

**Bases de Datos  
Curso 2014-2015  
Grado en Ingeniería del Software  
Examen Junio**

Nombre: \_\_\_\_\_

Se debe entregar esta hoja.

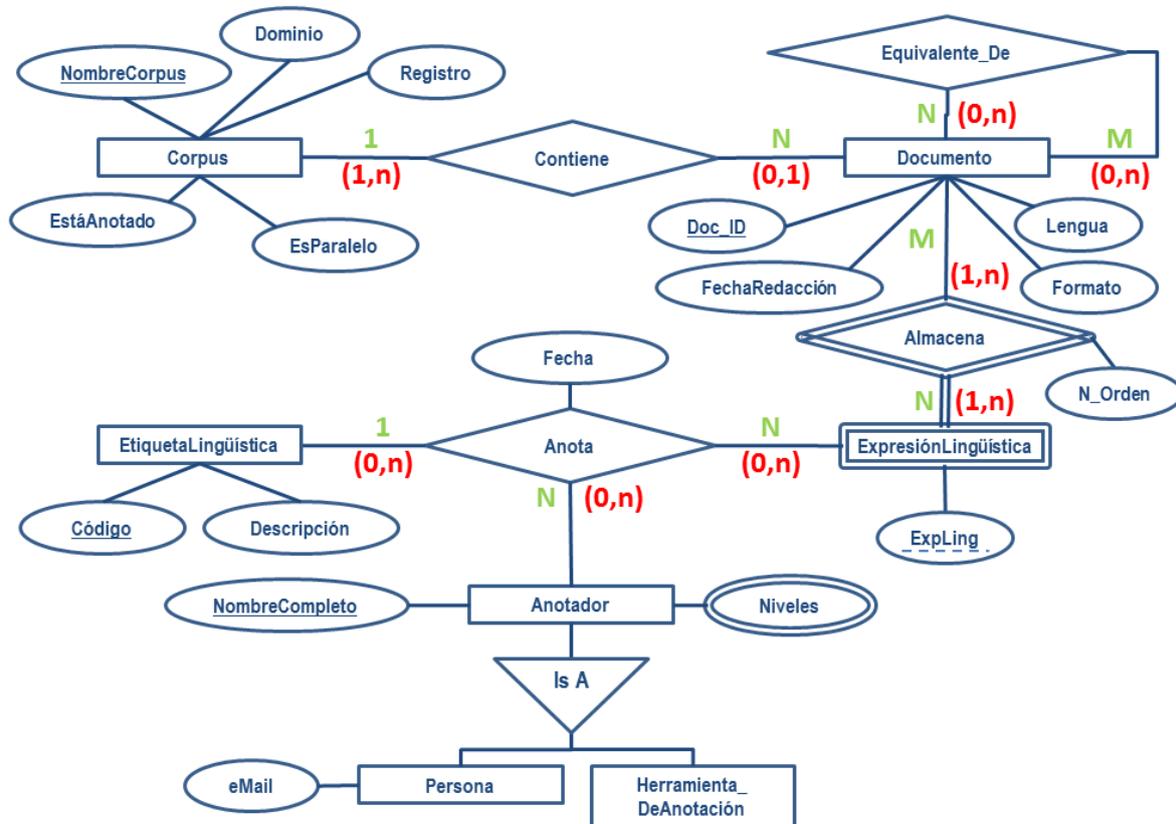
- 1) **(4,5 puntos) A partir de la información sobre la base de datos que se describe más abajo, se pide:**
  - a) **(2,5 puntos) Su modelo entidad-relación asociado, incluyendo atributos, claves, restricciones de cardinalidad y de participación.**
  - b) **(1,25 puntos) El modelo relacional que se obtiene a partir del modelo anterior al aplicar las transformaciones apropiadas, indicando las restricciones de integridad referencial resultantes.**
  - c) **(0,75 puntos) El desglose de todas las restricciones de integridad que no se hayan podido implementar en el modelo relacional (puedes omitir las restricciones de dominio), o que se hayan perdido en el proceso de transformación.**

Un grupo de investigación de la universidad necesita crear una base de datos de corpus lingüísticos para su procesamiento estadístico (como se explica más adelante, un corpus es una colección de documentos anotados lingüísticamente). La información que habrá que representar y almacenar en la misma es la siguiente:

- Los corpus están formados por uno o varios documentos, seleccionados de acuerdo con unos ciertos criterios lingüísticos y/o estadísticos. Por su parte, los documentos se incorporan a la base de datos aunque aún no pertenezcan a un determinado corpus, y cada documento pertenecerá como máximo a un corpus.
  - Cada corpus tiene un nombre distintivo y se caracteriza por (i) el dominio al que pertenecen sus documentos (por ejemplo, literatura, economía, etc.); (ii) el registro lingüístico en el que están escritos sus documentos (formal, informal o mixto); (iii) un campo lógico que indica si ya está anotado o está pendiente de anotación; y (iv) otro campo lógico adicional que, en el caso de corpus multilingües, indica si el corpus es o no paralelo.
- OBSERVACIÓN:** Se dice que un corpus es paralelo cuando (a) contiene documentos en varias lenguas; y (b) para cada documento escrito en una lengua dada es posible encontrar otro documento equivalente en el mismo, con los mismos contenidos, pero escritos en otra lengua distinta del corpus.
- Cada documento se nombra con un identificador único e independiente, y está escrito en una única lengua, que deberá declararse al introducir el documento en la base de datos. De cada documento se necesita saber asimismo su fecha de redacción y su formato (PDF, DOC, DOCX o HTML).
  - La base de datos debe reflejar aquellos documentos que son equivalentes a otros, según la definición anterior.
  - Los documentos almacenan una o más expresiones lingüísticas. Para cada expresión se guardará la expresión en sí y, además, la base de datos recogerá su orden dentro del documento al que pertenece. No se almacenan expresiones que no estén en ningún documento.

- El proceso de anotación consiste en que un anotador asigna una etiqueta a una expresión lingüística de un documento. Dicha información sobre la anotación debe registrarse en la base de datos junto con la fecha en la que se realizó. Inicialmente, las expresiones de un documento no tienen asignada anotación alguna.
- Las expresiones lingüísticas pueden ser anotadas por varios anotadores. Cada uno de ellos anotará la expresión con la etiqueta lingüística que considere oportuna (puede ser la misma o no). Por ello, a cada expresión lingüística le pueden llegar a corresponder varias etiquetas lingüísticas (pero de diferentes anotadores), pues un anotador sólo anotará como máximo una vez cada expresión de un documento (es decir, con una única etiqueta a lo sumo).
- De cada anotador (que puede ser humano o una aplicación software de anotación lingüística) se almacenará su nombre (completo) y los niveles de anotación en los que se especializa (gramatical, sintáctico y/o semántico). Asimismo, si el anotador es humano, se almacenará también su email.
- Los anotadores se guardan en la base de datos de antemano, es decir, se van registrando anotadores a medida que se van conociendo, aunque aún no haya en la base de datos documentos etiquetados por ese anotador.
- Las etiquetas lingüísticas con las que se anotan las expresiones lingüísticas quedan determinadas por su código identificativo único (“NCMS”, “SN”, etc.). A él se asociará una descripción en lenguaje natural de su significado (“Nombre común, masculino singular”, “Sintagma nominal”, etc.). Las etiquetas se han generado semiautomáticamente; por lo tanto, hay etiquetas que es posible que no lleguen a usarse nunca para anotar una expresión lingüística.

a) Modelo entidad-relación:



b) Modelo relacional:

- Corpus(NombreCorpus, Dominio, Registro, EstáAnotado, EsParalelo)
- Documento(Doc ID, NombreCorpus\*, Lengua, Formato, FechaRedacción)
- ExpresiónLingüística(Doc ID, ExpLing, N\_Orden)
- Anotador(NombreCompleto, EsAnotHumano, eMail\*)  
 NOTA: El atributo EsAnotHumano se añade para evitar el problema de la ambigüedad de los nulos (email desconocido o atributo no aplicable, en el caso de las aplicaciones de anotación automática).
- NivelesAnotador(NombreCompleto, Nivel)
- EtiquetaLingüística(Código, Descripción)
- Equivalente\_De(Doc ID1, Doc ID2)
- Anota(NombreCompleto, Doc ID, ExpLing, Código, Fecha)
- $\prod_{\text{NombreCorpus}}(\text{Documento}) \subseteq \prod_{\text{NombreCorpus}}(\text{Corpus})$
- $\prod_{\text{Doc\_ID}}(\text{ExpresiónLingüística}) \subseteq \prod_{\text{Doc\_ID}}(\text{Documento})$
- $\prod_{\text{NombreCompleto}}(\text{NivelesAnotador}) \subseteq \prod_{\text{NombreCompleto}}(\text{Anotador})$
- $\prod_{\text{Doc\_ID1}}(\text{Equivalente\_De}) \subseteq \prod_{\text{Doc\_ID}}(\text{Documento})$
- $\prod_{\text{Doc\_ID2}}(\text{Equivalente\_De}) \subseteq \prod_{\text{Doc\_ID}}(\text{Documento})$
- $\prod_{\text{NombreCompleto}}(\text{Anota}) \subseteq \prod_{\text{NombreCompleto}}(\text{Anotador})$
- $\prod_{\text{Código}}(\text{Anota}) \subseteq \prod_{\text{Código}}(\text{EtiquetaLingüística})$
- $\prod_{(\text{Doc\_ID}, \text{ExpLing})}(\text{Anota}) \subseteq \prod_{(\text{Doc\_ID}, \text{ExpLing})}(\text{ExpresiónLingüística})$

c) Restricciones de integridad adicionales:

c.1) Las restricciones de integridad que no se han podido implementar son las restricciones de dominio siguientes (no pedidas):

- $\Pi_{\text{Registro}}(\text{Corpus}) \subseteq \{\text{"formal"}, \text{"informal"}, \text{"mixto"}\};$
- $\Pi_{\text{EstáAnotado}}(\text{Corpus}) \subseteq \{\text{TRUE}, \text{FALSE}\};$
- $\Pi_{\text{EsParalelo}}(\text{Corpus}) \subseteq \{\text{TRUE}, \text{FALSE}\};$
- $\Pi_{\text{Formato}}(\text{Documento}) \subseteq \{\text{"PDF"}, \text{"DOC"}, \text{"DOCX"}, \text{"HTML"}\};$
- $\Pi_{\text{EsAnotHumano}}(\text{Anotador}) \subseteq \{\text{TRUE}, \text{FALSE}\};$
- $\Pi_{\text{Nivel}}(\text{NivelesAnotador}) \subseteq \{\text{"gramatical"}, \text{"sintáctico"}, \text{"semántico"}\};$
- Los atributos FechaRedacción y Fecha, pertenecientes a Documento y Anota, respectivamente, son de tipo FECHA (DATE o TIMESTAMP)

- Además, tal y como está especificada la base de datos, no es posible comprobar que

\* el valor del atributo EsParalelo (de Corpus) adopta un valor coherente con el contenido verdadero de la base de datos.

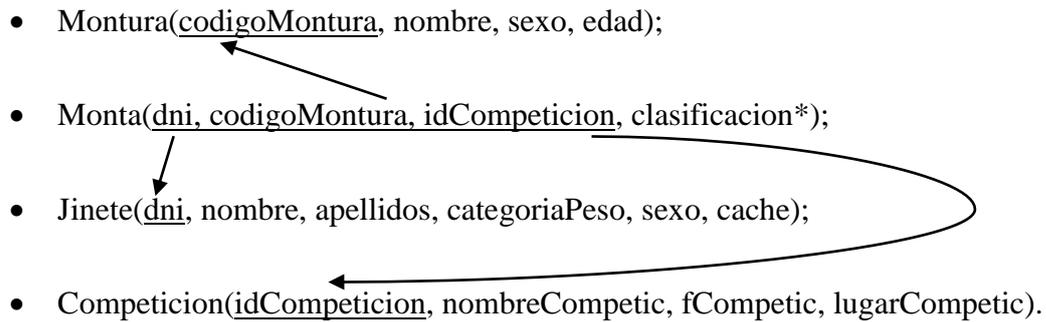
\* las tuplas que se insertan en la relación Equivalente\_De son semánticamente válidas, además por cada tupla que representa la equivalencia entre dos documentos, debería existir su contraria (si el documento A es equivalente al B, también el B es equivalente al A).

\* si el atributo EsAnotHumano es false el correo debe ser siempre NULL.

c.2) Restricciones de integridad que se pierden en la transformación:

- La restricción de participación mínima de Corpus respecto de la relación Contiene.
- La restricción de participación mínima de Documento respecto de la relación Almacena.
- La restricción de cardinalidad "un anotador sólo anotará como máximo una vez cada expresión de un documento (es decir, con una única etiqueta a lo sumo)".

2) (5,5 puntos) Dado el siguiente modelo relacional

- Montura(codigoMontura, nombre, sexo, edad);
  - Monta(dni, codigoMontura, idCompeticion, clasificacion\*);
  - Jinete(dni, nombre, apellidos, categoriaPeso, sexo, cache);
  - Competicion(idCompeticion, nombreCompetic, fCompetic, lugarCompetic).
- 

y las siguientes restricciones:

- La clasificacion de la relación Monta es un entero positivo que indica la posición ocupada finalmente por la montura en la competición.
- La categoría de peso de los jinetes se representa mediante los enteros siguientes: 1 ⇔ Menos de 60 kg; 2 ⇔ peso ∈ [60, 70) ; 3 ⇔ peso ∈ [70, 80) ; 4 ⇔ 80 kg o más.
- El cache de un Jinete es un número entero no negativo que indica lo que ingresa por carrera.
- El sexo se representa mediante un carácter:
  - en las monturas, con los caracteres 'M' ⇔ Macho, 'H' ⇔ Hembra;
  - en los jinetes, con los caracteres 'H' ⇔ Hombre, 'M' ⇔ Mujer.
- Los jinetes y sus monturas se inscriben en las competiciones de antemano, y el período de inscripción se abre una vez que se ha añadido una nueva entrada para la competición respectiva en la tabla Competicion. Al inscribirse,
  - el jinete y/o la montura tienen que darse de alta en el sistema previamente, si aún no ha(n) participado en ninguna competición;
  - se crea una nueva entrada en la relación Monta, con el atributo clasificacion inicializado a nulo, siempre que
    - tanto la montura y el jinete como la competición estén ya dados de alta en la base de datos (en sus tablas respectivas);
    - el jinete no haya sido dado de alta en esa misma competición con esa misma montura u otra distinta;
    - la montura no haya sido dada de alta en esa misma competición con ese mismo jinete u otro distinto.
- Excepto en los casos en los que indica lo contrario más arriba, ninguno de los atributos puede adoptar el valor nulo.

- a) (1 puntos) Escribe las sentencias SQL que permitan crear las tablas de su modelo físico, incluyendo las restricciones de integridad referencial y las otras restricciones de integridad descritas.

```

-- El orden de las tablas importa

CREATE TABLE Montura (
  codigoMontura VARCHAR(10) PRIMARY KEY
  , nombre VARCHAR(25) NOT NULL
  , sexo CHAR(1) NOT NULL
  , edad INTEGER NOT NULL CHECK (edad>0)
  , CONSTRAINT 'sexoCK1' CHECK (sexo IN ('H', 'M'))
);

CREATE TABLE Jinete (
  dni VARCHAR(10) PRIMARY KEY
  , nombre VARCHAR(25) NOT NULL
  , apellidos VARCHAR(100) NOT NULL
  , categoriaPeso INTEGER NOT NULL
  , sexo CHAR(1) NOT NULL
  , cache INTEGER NOT NULL CHECK (cache>=0)
  , CONSTRAINT 'catPesoCK' CHECK (categoriaPeso BETWEEN 1 AND 4)
  , CONSTRAINT 'sexoCK2' CHECK (sexo IN ('H', 'M'))
);

CREATE TABLE Competicion (
  idCompeticion VARCHAR(10) PRIMARY KEY
  , nombreCompetic VARCHAR(50) NOT NULL
  , fCompetic DATE NOT NULL
  , lugarCompetic VARCHAR(50) NOT NULL
);

-- La tabla Monta solo se puede crear al final
CREATE TABLE Monta (
  dni VARCHAR(10)
  , codigoMontura VARCHAR(10)
  , idCompeticion VARCHAR(10)
  , clasificación INTEGER DEFAULT NULL
  , PRIMARY KEY (dni, codigoMontura, idCompeticion)
  , FOREIGN KEY (dni) REFERENCES Jinete(dni)
  , FOREIGN KEY (codigoMontura) REFERENCES Montura(codigo)
  , FOREIGN KEY (idCompeticion) REFERENCES Competicion(idCompeticion)
  , CONSTRAINT 'clasificacionCK'
    CHECK ( clasificacion IS NULL OR clasificacion >= 1 )
);

```

**b) (0,5 puntos) Indica cuáles son las restricciones de integridad que no has podido incluir en las sentencias de creación de las tablas y cómo podrían implementarse.**

Las restricciones que no han podido incluirse en el apartado anterior son las siguientes comprobaciones requeridas para dar de alta a un jinete y su montura en una competición (relación Monta):

- el jinete no ha sido dado de alta en esa misma competición con otra montura distinta (si está dado de alta con la misma montura, la restricción de clave primaria se encarga de la comprobación);
- la montura no ha sido dada de alta en esa misma competición con otro jinete distinto (si está dado de alta con el mismo jinete, la restricción de clave primaria se encarga de la comprobación).

Existen varias formas de solucionar este problema, tales como usar

- **asertos** (aunque no están implementados en Oracle);
- **disparadores**.

- c) (0,5 puntos) Escribe una consulta eficiente en Álgebra Relacional para determinar el nombre y los apellidos de los jinetes (sean hombres o mujeres) que pesan entre 60 y 80 kg y que han ganado alguna competición.

$$\begin{aligned}
 R_1 &\leftarrow \sigma_{clasificacion=1}(Monta) \\
 R_2 &\leftarrow \prod_{dniJinete} (R_1) \\
 R_3 &\leftarrow \sigma_{(categoriaPeso=2) \vee (categoriaPeso=3)}(Jinete) \\
 R_4 &\leftarrow \prod_{dni, nombre, apellidos} (R_3) \\
 R_5 &\leftarrow R_2 \bowtie_{R_2.dni=R_4.dni} R_4 \\
 RES &\leftarrow \prod_{nombre, apellidos} (R_5)
 \end{aligned}$$

- d) (0,5 puntos) Escribe una sentencia SQL que permita subir un 5% el caché de todos los jinetes que hayan ganado alguna competición.

```

UPDATE Jinete
SET cache=cache*1.05
WHERE dni IN (SELECT DISTINCT DNI FROM Monta WHERE clasificacion=1);

```

- e) (3 puntos) Escribe consultas en SQL para determinar:

- a. (0,25 puntos) El número de competiciones celebradas en Madrid en 2014.

```

SELECT COUNT(*) AS NumeroCompeticMadrid2014
FROM Competicion
WHERE lugarCompetic = 'Madrid'
AND fCompetic BETWEEN '2014-01-01' AND '2014-12-31';

```

- b. (0,25 puntos) El caché medio de las mujeres jinetes en función de su categoría de peso.

```

SELECT categoriaPeso, AVG(cache) AS CacheMedio
FROM Jinete
WHERE sexo = 'M'
GROUP BY categoriaPeso;

```

- c. (0,5 puntos) El identificador y el nombre de la competición que ha tenido más participantes hasta la fecha, junto con el número de participantes que tuvo.

```

SELECT M.idCompeticion, C.nombreCompetic, COUNT(*) AS Participantes
FROM Monta M NATURAL INNER JOIN Competicion C
GROUP BY M.idCompeticion, C.nombreCompetic
HAVING COUNT(*) >= ALL ( SELECT COUNT(*)
FROM Monta
GROUP BY idCompeticion );

```

- d. (0,5 puntos) El nombre, los apellidos, la categoría de peso y el número de competiciones en las que han participado los jinetes masculinos que hayan concursado al menos en 100 competiciones pero no hayan ganado ninguna. Ordena los resultados por número de participaciones decreciente.

```
SELECT nombre, apellidos, categoriaPeso, COUNT(*) AS Participaciones
FROM Jinete NATURAL INNER JOIN MONTA
WHERE sexo='H'
GROUP BY dni, nombre, apellidos, categoriaPeso
HAVING COUNT(*) >= 100 AND dni NOT IN
    (SELECT DISTINCT dni
     FROM Monta
     WHERE clasificacion = 1)
ORDER BY Participaciones DESC;
```

o bien

```
SELECT nombre, apellidos, categoriaPeso, COUNT(*) AS Participaciones
FROM Jinete J NATURAL INNER JOIN MONTA
WHERE J.sexo='H'
GROUP BY dni, nombre, apellidos, categoriaPeso
HAVING COUNT(*) >= 100 AND
    NOT EXISTS (SELECT *
                FROM Monta M
                WHERE M.dni = J.dni AND M.clasificacion = 1)
ORDER BY Participaciones DESC;
```

- e. (0.75 puntos) El nombre, los apellidos, la categoría de peso y el caché de aquellas mujeres que superan el caché medio masculino de su categoría de peso.

```
SELECT J1.nombre, J1.apellidos, J1.categoriaPeso, J1.cache
FROM Jinete J1
WHERE J1.sexo='M' AND J1.cache >
    (SELECT AVG(J2.cache)
     FROM Jinete J2
     WHERE J2.sexo='H' AND
           J2.categoriaPeso = J1.categoriaPeso);
```

- f. (0.75 puntos) El caché medio de los jinetes que han hecho podio en una competición tanto con un caballo (macho), como con una yegua. PISTA: Utiliza para ello una subconsulta que obtenga el dni de todos los jinetes que cumplen dichas condiciones.

```
SELECT AVG(cache)
FROM JINETE J INNER JOIN
    (SELECT DISTINCT M1.dni
     FROM (Monta NATURAL INNER JOIN Montura) M1,
          (Monta NATURAL INNER JOIN Montura) M2
     WHERE M1.dni=M2.dni AND
           M1.clasificacion in (1,2,3) AND
           M2.clasificacion in (1,2,3) AND
           M1.sexo <> M2.sexo
     ) Q
WHERE J.DNI=M1.DNI;
```

-- Otra alternativa

```
SELECT AVG(cache)
  FROM JINETE
 WHERE DNI IN
      (SELECT DNI
        FROM Monta NATURAL INNER JOIN Montura
        WHERE clasificacion in (1,2,3) AND sexo='M')
 AND DNI IN
      (SELECT DNI
        FROM Monta NATURAL INNER JOIN Montura
        WHERE clasificacion in (1,2,3) AND sexo='H');
```