UNAH

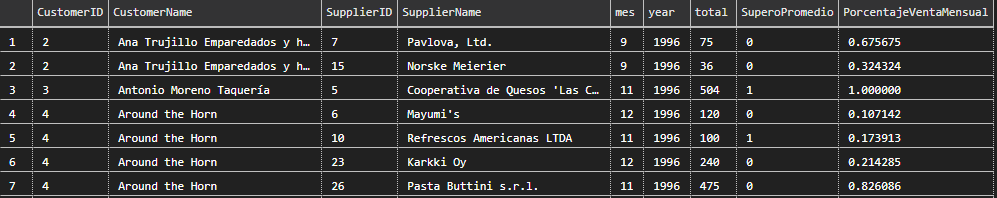
Examen práctico Bases de datos I

Victor Javier Amaya

#20161000203

**Ejercicio Uno (60%)**

Según la BD de productos realizar una consulta que retorne el siguiente resultado



*Definiciones a tener en cuenta:*

Total:

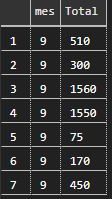
es la suma de todo lo que se Vendió agrupado por customer, supplier, mes y año.

SuperoPromedio:

esta transformación va en función del supplier, ejemplo, si analizamos solamente el customerID 2 para el supplier 7 notaremos lo siguiente en el mes de septiembre solamente facturo 75



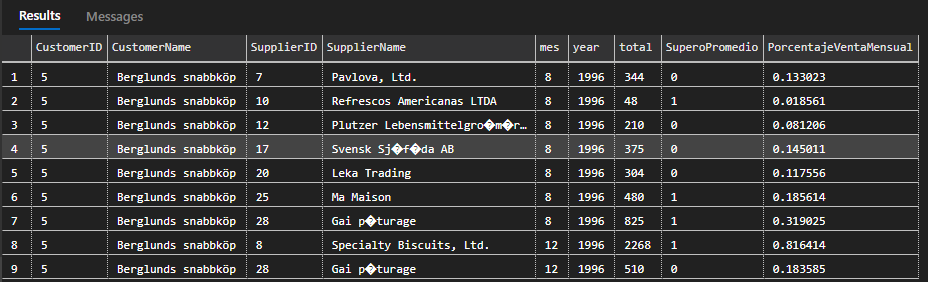
Pero si analizamos todo lo que facturo el supplier 7 en el mes de septiembre obtendremos:



Si sacamos el promedio (AVG) de eso : 659.285714 podemos notar q los 75 que facturo el customer 2 esta por debajo del promedio por lo que se coloca un 0 en caso de ser contrario colocar un 1.

PorcentajeVentaMensual:

Para este ejemplo tomemos como ejemplo al Customer 5

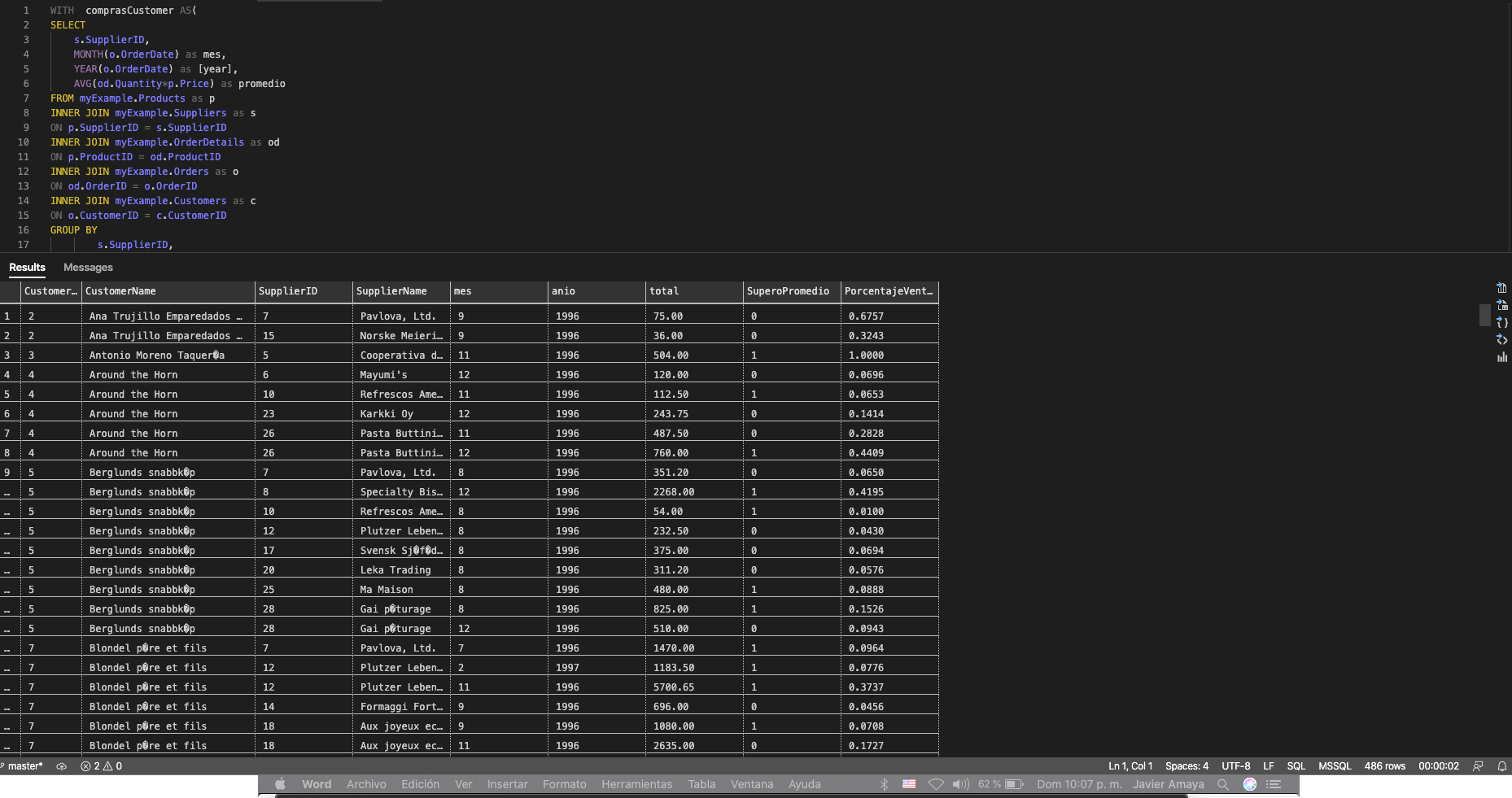


Como se puede observar este cliente realizo compras en dos meses octubre y diciembre si analizamos la línea del gasto del supplier 7 q fue en octubre se facturo 344 si sumamos todo lo que ese cliente facturo en ese mes tendremos 2586. Ahora si dividimos los 344/2586 obtendremos que ese gasto representa el 13 por ciento.

*Consideraciones a tener en cuenta:*

* El resultado principal que se busca el la primera imagen
* Se deben de usar las tablas de productos que ya se habían trabajado previamente
* Una vez terminado el query se debe de guardar en un .sql
* No se permite preguntar a otros compañeros como lo están haciendo o que le den ideas.

*RESPUESTA :*

**

**

**El archivo se llama QUERY\_REQUEST\_EXERCISE\_1.sql**

**Ejercicio dos (20%)**

Se esta pensando en automatizar el proceso de lectura de la consulta trabajada en el query anterior. Por lo que se busca lo siguiente:

* Según la estructura de la respuesta del query cree una nueva tabla con esas columnas en schema.fact\_summary.
* Esta tabla almacenara la información de un solo año, usted notara que tenemos información de dos años. Por lo que se pide que se cree una tabla de schema.settings y almacene en una tabla el año que se estará operando, esta tabla puede lucir de la siguiente manera:

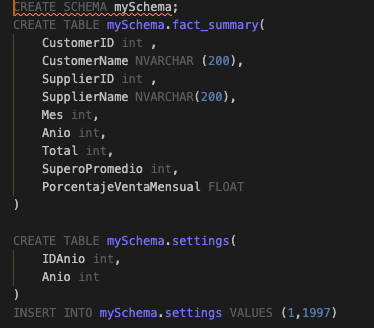
|  |  |
| --- | --- |
| Key | Value |
| Anio | 1997 |

* Crear un procedimiento almacenado que llene la información de esa tabla utilizando el query de antes (note que sí hace subconsultas para obtener el resultado esperado, se considera mala práctica dejar el filtro del año hasta el final)
* Tome en cuenta que usted puede popular una tabla haciendo un INSERT INTO TABLA SELECT \* FROM TABLA
* Valide cosas como: que sucede si corro varias veces el mismo SP.

Que se debe de entregar:

* El DDL de la nueva tabla que está creando (esta no tendrá PK y FK)
* La lógica del SP
* Si ocupo algo mas guarde la evidencia

**RESPUESTA**



El archivo de la logica del SP se llama SP\_FACT\_SUMMARY

Según la hermeneutica de este contexto de negocio, para validar el caso de ejecutar varias veces el mismo SP , lo que hago es una limpieza de la tabla fact\_summary cada vez que se ejecuta el SP.

En otras palabras primero limpia y despues actualiza la tabla.

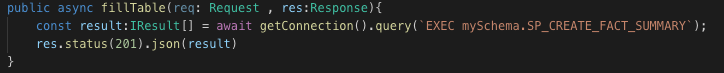
Debido al ejemplo que se da en este archivo la tabla settings solamente tiene el año 1997. Si se quisiera validar otros años tendria que añadir mas contexto de negocio a la situacion para tomar las validaciones pertinentes.

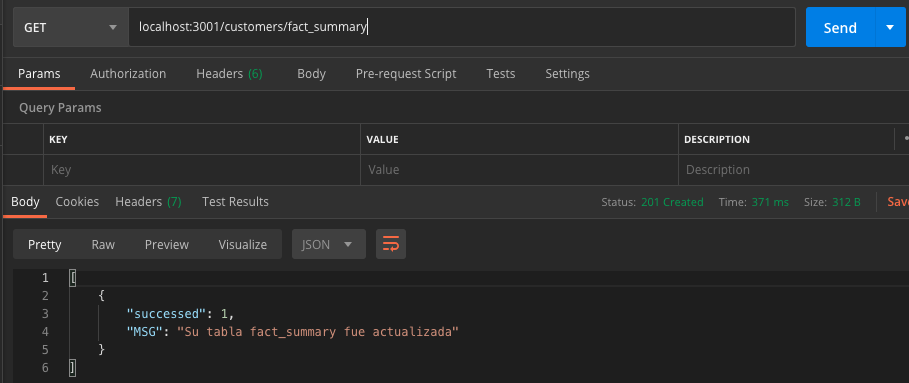
**Ejercicio tres (20%)**

* En el proyecto de restfulapi desarrolle un endpoint GET q llama el SP ya creado en el ejercicio anterior para actualizar la tabla de fact\_summary.

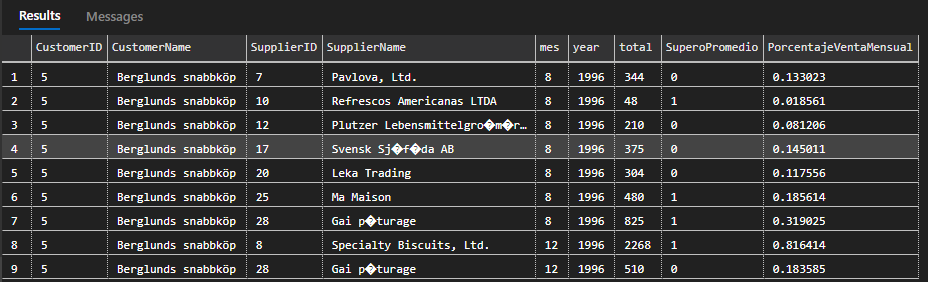
**RESPUESTA**

****

****

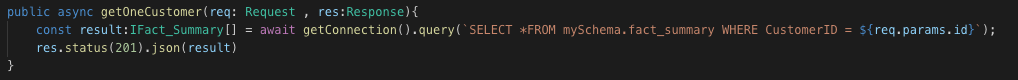
****

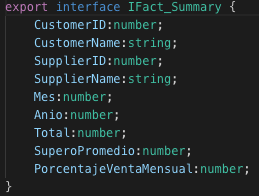
* En el proyecto restfulapi desarrolle un endpoint donde usted mande la información del customer id y le regrese en un arreglo; objetos que modele información como la que se muestra abajo

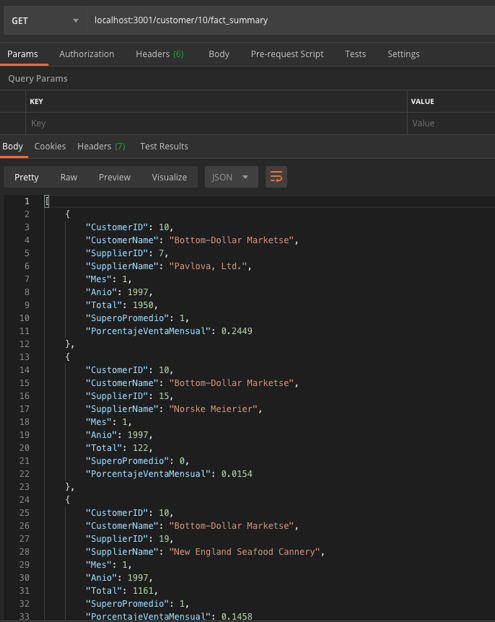


**RESPUESTA**

**

**

**

**

*Los dos primeros ejercicios los archivos los pueden anexar a una nueva carpeta del proyecto que hicieron fork (*<https://github.com/uayeb25/restfulapi-products-typeorm>*) sea ordenado de crear carpetas o nombre descriptivos para entender lo que revisare.*

**Link Git**

La carpeta de los files que pidio se llama **examenPractico**

<https://github.com/JavierAmaya/restfulapi-typorm-products>

El servicio se llama **customer.service**

La interface **fact\_summary**

El controller **customer.controller**

El archivo del query del ejercicio 1 se llama **QUERY\_REQUEST\_EXERCISE\_1**

El DDL para crear el schema y las tablas se llama **CREATE\_SCHEMA\_AND\_TABLES**

El archivo para el procedimiento almacenado se llama **SP\_FACT\_SUMMARY**