

1.

Un clustered index consiste en un árbol en la cual las hojas tienen la información que se busca. Solo puede haber uno por archivo de datos ya que se crea dentro del archivo y solo puede tener una estructura de datos. Es mejor que sea un columnar storage ya que permite guardarlo en memoria.

En un non-clustered index, las hojas son punteros hacia el archivo. Se puede tener multiples non-clustered index por tabla y sobre varias columnas debido a que no está en el archivo de datos.

Si utiliza INCLUDE para agregar columnas, el índice será más grande, pudiendo llegar a tal punto que no pueda estar en memoria. Hay que definir bien el índice ya que aunque las lecturas sean más rápidas, las escrituras se volverán más lentas debido a que se necesitará revisar y actualizar el índice.

2.

El memory footprint es la cantidad de memoria que está utilizando la aplicación o base de datos. Si el memory footprint del índice es bajo, se podrá subir a memoria, haciendo que las consultas sean mucho más rápidas ya que no tiene que ir al disco a sacar la información.

Cuando la aplicación tiene un memory footprint alto, significa que está utilizando mucha memoria. Si ya no hay memoria disponible, se ocupa realizar la paginación a disco. Esto ocasiona que el rendimiento de la aplicación disminuya ya que al utilizar el disco, las consultas serán más lentas comparado a si se utilizara la memoria.

3.

Utilizaría un hash en el que la llave sería el código del país. Este hash tendría almacenado un puntero a un árbol B ordenado por el date. Sus hojas tendrán un puntero al archivo de datos, en el que tendría un clustered index sobre la columna city.

Recomendaría una base de datos jerárquica o NoSQL ya que permiten almacenar grandes cantidades de información.