

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	09/06/2018	09:00



75.585 09 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	09/06/2018	09:00

### Enunciados

#### Ejercicio 1 – Modelo conceptual (30%)

Nos disponemos a diseñar parte de una base de datos que permita almacenar la información relativa a los destinos de las diferentes naves que envía la agencia espacial Elon.

De los destinos queremos guardar la designación de la International Astronomical Union (IAU Designation), que es un código alfanumérico que los identifica, y el nombre (si lo hay, que es posible que no). Los hay de dos tipos diferentes: planetas y asteroides. De los planetas queremos registrar el diámetro (en Km).

Los destinos son visitados por las sondas, teniendo en cuenta algunas restricciones que se presentan más adelante.

Cada planeta tiene varias ubicaciones. Una ubicación se identifica por una latitud (un número real) y una longitud (otro número real) relativa al planeta donde se encuentra. Dicho de otro modo, una misma longitud y latitud que identifica una ubicación se puede repetir en diferentes planetas. De cada ubicación queremos guardar la cantidad de combustible que se necesita para llegar.

Las naves que participan en los envíos contienen sondas. Una nave se identifica por el nombre del proyecto al que pertenece y, además, tiene un máximo responsable. De las sondas queremos guardar el código alfanumérico, que las identifica, el nombre, y cuál es su fuente de energía, que sólo puede ser combustible, paneles solares o energía nuclear. Cada sonda pertenece a una única nave. Interesa conocer qué sondas han aterrizado en qué ubicaciones de planetas y en qué períodos de tiempo. Un periodo de tiempo viene determinado por una fecha de inicio y una de fin. Además, sabemos que una sonda durante un período de tiempo puede visitar varias ubicaciones y que cada ubicación sólo puede ser visitada por una sonda durante un período. Así mismo una sonda puede visitar una misma ubicación en diferentes períodos de tiempo.

También puede darse el caso de que alguna sonda visite un asteroide, de lo cual queremos tener constancia. De hecho, queremos saber qué sondas han visitado cada asteroide y, en concreto, durante qué ventana de tiempo. Entendemos por "ventana de tiempo" un instante inicial y una duración que se asocia a cada visita. Sabemos que una sonda puede visitar un asteroide en diferentes ventanas de tiempo, que en una ventana de tiempo una sonda sólo puede visitar un asteroide y que en una ventana de tiempo un asteroide puede ser visitado sólo por una sonda. Además, de cada visita nos interesa saber el coste en combustible.

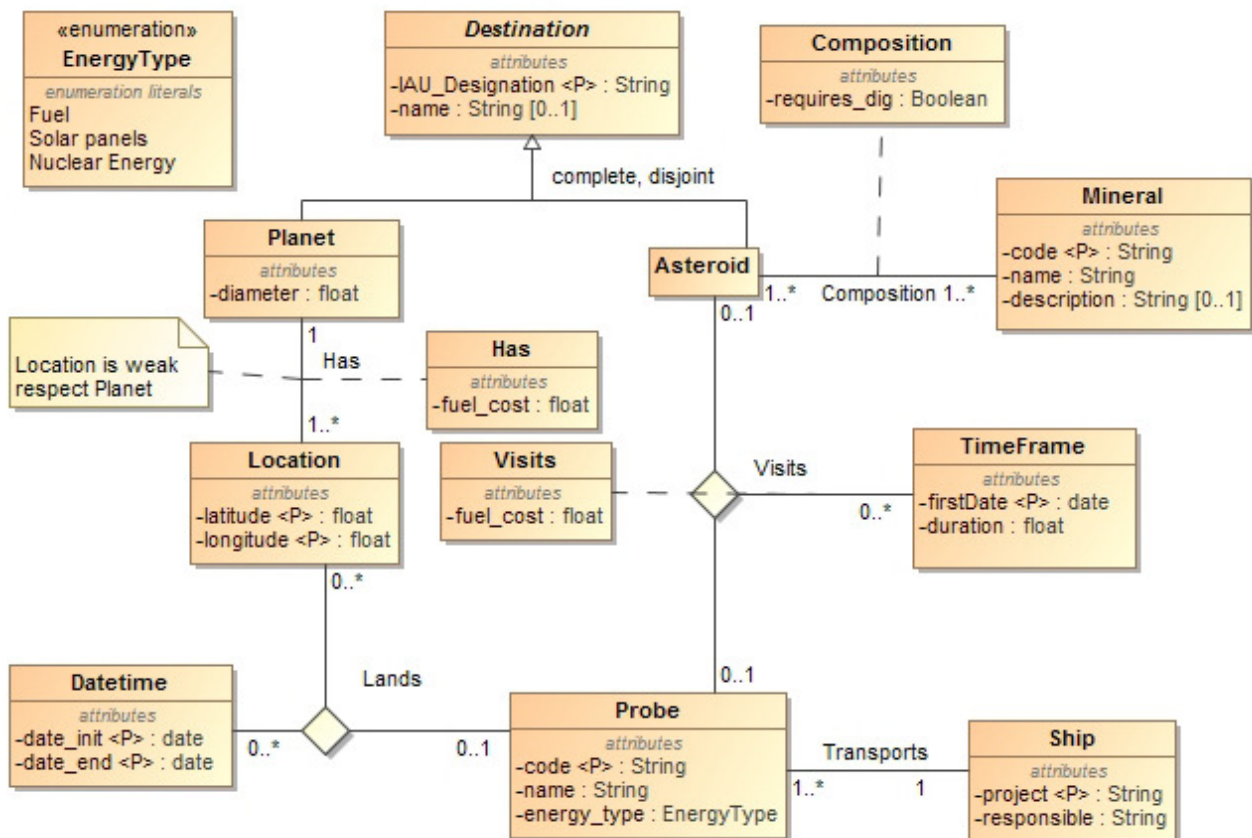
Los asteroides contienen minerales. En un asteroide se pueden encontrar uno o más minerales y un mineral se puede encontrar en uno o más asteroides. De los minerales queremos guardar un código identificador alfanumérico, el nombre y, si existe, una breve descripción. Dependiendo del asteroide, puede ser necesario minar o no, para obtener un mineral concreto. En este modelo no nos interesa registrar quién recoge los minerales, tan sólo su existencia.

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	09/06/2018	09:00

**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 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:



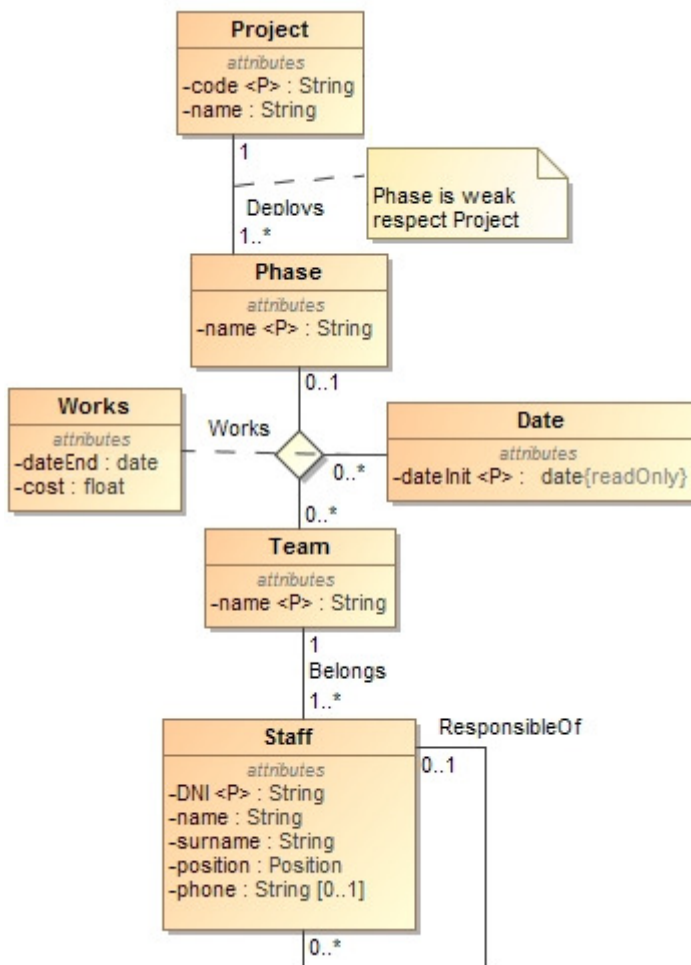
Dado que no se garantiza que las ventanas de tiempo no se puedan solapar, el modelo no puede garantizar que, en una ventana de tiempo, no existan varias sondas en un mismo asteroide, pues pueden estar allí por períodos de tiempo que se solapen, comenzando en momentos diferentes, pero coincidiendo en parte de su duración.

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	09/06/2018	09:00

### 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 pueden representar en el modelo lógico.

### SOLUCIÓN:

Project (code, **name**)

Phase (name, codeProject)  
{codeProject} is foreign key to Project

Date (dateInit)

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	09/06/2018	09:00

Team (name)

Works (dateInit, nameTeam, **namePhase**, **codeProject**, **dateEnd**, **cost**)

{namePhase, codeProject} is foreign key to Phase

{dateInit} is foreign key to Date

{nameTeam} is foreign key to Team

Staff (DNI, **name**, **surname**, **position**, phone, **nameTeam**, responsibleOf)

{nameTeam} is foreign key to Team

{responsibleOf} is foreign key to Staff

**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)**

Es decir, la clave primaria es la combinación {attr1, attr2} y no tenemos ninguna otra clave candidata. Contestad las siguientes preguntas y justificad las respuestas:

- Nos dicen que la relación está en 1FN, pero no en 2FN. En caso de existir, ¿qué dependencias tendríamos en la relación, aparte de las triviales debidas a la clave primaria?
- Ahora nos dicen que la relación está en 2FN, pero no en 3FN. En caso de existir, ¿qué dependencias tendríamos ahora, aparte de las triviales debidas a la clave primaria?
- Nos comunican ahora que la relación está en 3FN, pero no en FNBC. En caso de existir, ¿qué dependencias tendríamos ahora, aparte de las triviales debidas a la clave primaria?
- Finalmente, nos acaban comunicando que en realidad la relación está en FNBC. En caso de existir, ¿qué dependencias tenemos, incluyendo las triviales debidas a la clave primaria?

### SOLUCIÓN:

- Si la relación no está en 2FN, nos indica que algún atributo que no forma parte de ninguna clave (sólo tenemos attr3) está determinado tan sólo por una parte de la clave primaria. Con esta información, sólo podríamos decir que o bien tenemos la dependencia {attr1} → {attr3} o bien la dependencia {attr2} → {attr3}.

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	09/06/2018	09:00

- b) Esta afirmación es imposible. El hecho de que la relación no esté en 3FN implica que debemos tener algún atributo que no pertenezca a la clave (sólo tenemos *attr3*) que debe estar determinado (aunque sólo sea parcialmente) por algún atributo que no pertenezca a la clave, pero no tenemos ningún otro.
- c) Continuando con la misma explicación, si no está en FNBC indica que existe alguna dependencia entre alguna parte de la clave y algún conjunto de atributos que no sea una clave. Así, la conclusión es que debemos tener alguna de estas dependencias:  $\{attr3\} \rightarrow \{attr1\}$ ,  $\{attr3\} \rightarrow \{attr2\}$ ,  $\{attr3, attr2\} \rightarrow \{attr1\}$ ,  $\{attr3, attr1\} \rightarrow \{attr2\}$ ,  $\{attr1\} \rightarrow \{attr2\}$ ,  $\{attr2\} \rightarrow \{attr1\}$ .
- d) Si la relación está en FNBC, tendríamos que toda dependencia sería entre la clave de la relación y el resto de atributos. Por tanto, sólo tendríamos la dependencia trivial  $\{attr1, attr2\} \rightarrow \{attr3\}$  y ninguna otra.

### 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 periódicamente varios informes demográficos, para los que necesitan en todo momento disponer de datos de resumen actualizados, con la posibilidad de filtrar por Estado, sexo y/o edad. Cuando los funcionarios del DNE visualizan los datos, los anotan en otro programa y siguen con su trabajo.

¿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**, porque los datos que necesita el DNE para hacer los informes cambian cada vez y es requisito que estos estén actualizados. Por ese motivo, una vista materializada no tendría sentido porque los datos pueden cambiar en cada consulta y no es necesario almacenarlos, puesto que los introducen en otra aplicación. Una tabla temporal no es necesaria ya que los funcionarios del DNE simplemente necesitan visualizar los datos y anotarlos, y estos no serán necesarios en ningún otro proceso.

## Examen 2017/18-2

Asignatura	Código	Fecha	Hora inicio
Diseño de bases de datos	75.585	09/06/2018	09:00

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 NULL en el campo id?
- ¿Se puede insertar un valor NULL en el campo name?
- ¿Se puede insertar un valor NULL en el campo address?
- ¿Se puede insertar un valor NULL en el campo email?
- ¿Se puede insertar un valor NULL en el campo assoc?

### SOLUCIÓN:

- No**, no se puede insertar un valor NULL en el campo id, ya que este campo es la clave primaria de la tabla. Una clave primaria no acepta valores NULL.
- No**, no se puede insertar un valor NULL en el campo name. Este campo tiene una restricción específica de tipo NOT NULL.
- Sí**, sí se puede insertar un valor NULL en el campo address, no hay ninguna restricción que lo impida.
- Sí**, sí se puede insertar un valor NULL en el campo email. Aunque tenga una restricción de tipo UNIQUE, esta restricción acepta valores NULL, siempre que sea el primer valor NULL que se inserta en este campo.
- Sí**, sí se puede insertar un valor NULL en el campo assoc. Aunque tenga una restricción de tipo FOREIGN KEY, esta restricción acepta valores NULL.

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	09/06/2018	09:00

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	Edward	Fitzgerald	13/06/2001	999,57
F2	2	87654321A	Carlos	Castillo	01/03/2007	1415,27
F3	2	98127634T		Heith		1796,34
F4	4	82736455G	Lydia	Chevalier	12/22/2001	1126,12
F5	2	40897654J	Carlos	Castillo	01/08/2013	1400

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

### SOLUCIÓN:

F1: **No** permite insertarla ya que el campo salary no cumple la restricción CHECK que especifica que el campo salary debe ser mayor de 1000.

F2: **Sí** que permite insertarla.

F3: **Sí** que permite insertarla.

F4: **No** permite insertarla. El valor del campo hiredate debe ser una fecha válida y en este caso es errónea, puesto que 22 está fuera del dominio de los meses del año.

F5: **No** permite insertarla. El valor del campo name debe ser único y este ya se había introducido anteriormente en la fila 2.