

RA2 (17%)

Evaluación de los criterios:

- f) 25%. Se han desarrollado aplicaciones que modifican el contenido de la base de datos. EJERCICIO3
 h) 20%. Se han desarrollado aplicaciones que efectúan consultas. EJERCICIO1
 i) 25%. Se han ejecutado procedimientos en la base de datos. EJERCICIO2

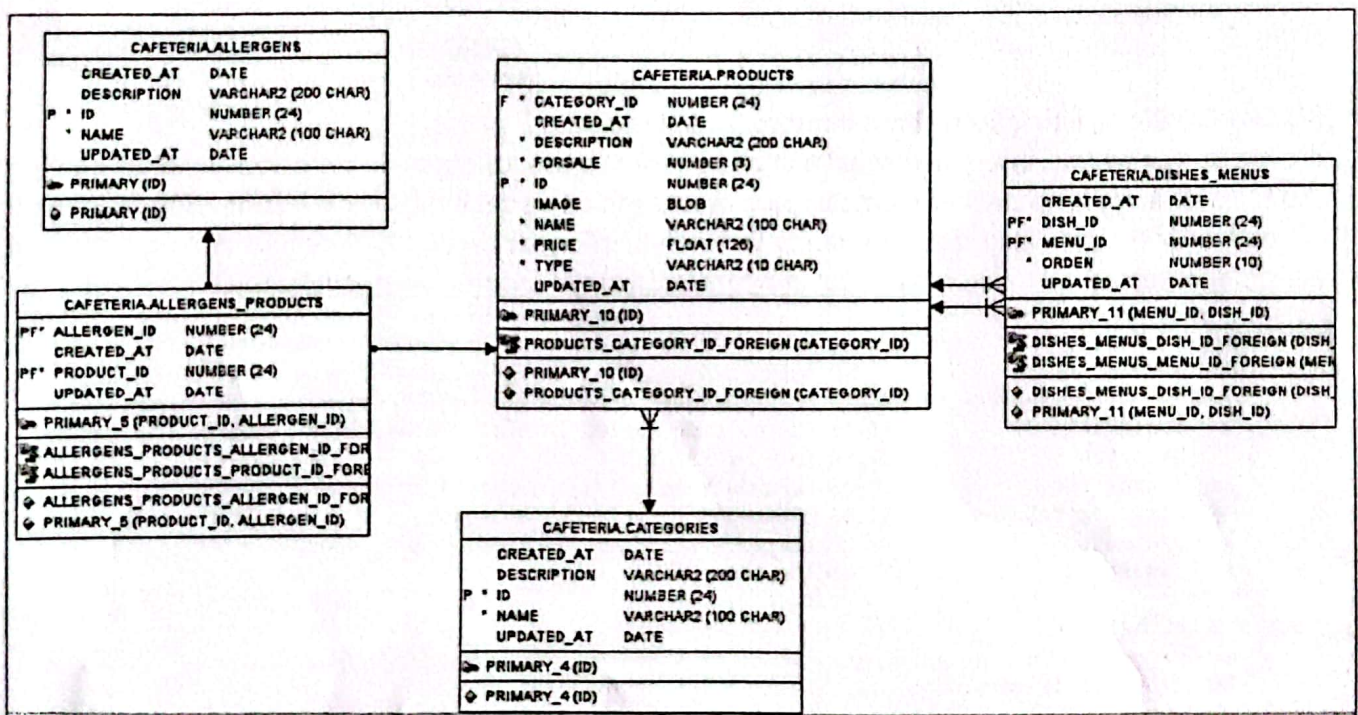
Una cafetería dispone de un catálogo de productos. Los productos pueden ser de dos tipos (*type*): **plato** y **menu**. Los datos de un producto son nombre (*name*), descripción (*description*), precio (*price*), una imagen (*image*) y si el producto está disponible para llevar o no (*for sale*).

Los productos **menu** deben indicar de qué platos están compuestos, junto con el orden de salida de cada plato. Un ejemplo es el siguiente "menú la primavera" tendrá vinculado los siguientes platos:

- Macarrones, con orden 1.
- Filete de Buey, con orden 2.
- Pechuga de pollo, con orden 3.

Cada producto puede pertenecer a una **categoría**, de cada categoría se quiere guardar su nombre (*name*) y una breve descripción (*description*). Cada producto puede también contener **alérgenos** y de cada alérgeno se quiere almacenar su nombre (*name*) y una breve descripción (*description*).

Toda esta información la tenemos almacenada en una base de datos ORACLE. Las tablas son las siguientes:



- **PRODUCTS:** contiene los datos de los productos que oferta la cafetería que pueden ser platos y menus. La clave es el *id*. Los productos pertenecen a una categoría, clave ajena *category_id*. Cada registro contiene un plato o un menú la columna *type* indica si se trata de plato o de menú.
- **ALLERGENS:** contiene la información de los alérgenos que se pueden presentar en los platos. La clave de cada registro es el campo *id*.
- **ALLERGENS_PRODUCTS:** contiene la información de los alérgenos que tienen los productos. Cada registro tiene dos columnas que además es la clave primaria de la tabla: *product_id* y *allergen_id* que son claves ajenas a las tablas **PRODUCTS** Y **ALLERGENS**.
- **CATEGORIES:** contiene la información de las categorías de los productos. Cada registro se identifica por el campo *id*.
- **DISHES_MENU:** contiene la información de los platos que hay en cada menú con el orden de salida del plato. La clave primaria está formada por los campos *menú_id* + *dish_id*.

En todas las tablas las columnas *created_at* y *updated_at* indican la fecha de creación y modificación de las tablas por defecto son nulas. No se usan para nada.

CREA EN ORACLE UN USUARIO DE NOMBRE **CAFETERIA** Y LA CONTRASEÑA **CAFETERIA** Y OTÓRGALE LOS PERMISOS NECESARIOS (ÓRDENES PARA CREAR USUARIO, primera línea versión 18 oracle express):

```
ALTER SESSION SET "_ORACLE_SCRIPT"=TRUE;
```

```
CREATE USER CAFETERIA IDENTIFIED BY CAFETERIA DEFAULT TABLESPACE USERS QUOTA 50M ON USERS;
```

```
GRANT DBA TO CAFETERIA;;
```

CREA LAS TABLAS PROPORCIONADAS POR EL SCRIPT.

Conexión para acceder a la BD (en mayúscula el usuario y la clave):

```
Connection conexion = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE",
        "CAFETERIA", "CAFETERIA");
```

A partir de estas tablas. Crear un proyecto Java de nombre **RA2RECU_APELLIDOSNOMBRE** y realiza los siguientes ejercicios (cada ejercicio es una clase Java):

SE PIDE:

(H) EJERCICIO1: Ejercicio1 ApellidosNombre.java. Consultas

Realiza una clase Java que muestre en pantalla un listado con los datos de los platos y menús que hay en cada una de las categorías. Por ejemplo, para la categoría 1 se debe mostrar la información de los platos y de los menús en esa categoría, mostrando también la información de los alérgenos en los platos y en los menús el orden de salida de cada plato, la información solicitada es la siguiente:

CATEGORIA: 1, Pasta				
PLATOS	ID	NOMBRE	PVP	ALERGENOS
=====				
	1	Espaguetis	12,00	(Leche y productos lácteos, Cacahuetes, Semillas y granos)
	2	Macarrones	11,00	(Huevos, Soja)
	3	Tortellinis	15,00	(Frutos secos, Trigo y otros cereales)
	4	Pasta de colores	12,00	(Pescados y mariscos, Mostaza)
	5	Raviolis	10,00	(Leche y productos lácteos, Huevos, Frutas y verduras)
	6	Lasaña	11,00	(Frutos secos)
=====				
MENÚS	ID	NOMBRE	ORDEN	PLATO
=====				
	39	Especial de la semana (45,00 €)	1	Tortellinis
			2	Ternera
			3	Higado encebollado

Si la categoría no tiene menús se debe mostrar un mensaje indicándolo. Y si no tiene platos también se debe mostrar un mensaje indicándolo. Ejemplo:

```
CATEGORIA: 7, Chuches
<<NO EXISTEN PLATOS EN LA CATEGORIA>>

<<NO EXISTEN MENÚS EN LA CATEGORIA>>
```

El listado debe quedar alineado y justificado, las cantidades numéricas deben mostrar la coma de los decimales. Listado ordenado por el id Categoría, y dentro por el id producto.

(I) EJERCICIO2: Ejercicio2ApellidosNombre.java. Ejecutar procedimientos de base de datos

Realiza una clase Java que disponga de un método de nombre **ejercicio2(id)** que reciba un ID de alérgeno y muestre en pantalla el nombre del alérgeno, todos los platos que llevan ese alérgeno y los menús que incluyen platos con ese alérgeno. Estos datos se deben obtener de procedimientos y/o funciones que debes crear en el programa Java. Si el ID no existe se debe mostrar un mensaje indicándolo, esa comprobación también se hará en los procedimientos y funciones. Se proporciona la clase Java que tendrás que rellenar.

Por ejemplo, para las siguientes llamadas, se debe mostrar la información que hay debajo que la debe devolver la función/procedimiento almacenado:

ejercicio2(22);

NO EXISTE EL ALERGENO CON ID: 22

ejercicio2(1);

ALÉRGENO: 1, Leche y productos lácteos

PLATOS:

Plato: 1: Espaguetis

Plato: 5: Raviolis

Plato: 10: Calamares

Plato: 14: Filete de Buey

Plato: 20: Pan con Tomate

Plato: 24: Madalenas

Plato: 25: Flan con helado

Plato: 28: Helado

Plato: 29: Tarta de queso

Plato: 30: Tarta San Marcos

MENUS:

Menu: 37: La Alcarria(Plato: 1: Espaguetis)

Menu: 38: La Primavera(Plato: 14: Filete de Buey)

ejercicio2(2);

ALÉRGENO: 2, Huevos

PLATOS:

Plato: 2: Macarrones

Plato: 5: Raviolis

Plato: 10: Calamares

Plato: 15: Muslos de Pollo

Plato: 21: Tostadas

Plato: 25: Flan con helado

MENUS:

Menu: 38: La Primavera(Plato: 2: Macarrones)

ejercicio2(3);

ALÉRGENO: 3, Frutos secos

PLATOS:

Plato: 3: Tortellinis

Plato: 6: Lasaña

Plato: 11: Boquerones

Plato: 15: Muslos de Pollo

Plato: 21: Tostadas

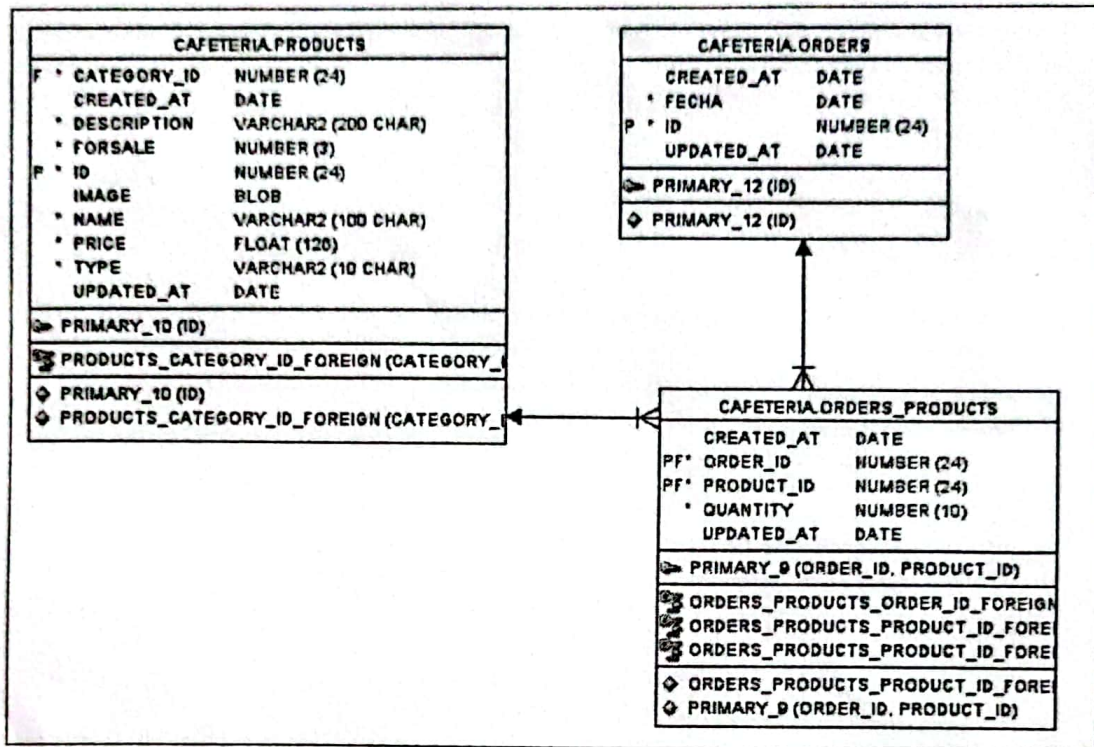
MENUS:

Menu: 39: Especial de la semana(Plato: 3: Tortellinis)

Para saltar línea puedes usar los caracteres `chr(13) || chr(10)`. Puedes crear una o varias funciones o procedimientos. Nombra a las funciones **FUN1_TUNOMBRE**, **FUN2_TUNOMBRE**, **PROC1_TUNOMBRE**, **PROC2_TUNOMBRE**, etc.

(F) EJERCICIO3: Ejercicio3ApellidosNombre.java. Modificar contenido de la BD.

Disponemos también de las tablas **ORDERS** en donde se registran los pedidos de productos, la clave es **order_id** y se genera automáticamente cada vez que se inserta un registro. Se dispone también de la tabla **ORDERS_PRODUCTS** donde se registran los detalles del pedido. Los campos son: el id del producto (**producto_id**) y la cantidad (**quantity**). La clave primaria estará formada por el **order_id + product_id**. Se muestra a continuación la relación de estas tablas con la tabla **PRODUCTS**.



Se pide realizar una clase java para insertar pedidos de productos. Diseña dentro de esa clase un método de nombre **ejercicio3(array1, array2)** que recibe dos arrays de enteros del mismo tamaño, uno contiene los id de productos a insertar y el otro la cantidad solicitada del producto. La fecha del pedido es la fecha actual. El proceso tiene que insertar pedidos. Se debe insertar en las tablas **ORDERS** y **ORDERS_PRODUCTS**. Ejemplos de llamadas para insertar 3 pedidos con varias líneas de pedido son:

```

int[] idproducto1 = { 1, 5, 7 };
int[] cantidad1 = { 1, 2, 1 };
ejercicio3(idproducto1, cantidad1);

int[] idproducto2 = { 3, 4 };
int[] cantidad2 = { 2, 3 };
ejercicio3(idproducto2, cantidad2);

int[] idproducto3 = { 8, 100, 9, 7, 11 };
int[] cantidad3 = { 1, 1, 2, 2, 3 };
ejercicio3(idproducto3, cantidad3);

```

La salida del proceso de la inserción será la siguiente, ojo con el producto 100 que no existe, se puede controlar el mensaje de error mediante excepciones. Ejemplo de salida al insertar los pedidos anteriores:

```

Pedido Insertado: 1
  Línea de pedido añadida: (1, 1, 1 )
  Línea de pedido añadida: (1, 5, 2 )
  Línea de pedido añadida: (1, 7, 1 )
Pedido Insertado: 2
  Línea de pedido añadida: (2, 3, 2 )
  Línea de pedido añadida: (2, 4, 3 )
Pedido Insertado: 3

```


Línea de pedido añadida: (3, 8, 1)
Problemas al insertar línea de pedido con el producto 100, No se insertará.
Línea de pedido añadida: (3, 9, 2)
Línea de pedido añadida: (3, 7, 2)
Línea de pedido añadida: (3, 11, 3)

Cada ejercicio se valorará con 10 puntos siempre y cuando realice lo que se pide y muestre la información solicitada.

Se proporcionan las clases Java del ejercicio 2 y 3 para que las completéis.

FORMATO DE ENTREGA

CREA UN FICHERO COMPRIMIDO DE NOMBRE: RA2RECU_ApellidosNombre.zip que contenga:

- El **proyecto Java** con nombre RA2RECU_ApellidosNombre, dentro estarán los tres ejercicios.
- Crear un fichero .jar para lanzar cada ejercicio: Ejercicio1.jar, Ejercicio2.jar y Ejercicio3.jar.

Los ejercicios se ejecutarán desde la línea de comandos del DOS. Versión del entorno Java donde se ejecutará el jar: 1.11.

Cualquier excepción no controlada que hace que se detenga la ejecución del programa resta 5 puntos del ejercicio.