

Caso T2

1) Un taller que se dedica a realizar revisiones de vehículos conserva, para cada revisión que realiza, una ficha en papel con la siguiente información:

- NIF del propietario del vehículo
- Apellidos
- Nombre
- Email
- Teléfono
- Matrícula del vehículo
- Marca
- Modelo
- Color
- Fecha de la revisión
- Kilometraje en el momento de la revisión
- Notas sobre la revisión

El propietario del taller decide informatizar su fichero, por lo que se requiere almacenar dicha información en una base de datos relacional. Describir las tablas y columnas que se utilizarán para diseñar un modelo de datos en 3ª forma normal que permita conservar la información de manera adecuada.

Información adicional:

- Se supone que el NIF de una persona la identifica unívocamente y no cambia nunca.
- Se supone que la matrícula de un vehículo lo identifica unívocamente, pero puede cambiar y es conveniente para los mecánicos disponer de la información de todas las revisiones que ha pasado un determinado vehículo aunque haya cambiado la matrícula. Por tanto, se debe implementar un método sencillo para conseguir que, en general, los mecánicos puedan identificar un vehículo aunque haya cambiado su matrícula.
- Un determinado vehículo puede cambiar de propietario, pero lo que interesa es mantener la información del propietario actual para poder avisarle de la fecha de la próxima revisión.
- Puesto que el propietario del taller quiere aplicar descuentos a los clientes habituales, es necesario que se pueda obtener toda la información de las revisiones que ha encargado un determinado cliente cuando era el propietario de un vehículo, aunque actualmente el propietario del mismo vehículo sea otro cliente.

Valor de la pregunta: 50% de la nota del caso

2) En el sistema operativo Linux se usan expresiones regulares en utilidades como grep, awk, vi, etc. para especificar patrones de texto que buscar o sustituir. Las implementaciones de los motores de expresiones regulares no son todas iguales, por lo que para el presente ejercicio se tomará un subconjunto de las expresiones regulares extendidas de GNU grep, definido en la guía de referencia que se adjunta.

Así pues, se requiere crear expresiones regulares, según la especificación contenida en la guía de referencia, para los casos que se plantean:

- a) Crear una expresión regular que concuerde con las diferentes formas de la palabra "gato": en masculino o femenino y sus respectivas versiones en plural. (Valor: 10%)
- b) Crear una expresión regular que concuerde con un nombre de host en el dominio "prueba.com". Se considera que los nombres de host antes del dominio pueden contener caracteres alfanuméricos en minúsculas y el guion, pero no pueden empezar por guion (por ejemplo, "servidor-web-1.prueba.com". (Valor: 10%)
- c) Crear una expresión regular que concuerde con el formato general de una dirección IP con los componentes individuales de la dirección siempre compuestos por 3 cifras entre 0 y 9 (no tiene por qué ser una dirección IP válida, ya que puede contener componentes mayores que 255). Por ejemplo: "010.850.000.305". (Valor: 10%)
- d) Crear una expresión regular que concuerde con las letras "abcd" al principio de una línea. Cualquier letra deberá poder aparecer en mayúscula o minúscula indistintamente. Por ejemplo "AbcD". (Valor: 10%)
- e) Crear una expresión regular que concuerde con una línea compuesta única y exclusivamente por dígitos impares (como mínimo 3 dígitos). (Valor: 10%)

Valor de la pregunta: 50% de la nota del caso

Guía de referencia

- La mayoría de caracteres, incluyendo los caracteres alfabéticos y numéricos son expresiones regulares que concuerdan con ellos mismos. Se distinguen las mayúsculas de las minúsculas.
- El punto (".") concuerda con cualquier carácter individual.
- Las expresiones de corchetes concuerdan con cualquier carácter individual que contengan. Por ejemplo, "[afj]" concuerda con la letra "a", la "f" o la "j". Se pueden especificar rangos con "-" (por ejemplo, "[a-zA-Z]"). Se puede hacer que la expresión concuerde con los caracteres que NO contiene usando el símbolo "^" como primer carácter (por ejemplo, "[^a-z]"). Para incluir el carácter "^" sin que se aplique su significado especial, se debe situar en cualquier lugar entre los corchetes excepto la primera posición ("^[a-z]"). Para incluir el carácter "-" sin que se aplique su significado especial, se debe situar entre los corchetes en la última posición ("[a-z-]"). Para incluir el carácter "]" sin que se cierre la expresión de corchetes, se debe situar justo después del corchete de apertura inicial ("[]a-z]").
- Los símbolos "^" y "\$" son meta-caracteres que concuerdan con una cadena vacía al inicio y final de la línea respectivamente.
- El símbolo "?" hace que el elemento anterior sea opcional, es decir, hace que concuerde si aparece 0 o 1 veces.
- El símbolo "*" concuerda cuando el elemento anterior está repetido 0 o más veces.
- El símbolo "+" concuerda cuando el elemento anterior está repetido 1 o más veces.
- Dos expresiones regulares concatenadas concuerdan con cualquier cadena formada por dos subcadenas concatenadas que concuerdan respectivamente con cada una de las dos expresiones regulares.
- El operador "|" se puede usar para especificar alternancia. Por ejemplo "uno|dos|tres" concordaría con cualquiera de las tres palabras.
- La precedencia de los operadores se aplica de la siguiente manera: en primer lugar, los operadores de repetición ("?", "*", "+"); en segundo lugar, la concatenación; y, en tercer lugar, la alternancia ("|"). La precedencia se puede modificar usando los paréntesis. Así, la expresión regular "(ab*|cd*|ef*)+", concordaría con cualquier cadena compuesta de uno o más elementos formados de la siguiente manera: una "a" seguida de un número indeterminado de "b"

(incluyendo 0) o bien una "c" seguida de un número indeterminado de "d" (incluyendo 0) o bien una "e" seguida de un número indeterminado de "f" (incluyendo 0). Un ejemplo podría ser "abbbbacdddddcaefffffffffff".

- Para hacer concordar los caracteres especiales (".", "*", etc.) con el carácter literal y no aplicar la operación especial que implica, se utiliza la barra inversa (también para la propia barra inversa). Por ejemplo, "google\\.com" o "c:\\windows".
- Por simplicidad, solo se considera un juego de caracteres ASCII de 7 bits, de tal manera que, por ejemplo, "[a-z]" concuerda con todas las letras minúsculas, puesto que no se contemplan otras "letras" como caracteres acentuados, específicos de idiomas concretos ("ñ", "ç"), etc., que requieren de tratamiento especial por parte del motor de expresiones regulares que se utiliza.