implementando nuestro propio proyecto

Trabajo de Alejandro Sainz Sainz

BD-ACTIVIDAD 2.3

Contenido

[CREANDO EL SCRIP DE NUESTRO PROYECTO 2](#_Toc185614617)

[AGREGANDO TABLAS DEPENDIENTES 8](#_Toc185614618)

[COMPROBACIÓN DE TABLAS Y DIAGRAMA 9](#_Toc185614619)

[MÁS RELACIONES RESULTAS 11](#_Toc185614620)

[AVANZANDO CON LA RESOLUCIÓN 19](#_Toc185614621)

[ULTIMAS RELACIONES A RESOLVER 21](#_Toc185614622)

[DIAGRAMA FINAL 32](#_Toc185614623)

[MODIFICACIONES 35](#_Toc185614624)

[MI DIAGRAMA RELACIONAL 36](#_Toc185614625)

[COMENZANDO CON CAMBIOS Y AJUSTES 36](#_Toc185614626)

[DIAGRAMA VIEJO 40](#_Toc185614627)

[DIAGRAMA NUEVO 41](#_Toc185614628)

[TERMINANDO DE MODIFICAR EL DIAGRAMA 42](#_Toc185614629)

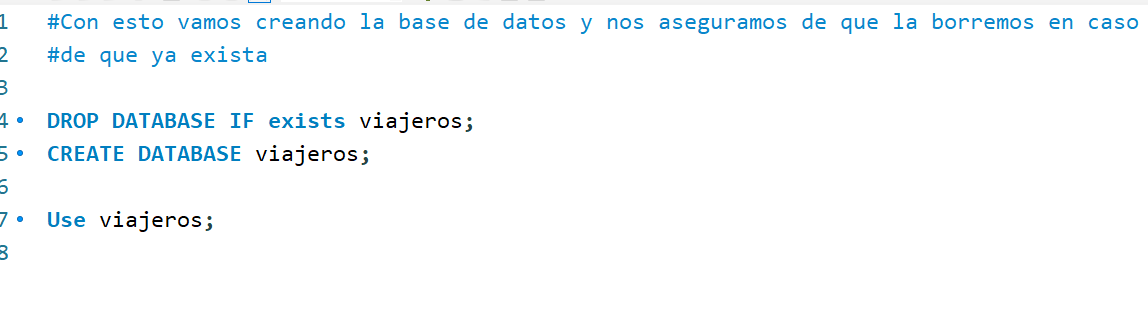
[MODIFICANDO LOS ATRIBUTOS 42](#_Toc185614630)

[COMPLETANDO EL DIAGRAMA 50](#_Toc185614631)

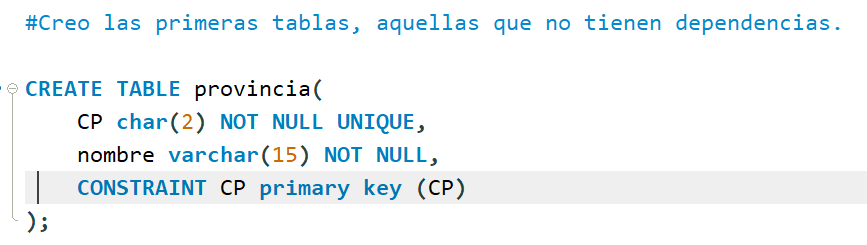
[DIAGRAMA FINAL 56](#_Toc185614632)

# CREANDO EL SCRIP DE NUESTRO PROYECTO

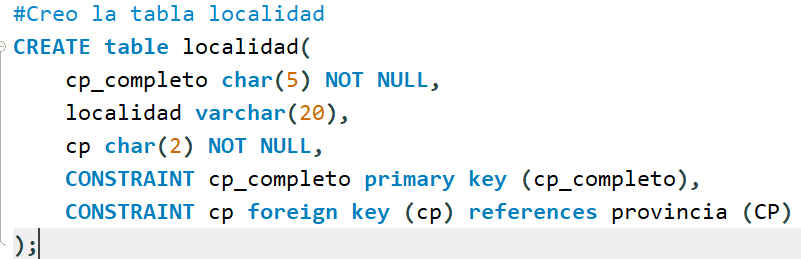
Lo primero es ir creando la BD en el script, e ir fijándonos en cuales son las tablas que no poseen dependencias para ir creándolas primero.



Lo primero que voy a hacer es crear la tabla provincia, que no tiene dependencias.

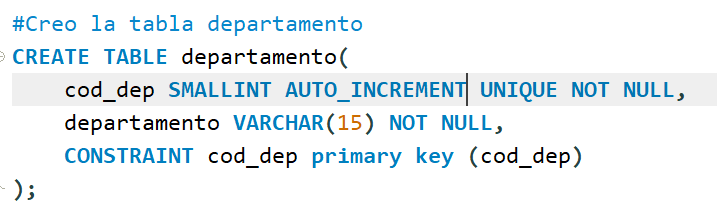


Ahora puedo crear la tabla localidad, ya que solo depende de la tabla provincia, así puedo dejar esa parte cerrada y seguir con otras tablas que no tengan dependencia.

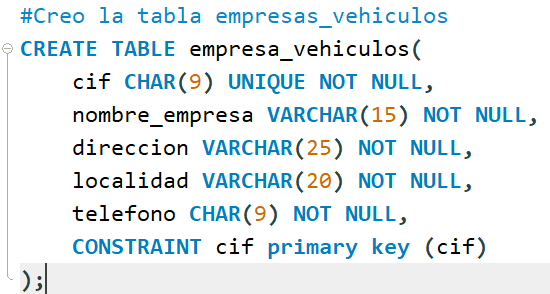


Al tener completadas estas dos, como luego necesito localidad para otras dos tablas como FK, ya puedo despreocuparme por ellas.

Voy a seguir con otras dos tablas independientes, que son la tabla departamentos y la tabla empresa vehículos.

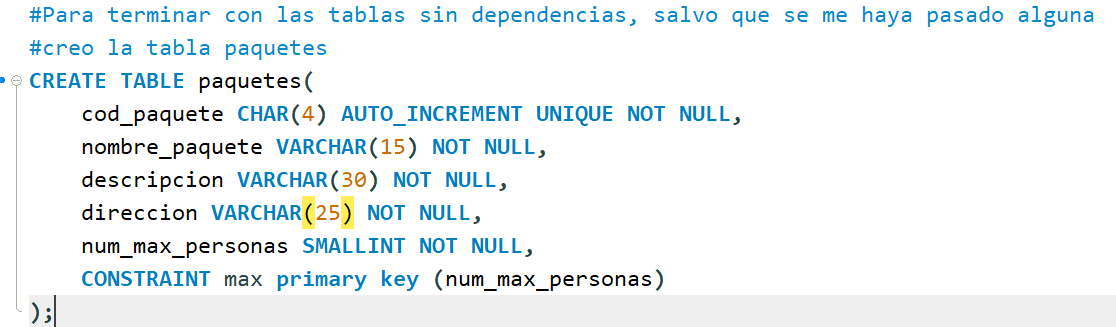


Esta es la parte del script correspondiente a la tabla departamento.



Con esta parte del script ya tengo terminada la tabla de empresas de vehículos de alquiler.

Ahora, si no me equivoco, creo que la única tabla que me quede de añadir que no tiene dependencias es la tabla paquetes, aquella que recoge los paquetes y su descripción.



Pues esto sería la tabla paquetes. Intento siempre ajustar los valores de los atributos a lo que yo creo que sería realista, por ejemplo, cod\_paquete CHAR(4), ya que no creo que el código de paquete ocupe más de cuatro caracteres.

Intento que el atributo dirección sea de la misma longitud en todas las tablas que lo incluyen, por coherencia más que nada.

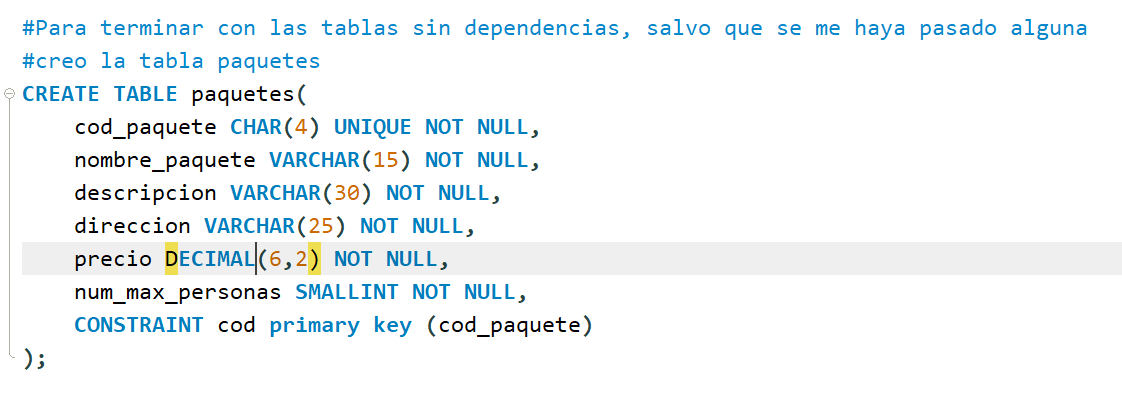
En el caso de num\_max\_personas he decidido que sea SMALLINT, simplemente por una cosa. Imaginemos que la empresa escala o necesita otra serie de cosas, como calcular media de clientes por actividad, etc, como incluir el precio de un paquete por persona, etc.

Ahora que lo he comentado, voy a añadir el precio de un paquete, completo, sin contar por grupo o con descuentos, que creo que así completo un poco al modelo.



Le añado el campo precio como ya dije antes.

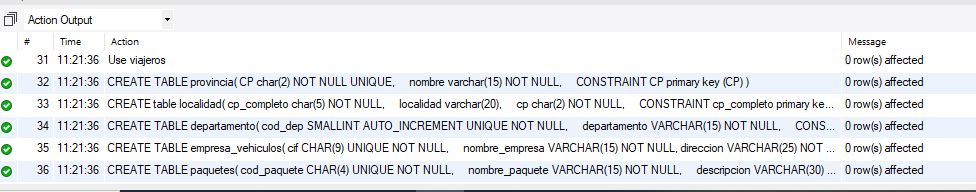
Llegado a este punto voy a ejecutar el script simplemente para ver que no da errores.



Gracias a hacer las pruebas de ejecutar el script como comprobación me di cuenta de un par de errores.

1. Puse un campo char como auto increment, cosa bastante improbable.
2. Me indicaba que marcar un campo como float era un practica deprecated y como indicó el profesor en clase lo cambie por decimal.

Una vez realizados los cambios el script quedó así y esto es lo que recoge la ejecución del scritp.



Vemos que ya no da ningún error o warning, así que ahora vamos a realizar un intento de ingeniería inversa, para ver como va quedando el diagrama, aunque me imagino que por ahora será bastante sencillo.

Como última comprobación, me aseguro de que se genera la parte del diagrama correspondiente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteMe acabo de fijar que la tabla alojamientos no depende de ninguna otra, según el modelo que yo generé, así que voy a crearla ahora.

Con esto, ahora sí, creo que ya tengo completadas todas las tablas independientes.

Vuelvo a ejecutar el script para comprobar todo de nuevo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ningún warning o error, así que, ahora sí, puedo hacer una prueba para ver cómo va quedando el diagrama.

Diagrama

Descripción generada automáticamente

Por ahora va quedando como esperaba, aunque no era muy difícil, ya que todavía no existe ninguna relación en lo que he creado, salvo en el caso de localidad y provincia.

En este punto, tengo que empezar a crear las tablas que son dependientes de otras, e ir viendo como quedan en comparación con el diagrama que fui desarrollando yo en la realización del ejercicio.

Creo ahora la tabla grupos, que me acabo de dar cuenta en una parte posterior del ejercicio que está sin crear.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

En este caso, he decidido darle a la clave principal, id\_grupo, un tipo int que se vaya incrementando según vayamos creando los grupos y a num\_personas un varchar(2) ya que no creo que el tamaño de un grupo esté en ningún caso por encima de 99.

Ahora que me doy cuenta, si quiero calcular medias de clientes y grupos, creo que es mejor cambiar el tipo de num\_personas a int, por si en algún caso deseo realizar operaciones con esos datos.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Lo dejo así.

# AGREGANDO TABLAS DEPENDIENTES

A partir de ahora, hay que crear las tablas que reciben PK de otras tablas como FK. Estas se crean a partir de este momento, pues en un inicio, si no se han creado las tablas de las que se reciben las PK como FK el programa lo reconoce como un error.

Creo que, en este caso, la más importante va a ser la tabla sedes, que es la tabla central de nuestro ejercicio.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Este sería el fragmento de script que corresponde a la tabla sedes.

En un mismo orden de cosas, tengo la tabla clientes, que tiene la misma dependencia, de la tabla localidad, y como ya mostré en el informe anterior, se pude incluir en el mismo ejemplo.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Pues aquí tenemos creada la tabla clientes.

## COMPROBACIÓN DE TABLAS Y DIAGRAMA

Para comprobar que estas dos tablas y la relación con la tabla localidad y provincia queda como resolví en el ejercicio anterior voy a ejecutar el script y la ingeniería inversa, y compararé el resultado.

Diagrama

Descripción generada automáticamente

Este diagrama es como quedó resuelto mi ejercicio, vamos a comprobar como queda en el Workbench.

Lo primero, al ejecutar el script, me da un error, hay dos FK con el mismo nombre dentro del modelo, cambio la FK de la tabla clientes por cp\_comp, para que no se repita con la FK de la tabla sedes. Después de eso, ya puedo comprobar el diagrama.

Diagrama

Descripción generada automáticamente

Si lo comparamos con la resolución del ejercicio vemos que quedan iguales.

Ahora prosigamos con los siguientes ejemplos.

## MÁS RELACIONES RESULTAS

Ahora voy a ir resolviendo el caso de los empleados, que creo que lo hice mal en el ejercicio anterior. En el diagrama E/R indiqué una herencia excluyente.

Sin embargo, al crear las tablas en ese mismo ejercicio, creé una tabla departamentos, y luego pasé la clave del departamento como FK a la tabla empleados, así que no sé si está bien resuelto como herencia, aunque como dos tablas independientes que se relacionan creo que si está resuelto de una forma correcta, así que optaré por esta segunda opción.

Diagrama

Descripción generada automáticamente

Esta es la forma en que lo resolví en el ejercicio anterior, y esta es la que voy a seguir al crear la base de datos en Workbench.

La tabla departamentos ya la tengo creada en el apartado anterior, de la misma forma que clientes, así que ahora sólo me queda crear la tabla empleados y relacionarlo todo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteEsta es la creación de la tabla empleados, que es una de las más largas que tenía en el ejercicio. De nuevo voy a ejecutar el script y a ver el diagrama en Workbench.

De nuevo compruebo que al ejecutar el script no hay fallos (no incluyo captura para no alargarlo todo demasiado).

Diagrama

Descripción generada automáticamente

La colocación es diferente a la que creé yo en draw.io pero el resultado es el mismo.

Continuo resolviendo tablas que tienen que ver con sedes, en este caso voy a resolver la relación entre sedes y paquetes. Como la relación era N:N debo de crear la tabla intermedia.

Diagrama

Descripción generada automáticamente

Como se ve aquí, en vez de crear un código para la relación, use las dos FK para crear un PK compuesta. Vamos a resolverlo en el Script.

Diagrama

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamenteEste es la porción de Script. Veamos el resultado.

Las dos entidades quedan relacionadas así. Prosigamos.

Me acabo de dar cuenta de que tenía otra tabla independiente que tenía que crear (esto me pasa por dejarme llevar y hacer un ejercicio muy largo).

Lo que voy a hacer es volver al apartado anterior a crearla y ponerlo allí y después resolver la relación correspondiente.

La tabla en cuestión es la tabla grupos.

Una vez subsanado esto, procedo a crear la tabla intermedia tal y como resolví el ejercicio anterior.

Diagrama

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Creando nombres distintos para la PK y las FK se genera esta tabla. El script se ejecuta de forma correcta. Vamos a ver el diagrama, que ya se empieza a liar un poco.

Diagrama

Descripción generada automáticamente

Ya tengo la relación resuelta, y como podemos observar, este fragmento del diagrama se parece bastante a la resolución del diagrama que yo creé.

Ahora que ya tenemos resulto el tema de los grupos, podemos pasar a resolver las reservas, que incluye todo esto que he hecho hasta ahora.

Diagrama

Descripción generada automáticamente

Como ya tengo creada la tabla grupos, puedo proceder a crear la tabla reservas. Vamos a ello.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

De esta forma he creado yo la tabla. Fecha\_fin\_reserva permito que sea null por si en la reserva, por el motivo que sea no se especifica fecha de fin. Incluyo el campo precio, un decimal, grupo completo (que sirve para indicar si un grupo completa el número de personas máximas que permite la actividad a realizar) indico que es del tipo boolean y por defecto su valor será false.

Después de esto indico la PK y la FK que forman parte de esta tabla. Al ejecutar el script todo correcto (que no quiere decir que esté bien, pero que se ejecuta sin ningún problema evidente).

Voy a crear ahora la tabla intermedia que usé para terminar de resolver todo este bloque de relaciones.

Esta tabla se llama actividad, ya que sería un listado de las actividades que se llevan a cabo como parte de lo contratado por un grupo en una reserva, indicando los paquetes que están contratados en una reserva en concreta. Vamos a ello.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteAsí es como resuelvo la creación de esta tabla intermedia. La ejecución en principio sin problema, veamos el diagrama para confirmar.

Diagrama

Descripción generada automáticamente

En principio todo correcto, el único problema, es que ya se empieza a hacer complicado que no haya líneas de relación por todas partes. Intento colocarlo todo de una forma en la que se vean de forma clara cuales son las líneas que corresponden a la relación resuelta en cada caso. Lo que me he dado cuenta al revisar, es que el campo boolean me le cambia el propio programa por un tinyint, que ya había visto en los apuntes que se considera lo mismo.

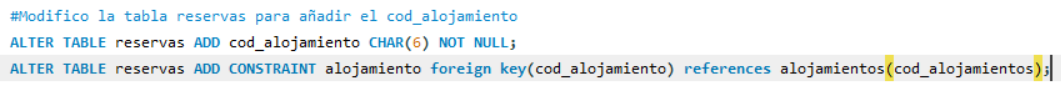
## AVANZANDO CON LA RESOLUCIÓN

Ahora que ya hemos ido resolviendo gran parte de lo que es el núcleo de nuestra BD, vamos a seguir con el resto de las relaciones.

Bien, este caso que viene ahora, lo dejé yo para este momento para probar comandos.

Como sabemos, en mi ejemplo, en la tabla reservas también se indica que alojamiento corresponde a cada reserva. Vamos a incluir eso ahora, y así tengo que usar comandos para modificar la tabla reservas.

Voy a usar el comando para añadir el cod\_alojamiento a la tabla reservas.



Con estos dos comandos, primero añado el nuevo campo a la tabla, después ya le añado a la tabla el CONSTRAINT indicando que ese nuevo campo es la foreign key y a que referencia. No se si se podría hacer en una sola línea, pero creo que así también es válido. Vamos a comprobar ahora el diagrama.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El atributo se ha añadido y se ha generado la relación, así que por ahora todo correcto.

Diagrama

Descripción generada automáticamenteDiagrama

Descripción generada automáticamente

Vamos a continuar con la parte de los alojamientos. En el ejercicio dije que las sedes se encargan de las reservas de los alojamientos, así que vamos a implementar eso en nuestra BD.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Aquí creo el código para esta tabla intermedia.

Texto

Descripción generada automáticamente con confianza media

Aquí el resultado de la ingeniería.

## ULTIMAS RELACIONES A RESOLVER

Llegados a este punto, ya casi he resulto todas las relaciones y las he añadido al script, que no quiere decir que las que queden sean más fáciles o ligeras.

Así que vamos con ellas.

Había hablado del equipamiento que podía pertenecer a las sedes, y que luego este equipamiento era revisado por nuestros técnicos, así que vamos a resolver esa parte.

Diagrama

Descripción generada automáticamente

Lo primero es centrarse en la tabla equipamiento. La cual recibe como FK la PK de sedes.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Más adelante me di cuenta que cod\_sede tenía que ser INT y lo cambié.

En la foto anterior se puede ver la creación de la tabla equipamiento. Como siempre el código, salvo en algunos casos, es un CHAR(6) ya que creo que es mejor para poder tener más variedad de código a crear.

Un par de campos descriptivos que son VARCHAR(15) como tipo de equipamiento y modelo. Plazas es un SMALLINT pero que puede tener un valor nulo ya que quizá ese tipo de equipamiento no se refiera a un vehículo propio.

Añado también el cod\_sede como FK.

Diagrama

Descripción generada automáticamente

Antes de echar un vistazo a como va quedando el diagrama voy a resolver todo lo relacionado con el equipamiento.

En primera instancia debo de crear la tabla equipo, luego las dos tablas intermedias en cualquier orden.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Este es el fragmento de script de esas 3 tablas. Lo hacemos así y ahorramos algo de texto y de tiempo. El script se ejecuta bien, vamos a intentar ver el diagrama.

Diagrama

Descripción generada automáticamente

De nuevo, de la misma forma que en el ejercicio original, se empieza a complicar la visualización, pero espero poder mejorarlo en la parte final del ejercicio. Así es como quedan estas tablas relacionadas después de ejecutar el script.

Diagrama

Descripción generada automáticamente

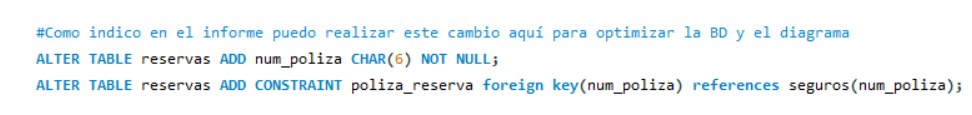
En el ejercicio también se hablaba de que se contrataban seguros para los grupos y los vehículos, así que vamos con esas partes, que ya son las últimas, y después veremos como se puede organizar el diagrama para que sea mínima mente legible y entendible.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En el script anterior se crean la tabla seguros y la tabla intermedia que la relaciona con grupos.

\*\*Incluyo esto aquí, ya que es una corrección. Después de revisar todo, me di cuenta de que podía ahorrarme esta tabla intermedia en la que los seguros se relacionan con los grupos, ya que es más eficiente si relaciono la tabla seguros con la tabla reservas, ya que esa tabla relaciona todo lo necesario con los grupos que contratan la reserva. Dicho esto, procedo a eliminar la tabla intermedia del script y a modificar la tabla reservas.



Eliminando el código de la tabla intermedia y añadiendo estas dos líneas puedo optimizar todo mucho más, o eso creo yo.

De todo eso me di cuenta al ver la parte de vehículos y su tabla alquileres.

Ahora vamos con la parte de vehículos.

Diagrama

Descripción generada automáticamente

La tabla empresas\_vehiculos ya esta creada, así que solo tengo que añadir la tabla vehículos y la intermedia, alquiler vehículos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Aquí están creadas ya la tabla vehículos y la de alquiler\_vehiculos. En estas partes me estoy saltando las muestras intermedias del diagrama para aligerar las cosas.

Vamos, ahora ya sí, con la parte de las agencias para terminar con todo ello.

Diagrama

Descripción generada automáticamente

Bueno, esto ya fue la última parte de mi ejercicio anterior, que quizá no hacía falta para entender y ejecutar el ejercicio, pero ya que lo presenté así hay que terminarlo.

En este caso tengo que crear primero la tabla agencias, después de eso, puedo crear ya las tablas intermedias.

Vamos a ello.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

La creación de la tabla agencias es bastante estándar. Luego las otras dos tablas intermedias también, bastante común a lo que ya he hecho en el ejercicio con anterioridad, PK compuesta de dos FK.

Ejecuto el script por última vez (supongo):

Texto

Descripción generada automáticamente

Todo Ok, salvo que no lo ejecuté por última vez. Al tener tantas tablas he creado nombres de FK repetidos, así que ha tocado modificar un par de cosas y probar hasta que todo queda como tiene que quedar.

# DIAGRAMA FINAL

Después de generar todo el código necesario vamos a ejecutar la ingeniería inversa y ver cómo queda todo.

Diagrama

Descripción generada automáticamente

En un principio el programa lo genera así. Voy a intentar ordenarlo un poco para que quede más legible.

Viendo el diagrama y moviendo las tablas me doy cuenta de que no añadí matrícula como FK a la tabla alquileres\_vehiculos, cosa que arreglo en este momento mientras voy ordenando el diagrama.

Diagrama

Descripción generada automáticamente

Este sería el diagrama, colocándolo un poco más, aún así, es un diagrama bastante grande, con muchas relaciones entre muchas tablas. A ratos se sigue haciendo difícil de leer.

Es bastante parecido al modelo que diseñé yo, este evidentemente más claro, y refleja lo que yo hice en el anterior ejercicio. Sigue siendo complicado de seguir en algunas partes ya que hay muchas tablas y para mí es difícil conseguir que no se corten demasiadas líneas, y eso que he intentado colocarlo lo mejor posible para que quede claro que líneas van de un sitio a otro.

Diagrama, Esquemático

Descripción generada automáticamente

Como se puede comprobar comparándolo con el mío, el mio es bastante más complicado de ver, aunque en estructura son bastante similares, ya que alguna cosa he ido cambiando al crear el código SQL.

# MODIFICACIONES

No he hecho demasiadas modificaciones en el ejercicio, añadir algún atributo nuevo y poco más.

El cambio más relevante ya lo expliqué con anterioridad al introducir el código SQL. Relacionando seguros con las reservas es más eficiente y práctico que en mi diseño inicial. Todo esto gracias a eliminar una tabla intermedia.

Creo que es la decisión correcta, en principio no me parece que genere problemas, al contrario, creo que los minimiza.

La otra opción que me da vueltas por la cabeza es sobre el tema de las agencias, que es algo de lo que al final me he terminado arrepintiendo. Se generan demasiadas tablas intermedias y no he encontrado la forma de mejorar eso.

Pensé en poner en la tabla reservas un atributo que se refiriese a la sede y otro a la agencia, y que los dos pudiesen tener valores nulos, dando a entender que en la tabla reserva se relaciona todo con la sede o la agencia que lo gestiona. Pero como ninguna de las opciones que pensé me terminaban de convencer, además de que no sabía como resolverlas de forma correcta, si es que la hay, he preferido dejarlo tal cual lo había planteado en un principio, para no terminar complicándolo todo más.

# MI DIAGRAMA RELACIONAL

Me imagino que lo lógico habría sido que esto fuese en primer lugar, pero tomé la decisión que, ya que tenía el diagrama hecho, ir generando la BD acorde a lo que ya tenía e ir modificándolo sobre la marcha.

Después de esto iría adaptando el diagrama al trabajo que he realizado, para que quedase todo acorde.

Así que ahora, lo que queda, es ir dejando el diagrama en condiciones, con los tipos de datos en las tablas y las cardinalidades bien puestas, que ya veré que tal me queda.

Lo primero que he hecho es asociar los seguros con las reservas, para que quede como en el ejercicio que he ido realizando. Con eso, elimino la tabla intermedia, y es algo menos a tener en cuenta.

Quizá podría darse el caso de que una reserva no lleve asociada una póliza de seguros, por lo que podría tener un valor de null. Ahora que lo pienso quizá el cambio puede tener sus puntos en contra, pero creo que voy a seguir adelante con ello.

## COMENZANDO CON CAMBIOS Y AJUSTES

Diagrama

Descripción generada automáticamente

Pare empezar a aclarar el diagrama que ya tenía, antes de añadir los tipos de atributos de cada tabla, lo que hago es colocar cardinalidades e intentar que cada cosa vaya resaltando por bloques.

En este caso, la parte referente a las empresas de vehículos, los vehículos mismo, el alquiler de los mismos y las pólizas de seguros que los cubren. Marco las cardinalidades, espero que estén de forma correcta.

Según he ido avanzando con las modificaciones y la ordenación del diagrama he ido cambiando también la forma en la que anotaba las cardinalidades.

Un ejemplo:

Diagrama

Descripción generada automáticamente

He añadido la cardinalidad a la línea de relación correspondiente, así es más difícil perderse y más fácil tener la cardinalidad bien relacionada con las líneas de relación de cada tabla.

He seguido este método para cada parte del diagrama, y aunque mucho más ordenado y visual, me ha sido imposible evitar algunos cruces de líneas. Ahora bien, aunque el diagrama es ahora más grande, no en tablas sino en el tamaño que ocupa, se ve mucho mejor, incluso en los cortes entre relaciones.

Voy a ir poniendo varias imágenes de las partes, y luego una en conjunto, para que se vea mejor.

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Con estos ejemplos vamos viendo como va quedando. Todo identificado por colores, cada bloque del diagrama tiene el mismo color para tablas principales e intermedias, la cardinalidad mostrada en cada línea de relación, etc.

Ahora vamos a mostrar el diagrama completo, que como ya dije, he sido incapaz de evitar que se solapen algunas líneas. Pero, para ser sinceros, al ponerlo junto al diagrama original es como la noche y el día.

## DIAGRAMA VIEJO

Diagrama, Esquemático

Descripción generada automáticamente

Aquí podemos apreciar, en comparación con el actual, el desorden y el caos.

## DIAGRAMA NUEVO

Escala de tiempo

Descripción generada automáticamente

Aunque siguen existiendo zonas que pueden generar algo de confusión, al estar dividida cada sección en colores, con las líneas de su mismo color y las cardinalidades marcadas dentro de cada línea relacional, se puede ver todo mucho más claro.

También he cambiado algo la disposición para intentar evitar que se crucen demasiado las líneas.

Pero, de todos modos, hay que tener en cuenta, que todavía tengo que modificar los nombres de los campos para que coincidan con aquellos dados en SQL, ya que SQL prefiere otro tipo de nomenclatura, y hay que añadir los tipos de datos, que eso añade algo más de complejidad al diagrama.

Así que vamos a ello.

# TERMINANDO DE MODIFICAR EL DIAGRAMA

## MODIFICANDO LOS ATRIBUTOS

Diagrama

Descripción generada automáticamente con confianza media

Comienzo por la parte de Provincia y Localidad. He cambiado los nombres de los atributos a minúsculas en los casos en los que en sql están de este modo, para que todo este de la misma forma.

En cuanto a los tipos, CP en provincia es CHAR(2) puesto que para identificar una provincia sólo necesito los dos primeros dígitos del código postal, en cp\_completo de la localidad es del mismo tipo pero de longitud fija 5 ya que necesito el código completo. En cuanto a localidad será un VARCHAR(20) para poder almacenar el nombre completo de la localidad.

Tabla

Descripción generada automáticamente

Tabla, Excel

Descripción generada automáticamente

Como aclaración comentar, que voy añadiendo los cambios en el mismo orden que he ido generando el script sql. Ya se que el orden parece aleatorio, pero en principio es porque empecé creando las tablas que no eran dependientes, luego fui añadiendo el resto.

En cuanto a estas dos tablas, decir de nuevo que fui cambiando los nombres para que fuesen iguales en la BD que en el diagrama. Los campos que en estas tablas son descriptivos o sólo texto, como dirección, localidad o departamento, comentar que les asigno a lo largo de todo el ejercicio el tipo VARCHAR, pues su longitud puede ser variable, y a aquellos que se repiten en más de una tabla intento darles siempre la misma longitud (creo que es así, quizá en alguno se me haya pasado y tenga una longitud diferente).

Lo mas reseñable en estas dos es:

cod\_dep: un SMALLINT con autoincrement, por si se crean más departamentos.

cif: CHAR(9) pues el cif siempre está formado por una letra y 8 dígitos.

telefono: en todas las tablas que aparece es CHAR(9) pues siempre tiene esa longitud.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Comentar también, que intento cambiar los nombres de las tablas para que coincidan con los de la BD.

En este caso, lo más reseñable:

cod\_paquete: CHAR(4) para definir el código de un paquete con números y letras me pareció más que suficiente.

Num\_max\_personas: SMALLINT. Podría haberlo definido como char también, pero consideré, que en caso de que se quisiese hacer un cálculo que involucre a las personas que pueden ejecutar una actividad SMALLINT me vendría muy bien.

Precio: Como me indicaste en clase cuando pregunte, DECIMAL(6,2). Creo que más que suficiente para alojar los precios. De aquí en adelante, el resto de atributos precio que se incluyen en el modelo son todos DECIMAL(6,2).

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamenteEn esta tabla sigo las mismas directrices que para las anteriores.

El único atributo que puede cambiar un poco es cod\_alojamiento que es un CHAR(6), para identificar a todos los alojamientos que se contratan creo que una combinación alfanumérica de 6 caracteres es lo más idóneo.

Tabla

Descripción generada automáticamenteEl resto sigue el estándar, son campos descriptivos que pueden tener una longitud variable. Teléfono va a ser del mismo tipo que en otras tablas.

Ahora era el tuno de grupos, y ya de paso hice el de la tabla componentes (que relaciona a grupos con clientes).

En grupos el id es int, para que pueda ser un campo auto\_incremente, y que cada vez que creemos un grupo nuevo se genere el número de forma secuencial y automática. num\_personas es int, por si se llegase a dar el caso de que quisiésemos hacer algún cálculo basado en las personas y precio, la media de personas por grupo, etc.

En la tabla componentes id\_grupo es int, pues la FK debe de ser del mismo tipo que el atributo al que hace referencia y dni es un CHAR(9) como en la tabla clientes, ya que el dni tiene una longitud fija de 9 caracteres, 8 dígitos y una letra.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

La tabla sedes, que es una de las centrales del ejercicio. Le cambio el nombre para que se ajuste bien al modelo. Todos sus atributos ya han aparecido en otras tablas, por lo que siguen la misma convención. Cp\_completo al ser una FK tiene el mismo tipo que en su tabla de origen, la cual ya expliqué anteriormente.

Cod\_sede por su parte es del tipo int, para que pueda ser auto\_incremente y cada vez que abramos una nueva sede, se incremente de forma automática.

Tabla, Excel

Descripción generada automáticamente

En esta tabla un par de aclaraciones. En el script se me olvidó inserta el campo f\_nacim, cosa que hice al completar esta tabla, dándole un tipo de dato DATE, que es el que más se ajusta. El resto de atributos reciben el mismo tipo de dato que ya les he dado en otras tablas y que no creo que tenga que volver a explicar, salvo dni, que aunque ya apareció en otra tabla anterior, recibe el tipo CHAR(9) por su longitud fija.

También tuve que corregir la relación con la tabla localidad, pues en el diagrama la había relacionado mediante el cp\_completo (al que ahora que me doy cuenta tengo que cambiar nombre) de forma incorrecta con el campo cp de la tabla localidad.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Arreglado.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamenteVarias cosas a comentar aquí:

Arreglo la falta de línea de relación con la tabla sedes, que no la había puesto, con su cardinalidad correspondiente.

Los tipos de datos de las FK son del mismo tipo que la tabla a la que referencian, SMALLINT e INT.

Num\_ss es un campo con una longitud fija, de ahí su tipo CHAR, la longitud no estoy seguro de que sea esa, pero creo que lo más importante es dar su tipo correcto.

Num\_matricula: tipo INT, para que pueda ser auto\_increment y generarse de forma automática al dar de alta a un nuevo empleado.

El resto de atributos, la gran mayoría, ya están explicados en tablas anteriores. Especialidad, que no había aparecido antes, es un campo descriptivo de la especialidad del trabajador, de ahí su tipo, VARCHAR.

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Esta tabla bastante sencillita es la de equipos. Sus dos atributos son SMALLINT. La PK para que pueda ser autoincremental, y el num\_tecnicos dado que va a almacenar la cantidad de técnicos participantes. También podría ser CHAR(2) por ser un atributo descriptivo más que para hacer operaciones con el.

Tabla, Excel

Descripción generada automáticamente

Esta es la última tabla totalmente independiente del modelo.

En esta todos los atributos son del tipo VARCHAR, pues son descriptivos y pueden tener una longitud variable.

Telefono sigue el mismo patrón que en casos anteriores, con su longitud fija de 9, y será un tipo CHAR.

Cod\_agencia en este caso, no la he tenido en cuenta como INT o SMALLINT para que sea auto\_increment, primero porque puede darse el caso de que haya más de una agencia en la misma provincia y necesite caracteres alfanuméricos para poder identificarlo así, y segundo, al no ser una entidad que controle nuestra empresa, no necesito específicamente que sea un dato correlativo y consecutivo.

Estas son las consideraciones que he tomado para esta tabla.

## COMPLETANDO EL DIAGRAMA

A partir de ahora, todas las tablas que me quedan por completar son tablas dependientes y que reciben una o varias FK. Por lo tanto como van a existir muchos atributos repetidos y que ya he explicado, iré colocando capturas de cada una de ellas con explicaciones más ligeras y escuetas.

Tabla

Descripción generada automáticamente

Recibe dos FK, que forman una PK compuesta. Sus tipos coinciden con los de los campos a los que hacen referencia.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

El cod\_reserva, un int para que pueda ser auto\_increment y secuencial.

Las fechas como tipo DATE.

Precio como en otro ejemplo, un Decimal (6,2).

Las FK como el tipo al que referencian.

Tabla

Descripción generada automáticamente

Matricula como CHAR(7) ya que creo que es la longitud completa de la matrícula.

Tipo un varchar, ya que es una descripción breve.

Plazas como SMALLINT, más por la posibilidad de que en algún momento se pueda hacer un cálculo, aunque bien podría ser un CHAR.

Cif, el mismo tipo declarado en la tabla origen.

Tabla

Descripción generada automáticamente

Cod\_alquiler como CHAR(6), ya he explicado en otras partes de este ejercicio porque en casos parecido a este elijo un char.

Las fechas de tipo DATE. Nunca DATETIME, ya que no creo que las horas sean relevantes.

Las FK de su tipo original.

Tabla

Descripción generada automáticamente con confianza media

Calendario

Descripción generada automáticamenteTipo de los atributos según atributos a los que referencia.

Mismo tipo de tabla que el caso anterior, así que los tipos de datos serán aquellos a los que referencian los atributos de la tabla.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Nada muy distinto a lo visto anteriormente.

Num\_poliza como CHAR(6), las fechas tipo DATE, los campos que son descriptivos son VARCHAR, de longitudes estimadas, no se muy bien lo que podrían requerir ese tipo de datos a ciencia cierta, y el precio, como en otros casos DECIMAL(6,2).

Imagen que contiene interior, verde, competencia de atletismo, público

Descripción generada automáticamenteOtra tabla intermedia cuyos atributos son PK de otras tablas.

Tabla

Descripción generada automáticamente

La PK, como en otros casos que creo una PK que no sea auto\_increment, quizá necesite algún carácter que no sea numérico para identificarlo mejor.

Los descriptivos, como VARCHAR.

Plazas, como en el caso de los vehículos de alquiler, un SMALLINT, salvo que este puede recibir un valor NULL.

La FK del mismo tipo origen.

Interfaz de usuario gráfica, Tabla, Excel

Descripción generada automáticamenteEn este caso:

Sigo el mismo patrón de tipo de dato para la PK.

La fecha es de tipo DATE.

Favorable, que en inicio iba a ser un campo BOOLEAN, el programa me lo creo como TINYINT, no se si será porque en el otro campo BOOLEAN del ejercicio le puse en el script un valor por defecto.

Tabla

Descripción generada automáticamenteLas FK del tipo original.

Tabla intermedia que sólo recibe FK como atributos, por lo tanto sus tipos ya están definidos.

Tabla

Descripción generada automáticamente

Ya por último. Una PK como las anteriores que he creado. Las fechas de tipo DATE y las FK del tipo original.

# Imagen que contiene Escala de tiempo Descripción generada automáticamenteDIAGRAMA FINAL

Espero que la calidad sea la suficiente. Así es como queda el diagrama al final de todo. Creo que es bastante más legible que el anterior, a pesar de llevar incluidas más cosas. Quizá tal y como está la imagen alguna cosa queda un poco borrosa, pero en principio creo que bien. Lo añadiré ahora en dos partes para que se vea mejor.

Tabla

Descripción generada automáticamente

Imagen que contiene Tabla

Descripción generada automáticamenteEspero que así en dos partes se vea mejor.