

Daniel de la Mata Lara

Para KSchool
8ª Edición Analítica Digital | Online
SQL

Lo que la IA te dice que aprendas

Noviembre 12, 2024

Objetivo

Queremos saber si lo que tratamos de enseñar está alineado con lo que la IA generativa más famosa nos indica.

Para ello, vamos a preguntar a Chat GPT qué conceptos son básicos en SQL. Tiraremos un poco del hilo y le preguntaremos además por conceptos genéricos de Bases de Datos y más concretamente por Google BigQuery.

SQL

Pregunta

¿Me das una lista de 15 comandos que consideras básicos para enseñar SQL?

Respuesta

¡Claro! Aquí tienes una lista de 15 comandos básicos de SQL que son fundamentales para principiantes:

1. **SELECT** - Extrae datos de una tabla.

```
SELECT column_name FROM table_name;
```

2. **WHERE** - Filtra registros en una consulta según una condición.

```
SELECT column_name FROM table_name WHERE condition;
```

3. **INSERT INTO** - Inserta nuevos datos en una tabla.

```
INSERT INTO table_name (column1, column2) VALUES (value1, value2);
```

4. **UPDATE** - Modifica datos existentes en una tabla.

```
UPDATE table_name SET column_name = value WHERE condition;
```

5. **DELETE** - Elimina registros de una tabla.

```
DELETE FROM table_name WHERE condition;
```

6. **ORDER BY** - Ordena los resultados de una consulta.

```
SELECT column_name FROM table_name ORDER BY column_name [ASC|DESC];
```

7. **GROUP BY** - Agrupa los datos para aplicar funciones de agregado.

```
SELECT column_name, COUNT(*) FROM table_name GROUP BY column_name;
```

8. **HAVING** - Filtra grupos de datos después de aplicar GROUP BY.

```
SELECT column_name, COUNT(*) FROM table_name GROUP BY column_name HAVING  
COUNT(*) > 1;
```

9. JOIN - Combina filas de dos o más tablas basadas en una relación entre columnas.

```
SELECT table1.column_name, table2.column_name  
  
FROM table1  
  
JOIN table2 ON table1.common_column = table2.common_column;
```

10. INNER JOIN - Devuelve filas con coincidencias en ambas tablas.

```
SELECT * FROM table1 INNER JOIN table2 ON table1.column_name =  
table2.column_name;
```

11. LEFT JOIN - Devuelve todas las filas de la tabla izquierda y las coincidencias de la tabla derecha.

```
SELECT * FROM table1 LEFT JOIN table2 ON table1.column_name =  
table2.column_name;
```

12. RIGHT JOIN - Devuelve todas las filas de la tabla derecha y las coincidencias de la tabla izquierda.

```
SELECT * FROM table1 RIGHT JOIN table2 ON table1.column_name =  
table2.column_name;
```

13. UNION - Combina los resultados de dos o más SELECT en un solo conjunto de resultados.

```
SELECT column_name FROM table1  
  
UNION  
  
SELECT column_name FROM table2;
```

14. DISTINCT - Selecciona valores únicos para evitar duplicados.

```
SELECT DISTINCT column_name FROM table_name;
```

15. COUNT, SUM, AVG, MIN, MAX - Funciones de agregado para cálculos en columnas.

```
SELECT COUNT(column_name) FROM table_name;
```

```
SELECT SUM(column_name) FROM table_name;
```

Estos comandos son un buen punto de partida para aprender SQL, ya que cubren las operaciones básicas para trabajar con bases de datos.

Comentario

La lista es bastante completa pero deberíamos haberle pedido que explicara cada comando con mayor profundidad.

Otras palabras reservadas

LIKE, BETWEEN, AS, WITH, AND, OR, OFFSET, LIMIT, CAST, SUBSTRING, CASE-WHEN, IF-THEN-ELSE, CREATE VIEW, CREATE INDEX, DROP, ALTER.

BASES DE DATOS

Pregunta

Y, previo a SQL, ¿qué cosas hay que saber acerca de bases de datos?

Respuesta

Antes de aprender SQL, es fundamental entender algunos conceptos básicos sobre bases de datos. Aquí tienes una lista de temas importantes que ayudan a sentar las bases para comprender cómo funcionan y cómo organizar datos de manera efectiva:

1. Qué es una Base de Datos: Comprender que una base de datos es una colección organizada de datos que se almacenan y administran electrónicamente. Las bases de datos permiten almacenar y recuperar información de forma estructurada.

2. Tipos de Bases de Datos: Conocer los diferentes tipos de bases de datos:

- **Bases de datos relacionales (RDBMS):** Organizan datos en tablas relacionadas y son las más comunes para SQL.
- **Bases de datos no relacionales (NoSQL):** Diseñadas para manejar grandes volúmenes de datos no estructurados, como documentos (MongoDB) o datos clave-valor (Redis).

3. Modelo Relacional: Comprender el modelo de bases de datos relacional, que organiza datos en **tablas** (o relaciones) compuestas por filas y columnas. Este modelo utiliza claves para relacionar tablas y asegurar la integridad de los datos.

4. Tablas, Filas y Columnas: Familiarizarse con las estructuras básicas:

- **Tablas:** Contenedores de datos organizados en columnas y filas.
- **Filas:** Cada fila representa un registro único.
- **Columnas:** Cada columna representa un atributo específico de los datos.

5. Claves Primarias y Extranjeras:

- **Clave Primaria (Primary Key):** Una columna (o conjunto de columnas) que identifica de manera única cada registro en una tabla.

- **Clave Foránea (Foreign Key):** Una columna que crea una relación entre dos tablas, apuntando a la clave primaria de otra tabla.

6. Normalización: Proceso de estructurar los datos en una base de datos para reducir la redundancia y mejorar la integridad de los datos. Incluye formas normales (1NF, 2NF, 3NF, etc.) que definen criterios para organizar los datos.

7. Integridad de Datos:

- **Integridad de entidad:** Asegura que cada fila en una tabla es única y tiene una clave primaria.
- **Integridad referencial:** Mantiene relaciones válidas entre tablas mediante claves foráneas.
- **Integridad de dominio:** Limita los valores posibles de una columna, como restricciones de tipo de datos (enteros, texto, etc.).

8. Transacciones y ACID: Concepto de transacciones como una secuencia de operaciones que se ejecutan como una sola unidad. Las bases de datos transaccionales cumplen con el principio **ACID**:

- **Atomicidad:** Todas las operaciones de una transacción se completan o ninguna se aplica.
- **Consistencia:** La transacción lleva la base de datos de un estado válido a otro.
- **Aislamiento:** Las transacciones se ejecutan de manera aislada sin interferencia de otras.
- **Durabilidad:** Una vez confirmada, una transacción queda guardada permanentemente.

9. Índices: Estructuras que mejoran la velocidad de acceso a los datos, pero que pueden aumentar el tiempo de escritura. Saber cuándo y cómo usar índices puede optimizar el rendimiento de una base de datos.

10. Backups y Recuperación de Datos: Conocer los métodos para hacer copias de seguridad y restauración de la base de datos en caso de fallos, algo crucial para la seguridad y la continuidad del negocio.

11. Roles y Permisos: Controlar el acceso a los datos con privilegios de usuario y roles específicos, asegurando que solo los usuarios autorizados pueden ver o modificar datos.

12. Lenguajes de Bases de Datos:

- **DDL (Data Definition Language):** Define estructuras de la base de datos (e.g., ``CREATE``, ``ALTER``, ``DROP``).
- **DML (Data Manipulation Language):** Manipula los datos almacenados (e.g., ``INSERT``, ``UPDATE``, ``DELETE``).
- **DCL (Data Control Language):** Controla permisos (e.g., ``GRANT``, ``REVOKE``).

Estos conceptos ayudan a construir una base sólida antes de escribir consultas SQL, ya que proporcionan el contexto necesario para comprender cómo interactuar con los datos de manera segura, eficiente y estructurada.

BigQuery

Pregunta

Y, ¿sobre Google Bigquery?

Respuesta

Google BigQuery es un servicio de almacenamiento y análisis de datos en la nube diseñado para gestionar y analizar grandes volúmenes de datos de forma rápida y escalable. Es parte de la plataforma de Google Cloud y permite a los usuarios realizar consultas en conjuntos de datos masivos usando SQL sin la necesidad de administrar infraestructura o servidores. Aquí te dejo los conceptos fundamentales para entender BigQuery:

1. Arquitectura sin Servidor (Serverless)

BigQuery es un servicio "sin servidor", lo que significa que Google gestiona automáticamente la infraestructura y el escalado de recursos. No necesitas configurar servidores, y el almacenamiento y procesamiento de datos se ajustan según la carga de trabajo. Esto permite a los usuarios enfocarse en consultas y análisis sin preocuparse por la infraestructura.

2. Almacenamiento Columnar

BigQuery usa un formato de almacenamiento basado en columnas (columnar storage), lo que facilita el procesamiento rápido y eficiente de grandes volúmenes de datos. Este enfoque mejora significativamente el rendimiento en consultas analíticas, ya que accede solo a las columnas necesarias, en lugar de recorrer todas las filas.

3. Consultas SQL y Soporte para ANSI SQL

BigQuery permite consultas utilizando SQL estándar compatible con ANSI SQL 2011. Esto hace que sea accesible para personas familiarizadas con SQL y simplifica la migración de consultas desde otras bases de datos SQL. También incluye extensiones específicas para BigQuery para mejorar la funcionalidad.

4. Escalabilidad Automática

BigQuery escala automáticamente los recursos para manejar tanto consultas pequeñas como consultas de petabytes de datos. Esta escalabilidad lo hace ideal para análisis de grandes volúmenes de datos y reportes en tiempo real.

5. Modelo de Pago por Consulta (o Almacenamiento y Consulta)

Existen dos modelos de pago en BigQuery:

- **Por consulta:** Solo pagas por los datos procesados en cada consulta, lo cual es ideal para consultas esporádicas.
- **Capacidad reservada:** Pagas una tarifa fija para obtener capacidad de procesamiento específica, adecuada para grandes organizaciones con cargas de trabajo constantes.

6. Almacenamiento a Largo Plazo y Gestión de Costos

BigQuery ofrece precios más bajos para almacenamiento a largo plazo, reduciendo el costo de almacenamiento de datos que no se modifican en 90 días. Este enfoque ayuda a gestionar el costo para datos históricos o de archivo que no requieren actualizaciones frecuentes.

7. Tablas, Conjuntos de Datos y Proyectos

- **Proyectos:** En Google Cloud, un proyecto organiza los recursos y datos asociados, como tablas y conjuntos de datos.
- **Conjuntos de Datos (Datasets):** Son contenedores para organizar tablas en BigQuery y pueden compararse a bases de datos en otros sistemas de gestión de bases de datos.
- **Tablas:** Almacenan los datos en filas y columnas dentro de conjuntos de datos. BigQuery admite tablas particionadas para mejorar la eficiencia de consultas.

8. Particiones y Clústeres

Las tablas se pueden dividir en **particiones** basadas en una columna de fecha o en un intervalo de tiempo. Esto mejora la eficiencia de las consultas al limitar los datos procesados. También pueden organizarse en **clústeres** por varias columnas para acelerar el rendimiento de las consultas.

9. Machine Learning Integrado (BigQuery ML)

BigQuery ML permite crear y entrenar modelos de machine learning directamente dentro de BigQuery usando SQL, sin necesidad de mover los datos a otra herramienta de ML. Esto es útil para tareas de análisis predictivo, como la clasificación y la regresión, dentro del entorno de BigQuery.

10. Funcionalidades de ETL (Extract, Transform, Load)

Aunque BigQuery no es una herramienta ETL, permite ejecutar tareas de transformación de datos. Además, es compatible con herramientas ETL externas y servicios como Dataflow o Dataprep, que facilitan el procesamiento previo a la carga de datos en BigQuery.

11. Seguridad y Control de Acceso

BigQuery se integra con el sistema de seguridad de Google Cloud, permitiendo un control de acceso granular con IAM (Identity and Access Management). Puedes definir quién tiene permisos para ver o consultar conjuntos de datos, tablas o columnas específicas.

12. Conectividad con Herramientas de Visualización y BI

BigQuery se integra con herramientas de análisis y visualización, como Google Data Studio, Tableau, Power BI, y Looker, facilitando la creación de dashboards y reportes. Estas integraciones lo hacen ideal para proyectos de análisis de datos y reportes en tiempo real.

13. Persistencia y Recuperación de Datos (Snapshots y Versiones)

BigQuery permite realizar **snapshots** de las tablas, proporcionando versiones de datos a lo largo del tiempo. Esto facilita la auditoría, recuperación y análisis de versiones históricas de datos.

14. Funciones de Geoespacial y BigQuery GIS

BigQuery incluye soporte para análisis geoespacial con BigQuery GIS, lo cual permite trabajar con datos geoespaciales (e.g., coordenadas GPS) directamente en BigQuery. Esto es útil para análisis de datos de ubicación y mapas.

15. Conectividad con Datos Externos

BigQuery permite acceder a datos externos mediante **External Tables** y **Federated Queries**. Por ejemplo, puedes consultar directamente archivos en Google Cloud Storage o bases de datos externas como Cloud Bigtable y Google Sheets sin necesidad de mover los datos.

BigQuery es una herramienta potente para análisis de datos masivos y permite realizar operaciones avanzadas con una interfaz SQL conocida, lo que lo hace una solución accesible y efectiva para la analítica empresarial.