

**Instituto Tecnológico de Costa Rica, Bases de datos II**  
**Prof. Rodrigo Núñez, Caso #1 – 10%, individual**

**Objetivo:** Retomar lecciones aprendidas sobre diseño normalizado de base de datos relacional del curso anterior y poner en práctica “corner cases” relacionados a threading, performance, cache, pooling y orms.

A saber, Aseni es una plataforma social para el enriquecimiento de la democracia y la información relacionada a los planes de gobierno. Existen dos tipos de usuario, encargados de campaña y ciudadanos. Los encargados de campaña pueden hacer las siguientes acciones:

- Para el perfil del usuario, agregar un bio personal, una foto, bandera y nombre del partido.
- Plan de gobierno, el plan de gobierno se compone de N acciones no mayores a 120 palabras. Para cada afirmación el usuario deberá cantón por cantón escribir un entregable específico de esa acción en ese cantón, especificar la fecha en que eso estará listo y el kpi “key performance indicator”, esto es el número entero que se quiere alcanzar de un ente abstracto. Pueden quedar cantones en blanco y podría ser que un cantón reciba K entregables. Por ejemplo:
  - Asfaltado de la carretera barrio nuevo, 10/05/2023, 20, kilómetros (entregable, fecha, valor de kpi, ente del kpi)
  - Apertura de startups de alta tecnología, 01/01/2023, 4, startups

Proceda por el momento a diseñar una base de datos en SQL Server para Aseni, solo tomando en cuenta lo de arriba, posteriormente se extenderá el modelo en otro caso. Luego de eso proceda a llenar la data de tal forma que existan mínimo 4 partidos, 3 acciones de plan y para cada acción entre 3 a 7 entregables en cantones diferentes, esto último aleatoriamente.

Ahora es necesario que haga un conjunto de pruebas para determinar el rendimiento y las capacidades de SQL Server en su modelo diseñado bajo ciertas condiciones:

	Query 1 JDBC + stored proc	Query 2 inline query	Query 3 ORM
java	<< time in ms >>	<< time in ms >>	<< time in ms >>
.net	<< time in ms >>	<< time in ms >>	<< time in ms >>
typescript	<< time in ms >>	<< time in ms >>	<< time in ms >>
	10 threads	10 threads with pool	10 Threads with pool and cache, each thread with different cache key

**Query1:** listar todos los entregables agrupados por partido para un cantón dado por parámetro, y cada hilo usa un cantón diferente.

**Query2:** listar todos los cantones que van a recibir entregables de máximo el 25% de los partidos, agregando la cantidad de entregables por cantón

**Query3:** Para cada partido y para cada acción del plan mostrar el cantón con menos entregables y la cantidad, y el cantón con más entregables y la cantidad.

Deberá entregar el ejercicio al asistente, al correo [kevinj.2002@gmail.com](mailto:kevinj.2002@gmail.com) el día 22 de febrero a las 6pm. Incluyendo:

- La tabla con los tiempos obtenidos
- Un repositorio en Git con los programas hechos en java, .net y ts respectivamente, cada programa debe ejecutar lo que se solicita en cada query
- Demuestre en cada programa por medio de los comentarios donde se cumple que la data que se solicita es la que pide el query, cómo y dónde se configura el pool y la cache, cómo y donde entra en acción el pool y la cache.
- Diseño de la base de datos en formato PDF
- Script para generar y llenar la base de datos, un único archivo .sql que sea idempotente (se puede ejecutar N veces sin problemas de creación de estructuras o de llenado de datos duplicados)
- Cualquier sospecha de copia anulará el trabajo

**Instituto Tecnológico de Costa Rica, Bases de datos II**  
**Prof. Rodrigo Núñez, Caso #2 – 15%, parejas**

**Objetivo:** Poner en práctica métodos de arquitectura de software orientado a bases de datos que han sido implementadas usando operaciones avanzadas de SQL.

A saber, Aseni es una plataforma social para el enriquecimiento de la democracia y la información relacionada a los planes de gobierno. Ya existe un diseño hecho en el caso #1, ahora proceda a complementar el diseño con la siguiente información:

- El segundo tipo de usuario es el ciudadano, este en su perfil usando el GPS se determina el cantón donde vive o bien se ingresa manualmente, además de agregar su número de cédula
- Posterior a eso el usuario verá en una pantalla todos los entregables para su cantón por partido político
- Para cada entregable tendrá una barra donde el extremo izquierdo significa muy insatisfecho y el extremo derecho muy satisfecho, internamente siendo esa barra un valor entre 0 y 1
- De esa forma ahora el ciudadano puede valorar en primera fila que tan viable, real o valioso encuentra dicho entregable para su cantón

Amplie el diseño de la base de datos de Aseni con estos nuevos requerimientos, incremente la cantidad de acciones y entregables en un 40% y ahora proceda a implementar un api de N capas con los siguientes requisitos:

- Stack de tecnologías: nodejs, expressjs, typescript, sql server
- Layers: api, routers, middleware, controllers and repositories
- El api puede hacerse usando graphql o restfull

El api deberá tener los siguientes endpoints:

Endpoint	Implementación en SQL	Output fields
Listar los cantones que recibirán entregables en los primeros 100 días de gobierno, pero que no recibirán nada en los últimos 100	Hacer uso según se necesite de except, intersect, set difference, datepart	Cantón, cantidad de beneficios que recibirán en los primeros 100 días
Para una misma acción en un mismo partido, sacar la densidad para todos los cantones que hay en los rangos de satisfacción del primer, segundo y tercer tercio	dense_rank, pivot tables	Partido, acción, tercio1, tercio2, tercio3
Listar por año, los 3 top meses del volumen de entregables por partido que estén relacionados a una lista de palabras proporcionadas	rank, datepart, full text	Partido, año, nombre del mes, % de entregables, position

Ranking por partido con mayores niveles de satisfacción en su plan en forma global pero cuya acción tenga el mismo comportamiento para todos los cantones donde habrá un entregable. Se consideran aceptables al top 30% de las calificaciones de satisfacción.	Rank, except, intersect, pivot tables, rank	Partido, % aceptación, posición, nota máxima obtenida
Reporte de niveles de satisfacción por partido por cantón ordenados por mayor calificación a menor y por partido. Finalmente agregando un sumariado por partido de los mismos porcentajes.	pivot tables, roll up	Partido, cantón, % insatisfechos, % medianamente satisfechos, % de muy satisfechos, sumariado
Dada un usuario ciudadano y un plan de un partido, recibir una lista de entregables para su cantón y las respectivas calificaciones de satisfacción para ser guardadas en forma transaccional.	Table value parameters, transactions, read committed, transaction error handling	200 OK

Pruebe que su api funciona correctamente creando un set de pruebas y llamadas usando postman. El servidor de base de datos debe estar en la computadora de un integrante, app de node y postman en la computadora del otro integrante, esto salvo que alguien trabaje solo.

Deberá subir todos los scripts, código fuente y set de postman a un repositorio de git a más tardar el sábado 11 de marzo a las 10pm, y las revisiones serán el domingo 12 durante el día con el profesor. Cualquier sospecha de copia anulará el trabajo