

Nicole Dávila Hernández

A01784217

Construcción de software y toma de decisiones (Gpo 501)

Esteban Castillo Juarez

5. Ejercicio de creación de una base de datos relacional en MySQL para el juego de cartas Pokemon TCG

Explicación de Cumplimiento con la Cuarta Forma Normal (4NF)

En el diagrama, base de datos y tuplas presentes en mi repositorio de GitHub, se trata de un modelo en su cuarta forma de normalización. Esto es debido a ciertos atributos que contiene el modelo, tales como:

Atomicidad de Atributos:

Todos los atributos en cada tabla son atómicos, lo que significa que contienen valores indivisibles. Por ejemplo, `player_username`, `card_name`, `match_date`, etc., son todos valores únicos y no pueden ser divididos más.

Eliminación de Dependencias Parciales:

El esquema asegura que todos los atributos no clave son completamente dependientes funcionalmente de la clave primaria. Esto se evidencia en el diseño de cada tabla. Por ejemplo, en la tabla `Player`, todos los atributos como `player_username`, `player_rank`, etc., dependen completamente de `player_id`.

Eliminación de Dependencias Transitivas:

Todas las dependencias transitivas están eliminadas. Por ejemplo, en la tabla `Card`, atributos como `card_name`, `card_type`, etc., son directamente dependientes de `card_id`, y ningún atributo es dependiente indirectamente de otro atributo no clave.

Eliminación de Dependencias Multivaluadas:

Cada tabla está diseñada para evitar dependencias multivaluadas. Por ejemplo, la tabla `Tournament_Player`, que enlaza jugadores con torneos:

```
CREATE TABLE Tournament_Player (  
    tournament_id INT,  
    player_id INT,  
    PRIMARY KEY (tournament_id, player_id),  
    FOREIGN KEY (tournament_id) REFERENCES Tournament(tournament_id),
```

```
FOREIGN KEY (player_id) REFERENCES Player(player_id)
);
```

Aquí, tournament_id no determina múltiples valores independientes de player_id sin considerar también el contexto de player_id en sí. Esto significa que la relación entre tournament_id y player_id es muchos-a-muchos pero se resuelve sin ninguna dependencia multivaluada independiente.

Separación Lógica de Datos:

Los datos están separados lógicamente en diferentes tablas basadas en sus entidades y relaciones, previniendo cualquier redundancia innecesaria. Por ejemplo, Player, Card, Deck, Tournament, etc., son todas entidades separadas con sus relaciones definidas en tablas asociativas como Tournament_Player y Deck_Cards.

Tomando en cuenta estas características, el diseño de la base de datos proporcionada cumple con la cuarta forma normal, asegurando la integridad de los datos, eliminando la redundancia y previniendo anomalías durante las operaciones de datos.