

# IBD 2011 – Curso de Recursantes

## Tercera Fecha de Examen Parcial (1/12/11)

### 1. Modelado Conceptual

Se debe modelar un servicio de mensajería electrónica.

Los *usuarios* se caracterizan por un número único, un nombre, un path opcional a una foto, y una dirección electrónica alternativa opcional.

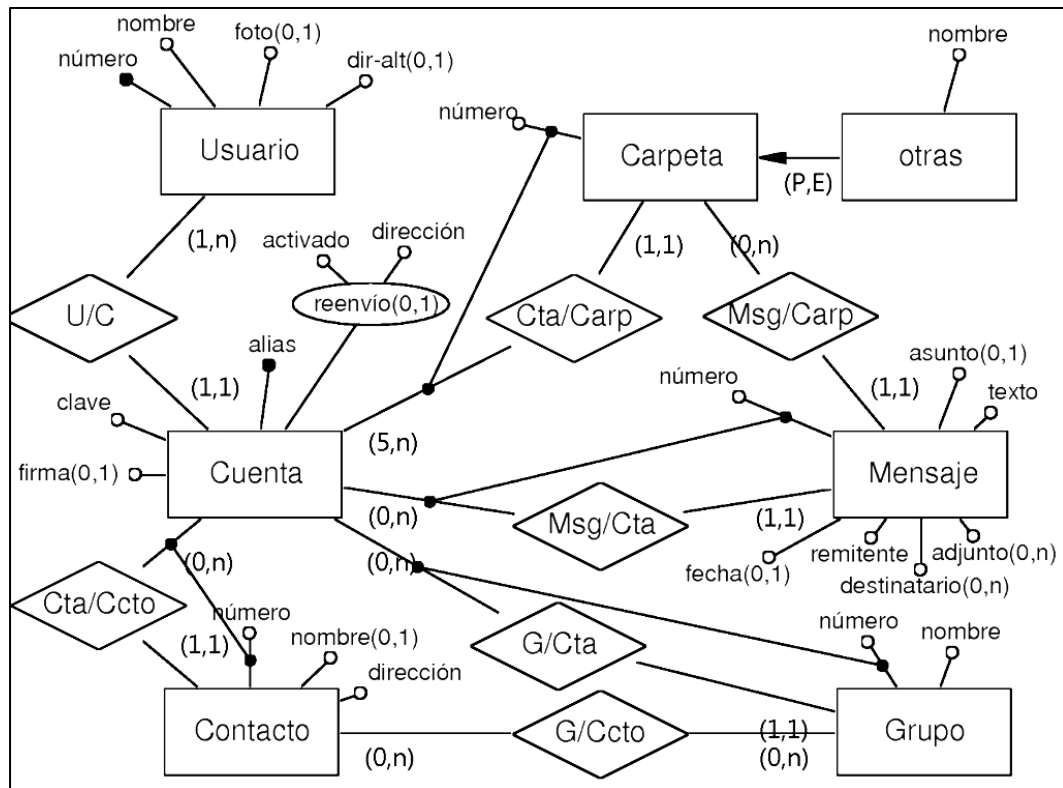
Las *cuentas* se caracterizan por el alias (nickname) del usuario propietario (único), la clave de acceso, un texto opcional para firma de mensajes, y una dirección electrónica opcional para reenvío de mensajes junto con la indicación de si el reenvío está activado o no. Un usuario puede tener más de una cuenta.

Las *carpetas de cuenta* se caracterizan por un número relativo a la cuenta (se numeran a partir de 1 para cada cuenta) y por un nombre opcional (las 5 primeras carpetas de cada cuenta serán siempre Recibidos, Enviados, Borradores, Spam y Papelera, y su número relativo determina su nombre por defecto, que no se registra, mientras que las carpetas que cree el propietario de la cuenta deben tener un nombre distintivo).

Los *contactos de cuenta* se caracterizan por un número relativo a la cuenta, el nombre opcional del propietario de la dirección y la dirección electrónica. Los contactos pueden estar organizados en *grupos* definidos para la cuenta, caracterizados por un número relativo a la cuenta y un nombre; un contacto puede no pertenecer a ningún grupo o pertenecer a más de uno.

Los *mensajes* se caracterizan por un número relativo a la cuenta, el texto, un asunto, la fecha de envío, la dirección electrónica del remitente, las direcciones electrónicas de los destinatarios (al menos una) y, opcionalmente, los paths de almacenamiento de archivos adjuntos (el asunto, la fecha y las direcciones pueden no tener valores asignados cuando el mensaje está asociado a la carpeta borradores). Todo mensaje debe estar asociado a una carpeta (recibidos, enviados, borradores, spam, papelera o otra carpeta creada por el usuario) correspondiente de la cuenta a la que pertenece.

**Nota:** verificar que cada entidad tenga **al menos** un identificador y la cardinalidad en **cada** relación que participe y que los atributos no obligatorios o no monovalentes tengan correctamente marcada su cardinalidad (cada omisión se considerará un error independiente, lo mismo que la no generalización o especialización cuando sea posible y la no denotación de identificadores externos o mixtos cuando sean naturales).



La especialización de carpetas en el subconjunto "Otras" con el nombre obligatorio, es equivalente a poner el nombre como opcional en Carpeta; también es admisible una especialización en dos subconjuntos (p.e. "Fijas", sin nombre, y "Otras", con nombre), o una especialización en seis subconjuntos (uno para cada carpeta fija de cuenta, sin nombre, más el subconjunto para las demás, con nombre). No es erróneo modelar subconjuntos de especialización sin atributos o relaciones propios, ya que implica mayor expresividad.

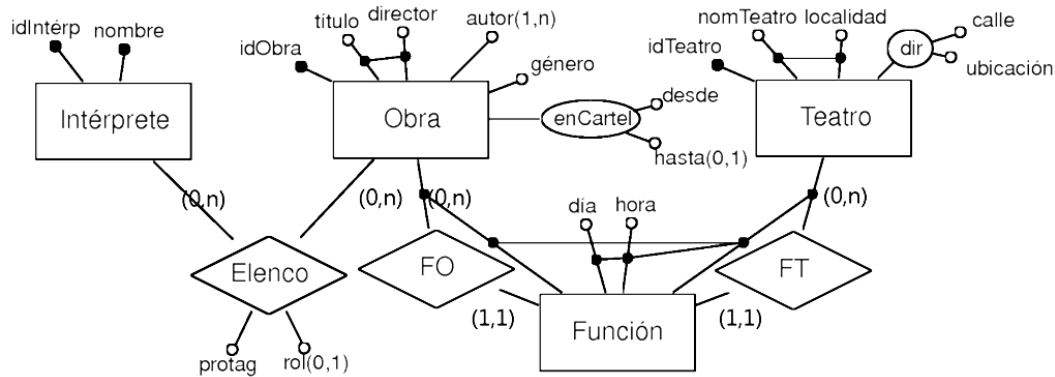
Entre Cuenta, Contacto y Grupo no hay un ciclo redundante por las cardinalidades de Contacto y Grupo en G/Ccto.

Los atributos de Mensaje pueden ser todos opcionales, excepto número.

**Notas: se aprueba con un mínimo de 6 (de 10) puntos, y**

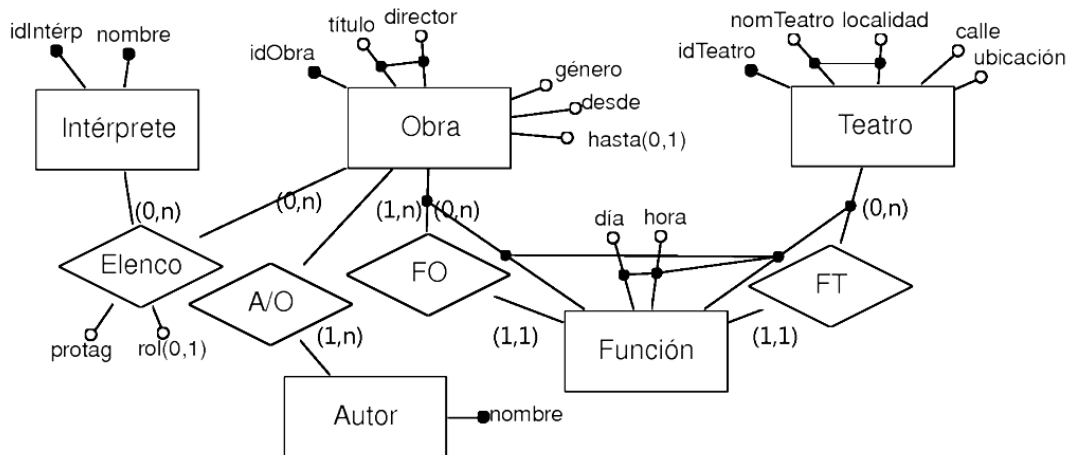
- La omisión de cualquier entidad o relación descuenta **2** puntos.
- La omisión o definición incorrecta (*por ejemplo, una cardinalidad distinta de **1,1** de la entidad débil en la relación con la que compone su identificador*) de cualquiera de los identificadores mixtos descuenta **1** punto .
- Cada error en de cardinalidad máxima de una entidad en una relación, así como una cardinalidad mínima distinta de 5 de Cuenta en Cta/Carp, descuenta  $\frac{1}{2}$  punto.
- La omisión de la cardinalidad de una entidad en una relación descuenta  $\frac{1}{2}$  punto.
- La no composición de los atributos de reenvío (Cuenta) descuenta  $\frac{1}{2}$  punto.
- La no denotación de opcionalidad en un atributo que tenga ese carácter descuenta  $\frac{1}{4}$  punto.
- La omisión de un atributo especificado en el enunciado o el agregado de uno que no se especifica descuenta  $\frac{1}{4}$  punto.

## 2. Modelos Lógico y Físico



Dado el diagrama conceptual de arriba, derivar un diagrama lógico y de éste derivar las tablas que le correspondan (indicando claves primarias y, cuando corresponda, claves candidatas y externas).

### Diagrama Lógico



Se eliminan los atributos compuestos enCartel (Obra) y dir (Teatro), y el polivalente autor (Obra).

La transformación de atributos compuestos en un único atributo simple es admisible, aunque no recomendada.

La conversión del atributo polivalente autor en una entidad dependiente de Obra y con identificador mixto de Obra más el nombre del autor es admisible (lo proponen Batini, Ceri y Navathe en su libro "Diseño Conceptual de Bases de Datos").

### Tablas

**Nota:** los atributos subrayados conforman claves primarias; fk (foreign key) denota claves foráneas o externas (también se pueden denotar con "cf" o "ce"); ak (alternative key) denota claves candidatas o alternativas (también se pueden denotar con "cc" o "ca"); el signo de cierre de interrogación denota opcionalidad (no es obligatorio denotarla).

Obra(idObra, (título, director)ak, género, desde, (hasta)?)

Autor(nombre)

A/O((idObra)fk, (nombre)fk)

Teatro(idTeatro, (nomTeatro, localidad)ak, calle, ubicación)

Función((idObra)fk, (idTeatro)fk, día, hora)

Intérprete(idIntérp, (nombre)ak)

Elenco((idIntérp)fk, (idObra)fk, protag, (rol)?)

**Nota:** las tablas de entidad pueden tener claves primarias automáticas, pero los identificadores originales deben marcarse como claves candidatas. Se aprueba con un mínimo de 6 (de 10) puntos y se descuenta 1 punto por cada transformación incorrecta en el modelo lógico y por cada clave primaria mal definida en una tabla; por cada clave externa o candidata no denotada o incorrecta se descuenta ½ punto.

3. Álgebra Relacional y SQL (se aprueba con al menos 4,5p en AR y 6p en SQL – 2 puntos c/op. para c/lenguaje)

**Posgrado**(codCarr, nomCarr), **Asignatura**(codCarr, codA, nomA)

**Alumno**(nroLeg, dni, nomAl, fNac, dom, tel, e-mail, (codCarr)fk)

**PromAsign**((nroLeg)fk, ((codCarr)fk, codA)fk, año, fecha, nota) // registra la aprobación de asignaturas (promoción)

**Nota: los errores de sintaxis en SQL no descuentan puntaje, siempre que la operación quede comprensible.**

- a. Registrar al alumno con DNI 88.888.888 y nomAl “González, Silvia Isabel” con nroLeg 7777 en el posgrado “Maestría en Ingeniería de Software” (el resto de los atributos se dejan con valor nulo).

$\text{Alumno} \leftarrow \text{Alumno} \cup \{(7777, 88.888.888, \text{'González, Silvia Isabel'}, \text{nulo}, \text{nulo}, \text{nulo}, \pi_{\text{codCarr}} (\sigma_{\text{nomCarr} = \text{'Maestría en Ingeniería de Software'}} (\text{Posgrado})))\}$

INSERT INTO Alumno (nroLeg, dni, nomAl, codCarr)

VALUE (7777, 88.888.888, 'González, Silvia Isabel', SELECT codCarr FROM Posgrado WHERE nomCarr = 'Maestría en Ingeniería de Software');

- b. Registrar fNac: “12-11-68” y e\_mail: “silvia.gonzalez@gmail.com” al alumno con dni 88.888.888.

$\delta_{\text{fNac} \leftarrow \text{'12-11-68'}, \text{e\_mail} \leftarrow \text{'silvia.gonzalez@gmail.com'}} (\sigma_{\text{dni} = 88.888.888} (\text{Alumno}))$

UPDATE Alumno

SET fNac = '12-11-68', e\_mail = 'silvia.gonzalez@gmail.com'

WHERE dni = 88.888.888;

- c. Listar nroLeg, dni y nomAl de los alumnos que hayan promocionado todas las asignaturas del posgrado “Esp. en Redes y Seguridad”. En SQL ordenar por nroLeg.

$\pi_{\text{nroLeg, dni y nomAl, codCarr}} (\text{PromAsign} \bowtie \text{Alumno}) \div \pi_{\text{codCarr, codA}} (\sigma_{\text{nomCarr} = \text{'Esp. en Redes y Seguridad'}} (\text{Posgrado} \bowtie \text{Asignatura}))$

**o bien:**

$\pi_{\text{nroLeg, dni y nomAl}} ((\text{PromAsign} \bowtie \text{Alumno}) \div \pi_{\text{codCarr, codA}} (\sigma_{\text{nomCarr} = \text{'Esp. en Redes y Seguridad'}} (\text{Posgrado} \bowtie \text{Asignatura})))$

**Nota:** también sería correcto en el *dividendo*

$\text{PromAsign} \bowtie \sigma_{\text{nomCarr} = \text{'Esp. en Redes y Seguridad'}} \text{Posgrado} \bowtie \text{Alumno}$  o el equivalente

$\text{PromAsign} \bowtie \sigma_{\text{nomCarr} = \text{'Esp. en Redes y Seguridad'}} (\text{Posgrado} \bowtie \text{Alumno})$  con cualquiera de las variantes previas pero **siempre con**

$\pi_{\text{codCarr, codA}} (\sigma_{\text{nomCarr} = \text{'Esp. en Redes y Seguridad'}} (\text{Posgrado} \bowtie \text{Asignatura}))$  como *divisor*.

SELECT nroLeg, dni y nomAl

FROM Alumno A

WHERE NOT EXISTS

(SELECT \*

FROM Asignatura A NATURAL JOIN Posgrado P

WHERE nomCarr = 'Esp. en Redes y Seguridad' AND NOT EXISTS

(SELECT \*

FROM PromAsign PA

WHERE PA.codCarr = P.codCarr AND PA.codA = A.codA AND PA.nroLeg = A.nroLeg))

**ORDER BY nroLeg;**

(alumnos tales que no exista asignatura del posgrado en cuestión que no hayan promocionado)

**o bien:**

SELECT nroLeg, dni y nomAl

FROM PromAsign NATURAL JOIN Alumno NATURAL JOIN Posgrado

WHERE nomCarr = 'Esp. en Redes y Seguridad'

GROUP BY nroLeg, dni y nomAl

HAVING COUNT(\*) = (SELECT COUNT(\*) FROM Asignatura NATURAL JOIN Posgrado WHERE nomCarr = 'Esp. en Redes y Seguridad')

**ORDER BY nroLeg;**

(alumnos cuya cantidad de asignaturas promocionadas del posgrado en cuestión sea igual a la cantidad total de las asignaturas de ese posgrado)

**Nota: la omisión del order by implica un descuento de 0,5 puntos.**

- d. Listar todas las asignaturas del posgrado en el que está inscripto el alumno con dni 22.222.222 junto con la fecha y nota de aprobación de las que haya promocionado. *En SQL ordenar por codA.*

$\pi_{\text{codA}, \text{nomA}, \text{fecha}, \text{nota}} (\sigma_{\text{dni} = 22.222.222} (\text{Alumno})) \mid \times \mid \text{Asignatura} \times \mid \text{PromAsign}$

(“ $\times$ ” implica producto izquierdo)

**Nota: la omisión de la proyección implica el descuento de 0,5 puntos; el uso innecesario de la tabla Posgrado también implica el descuento de 0,5 puntos**

*Sin usar producto izquierdo, la operación sería:*

$\text{Todas} \leftarrow \pi_{\text{nroLeg}, \text{codCarr}, \text{codA}, \text{nomA}} (\sigma_{\text{dni} = 22.222.222} (\text{Alumno})) \mid \times \mid \text{Asignatura}$

$\text{NoP} \leftarrow \pi_{\text{codCarr}, \text{codA}, \text{nomA}} (\text{Todas} - \pi_{\text{nroLeg}, \text{codCarr}, \text{codA}, \text{nomA}} (\text{Todas} \mid \times \mid \text{PromAsign})) \times \{(\text{nulo}, \text{nulo})\}$

$\pi_{\text{codCarr}, \text{codA}, \text{nomA}} (\text{Todas} \mid \times \mid \text{Promoción}) \cup \text{NoP}$

*(promocionadas  $\cup$  no promocionadas con fecha y nota nulas)*

**Nota: la omisión de la resta en NoP implica el descuento de 0,5 puntos; el uso innecesario de la tabla Posgrado también implica el descuento de 0,5 puntos**

SELECT codCarr, codA, nomA, fecha, nota

FROM Alumno A NATURAL JOIN Asignatura M LEFT JOIN PromAsign ON A.codCarr = M.codCarr

WHERE dni = 22.222.222

ORDER BY codA;

**Nota: la omisión del order by implica un descuento de 0,5 puntos; el uso innecesario de la tabla Posgrado también implica el descuento de 0,5 puntos**

- e. *Sólo SQL:* listar nomCarr, nomAl y el promedio de notas de las promociones de asignaturas del año 2010, de alumnos que tengan promedio igual o superior a 8.

SELECT nomCarr, nomAl, AVG(nota)

FROM PromAsign NATURAL JOIN Alumno NATURAL JOIN Posgrado

WHERE año = 2010

GROUP BY nomCarr, nroLeg, nomAl

HAVING AVG(nota) >= 8;

**Nota: no es indispensable incluir nroLeg en la cláusula ORDER BY**