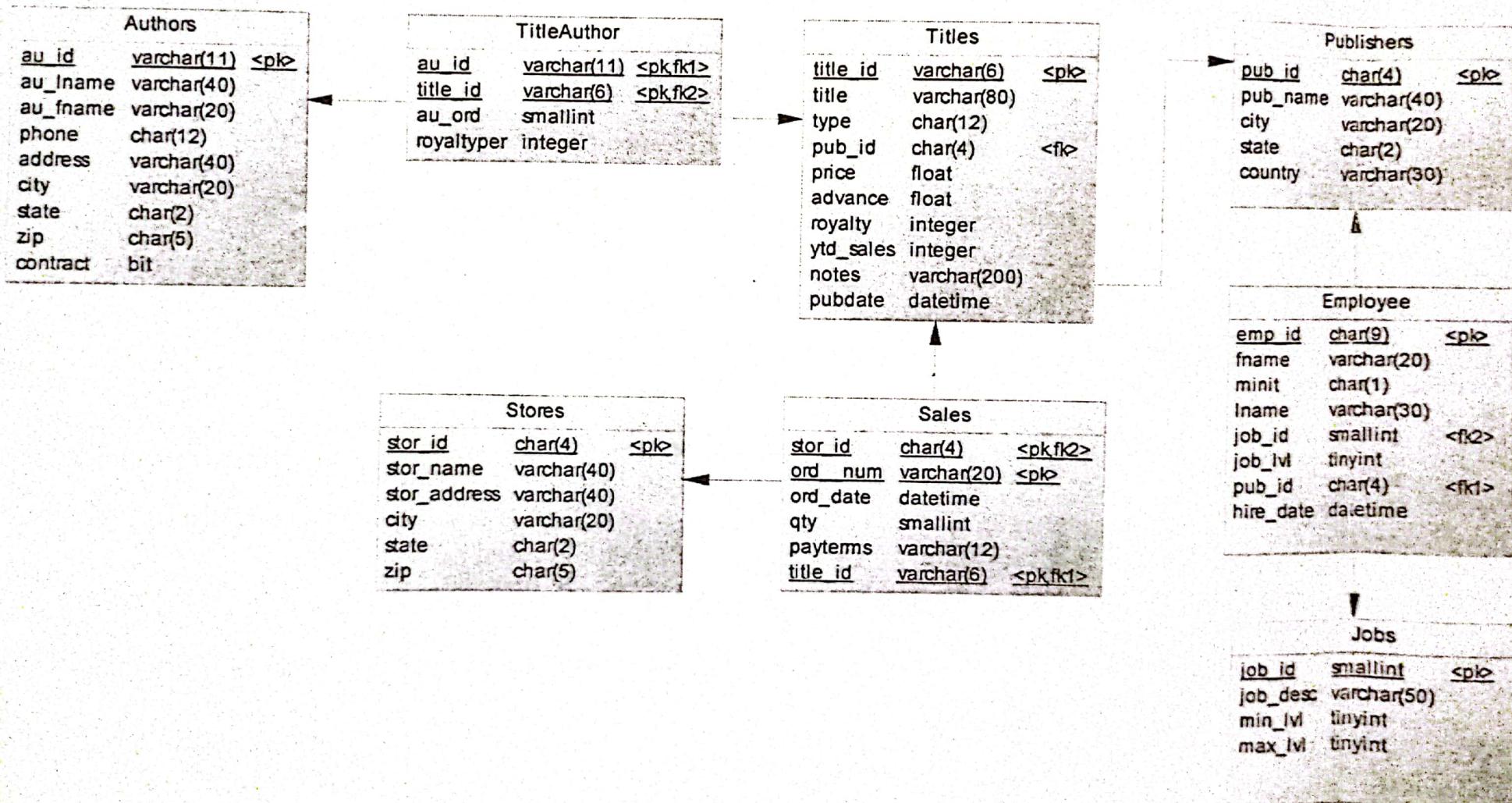


Bases de Datos - Examen Final

Ejercicio 1

Recordemos el modelo de la base de datos pubs:



La información de autoría o coautoría

Veamos el siguiente fragmento de los datos en la tabla TitleAuthor:

	au_id	title_id	au_ord	royaltyper
1	172-32-1176	PS3333	1	100
2	213-46-8915	BU1032	2	40
3	213-46-8915	BU2075	1	100
4	238-95-7766	PC1035	1	100
5	267-41-2394	BU1111	2	40
6	267-41-2394	TC7777	2	30
7	274-80-9391	BU7832	1	100
8	409-56-7008	BU1032	1	60
9	427-17-2319	PC8888	1	50
10	472-27-2349	TC7777	3	30

El porcentaje de coautoría se refleja en la columna royaltyper. Para una determinada publicación, un valor 100 indica que el autor posee el 100% de la autoría.

Por ejemplo, la publicación PS3333 registra la autoría del autor 172-32-1176 por un 100%

Analicemos ahora el caso de la publicación 'BU1032'.

	au_id	title_id	au_ord	royaltyper
1	172-32-1176	PS3333	1	100
2	213-46-8915	BU1032	2	40
3	213-46-8915	BU2075	1	100
4	238-95-7766	PC1035	1	100
5	267-41-2394	BU1111	2	40
6	267-41-2394	TC7777	2	30
7	274-80-9391	BU7832	1	100
8	409-56-7008	BU1032	1	60
9	427-17-2319	PC8888	1	50
10	472-27-2349	TC7777	3	30

Posee dos autores:

El autor 213-46-8915 posee un 40% de coautoría
El autor 409-56-7008 posee un 60% de coautoría

Se ha decidido implementar "campañas", en las que los autores registran **puntos** a partir de las **ventas** de las publicaciones de su autoría. Para ello se ha creado una tabla **PuntajeCampaign**:

PuntajeCampaign		
<u>idPuntajeCampaign</u>	integer	<pk>
<u>idCampaign</u>	integer	
<u>au_id</u>	varchar(11)	<fk>
<u>puntaje</u>	integer	
<u>usuario</u>	varchar(40)	
<u>fechaUltModif</u>	datetime	

Authors		
<u>au_id</u>	varchar(11)	<pk>
<u>au_lname</u>	varchar(40)	
<u>au_fname</u>	varchar(20)	
<u>phone</u>	char(12)	
<u>address</u>	varchar(40)	
<u>city</u>	varchar(20)	
<u>state</u>	char(2)	
<u>zip</u>	char(5)	
<u>contract</u>	bit	

TitleAuthor		
<u>au_id</u>	varchar(11)	<pk,fk1>
<u>title_id</u>	varchar(6)	<pk,fk2>
<u>au_ord</u>	smallint	
<u>royaltyper</u>	integer	

Titles		
<u>title_id</u>	varchar(6)	<pk>
<u>title</u>	varchar(80)	
<u>type</u>	char(12)	
<u>pub_id</u>	char(4)	<fk>
<u>price</u>	float	
<u>advance</u>	float	
<u>royalty</u>	integer	
<u>ytd_sales</u>	integer	
<u>notes</u>	varchar(200)	
<u>pubdate</u>	datetime	

Publishers		
<u>pub_id</u>	char(4)	<pk>
<u>pub_name</u>	varchar(40)	
<u>city</u>	varchar(20)	
<u>state</u>	char(2)	
<u>country</u>	varchar(30)	

Stores		
<u>stor_id</u>	char(4)	<pk>
<u>stor_name</u>	varchar(40)	
<u>stor_address</u>	varchar(40)	
<u>city</u>	varchar(20)	
<u>state</u>	char(2)	
<u>zip</u>	char(5)	

Sales		
<u>stor_id</u>	char(4)	<pk,fk2>
<u>ord_num</u>	varchar(20)	<pk>
<u>ord_date</u>	datetime	
<u>qty</u>	smallint	
<u>payterms</u>	varchar(12)	
<u>title_id</u>	varchar(6)	<pk,fk1>

Employee		
<u>emp_id</u>	char(9)	<pk>
<u>fname</u>	varchar(20)	
<u>minit</u>	char(1)	
<u>lname</u>	varchar(30)	
<u>job_id</u>	smallint	<fk2>
<u>job_lvl</u>	tinyint	
<u>pub_id</u>	char(4)	<fk1>
<u>hire_date</u>	datetime	

Jobs		
<u>job_id</u>	smallint	<pk>
<u>job_desc</u>	varchar(50)	
<u>min_lvl</u>	tinyint	
<u>max_lvl</u>	tinyint	

La tabla posee el siguiente esquema:

```
CREATE TABLE PuntajeCampaign
```

```
{  
    idPuntajeCampaign Integer NOT NULL,  
    idCampaign integer NOT NULL,  
    au_id varchar(11) NOT NULL,  
    puntaje integer NOT NULL,  
    usuario varchar(40),  
    fechaUltModif datetime  
}
```

PuntajeCampaign	
idPuntajeCampaign	integer
idCampaign	integer
au_id	varchar(11) <fk>
puntaje	integer
usuario	varchar(40)
fechaUltModif	datetime

```
ALTER TABLE PuntajeCampaign
```

```
    ADD CONSTRAINT PK_PuntajeCampaign PRIMARY KEY (idPuntajeCampaign)
```

```
ALTER TABLE PuntajeCampaign
```

```
    ADD CONSTRAINT PuntajeCampaign_FK  
    FOREIGN KEY (au_id) REFERENCES authors(au_id);
```

Cada entrada en la tabla PuntajeCampaign se identifica por una clave artificial idPuntajeCampaign.

Las campañas (idCampaign) se identifican con un número entero. La campaña actual es la 12.

La tabla posee una clave de negocios compuesta (clave candidata), formada por el ID de la campaña (idCampaign) y el código de autor (au_id).

Como se obtiene el puntaje

El puntaje a registrar depende del porcentaje de coautoría del autor sobre la publicación vendida.

El puntaje que se otorga es cantidad-vendida * (royaltyper/10)

Volvamos al ejemplo de la publicación 'BU1032'

	au_id	title_id	au_ord	royaltyper
1	172-32-1176	PS3333	1	100
2	213-46-8915	BU1032	2	40
3	213-46-8915	BU2075	1	100
4	238-95-7766	PC1035	1	100
5	267-41-2394	BU1111	2	40
6	267-41-2394	TC7777	2	30
7	274-80-9391	BU7832	1	100
8	409-56-7008	BU1032	1	60
9	427-17-2319	PC8888	1	50
10	472-27-2349	TC7777	3	30

Si, por ejemplo, se venden 2 unidades de la publicación 'BU1032':

```
INSERT Sales
    SELECT '6380', 'P2222', CURRENT_TIMESTAMP, 2, 'NET 30', 'BU1032'
```

Al autor 213-46-8915 que posee un 40% de coautoría, le corresponderá un puntaje (4 * 2).

4 por su (porcentaje de coautoría /10) y multiplicado por dos porque se vendieron dos unidades.

De manera similar, al autor 409-56-7008 que posee un 60% de coautoría, le corresponderá un puntaje (6 * 2).

Deberíamos obtener, en la tabla PuntajeCampaign, dos entradas como las siguientes:

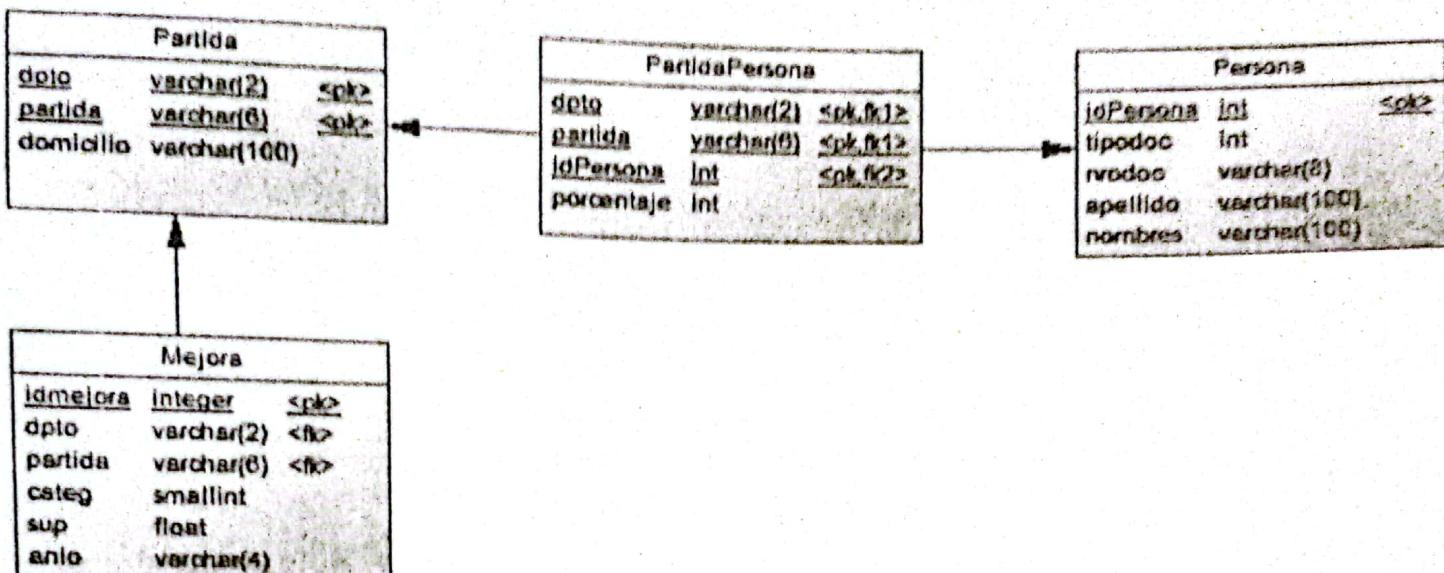
	idPuntajeCampaign	idCampaign	au_id	puntaje	usuario	fechUltModif
1	1	12	427-17-2319	10	NULL	NULL
2	2	12	213-46-8915	8	dbo	2020-10-26 00:59:53.550
3	3	12	409-56-7008	12	dbo	2020-10-26 00:59:53.550

Escriba un **trigger T-SQL** que, ante la inserción de una venta en la tabla Sales, registre el puntaje correspondiente en la tabla PuntajeCampaign para la campaña 12.

Tener en cuenta que:

- El puntaje para un autor se obtiene a partir de las NUEVAS ventas. Las ventas existentes no cuentan.
- Se debe obtener el nuevo idPuntajeCampaign incrementando en uno el valor del último IDPuntajeCampaign insertado.
- Si no existen entradas previas para la campaña y autor en cuestión, se debe crear una entrada nueva para el autor y la campaña, con el puntaje correspondiente.
- Si ya existen entradas previas para la campaña y autor en cuestión, se debe actualizar el puntaje correspondiente.
- Si el autor ya ha alcanzado el puntaje 400 para la campaña 12, NO se debe realizar ninguna operación ni INSERT ni UPDATE sobre PuntajeCampaign.
- Registrar fecha de última modificación (columna fechUltModif) y usuario de base de datos que realiza la inserción/actualización (columna usuario).

2. La Dirección de Catastro Provincial cuenta con un modelo de datos como el siguiente:



Cada **terreno** con características urbanas en la provincia se denomina **Partida** y es identificado por el número de departamento donde está ubicado (dpto) y un número de partida dentro de ese departamento. domicilio es el domicilio del terreno.

Por su parte los propietarios son identificados en la tabla Persona. Poseen un identificadorio (idPersona), poseen tipo y nro. de documento, apellido y nombres.

Una partida puede tener un único o varios propietarios. Los mismos se modelizan en la tabla PartidaPersona. La columna porcentaje especifica un valor entre 1 y 100 que indica el porcentaje de la propiedad correspondiente a cada persona.

La tabla Mejora representa las **construcciones** que se han realizado a una partida.

idmejora identifica únicamente a la mejora.

categ es la categoría de mejora realizada (un valor entre 1 y 5). Básicamente una categoría de mejora es un valor de clasificación al que se arriba luego de relevar la mejora de acuerdo a sus características, -sobre todo los materiales de construcción utilizados-, en una serie de rubros, como ser muros, pisos, revestimiento de fachada, etc. La manera en que se arriba a dicho valor y lo que significa exactamente cada valor entero no es relevante para este examen.

sup es la superficie de la mejora en metros cuadrados.

anio es el año en que fue declarada la mejora.

El siguiente es un listado que muestra algunos propietarios, sus propiedades y los porcentajes de copropiedad:

	Results	Messages				
	idPersona	apellido	nombres	dpto	partida	porcentaje
1	1	López	Ricardo	10	100100	50
2	3	Arroyo	Jorgelina	10	100100	50
3	1	López	Ricardo	10	111111	100
4	1	López	Ricardo	10	222222	50
5	2	Miranda	Pedro	10	222222	50
6	3	Arroyo	Jorgelina	10	333333	100
7	4	Pedrolini	Lucas	10	444444	100
8	1	López	Ricardo	10	505050	100

El siguiente es un listado de las mejoras de estas partidas:

	Results	Messages				
	idmejora	dpto	partida	categ	sup	anio
1	1	10	111111	1	250	2004
2	2	10	111111	2	300	2006
3	3	10	222222	1	350	2007
4	4	10	333333	4	100	2006
5	5	10	444444	1	450	2006
6	6	10	100100	1	50	1975
7	7	10	100100	2	70	1978
8	8	10	505050	1	60	1972
9	9	10	100100	1	100	1978
10	10	10	505050	2	100	1972
11	11	10	505050	3	100	1980

Obtener apellido y nombres de los propietarios de partidas cuando alguna de sus partidas posean mejoras de categoría 1 siempre y cuando esas partidas también posean otras mejoras de categoría 1 pero con exactamente el doble de superficie.

Resolver utilizando SQL directo, sin utilizar vistas, tablas auxiliares ni extensiones T-SQL o PL/pgSQL.

No utilizar sentencias o comandos no vistos en las Guías de Estudio.

3. En el modelo de la base de datos pubs:

Hay una única fila de detalle en Sales para el almacén '7066', ord_num 'QA7442.3', y corresponde a la publicación 'PS2091':

	Results	Messages
1	stor_id ord_num ord_date qty payterms title_id 7066 QA7442.3 1994-09-13 00:00:00.000 75 ON invoice PS2091	

Un usuario (Sesión 1) ejecuta las siguientes sentencias:

Sesión 1

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
```

```
BEGIN TRANSACTION
```

```
DELETE FROM sales  
WHERE stor_id = '7066' AND  
ord_num = 'QA7442.3' AND  
title_id = 'PS2091'
```

```
WAITFOR DELAY '00:00:30.000'; --Delay de 30 segundos
```

```
ROLLBACK TRANSACTION
```

Otro usuario (Sesión 2) ejecuta la siguiente sentencia:

Sesión 2

```
Select *  
FROM sales  
WHERE stor_id = '7066' AND  
ord_num = 'QA7442.3' AND  
title_id = 'PS2091'
```

Conteste con verdadero o falso:

- A. La sentencia de la Sesión 2 se ejecuta inmediatamente
- B. La sentencia de la Sesión 2 no obtiene filas.

Justifique ambas respuestas