**Benemérita Universidad Autónoma de Puebla**

**Facultad de Ciencias de la Computación**

**Ingeniería en Tecnologías de la Información**

**Administración de Bases de Datos. Indicaciones para Realizar el proyecto (otoño 2019)**

**Nombre del Docente: Carlos Armando Ríos Acevedo\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Nombre del Alumno: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fecha: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**Total, de puntos a obtener del proyecto: 100**

1. Con base en el modelo de la base de Datos denominada ***“scholar”,*** ingresar una muestra significativa de datos o registros (200), con diferentes fechas de años de la tabla de ***“curso”***. **(5 puntos)**
2. Particionar la tabla **“curso”** en los siguientes archivos **(5 puntos):**

* Una partición primaria (mdf).
* Una partición de bitácora (ldf).
* Seis particiones secundarias (ndf), divididas por año.

1. Diseñar una estrategia de respaldos que permita respaldar la base de datos en los siguientes términos **(5 puntos):**

* Generar una ubicación física (BACKUP-DEVICE) para los archivos full llamada **“FullScholar”**, donde se almacenen los respaldos mencionados.
* Generar una ubicación física (BACKUP-DEVICE) para los archivos diff llamada **“DiffScholar”,** donde se almacenen los respaldos mencionados.
* Generar una ubicación física (BACKUP-DEVICE) para los archivos logs llamada **“LogScholar”,** donde se almacenen los respaldos mencionados.

1. Generar un script con el archivo de respaldo especial “**tail log**”, que se ejecute en el caso de pérdida de información y se haga una recuperación manual de los datos **(5 puntos).**
2. Generar la automatización de los respaldos y restauraciones **(backup-recovery)**, de los archivos con el siguiente cronograma: el domingo a las doce la noche (semanal) realizar un full, todos los días a la misma hora (diario) un diferencial y cada hora uno del transaction log. Al final guardar los archivos txt, que se genere por cada respaldo para mostrar la evidencia **(5 puntos).**
3. Agregar una columna a la tabla de alumno, que se llame comentarios y que sea de tipo XML, donde se puedan colocar comentarios de los alumnos y después se pueda recuperar con la sentencia FOR XML. Los atributos de los elementos son: descripción del comentario, fecha del comentario, quien lo realizó y acciones que se tomaron **(5 puntos).**
4. La inserción en las tablas de: curso, lista, acta. Deben ser transaccionales y no deben permitir leer o escribir a otro usuario, si en momento preciso se está trabajando con la inserción, actualización, o eliminación de la información en esas tablas **(5 puntos).**
5. Programar los siguientes procedimientos almacenados **(10 puntos):**

* Recibiendo la matricula del alumno muestre sus calificaciones agrupadas por periodo.
* Muestre la siguiente información de los alumnos: matricula, nombre, paterno, materno, carrera, plan de estudios, nombre materia, calificación, la fecha, nrc, profesor que impartió el curso.
* Recibiendo como parámetro el nrc del curso, muestre el horario con salón, hora entrada, hora salida, profesor y día.

1. Usando uniones externas realizar las siguientes consultas **(10 puntos):**

* Que muestre los alumnos que aparecen en lista y no en actas (en caso de no existir ingresar datos para validar el ejercicio).
* Mostrar las asignaturas que no se han ofertado en cursos (en caso de no existir ingresar datos para valida rel ejercicio).

1. Usando las herramientas de volcado BCP y WIZARD exportar e importar datos en la tabla de alumnos, que sean creados en un archivo de texto **(10 puntos).**
2. Crear cuatro logins y usuarios con las siguientes características **(10 puntos):**

* USERACADEMICO. - tiene acceso a toda la información del sistema, pero no puede respaldar ni restaurar ningún dato.
* USERADMIN. - tiene acceso a toda la información del sistema y puede respaldar y restaurar.
* USERPROFESOR. – tiene acceso total a las tablas de listas y actas, pero no puede crear usuarios, dar privilegios, respaldar o restaurar información.
* USERALUMNO. – tienen acceso de lectura a cursos solamente.

1. Programar un disparador DDL que monitoree cualquier actividad de drop y alter en la base de datos. Programar una auditoría con SQL SERVER PROFILER en la base de datos de **“Scholar”**. Programar un Server Audit que almacené en el registro de aplicaciones de eventos de Windows (visor de eventos de Windows) **(10 puntos).**
2. En la tabla de acta cifrar con un certificado y llaves simétricas el campo de modo; es decir, que el alumno pueda no vea si pasó en ordinario o extraordinario **(5 puntos).**
3. Diseñar e implementar índices non-clúster en la tabla de listas y acta **(5 puntos).**
4. Verificar todas las tablas posibles donde existen índices non-clúster y verificar el índice de fragmentación y mejorarlo en su caso **(5 puntos)**.