**STRUCTURE FINALE DE LA BASE DE DONNÉES**

**1️⃣ Table des utilisateurs (users)**

Tous les utilisateurs (clients, vendeurs et administrateurs).

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| name | string | Nom complet |
| email | string (unique) | Adresse email |
| password | string | Mot de passe hashé |
| phone | string | Numéro de téléphone |
| role | enum (client, seller, admin) | Rôle de l’utilisateur |
| created\_at | timestamp | Date de création |
| updated\_at | timestamp | Date de mise à jour |

✅ **Permet de différencier les clients, vendeurs et administrateurs.**

**2️⃣ Table des vendeurs (sellers)**

Informations supplémentaires pour les vendeurs.

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| user\_id | UUID (FK) | Référence vers users |
| shop\_name | string | Nom de la boutique |
| description | text | Description de la boutique |
| logo\_url | string (URL) | Logo du vendeur |
| status | enum (pending, approved, rejected) | Statut du compte |
| created\_at | timestamp | Date d’enregistrement |

✅ **Tous les vendeurs sont d'abord des users, mais ils doivent être validés.**

**3️⃣ Table des catégories (categories)**

Organisation des produits par catégorie.

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| name | string | Nom de la catégorie |
| description | text | Description (optionnel) |

✅ **Permet d’organiser les produits.**

**4️⃣ Table des produits (products)**

Stocke tous les produits, qu’ils soient vendus par l’administration ou par des vendeurs.

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| seller\_id | UUID (FK) NULLABLE | Référence vers sellers (NULL si admin) |
| title | string | Nom du produit |
| description | text | Description |
| price | decimal | Prix |
| category\_id | UUID (FK) | Référence vers categories |
| stock | integer | Quantité disponible |
| image\_url | string (URL) | Image du produit |
| is\_admin\_product | boolean | **true** = commandable, **false** = vente libre |
| status | enum (available, out\_of\_stock, deleted) | État |
| created\_at | timestamp | Date de création |

✅ **Seuls les produits où is\_admin\_product = true peuvent être commandés.**

**5️⃣ Table des commandes (orders)**

Uniquement pour les produits **administratifs**.

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| client\_id | UUID (FK) | Référence vers users |
| total\_price | decimal | Prix total |
| status | enum (pending, paid, shipped, delivered, cancelled) | Statut de la commande |
| created\_at | timestamp | Date de création |

✅ **Seuls les produits is\_admin\_product = true peuvent être commandés ici.**

**6️⃣ Table des détails de commande (order\_items)**

Permet de commander plusieurs produits en une seule commande.

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| order\_id | UUID (FK) | Référence vers orders |
| product\_id | UUID (FK) | Référence vers products |
| quantity | integer | Quantité commandée |
| price | decimal | Prix unitaire |
| subtotal | decimal | quantity \* price |
| created\_at | timestamp | Date d'ajout |

✅ **Obligatoire pour gérer plusieurs produits dans une seule commande.**

**7️⃣ Table des paiements (payments)**

Gère les paiements des commandes **administratives**.

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| order\_id | UUID (FK) | Référence vers orders |
| user\_id | UUID (FK) | Référence vers users |
| amount | decimal | Montant payé |
| payment\_method | enum (card, paypal, mobile\_money, bank\_transfer) | Moyen de paiement |
| status | enum (pending, completed, failed) | Statut |
| transaction\_id | string | Référence transaction externe |
| created\_at | timestamp | Date du paiement |

✅ **Les paiements sont gérés uniquement par l’administration.**

**8️⃣ Table des messages (messages)**

Permet aux clients et vendeurs de discuter.

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| sender\_id | UUID (FK) | Référence vers users |
| receiver\_id | UUID (FK) | Référence vers users |
| product\_id | UUID (FK) NULLABLE | Produit concerné |
| content | text | Contenu du message |
| created\_at | timestamp | Date d'envoi |

✅ **Les clients peuvent contacter les vendeurs au sujet d'un produit.**

**9️⃣ Table des notifications (notifications)**

Pour informer les utilisateurs des événements importants.

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| user\_id | UUID (FK) | Référence vers users |
| message | text | Contenu de la notification |
| type | enum (order\_update, payment\_received, new\_message) | Type |
| is\_read | boolean | Lu ou non |
| created\_at | timestamp | Date d'envoi |

✅ **Permet d'informer les utilisateurs des mises à jour.**

**🔟 Table des favoris (favorites)**

Permet aux utilisateurs d’enregistrer leurs produits préférés.

| **Nom du champ** | **Type** | **Description** |
| --- | --- | --- |
| id | UUID (PK) | Identifiant unique |
| user\_id | UUID (FK) | Référence vers users |
| product\_id | UUID (FK) | Référence vers products |
| created\_at | timestamp | Date d'ajout |

✅ **Facilite l'expérience utilisateur.**

**📌 RÉSUMÉ FINAL**

✅ **Tout le monde peut vendre**, mais **seuls les produits de l'admin peuvent être commandés.**  
✅ **Les vendeurs publient des produits** et peuvent **discuter avec les clients**, mais sans vente en ligne.  
✅ **Les commandes, paiements et livraisons concernent uniquement l’administration.**  
✅ **Les favoris et notifications améliorent l’expérience utilisateur.**

💡 **Cette structure est complète et optimisée. Est-ce parfait pour vous ?**

**Vous avez dit :**

Est-ce y a-t-il moyen de relier dans un même programme de l'HTML, CSS, JavaScript et le langage C

**ChatGPT a dit :**

Oui, il est possible de **relier HTML, CSS, JavaScript et le langage C** dans un même programme, mais cela dépend du contexte d'utilisation et de l'architecture choisie. Voici plusieurs approches possibles :

**Services gratuits :**

1. **Création de compte client et vendeur**
2. **Consultation des produits et catégories**
3. **Recherche avancée avec filtres**
4. **Messagerie entre vendeurs et clients pour un produit spécifique**
5. **Avis et notes sur les vendeurs et produits**
6. **Wishlist (favoris)**
7. **Espace vendeur avec gestion de base (ajout/modification de produits, statistiques de base, etc.)**
8. **Notifications sur les nouvelles offres et promotions**
9. **Système de commentaires sous les produits**
10. **Support client par FAQ et chatbot**

**Services payants :**

1. **Mise en avant des produits (publicité sur la plateforme)**
2. **Statistiques avancées pour les vendeurs (analyse des ventes, audience, etc.)**
3. **Abonnement premium pour les vendeurs (meilleure visibilité, plus de fonctionnalités de gestion, etc.)**
4. **Vérification de compte vendeur pour une crédibilité accrue**
5. **Service de livraison intégré (si applicable, avec des frais supplémentaires)**
6. **Paiement sécurisé et intermédiaire (gestion des transactions avec une commission sur les ventes)**
7. **Assistance prioritaire pour les vendeurs premium**
8. **Personnalisation avancée de l’espace vendeur (branding, design, etc.)**

**Différenciation par rapport aux autres plateformes**

Pour vous démarquer, vous pouvez proposer des fonctionnalités uniques comme :

* **Messagerie enrichie** avec support d’images et vidéos pour mieux discuter des produits.
* **Négociation en temps réel**, permettant aux clients de faire des offres aux vendeurs.
* **Vidéos de démonstration des produits** directement sur les pages produits.
* **Livestream shopping**, où les vendeurs peuvent faire des présentations en direct de leurs produits.
* **Système de fidélité** basé sur des récompenses pour les achats ou interactions.

Est-ce que certaines de ces fonctionnalités vous intéressent particulièrement ?

### ****. Résumé des relations principales****

* **users (1) → (∞) products** : Un utilisateur (vendeur) peut avoir plusieurs produits.
* **users (1) → (∞) orders** : Un client peut passer plusieurs commandes.
* **orders (1) → (∞) order\_details** : Une commande peut contenir plusieurs produits.
* **orders (1) → (1) payments** : Chaque commande a un paiement unique.
* **users (∞) ↔ (∞) messages** : Les clients et vendeurs peuvent discuter.
* **users (∞) ↔ (∞) favorites** : Les clients peuvent enregistrer des produits en favoris.
* **products (1) → (∞) reviews** : Un produit peut recevoir plusieurs avis.
* **users (1) → (∞) notifications** : Chaque utilisateur peut recevoir des notifications.

. Prochaines étapes

Implémentez les contrôleurs et les routes pour gérer les fonctionnalités de votre site.

Testez les relations entre les tables en créant des données de test.

Ajoutez des validations et des politiques d'accès pour sécuriser votre application.