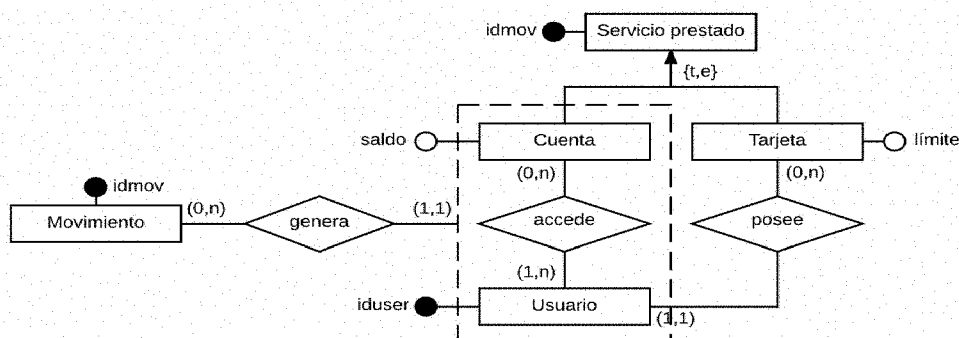


Grado en Ingeniería Informática
Diseño y Desarrollo de Sistemas de Información
Resolución del Examen de contenidos teóricos, Febrero de 2015

Enunciado

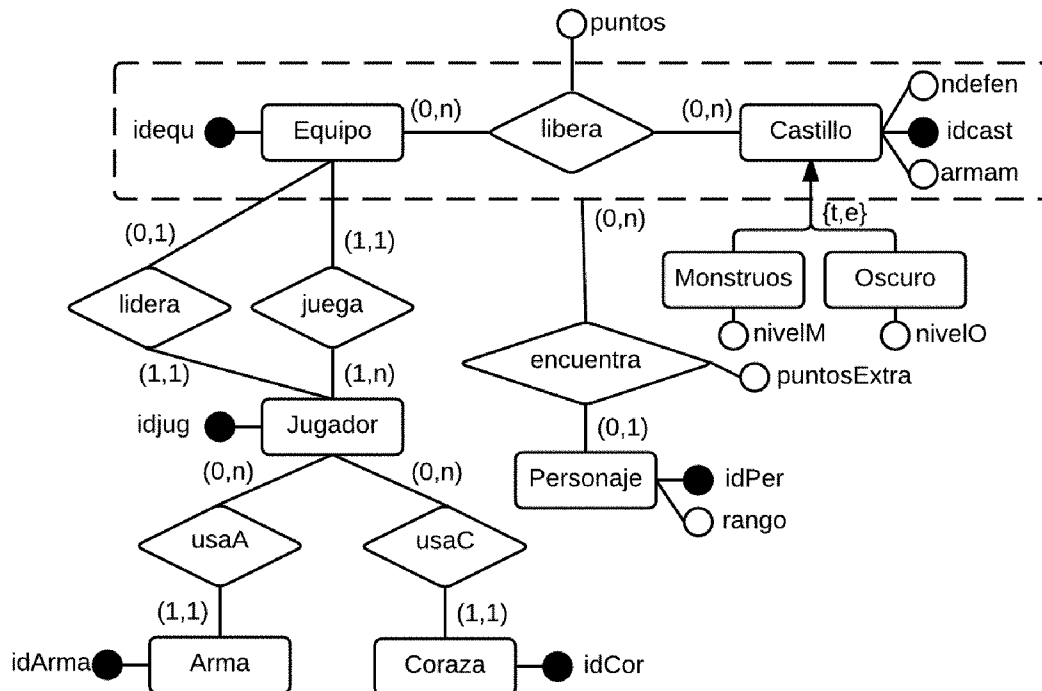
- (2,5 puntos)** En el presente ejercicio, se pretende desarrollar un Sistema de Información para un Videojuego por equipos llamado "Los reinos perdidos". Ambientado en un mundo de fantasía medieval, el juego consiste en intentar liberar castillos del mal, y se juega por equipos de participantes. Los equipos compiten en alcanzar la mayor puntuación posible con la liberación de castillos. Las particularidades del juego son las siguientes:
 - Varios equipos pueden coincidir en la liberación del mismo castillo.
 - De entre los jugadores de cada equipo habrá un líder que los dirigirá, y que va rotando entre ellos ante cada castillo a tomar.
 - Cada jugador escogerá un tipo de arma y coraza para los combates.
 - Hay dos tipos de castillos: castillos Oscuros y castillos Monstruos, que tienen características diferentes, unos niveles de oscuridad y los otros monstruos. De los castillos se conoce cantidad de defensores, y tipo de armamento.
 - El equipo al intentar tomar un castillo es posible que encuentre un personaje especial que habrá que liberar. Este personaje se caracteriza por rango de importancia. Su liberación la pueden hacer uno o varios equipos que luchan juntos, en un castillo determinado y en un tiempo de ataque. Este tiempo en liberarlo es importante porque acorde a ello es la puntuación extra que ganaría el equipo o equipos que lo libere.
- (2,5 puntos)** Redacta una descripción completa para el siguiente diagrama E/R:



- (2 puntos)** Realizar el paso a Modelo Relacional (paso a tablas) del diagrama E/R del ejercicio 2.
- (1,5 puntos)** Sea R una relación de esquema {A, B, C, D, E, F} que verifica el conjunto F de dependencias funcionales { $AD \rightarrow B$, $AF \rightarrow DE$, $E \rightarrow F$ }. Calcular todas las claves candidatas existentes en la relación R y justifica por qué son las únicas existentes.
- (1,5 puntos)** Sea R una relación de esquema {A, B, C, D, E}, con un conjunto CK de claves candidatas {AB, BC} y que verifica un conjunto F de dependencias funcionales { $B \rightarrow D$, $AD \rightarrow C$, $CD \rightarrow A$, $D \rightarrow E$ }. Comprobar si la relación está en Forma Normal de Boyce y Codd (sin comprobar si está en segunda forma normal o en tercera forma normal) y si no lo está, realizar una descomposición hasta que todas las relaciones de la descomposición estén en dicha forma normal.

Resolución

Ejercicio 1



Ejercicio 2

Los **servicios prestados** por la organización, cuyos datos almacena el sistema, se identifican por un identificador *idmov*, y son necesariamente de uno de dos tipos:

- **Cuenta**, que tiene además un *saldo*, y
- **Tarjeta**, que tiene además un *límite*.

Un **usuario** se identifica por un atributo *iduser*, y un **movimiento** se identifica por un atributo *idmov*.

Algunas restricciones adicionales en la información del sistema son:

- Un **usuario** puede poseer varias **tarjetas**, aunque una **tarjeta** pertenece a un único **usuario**.
- A una **cuenta** acceden varios **usuarios** y un **usuario** puede acceder a varias **cuentas**.
- Cuando un **usuario** accede a una **cuenta**, puede generar varios **movimientos**. Aunque un **movimiento** corresponde al acceso concreto de un **usuario** a una **cuenta**.

Ejercicio 3

El diseño lógico relacional correspondiente a este diagrama entidad-relación:

- Usuario (**iduser**)
- ServicioPrestado (**idmov**)
- Cuenta (**idmov**, saldo), con *idmov* llave externa a la tabla *ServicioPrestado*
- PoseeTarjeta (**idmov**, limite, iduser), con *idmov* llave externa a la tabla *ServicioPrestado* e *iduser* con una restricción *Not Null* sobre el atributo y llave externa a la tabla *Usuario*
- Accede (**iduser**, **idmov**), con *iduser* llave externa a la tabla *Usuario* e *idmov* llave externa a la tabla *Cuenta*

- GeneraMovimiento (**idmov**, iduser, idmov), con el par de atributos (*iduser*, *idmov*) con una restricción *Not Null* y llave externa a la tabla *Accede*.

Ejercicio 4

En primer lugar, sería recomendable establecer los tipos de atributos en función de si aparecen a la izquierda en todas las dependencias funcionales en las que aparece, a la derecha en todas las dependencias funcionales en las que aparece, a la izquierda de alguna y a la derecha en otra o en ninguna dependencia funcional:

- atributos independientes (no aparecen en ninguna dependencia): {C}
- atributos que aparecen a la izquierda en todas las dependencias en las que aparecen: {A}
- atributos que aparecen a la izquierda de alguna dependencia funcional y a la derecha de otra: {D, E, F}
- atributos que aparecen a la derecha en todas las dependencias en las que aparecen: {B}

Establecidos los tipos, podemos aplicar el algoritmo de extracción de claves candidatas:

- 1º Eliminación de atributos independientes:

Al haber ningún atributo independiente, la relación sin independientes queda como sigue $R_{SI}=R-\{C\}=\{A, B, D, E, F\}$

- 2º Eliminación de equivalencias:

Al no haber ninguna equivalencia entre parejas de atributos, $R_{SIE}=R_{SI}$ y $F_{SIE}=F$

- 3º Procesamiento de atributos que aparecen sólo a la izquierda:

$K_p=\{A\}=A$

Habiendo un candidato, tenemos que probar si es una clave:

$A^+=\{A\} \neq R_{SIE}$, $CK_{SIE}=\emptyset$ y, al no haber obtenido una clave en el paso 3, pasamos al paso 4

- 4º Procesamiento de atributos que aparecen a la izquierda y a la derecha:

Combinamos el candidato del paso 3 con cada atributo que esté a la izquierda y la derecha, y que no aparezca en el cierre de dicho candidato, para generar $K_p'=\{AD, AE, AF\}$

$AD^+=\{A, D, B\} \neq R_{SIE}$, $K_p'=\{AE, AF\}$, $CK_{SIE}=\emptyset$ y, al no haber obtenido una clave, tenemos que combinar el candidato con cualquier atributo que esté a la izquierda y a la derecha y que no aparezca en el cierre de dicho candidato; sin embargo, ADE es una extensión del candidato AE, por lo que no se incluye en el conjunto de candidatos, y ADF es una extensión de AF, por lo que no se incluye en el conjunto de candidatos.

$AE^+=\{A, E, F, D, B\}=R_{SIE}$, luego hemos encontrado una clave candidata para R_{SIE} lo que nos deja $K_p'=\{AF\}$ y $CK_{SIE}=\{AE\}$.

$AF^+=\{A, F, D, E, B\}=R_{SIE}$, luego hemos encontrado una clave candidata para R_{SIE} lo que nos deja $K_p'=\emptyset$ y $CK_{SIE}=\{AE, AF\}$.

Dado que no quedan candidatos, el paso 4 ha terminado.

- 5º Incorporación de atributos independientes:

Puesto que existían atributos independientes en R, es necesario incorporarlos a todas las claves, por lo que $CK'=\{ACE, ACF\}$

- 6º Incorporación de atributos equivalentes:

Al no haber atributos equivalentes, no se modifican las claves descubiertas, por lo que

$CK=\{ACE, ACF\}$

El hecho de haber aplicado el algoritmo de extracción de claves candidatas, garantiza que se han encontrado las únicas que hay.

Ejercicio 5

$R=\{A, B, C, D, E\}$, $CK=\{AB, BC\}$ y $F=\{B \rightarrow D, AD \rightarrow C, CD \rightarrow A, D \rightarrow E\}$

Podemos ver que R no está en Forma Normal de Boyce y Codd (BCNF) porque $B \rightarrow D$, $AD \rightarrow C$, $CD \rightarrow A$ y $D \rightarrow E$ son dependencias de F y B, AD, CD y D no son claves candidatas de CK.

Para normalizar, debemos aplicar el Teorema de Heath sobre una de las dependencias funcionales de F que no cumple la forma normal. A la hora de elegir, necesitamos saber qué atributo que está a la derecha puede causar un daño menor si se aplica el Teorema de Heath sobre dicho atributo:

- dependencias cuyo atributo a la derecha sólo está a la derecha de cualquier dependencia funcional ($\{E\}$): $\{D \rightarrow E\}$
- dependencias cuyo atributo a la derecha está a la izquierda de otra dependencia:
 - y que el atributo a la derecha no pertenezca a ninguna clave candidata ($\{D\}$): $\{B \rightarrow D\}$
 - y que el atributo a la derecha pertenezca a alguna clave candidata ($\{C, A\}$): $\{AD \rightarrow C, CD \rightarrow A\}$

En primer lugar, aplicamos el Teorema de Heath sobre la dependencia con E a la derecha:

$R_1 = \{D, E\}$, $F_1 = \{D \rightarrow E\}$, $CK_1 = \{D\}$, **R_1 está en BCNF** porque su clave candidata es D y todas sus dependencias tienen una clave candidata a la izquierda.

$R_2 = \{A, B, C, D\}$, $F_2 = \{B \rightarrow D, AD \rightarrow C, CD \rightarrow A\}$, $CK_2 = \{AB, BC\}$, R_2 no está en BCNF por culpa de la dependencias $B \rightarrow D$, $AD \rightarrow C$, $CD \rightarrow A$, y tenemos que aplicar de nuevo el Teorema de Heath para normalizar. Para aplicar dicho teorema, escogemos la dependencia $B \rightarrow D$, con lo que queda:

$R_{2,1} = \{B, D\}$, $F_{2,1} = \{B \rightarrow D\}$, $CK_{2,1} = \{B\}$, **$R_{2,1}$ está en BCNF** porque su única clave candidata es B y todas sus dependencias tienen una clave candidata a la izquierda.

$R_{2,2} = \{A, B, C\}$, $F_{2,2} = \emptyset$, $CK_2 = \{AB, BC\}$, **$R_{2,2}$ está en BCNF** porque todas sus dependencias funcionales (ninguna) tienen una clave candidata a la izquierda.

El resultado de nuestra normalización es el conjunto de relaciones:

$$\{(\{D, E\}, r_1), (\{B, D\}, r_{2,1}), (\{A, B, C\}, r_{2,2})\}$$

Aunque el ejercicio no requiere especificar si se ha perdido alguna dependencia funcional, es fácil ver que la dependencias $AD \rightarrow C$ y $CD \rightarrow A$ parecen no estar en ningún conjunto de dependencias del resultado de la normalización. Sin embargo, quizás pueden volver a obtenerse a partir de las demás. Para comprobarlo, basta con probar si C está en el conjunto AD^+ generado a partir de las dependencias que quedan en alguno de los conjuntos de dependencias del resultado, y hacer lo propio con A con respecto al conjunto CD^+ :

- $AD^+ = \{A, D, E\}$, que no incluye a C, por lo que la dependencia funcional se ha perdido efectivamente
- $CD^+ = \{C, D, E\}$, que no incluye a A, por lo que la dependencia funcional se ha perdido efectivamente