

Practica 8

Gestión de Autoescuelas

8.1. ENUNCIADO DEL PROBLEMA

Se desea analizar, para el posterior desarrollo de un sistema de información automatizado, la gestión de una autoescuela. En una empresa de este tipo se ofrecen una serie de servicios de enseñanza de conducción de vehículos a motor a sus clientes para prepararlos para los exámenes correspondientes que realiza la Dirección General de Tráfico; exámenes que, una vez superados, les permiten (a los clientes) obtener la licencia de conducción correspondiente capacitándolos para la conducción de vehículos de categoría igual (y a veces menor) al examen que han superado

La información que se maneja en una autoescuela es muy variada. Información correspondiente a los datos de los clientes, información correspondiente al personal de la autoescuela, información correspondiente a los medios físicos que utiliza la autoescuela para la impartición de sus clases, información correspondiente a las exigencias marcadas por las disposiciones legales, etc. En este ejercicio propuesto sólo se va considerar parte del sistema de información (por razones de simplicidad del mismo); considerándose el problema del personal sólo de forma colateral en lo que afecta a sus relaciones con los clientes y los medios físicos de la autoescuela.

Consideraremos, además, los siguientes supuestos semánticos para definir más concretamente el problema:

Supuesto 1: La autoescuela cuenta con una serie de profesores que están capacitados para impartir clases para la obtención de diferentes tipos de licencias.

Supuesto 2: No todos los profesores tienen categoría suficiente para impartir cualquier tipo de clase.

Supuesto 3: Con objeto de simplificar el problema propuesto se va a suponer que cada profesor tiene asignado un único vehículo para cada tipo de clase que imparte y viceversa, un vehículo está asignado a un único profesor.

Supuesto 4: Los tipos de clases hacen referencia a los tipos o grupos de licencias que los alumnos desean obtener y para cuyo examen se prepara a los mismos.

Supuesto 5: Además, hay que considerar que la autoescuela imparte clases teóricas y prácticas. Para la obtención de cualquier licencia el cliente (alumno) debe superar inicialmente una prueba teórica para poder presentarse a la prueba correspondiente.

Supuesto 6: Cuando el alumno cuenta con alguna licencia de conducción para la cual ha tenido que superar la prueba teórica, no necesitará presentarse de nuevo a esta prueba para la obtención de cualquier otra licencia.

Supuesto 7: Las clases teóricas son las mismas para todos los alumnos con independencia del tipo de licencia para la obtención de la cual se han matriculado en la autoescuela. Son, por tanto, las clases prácticas las que diferencian los tipos de licencias a obtener por los alumnos.

Supuesto 8: Un cliente puede matricularse en la autoescuela, al mismo o distinto tiempo, para su preparación en la obtención de más de un tipo de licencia.

Supuesto 9: Cada alumno de la autoescuela tiene que abonar unos costes de enseñanza que están divididos de

la forma siguiente: costes de matriculación, costes de tramitación de documentos, costes de enseñanza teórica, costes de enseñanza práctica, y otros costes. Los costes de matriculación y enseñanza teórica son únicos, pero los otros dependerán del número y tipo de enseñanza práctica y del número de veces que haya que tramitar su documentación en la DGT (Dirección General de Tráfico); es decir, el número de veces que se presente a examen.

Si el alumno se matricula para la preparación en la obtención de varias licencias de conducción sólo abonará una vez el coste por su enseñanza teórica, pero los costes de matriculación se abonarán para cada tipo de licencia en la que desea ser preparado.

Supuesto 10: Los costes de enseñanza práctica dependen del tipo de licencia para la cual se prepara y del tipo de vehículo que utiliza para esta preparación. Así, la flota de vehículos con que cuenta la autoescuela está clasificada en categorías según el tipo de licencia para el cual se pueden utilizar y, dentro de ésta, en la categoría, lujo o tamaño del vehículo.

Los costes de enseñanza práctica son abonados en base al número de horas que realiza el alumno. Para cada vehículo, o tipo de éstos, existe una tarifa base marcada por horas.

Supuesto 11: Cada alumno tiene asignado un único profesor para su enseñanza práctica y, a su vez, un empleado de la autoescuela puede ser profesor de varios alumnos.

Si el alumno se encuentra matriculado para su preparación en varias licencias su profesor puede ser el mismo o distinto para cada una de ellas.

Supuesto 12: Los costes que paga el alumno por la tramitación de su documentación en la DGT le dan derecho a la presentación en tres exámenes (teóricos y/o prácticos). Si no se obtiene la licencia en estas convocatorias estos costes se deben abonar de nuevo.

Supuesto 12+1: Cada vez que el alumno es examinado por la DGT en un examen práctico deberá abonar a la autoescuela el coste del mismo como si de una clase práctica más se tratara.

Supuesto 14: Dependiendo del tipo de licencia a alcanzar, el cliente debe presentar en la autoescuela una serie de documentos (fotocopia del dni, fotocopia de otras licencias que posea, en su caso, certificado médico, etc.). El seguimiento de esta documentación y el estado de la misma es de interés también para la autoescuela.

Supuesto 15: El alumno puede ir entregando a la autoescuela cantidades a cuenta del coste total que supone o supondrá (no se conoce a priori) su formación. El control económico del estado del pago de los alumnos es también de interés para la autoescuela.

Supuesto 16: A la autoescuela le interesa también mantener información correspondiente al consumo de sus vehículos y el número de kilómetros que son realizados por los mismos en cada clase práctica.

Supuesto 17: Interesa mantener también un seguimiento personal de las clases prácticas realizadas por los alumnos en las cuales el profesor correspondiente realizará un informe sobre el resultado de las mismas.

Supuesto 18: De vez en cuando, la autoescuela realiza pruebas de control de los conocimientos teóricos de los alumnos. El resultado final de estas pruebas también es de interés para la autoescuela.

8.2 MODELADO CONCEPTUAL : Analisis de Entidades

Dentro del sistema conformado por una autoescuela podemos apreciar unos subsistemas diferentes. Abordaremos el análisis de entidades en función del subsistema en el que se encuentran.

8.2.1 Gestión Económica

Dentro de este subsistema recogemos todo lo concerniente a los ingresos y gastos que se generan en la autoescuela cuando presta sus servicios a los alumnos o clientes. En esta línea y refiriéndonos a los supuestos del problema, tenemos que los únicos ingresos de capital a la autoescuela provienen de los pagos de los alumnos en relación a los servicios que usan de la misma. A su vez, cuando un alumno usa un servicio de la autoescuela, éste lleva asociado un coste que también debe asociarse al alumno. De esta manera realizamos un seguimiento de gastos y pagos del alumno hacia la autoescuela. Las entidades consideradas son las siguientes:

8.2.1.1 Pago

En esta entidad queremos reflejar cada pago que realiza el alumno a la autoescuela. Este pago puede ser por adelantado o una vez que el alumno ha usado algunos de los servicios de la autoescuela. Señalar que cada pago realizado a la autoescuela no tiene relación directa con el uso de un servicio concreto, sino que es una cantidad que el alumno entrega a la autoescuela hasta completar el coste total de su formación. Claramente se aprecia que se trata de una entidad débil por identificación de la entidad **matrícula**, ya que para identificar un pago necesitamos saber a qué matrícula se refiere el pago (y como consecuencia, a qué alumno). De esta manera podemos saber qué pagos de un alumno están referidos a qué matrículas, en el caso de que un alumno se matricule de más de una licencia (realice varias matrículas). Además, este tipo de entidad necesita un atributo externo que identifique unívocamente a cada ocurrencia, ya que en una matrícula se pueden realizar varios pagos a la autoescuela en vez de uno sólo por el total. Sus atributos son:

- **Nº_pago**: Atributo externo que asigna un número único a cada ocurrencia de la entidad
- **Importe**: Indica el importe del pago que realiza el alumno a la autoescuela.
- **Nº_matrícula**: Atributo heredado de la entidad con la que mantiene una relación de debilidad por identificación

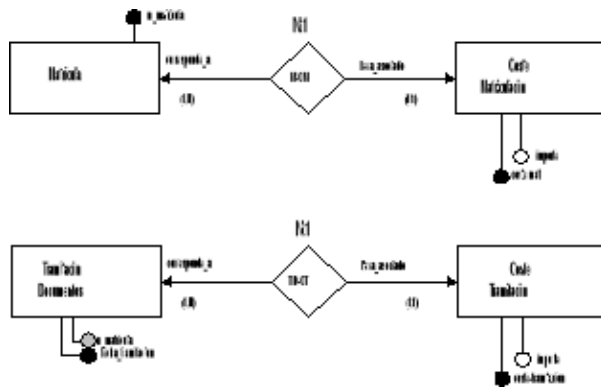
Señalar que la clave primaria de este tipo de entidad es el agregado de datos **Nº_pago+nº_matrícula**.

8.2.1.2 El coste de matriculación, de enseñanza teórica y de tramitación de documentos

Estudiamos estas tres entidades en el mismo apartado porque todas tienen algo en común: su valor (el coste que representan) es el mismo para todos los alumnos de la autoescuela. Es decir, cada matrícula realizada en la autoescuela lleva asociado un coste igual para todos los alumnos matriculados. Otro tanto ocurre con la tramitación de documentos en la DGT. La aplicabilidad del coste de enseñanza teórica a todos los alumnos dependerá de si tiene una licencia anterior o no.

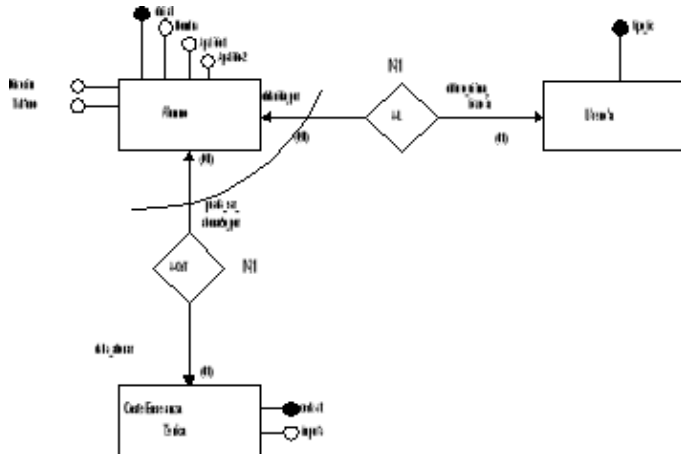
Para aplicar estos costes disponemos de dos opciones: La primera sería asociar a cada matrícula un atributo denominado **coste** que indicará el coste de matriculación, y seguir un tratamiento igual en las demás entidades. Esta consideración tiene el problema de que hay que introducir manualmente el valor de este coste siendo igual para todos. Además si alguna vez cambia el valor de este coste, tendríamos que cambiar todas las ocurrencias de las entidades afectadas por el cambio.

La segunda consideración es crear una entidad por cada coste fijo, y relacionarla con la entidad que genera el coste. Es decir, para la entidad **matrícula**, que como sabemos lleva asociado un coste de matriculación, creamos otra entidad **coste matriculación** de la que sólo existirá una ocurrencia que almacenaría este coste. De esta manera, cada vez que se cree una matrícula, se asociará esta con este coste. Igual tratamiento seguimos con la entidad **tramitación documentos**, como se puede apreciar a continuación:



De esta manera podemos calcular estos costes a la hora de cobrar al alumno.

El control del coste por una enseñanza teórica está más relacionado con el alumno que con la matrícula que realiza. Tenemos según los supuestos de problema, que si un alumno posee una licencia anterior no debe abonar estos gastos. Además tenemos que si el alumno se matricula de más de una licencia, es decir realiza más de una matrícula, el coste por su enseñanza teórica se abonará una sólo vez. Si representamos la última licencia que posee el alumno (en su caso) como una relación de la entidad **alumno** a **licencia**, y representamos una entidad **coste enseñanza teórica**, esta situación se podría representar como sigue:



La exclusividad de la relación A-CET y A-L nos indica que si el alumno posee una licencia anterior, no existirá la relación a **coste enseñanza teórica**. Y si no existe la relación a **licencia**, un alumno como mucho tendrá que abonar una sólo vez los portes de su enseñanza teórica.

8.2.2 Gestión del alumno

Dentro de este subsistema nos referimos a la gestión de la información del alumno en la autoescuela, como son los exámenes a los que se ha presentado, las clases prácticas realizadas, las pruebas teóricas, ect.

8.2.2.1 Alumno

Con esta entidad pretendemos reflejar al alumno que tenemos en la autoescuela. Veamos sus atributos

- **Nombre:** Cadena de caracteres que representa el nombre del alumno
- **Apellido-1:** Cadena de caracteres que representa el primer apellido
- **Apellido-2:** Cadena de caracteres que representa el segundo apellido
- **Dni_al:** Numero del documento nacional de identidad.
- **Dirección:** Dirección completa del alumno

- **Teléfono:** Teléfono del alumno

La clave primaria de esta entidad está formada por el atributo dni_al.

8.2.2.2 La Matrícula

Básicamente, cuando un alumno llega a una autoescuela se le realiza una matrícula en la que se recoge la licencia de la que desea prepararse. Este documento sirve como representación del alumno dentro de la empresa, de tal manera que para referirse a un alumno, la empresa se refiere a su matrícula. Desde este punto de vista, también es lógico pensar que un alumno que se matricula de más de una licencia consume recursos de la autoescuela como si fuera varias personas diferentes que necesitaran preparación en licencias diferentes. Es decir, realiza matrículas diferentes.

Sus atributos son:

- **Nº_matrícula:** Identifica el número de matrícula del alumno. Conformar el identificador de la entidad.
- **doc_presente:** Es un atributo booleano que indica si la documentación necesaria para la obtención de derechos a examen está presente en los archivos de la autoescuela.

8.2.2.3 Las Pruebas Teóricas

Básicamente la entidad **prueba teórica** identifica un test que un alumno realiza en la autoescuela. Veamos sus atributos:

- **Nº test:** identifica el número del test realizado.
- **Errores:** identifica el número de errores cometidos
- **Fecha_prueba:** identifica la fecha en la que se realizó el test.

Notar que esta entidad es débil por identificación de la entidad **matrícula**, puesto que para identificar una prueba concreta, necesitamos saber qué alumno (que matrícula) la realizó. De esta manera, la clave primaria de esta entidad es el agregado de datos **fecha_practica+nº_matrícula**.

8.2.2.4 Las Clases prácticas

La entidad **clase práctica** representa cada clase práctica que recibe un alumno en una licencia. En cada clase práctica interesa recordar un informe pormenorizado de lo sucedido en la misma (Supuesto 17), que recogemos en los atributos

- **Objetivos:** cadena de caracteres en la que el profesor indica los objetivos a cubrir en la clase práctica
- **Resultados:** cadena de caracteres en la que el profesor indica los resultados obtenidos en relación a los objetivos
- **Fecha_practica:** indica la fecha de realización de la práctica
- **Duración:** medida en horas de la duración de la clase

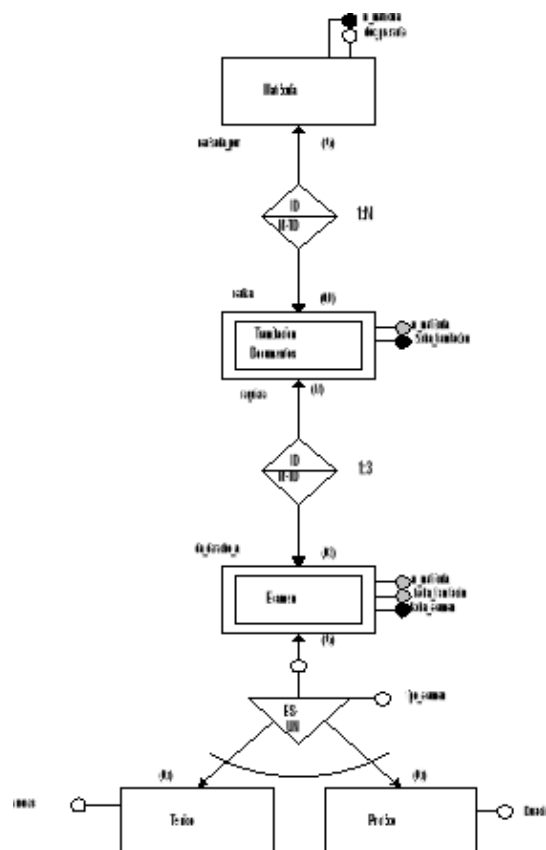
Como era de esperar, esta entidad es débil por identificación de la entidad **matrícula**, ya que todas las clases prácticas son aparentemente idénticas para todos los alumnos, y sólo podremos identificar una clase práctica si sabemos que alumno la realizó. De esta manera, la clave primaria de esta entidad es el agregado de datos **fecha_practica+nº_matrícula**, este último heredado de la relación de debilidad.

8.2.2.5 Los Exámenes y la Tramitación de Documentos

Los exámenes que realizan los alumnos en la autoescuela tienen que venir precedidos de una tramitación de su

documentación en la DGT. La entidad **exámen** representa, como su nombre indica, *'la realización de una prueba en la DGT'*. Por otra parte, la entidad **tramitación documentación** representa el hecho de que un alumno *'tramita su documentación con la intención de obtener derechos a exámen'*. Esta última entidad es débil por identificación de la entidad **matrícula**, ya que para identificar una tramitación de documentos, necesitamos saber a que matrícula está referida.

La entidad **exámen** está especializada en dos entidades diferentes: **exámen teórico** y **exámen práctico**. Esta consideración nos permite cumplir un supuesto implícito en el problema que es *'por cada tramitación de documentos se obtienen 3 convocatorias a exámen a repartir entre exámenes teóricos y exámenes prácticos'*. Como podemos ver en el diagrama inferior, para que un alumno pueda presentarse a un **exámen**, debe existir una ocurrencia de la entidad **tramitación documentos**, ya que la primera entidad es débil por identificación de la segunda. Esta debilidad podría ser por existencia (que quizás sea más lógico), pero si tomamos esta opción no sabríamos a que alumno (a que matrícula) pertenece cada examen (hay que considerar que todos los exámenes de todos los alumnos se almacenarán en la entidad **exámen**). Esta identificación también es posible ya que como se comentó anteriormente, la entidad **tramitación documentos** es débil por identificación de la entidad **matrícula**.



Veamos los atributos de la entidad **exámen**:

- **Fecha_exámen**: identifica la fecha de realización del exámen.

Notar que el identificador de esta entidad es el agregado de datos **nº_matrícula+fecha_tramitación+fecha_exámen**, estos dos últimos heredados de la debilidad por identificación.

En relación a la entidad **teórico**:

- **Errores**: entero que indica el número de errores cometidos en el teórico

En relación a la entidad **práctico**:

- **Duración**: indica la duración del examen para cobrarlo al alumno.

Los atributos de la entidad **tramitación documentación**:

- **Fecha_tramitación**: identifica la fecha de la tramitación

La clave primaria de esta entidad es el agregado de datos **nº_matrícula+fecha_tramitación**, el primero heredado de la entidad **matrícula**.

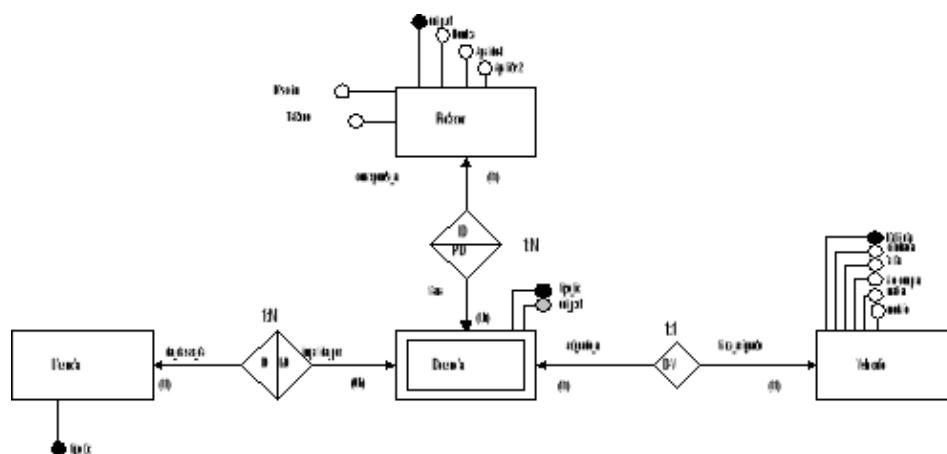
8.2.3 Gestión de Recursos de la Autoescuela

En este subsistema recogemos la gestión de la información referente a los profesores, los vehículos que se usan, los grupos de clases que tiene cada profesor, ect.

8.2.3.1 Profesor y Docencia

Estudiamos estas dos entidades conjuntamente, porque una es una representación de un profesor en la vida real (como una persona), y la otra es una representación de un profesor desde el punto de vista de la empresa (como un recurso de la misma).

La primera entidad es obvia: representa a una persona que ejerce de profesor en la autoescuela, proporcionando clases a los alumnos de la misma. Ahora se nos plantea un problema cuando consideramos que un profesor da clases de más de una licencia en la autoescuela. El problema viene al considerar a qué alumnos imparte clases el profesor, ya que éste puede impartir dos o más tipos de clases. Además tenemos que tener en cuenta el supuesto 11 que indica que **'si un alumno se matricula de más de una licencia, su profesor puede ser distinto o el mismo'**. De estas necesidades, surge la consideración de un profesor como un recurso que usa la autoescuela para la preparación de sus alumnos, que viene recogido en la entidad **Docencia**. Esta entidad representa a **un recurso presente en la autoescuela que imparte clases de una licencia, usando un vehículo para las mismas**. Desde este punto de vista, tenemos que un profesor puede tener asociadas una Docencia o más, en función de si da clases de una o más licencias. Y recíprocamente, una Docencia sólo puede estar asociada a un profesor (Un docente siempre es una persona). Esta representación nos permite el caso de que un alumno se matricule de más de una licencia, y su profesor pueda ser el mismo o distinto en función de si las dos entidades Docencia asociadas a las matrículas (las dos pertenecientes al mismo alumno) están asociadas a su vez al mismo profesor, o a distintos profesores. Veamos el diagrama:



Como podemos apreciar, una ocurrencia de **Docencia** está asociada a una y sólo una ocurrencia del tipo **Licencia**, indicando de esta manera la licencia de la que imparte clase una docencia concreta (asociada a un

profesor concreto).

Como era de esperar, la entidad **Docencia** es débil por identificación de la entidad **Profesor**: es evidente que si no existe un profesor, no podrá ejercer la capacidad de enseñar. También es evidente que no puede existir una entidad **Docencia** si no existe una entidad **Licencia** con la que este asociada. Desde este punto de vista, una Docencia necesita para identificarse y diferenciarse del resto el profesor que la imparte o representa. Si estamos considerando el caso de que un profesor puede tener asociadas una o mas docencias, y teniendo en cuenta que no pueden existir dos docencias asociadas al mismo profesor y asociadas a la misma licencia (seria como decir que un profesor da clase del B y del B) , surge la consideración de que la entidad Docencia es débil por identificación tanto de la entidad Profesor como de la entidad Licencia. La dependencia por identificación de la entidad Licencia, nos permite diferenciar entre las distintas docencias asociadas a un mismo profesor. Veamos los atributos:

En la entidad **profesor**, tenemos:

- **Nombre:** Cadena de caracteres que representa el nombre del profesor
- **Apellido-1:** Cadena de caracteres que representa el primer apellido
- **Apellido-2:** Cadena de caracteres que representa el segundo apellido.
- **Dni_prof:** Numero del documento nacional de identidad.
- **Dirección:** Dirección completa del profesor
- **Teléfono:** Teléfono del profesor

La clave primaria de esta entidad está formada por el atributo **DNI**.

Para la entidad **Docencia** no se han considerado atributos diferentes a los heredados como consecuencia de la debilidad que tiene respecto a **Profesor** y **Licencia**, aunque si que se ha justificado su existencia. Además, en futuras ampliaciones del sistema, por ejemplo, podría surgir la necesidad de controlar el sueldo que recibe un profesor concreto, en función de los tipos de clases que imparte. La consideración de la entidad **Docencia** es clave para este control.

En cualquier caso, por ahora el identificador de la entidad **Docencia** es el agregado de datos **tipo_lic+dni_prof**, heredados de **Licencia** y **Profesor** respectivamente.

8.2.3.2 Los Vehículos , el Consumo y el Lujo

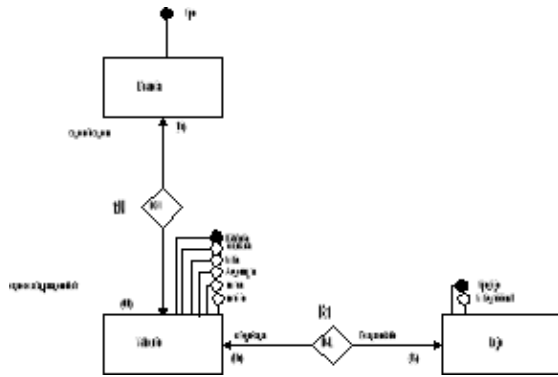
La entidad vehículo representa a los distintos vehículos de los que se vale la autoescuela para preparar a los alumnos en las distintas licencias que oferta. Los atributos son los siguientes:

- **Matrícula:** cadena de caracteres que identifica la matrícula del vehículo
- **Año_Compra:** Fecha de la compra del vehículo
- **Marca:** cadena de caracteres que identifica la marca del vehículo
- **Modelo:** cadena de caracteres que identifica el modelo del vehículo
- **Cilindrada:** Entero que indica la cilindrada del vehículo
- **Tarifa:** Entero que indica el coste asociado por hora al vehículo

La clave primaria de este tipo de entidad es el atributo **Matrícula**.

Como se puede apreciar, diferenciamos los vehículos en función de para qué licencia se pueden utilizar, relacionando cada vehículo con su licencia:

Además, con intención de considerar la presencia de vehículos de diferente lujo o categoría, hemos considerado una entidad **lujo** que indica a qué clase pertenece cada

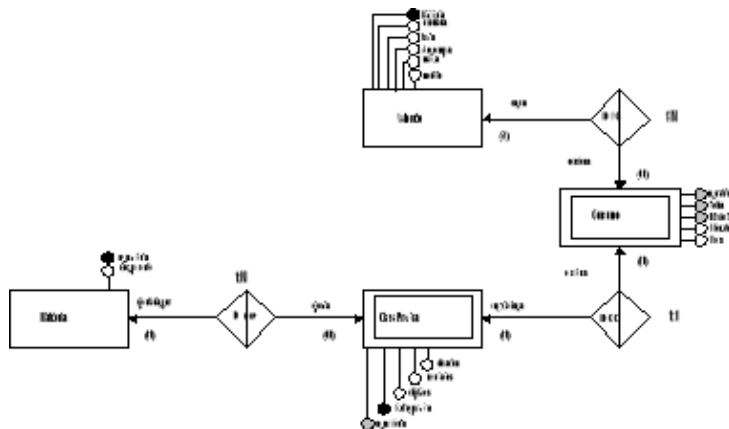


vehículo. Simplemente sirve para conocer la tarifa adicional que debe pagar el alumno sobre la normal en las clases prácticas. Sus atributos son:

- **Tipo_lujo:** cadena de caracteres que indica el tipo de lujo
- **Tarifa_adicional:** entero que indica la cantidad adicional a la tarifa base

La entidad **consumo** pretende reflejar el consumo en litros y los kilómetros recorridos por cada vehículo con objeto de conocer el rendimiento de los vehículos en cada clase práctica (Supuesto 16). Obviamente, este tipo de entidad es débil por identificación de **Vehículo**, ya que todos los consumos de los vehículos son iguales. Pero además, con intención de recoger este consumo en relación a cada clase práctica, esta entidad es además débil por identificación de la entidad **clase práctica**. De esta manera, la entidad **consumo** heredaría los atributos identificadores de las dos entidades anteriores, de tal manera que cada ocurrencia de la entidad **consumo** estaría relacionada siempre con un vehículo, y con una clase práctica. Los atributos de este tipo de entidad son:

- **Kilómetros:** Entero que refleja el número de kilómetros recorridos
- **Consumo:** Entero que refleja el consumo en litros del vehículo



8.2.3.3 Las Licencias y la Documentación

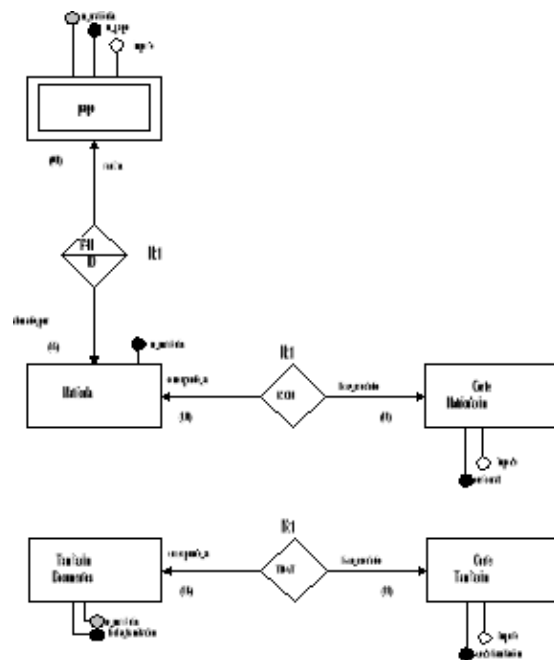
El primero representa a las distintas licencias de las que se da clase en la autoescuela. Sólo posee un atributo que indica el nombre de la licencia.

El segundo tipo de entidad representa los distintos documentos que se necesitan en la autoescuela para poder presentarse al examen práctico correspondiente. Como la entidad anterior posee un único atributo denominado nombre-doc y que es el nombre del documento, por ejemplo, certificado médico, DNI, licencia B, ect.

8.3 MODELADO CONCEPTUAL : ANALISIS DE INTERRELACIONES

8.3.1 Gestión Económica

Veamos el diagrama:



8.3.1.1 Tipo de Interrelación Matrícula/Pago(M-P)

Este tipo de interrelación refleja el pago que se realiza a cuenta de una matrícula en una licencia. Así, una matrícula puede tener asociados cero ó n pagos, mientras que un pago siempre está asociado a una matrícula

8.3.1.2 Tipo de Interrelación Tramitación Documentos/Coste Tramitación(TD-CT)

Esta interrelación fue explicada en el análisis de entidades. Sólo indicar que por cada tramitación siempre tiene asociado uno y sólo un coste de tramitación, mientras que un conste de tramitación está asociado a 1 o N tramitaciones.

8.3.1.3 Tipo de Interrelación Matrícula/Coste Matriculación(M-CM)

Con este tipo de interrelación pretendemos representar que una matrícula siempre lleva asociado un coste de matriculación, mientras que un coste de matriculación está asociado a 1 o N matrículas.

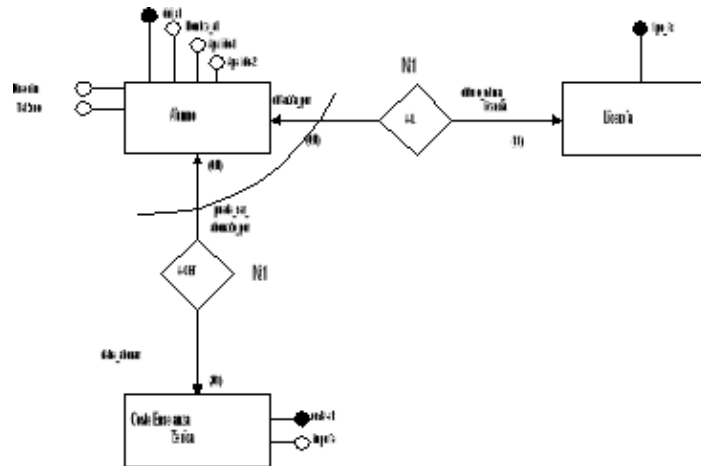
8.3.1.4 Tipo de Interrelación Alumno/Licencia(A-L)

Este tipo de interrelación representa la tenencia por parte de un alumno de una licencia, concretamente la última que ha obtenido. De esta manera, un alumno tendrá una o ninguna licencia, mientras que una licencia concreta pudo ser obtenida por cero o más alumnos.

8.3.1.5 Tipo de Interrelación Alumno/Coste Enseñanza Teórica(A-CET)

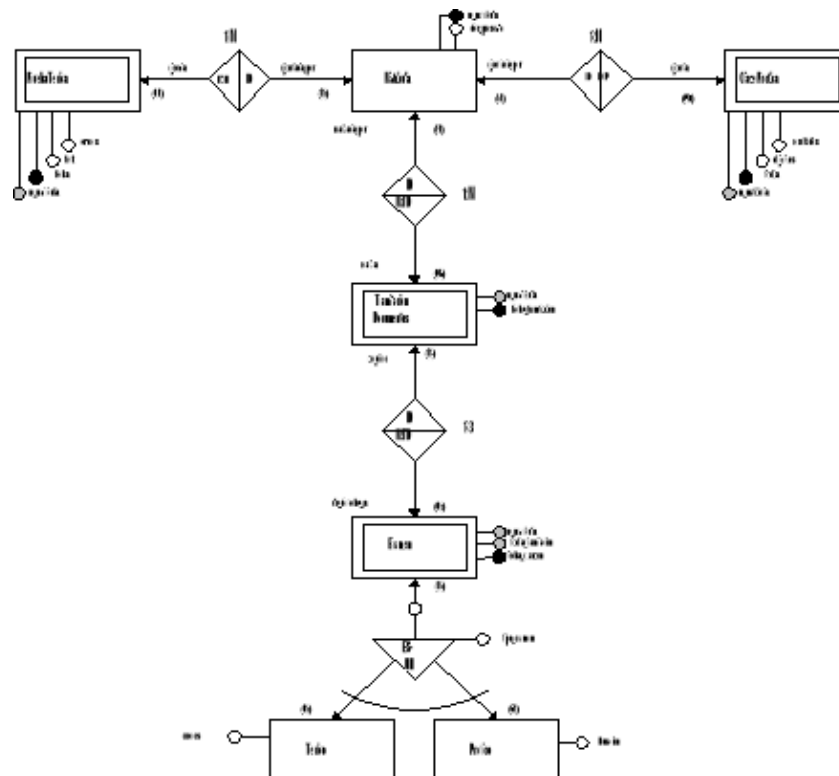
Este tipo de interrelación representa cuando un alumno debe pagar su enseñanza teórica. Este tipo de interrelación es exclusiva junto con la anterior respecto de la entidad **Alumno**. De esta manera, un alumno puede ser o no acreedor de un y sólo un coste por su enseñanza teórica. Y a su vez, el coste de enseñanza teórica puede ser abonado por cero o varios alumnos. Además se debe cumplir la restricción de que si un alumno posee una licencia, no pague su enseñanza teórica, y en caso de que la pague (porque no posea una

licencia anterior), sólo la abone una vez.



8.3.2 Gestión del Alumno

Veamos el diagrama:



8.3.2.1 Tipo de Interrelación Matrícula/Prueba Teórica(M-PT)

Este tipo de interrelación representa la asociación entre una matrícula, correspondiente a un alumno, y los productos de esta preparación en forma de ejercicios teóricos. Como podemos apreciar, una matrícula ejecuta o realiza 0 o N pruebas teóricas, mientras una prueba teórica, como es natural, es ejecutado por una y sólo una matrícula.

8.3.2.2 Tipo de Interrelación Matrícula/Clase Práctica

8.3.2.3 Tipo de Interrelación Matrícula/Alumno (MP-A)

8.3.2.4 Tipo de Interrelación Matrícula/Tramitación Documentación(M-TD)

8.3.2.5 Tipo de Interrelación Tramitación Documentación/Exámen(TD-E)

8.3.3 Gestión de Recursos de la Autoescuela

8.3.3.1 Tipo de Interrelación Docencia/Profesor (D-F)

8.3.3.2 Tipo de Interrelación Docencia/Licencia (D-L)

Este tipo de interrelación representa la licencia de la que da clase cada docencia. Así, una docencia imparte clases de una y sólo una licencia, mientras que una licencia concreta puede ser impartida por 0 (no hay profesores cualificados) ó N docencias.

8.3.3.3 Tipo de Interrelación Docencia/Vehículo(D-V)

Este tipo de interrelación representa la relación entre una docencia que imparte clases y el vehículo que necesita para ello. Así, una docencia tiene asignado uno y sólo un vehículo (supuesto 3), mientras que un vehículo está asociado a una y sólo una docencia.

8.3.3.4 Tipo de Interrelación Vehículo/Consumo (V-C)

Representa la relación entre un vehículo y los consumos asociados a este. Así un vehículo tiene asociados 0 ó N consumos, mientras que un consumo está asociado a un y sólo un vehículo.

8.3.3.5 Tipo de Interrelación Vehículo/Lujo(V-L)

Representa el lujo que un vehículo concreto de la autoescuela lleva asociado. Así, un lujo puede estar asociado a uno o más vehículos, mientras que un vehículo sólo puede estar asociado con un solo lujo.

8.3.3.6 Tipo de Interrelación Vehículo/Licencia(V-LC)

Representa el hecho de que para conducir un vehículo concreto es necesario poseer una licencia concreta. Más correctamente, un vehículo sólo puede servir para enseñar una licencia concreta. Así, un vehículo sirve para enseñar una y sólo una licencia, mientras que una licencia se puede enseñar en ninguno o muchos vehículos

8.3.3.7 Tipo de Interrelación Consumo/ClasePráctica(C-CP)

Este tipo de interrelación representa la dependencia de un consumo de la clase práctica en la que se realiza. La cardinalidad es uno a uno, ya que por cada clase práctica siempre se tiene asociado uno y sólo un consumo, mientras que un consumo siempre viene asociado a una y sólo una clase práctica.

8.3.3.8 Tipo de Interrelación Matrícula/Docencia(MP-D)

Mediante este tipo de interrelación podemos saber qué profesor se hace cargo de qué alumnos, y también a qué licencia corresponde cada matrícula. Se echa en falta una relación más directa entre la matrícula y el tipo de licencia matriculada, pero es la única manera (que hemos encontrado) que represente todos los supuestos del problema. Además, con este tipo de interrelación cumplimos un supuesto implícito del problema y es que un alumno no se podrá matricular de una licencia concreta si no existen docencias asociadas que impartan clases de esa licencia. Desde este punto de vista, tenemos que una matrícula está asignada a una y sólo una docencia, mientras una docencia puede tener asignadas 0 ó N matrículas.

8.3.3.9 Tipo de Interrelación Licencia/Documentación(L-D)

Este tipo de interrelación representa que documentos necesita que licencia para poder realizar la tramitación y obtener derechos a exámen. Así, un documento concreto puede ser requerido en 1 ó N licencias, mientras que una licencia concreta puede necesitar 1 ó N documentos diferentes.

DEFINICION SINTÁCTICA DEL PROBLEMA

GESTION DE AUTOESCUELAS
DEFINICION DE LOS TIPOS DE ENTIDAD

Atributos	Dominio	*	Tipo	restricciones
Licencia				
IP tipo_lic	CADENA(3)	1	a-z,0-9	
Documentacion				
IP nombre_doc	CADENA(30)	1	a-z,0-9	
Coste_Ensenianza_Teorica				
IP coste_et	CADENA(10)	1	a-z,0-9	
Importe	ENTERO(4)	1	0-9	
Lujo		1		
IP tipo_liujo	CADENA(10)	1	a-z	
Tarifa_adicional	ENTERO(4)	1	0-9	
Coste_Tramitacion		1		
IP coste_tram	CADENA(10)	1	a-z	
Importe	ENTERO(4)	1	0-9	
Profesor		1		
IP dni_prof	ENTERO(4)	1	0-9	
Nombre	CADENA(10)	1	a-z	
Apellido_1	CADENA(10)	1	a-z	
Apellido_2	CADENA(10)	1	a-z	
Direccion	CADENA(10)	1	a-z	
Telefono	ENTERO(9)	1	0-9	
Alumno		1		
IP dni_al	ENTERO(4)	1	0-9	
Nombre	CADENA(10)	1	a-z	
Apellido_1	CADENA(10)	1	a-z	
Apellido_2	CADENA(10)	1	a-z	
Direccion	CADENA(10)	1	a-z	
Telefono	ENTERO(9)	1	0-9	
Docencia		1		
IP dni_prof	ENTERO(4)	1	0-9	Heredado
IP tipo_lic	CADENA(3)	1	a-z,0-9	Heredado
Matricula		1		
IP n_matricula	ENTERO(10)	1	0-9	
Doc_presente	CADENA(1)	1	a-z	= S,N
Prueba Teorica		1		
IP fecha_prueba	Date	1	Date	
IP n_matricula	ENTERO(10)	1	0-9	Heredado
N_test	ENTERO(4)	1	0-9	
Errores	ENTERO(2)	1	0-9	
Pago		1		
IP n_pago	ENTERO(4)	1	0-9	
IP n_matricula	ENTERO(10)	1	0-9	Heredado
Importe	ENTERO(6)	1	0-9	

Clase_Practica		1		
IP fecha_practica	Date	1	Date	
IP n_matricula	ENTERO(10)	1	0-9	Heredado
Duracion	ENTERO(2)	1	0-9	
Objetivos	CADENA(50)	1	a-z,0-9	
Resultados	CADENA(50)	1	a-z,0-9	
Tramitacion_Documentos				
IP fecha_tramitacion	Date	1		
IP n_matricula	ENTERO(10)	1	0-9	Heredado
Examen				
IP fecha_examen	Date	1	Date	
IP n_matricula	ENTERO(10)	1	0-9	Heredado
IP fecha_tramitacion	Date	1	Date	Heredado
Teorico				
IP fecha_examen	Date	1	Date	Heredado
IP n_matricula	ENTERO(10)	1	0-9	Heredado
IP fecha_tramitacion	Date	1	Date	Heredado
Errores	NUMBER(2)	1	0-9	
Practico				
IP fecha_examen	Date	1	Date	Heredado
IP n_matricula	ENTERO(10)	1	0-9	Heredado
IP fecha_tramitacion	Date	1	Date	Heredado
Duracion	NUMBER(2)	1	0-9	
Vehiculo		1		
IP matricula	CADENA(6)	1	a-z,0-9	
Marca	CADENA(10)	1	a-z,0-9	
Modelo	CADENA(10)	1	a-z,0-9	
Cilindrada	ENTERO(4)	1	0-9	
Año_compra	Date	1	Date	
Tarifa	ENTERO(4)	1	0-9	
Consumo		1		
IP Matricula	CADENA(6)	1	a-z,0-9	Heredado
IP n_matricula	ENTERO(10)	1	0-9	Heredado
IP fecha_practica	Date	1	Date	Heredado
Kilometros	ENTERO(3)	1	0-9	
Litros	ENTERO(2)	1	0-9	

DEFINICION DE LOS TIPOS DE INTERRELACION						
Entidad	C(m)	C(M)	Entidad	C(m)	C(M)	Atributos
D_L						
Documentacion	1	N	Licencia	1	N	
V_LC						
Licencia	1	1	Vehiculo	0	N	

L_D					
Licencia	1	1	Docencia	0	N
P_D					
Profesor	1	1	Docencia	1	N
D_V					
Docencia	1	1	Vehiculo	1	1
D_M					
Docencia	1	1	Matricula	0	N
PT_M					
Prueba_Teorica	1	N	Matricula	1	1
CP_M					
Clase_Practica	0	N	Matricula	1	1
CP_C					
Clase_Practica	1	1	Consumo	1	1
C_V					
Consumo	0	N	Vehiculo	1	1
M_TD					
Tramitacion_Documentos	0	N	Matricula	1	1
TD_E					
Tramitacion_Documentos	1	1	Examen	1	3
ES_UN JERARQUICA TOTAL EXCLUSIVA DE EXAMEN					
Teorico	0	1	Practico	0	1
M_CM					
Matricula	1	N	Coste_Matriculacion	1	1
TD_CT					
Tramitacion_Documentos	1	N	Coste_Tramitacion	1	1
A_L					
Licencia	0	1	Alumno	0	N
A_CET					
Coste_Enseñanza_Teorica	0	1	Alumno	0	N
M_P					
Matricula	1	1	Pago	0	N
A_M					
Alumno	1	1	Matricula	1	N
V_L					
Vehiculo	1	N	Lujo	1	1

8.4 MODELO RELACIONAL

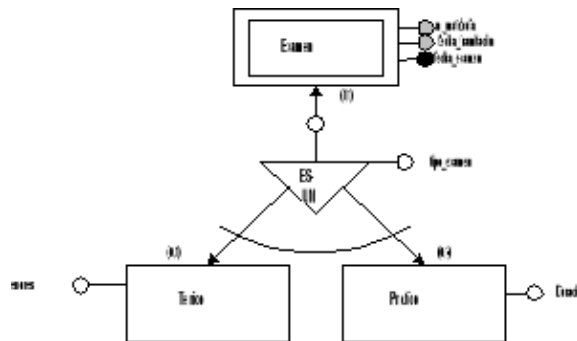
A continuación vamos a derivar el esquema relacional correspondiente al modelo conceptual descrito en la sección anterior.

Como podemos apreciar en el modelo conceptual, ninguna entidad contiene atributos múltiples ni compuestos, con lo que sólo debemos estudiar las relaciones jerárquicas que existen en el modelo antes de proceder a su traducción

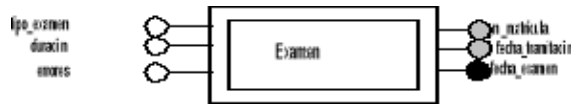
8.4.1 Eliminación de las Relaciones Jerárquicas

8.4.1.1 Los Exámenes

En el caso de la entidad **exámen** procedemos de la misma manera que con la entidad anterior. Como podemos apreciar en el diagrama conceptual, ninguno de los subtipos mantiene relaciones con ningún otro tipo de entidad del esquema:



Como consecuencia, aplicamos la regla **PRTECAR-4**, y obtenemos:



Esta solución, al igual que la anterior, tiene el problema de que si en el futuro se realiza alguna ampliación por el cambio de normativa que obligue a considerar entidades que de alguna manera se relacionen con los subtipos eliminados, los cambios tendrán un impacto mayor que si eliminamos la relación jerárquica en vez de los subtipos.

8.4.2 Traducción del Modelo Conceptual a Relacional

Vamos a considerar ahora las tablas generadas a partir de la aplicación de las reglas de traducción **RETCAR** para generar el modelo relacional.

8.4.2.1 El Alumno

Como podemos apreciar, el alumno mantiene relaciones con las entidades **matricula**, **licencia** y **coste-enseñanza-teórica**. Atendiendo a las cardinalidades, vemos que la tabla **matricula** sufre una modificación:

Matricula(nº_matricula, doc-presente, dni-al)

Para representar la exclusividad de las relaciones **A-L** y **A-CET**, podemos considerar la siguiente representación:

Alumno(dni-al, nombre, apellido-1, apellido-2, dirección, teléfono, **tipo-lic**, **coste-et**)

Y considerando la restricción de que los atributos *tipo-lic* y *coste-et* no pudieran a la vez tomar el valor NULL.

8.4.2.3 El Coste de Enseñanza Teórica

Este tipo de entidad sólo mantiene una interrelación con la entidad *alumno* y por el carácter de esta y la cardinalidad con la que participa (1,1), no requiere modificación. Así por la regla **RTECAR-1**:

Coste-Enseñanza-Teórica(coste-et,importe)

8.4.2.4 El Pago

La tabla pago, debido a la interrelación *M-P* quedaría de la siguiente forma

Pago (n_pago,nº_matricula,importe)

Ya que la dependencia de la entidad *matricula* es por identificación, y en la relación M-P participa con cardinalidad mínima cero y máxima muchos.

8.4.2.5 La Matricula

La tabla matricula mantiene relaciones con las entidades *clase_practica*, *tramitacion_documentos*, *pago*, *prueba_teorica*, *docencia* y *coste-matriculacion*. Sólo en las relaciones *M-CM* y *M-D* esta entidad participa con cardinalidad máxima muchos. Así, en virtud de la regla **RTECAR-3.1** tenemos:

Matricula(n_matricula,doc-presente,dni-al,dni-prof,tipo_lic,coste-mat)

Y considerando que las tres claves foráneas definidas no podrán tomar valores nulos en ninguna extensión de Matricula, preservando así la cardinalidad mínima de *docencia* y *coste-matriculación* en sus respectivas relaciones.

8.4.2.6 Las Licencias

Este tipo de entidad mantiene relaciones con las entidades *docencia*, *alumno* y *documentación*. En ninguna de ellas la tabla debe modificarse, con lo que la aplicación de la regla RTECAR-1 es suficiente:

Licencia(tipo-lic)

8.4.2.7 La Documentación

Este tipo de entidad sólo mantiene una relación con la entidad Licencia, y por la cardinalidad con la que participa y el carácter de la interrelación, su tabla no requiere modificación. Por tanto aplicando la regla RTECAR-1 obtenemos:

Documentación(nombre-doc)

8.4.2.8 El Coste de Matriculación

Esta entidad sólo mantiene relación con la entidad matrícula, y por el carácter de ésta no sufre modificaciones y queda:

Coste_Matriculacion(coste-mat,importe)

8.4.2.9 Las Pruebas Teóricas

Esta entidad sólo mantiene relación con **matricula** que le da carácter de debilidad por identificación. Como consecuencia, la representación de este tipo de entidad queda como sigue:

Prueba_Teórica(fecha-prueba, nº matricula, nº_test,errores)

Tabla obtenida en virtud de la regla **RTECAR-3.1**, ya que la cardinalidad con la que participa esta entidad en la interrelación **PT-M** es (0,N).

8.4.2.10 Las Clases Practicas

Esta entidad, al igual que la anterior, mantiene una relación con la entidad **matricula** de la que es débil por identificación. Como consecuencia, esta entidad heredaré el identificador de **matricula**. Además como participa en la interrelación con cardinalidad (0,N), aplicando la regla **PRTECAR-3.1** obtenemos:

Clase_Practica(fecha-practica, nº-matricula ,duracion,objetivos,resultados)

La interrelación que esta entidad mantiene con la entidad consumo, aparentemente (su cardinalidad en esta interrelación es (1,1)) modificaría la tabla, pero como veremos más adelante no es así.

8.4.2.11 La Tramitación de Documentos

Esta entidad mantiene relación con la entidad matrícula de la que es débil por identificación. En esta interrelación **M-TD** la entidad **tramitación_documentos** participa con cardinalidad (0,N). Además tenemos una relación uno a muchos con la entidad **coste_tramitacion**. Aplicando la regla **PRTECAR-3.1** tenemos:

Tramitacion_documentos(fecha_tramitacion,nº matricula,coste_tram)

8.4.2.12 El Coste de Tramitación de Documentos

Esta entidad no sufre modificaciones ya que la única interrelación que mantiene, la mantiene con la entidad **Tramitación_documentos**, en la que participa con cardinalidad máxima y mínima uno:

Coste_Tramitacion(coste-tram,importe)

8.4.2.13 Los Exámenes

La entidad **examen** mantiene relaciones con la entidad tramitación de documentos de la que depende por identificación. Aunque la cardinalidad con la que la entidad **examen** participa en la interrelación **TD-E** es (0,3), podemos considerar que esta relación es uno a muchos, y controlar mediante software la restricción de que no puede haber más de tres exámenes relacionados con la misma tramitación de documentos. De esta manera, aplicando la regla **RTECAR-3.1** obtenemos:

Examen(fecha-examen,nº-matricula,fecha-tramitacion,tipo-examen

,duracion, errores)

8.4.2.14 Los Profesores

Esta entidad no sufre modificaciones ya que en la única interrelación que mantiene con docencia, participa con cardinalidad máxima y mínima igual a uno. La tabla queda como sigue:

Profesor(dni-prof, nombre, apellido-1, apellido-2, direccion, telefono)

8.4.2.15 Las Docencias

Este tipo de entidad mantiene relaciones con las entidades **licencia**, **profesor** y **vehículo**. En los tipos de interrelación **P-D** y **L-D** esta entidad participa con cardinalidad (0,N) y (1,N). Para ambas podemos aplicar la regla **RTECAR-3.1** obteniendo:

Docencia(dni-prof , tipo-lic)

Considerando que la entidad es débil por identificación de **Licencia** y **Profesor**.

Sin embargo en el tipo de interrelación **D-V**, las dos entidades participan con cardinalidad máxima y mínima igual a 1. Por ello y en virtud de la regla RTECAR-2.1 obtenemos

Docencia(dni-prof , tipo-lic , matricula)

8.4.2.16 Los Vehículos

La entidad **vehículo** sólo mantiene relaciones con las entidades **lujo**, **docencia** , **Licencia** y **Consumo**. De estas interrelaciones, las únicas que introducen modificaciones en las tablas son las tres primeras, ya que en el tipo de interrelación **D-V** participa con cardinalidad máxima y mínima igual a uno, en el tipo de interrelación **V-L** participa con cardinalidad (1,N) y en el tipo de interrelación **LC-V** con cardinalidad (0,N) . Aplicando la regla **RTECAR-2.1** y la **RTECAR-3.1** obtenemos:

Vehiculo(matricula, marca, modelo, cilindrada, año-compra, tarifa, dni-prof

,tipo-lic, tipo-lujo , licencia_necesaria)

Recordando lo expuesto en el modelo conceptual, teníamos una restricción éste no cumplía referente a las docencias y los vehículos. Se nos presentaba la restricción de que una docencia sólo puede tener asociado un vehículo en el que se pudiera impartir la licencia que enseña la docencia. Pues bien, si imponemos en el modelo relacional que el valor del atributo tipo-lic y el atributo tipo-lic1 tengan el mismo valor, obligamos a que si un vehículo está asociado a una docencia coincidan la licencia que se imparte y la licencia necesaria del vehículo. Notar que hemos denominado licencia_necesaria a la clave foránea que representa la interrelación entre los vehículos y las licencias.

8.4.2.17 El Lujo de los Vehículos

Este tipo de entidad sólo mantiene un tipo de interrelación con la entidad vehículo, en la que participa con cardinalidad máxima y mínima igual a uno con lo que no sufre modificaciones, quedando de la siguiente forma:

Lujo(tipo-lujo, tarifa-adicional)

8.4.2.18 El Consumo

Este tipo de entidad mantiene dos tipos de interrelación. Uno con la entidad **vehículo** y otro con la entidad **clase-practica**. Esta entidad es débil por identificación respecto a las dos entidades, pero en el tipo de

interrelación **C-V** participa con cardinalidad (0,N), mientras que en el tipo de interrelación **C-CP** las dos entidades participan con cardinalidad máxima y mínima igual a uno. Siguiendo la regla RTECAR-2.1 para la interrelación **C-CP** y la regla RTECAR-3.1 para la interrelación **C-V** obtenemos:

Consumo(matricula , n°_matricula , fecha-practica ,kilometros,litros)

Como podemos comprobar, esta representación de la relación entre clase práctica y consumo permite semánticamente que existan ocurrencias de la entidad **Clase-práctica** que no tengan un consumo asociado, aunque la cardinalidad máxima y mínima de **Consumo** sea 1. El problema viene de que si consideramos que esta es una interrelación uno a uno en la que los dos tipos de entidad participan de forma completa, ambos tipos de interrelación tendrán que referirse de alguna manera a la otra. Pero si hacemos esto así, estaríamos repitiendo los atributos identificadores en la entidad **Clase-práctica**, ya que estos forman parte del identificador de **Consumo** (recordemos que es débil por identificación). Debido a las complicaciones que esto tendría, consideraremos que el control de la cardinalidad de la entidad **Consumo** se realizará vía software, obligando a que cada vez que se haga una inserción en la tabla **Clase-Practica**, se haga la correspondiente inserción en la tabla **Consumo**.

8.4.2.19 La Interrelación D-L

Esta es el único tipo de interrelación que necesita una tabla propia en el esquema por su carácter de muchos a muchos. De esta manera y aplicando la regla RTECAR-4 obtenemos:

D-L(tipo-lic,nombre-doc)

8.5 ESQUEMA RELACIONAL

Alumno(dni-al,nombre,apellido-1,apellido-2,direccion,telefono, **tipo-lic**, **coste-et**)

Coste-Enseñanza-Teórica(coste-et,importe)

Pago (n°_pago, n°_matricula,importe)

Matricula(n°_matricula,doc-presente,**dni-al**,**dni-prof**,**codigo-docencia**,**coste-mat**)

Licencia(tipo-lic)

Documentación(nombre-doc)

Coste_Matriculacion(coste-mat,importe)

Prueba_Teórica(fecha-prueba, **n-matricula**,n_test,errores)

Clase_Practica(fecha-practica, **n-matricula** ,duracion,objetivos,resultados)

Tramitacion_documentos(fecha_tramitacion,n-matricula,coste_mat)

Coste_Tramitacion(coste-tram,importe)

Examen(fecha-examen,n°-matricula,fecha-tramitacion,tipo-xamen,duracion,errores)

Profesor(dni-prof,nombre,apellido-1,apellido-2,direccion,telefono)

Docencia(dni-prof , tipo-lic , matricula)

Vehiculo(matricula,marca,modelo,cilindrada,año-compra,tarifa,tipo-lic, dni-prof,
tipo-lujo , **licencia_necesaria**)

Lujo(tipo-lujo,tarifa-adicional)

Consumo(matricula , n°_matricula , fecha-practica ,kilometros,litros)

D-L(tipo-lic,nombre-doc)

8.6 NORMALIZACION DEL MODELO

Como podemos apreciar, todas las relaciones generadas en la traduccion del modelo conceptual se encuentran en FNBC. Ahora mostramos el diagrama de dependencias

Alumno

dni nombre,apellido-1,apellido-2,direccion,telefono, **tipo_lic**, **coste_et**

Coste_Enseñanza_Teórica

coste-et importe

Pago

n_pago,n_matricula importe

Matricula

n_matricula doc-presente,**dni-al**,**dni-prof**,**codigo-docencia**,**coste-mat**

Licencia

tipo-lic

Documentación

Nombre_doc

Coste_Matriculacion

Coste_mat importe

Prueba_Teórica

Fecha_prueba,n_matricula n°-test,errores

Clase Practica

Fecha_practica, n_matricula duracion,objetivos,resultados

Tramitacion_documentos

Fecha_tramitacion,n_matricula coste_tram

Coste_Tramitacion

Coste_tram importe

Examen

Fecha_examen, tipo-xamen,duracion,errores

n_matricula,fecha_tramitacion

Profesor

dni-prof nombre,apellido-1,apellido-2,direccion,telefono

Docencia

dni-prof , tipo-lic matricula

Vehiculo

Matricula marca,modelo,cilindrada,año-compra,tarifa,tipo-lic, dni-prof,

tipo-lujo , licencia_necesaria

tipo-lic, dni-prof marca,modelo,cilindrada,año-compra,tarifa

tipo-lujo , licencia_necesaria, matricula

Lujo

tipo-lujo tarifa-adicional

Consumo

matricula , n°_matricula , fecha-practica kilometros,litros

D-L

tipo-lic,nombre-doc

8.7 SENTENCIAS DE DEFINICION Y MANIPULACION DE LA BASE DE DATOS

8.7.1 Sentencias de definicion sintactica de las tablas de la base de datos

CREATE TABLE Licencia

(

```

tipo_lic VARCHAR(3) NOT NULL,

CONSTRAINT pk_Licencia

PRIMARY KEY (tipo_lic)

);

CREATE TABLE Documentacion

(

nombre_doc VARCHAR(30) NOT NULL,

CONSTRAINT pk_Documentacion

PRIMARY KEY (nombre_doc)

);

CREATE TABLE Coste_Ensenianza_Teorica

(

coste_et VARCHAR(10) NOT NULL,

importe NUMBER(4) NOT NULL,

CONSTRAINT pk_Coste_Ensenianza_Teorica

PRIMARY KEY (coste_et)

);

CREATE TABLE Coste_Matriculacion

(

coste_mat VARCHAR(14) NOT NULL,

importe NUMBER(4) NOT NULL,

CONSTRAINT pk_Coste_Matriculacion

PRIMARY KEY (coste_mat)

);

CREATE TABLE Lujo

(

```

```

tipo_lujo VARCHAR(10),

tarifa_adicional NUMBER(4),

CONSTRAINT pk_Lujo

PRIMARY KEY (tipo_lujo)

);

CREATE TABLE Coste_Tramitacion

(

coste_tram VARCHAR(10) NOT NULL,

importe NUMBER(4),

CONSTRAINT pk_Coste_Tramitacion

PRIMARY KEY (coste_tram)

);

CREATE TABLE Profesor

(

dni_prof NUMBER(4) NOT NULL,

nombre VARCHAR(10) NOT NULL,

apellido_1 VARCHAR(10) NOT NULL,

apellido_2 VARCHAR(10) NOT NULL,

direccion VARCHAR(20),

telefono NUMBER(9),

CONSTRAINT pk_Profesor

PRIMARY KEY (dni_prof)

);

CREATE TABLE Alumno

(

dni_al NUMBER(4) NOT NULL ,

```

```

nombre VARCHAR(10) NOT NULL ,
apellido_1 VARCHAR(10) NOT NULL ,
apellido_2 VARCHAR(10) NOT NULL ,
direccion VARCHAR(20) ,
telefono NUMBER(9) ,
tipo_lic VARCHAR(3) ,
coste_et VARCHAR(10) ,
CONSTRAINT pk_Alumno
PRIMARY KEY (dni_al) ,
CONSTRAINT fk_Licencia
FOREIGN KEY (tipo_lic)
REFERENCES Licencia(tipo_lic) ,
CONSTRAINT fk_Coste_Ensenianza_Teorica
FOREIGN KEY (coste_et)
REFERENCES Coste_Ensenianza_Teorica(coste_et),
CONSTRAINT ck_exclusiva_lic_cet
CHECK (
( (NOT(coste_et IS NULL)) AND (tipo_lic IS NULL) )
OR
( (coste_et IS NULL) AND (NOT(tipo_lic IS NULL)) )
)
);
CREATE TABLE Docencia
(
dni_prof NUMBER(4) NOT NULL,
tipo_lic VARCHAR(3) NOT NULL,

```

```

matricula VARCHAR(6),

CONSTRAINT pk_Docencia

PRIMARY KEY (dni_prof, tipo_lic),

CONSTRAINT fk_Profesor

FOREIGN KEY (dni_prof)

REFERENCES Profesor(dni_prof)

ON DELETE CASCADE,

CONSTRAINT fk_tipo_lic

FOREIGN KEY (tipo_lic)

REFERENCES Licencia(tipo_lic)

ON DELETE CASCADE,

CONSTRAINT sk_docencia

UNIQUE (matricula),

CONSTRAINT fk_vehiculo

FOREIGN KEY (matricula)

REFERENCES vehiculo(matricula),

CONSTRAINT ck_docencia

CHECK (matricula <> NULL)

);

CREATE TABLE Matricula

(

n_matricula NUMBER(10) NOT NULL,

doc_presente NUMBER(1) NOT NULL,

dni_al NUMBER(4) NOT NULL,

dni_prof NUMBER(4) ,

tipo_lic VARCHAR(3) ,

```

```

coste_mat VARCHAR(14) NOT NULL,

CONSTRAINT pk_Matricula

PRIMARY KEY (n_matricula),

CONSTRAINT fk_Alumno

FOREIGN KEY (dni_al)

REFERENCES Alumno(dni_al),

CONSTRAINT fk_Docencia

FOREIGN KEY (tipo_lic,dni_prof)

REFERENCES Docencia(tipo_lic,dni_prof),

CONSTRAINT fk_Coste_Matriculacion

FOREIGN KEY (coste_mat)

REFERENCES Coste_Matriculacion(coste_mat),

CONSTRAINT ck_doc_presente

CHECK ( ((doc_presente <= 1)

OR (doc_presente > 0))

)

);

CREATE TABLE Prueba_Teorica

(

fecha_prueba DATE NOT NULL,

N_matricula NUMBER(10) NOT NULL,

N_test NUMBER(4) NOT NULL,

errores NUMBER(2) NOT NULL,

CONSTRAINT pk_Prueba_Teorica

PRIMARY KEY (fecha_prueba,N_matricula),

CONSTRAINT fk_Matricula

```

```

FOREIGN KEY (N_matricula)

REFERENCES Matricula(N_matricula)

ON DELETE CASCADE

);

CREATE TABLE Pago

(

N_pago NUMBER(4) NOT NULL,

N_matricula NUMBER(10) NOT NULL,

importe NUMBER(6) NOT NULL,

CONSTRAINT pk_Pago

PRIMARY KEY (N_pago,N_matricula),

CONSTRAINT ck_importe

CHECK ( importe > 0 ),

CONSTRAINT fk_Mat

FOREIGN KEY (N_matricula)

REFERENCES Matricula(N_matricula)

ON DELETE CASCADE

);

```

```

CREATE TABLE Clase_Practica

(

fecha_practica DATE NOT NULL,

N_matricula NUMBER(10) NOT NULL,

duracion NUMBER(2) NOT NULL,

objetivos VARCHAR(50),

resultados VARCHAR(50),

CONSTRAINT pk_Clase_Practica

```

```

PRIMARY KEY (fecha_practica,N_matricula),

CONSTRAINT fk_Matric

FOREIGN KEY (N_matricula)

REFERENCES Matricula(n_matricula)

ON DELETE CASCADE,

CONSTRAINT ck_duracion

CHECK ( duracion < 10)

);

CREATE TABLE Tramitacion_Documentos

(

fecha_tramitacion DATE NOT NULL,

N_matricula NUMBER(10) NOT NULL,

coste_tram VARCHAR(10) NOT NULL,

CONSTRAINT pk_Tramitacion_Documentos

PRIMARY KEY (fecha_tramitacion,N_matricula),

CONSTRAINT fk_Matr

FOREIGN KEY (N_matricula)

REFERENCES Matricula(N_matricula)

ON DELETE CASCADE,

CONSTRAINT fk_coste_tram

FOREIGN KEY (coste_tram)

REFERENCES coste_tramitacion(coste_tram)

);

CREATE TABLE Examen

(

fecha_examen DATE NOT NULL,

```

```

N_matricula NUMBER(10) NOT NULL,

fecha_tramitacion DATE NOT NULL,

tipo_examen VARCHAR(1) NOT NULL,

duracion NUMBER(2),

errores NUMBER(2),

CONSTRAINT pk_Examen

PRIMARY KEY (fecha_examen,

N_matricula,fecha_tramitacion),

CONSTRAINT ck_tipo_examen

CHECK (tipo_examen IN ('T','P')),

CONSTRAINT ck_valor_error_duracion

CHECK (

( (tipo_examen = 'T') AND (errores <> NULL)

AND (duracion = NULL) )

OR

( (tipo_examen = 'P') AND (duracion <> NULL)

AND (errores = NULL) )

),

CONSTRAINT fk_Tramitacion_Documentos

FOREIGN KEY (fecha_tramitacion,N_matricula)

REFERENCES Tramitacion_Documentos(

fecha_tramitacion,N_matricula)

ON DELETE CASCADE

);

CREATE TABLE Vehiculo

(

```

```

matricula VARCHAR(6) NOT NULL,
marca VARCHAR(10) NOT NULL,
modelo VARCHAR(10) NOT NULL,
cilindrada NUMBER(4) NOT NULL,
anio_compra DATE NOT NULL,
tarifa NUMBER(4) NOT NULL,
tipo_lic VARCHAR(3),
dni_prof NUMBER(4),
tipo_lujo VARCHAR(10) NOT NULL,
licencia_necesaria VARCHAR(3) NOT NULL,
CONSTRAINT pk_Vehiculo
PRIMARY KEY (matricula),
CONSTRAINT ck_Licencia_Vehiculo
CHECK (tipo_lic = licencia_necesaria ),
CONSTRAINT sk_Vehiculo
UNIQUE (dni_prof,tipo_lic),
CONSTRAINT fk_Docen
FOREIGN KEY (dni_prof,tipo_lic)
REFERENCES Docencia(dni_prof,tipo_lic),
CONSTRAINT fk_Licenc
FOREIGN KEY (licencia_necesaria)
REFERENCES Licencia(tipo_lic),
CONSTRAINT fk_Lujo
FOREIGN KEY (tipo_lujo)
REFERENCES Lujo(tipo_lujo)
);

```

```

CREATE TABLE Consumo
(
matricula VARCHAR(6) NOT NULL,
N_matricula NUMBER(10) NOT NULL ,
fecha_practica DATE NOT NULL,
kilometros NUMBER(3) NOT NULL,
litros NUMBER(2) NOT NULL,
CONSTRAINT pk_Consumo
PRIMARY KEY (matricula,N_matricula,fecha_practica),
CONSTRAINT fk_Clase_Practica
FOREIGN KEY (fecha_practica,N_matricula)
REFERENCES Clase_Practica
(fecha_practica,N_matricula)
ON DELETE CASCADE,
CONSTRAINT fk_Vehic
FOREIGN KEY (matricula)
REFERENCES Vehiculo(matricula)
ON DELETE CASCADE
);

CREATE TABLE D_L
(
tipo_lic VARCHAR(3) NOT NULL,
nombre_doc VARCHAR(30) NOT NULL,
CONSTRAINT pk_D_L
PRIMARY KEY (tipo_lic,nombre_doc),
CONSTRAINT fk_Lic

```

```
FOREIGN KEY (tipo_lic)
REFERENCES Licencia(tipo_lic)
ON DELETE CASCADE,
CONSTRAINT fk_Documentacion
FOREIGN KEY (nombre_doc)
REFERENCES Documentacion(nombre_doc)
ON DELETE CASCADE
);
```

8.7.2 Sentencias de borrado de las tablas de la base de datos

```
ALTER TABLE DOCENCIA
DROP CONSTRAINT FK_VEHICULO;
DELETE D_L;
DROP TABLE D_L;
DELETE DOCUMENTACION;
DROP TABLE DOCUMENTACION;
DELETE PRUEBA_TEORICA;
DROP TABLE PRUEBA_TEORICA;
DELETE CONSUMO;
DROP TABLE CONSUMO;
DELETE VEHICULO;
DROP TABLE VEHICULO;
DELETE CLASE_PRACTICA;
DROP TABLE CLASE_PRACTICA;
DELETE LUJO;
DROP TABLE LUJO;
DELETE EXAMEN;
```

DROP TABLE EXAMEN;
DELETE TRAMITACION_DOCUMENTOS;
DROP TABLE TRAMITACION_DOCUMENTOS;
DELETE PAGO;
DROP TABLE PAGO;
DELETE MATRICULA;
DROP TABLE MATRICULA;
DELETE COSTE_MATRICULACION;
DROP TABLE COSTE_MATRICULACION;
DELETE DOCENCIA;
DROP TABLE DOCENCIA;
DELETE PROFESOR;
DROP TABLE PROFESOR;
DELETE ALUMNO;
DROP TABLE ALUMNO;
DELETE COSTE_ENSENIANZA_TEORICA;
DROP TABLE COSTE_ENSENIANZA_TEORICA;
DELETE LICENCIA;
DROP TABLE LICENCIA;
DELETE COSTE_TRAMITACION;
DROP TABLE COSTE_TRAMITACION;

8.7.3 Sentencias de insercion de informacion en la base de datos

INSERT INTO Licencia
VALUES
 ('A');
INSERT INTO Licencia

```

VALUES ('B');

INSERT INTO Licencia

VALUES ('C');

INSERT INTO Documentacion

VALUES ('DNI');

INSERT INTO Documentacion

VALUES ('CERT MEDICO');

INSERT INTO Documentacion

VALUES ('CERT PSICOL');

INSERT INTO Documentacion

VALUES ('LICENCIA A');

INSERT INTO Documentacion

VALUES ('LICENCIA B');

INSERT INTO Documentacion

VALUES ('LICENCIA C');

INSERT INTO Documentacion

VALUES ('CERT DGT');

INSERT INTO Profesor

VALUES (1,'Rafael','Andino','Pozo','Onieva 3',00001);

INSERT INTO Profesor

VALUES (2,'Ana','Ruiz','Romero','Garberas 16',00002);

INSERT INTO Profesor

VALUES (3,'Carolina','Arredondo','Ruiz','Liberacion 3',00003);

INSERT INTO Profesor

VALUES (4,'Nestor','Arredondo','Martinez','Liberacion 3',00004);

INSERT INTO Docencia

```

```

VALUES (1,'A','000005');

INSERT INTO Docencia

VALUES (2,'A','000006');

INSERT INTO Docencia

VALUES (2,'B','000001');

INSERT INTO Docencia

VALUES (3,'B','000002');

INSERT INTO Docencia

VALUES (3,'C','000003');

INSERT INTO Docencia

VALUES (4,'C','000004');

INSERT INTO Coste_Ensenianza_Teorica

VALUES ('coste_e_t',4500);

INSERT INTO Coste_Matriculacion

VALUES ('coste_mat',3000);

INSERT INTO Coste_Tramitacion

VALUES ('coste_tram',3000);

INSERT INTO Lujo

VALUES ('normal',500);

INSERT INTO Lujo

VALUES ('extra',1000);

INSERT INTO Lujo

VALUES ('ejecutivo',2000);

INSERT INTO Lujo

VALUES ('exclusivo',5000);

INSERT INTO Vehiculo

```

VALUES ('000001','Ford','Escort',2000,TO_DATE('21/3/99','DD/MM/YY'),

1000,'B',2,'normal','B');

INSERT INTO Vehiculo

VALUES ('000002','Wolkswagen','Golf TDI',1900,TO_DATE('31/3/95',

'DD/MM,YY'),1000,'B',3,'extra','B');

INSERT INTO Vehiculo

VALUES ('000003','Scania','Globe',4500,TO_DATE('7/7/97',

'DD/MM/YY'),3000,'C',3,'normal','C');

INSERT INTO Vehiculo

VALUES ('000004','Scania','RoadRun',6000,TO_DATE('1/6/99',

'DD/MM/YY'),3000,'C',4,'ejecutivo','C');

INSERT INTO Vehiculo

VALUES ('000005','Suzuki','MRZ',500,TO_DATE('27/8/98','DD/MM/YY'),

700,'A',1,'exclusivo','A');

INSERT INTO Vehiculo

VALUES ('000006','Honda','MRV',500,TO_DATE('5/2/95','DD/MM/YY'),

700,'A',2,'normal','A');

INSERT INTO Alumno

VALUES (1,'Jose','Garcia','Perez','Liberacion 4',1,'A',NULL);

INSERT INTO Alumno

VALUES (2,'Ana','Belen','Ruiz','Plaza 1',2,'B',NULL);

INSERT INTO Alumno

VALUES (3,'Elisa','Roca','Perez','Plaza 3',3,NULL,'coste_e_t');

INSERT INTO Alumno

VALUES (4,'Petra','Garcia','Obregon','Orgullo 1',4,'B',NULL);

INSERT INTO Alumno

VALUES (5,'Pepa','Martos','Garcia','Canto 1',5,'A',NULL);

INSERT INTO Alumno

VALUES (6,'Antonio','Molina','Mora','Campillo 7',6,'B',NULL);

INSERT INTO Alumno

VALUES (7,'Elisabeth','Ruiz','Romero','Garberas 1',7,NULL,'coste_e_t');

INSERT INTO Alumno

VALUES (8,'Juan','Martinez','Fontiveros','Amador 1',8,NULL,'coste_e_t');

INSERT INTO Alumno

VALUES (9,'Susana','Ruiz','Pozo','Amador 2',9,NULL,'coste_e_t');

INSERT INTO Alumno

VALUES (10,'Pepe','Barba','Martinez','Prado 9',10,'B',NULL);

INSERT INTO Matricula

VALUES (0,0,1,2,'B','coste_mat');

INSERT INTO Matricula

VALUES (1,0,2,3,'C','coste_mat');

INSERT INTO Matricula

VALUES (2,0,3,1,'A','coste_mat');

INSERT INTO Matricula

VALUES (3,0,4,4,'C','coste_mat');

INSERT INTO Matricula

VALUES (4,0,5,2,'B','coste_mat');

INSERT INTO Matricula

VALUES (5,0,5,4,'C','coste_mat');

INSERT INTO Matricula

VALUES (6,0,6,4,'C','coste_mat');

INSERT INTO Matricula

```

VALUES (7,0,7,1,'A','coste_mat');

INSERT INTO Matricula

VALUES (8,0,8,2,'A','coste_mat');

INSERT INTO Matricula

VALUES (9,0,8,3,'B','coste_mat');

INSERT INTO Matricula

VALUES (10,0,8,4,'C','coste_mat');

INSERT INTO Matricula

VALUES (11,0,9,2,'A','coste_mat');

INSERT INTO Matricula

VALUES (12,0,9,2,'B','coste_mat');

INSERT INTO Matricula

VALUES (13,0,10,4,'C','coste_mat');

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('1/2/99','DD/MM/YY'),1,1,4);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('3/4/98','DD/MM/YY'),1,2,3);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('2/4/99','DD/MM/YY'),3,4,1);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('3/4/98','DD/MM/YY'),5,1,3);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('5/5/98','DD/MM/YY'),7,5,2);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('13/3/98','DD/MM/YY'),1,5,6);

INSERT INTO Prueba_Teorica

```

```

VALUES (TO_DATE('4/4/98','DD/MM/YY'),9,9,6);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('2/3/98','DD/MM/YY'),6,7,8);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('5/7/97','DD/MM/YY'),3,4,5);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('3/6/98','DD/MM/YY'),3,5,6);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('25/6/98','DD/MM/YY'),7,2,1);

INSERT INTO Pago

VALUES (1,1,1000);

INSERT INTO Pago

VALUES (2,1,20000);

INSERT INTO Pago

VALUES (1,3,10000);

INSERT INTO Pago

VALUES (1,4,10000);

INSERT INTO Pago

VALUES (2,4,5000);

INSERT INTO Pago

VALUES (1,5,1000);

INSERT INTO Pago

VALUES (1,6,1000);

INSERT INTO Clase_Practica

VALUES (TO_DATE('2/2/99','DD/MM/YY'),1,2,'rampas','mejorar');

INSERT INTO Consumo

```

```

VALUES ('000003',1,TO_DATE('2/2/99','DD/MM/YY'),10,1);

INSERT INTO Clase_Practica

VALUES (TO_DATE('2/2/99','DD/MM/YY'),2,1,'reduccion marchas',
'mejorar');

INSERT INTO Consumo

VALUES ('000005',2,TO_DATE('2/2/99','DD/MM/YY'),5,1);

INSERT INTO Clase_Practica

VALUES (TO_DATE('1/2/99','DD/MM/YY'),1,1,'rampas','mejorar');

INSERT INTO Consumo

VALUES ('000003',1,TO_DATE('1/2/99','DD/MM/YY'),25,3);

INSERT INTO Clase_Practica

VALUES (TO_DATE('3/3/99','DD/MM/YY'),2,3,'rampas','mejorar');

INSERT INTO Consumo

VALUES ('000005',2,TO_DATE('3/3/99','DD/MM/YY'),15,3);

INSERT INTO Clase_Practica

VALUES (TO_DATE('4/12/98','DD/MM/YY'),2,2,'aparcar','bien');

INSERT INTO Consumo

VALUES ('000005',2,TO_DATE('4/12/98','DD/MM/YY'),20,4);

INSERT INTO Clase_Practica

VALUES (TO_DATE('5/12/98','DD/MM/YY'),3,1,'reduccion','bien');

INSERT INTO Consumo

VALUES ('000004',3,TO_DATE('5/12/98','DD/MM/YY'),30,10);

INSERT INTO Clase_Practica

VALUES (TO_DATE('5/11/98','DD/MM/YY'),4,1,'aparcar','mejorar');

INSERT INTO Consumo

VALUES ('000001',4,TO_DATE('5/11/98','DD/MM/YY'),40,10);

```

```

INSERT INTO Clase_Practica
VALUES (TO_DATE('6/12/98','DD/MM/YY'),4,1,'espejos','bien');

INSERT INTO Consumo
VALUES ('000001',4,TO_DATE('6/12/98','DD/MM/YY'),30,8);

INSERT INTO Clase_Practica
VALUES (TO_DATE('6/12/98','DD/MM/YY'),5,2,'giros','bien');

INSERT INTO Consumo
VALUES ('000004',5,TO_DATE('6/12/98','DD/MM/YY'),15,2);

INSERT INTO Clase_Practica
VALUES (TO_DATE('7/7/99','DD/MM/YY'),6,1,'rampas','bien');

INSERT INTO Consumo
VALUES ('000004',6,TO_DATE('7/7/99','DD/MM/YY'),6,1);

INSERT INTO Clase_Practica
VALUES (TO_DATE('7/7/99','DD/MM/YY'),7,1,'señales','mejorar');

INSERT INTO Consumo
VALUES ('000005',7,TO_DATE('7/7/99','DD/MM/YY'),7,1);

INSERT INTO Clase_Practica
VALUES (TO_DATE('10/12/99','DD/MM/YY'),8,1,'señanles','mejorar');

INSERT INTO Consumo
VALUES ('000006',8,TO_DATE('10/12/99','DD/MM/YY'),8,1);

INSERT INTO Clase_Practica
VALUES (TO_DATE('11/12/99','DD/MM/YY'),8,2,'señanles','mejorar');

INSERT INTO Consumo
VALUES ('000006',8,TO_DATE('11/12/99','DD/MM/YY'),14,2);

INSERT INTO Clase_Practica
VALUES (TO_DATE('10/1/99','DD/MM/YY'),9,1,'señanles','mejorar');

```

```

INSERT INTO Consumo

VALUES ('000002',9,TO_DATE('10/1/99','DD/MM/YY'),9,1);

INSERT INTO Clase_Practica

VALUES (TO_DATE('10/2/99','DD/MM/YY'),10,3,'señales','mejorar');

INSERT INTO Consumo

VALUES ('000004',10,TO_DATE('10/2/99','DD/MM/YY'),10,3);

INSERT INTO Clase_Practica

VALUES (TO_DATE('11/2/99','DD/MM/YY'),10,2,'señales','mejorar');

INSERT INTO Consumo

VALUES ('000004',10,TO_DATE('11/2/99','DD/MM/YY'),20,2);

INSERT INTO Clase_Practica

VALUES (TO_DATE('10/12/99','DD/MM/YY'),11,1,'señales','mejorar');

INSERT INTO Consumo

VALUES ('000006',11,TO_DATE('10/12/99','DD/MM/YY'),11,1);

INSERT INTO Clase_Practica

VALUES (TO_DATE('10/12/99','DD/MM/YY'),12,2,'señales','mejorar');

INSERT INTO Consumo

VALUES ('000001',12,TO_DATE('10/12/99','DD/MM/YY'),12,2);

INSERT INTO Clase_Practica

VALUES (TO_DATE('10/12/99','DD/MM/YY'),13,3,'señales','mejorar');

INSERT INTO Consumo

VALUES ('000004',13,TO_DATE('10/12/99','DD/MM/YY'),13,3);

INSERT INTO Tramitacion_Documentos

VALUES (TO_DATE('2/2/99','DD/MM/YY'),2,'coste_tram');

INSERT INTO Examen

VALUES (TO_DATE('12/5/99','DD/MM/YY'),2,TO_DATE('2/2/99',

```

```

'DD/MM/YY'),'T',NULL,6);

INSERT INTO Examen

VALUES (TO_DATE('17/5/99','DD/MM/YY'),2,TO_DATE('2/2/99',
'DD/MM/YY'),'T',NULL,9);

INSERT INTO Tramitacion_Documentos

VALUES (TO_DATE('11/10/99','DD/MM/YY'),1,'coste_tram');

INSERT INTO Examen

VALUES (TO_DATE('1/6/99','DD/MM/YY'),1,TO_DATE('11/10/99',
'DD/MM/YY'),'T',NULL,0);

INSERT INTO Examen

VALUES (TO_DATE('3/6/99','DD/MM/YY'),1,TO_DATE('11/10/99',
'DD/MM/YY'),'P',1,NULL);

INSERT INTO Tramitacion_Documentos

VALUES (TO_DATE('19/5/99','DD/MM/YY'),6,'coste_tram');

INSERT INTO Examen

VALUES (TO_DATE('27/5/99','DD/MM/YY'),6,TO_DATE('19/5/99',
'DD/MM/YY'),'T',NULL,1);

INSERT INTO D_L

VALUES ('A','DNI');

INSERT INTO D_L

VALUES ('A','CERT MEDICO');

INSERT INTO D_L

VALUES ('B','LICENCIA A');

INSERT INTO D_L

VALUES ('B','DNI');

INSERT INTO D_L

```

```
VALUES ('B','CERT MEDICO');
```

```
INSERT INTO D_L
```

```
VALUES ('C','LICENCIA B');
```

```
INSERT INTO D_L
```

```
VALUES ('C','DNI');
```

```
INSERT INTO D_L
```

```
VALUES ('C','CERT MEDICO');
```

```
INSERT INTO D_L
```

```
VALUES ('C','CERT PSICOL');
```

8.7.4 Sentencias de consulta contra la base de datos

1) Obtener el alumno que mas errores comete en las pruebas teoricas, contando todos los errores de todos los test

Para ello primero tenemos que calcular para cada alumno todos los errores que comete en todos los test que realiza. Esa es la mision del select mas interior, del que lo unico que hay que comentar es que hacemos la reunion por el atributo n_matricula, haciendo entonces aparecer en el producto cartesiano todas las matriculas con sus pruebas teoricas. Como es obvio, saldrán varias tuplas con el dni_al iguales, y esta es la clave para realizar los grupos en funcion del dni_al y calcular la suma de errores. Vemos como calculamos el maximo del conjunto de alumnos. Despues (en el select mas exterior) volvemos a generar la misma tabla anterior, pero solo mostramos los grupos (clausula HAVING) que tienen una suma igual al maximo

```
select a.dni_al dni,a.nombre,a.apellido_1,a.apellido_2,sum(pt.errores)
```

```
from alumno a,matricula m,prueba_teorica pt
```

```
where a.dni_al = m.dni_al AND m.n_matricula = pt.n_matricula
```

```
group by a.dni_al,a.nombre,a.apellido_1,a.apellido_2
```

```
having sum(pt.errores) =
```

```
(select max(err_al.errores)
```

```
from (select m.dni_al,sum(pt.errores) errores
```

```
from matricula m,prueba_teorica pt
```

```
where pt.n_matricula = m.n_matricula
```

```
group by m.dni_al) err_al
```

```
);
```

y la salida según los datos introducidos en la base de datos es:

DNI NOMBRE APELLIDO_1 APELLIDO_2 SUM(PT.ERRORES)

2 Ana Belen Ruiz 13

2) Obtener el total de ingresos en la autoescuela desglosada por alumnos

La orden de consulta es sencilla. Necesitamos generar una tabla en la que para cada alumno, vengar eflejado cada pago que realiza, es decir, que por cada pago realizado por un alumno tengamos una tupla en esa tabla. Una vez realizado esto, calculamos la suma total para cada alumno agrupando por el identificador del alumno.

```
select a.dni_al DNI,a.nombre nombre,a.apellido_1 app1,  
  
a.apellido_2 app2 ,sum(p.importe) Ptas  
  
from alumno a,matricula m,pago p  
  
where a.dni_al = m.dni_al AND m.n_matricula = p.n_matricula  
  
group by a.dni_al,a.nombre,a.apellido_1,a.apellido_2;
```

Y la salida seria:

DNI NOMBRE APP1 APP2 PTAS

2 Ana Belen Ruiz 21000

4 Petra Garcia Obregon 10000

5 Pepa Martos Garcia 16000

6 Antonio Molina Mora 1000

3) Calcular lo que cada alumno adeuda a la autoescuela en concepto de clases practicas para todas las licencias de las que se ha matriculado, mostrando tambien los litros totales consumidos y los kilometros totales realizados. Como cada clase practica se realiza en un solo vehiculo, tendremos que calcular para cada alumno y para cada vehiculo que usa en las clases practicas de las licencias en que se ha matriculado, el importe que debe a la autoescuela.

La resolucion de esta consulta involucra a seis tablas: alumno,matricula, clase_practica, vehiculo, lujo y consumo. Para realizar el calculo, necesitamos averiguar cuanto debe un alumno por cada clase practica. Para ello tenemos que multiplicar el numero de horas realizadas por la suma de la tarifa base del vehiculo mas la tarifa adicional en base al lujo del mismo. Y una vez realizado esto, tenemos que sumar para cada alumno el coste de cada clase practica. La clausula WHERE se encarga de seleccionar las tuplas de tal manera que aparezcan tantas tuplas con el mismo dni y tipo de licencia como clases practicas haya realizado el alumno en ese tipo de licencia. La clausula GROUP BY se encarga de hacer los grupos en base al identificador del alumno, el nombre los apellidos y el tipo de licencia del que se ha matriculado.

```

select a.dni_al dni,a.nombre,a.apellido_1,a.apellido_2,v.tipo_lic licencia,sum(cp.duracion)
Horas,sum(c.kilometros) Km, sum(c.litros) Litros, sum(cp.duracion * (v.tarifa +
l.tarifa_adicional)) Ptas
from matricula m,clase_practica cp,vehiculo v,lujo l,consumo c,alumno a
where a.dni_al = m.dni_al AND m.n_matricula = cp.n_matricula AND cp.n_matricula =
c.n_matricula AND cp.fecha_practica = c.fecha_practica AND c.matricula =
v.matricula AND v.tipo_lujo = l.tipo_lujo
group by a.dni_al,a.nombre,a.apellido_1,a.apellido_2,v.tipo_lic;

```

Y la salida esperada es:

DNI NOMBRE APELLIDO_1 APELLIDO_2 LIC HORAS KM LITROS PTAS

```

2 Ana Belen Ruiz C 3 35 4 10500
3 Elisa Roca Perez A 6 40 8 34200
4 Petra Garcia Obregon C 1 30 10 5000
5 Pepa Martos Garcia B 2 70 18 3000
5 Pepa Martos Garcia C 2 15 2 10000
6 Antonio Molina Mora C 1 6 1 5000
7 Elisabeth Ruiz Romero A 1 7 1 5700
8 Juan Martinez Fontiveros A 3 22 3 3600
8 Juan Martinez Fontiveros B 1 9 1 2000
8 Juan Martinez Fontiveros C 5 30 5 25000
9 Susana Ruiz Pozo A 1 11 1 1200
9 Susana Ruiz Pozo B 2 12 2 3000
10 Pepe Barba Martinez C 3 13 3 15000

```

4) Suma desglosada de los litros, kilometros, horas de practicas de cada alumno en cada vehiculo y con cada profesor en la autoescuela.

Simplemente observamos que un vehiculo esta asociado a una docencia, y una o mas de estas a un profesor. Organizamos los grupos primero en base al alumno, despues en base al profesor y por ultimo en base al vehiculo, ya que los consumos estan asociados a los vehiculos. Aunque habra que tener en cuenta las clases practicas porque ahi viene reflejada la duracion de cada practica. Veamos

```
select m.dni_al,m.tipo_lic,m.dni_prof,c.matricula,sum(cp.duracion) horas,
sum(c.kilometros) Km, sum(c.litros) l
from matricula m,clase_practica cp,consumo c
where m.n_matricula = cp.n_matricula AND cp.n_matricula = c.n_matricula AND
cp.fecha_practica = c.fecha_practica
group by m.dni_al,m.tipo_lic,m.dni_prof,c.matricula;
```

Y la salida debe ser:

DNI_AL TIP DNI_PROF MATRIC HORAS KM L

```
-----
2 C 3 000003 3 35 4
3 A 1 000005 6 40 8
4 C 4 000004 1 30 10
5 B 2 000001 2 70 18
5 C 4 000004 2 15 2
6 C 4 000004 1 6 1
7 A 1 000005 1 7 1
8 A 2 000006 3 22 3
8 B 3 000002 1 9 1
8 C 4 000004 5 30 5
9 A 2 000006 1 11 1
9 B 2 000001 2 12 2
10 C 4 000004 3 13 3
```

5) Encontrar Los alumnos que superan la media de litros consumidos en las practicas. Mostrar ademas la media de litros gastados.

Para esta consulta necesitamos saber los litros que cada alumno ha gastado en el total de sus clases practicas para cualquier licencia de la que se ha matriculado, y conocer la media de litros gastados en todas las clases practicas. Para conocer el primer conjunto de datos, usamos el primer select anidado dentro de la clausula from que hemos etiquetado con el nombre litros_alumno. Como puede observarse, esta sentencia agrupa todos los gastos de un alumno para cualquier licencia de la que tiene registradas practicas, y calcula el gasto total de litros. Para conocer la media de litros gastados en las clases practicas, como sabemos que por cada clase practica hay uno y solo un consumo, simplemente calculamos la media en base a la tabla consumo. El segundo select de la clausula from realiza este calculo. Lo hemos etiquetado como c para poder referenciarlo en el select superior y asi poder mostrar la media en todos los alumnos. Por ultimo, comentar que en el select mas exterior se ven involucradas las tablas alumno y las dos anteriormete generadas. La clausula WHERE de este select mas externo, se encarga de relacionar a cada alumno con su gasto en litros y ademas de mostrar solo los alumnos que han superado la media de litros gastados

```
select a.nombre,a.apellido_1,a.apellido_2,a.dni_al,litros_alumno.suma,c.media
```

```
from alumno a,
```

```
(select a.dni_al,sum(c.litros) suma
```

```
from alumno a,matricula m,clase_practica cp,consumo c
```

```
where a.dni_al = m.dni_al AND m.n_matricula = cp.n_matricula AND
```

```
cp.n_matricula = c.n_matricula AND cp.fecha_practica =c.fecha_practica
```

```
group by a.dni_al) litros_alumno,
```

```
(select avg(consumo. litros) media from consumo) c
```

```
where litros_alumno.suma >c.media AND litros_alumno.dni_al = a.dni_al;
```

Y la salida sera:

```
NOMBRE APELLIDO_1 APELLIDO_2 DNI_AL SUMA MEDIA
```

```
-----
```

```
Ana Belen Ruiz 2 4 3,1052632
```

```
Elisa Roca Perez 3 8 3,1052632
```

```
Petra Garcia Obregon 4 10 3,1052632
```

```
Pepa Martos Garcia 5 20 3,1052632
```

```
Juan Martinez Fontiveros 8 9 3,1052632
```

8.7.5 Sentencias de modificacion de la intencion de la base de datos

1)Hacer que sea posible introducir una duracion por clase practica de mas de 10 horas

Simplemente, deshabilitamos la restriccion que nos impide introducir una clase practica de mas de 10 horas.

ALTER TABLE clase_practica

DISABLE CONSTRAINT ck_duracion;

2)Deshabilitar la restriccion de clave foranea en tramitacion de documentos a la tabla coste_tramitacion para hacer un cambio en la extension de esta ultima

ALTER TABLE tramitacion_documentos

DISABLE CONSTRAINT fk_coste_tram;

3) Añadir un atributo codigo_postal a la entidad alumno, de tipo NUMBER

ALTER TABLE alumno

ADD (codigo_postal **NUMBER**(10));

3)Obligar a que se facilite una direccion para cada alumno

ALTER TABLE alumno

ADD CONSTRAINT ck_direccion

CHECK (direccion <> **NULL**);

8.7.6 Sentencias de modificacion de la extension de la base de datos

1) El alumno con dni igual a 1 se ha mudado a la direccion 'Puerta Real 7' y con telefono 867043

UPDATE alumno **SET**

direccion = 'Puerta Real 7' ,

telefono = 867043

WHERE dni_al = 1;

2)Cambiar el coste de matriculacion a 3500 Ptas

UPDATE coste_matriculacion **SET**

importe = 3500

WHERE coste_mat = 'coste_mat';

3)Subir un 30% el lujo EXCLUSIVO y EJECUTIVO de los vehiculos

UPDATE lujo **SET**

tarifa_adicional = tarifa_adicional + tarifa_adicional * 0.30

WHERE tipo_lujo = 'ejecutivo' **OR** tipo_lujo = 'exclusivo';

4)Reducir la tarifa base un 40% de los vehiculos que superan en la media el numero total de horas de practicas que se han realizado en la autoescuela

```
UPDATE vehiculo SET
tarifa = tarifa - tarifa*0.30
WHERE matricula IN ( SELECT seleccionados.matricula
FROM (SELECT v.matricula, SUM(cp.duracion)
FROM vehiculo v,consumo c,clase_practica cp
WHERE v.matricula = c.matricula AND
c.fecha_practica = cp.fecha_practica
AND c.n_matricula = cp.n_matricula
GROUP BY v.matricula
HAVING SUM(cp.duracion) > (SELECT AVG(duracion)
FROM clase_practica
)
) seleccionados
);
```

5)Reducir un 2% la duracion de cada clase practica del alumno que mas horas haya realizado en la autoescuela para la licencia 'B'

```
UPDATE clase_practica SET
duracion = duracion - duracion*0.02
WHERE n_matricula IN
(SELECT maximos.n_matricula
FROM (SELECT cp.n_matricula, SUM(cp.duracion)
FROM matricula m,clase_practica cp
WHERE m.n_matricula = cp.n_matricula AND m.tipo_lic = 'B'
GROUP BY cp.n_matricula
HAVING SUM(cp.duracion) = (SELECT MAX(sumas.recuento)
```

```

FROM (SELECT cp.n_matricula,
SUM(cp.duracion) recuento
FROM matricula m,clase_practica cp
WHERE m.n_matricula = cp.n_matricula
AND m.tipo_lic = 'B'
GROUP BY cp.n_matricula) sumas
)
) maximos
);

```

Quizás esta sentencia UPDATE si necesite comentario. Basicamente, lo que buscamos es el numero de la matricula del alumno que mas horas ha dado clase en la licencia `B`. La sentencia SELECT mas interna calcula la suma de horas para cada matricula que se ha realizado en la licencia B. La sentencia SELECT que envuelve a esta ultima, calcula el máximo de esas horas. El siguiente SELECT exterior, devuelve una tupla con dos atributos que son el numero de matricula y el total de horas de clase. Esta tupla corresponde al maximo. Y por ultimo, el SELECT mas exterior que sirve para realizar una proyeccion y devolver sólo el numero de matricula correspondiente al maximo de horas.

6)Aumentar un 10% los consumos realizados por el alumno con dni igual a 8 en todas las licencias en las que se haya matriculado

```

UPDATE consumo SET
litros = litros + litros*0.1,
kilometros = kilometros + kilometros*0.1
WHERE n_matricula IN(
SELECT m.n_matricula
FROM matricula m
WHERE m.dni_al = 8
);

```

7)Eliminar todas las matriculas del alumno con dni = 1 por defuncion del mismo

```

DELETE FROM matricula
WHERE matricula.dni_al = 8;

```

8)Borrar todas las clases practicas realizadas entre el 6/12/98 y el 7/7/99

DELETE FROM clase_practica

WHERE fecha_practica **BETWEEN** TO_DATE('06/12/98','DD/MM/YY') **AND**

TO_DATE('07/07/99','DD/MM/YY');

9)Borrar todos los pagos del alumno con dni = 1 en todas las matriculas que haya realizado

DELETE FROM pago

WHERE n_matricula **IN** (

SELECT m.n_matricula

FROM matricula m

WHERE m.dni_al = 1

);

10) Borrar todas las tramitaciones de documentos del alumno con dni = 6 y que ademas hayan sido realizadas del 18/5/99 hasta el dia de hoy, ademas de toda la informacion referente a los exámenes que hayan sido realizados por el alumno usando las convocatorias a las que da derecho esas tramitaciones de documentos.

Esto ultimo se realiza automaticamente debido a la clausula ON DELETE CASCADE de SQL en la tabla EXAMEN

DELETE FROM tramitacion_documentos

WHERE n_matricula **IN** (

SELECT DISTINCT m.n_matricula

FROM matricula m,tramitacion_documentos td

WHERE m.dni_al = 6 **AND** m.n_matricula = td.n_matricula

)

AND fecha_tramitacion **BETWEEN** TO_DATE('15/05/99','DD/MM/YY') **AND** SYSDATE;

11)Eliminar el coche que mas kilometros ha recorrido de los que tiene se tiene en la autoescuela impartiendo licencias del tipo B.

Para poder hacer esto hay que deshabilitar la condicion de clave foranea en la tabla DOCENCIA a VEHICULO, para que nos deje borrar esas tuplas. Evidentemente, el esquema queda inconsistente pero se supone que esta es una orden que ira seguida de una insercion de otro vehiculo que reemplace al que la sentencia DELETE elimine

ALTER TABLE Docencia

```

DISABLE CONSTRAINT fk_vehiculo;

DELETE FROM vehiculo

WHERE matricula IN

( SELECT maximo_kilometros.matricula

FROM(

SELECT v.matricula,SUM(c.kilometros)

FROM vehiculo v,consumo c

WHERE v.tipo_lic = 'B' AND v.matricula = c.matricula

GROUP BY v.matricula

HAVING SUM(c.kilometros) =

(

SELECT max(km.recuento)

FROM (SELECT v.matricula,SUM(c.kilometros)

recuento

FROM vehiculo v,consumo c

WHERE v.tipo_lic = 'B' AND

v.matricula = c.matricula

GROUP BY v.matricula) km

)

) maximo_kilometros

);

```

12) Borrar todas las matriculas de los alumnos que alguna vez hayan dado una practica en el coche con matricula = 000002

```

DELETE FROM matricula

WHERE m.n_matricula IN (

SELECT DISTINCT m.n_matricula

FROM matricula m,clase_practica cp,consumo c

```

```

WHERE m.n_matricula = cp.n_matricula AND cp.n_matricula =
c.n_matricula AND cp.fecha_practica = c.fecha_practica AND
c.matricula ='000001'
);

```

8.7.6 PROCEDIMIENTOS PL/SQL

1) Crear un procedimiento en PL/SQL que presente por pantalla el dni del alumno que ha gastado mas de un tanto por ciento del total del combustible, tanto por ciento que se introducirá como parámetro en el procedimiento.

Este procedimiento realiza un trabajo que puede realizarse con una orden SELECT compleja, aunque siempre perderemos la posibilidad de parametrización que nos permite un procedimiento. La complejidad reside en calcular para cada alumno, cuanto combustible gasta en total en cualquier licencia de la que se haya matriculado y haya dado al menos una clase practica. La agrupación que necesitamos realizar para calcular el total de cada uno de los alumnos la realizamos procesando el cursor **'al_litros'** que como vemos devuelve por cada clase practica de un alumno, el numero de litros consumidos en esa practica.

Como podemos ver más adelante, al procesar el cursor usamos una tabla con una estructura igual que la que devuelve el cursor, y aprovechamos el índice de la tabla para agrupar y sumar los litros de cada alumno. Al final, devolvemos el dni del alumno y su suma de litros consumidos.

La única excepción que se considera es la de que no haya consumos almacenados en la base de datos, con lo que no hay clases practicas.

```

CREATE OR REPLACE PROCEDURE gasto_alumno(gasto IN NUMBER) AS

```

```

/* zona de declaracion de variables */

```

```

/* este cursor contiene el conjunto de los gastos en litros para cada alumno*/

```

```

CURSOR al_litros IS

```

```

SELECT m.dni_al,c.litros

```

```

FROM matricula m,clase_practica cp, consumo c

```

```

WHERE m.n_matricula = cp.n_matricula AND

```

```

cp.n_matricula = c.n_matricula AND

```

```

cp.fecha_practica = c.fecha_practica;

```

```

/* esta tabla la usamos para recoger los resultados y agrupar los

```

```

litros gastados por cada alumno */

```

```

TYPE t_litros IS TABLE OF al_litros%ROWTYPE INDEX BY BINARY_INTEGER;

```

```

t_al_litros t_litros;

/* Esta variable recogerá el total de litros de todos los consumos */

v_total_litros NUMBER (10) := 0;

/* variable auxiliar que recoge las extracciones del cursor */

auxiliar al_litros%ROWTYPE;

encontrado BOOLEAN := FALSE;

/* esta excepcion captura el error que se produce cuando no hay
gastos de combustible en la base de datos */

no_hay_gastos EXCEPTION;

BEGIN

/* Calculamos el total */

SELECT sum(c.litros)

INTO v_total_litros

FROM consumo c ;

IF SQL%NOTFOUND THEN

RAISE no_hay_gastos;

END IF;

/* realizamos el calculo del tanto por ciento correspondiente */

v_total_litros := v_total_litros * gasto;

OPEN al_litros;

/* mientras en la iteracion anterior se saco una fila */

FETCH al_litros INTO auxiliar;

WHILE al_litros%FOUND AND NOT (al_litros%FOUND IS NULL) LOOP

IF t_al_litros.EXISTS(auxiliar.dni_al) THEN

t_al_litros(auxiliar.dni_al).litros := t_al_litros(auxiliar.dni_al).litros + auxiliar.litros;

ELSE

```

```

t_al_litros(auxiliar.dni_al) := auxiliar;

END IF;

FETCH al_litros INTO auxiliar;

END LOOP;

CLOSE al_litros;

/* ya tenemos el calculo en la variable de tabla t_al_litros */

/* mostramos en pantalla los alumnos que superan el

tanto por ciento indicado en el parametro formal */

FOR v_contador IN t_al_litros.FIRST .. t_al_litros.LAST LOOP

IF t_al_litros(v_contador).litros > v_total_litros THEN

DBMS_OUTPUT.PUT_LINE ( 'El alumno con dni' || t_al_litros(v_contador).dni_al

|| 'ha superado el gasto indicado de'

|| v_total_litros);

END IF;

END LOOP;

/* Zona de EXCEPCIONES */

EXCEPTION

WHEN no_hay_gastos THEN

DBMS_OUTPUT.PUT_LINE('No hay gastos de combustible almacenados');

END gasto_alumno;

```

2) *Escribir un procedimiento que realice la matriculacion de un alumno en una licencia concreta.*

Concretemos un poco más. El procedimiento debe ocuparse de:

- Si no existe el alumno, crear la entrada en alumno
- Si ya existe, debe intentar matricular al alumno en la licencia que se desea. Habrá que controlar también si vamos a matricularlo en una licencia de la que ya está matriculado
- Deberá asociar la matrícula a la docencia que menos alumnos tenga matriculados. Para esto declararemos un subprocedimiento, llamado hallar_docencia que devuelve la docencia que cumple los requisitos anteriores
- Deberá gestionar los errores que puedan aparecer:

- Que el alumno ya este matriculado de esa licencia
- Si el alumno no existe en la autoescuela, comprobar que los datos del alumno, en especial la licencia anterior, son correctos
- Comprobar que la licencia de la que queremos matricular existe en la base de datos y tiene al menos un profesor asociado.

```

CREATE OR REPLACE PROCEDURE crearalumno( dni IN alumno.dni_al%TYPE,

nombre IN alumno.nombre%TYPE,

apellido_1 IN alumno.apellido_1%TYPE,

apellido_2 IN alumno.apellido_2%TYPE,

direccion IN alumno.direccion%TYPE,

telefono IN alumno.telefono%TYPE,

licencia_anterior IN alumno.tipo_lic%TYPE,

licencia_matriculacion IN

matricula.tipo_lic%TYPE

) IS

numero_matricula BINARY_INTEGER:=0; /*recogera el numero de matricula mas alto en la tabla

matricula */

v_docencia docencia%ROWTYPE; /*recoge a la docencia seleccionada*/

al_prev_mat EXCEPTION; /* captura el error de que ya haya un alumno igual en la base de

datos*/

lic_inex EXCEPTION; /* captura el error de solicitar una licencia que no existe */

lic_ant_inex EXCEPTION; /* la licencia anterior introducida no es valida */

basura VARCHAR(50);

CURSOR c_alumno (dni_alumno alumno.dni_al%TYPE) IS

SELECT a.dni_al

FROM alumno a

WHERE a.dni_al = dni_alumno; /*este ultimo es un parametro de entrada */

CURSOR c_licencia (tipo_licencia licencia.tipo_lic%TYPE) IS

```

```

SELECT l.tipo_lic

FROM licencia l

WHERE l.tipo_lic = tipo_licencia;

CURSOR c_max_matricula IS

SELECT max(m.n_matricula)

FROM matricula m;

CURSOR c_matricula_alumno (dni_alumno alumno.dni_al%TYPE,

licencia_matriculacion licencia.tipo_lic%TYPE) IS

SELECT m.tipo_lic

FROM matricula m

WHERE m.dni_al = dni AND /* buscamos una matricula asociada al alumno */

m.tipo_lic = licencia_matriculacion; /* y en la licencia que queremos matricularlo */

/* INICIO DE LA DECLARACION DEL SUBPROCEDIMIENTO CALCULAR_DOCENCIA*/

FUNCTION calcular_docencia(licencia_seleccionada IN matricula.tipo_lic%TYPE,

docencia_seleccionada OUT docencia%ROWTYPE

) RETURN BOOLEAN IS

/* Este procedimiento devuelve la docencia que tiene menos matriculados en la licencia

que se especifica en el argumento de la funcion*/

/* DECLARACION DE VARIABLES*/

/* Este cursor apunta a una sentencia select que devuelve el numero de matriculados

de todas las docencias que den clase de la licencia especificada en el argumento

de la funcion */

CURSOR c_docencia (lm matricula.tipo_lic%TYPE) IS

SELECT d.dni_prof,d.tipo_lic,d.matricula,count(m.n_matricula) recuento

FROM docencia d,matricula m

WHERE d.dni_prof = m.dni_prof AND

```

```

m.tipo_lic = d.tipo_lic AND

d.tipo_lic = lm

GROUP BY d.dni_prof,d.tipo_lic,d.matricula;

/* Esta variable contendra el minimo obtenido*/

minimo_actual c_docencia%ROWTYPE;

/* Esta variable sirve para recoger las tuplas extraidas del cursor */

auxiliar c_docencia%ROWTYPE;

/* INICIO*/

BEGIN

/* vamos a comprobar primero si el conjunto activo del cursor
tiene al menos una tupla. En caso de que no, esto quiere decir
que se introdujo una licencia de la que no se da clase en
la autoescuela */

minimo_actual.recuento := 1000;

OPEN c_docencia(licencia_seleccionada);

FETCH c_docencia INTO auxiliar;

IF c_docencia%NOTFOUND THEN /*si se realizo una extraccion y no hubo tupla
extraida */

CLOSE c_docencia;

docencia_seleccionada.dni_prof := NULL;

docencia_seleccionada.tipo_lic := NULL;

docencia_seleccionada.matricula := NULL;

RETURN FALSE; /* devolvemos FALSE como respuesta */

END IF;

/* Como en este caso hay tuplas en el cursor */

CLOSE c_docencia;

```

```

OPEN c_docencia(licencia_seleccionada);

LOOP

FETCH c_docencia INTO auxiliar;

EXIT WHEN c_docencia%NOTFOUND;

/* si no encontramos datos nos salimos */

IF minimo_actual.recuento >= auxiliar.recuento THEN

/* si encontramos alguno menor */

minimo_actual := auxiliar;

/* actualizamos el minimo */

END IF;

END LOOP;

CLOSE c_docencia;

docencia_seleccionada.dni_prof := minimo_actual.dni_prof;

docencia_seleccionada.tipo_lic := minimo_actual.tipo_lic;

docencia_seleccionada.matricula := minimo_actual.matricula;

RETURN TRUE; /*hemos encontrado el minimo. La funcion ha tenido exito */

END calcular_docencia;

/* FIN DE LA DEFINICION DEL SUBPROCEDIMIENTO CALCULAR_DOCENCIA*/

/* Inicio procedimiento principal */

BEGIN

/* primero vamos a comprobar que el alumno no esta ya registrado en la base de datos*/

OPEN c_alumno(dni);

FETCH c_alumno INTO basura;

IF c_alumno%NOTFOUND THEN /* Usamos el cursor para detectar si hubo filas */

/* Estamos en el caso de que el alumno no existe */

/* calculemos ahora la docencia que se encargara de sus clases */

```

```

IF NOT calcular_docencia(licencia_matriculacion,v_docencia) THEN

/* Si la licencia no existe */

RAISE lic_inex;

ELSE

/* Si la licencia existe */

/*ahora tenemos que saber si tiene una licencia anterior y cual es esta*/

IF licencia_anterior IS NULL THEN /*no hay licencia anterior*/

/* luego tenemos que asociarle el coste de enseñanza teorica */

INSERT INTO alumno VALUES

(dni,nombre,apellido_1,apellido_2,direccion,telefono,NULL,'coste_e_t');

ELSE

/* vamos a ver si la licencia anterior es valida */

OPEN c_licencia(licencia_anterior);

FETCH c_licencia INTO basura;

IF c_licencia%NOTFOUND THEN

/* la licencia anterior no existe */

CLOSE c_licencia;

RAISE lic_ant_inex;

ELSE

/* la licencia anterior existe */

/* como tiene una licencia anterior,se la asociamos */

INSERT INTO alumno VALUES

(dni,nombre,apellido_1,apellido_2,direccion,telefono,licencia_anterior,NULL);

CLOSE c_licencia;

END IF;

END IF;

```

```

/* Una vez resulelto esto hay que realizar una entrada en la tabla matricula */

/* Vamos a calcular el numero mas alto del atributo n_matricula para asociarlo a la nueva
matricula */

OPEN c_max_matricula;

FETCH c_max_matricula INTO numero_matricula;

IF c_max_matricula%NOTFOUND THEN

/* no hay matriculas */

numero_matricula := 1;

CLOSE c_max_matricula;

ELSE

/* incrementamos el numero de matricula */

numero_matricula := numero_matricula +1;

/* creamos una ocurrencia en la tabla matricula para matricular al alumno */

INSERT INTO matricula VALUES

(numero_matricula,0,dni,v_docencia.dni_prof,

v_docencia.tipo_lic,'coste_mat');

CLOSE c_max_matricula;

END IF;

END IF;

CLOSE c_alumno;

ELSE

/* Estamos en el caso de que el alumno ya esta matriculado en alguna licencia*/

/* Habra que ver si ya esta matriculado en la licencia de la que quiere matricularse */

OPEN c_matricula_alumno(dni,licencia_matriculacion);

FETCH c_matricula_alumno INTO basura;

IF c_matricula_alumno%NOTFOUND THEN /* si no encontramos respuesta, es que no esta

```

```

matriculado */

/* en la licencia de la que ahora vamos a matricularle */

IF NOT calcular_docencia(licencia_matriculacion,v_docencia) THEN

dbms_output.put_line( v_docencia.dni_prof || v_docencia.tipo_lic);

/* no existe la licencia de la que vamos a matricularlo */

RAISE lic_inex;

ELSE

/* si existe*/

/* procedemos a la matriculacion */

OPEN c_max_matricula;

FETCH c_max_matricula INTO numero_matricula;

IF c_max_matricula%NOTFOUND THEN

numero_matricula :=1;

ELSE

numero_matricula:= numero_matricula +1;

END IF;

/* matriculamos */

INSERT INTO matricula VALUES

(numero_matricula,0,dni,v_docencia.dni_prof,v_docencia.tipo_lic,'coste_mat');

END IF;

ELSE

/* Como la sentencia select ha encontrado al alumno matriculado en la licencia de la

que queriamos matricularle, levantaremos una excepcion */

RAISE al_prev_mat;

END IF;

CLOSE c_matricula_alumno;

```

END IF;

EXCEPTION

WHEN lic_inex **THEN**

DBMS_OUTPUT.PUT_LINE('La licencia introducida no tiene profesor asociado o no existe');

WHEN al_prev_mat **THEN**

DBMS_OUTPUT.PUT_LINE('el alumno ya esta matriculado de esa licencia');

WHEN lic_ant_inex **THEN**

DBMS_OUTPUT.PUT_LINE('La licencia anterior es inválida');

END crearalumno;

3) Realizar un procedimiento que se encargue de realizar las inserciones correctamente.

Como indicamos en el apartado del modelado relacional, la entidad tramitacion_documentos y la entidad examen mantenian una relacion 1:3. Este procedimiento nos ayudará a mantener esta restriccion, impidiendo la asociacion de más de tres exámenes a una misma tramitacion de documentos. Concretando un poco, el procedimiento deberá realizar:

- Comprobar si existe una matricula asociada con el alumno concreto y en la licencia concreta
- Comprobar si existe una tramitacion de documentos asociada a esa matricula
- De las tramitaciones de documentos disponibles, escoger la que tenga una fecha mas reciente y comprobar si tiene o no 3 exámenes asociados. Para esto usaremos una subfuncion, hallar_tramitacion, que en realidad es la que se lleva la mayoria del procesamiento del procedimiento. Retornará TRUE si todo va bien junto con la tramitacion elegida, y en caso de cualquier problema devolverá FALSE.
- Realizar la insercion del examen si todo va bien. En caso contrario, devolver el error al usuario

CREATE OR REPLACE PROCEDURE inserta_examen

(dni_alumno IN alumno.dni_al%TYPE,

licencia_examen IN licencia.tipo_lic%TYPE,

fecha_examen IN examen.fecha_examen%TYPE,

tipo_ex IN examen.tipo_examen%TYPE,

err_teorico IN examen.errores%TYPE,

dur_practico IN examen.duracion%TYPE) AS

t_a tramitacion_documentos%ROWTYPE;

no_hallada EXCEPTION;

/* Este subprocedimiento busca una tramitacion de documentos para un alumno en una licencia concreta, devolviendo FALSE si no existe una tramitacion o si existe y ademas tiene 3 examenes asociados, con lo que no podemos asociar un nuevo examen. En caso de que exista una tramitacion adecuada, se devuelve en el parametro de salida tram_correcta */

FUNCTION hallar_tramitacion(

dni IN alumno.dni_al%TYPE,

licencia_examen IN licencia.tipo_lic%TYPE,

tram_correcta OUT tramitacion_documentos %ROWTYPE

) **RETURN BOOLEAN IS**

/* Este cursor continene una tabla en la que se muestra para un alumno y en una licencia concreta, y para cada tramitacion de documentos que realiza el numero de examenes asociado. De esta manera recorriendo el cursor, podemos hallar la tramitacion mas actual recorriendo el cursor y comprobar si ya tiene agotadas las tres convocatorias pertinentes */

CURSOR tramitacion (dni1 alumno.dni_al%TYPE,licencia_examen1 licencia.tipo_lic%TYPE)

IS

SELECT m.dni_al,m.n_matricula,td.fecha_tramitacion,count(e.fecha_examen)

recuento

FROM matricula m,tramitacion_documentos td,examen e

WHERE m.dni_al = dni1 AND m.tipo_lic = licencia_examen1 AND

m.n_matricula = td.n_matricula AND

e.fecha_tramitacion = td.fecha_tramitacion AND

e.n_matricula = td.n_matricula

GROUP BY m.dni_al,m.n_matricula,td.fecha_tramitacion;

```

/* Aqui almacenamos las tuplas que leemos del cursor superior */

CURSOR tram_examen (dni1 alumno.dni_al%TYPE,licencia_examen1
licencia.tipo_lic%TYPE) IS

SELECT max(td.fecha_tramitacion) fecha

FROM tramitacion_documentos td,matricula m

WHERE m.dni_al = dni1 AND m.tipo_lic = licencia_examen1 AND
m.n_matricula = td.n_matricula;

r_tram_examen tram_examen%ROWTYPE;

r_tramitacion tramitacion%ROWTYPE;

tramitacion_tardia tramitacion%ROWTYPE;

BEGIN

OPEN tramitacion(dni,licencia_examen);

FETCH tramitacion INTO tramitacion_tardia;

IF tramitacion%NOTFOUND THEN

/* El cursor esta vacio. no hay tramitaciones asociadas con el alumno
en esa matricula*/

CLOSE tramitacion;

RETURN FALSE;

END IF;

/* llegados a este punto si que existen tramitaciones de documentos */

CLOSE tramitacion;

OPEN tramitacion (dni,licencia_examen);

LOOP

FETCH tramitacion INTO r_tramitacion;

IF tramitacion%NOTFOUND THEN

CLOSE tramitacion;

```

```

EXIT;

END IF;

/* vamos a procesar el cursor */

/* vamos a calcular la tramitacion mas tardia */

IF tramitacion_tardia.fecha_tramitacion <= r_tramitacion.fecha_tramitacion THEN

tramitacion_tardia := r_tramitacion;

END IF;

END LOOP;

/* ahora tenemos en tramitacion_tardia la tramitacion mas actual

ahora lo que tenemos que hacer es ver si la tramitacion tiene

asociadas tres exámenes, con lo que no podría ser considerado un

nuevo examen */

OPEN tram_examen (dni,licencia_examen);

FETCH tram_examen INTO r_tram_examen;

CLOSE tram_examen;

IF tramitacion_tardia.fecha_tramitacion = r_tram_examen.fecha THEN

IF tramitacion_tardia.recuento < 3 THEN

/* Pueden asociarse mas exámenes */

tram_correcta.n_matricula := tramitacion_tardia.n_matricula;

tram_correcta.fecha_tramitacion := tramitacion_tardia.fecha_tramitacion;

tram_correcta.coste_tram := 'coste_tram';

ELSE

/* No pueden asociarse mas exámenes */

RETURN FALSE;

END IF;

ELSE

```

```

tram_correcta.fecha_tramitacion := r_tram_examen.fecha;

tram_correcta.n_matricula := tramitacion_tardia.n_matricula;

tram_correcta.coste_tram := 'coste_tram';

END IF;

RETURN TRUE;

END hallar_tramitacion ;

/* COMENZAMOS LA SECCION EJECUTABLE DEL BLOQUE */

BEGIN

/* Buscamos una tramitacion */

IF hallar_tramitacion(dni_alumno,licencia_examen,t_a) THEN

/* hay una tramitacion en la que podemos hacer un nuevo examen */

IF tipo_ex = 'T' THEN

/*es teorico*/

INSERT INTO examen VALUES

( fecha_examen,t_a.n_matricula,t_a.fecha_tramitacion,'T',NULL,err_teorico);

ELSE

/*es practico*/

INSERT INTO examen VALUES

( fecha_examen,t_a.n_matricula,t_a.fecha_tramitacion,'P',dur_practico,NULL);

END IF;

ELSE

/* No se hallo la tramitacion asociada */

RAISE no_hallada;

END IF;

/* SECCION DE EXCEPCIONES */

EXCEPTION

```

WHEN no_hallada **THEN**

DBMS_OUTPUT.PUT_LINE(' No se encontro una tramitacion con menos de 3 examenes ');

END inserta_examen;

2 @iscbd GESTIÓN DE AUTOESCULAS

GESTIÓN DE AUTOESCUELAS @iscbd 3

