EMPRESA DE TAXIS (JOINS)

Trabajo de Alejandro Sainz Sainz

BD-ACTIVIDAD 3.2

INICIANDO EL SCRIPT Y CREANDO LA BD 3

PRIMEROS PASOS 3

INSERTANDO DATOS 6

COMENZAMOS CON LAS CONSULTAS 8

MOSTRAR COCHES CON CONDUCTOS ASIGNADO 8

MOSTRAR ORIGEN/DESTINO/MATRICULA/ADAPTADO 9

LISTAR CONDUCTORES Y COCHES RESPECTIVOS 10

LISTAR TODOS LOS DATOS 11

TODOS LOS SERVICIOS Y TAXIS 12

CONCLUSIÓN 13

[1 Primera parte del script SQL 3](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949615)

[2 Creación de la primera tabla: Taxis 3](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949616)

[3 Creación de la tabla Choferes/Conductores 4](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949617)

[4 Creación de la última tabla: Carreras/Viajes 5](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949618)

[5 Ingeniería Inversa de la BD 6](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949619)

[6 Inserts de la tabla Taxis 7](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949620)

[7 Inserts de la tabla choferes 7](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949621)

[8 Inserts de la tabla carreras 8](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949622)

[9 Primera Consulta 8](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949623)

[10 Segunda Consulta 9](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949624)

[11 Tercera Consulta 10](file:///C:\Users\daw1.COMPU-S026\Documents\Daw1\Bases%20de%20Datos\Actividad%203.3\Empezando%20con%20consultas.docx#_Toc188949625)

[12 Cuarta consulta UNION 11](#_Toc188949626)

[13 Quinta consulta UNION 2 12](#_Toc188949627)

# INICIANDO EL SCRIPT Y CREANDO LA BD

Vamos a comenzar primero con el script para luego ejecutar una ingeniería inversa y ver así el modelo relacional. Al lío.

# PRIMEROS PASOS

Texto

Descripción generada automáticamenteVamos con la primera parte del script, que es la creación de la base de datos en si misma y de las tablas que la componen.

1 Primera parte del script SQL

Esta parte no tiene mucho que explicar, es la creación de la base de datos, a la que he llamado compañía\_taxis. Indico en el script que la borre en caso de que exista y después de crearla, que la use.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

2 Creación de la primera tabla: Taxis

Creo la tabla taxis, que es la única que no recibe claves ajenas.

Matrícula como primary key y como campo único, un char(7).

Marca y modelo, campos descriptivos, como varchar de tamaño 15, creo que serán más que suficientes.

Num\_pasajeros, un char(1), es más que suficiente para indicar las plazas del taxi, dudo que en ningún momento pase de un dígito.

Adaptado. Lo vi en el ejercicio anterior para el sexo de los empleados y me gustó. Uso un enum, s para sí, n para no.

Luego indico cual es la primary key.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteDespués creo la tabla de los conductores a la que he llamado choferes.

3 Creación de la tabla Choferes/Conductores

Bastante estándar, muy parecido a lo usado en ejercicios anteriores que incluían tablas de empleados o personas.

Lo único, incluir un campo vehículo, que es el que se refiere a la matrícula del taxi que tiene asignado este conductor.

Indico después la PK y la FK.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteAquí la tabla carreras, no me convence mucho el nombre, pero como en el enunciado lo indica así, lo seguimos.

4 Creación de la última tabla: Carreras/Viajes

Incluyo un campo, cod\_carrera, para que me sirva como identificador único. Como tipo le doy un char(6), más que de sobra para códigos de identificación alfanuméricos.

Origen y destino se refieren a direcciones, espero que con una longitud de 15 del tipo varchar sea más que suficiente.

Precio, quizá me he ido un poco largo, pero nunca se sabe, un decimal (6,2). Quizá podría optar por un char o varchar, ya que podría ser simple texto, pero así me parecía mejor.

Conductor y taxi. Las FK que recibe esta tabla. Conductor refiere a chofer y taxi a taxis.

Voy a ejecutar ahora el script, y a realizar la ingeniería inversa.

Diagrama

Descripción generada automáticamente

5 Ingeniería Inversa de la BD

La BD no es muy complicada, así que el diagrama tampoco lo es. Aquí vemos las tres tablas y sus relaciones.

## INSERTANDO DATOS

Lo primero que tengo que hacer es insertar datos para cada una de las tablas, pero tengo que tomar como referencia aquellos que se proporcionan como ejemplo, por coherencia, para luego poder insertar datos en la tabla carreras y que obtengamos el mismo resultado que el proporcionado por el ejemplo.

Antes de nada, como en el ejemplo hay un espacio en blanco en las matrículas, voy a corregirlo y a ese campo le voy a dar un tipo de char(8).

Vamos a ello.

Texto

Descripción generada automáticamenteInserto primero registros en la tabla taxis, primero los valores de matrículas que aparecen en la tabla de ejemplo, luego creo algunos más, por si me hiciesen falta para los ejercicios.

6 Inserts de la tabla Taxis

Vamos ahora con la tabla choferes.

Me aseguro que, siguiendo la tabla del ejercicio, a los dnis que aparecen les asigno la matrícula que les corresponde.

Texto

Descripción generada automáticamente con confianza media

7 Inserts de la tabla choferes

Hecho. Me aseguro de que todos los valores concuerden a como se pide. He incluido unos cuantos más por si acaso.

Ahora toca la tabla carreras.

Vamos a asegurarnos también de que todo concuerda con el enunciado.

Según voy avanzando me doy cuenta de fallos, se me olvidó añadir el atributo para saber si una carrera era nocturna. Voy a añadir un atributo del tipo enum, valores sÍ o no, como en el caso anterior. Luego, al final del ejercicio mostraré capturas con las correcciones.

Imagen que contiene Diagrama

Descripción generada automáticamenteAquí esta lo que he ido añadiendo a la tabla, buscando que concuerde con el ejemplo.

8 Inserts de la tabla carreras

# COMENZAMOS CON LAS CONSULTAS

## MOSTRAR COCHES CON CONDUCTOS ASIGNADO

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamentePara este apartado si voy a tener que usar un join, ya que debo de mostrar aquello coches cuya matrícula aparezca también en la tabla choferes.

9 Primera Consulta

Aquí el comando usado y el resultado. Indicar que como el atributo matrícula tiene el mismo nombre en las dos tablas, tuve que indicar taxis.matricula después del select.

## MOSTRAR ORIGEN/DESTINO/MATRICULA/ADAPTADO

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteOtra consulta JOIN que vamos a utilizar para este ejercicio, ya que necesito datos de las dos tablas.

10 Segunda Consulta

Como vemos en la imagen, con ese comando obtenemos el resultado que se muestra debajo.

## LISTAR CONDUCTORES Y COCHES RESPECTIVOS

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteVamos a ello.

11 Tercera Consulta

Aquí el resultado. El order by del final ha sido por probar comandos, y me he quedado con este resultado ordenado por nombre.

Aquí el comando ya empieza a ser un poco largo, así que voy a copiarlo aquí para que se vea un poco mejor.

**SELECT nombre, apellidos,dni, marca, modelo, taxis.matricula, num\_pasajeros FROM choferes INNER JOIN taxis WHERE choferes.matricula = taxis.matricula ORDER BY nombre;**

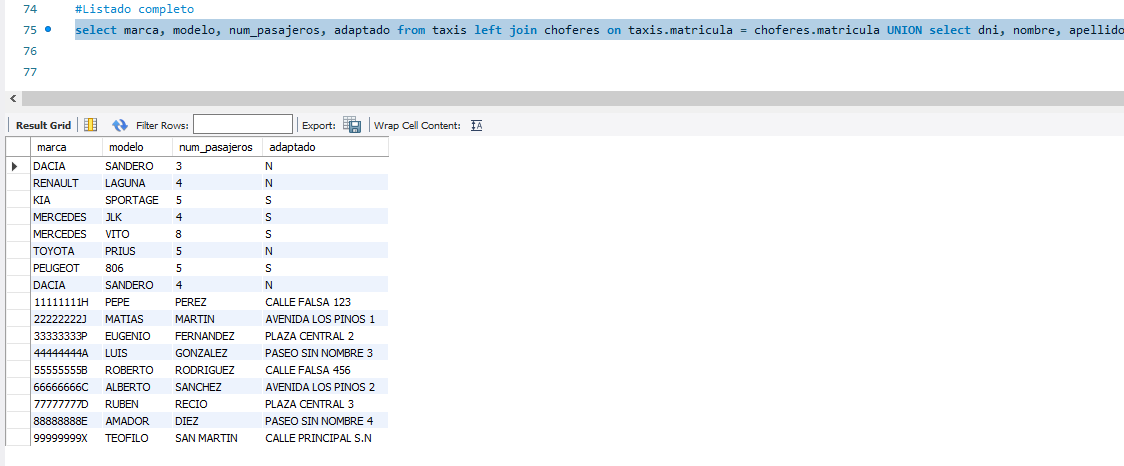
## LISTAR TODOS LOS DATOS

Para este ejercicio supongo que debo de usar un full outer join. Para así poder listar todos los resultados.

Bien, después de mirar los apuntes y en internet, veo que MySql no acepta la full outter join. Después de hablar con el profesor, y de mirar documentación y w3schools, ya me explicó que debo de hacer dos joins, una right y una left, indicando los atributos que quiero mostrar, concatenándolas mediante la sentencia UNION. Como el comando es un poco largo, primero le voy a escribir aquí, antes de insertar la captura.

**SELECT marca, modelo, num\_pasajeros, adaptado FROM taxis LEFT JOIN choferes ON taxis.matricula = choferes.matricula UNION SELECT dni, nombre, apellidos, direccion FROM taxis RIGHT JOIN choferes ON choferes.matricula = taxis.matricula;**

Este sería el comando, ahora inserto la captura para poder ver el resultado.



12 Cuarta consulta UNION

Lo único que tengo que tener en cuenta, es que en cuanto cambio una columna a mostrar el resultado puede variar mucho. Por eso tengo que elegir yo las columnas, para que no se duplique el atributo matrícula no puedo usar el carácter comodín \*.

También en la documentación de MySql encontré la opción de CROSS JOIN, pero eso cruza cada elemento de una tabla con cada elemento de otra, y eso no produce los resultados deseados.

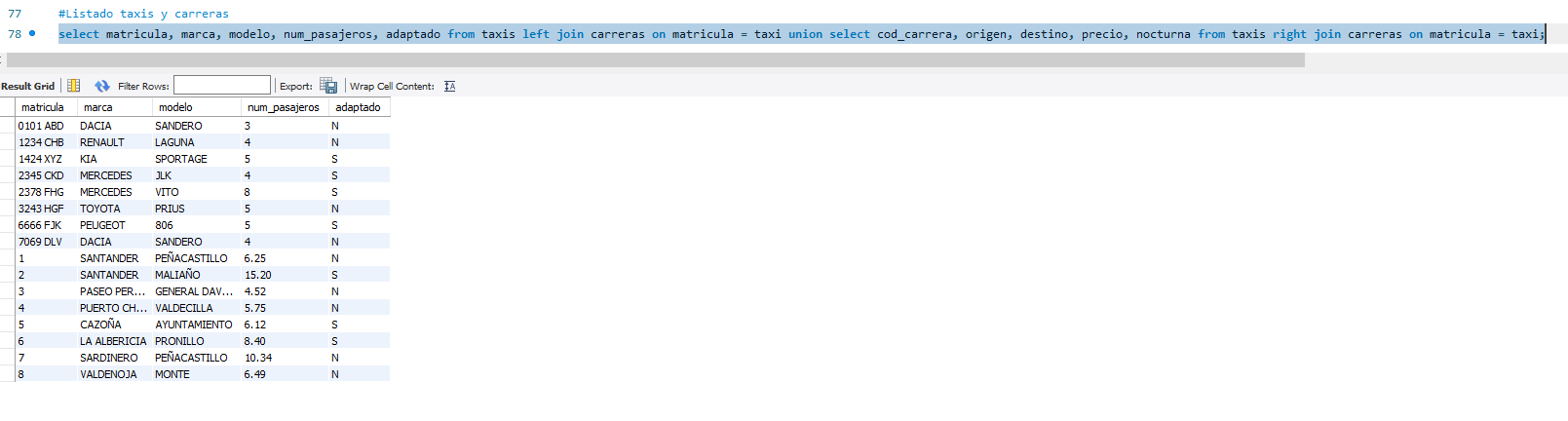
## TODOS LOS SERVICIOS Y TAXIS

En este caso, creo que también tendré que usar un UNION de dos JOIN.

**SELECT matricula, marca, modelo, num\_pasajeros, adaptado FROM taxis LEFT JOIN carreras ON matricula = taxi UNION SELECT cod\_carrera, origen, destino, precio, nocturna FROM taxis RIGHT JOIN carreras ON matricula = taxi;**

De nuevo este es un comando muy largo, así que primero lo pongo aquí para que se pueda ver bien, ya que en la captura puede que no se vea completo o que se vea muy pequeño.

Antes de nada, para comentar, pensé que si en el primer join, que era para taxis podría usar el carácter comodín \* y luego indicar las 5 columnas que quería de carreras ya que la tabla taxis es más pequeña, pero eso tampoco está permitido, así que tuve que indicar las 5 columnas de cada tabla manualmente.



13 Quinta consulta UNION 2

Como ya dije, el comando se ve muy pequeño, pero el resultado se ve de forma correcta.

# CONCLUSIÓN

Como ya se habló por teams, quizá tendría que insertar sólo los registros indicado en vez de añadir registros adicionales, aunque la tabla carreras la copié tal cual era en el ejemplo y no le añadí ningún registro adicional. No es que me haya generado ningún problema per se, pero a veces en la visualización el resultado es más amplio que el esperado.

Por lo demás, todo bien, pero al no encontrar en los apuntes claramente el comando UNION eso me ha traído algún que otro problema. De todas formas nada que no se pudiese solucionar con alguna que otra pregunta e investigando por internet en la documentación oficial.