

# 1ª CONVOCATORIA 2018-19

Tiempo: 2 horas



**1.-** La empresa de trenes “A Toda Máquina” que tiene su sede en el país DBD-UA, quiere que le construyamos una base de datos para alojar la información que a continuación se describe. **(5 puntos)**

Esta empresa tiene varios trenes destinados al transporte de pasajeros dentro de DBD-UA. Estos trenes están identificados por un número y se conoce su fecha de puesta en funcionamiento y obligatoriamente la fecha de su última revisión. Según su recorrido tiene trenes de corta distancia y trenes larga distancia. De los de corta distancia nos interesa saber por cuantas comunidades autónomas pasan. Por otro lado, los trenes se clasifican como A, B, C, D o E en función de una serie de parámetros, pudiendo cada tren pertenecer a una única de estas categorías. Se conocerá el periodo máximo de tiempo que debe transcurrir entre las revisiones de los trenes. Todos los trenes de la misma categoría tienen el mismo periodo máximo.

La rutina es siempre la misma, diariamente cada tren se utiliza para el trayecto de ida de una localidad a otra, y luego para el trayecto de vuelta, considerándose estos dos trayectos independientes entre sí y realizándose solo una vez por día. Un tren hace durante toda su vida siempre el mismo trayecto diario. Interesa conocer para cada tren cuáles son sus localidades extremo en su recorrido (sin tener en cuenta otras localidades intermedias en las que pueda hacer paradas). Dos trenes pueden tener los mismos trayectos diario. Cada tren tiene un horario fijo de salida desde el origen (un extremo) y de llegada (al otro extremo) y otro horario fijo de salida y de llegada en el trayecto inverso. Estos horarios se deben conocer obligatoriamente. Cada tren tiene siempre asignados los mismos vagones. Estos vagones están numerados correlativamente comenzando con 1 en cada tren. De cada uno de estos vagones se conoce si dispone de aseo o no en el vagón y si tienen o no zona auxiliar de equipaje, además algunos vagones realizarán la función de bar y en este caso se conocerá, además de lo anterior, si tiene o no microondas y la capacidad de la nevera.

Los vagones, excepto los que son bar, tienen asientos que siguen la misma numeración dentro de cada vagón (en todos los vagones tendremos 1A, 1B, 1C, 1D, 2A, 2B, 2C, 2D, ...). De cada asiento se sabe su orientación (izquierda o derecha, no se refiere al sentido de la marcha). Este dato depende del asiento concreto, es decir el asiento 4A del vagón 2 del tren 1 puede ser orientación izquierda y el 4A del vagón 2 del tren 1 (o del tren 3) puede tener orientación derecha.

Se considera un viaje de un tren a la realización de cada uno de los trayectos (el trayecto de ida es un viaje y el de vuelta, otro) entre sus dos localidades extremo en una fecha concreta. Nos interesa conocer para cada viaje los asientos comprados(ocupados), así como el cliente que realiza la compra. Un mismo cliente podrá comprar varios asientos, pero un asiento en un viaje como mucho lo puede comprar un cliente.

De todo cliente se almacenará obligatoriamente tanto su NIF, como su nombre y su fecha de nacimiento. Además, una vez finalizado un viaje la empresa registra para cada viaje la hora real de salida del tren y la hora real de llegada. Estas horas se deben conocer para el caso en el que se quieran realizar reclamaciones relacionadas con la puntualidad. La empresa establece para ello una tabla pública de devoluciones de dinero en función del porcentaje de tiempo extra que se invierte en el viaje, si se invierte un 20% más de tiempo te devuelven el 10% de lo que has pagado, si se invierte un 40% más te devuelven un 25%, etc.

En cuanto a las localidades en las que operan estos trenes, se conoce de ellas su nombre que sirve para identificarlas, y el número de habitantes censado. De estas ciudades nos interesa conocer la distancia entre ellas por carretera (la distancia puede ser distinta si se va de la localidad A a B que de B a A) y, además para alguna de ellas se sabe qué localidad (solo una) es la que más se le parece en cuanto a infraestructuras.

La empresa organiza cursillos para su personal, estos cursillos están identificados por un código y se conoce obligatoriamente la descripción del mismo y el nivel que, como mínimo, debe tener el personal asistente al cursillo. Estos cursillos se realizan en las distintas ciudades donde operan los trenes de la empresa. Se debe conocer el calendario previsto de cursillos sabiendo que en cada ciudad se realiza como máximo un cursillo diario. Como los cursillos los imparten especialistas, el mismo cursillo no se podrá realizar a la vez en dos ciudades.

Algunos de los clientes se pueden dar de alta en la web de la empresa pasando a tener un usuario identificado por su dirección de correo electrónico (no es posible que dos clientes tengan el mismo correo electrónico). Entre estos usuarios registrados se sortean estancias en hoteles (sin ningún tipo de limite). Interesa saber los usuarios agraciados, así como el número de veces que les ha correspondido un hotel. Un hotel de los que entra en sorteo está identificado por un código, se conoce su nombre y está obligatoriamente situado en una y solo una de las localidades en las que operan los trenes. En una localidad no puede haber dos hoteles que entren en los sorteos.

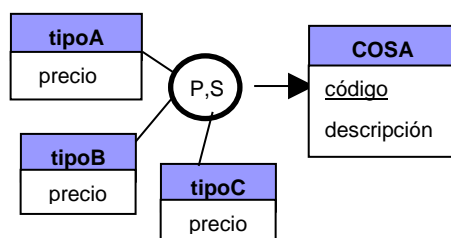
**2.-** Dado el siguiente esquema lógico relacional, obtener un diagrama EER del que se pueda haber extraído, sabiendo que no se han necesitado comentarios sobre pérdidas expresivas. **(2,5 puntos)**

<b>A(a0, a1, a2, a3)</b> CP: a0	<b>B(b0, b1)</b> CP: b0 Caj: b0 → A V.N.N.:b1	<b>C(c0, c1)</b> CP: c0 C.ajena: c0 → A C.ajena: c1 → A	<b>D(d0, d1, d2, d3)</b> CP: (d0, d1) C.ajena: d2 → B VNN: d2 VNN: d3	<b>E(e0,e1)</b> CP: e0 Caj: e0 → B
<b>F(f0,f1,f2)</b> CP: (f0,f1,f2) C.ajena: f0 → B C.ajena:f1 → E	<b>G(g0, g1, g2, g3, g4)</b> CP: (g0, g1) C.ajena: g0 → E C.ajena: g3, g4 → D V.N.N.:g3, g4	<b>I (i0, i1, i2, i3, i4, i5)</b> CP: i0 C.alternativa: i2 C.alternativa: i5 Cajena: i2 → B Cajena: i4 → C Cajena: i5 → H VNN: i4	<b>J(j0, j1, j2, j3, j4, j5)</b> CP: j0, j1 C. alternativa: j2, j3 C.ajena: j0, j1 → G C.ajena: j2, j3 → D C. ajena: j4 → H V.N.N.: j4	
<b>H(h0, h1, h2)</b> CP: h0 V.N.N.:h1				

**3.-** Contesta a las siguientes cuestiones (pon en tu respuesta el número de cuestión y la respuesta)  
(0 fallos: 1,75 puntos, 1 fallo: 1,25 puntos, 2 fallos: 0,75 puntos, 3 fallos: 0,25 puntos)

- En el diseño físico, ¿qué mecanismos nos ofrecen algunos SGBDR para gestionar las reglas de negocio que no hayan podido ser recogidas en el esquema relacional?
- ¿Qué características debería tener una generalización en el esquema conceptual para que al transformarla a tablas decidas definir una única tabla para toda la generalización y añadir una columna "tipo" a esa tabla, en lugar de definir una tabla para el objeto general y otra para cada uno de los subtipos?
- ¿Cómo quedaría la siguiente colección tras ejecutar la siguiente instrucción en mongoDB?  
db.prueba.update({codigo:"NO"},{\$set:{codigo:"YES"},\$inc:{valor:10},{upsert:true})  
{ "\_id": "xcd05", "codigo": "YES", "nombre": "No se puede", "valor": 5 }  
{ "\_id": "xcd04", "codigo": "NO", "nombre": "Sí se puede", "valor": 4, "puntos": 3 }
- Indica cuál de las siguientes sentencias en mongoDB serviría para recuperar el nombre de las pruebas que contengan en su código YES, ordenado descendientemente.  
a.- db.prueba.find({codigo:{\$regex:"YES"}},{nombre:1}). pretty({codigo:-1})  
b.- db.prueba.find({codigo:{\$regex:"YES"}},{\_id:0, nombre:1}). sort({nombre:-1})  
c.- db.prueba.find({codigo:{\$in:"YES"}},{\_id:0, nombre:1}). sort({codigo:-1})
- ¿Cómo quedaría la siguiente colección tras ejecutar la siguiente instrucción en mongoDB?  
db.prueba.update({codigo:"NO", puntos:3},{valor:12},{upsert:true})  
{ "\_id": "xcd05", "codigo": "YES", "nombre": "No se puede", "valor": 5 }  
{ "\_id": "xcd04", "codigo": "NO", "nombre": "Sí se puede", "valor": 4, "puntos": 3 }

**4.-** ¿Son equivalentes los siguientes 2 esquemas? Si no es así se debe indicar la diferencia, y si lo son se debe indicar como se recogen las características del esquema conceptual en el esquema relacional **(0,75 puntos)**



COSA(código, descripción) C.Primaria: código  
 tipoA(cosa, precio) C.Primaria: cosa  
 C.Ajena: cosa → COSA  
 tipoB(cosa, precio) C.Primaria: cosa  
 C.Ajena: cosa → COSA  
 tipoC(cosa, precio) C.Primaria: cosa  
 C.Ajena: cosa → COSA