Retailer (Market Mint) Database

The **Marketmint retailer database** is a well-structured relational schema designed to manage retail operations, ranging from product cataloguing and inventory to customer management, order processing, billing, and vendor shipments. The central components of the system are organized into multiple interconnected tables.

1. **Product Table**: This table stores the catalog of products available in the system. Each product has a unique product_id and includes fields like product_name, brand_id, type_id, UPC, size, and price. Products are associated with a brand and a product type, allowing for categorization and filtering.
2. **Brand Table**: Contains details about product brands, with a unique brand_id and the brand_name. This allows linking multiple products to a single brand.
3. **Product_Type Table**: Defines categories of products, using type_id and type_name. This table helps classify products (e.g., electronics, clothing, etc.).
4. **Store Table**: Represents physical or online retail outlets. It includes store_id, store_name, address, city, and state, supporting location-based operations and analytics.
5. **Inventory Table**: Tracks the quantity of each product available at each store. It uses a composite key of store_id and product_id, and the field quantity_available shows stock levels. This supports inventory checks and replenishment decisions.
6. **Customer Table**: Holds customer information, including customer_id, customer_name, email, phone, address, city, and date_joined. This data enables customer relationship management and segmentation, especially by city.
7. **Orders Table**: Records customer purchases, with each order having an order_id, customer_id, store_id, order_date, and net_amount. It links customers and stores, supporting sales tracking and reporting.
8. **Bill Table**: Provides a breakdown of each order into specific products. It contains order_id, product_id, quantity, unit_price, and total_price, offering detailed billing and revenue insights.
9. **Vendor Table**: Stores information about suppliers, including vendor_id, vendor_name, and contact_info. Vendors supply products to stores via shipments.
10. **Shipment Table**: Tracks product shipments from vendors to stores. Each shipment includes shipment_id, vendor_id, store_id, shipment_date, and delivery_date, allowing for logistics and delivery analysis.
11. **ShipmentDetails Table**: Breaks down each shipment by product, storing shipment_id, product_id, and quantity_shipped, which links logistics with inventory replenishment.

This schema provides comprehensive support for managing a retail ecosystem. City-level information is stored for both **customers** and **stores**, enabling powerful **location-based queries**, such as analyzing which cities generate the most orders or identifying customer clusters. The **separation of orders and bills**, as well as **shipments and their details**, promotes normalization, ensuring efficiency, consistency, and flexibility in querying and reporting across business functions like sales, supply chain, and customer service.