

A decorative graphic on the left side of the slide, consisting of a network of orange lines and circles that resemble a circuit board or a data network. The lines are of varying thicknesses and the circles are of varying sizes, creating a complex, branching pattern that extends from the top to the bottom of the slide.

PROJETO DE BANCO DE DADOS

AVALIAÇÃO 2

CENÁRIO

O banco de dados que modelamos está relacionado ao futebol, com uma organização estruturada, diversas áreas funcionais, como jogadores, treinadores, partidas, arbitragem, e eventos relacionados a treinos e lesões.



DESCRIÇÃO DO CENÁRIO

- **Clube:** Cada clube de futebol tem um nome, data de fundação e um estádio. O clube contrata jogadores e um treinador.
- **Jogadores:** Os jogadores têm nome, posição, data de nascimento e estão vinculados a um clube. Eles participam de partidas e treinamentos, e podem sofrer lesões.
- **Treinador:** Cada clube tem um treinador que é responsável por treinar os jogadores e liderar o time nas partidas.
- **Partidas:** Os clubes jogam partidas contra outros times. As partidas registram os gols de cada time e são supervisionadas por um árbitro.
- **Arbitragem:** Um árbitro supervisiona cada partida, garantindo que as regras sejam seguidas.
- **Escalação:** Antes de cada partida, os jogadores são escalados, ou seja, escolhidos para jogar, com suas posições definidas.
- **Lesões:** Se um jogador se machuca, a lesão é registrada, incluindo a data e o tipo de lesão.
- **Treinamentos:** Os clubes organizam treinamentos regulares para os jogadores. A participação de cada jogador é registrada.
- **Contratos:** Cada jogador tem um contrato com o clube, que define o período de trabalho e o salário.

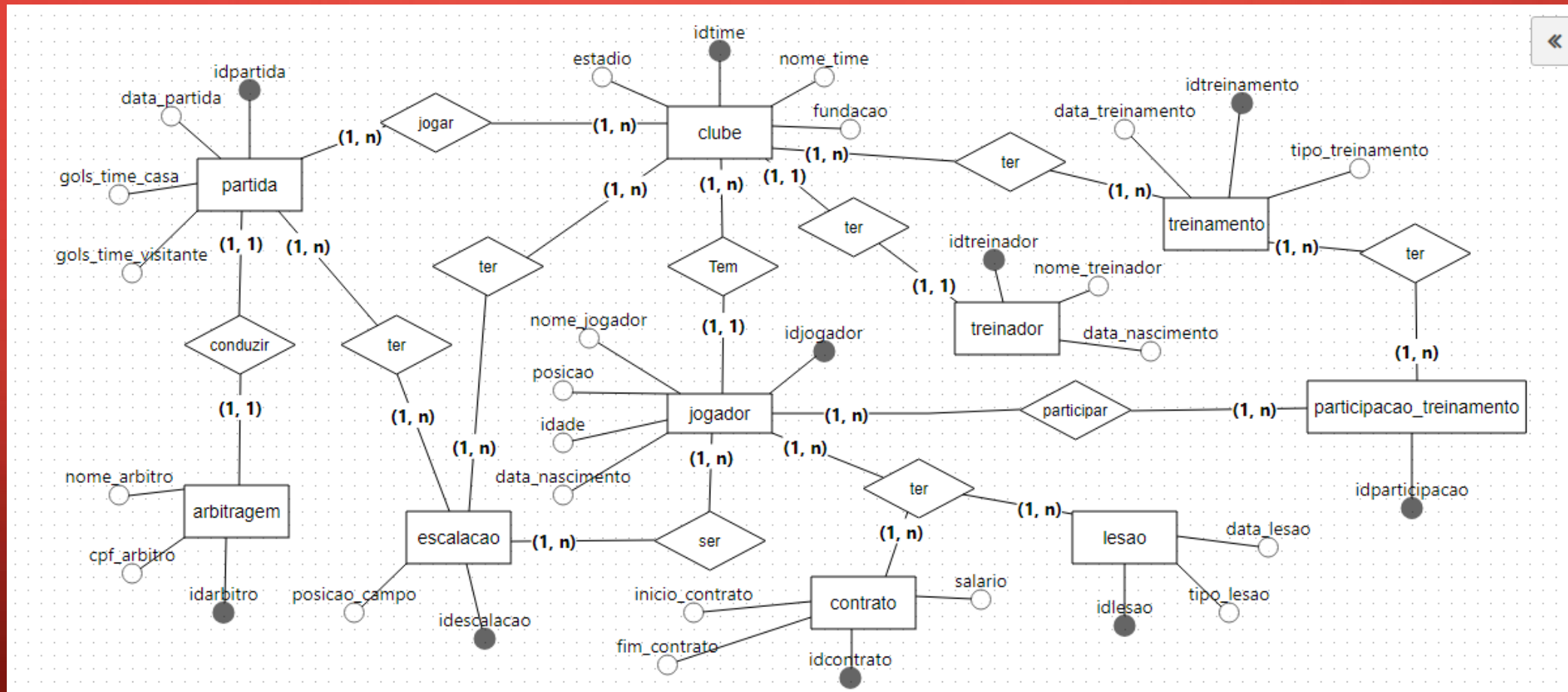
MODELO CONCEITUAL

O **Modelo Conceitual** é a representação abstrata e de alto nível das principais entidades do sistema e de seus relacionamentos. Neste cenário, o modelo conceitual foca em identificar e definir as entidades e os relacionamentos principais (Clube, Jogador, Treinador, etc.).

- **Relacionamentos:**

- **Clube e Jogador:** Um clube pode ter muitos jogadores, mas cada jogador está associado a um único clube.
- **Clube e Treinador:** Cada clube tem um treinador, e um treinador só pode estar associado a um clube por vez.
- **Clube e Partida:** Um clube participa de várias partidas como time da casa ou visitante.
- **Partida e Arbitragem:** Cada partida tem um árbitro designado.
- **Partida e Escalação:** Uma partida envolve uma escalação de jogadores, e cada jogador participa em uma determinada posição.
- **Jogador e Lesão:** Um jogador pode ter várias lesões ao longo do tempo.
- **Treinamento e Participacao_Treinamento:** Os jogadores participam de vários treinamentos, e suas participações são registradas.
- **Jogador e Contrato:** Cada jogador tem um contrato que define sua relação formal com o clube.

MODELO CONCEITUAL



MODELO LÓGICO

O **Modelo Lógico** é a forma como os dados são organizados em um banco de dados relacional. Ele define as tabelas, chaves primárias e estrangeiras, e os tipos de dados de cada coluna.

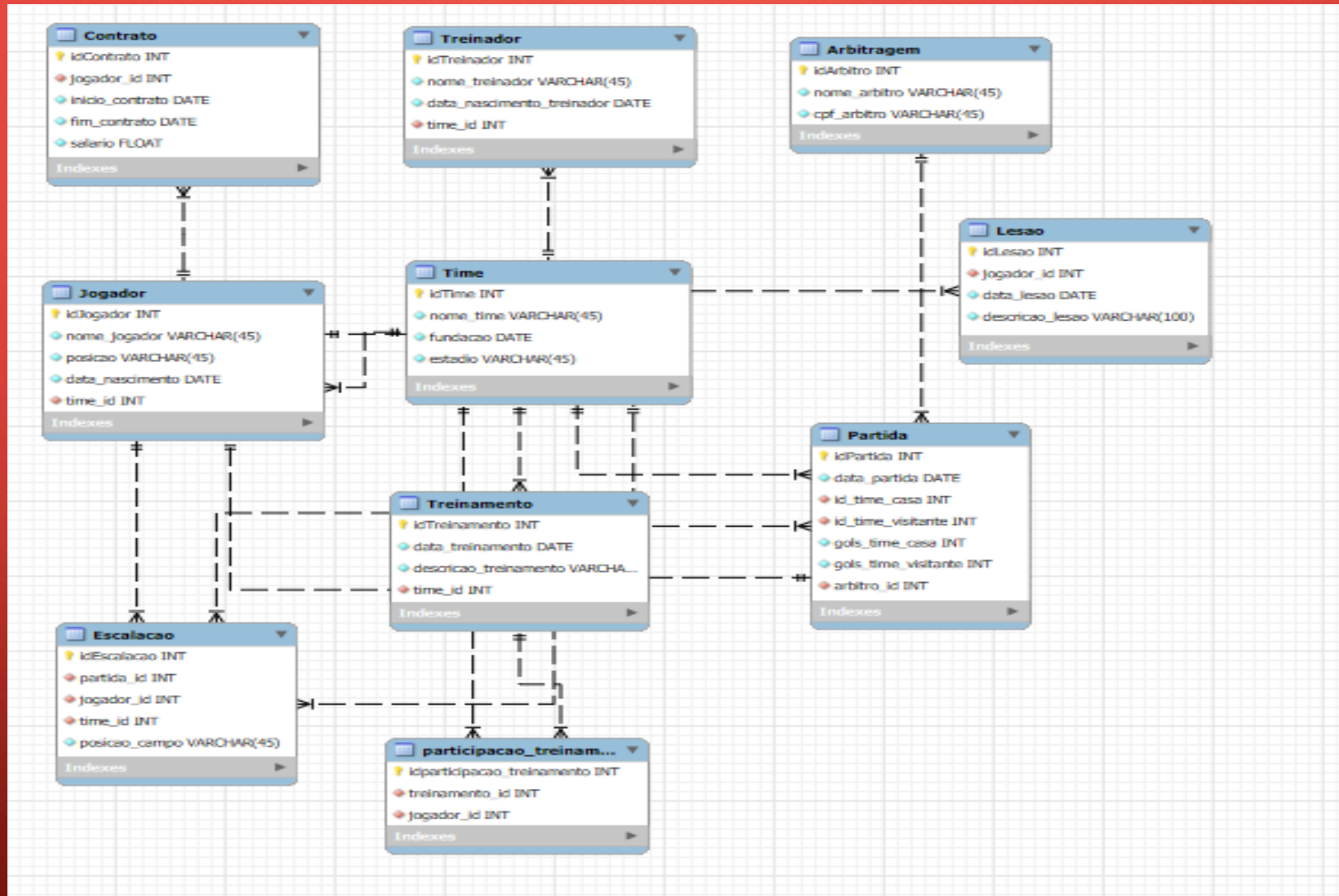
Principais Tabelas:

1. **Clube:** Armazena informações sobre os times, como idtime (chave primária), nome do time, data de fundação e estádio.
2. **Jogador:** Contém dados dos jogadores, incluindo idjogador (chave primária), nome, posição, data de nascimento e time_id (referenciando a tabela Clube).
3. **Treinador:** Guarda informações sobre os treinadores, com idtreinador (chave primária) e time_id (referenciando a tabela Clube).
4. **Partida:** Registra os detalhes das partidas, como idpartida (chave primária), times que jogaram (id_time_casa, id_time_visitante), gols e arbitro_id (referenciando Arbitragem).
5. **Arbitragem:** Armazena os árbitros, com idarbitro (chave primária), nome e CPF.
6. **Escalação:** Relaciona jogadores às partidas, com idescalacao (chave primária), partida_id, jogador_id, e time_id.
7. **Lesão:** Registra as lesões dos jogadores, com idlesao (chave primária) e jogador_id (referenciando Jogador).
8. **Treinamento:** Armazena informações sobre os treinos, com idtreinamento (chave primária) e time_id (referenciando Clube).
9. **Participação no Treinamento:** Registra quais jogadores participaram de quais treinos, com idparticipacao (chave primária), treinamento_id, e jogador_id.
10. **Contrato:** Armazena os contratos dos jogadores, com idcontrato (chave primária), jogador_id (referenciando Jogador), duração e salário.

Chaves Estrangeiras:

As chaves estrangeiras (time_id, arbitro_id, etc.) conectam as tabelas entre si, garantindo que os dados estejam relacionados de forma consistente.

MODELO LÓGICO



QUERY'S DE CRIAÇÃO DE TABELA

```
CREATE TABLE `futebol_db`.`clube` (  
  `idTime` INT NOT NULL AUTO_INCREMENT,  
  `nome_time` VARCHAR(45) NOT NULL,  
  `fundacao` DATE NOT NULL,  
  `estadio` VARCHAR(100) NOT NULL,  
  PRIMARY KEY (`idTime`));
```

```
CREATE TABLE `futebol_db`.`arbitragem` (  
  `idarbitragem` INT NOT NULL AUTO_INCREMENT,  
  `nome_arbitro` VARCHAR(45) NOT NULL,  
  `cpf_arbitro` VARCHAR(12) NOT NULL,  
  PRIMARY KEY (`idarbitragem`));
```

```
CREATE TABLE `futebol_db`.`treinador` (  
  `idtreinador` INT NOT NULL AUTO_INCREMENT,  
  `nome_treinador` VARCHAR(45) NOT NULL,  
  `data_nascimento` DATE NOT NULL,  
  `time_id` INT NOT NULL,  
  PRIMARY KEY (`idtreinador`),  
  INDEX `fk_time_id_idx` (`time_id` ASC) VISIBLE,  
  CONSTRAINT `fk_time_id`  
    FOREIGN KEY (`time_id`)  
    REFERENCES `futebol_db`.`clube` (`idTime`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```


QUERY'S DE CRIAÇÃO DE TABELA

```
CREATE TABLE `futebol_db`.`jogador` (  
  `idjogador` INT NOT NULL AUTO_INCREMENT,  
  `nome_jogador` VARCHAR(100) NOT NULL,  
  `posicao` VARCHAR(45) NOT NULL,  
  `data_nascimento` DATE NOT NULL,  
  `idade` INT NOT NULL,  
  `time_id` INT NOT NULL,  
  PRIMARY KEY (`idjogador`),  
  INDEX `fk_time_id2_idx` (`time_id` ASC) VISIBLE,  
  CONSTRAINT `fk_time_id2`  
    FOREIGN KEY (`time_id`)  
    REFERENCES `futebol_db`.`clube` (`idTime`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

QUERY'S DE CRIAÇÃO DE TABELA

```
CREATE TABLE `futebol_db`.`treinamento` (  
  `idtreinamento` INT NOT NULL AUTO_INCREMENT,  
  `data_treinamento` DATE NOT NULL,  
  `tipo_treinamento` VARCHAR(45) NOT NULL,  
  `time_id` INT NOT NULL,  
  PRIMARY KEY (`idtreinamento`),  
  INDEX `fk_time_id3_idx` (`time_id` ASC) VISIBLE,  
  CONSTRAINT `fk_time_id3`  
    FOREIGN KEY (`time_id`)  
    REFERENCES `futebol_db`.`clube` (`idTime`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

QUERY'S DE CRIAÇÃO DE TABELA

```
CREATE TABLE `futebol_db`.`partida` (  
  `idpartida` INT NOT NULL AUTO_INCREMENT,  
  `data_partida` DATE NOT NULL,  
  `hora_partida` TIME NOT NULL,  
  `id_time_casa` INT NOT NULL,  
  `id_time_visitante` INT NOT NULL,  
  `gols_time_casa` INT NOT NULL,  
  `gols_time_visitante` INT NOT NULL,  
  `arbitro_id` INT NOT NULL,  
  PRIMARY KEY (`idpartida`),  
  INDEX `fk_id_time_casa_idx` (`id_time_casa` ASC) VISIBLE,  
  INDEX `fk_id_time_visitante_idx` (`id_time_visitante` ASC) VISIBLE,  
  INDEX `fk_arbitro_id_idx` (`arbitro_id` ASC) VISIBLE,  
  CONSTRAINT `fk_id_time_casa`  
    FOREIGN KEY (`id_time_casa`)  
      REFERENCES `futebol_db`.`clube` (`idTime`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_id_time_visitante`  
    FOREIGN KEY (`id_time_visitante`)  
      REFERENCES `futebol_db`.`clube` (`idTime`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_arbitro_id`  
    FOREIGN KEY (`arbitro_id`)  
      REFERENCES `futebol_db`.`arbitragem` (`idarbitragem`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION);
```

QUERY'S DE CRIAÇÃO DE TABELA

```
CREATE TABLE `futebol_db`.`escalacao` (  
  `idescalacao` INT NOT NULL AUTO_INCREMENT,  
  `partida_id` INT NOT NULL,  
  `time_id` INT NOT NULL,  
  `jogador_id` INT NOT NULL,  
  `posicao_campo` VARCHAR(100) NOT NULL,  
  PRIMARY KEY (`idescalacao`),  
  INDEX `fk_time_id4_idx` (`time_id` ASC) VISIBLE,  
  INDEX `fk_partida_id_idx` (`partida_id` ASC) VISIBLE,  
  INDEX `fk_jogador_id_idx` (`jogador_id` ASC) VISIBLE,  
  CONSTRAINT `fk_time_id4`  
    FOREIGN KEY (`time_id`)  
      REFERENCES `futebol_db`.`clube` (`idTime`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_partida_id`  
    FOREIGN KEY (`partida_id`)  
      REFERENCES `futebol_db`.`partida` (`idpartida`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_jogador_id`  
    FOREIGN KEY (`jogador_id`)  
      REFERENCES `futebol_db`.`jogador` (`idjogador`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION);
```

QUERY'S DE CRIAÇÃO DE TABELA

```
CREATE TABLE `futebol_db`.`lesao` (  
  `idlesao` INT NOT NULL AUTO_INCREMENT,  
  `jogador_id` INT NOT NULL,  
  `data_lesao` DATE NOT NULL,  
  `tipo_lesao` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`idlesao`),  
  INDEX `fk_jogador_id2_idx` (`jogador_id` ASC) VISIBLE,  
  CONSTRAINT `fk_jogador_id2`  
    FOREIGN KEY (`jogador_id`)  
      REFERENCES `futebol_db`.`jogador` (`idjogador`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

```
CREATE TABLE `futebol_db`.`participacao_treinamento` (  
  `idparticipacao_treinamento` INT NOT NULL AUTO_INCREMENT,  
  `treinamento_id` INT NOT NULL,  
  `jogador_id` INT NOT NULL,  
  PRIMARY KEY (`idparticipacao_treinamento`),  
  INDEX `fk_treinamento_id_idx` (`treinamento_id` ASC) VISIBLE,  
  INDEX `fk_jogador_id3_idx` (`jogador_id` ASC) VISIBLE,  
  CONSTRAINT `fk_treinamento_id`  
    FOREIGN KEY (`treinamento_id`)  
      REFERENCES `futebol_db`.`treinamento` (`idtreinamento`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_jogador_id3`  
    FOREIGN KEY (`jogador_id`)  
      REFERENCES `futebol_db`.`jogador` (`idjogador`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```


QUERY'S DE CRIAÇÃO DE TABELA

```
CREATE TABLE `futebol_db`.`contrato` (  
  `idcontrato` INT NOT NULL AUTO_INCREMENT,  
  `jogador_id` INT NOT NULL,  
  `inicio_contrato` DATE NOT NULL,  
  `fim_contrato` DATE NOT NULL,  
  `salario` FLOAT NOT NULL,  
  PRIMARY KEY (`idcontrato`),  
  INDEX `fk_jogador_id4_idx` (`jogador_id` ASC) VISIBLE,  
  CONSTRAINT `fk_jogador_id4`  
    FOREIGN KEY (`jogador_id`)  
    REFERENCES `futebol_db`.`jogador` (`idjogador`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

QUERY'S DE POPULAÇÃO DE TABELA

```
INSERT INTO clube(nome_time, fundacao, estadio)
VALUES
('FC Barcelona', '1899-11-29', 'Camp Nou'),
('Real Madrid', '1902-03-06', 'Santiago Bernabéu'),
('São Paulo FC', '1930-01-25', 'Morumbi');
```

```
INSERT INTO arbitragem(nome_arbitro, cpf_arbitro)
VALUES
('Wilton Pereira Sampaio', '12345678901');
```

```
INSERT INTO contrato(jogador_id, inicio_contrato, fim_contrato, salario)
VALUES
(1, '2023-07-02', '2026-06-30', '1600000'),
(2, '2024-07-16', '2029-07-26', '15000000'),
(3, '2022-07-16', '2026-07-18', '23000000'),
(4, '2018-07-20', '2027-07-20', '20000000');
```

```
INSERT INTO escalacao(partida_id, jogador_id, time_id, posicao_campo)
VALUES
(1, 1, 1, 'Meio-Campo'),
(1, 2, 2, 'Atacante'),
(1, 3, 1, 'Atacante'),
(1, 4, 2, 'Ponta-Esquerda');
```

```
INSERT INTO jogador(nome_jogador, posicao, data_nascimento, idade, time_id)
VALUES
('Lamine Yamal', 'Meio-Campo', '2007-07-13', 17, 1),
('Kylian Mbappé', 'Atacante', '1998-12-20', 25, 2),
('Robert Lewandowski', 'Atacante', '1988-08-21', 36, 1),
('Vini Jr', 'Ponta-Esquerda', '2000-07-12', 24, 2);
```

```
INSERT INTO lesao(jogador_id, data_lesao, tipo_lesao)
VALUES
(1, '2024-05-06', 'Estiramento na coxa'),
(3, '2024-04-19', 'Estiramento na coxa'),
(2, '2024-05-06', 'Rompimento de ligamento');
```

QUERY'S DE POPULAÇÃO DE TABELA

```
INSERT INTO participacao_treino(treinamento_id, jogador_id)
VALUES
(1, 2),
(1, 4);
```

```
INSERT INTO partida(data_partida, hora_partida, id_time_casa, id_time_visitante, gols_time_casa, gols_time_visitante, arbitro_id)
VALUES
('2024-09-15', '16:00', 1, 2, 1, 3, 1);
```

```
INSERT INTO treinador(nome_treinador, data_nascimento, time_id)
VALUES
('Hansi Flick', '1965-02-24', 1),
('Carlo Ancelotti', '1959-06-10', 2)
```

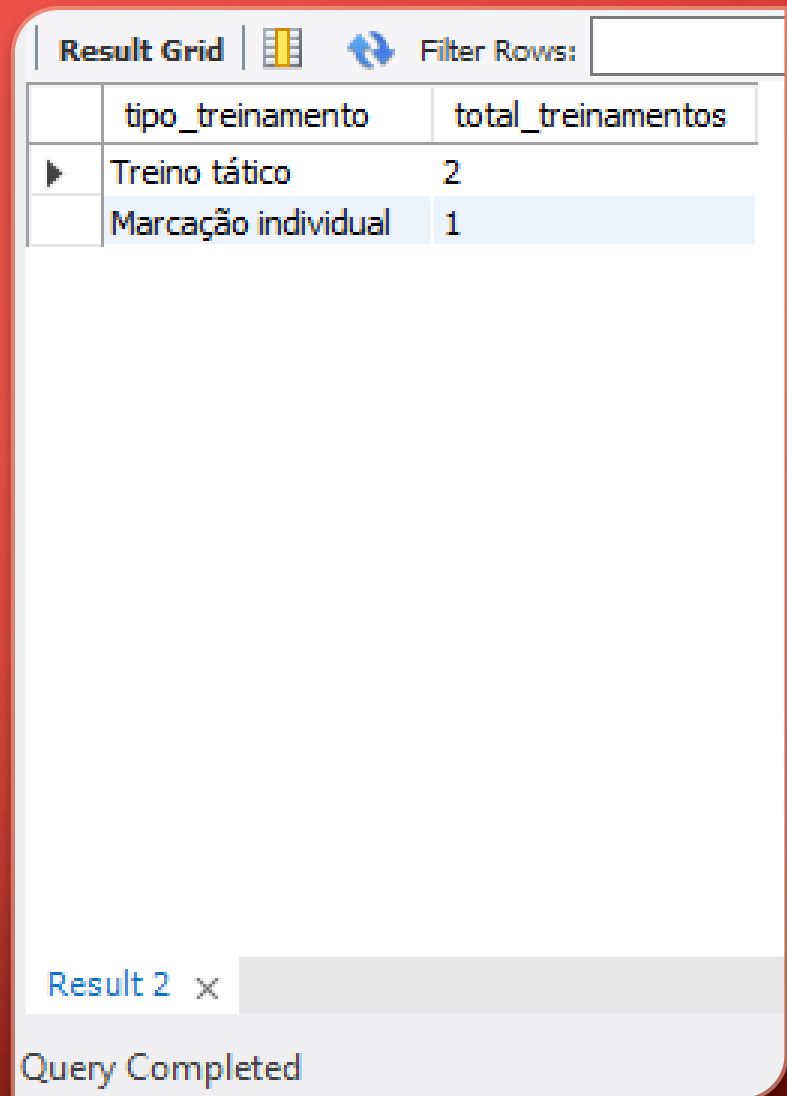
```
INSERT INTO treinamento(data_treinamento, tipo_treinamento, time_id)
VALUES
('2024-09-12', 'Treino tático', 2)
('2024-09-12', 'Marcação individual', 2)
('2024-09-10', 'Treino tático', 1)
```

QUERY GROUP BY

```
SELECT tipo_treinamento, COUNT(idtreinamento) AS  
total_treinos FROM Treinamento
```

```
GROUP BY tipo_treinamento;
```

Esse código tem função de contar o total de treinos que existem para cada tipo de treinamento registrado na tabela `Treinamento`. A consulta está agrupando os dados com base no tipo de treinamento e contando quantos registros existem em cada tipo.



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a SQL query. The grid has two columns: 'tipo_treinamento' and 'total_treinos'. There are two rows of data: 'Treino tático' with a count of 2, and 'Marcação individual' with a count of 1. The second row is highlighted. Below the grid, it says 'Result 2' and 'Query Completed'.

	tipo_treinamento	total_treinos
▶	Treino tático	2
	Marcação individual	1

QUERY GROUP BY

```
SELECT posicao, COUNT(idjogador) AS total_jogadores  
FROM Jogador
```

```
GROUP BY posicao;
```

Este código conta quantos jogadores existem para cada posição registrada na tabela `Jogador`. A consulta agrupa os dados com base na posição dos jogadores e conta quantos jogadores estão em cada posição.

	posicao	total_jogadores
▶	Meio-Campo	1
	Atacante	2
	Ponta-Esquerda	1

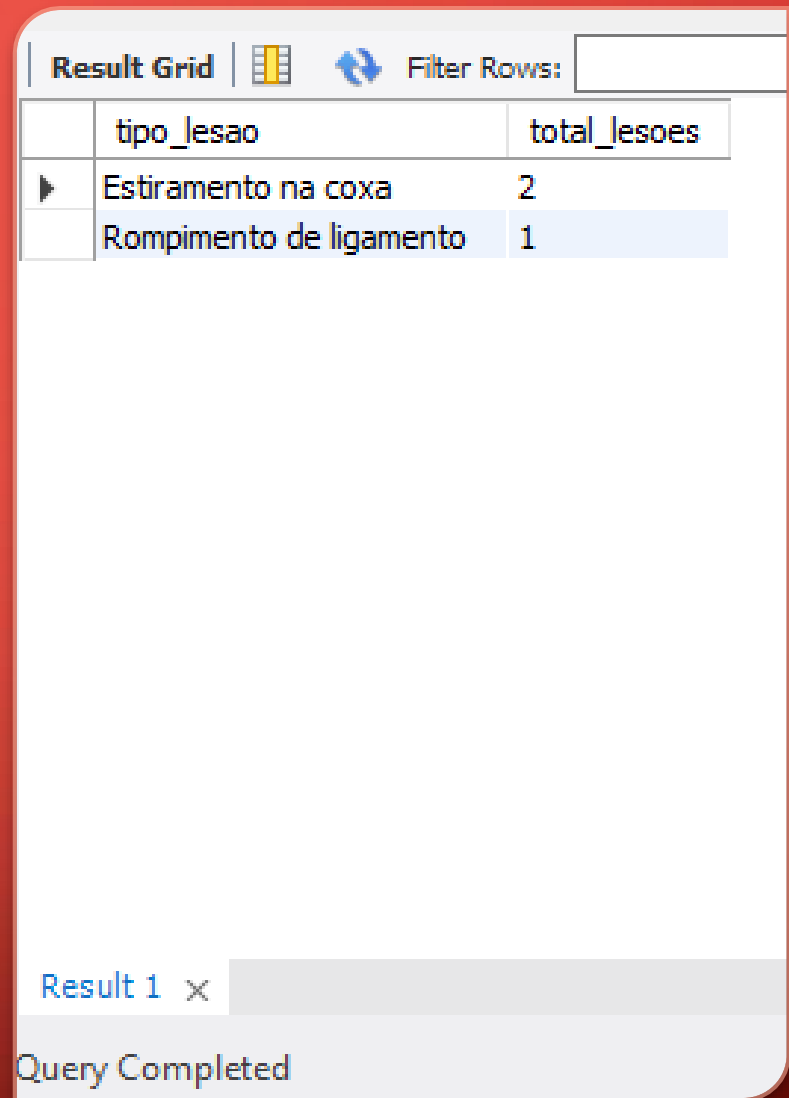
Result 1 x

QUERY GROUP BY

```
SELECT tipo_lesao, COUNT(idlesao) AS total_lesoes  
FROM Lesao
```

```
GROUP BY tipo_lesao;
```

O código é usado para contar o número de lesões de cada tipo registradas na tabela `Lesao`. A consulta agrupa os dados com base no tipo de lesão e conta o total de lesões de cada tipo que foi registrada.



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a SQL query. The grid has two columns: 'tipo_lesao' and 'total_lesoes'. There are two rows of data. The first row shows 'Estiramento na coxa' with a count of 2. The second row shows 'Rompimento de ligamento' with a count of 1. Below the grid, it says 'Result 1' and 'Query Completed'.

	tipo_lesao	total_lesoes
▶	Estiramento na coxa	2
	Rompimento de ligamento	1

Result 1 x

Query Completed

QUERY INNER JOIN

```
SELECT p.idpartida, t1.nome_time AS time_casa,  
t2.nome_time AS time_visitante, p.gols_time_casa,  
p.gols_time_visitante, t1.estadio as estadio,  
p.data_partida, p.hora_partida FROM Partida p
```

```
INNER JOIN clube t1 ON p.id_time_casa =  
t1.idtime INNER JOIN clube t2 ON p.id_time_visitante =  
t2.idtime;
```

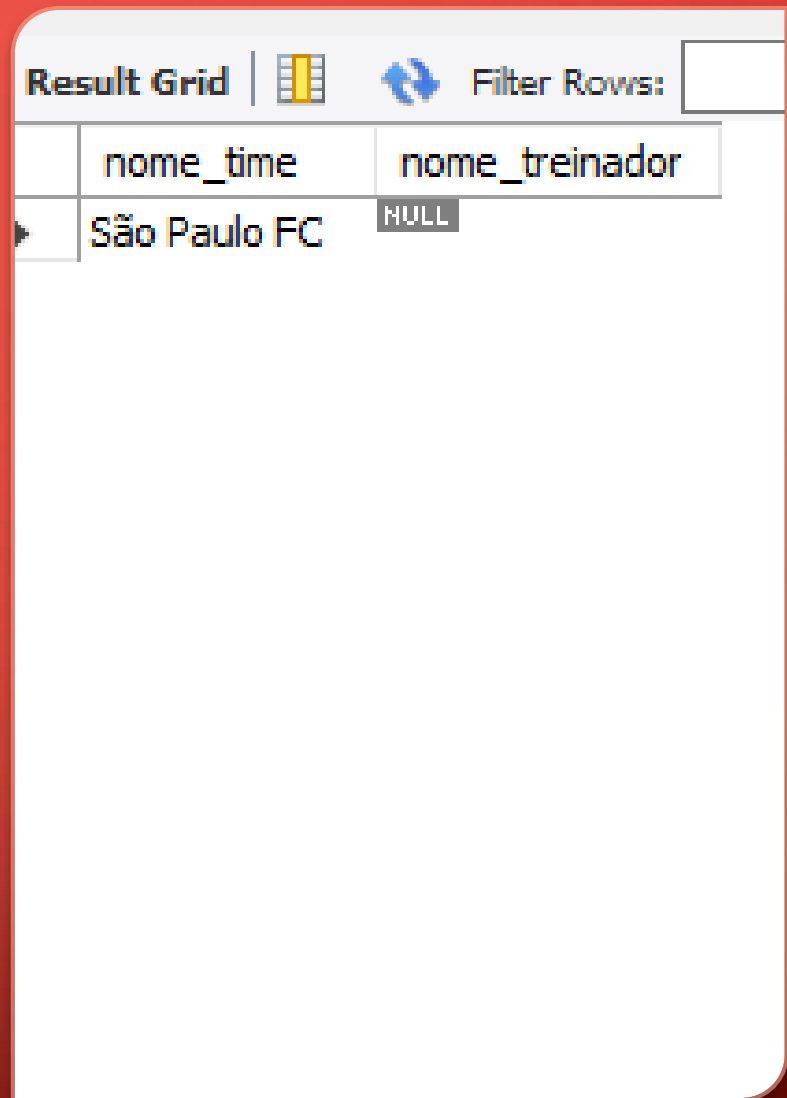
	idpartida	time_casa	time_visitante	gols_time_casa	gols_time_visitante	estadio	data_partida	hora_partida
▶	1	FC Barcelona	Real Madrid	1	3	Camp Nou	2024-09-15	16:00:00

O código tem a função de obter detalhes das partidas, incluindo os nomes dos times (casa e visitante), gols marcados, estádio, data e hora da partida. Ele usa um INNER JOIN para combinar as tabelas Partida e Clube, para trazer os nomes dos times e o estádio onde a partida foi realizada.

QUERY JOIN LEFT EXCLUSIVO

```
SELECT t.nome_time, tr.nome_treinador FROM clube t  
LEFT JOIN Treinador tr ON t.idtime = tr.time_id WHERE  
tr.idtreinador IS NULL;
```

Este código faz uma consulta para encontrar os times na tabela clube que não têm um treinador associado na tabela Treinador.



	nome_time	nome_treinador
▶	São Paulo FC	NULL

QUERY JOIN RIGHT INCLUSIVO

```
SELECT j.nome_jogador, t.nome_time FROM Jogador j  
RIGHT JOIN clube t ON j.time_id = t.idtime;
```

Este código está realizando uma consulta para listar todos os jogadores e os seus respectivos times, mas também inclui todos os times, mesmo aqueles que não têm jogadores associados

nome_jogador	nome_time
Robert Lewandowski	FC Barcelona
Lamine Yamal	FC Barcelona
Vini Jr	Real Madrid
Kylian Mbappé	Real Madrid
NULL	São Paulo FC