

Database Schema Explanation for Supermarket WhatsApp Bot Project – ASKMART

This document explains the structure of the database for the **Supermarket WhatsApp Bot Project**. The database is designed to manage users, products, orders, feedback, promotions, loyalty points, shipping information, and admin users. The schema also supports efficient querying through indexes.

1. Users Table

Purpose:

The `users` table stores information about customers using the system. It helps track their contact details, loyalty points, language preferences, and account status.

Fields:

- `user_id` (Primary Key): A unique identifier for each user.
 - `name` : Name of the user.
 - `phone_number` : Unique phone number of the user.
 - `email` : User's email address (unique).
 - `loyalty_points` : Points accumulated by the user for loyalty rewards.
 - `language` : Preferred language for communication (default is English).
 - `created_at` : Timestamp when the user record was created.
 - `updated_at` : Timestamp when the user record was last updated.
 - `status` : Status of the user's account (e.g., active, suspended).
 - `last_login` : Timestamp of the user's last login.
-

2. Products Table

Purpose:

The `products` table stores information about the products available in the supermarket. It tracks product details, stock levels, pricing, and categorization.

Fields:

- `product_id` (Primary Key): A unique identifier for each product.
 - `name` : Name of the product.
 - `description` : Detailed description of the product.
 - `price` : Price of the product.
 - `stock_quantity` : The quantity of the product available in stock.
 - `image_url` : URL of the product image.
 - `category` : Category to which the product belongs (e.g., dairy, fruits).
 - `brand` : Brand of the product.
 - `sku` : A unique SKU identifier for the product.
 - `weight` : The weight of the product (in kilograms or preferred unit).
 - `is_active` : Boolean indicating whether the product is currently active.
 - `created_at` : Timestamp when the product was created.
 - `updated_at` : Timestamp when the product was last updated.
-

3. Orders Table

Purpose:

The `orders` table records the details of customer orders, including the total price, order status, shipping details, and payment information.

Fields:

- `order_id` (Primary Key): A unique identifier for each order.
- `user_id` (Foreign Key): The ID of the user who placed the order.
- `status` : Status of the order (e.g., pending, shipped, delivered).
- `total_price` : Total price of the order.
- `shipping_address` : The address where the order is to be delivered.
- `delivery_date` : Estimated or actual delivery date.
- `payment_status` : Status of the payment (e.g., pending, paid).
- `payment_method` : Payment method used for the order (e.g., credit card, PayPal).
- `shipping_method` : Shipping method chosen (e.g., home delivery, pick-up).

- `order_date` : Timestamp when the order was created.
 - `updated_at` : Timestamp when the order was last updated.
-

4. Order Items Table

Purpose:

The `order_items` table stores the details of each product included in an order, including the quantity and price.

Fields:

- `order_item_id` (Primary Key): A unique identifier for each item in an order.
 - `order_id` (Foreign Key): The ID of the associated order.
 - `product_id` (Foreign Key): The ID of the product being purchased.
 - `quantity` : The number of units of the product in the order.
 - `unit_price` : The price of a single unit of the product.
 - `discount` : Discount applied to the product.
-

5. Shipping Information Table

Purpose:

The `shipping_info` table records shipping details for each order, including the shipping method, tracking number, and pick-up option.

Fields:

- `shipping_id` (Primary Key): A unique identifier for each shipping record.
- `order_id` (Foreign Key): The ID of the associated order.
- `shipping_method` : The method used for shipping (e.g., delivery, pick-up).
- `tracking_number` : Tracking number for the order shipment.
- `shipping_address` : The address where the order is shipped.
- `status` : Current shipping status (e.g., pending, delivered).
- `pick_up_option` : Boolean indicating if the customer opted for pick-up.
- `shipping_date` : Date when the order was shipped.

- `pick_up_location` : Location where the customer can pick up the order.
-

6. Admin Users Table

Purpose:

The `admin_users` table stores information about the business administrators who manage the bot and web dashboard.

Fields:

- `admin_id` (Primary Key): A unique identifier for each admin user.
 - `name` : Name of the admin user.
 - `email` : Admin's email address.
 - `password_hash` : Hash of the admin's password for security.
 - `role` : Role of the admin (e.g., staff, manager, super admin).
 - `created_at` : Timestamp when the admin record was created.
 - `last_login` : Timestamp of the admin's last login.
 - `status` : Indicates whether the admin account is active or suspended.
 - `password_reset_token` : Token for resetting the admin password.
 - `reset_token_expiration` : Expiry timestamp for the reset token.
-

7. Feedback Table

Purpose:

The `feedback` table stores customer feedback for products, including ratings and comments, to gather insights into product performance and customer satisfaction.

Fields:

- `feedback_id` (Primary Key): A unique identifier for each feedback record.
- `user_id` (Foreign Key): The ID of the user who provided the feedback.
- `product_id` (Foreign Key): The ID of the product being reviewed.
- `rating` : Rating given by the customer (between 1 and 5).
- `comments` : Additional comments by the customer.

- `resolved` : Boolean indicating if the feedback issue has been resolved.
 - `anonymous` : Boolean indicating whether the feedback is anonymous.
-

8. Loyalty Points Table

Purpose:

The `loyalty_points` table tracks the loyalty points for each user, including the points earned, redeemed, and current balance.

Fields:

- `loyalty_id` (Primary Key): A unique identifier for each loyalty record.
 - `user_id` (Foreign Key): The ID of the user to whom the loyalty points belong.
 - `points_earned` : The total points the user has earned.
 - `points_redeemed` : The total points the user has redeemed.
 - `current_points` : The current points balance for the user.
 - `tier` : The tier level of the user (e.g., bronze, silver, gold).
 - `points_expiration` : Expiry date for the loyalty points.
 - `last_updated` : Timestamp when the loyalty record was last updated.
-

9. Promotions Table

Purpose:

The `promotions` table stores details about ongoing promotions or discounts, including the products or categories involved and the discount percentage.

Fields:

- `promotion_id` (Primary Key): A unique identifier for each promotion.
- `title` : Title of the promotion.
- `description` : Detailed description of the promotion.
- `discount_percentage` : Discount percentage offered by the promotion.
- `start_date` : Start date of the promotion.
- `end_date` : End date of the promotion.

- `product_category` : The category of products that the promotion applies to.
 - `created_at` : Timestamp when the promotion was created.
-

Indexes for Performance:

To optimize query performance, several indexes are created on commonly queried fields:

- `idx_users_phone` : Index on the user's phone number for faster search and lookups.
 - `idx_users_email` : Index on the user's email address for faster search.
 - `idx_products_category` : Index on the product category to speed up category-based queries.
 - `idx_products_brand` : Index on the product brand for efficient brand-based searches.
 - `idx_orders_user` : Index on the `user_id` in orders for fast user order lookups.
 - `idx_orders_status` : Index on the order status for quick filtering of order statuses.
 - `idx_order_items_order` : Index on `order_id` for fast order-item lookups.
 - `idx_feedback_user` : Index on `user_id` for quick access to a user's feedback.
 - `idx_loyalty_user` : Index on `user_id` for fast access to loyalty points.
 - `idx_shipping_order` : Index on `order_id` for quick shipping information retrieval.
-

Conclusion

This schema is designed to efficiently manage customer data, orders, feedback, shipping, loyalty programs, and promotions. It is scalable, with performance optimizations via indexes, and provides the necessary structure to support the Supermarket WhatsApp Bot and its accompanying web dashboard for business owners.