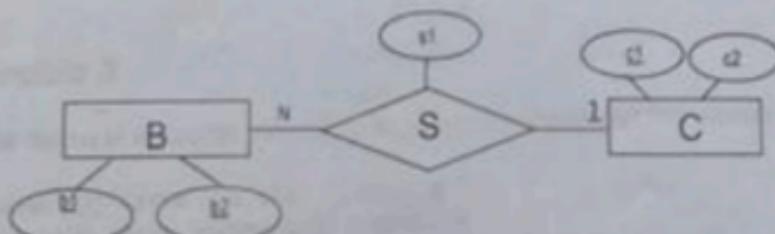


Ejercicio 1

Indicar para cada una de las afirmaciones siguientes **Verdadero o Falso y justificar** (explicar claramente y ejemplificar con instancias de esquemas, etc.):

1.1) "Cuando hay una relación de participación TOTAL entre 2 tipos de entidades es seguro que una de ellas es DEBIL"

1.2) "La relación binaria S por ser N:1 puede mapearse sin tener que armar una tabla aparte de las tablas B y de C, es decir puede embeberse. El mecanismo consiste en agregar a la tabla C los atributos clave de la entidad B"



1.1) Falso pues puede tener participacion total sin ser una entidad debil

1.2) Falso pues habria que meterla en la clase B

Ejercicio 2

Se ha creado la siguiente tabla

```
create table opinion
(
    username text NOT NULL PRIMARY KEY,
    rating integer NOT NULL,
    age integer NOT NULL
);
```

Y se tiene la siguiente instancia:

| username | opinion | |
|----------|---------|-----|
| | rating | age |
| Leti | 7 | 44 |
| Jose | 10 | 34 |
| Luis | 8 | 60 |
| Leti22 | 10 | 48 |
| Maria | 8 | 60 |
| Ceci | 7 | 21 |
| Ceci111 | 7 | 33 |
| Pablo | 7 | 39 |

2.1) Mostrar el resultado que se obtiene (header y tuplas) al ejecutar la siguiente consulta

```
SELECT S.username
FROM opinion S
WHERE S.rating >= ALL (SELECT S2.rating
                        FROM opinion S2
                        WHERE S2.rating <> (SELECT MAX(S3.rating)
                                              FROM opinion S3 )
)
AND S.rating <> (SELECT MAX(S3.rating) FROM opinion S3 )
```

| username |
|----------|
| Luis |
| Maria |

quiero los username de los que
hayan puesto el segundo mayor rating

2.2) Mostrar el resultado que se obtiene (header y tuplas) al ejecutar la siguiente consulta

```
SELECT S.rating , MIN( S.age ) AS calculated
FROM opinion S
GROUP BY S.rating
HAVING 1 <(SELECT COUNT(*) FROM opinion S2 WHERE S.rating = S2.rating)>
ORDER BY S.rating DESC
```

| rating | calculated |
|--------|------------|
| 10 | 34 |
| 8 | 60 |
| 7 | 33 |

quiero rating que se hayan dado mas de
una vez con la menor edad que lo dio,
ordenados por rating de manera descendiente

Ejercicio 3

Se tienen el siguiente esquema de relación. Ningún atributo puede ser null.

empleado(id, name, title, city) — clave id

proyecto(id, name, budget, city) — clave id

asignacion(emplID, proyID, duration) -- clave emplID + proyID

Las tablas ya están creadas.

Si se ejecutara la siguiente consulta del cálculo relacional de dominio se obtendrían ciertas tuplas.

{ e, p, d | (\exists en, r, pn, b, c) (empleado(e, en, r, c) \wedge proyecto(p, pn, b, c) \wedge d = 0) }

Escribir la sentencia SQL que permita insertar en la tabla **asignacion** las tuplas que se obtendría de la ejecución de la consulta del cálculo mostrada. Es decir, traducir la consulta del cálculo a SQL y usarla para expresar una sentencia SQL que inserte en la tabla **asignacion** esos valores.

quiero e, p, d tal que hay un empleado con e=id, en=name, r=title, c=city y que haya estado en el proyecto con p=id, pn=name, c=city y que d = 0

quiero id del empleado con el id del proyecto tal que estan en la misma ciudad, con una columna extra d que sea 0.

INSERT INTO asignacion(emplID, proyID, duration)

SELECT e.id, p.id, 0

FROM empleado e, proyecto p

WHERE e.city = p.city

Ejercicio 4

Dado el esquema de relación R(A, B, C, D, E, F) con Dep = { AB → B, B → E, A → C, CD → EF }
Demostrar con Axiomas de Armstrong que BC → F puede inferirse. De no poder demostrarse
proponer una instancia que satisfaga Dep pero no BC → F

(BC)+ = { B, C, E } luego como F no esta en (BC)+, no es cierto que BC → F

Contraejemplo

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | 2 | 3 | 0 | 5 | 0 |

$$AB \rightarrow B \quad \checkmark$$

$$B \rightarrow E \quad \checkmark$$

$$A \rightarrow C \quad \checkmark$$

$$CD \rightarrow EF \quad \checkmark$$

Vemos que BC → F no se cumple pues ambas tuplas coinciden en B y C, mas no en F

Ejercicio 5

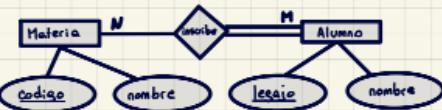
5.1) Enumerar todos los tipos de participaciones que pueden aparecer en un DER. Definirlos y ejemplificar.

participacion total: ejemplo una entidad debil

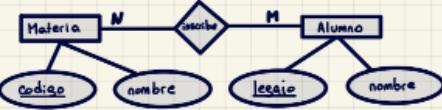
participacion parcial: cualquier cosa, tlt ahi pego la de nash

En un Diagrama de Entidad/Relación (DER) la participación puede ser:

▷ total: una entidad debe participar en al menos una ocurrencia de la relación, es decir que cada instancia (o fila) está obligada a estar asociada con al menos una instancia de otra entidad a través de una relación. Supongamos una universidad donde no pueden haber alumnos no inscriptos a ninguna materia:



▷ parcial: una entidad puede participar en una ocurrencia de la relación. Para el caso anterior, supongamos que ahora sí puede haber alumnos no inscriptos. Entonces tenemos:



con el primer ejemplo de nash ya se estan ejemplificando ambas

Materia puede existir por mas de no tener ningun alumno anotado (participacion parcial)

Alumno debe estar asociado a alguna materia (participacion total)

~~5.2)~~

Sean las siguientes 2 tablas

```
CREATE TABLE geo
(
    province TEXT,
    country TEXT,
    area FLOAT, -- km2
    primary key(province, country)
);
```

```
CREATE TABLE pais
(
    country TEXT,
    currency TEXT,
    primary key(country)
);
```

Un ejemplo de instancia válida sería:

| GEO | | |
|----------|-----------|--------|
| province | country | area |
| Salta | Argentina | 155488 |
| La Rioja | Argentina | 89680 |
| La Rioja | España | 5045 |

| PAIS | |
|-----------|----------|
| country | currency |
| Argentina | peso |
| España | euro |

Ambas tablas tienen la columna **country**. Se quiere agregar una clave foránea desde la tabla **pais** hacia la tabla **geo**, obviamente en el atributo **country**. No agregar más columnas de las ya existentes. ¿Es esto posible? Si lo es, escribir con una sentencia **ALTER TABLE PAIS ...** la modificación solicitada. De no ser posible explicar el problema.

No es posible pues una fk debe hacer referencia a una clave candidata. Osea como **country** no es clave, no puede ser clave foranea.

~~5.3)~~ En teoría de normalización, ¿a qué se refiere con **anomalia de actualización?** Explicar detalladamente

Anomalia de actualización se da cuando al hacer un **UPDATE** de algun valor, haya que actualizar este en varias tuplas, siendo que estas pueden estar en distintos bloques del disco, empeorando la performance.