Use Case

Building a Cart Analysis for Myph

Aim: The primary aim of this experiment is to design and test an amended producer data model that facilitates efficient product retrival by category and to evaluate the capability of a relational database application to handle these transactions and subsequent surplus selections

① Initial Data Model

The existing model focuses on individual product details

Field	Type	Description
Product_id	Unique ID	primary key, identifies the product
title	string	product name
description	Text	Detais about the product
stock_quality	integer	current inventory level
pricing	decimal	product price
category_name	string	The category that product belongs to (eg: 'smartphones', (Accessories).

② Proposed Amended Data Model

To enable quick betrival of all products within a category and support a category tree structure, a new model component (likely ^ seperate collection/table) is proposed.

Field	Type	Description
category.id	UniqueId	Primary key for the category
category-name	string	Name of the category (eg: 'smart phones')
category-path	Array/string	The path in the category tree (eg; Electronics/phones/Smartphones) crucial for the requirement
product_list	Array of ID's	collection of product ID's belonging for the requirement.

Procedure and Queries

Step	Procedure	Example Queries
Data setup	Populate the 'Product' and 'Category' models wih sample data for myph phones	Insert: Product (id: P0001, title: Myph21, category_name: 'Smart phones')

Category Retrival Test	Query the category model to retrive the list of products for a specific category	Query (Goal):
		SELECT product_list FROM category WHERE category-name = 'smartphone'

Relational Database Evaluation	Test standard SQL transactional properties (ACID) during a cart update/checkout process	SQL Transaction:
		BEGIN TRANSACTION;
		UPDATE stock SET quantity =quantity-1 WHERE id = 'P001';
		INSERT INTO orders (pool, userid) VALUES ('P001', 'U007');
		COMMIT;

Outlier Analysis Test	Run an analysis query to identify surplus selections (products often added to the cart but never purchased).	Query (Goal):
		SELECT product_id, COUNT(cart_adds)/ COUNT(purchases) AS add-to-purchase_ratio FROM analysis-log WHERE add-to-purchase_ratio> [Threshold]

① Product Retrival by category

- Query output: The query in step2 directly returns the list of the product ID's (eg: (P001, P002, P007)), requiring only one lookup in the category collection/table.

② outlier/surplus selection Analysis

By tracking add-to-cart event, remove from cart events and find purchases, one can identify surplus selections.

③ Relational Database Application

- Relational database application can answer these transactions? Yes, definitely
- Relational databases (like postgresQL MySQL) are specificclly designed to handle e-commerce transactions using ACID (Atomicity consistency, isolation, Durability) properties.

④ Recovery through carting and commerce.

- Transactional recovery :- This is handled by the Relational database management system (RDMS). If a transaction fails mid-way the RDBms automatically rolls back the changes, ensuring data integrity
- Customer recovery (carting) : This often involves "Abandoned cart recovery" campaigns. The data recorded in the cart is used to send remainders to the cart is used to send remainders to the customers to encourage them to complete the purchase, thus "recovering" a

potentially lost sale.

Result:-

The amended data model successfully supports cart analysis by enabling quick retrival of product by category and facilitates the identification of outlier selections. Relational database (RDBMS) are idea/ for handling these transactions and ensuring data recovery via transactional roll back and abandoned cart strategies.