



NOMBRE ALUMNO:

EJERCICIO 1: modelo conceptual

El fabricante de vehículos eléctricos TESMA necesita diseñar una base de datos para almacenar información relativa a la venta de sus vehículos.

Los clientes de TESMA deben hacer una reserva del vehículo que quieren adquirir. Se quiere registrar la fecha en que cada cliente reserva cada vehículo. Así, un cliente puede reservar un determinado vehículo un día concreto. El mismo cliente puede reservar diversos vehículos a lo largo del tiempo (en fechas diferentes). Un vehículo solo puede ser reservado una vez y por un único cliente. De cada reserva se quiere saber el importe final y el depósito que ha dejado el cliente de paga y señal.

De cada cliente se quiere guardar su nombre, la fecha de nacimiento y un código numérico que lo identifica.

De cada vehículo se quiere almacenar:

- Un código, que lo identifica.
- El modelo.
- De qué color es.
- La matrícula.
- El concesionario donde se entregará al cliente.

Cada modelo de vehículo tiene un nombre, el año de salida al mercado y la autonomía (kilómetros que puede circular con la carga completa). Un modelo se identifica por el nombre y el año de salida.

Cada vehículo es de un solo color. Los colores tienen un código que los identifica y un nombre. Puede haber diversos vehículos del mismo color.

Una vez el vehículo está a punto para ser entregado al cliente se matricula. Esta matrícula se identifica por un código alfanumérico que es único en un país determinado, pero que se puede repetir en diferentes países. De cada país se quiere guardar la abreviatura, que lo identifica, y el nombre.



NOMBRE ALUMNO:

Cada vehículo se entrega en un único concesionario y en cada concesionario se entregan diversos vehículos. Se quiere guardar la fecha en que cada vehículo llega al concesionario (final de fabricación) y la fecha en que el cliente lo recoge. Los concesionarios tienen un nombre y un código que los identifica. No tendremos en la base de datos concesionarios que no hayan entregado como mínimo un vehículo.

Los modelos de vehículos pueden ser estándar (no se pueden escoger las características) o personalizados. En el caso de los modelos estándar se quiere guardar su potencia. En el caso de los modelos personalizados se quiere guardar qué opciones ha escogido el cliente. Un modelo personalizado puede tener una o más opciones y cada opción se puede escoger en diversos modelos personalizados.

Cada opción tiene una descripción, un valor, el precio adicional que cuesta y un código numérico identificador. Algunos ejemplos de opciones serían:

- Descripción: asientos calefactados. Valor: Sí o No. Precio: 250€. Código: 1
- Descripción: más autonomía. Valor: 200 km extra. Precio: 1500€. Código: 2

Cada modelo personalizado de vehículo se monta en una fábrica determinada. Cada fábrica monta diferentes modelos personalizados.

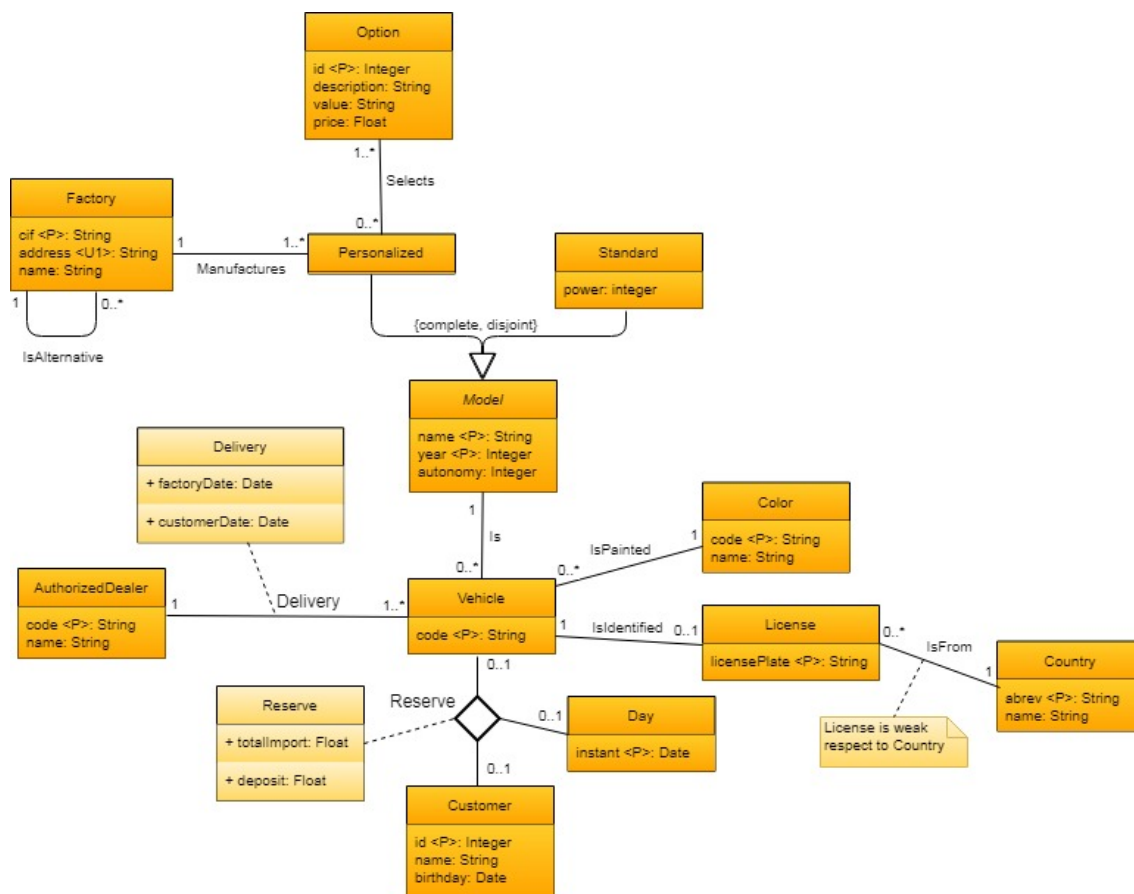
Las fábricas tienen un nombre, un CIF (que los identifica) y una dirección (que es única). Cada fábrica tiene otra fábrica alternativa para cuando cierra por vacaciones. Una fábrica puede ser alternativa de otras fábricas, o no.

Se solicita: realizar el diseño conceptual, mediante un diagrama de clases UML, que recoja toda la semántica del enunciado, utilizando una herramienta de modelado (Draw.io o similar). Es necesario indicar los atributos de los tipos de entidades con los tipos de datos correspondientes, las cardinalidades de los tipos de relaciones, todas las restricciones (claves primarias, alternativas...), así como los requisitos que no han quedado reflejados en el esquema propuesto. Si se ha realizado alguna suposición semántica adicional también se debe indicar.



NOMBRE ALUMNO:

SOLUCIÓN:

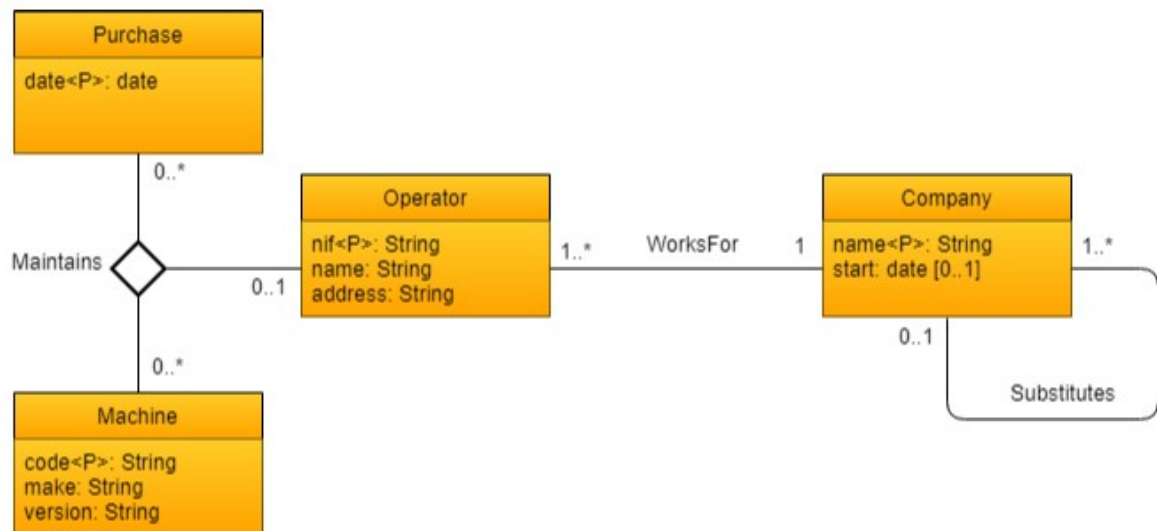




NOMBRE ALUMNO:

EJERCICIO 2: Modelo lógico

Dado el siguiente modelo conceptual:



Se solicita: obtener el modelo lógico relacional correspondiente indicando si, a causa de la transformación, aparecen atributos que pueden tomar valor NULL. Anotad también los requisitos que pueden existir en el modelo conceptual, que no se puedan representar en el modelo lógico.



NOMBRE ALUMNO:

SOLUCIÓN:

Company (name, start, substitute)

{substitute} is foreign key to Company

Machine (code, **make**, **version**)

Operator (nif, **name**, **address**, **worksFor**)

{worksFor} is foreign key to Company

Purchase (date)

Maintains (machine, date, **operator**)

{machine} is foreign key to Machine

{date} is foreign key to Purchase

{operator} is foreign key to Operator

Las claves primarias aparecen con subrayado continuo. Los atributos en **negrita** son NOT NULL.



NOMBRE ALUMNO:

EJERCICIO 3: Normalización

Nos proporcionan la siguiente relación en modelo relacional:

Relation (attr1, attr2, attr3, attr4, attr5)

Es decir, la clave primaria es la combinación {attr1, attr2} y no tenemos ninguna otra clave candidata. Aparte de las dependencias triviales debidas a la clave primaria, tenemos también esta dependencia: {attr3} → {attr5}. Contestad las siguientes preguntas y justificad las respuestas:

- a) ¿En qué forma normal se encuentra la relación?
- b) ¿En qué forma normal se encontraría la relación si añadimos una clave candidata {attr3} a la relación, manteniendo la misma dependencia mencionada en el enunciado?
- c) ¿En qué forma normal se encontraría la relación si la clave primaria fuese la combinación de atributos {attr1, attr2, attr3, attr4}, manteniendo también la dependencia mencionada en el enunciado y sin ninguna otra clave candidata?



NOMBRE ALUMNO:

SOLUCIÓN:

- a) La relación se encuentra en 2FN. Podemos asegurar que se encuentra en 1FN porque el enunciado nos dice que está en modelo relacional. También está en 2FN porque todos los atributos dependen de la clave al completo. Pero no está en 3FN porque el atributo *attr5*, además de depender de la clave, tiene también una dependencia con otro atributo (*attr3*) que no pertenece a la clave.
- b) En este caso la relación pasaría a estar en FNBC. Todos los atributos, incluso ahora también el atributo *attr5*, pasarían a depender de todas las claves al completo, con lo que la relación se encontraría en 3FN. También estaría en FNBC ya que el enunciado no menciona ninguna dependencia entre claves.
- c) La relación pasaría a estar solo en 1FN. No estaría en 2FN porque, teniendo en cuenta la dependencia del enunciado, el atributo *attr5* estaría ahora determinado por solo una parte de la clave.



NOMBRE ALUMNO:

EJERCICIO 4: preguntas teóricas

- 1) Disponemos de un SGBD que almacena datos sobre las ventas de una cadena de electrodomésticos. Cada día se almacenan los miles de transacciones que se efectúan. Para poder hacer promociones en función de las ventas de los productos, Marketing necesita disponer de las estadísticas y los valores acumulados de las ventas durante los últimos 12 meses anteriores al mes actual. Durante el mes actual se irán consultando estos datos para tomar las decisiones pertinentes. Estos valores acumulados se calculan a partir de una complicada serie de factores y otras tablas relacionadas (también con miles de registros). ¿Qué estrategia será más eficiente para proporcionar estos datos de forma instantánea?
- a) Una vista.
 - b) Una vista materializada.
 - c) Una tabla temporal.

SOLUCIÓN:

Una vista materializada. Los datos que necesita el departamento para hacer el seguimiento de las ventas pueden llegar a ser muy costosos de conseguir. Puesto que se indica que necesitan la información de forma instantánea, una vista materializada permite recalcular y ajustar los valores necesarios del resultado cada cuando se indique, dejándolos 'consolidados'. De este modo, cada consulta no implica volver a recalcularlo todo y el departamento puede conseguir los datos de forma casi instantánea. Con una vista los cálculos se tendrían que hacer cada vez que se necesita actualizar, cosa que podría llegar a ser prohibitiva en cuanto a coste computacional. La tabla temporal tampoco nos sirve puesto que solo guardaría los datos para una sesión determinada.

**NOMBRE ALUMNO:**

- 2) Nos muestran esta instrucción SQL. Servirá para crear una tabla que almacenará los datos del personal de una empresa:

```
CREATE TABLE Employee (  
    dept INTEGER CONSTRAINT PK_dept PRIMARY KEY,  
    NIF VARCHAR2(9) CONSTRAINT AK_NIF UNIQUE,  
    name VARCHAR2(30) CONSTRAINT NN_name NOT NULL,  
    surname1 VARCHAR2(30) CONSTRAINT NN_surname1 NOT NULL,  
    surname2 VARCHAR2(30),  
    startDate DATE,  
    salary NUMBER(4,2) CONSTRAINT CK_salary CHECK (salary > 0),  
    CONSTRAINT PK_Employees PRIMARY KEY (dept, NIF) );
```

Indicad si al ejecutar la creación de esta tabla se nos mostraría algún error. Si existieran posibles errores, explicad cuáles son y cómo solucionarlos.

SOLUCIÓN:

La ejecución dará un error: no puede haber más de una clave primaria en una tabla. En la sentencia SQL que se presenta hay una clave primaria a nivel de campo dept y otra clave primaria a nivel de tabla (dept, NIF). Por la restricción de unicidad de la clave primaria, la clave primaria tiene que ser única, no puede tener repetidos y con esta definición de tabla estamos definiendo dos claves primarias.

- 3) Nos dicen que al hacer un diseño nos aparece una tabla donde se almacenan los datos básicos de personas. Esta tiene como clave primaria el campo DNI. También nos dicen que existe una tabla llamada 'Department', y que tiene como clave primaria id. Finalmente nos corroboran que entre las reglas de negocio consta que todo empleado trabaja en un departamento. Con toda esta información nos piden crear la siguiente tabla:

**NOMBRE ALUMNO:**

```
CREATE TABLE Employee (  
  DNI CHAR(9) CONSTRAINT PK_Person PRIMARY KEY,  
  name VARCHAR2(30 CHAR) CONSTRAINT NN_name NOT NULL,  
  surname VARCHAR2(40 CHAR) CONSTRAINT AK_surname UNIQUE,  
  department CHAR(4) CONSTRAINT FK_Department REFERENCES Department(id),  
  CONSTRAINT FK_DNIPerson FOREIGN KEY (DNI) REFERENCES Person(DNI) );
```

Anotad los posibles errores lógicos y/o conceptuales que encontréis e indicad, si es necesario, cómo se tendrían que corregir.

SOLUCIÓN:

Nos indican que todo empleado pertenece a un departamento, por lo que tendría que constar la cláusula NOT NULL en la columna department.

También se observa que se ha indicado que el campo surname es de tipo UNIQUE. Este hecho es un error grave, puesto que no podría introducirse en la base de datos dos personas con los mismos apellidos (hermanos, por ejemplo). Aun así, y por lógica de contenido de información de cada tabla, el campo name y surname serían redundantes en la tabla Employee, puesto que tendrían que constar en la tabla Person.

EJERCICIO 5: cuestionarios autoevaluación

Este ejercicio consiste en hacer los cuestionarios de autoevaluación que tenéis disponibles en el aula.



NOMBRE ALUMNO:

Recursos

Para resolver esta PEC es necesario utilizar los contenidos estudiados en los módulos 1 a 5 del material docente.

Criterios de valoración

Se valorarán los ejercicios con los criterios introducidos en la PEC1 y en las Prácticas, por lo que se deberán seguir las indicaciones que se hicieron constar en las mismas, tales como que en el diseño conceptual hay que anotar las PK en todas las entidades e indicar las cardinalidades, utilizar los tipos Camel y Pascal en el diseño lógico relacional, argumentar las respuestas al indicar las formas normales y responder a las preguntas teóricas, etcétera.

El peso de los ejercicios en la nota total de la PEC es:

- Ejercicio 1: 25%
- Ejercicio 2: 20%
- Ejercicio 3: 15%
- Ejercicio 4: 20%
- Cuestionarios de autoevaluación: 20%

Esta PEC tiene un peso del 50% en la nota de evaluación continua.

Formato y fecha de entrega

El formato del archivo debe de ser Word u OpenOffice, y se debe entregar una versión del mismo en PDF. Haced envíos independientes para la versión doc/odt y el PDF.

El nombre del archivo deberá tener el siguiente formato:

PEC1_Apellido1_Apellido2_Nombre.extensión

Los apellidos se escribirán sin acentos. Por ejemplo, un estudiante que se llame Alfredo García Melgar tendría el siguiente nombre de archivo:

PEC1_Garcia_Melgar_Alfredo.doc (u .odt) y .pdf

IMPORTANTE: El nombre y apellidos del estudiante también tienen que aparecer en la portada y/o encabezado del documento con la solución.

Es responsabilidad del estudiante asegurarse que los documentos entregados se han subido correctamente y corresponden a la actividad que hay que presentar.



NOMBRE ALUMNO:

La fecha límite para entregar la PEC 2 es el viernes 21 de diciembre de 2018.

Nota: Propiedad intelectual

A menudo es inevitable, al producir una obra multimedia, hacer uso de recursos creados por terceras personas. Es por lo tanto comprensible hacerlo en el marco de una práctica de los estudios del Grado de Informática, siempre y cuando se documente claramente y no suponga plagio en la práctica.

Por lo tanto, al presentar una práctica que haga uso de recursos ajenos, se tiene que presentar junto con ella un documento en el que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el sitio de donde se obtuvo y su estatus legal: si la obra está protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL...). El estudiante tendrá que asegurarse que la licencia que sea no impide específicamente su uso en el marco de la práctica. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por el copyright.

Además, de deberán adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente si corresponde.

Otro punto a considerar es que cualquier práctica que haga uso de recursos protegidos por el copyright no podrá en ningún caso publicarse en Mosaico, la revista del Graduado en Multimedia a la UOC, a no ser que los propietarios de los derechos intelectuales den su autorización explícita.