

8. Supabase Database Schema (SQL & Explanations)

Use PostgreSQL on Supabase. Below are the core tables and an example CREATE TABLE DDL for each. Run them in Supabase SQL editor or your migration tool.

Note: UUIDs, timestamps, and JSONB are used for flexibility and compatibility with personalization.

1) Table: users

| Column | Type | Default | Constraints | Notes |
|----------------|------------|-------------------|------------------|-------------------------------|
| id | uuid | gen_random_uuid() | Primary key | Unique user ID |
| name | text | — | NOT NULL | Full name of user |
| email | text | — | NOT NULL, UNIQUE | Used for login |
| password_hash | text | — | NOT NULL | Hashed password |
| email_verified | bool | false | — | If email is verified or not |
| phone | text | | | Phone/mobile number |
| is_active | bool | true | — | If the user account is active |
| created_at | timestampz | now() | — | Timestamp of registration |
| updated_at | timestampz | now() | — | Timestamp of updattion |
| last_login | timestampz | | — | Timestamp of last login |

2) Table: addresses

| Column | Type | Default | Constraints | Notes |
|--------------|------------|-------------------|---------------------------|------------------------------------|
| id | uuid | gen_random_uuid() | Primary key | Unique address ID |
| user_id | uuid | — | FOREIGN KEY → users.id | Owner of the address |
| line1 | text | — | NOT NULL | Address line 1 |
| line2 | text | — | — | Optional additional address |
| city | text | — | NOT NULL | City name |
| state | text | — | NOT NULL | State/Province |
| country | text | 'India' | — | Default India |
| zip_code | text | — | NOT NULL | Postal code |
| is_default | bool | false | — | If the address is default for user |
| address_type | text | 'home'::text | — | Home, work, etc |
| phone | text | — | — | Receipients phone number |
| landmark | text | — | — | Address near landmark |
| created_at | timestampz | now() | — | Created timestamp |

3) Table: products

| Column | Type | Default | Constraints | Notes |
|--------------|------------|-------------------|----------------------|------------------------------|
| id | uuid | gen_random_uuid() | Primary key | Product ID |
| title | text | — | NOT NULL | Product name |
| description | text | — | — | Product details |
| price | int4 | — | NOT NULL | Base price (no decimals) |
| stock | Int4 | — | NOT NULL | No of stock |
| category_id | uuid | — | FK to category.id | Category name |
| img_url | text | — | — | Cloudinary URLs |
| created_at | timestampz | now() | — | Created timestamp |
| sku | text | — | UNIQUE | Product SKU |
| has_variants | bool | false | — | If the product has vairants. |

4) Table: product_variants

| Column | Type | Default | Constraints | Notes |
|------------|--------|-------------------|--|---|
| id | uuid | gen_random_uuid() | Primary key | Variant ID |
| product_id | uuid | — | FOREIGN KEY → products.id ON DELETE CASCADE | Parent product |
| sku | text | — | UNIQUE | Variant SKU |
| attributes | jsonb | — | — | e.g. {"color": "Pink", "size": "Small"} |
| price | int4 | — | — | Variant-specific price |
| stock | int4 | — | — | Available stock |
| is_default | bool | false | — | Default variant for a product |
| weight | float8 | — | — | Weight of the product |
| img_url | text | — | — | Variant's img url from cloudinary |

5) Table: orders

| Column | Type | Default | Constraints | Notes |
|------------------|------------|-------------------|---------------------------|--|
| id | uuid | gen_random_uuid() | Primary key | Order ID |
| user_id | uuid | — | FOREIGN KEY → users.id | Buyer |
| status | text | 'pending' | — | pending / confirmed / shipped / delivered / cancelled |
| subtotal | int4 | — | — | Item total before shipping/discount |
| shipping_cost | int4 | — | — | Delivery charge |
| final_total | int4 | — | — | Final payable amount |
| shipping_address | jsonb | — | — | Snapshot of delivery address |
| payment_status | text | 'unpaid' | — | unpaid / paid / refunded / failed |
| payment_id | text | — | UNIQUE | Transaction ID (Razorpay/Stripe) |
| created_at | timestampz | now() | — | Order created timestamp |
| placed_at | timestampz | now() | — | Time at which order was placed. |
| custom_bundle_id | uuid | — | FK → custom_bundles.id | Links order to a specific user- created bundle |
| bundle_type | text | 'single' | — | 'single', 'brand_bundle', or 'custom_bundle' |

6) Table: order_items

| Column | Type | Default | Constraints | Notes |
|--------------------|------------|-------------------|-----------------------------------|--|
| id | uuid | gen_random_uuid() | Primary key | Order item ID |
| order_id | uuid | — | FOREIGN KEY → orders.id | Associated order |
| product_variant_id | uuid | — | FOREIGN KEY → product_variants.id | Purchased variant |
| quantity | int4 | 1 | NOT NULL | Quantity ordered |
| price | int4 | 0 | NOT NULL | Price snapshot at order time |
| created_at | timestampz | now() | — | Order item created timestamp |
| bundle_origin | text | — | — | 'brand_bundle', 'custom_bundle', or NULL |
| bundle_id | uuid | — | — | Links to bundles.id, custom_bundles.id, or bundle_user_public.id |

7) Table: payments

| Column | Type | Default | Constraints | Notes |
|-------------|------------|-------------------|-------------------------|----------------------------------|
| id | uuid | gen_random_uuid() | Primary key | Payment ID |
| user_id | uuid | — | FOREIGN KEY → user.id | FK to user.id |
| order_id | uuid | — | FOREIGN KEY → orders.id | FK to order.id |
| provider | text | 'RazorPay'::text | NOT NULL | Razorpay/Stripe |
| payment_id | text | — | — | Razorpay payment id |
| amount | int4 | — | NOT NULL | Final amount to be paid |
| currency | text | 'INR'::text | NOT NULL | Which Currency |
| status | text | — | NOT NULL | Status of payment. |
| is_success | bool | false | — | If the payment is succeeded |
| failure_msg | text | — | — | Error msg if payment gets failed |
| created_at | timestampz | now() | — | Timestamp |

8) Table: shipments

| Column | Type | Default | Constraints | Notes |
|----------|------|-------------------|-------------------------|---------------|
| id | uuid | gen_random_uuid() | Primary key | Shipment ID |
| order_id | uuid | — | FOREIGN KEY → orders.id | Related order |

| | | | | |
|--------------------|------------|-----------------|--------|----------------------------------|
| courier | text | — | — | Courier name |
| awb | text | — | UNIQUE | Air Waybill (tracking ID) |
| status | text | “packing”::text | — | shipped / in_transit / delivered |
| estimated_delivery | date | — | — | Expected arrival date |
| tracking_history | jsonb | — | — | Courier updates |
| created_at | timestampz | now() | — | Timestamp created |

9) Table: gift_previews

| Column | Type | Default | Constraints | Notes |
|-----------------|------------|-------------------|---------------------------|---------------------------------------|
| id | uuid | gen_random_uuid() | Primary key | Preview ID |
| user_id | uuid | — | FOREIGN KEY → users.id | Owner |
| preview_url | text | — | — | Cloudinary link for preview |
| personalization | jsonb | — | — | e.g. {"text": "Happy Birthday Kajal"} |
| created_at | timestampz | now() | — | Created timestamp |

10) Table: categories

| Column | Type | Default | Constraints | Notes |
|-------------|------------|-------------------|---------------------|--------------------------------|
| id | uuid | gen_random_uuid() | Primary key | Category ID |
| name | text | — | NOT NULL, UNIQUE | Category name (e.g. 'Pouches') |
| description | text | — | — | Optional category description |
| created_at | timestampz | now() | — | Created timestamp |

11) Table: coupons

| Column | Type | Default | Constraints | Notes |
|-----------------|------|-------------------|---------------------|----------------------------|
| id | uuid | gen_random_uuid() | Primary key | Coupon ID |
| code | text | — | NOT NULL, UNIQUE | Coupon code string |
| description | text | — | — | coupon description |
| discount_type | text | ‘Percent’::text | — | flat / percent |
| discount_value | int4 | — | NOT NULL | Discount amount |
| min_order_value | int4 | — | — | Minimum subtotal required |
| max_discount | int4 | — | — | Maximum discountable value |
| start_date | date | — | — | Start date of coupon |
| end_date | date | — | — | Expiration date |

| | | | | |
|----------------|------------|-------|---|---|
| is_active | boolean | true | — | Active status |
| usage_limit | int4 | — | — | Number of coupons limit |
| usage_per_user | int4 | 1 | — | Limit of a coupon for a user |
| usage_count | int4 | 0 | — | Number of universal usage for a coupon. |
| created_at | timestampz | now() | — | Created timestamp |

12) Table: coupons_applied

| Column | Type | Default | Constraints | Notes |
|-----------------|------------|-------------------|-----------------------------|-----------------------|
| id | uuid | gen_random_uuid() | Primary key | Record ID |
| order_id | uuid | — | FOREIGN KEY → orders.id | Which order used it |
| coupon_id | uuid | — | FOREIGN KEY → coupons.id | Which coupon applied |
| discount_amount | int4 | — | — | Amount discounted (₹) |
| created_at | timestampz | now() | — | When applied |

13) Table: reviews

| Column | Type | Default | Constraints | Notes |
|------------|------------|-------------------|------------------------------|-------------------|
| id | uuid | gen_random_uuid() | Primary key | Review ID |
| product_id | uuid | — | FOREIGN KEY → products.id | Product reviewed |
| user_id | uuid | — | FOREIGN KEY → users.id | Reviewer |
| rating | int2 | — | NOT NULL | Rating (1–5) |
| comment | text | — | — | Optional feedback |
| created_at | timestampz | now() | — | Timestamp |

14) Table: wishlist

| Column | Type | Default | Constraints | Notes |
|------------|------------|-------------------|------------------------------|--------------------|
| id | uuid | gen_random_uuid() | Primary key | Wishlist item ID |
| user_id | uuid | — | FOREIGN KEY → users.id | Owner |
| product_id | uuid | — | FOREIGN KEY → products.id | Wishlisted product |
| created_at | timestampz | now() | — | Added date |

15) Table: carts

| Column | Type | Default | Constraints | Notes |
|------------|------------|-------------------|------------------------|-----------------------|
| id | uuid | gen_random_uuid() | Primary key | One Cart ID per user |
| user_id | uuid | — | FOREIGN KEY → users.id | Owner |
| session_id | text | — | — | Guest user session id |
| expires_at | timestampz | now() + 7 days | — | Autoclear guest carts |
| created_at | timestampz | now() | — | Added timestamp |
| updated_at | timestampz | now() | — | Added timestamp |

16) Table: cart_items

| Column | Type | Default | Constraints | Notes |
|--------------------|------------|-------------------|-----------------------------------|--|
| id | uuid | gen_random_uuid() | Primary key | Cart item ID |
| user_id | uuid | — | FOREIGN KEY → users.id | Owner |
| product_variant_id | uuid | — | FOREIGN KEY → product_variants.id | Product in cart |
| quantity | int4 | 1 | NOT NULL | Quantity |
| created_at | timestampz | now() | — | Added timestamp |
| bundle_origin | text | — | — | Optional: 'brand_bundle', 'custom_bundle', or NULL |
| bundle_id | uuid | — | — | Optional: links to bundles.id, custom_bundles.id, or bundle_user_public.id |

17) Table: bundles

| Column | Type | Default | Constraints | Notes |
|-------------|------|-------------------|-------------|--|
| id | uuid | gen_random_uuid() | Primary key | Unique bundle ID |
| title | text | — | NOT NULL | Bundle name (e.g., "Birthday Bloom Box") |
| description | text | — | — | Short description |

| | | | | |
|------------|------------|-------|----------|--------------------------------|
| price | int4 | — | NOT NULL | Total price for bundle |
| img_url | text | — | — | Display image (Cloudinary URL) |
| is_active | bool | true | — | Visibility toggle |
| created_at | timestampz | now() | — | Created timestamp |
| updated_at | timestampz | now() | — | Updated timestamp |

18) Table: bundle_items

| Column | Type | Default | Constraints | Notes |
|--------------------|------------|-------------------|-----------------------------------|--|
| id | uuid | gen_random_uuid() | Primary key | Record ID |
| bundle_id | uuid | — | FK → bundles.id ON DELETE CASCADE | Parent bundle |
| product_id | uuid | — | FK → products.id | Base product |
| product_variant_id | uuid | — | FK → product_variants.id | Optional variant |
| quantity | int4 | 1 | — | Quantity of product |
| created_at | timestampz | now() | — | Timestamp |
| CHECK | — | — | — | (product_id IS NOT NULL OR product_variant_id IS NOT NULL) |

19) Table: bundles_custom

| Column | Type | Default | Constraints | Notes |
|-------------|------|-------------------|----------------|--|
| id | uuid | gen_random_uuid() | Primary key | User's custom bundle ID |
| user_id | uuid | — | FK → users.id | Bundle owner |
| order_id | uuid | — | FK → orders.id | Linked order after purchase |
| title | text | — | — | Optional name (e.g., "My Pink Surprise Box") |
| total_price | int4 | — | — | Bundle total at |

| | | | | |
|--------------|------------|-------|---|-----------------------------|
| | | | | checkout |
| is_published | bool | false | — | Whether user made it public |
| created_at | timestampz | now() | — | Created timestamp |

20) Table: bundle_custom_items

| Column | Type | Default | Constraints | Notes |
|--------------------|------------|-------------------|---|--|
| id | uuid | gen_random_uuid() | Primary key | Record ID |
| custom_bundle_id | uuid | — | FK → custom_bundles.id ON DELETE CASCADE | Parent bundle |
| product_id | uuid | — | FK → products.id | Base product |
| product_variant_id | uuid | — | FK → product_variants.id | Optional variant |
| quantity | int4 | 1 | — | Quantity of product |
| created_at | timestampz | now() | — | Timestamp |
| CHECK | — | — | — | (product_id IS NOT NULL OR product_variant_id IS NOT NULL) |

21) Table: bundle_user_public

| Column | Type | Default | Constraints | Notes |
|------------------|------|-------------------|------------------------|--|
| id | uuid | gen_random_uuid() | Primary key | Public bundle ID |
| custom_bundle_id | uuid | — | FK → custom_bundles.id | Source bundle |
| user_id | uuid | — | FK → users.id | Creator (public sharer) |
| title | text | — | NOT NULL | Public name (e.g., “Kajal’s Glam Box”) |
| username | text | — | — | Display name of creator |
| description | text | — | — | Optional description/message |

| | | | | |
|--------------|-------------|-------|---|----------------------------------|
| total_price | int4 | — | — | Price snapshot |
| upvote_count | int4 | 0 | — | Cached upvote counter |
| img_url | text | — | — | Thumbnail / representative image |
| created_at | timestamptz | now() | — | Created timestamp |

22) Table: bundle_user_items_public

| Column | Type | Default | Constraints | Notes |
|--------------------|-------------|-------------------|---|--|
| id | uuid | gen_random_uuid() | Primary key | Record ID |
| user_bundle_id | uuid | — | FK → bundle_user_public.id ON DELETE CASCADE | Public parent bundle |
| product_id | uuid | — | FK → products.id | Base product |
| product_variant_id | uuid | — | FK → product_variants.id | Optional variant |
| quantity | int4 | 1 | — | Quantity of product |
| created_at | timestamptz | now() | — | Timestamp |
| CHECK | — | — | — | (product_id IS NOT NULL OR product_variant_id IS NOT NULL) |

23) Table: bundle_user_upvotes

| Column | Type | Default | Constraints | Notes |
|-----------------|-------------|-------------------|----------------------------|---------------------|
| id | uuid | gen_random_uuid() | Primary key | Record ID |
| user_id | uuid | — | FK → users.id | Who liked |
| user_bundle_id | uuid | — | FK → bundle_user_public.id | Which public bundle |
| created_at | timestamptz | now() | — | Timestamp |
| UNIQUE(user_id, | — | — | — | Prevent duplicate |

user_bundle_id)

upvotes