Prueba Técnica - Ingeniero de Datos

Gabriela Eugenia Orellana Medina

27 de mayo de 2023

INTRODUCCIÓN

Olist, es una tienda por departamentos más grande de los mercados brasileños. Conecta pequeñas empresas de todo Brasil a los canales sin problemas y con un solo contrato. Esos comerciantes pueden vender sus productos a través de la Tienda Olist y enviarlos directamente a los clientes utilizando los socios logísticos con los que cuenta Olist.
Después de que un cliente compra el producto, se notifica a un vendedor para cumplir con ese pedido. Una vez que el cliente recibe el producto, o vence la fecha estimada de entrega, el cliente recibe una encuesta de satisfacción por correo electrónico donde puede dejar una nota sobre la experiencia de compra y anotar algunos comentarios.

Olist Store compartió un conjunto de datos. El conjunto de datos tiene información de 100k pedidos de 2016 a 2018 realizados en múltiples mercados en Brasil. Sus funciones permiten ver un pedido desde múltiples dimensiones: desde el estado del pedido, el precio, el pago y el rendimiento del flete hasta la ubicación del cliente, los atributos del producto y, finalmente, las reseñas escritas por los clientes. También un conjunto de datos de geolocalización que relaciona los códigos postales brasileños con las coordenadas latitud/longitud.

A partir de este conjunto se decide elegir 3 archivos, los cuales se tiene que elaborar el modelado de datos y su respectiva automatización del proceso en el cual se desarrolla un flujo automatizado para la extracción, transformación y carga (ETL) de los archivos hacia la base de datos relacional creada en el modelado de datos.

DESCRIPCIÓN DE LA PRUEBA

1. *Modelado de datos (35%):*
   Diseñar y crear un modelo de datos relacional adecuado para el conjunto de datos proporcionado.

2. *Automatización del proceso (35%):*
   Desarrollar un flujo automatizado para la extracción, transformación y carga (ETL) de los archivos provistos hacia la base de datos relacional creada en el paso anterior.

3. *Documentación técnica y publicación del proyecto (30%):*
   Generar un documento técnico describiendo los aspectos relevantes de la solución con base a tu experiencia y publicar fuentes en GITHUB. El enlace al repositorio debe incluirse en el documento técnico.

MODELADO DE DATOS

Se decidió tomar como base los archivos:

- olist_orders_dataset
- olist_order_items_dataset
- olist_customers_dataset

Se consideraron importantes porque esta información servirá para planificar y gestionar los productos y es una prueba y garantía de que las ordenes se han realizado. Esta información, además, servirá para planificar las intervenciones y hacer un seguimiento y monitoreo de ellas. En cuanto a la base de customers, ayudará a saber a que estado y municipio de Brasil se enviarán los productos de las ordenes realizadas.

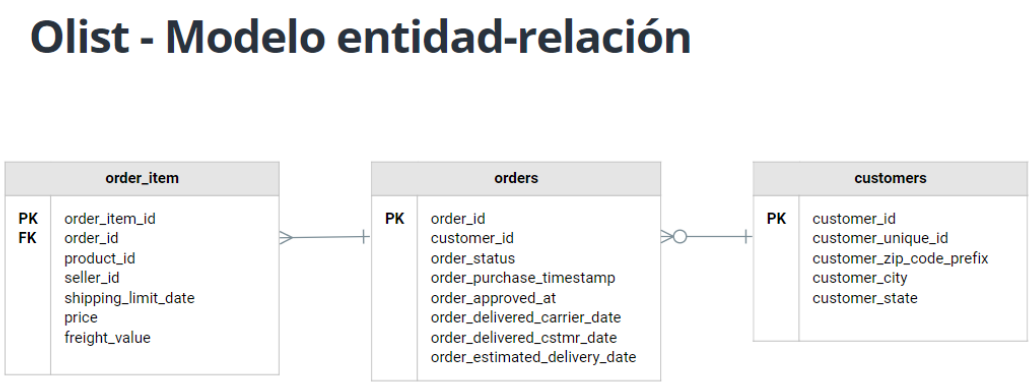

Figura 1: Modelo entidad – relación ordenes Olist

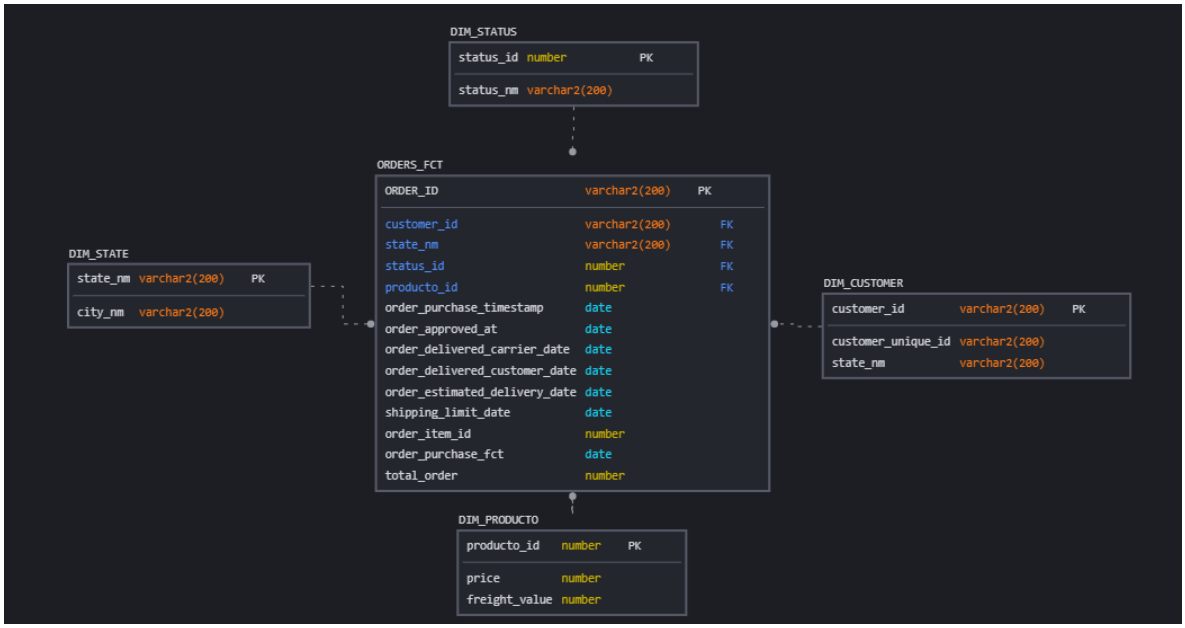

Figura 2: Esquema estrella – Ordenes Olist

Script de creación de tablas dimensionales

Tabla dimensional state

```
CREATE TABLE DIM_STATE
(
  STATE_NM          VARCHAR2(30 BYTE),
  CITY_NM   VARCHAR2(200 BYTE)
);
```

Tabla dimensional customer

```
CREATE TABLE DIM_CUSTOMER
(
  CUSTOMER_ID        VARCHAR2(200 BYTE),
  CUSTOMER_UNIQUE_ID  VARCHAR2(200 BYTE),
  STATE_NM          VARCHAR2(30 BYTE)
);
```

Tabla dimensional producto

```
CREATE TABLE DIM_PRODUCTO
(
  PRODUCT_ID      VARCHAR2(200 BYTE),
  PRICE                NUMBER(38,2),
  FREIGHT_VALUE  NUMBER(38,2)
);
```

Tabla dimensional status

```
CREATE TABLE DIM_STATUS
(
  STATUS_ID         NUMBER,
  ORDER_STATUS  VARCHAR2(200 BYTE)
);
```

Insert a la tabla dimensional status (única tabla dimensional que quedara fija)

```
INSERT INTO  DIM_STATUS
SELECT
CASE WHEN ORDER_STATUS = 'created' THEN 1
    WHEN ORDER_STATUS = 'processing' THEN 2
    WHEN ORDER_STATUS = 'approved' THEN 3
    WHEN ORDER_STATUS = 'invoiced' THEN 4
    WHEN ORDER_STATUS = 'unavailable' THEN 5
    WHEN ORDER_STATUS = 'shipped' THEN 6
    WHEN ORDER_STATUS = 'delivered' THEN 7
    WHEN ORDER_STATUS = 'canceled' THEN 8
    END STATUS_ID,
    A.*
FROM
(SELECT DISTINCT ORDER_STATUS FROM TBL_OLIST_ORDERS) A
ORDER BY 1;
COMMIT;
```

Creación de tablas intermedias para poder llenar la tabla hechos

```
CREATE TABLE TBL_ORDERS_STATUS
(
 ORDER_ID                  VARCHAR2(200 BYTE),
 CUSTOMER_ID               VARCHAR2(200 BYTE),
 ORDER_STATUS              VARCHAR2(200 BYTE),
 ORDER_PURCHASE_TIMESTAMP      DATE,
 ORDER_APPROVED_AT         DATE,
 ORDER_DELIVERED_CARRIER_DATE   DATE,
 ORDER_DELIVERED_CUSTOMER_DATE  DATE,
 ORDER_ESTIMATED_DELIVERY_DATE  DATE,
 STATUS_ID                 NUMBER
);
```

```
CREATE TABLE TBL_DTL_ORDERS_STATUS
(
  ORDER_ID                    VARCHAR2(200 BYTE),
  CUSTOMER_ID                 VARCHAR2(200 BYTE),
  ORDER_STATUS                VARCHAR2(200 BYTE),
  ORDER_PURCHASE_TIMESTAMP    DATE,
  ORDER_APPROVED_AT           DATE,
  ORDER_DELIVERED_CARRIER_DATE  DATE,
  ORDER_DELIVERED_CUSTOMER_DATE  DATE,
  ORDER_ESTIMATED_DELIVERY_DATE  DATE,
  STATUS_ID                   NUMBER,
  ORDER_ITEM_ID               NUMBER(38),
  PRODUCT_ID                  VARCHAR2(200 BYTE),
  SHIPPING_LIMIT_DATE         DATE
);
```

Creación de tabla hechos

```
CREATE TABLE TBL_ORDERS_FCT
(
  ORDER_ID                    VARCHAR2(200 BYTE),
  ORDER_PURCHASE_FCT          DATE,
  CUSTOMER_ID                 VARCHAR2(200 BYTE),
  CUSTOMER_UNIQUE_ID          VARCHAR2(200 BYTE),
  STATUS_ID                   NUMBER,
  ORDER_STATUS                VARCHAR2(200 BYTE),
  ORDER_ITEM_ID               NUMBER(38),
  STATE_NM                    VARCHAR2(30 BYTE),
  CITY_NM                     VARCHAR2(200 BYTE),
  ORDER_PURCHASE_TIMESTAMP    DATE,
  ORDER_APPROVED_AT           DATE,
  SHIPPING_LIMIT_DATE         DATE,
  ORDER_DELIVERED_CARRIER_DATE  DATE,
```

```
  ORDER_DELIVERED_CUSTOMER_DATE  DATE,
  ORDER_ESTIMATED_DELIVERY_DATE  DATE,
  PRICE                      NUMBER(38,2),
  FREIGHT_VALUE              NUMBER(38,2),
  TOTAL_ORDER                NUMBER
);
```

Creación de table resumen

```
CREATE TABLE TBL_SMMRY_ORDERS_FCT
(
  ORDERS                     NUMBER,
  ORDER_PURCHASE_FCT         DATE,
  CUSTOMERS                  NUMBER,
  STATUS_ID                  NUMBER,
  ORDER_STATUS               VARCHAR2(200 BYTE),
  ORDER_ITEM_ID              NUMBER(38),
  STATE_NM                   VARCHAR2(30 BYTE),
  CITY_NM                    VARCHAR2(200 BYTE),
  ORDER_PURCHASE_TIMESTAMP   DATE,
  ORDER_APPROVED_AT          DATE,
  SHIPPING_LIMIT_DATE        DATE,
  ORDER_DELIVERED_CARRIER_DATE   DATE,
  ORDER_DELIVERED_CUSTOMER_DATE  DATE,
  ORDER_ESTIMATED_DELIVERY_DATE  DATE,
  TOTAL_ORDER                NUMBER
);
```

AUTOMATIZACIÓN DEL PROCESO

```
CREATE OR REPLACE PROCEDURE PRCSS_OLIST_ORDERS (P_DT IN DATE DEFAULT TRUNC(SYSDATE-2) )
AS
V_DT DATE := P_DT;


BEGIN

        /* INSERT DE TABLAS DIMENSIONALES QUE SE TRUNCARAN A DIARIO*/

        EXECUTE IMMEDIATE 'TRUNCATE TABLE DIM_STATE';
        EXECUTE IMMEDIATE 'TRUNCATE TABLE DIM_CUSTOMER';
        EXECUTE IMMEDIATE 'TRUNCATE TABLE DIM_PRODUCTO';

        INSERT INTO DIM_STATE
        SELECT DISTINCT CUSTOMER_STATE STATE_NM, CUSTOMER_CITY CITY_NM
        FROM TBL_OLIST_CUSTOMERS
        ORDER BY 1;
        COMMIT;

        INSERT INTO DIM_CUSTOMER
        SELECT DISTINCT CUSTOMER_ID, CUSTOMER_UNIQUE_ID, CUSTOMER_STATE STATE_NM
        FROM TBL_OLIST_CUSTOMERS;
        COMMIT;

        INSERT INTO DIM_PRODUCTO
        SELECT DISTINCT PRODUCT_ID, PRICE, FREIGHT_VALUE
        FROM TBL_OLIST_ORDER_ITEMS;
        COMMIT;

        --- CREACIÓN DE TABLAS INTERMEDIAS PARA LLEGAR A LA TABLA FINAL ---

       /* TABLA QUE TRAE EL DETALLE DE LA ORDEN Y SE AGREGA EL STATUS ID
          SE TRUNCARA A DIARIO, DEPENDIENDO DE LA HORA QUE SE CARGUE LA INFO */
```

```
    EXECUTE IMMEDIATE 'TRUNCATE TABLE TBL_ORDERS_STATUS';

    INSERT INTO TBL_ORDERS_STATUS
    SELECT DISTINCT A.*, CASE WHEN ORDER_STATUS = 'created' THEN 1
    WHEN ORDER_STATUS = 'processing' THEN 2
    WHEN ORDER_STATUS = 'approved' THEN 3
    WHEN ORDER_STATUS = 'invoiced' THEN 4
    WHEN ORDER_STATUS = 'unavailable' THEN 5
    WHEN ORDER_STATUS = 'shipped' THEN 6
    WHEN ORDER_STATUS = 'delivered' THEN 7
    WHEN ORDER_STATUS = 'canceled' THEN 8
    END STATUS_ID
    FROM TBL_OLIST_ORDERS A
    WHERE TRUNC(ORDER_PURCHASE_TIMESTAMP) = V_DT;
    COMMIT;

/* SE REUTILIZA LA TABLA CREADA ANTERIORMENTE Y AGREGA EL PRODUCT ID
Y LA FECHA LIMITE DE ENTREGA AL REPARTIDOR, TAMBIÉN SE TRUNCARA A DIARIO */

    EXECUTE IMMEDIATE 'TRUNCATE TABLE TBL_DTL_ORDERS_STATUS';

    INSERT INTO TBL_DTL_ORDERS_STATUS
    SELECT A.*, B.ORDER_ITEM_ID, B.PRODUCT_ID, B.SHIPPING_LIMIT_DATE
    FROM TBL_ORDERS_STATUS A
    LEFT JOIN TBL_OLIST_ORDER_ITEMS B ON A.ORDER_ID = B.ORDER_ID;
    COMMIT;
```

```
/* TABLA HECHOS, TABLA FINAL*/

EXECUTE IMMEDIATE 'TRUNCATE TABLE TBL_ORDERS_FCT';

INSERT INTO TBL_ORDERS_FCT
SELECT DISTINCT A.ORDER_ID, TRUNC(A.ORDER_PURCHASE_TIMESTAMP)ORDER_PURCHASE_FCT, A.CUSTOMER_ID, B.CUSTOMER_UNIQUE_ID,
D.STATUS_ID, A.ORDER_STATUS, A.ORDER_ITEM_ID, B.STATE_NM, C.CITY_NM, A.ORDER_PURCHASE_TIMESTAMP, A.ORDER_APPROVED_AT,
A.SHIPPING_LIMIT_DATE, A.ORDER_DELIVERED_CARRIER_DATE, A.ORDER_DELIVERED_CUSTOMER_DATE, A.ORDER_ESTIMATED_DELIVERY_DATE,
E.PRICE, E.FREIGHT_VALUE, SUM((E.PRICE*A.ORDER_ITEM_ID)+(E.FREIGHT_VALUE*A.ORDER_ITEM_ID)) TOTAL_ORDER
FROM TBL_DTL_ORDERS_STATUS A
LEFT JOIN DIM_CUSTOMER B ON A.CUSTOMER_ID = B.CUSTOMER_ID
LEFT JOIN DIM_STATE C ON B.STATE_NM = C.STATE_NM
LEFT JOIN DIM_STATUS D ON A.STATUS_ID = D.STATUS_ID
LEFT JOIN DIM_PRODUCTO E ON A.PRODUCT_ID = E.PRODUCT_ID
WHERE TRUNC(A.ORDER_PURCHASE_TIMESTAMP) = V_DT
GROUP BY A.ORDER_ID, TRUNC(A.ORDER_PURCHASE_TIMESTAMP), A.CUSTOMER_ID, B.CUSTOMER_UNIQUE_ID, D.STATUS_ID, A.ORDER_STATUS,
A.ORDER_ITEM_ID, B.STATE_NM, C.CITY_NM, A.ORDER_PURCHASE_TIMESTAMP, A.ORDER_APPROVED_AT, A.SHIPPING_LIMIT_DATE, A.ORDER_DELIVERED_CARRIER_DATE,
A.ORDER_DELIVERED_CUSTOMER_DATE, A.ORDER_ESTIMATED_DELIVERY_DATE, E.PRICE, E.FREIGHT_VALUE;
COMMIT;
```

```
        /* TABLA RESUMEN PARA PODER LLEVARLO A UN REPORTE DE VENTAS O UN REPORTE QUE SEA NECESARIO,
        SE AGREGA EL DELETE POR SI SE NECESITA HACER UN REPROCESO, ASÍ NO SE DUPLICAN LOS DATOS */

        DELETE FROM TBL_SMMRY_ORDERS_FCT  WHERE ORDER_PURCHASE_FCT = V_DT;
        COMMIT;

        INSERT INTO TBL_SMMRY_ORDERS_FCT
        SELECT SUM(CASE WHEN ORDER_ID IS NOT NULL THEN 1 ELSE 0 END) ORDERS, ORDER_PURCHASE_FCT,SUM(CASE WHEN CUSTOMER_ID IS NOT NULL THEN 1 ELSE 0 END)CUSTOMERS,
        STATUS_ID, ORDER_STATUS, ORDER_ITEM_ID, STATE_NM, CITY_NM, TRUNC(ORDER_PURCHASE_TIMESTAMP)ORDER_PURCHASE_TIMESTAMP,
        TRUNC(ORDER_APPROVED_AT)ORDER_APPROVED_AT, TRUNC(SHIPPING_LIMIT_DATE)SHIPPING_LIMIT_DATE,
        TRUNC(ORDER_DELIVERED_CARRIER_DATE)ORDER_DELIVERED_CARRIER_DATE, TRUNC(ORDER_DELIVERED_CUSTOMER_DATE)ORDER_DELIVERED_CUSTOMER_DATE,
        TRUNC(ORDER_ESTIMATED_DELIVERY_DATE)ORDER_ESTIMATED_DELIVERY_DATE, SUM(TOTAL_ORDER)TOTAL_ORDER
        FROM TBL_ORDERS_FCT
        WHERE ORDER_PURCHASE_FCT = V_DT
        GROUP BY ORDER_PURCHASE_FCT, STATUS_ID, ORDER_STATUS, ORDER_ITEM_ID, STATE_NM, CITY_NM,TRUNC(ORDER_PURCHASE_TIMESTAMP),
        TRUNC(ORDER_APPROVED_AT), TRUNC(SHIPPING_LIMIT_DATE), TRUNC(ORDER_DELIVERED_CARRIER_DATE), TRUNC(ORDER_DELIVERED_CUSTOMER_DATE),
        TRUNC(ORDER_ESTIMATED_DELIVERY_DATE) ;
        COMMIT;

END;
```

CONCLUSIONES

Los archivos con los que se trabajo son fundamentales para la gestión y seguimiento de las órdenes, porque proporcionan específicos sobre fechas importantes en cada orden, así como la cantidad, el precio y el flete por producto. Este conjunto de datos permitirá una planificación precisa y un seguimiento detallado de los productos en las órdenes, así como también, la información sobre los clientes, que incluye detalles de ubicación, como el estado y el municipio de envío que serán esenciales para la planificación y la logística de las entregas de los productos a los clientes.

REFERENCIAS

Brazilian E-Commerce Public Dataset by Olist. (2021, October 1). Kaggle. https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce

SqlDBM - Online Database Modeler. (n.d.). https://app.sqldbm.com/Oracle/Draft/

GabsOrellana. (n.d.). *GitHub - GabsOrellana/Prueba-Tecnica-_-Ingeniero-de-Datos*. GitHub. https://github.com/GabsOrellana/Prueba-Tecnica-_-Ingeniero-de-Datos

ANEXOS

A partir de la tabla resumen TBL_SMMRY_ORDERS_FCT se opto por hacer un gráfico de barras del top 10 de ciudades con sus respectivos ingresos diarios, como ejemplo de un posible reporte o KPI acerca de las órdenes generadas en Olist.

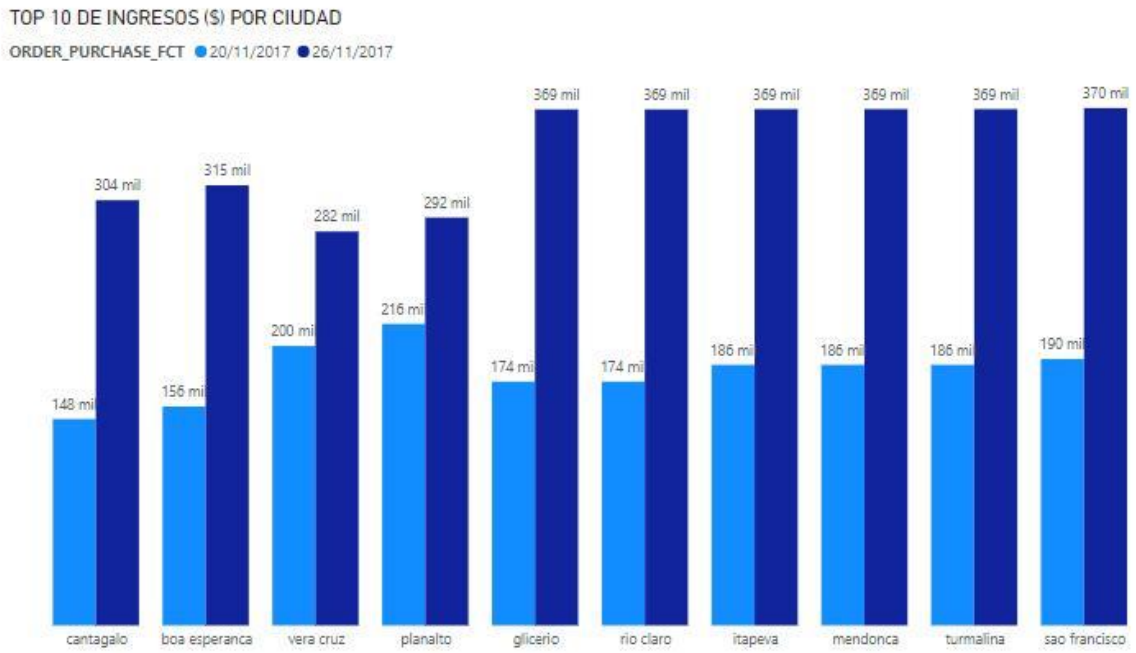

Gráfico 1: Gráfico de barras de top 10 de ingresos diarios por ciudad para el 20 y 26 de noviembre 2017.