

Nombre:.....

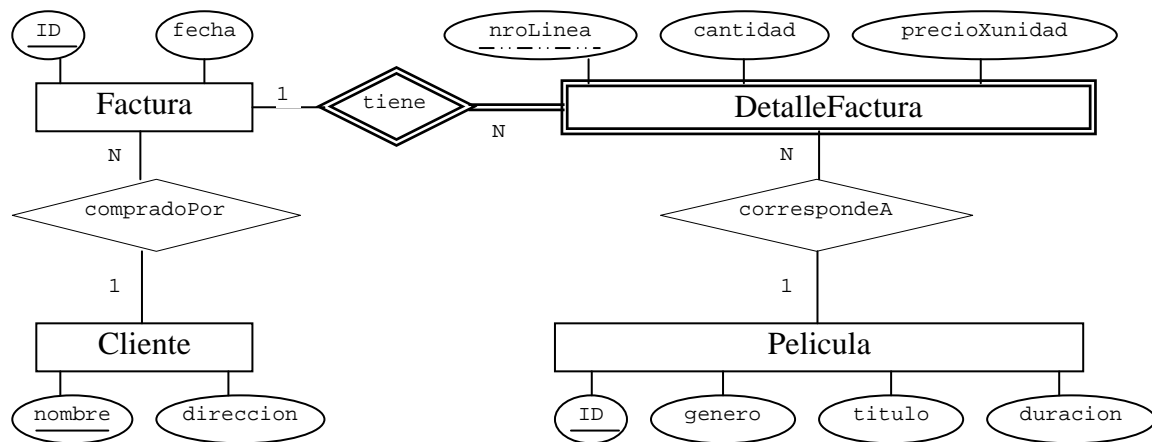
Legajo:.....

## Primer Parcial de Base de Datos I

Ej. 1	Ej. 2	Ej. 3	Ej. 4	Ej. 5	Nota
/2	/2.4	/0.8	/3.3	/1.5	

### Ejercicio 1

El siguiente diagrama de E/R representa las películas que graba la Warner Bros y su comercialización en el mercado.



#### 1.1)

Escribir las sentencias SQL que permitan crear las tablas y todas las restricciones mostradas en el diagrama. No asumir nada que no esté indicado en el diagrama E/R.

#### 1.2)

Indicar para cada una de las afirmaciones siguientes **V** o **F** y **justificar** (explicar claramente y ejemplificar con instancias de esquemas, etc.) . No asumir nada que no aparezca en el diagrama de E/R.

- 1.2.1) Una misma película no puede haberse vendido a dos precios diferentes.
- 1.2.2) No puede aparecer en una factura el registro de dos clientes distintos.
- 1.2.3) Si un cliente cambia de dirección las facturas pasadas quedan automáticamente con la nueva dirección.
- 1.2.4) Los números de líneas no pueden repetirse para facturas distintas.
- 1.2.5) En la misma factura no puede haber dos líneas con la misma película.
- 1.2.6) Si se borra una línea de detalle factura, automáticamente se borra la factura correspondiente.

## Ejercicio 2

Se dice que dos consultas SQL son equivalentes si al ejecutarlas para cualquier instancia, producen el mismo resultado (salvo quizás el orden de aparición de las tuplas resultantes que es lo único en lo que podrían diferir). Para que dos consultas SQL se consideren equivalente sus resultados deberían coincidir hasta en la cantidad de tuplas.

A continuación se muestran pares de consultas para investigar si son equivalentes. Notar que se está trabajando sobre relaciones en SQL y no en el modelo puro relacional.

Para cada ítem se pide indicar si las consultas son o no equivalentes. De **no serlo**, mostrar un contraejemplo, o sea, una instancia que evidencie el problema, indicando los resultados que generarían Q1 y Q2. Si el problema es que alguna de las consultas ni siquiera compila, señalarlo claramente explicando su error sintáctico.

Todas las consultas se aplican sobre los siguientes esquema de relación:

**R1( A, B, C)** tiene como clave al atributo A

**R2( F, G, H)** tiene una clave foránea sobre el atributo F que referencia la relación R1

Ningún atributo puede ser NULL (todos fueron definidos NOT NULL)

2.1)

**Q1:**

```
SELECT A
FROM R1
WHERE EXISTS (SELECT * FROM R2 WHERE F = A)
```

**Q2:**

```
SELECT F
FROM R2
WHERE EXISTS (SELECT * FROM R1 WHERE F = A)
```

2.2)

**Q1:**

```
SELECT A
FROM R1
WHERE EXISTS (SELECT * FROM R2 WHERE F = A)
```

**Q2:**

```
SELECT F
FROM R2
WHERE F IN (SELECT A FROM R1)
```

2.3)

**Q1:**

```
SELECT R1.*, R2.*
FROM R1 JOIN R2 ON A = F
```

**Q2:**

```
SELECT R1.*, R2.*
FROM R2 LEFT OUTER JOIN R1 ON A = F
```

2.4)

**Q1:**

```
SELECT R1.*, R2.*
FROM R1 JOIN R2 ON A = F
```

**Q2:**

```
SELECT R1.*, R2.*
FROM R2 RIGHT OUTER JOIN R1 ON A = F
```

2.5)

**Q1:**

```
SELECT  A
FROM R1, R2
WHERE A = F
GROUP BY A
HAVING A > 1
```

**Q2:**

```
SELECT  DISTINCT A
FROM R1, R2
WHERE A = F AND A > 1
```

2.6)

**Q1:**

```
SELECT  A
FROM R1, R2
WHERE A = F
GROUP BY A
HAVING COUNT(B) > 1
```

**Q2:**

```
SELECT DISTINCT A
FROM R1, R2
WHERE A = F AND COUNT(B) > 1
```

### Ejercicio 3

Se tiene el archivo de texto DATOS.TXT que contiene la siguiente información:

```
10&Maria Clara&20.34&12/10/2000
55&Soledad&40.2&28/10/2000
20&Fernanda&80.44&12/09/2001
17&Martina&20.34&26/11/2001
.....
```

La primera columna representa el legajo, la segunda el nombre, la tercera el sueldo y la cuarta la fecha. A modo de ejemplo, la primera línea podría leerse como “existe una persona con legajo 10 cuyo nombre es Maria Clara que cobra 20.34\$ y que ingresó el 12/10/2000”.

Se quiere importar su información en la tabla PERSONAL que fue construida con la siguiente sentencia, y ya posee tuplas insertadas:

```
CREATE TABLE personal
(
    nombre      CHAR(20) NOT NULL,
    fecha       DATE,
    legajo       INT NOT NULL PRIMARY KEY
)
```

Indicar todos los pasos y comandos que habría que usar para insertar, en forma masiva, la información del archivo en la tabla PERSONAL con utilitarios DB2 (no con sucesivos “insert”) . No deben insertarse los valores de los sueldos en esta tabla.

## Ejercicio 4

Se tiene el siguiente esquema de base de datos, que representa recetas de cocina.

La relación **RECETA** está identificada por un **id**, y tiene asociada una **descripción** y la **cantidad de porciones** que salen de ella. Ejemplo: la receta **1** se llama **Lemon Pie** y se puede construir a partir de ella **4** porciones.

La relación **INGREDIENTE** posee un **nombre** que la identifica, su **precio** en el mercado asociado a un **peso en gramos**. Ejemplo: **500 g** de **Harina** cuesta **\$1**. Todos los ingredientes aparecen en la tabla medidos en la unidad gramos.

La relacion **FORMADAPOR** muestra la asociación entre RECETA, INGREDIENTE que la compone y **cantidad** de los mismos en su composición (siempre medida en gramos). Obviamente tiene dos claves foráneas: una a la tabla **RECETA** y otra a la tabla **INGREDIENTE**. Ejemplo: la receta con ID **1** lleva en su composición **1000 g** de **Harina**, también lleva en su composición **600 g** de **Leche**, etc.

RECETA		
<u>Id</u>	Descrip	#porcion

INGREDIENTE		
<u>Nombre</u>	Gramaje	Precio

FORMADAPOR		
<u>Id</u>	<u>Nombre</u>	Cant

### Ejemplo:

Una posible instancia del esquema de base de datos sería

RECETA		
ID	Descrip	#Porcion
1	'Lemon Pie'	4
2	'Pan Arabe'	3
3	'Tallarines'	4

INGREDIENTE		
Nombre	Gramaje	Precio
'Limon'	1000	0.8
'Harina'	500	1
'Manteca'	200	1.3
'Leche'	1000	1
'Azucar'	500	1.3
'Huevo'	100	0.5
'Agua'	100	0

FormadaPor		
ID	Nombre	Cantidad
1	'Limon'	200
1	'Harina'	1000
1	'Manteca'	200
1	'Leche'	600
1	'Azucar'	50
1	'Huevo'	200
1	'Agua'	100
3	'Huevo'	200
3	'Harina'	1500
3	'Agua'	100
2	'Agua'	400
2	'Harina'	1000

### 4.1)

Escribir una consulta SQL en DB2 para mostrar los nombres de los ingredientes tales que en todas las recetas en las cuales lograron intervenir lo hacen en una proporción correspondiente a por lo menos el 20% de la composición (en gramos) de dicha receta.

Con el ejemplo anterior debe obtenerse:

NOMBRE
-----
Harina
Leche

Como la receta **1** usa **2350 gramos**, la **2** usa **1400 gramos** y la **3** usa **1800 gramos**, si desglosamos por ingrediente encontramos que:

ID	NOMBRE	3
1	Limon	0,0851063829787234042553191
1	Harina	0,4255319148936170212765957
1	Manteca	0,0851063829787234042553191
1	Leche	0,2553191489361702127659574
1	Azucar	0,0212765957446808510638297
1	Huevo	0,0851063829787234042553191
1	Agua	0,0425531914893617021276595
2	Agua	0,2857142857142857142857142
2	Harina	0,7142857142857142857142857
3	Huevo	0,1111111111111111111111111
3	Harina	0,8333333333333333333333333
3	Agua	0,0555555555555555555555555

Se marcó en gris aquellos ingredientes que logran intervenir en más un 20% de la composición correspondiente, y de ellas se observa que la que lo logran en todas las recetas en las que intervienen son sólo Harina y Leche.

#### 4.2)

Escribir una consulta SQL en DB2 para mostrar el nombre de cada receta y cuanto costaría preparar una **sola porción** de la misma, sólo para aquellas recetas que lleven no más de 4 productos en su composición.

Con el ejemplo anterior debe obtenerse:

DESCRIP	CUESTA
Pan Arabe	0.66
Tallarines	1.0

#### 4.3)

Escribir una consulta SQL en DB2 para averiguar el identificador de las recetas que son amargas y usaron todos los ingredientes posibles registrados en INGREDIENTE. Se consideran amargas aquellas recetas que usaron como mucho un 5% de su composición en azúcar.

Aclaración: Se asegura que el ingrediente Azucar está presente en la tabla ingrediente.

#### Ejemplo 1:

Con el ejemplo anterior debe obtenerse:

ID
1

ya que la única receta que uso todos los ingredientes es Lemon Pie (o sea la 1) y como el azúcar es de 50 g frente al total de 2350g, representa un 2.12% de la misma, por lo tanto es amarga.

#### Ejemplo 2:

Si se tuvieran los datos del ejemplo anterior pero la cantidad usada de azúcar por la receta 1 fuera 200g, entonces no se mostraría porque resultaría ser dulce (8.51 % de la composición)

### Ejercicio 5

Se tiene la relación **CURSADA** que almacena las notas de cursada de los alumnos de una facultad en la cual las materias no pueden dictarse en más de un cuatrimestre por año. Su esquema es **CURSADA(legajo, materia, año, nota)**.

Por ejemplo, la tupla (33345, 'Base de Datos I', 2000, 4) representa que el estudiante con legajo 33345 cursó durante el 2000 la materia Base de Datos I y aprobó el TP con 4.

Dada la siguiente consulta sobre la tabla CURSADA:

```
SELECT legajo
FROM CURSADA
GROUP BY legajo
HAVING MAX(nota) > 4.0
```

#### 5.1)

Indicar si en el **álgebra relacional vista en clase** puede realizarse una consulta que produzca el mismo resultado que la dada. En caso afirmativo escribirla y en caso negativo justificar la imposibilidad.

#### 5.2)

Indicar si en el **cálculo relacional de tuplas visto en clase** puede realizarse una consulta que produzca el mismo resultado que la dada. En caso afirmativo escribirla y en caso negativo justificar la imposibilidad.

#### 5.3)

Indicar si en el **cálculo relacional de dominios visto en clase** puede realizarse una consulta que produzca el mismo resultado que la dada. En caso afirmativo escribirla y en caso negativo justificar la imposibilidad.