-02-05', '2025-11-29'),
('Campeonato Brasileiro', 'CBF', '2025-03-29', '2025-12-21'),
('Copa do Brasil', 'CBF', '2025-02-19', '2025-11-09'),
('Sulamericana', 'CONMEBOL', '2025-03-05', '2025-11-22');
```

```
insert into tbTorcedor (CPF, nome) values
('395.821.467-32', 'Carlos Mendes'),
('718.264.905-78', 'Fernanda Souza'),
('562.109.843-21', 'Ricardo Müller'),
('804.376.291-55', 'Julia Alves'),
('139.528.760-99', 'Thiago Pereira'),
('627.914.382-44', 'Amanda Ribeiro'),
('250.671.839-60', 'Lucas Martins');
```

```
insert into tbEmailTorcedor(idTorcedor, email) values
```

```
(1, 'carlosmendes.santos66@yahoo.com.br'),
(1, 'carlosms66@gmail.com'),
(2, 'nandasouzapr1@hotmail.com'),
(3, 'ricardo.muller07@gmail.com'),
(4, 'jualves21.02@gmail.com'),
(4, 'julia.alvesmartins2005@gmail.com'),
(5, 'thiagope.200392@outlook.com'),
(5, 'thiago92alves.pereira@gmail.com'),
(6, 'amanda03ribeiro09@gmail.com'),
(7, 'lucas.martinslm2004@gmail.com');
```

insert into tbTelefoneTorcedor(idTorcedor, telefone) values

```
(1, '11983476291'),
(2, '21967584930'),
(2, '21998234756'),
(3, '31945672890'),
(4, '35923784567'),
(5, '51978456239'),
(5, '51993284756'),
(6, '61984567213'),
(7, '71992837465'),
(7, '71967382941');
```

insert into tbFuncionario(CPF, nome, email, salario, idEstadio) values

```
('251.785.496-33', 'Carlos Silva', 'carlos.silva77@gmail.com', 2000, 2),
('537.924.881-45', 'Ana Souza', 'ana.souza98@hotmail.com', 1650, 3),
('174.368.102-76', 'Marcos Takahashi', 'marcos.takahashi21@outlook.com', 2050, 1),
('684.120.935-59', 'Juliana Costa', 'juliana.costa35@outlook.com', 2270, 1),
('491.623.758-20', 'Roberto Almeida', 'roberto.almeida99@yahoo.com.br', 1550, 7),
('832.174.659-61', 'Rafaela Rossi', 'rafaela.rossi43@gmail.com', 3000, 8),
('920.546.831-87', 'Ricardo Lima', 'ricardo.lima04@outlook.com', 3120, 6),
('621.553.099-96', 'Carolina Fonseca', 'carolinafsc32@gmail.com', 1820, 10),
('759.976.140-34', 'Juliano Vasques', 'juliano.vasques93@outlook.com', 2130, 9);
```

insert into tbTelefoneFuncionario(idFuncionario, telefone) values

```
(1, '21981074604'),
(1, '24924987434'),
(2, '13975079206'),
(3, '11937372067'),
(4, '11968403911'),
(4, '11973585148'),
(5, '11902774665'),
(6, '51935034449'),
(7, '31932678866'),
(8, '71947345507'),
(9, '51981328182'),
(9, '51991246719');
```

insert into tbSetor (id, nome, capacidade, preco) values

```
(1, 'Arquibancada Inferior', 20000, 50),
(2, 'Arquibancada Superior', 15000, 40),
```

```
(3, 'Camarote VIP', 5000, 300.00),  
(4, 'Arquibancada Leste', 25000, 60),  
(5, 'Arquibancada Oeste', 25000, 60),  
(6, 'Cadeiras Cobertas', 20000, 90),  
(7, 'Camarote Especial', 5000, 400);
```

```
insert into tbEstadioSetor (idEstadio, idSetor) values
```

```
(1, 1), (1, 2), (1, 3), (1, 6), (1, 7),  
(2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7),  
(3, 1), (3, 2), (3, 6),  
(4, 1), (4, 2), (4, 3), (4, 5), (4, 7),  
(5, 1), (5, 2), (5, 4), (5, 6), (5, 7),  
(6, 1), (6, 2), (6, 3), (6, 5), (6, 7),  
(7, 1), (7, 2), (7, 4), (7, 6), (7, 7),  
(8, 1), (8, 3), (8, 5), (8, 6),  
(9, 1), (9, 2), (9, 3), (9, 4), (9, 6),  
(10, 1), (10, 2), (10, 4), (10, 5), (10, 6), (10, 7);
```

```
insert into tbJogo (idEstadio, timeCasa, timeVisitante, capacidadeMax, dataJogo, horaJogo,  
idCampeonato) values
```

```
(1, 1, 2, 50000, '2025-03-10', '19:00:00',1),  
(3, 3, 4, 45000, '2025-03-11', '20:30:00',1),  
(4, 5, 6, 60000, '2025-03-12', '21:00:00',2),  
(2, 7, 8, 48000, '2025-03-13', '18:00:00',3),  
(8, 9, 10, 55000, '2025-03-14', '19:30:00',2),  
(6, 11, 12, 53000, '2025-03-15', '20:00:00',2),  
(10, 13, 1, 40000, '2025-03-16', '21:45:00',3),  
(7, 2, 3, 47000, '2025-03-17', '18:30:00',1),  
(5, 4, 5, 49000, '2025-03-18', '19:15:00',4),  
(2, 6, 7, 62000, '2025-03-19', '20:45:00',4);
```

```
insert into tbIngressos(tipo, Data_compra, idJogo, idTorcedor, idSetor) values
```

```
('Inteira', '2025-03-01', 1, 3, 3),  
( 'Meia-entrada', '2025-03-03', 4, 1, 2),  
( 'Inteira', '2025-03-10', 7, 4, 4),  
( 'Inteira', '2025-02-28', 1, 1, 2),  
( 'Meia-entrada', '2025-03-12', 8, 6, 5);
```

```
insert into tbVenda(idIngresso, forma, qtdParcelas, idSetor) values
```

```
(1, 'Crédito',2,1),  
(2, 'Crédito',1,3),  
(3, 'Débito',5,2),  
(4, 'Pix',1,3),  
(5, 'Débito',2,1);
```

6. CONSULTAS SQL

Em seguida, os scripts para as consultas, de acordo com as instruções que foram dadas previamente, todos tendo sua relevância de acordo com o contexto dado pelo mini-mundo:

- I. Elabore e resolva 2 consultas simples (SELECT ... FROM ... WHERE..) que envolva 3 ou mais tabelas. Use o ORDER BY.

1.

Mostra todos os jogos, suas equipes participantes, o nome do estádio, do campeonato e data e horário

```
select j.id, eC.nomeEquipe as 'Time Casa', eV.nomeEquipe as 'Time Visitante', es.nome as 'Estádio', c.nome as 'Campeonato',  
j.dataJogo as 'Data', j.horaJogo 'Horário'  
from tbJogo j inner join tbEquipe eC on j.timeCasa = eC.id  
inner join tbEquipe eV on j.timeVisitante = eV.id  
inner join tbEstadio es on j.idEstadio = es.id  
inner join tbCampeonato c on j.idCampeonato = c.id  
order by j.dataJogo desc;
```

2.

Mostrar qual foi o último jogo de determinada equipe e em qual estádio seja como time casa ou time visitante

```
select e.id as idEquipe, e.nomeEquipe, e.nomeTecnico, j.dataJogo as 'Última Partida', es.nome as 'Estádio'  
from tbJogo j inner join tbEquipe e on e.id = j.timeCasa or e.id = j.timeVisitante  
inner join tbEstadio es on es.id = j.idEstadio  
where j.dataJogo = (select max(dataJogo) from tbJogo j where e.id = j.timeCasa or e.id = j.timeVisitante)  
group by j.dataJogo, e.id, e.nomeEquipe, e.nomeTecnico, es.nome order by j.dataJogo desc;
```

- II. Elabore e resolva uma consulta que necessite do DISTINCT.

Mostra em quantos estádios diferentes cada time vai jogar

```
select e.nomeEquipe, count(distinct j.idEstadio) as totalEstadios  
from tbJogo j  
join tbEquipe e on j.timeCasa = e.id or j.timeVisitante = e.id  
group by e.nomeEquipe  
order by totalEstadios desc;
```

- III. Elabore e resolva uma consulta que use alguma função agregada (MAX, MIN, COUNT, AVG, SUM) com Group by.

Mostra o nome, id, email e salário, junto com o estádio que estão trabalhando os

colaboradores (Que recebem mais que a média dos trabalhadores)

```
select f.id, f.nome, f.email, f.salario, e.nome as Estadio from tbFuncionario f
inner join tbEstadio e on e.id = f.idEstadio
where f.idEstadio not in(8, 9, 10)
and f.salario > (select avg(salario) from tbFuncionario);
```

Além dessa consulta, a consulta do item V utiliza a função COUNT, e a consulta do item VIII usa a função MAX.

IV. Elabore e resolva uma consulta que precise do operador IN (ou NOT IN).

Na consulta anterior foi utilizado not in; na consulta do item VII, foi usado in.

V. Elabore e resolva uma consulta que precise do HAVING.

Mostra quantos jogos um estádio sediará por campeonato

```
select e.nome as "Nome Estadio", count(j.idEstadio) as "Quantidade de jogos" from tbJogo j
inner join tbCampeonato c on j.idCampeonato = c.id
inner join tbEstadio e on j.idEstadio = e.id
where c.nome = "Libertadores"
group by e.nome
having count(j.idEstadio > 0);
```

VI. Elabore e resolva uma consulta que precise do LEFT JOIN.

Mostra o id, o CPF e nome dos torcedores que não compraram nenhum ingresso

```
select t.id, t.CPF, t.nome from tbTorcedor t
left join tbIngressos i on t.id = i.idTorcedor where i.idTorcedor is null
order by t.id;
```

VII. Elabore e resolva uma consulta que utilize o UNION ou INTERSECT ou MINUS (not in).

Mostra quais times irão se enfrentar baseado no id de um jogo

```
select e.nomeEquipe as "Jogo" from tbEquipe e where e.id in(select timeCasa from tbJogo where
id = 1)
union
select e.nomeEquipe from tbEquipe e where e.id in(select timeVisitante from tbJogo where id =
1);
```

VIII. Elabore e resolva uma consulta que utilize subselect

Mostrar qual foi o último jogo de determinada equipe e em qual estádio seja como time casa ou time visitante

```
select e.id as idEquipe, e.nomeEquipe, e.nomeTecnico, j.dataJogo as 'Última Partida', es.nome as 'Estádio'
from tbJogo j inner join tbEquipe e on e.id = j.timeCasa or e.id = j.timeVisitante
inner join tbEstadio es on es.id = j.idEstadio
where j.dataJogo = (select max(dataJogo) from tbJogo j where e.id = j.timeCasa or e.id = j.timeVisitante)
group by j.dataJogo, e.id, e.nomeEquipe, e.nomeTecnico, es.nome order by j.dataJogo desc;
```

O item seguinte também utiliza subselect.

IX. Elabore e resolva uma consulta que utilize o ALL ou o SOME.

Mostra os estádios que tem a capacidade maior que pelo menos um estádio em que jogou o Corinthians

```
select id, nome from tbEstadio
where capacidade > some
(select es.capacidade from tbJogo j
inner join tbEstadio es on j.idEstadio = es.id
where j.timeCasa = 2 or j.timeVisitante = 2);
```

7. GATILHO

Foi necessária a criação de um gatilho para gerar automaticamente as parcelas do pagamento para a venda dos ingressos. O gatilho busca através de um select o preço do setor, evita através de um if, a divisão por zero (seria o caso da compra feita à vista), e atribui o valor da parcela como a razão entre o valor total e a quantidade de parcelas. Após isso é feito um loop para adicionar todas as parcelas a tabela tbPagParcela. A cada insert na tabela tbVenda, o gatilho é acionado.

```
DELIMITER $$
```

```
CREATE TRIGGER trg_pos_venda
AFTER INSERT ON tbVenda
FOR EACH ROW
BEGIN
    DECLARE parcela_valor DECIMAL(10, 2);
    DECLARE i INT DEFAULT 1;
    DECLARE valor_total DECIMAL(10,2);

    -- Buscar o preço do setor correspondente à venda
    SELECT s.preco
    INTO valor_total
    FROM tbSetor s
    WHERE s.id = NEW.idSetor
    LIMIT 1;

    -- Evitar divisão por zero
    IF NEW.qtdParcelas > 0 THEN
        SET parcela_valor = valor_total / NEW.qtdParcelas;

        -- Loop para inserir parcelas
        WHILE i <= NEW.qtdParcelas DO
            INSERT INTO tbPagParcela (dataPag, descricao, valorParcela)
            VALUES (CURDATE(), CONCAT('Parcela ', i, ' de ', NEW.qtdParcelas), parcela_valor);

            SET i = i + 1;
        END WHILE;
    END IF;
END $$

DELIMITER ;
```