**Ejercicio 20.2:** Discuss the decisions made during physical database design

***Decisiones acerca de indexación***

Sabemos que el *performance* de los queries en una base de datos depende en gran parte de la elección de índices en el diseño de la base de datos. Los aspectos de diseño a tomar en cuenta pueden resumirse en lo siguiente:

* Sabemos que para indexar un atributo se debe cumplir que, o este es una llave o es utilizado por un query de selección condicional o de unión.
* Los índices pueden ser compuestos, es decir, conformados por más de un atributo. Por lo tanto, debemos estar atentos a atributos que, para varios queries, actúan juntos.
* Si una tabla requiere varios índices, las decisiones de cuál índice debe ser primario y cuál debe ser *clustered*, dependen en si la tabla debe estar ordenada por ese atributo o no.
* La decisión correcta para escoger la estructura del índice, la cual puede ser un *hash index* o un *tree index*, depende de si los queries a ejecutar serán de igualdad o unión. Si se necesitan los dos tipos, la opción correcta es usar un *tree index*. Si se necesita solo ejecutar queries de igualdad, el *hash index* debe escogerse.
* Si nuestros archivos tienden a tener un crecimiento continuo, se debe considerar la opción de utilizar un hashing dinámico.

**Ejercicio 22.2** Prove that strict two-phase locking guarantees strict schedules.

Por propiedad, se dice que una transacción sigue el protocolo Two-phase locking cuando todas las operaciones de bloque, *read\_lock*  y *write\_lock*, deben ocurrir antes de la primera operación de desbloqueo *unlock*. Dicho esto, supongamos que T1 y T2 son transacciones que cumplen las propiedades de el Two-phase locking.

T1 pide los bloqueos *read\_lock*  y *write\_lock* y, al estar disponibles, se le otorgan. Al mismo tiempo, T2 pide permiso para bloquearse. Como T1 tiene los bloqueos tomados y no ha los ha desbloqueado, T2 se verá obligado a esperar a que T1 desbloquee los *locks* para poder otorgársele los permisos de bloqueo. De esta manera, se garantiza que los schedules serán completamente serializables, ya que solo una transacción tendrá la sección crítica y ninguna otra puede acceder a ella hasta que la transacción que esté adentro termine sus operaciones.