

GreenTec



I. Aspectos generales

GreenTec es un sistema de base de datos e infraestructura de soporte necesaria para apoyar a un sistema de información encargado de los parques naturales.

II. Infraestructura

A continuación, se define la red que deberá crearse para poder dar soporte de infraestructura por medio de un clúster.

- Instale tres máquinas virtuales utilizando ya sea VmWare, HyperV, VirtualBox e instale Windows Server Enterprise Edition.
- Interconecte en red las 3 máquinas de forma que sea visible toda la red virtual vía IP.
- Una de las máquinas anteriores será su controlador de dominio donde creará los usuarios y asignará los permisos necesarios, adicionalmente también brindará el servicio de DNS
- Siga los pasos para instalar SQL Server Management Studio en el virtualizador seleccionado en modo Cluster, utilizando clustering service.

- Sea cuidadoso con los pasos que indique el tutorial que encuentren porque faltar a alguno o hacerlo equívocamente podría implicar empezar de cero
- Instale los SQL Server con replication services, integration services, reporting services y analysis services.
- Utilice cualquier base de datos dummy para probar si el cluster funciona correctamente como fail over, para ello se conecta al cluster sql con un cliente, se ejecuta un query, responde y se apaga posteriormente el servidor master, posteriormente el slave entra a funcionar y el query puede ser atendido nuevamente

III. Base de datos relacional

Para el modelado de la base de datos considere lo siguiente:

1. Una comunidad puede tener varios parques naturales. En toda comunidad existe uno y sólo un organismo responsable de los parques. Un parque puede estar compartido por más de una comunidad.
2. Un parque natural se identifica por un nombre, fue declarado en una fecha, se compone de varias áreas identificadas por un nombre y caracterizadas por una determinada extensión. Por motivos de eficiencia se desea favorecer las consultas referentes al número de parques existentes en cada comunidad y la superficie total declarada parque natural en cada comunidad.
3. En cada área forzosamente residen especies que pueden ser de tres tipos: vegetales, animales y minerales. Cada especie tiene una denominación científica, una denominación vulgar y un número inventariado de individuos por área. De las especies vegetales se desea saber si tienen floración y en qué periodo se produce ésta; de las animales se desea saber su tipo de alimentación (herbívora, carnívora u omnívora) y sus periodos de celo; de las minerales se desea saber si se trata de cristales o de rocas.
4. Además, interesa registrar qué especies sirven de alimento a otras especies, teniendo en cuenta que ninguna especie mineral se considera alimento de cualquier otra especie y que una especie vegetal no se alimenta de ninguna otra especie.
5. Del personal del parque se guarda el número de identificación, nombre, dirección, teléfonos (domicilio, móvil) y sueldo. Se distinguen los siguientes tipos de personal:
 - Personal de gestión: registra los datos de los visitantes del parque y están destinados en una entrada del parque (las entradas se identifican por un número).

- Personal de vigilancia: vigila un área determinada del parque que recorre en un vehículo (tipo y matrícula).
 - Personal investigador: Tiene una titulación que ha de recogerse y pueden realizar (incluso conjuntamente) proyectos de investigación sobre una determinada especie. Un proyecto de investigación tiene un presupuesto y un periodo de realización.
 - Personal de conservación: mantiene y conserva un área determinada del parque. Cada uno lo realiza en una especialidad determinada (limpieza, caninos...).
6. Un visitante (número de cédula, nombre, domicilio y profesión) debe alojarse dentro de los alojamientos de que dispone el parque; éstos tienen una capacidad limitada y tienen una determinada categoría.
 7. Los alojamientos organizan excursiones al parque, en vehículo o a pie, en determinados días de la semana y a una hora determinada. A estas excursiones puede acudir cualquier visitante del parque.
 8. Por comodidad, suponemos que un visitante tiene, obligatoriamente, que alojarse en el parque. Suponemos también, que cada vigilante tiene su vehículo propio que sólo utiliza él.

Basado en lo anterior proceda a:

- Diseñar e implementar en SQL Server Management Studio con el diagramador de SQL la base de datos.
- Hacer los scripts de llenado de la base de datos de tal forma que logre rellenar un mínimo de 30000 excursiones en al menos 100 parques naturales con al menos 1000 visitantes disponibles a lo largo del 2012 y 2018.
- Utilizar en al menos 3 tablas algún método de *bulkcopy* o *bcp* para el llenado.
- Hacer 1 revisión preliminar de la base de datos únicamente, la cual deberá ser a más tardar el 16 de octubre.
- Montar la base de datos en el clúster y asegúrese que funciona correctamente.

IV. ETL a Base de datos de grafos

- Diseñe una base de datos para *Neo4j* la cual solo va a contener la información del punto 4 de la sección III. De tal forma que en un futuro esta base de datos funcione como fuente de datos para que algún sistema pueda consultar de manera rápida dada una especie y no tener que consumir la base de datos principal.
- Utilizando *Integration Services* cree un paquete ETL que haga los siguientes pasos:

- Eliminar del *Neo4j* todos los registros.
- Migrar los datos de la base de datos central al *Neo4j* con los datos más actuales.
- No se implementará el sistema de consulta, solo la base de datos en *Neo4j* y el ETL de carga de datos.
- Cree las consultas necesarias en *Neo4j* para que a partir de los datos de una especie se extraiga la información sobre a cuales especies es depredador y a cuales sirve como presa.

V. Replication services

- Ahora el estudiante con un 4to computador físico o virtual que también posea SQL Server procederá a programar un servicio de replicación.
- La base de datos en el clúster deberá replicarse en forma bidireccional a otro servidor.
- Utilizando el SQL Management Studio y vía *insert*, se probará que los datos que se inserten en la base en el clúster son replicados al nuevo servidor y viceversa
- Utilice *replication services* para dicho cometido.

VI. Scripts

Seguidamente el estudiante realizará una serie de ejercicios con los cuales fortalecerá y pondrá en práctica los conocimientos sobre SQL Avanzado

Crear los stored procedures transaccionales, funciones o triggers necesarios para el llenado de la base de datos. Además, implemente los siguientes puntos:

- Cree 1 vista de datos utilizables por el sistema que requieran al menos 4 tablas, dicha vista debe ser materializada.
- Escriba 1 *store procedure* transaccional que haga escritura en al menos 4 tablas.
- Cree 1 consulta que requiera al menos 4 JOINS, 2 funciones agregadas, tres *subqueries*, un case, un CONVERT, un ORDER BY y una función, someta la consulta al *Show Estimated Execution Plan* y mejórela basado en las métricas de tiempo y resultados obtenidos. Conserve ambos *queries* antes y después de la optimización y compare los resultados
- Cree un *stored procedure* transaccional que llame dentro de la transacción a otro SP transaccional y este a otro SP que también sea transaccional. (De dos niveles). Cada *stored procedure* debe afectar al menos a dos tablas. Demuestre en este SP el funcionamiento del *commit* y el *rollback* tanto para un caso de éxito como uno de fallo

- Cree un *stored procedure* que inserte N registros en una tabla, tomando dichos valores de un *xml* suministrado por parámetro. El *stored procedure* parsea el *xml* con las librerías que posee para ello e inserta los datos.
- Realice el mismo trabajo anterior, pero usando *table valued parameters*.
- Cree un(os) *query(s)* para cada una de los siguientes tipos de instrucciones y/o operadores de tal manera que se demuestre claramente su uso:
 - Uso de cursor local, demuestre por qué es local
 - Uso de cursor global, demuestre por qué es global
 - Uso de un *trigger*
 - Sp_recompile
 - Merge
 - Coalesce
 - Substring
 - LTRIM
 - AVERAGE con algún grupo
 - TOP
 - CASE
 - &&
 - Execute as
 - Union
 - Distinct

VII. Mantenimiento

Cree un plan de mantenimiento que:

- Haga un backup completo cada semana a la 4am todos los sábados.
- Demuestre como restaurar un backup full y uno incremental.

VIII. Seguridad

Creando usuarios de prueba proceda a los siguientes pasos

- Demostrar que un usuario puede o no acceder a una base de datos.
- Que un usuario tenga permisos de hacer operaciones de *select* y no de escritura sobre una tabla.
- Que un usuario pueda o no ejecutar *stored procedures*.

Creando roles demuestre

- Que los usuarios miembros pueden hacer operaciones de *select* sobre unas tablas y no de escritura.
- Que los usuarios miembros pueden ejecutar o no *stored procedures*.

IX. Concurrencia

- Para las pruebas de concurrencia deberá usar operaciones nativas y NO operadores de LOCK que bloqueen expeditamente tablas, registros o columnas
- Cree situaciones ficticias que hagan que el motor de base de datos se someta a situaciones que fuercen a producirse los resultados solicitados
- Escriba dos scripts transaccionales que se aseguren de crear un *deadlock* usando operaciones de *select* y *update* sobre dos tablas distintas
- Demuestre con scripts cómo funcionan los niveles de insolación READ UNCOMMITTED, READ COMMITED, REPEATABLE READ y SERIALIZABLE, dejando claro los problemas que solucionan, así como los problemas que no solucionan.
- Demuestre como podrían darse *Rollbacks* cuando hay casos de *Timeouts*.
- Cree un cursor que bloquee los datos que un cursor utiliza, para ello use un cursor para update, luego invoque un script que demuestre que sucede un deadlock mientras el cursor está haciendo el recorrido, acá debe ser evidente que los registros o campos son bloqueados por el cursor conforme avanza por los registros y no así por los registros que no haya visitado aún. Los registros deben ser bloqueados por el cursor en su barrido sin tener que hacer operaciones adicionales sobre los mismos.

X. Notas

- Todas las consultas de información deben tener sentido semántico para el lector de los resultados, de hecho, asígnele un nombre a las mismas.
- Prácticamente todo el trabajo es hecho vía scripts y ejecutado desde el *Console Management* de SQL Server.
- Todos los scripts deben hacer un *print* indicando porque paso se va ejecutando, excepto cuando se trata de ciclos donde nada más debe existir un *print* al inicio y al salir del ciclo.
- Muchas de las tareas son experimentales por lo cual no tienen resultados concretos, en su caso se evalúa lo bien justificado que esté el experimento.
- Puede utilizar de 1 a N scripts en general o por tarea a realizar, perfectamente un script o *query* podría solucionar varios de los ejercicios a la vez.

XI. Aspectos administrativos

- La tarea es en grupos de máximo 4 personas y se entrega el día sábado 27 de octubre del 2018 en la cita de revisión **presencial** indicada por el profesor.

- No se permitirá defensa por medio de video llamadas o algún otro medio. De no presentarse o llegar 15 minutos tarde a la cita asignada pierde la totalidad del puntaje asignado a la prueba.
- **Asunto del correo:** [II-2018-CE-3101] II Proyecto, debe contener:
 - Carnet y nombre de los estudiantes.
 - Link al repositorio o carpeta de google drive.
- Cualquier duda, omisión o contradicción en la especificación se debe aclarar con el profesor.
- Cualquier clase de copia de código será sancionada de acuerdo con el reglamento vigente y se llevará hasta la consecuencia de amonestación con carta al expediente.
- El código debe estar suficientemente documentado de tal forma que usted se pueda orientar en el fácilmente durante la defensa.
- Deben entregar un PDF documentando los pasos que siguieron para cumplir cada punto del enunciado. Por ejemplo que incluya el nombre de todos los scripts y una breve descripción de la funcionalidad, configuración de la red del clúster etc.

XII. Bonificación

Para optar por los puntos de bonificación, debe haber completado todos los puntos solicitados en el enunciado.

Implemente un sistema con interfaz gráfica para poder utilizar el sistema.

XIII. Evaluación

- Infraestructura 5%
- Base datos 2.5%
- ETL 5%
- Replication Services 5%
- Scripts 4%
- Mantenimiento 2%
- Seguridad 2.5%
- Concurrencia 3%
- Documentación 1%
- Bonificación: 5%

Total: 30% + 5% de bonificación