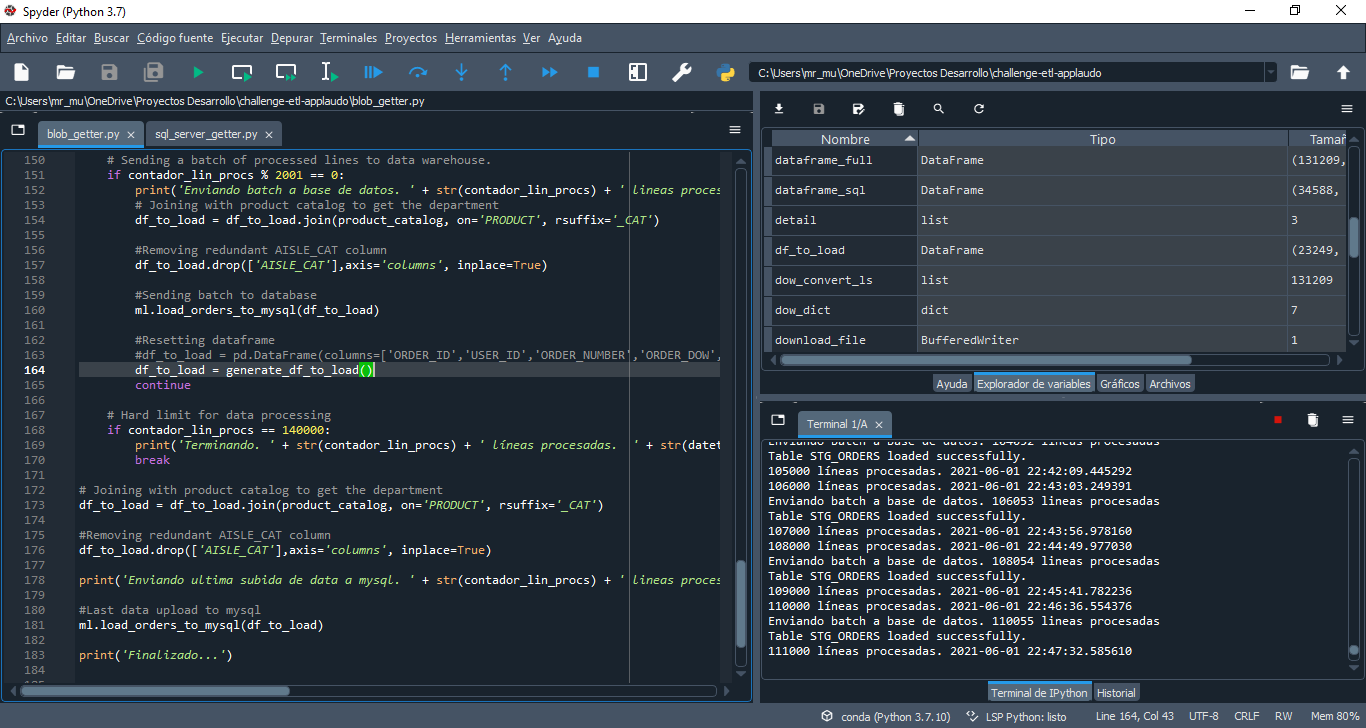
Coding Challenge: ETL Test

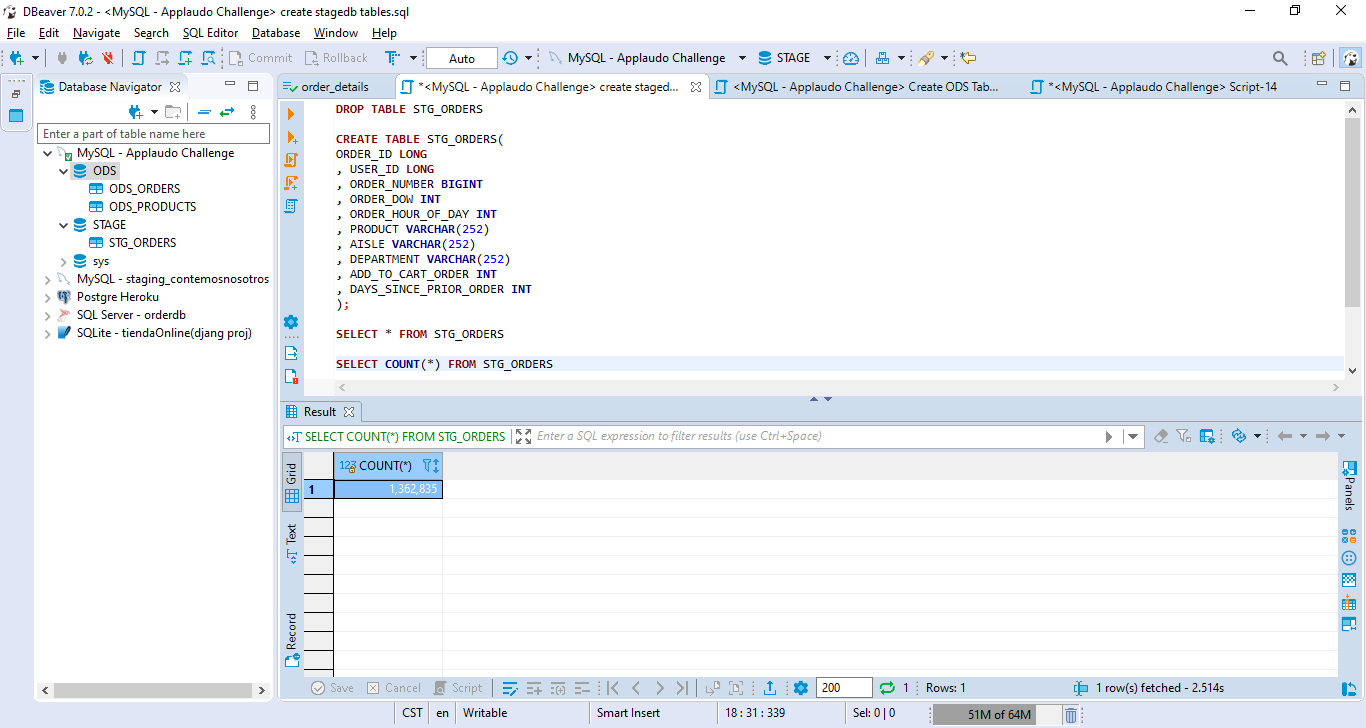
# Obtención de información de los repositorios Azure, SQL Server y API de productos

La solución consta de un conjunto de scripts de Python que realizan la tarea de obtener la información desde el blob store de Azure y la base de datos SQL Server. Se construye un dataframe para unirlo con el catálogo de productos procesado desde formato JSON y luego se carga hacia nuestra base de datos MySQL.

El script presenta mensajes informativos del estado del proceso y marcas de tiempo al ejecutarse desde el IDE Spyder:



Con la información del script se puede identificar el estado de progreso. En total se tienen más de 130,000 líneas de órdenes, las cuales se deben normalizar para generar una línea por cada articulo en cada orden específica, generando más de 1,300,000 registros cargados a la tabla STG\_ORDERS de nuestra base de datos STAGE.



En el proceso de carga se encontró que la data era relativamente homogénea o que era fácil de convertir a un tipo de datos específico. A excepción de la columna ORDER\_DOW, el cual contenía números en la mayoría de los casos, pero que contenía el nombre del día de la semana en otros. Se hizo una conversión para que todos los campos quedaran expresados como número.

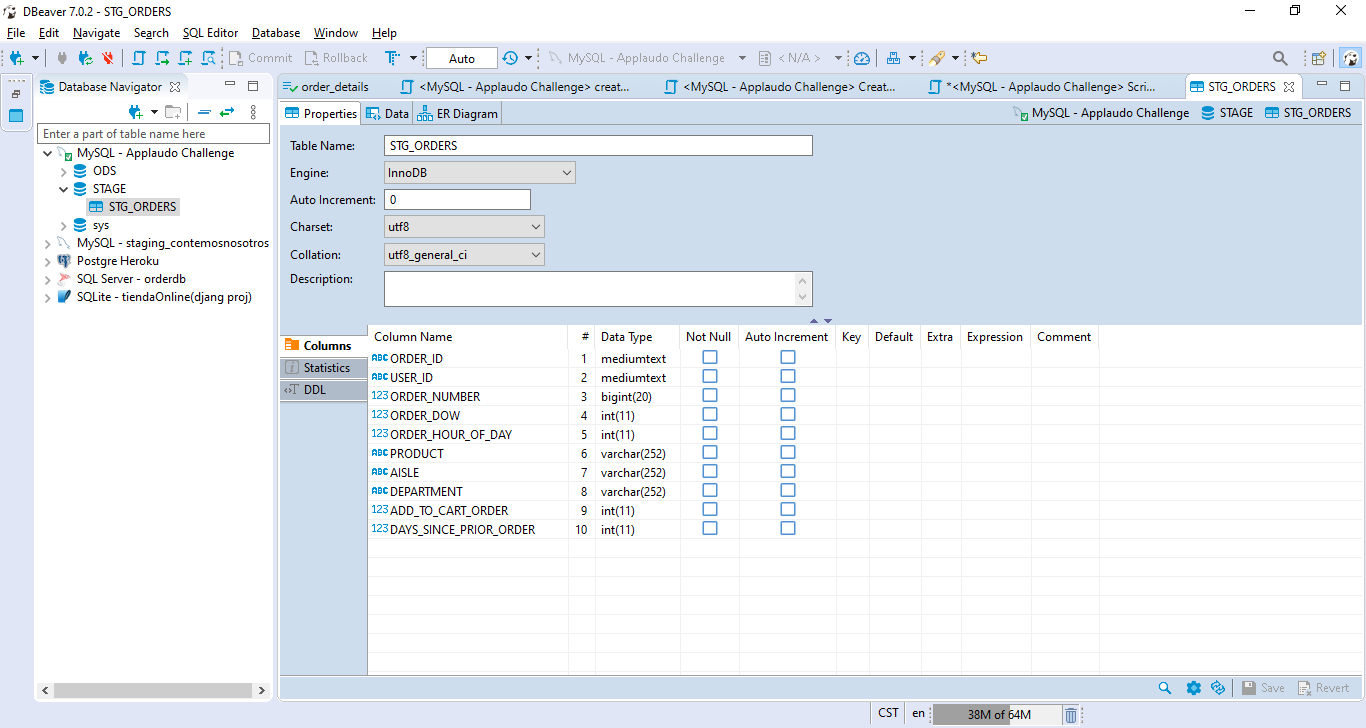
Los scripts Python y SQL utilizados se adjuntan con esta entrega.

# Base de datos creada para la solución

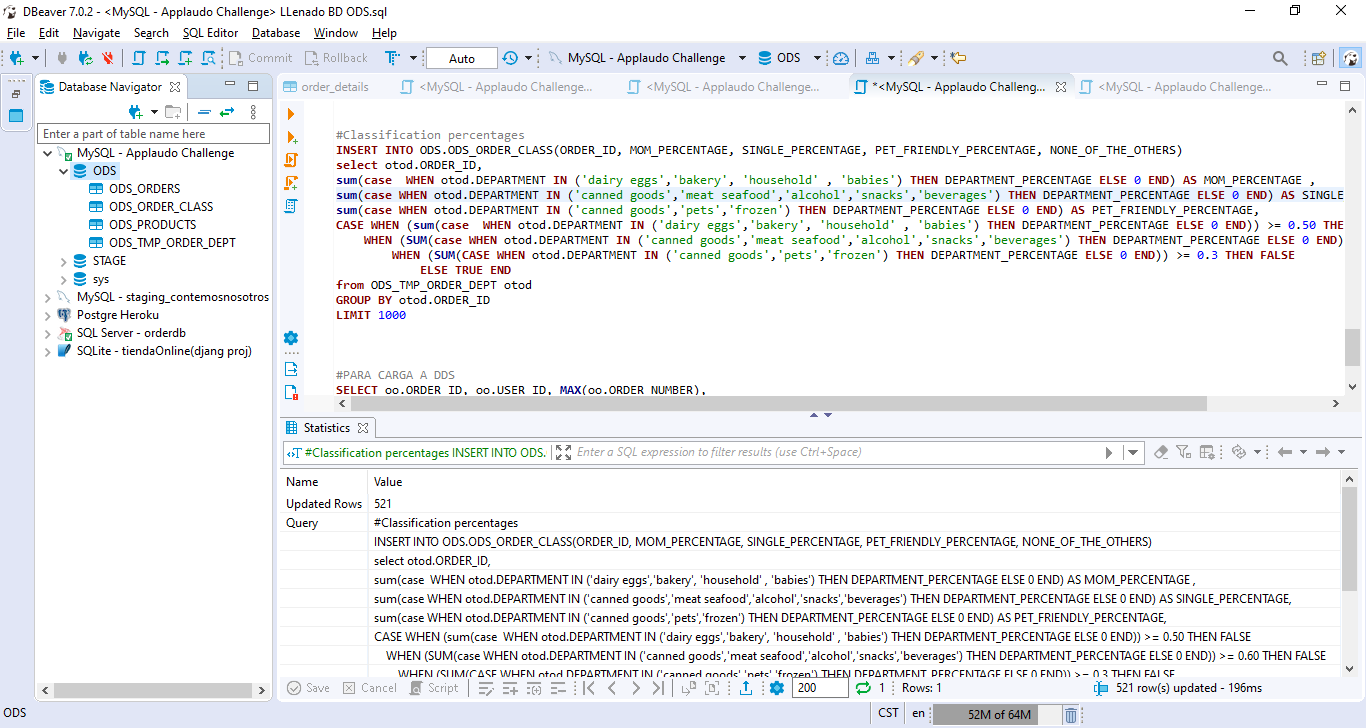
Se crearon bases de datos para la solución analítica en un servidor mysql de Google Cloud.

## STAGE

Se ha generado una base de datos STAGE para la carga de la información proveniente de la unión del repositorio azure y sql server. La cual contiene una única tabla para almacenar la información:



## ODS



La base de datos operacionales (ODS) contiene una tabla de catálogo de productos, la cual se ha llenado con la información enviada con los pedidos. Dicha tabla incluye un campo flag para indicar si el producto tiene registro correspondiente en el catálogo o si no.

La creación de las tablas actuales de ODS es la siguiente:

**DROP** **TABLE** **IF** **EXISTS** ODS\_PRODUCTS;

**CREATE** **TABLE** ODS\_PRODUCTS(

ID **BIGINT** **NOT** **NULL** **AUTO\_INCREMENT**

, PRODUCT **VARCHAR**(252)

, AISLE **VARCHAR**(252)

, DEPARTMENT **VARCHAR**(252)

, FROM\_CATALOG **BOOLEAN**

, **PRIMARY** **KEY**(ID)

);

**DROP** **TABLE** **IF** **EXISTS** ODS\_ORDERS;

**CREATE** **TABLE** ODS\_ORDERS(

ID **INT** **NOT** **NULL** **AUTO\_INCREMENT**

, ORDER\_ID **BIGINT**

, USER\_ID **BIGINT**

, ORDER\_NUMBER **INT**

, ORDER\_HOUR\_OF\_DAY **SMALLINT**

, ORDER\_DOW **SMALLINT**

, DAYS\_SINCE\_PRIOR\_ORDER **INT**

, ADD\_TO\_CART\_ORDER **INT**

, PRODUCT **BIGINT**

, **PRIMARY** **KEY** (ID)

, **INDEX** product\_id (PRODUCT)

, **FOREIGN** **KEY** (PRODUCT) **REFERENCES** ODS\_PRODUCTS(ID) **ON** **DELETE** **RESTRICT**

)

Los siguientes scripts llenan el catálogo de productos indicando si cada registro se encuentra efectivamente en el catálogo o si no es posible identificar el producto dentro del catálogo (productos nuevos)

#INSERCION CATALOGO DE PRODUCTOS A ODS

**INSERT** **INTO** ODS.ODS\_PRODUCTS (PRODUCT,AISLE,DEPARTMENT,FROM\_CATALOG)

(**SELECT** **DISTINCT** PRODUCT, AISLE, DEPARTMENT, **TRUE**

**FROM** STG\_ORDERS

**WHERE** ISNULL(DEPARTMENT)=0

)

#INSERCION PRODUCTOS NO EN CATALOGO A ODS

**INSERT** **INTO** ODS.ODS\_PRODUCTS (PRODUCT,AISLE,DEPARTMENT,FROM\_CATALOG)

(**SELECT** **DISTINCT** PRODUCT, AISLE, 'UNKNOWN', **FALSE**

**FROM** STG\_ORDERS

**WHERE** ISNULL(DEPARTMENT)=1

)

Se creó una tabla adicional como tabla temporal para almacenar los cálculos de productos por departamento y el porcentaje de estos, para categorizar a los clientes de la forma solicitada por el departamento de Marketing.

**CREATE** **TABLE** ODS.ODS\_TMP\_ORDER\_DEPT(

ID **INT** **NOT** **NULL** **AUTO\_INCREMENT**,

ORDER\_ID **BIGINT**,

DEPARTMENT **VARCHAR**(252),

DEPARTMENT\_COUNT **INT**,

DEPARTMENT\_PERCENTAGE **DECIMAL**(6,3),

**PRIMARY** **KEY** (ID)

);

El llenado de dicha tabla se realiza con el siguiente script (NOTA: El script utiliza la cláusula LIMIT para propósito de pruebas de concepto ya que la inserción completa demora mucho tiempo.)

**INSERT** **INTO** ODS.ODS\_TMP\_ORDER\_DEPT (ORDER\_ID, DEPARTMENT, DEPARTMENT\_COUNT, DEPARTMENT\_PERCENTAGE)

**SELECT** oo.ORDER\_ID , op.DEPARTMENT, **COUNT**(op.DEPARTMENT),

**COUNT**(op.DEPARTMENT)/(**SELECT** **COUNT**(oo1.PRODUCT)

**FROM** ODS\_ORDERS oo1 **WHERE** oo1.ORDER\_ID = oo.ORDER\_ID **AND** oo1.USER\_ID = oo.USER\_ID **AND** oo1.ORDER\_NUMBER = oo.ORDER\_NUMBER )

**FROM** ODS\_ORDERS oo **INNER** **JOIN** ODS\_PRODUCTS op **ON** oo.PRODUCT = op.ID

#WHERE oo.ORDER\_ID = 1

**GROUP** **BY** oo.ORDER\_ID, oo.USER\_ID, oo.ORDER\_NUMBER ,op.DEPARTMENT

Con este script se obtiene la proporcionalidad de departamentos de productos en cada orden, con lo cual se etiquetarían los clientes de la forma solicitada por el departamento de Marketing.

#Classification percentages

**select** otod.ORDER\_ID,

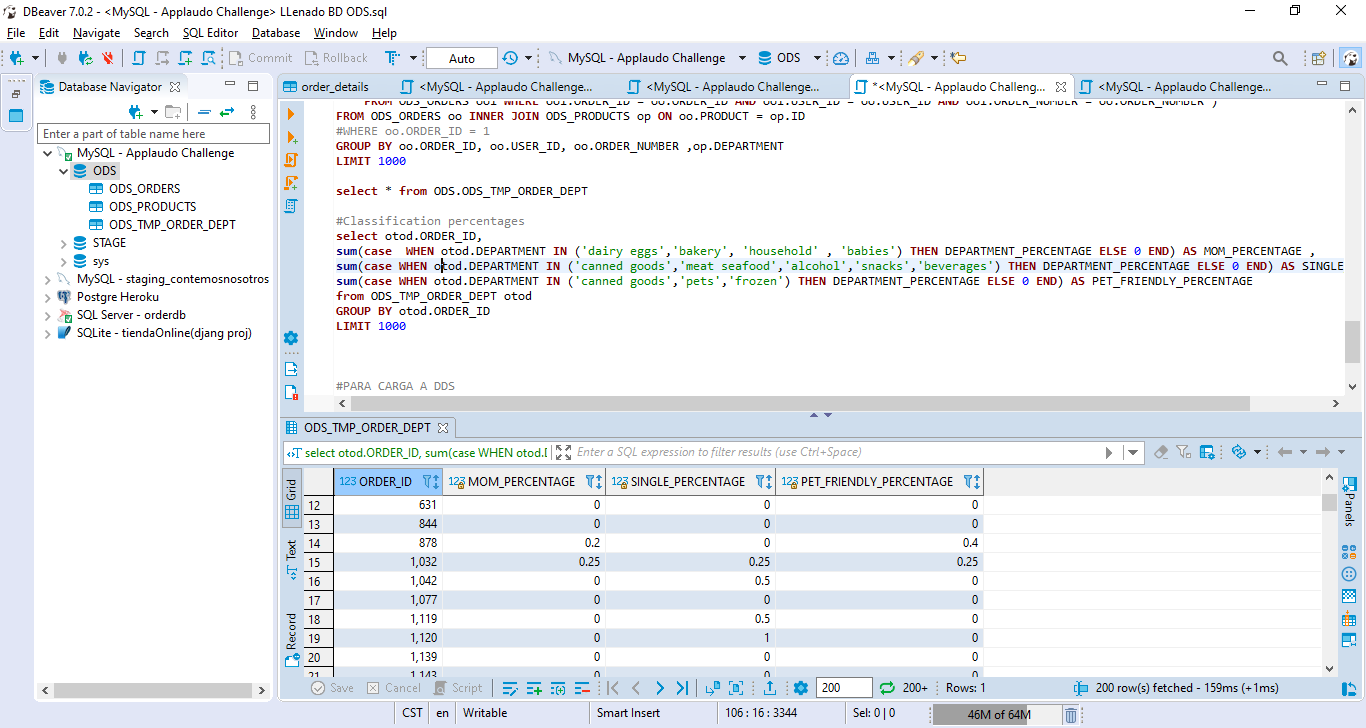
**sum**(**case** **WHEN** otod.DEPARTMENT **IN** ('dairy eggs','bakery', 'household' , 'babies') **THEN** DEPARTMENT\_PERCENTAGE **ELSE** 0 **END**) **AS** MOM\_PERCENTAGE ,

**sum**(**case** **WHEN** otod.DEPARTMENT **IN** ('canned goods','meat seafood','alcohol','snacks','beverages') **THEN** DEPARTMENT\_PERCENTAGE **ELSE** 0 **END**) **AS** SINGLE\_PERCENTAGE,

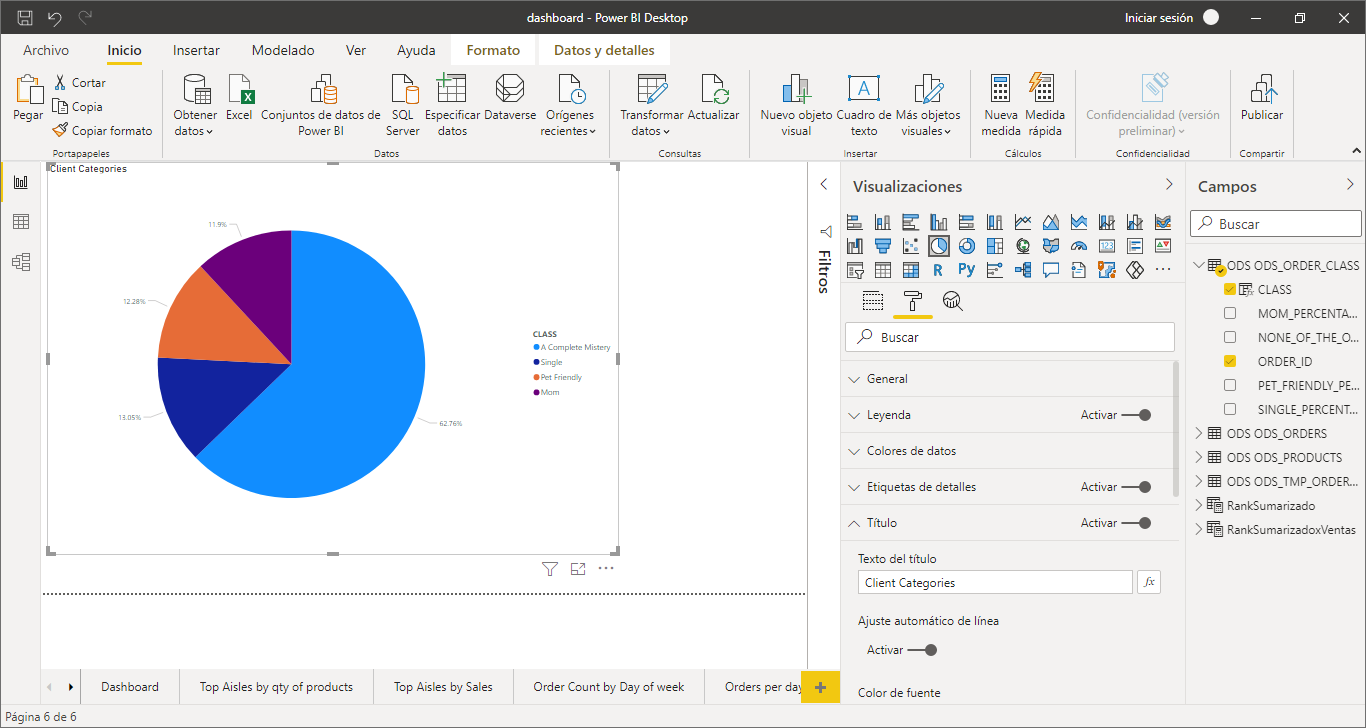
**sum**(**case** **WHEN** otod.DEPARTMENT **IN** ('canned goods','pets','frozen') **THEN** DEPARTMENT\_PERCENTAGE **ELSE** 0 **END**) **AS** PET\_FRIENDLY\_PERCENTAGE

**from** ODS\_TMP\_ORDER\_DEPT otod

**GROUP** **BY** otod.ORDER\_ID



Al cargar la información de etiquetas en el modelo y probarlo con un subconjunto de los datos se obtiene la proporción de las órdenes recibidas



# Próximos pasos

Debido a contratiempos, no se logró completar con toda la información requerida. Un siguiente paso sería clasificar y visualizar los datos solicitados por el departamento de Customer Loyalty. Así como hacer pruebas con una cantidad de datos mayores y una infraestructura que mejore los tiepmos de respuesta para carga, procesamiento y análisis de información.