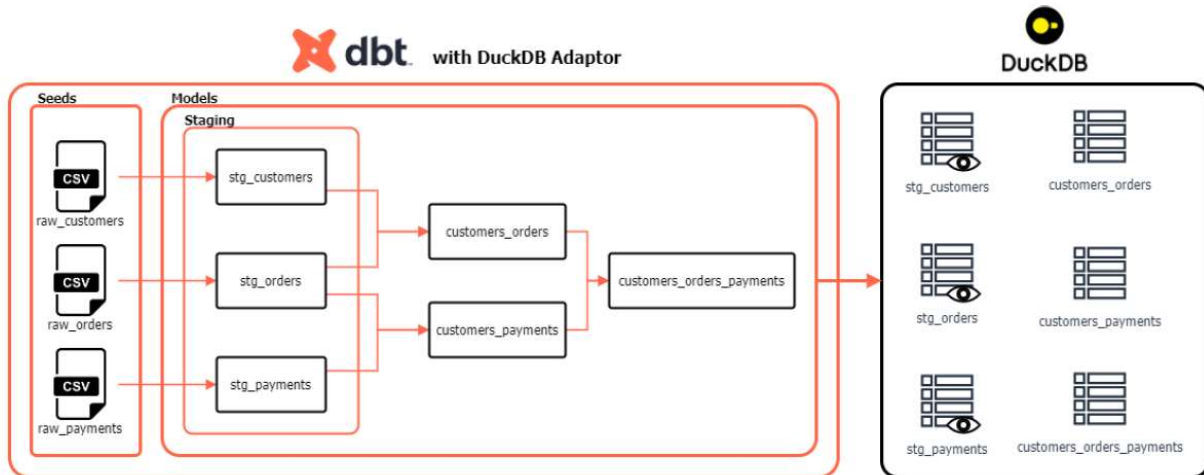


## EJERCICIO 1. CREACIÓN DEL MODELO

Recordemos la siguiente imagen:



### Staging Models

- **Objetivos:** Crear el staging models para todas las tres tablas, `stg_customers`, `stg_orders` y `stg_payments` y materializar como vistas en DuckDB database.
- **Instrucciones:**
  - Crea una carpeta llamada **Staging** dentro de la carpeta **Models** para los archivos SQL que tendrán definido los modelos.
  - Modifica el fichero **dbt\_project.yml** para indicar los modelos dentro de la carpeta staging que serán materializados como vistas.
  - Crea el fichero SQL dentro de la carpeta staging. Los nombres de los ficheros serán los nombres de cada uno de los modelos. Las vistas serán simplemente seleccionando todas las columnas de las tablas origen, pero renombrando algunas columnas:
    - **customers:**
      - `id` renombra `customer_id`
    - **orders:**
      - `id` renombra `order_id`
      - `user_id` renombra `customer_id`
    - **payment**
      - `id` renombra `payment_id`
      - `amount /100` renombra `amount`
  - Crea un fichero **schema.yml** dentro de la carpeta staging y anota todas las columnas de todos los modelos.
  - Puedes realizar cada modelo de forma individual o todos a la vez con el comando `dbt build`.
  - Comprueba si las vistas están creadas seleccionando sus datos en DuckDB database usando `duckcli`.

### Comandos:

- Cargar los ficheros csv ejecutando el comando:  
`dbt seed`
- Compilar un sólo modelo dentro `dbt build`:

*dtb build -f -s <nombre del modelo>*

- Compilar todos los modelos:

*dbt build*

- Ejecutar duckcli y realizar la consulta select from table:

*cd app/src/jaffle\_shop/*

*duckcli jaffle\_shop.duckdb*

*select \* from <nombre de la tabla>*

- Salir de duckcli

*exit*