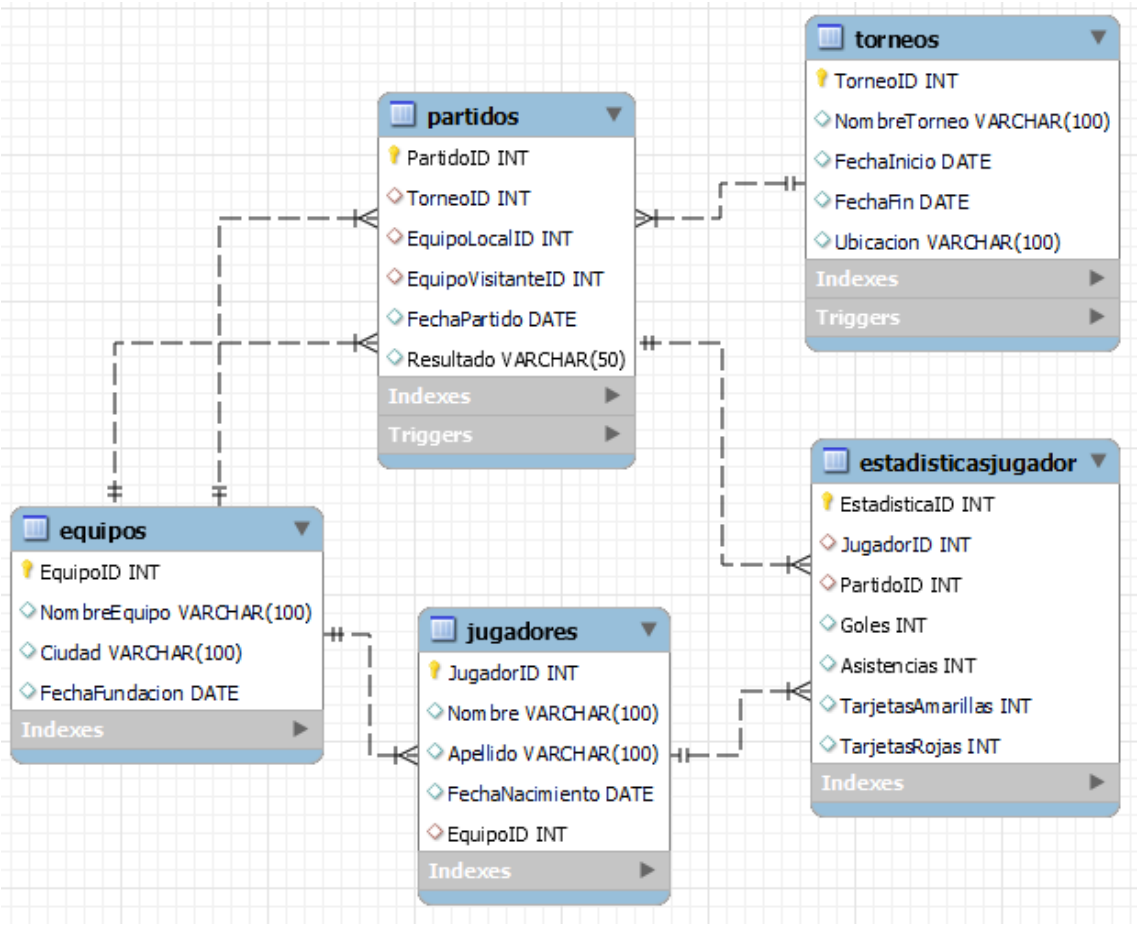
Simulacro Parcial



Script

```
create database simulacro;

CREATE TABLE Equipos (
    EquipoID INT PRIMARY KEY AUTO_INCREMENT,
    NombreEquipo VARCHAR(100),
    Ciudad VARCHAR(100),
    FechaFundacion DATE
);

INSERT INTO Equipos (NombreEquipo, Ciudad, FechaFundacion)
VALUES
('Tigres FC', 'Buenos Aires', '1985-03-21'),
('Águilas Doradas', 'Mendoza', '1990-07-15'),
('Leones Negros', 'Cordoba', '1978-11-05');

CREATE TABLE Jugadores (
    JugadorID INT PRIMARY KEY AUTO_INCREMENT,
    Nombre VARCHAR(100),
    Apellido VARCHAR(100),
    FechaNacimiento DATE,
    EquipoID INT,
```

```
    FOREIGN KEY (EquipoID) REFERENCES Equipos(EquipoID)
);

INSERT INTO Jugadores (Nombre, Apellido, FechaNacimiento, EquipoID)
VALUES
('Lucas', 'González', '1995-04-12', 1),
('Matías', 'Rodríguez', '1993-09-23', 2),
('Diego', 'Fernández', '1990-01-15', 3);

CREATE TABLE Torneos (
    TorneoID INT PRIMARY KEY AUTO_INCREMENT,
    NombreTorneo VARCHAR(100),
    FechaInicio DATE,
    FechaFin DATE,
    Ubicacion VARCHAR(100)
);

INSERT INTO Torneos (NombreTorneo, FechaInicio, FechaFin, Ubicacion)
VALUES
('Torneo Apertura', '2025-04-01', '2025-04-30', 'Buenos Aires'),
('Torneo Clausura', '2025-05-01', '2025-05-31', 'Cordoba');

CREATE TABLE Partidos (
    PartidoID INT PRIMARY KEY AUTO_INCREMENT,
    TorneoID INT,
    EquipoLocalID INT,
    EquipoVisitanteID INT,
    FechaPartido DATE,
    Resultado VARCHAR(50),
    FOREIGN KEY (TorneoID) REFERENCES Torneos(TorneoID),
    FOREIGN KEY (EquipoLocalID) REFERENCES Equipos(EquipoID),
    FOREIGN KEY (EquipoVisitanteID) REFERENCES Equipos(EquipoID)
);

INSERT INTO Partidos (TorneoID, EquipoLocalID, EquipoVisitanteID, FechaPartido,
Resultado)
VALUES
(1, 1, 2, '2025-04-10', '2-1'),
(1, 2, 3, '2025-04-15', '0-0');
```

Ejercicios

Procedimientos Almacenados

1. **Registrar un nuevo partido:** procedimiento almacenado llamado registrarPartido. Este procedimiento debe:
    - Recibir como parámetros los IDs de los equipos (local y visitante), el ID del torneo y la fecha del partido.
    - Verificar que los IDs de los equipos y del torneo existan.
    - Registrar un nuevo partido en la tabla Partidos.
    - Manejar errores mediante transacciones (ROLLBACK en caso de IDs inexistentes o errores de SQL).

- Devolver un mensaje indicando el éxito o el error de la operación.

2. **Registrar estadísticas de un jugador:** procedimiento almacenado llamado registrarEstadisticasJugador que:
   - Reciba como parámetros el ID del jugador, el ID del partido, goles, asistencias y tarjetas (amarillas y rojas).
   - Verifique que los IDs proporcionados (jugador y partido) existan.
   - Si un ID no existe, genere un error personalizado usando SIGNAL.
   - Inserte las estadísticas del jugador en la tabla EstadisticasJugador.
   - Devuelva un mensaje de confirmación si la operación es exitosa.
3. **Actualizar estadísticas de un jugador:** procedimiento almacenado llamado actualizarEstadisticasJugador que:
   - Reciba como parámetros el ID del jugador, el ID del partido, goles, asistencias y tarjetas.
   - Verifique que los IDs existan.
   - Actualice las estadísticas del jugador en la tabla EstadisticasJugador.
   - Maneje errores mediante transacciones, usando ROLLBACK en caso de fallas.
   - Devuelva un mensaje de éxito o error.

Triggers

4. **Auditar inscripciones a torneos:** TRIGGER llamado auditarInscripcionTorneo que:
   - Se dispare después de insertar un registro en la tabla Torneos.
   - Inserte en la tabla AuditoriaInscripciones el ID del torneo y su nombre.
5. **Auditar cambios en resultados de partidos:** TRIGGER llamado auditarCambioResultado que:
   - Se dispare después de actualizar un registro en la tabla Partidos.
   - Si el resultado del partido cambia, registre los valores antiguos y nuevos en la tabla AuditoriaResultados.

UDF's

6. **Calcular el promedio de goles por jugador:** función llamada promedioGolesPorJugador que:
   - Reciba como parámetro el ID de un jugador.
   - Calcule el promedio de goles del jugador dividiendo el total de goles entre la cantidad de partidos jugados.
   - Devuelva el resultado como un valor decimal.

7. **Obtener estadísticas de un jugador:** función llamada estadisticasJugador que:
    - Reciba como parámetro el ID de un jugador.
    - Devuelva una cadena de texto con el nombre completo del jugador, el total de goles y asistencias registrados en la tabla EstadisticasJugador.

CTE's

8. **Crear una consulta utilizando una CTE llamada TarjetasPorJugador que:**
    - Agrupe las tarjetas amarillas y rojas de cada jugador en partidos.
    - Combine las tarjetas acumuladas utilizando UNION ALL.
    - En la consulta principal, muestre el nombre del jugador y la cantidad total de tarjetas acumuladas (amarillas + rojas), ordenadas en orden descendente.
8. **Estadísticas de los mejores jugadores:** CTE llamada TopJugadores que:
    - Seleccione los jugadores con un total de goles mayor a 5 y asistencias mayor o igual a 3.
    - En la consulta principal, muestra el nombre, apellido, equipo, total de goles y asistencias de los jugadores destacados.