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?

**R/ Una llave primaria es una llave única que identifica una instancia de una entidad de las otras instancias. La función de la llave primaria es identificar una instancia de las otras a través de esta llave única, además de relacionarse a través de esta llave con otras entidades.**

¿Cómo se relacionan las llaves primarias con las llaves foráneas? ¿Por qué son importantes para establecer relaciones entre tablas? Ejemplo: Piensa en la llave primaria como un identificador único en una tabla.

**R/Una llave foránea es una llave primaria de una entidad la cual esta adentro de otra entidad, esto se utiliza para relacionar entidades y mostrar cuales entidades se relacionan con cuales.**

¿Por qué crees que una llave primaria no puede tener valores repetidos ni nulos? Piensa en la llave foránea como una forma de conectar tablas. ¿Por qué es útil en una base de datos relacional?

**R/Porque o sino al momento de relacionar una tabla con otra va a haber un error en la relación que se establece entre las entidades, no se va a saber que atributo específicamente se quiere traer de la tabla que se relaciona.**

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: 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.

**R/SQL es un lenguaje relacional que permite hacer operaciones entre entidades de una base de datos y gestionar de una manera eficiente los datos que hay dentro de estas.**

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

**R/Porque SQL realiza operaciones de relación dentro de las bases de datos, por lo tanto, es importante saber cómo interactúan para el momento de realizar consultas hacerlos de manera correcta, ya que al hacer una consulta se realizan operaciones de relación entre las entidades de la base de datos.**

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

**R/ los términos que más interesantes me parecieron fue los de llave primaria y llave forana; ya que la buena elección y asignación de la llave primaria va a garantizar el buen funcionamiento de las relaciones de las tablas cuando se llame en otra tabla como llave foránea.**

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?

• LEFT JOIN (LEFT OUTER JOIN): ¿Qué hace con las filas de la tabla

izquierda que no tienen coincidencias?

• RIGHT JOIN (RIGHT OUTER JOIN): ¿Qué hace con las filas de la tabla

derecha que no tienen coincidencias?

• FULL JOIN (FULL OUTER JOIN): ¿Qué sucede con las filas de ambas

tablas que no tienen coincidencias?

INNER JOIN: Se utiliza cuando se quieren tomar los datos que están comunes en dos comunes, es decir la intersección de estos. .

Ejemplo de uso: Tengo dos entidades, una entidad empleados y otra tabla departamentos. La tabla empleados tiene como llave foránea id\_departamentos. Si quiero saber que empleados tienen id\_departamento en común con la tabla departamento, entonces utilizo un INNER JOIN para hallar la intersección de las dos tablas.

LEFT JOIN: Ejemplo tengo dos entidades, una a la derecha y otra a la izquierda. El LEFT JOIN me va a traer todas las instancias de la tabla de la izquierda más los que están en intersección con la tabla de la derecha.

Ejemplo: Tengo una tabla clientes y otra tabla pedidos. Si necesito obtener todos los clientes así tengan pedidos o no, entonces utilizo el LEFT JOIN para unir la tabla pedidos con clientes y que se me impriman todos los clientes y sus pedidos, sin embargo los clientes que no tengan pedidos también se me van a imprimir porque LEFT JOIN trae todas las instancias de clientes. En los pedidos de los clientes que no tienen va a aparecer NULL.

RIGHT JOIN: Tiene un caso de uso similar al del LEFT JOIN, solo que el RIGHT JOIN devuelve todos los campos de la tabla derecha y la intersección con la izquierda.

Ejemplo de uso: Tengo una tabla calificación de restaurantes y otra tabla restaurantes, así que quiero imprimir todos los restaurantes y su calificación, aunque si el restaurante no tiene calificación también debe ser retornado.}

FULL JOIN: Muestra todas las filas de la tabla izquierda y de la tabla derecha sin importar si coinciden o no.

Ejemplo de uso: Quiero traer todos los campos de una tabla empleados y de otra tabla departamentos sin importar si cada empleado tiene un departamento asignado o no.

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.

AVG -> Calcula el promedio de los valores en una columna. Resulta importante cuando se quiere obtener el promedio de valores

BETWEEN -> Selecciona valores dentro de un rango

IS NULL -> Filtra filas que que su valor es NULL. Resuta importante cuando quiero saber si un campo es NULL, o también si no es nulo IS NOT NULL

LIKE -> Busca un patron dentro de los valores de una columna. Resulta útil cuando se quiere encontrar un patron que sigan los valores

IN -> Especifica valores dentro de una clausula WHERE

AS -> Renombra una columna o una tabla con un alias

LIMIT -> Limita el numero de filas retornadas en una consulta

SUM -> Retorna la suma de elementos de una columna, o conjunto de elementos.