

DAMG 6210 - DATA MANAGEMENT AND DATABASE DESIGN FINAL PROJECT

GROUP 08 - MEMBERS

Name	NUID
Muskan Deepak Raisinghani	002825398
Yashwanth Behara	002299461
Kameswara Krishna Sai Praneeth Kattamuri	002294248
Dev Mithunisvar Premraj	002893541

TABLE OF CONTENTS

PROBLEM STATEMENT	2
STEPS FOR EXECUTION	2
ER DIAGRAM	3
BUSINESS RULES	4
BUSINESS FLOW	5, 6
DATA FLOW DIAGRAMS	7
DESCRIPTION OF PROC AND VIEWS	8,9
ANALYSIS	9

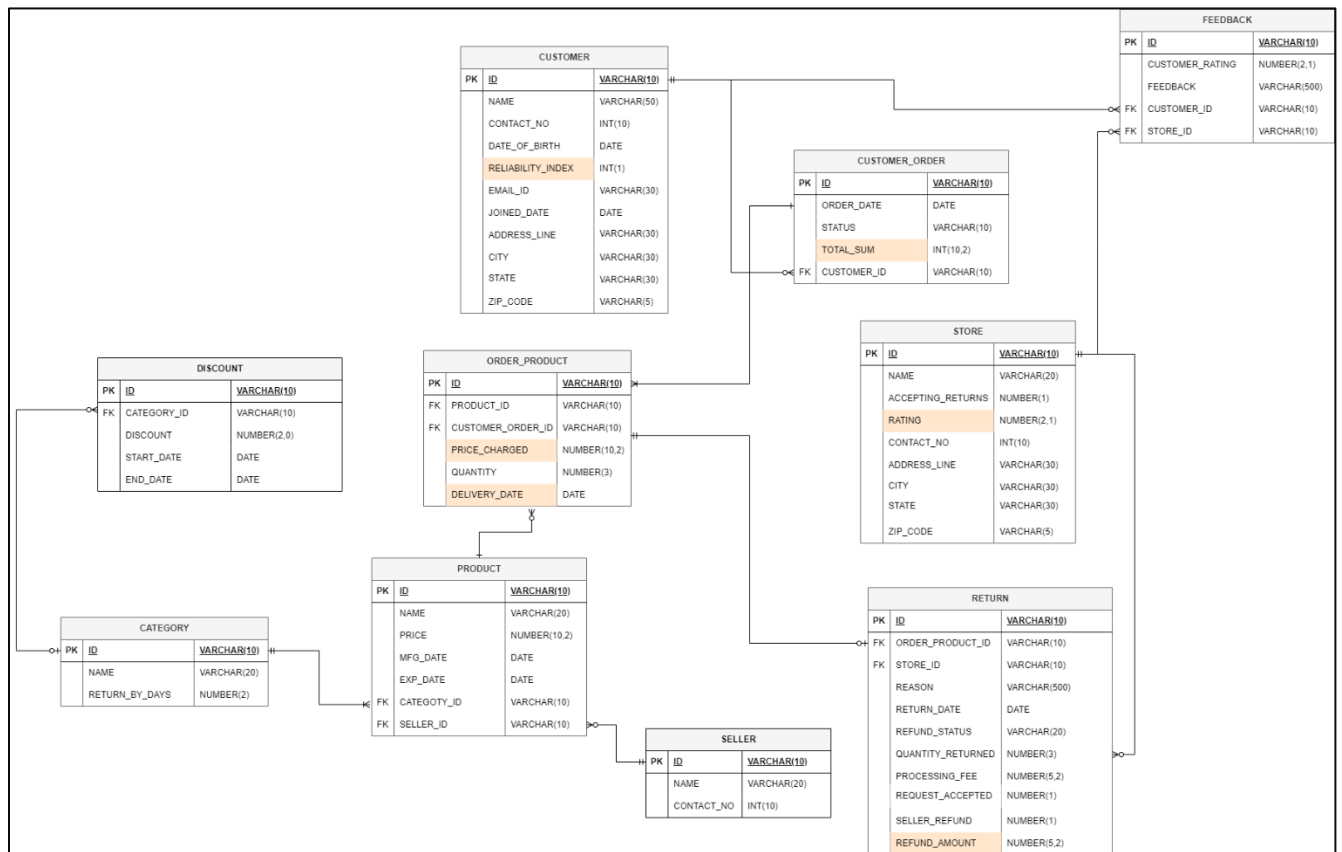
Problem Statement:

Managing returns and refunds in today's supply chain landscape presents notable challenges, encompassing elevated return rates, intricate processing workflows, and insufficient assessment of customer reliability. These hurdles translate into amplified costs, possible disruptions, and customer discontent. To tackle these issues effectively, we advocate for the implementation of a robust online transaction management system.

Steps to execute the scripts, to get the expected output and following the flow

1. **Login with admin credentials.**
2. Execute the admin_grants.sql to grant permissions to the business manager.
3. **Login with business manager credentials.**
 - a. Execute the DDL_DML.sql file to create tables and populate them with necessary default data.
 - b. Execute the views.sql file to create the required views.
 - c. Execute the reports.sql to create views for analysis.
 - d. Execute the procedures.sql file to create the required procedures.
4. Login with customer credentials and execute the 1_customer_script.sql
5. Login with seller credentials and run the 2_seller_script.sql.
6. Login with customer credentials and execute the 3_customer_script.sql
7. Login with store credentials and execute the 4_store_script.sql
8. Login with seller credentials and execute the 5_seller_script.sql
9. Login with store credentials and execute the 6_store_script.sql
10. Login with business manager credentials and execute the 7_reports_script.sql

ER DIAGRAM



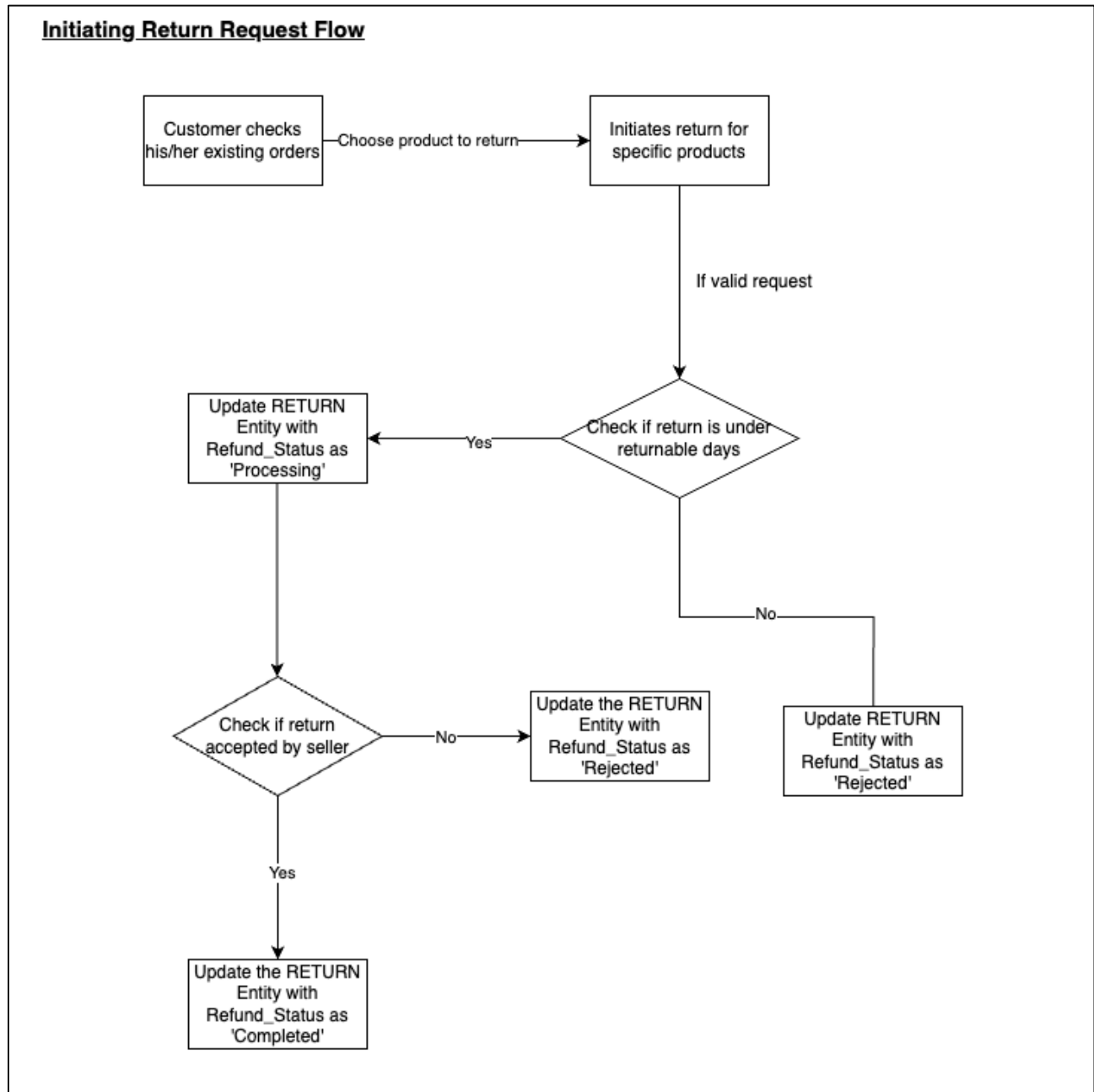
BUSINESS RULES

- Each product is categorized, influencing the final price for customers. On special dates, each product category is associated with a distinct discount rate, resulting in variable final prices based on the date of purchase.
 - The total order value is a derived attribute, calculated by considering the number of products in the order and their respective prices at the time of purchase.
 - Products are ineligible for return if the customer initiates a return request after the specified "returnable date," which varies by product category, ensuring a streamlined and standardized return process.
 - When a customer initiates a return order, the system initially verifies its conformity with the product's returnable date. Subsequently, a further examination is conducted based on the customer's reliability index to determine the appropriate refund amount.
 - Customers can share feedback about the store where they processed their product return. This feedback serves as valuable information for other customers, assisting them in making informed decisions about which store to make returns.
 - Each store receives a rating derived from the average of customer feedback, offering a concise and informative measure of its overall performance.
 - Customers will receive a notification specifying the reason for the rejection of their return request.
 - If the request is rejected, then the processing fee will be 0.
 - The processing fee for the return request would be:
 - $\text{Total Refund} = \text{Price charged} - \text{Processing fee}$
 - Order delivery date would be derived attribute and will be derived from Purchased date.
 - For food products, it will be 1-day delivery.
 - For the other products, it will be 2 days.
 - Only existing customers can initiate a return request and access order and product tables.
 - Only existing sellers can fetch categories and add products.
- Only existing store managers can update the status of the stores.

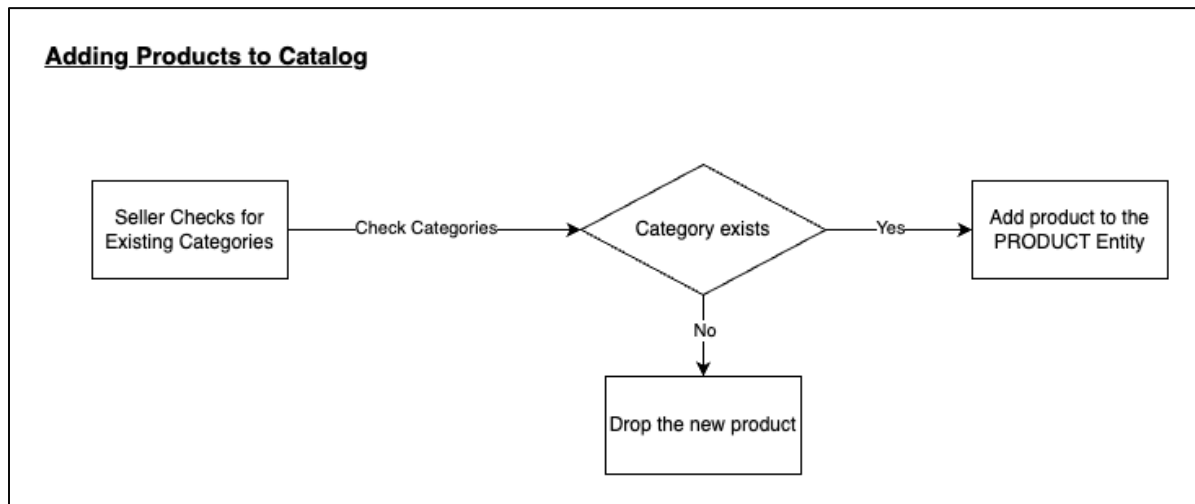
BUSINESS FLOW

There are 4 business flows in our Returns and Refunds system:

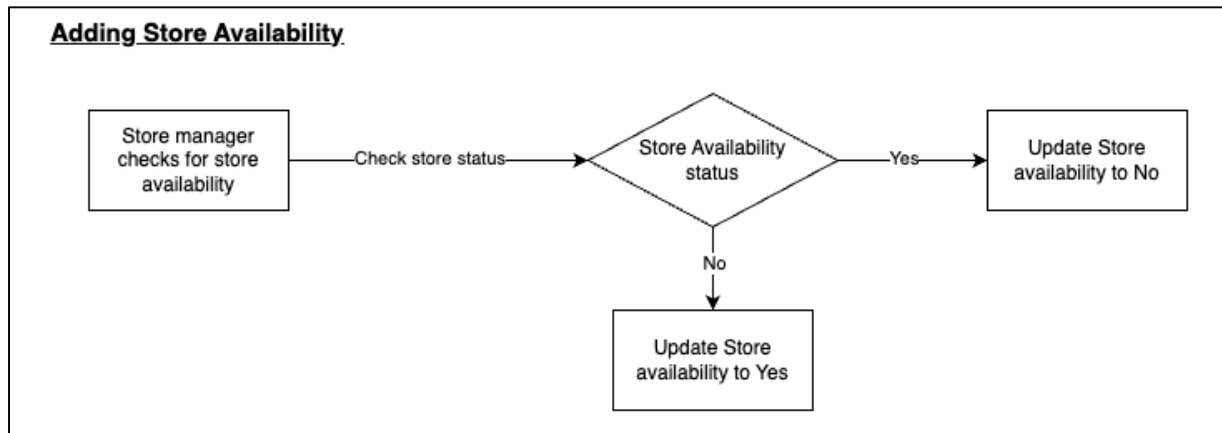
1. Initiating Return Request



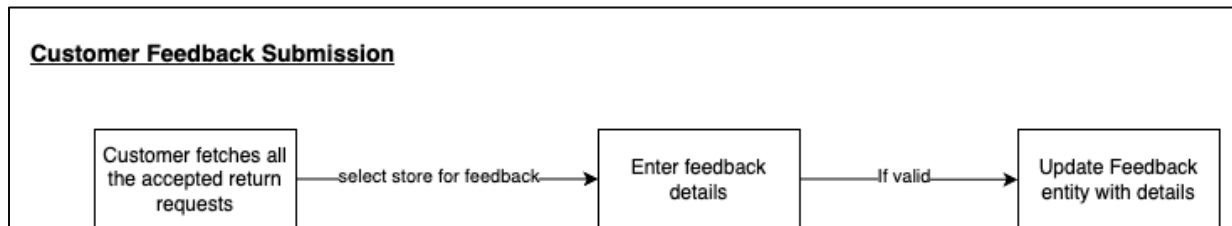
2. Adding products to catalog



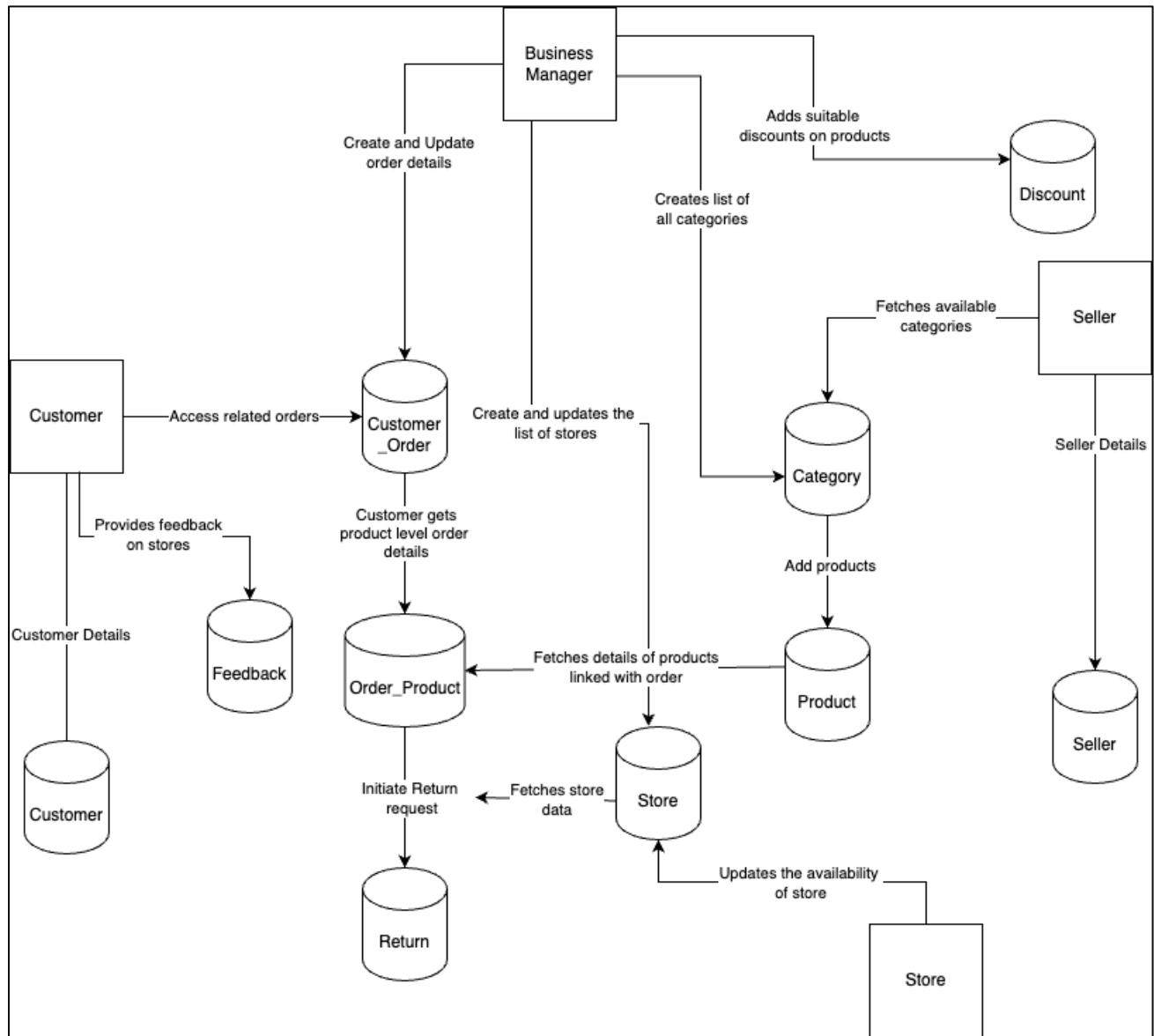
3. Adding store availability



4. Customer Feedback Submission



DATA FLOW DIAGRAM



DESCRIPTION OF PROCEDURES AND VIEWS

PROCEDURES

1. CREATE_RETURN
Procedure for customer to create return.
2. SUBMIT_FEEDBACK
Procedure for customer to submit feedback to stores after the return.
3. SHOW_RETURNS_REQUEST
Procedure for customer to show the return requests they have raised.
4. GET_FEEDBACK_FOR_STORE
Procedure for customer to view the ratings and feedback of stores.
5. GET_RETURNED_PRODUCTS
Procedure for customer to view the products they have received.
6. FILTER_STORE_FOR_FEEDBACK
Procedure for customer to filter the store based on the order returned.
7. ADD_PRODUCT
Procedure for seller to add the product.
8. UPDATE_SELLER_REFUND
Procedure for seller to update the refund status. The seller can accept or reject the return with this procedure.
9. UPDATE_STORE_AVAILABILITY
Procedure for store to update their availability.
10. GET_STORE_RATING
Procedure to get the rating for particular store.
11. FILTER_ACCEPTED_RETURNS
Procedure to filter the accepted returns.
12. GET_SYSTEM_APPROVED_RETURNS
Procedure for seller to view the return requests that are pending approval.
13. GET_RETURNED_PRODUCT_ANALYSIS
Procedure for seller to view which products are returned with their frequencies.
14. VIEW_CATEGORIES_AVAILABLE
Procedure for seller to check categories available in the system.

VIEWS

1. CUSTOMER_RELIABILITY_INDEX
View for calculating customer reliability index based on the returned orders and total orders.
2. ORDER_DELIVERY_DATE
View for calculating delivery date based on the category of product.
3. NUMBER_OF_RETURNABLE_DAYS
View for calculating the returnable days for any product based on the category.
4. CATEGORY_VIEW
View for seller to check all the categories available in the system.

5. RETURNED_PRODUCTS_DETAILS
View for seller to check the product and the frequency of returns made to make informed decisions before adding product.
6. CHECK_APPROVED_RETURNS_BY_SYSTEM
View for seller to check the returns they have to approve or reject based on the product condition.
7. PRODUCT_DISCOUNT_ASSOCIATION
View for calculating price charged for order product. If a discount is available, price charged is equal to product price – discount applied.
8. ORDER_TOTAL_PRICE_PER_UNIT
View for customer to see total order value.
9. REFUND_AMOUNT_VIEW
View for calculating refund amount after deducting processing fee from price charged.
10. ACCEPTED_RETURNS_VIEW
View for customer to check the approved returns by seller.
11. QTY_AVAILABLE_FOR_RETURN
View for calculating the product count available for return.
12. PROCESSING_FEES_BY_YEAR_MONTH
View for calculating the processing fee based on the reliability index.
13. SELLER_RETURNED_PRODUCTS
View required to check the seller and the products returned associated with them.
14. STORE_FOR_FEEDBACK
View for customer to check the feedback given to store.
15. STORE_AVERAGE_RATING_VIEW
View for checking the store rating compared to average rating.

ANALYSIS

1. Month-wise returns
The month-wise returns provide insights into the returns made in particular month for each product. It provides summary of the total number of product returns, grouped by year, month (with the month name), and product name. This view facilitates analysis of product return patterns over time.
2. Product-wise returns
The product-wise returns provide the summary of the total number of returns for products from each seller, grouped by year and month (with the month name). This view helps in assessing seller performance and product quality by analyzing return trends associated with sellers.
3. Total processing fees
Total processing fees provides analysis on the amount received as a part of processing fees.

4. Seller-wise returns

The seller-wise returns provide insights for the products that have been returned.

5. Comparison of average store ratings

The above analysis provides insights about the store rating and the average rating to see how a store is performing compared to other stores.