**EXAMEN DE CONOCIMIENTO**

**SQL SERVER**

1. **Examen Práctico**

Adjunto a este examen se encuentra un documento Excel llamado “Examen Práctico” en el cual cada hoja del documento hace referencia a una tabla SQL, tomando como base este documento, conteste lo siguiente

Que script se utilizaría para conocer el dato de:

* 1. Insertar un registro en la tabla *Tbl\_Recuperacion*

**La respuesta esta en el archivo de nombre respuestas.sql**

* 1. Eliminar de la tabla Dim\_Cliente los registros de los clientes que pertenezcan al AgrupadorCliente 543.

**La respuesta esta en el archivo de nombre respuestas.sql**

* 1. Conocer el valor Total de ImpRecuperacion e ImpObjetivo por cliente correspondiente al año anterior (2020).

**La respuesta esta en el archivo de nombre respuestas.sql**

* 1. Conocer el valor de %ImpRecuperacion utilizando la fórmula:

*ImpRecuperacion/(ImpObjetivo(Mes Anterior)-ImpBonificacion)*

**La respuesta esta en el archivo de nombre respuestas.sql**

* 1. Utilizando la herramienta Tableau diseñe un dashboard que represente los datos de: **ImpRecuperacion, ImpObjetivo e ImpBonificaciones** de cada cliente y por periodo mensual.

**La respuesta esta en el archivo de nombre ReporteTableauDavidAmaya.pdf**

Sobre la página oficial de Tableau puede descargar la versión de prueba, la liga de descarga es: <https://www.tableau.com/support/releases/desktop/2021.1.1>

* 1. Adjunto a este examen se encuentra un documento .txt llamado “SP”, este contiene un Stored Procedure que se utiliza para generar las estrellas, mencione que es lo que realiza el procedimiento y que resultado arroja.

**Es un proceso de carga de cartera a historico, la información que se puede procesar son los saldos, pagos o ambos conceptos juntos, maneja un periodo de fechas (inicial y final) para su ejecución y dependiendo de uno de los parametros de entrada (@debug) almacena en una bitacora el evento(proceso ejecutado), si el parametro de entrada @MigraHistoria esta marcado como "SI" se borra de la tabla DwhVtaHCartera los datos del concepto a migrar (saldos, pagos o ambos) para que se coloquen en esta misma entidad, en caso de que este parametro este marcado como "NO" no se realiza este borrado de información y solo se almacena en esta misma tabla los nuevos datos generados.**

**El datamart o almacen origen del que extrae esta información es el de Ventas**

**Resultado Final:**

**La información final que es la que genera la estella de cartera la cual se obtiene de la ejecucion del store CVtaCarteraSel, el resultado de este exec se guarda en una tabla temporal de nombre DwhVtaTempCartera para finalmente pasarla de esta a la tabla DwhVtaHCartera.**

* 1. Cuenta con un archivo de tipo XML llamado Ventas.EsquemaFacturas
     + - * Usted necesita declarar una variable de tipo XML llamada XML1.
         * La solución deberá asegurar que XML1 ésta validada para utilizarse por Ventas.EsquemaFacturas

**declare @XML1 xml (Ventas.EsquemaFacturas)**

* 1. Necesitas desplegar los registros de la tabla “Ordenes” para los Clientes de forma secuencial, considerando el formato XML como a continuación se visualiza:

<row OrdenID = “1” FechaOrden = “2016-01-01 23:59” Total = “2800.00” Nombre = “Cliente A” País = ‘México” />

<row OrdenID = “2” FechaOrden = “2016-02-01 20:00” Total = “3200.00” Nombre = “Cliente B” País = ‘México” />

¿Cuál de las siguientes T-SQL debería usar?

1. **SELECT OrdenID, FechaOrden, Total, Nombre, Pais FROM Ordenes INNER JOIN Clientes ON Ordenes.ClienteID = Clientes.ClienteID WHERE Clientes.ClienteID = 1**

**FOR XML RAW**

1. SELECT OrdenID, FechaOrden, Total, Nombre, Pais FROM Ordenes INNER JOIN Clientes ON Ordenes.ClienteID = Clientes.ClienteID WHERE Clientes.ClienteID = 1

FOR XML RAW, ELEMENTS

1. SELECT OrdenID, FechaOrden, Total, Nombre, Pais FROM Ordenes INNER JOIN Clientes ON Ordenes.ClienteID = Clientes.ClienteID WHERE Clientes.ClienteID = 1

FOR XML AUTO

1. SELECT OrdenID, FechaOrden, Total, Nombre, Pais FROM Ordenes INNER JOIN Clientes ON Ordenes.ClienteID = Clientes.ClienteID WHERE Clientes.ClienteID = 1

FOR XML RAW, AUTO

* 1. Necesitas crear una tabla con el siguiente nombre "DetalleOrdenes". Esta deberá reunir los siguientes requerimientos:

1. Escribir a disco el resultado
2. Contener una columna llamada TotalPorItem que almacene por cada producto el Precio de lista por la cantidad de la tabla Productos.

¿Cuál de los siguientes T-SQL deberá utilizar?

CREATE TABLE DetalleOrdenes (PrecioLista money not null, Cantidad int not null,

TotalPorItem as (PrecioLista \* Cantidad) PERSISTED TO DISK)

**CREATE TABLE DetalleOrdenes (PrecioLista money not null, Cantidad int not null,**

**TotalPorItem as (PrecioLista \* Cantidad) PERSISTED)**

CREATE TABLE DetalleOrdenes (PrecioLista money not null, Cantidad int not null,

TotalPorItem as (PrecioLista \* Cantidad) TO DISK)

* 1. Una aplicación comienza a presentar lentitud, usted descubre que una gran cantidad de memoria es consumida por un simple query dinámico.

Necesitas reducir el procedimiento cache de los querys sin la necesidad de crear ningún índice adicional.

¿Cuál opción deberá elegir para mejorar esto?

1. Add a HASH hint to the query
2. Add a LOOP hint to the query
3. Add a FORCESEEK hint to the query
4. Add an INCLUDE clause to the index
5. Add a FORCESCAN hint to the Attach query
6. Add a COLUMNSTORE index to cover the query
7. **Enable the optimize for ad hoc workloads option**
8. Cover the unique clustered index with a COLUMNSTORE index
9. Include a SET FORCEPLAN ON statement before you run the query
10. Include a SET STATISTICS PROFILE ON statement before you run the query
11. Include a SET STATISTICS SHOWPLAN\_XML ON statement before you run the query
12. Include a SET TRANSACTION ISOLATION LEVEL REPEATABLE READ statement before you run the query
    1. Usted tiene las siguientes definiciones en dos tablas:

|  |  |
| --- | --- |
| **CREATE TABLE CLIENTES** (  ClientID INT NOT NULL PRIMARY KEY,  Nombrecliente VARCHAR(100) NOT NULL, Direccion VARCHAR(100) NOT NULL) | **CREATE TABLE ORDENES** (  OrdenID INT NOT NULL PRIMARY KEY, ClienteID INT NOT NULL, DescripcionOrden VARCHAR(200)) |

Usted necesita asegurar que la columna ClienteID en la tabla “Ordenes” contiene solo valores que existen en la columna ClienteID de la Tabla Clientes.

¿Cuál de las T-SQL deberá utilizar?

1. ALTER TABLE CLIENTES ADD CONSTRAINT FX\_Clientes\_ClienteID FOREIGN KEY (ClienteID) REFERENCES Ordenes (ClienteID)
2. **ALTER TABLE ORDENES ADD CONSTRAINT FX\_Ordenes\_ClienteID FOREIGN KEY (ClienteID) REFERENCES Clientes (ClienteID)**
3. ALTER TABLE ORDENES ADD CONSTRAINT CK\_Ordenes\_ClienteID CHECK (ClienteID IN SELECT ClienteID FROM Clientes))
4. ALTER TABLE ORDENES ADD CONSTRAINT FK Ordenes ClienteID PRIMARY KEY (ClienteID)
   1. Usted crea una tabla que tiene las columnas ProductoID, DescripcionCodigo y Totales, registra una marca a mitad de año para los productos en esa tabla.

La tabla tiene totales obtenidos por 50 productos para varias marcas.

Usted Necesita recuperar productos que obtuvieron los totales más altos para cada marca junto con el detalle de los totales.

¿Cuál de los T-SQL deberá utilizar?

* 1. SELECT ProductID AS Code, RANK() OVER(Order BY avg(Marcas) DESC) AS Value

FROM ProductosMarcados GROUP BY ProductID

* 1. SELECT ProductID, Nombre, Marca DENSE\_RANK() OVER(Order BY Marcas DESC) AS Rank

FROM ProductosMarcados

* 1. SELECT ProductID as Code, DENSE\_RANK() OVER(Order BY avg(Marcas) DESC) AS Value

FROM ProductosMarcados GROUP BY ProductID

* 1. SELECT ProductID as Code, NTILE(2) OVER(Order BY avg(Marcas) DESC) AS Value

FROM ProductosMarcados

GROUP BY ProductID

* 1. SELECT ProductID as Code, Marcas as Value

FROM (SELECT ProductoID, Marcas AS Marcas,RANK() OVER(PARTITION BY DescripcionCodigo ORDER BY Marcas ASC) AS Rank FROM ProductosMarcados) tmp

WHERE Rank = 1

* 1. SELECT ProductID as Code, Marcas as Value

FROM (SELECT ProductoID, Marcas AS Marcas,RANK() OVER(PARTITION BY DescripcionCodigo ORDER BY Marcas DESC) AS Rank

FROM ProductosMarcados) tmp

WHERE Rank = 1

* 1. **SELECT ProductID as Code, Marcas as Value**

**FROM (SELECT ProductoID, Marcas AS Marcas,RANK() OVER(PARTITION BY ProductoID ORDER BY Marcas ASC) AS Rank FROM ProductosMarcados) tmp**

**WHERE Rank = 1**

* 1. Usted comprueba que durante periodos se detectan picos altos en consumo de CPU, ciertas operaciones están tomando más tiempo que el esperado. Al parecer el problema son bloqueos.

Usted necesita reunir más datos para determinar cuáles procesos están bloqueando y poder identificar la causa-raíz. ¿Qué debemos hacer?

* 1. Comenzar un trace utilizando SQL Server Profile para cachar el evento Deadlock
  2. Utilizar SP\_CONFIGURE y poner un parámetro para el proceso bloqueado y Comenzar un Trace con el SQL Server Profile para cachar el proceso bloqueado.
  3. Programar un JOB en el agente y ejecutarlo cada 60 segundos e insertar los resultados de la ejecución de la vista dinámica sys.dm\_os\_wait\_stats dentro de una tabla.
  4. **Utilizar el sistema de monitoreo para cachar el evento: Lock Waits/sec**
  5. Detecta que el índice "IDX\_TransaccionProductos\_CodidoTransaccion" es pobre debido a una alta fragmentación.

Usted necesita desfragmentar el índice y asegurar que los querys en uso son capaces de indexar durante el proceso de fragmentación. ¿Cuál opción deberá utilizar?

ALTER INDEX ALL ON TransaccionProductos REBUILD

ALTER INDEX IDX\_TransaccionProductos\_CodidoTransaccion ON TransaccionProductos\_CodidoTransaccion REORGANIZE

**ALTER INDEX IDX\_TransaccionProductos\_CodidoTransaccion ON TransaccionProductos\_CodidoTransaccion REBUILD**

CREATE INDEX IDX\_TransaccionProductos\_CodidoTransaccion ON IDX\_TransaccionProductos\_CodidoTransaccion WITH DROP EXISTING

* 1. Hay alta contención de Lecturas/Escrituras en varias tablas utilizadas por su transacción.

Usted necesita minimizar el espacio en la base de datos TEMPDB.

¿Cuál nivel de ISOLATION deberá utilizar?

**SERIALIZABLE**

SNAPSHOT

READ COMMINTTED SNAPSHOT

REPEATABLE READ