

Hola,

No soy seguro de comprender bien todo tu mensaje.

Si me preguntas si es posible de trabajar con una sola tabla, la respuesta es sí. Piensas a una hoja Excel donde cada columna corresponde a un campo de los datos de tu aplicación. Cuando no existe una información, la celda quedar vacía. Y cuando viene la información, completamos la celda.

Por ejemplo, cuando no hay información por los columnas corresponden a tu tabla "Actas", estas celdas quedar vacías y también las celdas de las columnas FEntrega y FActa qui existen una sola vez en la hoja XLS, pero 2 veces en tus tablas no organizadas.

Para consultar o tratar, es siempre leer, o agregar, o actualizar, o suprimir una línea (o todas líneas), con varios campos (o todos campos), correspondiendo a un valor que gustamos (clausula WHERE).

No digo que la hoja XLS es buena solución, pero esta es más fácil a programar. Su gran defecto es la redundancia: más de octetos, más de trabajo para las correcciones. Las correcciones están más fáciles en Excel porque hacemos Buscar-Reemplazar sobre una la columna que queremos. En la aplicación VB, debemos programar las herramientas que permiten Buscar-Reemplazar sobre un campo para corregir.

Yo también, he escrito aplicación con los datos en una sola tabla, o mismo con los datos registrados en un hoja XLS. Lo he hecho porque están pequeñas DB, menos que 500 líneas de 30 hasta 50 campos, y sobre todos, estada para hacer rápido, sin análisis, porque soy un poco perezoso....

Si las tablas están organizadas según un análisis relacional (hablamos de un DB normalizada, acordada a una norma), no es necesario el Buscar-Reemplazar para corregir, un dato existe una sola vez en la DB. Si contiene un error, corregimos directamente, una sola vez. Más importante, la DB completa tendra menos de octetos que la hoja XLS o que tu DB.

Por ejemplo; en tu tabla SANCIONES, veo un dato ARIAS CHOQUEHUANCA ABRAHAN. Supongo que es el nombre del dueño del vehículo. Si existe una sola vez en toda la DB, perdemos octetos con el sistema normalizado. Pero, si existe 2 veces o más, estamos ganadores. En tu DB, este nombre existe en SANCIONES y en ACTAS, y quizá en otro lugar (supongo que la DB que me envías es solamente una parte del todo).

Explico este ejemplo.

ARIAS CHOQUEHUANCA ABRAHAN contiene 26 caracteres. En tus ambas tablas, este nombre tiene 52 caracteres.

En la DB normalizada, existe una tabla TPersonas donde este nombre es registrado con un IdPersona que es un numérico (Auto o no) de 8 octetos (8 octetos si Long, 4 octetos si Integer, 2 si Short, no siempre necesario de utilizar Long. Por ejemplo, una tabla TTipoVehiculo podría tener un IdTipo de tipo Byte, porque hay no más que 256 tipos diferentes de vehículos).

Bueno, esta persona ARIAS CHOQUEHUANCA ABRAHAN tiene 34 octetos (26 + 8) en TPersonas y donde debemos referenciar esta persona, utilizamos un duplico de su IdPersona, pues 8 octetos. Cada vez que utilizas 26 octetos para esta persona, aquí utilizamos 8 octetos (CopIdPersona). Si es necesario de referenciar en SANCIONES y en ACTAS, tenemos 8 octetos en SANCIONES y 8 octetos en ACTAS, y 34 octetos en TPersonas, pues un total de 50 octetos. Ganamos 2 octetos, no es mucho, pero es solamente 2 referencias de una misma persona. Si esta persona es referenciada un vez de más, ganamos 20 caracteres, y más aún si más referencias....

En verdad, el nombre no es limitado a hasta 26 caracteres, pero 128 o 256, según la tamaño que damos al campo. En verdad, el nombre no es limitado a hasta 26 caracteres, pero 128 o 256, según la tamaño que damos al campo. Pero el principio del numeroso de octetos es mismo.

Y sobre las correcciones, cuando debes corregir la ortografía de este nombre, debes lo hacer 2 veces con tus ambas tablas. Una sola vez con la DB normalizada.

Sobre tu aplicación, no conozco todo el problema pero veo que existe ninguna normalización, ni ahora, ni en las partes anteriores. Si no tiene obligación sobre la tamaño del DB, sería más simple de hacer una sola tabla para todo. La herramienta de corrección (Buscar-Reemplazar) sería más fácil a hacer, es bastante de dar el malo valor, el bueno valor, y el nombre del campo, y escribir un "UPDATE LaTabla SET EICampo = EIBuenoValor WHERE EICampo = EIMaloValor".

El solo "malo" del DB normalizada es que cada tabla de descripciones (TPersonas, TTipos, TMarca, TColor, ...) necesita su gestión particular (Form, posibilidad de crear, consultar, modificar, suprimir).

Tambien, las consultas SQL (SELECT) sobre las tablas en relación es un poco más complejas.

El **SELECT ... FROM UnaTabla ...** cambia en **SELECT ... FROM UnaTabla WHERE (SELECT ... FROM OtraTabla ...) ...** o tambien **SELECT ... FROM UnaTabla, OtraTabla INNER JOIN ...**

A ti de ver. Tu aplicación ya es bien desarrollada, ya registras y consultas muchas informaciones. Si no tiene error, quizá es bien de continuar como has comenzado.

Espero este te ayudará ...