Informe creado por:

Javier Villarreal Rodríguez  06030112-H

Rafael Gómez Bermejo  51113110-A

Este informe contiene pruebas de testing de la práctica 2 de Bases de datos realizada por nosotros.

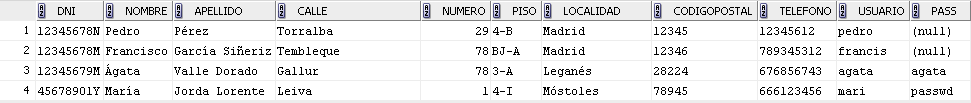
**Apartado 1:**

Procedimiento llamado***PEDIDOS\_CLIENTE*** que recibe como parámetro el DNI de un cliente y lo muestra por pantalla junto con sus datos personales y con un listado con los datos de los pedidos que ha realizado ordenados crecientemente por fecha. En caso de error deberá mostrarse por pantalla un mensaje de advertencia explicando el error. Al finalizar el listado se deberá mostrar la suma de los importes de todos los pedidos del cliente. Incluye un bloque de código anónimo para probar el procedimiento para cada uno de los casos posibles .

**Casos de prueba:**

*Que el cliente no exista:*

Este es el estado de la tabla Clientes antes de ejecutar el procedimiento:



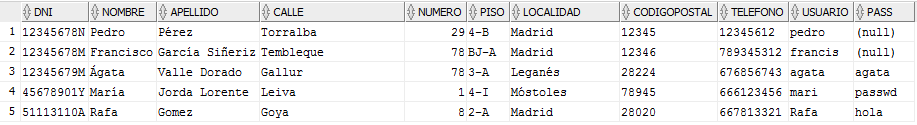
Vamos a probar con el usuario con DNI 12345678A que como vemos en la imagen anterior no existe.

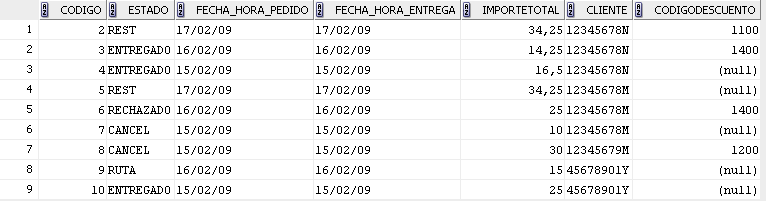
|  |
| --- |
| **Bloque anónimo**  DECLARE  DNICLIENTE CHAR(200);  BEGIN  DNICLIENTE := '12345678A';  PEDIDOS\_CLIENTE(  DNICLIENTE => DNICLIENTE  );  --rollback;  END; |

En dicho bloque anónimo hemos usado un parámetro en el procedimiento que indica un DNI de un cliente que no existe. En la salida por pantalla sale el siguiente mensaje de error “El cliente no existe” indicando que el cliente no existe.

*Que el cliente exista y no tenga pedidos:*

Este es el estado de la tabla de Clientes y Pedidos respectivamente antes de ejecutar el procedimiento para el caso de prueba::





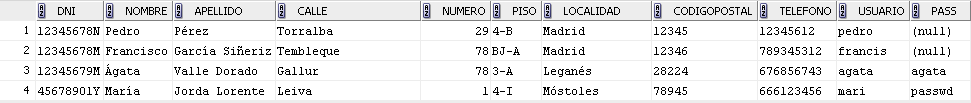
Como vemos, hemos creado un usuario con DNI 51113110A, el cual no tiene pedidos.

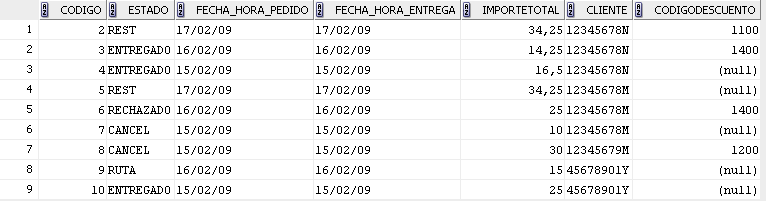
|  |
| --- |
| **Bloque anónimo**  DECLARE  DNICLIENTE CHAR(200);  BEGIN  DNICLIENTE := '51113110A';  PEDIDOS\_CLIENTE(  DNICLIENTE => DNICLIENTE  );  --rollback;  END; |

En dicho bloque anónimo hemos usado un parámetro en el procedimiento que indica un cliente no tiene pedidos asociados. En la salida por pantalla sale el siguiente mensaje de error “El cliente no tiene pedidos asociados” indicando que el cliente no ha realizado ninguno.

*Que el cliente exista y tenga pedidos:*

Este es el estado de la tabla Clientes y Pedidos, respectivamente antes de ejecutar el procedimiento para el caso de prueba:

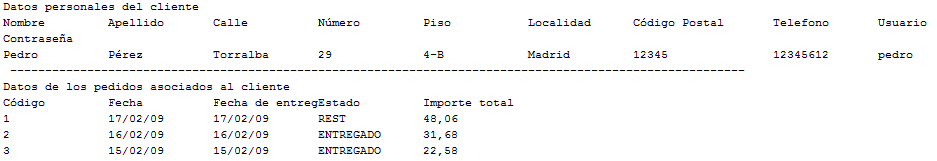




Como vemos, el usuario con DNI 12345678N existe y tiene pedidos asociados.

|  |
| --- |
| **Bloque anónimo**  DECLARE  DNICLIENTE CHAR(200);  BEGIN  DNICLIENTE := '12345678N';  PEDIDOS\_CLIENTE(  DNICLIENTE => DNICLIENTE  );  --rollback;  END; |

En dicho bloque anónimo hemos usado un parámetro en el procedimiento que indica un DNI de un cliente que existe y tiene pedidos asociados. En la salida por pantalla sale lo siguiente:



**Apartado 2**

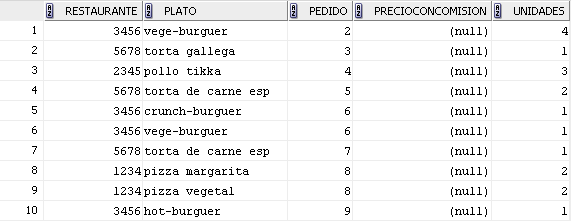
Procedimiento almacenado llamado***REVISA\_PRECIO\_CON\_COMISION***(sin argumentos) cuya misión es comprobar la consistencia de los datos de todos los precios con comisión en la tabla Contiene. El campo “precio con comisión” de la tabla “Contiene” debe almacenar el precio del plato incluyendo el porcentaje de la comisión de su restaurante. El procedimiento debe verificar y actualizar estos datos de modo que resulten consistentes. Si todos los datos son correctos, se mostrará un mensaje indicando “Ningún cambio en los datos de Contiene”. En caso contrario se indicará el número de filas modificadas en Contiene.

**Casos de prueba**

|  |
| --- |
| **Bloque anónimo**  BEGIN  REVISA\_PRECIO\_CON\_COMISION();  --rollback;  END; |

*Hay filas que no son consistentes:*

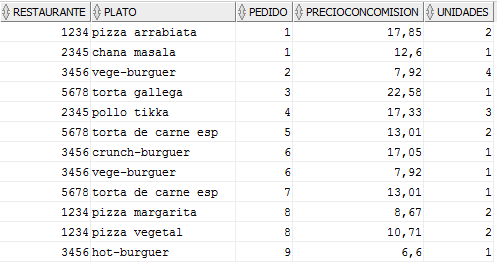
Este es el estado de la tabla Contiene antes de ejecutar el procedimiento para el caso de prueba:



Al principio, la tabla Contiene tiene todas las filas con el campo PRECIOCONCOMISION nulo.Cuando ejecutas dicho bloque anónimo sale el siguiente mensaje de error “Se han modificado 12 filas de la tabla contiene.” indicando el número de filas que se han modificado.

*Todas las filas son consistentes:*

Este es el estado de la tabla Contiene antes de ejecutar el procedimiento para el caso de prueba:



Una vez ejecutado, la primera vez dicho procedimiento se actualizan todas las filas de la tabla Contiene. Con lo cual, al ejecutarlo por segunda vez, sale el siguiente mensaje: “Ningun cambio den los datos de contiene”.También si cambias alguna comisión de un restaurante y ejecutas el procedimiento se ejecuta correctamente y te indica que se han modificado las filas de la tabla Contiene.

**Apartado 3**

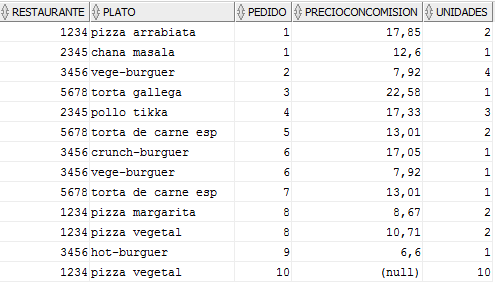
Procedimiento almacenado llamado ***REVISA\_PEDIDOS*** (sin argumentos) cuya misión es comprobar la consistencia de los datos de todos los pedidos. El campo “importe total” de la tabla “Pedidos” debe almacenar la suma de los “precio con comisión” de los platos del pedido multiplicados por su cuantía. Se pide usar un cursor FOR UPDATE El procedimiento debe verificar y actualizar estos datos para todos los pedidos, de modo que resulten consistentes. Si todos los datos son correctos, se mostrará un mensaje indicando “Ningún cambio en los datos de la tabla Pedidos”. En caso contrario se indicará el número de filas modificadas en en la tabla Pedidos.

|  |
| --- |
| BEGIN  REVISA\_PEDIDOS();  --rollback;  END; |

**Casos de prueba**

*Hay filas de la tabla Contiene que tienen el campo PRECIOCONCOMISION nulo*.

Este es el estado de la tabla Contiene antes de ejecutar el procedimiento para el caso de prueba:

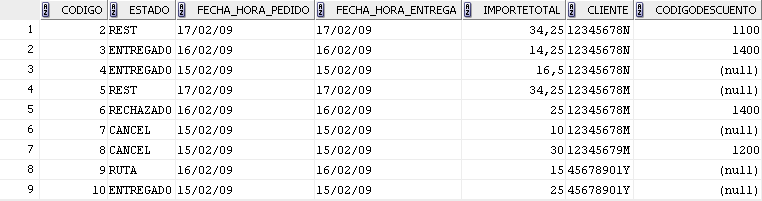


Como vemos, hay una fila de la tabla de Contiene que tiene el campo PRECIOCONCOMISION nulo.

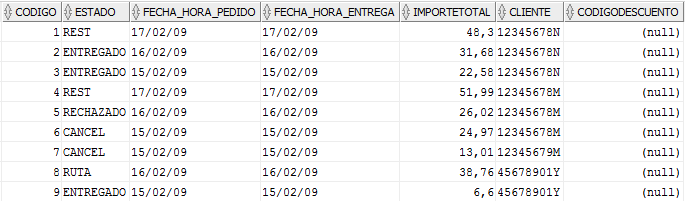
Al ejecutar el procedimiento, éste comprueba si hay alguna fila con el campo nombrado a nulo y en este caso muestra el mensaje “Hay filas de la tabla Contiene que tienen el campo PRECIOCONCOMISION NULO” y finaliza el proceso.

*Hay filas que no son consistentes.*

Este es el estado de la tabla Pedidos antes de ejecutar el procedimiento para el caso de prueba:



En caso de haber inconsistencias en las filas se mostraría el mensaje “Se han modificado ‘n’ filas de la tabla pedidos.” , siendo n el número de filas inconsistentes. Dichas modificaciones arreglaría las inconsistencias actualizando el valor del pedido. Al ejecutar dicho procedimiento, el estado de la tabla Pedidos queda así:



*Todas las filas son consistentes*

Si todas las filas son consistentes, como es el caso, de haber ejecutado el procedimiento para el caso de prueba anterior, se mostraría el mensaje “Ningún cambio den los datos de pedidos” dando a entender que todos los valores son correctos.

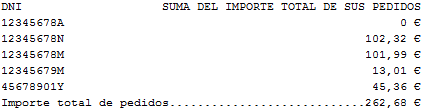
**Apartado 4**

Procedimiento ***DATOS\_CLIENTES*** que recorre todos los Clientes con un FOR y muestra todos sus datos junto con la suma de importe total de todos sus pedidos. Finalmente, muestra la suma total de los importes de todos los pedidos de todos los clientes.

**Casos de prueba**

|  |
| --- |
| **Bloque anónimo**  BEGIN  DATOS\_CLIENTES();  --rollback;  END; |

En dicho bloque anónimo hemos llamado al procedimiento. En la salida por pantalla sale lo siguiente:



El usuario con DNI 12345678A ha sido creado a posta para testear que se muestran los clientes que no tienen pedidos.

**Apartado 5**

Procedimiento que llama a REVISA\_PRECIO\_CON COMISION, REVISA\_PEDIDOS y DATOS CLIENTES. Incluye un bloque anónimo de prueba. El procedimiento se llama LLAMADA.

|  |
| --- |
| BEGIN  LLAMADA();  --rollback;  END; |