

Hemos dado los primeros pasos, oops como en todos primeros pasos hay algún tropiezo, creamos una tabla en la base de datos equivocada, aprovechemos para aprender a borrar una tabla y crearla en otro sitio todo esto es práctica y es muy interesante que lo veamos, hablemos también por ejemplo de restricciones o constraints, podemos hablar de datos de tipo y unique, únicos, podemos crear claves para que nuestra tabla esté ordenada y para no violar un principio fundamental de las bases de datos y es que no exista de ninguna manera un registro duplicado, en fin, estas cositas que son muy comunes en el mundo de las bases de datos pero que tenemos que aprender, vamos a aprenderlo. Y si ahora pedimos que nos muestre las tablas `show tables`, bueno nos dice que aquí están las tablas en la base de datos personas, aquí está nuestra tabla usuarios, no contiene información todavía eso, ya lo sabemos, entonces vamos a pedir que nos describa, `describe usuarios` y debería mostrarnos la estructura que acabamos de crear, ahí está, un Id, un nombre y un apellido, muy bien, supongan que queremos cambiar la estructura de esta tabla, queremos agregar por ejemplo una nueva columna, email por ejemplo, cualquiera no importa, bien, para eso vamos a utilizar la sintaxis `alter table`, este es el comando `alter`, o sea, alterar, modificar una tabla y luego de esta instrucción tenemos que indicar por supuesto cuál es la tabla que queremos modificar, se llama usuarios la que acabamos de crear, ok, y ahora tenemos varias posibilidades para modificar, podríamos decir reitero aquí no hay que hacer un entero, ok, se puede continuar escribiendo todo aquí en línea, a mi mentalmente me ayuda mucho a hacer este espacio, podríamos indicar que queremos agregar, que queremos agregar, bueno una columna, un campo, una columna que se llame email que sea de tipo `varchar`, 255 esa es su longitud y aquí simplemente yo tendría que dar enter, voy a poner un punto de coma sólo por costumbre, control enter, todo ha ido bien, vamos a ver ahora otra vez la estructura de nuestra tabla usuarios con `describe usuarios`, deberíamos ver que apareció el campo email, perfecto ahí está nuestro campo, bien, cómo hacemos para eliminar una columna de una tabla, bueno, vamos a mostrarles otra vez esto también es con el comando `alter table` yo quiero conservar el campo email por lo tanto voy a agregar uno cualquiera para luego borrarlo, por ejemplo, y de paso repasamos otra vez cómo agregar un campo, `alter table usuarios ADD test int`, y aquí un campo que se llame test, va a ser de tipo `int`, y no le voy a poner longitud, le doy a enter, ahora vamos a describir la tabla para que ustedes vean que el cambio efectivamente se ha efectuado, `describe usuarios` y aquí está, bien, agregado mi campo test, noten que cuando al campo `int` no le pongo una longitud por defecto me agrega 11, bueno, voy a borrar ese campo, quiero borrar, como hacemos otra vez, con `alter table usuarios drop test` pero ahora y reitero no es necesario escribirlo en mayúsculas simplemente lo hago por costumbre y aquí entonces podríamos indicar que queremos borrar un campo, como borramos, con `drop` pero primero debo decir claramente sobre qué tabla estoy trabajando, bueno, sobre la tabla usuarios y ahora diré `drop table usuarios` sí, y qué es lo que voy a borrar, bueno, una columna, `drop column`, `drop column` y tengo obviamente que escribir el nombre de la columna, como se llama, test recuerdan, la agregamos recién y bueno solamente la habíamos agregado para borrarla cosa que vamos a hacer ahora mismo, realmente desea ejecutar `alter table usuarios`, etcétera, etcétera, cada vez que borró algo me pregunta, por qué borrar una

columna significa modificar la integridad de los datos, eso es lo que hace My SQL, por eso me hace estas preguntas, noten que cuando creó algo no me pregunta nada pero sí cuando destruyó, por qué, no por qué le preocupe mucho la estructura de mi tabla sino que le preocupan los datos que pueda contener, ok, vamos a decirle que sí, que está todo bien, borro, la hasta luego y se ha borrado, seguramente ustedes no necesitan que lo comprobemos pero ya que estamos vamos a comprobarlo usuarios y test no está más, muy bien, pero también podemos por último utilizar alter table para modificar, supongan que nos han avisado desde el sector donde hicieron la arquitectura de nuestra base de datos que ha habido un error, que no es necesario que email tenga 255 caracteres sino, bueno, 128 por poner un número, ok, entonces vamos a modificar, vamos a otra vez alterar nuestra tabla, alter table, usuarios y vamos a escribir modify, modify, ahora lo escribo al lado pero recuerden que esto es indistinto, hasta ahora veníamos escribiéndolo abajo por una cuestión de practicidad, a mí me parece más práctico ustedes pueden hacerlo como mejor les acomode, bueno, ya que lo puse abajo lo dejo así, que queremos modificar una columna, modify column y ahora el nombre de la columna, bueno, nuestra columna se llama email, es de tipo varchar, podríamos modificar el tipo, va a seguir siendo de tipo varchar pero no 255 sino de 128 y control enter y esto se ha modificado, seguramente ustedes me creen pero por las dudas, describe, usuarios y veremos que ya impacto aquí el cambio, muy bien, ahora hablemos de las restricciones o constraints, las restricciones pueden ser especificadas cuando se crea la tabla con la opción create table o después de creada, con el comando alter table, las restricciones básicamente nos permiten especificar reglas para los datos en la tabla, se usan para limitar el tipo de datos que pueden ir dentro de la tabla, esto asegura la precisión y confiabilidad de los datos, si hay cualquier violación entre una restricción y una acción sobre los datos la acción es anulada, por ejemplo, podemos indicar que un campo no admita valores nulos, es decir, que sí o sí haya que tipear algún valor para ese campo, en tanto que si no indicamos lo contrario, bueno el campo podrá quedar vacío, fíjense aquí nombre, apellido, e-mail todos tienen la columna aquí “null” marcada como yes, es decir, que admiten valores nulos, queremos decir que el campo de tipo email no pueda ser nulo y creemos que sí o sí contenga un valor, que estén obligados a incluir un valor, bueno, podemos hacerlo con alter table, ok, como lo vamos a hacer esto, bueno, reitero podemos hacerlo cuando creamos la tabla pero como la tabla ya está creada vamos a modificarla, vamos a hacer a alter, table, el nombre es usuarios y ahora vamos a indicar por ejemplo, el campo, tendríamos que indicar el campo que queremos modificar, aquí por ejemplo, vamos a decir modify, hay que poner un espacio, otra vez esto de corrido o abajo como ustedes quieran, esta vez lo voy a dejar aquí, modify, email, el campo email y ahora vamos a decir, bueno, otra vez que es varchar 128 y vamos a agregar not null, ok, si ahora miramos otra vez con describe, usuarios, veremos que a este campo la propiedad null ya se le ha cambiado, este campo estaba como not null, no admite valores nulos, bien ok, espero que eso les quede claro pero además tenemos otra restricción que es la restricción de tipo unique, esto bueno hace que por ejemplo un dato no pueda estar repetido, podríamos indicar por ejemplo que queremos que apellido no estuviera repetido, en la vida real esto no tendría mucho sentido porque puede haber, digamos, más de un usuario con un apellido, por lo tanto, bueno miren vamos a hacer algo más parecido a la vida real, agregamos un campo que se llame username o nombre de usuario o simplemente usuario, sí, y ahí sí tiene sentido que no haya dos usuarios con el mismo nombre, o sea, alter table se me ocurre que podríamos

cambiar la tabla, alter table aquí sería ADD verdad, para agregar una columna, pero primero pongamos el nombre de la tabla usuario y ADD, que queremos aquí agregar, bueno, una columna que se llame usuario, usuario de tipo varchar y bueno 32 no vamos a admitir que sean tan largos los nombres de usuario, esto como máximo obviamente no, aquí le damos a enter y ya tenemos un nuevo campo, entonces ahora sí, bueno aquí mismo podría haber podría haberlo hecho en un solo paso, podría haberlo hecho en un solo paso, agregando aquí además luego que ese campo es de tipo unique, o sea, que no puede haber dos campos que tengan el mismo nombre de usuario, pero bueno no me di cuenta de hacerlo así que vamos a hacerlo ahora, vamos a cambiarlo, vamos a hacer alter table, aquí sería modify, vamos a modificar, vamos a modificar el campo usuario que es de tipo varchar, 32 pero vamos a indicar que sea de tipo unique, es decir, que no puede haber dos usuarios con el mismo exacto nombre pero también le vamos a decir que sean not null, podemos encadenar estas posibilidades porque también vamos a decir de esta manera que no queremos, vamos a ponerlo en este orden, no queremos que haya algún registro que no tenga un nombre de usuario, es decir, ya lo hicimos con email ahora también lo hacemos con usuario, o sea, el usuario podrá poner o no su nombre y su apellido pero si o si tendrá que escribir un email y tendrá que escribir un nombre de usuario y además el nombre de usuario no podrá ser igual al de otro, le damos enter y la tabla ya se ha modificado, si ahora hacemos un describe de nuestra tabla, usuarios veremos que estas constraints, estas restricciones, ya se están aplicando, ya sabemos que el usuario no puede ser nulo y además es de tipo unique, claro aquí al ponerlo como de tipo no nulo y único nos dice que se ha convertido en nuestra clave primaria pero eso lo vamos a cambiar de inmediato porque en realidad nuestra clave primaria, es decir, el ID que va a identificar de manera unívoca a un registro para diferenciarlo de otro va a ser, bueno, en este campo, el campo ID, bien, aquí ha tomado la decisión por nosotros la base de datos, el administrador de bases de datos y está muy bien porque usuario al ser un campo que no puede estar vacío y que es de tipo único, es decir, que no puede estar repetido bien podría ser nuestra clave única, ok, pero lo que haremos nosotros ahora será indicar que ID en realidad va a ser nuestra clave única, bien ok, vamos a hacerlo de esta manera, vamos a indicar por ejemplo aquí, vamos a cambiar la tabla, alter, table, la tabla se llama usuarios y ahora lo que haremos será agregar una clave única, una clave primaria, una primary key, ok, pero vamos a definir nosotros cuáles, no dejaremos que MySQL cuál lo haga por nosotros, aunque lo que hizo hasta ahora está bien, no está mal pero nosotros lo vamos a mejorar tomando control sobre el ID porque cuando esta tabla fue diseñada se pensó que ID iba a ser el mejor campo candidato para hacer el campo clave de primary key, es decir, la clave única de nuestra tabla, porque no habrá de ninguna manera dos claves iguales, tampoco habrá dos usuarios, dos nombres de usuario, pero es mejor según lo entendemos o según entendió quien diseñó esta tabla supongamos el arquitecto de base de datos que nos dio la especificación para la creación decidió que el ID sea el campo clave para primary key los cuales por otra parte es muy común en la vida real verdad, bien, entonces aquí lo que haremos ahora es vamos a agregar ADD, primary, primary key y vamos a indicar el nombre del campo que va a ser primary key bueno, cuales es, ID el campo ID, apretamos enter y esto está terminado, ahora simplemente si volvemos a inspeccionar con describe, usuarios, bueno ahí quedó en minúsculas, describe van a ver que efectivamente no hace la diferencia y ahora tenemos, fíjense aquí que se produjo un cambio, usuario está como unique porque así lo definimos en

primera instancia sólo que como no teníamos seteada una clave primaria SQL decidió con buen tino por otra parte que éste iba a ser el campo que identificaría de manera unívoca a un registro de otro, ahora nosotros al identificar claramente que esta es nuestra clave primaria, bueno pues la cosa ha vuelto a la normalidad, es decir, usuario está como que no admite nulo y además es de tipo único, noten aquí que ID se ha puesto como no en nulo, es decir, no admite nulos nosotros no hicimos eso, claro, pero al ser la clave primaria evidentemente de ninguna manera puede quedar este valor vacío, si, eso tiene que estar bueno pues muy muy claro, bien, obviamente nosotros también podemos quitar restricciones que hayamos puesto aquí, podríamos por ejemplo en algún momento decidir que el apellido fuera un campo único, aquí bien, podríamos definirlo y si luego quisiéramos eliminar eso porque la verdad es que no tiene mucho sentido, la verdad que no tiene mucho sentido que un campo apellido, ya lo contamos hoy no, sea de tipo unique, bueno, podríamos borrarlo, vamos a hacerlo esto, vamos a agregarlo solo para luego quitarlo, bien, como lo vamos a hacer aquí vamos a poner otra vez alter table, bueno aquí ya estoy mezclando minúsculas con mayúsculas, no importa igual funciona perfectamente, no hay ningún problema, es la tabla usuarios y aquí lo que vamos a hacer ahora es agregar una restricción de tipo unique, bien, vamos a indicar por ejemplo ADD, unique y vamos a indicar cuál es el campo, también podemos hacerlo de esta manera, mucho más rápido, vamos a indicar que unique que es el campo, apellido verdad, apellido, perfecto, esto ya está hecho describe, usuarios y ahí tenemos como unique apellido y decimos oops aquí nos hemos equivocado, esto va a ser un problema, seguramente habrá más de un apellido similar y entonces qué ocurre cuando alguien intente ingresar un registro nuevo y el apellido ya exista en otro registro la base de datos no le va a permitir ingresar el nuevo registro y eso está mal, por lo tanto ahora decidimos que queremos eliminar esa restricción, ese constraints, bueno, podemos hacerlo perfectamente, vamos a eliminar una restricción esto podemos hacerlo aquí muy fácilmente, vamos a otra vez utilizar la sintaxis alter table, muy bien, tenemos que indicar cuál es el nombre de la tabla, usuarios y ahora vamos a indicar lo siguiente para borrar una restricción de tipo unique, vamos a indicar drop, index y cuál es el campo que queremos modificar, es decir, que no deseamos que tenga ya la restricción de tipo unique, es el campo apellido, apellido, realmente desea ejecutar, etcétera, etcétera, si, lo desea, muy bien, y ahora si vemos nuevamente con describe y user, perdón los usuarios, vamos a ver que el campo apellido ya no tiene la restricción de tipo unique, ok, así que ya hemos aprendido cómo poner restricciones, como quitarlas, como digamos trabajar con ellas, solamente nos falta algo que es muy importante hemos decidido que ID sea nuestra clave primaria y que no admita valores nulos pero también queremos para que esto sea totalmente completo que su valor se actualice automáticamente, que el usuario no tenga que poner un número de ID, que cuando se crea un registro el primero sea 1, el segundo 2, etcétera que la base de datos se encargue de eso por nosotros, lo cual es una muy buena práctica en este sentido, si, no dejar que el usuario ingrese un ID porque podría colisionar con otro que ya exista y no se rompería nada porque como ya tenemos la restricción aquí de clave primaria el registro nuevo directamente no se crearía, pero no queremos que ocurra eso, queremos que cuando se crea un registro, bueno, no haya ningún problema, que se cree y ya está, para eso dejamos en manos de la base de datos la creación de el ID correspondiente, le vamos a decir que además de ser clave primaria sea de tipo auto incremental, bien, eso obviamente todo esto lo podemos hacer cuando creamos las tablas de la misma forma que,

con los mismos comandos que vemos aquí pero bueno lo vamos mostrando como sobre una tabla que se creó en principio muy pequeña para que ustedes vean en detalle qué hace cada una de las instrucciones, bien, a ver entonces, con esto ya terminaremos aquí lo que tendríamos que hacer simplemente es bueno nuevamente alterar nuestra tabla, alter table, el nombre de la tabla, usuarios, sólo para ser un poco más ordenados voy aquí abajo y vamos a modificarlo, vamos a modificar un campo que ya existe que es el campo ID que es de tipo entero ya es una primary key por lo tanto esto no hay que no hay que agregarlo aquí porque eso nos daría un problema, ya está seteado como clave primaria pero le falta para ser completo el constraints o la restricción auto incremento, ok, y si ahora miramos aquí con describe, table, bueno, en realidad el nombre de la tabla directamente, usuarios, venimos acá abajo y vemos que ya nuestro ID además de ser clave primaria es de tipo auto incremental, ya tenemos lista nuestra estructura, a partir de la próxima clase comenzaremos a hacer las consultas, cargar información, verificarla, modificarla y borrarla.