**Prueba Técnica Especialista Soluciones Digitales (Datos)**

**Nombre: David Santiago Moreno Medina**

Por favor contestar las siguientes preguntas, selecciona la respuesta en rojo la respuesta correcta:

1. Escriba una query que regrese el nombre y salario del profesor con el salario más alto.

SALARIES (

Professor\_Name (TEXT),

Department (TEXT),

Salary (INT)

)

1. SELECT Professor\_Name, MAX(Salary) FROM Department;
2. SELECT Professor\_Name, MAX(Salary) FROM SALARIES;
3. SELECT MIN(Salary),Professor\_Name, FROM Department;
4. Ninguna de las anteriores

2. Escriba una query que regrese el departamento con el salario promedio más alto junto con ese salario promedio.

SALARIES (

Professor\_Name (TEXT),

Department (TEXT),

Salary (INT)

)

1. SELECT Department, AVG(Salary) FROM SALARIES GROUP BY DEPARTMENT ORDER BY AVG(Salary);
2. SELECT Department, AVG(Salary) FROM SALARIES ORDER BY AVG(Salary) DESC LIMIT 1;
3. SELECT Department, AVG(Salary) FROM SALARIES GROUP BY AVG(Salary) DESC LIMIT 1;
4. SELECT Department, AVG(Salary) FROM SALARIES GROUP BY DEPARTMENT ORDER BY AVG(Salary) DESC LIMIT 1;

3. Escriba un query que regrese todos los departamentos en la table (sin duplicados).

SALARIES (

Professor\_Name (TEXT),

Department (TEXT),

Salary (INT)

)

1. Select department from salaries where salary=0;
2. SELECT DISTINCT Department FROM SALARIES;
3. SELECT all Department FROM SALARIES;
4. SELECT DISTINC Department FROM SALARIES;
5. Explícame cómo se asegura la atomicidad, consistencia, aislamiento y durabilidad en una transacción SQL y cómo se manejan los errores y las situaciones de fallo.

Para garantizar la integridad de los datos, queremos que nuestras transacciones sean confiables: lo que significa que deseamos que cumplan con propiedades como automaticidad, consistencia, aislamiento y durabilidad.

En cuanto a la automaticidad, Queremos que las transacciones se realicen con éxito o que ninguna se realice. Esto se traduce en la garantía de que los errores no causen efectos parciales y en caso de un fallo, la operación se reinicie desde el principio.

En relación con la Consistencia: Nuestra meta es mantener un seguimiento de los estados en las transacciones. Para ello establecemos estados válidos y, mediante esta propiedad, asegurarnos que solo sea posible transaccionar de un estado válido a otro estado válido, para así mantener la consistencia en los datos.

El Aislamiento es esencial para asegurar que cada transacción se ejecute de manera independiente de las demás que estén en curso. Esto previene interferencias y conflictos entre transacciones, garantizando una mayor calidad en las ejecuciones

Finalmente, la Durabilidad: Es una propiedad que asegura que una vez que una transacción se completa con éxito, los cambios realizados de esa transacción permanecen en la base de datos, incluso si ese produce un fallo. Esto nos brinda garantías de que la información no se perderá independiente de las circunstancias adversas.

1. Dada una tabla

A (

id {varchar(30) , PK, NOT NULL},

name {varchar(30)})

y una tabla

B (

id {varchar(30) , PK, NOT NULL},

A\_id {varchar(30), FK (A.id), NOT NULL },

value {decimal})

Cómo obtendría una vista que permita devolver B.id, B.value, A.name?

Create view vista as

Select B.id, B.value,A.name from B

Inner join on A.id=B.A\_id

1. Por favor leer el siguiente caso de uso y posteriormente contestar las preguntas relacionadas al caso:

**Trabajando con facturas:**

Las facturas de venta se almacenan en la tabla (***SOE\_CustInvoiceJourStaging***) por compañía, identificada con el campo (***DataAreaId***), en dicha tabla se encuentran tanto las facturas (GFV0…, RFV0…) como las notas crédito (GCV0…, RCV0…) identificadas en el campo (***InvoiceID***). Estas facturas pueden estar en **diferentes monedas** (***CurrencyCode***) y provienen de órdenes de venta (***SalesId***), que se encuentran en la tabla (***XAP\_BISalesTableStaging***), y también se almacenan por compañía (***DataAreaId1***). En la tabla de las órdenes de venta se identifica el almacén donde se realizó la venta (***InventLocationId***). Esta facturación es tanto para clientes internos, como para clientes externos. Los clientes se identifican con la cuenta de cliente (InvoiceAccount), la cual, para los clientes externos inicia con la letra “X”.

Solo trabajaremos con las compañías ABC (010) y XYZ (020)

Se requiere:

1. Realizar una consulta SQL que permita sumar los valores del total de las ***facturas*** (***InvoiceAmount***) para cada compañía y almacén. En primer lugar, debe presentarse el almacén con mayores ventas y así sucesivamente.

select DataAreaId,InventLocationId,sum(InvoiceAmount) as T\_venta from SOE\_CustInvoiceJourStaging

where DataAreaId in ('010','020')

group by DataAreaId,InventLocationId

order by T\_venta desc

1. Realizar una consulta SQL que permita calcular el total de las **facturas**, sumando los valores del subtotal (SalesBalance) y los impuestos (**SumTax**), únicamente para los clientes externos, para cada cliente por compañía.

select DataAreaId,InoiceAccount,sum(SalesBalance+SumTac) as T\_Factura

from SOE\_CustInvoiceJourStaging

where DataAreaId in ('010','020') and left(InvoiceAccount,1)='X'

group by DataAreaId,InvoiceAccount

1. Realizar una consulta SQL que permita calcular el promedio facturado por cada cliente por mes de facturación (***InvoiceDate***)

select DataAreaId,InoiceAccount,MONTH(InvoiceDate),AVG(InvoiceAmount) as Promedio\_Fac

from SOE\_CustInvoiceJourStaging

where DataAreaId in ('010','020')

group by DataAreaId,InvoiceAccount,MONTH(InvoiceDate)

1. Describa que hace el siguiente código:

DataSet ds = new DataSet();

ds.Locale = CultureInfo.InvariantCulture;

FillDataSet(ds);

DataTable products = ds.Tables["Product"];

IEnumerable<DataRow> query =

from product in products.AsEnumerable()

select product;

Creamos un objeto Dataset, llenamos el dataset con una tabla llamada “Product” y luego seleccionamos las filas de la tabla y llenamos Dataset con las filas de nuestra tabla.

1. Describa que hace el siguiente código:

DataSet ds = new DataSet();

ds.Locale = CultureInfo.InvariantCulture;

FillDataSet(ds);

DataTable contacts = ds.Tables["Contact"];

DataTable orders = ds.Tables["SalesOrderHeader"];

var query =

from contact in contacts.AsEnumerable()

from order in orders.AsEnumerable()

where contact.Field<int>("ContactID") == order.Field<int>("ContactID")

&& order.Field<decimal>("TotalDue") < 500.00M

select new

{

ContactID = contact.Field<int>("ContactID"),

LastName = contact.Field<string>("LastName"),

FirstName = contact.Field<string>("FirstName"),

OrderID = order.Field<int>("SalesOrderID"),

Total = order.Field<decimal>("TotalDue")

};

El código crea un Dataset, establece una configuración de cultura, llena el dataset con datos y luego realiza una consulta. La consulta combina dos tablas para seleccionar filas relacionadas que cumplan ciertas condiciones. El resultado es una secuencia de objetos que representan los datos de ContactId,LastName,FistName,OrderId,Total.

1. Explique la funcionalidad del siguiente fragmento de código:

using System.ComponentModel.DataAnnotations;

namespace ConsoleApp1;

internal class Class1

{

public Class1(string name, int valor) {

Id = Guid.NewGuid();

Name = name;

Valor = valor;

}

[Key]

public Guid Id { get; set; }

[Required]

public string Name { get; set; }

[Required]

public int Valor { get; set; }

public override bool Equals(object? obj)

{

if (obj == null || GetType() != obj.GetType())

{

return false;

}

return Name == ((Class1)obj).Name && Valor == ((Class1)obj).Valor;

}

public override int GetHashCode() { return 0; }

}

El código define una clase “Class1” con propiedades “Id”,”Name” y “Valor”, que son validados para asegurar que “Name” y “Valor” no sean nulos ni vacíos. Además, la clase implementa un método   
“Equals” para comparar instancias por igualdad y un método “GetHashCode” que devuelve un valor hash constante de 0

1. ¿Cuál es el proceso que sigues para analizar datos?

El proceso de analizar datos comienza con la obtención de estadísticas descriptivas para las variables clave. Esto involucra calcular la media, la desviación estándar, los percentiles, y evaluar medidas de asimetría como el skew y la asimetría. Estos datos iniciales nos ayudan a comprender la variabilidad en nuestros datos, identificar valores atípicos y determinar si son relevantes en nuestro contexto, además de explorar la distribución de las variables a través de histogramas. Aquí nos interesa ver que patrón siguen nuestros datos, queremos ver distribuciones de probabilidad conocidas para hacer pruebas de que nuestros datos sigan alguna de estas. Esto nos sirve para crear intervalos de confianza, y en general nos ayuda a responder preguntas de la empresa y crear un modelo mas robusto.

Luego, se procede a examinar las correlaciones entre las variables, lo que permite identificar relaciones o redundancias que puedan afectar el modelo. Las correlaciones fuertes pueden sugerir que algunas variables aportan información similar, lo que puede llevar a considerar la eliminación de variables innecesarias y reducir la dimensionalidad.

El siguiente paso implica la realización de pruebas estadísticas, como la prueba t o chi-cuadrado, dependiendo de la naturaleza de los datos y los objetivos de la investigación. Estas pruebas ayudan a evaluar hipótesis específicas y determinar si existen relaciones estadísticamente significativas entre las variables.

Finalmente, se pueden aplicar técnicas de modelado estadístico o de aprendizaje automático, como regresión lineal, regresión logística, árboles de decisión o algoritmos de agrupamiento no supervisado, para comprender mejor cómo las variables interactúan y cómo pueden predecir o explicar resultados de interés

1. ¿Cuál crees que son los criterios para decir que una estructura de datos es buena o no?

Existen varios criterios a considerar, pero considero que los más importantes son la eficiencia, ya que la velocidad de los algoritmos influye significativamente en el tiempo de ejecución, y las diferentes estructuras de datos pueden tener tiempos de acceso variados (por ejemplo, O(n^2) o O(log(n)\*n), según el caso). La elección depende de la naturaleza de los datos y el contexto de la aplicación. Otro criterio crucial es la escalabilidad, lo que implica que la estructura debe mantener su eficiencia incluso cuando se trabaja con grandes volúmenes de datos. Además, la seguridad es fundamental, ya que se busca minimizar la probabilidad de fallos en la estructura de datos.

1. ¿Cómo manejarías datos faltantes en un conjunto de datos?

Tenemos varias formas de lidiar con datos faltantes. Una opción es simplemente eliminar los registros con valores faltantes, otra es reemplazarlos por la media de cada columna, y la tercera consiste en utilizar modelos predictivos para estimar esos valores faltantes sin perder información. En mi experiencia, si nuestros datos son bastante consistentes, el error es bajo y la variabilidad es estable, me inclino por el enfoque de modelos predictivos, ya que nos permite conservar la información. Sin embargo, la elección depende del caso y a veces es mejor optar por agregar la media o eliminar los datos faltantes.

1. ¿Qué es la normalización y por qué es relevante en el análisis de datos?

La normalización en el contexto de datos se refiere a escalar los valores de una variable para que estén en el rango de 0 a 1. Es relevante en el análisis de datos para permitir la comparación y combinación de variables con diferentes unidades o escalas, lo que facilita el análisis y la toma de decisiones

1. ¿Enumera y explica los esquemas de diseño en el modelado de datos?
2. Esquema de Tabla: Creamos una tabla puede ser en sql que define las columnas y tipos de datos que componen la tabla.
3. Esquema Clave primaria: Definimos las claves primarias, para esto ya debemos establecer las relaciones entre tablas mediante el uso de estas claves priamarias.
4. Creamos índices para cada columna en la tabla
5. Esquema de seguridad: Controlamos quién tiene acceso a la base de datos y qué operaciones puede realizar, también el acceso de qué personas tienen acceso a la base y quienes no.
6. Esquema vistas: Definimos consultas que le permiten al usuario acceder a los datos específicos sin traer todos los datos detrás de esa vista
7. Esquema de Procedimientos Almacenados: Definen ya los procesos que contienen alguna lógica en especial y sirven para operaciones que ya hayamos creado.
8. ¿Qué estrategias empleas para optimizar el rendimiento de las canalizaciones de datos y garantizar escalabilidad?

para optimizar el rendimiento de las canalizaciones de datos y garantizar la escalabilidad, utilizamos la estrategia de paralelización, que consiste en dividir el flujo de datos en tareas más pequeñas que se ejecutan en paralelo. Esto acelera significativamente el procesamiento y es especialmente efectivo en entornos con grandes volúmenes de datos.

Además, podemos aprovechamos servicios en la nube como AWS (Amazon Web Services) que nos permiten escalar automáticamente nuestra infraestructura en función de la demanda.