Actividad SQL Noviembre 5 de 2024

Taller de SQL: Fundamentos y Comandos Avanzados

Objetivo: Entender los conceptos básicos de SQL, las claves primarias y foráneas, el uso de JOINs y explorar comandos útiles para consultas.

Actividad 1: Llaves Primarias y Foráneas Instrucciones:

Escribe en tus palabras lo que entiendes sobre las llaves primarias y las llaves foráneas en SQL. Asegúrate de responder a las siguientes preguntas:

¿Qué es una llave primaria y cuál es su función en una tabla?

La llave primaria es por asi decirlo una columna la cual identifica de manera única cada tabla, en la tabla se almacenan los atributos

¿Cómo se relacionan las llaves primarias con las llaves foráneas? ¿Por qué son importantes para establecer relaciones entre tablas?

La relación entre la llave primaria con la foranea es que la foránea concatena una tabla con la otra, para la existencia de una llave foránea es necesario de una primaria, pero una primaria no necesita a una foránea

Ejemplo: La clave primaria sobre una lista de empleados podría ser la cedula, debido a que es un dato que nunca se repetirá y podrá ser distintivo, esta podría ir conectada a otra tabla mediane una llave foránea como por ejemplo la eps, en la cual se identifique que tipo de eps tiene, si es afiliado únicamente o prepagada, la eps podría ser la otra llave primaria

Piensa en la llave primaria como un identificador único en una tabla. ¿Por qué crees que una llave primaria no puede tener valores repetidos ni nulos?

Debido a que estos valores tienen que ser únicos e irrepetibles por lo que son importantes para mantener un orden sin ningún posible colapso, ya que al no haber repeticiones se identifica como una característica clave

Piensa en la llave foránea como una forma de conectar tablas. ¿Por qué es útil en una base de datos relacional?

Debido a que podemos relacionar distintas informaciones de distintas tablas, esto ayuda a que sea un poco mas independiente cada tabla pero al momento de que yegase la necesidad de recurrir a información un poco mas especifica es muy útil contar con esta principal característica de las bases de datos relacionales

Actividad 2: ¿Qué es SQL?

Instrucciones:

Explica, en tus palabras, qué es SQL (Structured Query Language). En tu explicación, menciona lo que entendiste de los nuevos términos que aprendiste en la clase.

Pautas para los estudiantes: Es un lenguaje de computador por asi decirlo en el cual le preguntas a la aplicación por los datos ya almacenados en la base de datos, en cual puedes borarr, almacenar, agregar o editar valores

Puedes pensar en SQL como un lenguaje de programación que te permite trabajar con bases de datos.

En tu explicación, trata de incluir al menos dos de los siguientes términos:

consulta, transacción, relación, tabla, columna, o registro.

Podria llamarse com un lenguaje pero no de programación, ya que viene siendo como una tabla en la cual todo es visual, tiene algo de relación ya que puedes manejar distintos datos y relacionarlos como lo que se realiza en la concatenación de la llave primaria o foránea dependiendo de si este estructurada en una misma columna varias llaves o etc, lo que quiero decir es que no son lo mismo, el lenguaje de programación tiene la particularidad de que puedes pedirle al programa hacer distintos bucles o cosas un poco mas complejas que solo extraer , modificar, eliminar , agregar o relacionar informacion

Preguntas para reflexionar:

¿Por qué crees que es importante saber cómo interactuar con bases de datos usando SQL?

Debido a que para una mejor planificación es muy puntual entender la lógica sobre las bases de datos, es muy practico ya que es uno de los mas pupulares y fáciles de entender

¿Cuál de los términos te resultó más interesante y por qué?

Me resultó interesante mas que todo las llaves foráneas y primarias como van relacionadas y también la concatenación debido a que con esto puedes tener una buena estructura para empezar a desarrollar una base de datos con información que se relacione y cumpla con su proposito

Actividad 3: Tipos de JOINS y Sus Usos Instrucciones:

Investiga los diferentes tipos de JOINs que existen en SQL y realiza una breve comparación de cada uno. Menciona cómo se usan y en qué casos pueden ser útiles.

Aspectos a cubrir en tu investigación:

* INNER JOIN: ¿Cómo combina filas y en qué casos es más útil?

Combina las tablas con datos precisos comunes en una nueva tabla con un conjunto de datos en común

La cual esta nueva tabla por asi decirlo es una combinación sobre la estructura de ambas y se le agrega una letra para poder identificarla

* LEFT JOIN (LEFT OUTER JOIN): ¿Qué hace con las filas de la tabla izquierda que no tienen coincidencias?
* Left join lo que hace es buscar con los datos en común una relación y los organiza según las llaves primarias, si no encuentra datos relacionados arroja NULL
* RIGHT JOIN (RIGHT OUTER JOIN): ¿Qué hace con las filas de la tabla derecha que no tienen coincidencias?   
  El right join lo que hace es combinar dos filas de 2 tablas basándose en la coincidencia, la cual aparecerá todas las filas de la tabla derecha después del right join si no hay coincidencias mostrara NULL
* FULL JOIN (FULL OUTER JOIN): ¿Qué sucede con las filas de ambas tablas que no tienen coincidencias?

El full join es la combinación entre dos tablas las cuales relacionan si hay coincidencias con la otra, en caso tal que no tengan coincidencias la tabla arroja NULL

Pautas:

Puedes hacer una tabla comparativa para visualizar mejor las diferencias. Incluye ejemplos de cómo se vería una consulta con cada tipo de JOIN.

|  |  |  |  |
| --- | --- | --- | --- |
| Join | funcion | Resultado de filas | Anexo |
| Inner join | Devuelve únicamente las filas que tienen concidencias con las tablas relacionadas | Solo arroja las filas que tienen coincidencias basadas en ON | No devuelve NULL |
| Left join | Arroja todas las filas de la tabla iquierda junto con las coincidencias de la derecha | Las filas de la tabla izquierda y NULL si no hay coincidencia, en caso tal de que la haya arroja las respectivas filas de la tabla derecha | Arroja NULL en la derecha(si no encuentra coincidencia) |
| Right join | Arroja todas las filas de la tabla derecha junto con las coincidencias de la izquierda | Arrojara las filas de la tabla derecha y en la columna izquierda arroja las coincidencias | Arroja NULL en la izquierda(si no encuentra coincidencia) |
| Full join | Devuelve todas las filas de las dos tablas, si no hay coincidencia arroja NULL | Arrojará todas las filas de ambas tablas | Arroja NULL(si no encuentra coincidencia) |

Actividad 4: Comandos Útiles para Consultas en SQL Instrucciones:

Busca al menos 8 comandos de SQL que no hayas visto en clase y que te parezcan útiles para hacer consultas. Explica brevemente cada comando y por qué te parece útil. Puedes hacer tu búsqueda en Internet para encontrar ejemplos y comprender mejor su uso.

Ejemplo de comandos a investigar:

* DISTINCT: Filtra valores duplicados en los resultados de una consulta.
* LIMIT: Restringe el número de filas devueltas en una consulta.
* CASE: Proporciona condiciones y permite hacer consultas más complejas.
* COALESCE: Devuelve el primer valor no nulo en una lista.
* UNION: Combina los resultados de dos consultas en una sola.

1. ALTER- Modifica la tabla, vista, atributos
2. DROP- elimina la base de datos
3. SELECT- consultar datos
4. REVOKE- quitar privilegios a los usuarios
5. SAVEPOINT- es como un checkpoint donde guardas hasta ese proceso
6. ORDER BY- ordenar ya sea asc(ascendente) o desc(descendente)
7. AS- nombrarlo de cierta manera
8. GROUP BY- (nombres) agrupa