{desafío}
latam_

Base de datos relacionales _

Parte II

Realizando consultas

Competencias

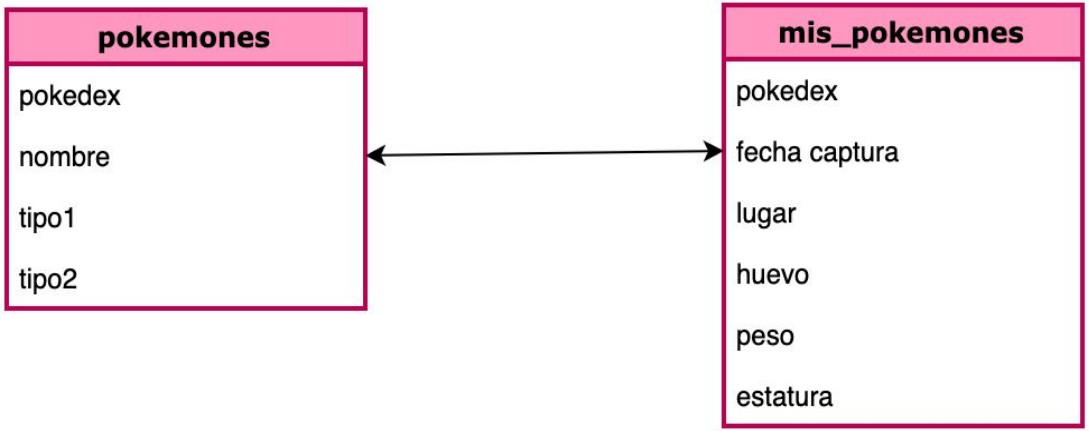
- Implementar alias en consultas SQL para un manejo personalizado de los campos y las tablas involucradas.
- Reconocer las funciones básicas que se pueden usar en consultas para obtener datos calculados.
- Realizar consultas con funciones a las tablas de la base de datos.
- Crear índices en las tablas para agilizar las consultas.

.

Creando las tablas para los Pokemones

Supongamos que estamos creando un videojuego con la temática de pokémon, donde debemos registrar en nuestra base de datos los pokemones que se usarán en el juego. Cada pokémon debe ser identificable por un número definido por la pokédex y la trama original de la serie. El videojuego tiene un protagonista maestro pokémon y debe hacerse un registro por cada pokémon capturado y los datos de esta acción.

Creando las tablas para los Pokemones



Importando datos de un .csv

Un archivo CSV es un archivo de texto que almacena datos en forma de columnas, separadas generalmente por comas.

El flujo con el cual podemos ingresar una serie de registros alojados en un .csv es el siguiente:

1. Abrir el archivo en un editor de texto, de recomendación Visual Studio code o Atom
2. Si las filas están encerradas entre comillas, elimínelas
3. Crear la tabla a la cual desea agregar el archivo csv.

Cargando la data

- Para importar la información procede con el siguiente comando.
- Se dejó header como parámetro, ya que si revisamos el archivo tiene la primera fila con los nombres de las columnas, y haremos lo mismo para ambas tablas.

```
\copy pokemones FROM 'directorio/pokemonesKanto.csv' csv header;
```

Consultando tablas

- Ver una columna en específico, la sintaxis es la siguiente:

```
SELECT columna1 FROM nombre_tabla;
```

- Ver más de una columna

```
SELECT columna1, columna2 FROM
nombre_tabla;
```

- Seleccionar todos los datos de la tabla:

```
SELECT * FROM nombre_tabla;
```

- Para ver una fila en específico:

```
SELECT columna FROM tabla WHERE
condicion;
```

Operadores condicionales

Operador	Descripción
=	Igual
>	Mayor que
<	Menor que
<> , !=	Distinto. El operador != es solo válido en algunas versiones de SQL
>=	Mayor o igual
<=	Menor o igual
BETWEEN	Entre cierto rango
LIKE	Por patrón
IN	Para especificar múltiples valores posibles para una columna

Ejercicio propuesto

Utilizar el mismo ejercicio de esta lectura pero realiza las siguientes consultas:

1. Listar los pokemones cuya columna pokédex sea mayor a 50.
2. Listar pokemones de tipo psíquico.
3. Listar los primeros 22 registros de la tabla pokemones.

Alias

Se aplica cuando se quiere dar un nombre más pequeño o más fácil de usar, cuando este se utilizará varias veces.

Sintaxis:

```
SELECT
nombre_tabla_referencia.nombre_columna as nombre_referencia
FROM nombre_tabla as nombre_tabla_referencia;
```

Ejercicio propuesto

Realizar una consulta a la tabla pokemones donde la columna nombre tenga un alias “pokename” y pokédex tenga el alias “nro_pokedex”. La consulta debe obtener como límite 30 registros.

Funciones en consultas

MIN	Entrega el mínimo de los datos de una columna.
MAX	Entrega el máximo de los datos de una columna.
LENGTH	Calcula el largo de los datos en una columna.
COUNT	Cuenta la cantidad de ocurrencias de las filas.
SUM	Suma los valores de una columna ignorando los valores null.

Ejercicio propuesto

1. Realizar una consulta para saber cuántos registros se tienen en la tabla pokemones. Ocupa el comando COUNT para esto.
2. Realizar una consulta para saber cuantos pokemones tienen como segundo tipo, el tipo roca en la tabla pokemones.

Agrupación con GROUP BY

- La instrucción GROUP BY agrupa filas que tienen los mismos valores en filas de resumen, como "buscar el número de clientes en cada país".
- Se utiliza con funciones agregadas (COUNT, MAX, MIN, SUM) para agrupar el conjunto de resultados por una o más columnas.

```
SELECT columna1, columna2, columna3
FROM tabla
GROUP BY columna1;
```

Ejercicio propuesto

Realizar una consulta SQL que devuelva los pokemones de la tabla pokemones agrupados por la columna tipo2.

Ordenamiento

- Se utiliza para ordenar filas según los valores que tengan en algunas columnas.
- Ejemplo común sería ordenar de menor a mayor.
  - DESC: Orden decreciente.
  - ASC: Orden ascendente.

```
SELECT columna1, columna2, ...
FROM nombre_tabla
[WHERE condiciones]
ORDER BY columna2 [DESC | ASC];
```

Ejercicio propuesto

Realizar una consulta a la tabla pokemones para obtener los pokemones ordenados según su id de pokédex de forma descendiente.

Creando índices

Cuando creamos un índice en una tabla de la base de datos, lo que hacemos es agilizar el tiempo de respuesta de esta búsqueda pues preparamos al motor de base de datos para próximas búsquedas relacionadas a las columnas que tengan índices.

Sintaxis:

```
CREATE INDEX nombre_indice on nombre_tabla(nombre_columna);
```

Ejercicio propuesto

Agregar a la columna pokédex un índice y verificar que este fue creado.

Eliminando índices

Si queremos eliminar un índice, basta con ejecutar el comando:

```
drop index nombre_indice;
```

Ejercicio propuesto

Elimina el índice que le has creado a la columna pokedex y verifica que fue eliminado.

Ejercicio guiado: Pongamos a prueba los conocimientos

La empresa japonesa Mawashi Cars Spa hará una reinauguración muy pronto y ha decidido dejar de registrar todos los movimientos en excel y empezar a usar el software que compró el año pasado pero nunca estrenó, sin embargo, cuando lo intentó utilizar se llevó la sorpresa de que la aplicación necesita conectarse a una base de datos. La empresa se contactó con un desarrollador para esto y le envió un archivo en formato csv que exportaron de excel con unos pocos registros de autos para que proceda con la creación de la base de datos y pueda hacer las pruebas que necesite. En el apoyo lectura de esta sesión encontrarás el fichero .csv con la data de los autos.

Luego del levantamiento de requerimientos que le hizo el desarrollador a la empresa se concluyó que necesitaremos crear las siguientes tablas con los siguientes campos:

Manos al código - Pongamos a prueba los conocimientos

id	Marca	Modelo	Año	Color

Modelo de la tabla Autos

Fecha	id_auto	Cliente	Referencia	Cantidad

Modelo de la tabla Ventas

Manos al código - Pongamos a prueba los conocimientos

Las pruebas a realizar deben ser:

1. Creación de la base de datos.
2. Conexión con la base de datos.
3. Crear las tablas Autos y Ventas enlazadas por claves primaria-foranea.
4. Importar el archivo autos.csv en la tabla Autos.
5. Verificar la carga exitosa de la data en la tabla Autos.
6. Hacer una consulta con Alias.
7. Realizar 2 inserciones a la tabla Ventas.
8. Realizar una consulta con la función SUM.
9. Agregar una columna a una tabla.
10. Agregar un registro a la tabla Autos.
11. Hacer una actualización de información a una tabla.
12. Agrupar columnas en una tabla.
13. Ejecutar una consulta con ORDER BY.
14. Agregar dos índices a dos columnas.
15. Eliminar un índice a una columna.

{desafío} latam_

Academia de talentos digitales

www.desafiolatam.com