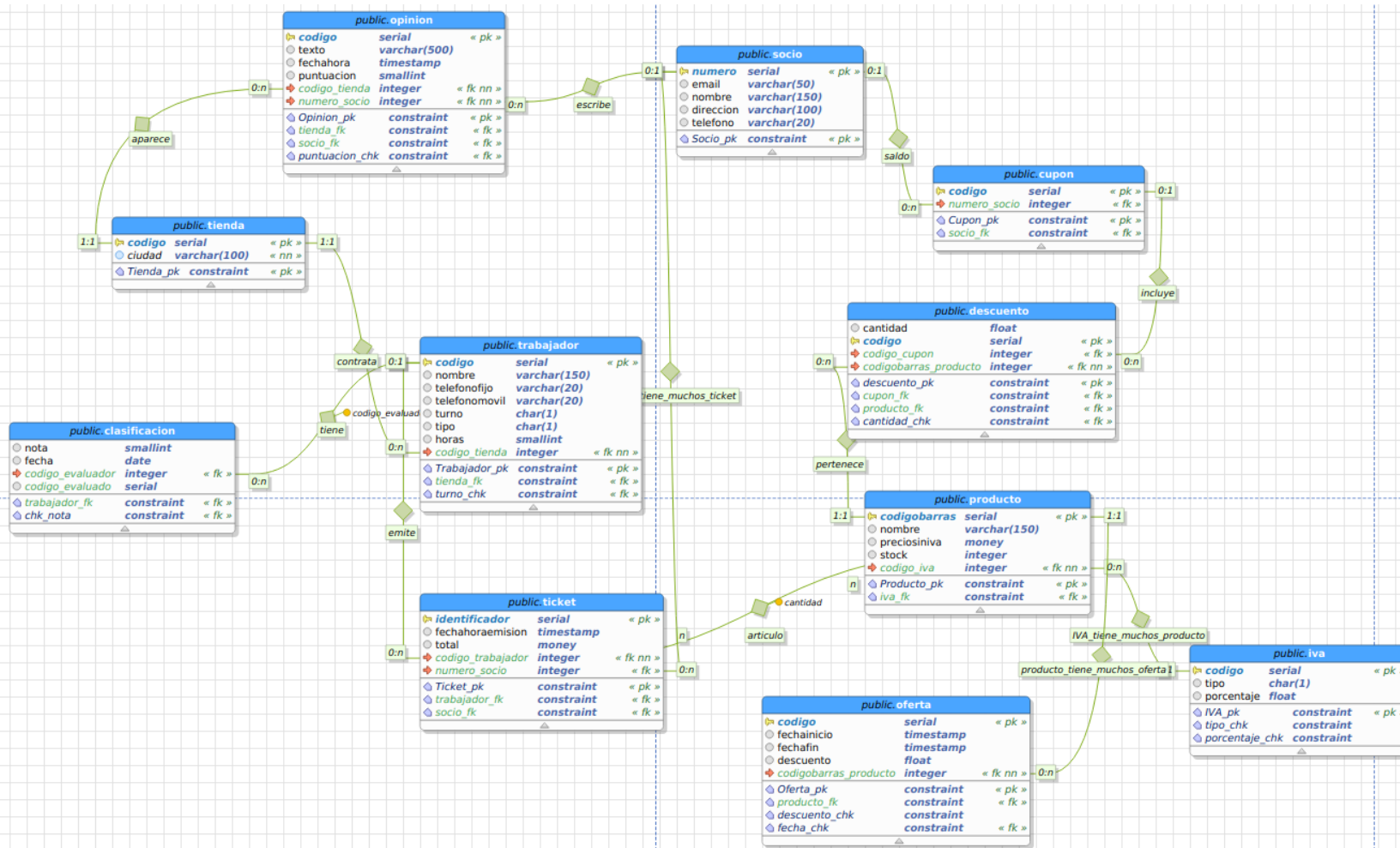


**Bases de Datos
Grado II**

PECL2 2019-20

Grupo: Ñ
Componentes del grupo:
Marcel Andrei Voicu
Carlos Javier Hellín Asensio



Se han realizado las siguientes modificaciones a la PECL1:

- La relación de producto – oferta ha pasado de ser N:1 a 1:N para que la tabla oferta tenga la clave primaria de producto y no al revés, con la ventaja de poder realizar mejor las consultas SQL.
- Se eliminado la relación ES de trabajador y se ha añadido un atributo llamado tipo a la tabla trabajador que indica con una letra si es cajero (C) o reponedor (R), con el objetivo de dejar una base de datos menos compleja.

-- 1

SELECT codigoBarras, precioSinIva FROM producto;

	codigoBarras [PK] integer	precioSinIva money
1	1	82,01 €
2	2	8,21 €
3	3	63,95 €
4	4	80,06 €
5	5	46,19 €
6	6	19,45 €
7	7	23,54 €
8	8	18,43 €
9	9	86,01 €
10	10	9,83 €
11	11	17,09 €
12	12	93,60 €

-- 2

select nombre, case tipo when 'C' then 'Cajero' else 'Reponedor' end as tipo from trabajador;

	nombre character varying (150)	tipo text
1	Lissie Brumble	Cajero
2	Cacilia Bonass	Cajero
3	Ali Angood	Reponedor
4	Kory Adriano	Reponedor
5	Elmore O'Cahsedy	Reponedor
6	Nanette Fried	Reponedor
7	Jermayne Duchart	Cajero
8	Derek Laugheran	Cajero
9	Putnem Gounel	Cajero
10	Karlyn Memmory	Reponedor
11	Anselma Westoff	Reponedor
12	Arda Bradlaugh	Cajero

-- 3

SELECT nombre

FROM trabajador

WHERE tipo = 'R' AND horas > 4; -- Considerando que los días laborales son de Lunes a Viernes y que 4 horas diarias * 5 días hacen 20 semanales

	nombre character varying (150)
1	Ali Angood
2	Kory Adriano
3	Elmore O'Cahsedy
4	Nanette Fried
5	Karlyn Memmory
6	Anselma Westoff
7	Grace Cutchey
8	Kristi de Quesne
9	Brock Probbin
10	Ike Tipper
11	Blondy Verduin
12	Hobie Joberne





-- 4

select sum(total) as total_facturado from ticket;

	total_facturado money
1	1.725,11 €



-- 5

```
SELECT codigo, nombre, cantidad
FROM (SELECT cupon.codigo, codigoBarras_producto, cantidad
      FROM cupon
      INNER JOIN descuento ON cupon.codigo = codigo_cupon) AS temp1
INNER JOIN producto ON codigoBarras = temp1.codigoBarras_producto;
```

		codigo integer		nombre character varying (150)		cantidad double precision	
1		1		Cheese - Gouda		0.1	
2		2		Caviar - Salmon		0.1	
3		3		Cheese - Gouda		0.2	
4		4		Cream - 10%		0.2	
5		5		Pears - Bartlett		0.5	
6		6		Basil - Fresh		0.4	
7		7		Juice - Happy Planet		0.2	
8		8		Beef - Tenderloin		0.1	
9		9		Foil Cont Round		0.4	
10		10		Seedlings - Mix, Organic		0.2	
11		11		Jam - Apricot		0.4	
12		12		Pineapple - Canned, Rings		0.2	



-- 6

```
select nombre from producto join descuento on producto.codigoBarras =
descuento.codigoBarras_producto limit 5;
```

		nombre character varying (150)	
1		Cheese - Gouda	
2		Caviar - Salmon	
3		Cheese - Gouda	
4		Cream - 10%	
5		Pears - Bartlett	

-- 7

```
SELECT SUM(puntuacion)/COUNT(puntuacion) AS notaMedia FROM opinion;
```

		notamedia bigint	
1		5	




-- 8

```
select count(ticket.identificador) as numero_tickets, nombre, ciudad from ticket join trabajador on
ticket.codigo_trabajador = trabajador.codigo join tienda on trabajador.codigo_tienda =
tienda.codigo group by codigo_trabajador, nombre, ciudad order by numero_tickets desc;
```

		numero_tickets bigint		nombre character varying (150)		ciudad character varying (100)	
1		3		Arda Bradlaugh		Madrid	
2		3		Brock Probbin		Torrejon	
3		3		Elmore O'Cahsedy		Alcala de Henares	
4		2		Kory Adriano		Torrejon	
5		2		Cacilia Bonass		Madrid	
6		2		Lissie Brumble		Getafe	
7		1		Derek Laugheran		Alcala de Henares	
8		1		Leah Idel		Alcala de Henares	
9		1		Harbert Conaghy		Madrid	
10		1		Nanette Fried		Torrejon	
11		1		Ellene Feore		Torrejon	
12		1		Grace Cutchev		Getafe	

-- 9

```
SELECT tienda.codigo AS tienda, COUNT(trabajador) AS nTrabajadores
FROM tienda
INNER JOIN trabajador ON tienda.codigo = codigo_tienda
GROUP BY tienda.codigo
ORDER BY nTrabajadores ASC;
```

		tienda integer		ntrabajadores bigint	
1		9		3	
2		3		3	
3		5		3	
4		4		3	
5		10		3	
6		6		3	
7		2		3	
8		7		3	
9		1		3	
10		8		3	

-- 10

```
select nombre, telefonofijo, telefonomovil, (sum(nota) / count(codigo_evaluador)) as media from
clasificacion join trabajador on trabajador.codigo = clasificacion.codigo_evaluado group by
nombre, telefonofijo, codigo_evaluador, telefonomovil having (sum(nota) /
count(codigo_evaluador)) >= 10;
```

	nombre character varying (150)	telefonofijo character varying (20)	telefonomovil character varying (20)	media bigint
1	Kristi de Quesne	459 180 0674	489 951 0530	10
2	Atlanta Yegorovnin	291 182 8604	964 314 3658	10

-- 11

```
SELECT codigobarras_producto, descuento
```

```
FROM oferta
```

```
WHERE fechainicio BETWEEN '2019/05/01 00:00:00' AND '2019/05/01 23:59:59';
```

	codigobarras_producto integer	descuento double precision
1	2	0.4

-- 12

```
select distinct email from socio join ticket on numero = numero_socio join articulo on identificador
= identificador_ticket join oferta on articulo.codigobarras_producto =
oferta.codigobarras_producto and fechainicio >= (DATE('2019-05-31 12:00:00') - 7) and fechafin <=
'2019-05-31 23:59:59';
```

	email character varying (50)
1	aquailb@nature.com

-- 13

```
SELECT nombre
```

```
FROM (SELECT codigo FROM tienda
```

```
WHERE ciudad LIKE 'M%') AS temp1
```

```
INNER JOIN trabajador ON codigo_tienda = temp1.codigo
```

```
ORDER BY nombre ASC;
```

	nombre character varying (150)
1	Arda Bradlaugh
2	Cacilia Bonass
3	Harbert Conaghy
4	Hobie Joberne
5	Ike Tipper
6	Putnem Gounel

-- 14

```
select email from socio join cupon on numero = numero_socio join descuento on cupon.codigo =  
codigo_cupon group by email order by sum(cantidad) desc limit 1;
```

	email	
	character varying (50)	
1	eshaplint@cpanel.net	

-- 15

```
SELECT nombre  
FROM (SELECT nombre, COUNT(*) AS cantidad  
      FROM (SELECT identificador  
            FROM ticket  
              WHERE total < '€0.00') AS temp1  
      INNER JOIN articulo ON temp1.identificador = identificador  
      INNER JOIN producto ON articulo.codigoBarras_producto = codigoBarras  
      GROUP BY nombre  
      ORDER BY cantidad DESC  
      LIMIT 1) AS temp2;
```

	nombre	
	character varying (150)	
1	Plastic Wrap	

-- 16

```
select nombre from ticket join trabajador on ticket.codigo_trabajador = trabajador.codigo join  
tienda on trabajador.codigo_tienda = tienda.codigo group by codigo_trabajador, nombre, ciudad  
order by count(codigo_trabajador) desc limit 1;
```

	nombre	
	character varying (150)	
1	Brock Probbin	

-- 17

```
SELECT nombre  
FROM (SELECT nombre  
      FROM (SELECT numero_socio, MAX(puntuacion) AS maxima  
            FROM opinion  
              GROUP BY numero_socio  
              ORDER BY maxima DESC  
              LIMIT 1) AS temp1  
      INNER JOIN socio ON temp1.numero_socio = numero) AS temp2;
```

	nombre	
	character varying (150)	
1	Jameson Zorzutti	

-- 18

select identificador, fechahoraemision, total from ticket join trabajador on
ticket.codigo_trabajador = trabajador.codigo join tienda on trabajador.codigo_tienda =
tienda.codigo where nombre like 'A%' and ciudad like 'M%';

	Identificador [PK] integer	fechahoraemision timestamp without time zone	total money
1	3	2019-06-28 05:43:56	54,23 €
2	23	2019-06-24 20:52:20	132,77 €
3	25	2019-09-10 16:07:43	-19,90 €

-- 19

SELECT nombre, identificador AS ticket
FROM (SELECT nombre, trabajador.codigo
FROM (SELECT codigo
FROM tienda
WHERE ciudad = 'Alcala de Henares') AS temp1
INNER JOIN trabajador ON temp1.codigo = codigo_tienda) AS temp2
INNER JOIN ticket ON temp2.codigo = codigo_trabajador;

	nombre character varying (150)	ticket integer
1	Gisela Noonan	1
2	Elmore O'Cahsedy	5
3	Grady Cavell	6
4	Elmore O'Cahsedy	8
5	Elmore O'Cahsedy	9
6	Leah Idel	15
7	Jermayne Duchart	22
8	Derek Laughheran	26

-- 20

select identificador, nombre from ticket join trabajador on ticket.codigo_trabajador =
trabajador.codigo join tienda on trabajador.codigo_tienda = tienda.codigo join articulo on
identificador = identificador_ticket left join oferta on articulo.codigobarras_producto =
oferta.codigobarras_producto where oferta.codigo is null and numero_socio is null and ciudad =
'Alcala de Henares';

	Identificador integer	nombre character varying (150)
1	26	Derek Laughheran
2	22	Jermayne Duchart