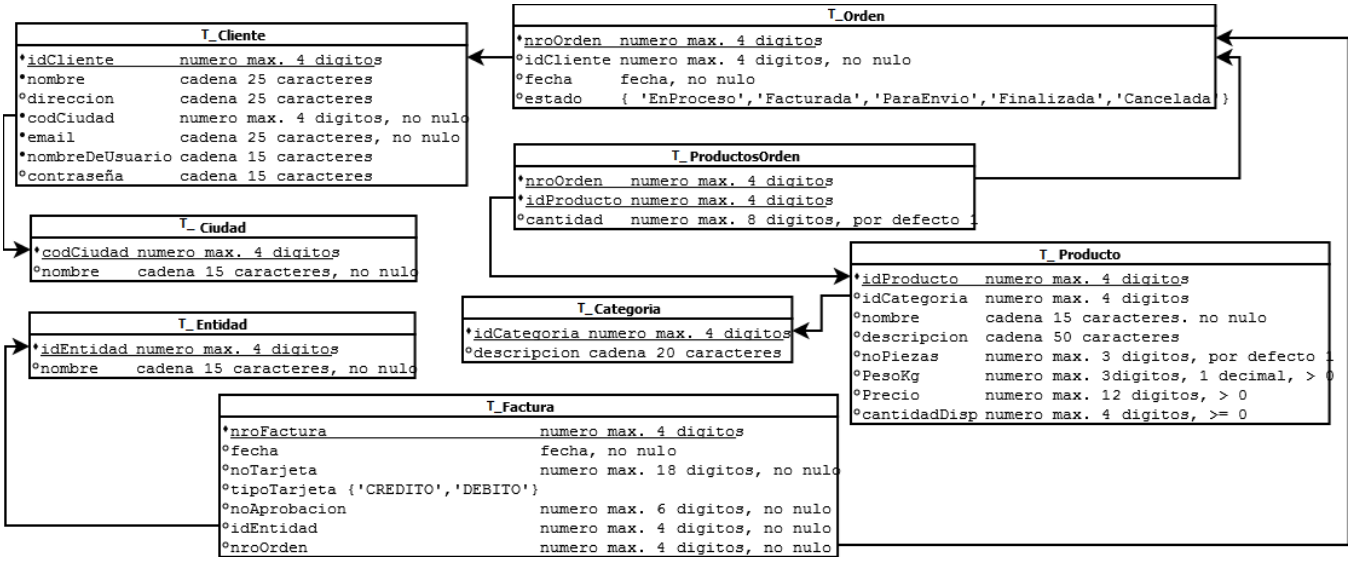
En este laboratorio haremos un repaso de la programación en PL/SQL. Para ello, usaremos los datos una base de datos de una tienda virtual.

La tienda tiene clientes, que se registran en la plataforma con un nombre de usuario y una contraseña. Una vez registrados pueden hacer ordenes (carritos de compra) que incluyen los productos que el cliente quiere comprar. En el proceso de la compra, la orden cambia de estado. Inicia con estado "EnProceso" que indica que el cliente está decidiendo que productos va a comprar. Una vez el cliente decide hacer la compra y realiza el pago, la orden para a estado "Facturada". Durante el proceso de envío (desde la recopliación de los productos hasta la entrega) el estado es "ParaEnvio", y una vez ha sido entregada, el estado es "Finalizada". Por otro lado, si el cliente decide no comprar, la orden pasa de "EnProceso" a "Cancelada".

Modelo de Datos

Los datos están cargados en **su** esquema; las tablas de este laboratorio inician con "T_", como se muestra en la siguiente Figura que presenta el diagrama del modelo relacional.



Ejercicios

Usando el esquema dado, se solicitan los siguientes requerimientos.

1. Cree los índices necesarios para mejorar el tiempo de respuesta de las siguientes consultas:

```
SELECT * FROM T_Cliente WHERE nombreDeUsuario = 'Calvin';
```

```
SELECT * FROM T_Cliente NATURAL JOIN T_Orden
WHERE fecha BETWEEN '01/06/2020' AND '31/10/2020';
```

Para mirar el plan de ejecución, agregue la sentencia EXPLAIN PLAN FOR antes del SELECT:

```
EXPLAIN PLAN FOR SELECT * FROM T_Cliente WHERE nombreDeUsuario = 'Calvin';
```

Una vez ejecutado el comando, use la siguiente sentencia para desplegar el plan de ejecución:

```
SELECT PLAN_TABLE_OUTPUT FROM TABLE(DBMS_XPLAN.DISPLAY());
```

2. Escriba un trigger que evite agregar a una orden un producto en una cantidad mayor a la que está disponible. En el mismo sentido, si se aumenta la cantidad de un producto en una orden, evite que la diferencia supere la cantidad disponible del producto. Incluya en el script los casos de prueba (casos en que se aborta la operación, casos exitosos)

3. Se va a otorgar un bono de descuento a los n clientes con mayores compras (valor comprado) de cada ciudad. Escriba una función PL que reciba 3 parámetros: El primero, n, indica cuantos clientes elije de cada ciudad; el segundo, maxp, es el máximo porcentaje de descuento, que se da al cliente con mayor valor comprado; y el tercero minp, es el porcentaje mínimo de descuento que se va a otorgar. La función debe retornar una tabla con la identificación y nombre de cada cliente seleccionado, la dirección, el nombre de la ciudad donde vive, y el valor del bono de descuento. En cada ciudad, el cliente con mayor valor comprado recibe el maxp descuento, el segundo cliente recibe maxp – 0.1, el siguiente maxp – 0.2, luego maxp – 0.3 y así sucesivamente hasta llegar a minp, a partir de ese punto todos los clientes reciben el porcentaje minp, hasta completar n clientes.

4. Escriba un procedimiento que use el cursor for update para actualizar el precio de los productos que al listar los productos en orden ascendente de su nombre ocupan las posiciones 3, 5, 8 y 12. El nuevo valor de esos productos llega como parámetro.

5. Cuando se borra una factura se debe borrar la orden asociada a ella. Escriba un procedimiento que reciba como parámetro el número de la factura, y se encargue de borrar la factura y su orden asociada. Tenga en cuenta incluir el manejo transaccional y de excepciones.

Al finalizar la sesión, cada estudiante debe enviar el script a mcpabon@javerianacali.edu.co.