

# **Sistema de Consulta SQL con LLM**

## **Integrantes**

- 📧 Matías Irala
- 📧 Richar Núñez

## **Resumen**

El siguiente trabajo presenta la implementación de un sistema de consulta en lenguaje natural para bases de datos SQL. Utiliza un modelo de lenguaje de gran escala (LLM) para traducir automáticamente preguntas formuladas en lenguaje natural a consultas SQL válidas. El estudio toma como caso de estudio la base de datos Chinook y Llama 4.

El objetivo es facilitar el acceso a la información contenida en la base de datos sin requerir conocimientos técnicos en SQL por parte del usuario. Esta solución combina procesamiento de lenguaje natural con ejecución de consultas estructuradas, brindando una interfaz más accesible e intuitiva para la exploración de datos.

## **1. Planteamiento del problema**

El acceso a bases de datos relacionales como SQLite requiere habitualmente conocimientos específicos de SQL, lo que representa una barrera significativa para usuarios no técnicos que desean consultar y analizar información almacenada en estos sistemas. Esta limitación restringe el aprovechamiento de datos por parte de profesionales de otras áreas, como analistas de negocios, periodistas o investigadores, que podrían beneficiarse del acceso directo a la información sin depender de intermediarios técnicos.

Con el avance de los modelos de lenguaje natural (LLM), surge la posibilidad de traducir automáticamente preguntas formuladas en lenguaje humano a consultas SQL ejecutables. Sin embargo, esta tarea implica varios desafíos, como la correcta interpretación de la intención del usuario, la generación de consultas sintácticamente válidas y semánticamente relevantes, y la necesidad de asegurar que las respuestas proporcionadas sean precisas y seguras.

En este contexto, se plantea la necesidad de diseñar e implementar un modelo que permita a los usuarios interactuar con bases de datos relacionales mediante lenguaje natural, aprovechando las capacidades de los modelos de lenguaje para reducir la barrera técnica del acceso a los datos. El presente trabajo se enfoca en abordar este problema utilizando la base de datos Chinook como caso de estudio, apoyado en el modelo Llama 4.

## **2. Descripción del corpus (base de datos)**

Para este estudio se utilizó la base de datos Chinook, una base de datos relacional en formato SQLite ampliamente utilizada en contextos educativos y de pruebas.

Chinook simula un entorno comercial de una tienda digital de música, similar a plataformas como iTunes, e incluye un conjunto representativo de datos que abarcan clientes, facturas, empleados, pistas musicales, álbumes, artistas y géneros.

La base de datos contiene múltiples tablas interrelacionadas, lo que la convierte en un recurso valioso para evaluar la capacidad de los modelos de lenguaje para generar consultas SQL sobre estructuras complejas. Chinook está compuesta por datos en inglés y su esquema refleja una organización relacional típica, con claves primarias, claves foráneas y restricciones de integridad.

### **3. Metodología**

La metodología implementada en este trabajo consiste en un sistema integrado para traducir preguntas en lenguaje natural a consultas SQL ejecutables, utilizando un enfoque basado en recuperación aumentada con LLM. El sistema sigue un flujo estructurado que combina múltiples técnicas para mejorar la precisión de las traducciones.

A continuación, se detallan los principales componentes y etapas del sistema:

#### **❑ Infraestructura base y conexión a datos**

Se establece una conexión a una base de datos SQLite (Chinook) que contiene información sobre música, artistas, álbumes y ventas

Se utilizó el modelo Llama 4 (meta-llama/llama-4-scout-17b-16e-instruct), accedido mediante el proveedor Groq, por su capacidad de procesamiento eficiente y respuestas contextualizadas en tareas de generación de texto. Este modelo sirvió como núcleo para interpretar preguntas en lenguaje natural y generar consultas SQL correspondientes.

#### **❑ Arquitectura de recuperación aumentada**

La arquitectura implementa un enfoque RAG (Retrieval-Augmented Generation) con dos componentes principales de recuperación:

- Recuperación de ejemplos SQL similares:
  - Se creó un corpus de pares pregunta-consulta SQL (ejemplos few-shot) que cubren diversos escenarios de consulta.
  - Estos ejemplos se vectorizan utilizando embeddings de HuggingFace (sentence-transformers/all-mpnet-base-v2).
  - Se implementa un índice FAISS para recuperar eficientemente los ejemplos más similares a la nueva pregunta.
  - Se utiliza create\_retriever\_tool para convertir este recuperador en una herramienta utilizable por el agente, el tool es nombrado example\_retriever\_tool.
- Recuperación de entidades nombradas:

- Se extrae y vectoriza un vocabulario de entidades específicas del dominio (nombres de artistas y álbumes).
- Un segundo índice FAISS permite identificar y normalizar referencias a estas entidades en las consultas del usuario.
- Se crea otra herramienta de recuperación con `create_retriever_tool` para buscar nombres propios, el tool es nombrado `retriever_tool`.

#### ❓ **Extensión del toolkit SQL:**

- Se inicializa un `SQLDatabaseToolkit` estándar que proporciona herramientas básicas para interactuar con la base de datos.
- Este toolkit se extiende añadiendo las herramientas de recuperación personalizadas (`retriever_tool` y `example_retriever_tool`) al conjunto de herramientas disponibles.
- La extensión permite que el agente alterne fluidamente entre operaciones de base de datos y consultas semánticas al contexto y entidades.

#### ❓ **Proceso de ejecución:**

El sistema sigue un flujo estructurado de pasos para procesar cada consulta:

1. Recibe una pregunta en lenguaje natural.
2. Obtiene la lista de tablas disponibles en la base de datos usando herramientas del toolkit SQL.
3. Consulta el esquema de las tablas relevantes.
4. Recupera ejemplos SQL similares a la pregunta actual usando la herramienta de recuperación personalizada.
5. Identifica posibles entidades nombradas en la consulta mediante la herramienta de búsqueda de nombres propios.
6. Genera una consulta SQL válida utilizando el modelo LLM, guiado por un prompt de sistema específico.
7. Ejecuta la consulta en la base de datos.
8. Formatea y presenta los resultados al usuario.

#### ❓ **Integración mediante ReAct Agent:**

Se utiliza la arquitectura `create_react_agent` de `LangGraph` para orquestar las diferentes herramientas.

El agente implementa un patrón de razonamiento y acción (ReAct) que permite al modelo planificar los pasos necesarios, ejecutar herramientas adecuadas en cada fase y razonar sobre los resultados.

El prompt de sistema guía al agente para seguir un orden específico de operaciones, mejorando la consistencia y calidad de las consultas generadas.

#### ❓ **Procesamiento de resultados y debugging interno:**

Se diseñó una función auxiliar para transformar los resultados de las consultas SQL en listas limpias y legibles. También se activaron mecanismos de depuración interna para inspeccionar la generación de consultas y evaluar la lógica de recuperación utilizada por el modelo.

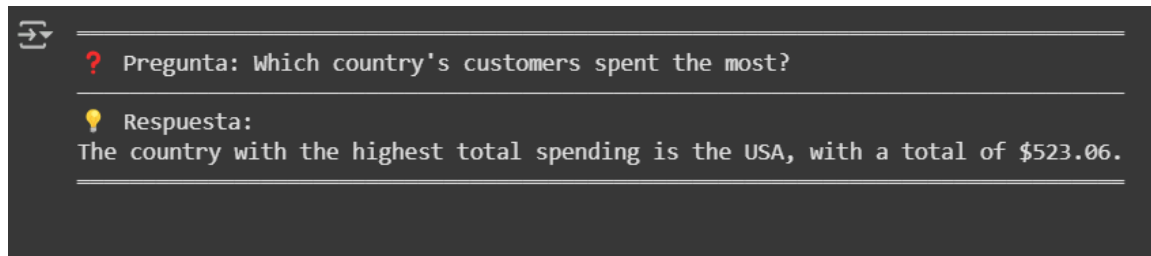
#### **Interfaz de consulta final:**

Se definió una función de interacción que permite al usuario formular preguntas en lenguaje natural. Esta función coordina la recuperación de información contextual (ejemplos o nombres propios), la generación de la consulta SQL por parte del modelo, su ejecución sobre la base de datos y la presentación del resultado final al usuario.

### **4. Evaluación de resultados**

Para evaluar la efectividad del sistema propuesto, se realizaron varias consultas en lenguaje natural que cubren distintos tipos de operaciones sobre la base de datos Chinook: agregación, filtrado y recuperación por entidad específica. A continuación, se presenta un análisis cualitativo de los resultados obtenidos:

#### **Pregunta 1:**




#### **# Solución Humana:**

```
db.run("SELECT Country, SUM(Total) AS TotalPurchase FROM Invoice JOIN  
Customer ON Invoice.CustomerId = Customer.CustomerId GROUP BY Country  
ORDER BY TotalPurchase DESC LIMIT 1")  
[('USA', 523.0600000000003)]
```

La consulta generada es semánticamente correcta, y el resultado coincide con el valor esperado según la lógica del dominio. El modelo entendió correctamente la intención de comparar países según su gasto acumulado.

#### **Pregunta 2:**



**Pregunta:** List all albums by AC/DC

---

**Respuesta:**  
The albums by AC/DC are:


1. For Those About To Rock We Salute You
2. Let There Be Rock

# Solución Humana:

```
db.run("SELECT * FROM Album WHERE ArtistId = (SELECT ArtistId FROM Artist WHERE Name = 'AC/DC')")
[(1, 'For Those About To Rock We Salute You', 1), (4, 'Let There Be Rock', 1)]
```

La respuesta es precisa y completa. El modelo reconoció correctamente el nombre de un artista y extrajo información asociada a él, lo cual implica una recuperación semántica exitosa y uso adecuado de claves foráneas.

### **Pregunta 3:**



**Pregunta:** What is the average duration of tracks in the Rock genre?

---


**Respuesta:**  
The average duration of tracks in the Rock genre is approximately 283.91 seconds or 4.73 minutes.

# Solución Humana:

```
db.run("SELECT AVG(Milliseconds) FROM Track WHERE GenreId = (SELECT GenreId FROM Genre WHERE Name = 'Rock')")
[(283910.0431765613,)]
```

El resultado es coherente y correctamente presentado, con conversión de unidades incluida, lo cual indica una capacidad del modelo para interpretar contexto y enriquecer la respuesta de forma útil para el usuario.

### **Pregunta 4:**



**Pregunta:** Which artist has the most albums in the database?

---

**Respuesta:**  
The artist with the most albums in the database is Iron Maiden with 21 albums.

# Solución Humana:

```
db.run("SELECT ar.Name AS ArtistName, COUNT(al.AlbumId) AS AlbumCount
```

```
FROM Artist ar JOIN Album al ON ar.ArtistId = al.ArtistId GROUP BY  
ar.Name ORDER BY AlbumCount DESC LIMIT 1;")  
[('Iron Maiden', 21)]
```

Consulta correcta y resultado válido. La solución demuestra dominio de jerarquías.

Los resultados muestran un alto grado de precisión tanto en la comprensión semántica de las preguntas como en la formulación de consultas SQL. En los cuatro casos:

- ☐ Las respuestas finales fueron comprensibles, completas y, cuando fue necesario, enriquecidas con conversiones de unidades o formatos amigables.
- ☐ El sistema demostró habilidad para trabajar con relaciones entre tablas, filtros específicos, funciones de agregación y ordenamientos complejos.

Aunque esta evaluación es de carácter cualitativo, se observa que el modelo logra traducir intenciones del lenguaje natural en operaciones SQL con un alto grado de fidelidad. Esto refuerza la viabilidad del enfoque basado en modelos de lenguaje y recuperación contextual como herramienta para democratizar el acceso a datos estructurados.

## 5. Conclusiones y recomendaciones

Este trabajo demostró la viabilidad de integrar un modelo de lenguaje con una base de datos relacional para permitir consultas en lenguaje natural. La solución desarrollada permite que usuarios sin conocimientos técnicos en SQL accedan a información compleja de forma intuitiva y eficiente. A través de ejemplos prácticos, se validó que el modelo fue capaz de:

- ☐ Interpretar correctamente preguntas en lenguaje natural.
- ☐ Generar consultas SQL sintáctica y semánticamente válidas.
- ☐ Entregar respuestas claras y precisas, incluso en casos que involucran agregaciones, filtros y relaciones múltiples entre tablas.

Además, la incorporación de técnicas como few-shot learning y el uso de un sistema de recuperación basado en vectores (FAISS) mejoró significativamente la precisión de las respuestas, facilitando una traducción más contextualizada y específica.

De este modo se podría ampliar el set de ejemplos de few-shot learning para cubrir un espectro más amplio de estructuras y operaciones SQL, mejorando así la generalización del modelo ante consultas nuevas, explorar otras bases de datos con mayor complejidad y volumen para evaluar la escalabilidad de la solución.

Desplegar la solución como servicio web o asistente interactivo, facilitando su uso en contextos educativos, empresariales o analíticos sin necesidad de entorno local.