

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30



75.585 16 06 18 EX

Espacio para la etiqueta identificativa con el código personal del **estudiante**.
Examen

Ficha técnica del examen

- Comprueba que el código y el nombre de la asignatura corresponden a la asignatura matriculada.
- Debes pegar una sola etiqueta de estudiante en el espacio correspondiente de esta hoja.
- No se puede añadir hojas adicionales, ni realizar el examen en lápiz o rotulador grueso.
- Tiempo total: **2 horas** Valor de cada pregunta: **Consta en cada pregunta.**
- En el caso de que los estudiantes puedan consultar algún material durante el examen, ¿cuáles son?: **No se puede consultar ningún material.**
- En el caso de poder usar calculadora, de que tipo? **NINGUNA**
- En el caso de que haya preguntas tipo test: ¿descuentan las respuestas erróneas? **NO**
¿Cuánto?
- Indicaciones específicas para la realización de este examen

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

Enunciados

Ejercicio 1 – Modelo conceptual (30%)

Nos disponemos a diseñar una base de datos para almacenar información sobre los elementos de transporte de los centros espaciales de la agencia espacial Elon.

Los centros espaciales se identifican por un nombre y queremos guardar el nombre del municipio donde está ubicado. Cada centro espacial contiene varias plataformas de lanzamiento (una como mínimo). Estas se identifican con el código alfanumérico relativo al centro espacial donde están ubicadas, es decir, el código alfanumérico que las identifica se puede repetir en diferentes Centros Espaciales. Tienen además un nombre y guardaremos la superficie que tienen en m2.

Cada una de las plataformas de lanzamiento tiene uno o más hangares, donde se montan y almacenan naves. De los hangares, identificados con un código alfanumérico, guardaremos el número de naves que pueden almacenar. Las naves se identifican por un código alfanumérico y sólo guardaremos las medidas (anchura, longitud y altura) en metros y el peso en kilogramos. Interesa disponer, además del número de naves que hay en el hangar actualmente, de un registro de las entradas y salidas de las naves de los hangares. Esto quiere decir conocer la fecha en que la nave entró en el hangar y la fecha en que salió. Cada nave puede haber entrado en un hangar más de una vez (en diferentes fechas) y, en una fecha determinada, en un hangar pueden entrar varias naves. Sin embargo, en una fecha determinada una nave sólo puede haber entrado en un hangar.

En los hangares operan las compañías de montaje de naves. Las compañías se identifican con un nombre y guardaremos su número de empleados. No todas las compañías trabajan en algún hangar, pero puede ser que trabajen en más de uno. Puede darse el caso de que algunos hangares estén en desuso y no trabaje ninguna compañía en ellos. Por otro lado, interesa tener información sobre los modelos de naves que se montan en los hangares, que se hacen de acuerdo con un patrón o modelo conocido. Un modelo de nave se identifica por una categoría alfanumérica y tiene asociado uno de los siguientes tipos de nave: Módulo de estación espacial, Nave Extrasolar, Nave de órbita geoestacionaria, Nave de órbita baja. Dado que un mismo modelo de nave lo pueden montar diversas compañías, se quiere conocer el coste de montaje para cada caso. Toda compañía monta como mínimo un modelo de nave y todo modelo lo monta como mínimo una compañía.

Los trabajadores llegan a los centros espaciales en transporte público (autobús o tren). De cada tren y/o autobús guardaremos un código alfanumérico (que los identifica) y la hora de salida. De los transportes que sean autobuses guardaremos su aforo. Se quiere disponer de la información de los trenes y autobuses que conectan con los centros espaciales. Cada centro tiene al menos un transporte que lo conecta. Cada transporte tiene conexión con un solo centro.

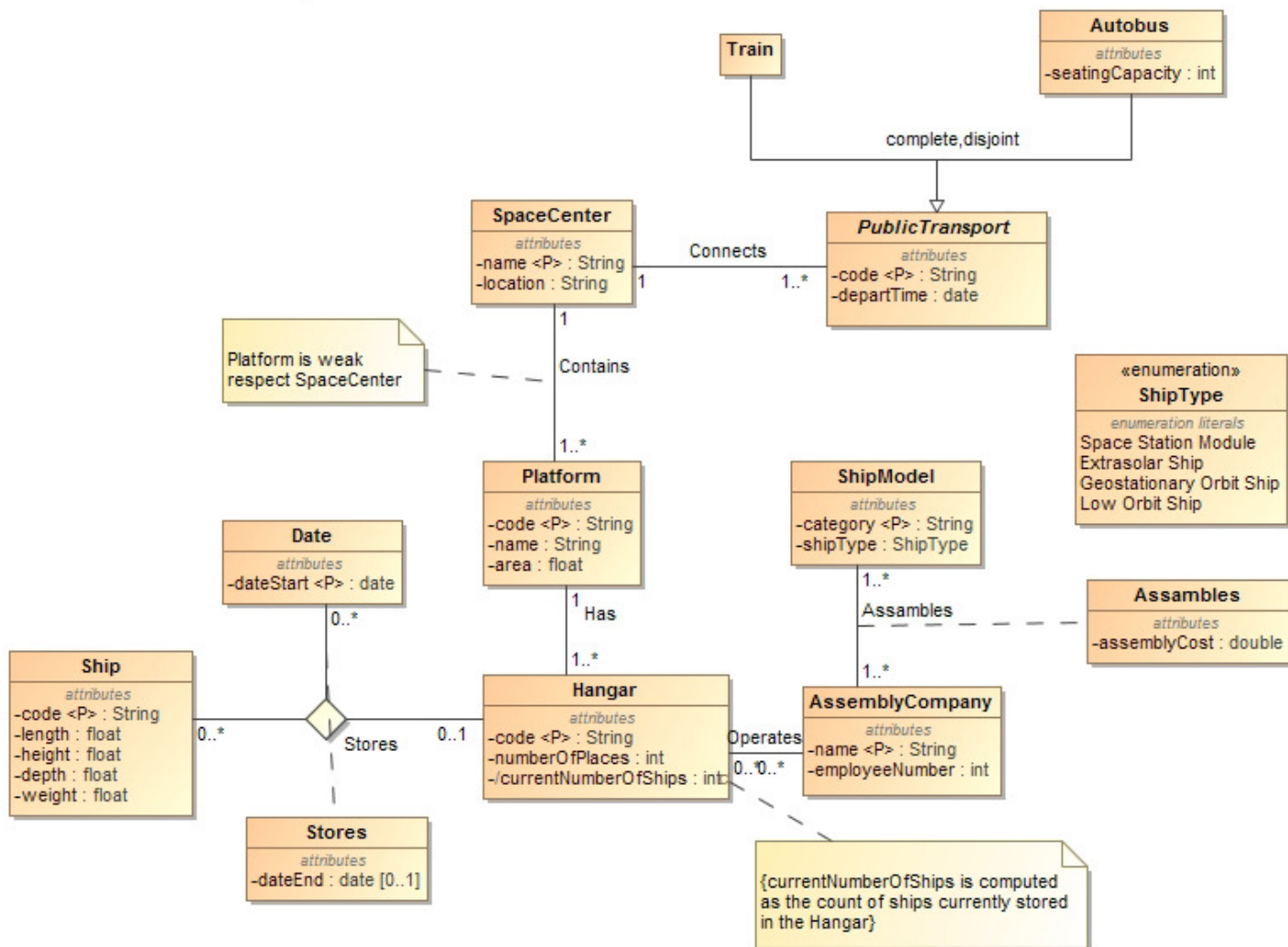
Se pide: realizad el diseño conceptual, mediante un diagrama de clases UML, que recoja toda la semántica del enunciado. Hay que indicar los atributos de los tipos de entidades con los tipos de datos correspondientes, las cardinalidades de los tipos de relaciones, todas las restricciones (claves primarias, alternativas...), así como los requisitos que no han quedado

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

reflejados en el esquema propuesto. Si se ha realizado alguna suposición semántica adicional también se indicará. No presupongáis otras relaciones, datos o funcionalidades que no consten explícitamente en el enunciado.

SOLUCIÓN:



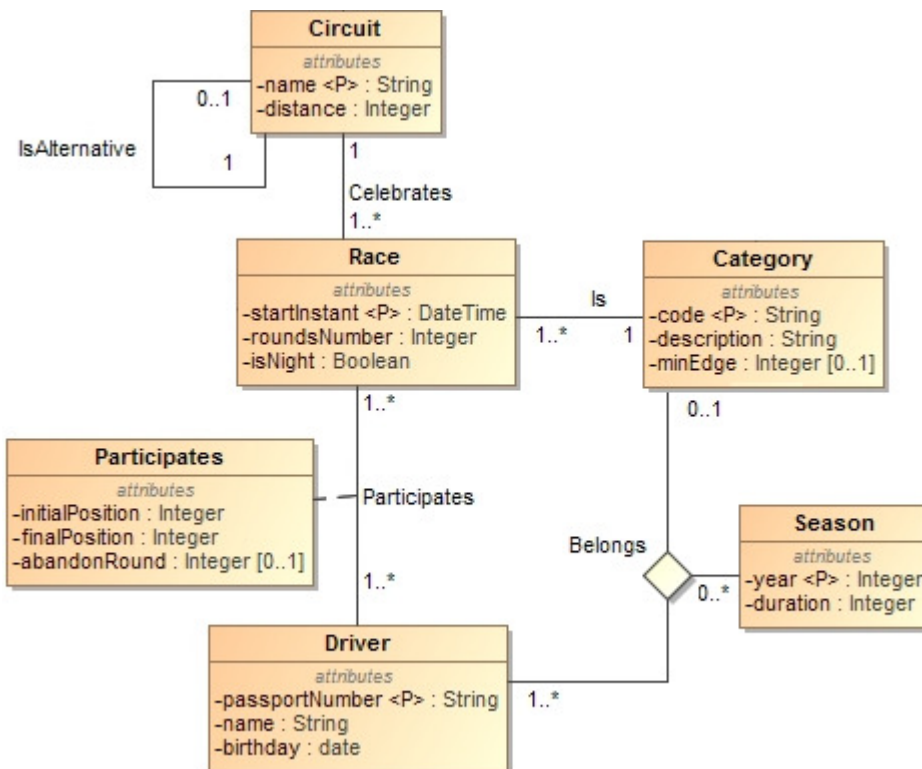
El modelo no garantiza que una misma nave no pueda estar en dos hangares en el mismo momento. Dado que no se garantiza que los periodos de estancia de una nave en un hangar no se puedan solapar, una misma nave puede estar relacionada con dos periodos que comiencen en tiempos diferentes, pero con fechas de finalización que hagan que parte de estos periodos coincidan.

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

Ejercicio 2 – Modelo lógico (20%)

Dado el modelo conceptual siguiente:



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

SOLUCIÓN:

Circuit (name, **distance**, isAlternative)
{isAlternative} is foreign key to Circuit

Category (code, **description**, minEdge)

Race (startInstant, **roundsNumber**, **isNight**, **codeCategory**, **nameCircuit**)
{codeCategory} is foreign key to Category
{nameCircuit} is foreign key to Circuit

Driver (passportNumber, **name**, **birthday**)

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

Participates (startInstantRace, passportNumberDriver, **initialPosition**, **finalPosition**, abandonRound)

{startInstantRace} is foreign key to Race

{passportNumberDriver} is foreign key to Driver

Season (year, **duration**)

Belongs (passportNumberDriver, yearSeason, **codeCategory**)

{passportNumberDriver} is foreign key to Driver

{yearSeason} is foreign key to Season

{codeCategory} is foreign key to Category

El modelo no permite asegurar que para toda temporada y categoría tendremos al menos un conductor que participe.

Nota: Los atributos que no son clave y no pueden ser NULL están en negrita. Las claves primarias están subrayadas con línea continua.

Ejercicio 3 – Normalización (20%)

Tenemos la siguiente relación en modelo relacional:

Relation (attr1, attr2, attr3, attr4)

Es decir, la clave primaria es la combinación {attr1, attr3} y no tenemos ninguna otra clave candidata. Aparte de las dependencias triviales debidas a la clave primaria, tenemos una dependencia {attr1, attr2} → {attr4}. Contestad las siguientes preguntas y justificad las respuestas:

- ¿En qué forma normal se encuentra la relación?
- Eliminamos la dependencia {attr1, attr2} → {attr4}. ¿En qué forma normal se encontraría ahora la relación?
- Una vez eliminada la dependencia anterior, incluimos en la relación un nuevo atributo attr5, que añade una dependencia {attr5} → {attr2}. La clave primaria sigue siendo adecuada. ¿En qué forma normal se encontraría ahora la relación?
- Para acabar, nos damos cuenta de que el nuevo atributo attr5 es una clave candidata de la relación. ¿En qué forma normal acaba la relación ante este hecho?

SOLUCIÓN:

- La relación está en primera forma normal (1FN). Al estar en modelo relacional, podemos estar seguros de que cumple la 1FN. Aparte de las dependencias triviales, tenemos que el atributo attr4 no está determinado por la clave al completo {attr1, attr3}, con lo que no cumple la 2FN.

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

- b) La relación pasaría a estar en FNBC. Como ya no tendríamos ninguna otra dependencia aparte de las triviales de la clave primaria, se da que los atributos que no pertenecen a la clave (*attr2* i *attr4*) están determinados por (y sólo por) la clave.
- c) La relación estaría ahora en 2FN. Nos dicen que la clave primaria sigue siendo adecuada y, por tanto, el nuevo atributo *attr5* está determinado por la clave. Pero el nuevo atributo implica una nueva dependencia en la que un atributo que no pertenece a ninguna clave (*attr2*) está determinado, aunque sea en parte, por un atributo que no pertenece a ninguna clave (*attr5*). Esto no cumple la 3FN.
- d) Al pasar a ser el atributo *attr5* una clave candidata, la nueva dependencia que añadíamos no contradice la 3FN y, además, por el hecho de ser clave candidata, el resto de atributos también estarían determinados por *attr5*. Así, la relación se encontraría en FNBC.

Ejercicio 4 (30%)

- 1) Disponemos de un SGBD que almacena datos censales de los EEUU (unos 325 millones de habitantes), donde cada día se registran cientos de nacimientos y defunciones. Los funcionarios del Departamento Nacional de Estadística (DNE) deben preparar, mensualmente, un informe demográfico con el total de habitantes por Estado, sexo, rangos de edad, etc. y un informe resumen anual con la misma información.

¿Qué estrategia será más eficiente para proporcionar estos datos? Justificad la respuesta.

- a) Una vista.
- b) Una vista materializada.
- c) Una tabla temporal.

SOLUCIÓN:

La estrategia más eficiente será utilizar una **vista materializada**, puesto que nos permite ejecutarla una sola vez y almacenar los datos calculados de cada mes, para así poder realizar de una manera muy efectiva el acumulado anual. Una vista se debería ejecutar cada fin de mes durante todo el año y guardar los resultados en una tabla auxiliar para poder calcular el acumulado anual de manera eficiente. En el caso de una tabla temporal, los datos calculados no serán persistentes más allá de la sesión o transacción en que se haga el cálculo, con lo cual nos encontramos con el mismo problema que para el caso de la vista.

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

2) Considerad la siguiente instrucción SQL:

```
CREATE TABLE Client (
  id INTEGER CONSTRAINT PK_Client PRIMARY KEY,
  name VARCHAR2(50 CHAR) CONSTRAINT NN_ClientName NOT NULL,
  address VARCHAR2(70 CHAR),
  email VARCHAR2(50 CHAR) CONSTRAINT AK_ClientEmail UNIQUE,
  assoc INTEGER,
  CONSTRAINT FK_ClientAssociation FOREIGN KEY (assoc)
    REFERENCES Association (id)
);
```

Contestad las siguientes preguntas con un (SI / NO) y justificad la respuesta. No se valorarán las respuestas que no estén debidamente justificadas.

- ¿Se puede insertar un valor duplicado (no único) en el campo id?
- ¿Se puede insertar un valor duplicado (no único) en el campo name?
- ¿Se puede insertar un valor duplicado (no único) en el campo address?
- ¿Se puede insertar un valor duplicado (no único) en el campo email?
- ¿Se puede insertar un valor duplicado (no único) en el campo assoc?

SOLUCIÓN:

- No**, no se puede insertar un valor duplicado en el campo id, ya que este campo es la clave primaria de la tabla. Una clave primaria no acepta valores duplicados, este debe ser único.
- Sí**, sí se puede insertar un valor duplicado en el campo name. Aunque tenga una restricción de tipo NOT NULL, esta restricción acepta valores duplicados.
- Sí**, sí se puede insertar un valor duplicado en el campo address, no hay ninguna restricción que lo impida.
- No**, no se puede insertar un valor duplicado en el campo email. Este campo tiene una restricción específica de tipo UNIQUE.
- Sí**, sí se puede insertar un valor duplicado en el campo assoc. Aunque tenga una restricción de tipo FOREIGN KEY, esta restricción acepta valores duplicados. Sólo debe cumplir la restricción de integridad referencial con la tabla Association.

3) Considerad la siguiente instrucción SQL:

```
CREATE TABLE Employees (
  dept INTEGER,
  NIF VARCHAR2(9 CHAR),
  name VARCHAR2(30 CHAR) CONSTRAINT AK_Name UNIQUE,
  surname VARCHAR2(30 CHAR) CONSTRAINT NN_Surname NOT NULL,
  hiredate DATE,
  salary NUMBER(6,2) CONSTRAINT CH_Salary CHECK (salary>1000),
  CONSTRAINT PK_Employees PRIMARY KEY (dept, NIF)
);
```

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

Indicad si esta definición de tabla permitirá insertar las filas (F1..F5) en el mismo orden en el que aparecen en la siguiente tabla:

	Dept	NIF	name	surname	hiredate	salary
F1	1	1234567-F		Fitzgerald	13/06/2001	1300
F2		87654321A	Carlos	Castillo	01/03/2007	1415,27
F3	2	98127634T		Heith		1796,34
F4	2	87654321A	Lydia	Chevalier	22/12/2001	1126,12
F5	1	87654321A	Carla	Castillo	01/08/2013	850

En el caso de que la fila no se pueda insertar, justificad todos los motivos por los que esta será descartada.

SOLUCIÓN:

F1: **Sí** que permite insertarla.

F2: **No** permite insertarla. Se produce una violación de entidad de la clave primaria, el campo dept forma parte de la PK y, consecuentemente, no puede ser NULL.

F3: **No** permite insertarla porque el campo name es UNIQUE y sólo permitiría el primer valor NULL, el de la fila 1.

F4: **Sí** que permite insertarla.

F5: **No** permite insertarla ya que el campo salary no cumple la restricción CHECK que especifica que el campo salary debe ser mayor de 1000.

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30

Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	16/06/2018	18:30