

Profesor:	DAVID APARICIO ESCRIBANO	Grupo	81
Alumno/a:	MARIO RAMOS SALSON	NIA:	100495849
Alumno/a:	VICTOR MARTINEZ DE LAS HERAS	NIA:	100495829
Alumno/a:	MIGUEL FIDALGO GARCIA	NIA:	100495770

1. Introducción

La compañía “*Caffeine maniacs*” quiere introducir nuevas funcionalidades a su base de datos para ayudarles con la logística, estrategias de marketing y funcionalidades básicas de su empresa. Para ellos nos han pedido realizar dos consultas que les ayudan a observar cuales son los países que más venden y más compran de un determinado café, junto con un informe de las referencias más vendidas por mes en el último año. Esto les ayuda a mejorar su estrategia de marketing e incrementar el negocio de la empresa.

Por otro lado, también hemos implementado funcionalidades como convertir todos los borradores de pedidos en pedidos confirmados y mostrar una gran cantidad de información acerca de un proveedor en específico. Esto les ayuda a la hora de elegir proveedor para escoger el más apto y el que más beneficio les aporte.

Además, nos han pedido la implementación de 3 vistas diferentes las cuales muestran las compras de un usuario en concreto, la información de un usuario en concreto y los comentarios de cualquier usuario, dándole la posibilidad a este de editar el texto de su comentario si y sólo si los likes son 0.

Para finalizar, hemos implementado una serie de disparadores que hacen más resistentes las inserción, consultas y actualizaciones de la base de datos. Estas se encargan de actualizar atributos importantes, mover la información de un cliente que se da de baja, impedir que un cliente anónimo compre con una tarjeta de un cliente registrado. Y para finalizar, que cuando se haga un pedido, se reste del total de stock la cantidad comprada y ,en caso de que se supere el stock mínimo, se realice otro pedido de reposición.

2. Consultas

2.1 Bestsellers Geographic Report:

a) Su diseño en álgebra relacional

$$P_{AUX}(\sigma_{Orderdate > ADD_MONTHS(SYS_DATE, -12)}(\pi_{barcode, quantity, price, country, orderdate}(CLIENT_LINES)) \cup$$

$$\sigma_{Orderdate > ADD_MONTHS(SYS_DATE, -12)}(\pi_{barcode, quantity, price, country, orderdate}(ANONIM_LINES)))$$
;

$$P_{AUX2}(\pi_{barcode, \underbrace{count('x')}_{n-compradores}, \underbrace{sum(quantity)}_{ventas}, price, \underbrace{sum(quantity) * price}_{ingresos}, \underbrace{round(avg(quantity), 2)}_{promedio-ventas}})$$

$$Group\ By\ barcode, price\ (aux)$$

$$P_A(\pi_{barcode, varietal, origen} (PRODUCTS * REFERENCES))$$

$$\uparrow$$

$$product$$

$$P_{ORDEN_VENTAS}(\pi_{origen, varietal, n-compradores, price, ingresos, promedio-ventas}(\pi_{AUX2 * A}))$$

$$\uparrow$$

$$barcode$$

$$P_{MAX_VENTAS_VARIETAL}(\sigma_{RANK=1}(\pi_{origen, varietal, ventas, n-compradores, price, ingresos, promedio-ventas}(ORDEN_VENTAS)))$$

$$P_{CONSUMIDORES_1}(\pi_{varietal, country, \underbrace{sum(quantity)}_{cantidad}} GroupBy varietal, country (A * AUX))$$

$$\uparrow$$

$$barcode$$

$$P_{CONSUMIDORES_2}(\pi_{varietal, country, cantidad, ventas, \underbrace{round(cantidad/ventas, 2) * 100}_{percentage}}(CONSUMIDORES_1 * MAX_VENTAS_VARIETAL))$$

$$\uparrow$$

$$varietal$$

$$P_{CONSUMIDORES_3}(\sigma_{percentage > 1}(\pi_{varietal, \underbrace{count('x')}_{países-potenciales}} GroupBy varietal (CONSUMIDORES)))$$

$$\pi_{origen, varietal, ventas, n-compradores, price, ingresos, promedio-ventas, países-potenciales}(MAX_VENTAS_VARIETAL * CONSUMIDORES)$$

$$\uparrow$$

$$varietal$$

b) Su implementación en SQL

```
-- CONSULTA 1
WITH AUX AS (
    SELECT BARCODE, QUANTITY, PRICE, ORDERDATE, COUNTRY FROM CLIENT_LINES
    WHERE ORDERDATE > ADD_MONTHS(SYS_DATE, -12)
```

```
UNION
    SELECT BARCODE, TO_CHAR(QUANTITY), PRICE, ORDERDATE, DLIV_COUNTRY
FROM LINES_ANONYM
WHERE ORDERDATE > ADD_MONTHS(SYSDATE, -12)
),
AUX2 AS (
    SELECT BARCODE, COUNT('X') AS N_COMPRADORES, SUM(QUANTITY) AS VENTAS,
PRICE, SUM(QUANTITY)*PRICE AS INGRESOS, ROUND(AVG(QUANTITY),2) AS
PROMEDIO_VENTAS
    FROM AUX
    GROUP BY BARCODE, PRICE
),
-- Sirve para mezclar los barcodes y saber el varietal de cada uno
A AS (
    SELECT DISTINCT barcode, varietal, origin FROM PRODUCTS NATURAL JOIN
REFERENCES
),
-- Sirve para mezclar la subconsulta anterior con los datos de cada
varietal
ORDEN_VENTAS AS (
    SELECT ORIGIN, VARIETAL, N_COMPRADORES, VENTAS, PRICE, INGRESOS,
PROMEDIO_VENTAS,
        ROW_NUMBER() OVER (PARTITION BY ORIGIN ORDER BY VENTAS DESC)
AS RN
    FROM A NATURAL JOIN AUX2
),
--SELECT ORIGIN, VARIETAL, VENTAS, RN FROM VENTAS_RANKED; --Para ver como
se hace la partición
MAX_VENTAS_VARIETAL AS (
    SELECT ORIGIN, VARIETAL, VENTAS, N_COMPRADORES, PRICE, INGRESOS,
PROMEDIO_VENTAS
    FROM ORDEN_VENTAS
    WHERE RN = 1
),
-- Cantidades totales del varietal por país comprador
CONSUMIDORES_1 AS (
    SELECT VARIETAL, COUNTRY, SUM(QUANTITY) AS CANTIDAD
```

```
FROM A NATURAL JOIN AUX GROUP BY VARIETAL, COUNTRY
),

-- Se añade la columna con máximo de ventas por varietal
CONSUMIDORES_2 AS (
    SELECT VARIETAL, COUNTRY, CANTIDAD, VENTAS,
    ROUND(CANTIDAD/VENTAS,3)*100 AS PORCENTAJE
    FROM CONSUMIDORES_1
    NATURAL JOIN MAX_VENTAS_VARIETAL
),

-- Filtras por los mayores del 1%
CONSUMIDORES_3 AS(
    SELECT VARIETAL, COUNT('X') AS PAISES_POTENCIALES FROM CONSUMIDORES_2
    WHERE PORCENTAJE > 1 GROUP BY VARIETAL
)

SELECT ORIGIN, VARIETAL, VENTAS, N_COMPRADORES, PRICE, INGRESOS,
    PROMEDIO_VENTAS, PAISES_POTENCIALES
FROM MAX_VENTAS_VARIETAL NATURAL JOIN CONSUMIDORES_3;
```

c) Las pruebas realizadas para demostrar que funciona correctamente

ORIGIN	VARIETAL	VENTAS	N_COMPRADORES	PRICE	INGRESOS	PROMEDIO_VENTAS	PAISES_POTENCIALES
1 BangladOs	Laurentii	137	13	72,35	9911,95	10,54	22
2 Brasil	Catuai	193	10	0,85	164,05	19,3	33
3 CamerOn	Charrieriana	148	12	30,95	4580,6	12,33	60
4 Ceylan	Ceylan	122	8	27,22	3320,84	15,25	60
5 Colombia	Sierra Nevada de Santa Marta	140	7	2,2	308	20	10
6 Costa Rica	Villalobos	184	15	4,3	791,2	12,27	69
7 Ecuador	Pacamara	141	9	34,09	4806,69	15,67	59
8 El Salvador	Bourbon Naranja	174	9	16,77	2917,98	19,33	118
9 EtiopDa	Harrar	144	13	16,65	2397,6	11,08	83
10 Filipinas	Baraco	136	13	3,35	455,6	10,46	54
11 Guadalupe	Bonifieur	108	11	36,25	3915	9,82	34
12 Guatemala	Huehuetenango	249	14	22,45	5590,05	17,79	37
13 Hawai	Kona	66	8	35,94	2372,04	8,25	31
14 India	Sarchimor	145	11	4,81	697,45	13,18	60
15 Indonesia	Mandheling	246	17	33,05	8130,3	14,47	114
16 Jamaica	Blue Mountain	154	9	4,85	746,9	17,11	59
17 Kenia	SL-34	120	10	1,2	144	12	88
18 Madagascar	Bourbon	186	11	7,89	1467,54	16,91	106
19 Malasia	Dewevrei	131	10	19,85	2600,35	13,1	35
20 Nicaragua	Caturra	81	4	0,7	56,7	20,25	10
21 Nueva Guinea	Arusha	166	6	24,14	4007,24	27,67	107
22 Panamá	Geisha	159	11	12,49	1985,91	14,45	87
23 Perú	Pache Colis	205	11	2,65	543,25	18,64	63
24 Perú	Catimor	139	12	34,25	4760,75	11,58	55
25 R.D.Congo	Excelsa	102	8	26,75	2728,5	12,75	12
26 Ruanda	MayagDez	123	10	9,75	1199,25	12,3	47
27 Tailandia	Black Ivory	108	9	38,75	4185	12	61
28 Tanzania	Peaberry	127	7	13,25	1682,75	18,14	100
29 Uganda	Bugishu	124	8	108,05	13398,2	15,5	82
30 Vietnam	Robusta	102	7	36,75	3748,5	14,57	92
31 Yemen	Moka	135	12	46,25	6243,75	11,25	68

2.2 Business way of life:

a) Su diseño en álgebra relacional

$$PAUX1 \left(\sigma_{orderdate > NOW - MONTHS(SYSDATE, -12)} \left(\pi_{barcode, quantity, price, orderdate(mes)} (CLIENT - LINES) \right) \cup \right. \\ \left. \sigma_{orderdate > NOW - MONTHS(SYSDATE, -12)} \left(\pi_{barcode, quantity, price, orderdate(mes)} (ANONYM - LINES) \right) \right)$$

$$PAUX2 \left(\pi_{mes, barcode, \underbrace{count('x')}_{n_pedidos}, \underbrace{sum(quantity)}_{ventas}, \underbrace{sum(quantity) * price}_{ingresos}} \text{ GroupBy } barcode, mes, price(AUX1) \right)$$

$$PAUX3 \left(\pi_{DISTINCT mes, \underbrace{max(ventas)}_{ventas}} \text{ GroupBy } mes(AUX2) \right)$$

$$PAUX4 \left(\pi_{mes, barcode, n_pedidos, ventas, ingresos} (AUX3 * AUX2) \right)$$

↑
mes

$$\pi_{mes, barcode, n_pedidos, ventas, ingresos, \underbrace{ingresos - ventas * max(cost)}_{beneficio}} \text{ GroupBy } mes, barcode, n_pedidos, ventas, ingresos$$

$$\text{Order By } mes(AUX4 * SUPPLY - LINES)$$

↑
barcode

b) Su implementación en SQL

```
-- CONSULTA 2
WITH AUX1 AS (
    SELECT BARCODE, QUANTITY, PRICE, SUBSTR(ORDERDATE, 4, 2) AS MES FROM
CLIENT_LINES
    WHERE ORDERDATE > ADD_MONTHS(SYSDATE, -12)
    UNION
    SELECT BARCODE, TO_CHAR(QUANTITY), PRICE, SUBSTR(ORDERDATE, 4, 2) AS
MES FROM LINES_ANONYM
    WHERE ORDERDATE > ADD_MONTHS(SYSDATE, -12)
),
AUX2 AS (
    SELECT MES, BARCODE, COUNT('X') AS N_PEDIDOS, SUM(QUANTITY) AS
VENTAS, SUM(QUANTITY)*PRICE AS INGRESOS
    FROM AUX1
    GROUP BY BARCODE, MES, PRICE
),
AUX3 AS (
    SELECT DISTINCT MES, MAX(VENTAS) AS VENTAS FROM AUX2 GROUP BY MES
),
AUX4 AS (
    SELECT MES, BARCODE, N_PEDIDOS, VENTAS, INGRESOS FROM AUX3 NATURAL
JOIN AUX2
)
SELECT MES, BARCODE, N_PEDIDOS, VENTAS, INGRESOS, INGRESOS - VENTAS *
MAX(COST) AS BENEFICIO FROM AUX4 NATURAL JOIN SUPPLY_LINES
GROUP BY MES, BARCODE, N_PEDIDOS, VENTAS, INGRESOS ORDER BY MES;
```

c) Las pruebas realizadas para demostrar que funciona correctamente

MES	BARCODE	N_PEDIDOS	VENTAS	INGRESOS	BENEFICIO
01	IQQ872240317472	2	78	389,22	39
02	OQI86097I858215	2	60	795	79,8
03	QOQ12753Q623045	1	39	2843,1	312,78
04	OII22831Q738220	4	71	1593,95	191,7
05	IQQ28760I789472	2	66	359,7	36,3
06	QIQ79610I616590	3	75	936,75	93,75
07	QOI21664I685931	3	59	6138,95	1043,71
08	III29452Q587105	4	63	2082,15	208,53
09	OOI63477Q670918	3	59	41,3	5,31
10	QIO96770Q227716	6	97	62,08	7,76
11	OIQ18613Q276154	3	71	241,4	36,21
12	IIQ67472I907468	3	80	68	8,8

3. Paquete

Los paquetes son objetos que almacenan un conjunto de variables, funciones o procedimientos. Se compilan y se guardan en la base de datos, donde pueden ser llamados y reutilizados en varias partes de una aplicación. Los paquetes tienen dos partes: la descripción, donde se declaran las variables, procedimientos, etc. Y el cuerpo, donde se especifica el código de cada procedimiento.

Nuestro paquete se llama 'caffeine' y tiene los procedimientos 'actualizar' e 'info'.

La descripción del paquete en sql es la siguiente:

```
CREATE OR REPLACE PACKAGE caffeine AS
    PROCEDURE actualizar;
    PROCEDURE info (proveedor in VARCHAR2);
END caffeine;
```

El cuerpo es el siguiente:

```
CREATE OR REPLACE PACKAGE BODY caffeine AS

    -- Definir variables para almacenar los resultados
    v_taxid REPLACEMENTS.TAXID%TYPE;
    v_confirmados NUMBER;
```

```

v_media_entrega NUMBER(5,2);
v_coste_media NUMBER(5,2);
v_coste_min NUMBER;
v_coste_max NUMBER;
v_resta_media NUMBER;
v_resta_min NUMBER;

PROCEDURE actualizar AS
BEGIN
. . .
END;

PROCEDURE info (proveedor in VARCHAR2) AS
BEGIN
. . .
END;

END caffeine;
/

```

Para las pruebas hemos introducido los siguientes datos en la tabla de replacements, ya que antes un mismo proveedor no tenía varias ofertas para el mismo producto. También insertamos un proveedor distinto para asegurarnos de que selecciona el que queremos y no otro.

1	J38320106J	OOO28882I889528	07/01/24	D	25	14/01/24	1750
2	J38320106J	OOO28882I889528	10/02/24	D	72	14/02/24	5277,6
3	J38320106J	OOO28882I889528	18/03/24	D	138	20/03/24	10446,6
4	J38320106J	OIQ47059I515289	07/01/24	D	22	18/01/24	402,6
5	J38320106J	OIQ47059I515289	05/02/24	D	89	14/02/24	2029,2
6	J38320106J	OIQ47059I515289	12/03/24	D	53	17/03/24	1128,9
7	J38320106J	IQQ99988O827109	07/02/24	D	13	09/02/24	455
8	J38320106J	IQQ99988O827109	14/02/24	D	53	15/02/24	2119,47
9	J38320106J	IQQ99988O827109	28/03/24	D	168	30/03/24	7056
10	J38320106J	OOO28882I889528	22/01/07	D	25	14/01/24	1750
11	Q53972745W	III07057O402995	07/01/24	D	25	14/01/24	1750
12	Q53972745W	III07057O402995	10/02/24	D	72	14/02/24	5277,6
13	Q53972745W	III07057O402995	18/03/24	D	138	20/03/24	10446,6

3.1 PROCEDURE actualizar

- a) Su diseño (entradas, salidas, lógica del bloque principal), y en caso de haber necesitado hacer uso de elementos auxiliares (consultas, vistas, otros procedimientos/funciones...) también se debe incluir el diseño de estos (a menos que sean consultas triviales).

Este procedimiento actualiza el estado de todos los pedidos. Los que tengan estado de borrador ('D') pasan a confirmado ('P').

Aparte de esta funcionalidad, no tiene ninguna entrada, elemento auxiliar o consulta.

- b) Su implementación en SQL

```
PROCEDURE actualizar AS
BEGIN
    -- Actualizamos la tabla REPLACEMENTS

    UPDATE REPLACEMENTS
    SET STATUS = 'P'
    WHERE STATUS = 'D';

END;
```

- c) Pruebas

Ejecutamos `exec caffeine.actualizar()` y nos damos cuenta que el procedimiento ha hecho lo que tenía que hacer, cambiar todos los borradores a confirmados.

	TAXID	BARCODE	ORDERDATE	STATUS	UNITS	DELDATE	PAYMENT
1	J38320106J	00028882I889528	07/01/24	P	25	14/01/24	1750
2	J38320106J	00028882I889528	10/02/24	P	72	14/02/24	5277,6
3	J38320106J	00028882I889528	18/03/24	P	138	20/03/24	10446,6
4	J38320106J	OIQ47059I515289	07/01/24	P	22	18/01/24	402,6
5	J38320106J	OIQ47059I515289	05/02/24	P	89	14/02/24	2029,2
6	J38320106J	OIQ47059I515289	12/03/24	P	53	17/03/24	1128,9
7	J38320106J	IQQ999880827109	07/02/24	P	13	09/02/24	455
8	J38320106J	IQQ999880827109	14/02/24	P	53	15/02/24	2119,47
9	J38320106J	IQQ999880827109	28/03/24	P	168	30/03/24	7056
10	J38320106J	00028882I889528	22/01/07	P	25	14/01/24	1750
11	Q53972745W	III070570402995	07/01/24	P	25	14/01/24	1750
12	Q53972745W	III070570402995	10/02/24	P	72	14/02/24	5277,6
13	Q53972745W	III070570402995	18/03/24	P	138	20/03/24	10446,6

3.2 PROCEDURE info (proveedor in VARCHAR2)

- a) Su diseño (entradas, salidas, lógica del bloque principal), y en caso de haber necesitado hacer uso de elementos auxiliares (consultas, vistas, otros procedimientos/funciones...) también se debe incluir el diseño de estos (a menos que sean consultas triviales).

Este procedimiento coge los pedidos confirmados de un determinado proveedor que se pasa como argumento. Da una salida por pantalla que muestra el CIF del proveedor, el número de pedidos confirmados y el promedio del tiempo de entrega. Por otro lado, para cada uno de sus productos, imprime el coste actual, mínimo, máximo, medio, la diferencia entre el coste actual y el medio, y la diferencia entre el coste actual y el mínimo.

Para ello hemos usado dos consultas.

La primera es muy sencilla, simplemente agrupamos todas las ofertas del proveedor por el ID, el estado confirmado y que se hayan hecho en el último año. de esta agrupación sacamos el CIF, La media del tiempo de entrega y el número de pedidos confirmados en el último año.

La segunda consulta va dentro de un bucle, para mostrar los datos de cada producto que oferta el proveedor.

$\sigma_{\text{taxid} = \text{proveedor}, \text{status} = 'P', \text{orderdate} \geq \text{'último año'}} (\pi_{\text{taxid}, \text{avg}(\text{deldate} - \text{orderdate}), \text{count}('X')} \text{Group By taxid (REPLACEMENTS)})$

$\sigma_{\text{taxid} = \text{proveedor}, \text{status} = 'P', \text{orderdate} \geq \text{'último año'}} (\pi_{\text{min}(\text{payment} / \text{units}), \text{max}(\text{payment} / \text{units}), \text{avg}(\text{payment} / \text{units})} \text{ (REPLACEMENTS)})$

b) Su implementación en SQL

```
PROCEDURE info (proveedor in VARCHAR2) AS
BEGIN
    SELECT TAXID, AVG(deldate - orderdate), COUNT('X')
        INTO v_taxid, v_media_entrega, v_confirmados
        FROM REPLACEMENTS
            WHERE TAXID = proveedor AND STATUS = 'P' AND ORDERDATE >=
ADD_MONTHS(SYSDATE, -12) GROUP BY TAXID;

    DBMS_OUTPUT.PUT_LINE('Tax ID: ' || v_taxid || ', Pedidos confirmados: ' || v_confirmados
        || ', Promedio entrega: ' || v_media_entrega || ' días, Ofertas del proveedor: ');

    FOR fila IN (SELECT * FROM SUPPLY_LINES WHERE TAXID = proveedor)
    LOOP
        SELECT MIN(payment/units), MAX(payment/units), AVG(payment/units)
            INTO v_coste_min, v_coste_max, v_coste_media
            FROM REPLACEMENTS
```

```

        WHERE TAXID = proveedor AND STATUS = 'P' AND BARCODE =
fila.BARCODE AND ORDERDATE >= ADD_MONTHS(SYSDATE, -12);

        v_resta_media := fila.cost - v_coste_media;
        v_resta_min := fila.cost - v_coste_min;

        DBMS_OUTPUT.PUT_LINE('Barcode: ' || fila.barcode || ', Coste
actual: ' || fila.cost ||
        ', Coste mínimo: ' || v_coste_min || ', Coste máximo: ' ||
v_coste_max || ', Coste medio: ' || v_coste_media ||
        ', Coste actual - coste media: ' || v_resta_media ||
        ', Coste actual - coste mínimo: ' || v_resta_min);

    END LOOP;

EXCEPTION
when no_data_found then dbms_output.put_line('No hay datos');
when too_many_rows then dbms_output.put_line('Demasiadas filas');
when others then dbms_output.put_line('Se ha producido otro error');
END;
```

c) Pruebas

Primero ejecutamos el procedimiento anterior con `exec caffeine.actualizar()` para que los borradores pasen a ser pedidos confirmados. Después ejecutamos `SET SERVEROUTPUT ON`, para ver la salida del procedimiento por pantalla, y `exec caffeine.info('J38320106J')`, que devuelve la información del proveedor con el ID introducido como argumento. Devuelve:

Tax ID: J38320106J, Pedidos confirmados: 9, Promedio entrega: 4,78 días, Ofertas del proveedor:

Barcode: IQQ999880827109, Coste actual: 41,29, Coste mínimo: 35, Coste máximo: 42, Coste medio: 39, Coste actual - coste media: 2,29, Coste actual - coste mínimo: 6,29

Barcode: OIQ47059I515289, Coste actual: 20,82, Coste mínimo: 18,3, Coste máximo: 22,8, Coste medio: 20,8, Coste actual - coste media: ,02, Coste actual - coste mínimo: 2,52

Barcode: OOO28882I889528, Coste actual: 72,16, Coste mínimo: 70, Coste máximo: 75,7, Coste medio: 73, Coste actual - coste media: -,84, Coste actual - coste mínimo: 2,16

Procedimiento PL/SQL terminado correctamente.

4. Diseño externo

- **Vista ‘Mis_compras’:** En esta vista se muestran todas las compras de un usuario en específico, o en este caso del usuario actual.

a) Su diseño en álgebra relacional

a) $\sigma_{\text{username}=\text{USER}}(\text{CLIENT_LINES})$ Solo lectura

b) Su implementación en SQL

```
CREATE OR REPLACE VIEW Mis_compras AS
SELECT * FROM CLIENT_LINES
--WHERE USERNAME = USER
-- Para las pruebas
WHERE USERNAME = 'abby'
WITH READ ONLY;
```

- c) **Pruebas:** Para probar este código, hemos buscado en las tablas de compras en nombre de algún cliente para que pudiese funcionar. Posteriormente, hemos reemplazado temporalmente la función USER por el de este cliente y todo ha funcionado correctamente.

Base de datos Aulas Informáticas MIS_COMPRAS										
Columnas Datos Permisos Dependencias Detalles Disparadores SQL Errores										
	ORDERDATE	USERNAME	TOWN	COUNTRY	BARCODE	PRICE	QUANTITY	PAY_TYPE	PAY_DATETIME	CARDNUM
1	23/12/14	abby	Valtorres	Cook Islands	IQI099580821265	54,85 7		COD	25/12/14	(null)
2	23/12/14	abby	Valtorres	Cook Islands	OII990610387669	37,4 2		COD	25/12/14	(null)
3	24/11/18	abby	Ciudad del Robledal	Denmark	OQI29831I701907	18,9 8		COD	26/11/18	(null)
4	26/09/19	abby	Valtorres	Cook Islands	OQI546130493020	8,3 2		COD	28/09/19	(null)
5	06/05/17	abby	Valtorres	Cook Islands	000074250766667	4,85 36		COD	06/05/17	(null)
6	20/03/14	abby	Valtorres	Cook Islands	QII83306Q657842	31,85 4		COD	22/03/14	(null)
7	05/07/15	abby	Ciudad del Robledal	Denmark	IOI84906Q721225	28,99 8		COD	06/07/15	(null)
8	24/12/14	abby	Valtorres	Cook Islands	QQQ96063I470990	12,65 9		COD	25/12/14	(null)
9	16/08/15	abby	Ciudad del Robledal	Denmark	IOQ37063I759576	45,05 12		COD	18/08/15	(null)
10	20/03/14	abby	Valtorres	Cook Islands	QQQ18785Q986489	23,15 5		COD	22/03/14	(null)
11	10/04/18	abby	Valtorres	Cook Islands	OIQ93476Q682030	24,15 7		COD	11/04/18	(null)
12	26/09/19	abby	Valtorres	Cook Islands	OIO20279Q827300	2,5 9		COD	28/09/19	(null)
13	26/09/19	abby	Valtorres	Cook Islands	QQQ64961Q492127	78,1 14		COD	28/09/19	(null)
14	11/05/18	abby	Ciudad del Robledal	Denmark	IIQ144400273452	38,55 2		COD	14/05/18	(null)
15	23/12/14	abby	Valtorres	Cook Islands	OIO17505Q856187	1,5 8		COD	25/12/14	(null)
16	24/12/14	abby	Valtorres	Cook Islands	OII97326Q672619	34,6 13		COD	25/12/14	(null)
17	12/01/17	abby	Ciudad del Robledal	Denmark	OQI114550823803	6,8 3		COD	12/01/17	(null)
18	17/07/19	abby	Valtorres	Cook Islands	OIO810650540784	92,88 16		COD	18/07/19	(null)
19	16/07/19	abby	Valtorres	Cook Islands	IQQ46127Q226614	10,7 2		COD	18/07/19	(null)
20	16/07/19	abby	Valtorres	Cook Islands	QQQ66607I732727	4,3 3		COD	18/07/19	(null)
21	07/12/15	abby	Ciudad del Robledal	Denmark	QQQ46183I480600	12,75 9		COD	08/12/15	(null)
22	12/03/15	abby	Valtorres	Cook Islands	OIO9067I423900	108,06 25		COD	14/03/15	(null)
23	21/08/19	abby	Ciudad del Robledal	Denmark	QIO018570342186	1,55 5		COD	23/08/19	(null)
24	27/09/19	abby	Valtorres	Cook Islands	OII918850847857	2,4 6		COD	28/09/19	(null)
25	10/04/18	abby	Valtorres	Cook Islands	OIO38647I173366	30,45 27		COD	11/04/18	(null)
26	24/11/18	abby	Ciudad del Robledal	Denmark	III62697Q978531	132,15 27		COD	26/11/18	(null)
27	10/04/18	abby	Valtorres	Cook Islands	QII32920I216555	54,55 3		COD	11/04/18	(null)
28	17/08/15	abby	Ciudad del Robledal	Denmark	OIO65483I132968	90,75 3		COD	18/08/15	(null)
29	16/08/15	abby	Ciudad del Robledal	Denmark	OIQ822010212659	14,25 20		COD	18/08/15	(null)
30	16/08/15	abby	Ciudad del Robledal	Denmark	OII319780673570	2,85 2		COD	18/08/15	(null)

- **Vista ‘Mi_perfil’:** En esta vista se muestran todos los datos personales de un usuario en concreto o del usuario actual.

a) Su diseño en álgebra relacional

$$\rho_{AUX}(\sigma_{USERNAME=USER} \pi_{USERNAME, NAME, SURN1, CARDNUM} (CLIENTS \bowtie_{CARDNUM} CLIENT_CARDS)) \bowtie_{USERNAME} (CLIENT_ADDRESSES \bowtie_{AUX})$$

b) Su implementación en SQL

```
CREATE OR REPLACE VIEW Mi_perfil AS
WITH AUX AS (
    SELECT USERNAME, NAME, SURN1, CARDNUM
    FROM CLIENTS
    NATURAL JOIN CLIENT_CARDS)
SELECT * FROM AUX NATURAL JOIN CLIENT_ADDRESSES
-- WHERE USERNAME = USER
-- Para las pruebas
WHERE USERNAME = 'ojeda'
WITH READ ONLY;
```

- c) Pruebas:** Para probar este código, hemos buscado en las tablas de usuarios en nombre de algún cliente para que pudiese funcionar. Posteriormente, hemos reemplazado temporalmente el username por el de este cliente y todo ha funcionado correctamente.

USER...	NAME	SURN1	CARDNUM	WAYTYPE	WAYNAME	GATE	BLOCK	STAIRW	FLOOR	DOOR	ZIP	TOWN	COUNTRY
ojeda	Ambrosia	Ojeda	10173148360	Avenue	San Juan de la Cruz	104	(null)	(null)	20	(null)	19817	Villachimeneas	Norfolk Island

- **Vista a ‘mis_comentarios’ :** En esta vista enumeramos los comentarios de un usuario y le damos la posibilidad de insertar, eliminar o cambiar el texto de sus comentarios si y solo si sus likes son 0.

a) Su diseño en álgebra relacional

$$\sigma_{USERNAME=USER} \pi_{ROW_NUMBER(POSTDATE \uparrow), USERNAME, POSTDATE, BARCODE, PRODUCT, SCORE, TITLE, TEXT, LIKES, ENDORSED} (POSTS)$$

b) Su implementación en SQL

```
CREATE OR REPLACE VIEW mis_comentarios AS
SELECT ROW_NUMBER() OVER (ORDER BY POSTDATE ASC) as RN,
    USERNAME, POSTDATE, BARCODE, PRODUCT, SCORE, TITLE, TEXT, LIKES,
    ENDORSED
FROM POSTS
WHERE USERNAME = USER
WITH CHECK OPTION;
```

Disparador asociado:

```
CREATE OR REPLACE TRIGGER CHK_comentarios
BEFORE DELETE OR UPDATE OR INSERT ON POSTS
FOR EACH ROW
    --DECLARE mg NUMBER;
BEGIN
    --SELECT NEW.LIKES FROM POSTS WHERE LIKES = 0;
    if (:OLD.LIKES != 0) then
        RAISE_APPLICATION_ERROR(-20002, 'No puedes
eliminar/modificar/insertar si los likes no son 0');
    END IF;

    IF UPDATING('USERNAME') OR UPDATING('POSTDATE') OR
UPDATING('BARCODE') OR UPDATING('PRODUCT') OR UPDATING('SCORE') OR
UPDATING('TITLE') OR UPDATING('LIKES') OR UPDATING('ENDORSED') THEN
        RAISE_APPLICATION_ERROR(-20002, 'Solo se puede editar
texto');
    END IF;
END;
/
```

c) Pruebas:

Para probar la funcionalidad de este código, hemos cambiado a propósito dos likes a 0 de un usuario en concreto para así poder probar todas las combinaciones que nuestro código proporciona.

```
ALTER TRIGGER CHK_COMENTARIOS DISABLE;

UPDATE POSTS SET LIKES = 0 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '23/01/15';
UPDATE POSTS SET LIKES = 0 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21';
```

	USERNAME	POSTDATE	BARCODE	PRODUCT	SCORE	TITLE	TEXT	LIKES	ENDORSED
1	abelarda	23/...	IOQ...	Lug...		1Kd ...	TK ziLDn ...	0 (null)	
2	abelarda	05/...	(null)	Gla...		3Rqf...	IsfpvOltc...	0 (null)	
3	abelarda	24/...	OIO...	Pri...		1SnT...	VPtPour r...	890 (null)	
4	abelarda	28/...	(null)	Sil...		4VzZ...	ZNdhGnWeA...	2498 (null)	
5	abelarda	31/...	(null)	Guerra		1Gz ...	XvsaQuWVK...	2930 (null)	
6	abelarda	28/...	(null)	Sereno		2Mky...	WPSPkqf r...	3311 (null)	
7	abelarda	24/...	OOQ...	Mar...		5YPQ...	TqFQjSoL ...	3955 (null)	
8	abelarda	02/...	OIQ...	Reves		3YWQ...	SWPsYbVXJ...	6609 (null)	
9	abelarda	16/...	OIO...	Tw...		2TQU...	JtJLP AQH...	7100 (null)	
10	abelarda	24/...	(null)	Hom...		4RQw...	FNk wYmSD...	10948 (null)	

Después, hemos habilitado el trigger de vuelta y hemos probado a borrar un comentario con 0 likes y vemos que si va.

```
ALTER TRIGGER CHK_COMENTARIOS ENABLE;

DELETE FROM POSTS WHERE USERNAME = 'abelarda' and TO_CHAR(POSTDATE) =
'23/01/15'; -- Se elimina una fila porque los likes son 0
```

También, hemos probado a cambiar algo más que el texto en un comentario que si tiene 0 likes, y vemos que no deja porque solo se puede borrar el texto.

```
UPDATE POSTS SET LIKES = 0 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'; -- Error solo se puede editar texto
UPDATE POSTS SET SCORE = 3 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'; -- Error solo se puede editar texto
UPDATE POSTS SET TEXT = 'HOLA', SCORE = 3 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'; -- Error solo se puede editar texto
```

Error que empieza en la línea: 507 del comando :

```
UPDATE POSTS SET LIKES = 0 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'
Informe de error -
ORA-20002: Solo se puede editar texto
ORA-06512: en "FSDB139.CHK_COMENTARIOS", línea 9
ORA-04088: error durante la ejecución del disparador
'FSDB139.CHK_COMENTARIOS'
```

Error que empieza en la línea: 508 del comando :

```
UPDATE POSTS SET SCORE = 3 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'
Informe de error -
ORA-20002: Solo se puede editar texto
ORA-06512: en "FSDB139.CHK_COMENTARIOS", línea 9
```


ORA-04088: error durante la ejecución del disparador
'FSDB139.CHK_COMENTARIOS'

Error que empieza en la línea: 509 del comando :

UPDATE POSTS SET TEXT = 'HOLA', SCORE = 3 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'

Informe de error -

ORA-20002: Solo se puede editar texto

ORA-06512: en "FSDB139.CHK_COMENTARIOS", línea 9

ORA-04088: error durante la ejecución del disparador
'FSDB139.CHK_COMENTARIOS'

Por último, intentamos editar el texto de un comentario que si tiene 0 likes y todo funciona correctamente.

```
UPDATE POSTS SET TEXT = 'HOLA' WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'; -- Se modifica con éxito
```

5. Disparadores

- a) $\sigma_{\text{username} = :new_username, \text{barcode} = :new_barcode} \pi_{\text{count}('x')}(\text{CLIENT_LINES})$
- b) $\sigma_{\text{username} = :old_username} \pi_{\text{orderdate}, v_contacto1, v_contacto2, \text{dliv_datetime}, :old_vname, :old_sum1, :old_sum2, \text{waytype}, \text{wayname}, \dots}(\text{CLIENT_ADDRESSES} * \text{ORDERS_CLIENTS})$
- $\sigma_{\text{username} = :old_username} \pi_{\text{orderdate}, v_contacto1, \text{town}, \text{country}, \text{barcode}, \text{price}, \text{quantity}, \text{pay_type}, \text{pay_datetime}, \text{card_comp}, \text{cardnum}, \text{card_holder}, \text{card_expir}}(\text{CLIENT_LINES} * \text{CLIENT_CARDS})$
- $\sigma_{\text{username} = :old_username} \pi_{\text{postdate}, \text{barcode}, \text{product}, \text{score}, \text{title}, \text{text}, \text{likes}, \text{endorse}}(\text{POSTS})$
- c) $\sigma_{\text{cardnum} = :new_card_num} \pi_{\text{count}('x')}(\text{CLIENT_CARDS})$
- d) $\sigma_{\text{barcode} = :new_barcode} \pi_{\text{cur_stock}, \text{min_stock}, \text{max_stock}}(\text{References})$
- $\sigma_{\text{barcode} = :new_barcode} \pi_{\text{min}(\text{cost})} \text{Group by Barcode}(\text{Supply_Lines})$
- * Errores 4.2 misma consulta sobre Likes-endorse

Para cada disparador resuelto se debe incluir una subsección que contenga:

a) TRIGGER_1:

- **Descripción del diseño:** Se dispara antes de que se inserte o edite un comentario en la tabla 'POSTS'. El disparador va fila por fila y busca si el cliente que está actualizando o insertando el comentario ha comprado el producto del que habla. Si se encuentra una compra de ese cliente a dicho producto en la tabla CLIENT_LINES, el atributo

endorsed de POST pasará a ser 'Y', en caso contrario, será 'N'. Antes de probar el disparador alteramos la tabla POST para que 'endorsed' sea de tipo VARCHAR, ya que originalmente era de tipo DATE.

- **Código (en SQL):**

```
ALTER TABLE POSTS MODIFY ENDORSED VARCHAR2(1);

CREATE OR REPLACE TRIGGER trigger_1
  BEFORE UPDATE OR INSERT ON POSTS
  FOR EACH ROW
  DECLARE v_endorsed NUMBER;
  BEGIN

    SELECT COUNT('X') INTO v_endorsed FROM CLIENT_LINES
    WHERE USERNAME = :NEW.USERNAME AND BARCODE = :NEW.BARCODE;

    IF v_endorsed >= 1 THEN
      :NEW.ENDORSED := 'Y';
    ELSE
      :NEW.ENDORSED := 'N';
    END IF;
  END;
/
```

- **Pruebas:**

Primero probamos las inserciones:

```
INSERT INTO POSTS (USERNAME, POSTDATE, BARCODE, PRODUCT, SCORE, TEXT,
LIKES) VALUES ('austro', SYSDATE, 'Q0014099I548032', 'Puercos',
3, 'QUEASE', 10); -- INSERTA 1 FILA CON ENDORSED = 'Y' PORQUE AUSTRO HA
COMPRADO ANTERIORMENTE ESE PRODUCTO

INSERT INTO POSTS (USERNAME, POSTDATE, BARCODE, PRODUCT, SCORE, TEXT,
LIKES) VALUES ('austro', TO_DATE('10/04/24', 'DD/MM/YY'),
'OIQ75670Q167198', 'Milagros o joyas',
3, 'QUEASE', 10); -- INSERTA 1 FILA CON ENDORSED = 'N' PORQUE AUSTRO NO
HA COMPRADO ANTERIORMENTE ESE PRODUCTO
```

USE...	POSTDATE	BARCODE	PRODUCT	SCORE	TITLE	TEXT	LIKES	ENDORSED
1 austro	12/04/24	Q0014099I548032	Puercos	3 (null)		QUEASE	10	Y
2 austro	10/04/24	OIQ75670Q167198	Milagros o joyas	3 (null)		QUEASE	10	N

Luego probamos las actualizaciones:

Para la primera, nos fijamos en un usuario que tenga comentarios de productos que haya comprado. Nosotros elegimos a 'abelarda', que tiene varios comentarios como los siguientes.

	USER...	POSTDATE	BARCODE	PRODUCT	SCORE	TITLE	TEXT	LIKES	ENDORSED
1	abelarda	23/01/15	IOQ4445...	Lugaren...		1 Kd KMSb...	TK ziLD...	32406	(null)
2	abelarda	06/02/15	III9080...	Ninos		2 MpL hPu...	QRdTSzq...	11235	(null)
3	abelarda	08/02/15	OQ03267...	Princip...		4 XhBMX n...	YsDHVhZ...	33601	(null)
4	abelarda	09/02/15	IOI1004...	Caballo...		2 OUwIsr ...	Z1P GWK...	13311	(null)
5	abelarda	10/02/15	OOQ7539...	Oficial...		3 LCzwBjK...	PcJohPM...	22671	(null)
6	abelarda	11/02/15	(null)	Sirena ...		5 AsDT Mk...	Gkrvwoc...	25238	(null)
7	abelarda	17/02/15	QII8563...	Discusion		4 DNvR	CbiUQNL...	21981	(null)

Ejecutamos el update de uno de ellos:

```
UPDATE POSTS SET TEXT = 'HOLA' WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '10/02/15';
-- ACTUALIZA 1 FILA CON ENDORSED = 'Y' PORQUE AUSTRO HA COMPRADO
ANTERIORMENTE ESE PRODUCTO
```

Y nos fijamos de que haya hecho bien los cambios, que en este caso era poner en el cuerpo del comentario 'HOLA' y que se actualice la columna de 'endorsed'.

	USER...	POSTDATE	BARCODE	PRODUCT	SCORE	TITLE	TEXT	LIKES	ENDORSED
1	abelarda	23/01/15	IOQ4445...	Lugaren...		1 Kd KMSb...	TK ziLD...	32406	(null)
2	abelarda	06/02/15	III9080...	Ninos		2 MpL hPu...	QRdTSzq...	11235	(null)
3	abelarda	08/02/15	OQ03267...	Princip...		4 XhBMX n...	YsDHVhZ...	33601	(null)
4	abelarda	09/02/15	IOI1004...	Caballo...		2 OUwIsr ...	Z1P GWK...	13311	(null)
5	abelarda	10/02/15	OOQ7539...	Oficial...		3 LCzwBjK...	HOLA	22671	Y
6	abelarda	11/02/15	(null)	Sirena ...		5 AsDT Mk...	Gkrvwoc...	25238	(null)
7	abelarda	17/02/15	QII8563...	Discusion		4 DNvR	CbiUQNL...	21981	(null)

También hay que probar el caso de que no haya comprado el producto con anterioridad.

Para ello, modificamos uno de los comentarios que hemos introducido antes y nos damos cuenta de que el texto del comentario ha cambiado pero la columna endorsed no, ya que el usuario sigue sin haber comprado el producto.

```
UPDATE POSTS SET TEXT = 'asdfasfasdfasd' WHERE USERNAME = 'austro' and
TO_CHAR(POSTDATE) = '10/04/24';
-- ACTUALIZA 1 FILA CON ENDORSED = 'N' PORQUE AUSTRO NO HA COMPRADO
ANTERIORMENTE ESE PRODUCTO
```

	USER...	POSTDATE	BARCODE	PRODUCT	SCORE	TITLE	TEXT	LIKES	ENDORSED
1	austro	10/04/24	OIQ7567...	Milagro...	3 (null)		asdfasfasdfasd	10	N
2	austro	12/04/24	QO01409...	Puercos	3 (null)		QUEASE	10	Y

b) TRIGGER_2:

- **Descripción del diseño:** Se dispara antes de borrar un cliente de la tabla CLIENTS . Su objetivo es trasladar las compras y comentarios del usuario registrado, a otras tablas de usuarios no registrados. El disparador va fila por fila y busca todos los datos necesarios del cliente eliminado para realizar la inserción en las tablas de usuarios anónimos . Como en las tablas proporcionadas no se guarda la dirección de facturación (solo el país y pueblo), hemos dado por hecho que es la misma que la de entrega. Antes de probar el disparador alteramos las tablas CLIENT_ADDRESSES y CLIENT_CARDS para tengan las mismas restricciones originales pero sin el borrado en cascada, para evitar el error de tabla mutante. Obviamente al quitar este tipo de borrado tenemos que eliminar los datos relacionados con esas tablas manualmente.

- **Código (en SQL):**

```
ALTER TABLE CLIENT_ADDRESSES
DROP CONSTRAINT fk_addresses_clients;
ALTER TABLE CLIENT_ADDRESSES
ADD CONSTRAINT fk_addresses_clients FOREIGN KEY(username) REFERENCES
Clients;

ALTER TABLE CLIENT_CARDS
DROP CONSTRAINT fk_cards_clients;
ALTER TABLE CLIENT_CARDS
ADD CONSTRAINT fk_cards_clients FOREIGN KEY(username) REFERENCES Clients;

CREATE OR REPLACE TRIGGER trigger_2
  BEFORE DELETE ON CLIENTS
  FOR EACH ROW
  DECLARE v_contacto1 VARCHAR2(60);
          v_contacto2 VARCHAR2(60);
  BEGIN
    IF :OLD.PREFERENCE = 'EMAIL' THEN
      v_contacto1 := :OLD.EMAIL;
      v_contacto2 := TO_CHAR(:OLD.MOBILE);
    ELSIF :OLD.PREFERENCE = 'SMS' THEN
      v_contacto1 := TO_CHAR(:OLD.MOBILE);
      v_contacto1 := :OLD.EMAIL;
    END IF;

    INSERT INTO ORDERS_ANONYM(
```

```

        SELECT ORDERDATE, v_contacto1, v_contacto2, DLIV_DATETIME,
:OLD.NAME, :OLD.SURN1, :OLD.SURN2, WAYTYPE, WAYNAME, GATE, BLOCK, STAIRW,
FLOOR, DOOR, ZIP, TOWN, COUNTRY, WAYTYPE, WAYNAME, GATE, BLOCK, STAIRW,
FLOOR, DOOR, ZIP, TOWN, COUNTRY
        FROM CLIENT_ADDRESSES NATURAL JOIN ORDERS_CLIENTS
        WHERE USERNAME = :OLD.USERNAME);

INSERT INTO LINES_ANONYM(
        SELECT ORDERDATE, v_contacto1, TOWN, COUNTRY, BARCODE, PRICE,
QUANTITY, PAY_TYPE, PAY_DATETIME, CARD_COMP, CARDNUM, CARD HOLDER,
CARD_EXPIR
        FROM CLIENT_LINES NATURAL JOIN CLIENT_CARDS
        WHERE USERNAME = :OLD.USERNAME);

INSERT INTO ANONYPOSTS(
        SELECT POSTDATE, BARCODE, PRODUCT, SCORE, TITLE, TEXT, LIKES,
ENDORSED
        FROM POSTS
        WHERE USERNAME = :OLD.USERNAME);

DELETE FROM Orders_Clients
        WHERE (username, town, country) IN (SELECT username, town,
country FROM Client_Addresses WHERE username = :OLD.USERNAME);

DELETE FROM Posts WHERE username = :OLD.USERNAME;
DELETE FROM CLIENT_ADDRESSES WHERE USERNAME = :OLD.USERNAME;
DELETE FROM CLIENT_CARDS WHERE USERNAME = :OLD.USERNAME;
DELETE FROM CLIENT_LINES WHERE USERNAME = :OLD.USERNAME;
END;
/

```

- Pruebas:

DELETE FROM CLIENTS WHERE USERNAME = 'pedro';

CLIENTS:

USERN...	REG_DATETIME	USER_PASSW	NAME	SURN1	SURN2	EMAIL	MOBILE	PREFERENCE	VOUCHER	VOUCHER_EXP
1 pedro	01/06/14	ywsXYvdRH	Gerardo Pedro	Campos	Barrios	pedro@serviciocorreo.ucetreseme.kom	555387192 EMAIL			0 (null)

Memoria Práctica 2 – Consultas, bloques, diseño externo y disparadores



ORDERS_CLIENTS:

	ORDERDATE	USER...	TOWN	COUNTRY	DLIV_DATETIME	BILL_TOWN	BILL_COUNTRY	DISCOUNT
1	23/11/21	pedro	Val de las Setas	Nepal	24/11/21	Val de las Setas	Nepal	0
2	06/09/16	pedro	Val de las Setas	Nepal	07/09/16	Val de las Setas	Nepal	0
3	07/09/16	pedro	Val de las Setas	Nepal	08/09/16	Val de las Setas	Nepal	0
4	15/08/18	pedro	Val de las Setas	Nepal	16/08/18	Val de las Setas	Nepal	0
5	15/08/18	pedro	Val de las Setas	Nepal	16/08/18	Val de las Setas	Nepal	0
6	14/08/18	pedro	Val de las Setas	Nepal	16/08/18	Val de las Setas	Nepal	0
7	09/05/22	pedro	Val de las Setas	Nepal	10/05/22	Val de las Setas	Nepal	0
8	23/03/15	pedro	Val de las Setas	Nepal	25/03/15	Val de las Setas	Nepal	0
9	23/03/15	pedro	Val de las Setas	Nepal	25/03/15	Val de las Setas	Nepal	0
10	05/10/15	pedro	Val de las Setas	Nepal	06/10/15	Val de las Setas	Nepal	0

POSTS:

USER...	POSTDATE	BARCODE	PRODUCT	SCORE	TITLE	TEXT	LIKES	ENDORSED
1 pedro	22/08/14	OII4164...	Verdad		5 ANHSNF Vj	YYAVATewsEN MCrX...	8205 (null)	
2 pedro	29/11/14	(null)	Ciervo ...		1 Fdgmong...	EaSJHy JyYK vvdcd...	14782 (null)	
3 pedro	23/02/15	IQI7196...	Verdad		5 Q2vjG b...	OcsRiR UqsDzdpfk...	14482 (null)	
4 pedro	14/09/16	(null)	Pegado ...		1 ZYjbHSF...	AUApaPBKO RiCvpo...	14398 (null)	
5 pedro	30/09/16	OII3851...	Princip...		4 KUF OZb...	TZpwFjSOh oSxz u...	27761 (null)	
6 pedro	04/06/19	OQI4606...	Tomatera		2 GDTbHi ...	MwKQX stREb OpiI...	16745 (null)	
7 pedro	25/11/21	OQI3830...	Sonador...		4 IZzdWPpy...	IdNoHL iETBjCMz ...	19603 (null)	
8 pedro	28/11/21	(null)	Oficial...		1 IUFsVft...	GqCLCE iLZqjpfXW...	29675 (null)	
9 pedro	01/12/21	(null)	Venus		5 QwDIxim...	BxgI fgZavYrkp x...	7130 (null)	
10 pedro	07/12/21	(null)	Bandolero		4 ZVvMO	EOdVrAfah f FBTo ...	13413 (null)	
11 pedro	08/12/21	(null)	Lugaren...		1 UvUWqLS...	UfE dUJp Xesy Hr...	2053 (null)	
12 pedro	10/12/21	OIQ1652...	Gracia		2 Qc UZYy...	RwViIPbLbAZ A Wn...	14585 (null)	
13 pedro	11/12/21	(null)	Cantar ...		2 LmHsk Z...	CjB Vsjlpa dQDCA...	17244 (null)	
14 pedro	17/12/21	OQI7784...	Razones...		1 AuoRWX ...	EjfVvw s UelcpQ ...	27220 (null)	
15 pedro	17/12/21	IQQ9068...	Enigma		5 BxYsFzx...	AHsCXn ORFckout ...	7615 (null)	
16 pedro	19/12/21	OII2283...	Vendedor		1 EmSZhEn...	TGZjprGwjJ ZdUnF...	4604 (null)	
17 pedro	22/12/21	(null)	Valla		1 FePc y ...	JpYhR gA SeC xKR...	38842 (null)	
18 pedro	01/01/22	IIO2712...	Cantar		2 XpTjvvx	QIvzHBeldDX QtFG...	35620 (null)	
19 pedro	02/01/22	OOO9021...	Perdon		4 JwZVKHB...	GyL ptvRU QQ fJ ...	26237 (null)	

CLIENT ADDRESSES:

USER...	WAYTYPE	WAYNAME	GATE	BLOCK	STAIRW	FLOOR	DOOR	ZIP	TOWN	COUNTRY
1 pedro	Circus	Chesterton	107	(null)	(null)	8	1	99220	Val de las Setas	Nepal

CLIENT CARDS:

	CARDNUM	USE...	CARD_CO...	CARD_HO...	CARD_EX...
1	4436277...	pedro	Andorra...	G Pedro...	01/05/26
2	9350376...	pedro	Cabestro	G Pedro...	01/02/27

Después del borrado de pedro, las anteriores quedan vacías y en las de usuarios no registrados se pasan sus datos:

ORDERS ANONYM:

ORDERDATE	CONTACT	CONTACT2	DLIV_DAT...	NAME	SURN1	SURN2	BILL_W...	BILL_WA...	B...	BELL...	BILL...	BILL...	BILL_ZIP	BILL_TOWN	BILL_COU...	DLIV_WA...	DLIV_WA...	DL...	DLIV...	DLIV...	DLIV...	DLIV_ZIP	DLIV_TOWN	DLIV_CO...
1 02/06/14	pedsro...	555387192 04/06/14		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	
2 02/06/14	pedsro...	555387192 04/06/14		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	
3 02/06/14	pedsro...	555387192 04/06/14		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	
4 02/06/14	pedsro...	555387192 04/06/14		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	
5 03/06/14	pedsro...	555387192 03/06/14		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	
6 03/06/14	pedsro...	555387192 03/06/14		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	
7 01/08/14	pedsro...	555387192 04/08/14		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	
8 01/08/14	pedsro...	555387192 04/08/14		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	
9 02/08/14	pedsro...	555387192 04/08/14		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	
10 07/09/16	pedsro...	555387192 08/09/16		Gerardo Pedro	Campos	Barrios	Circus...	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	... Circus	Chester...	107	(null)	(null)	8	1 99220	Val de ... Nepal	...	

LINES_ANONYM:

ORDERDATE	CONTACT	DLIV_TOWN	DLIV_COUNTRY	BARCODE	PRICE	QUANTITY	PAY_TYPE	PAY_DATETIME	CARD_COMP	CARD_NUM	CARD HOLDER	CARD_EXPIR
1 02/06/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	OII35480Q569349	36,75	8	credit card	02/06/14	Andorran Stress	443627774696 G	Pedro Campos	01/05/26
2 02/06/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	II013320I758183	39,8	7	credit card	02/06/14	Andorran Stress	443627774696 G	Pedro Campos	01/05/26
3 02/06/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	OI089244Q896049	2	4	credit card	02/06/14	Andorran Stress	443627774696 G	Pedro Campos	01/05/26
4 02/06/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	QI0737190399122	12,45	3	credit card	02/06/14	Andorran Stress	443627774696 G	Pedro Campos	01/05/26
5 03/06/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	IQI34789198893	8,1	13	credit card	03/06/14	Andorran Stress	443627774696 G	Pedro Campos	01/05/26
6 03/06/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	OQ0831390452544	2,99	2	credit card	03/06/14	Andorran Stress	443627774696 G	Pedro Campos	01/05/26
7 01/08/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	OQ0860751584314	16,55	3	credit card	01/08/14	Cabestro	935037699551 G	Pedro Campos	01/02/27
8 01/08/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	IOI64831I132968	90,75	1	credit card	01/08/14	Cabestro	935037699551 G	Pedro Campos	01/02/27
9 03/09/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	IQ0776110115881	18,55	4	credit card	03/09/14	Cabestro	935037699551 G	Pedro Campos	01/02/27
10 03/09/14	pedro@servooorreo.uoetreseme.kom	Val de las Setas	Nepal	00082944Q821067	120,95	23	credit card	03/09/14	Cabestro	935037699551 G	Pedro Campos	01/02/27

ANNONYMPOSTS:

POSTDATE	BARCODE	PRODUCT	SCORE	TITLE	TEXT	LIKES	ENDORSED
1 28/11/21	(null)	Oficial y trovar	1	IUFsVftAdZ COYGosGmtY ISwKs oilfal arkHaw	GqCLCE iLZqjpfXWJ fFd bcF QP azPoZIU...	29675	(null)
2 01/12/21	(null)	Venus	5	QwDIXimGHTV R apg nmNcU EoD mj	BxgI fgZavYrkp xRiFgUiuC HRkSOIUQ I ...	7130	(null)
3 07/12/21	(null)	Bandolero	4	ZVvMO	EodVrAfh f FBTo c JbhwWpWY NGmY JV...	13413	(null)
4 08/12/21	(null)	Lugarenos torero	1	UvUWqLSjs GPHlo qNDVog E	UFE dUJp Xesy Hrfxrx jmZ Fxate dBvbw...	2053	(null)
5 10/12/21	OIQ16525I403549	Gracia	2	Qc UZYYSZUG MMjQjnCvjm	RwViPbLbAZ A Wnbk Loro HNgK zZJD ...	14585	(null)
6 11/12/21	(null)	Cantar de lejanía	2	LmHsk ZN muMGvRycZ mXaEY	CjB VajlpA dQDCAJ fmLu AbwV vd Nbmkv...	17244	(null)
7 17/12/21	OQI778450117018	Razones y margaritas	1	AuoRWX Z cBy LZ EQ TCSltgt	EjfvVw s UelcpQ jAEgydX u tENyav jv ...	27220	(null)
8 17/12/21	IQQ90681Q408278	Enigma	5	BxYsFzBv GNtkwOYUW HhOUEm pvcIvJnBtm cRZJu	AHsCXn ORFckout JqJuPn Wqptg AHEW gj...	7615	(null)
9 19/12/21	OII22831Q738220	Vendedor	1	EmSZhEngpsZ EaQdZLr gt	TGZjprGwj ZdUnFLr pK OohV LZpwr sUX...	4604	(null)
10 22/12/21	(null)	Valla	1	FePc y YePflRhd sP BUWDAEz UuLGddZN uletrTlP	JpYhR gA SeC xKrBiosyLS Ck QQQOXpd E...	38842	(null)
11 01/01/22	II0271250889794	Cantar	2	XpTjvvx	QIvzHBeldDX QtFG DgxBoPdVg OUBOFZor ...	35620	(null)
12 02/01/22	O00902101895216	Perdon	4	JWZVKHbGIf ZnmsIU Yc anFbr	GVL pteVRU QQ fJ vjUrMbOKYP VNIhr NDY...	26237	(null)
13 22/08/14	OII416440108939	Verdad	5	ANHSNF Vj	YYAVATewsEN MCrX WeDQZk c wNDUAKtyf ...	8205	(null)
14 29/11/14	(null)	Ciervo rock	1	FdgmonG o gnX	EaSJHy JyYK vvdcqMN Ds Mv Nvy FrvCDF...	14782	(null)
15 23/02/15	IQI71960I571975	Verdad	5	QZvjG bfbG bKbXh K XnUnILLsQ kIziBsZcOh	OcsRiR UgsDsdpfkn EDjfpdPBnp o gwQED...	14482	(null)
16 14/09/16	(null)	Pegado o zaleo	1	ZYjbHSPdhiU n h QPftBzXc	AUApaPBKO RTcVpocU LLNZGI HAbaoTvi g...	14398	(null)
17 30/09/16	OII38510Q613606	Principio triunfo	4	KUF OZbmyHt jfewOgBx auOeOUT	TZpwFjSOH oSxz u VzHXjCU niZAsMN nEY...	27761	(null)
18 04/06/19	OQI460610493876	Tomatera	2	GDtBhi zIbpoP k kdVnzM	MwKQX stREe OpiGn HKDQIRgs ShmqAese...	16745	(null)
19 25/11/21	OQI38308Q863270	Sonador de pasar	4	IzZdwPyrr SnqDRVCqBM Emlu NWyZAc ZNcrfeatE	IdNoHL iETBjCmZ OyTh qQijw wufocpAZB...	19603	(null)

c) TRIGGER_3:

- **Descripción del diseño:** Se dispara antes de que se inserten compras anónimas. Se ejecuta fila a fila y su objetivo es impedir que un usuario no registrado realice una compra con la tarjeta de un usuario registrado. Para ello miramos si la tarjeta introducida está en la tabla de 'CLIENT_CARDS'. Si no lo está, no hace nada y hace el insert normal y si está, manda una excepción que impide que se haga la inserción.

- Código (en SQL):

```
CREATE OR REPLACE TRIGGER trigger_3
BEFORE INSERT ON LINES_ANONYM
FOR EACH ROW
DECLARE v_tarjeta NUMBER;
BEGIN
    SELECT COUNT('X') INTO v_tarjeta FROM CLIENT_CARDS
    WHERE CARDNUM = :NEW.CARD_NUM;

    if v_tarjeta > 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'La tarjeta ya está
registrada como tarjeta de cliente');
    END IF;
```



```
END;
```

```
/
```

- Pruebas:

Tarjeta ya registrada:

```
INSERT INTO LINES_ANONYM VALUES (SYSDATE, 'diagi@servcorreo.ucetreseme.kom', 'Val de las Caballerizas', 'Namibia', 'QOO04562I911080', '1,15', '2', 'credit card', '25/06/16', 'Cabestro', 10694341628, 'M Altagracia Lago', '01/03/26');
```

La tarjeta 10694341628 ya existe, por lo que nos devuelve:

Error que empieza en la línea: 497 del comando :

```
INSERT INTO LINES_ANONYM VALUES(SYSDATE, 'diagi@servcorreo.ucetreseme.kom', 'Val de las Caballerizas', 'Namibia', 'QOO04562I911080', '1,15', '2', 'credit card', '25/06/16', 'Cabestro', 10694341628, 'M Altagracia Lago', '01/03/26')
```

Informe de error -

ORA-20002: La tarjeta ya está registrada como tarjeta de cliente

ORA-06512: en "FSDB139.TRIGGER_3", línea 7

ORA-04088: error durante la ejecución del disparador 'FSDB139.TRIGGER_3'

Tarjeta no registrada:

Insertamos primero en 'ORDERS_ANONYM' para que no haya problemas de foreign key.

Es importante notar que tenemos que hacer las dos inserciones a la vez para que compartan el SYSDATE.

```
INSERT INTO ORDERS_ANONYM VALUES (sysdate, '1234@gmail.com', '', '', 'John', 'Doe', '', 'Street', '123 Main St', 'A', '', '', '', '12345', 'UC3M', 'Spain', 'Calle', 'Nano', 'C', '', '', '13579', 'UC3M', 'Spain');

INSERT INTO LINES_ANONYM VALUES (sysdate, '1234@gmail.com', 'UC3M', 'Spain', 'QII85636I380417', 13, 2, 'credit card', SYSDATE, 'khkhgklghkl', 333333333, 'Mariogod', sysdate);
```

LINES_ANONYM:

ORDERDATE	CONTACT	DLIV_TOWN	DLIV_COUNTRY	BARCODE	PRICE	QUANTITY	PAY_TYPE	PAY_DATETIME	CARD_COMP	CARD_NUM	CARD HOLDER	CARD_EXPIR
12/04/24	1234@gmail.com	UC3M	Spain	QII856...	13	2	credi...	12/04/24	khkhgk...	3333...	Mariogod	12/0...

d) TRIGGER_4:

- **Descripción del diseño:** Se dispara después de que se inserte una compra en la tabla 'CLIENT_LINES' o 'LINES_ANOMYN' y restará el stock del producto dependiendo de las unidades que se han comprado. Si el stock se queda por debajo del mínimo, se hará un pedido en 'REPLACEMENTS' para rellenarlo hasta el máximo con el mínimo del precio que ofrezcan los proveedores.

Con el código de barras del producto, buscamos su stock máximo en 'REFERENCES' y el precio mínimo al que se oferta en 'SUPPLY_LINES'

No hemos encontrado la manera de hacer que salte un disparador al insertar en una tabla, por lo que hemos hecho dos idénticos cambiando la tabla que hace que se dispare.

- **Código (en SQL):**

```
ALTER TABLE REPLACEMENTS
DROP CONSTRAINT pk_replacements; -- Elimina la clave primaria que deseas
eliminar

-- Agregar una nueva clave primaria
ALTER TABLE REPLACEMENTS
ADD CONSTRAINT nueva_pk PRIMARY KEY (barcode,orderdate);
--Agrega una nueva clave primaria con las columnas deseadas

create or replace TRIGGER trigger_4_1
AFTER INSERT ON CLIENT_LINES
--OR AFTER INSERT ON ANONYM_LINES
FOR EACH ROW
DECLARE v_stock NUMBER;
        v_min_stock NUMBER;
        v_max_stock NUMBER;
        v_resta NUMBER;
        v_coste_unidad NUMBER;
BEGIN

        SELECT CUR_STOCK, MIN_STOCK, MAX_STOCK INTO v_stock, v_min_stock,
v_max_stock FROM REFERENCES
        WHERE BARCODE = :NEW.BARCODE;

        SELECT MIN(COST) INTO v_coste_unidad FROM SUPPLY_LINES
        WHERE BARCODE = :NEW.BARCODE GROUP BY BARCODE;

        v_resta := v_stock - :NEW.QUANTITY;
        if v_resta >= v_min_stock then
```



```
        UPDATE REFERENCES
        SET CUR_STOCK = CUR_STOCK - :NEW.QUANTITY;
    else
        UPDATE REFERENCES
        SET CUR_STOCK = CUR_STOCK - :NEW.QUANTITY;
        INSERT INTO REPLACEMENTS values(NULL, :NEW.BARCODE, SYSDATE, 'D',
v_max_stock - v_resta, NULL, v_coste_unidad * :NEW.QUANTITY);

    END IF;
END;
/

create or replace TRIGGER trigger_4_2
AFTER INSERT ON LINES_ANONYM
--OR AFTER INSERT ON ANONYM_LINES
FOR EACH ROW
DECLARE v_stock NUMBER;
        v_min_stock NUMBER;
        v_max_stock NUMBER;
        v_resta NUMBER;
        v_coste_unidad NUMBER;
BEGIN

        SELECT CUR_STOCK, MIN_STOCK, MAX_STOCK INTO v_stock, v_min_stock,
v_max_stock FROM REFERENCES
        WHERE BARCODE = :NEW.BARCODE;

        SELECT MIN(COST) INTO v_coste_unidad FROM SUPPLY_LINES
        WHERE BARCODE = :NEW.BARCODE GROUP BY BARCODE;

        v_resta := v_stock - :NEW.QUANTITY;
        if v_resta >= v_min_stock then
            UPDATE REFERENCES
            SET CUR_STOCK = CUR_STOCK - :NEW.QUANTITY;

        else
            UPDATE REFERENCES
            SET CUR_STOCK = CUR_STOCK - :NEW.QUANTITY;
            INSERT INTO REPLACEMENTS values(NULL, :NEW.BARCODE, SYSDATE, 'D',
v_max_stock - v_resta, NULL, v_coste_unidad * :NEW.QUANTITY);
```

```
END IF;  
END;  
/
```

- Pruebas:

Insertamos algo conocido en 'ORDERS_CLIENTS' para que no de error de foreign key.

```
insert into orders_clients  
values(TO_DATE('20/02/23'), 'cristi', 'Serrania de las Tripas', 'Congo',  
NULL, 'Serrania de las Tripas', 'Congo', NULL);
```

En referencias con el barcode = III01816Q633880 está este producto:

BARCODE	PRODUCT	FORMAT	PACK_TYPE	PACK_UNIT	QUANTITY	PRICE	CUR_STOCK	MIN_STOCK	MAX_STOCK
1 III01816Q633880	Gardenias de fic...	P	cup	ml.	200	1,99	1438	20	3780

Insertamos una compra de dos unidades de este producto:

```
-- Se actualiza el stock en referencias  
insert into client_lines  
values(TO_DATE('20/02/23'), 'cristi', 'Serrania de las Tripas', 'Congo',  
'III01816Q633880', 13, 2, 'COD', SYSDATE, NULL);
```

En la tabla 'REFERENCES' se restan dos unidades del stock actual.

BARCODE	PRODUCT	FORMAT	PACK_TYPE	PACK_UNIT	QUANTITY	PRICE	CUR_STOCK	MIN_STOCK	MAX_STOCK
1 III01816Q633880	Gardenias de fic...	P	cup	ml.	200	1,99	1436	20	3780

Borramos la compra para insertarla nuevamente pero con más unidades para que se quede debajo del mínimo.

```
delete from client_lines  
where username = 'cristi' and orderdate = TO_DATE('20/02/23') and town =  
'Serrania de las Tripas' and country = 'Congo';
```

Modificamos el tipo de dato de 'QUANTITY' para que se puedan comprar más de 99 unidades e insertamos la compra.

```
-- Inicialmente hay 1438 unidades y el stock mínimo es 20  
ALTER TABLE CLIENT_LINES MODIFY QUANTITY VARCHAR2(10);  
insert into client_lines  
values(TO_DATE('20/02/23'), 'cristi', 'Serrania de las Tripas', 'Congo',  
'III01816Q633880', 13, 1425, 'COD', SYSDATE, NULL);
```

BARCODE	PRODUCT	FORMAT	PACK_TYPE	PACK_UNIT	QUANTITY	PRICE	CUR_STOCK	MIN_STOCK	MAX_STOCK
1 III01816Q633880	Gardenias de fic...	P	cup	ml.	200	1,99	11	20	3780

Como las unidades de stock actual están por debajo del mínimo se ha hecho la siguiente inserción en la tabla de 'REPLACEMENTS'

	TAXID	BARCODE	ORDERDATE	STATUS	UNITS	DELDATE	PAYMENT
1	(null)	III01816Q633880	12/04/24	D	3769	(null)	2322,75

En nuestro diseño hemos decidido pedir el número de unidades hasta llegar al máximo, en este caso $3780 - 11 = 3769$.

Cabe destacar que para la correcta ejecución de todo el código, cada vez que ejecutamos un trigger y sus pruebas correspondientes, deshabilitamos el trigger. Esto se debe a que algunos triggers se solapan entre sí, porque uno no permite borrar comentarios si sus likes no son 0, pero otro dice que si se elimina un usuario, toda su información ha de desaparecer. Por ello, para evitar estos problemas, se habilitan y deshabilitan en función de lo que queremos probar.

6. Conclusiones

Gracias a las decisiones de diseño implementadas a lo largo de todo el trabajo, hemos conseguido una solución sólida y robusta para cubrir todas las necesidades de la empresa que nos ha contratado. Todas las implementaciones, consultas y controladores funcionan correctamente y son versátiles para cubrir cualquier necesidad tanto válida como no válida.

En cuanto al esfuerzo realizado, nos parecen unas prácticas excesivamente largas para lo que valen de la nota final. Es incoherente que prácticas que pueden requerir semanas de trabajo se valoren como la mitad de un examen corto tipo test. Para futuros años, recomendamos que la práctica se acorte un poco o que tenga una mayor valoración. Por otro lado, han sido unas prácticas muy didácticas donde hemos aprendido diversos conocimientos y a trabajar en equipo. Otra recomendación sería disminuir los datos que hay en algunas tablas proporcionadas. No son necesarios más de 50.000 datos para poder probar la solución. Esto generaba que algunos días no se pudiera trabajar en el código, ya que la base de datos dejaba de funcionar.

7. Scrip

```
-----INICIAMOS LA SALIDA DEL SERVIDOR-----

SET SERVEROUTPUT ON;

-----CONSULTA 1-----

WITH AUX AS (
    SELECT BARCODE, QUANTITY, PRICE, ORDERDATE, COUNTRY FROM CLIENT_LINES
    WHERE ORDERDATE > ADD_MONTHS(SYSDATE, -12)
    UNION
    SELECT BARCODE, TO_CHAR(QUANTITY), PRICE, ORDERDATE, DLIV_COUNTRY
FROM LINES_ANONYM
    WHERE ORDERDATE > ADD_MONTHS(SYSDATE, -12)
),

AUX2 AS (
    SELECT BARCODE, COUNT('X') AS N_COMPRADORES, SUM(QUANTITY) AS VENTAS,
    PRICE, SUM(QUANTITY)*PRICE AS INGRESOS, ROUND(AVG(QUANTITY),2) AS
    PROMEDIO_VENTAS
    FROM AUX
    GROUP BY BARCODE, PRICE
),

-- Sirve para mezclar los barcodes y saber el varietal de cada uno
A AS (
    SELECT DISTINCT barcode, varietal, origin FROM PRODUCTS NATURAL JOIN
    REFERENCES
),

-- Sirve para mezclar la subconsulta anterior con los datos de cada
varietal
ORDEN_VENTAS AS (
    SELECT ORIGIN, VARIETAL, N_COMPRADORES, VENTAS, PRICE, INGRESOS,
    PROMEDIO_VENTAS,
        ROW_NUMBER() OVER (PARTITION BY ORIGIN ORDER BY VENTAS DESC)
AS RN
```

```

        FROM A NATURAL JOIN AUX2
    ),
    --SELECT ORIGIN, VARIETAL, VENTAS, RN FROM VENTAS_RANKED; --Para ver como
    se hace la partición
    MAX_VENTAS_VARIETAL AS (
        SELECT ORIGIN, VARIETAL, VENTAS, N_COMPRADORES, PRICE, INGRESOS,
        PROMEDIO_VENTAS
        FROM ORDEN_VENTAS
        WHERE RN = 1
    ),

    -- Cantidades totales del varietal por país comprador
    CONSUMIDORES_1 AS (
        SELECT VARIETAL, COUNTRY, SUM(QUANTITY) AS CANTIDAD
        FROM A NATURAL JOIN AUX GROUP BY VARIETAL, COUNTRY
    ),

    -- Se añade la columna con máximo de ventas por varietal
    CONSUMIDORES_2 AS (
        SELECT VARIETAL, COUNTRY, CANTIDAD, VENTAS,
        ROUND(CANTIDAD/VENTAS,3)*100 AS PORCENTAJE
        FROM CONSUMIDORES_1
        NATURAL JOIN MAX_VENTAS_VARIETAL
    ),

    -- Filtras por los mayores del 1%
    CONSUMIDORES_3 AS (
        SELECT VARIETAL, COUNT('X') AS PAISES_POTENCIALES FROM CONSUMIDORES_2
        WHERE PORCENTAJE > 1 GROUP BY VARIETAL
    )

    SELECT ORIGIN, VARIETAL, VENTAS, N_COMPRADORES, PRICE, INGRESOS,
    PROMEDIO_VENTAS, PAISES_POTENCIALES
    FROM MAX_VENTAS_VARIETAL NATURAL JOIN CONSUMIDORES_3;

    -----CONSULTA 2-----

    WITH AUX1 AS (

```

```

SELECT BARCODE, QUANTITY, PRICE, SUBSTR(ORDERDATE, 4, 2) AS MES FROM
CLIENT_LINES
WHERE ORDERDATE > ADD_MONTHS(SYSDATE, -12)
UNION
SELECT BARCODE, TO_CHAR(QUANTITY), PRICE, SUBSTR(ORDERDATE, 4, 2) AS
MES FROM LINES_ANONYM
WHERE ORDERDATE > ADD_MONTHS(SYSDATE, -12)
),

AUX2 AS (
SELECT MES, BARCODE, COUNT('X') AS N_PEDIDOS, SUM(QUANTITY) AS
VENTAS, SUM(QUANTITY)*PRICE AS INGRESOS
FROM AUX1
GROUP BY BARCODE, MES, PRICE
),

AUX3 AS (
SELECT DISTINCT MES, MAX(VENTAS) AS VENTAS FROM AUX2 GROUP BY MES
),

AUX4 AS (
SELECT MES, BARCODE, N_PEDIDOS, VENTAS, INGRESOS FROM AUX3 NATURAL
JOIN AUX2
)

SELECT MES, BARCODE, N_PEDIDOS, VENTAS, INGRESOS, INGRESOS - VENTAS *
MAX(COST) AS BENEFICIO FROM AUX4 NATURAL JOIN SUPPLY_LINES
GROUP BY MES, BARCODE, N_PEDIDOS, VENTAS, INGRESOS ORDER BY MES;

-----PAQUETE-----

DELETE FROM REPLACEMENTS;
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('J38320106J','00028882I889528', TO_DATE('07/01/24', 'DD/MM/YY'),
25, TO_DATE('14/01/24'), 70*25);
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)

```

```
VALUES ('J38320106J','00028882I889528', TO_DATE('10/02/24', 'DD/MM/YY'),
72, TO_DATE('14/02/24'), 73.3*72);
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('J38320106J','00028882I889528', TO_DATE('18/03/24', 'DD/MM/YY'),
138, TO_DATE('20/03/24'), 75.7*138);

INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('J38320106J','OIQ47059I515289', TO_DATE('07/01/24', 'DD/MM/YY'),
22, TO_DATE('18/01/24'), 18.3*22);
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('J38320106J','OIQ47059I515289', TO_DATE('5/02/24', 'DD/MM/YY'),
89, TO_DATE('14/02/24'), 22.8*89);
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('J38320106J','OIQ47059I515289', TO_DATE('12/03/24', 'DD/MM/YY'),
53, TO_DATE('17/03/24'), 21.3*53);

INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('J38320106J','IQQ999880827109', TO_DATE('07/02/24', 'DD/MM/YY'),
13, TO_DATE('09/02/24'), 35*13);
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('J38320106J','IQQ999880827109', TO_DATE('14/02/24', 'DD/MM/YY'),
53, TO_DATE('15/02/24'), 39.99*53);
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('J38320106J','IQQ999880827109', TO_DATE('28/03/24', 'DD/MM/YY'),
168, TO_DATE('30/03/24'), 42*168);

INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('J38320106J','00028882I889528', TO_DATE('22/01/07', 'DD/MM/YY'),
25, TO_DATE('14/01/24'), 70*25);

-- Añadimos otros proveedores a la base de datos
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
```

```
VALUES ('Q53972745W','III070570402995', TO_DATE('07/01/24', 'DD/MM/YY'),
25, TO_DATE('14/01/24'), 70*25);
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('Q53972745W','III070570402995', TO_DATE('10/02/24', 'DD/MM/YY'),
72, TO_DATE('14/02/24'), 73.3*72);
INSERT INTO REPLACEMENTS(taxID, barCode, orderdate, units, deldate,
payment)
VALUES ('Q53972745W','III070570402995', TO_DATE('18/03/24', 'DD/MM/YY'),
138, TO_DATE('20/03/24'), 75.7*138);

CREATE OR REPLACE PACKAGE caffeine AS
    PROCEDURE actualizar;
    PROCEDURE info (proveedor in VARCHAR2);

END caffeine;
/

CREATE OR REPLACE PACKAGE BODY caffeine AS

    -- Definir variables para almacenar los resultados

    v_taxid REPLACEMENTS.TAXID%TYPE;
    v_confirmados NUMBER;
    v_media_entrega NUMBER(5,2);
    v_coste_media NUMBER(5,2);
    v_coste_min NUMBER;
    v_coste_max NUMBER;
    v_resto_media NUMBER;
    v_resto_min NUMBER;

    PROCEDURE actualizar AS
    BEGIN
        -- Actualizamos la tabla REPLACEMENTS

        UPDATE REPLACEMENTS
        SET STATUS = 'P'
        WHERE STATUS = 'D';

    END;

    PROCEDURE info (proveedor in VARCHAR2) AS
    BEGIN
```



```
SELECT TAXID, AVG(deldate - orderdate), COUNT('X')
    INTO v_taxid, v_media_entrega, v_confirmados
    FROM REPLACEMENTS
    WHERE TAXID = proveedor AND STATUS = 'P' AND ORDERDATE >=
ADD_MONTHS(SYSDATE, -12) GROUP BY TAXID;

DBMS_OUTPUT.PUT_LINE('Tax ID: ' || v_taxid || ', Pedidos confirmados: '
|| v_confirmados
|| ', Promedio entrega: ' || v_media_entrega || ' días, Ofertas del
proveedor:');

FOR fila IN (SELECT * FROM SUPPLY_LINES WHERE TAXID = proveedor)
LOOP
    SELECT MIN(payment/units), MAX(payment/units), AVG(payment/units)
    INTO v_coste_min, v_coste_max, v_coste_media
    FROM REPLACEMENTS
    WHERE TAXID = proveedor AND STATUS = 'P' AND BARCODE =
fila.BARCODE AND ORDERDATE >= ADD_MONTHS(SYSDATE, -12);
    v_resta_media := fila.cost - v_coste_media;
    v_resta_min := fila.cost - v_coste_min;

    DBMS_OUTPUT.PUT_LINE('Barcode: ' || fila.barcode || ', Coste
actual: ' || fila.cost ||
    ', Coste mínimo: ' || v_coste_min || ', Coste máximo: ' ||
v_coste_max || ', Coste medio: ' || v_coste_media ||
    ', Coste actual - coste media: ' || v_resta_media ||
    ', Coste actual - coste mínimo: ' || v_resta_min);

END LOOP;

EXCEPTION
when no_data_found then dbms_output.put_line('No hay datos');
when too_many_rows then dbms_output.put_line('Demasiadas filas');
when others then dbms_output.put_line('Se ha producido otro error');
END;

END caffeine;

/

exec caffeine.actualizar();
exec caffeine.info('J38320106J');
```

-----VISTA 1-----

```
CREATE OR REPLACE VIEW Mis_compras AS
  SELECT * FROM CLIENT_LINES
  WHERE USERNAME = USER
  -- Para las pruebas
  --WHERE USERNAME = 'abby'
  WITH READ ONLY;
```

-----VISTA 2-----

```
CREATE OR REPLACE VIEW Mi_perfil AS
  WITH AUX AS (
    SELECT USERNAME, NAME, SURN1, CARDNUM
    FROM CLIENTS
    NATURAL JOIN CLIENT_CARDS)
  SELECT * FROM AUX NATURAL JOIN CLIENT_ADDRESSES
  WHERE USERNAME = USER
  -- Para las pruebas
  -- WHERE USERNAME = 'ojeda'
  WITH READ ONLY;
```

-----VISTA 3-----

```
CREATE OR REPLACE VIEW mis_comentarios AS
  SELECT ROW_NUMBER() OVER(ORDER BY POSTDATE ASC) as RN,
  USERNAME, POSTDATE, BARCODE, PRODUCT, SCORE, TITLE, TEXT, LIKES,
  ENDORSED
  FROM POSTS
  WHERE USERNAME = USER
  WITH CHECK OPTION;
```

```
CREATE OR REPLACE TRIGGER CHK_comentarios
  BEFORE DELETE OR UPDATE OR INSERT ON POSTS
```

```

FOR EACH ROW
    --DECLARE mg NUMBER;

BEGIN
    --SELECT NEW.LIKES FROM POSTS WHERE LIKES = 0;
    if (:OLD.LIKES != 0) then
        RAISE_APPLICATION_ERROR(-20002, 'No puedes
eliminar/modificar/insertar si los likes no son 0');
    END IF;

    IF UPDATING('USERNAME') OR UPDATING('POSTDATE') OR
UPDATING('BARCODE') OR UPDATING('PRODUCT') OR UPDATING('SCORE') OR
    UPDATING('TITLE') OR UPDATING('LIKES') OR UPDATING('ENDORSED')
THEN
        RAISE_APPLICATION_ERROR(-20002, 'Solo se puede editar
texto');
    END IF;
END;
/

ALTER TRIGGER CHK_COMENTARIOS DISABLE;

UPDATE POSTS SET LIKES = 0 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '23/01/15';
UPDATE POSTS SET LIKES = 0 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21';

ALTER TRIGGER CHK_COMENTARIOS ENABLE;

DELETE FROM POSTS WHERE USERNAME = 'abelarda' and TO_CHAR(POSTDATE) =
'23/01/15'; -- Se elimina una fila porque los likes son 0
UPDATE POSTS SET LIKES = 0 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'; -- Error solo se puede editar texto
UPDATE POSTS SET SCORE = 3 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'; -- Error solo se puede editar texto
UPDATE POSTS SET TEXT = 'HOLA', SCORE = 3 WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'; -- Error solo se puede editar texto
UPDATE POSTS SET TEXT = 'HOLA' WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '05/06/21'; -- Se modifica con éxito

ALTER TRIGGER CHK_COMENTARIOS DISABLE;
    
```

```

-----TRIGGER 1-----

ALTER TABLE POSTS MODIFY ENDORSED VARCHAR2(1);

CREATE OR REPLACE TRIGGER trigger_1
  BEFORE UPDATE OR INSERT ON POSTS
  FOR EACH ROW
    DECLARE v_endorsed NUMBER;
  BEGIN

    SELECT COUNT('X') INTO v_endorsed FROM CLIENT_LINES
    WHERE USERNAME = :NEW.USERNAME AND BARCODE = :NEW.BARCODE;

    IF v_endorsed >= 1 THEN
      :NEW.ENDORSED := 'Y';
    ELSE
      :NEW.ENDORSED := 'N';
    END IF;
  END;

/

INSERT INTO POSTS(USERNAME, POSTDATE, BARCODE, PRODUCT, SCORE, TEXT,
LIKES) VALUES ('austro', SYSDATE, 'Q0014099I548032', 'Puercos',
3, 'QUEASE', 10); -- INSERTA 1 FILA CON ENDORSED = 'Y' PORQUE AUSTRO HA
COMPRADO ANTERIORMENTE ESE PRODUCTO
INSERT INTO POSTS(USERNAME, POSTDATE, BARCODE, PRODUCT, SCORE, TEXT,
LIKES) VALUES ('austro', TO_DATE('10/04/24', 'DD/MM/YY'),
'OIQ75670Q167198', 'Milagros o joyas',
3, 'QUEASE', 10); -- INSERTA 1 FILA CON ENDORSED = 'N' PORQUE AUSTRO NO
HA COMPRADO ANTERIORMENTE ESE PRODUCTO
UPDATE POSTS SET TEXT = 'HOLA' WHERE USERNAME = 'abelarda' and
TO_CHAR(POSTDATE) = '10/02/15';
-- ACTUALIZA 1 FILA CON ENDORSED = 'Y' PORQUE AUSTRO HA COMPRADO
ANTERIORMENTE ESE PRODUCTO
UPDATE POSTS SET TEXT = 'asdfasdfasdfasd' WHERE USERNAME = 'austro' and
TO_CHAR(POSTDATE) = '10/04/24';
-- ACTUALIZA 1 FILA CON ENDORSED = 'N' PORQUE AUSTRO NO HA COMPRADO
ANTERIORMENTE ESE PRODUCTO

```

```
ALTER TRIGGER trigger_1 DISABLE;

-----TRIGGER 2-----

ALTER TABLE CLIENT_ADDRESSES
DROP CONSTRAINT fk_addresses_clients;
ALTER TABLE CLIENT_ADDRESSES
ADD CONSTRAINT fk_addresses_clients FOREIGN KEY(username) REFERENCES
Clients;

ALTER TABLE CLIENT_CARDS
DROP CONSTRAINT fk_cards_clients;
ALTER TABLE CLIENT_CARDS
ADD CONSTRAINT fk_cards_clients FOREIGN KEY(username) REFERENCES Clients;

CREATE OR REPLACE TRIGGER trigger_2
  BEFORE DELETE ON CLIENTS
  FOR EACH ROW
  DECLARE v_contacto1 VARCHAR2(60);
          v_contacto2 VARCHAR2(60);
  BEGIN
    IF :OLD.PREFERENCE = 'EMAIL' THEN
      v_contacto1 := :OLD.EMAIL;
      v_contacto2 := TO_CHAR(:OLD.MOBILE);
    ELSIF :OLD.PREFERENCE = 'SMS' THEN
      v_contacto1 := TO_CHAR(:OLD.MOBILE);
      v_contacto1 := :OLD.EMAIL;
    END IF;

    INSERT INTO ORDERS_ANONYM(
      SELECT ORDERDATE, v_contacto1, v_contacto2, DLIV_DATETIME,
:OLD.NAME, :OLD.SURN1, :OLD.SURN2,
      WAYTYPE, WAYNAME, GATE, BLOCK, STAIRW, FLOOR, DOOR, ZIP,
TOWN, COUNTRY,
      WAYTYPE, WAYNAME, GATE, BLOCK, STAIRW, FLOOR, DOOR, ZIP,
TOWN, COUNTRY
      FROM CLIENT_ADDRESSES NATURAL JOIN ORDERS_CLIENTS
      WHERE USERNAME = :OLD.USERNAME);
```

```
INSERT INTO LINES_ANONYM(
    SELECT ORDERDATE, v_contacto1, TOWN, COUNTRY, BARCODE, PRICE,
QUANTITY,
    PAY_TYPE, PAY_DATETIME, CARD_COMP, CARDNUM, CARD_HOLDER,
CARD_EXPIR
    FROM CLIENT_LINES NATURAL JOIN CLIENT_CARDS
    WHERE USERNAME = :OLD.USERNAME);

INSERT INTO ANONYPOSTS(
    SELECT POSTDATE, BARCODE, PRODUCT, SCORE, TITLE, TEXT, LIKES,
ENDORSED
    FROM POSTS
    WHERE USERNAME = :OLD.USERNAME);

DELETE FROM Orders_Clients
    WHERE (username, town, country) IN (SELECT username, town,
country FROM Client_Addresses WHERE username = :OLD.USERNAME);

DELETE FROM Posts WHERE username = :OLD.USERNAME;
DELETE FROM CLIENT_ADDRESSES WHERE USERNAME = :OLD.USERNAME;
DELETE FROM CLIENT_CARDS WHERE USERNAME = :OLD.USERNAME;
DELETE FROM CLIENT_LINES WHERE USERNAME = :OLD.USERNAME;
END;
/

DELETE FROM CLIENTS WHERE USERNAME = 'alberto';
DELETE FROM CLIENTS WHERE USERNAME = 'arce';
DELETE FROM CLIENTS WHERE USERNAME = 'pedro';

ALTER TRIGGER trigger_2 DISABLE;

-----TRIGGER 3-----

CREATE OR REPLACE TRIGGER trigger_3
    BEFORE INSERT ON LINES_ANONYM
    FOR EACH ROW
    DECLARE v_tarjeta NUMBER;
    BEGIN
```

```

SELECT COUNT('X') INTO v_tarjeta FROM CLIENT_CARDS
WHERE CARDNUM = :NEW.CARD_NUM;

if v_tarjeta > 0 THEN
    RAISE_APPLICATION_ERROR(-20002, 'La tarjeta ya está
registrada como tarjeta de cliente');
END IF;
END;
/

-- No puedes hacer la compra porque la tarjeta ya pertenece a un cliente
registrado -> Error: "La tarjeta ya está registrada como tarjeta de
cliente"
INSERT INTO LINES_ANONYM VALUES (SYSDATE,
'diagi@servcorreo.ucetreseme.kom', 'Val de las Caballerizas', 'Namibia',
'QQO04562I911080', '1,15', '2', 'credit card', '25/06/16', 'Cabestro',
10694341628, 'M Altagracia Lago', '01/03/26');

INSERT INTO ORDERS_ANONYM VALUES (sysdate, '1234@gmail.com', '', '',
'John', 'Doe', '', 'Street', '123 Main St', 'A',
'', '', '', '', '12345', 'UC3M', 'Spain', 'Calle', 'Nano', 'C', '', '',
'', '', '13579', 'UC3M', 'Spain');

INSERT INTO LINES_ANONYM VALUES (sysdate, '1234@gmail.com', 'UC3M',
'Spain', 'QII85636I380417', 13, 2, 'credit card', SYSDATE, 'khkhgklghkl',
333333333, 'Mariogod', sysdate);

ALTER TRIGGER trigger_3 DISABLE;

-----TRIGGER 4-----

ALTER TABLE REPLACEMENTS
DROP CONSTRAINT pk_replacements; -- Elimina la clave primaria que deseas
eliminar

-- Agregar una nueva clave primaria
ALTER TABLE REPLACEMENTS
ADD CONSTRAINT nueva_pk PRIMARY KEY (barcode,orderdate);
--Agrega una nueva clave primaria con las columnas deseadas

```

```
create or replace TRIGGER trigger_4_1
AFTER INSERT ON CLIENT_LINES
--OR AFTER INSERT ON ANONYM_LINES
FOR EACH ROW
DECLARE v_stock NUMBER;
        v_min_stock NUMBER;
        v_max_stock NUMBER;
        v_resta NUMBER;
        v_coste_unidad NUMBER;
BEGIN

        SELECT CUR_STOCK, MIN_STOCK, MAX_STOCK INTO v_stock, v_min_stock,
v_max_stock FROM REFERENCES
        WHERE BARCODE = :NEW.BARCODE;

        SELECT MIN(COST) INTO v_coste_unidad FROM SUPPLY_LINES
        WHERE BARCODE = :NEW.BARCODE GROUP BY BARCODE;

        v_resta := v_stock - :NEW.QUANTITY;
        if v_resta >= v_min_stock then
            UPDATE REFERENCES
            SET CUR_STOCK = CUR_STOCK - :NEW.QUANTITY;

        else
            UPDATE REFERENCES
            SET CUR_STOCK = CUR_STOCK - :NEW.QUANTITY;
            INSERT INTO REPLACEMENTS values(NULL, :NEW.BARCODE, SYSDATE, 'D',
v_max_stock - v_resta, NULL, v_coste_unidad * :NEW.QUANTITY);

        END IF;
END;
/

create or replace TRIGGER trigger_4_2
AFTER INSERT ON LINES_ANONYM
--OR AFTER INSERT ON ANONYM_LINES
FOR EACH ROW
DECLARE v_stock NUMBER;
        v_min_stock NUMBER;
```



```

        v_max_stock NUMBER;
        v_resta NUMBER;
        v_coste_unidad NUMBER;
BEGIN

    SELECT CUR_STOCK, MIN_STOCK, MAX_STOCK INTO v_stock, v_min_stock,
v_max_stock FROM REFERENCES
    WHERE BARCODE = :NEW.BARCODE;

    SELECT MIN(COST) INTO v_coste_unidad FROM SUPPLY_LINES
    WHERE BARCODE = :NEW.BARCODE GROUP BY BARCODE;

    v_resta := v_stock - :NEW.QUANTITY;
    if v_resta >= v_min_stock then
        UPDATE REFERENCES
        SET CUR_STOCK = CUR_STOCK - :NEW.QUANTITY;

    else
        UPDATE REFERENCES
        SET CUR_STOCK = CUR_STOCK - :NEW.QUANTITY;
        INSERT INTO REPLACEMENTS values(NULL, :NEW.BARCODE, SYSDATE, 'D',
v_max_stock - v_resta, NULL, v_coste_unidad * :NEW.QUANTITY);

    END IF;
END;
/
insert into orders_clients
values(TO_DATE('20/02/23'), 'cristi', 'Serrania de las Tripas', 'Congo',
NULL, 'Serrania de las Tripas', 'Congo', NULL);

-- Se actualiza el stock en referencias
insert into client_lines
values(TO_DATE('20/02/23'), 'cristi', 'Serrania de las Tripas', 'Congo',
'III01816Q633880', 13, 2, 'COD', SYSDATE, NULL);

delete from client_lines
where username = 'cristi' and orderdate = TO_DATE('20/02/23') and town =
'Serrania de las Tripas' and country = 'Congo';

-- Inicialmente hay 1438 unidades y el stock mínimo es 20
ALTER TABLE CLIENT_LINES MODIFY QUANTITY VARCHAR2(10);

```

Titulación: GRADO INGENIERIA INFORMATICA

Año Académico: 2023/24 -- Curso: 2º

Asignatura: Ficheros y Bases de Datos

Memoria Práctica 2 – Consultas, bloques, diseño externo y disparadores



uc3m | Universidad Carlos III de Madrid

```
insert into client_lines
values(TO_DATE('20/02/23'), 'cristi', 'Serrania de las Tripas', 'Congo',
'III01816Q633880', 13, 1425, 'COD', SYSDATE, NULL);

ALTER TRIGGER trigger_4_1 DISABLE;
ALTER TRIGGER trigger_4_2 DISABLE;
```