

1.

Un clustered index consiste en armar un heap ordenado en el que las hojas son los datos en vez de punteros a esos datos. Solo puede haber uno por archivo de datos ya que se crea dentro del archivo. Las columnas del índice se escogen si son modificadas poco frecuente porque el árbol se ocupa acomodar por cada cambio.

Un non-clustered index consiste en un heap no ordenado en el que las hojas son punteros hacia el heap. Puede tener multiples non-clustered index por tabla, pero entre más índices tenga, menor será el rendimiento. Si utiliza INCLUDE para agregar columnas, el índice será más grande y lento porque el índice agrega las columnas en las hojas, además de las ya asignadas antes del INCLUDE, ocupando muchos más procesos ya que por cada columna que se va a recuperar, se necesita abrir el archivo para sacar la información.

2.

El memory footprint es la cantidad de memoria que usa la aplicación o base de datos. Al crearse un índice, si su memory footprint es bajo, el índice será más pequeño, por lo que podrá estar en memoria en lugar del disco, haciéndolo más rápido ya que no se necesita consultar al disco por el índice, que es más lento que consultarlo en memoria.

Cuando se tiene un memory footprint alto y el índice es tan grande que se encuentra en memoria y en disco, se ocupa realizar la paginación. Esto consiste en traer partes de la memoria a disco porque el índice completo no puede estar en memoria, por lo que el índice también consulta en el disco, gastando el espacio de 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 heap ordenado, por lo que usaría un clustered index.

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