
Torneo de Pokémon Go

Modelo Relacional



1. CORRECCIONES

Registrar:

Cambiamos la participación de la relación Registrar entre encargado registro y ParticipanteUNAM total de lado de participanteUNAM pues debe ser registrado por un encargado y parcial de lado de encargado registro pues no forzosamente ha registrado a alguien.

Llave numcuenta:

Modificamos la llave primaria de ParticipanteUNAM a id_persona.

Inscribir

Renombramos la tabla Inscribir de la relación entre EncargadoRegistro y ParticipanteUNAM a EncargadoInscribirParticipante y la tabla de la relación entre Evento y ParticipanteUNAM a ParticipanteInscribirEvento.

Registrar distancia recorrida:

La relación registrar entre DistanciaRecorrida y CuentaPokemonGo al ser de uno a muchos. Por lo que se agrega a la tabla DistanciaRecorrida la llave de CuentaPokemonGo quedando: DistanciaRecorrida(edición, id_torneo, id_distancia, locación, fecha, hora, id_persona, CódigoDeEntrenador)

Registrar captura:

La relación registrar entre Pokemon, CapturaPokemon y CuentaPokemonGo debe tener las llaves de las tres entidades quedando Registrar(id_pokemon, edición, id_torneo, id_captura, id_persona, CódigoDeEntrenador)

2. DOMINIOS:

- **Evento**

- Edición : Será de tipo Integer y deberá cumplir las restricciones de NOT NULL y UNIQUE ya que es la llave primaria.
- Fecha: Es de tipo Date y deberá cumplir la restricción NOT NULL.

- **CuentaPokemon**

- id_persona: Es de tipo Integer ya que hace referencia a la columna id_persona de ParticipanteUNAM. Ya que es parte de la llave primaria debe cumplir con la condición de NOT NULL pero no necesariamente es única pues un participante puede tener multiples cuentas.

- **CodigoDeEntrenador:** Es de tipo Integer, al ser componente de la llave primaria debe cumplir con ser NOT NULL y como es la llave de la entidad débil debe cumplir con UNIQUE. Para asignar estos códigos automáticamente agregamos Serial.
- **Equipo:** Será de tipo VARCHAR(20), es decir, una cadena máximo 20 caracteres, toda cuenta debe pertenecer a un equipo por lo que debe cumplir la condición de NOT NULL además tiene un CHECK pues debe ser una de tres opciones segun el caso de uso.
- **Nivel:** Es de tipo smallint, además que toda cuenta debe tener un nivel y estos van desde 0 por lo que debe cumplir las restricciones NOT NULL y CHECK (Nivel >= 1).
- **NombreDeUsuario:** Es de tipo VARCHAR(30), es decir, una cadena de a lo más 30 caracteres y como toda cuenta requiere de un nombre será de tipo NOT NULL, además consideramos que todas las cuentas deben tener un nombre único por lo que agregamos la restricción UNIQUE.

• **Pokemon**

- **id_pokemon:** Será de tipo Integer, como es la llave primaria debe cumplir con ser NOT NULL y UNIQUE. Para asignar estos códigos automáticamente agregamos Serial.
- **id_persona y CodigoDeEntrenador** son la llave compuesta de CuentaPokemonGo que estamos usando aquí como llave foránea, ambas son de tipo Integer. Como consideramos que todo Pokemon esta asociado a una cuenta deben cumplir la restricción de NOT NULL.
- **Nombre:** Es de tipo VARCHAR(50), es decir, cadenas de máximo 50 caracteres. Todo pokemon debe tener nombre por lo que debe cumplir la restricción de NOT NULL.
- **Sexo:** Como usaremos la convención 'M' para denotar macho, 'H' para denotar hembra y 'Otro' para otro caso, usaremos el tipo Varchar(4), debe tener uno asignado así que debe cumplir las restricciones NOT NULL CHECK (Sexo IN ('M', 'H', 'Otro'))).
- **Peso:** Es de tipo real pues puede ser decimal, todos deben tener registrado su peso y este valor solo puede positivo así que debe cumplir las restricciones NOT NULL y CHECK (Peso >0).
- **PuntosDeCombate:** Son datos de tipo Integer, los cuales solo pueden ser positivos y todo pokemon debe tenerlos asignados así que deben cumplir con las restricciones NOT NULL CHECK (PuntosDeCombate >= 0).
- **Shiny:** Este valor se refiere a si un pokemon es o no es shiny por lo que es de tipo Boolean y debe cumplir la restricción de NOT NULL.
- **Tipo:** El tipo será especificado en una cadena de tipo VARCHAR(20), es decir, con longitud máxima de 20 caracteres y debe cumplir la restricción de NOT NULL.

- Especie: Se especificara con una cadena de máximo 20 caracteres por lo que usamos Varchar(20) y debe cumplir la restricción de NOT NULL

- **TorneoCapturaShinys**

- Edicion: Es una llave foranea referente a la columna edición de la tabla Evento por lo que es de tipo Integer, como además es parte de la llave primaria debe cumplir con ser NOT NULL.
- id_torneo: Es de tipo Integer y es parte de la llave primaria y además la llave de la entidad débil por lo que además de cumplir la restricción de NOT NULL cumple con ser UNIQUE. Para asignar estos identificadores automáticamente agregamos Serial.
- id_persona: Al ser una llave foranea de ParticipanteUNAM respectamos su tipo Integer. Este dato hace referencia al ganador del torneo por lo que puede ser null.
- CantidadAPremiar: Esta información al referirse a dinero será de tipo real y según lo especificado en el caso de uso tendrá el valor de DEFAULT 500.0.

- TorneoDistanciaRecorrida

- Edicion: Al ser una llave foranea a edicion de evento respecta el tipo Integer y como es parte de la llave compuesta debe cumplir la restricción de NOT NULL
- id_torneo: Es de tipo integer y al ser la llave primaria de la entidad cumple NOT NULL Y UNIQUE. Para asignar estos identificadores automáticamente agregamos Serial.
- id_persona: Es una llave foranea a ParticipanteUNAM(id_persona) por lo que es de tipo Integer, al reflejar al ganador puede ser null en algunos casos.
- CantidadAPremiar: Usamos Real ya que el monto del premio podría incluir centavos, además asignamos DEFAULT 500.0 que es o que especifica el caso de uso.

- CapturaPokemon

- Edicion: Es de tipo Integer, como es parte de la llave compuesta debe cumplir con ser NOT NULL.
- id_torneo: Es de tipo Integer, como es parte de la llave compuesta debe cumplir con ser NOT NULL.
- id_captura: Es de tipo Integer, como es parte de la llave compuesta debe cumplir con ser NOT NULL, además como es la llave primaria de la entidad débil debe cumplir con la restricción de UNIQUE y usamos SERIAL para que este identificador se asigne automáticamente.

- Registrar

- id_pokemon: Es una llave foranea de tipo Integer, como todo registro de una captura pokemon necesita especificar un pokemon cumple la restricción de NOT NULL.

- CódigoDeEntrenador: INTEGER NOT NULL, id_persona INTEGER NOT NULL,
 - id_captura *INTEGER NOT NULL*,
 - Edicion INTEGER NOT NULL,
 - id_torneo *INTEGER NOT NULL*,
 - Fecha DATE NOT NULL,
 - Hora TIMETZ NOT NULL
- DistanciaRecorrida
 -
- **ComprarCuidador**
 - id_persona. Se define como INTEGER para representar una llave foránea con la tabla Cuidador. Se coloca NOT NULL porque siempre debe existir un cuidador asociado a la compra.
 - id_alimento. Se define como INTEGER para representar una llave foránea con la tabla Alimento. Se coloca NOT NULL para garantizar la existencia del producto comprado.
 - MetodoDePago. Se define como VARCHAR(20), un texto corto, porque solo se almacenan nombres de métodos de pago, válidos por CHECK('Tarjeta', 'Efectivo', 'Transferencia'). Se coloca NOT NULL porque cada que compras algo debes de pagarlo.
 - Cantidad. Se define como REAL estrictamente positivo, para considerar todos los tipos de precios. Se coloca como NOT NULL para asegurar que siempre se registre una cantidad.
- **Limpiador**
 - **CorreoLimpiador**
 - **TelefonoLimpiador**
 - **TrabajarLimpiador**
 - **ComprarLimpiador**
 - id_persona. Se define como INTEGER para representar una llave foránea con la tabla Limpiador. Se coloca NOT NULL porque siempre debe existir un limpiador asociado a la compra.
 - id_alimento. Se define como INTEGER para representar una llave foránea con la tabla Alimento. Se coloca NOT NULL para garantizar la existencia del producto comprado.
 - MetodoDePago. Se define como VARCHAR(20), un texto corto, porque solo se almacenan nombres de métodos de pago, válidos por CHECK('Tarjeta', 'Efectivo', 'Transferencia'). Se coloca NOT NULL porque cada que compras algo debes de pagarlo.

- Cantidad. Se define como REAL estrictamente positivo, para considerar todos los tipos de precios. Se coloca como NOT NULL para asegurar que siempre se registre una cantidad.

- **TorneoPelea**
- **PeleaTorneo**
- **Utilizar**