

Metodología de trabajo

Debéis crear una carpeta y añadir un fichero con extensión sql para escribir cada consulta y verificar su funcionamiento en la base de datos, en este mismo fichero detallar la descripción sobre su funcionamiento y cualquier aspecto que consideréis relevante y subiereis a Git. Al finalizar evaluaremos la eficiencia de la formación en base al proyecto que habéis creado. Este método os permitirá memorizar algunos conceptos. Podéis usar Chatgpt, Youtube o cualquier otro medio de información, pero si copiáis y pegáis código sin entender realmente los conceptos no aprenderéis nada.

Podéis trabajar en parejas y establecer cualquier estrategia que consideréis adecuada, repartiros las tareas y que luego cada uno presente un ejercicio, elegir una forma de colaborar para hacerlo más productivo. El equipo debe tomar decisiones de forma consensuada, elegir un líder por cada grupo (podéis rotarle si consideráis oportuno) en caso de conflicto que tome la decisión, podéis dividir los equipos o cambiar la estrategia si algo no os funciona. El objetivo es que aprendáis y que nadie se quede atrás y esto **último es responsabilidad de todos**. Si alguien finaliza todos los ejercicios **su trabajo será ayudar a que los demás finalicen los suyos**. La formación finaliza cuando todo el equipo ha terminado así que iréis a la velocidad del más lento.

Sabemos que hay personas a las que las cuesta más, es algo normal y no debe preocuparlos, es cuestión de experiencia, repetir los ejercicios una y otra vez hasta entenderlos, buscar información. Antes de solicitar ayuda intentar resolverlo por vuestra cuenta, leer sobre las posibles soluciones, el camino para resolver un problema te enseña muchos aspectos que difícilmente vais a aprender de otra forma, pero solicitar ayuda cuando se necesita demuestra inteligencia, no tengáis miedo de hacerlo. Algunos de los conceptos que pongo aquí no están en los cursos, algunos conceptos me han ayudado a diseñar mejor las bases de datos y resolver distintos problemas.

Tener en cuenta que hay algunos conceptos que usareis en este ejercicio que hay que utilizar con precaución, preguntarlos donde deben estar las reglas de negocio (hay argumentos en defensa de distintas estrategias), cada proyecto y escenario son distintos, hay mucha gente que intenta no usar Store Procedures, sin embargo, en algunas ocasiones representan una clara ventaja si queremos obtener un buen rendimiento, de la misma forma el uso de cursor, índices, etc. pueden degradar el sistema, los bloqueos continuos pueden provocar Deadlocks y otros problemas. Lo importante es conocer las ventajas y desventajas de cada aproximación para seleccionar la solución más adecuada en cada caso.

Como hemos comentado, existe un déficit general sobre el conocimiento del diseño y desarrollo de bases de datos, muchos de vosotros usáis Entity Framework, Datasets, comandos de Ado 2.0 o Dapper que representan una capa de abstracción, sin embargo, todas estas tecnologías generan o ejecutan sentencias SQL y cuando tengáis un problema de rendimiento tendréis que conocer cómo resolverlo. Con un ORM como EF es fácil cometer errores, a veces la generación de consultas no es la más optima y en ocasiones debéis poder analizarlas en detalle para dar con la solución adecuada. Por otro lado, diseñar una aplicación usando una aproximación como Code First requiere conocer en profundidad como definir los tipos de datos, crear reglas de integridad referencial, índices, relaciones y otros conceptos que requieren una base sólida.

Debéis razonar las respuestas, debatir cada solución, pensar cuando es más adecuado usar una u otra aproximación o incluso evitar su uso porque en distintos escenarios pueden suponer una mala práctica. Ampliar el conocimiento sobre el diseño y desarrollo en Base de datos es esencial para cualquier programador, científico de datos, DBA, etc. Su aplicación en sistemas de Big data e IA es siempre relevante. Este ejercicio requiere esfuerzo, si no encontráis una solución por vosotros mismos, buscarla entre todos. Esforzarse por encontrar una solución os hará aprender otros conceptos importantes, nuestra intención es que aprendáis y logréis autonomía cuando no sepáis resolver algún problema. No olvidéis que el esfuerzo provoca que gran parte de los conceptos no se olviden fácilmente. 😊

Ejercicios Base de datos Propia

Puedes configurar Docker y las bases de datos de AdventureWorks usando las instrucciones de este link:

<https://forms.office.com/e/240tf517Cu>

Deberéis instalar además Azure Data Studio – Sql Management Studio – Github – Visual Studio Code (Opcional)

Crea una carpeta llamada SqlServerTest en tu disco C:

- Dentro de la carpeta crea un documento en Word llamado SqlDocument, en este documento tomaras notas de todo aquello que consideres importante.
- Usando los comandos de Github sube esta carpeta a una cuenta en Github. Si no tienes una puedes abrir una gratuita con tu email de Semicrol o tu email personal, como prefieras. Enviar el link a Lucia para que lo incorpore este en el proyecto de formación de Devops.
- Por cada pregunta crear un archivo con extensión .sql. Ejemplos:

PivotExample.sql
RankExample.sql
DatabaseTypes.sql

Podeis adjuntar la información de lo que hace cada sentencia usando las notaciones de comentarios para Sql Server como /* */ o '. Cada fichero Sql Server tenga la explicación y la sentencia de esta forma:

```
/*
    Esta query muestra cómo crear un login y usuario
*/
```

De esta forma podéis subir los archivos al repositorio de Github de vuestro proyecto.

- Desde Sql Server Data Studio o Sql Management Studio conéctate al servidor Sql Server de tu equipo Docker.
 - o Crea una nueva base de datos llamada 'smcdb1'. Establece el collation de smcdb1 en Modern_spanish_ci_ai
 - o Define en el documento que significado tienen las terminaciones ci, ai y que otras opciones existen.
 - o Dentro de smcdb1, crea una tabla llamada 'Test'
 - o Crea una segunda base de datos llamada 'smcdb2'. Establece el collation de smcdb2 en Latin1_general_cs_as
- Dentro de la tabla 'Test' crea un campo clave llamado Code de tipo string char(20).
- Añade un campo por cada tipo diferente (char, nchar, unique identifier, int, double, money, etc).
- Abre el documento Word y escribe para que se usa cada uno de los tipos, su tamaño en bytes, las diferencias entre campos con un tipo similar y describe cuando es mejor usar uno u otro, compara los tipos numéricos, string, datetimes, unique identifier, etc.
 - o Ejemplos char vs nchar, char vs varchar, varchar vs nvarchar, datetime vs smalldatetime, int vs bigint vs decimal vs money vs smallmoney, datetime vs smalldatetime vs datetimeoffset y todas las relaciones que veas.
 - o Describe las ventajas y desventajas de usar un int autonumérico vs unique identifier.
- Describe cuando es adecuado usar default values en el diseño de una tabla, escribe distintos ejemplos de default values usando distintos tipos de campos.
- **Crea algunos datos de prueba en la tabla Test.**
- Copia la tabla 'Test' desde la base de datos smcdb1 a smcdb2
- Intenta hacer una query usando un join sobre de la tabla smcdb1.dbo.Test hacia la otra base de datos smcdb2.dbo.Test has creado previamente Acuérdate que el collation es diferente en ambas bases de datos.

- Crea una tabla llamada Sales.InvoicesHeader,
La estructura de esta tabla será la siguiente:

Invoiceld
 InvoiceDate
CustomerId
AddressId (crea una tabla relacionada llamada Address y establecimiento AddressId como campo clave, los campos que consideres importantes y un campo adicional denominado CountryId, de la misma forma crea la tabla Countries relacionada con esta última)
 TaxBase (Base Imponible)
 TotalVat (Total Iva)
 Total

- Crea otra tabla denominada Sales.InvoicesDetail, relaciona ambas tablas por los dos campos y establece una integridad referencial completa.

La estructura de la segunda tabla será la siguiente:

Invoiceld
 RowNumber
ProductId
 Description
 Quantity
 UnitPrice
 Discount (Descuento en porcentaje)
VatTypeId (Tipo de Iva, 1,2,3...)
 TotalLine

- Crea las tablas adicionales que creas que necesitas de los campos en rojo para complementar este modelo, define los tipos de datos que consideres más adecuados, teniendo en cuenta que estas tablas pueden ser usadas por muchos usuarios al mismo tiempo y que el tamaño de las tablas es un factor importante a tener en cuenta, a menor tamaño mayor rendimiento y menor coste de almacenamiento.
- Analiza que columnas deben usar nulos y cuáles no, define ventajas y desventajas usando columnas con o sin nulos.
- Crea un diagrama de base de datos y verifica las reglas de integridad referencial, ten en cuenta que no podrás eliminar un cliente que tenga facturas o un artículo que tenga líneas de detalle,
- Los campos clave pueden estar compuestos de varios campos, describe cuando usarías esta estrategia y si te parece una buena práctica.
- Usando triggers o campos calculados haz que los campos Total y TotalLine de las tablas se actualice cada vez que un campo relacionado con estos cálculos se actualice. Razona y describe en el documento (Antes de buscarlo en ChatGPT) 😊, qué ventajas y desventajas representa el uso de campos calculados o triggers.
- Crea los datos para varias facturas, crea y guarda un programa en TSQL que te rellene al menos 10.000 facturas con al menos 50 líneas de detalle por cada una de forma aleatoria, cada factura debe tener al menos dos tipos de Iva diferentes.
- (**Nuevo**) Diferentes campos de una tabla pueden tener distintos collation. Describe las limitaciones de su uso y diseña una tabla con este ejemplo.

- Resuelve la siguiente consulta complementando con los 12 meses, no tengas en cuenta los datos, son solo para el ejemplo.

Type	Year	Total Invoices	Total Vat	Quantity	Avg Total Invoice	Stdev Total Invoice	January	February	Marzo, Abril, Mayo...
Invoices	2020	5.203.200 €	1.092.630 €	2.210 €	14.000 €	9.000 €	412.000 €	450.000 €
Payment (Abonos)	2020	-210.000 €	4.410 €	-241 €	1.200 €	7.00 €	1.700 €	1.610 €
	2019								

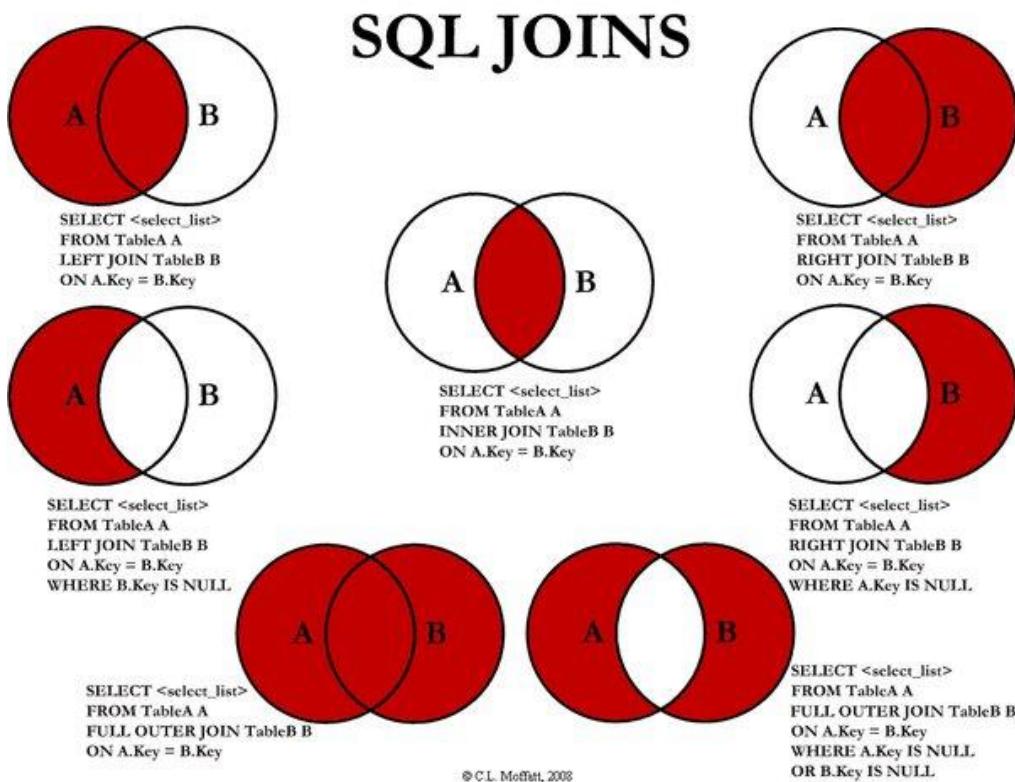
- Crea una consulta con los 5 clientes que más facturan.
- Crea una consulta ordenada por los países que más facturan.
- Crea una consulta que calcule el total de facturación de los últimos tres meses a partir del día actual.
- Elimina la integridad referencial entre ambas tablas.
- Crea un sistema de integridad referencial en ambas tablas usando solo Triggers.
- Añade un campo calculado a la tabla SalesInvoicesHeader en el que el total de la factura se la suma de la base_imponible + iva. Describe las ventajas y desventajas de usar campos calculados.

Ejercicios con Adventure Works

Selecciona la base de datos AdventureWorks2022, sobre ella vamos a realizar distintos ejemplos, cada ejemplo lo deberéis guardar un fichero de extensión Sql y la descripción de lo que hace, podéis usar los comentarios de Sql Server para explicarlo.

- Realiza un listado de los empleados de cada departamento que tenga más de 5 empleados.
- Realiza una consulta para encontrar el mejor vendedor de cada mes basado en el total vendido.
- Realiza una consulta con los productos menos vendidos ordenados de mayor a menor.
- Realiza una consulta de ventas por territorio,
- Se está realizando un proceso de actualización de salarios en la tabla. El incremento es del 5% para los empleados de un determinado departamento. Se requiere que toda la operación se realice como una transacción y que se manejen posibles errores sin afectar la integridad de los datos.
- Crea un diagrama por cada esquema de base de datos y añade las tablas en cada uno de ellos. Cambia la visualización de cada tabla a Custom y reordena la vista. Revisa el modelo de relación e integridad referencial y aprender a crear relaciones gráficamente usando esta herramienta.
- Escribe el bloque de código T-SQL que realice la actualización dentro de una transacción y que incluya manejo de errores con TRY...CATCH. Describe cómo funciona.
- Crea algunos ejemplos usando Group y Having, usando Sum, Avg, Stdev() y otros comandos similares
- Crea una consulta de ejemplo usando Pivot y Unpivot
- Explica la diferencia entre Where y Having
- Crea un ejemplo usando una Common Table Expression o CTE
- Crea un ejemplo de consulta recursiva usando una expresión CTE con una clave numérica
- Define la diferencia entre un Índice Clusterizado y un Índice No-Clusterizado
- Que otros tipos de índices existen
- Como y porque debes actualizar los índices.
- Escribe un ejemplo de creación de índices con distintas opciones, define en el documento para que se usa cada una, que es Fillfactor
- Que es un índice único y para qué sirve.
- Crea un ejemplo de uso con Full Text Search, actívalo en una columna de una tabla, realiza varios ejemplos de búsqueda por contenido usando Contains, Freetext, Containstable y FreeTextTable y verifica los tiempos.
- Fuerza el uso de un índice en una query, describe un ejemplo de cuando debes hacerlo.
- Crea un ejemplo usando un índice filtrado y explica las ventajas de su uso.
- Crea un ejemplo usando SQL LIMIT/OFFSET
- Crea varios ejemplos usando Fecha y sus distintas configuraciones

- Crea un ejemplo usando For LOOP
- Crea un ejemplo usando un bucle While y break
- Crea un ejemplo usando los comandos Top y Limit y Percent
- Crea un ejemplo usando Rank y explica cómo funciona
- Crea un ejemplo usando DenseRank y explica cómo funciona
- Crea un ejemplo usando Ntile y explica cómo funciona
- Pon un ejemplo de uso de Scope_Identity(), Identity(), @@Identity, and Ident_Current() y describe la diferencia entre cada uno.
- Para qué se usa el lenguaje DCL, pon algunos ejemplos.
- Crea dos tablas con tres campos claves y establece el modelo de integridad referencial
- Crea un ejemplo particionando una de las tablas y explica para qué se debe usar el particionado, los requisitos que tiene y qué ventajas e inconvenientes representan esto. Si no fuera posible realizar un particionado, qué otras estrategias se te ocurren, descríbelo en el documento.
- Crea un ejemplo de consulta usando los diferentes métodos de Join con las tablas que escogas de Adventure Works.

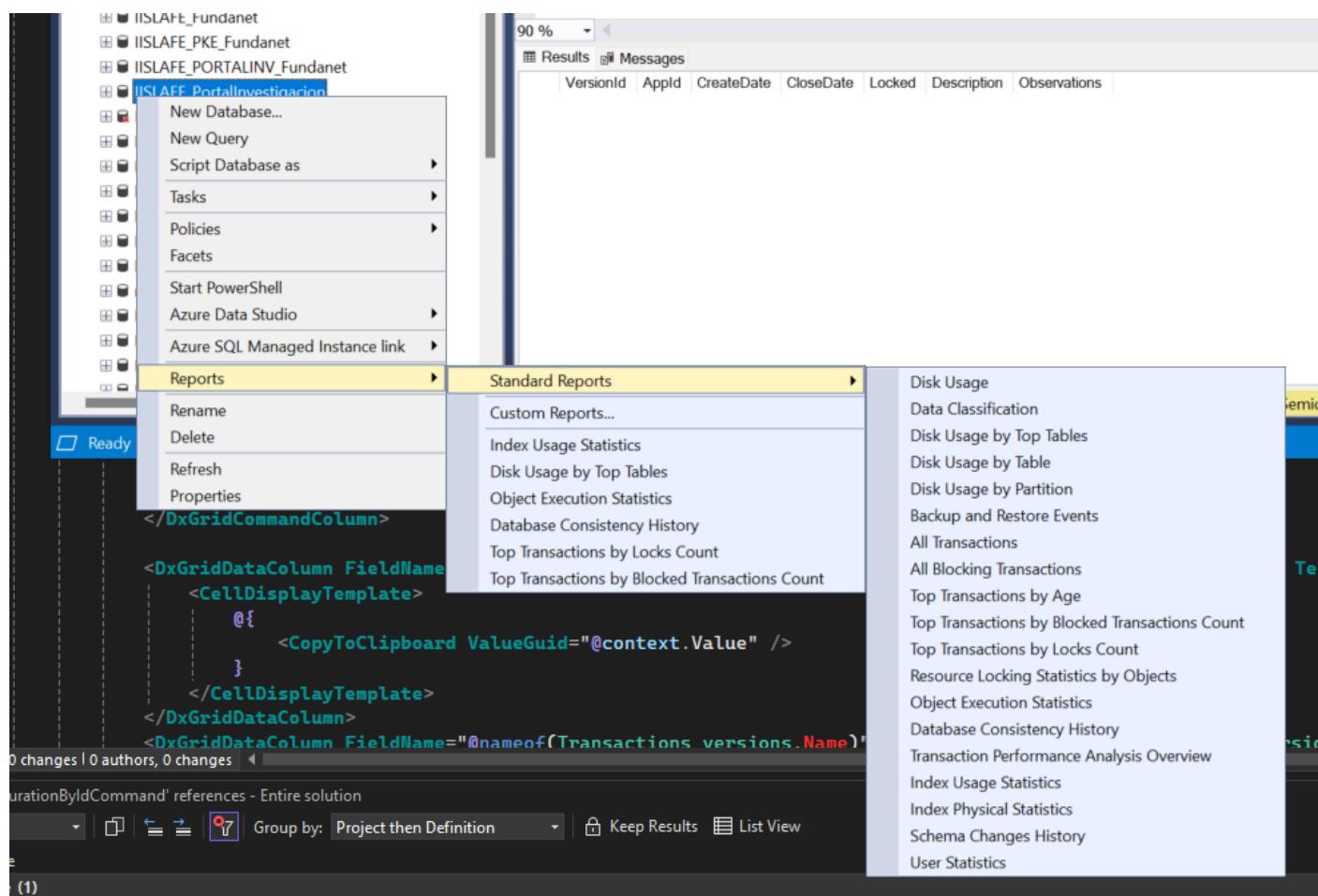


- Escribe un ejemplo usando una función de agregado como Sum o Avg sobre una partición determinada.
- Crea una tabla usando el tipo de dato HierarchyId
- Escribe una consulta añadiendo un nodo
- Escribe una consulta para devolver el nodo padre de un nodo determinado.
- Escribe una consulta usando la cláusula TableSample, explica para qué se utiliza.
- ¿Qué es un plan de ejecución y cómo se puede analizar?
- Describe las estrategias de optimización de consultas.
- ¿Qué es la inyección SQL y cómo se previene? Pon un ejemplo.
- ¿Cómo se configura el cifrado de datos en SQL Server?
- Explica el uso de tablas temporales y variables de tabla.
- Cuando y porque deberías usar Alias con cada tabla.
- Escribe una consulta bajo un bloque try y catch, describe las ventajas y desventajas de su uso.
- Realiza una consulta que realice varias actualizaciones que grabe los cambios si es un día entre semana y la cancele en caso contrario.

- Crea un sinónimo en la base de datos con una tabla sobre la base de datos [AdventureWorksLT2022] y mapea la tabla SalesLT.Customer
- Dentro de un plan de ejecución, existen varias operaciones, describe cada una de ellas, ¿cómo identificas un bajo rendimiento? Define los pasos que has seguido para optimizarla.
- Crea un login sobre la base de datos con que solo tenga acceso de lectura sobre las tablas del esquema Person y verifica que funciona correctamente.
- Desde Sql Server Management Studio aprende a usar el Sql Server Profiler, configúralo para capturar trazas, realiza algún cambio en algún registro de las tablas de la base de datos y captura las sentencias que lo producen, aprende a optimizar índices de forma automática a partir de un archivo de trazas.
- Realiza una consulta con el número de registros de cada tabla ordenandola de mayor a menor.
- Averigua cuales son las consultas más costosas de la base de datos.
- Escribe una consulta en TSQL para conocer cuáles son las consultas que producen más bloqueos
- Escribe una consulta que provoque un Deadlock en la base de datos y los pasos a seguir para resolver este problema, así como la importancia del uso de los diferentes niveles de aislamiento.
- En AdventureWorks crea un nuevo login de acceso a la base de datos y añádele un rol para que solo permita realizar operaciones de lectura, sobre las tablas del esquema 'Person'. Comprueba que funciona.
- Define las diferencias entre usar grupos de usuarios vs usuarios independientes.
- Crea un trigger de base de datos. Explica que es para qué sirve
- Explica que diferencia existe con un trigger en una tabla.
- Crea una tabla a partir de la estructura a cualquier otra añadiendo una columna adicional de tipo nchar(10) con el nombre 'estado', con el mismo nombre + '_history'. Crea en la tabla origen nuevos triggers para actualización, borrado e inserción de registros en la tabla origen a partir de las tablas inserted y deleted de forma que cualquier modificación quede registrada con el ultimo estado del registro.
- Escribe un ejemplo explicando que es y para qué sirve una Función de Tabla
- Escribe un ejemplo explicando que es y para qué sirve una Función escalar
- Crea un procedimiento almacenado que devuelva el nombre del tercero a partir de su ID
- Crea un procedimiento almacenado usando un table type para pasar un conjunto de parámetros y devolver valores relacionados con estos.
- Crea un procedimiento almacenado que use un cursor de tipo Fast Foward, describe que es un cursor, porque debes liberarlo al finalizar el procedimiento y los distintos tipos de configuración como (LOCAL, GLOBAL, FORWARD_ONLY | SCROLL, STATIC | KEYSET | DYNAMIC | FAST_FORWARD, READ_ONLY | SCROLL_LOCKS | OPTIMISTIC).
- Busca información sobre el uso de TempDB, describe que es y cómo funciona, las distintas estrategias de uso que pueden afectar al rendimiento, división de múltiples archivos, porque es una buena práctica ubicar Tempdb en unidades distintas a las de la base de datos y otros conceptos que consideres interesantes.
- Define los distintos tipos de transacciones que existen en Sql Server y la diferencia entre cada una de ellas
- Crea un nuevo tipo de dato llamado CustomMoney con 14 dígitos numéricos y 4 decimales con valor de 0 por defecto para que no permita nulos. ¿Crees que es una buena práctica?, arguméntalo exponiendo las ventajas y desventajas sobre su uso.
- Crea una tabla en memoria. Describe su función, ventajas y desventajas. En que tablas de Fundanet usarías esta configuración, argumenta tu respuesta. Aprende a usar el asistente Table Memory Optimization Advisor e intenta migrar una tabla en memoria, describe el proceso.
- Explica que es y para qué sirve una Función de Tabla
- Explica que es un sinónimo y para que se usa, pon un ejemplo práctico usando dos bases de datos distintas.
- Crea una nueva base de datos a partir del siguiente Script [Northwind Reborn - Base de datos de ejemplo de SQL Server · GitHub](#)
- Explica para que sirve el versionado de tablas, activa el versionado en una de las tablas y explica cómo funciona. Recuerda que esta opción no está activa en todas las versiones de Sql Server.
- Genera un Backup de la base de datos y restáurala con otro nombre. Escribe las sentencias sql en el proyecto.
- Realiza algún cambio en la base de dataos. Haz una copia de seguridad después de haber realizados varios cambios en diferentes momentos y restáurala usando el registro de transacciones para restaurarla 5 minutos

antes de la copia, verifica no hay cambios posteriores de este momento. Describe la importancia del log de transacciones.

- Explica la diferencia entre DELETE y TRUNCATE y realiza un ejemplo, explica la diferencia de tiempos.
- Crea varios ejemplos con operaciones Bulk Insert desde una tabla, vista y fichero csv y verifica los tiempos frente a una operación Insert tradicional. Documenta el funcionamiento y las diferencias, define cuando es más adecuado usar una u otra aproximación.
- Explica la diferencia entre UNION y UNION ALL y realiza un ejemplo.
- Explica que son y cómo funcionan los Bloqueos, los distintos niveles de Aislamiento y el versionado de filas, Row Versioning, pon un ejemplo con cada una de ellas. Describe cuales son las recomendaciones y mejores prácticas sobre esto.
- Cuál es el objetivo de ROWGUIDCOL, realiza un ejemplo y describe cuando deberías utilizarlo.
- Estudia la estructura de la base de datos AdventureWorks, verás que usa esquemas y distintas reglas de estilo en la definición de las tablas, describe cuales son y porque suponen una buena práctica.
- Describe las ventajas y desventajas de usar procedimientos almacenados.
- Crea un ejemplo usando JSON_VALUE, JSON_QUERY y OPEN_JSON
- Aprende como copiar una base de datos en Docker y restaurarla. Para ello crea una importa una base de datos, describe los pasos que has realizado para hacerlo.
- Una buena herramienta para obtener información sobre tu base de datos es el uso de Informes, realiza una pequeña descripción de cada informe.



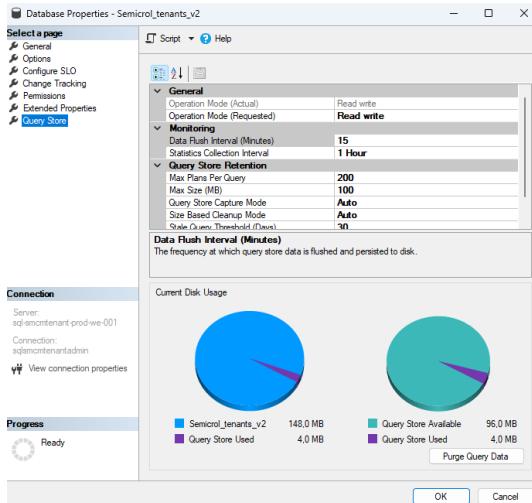
- Los comandos DBCC son muy usados, realiza un ejemplo con cada uno y describe su función.

Planes de ejecución y Query Plan

- Echa un vistazo a los videos de Youtube sobre optimización de consultas con Planes de ejecución y uso el uso del QueryPlans. [planes de ejecucion sql server - YouTube](#), [SQL Server Query Plan Analysis: The 5 Culprits That Cause 95% of Your Performance Headache \(youtube.com\)](#), [Funeral Big Campaign 2024 - Acquisition 30 sec B \(youtube.com\)](#)
- En el documento define lo que es un Query Plans y para que se usa.
- Crea un ejemplo del uso de un Query Plan, obtén el XML sobre el plan de ejecución.
- Que es Change Tracking en Sql Server y cómo funciona, activalo y realiza algún ejercicio sobre esto.
- ¿Puedes forzar un plan de ejecución?, pon un ejemplo.
- Describe con tus palabras cada elemento que compone un plan de ejecución. [Graphical Execution Plans for Simple SQL Queries - Simple Talk \(red-gate.com\)](#)
- Investiga sobre las funciones que realiza un Query Store, aprende a manejarlo, usa las consultas de regresión, describe tu experiencia con este esta herramienta. Deberás aprender sobre:
 - o Localizar y solucionar consultas con regresiones de elección del plan
 - o Identificación y optimización de las consultas que consumen más recursos
 - o Pruebas A/B
 - o Mantener la estabilidad del rendimiento al actualizar a una versión más reciente de SQL Server
 - o Identificación y mejora de las cargas de trabajo puntuales

Sobre una consulta de Adventure Works que tenga varios inner join y distintos parámetros de selección, realiza pruebas A/B forzando índices o creando índices y evaluando las diferencias.

- Uno de los grandes desconocidos y algo muy usado son los informes de estado sobre SQL,



- Con los conocimiento obtenidos, ‘antes de usar Chatgpt’ :), intenta optimizar desde la base de datos de Fundanet que tenéis en Docker las siguientes vistas Fundanet llamada [dbo].[pi_vAutoresFiliaciones], [dbo].[proyectos_derecho] y [dbo].[v_ec_TareasPacientes], los procedimientos almacenados [dbo].[pi_proyectosPersonalUnidades] y [dbo].[RecalcularStock] Detalla que operaciones has realizado, la mejora en la consulta y cualquier información que consideres relevante. Revisa las incidencias antes y después de la refactorización. Almacena dos archivos el Sql Previo y el posterior para ver los resultados obtenidos.
- Crea una vista indizada, pon un ejemplo de funcionamiento y verifica el rendimiento antes y después de aplicar un índice.
- Realiza una búsqueda usando Like y caracteres de búsqueda como %
- Habilita una búsqueda de texto completo, FullTextSearch, crea un índice y usar las cláusulas Contains y FreeText, describe las diferencias que hay con Like

Si has llegado hasta aquí, enhorabuenaii

Tienes conocimientos de Sql Server suficientes para realizar un correcto diseño de bases de datos, recuerda que faltan muchas cosas, TSQL tiene más de cientos de instrucciones que seguramente no uses en la mayoría de las ocasiones por que ni siquiera las conoces, es importante que continues aprendiendo sobre otros conceptos, cada versión de Sql Server incorpora nuevos comando y operaciones, tener en cuenta que algunas no estarán disponibles en versiones antiguas.