



Tecnológico de Monterrey

Nombre:

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

Materia:

Diseño de arquitecturas, uso y administración de datos

Estudiante:

Alejandro Acevedo Granados

IDs:

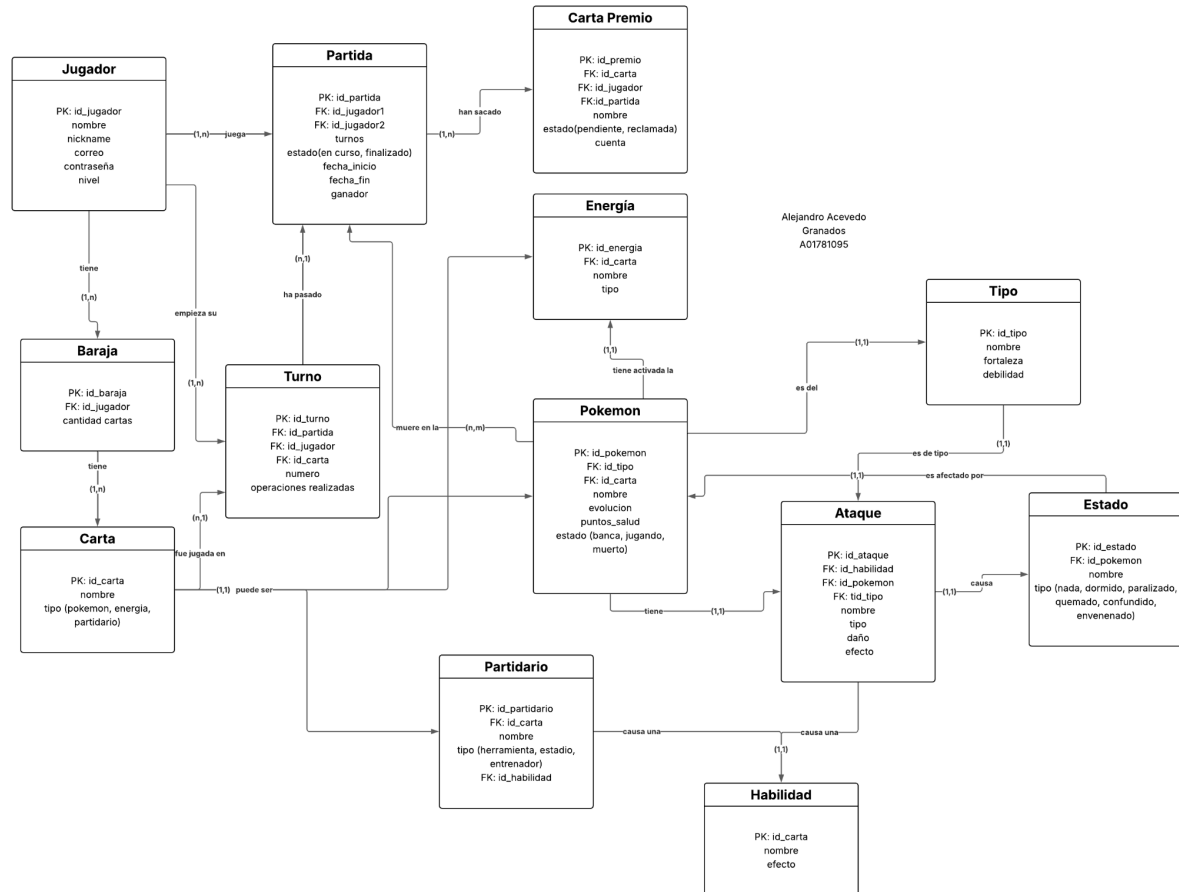
A01781095

Profesor/es:

Esteban Castillo Juarez

Fecha de entrega:

Domingo, 27 de Abril 2025



Se agregó la entidad turno basado en comentarios

En este diagrama se puede apreciar mi modelo de una partida de Pokemon TCG, en mi caso puse 13 entidades, a continuación voy a explicar cada primary y foreign key una de ellas, así como su nivel de normalización y restricciones de integridad:

Claves Primarias (Primary Keys)

1. Jugador: id_jugador

- Justificación: Identificador único para cada jugador del juego.

2. **Partida:** id_partida (con AUTO_INCREMENT)

- Justificación: Identifica de manera única cada partida que se juega entre dos jugadores.

3. **Carta Premio:** id_premio (con AUTO_INCREMENT)

- Justificación: Identifica cada carta premio de manera única en el sistema.

4. **Baraja:** id_baraja (con AUTO_INCREMENT)

- Justificación: Identifica cada mazo o baraja de cartas de manera única.

5. **Carta:** id_carta (con AUTO_INCREMENT)

- Justificación: Identificador único para cada carta del juego, independientemente de su tipo.

6. **Turno:** id_turno (con AUTO_INCREMENT)

- Justificación: Identifica cada turno de juego de manera única.

7. **Energía:** id_energia (con AUTO_INCREMENT)

- Justificación: Identifica cada carta de energía de manera única.

8. **Pokemon:** id_pokemon (con AUTO_INCREMENT)

- Justificación: Identifica cada carta Pokémon de manera única.

9. **Partidario:** id_partidario (con AUTO_INCREMENT)

- Justificación: Identifica cada carta de entrenador/herramienta/estadio de manera única.

10. **Tipo:** id_tipo (con AUTO_INCREMENT)

- Justificación: Identifica cada tipo de Pokémon en el juego.

11. **Ataque:** id_ataque (con AUTO_INCREMENT)

- Justificación: Identifica cada ataque de manera única.

12. **Estado:** id_estado (con AUTO_INCREMENT)

- Justificación: Identifica cada estado especial de un Pokémon.

13. **Habilidad:** id_carta

- Justificación: Usa el id_carta como clave primaria, lo que indica que cada carta tiene una habilidad específica.

Claves Foráneas (Foreign Keys)

1. **Partida:**

- `id_jugador1` → Jugador(`id_jugador`)

- `id_jugador2` → Jugador(`id_jugador`)

- `ganador` → Jugador(`id_jugador`)

- Justificación: Conecta la partida con los dos jugadores participantes y el ganador.

2. **Carta_Premio:**

- `id_carta` → Carta(`id_carta`)

- `id_jugador` → Jugador(`id_jugador`)

- `id_partida` → Partida(`id_partida`)

- Justificación: Relaciona cada carta premio con la carta específica, el jugador que la posee y la partida a la que pertenece.

3. Baraja:

- `id_jugador` → Jugador(`id_jugador`)
- Justificación: Conecta cada baraja con su propietario.

4. Turno:

- `id_partida` → Partida(`id_partida`)
- `id_jugador` → Jugador(`id_jugador`)
- `id_carta` → Carta(`id_carta`)
- Justificación: Relaciona cada turno con la partida, el jugador que lo realiza y la carta que juega.

5. Energia:

- `id_carta` → Carta(`id_carta`)
- Justificación: Conecta la carta de energía con su entrada en la tabla Carta.

6. Pokemon:

- `id_tipo` → Tipo(`id_tipo`)
- `id_carta` → Carta(`id_carta`)
- Justificación: Conecta cada Pokémon con su tipo y su entrada en la tabla Carta.

7. Partidario:

- `id_carta` → Carta(`id_carta`)
- `id_habilidad` → Habilidad(`id_carta`)

- Justificación: Conecta la carta de partidario con su entrada en la tabla Carta y su habilidad.

8. **Ataque:**

- `id_habilidad` → Habilidad(`id_carta`)
- `id_pokemon` → Pokemon(`id_pokemon`)
- `id_tipo` → Tipo(`id_tipo`)

- Justificación: Conecta cada ataque con la habilidad, el Pokémon que lo realiza y el tipo al que pertenece.

9. **Estado:**

- `id_pokemon` → Pokemon(`id_pokemon`)
- Justificación: Conecta cada estado con el Pokémon que lo sufre.

10. **Habilidad:**

- `id_carta` → Carta(`id_carta`)
- Justificación: Conecta cada habilidad con la carta a la que pertenece.

Analizando tu esquema de base de datos Pokémon TCG, confirmo que no alcanza la 4ta Forma Normal debido a que hay algunas dependencias multivaluadas. Ahora, te presentaré las restricciones de integridad presentes en cada tabla:

Restricciones de integridad por tabla

Tabla Jugador

id_jugador debe ser único para cada jugador (clave primaria)

nombre, nickname, correo y contraseña no pueden ser nulos

El nivel por defecto es 1 para un nuevo jugador

Tabla Partida

id_partida debe ser único para cada partida (clave primaria)

id_jugador1 e id_jugador2 debe existir en la tabla Jugador

ganador, si está definido, debe ser un id_jugador válido

estado sólo puede tener los valores 'en curso' o 'finalizado'

El valor por defecto de turnos es 0

El valor por defecto de estado es 'en curso'

fecha_inicio por defecto toma la fecha y hora actuales

Tabla Carta_Premio

id_premio debe ser único para cada premio (clave primaria)

id_carta debe existir en la tabla Carta

id_jugador debe existir en la tabla Jugador

id_partida debe existir en la tabla Partida

nombre no puede ser nulo

estado sólo puede ser 'pendiente' o 'reclamada'

El valor por defecto de estado es 'pendiente'

El valor por defecto de cuenta es 0

Tabla Baraja

id_baraja debe ser único para cada baraja (clave primaria)

id_jugador debe existir en la tabla Jugador

El valor por defecto para cantidad_cartas es 60

Tabla Carta

id_carta debe ser único para cada carta (clave primaria)

nombre no puede ser nulo

tipo solo puede ser 'pokemon', 'energia' o 'partidario'

Tabla Turno

id_turno debe ser único para cada turno (clave primaria)

id_partida debe existir en la tabla Partida

id_jugador debe existir en la tabla Jugador

id_carta debe existir en la tabla Carta

El valor por defecto para numero es 0

El valor por defecto para operaciones_realizadas es 0

Tabla Energia

id_energia debe ser único para cada carta de energía (clave primaria)

id_carta debe existir en la tabla Carta

nombre y tipo no pueden ser nulos

Tabla Pokemon

id_pokemon debe ser único para cada Pokémon (clave primaria)

id_tipo debe existir en la tabla Tipo

id_carta debe existir en la tabla Carta

nombre no puede ser nulo

puntos_salud no puede ser nulo

estado sólo puede ser 'banca', 'jugando' o 'muerto'

El valor por defecto para estado es 'banca'

Tabla Partidario

id_partidario debe ser único para cada carta partidario (clave primaria)

id_carta debe existir en la tabla Carta

id_habilidad debe existir en la tabla Habilidad

nombre no puede ser nulo

tipo solo puede ser 'herramienta', 'estadio' o 'entrenador'

Tabla Tipo

id_tipo debe ser único para cada tipo (clave primaria)

nombre no puede ser nulo

Tabla Ataque

id_ataque debe ser único para cada ataque (clave primaria)

id_habilidad debe existir en la tabla Habilidad

id_pokemon debe existir en la tabla Pokemon

id_tipo debe existir en la tabla Tipo

nombre, tipo y daño no pueden ser nulos

Tabla Estado

id_estado debe ser único para cada estado (clave primaria)

id_pokemon debe existir en la tabla Pokemon

nombre no puede ser nulo

tipo solo puede ser 'nada', 'dormido', 'paralizado', 'quemado', 'confundido' o 'envenenado'

El valor por defecto para tipo es 'nada'

Tabla Habilidad

id_carta debe ser único para cada habilidad (clave primaria)

id_carta debe existir en la tabla Carta

nombre no puede ser nulo

Primera Forma Normal (1FN)

El esquema cumple con la 1FN porque:

- Todos los atributos son atómicos (no hay valores multivaluados o compuestos).
- Cada tabla tiene una clave primaria identificada.
- No hay grupos repetitivos en las tablas.

Segunda Forma Normal (2FN)

El esquema cumple con la 2FN porque:

- Ya cumple con la 1FN.
- Todos los atributos no clave dependen completamente de la clave primaria, no de parte de ella.
- Por ejemplo, en la tabla Pokemon, atributos como nombre, evolucion, puntos_salud y estado dependen completamente del id_pokemon, no de una parte de la clave.

Tercera Forma Normal (3FN)

El esquema cumple con la 3FN porque:

- Ya cumple con la 2FN.
- No hay dependencias transitivas (donde un atributo no clave depende de otro atributo no clave).
- La información está distribuida en tablas específicas: datos de jugadores en Jugador, datos de partidas en Partida, etc.

No alcanza la 4ta Forma Normal debido a que hay algunas dependencias multivaluadas.

Existen tablas donde un atributo puede determinar varios valores independientes entre sí. Por ejemplo, en la tabla Pokémon, un mismo Pokémon puede tener múltiples ataques y múltiples evoluciones, pero estos atributos son independientes entre sí (los ataques no dependen de las evoluciones).