Instituto Tecnológico de Costa Rica, Escuela de Computación, Bases de Datos
Prof. Rodrigo Núñez N, Proyecto II – 35%

LevelUP

LevelUP es un sistema de carrera y logros para las organizaciones; en dicho sistema el personal de administración, motivación de personal o recursos humanos configura un conjunto de condiciones, reglas, puntos, logros y premios para los empleados de la organización.

En el sistema es posible ingresar los expedientes de los empleados, ahí se almacenan sus datos personales, fechas de ingreso, los distintos contratos que la persona ha tenido, observaciones de los expedientes con respecto a levelUP, la lista de títulos y puestos así como su categoría ganada en levelUP.

Todos los participantes dentro de la organización inician sin ninguna condecoración y con cero puntos.

Debe existir un catálogo de condecoraciones por departamento las cuales son nombre, icono o imagen de la condecoración y las reglas de cómo se obtiene, las reglas pueden ser por tiempo o por tipos de logros alcanzados.

Adicionalmente los administradores pueden configurar un catálogo de logros donde hay estudios realizados, nuevos idiomas, charlas hechas o recibidas, seminarios, procesos mejorados a lo interno de la empresa, mejoras o innovaciones empresariales, trabajo social, etc existe un catálogo amplio de logros. Estos tipos de logros tienen una valoración en puntos y cada vez que alguien alcanza un logro gana esos puntos.

Por otro lado existen los premios que son sorpresas, refrescos, días libres, dinero, viajes, ropa entre otros.

Por otro lado debe ser posible configurar las reglas de la organización, por ejemplo se pueden ganar premios al obtener una condecoración, o bien puntos. Se pueden ganar premios y puntos después de obtener cierta cantidad de logros de algún tipo o del conjunto de tipos. Cuando se aplica esa regla es importante especificar la cantidad de veces no repetidas que debe pasar ese tipo de logro para obtener ese premio o condecoración o puntos.

En otro módulo los jefes superiores o bien las personas autorizadas hacen ingreso mes a mes de los logros incurridos por los empleados y así revisan los logros y premios obtenidos para poder premiar y homenajear a los acreedores.

Aspecto importante es que éste sistema es multi idioma y multi moneda.

Infraestructura
La configuración de red de LevelUP es la siguiente:

1. **Base de datos**

- Dos servidores SQL Server 2008 Enterprise Edition o Superior, instalados sobre un Windows 2008 Server Enterprise R2
- Un servidor controlador de dominio en Windows 2008 Server
- Un servidor web ya sea con IIS, Tomcat o Apache
- Todos los servidores serán instalados como máquinas virtuales
- Los dos servidores SQL Server serán instalados y configurados haciendo Mirroring con witness, el witness será el controlador de dominio esto para tener alta disponibilidad y recuperación de datos ante fallas
- Los clientes deberán conectarse al SQL Server usando un primary y secondary en la conexión para que el paso del servidor maestro al esclavo en caso de fallo del primero sea automático tan pronto lo indique el witness
- Todo lo anterior lo puede instalar y configurar usando una DB dummy para luego aplicarlo a la verdadera en caso de ser necesario
- Asegúrese que la base de datos dentro del mirror no vaya a sufrir cambios estructurales una vez que ya esté en modo mirror
- El servidor Web servirá de hosting para un REST service que será consumido por la aplicación de pruebas
- El servidor Web deberá abrir una conexión segura usando HTTPS, para ello utilice certificados generados por usted mismo usando ya sea los utilitarios de Linux o Windows

Base de datos

- Proceda a diseñar e implementar en SQL Server la base de datos de LevelUP, se recomienda usar el editor de diagramas del mismo SQL Server
- Haga los scripts de llenado de la base de datos utilizando únicamente código SQL a manera de script siguiendo las siguientes condiciones:
    - El sistema tiene al menos dos monedas
    - Deben haber entre 200 y 400 empleados
    - Existe un catálogo de logros de al menos 15 tipos de logros, entre 50 y 100 premios, entre 10 a 15 condecoraciones por departamento y al menos 3 departamentos
    - Un total de datos de 4 años de uso del sistema que incluye los logros de los empleados, los puntos, condecoraciones y premios obtenidos, sumando entre todo eso un total aproximado de 120000 registros. Para esto último utilice varios randoms sobre rangos y valores controlados todos usando SQL scripting con T-SQL
- No utilice lenguajes de programación de alto nivel, utilice solo el SQL que ofrece MSSQL y su lenguaje
- Podrá hacer 1 revisión preliminar de la base de datos únicamente la cual deberá ser enviada a más tardar el 25 de setiembre
- Monte la base de datos en el cluster y asegúrese que funciona correctamente
- Ahora con dicha base de datos ya poblada proceda a realizar los siguientes ejercicios

Consultas misceláneas

- Cree 1 vista de datos utilizables por el sistema que requieran al menos 4 tablas, recuerde indexar la vista y hacerla dinámica
- Escriba 1 store procedure transaccional útil para el sistema que haga escritura en al menos 4 tablas
- Haga un select que cree una columna dinámica de una consulta cuya columna numere posibles grupos de datos, se recomienda la instrucción CASE
- Cree 1 consulta que requiera al menos 4 joins sobre tablas, 2 funciones agregadas, tres subqueries, un case, un convert, un order by, un having, una función y alguna de éstas operaciones: IN, NOT IN, EXISTS. Someta la consulta al analizador/optimizador y mejórela basado en las métricas de tiempo y resultados obtenidos. Conserve ambos queries antes y después de la optimización y compare los resultados

- Realice una consulta que contenga al menos 3 JOINS la cual deba resolver un problema de SET DIFFERENCE y de INTERSECTION
- 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, para probarlo tan solo se invoca con valores correctos y con valores incorrectos
- Cree un stored procedure que retorne el resultado de una consulta de al menos 3 tablas utilizando xml creado con el método implicit, retornando los nombres de los campos como nombres de elements. Asegúrese que en el xml resultado sea claro la jerarquía de las tablas en el xml mismo, es decir que se puede determinar al leer el xml a que tabla corresponde que valor
- Cree un stored procedure que inserte N registros en una tabla, tomando dichos valores de un xml suministrado por parámetro, asegurese que los datos del xml ocupen hacer join con una tabla real y el resultado de esa consulta sea el insert
- Realice el mismo trabajo anterior pero usando table valued parameters
- Haga un query que genere un archivo CSV de datos basado en dos tablas
- 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:
  .1. Uso de cursor local, demuestre por qué es local
  .2. Uso de cursor global, demuestre por qué es global
  .3. Uso de un trigger
  .4. Sp_compile
  .5. Merge
  .6. Coalesce
  .7. Substring
  .8. LTRIM
  .9. AVERAGE con algún grupo
  .10. TOP
  .11. &&
  .12. SCHEMA BINDING
  .13. WITH ENCRYPTION
  .14. Execute as
  .15. Union
  .16. Distinct

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
  - Que un usuario pueda ejecutar solo ciertos SPs pero que no pueda hacer ninguna otra operación sobre las tablas

Concurrencia

- Para las pruebas de concurrencia deberá usar operaciones nativas de T-SQL 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
- Trate de crear un deadlock en cascada que produzca alguna de las siguientes situaciones: que hayan datos inconsistentes en la base de datos por al menos 5 minutos. Crear más deadlocks en cascada de los otros deadlocks hasta que el motor de base de datos intente colapsar.
- Demuestre con scripts cómo funcionan los niveles de isolación READ UNCOMMITED, READ COMMITED, REPEATABLE READ y SERIALIZABLE. En los scripts de prueba o bien transacciones que cree para demostrar su uso, debe asegurarse no solo de demostrar los problemas que previene si no también los problemas que produce, dejando claro cuál es la capacidad o la acción interna que realiza cada tipo de isolación.
- Cree un cursor que bloquee los datos que el 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 aun. Los registros deben ser bloqueados por el cursor en su barrido sin tener que hacer operaciones adicionales sobre los mismos
- Escriba un script que produzca un deadlock en cascada de 3 niveles, es decir los SP se bloquean A->B->C->A, dicha cascada debe ser visible en el monitor de procesos de SQL Server o por medio de un script o herramienta de monitoreo

Webservice y Aplicación

- El webservice deberá hacerse en la tecnología que desee y hosteado en el servidor Web instalado
- El Webservice tendrá los métodos necesarios para poder atender las funciones de la aplicación de prueba
- El WS deberá ser tipo REST y usar JSon como formato de intercambio
- El webservice debe tener N capas, la capa del endpoint del servicio, una capa de implementación que lleva la lógica y una capa de acceso a datos. La capa de acceso a datos podrá trabajar en modo Pool de máximo 4 conexiones o bien en modo abierto de conexiones sin pool. Por medio de algún archivo de configuración o propiedades del sistema se podrá configurar si el WS va usar pool o no. Esta virtud se comprobará con el monitor de conexiones de SQL Server
- La aplicación de prueba deberá tener dos funciones fundamentales
  - Una pantalla de configuración y creación de reglas que permita crear, editar y ver las reglas del sistema LevelUP
  - Una pantalla para insertar los registros de los logros de los empleados
- La aplicación se debe hacer en la tecnología que guste, solo debe soportar comunicación REST con el WS
- Deberá seguir el patrón de diseño MVC para la interface gráfica de la aplicación, lo cual NO es lo mismo de usar la tecnología Web MVC de tecnologías como Visual Studio o similares, si no que estén claramente delimitadas las clases del Model, View y Controllers.
- Recuerde solicitar el mínimo de información al usuario para realizar dichas operaciones, toda la información que se muestre en pantalla deberá leida de la base de datos
- Recuerde que tiene que tener soporte de HTTPs para poder tener comunicación con el WS
- El WS se conecta al mirror, cuando se apague el servidor principal el WS va a retornar errores de conexión y datos pero en algunos segundos tan pronto se active el servidor esclavo la respuesta correcta del WS se hace evidente.

Otros Aspectos

- 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, esto es de ayuda para el seguimiento de las tareas
- 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
- Cualquier sospecha de copia anulará el trabajo
- El trabajo puede hacerse en grupos de máximo 4 personas
- Todos los miembros del equipo de trabajo están sujetos a revisión sobre cualquier tema o ejercicio
- El proyecto se revisará contra cita:
    - Miércoles 2 de octubre, mirroring funcional y web server instalado y respondiendo via https. Para esto último se puede usar cualquier browser para garantizar el uso del certificado
    - Revisión final 19 y 20 de octubre