

PROYECTO 2: CASO POKÉMON

CONTENIDO

Introducción	2
Objetivo.....	2
Base de datos Pokémon	2
Creación de nuevas tablas	3
Entrega de creación de tablas.....	5
Creación de consultas.....	5
Entrega de consultas	8
Finalización.....	8

INTRODUCCIÓN

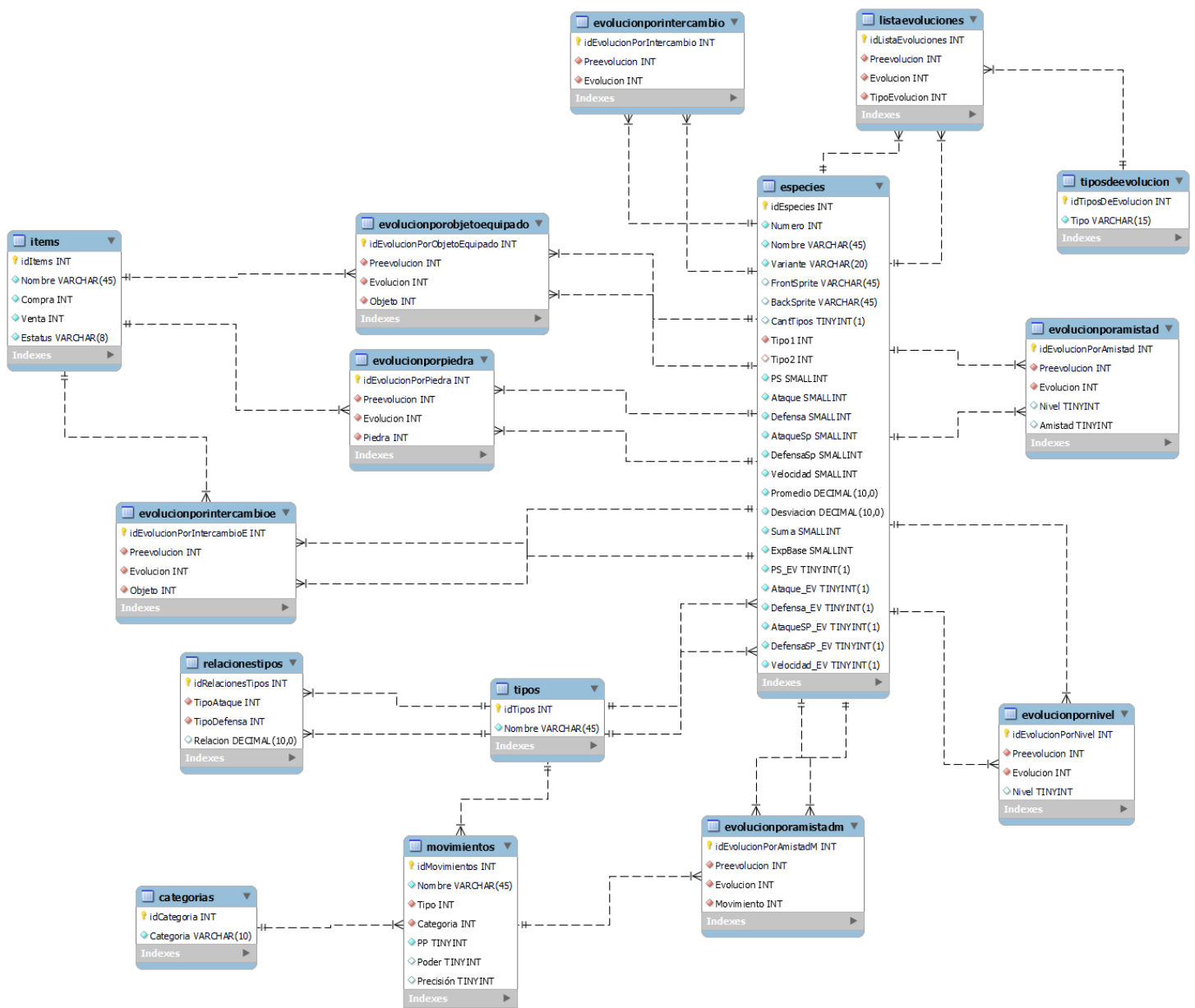
Con lo visto hasta ahora en el curso deberías ser capaz de esquematizar bases de datos complejas y tan grandes como sea necesario. Por supuesto, no hemos visto herramientas de optimización, pero asimismo tu capacidad ya está ahí. En este proyecto nos enfocaremos justamente en validar este entendimiento de forma que, al intentar hacerlo por ti mismo/a, te enfrentes al reto y, en consecuencia, puedas trabajar con el equipo docente para solventar dudas y fortalecer aquellos aspectos que son necesarios.

OBJETIVO

Trabajar con una base de datos ya existente para agregar más estructuras a ella y generar luego las consultas SQL necesarias para obtener información, tal como se describirá en este documento.

BASE DE DATOS POKÉMON

El/La estudiante ha de haber generado, tanto en **MySQL** como en **SQLite**, la base de datos basada en los videojuegos de Pokémon, tal como ha sido diagramada en las clases. Será a partir de dicho esquema que este proyecto será llevado a cabo. A continuación se muestra el diagrama visto en las clases:



Si bien se ha mencionado que este diagrama puede ser mejorado, lo cual es cierto, de momento basaremos nuestra práctica en él, ya que es perfecto para trabajar con consultas avanzadas y el entendimiento de un modelo de base de datos relacional complejo.

CREACIÓN DE NUEVAS TABLAS

Lo que hemos diagramado en la base de datos es, entre otras cosas, la lista de especies Pokémon existentes, cada una de las cuales posee un montón de atributos o campos de información. Ahora bien, dentro de estos videojuegos aparecen Pokémon salvajes y de otros entrenadores contra los cuales se puede batallar y/o capturar (si son salvajes):



En el ejemplo de la izquierda puedes ver que apareció un Poochyena salvaje, el cual es de nivel 2, extremadamente bajo. En nuestra base de datos vemos que la especie Poochyena tiene ID 292, así que podrás encontrar allí todos los datos relacionados con ella. Sin embargo aquí tenemos a un individuo de nivel 2 de dicha especie, el cual tendrá sus propios atributos, que serían diferentes si su nivel fuera 50 (de hecho el nivel es un atributo).

A la derecha puedes ver al Pokémon de la especie Treecko (que en el ejemplo tiene como nombre Lippy, es como tener un perro Pastor Alemán y llamarlo Lippy). Se ve en esa pantalla que Lippy tiene Nivel 5, luego se indica que sus puntos de salud son de 20 sobre 20 (si el Pokémon es dañado en batalla su salud bajará, pero nunca será más de 20 hasta que suba de nivel). Luego tienes los puntos de ataque (12), defensa (8),



ataque especial (12), defensa especial (11), velocidad (13), puntos de experiencia (152) y puntos faltantes para subir de nivel (27). Además se muestra qué objeto tiene equipado y qué cintas (en este caso no tiene nada). A efectos de esta práctica no diferenciaremos entre objetos y cintas. Finalmente vemos a la izquierda los ataques que el Pokémon conoce y cuántos PP restantes le quedan de cada uno.

Por tanto se hace necesario crear una estructura que permita tener los individuos Pokémon que se van generando en el juego ya que los datos de cada uno son independientes y van modificándose en el tiempo. Así, se hace necesario crear una tabla con los siguientes campos:

Tabla Pokémon	
Campo	Descripción
IdPokemon	El ID del registro. Debe ser autoincremental. Clave primaria.
Especie	La especie a la cual pertenece este individuo. No puede ser nulo.
Mote	El nombre que se le ha dado al Pokémon. Puede ser NULL.
Nivel	El nivel actual. No será nunca mayor a 255. No puede ser nulo.
Salud máxima	Lo máximo de salud que se puede tener en el nivel actual. No nulo.
Salud actual	Los puntos de salud actuales. No nulo.
Ataque	La capacidad de ataque. No será nunca mayor a 255. No puede ser nulo.
Defensa	Ídem anterior.
Ataque especial	Ídem anterior.
Defensa especial	Ídem anterior.
Velocidad	Ídem anterior.
Puntos de experiencia	La cantidad de experiencia acumulada. No puede ser nulo
Objeto equipado	Referencia a la tabla de ítems equipados. Puede ser NULL.
Estado	Referencia a la tabla de estados. Puede ser NULL.

El último campo, llamado Estado, indica que es una referencia a la tabla de **estados**, la cual debe ser creada como sigue:

Tabla Estados	
Campo	Descripción
idEstados	El ID del registro. No debe ser autoincremental. Clave primaria.
Estado	El nombre del estado.

Esta tabla, al ser de consultas con datos que no variarán en el tiempo, debe ser completada con la siguiente información:

Tabla Estados	
idEstados	Estado
1	Paralizado
2	Quemado
3	Envenenado
4	Gravemente envenenado
5	Dormido
6	Congelado
7	Pokérus
8	Debilitado

Estos son los 8 estados posibles que existen. Su uso es exclusivo para las mecánicas del juego. Por ejemplo, un Pokémon **paralizado** tendrá muchas posibilidades de no poder atacar en cada turno; uno **quemado** o **envenenado** perderá puntos de salud automáticamente en cada turno; uno **dormido** no podrá atacar, entre muchas otras cosas.

Finalmente se hace necesario tener una tabla que sirva para registrar los ataques que un Pokémon ha aprendido, por tanto se ha de unir al Pokémon específico con el ataque concreto, así como registrar el PP actual y máximo. Por tanto, la estructura de la tabla debería ser como la que sigue a continuación:

Tabla Ataques	
Campo	Descripción
idAtaques	El ID del registro. Debe ser autoincremental. Clave primaria.
Pokemon	Referencia al Pokémon que ha aprendido dicho ataque. No nulo
Movimiento	Referencia a la tabla de movimientos. No nulo.
PP máximo	La cantidad máxima de PP para este ataque.
PP actual	La cantidad de PP actuales disponibles para usar.

ENTREGA DE CREACIÓN DE TABLAS

Lo primero que debes crear para entregar con este proyecto son las sentencias MySQL y SQLite necesarias para generar las tablas descritas anteriormente. En concreto:

- ✓ Sentencia para crear la tabla de Estados.
- ✓ Sentencia para insertar en la tabla de Estados los valores indicados para ella.
- ✓ Sentencia para crear la tabla de Pokémon.
- ✓ Sentencia para crear la tabla de Ataques.

Como se mencionó, escribe las instrucciones para MySQL y SQLite, generando así dos archivos distintos. Nosotros deberíamos poder copiar y pegar tu código SQL en los editores de **MySQL Workbench** y **SQLite Studio** de forma que al ejecutarlos se generaran las tablas y los datos automáticamente sin mayor esfuerzo de nuestra parte.

CREACIÓN DE CONSULTAS

Teniendo la estructura completada insertaremos algunos datos ficticios para hacer pruebas. Así, deberás insertar al menos 10 registros en la tabla de Pokémon. Por ejemplo:

id	Especie	Mote	Nvl	PSmX	PS	Atk	Def	AtkE	DefE	Vel	Exp	Obj	Est
1	283	Lippy	5	20	20	12	8	12	11	13	152	NULL	NULL
2	27	Pika	10	30	30	20	16	16	18	26	523	210	NULL
3	179	Unico	50	181	181	143	110	156	110	150	2600	603	1

En este ejemplo se ingresó a Treecko (283), Pikachu (27) y Mewtwo (179), o mejor dicho, un Treecko, un Pikachu y un Mewtwo; bien podríamos registrar más, cada uno con sus atributos individuales. En el caso de Treecko no tiene ningún objeto ni estado asociados; Pikachu tiene el objeto **Electrizador** (210) y ningún estado; Mewtwo tiene el objeto **Restos** (603) y el estado **Paralizado** (1). En tu caso deberías ingresar más registros, sin importar demasiado qué especie referencias, ni qué atributos pones. Lo que sí importa es que incluyas registros sin referenciar objetos ni estados (como hice con Lippy), otros solo con un objeto o un estado (como Pika) y otros con ambas referencias. Esto es importante por cuestiones que veremos más adelante. Finalmente veamos un ejemplo de relleno de datos para la tabla de Ataques:

id	Pok	Mov	PPMx	PPAc
1	1	1	35	35
2	2	39	30	30
3	1	43	30	30
4	3	94	10	10
5	2	84	30	30
6	1	71	25	25

He puesto tres ataques para Lippy (referencia con número 1 en la columna Pok), dos ataques para Pika (referencia con número 2 en la columna Pok) y un ataque para Unico (referencia con número 3). Las referencias están desordenadas porque nunca se sabe en qué orden serán agregados los registros. De pronto Pikachu aprende un ataque nuevo, luego Mewtwo y después otra vez Pikachu. Incluso podría pasar que un Pokémon no sepa ningún ataque.

Teniendo entonces algunos datos en las tablas, más la información que te brindamos para las importaciones de todas las tablas anteriormente creadas en las clases, tu principal tarea será generar las siguientes sentencias de consulta para poder visualizar:

- Una tabla con nombres de especie, mote y nivel de cada pokemon:

Nombre	Mote	Nivel
Treecko	Lippy	5
Pikachu	Pika	10
Mewtwo	Unico	50

- Una tabla que incluya el nombre de la especie, el mote del pokemon, los ataques que aprendió y el valor de PP actual para cada ataque. Con la información llenada anteriormente tendrías algo como lo que se muestra debajo:

Especie	Mote	Ataque	PP Actual
Treecko	Lippy	Pound	35
Treecko	Lippy	Leer	30
Treecko	Lippy	Absorb	25
Pikachu	Pika	Tail Whip	30
Pikachu	Pika	Thunder Shock	30
Mewtwo	Unico	Psychic	10

Si en tu tabla **Pokemon** incluyes algún registro de individuo que no ha aprendido ningún movimiento, es decir, que no haya ni un registro en la tabla **Ataques** para ese Pokémon concreto, no lograrás que aparezca dicho individuo en la consulta anterior. En un caso como ese ya se necesitaría conocer otras formas de consultar las tablas.

- Una tabla como la anterior pero que además incluya el estado del Pokémon. Para los datos cargados en estos ejemplos obtendrías algo así:

Especie	Mote	Ataque	PP Actual	Estado
Mewtwo	Unico	Psychic	10	Paralizado

En este caso ocurre claramente lo que se comentó para el caso anterior: no aparecen en los resultados los Pokémon que no tienen un estado asociado. Para solucionar este problema se requiere conocer otros tipos de consultas llamadas JOIN. Otra solución es no permitir que hayan calves foráneas nulas (NULL), por ejemplo, definiendo un **Estado** llamado 'Ninguno' para que sí o sí exista siempre una referencia a la tabla de estados.

- Una tabla conteniendo el nombre de la especie, el mote del pokemon y el primer tipo:

Especie	Mote	Tipo
Treecko	Lippy	Planta
Pikachu	Pika	Eléctrico
Mewtwo	Unico	Planta

Si en esta consulta se incluyera el segundo tipo de cada especie obtendrías un resultado vacío, porque ninguno de los pokemon listados tiene un segundo tipo, solo tienen uno.

- Una tabla que contenga el nombre de cada especie Pokémon y su primer tipo (Tipo 1), así como también veremos la variante que contenga ambos tipos y un conteo básico:

Especie	Tipo 1
Abra	Planta
Aerodactyl	Roca
Aipom	Normal
Alakazam	Planta
Ampharos	Eléctrico
Arbok	Veneno
Arcanine	Fuego
Ariados	Bicho
Articuno	Hielo
Articuno	Planta
Azumarill	Agua
Azurill	Normal
Bayleef	Planta
Beautifly	Bicho
Beedrill	Bicho
Bellossom	Planta
Bellsprout	Planta
Blastoise	Agua
Blaziken	Fuego
Blissey	Normal
Breloom	Planta
Bulbasaur	Planta
Butterfree	Bicho
Cascoon	Bicho

Evidentemente que la lista resultante quedará mucho más larga. Nótese que los resultados están ordenados por el nombre de la especie de forma ascendente, lo cual tú debes lograr también. Esta consulta es extremadamente simple, sin embargo te pregunto ¿qué ocurre si quieres incluir el segundo tipo para obtener algo así?:

Especie	Tipo 1	Tipo 2
Aerodactyl	Roca	Volador
Ariados	Bicho	Veneno
Articuno	Hielo	Volador
Articuno	Planta	Volador
Azumarill	Agua	Hada
Azurill	Normal	Hada
Beautifly	Bicho	Volador
Beedrill	Bicho	Veneno
Bellsprout	Planta	Veneno

Lo primero que ocurre en este resultado es que perderemos todas las especies que no tengan un segundo tipo asociado, sin embargo ese no es el problema aquí, sino que tendrás un problema al intentar generar esta tabla. Como parte de este proyecto

deberás, en primer lugar, darte cuenta de cuál es la dificultad de esta segunda consulta donde queremos incluir el segundo tipo de cada especie, y en segundo lugar, deberás INVESTIGAR y solventar dicha dificultad para lograr este resultado. Finalmente genera el SQL necesario para contar la cantidad de especies por su primer tipo:

Tipo 1	Cantidad
Acero	3
Agua	54
Bicho	32

El resultado final contendrá todos los tipos existentes y estará ordenado alfabéticamente por el nombre de cada tipo existente.

- Genera una consulta SQL que muestre la lista de pokemon registrados en la tabla de pokemon indicando el nombre de la especie, el mote y la evolución inmediata. Por ejemplo, la especie Treecko evoluciona en Grovyle al nivel 16, así que es una evolución por nivel. Tendrás que generar una consulta diferente por cada tipo de evolución existente, ya que no podrás agrupar todo en un mismo resultado. En cada caso deben aparecer los campos mencionados más todos los relativos al tipo de evolución concreto. Por ejemplo:

Pokemon	Evolucion	Nivel
Treecko	Grovyle	16

Pokemon	Evolucion	Piedra
Pikachu	Raichu	Piedra trueno

En estos dos ejemplos tenemos un resultado por cada consulta ya que con los datos ingresados anteriormente no da para más. Incluso, como Mewtwo no tiene evolución pues no aparecerá en ningún resultado. Con los datos dados en los archivos de importación tampoco tendrás demasiado ya que no están completos, pero te servirán para entender el mecanismo y generar todas las consultas SQL necesarias para todos los tipos de evolución.

ENTREGA DE CONSULTAS

Además de las sentencias SQL para crear las tablas antes descritas, incluirás también las sentencias INSERT con los datos que tú hayas agregado a cada tabla, de forma tal que al ejecutarlas nosotros tengamos dicha información incluida en nuestra base de datos.

Hecho eso, deberás entregar todas las sentencias SQL necesarias para generar los resultados descritos así como la solución al problema de los tipos 1 y 2, ya que esto se hará presente también con los campos Preevolucion y Evolucion de las tablas que consultarás para listar las evoluciones de cada especie. La dificultad está en que hay tablas dos claves foráneas a un mismo campo de una misma tabla. Por ejemplo, la tabla Especies tiene los campos Tipo1 y Tipo2 que referencian, ambos, al ID de la tabla tipos. La tabla ListaEvoluciones tiene los campos Preevolucion y Evolucion que referencian ambos al ID de la tabla Especies. Eso complica un poco las cosas.

FINALIZACIÓN

Debes entregar los siguientes archivos para aprobar este proyecto:

1. **Creación MySQL**, el cual debe contener:
 - a. Sentencias de creación de las tablas Estados, Pokemon y Ataques.
 - b. Sentencias de inserción de datos en todas las tablas.
2. **Creación SQLite**: ídem anterior pero para SQLite.
3. **Consultas SQL**: un solo archivo conteniendo todas las sentencias SQL con las consultas descritas anteriormente para obtener los resultados indicados. No hay variación entre MySQL y SQLite para dichas sentencias por lo que con un solo archivo basta.

Envía estos archivos al correo electrónico bedelia@kaedusoft.edu.uy. Si tienes dudas o consultas puedes evacuarlas a dicha dirección de correo o bien al WhatsApp o Telegram +598 94 815 035.