



PRIMARY KEY



regular line

**total participation**

## Rules

- Each game has exactly one white player and one black player
- Same player pair cannot play same round in same event
- No two events with same name on same date
- Game identified by Event + Round combination

## Participation Constraints

(O, N) - Optional, Many

(1, 1) - Mandatory, One

(1, N) - Mandatory, Many

```

-- Drop tables if they exist (for clean setup)
DROP TABLE IF EXISTS Game;
DROP TABLE IF EXISTS Event;
DROP TABLE IF EXISTS Player;

-- Create Player Table (Strong Entity)
CREATE TABLE Player (
    PlayerID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    EloRating INT NOT NULL
);

-- Create Event Table (Strong Entity)
CREATE TABLE Event (
    EventID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Site VARCHAR(100) NOT NULL,
    Date DATE NOT NULL,

    -- Constraint: No two events with same name can occur on same date
    CONSTRAINT unique_event_name_date UNIQUE (Name, Date)
);

-- Create Game Table (Weak Entity - depends on Event)
-- Game is identified by EventID + Round (composite key)
-- Includes separate foreign keys for white and black players
CREATE TABLE Game (
    EventID INT NOT NULL,
    Round INT NOT NULL,
    WhitePlayerID INT NOT NULL,
    BlackPlayerID INT NOT NULL,
    Result VARCHAR(10) NOT NULL,
    Moves TEXT(1000),

    -- Composite Primary Key (EventID + Round identifies a game)
    PRIMARY KEY (EventID, Round),

    -- Foreign Key to Event (Identifying Relationship - Hosts)
    CONSTRAINT fk_game_event
        FOREIGN KEY (EventID) REFERENCES Event(EventID)
        ON DELETE CASCADE ON UPDATE CASCADE,

    -- Foreign Key to Player for White (Plays White Relationship)
    CONSTRAINT fk_game_white_player
        FOREIGN KEY (WhitePlayerID) REFERENCES Player(PlayerID)
        ON DELETE RESTRICT ON UPDATE CASCADE,

    -- Foreign Key to Player for Black (Plays Black Relationship)
    CONSTRAINT fk_game_black_player
        FOREIGN KEY (BlackPlayerID) REFERENCES Player(PlayerID)
        ON DELETE RESTRICT ON UPDATE CASCADE,

    -- Business Rule Constraints
    CONSTRAINT valid_result
        CHECK (Result IN ('1-0', '0-1', '1/2-1/2')),

    -- Ensure white and black players are different
    CONSTRAINT different_players
        CHECK (WhitePlayerID != BlackPlayerID),

    -- Constraint: Same players cannot play same round in same event
    CONSTRAINT unique_players_round_event
        UNIQUE (EventID, Round, WhitePlayerID, BlackPlayerID)
);

```