**Evaluación técnica de programación**

**1. CONOCIMIENTOS SQL**

**1.1) Describe el funcionamiento general de la sentencia JOIN.**

La sentencia JOIN en SQL se utiliza para combinar filas de dos o más tablas en una base de datos, basándose en una relación lógica entre ellas, por ejemplo, “INNER JOIN” esta combina filas de dos tablas que cumplen con una condición especifica y solo devuelve las filas donde hay una coincidencia en ambas tablas.

**1.2) ¿Cuáles son los tipos de JOIN y cuál es el funcionamiento de los mismos?**

1. INNER JOIN

Devuelve solo las filas con coincidencias en ambas tablas.

**SELECT** empleados.nombre, departamentos.nombre

**FROM** empleados

**INNER JOIN** departamentos

**ON** empleados.departamento\_id = departamentos.id;

2. LEFT JOIN

Devuelve todas las filas de la tabla de la izquierda y las coincidencias de la tabla de la derecha. Las filas de la izquierda sin coincidencias mostrarán NULL para las columnas de la derecha.

**SELECT** empleados.nombre, departamentos.nombre

**FROM** empleados

**LEFT JOIN** departamentos

**ON** empleados.departamento\_id = departamentos.id;

3. RIGHT JOIN

Devuelve todas las filas de la tabla de la derecha y las coincidencias de la tabla de la izquierda. Las filas de la derecha sin coincidencias mostrarán NULL para las columnas de la izquierda.

**SELECT** empleados.nombre, departamentos.nombre

**FROM** empleados

**RIGHT** **JOIN** departamentos

**ON** empleados.departamento\_id = departamentos.id;

4. FULL JOIN

Devuelve todas las filas donde hay una coincidencia en una de las tablas. Las filas sin coincidencia tendrán NULL en el lado que no coincide.

**SELECT** empleados.nombre, departamentos.nombre

**FROM** empleados

**FULL** **OUTER** **JOIN** departamentos

**ON** empleados.departamento\_id = departamentos.id;

5. CROSS JOIN

Devuelve el producto cartesiano, es decir, todas las combinaciones posibles de filas entre las dos tablas.

**SELECT** empleados.nombre, proyectos.nombre

**FROM** empleados

**CROSS** **JOIN** proyectos;

6. SELF JOIN

Une una tabla consigo misma para relacionar filas dentro de la misma tabla.

**SELECT** e1.nombre **AS** empleado, e2.nombre **AS** jefe

**FROM** empleados e1

**JOIN** empleados e2

**ON** e1.jefe\_id = e2.id;

**1.3) ¿Cuál el funcionamiento general de los TRIGGER y qué propósito tienen?**

Los **triggers** o disparadores en bases de datos son como alarmas automáticas. Se activan cuando haces algo especifico en una tabla, como agregar, actualizar o borrar datos. Se puede decir que son reglas condicionales que cada vez que se realiza una acción antes mencionada se realiza otra ya automatizada previamente.

**1.4) ¿Qué es y para qué sirve un STORED PROCEDURE?**

Es como un mini programa dentro del sistema de bases de datos que puede hacer cosas específicas, como consultar datos, actualizarlos o realizar cálculos. Puedes pasarle ciertos parámetros y obtener resultados. Es útil porque se puede automatizar y se ocupa de realizar las tareas de manera eficiente.