# Part 1

**Task 1 : Adding New Tables and Modifying Existing Tables**

1. **Description**

|  |  |  |
| --- | --- | --- |
| **New/**  **Old** | **Table** | **Description** |
| New | stores | Stores information about each store location that will be opened in the app |
| New | cart | Stores information about products added to the shopping cart by each customer |
| New | wishlist | Stores information about products that customers want to save as favorites or “wishlisted.” |
| New | payments | Stores information about available payment methods and customer transactions. |
| New | products\_reviews | Stores customer reviews for products, including ratings and optional review text. |
| Old | products | Modify, add new column store\_id |
| Old | orders | Modify, add new coloumn payment\_status (Paid, Pending, Failed) |

1. **Table Structure**
2. **Store Table**

|  |  |  |
| --- | --- | --- |
| **stores** | | |
| store\_id : int | Not null | PK |
| store\_name : varchar(100) |  |  |
| address: varchar(255) |  |  |

1. **Cart Table**

|  |  |  |
| --- | --- | --- |
| **cart** | | |
| card\_id : int | Not null | PK |
| customer\_id : int | Not null | FK |
| product\_id : int | Not null | FK |
| quantity : int | > 0 |  |

1. **Wishlist Table**

|  |  |  |
| --- | --- | --- |
| **wishlist** | | |
| wishlist\_id: int |  | PK |
| customer\_id : int | Not null | FK |
| product\_id : int | Not null | FK |

1. **Payments**

|  |  |  |
| --- | --- | --- |
| **payments** | | |
| payment\_id : int |  | PK |
| order\_id : int | Not null | FK |
| payment\_method: varchar(50) | Not null |  |
| amount: numeric | > 0 |  |
| payment\_date: timestamp |  |  |

1. **Product\_reviews**

|  |  |  |
| --- | --- | --- |
| **product\_reviews** | | |
| review\_id : int |  | PK |
| customer\_id: int | Not null | FK |
| product\_id: int | Not null | FK |
| rating: int | Between 1-5 |  |
| review\_text: text |  |  |
| review\_date: timestamp |  |  |

1. **DDL**
2. **Store table**

-- Membuat tabel stores

CREATE TABLE public.stores (

    store\_id int4 NOT NULL,

    store\_name varchar(100) NULL,

    address varchar(255) NULL,

    CONSTRAINT stores\_pk PRIMARY KEY (store\_id)

);

1. **Cart table**

-- Membuat tabel cart

CREATE TABLE public.cart (

    cart\_id int4 NOT NULL,

    customer\_id int4 NOT NULL,

    product\_id int4 NOT NULL,

    quantity int4 CHECK (quantity > 0),

    CONSTRAINT cart\_pk PRIMARY KEY (cart\_id),

    FOREIGN KEY (customer\_id) REFERENCES public.customers(customer\_id),

    FOREIGN KEY (product\_id) REFERENCES public.products(product\_id)

);

1. **Wishlist table**

-- Membuat tabel wishlist

CREATE TABLE public.wishlist (

    wishlist\_id int4 NOT NULL,

    customer\_id int4 NOT NULL,

    product\_id int4 NOT NULL,

    CONSTRAINT wishlist\_pk PRIMARY KEY (wishlist\_id),

    FOREIGN KEY (customer\_id) REFERENCES public.customers(customer\_id),

    FOREIGN KEY (product\_id) REFERENCES public.products(product\_id)

);

1. **Payment table**

-- Membuat tabel payments

CREATE TABLE public.payments (

    payment\_id int4 NOT NULL,

    order\_id int4 NOT NULL,

    payment\_method varchar(50) NOT NULL,

    amount numeric NOT NULL CHECK (amount > 0),

    payment\_date timestamp DEFAULT CURRENT\_TIMESTAMP,

    CONSTRAINT payments\_pk PRIMARY KEY (payment\_id),

    FOREIGN KEY (order\_id) REFERENCES public.orders(order\_id)

);

1. **Product\_reviews**

-- Membuat tabel product\_reviews

CREATE TABLE public.product\_reviews (

    review\_id int4 NOT NULL,

    customer\_id int4 NOT NULL,

    product\_id int4 NOT NULL,

    rating int4 CHECK (rating BETWEEN 1 AND 5),

    review\_text text,

    review\_date timestamp DEFAULT CURRENT\_TIMESTAMP,

    CONSTRAINT review\_pk PRIMARY KEY (review\_id),

    FOREIGN KEY (customer\_id) REFERENCES public.customers(customer\_id),

    FOREIGN KEY (product\_id) REFERENCES public.products(product\_id)

);

1. **Products table**

Adds a shop\_id column to link each product to a specific store.

-- Menambahkan kolom store\_id ke tabel products

ALTER TABLE public.products

ADD COLUMN store\_id int4,

ADD CONSTRAINT fk\_store FOREIGN KEY (store\_id) REFERENCES public.stores(store\_id);

1. **Order table**

Adding a payment\_status column to track whether an order is Paid, Pending, or Failed.

-- Menambahkan kolom payment\_status dan total\_amount ke tabel orders

ALTER TABLE public.orders

ADD COLUMN payment\_status varchar(20) DEFAULT 'Pending';

1. **NEW ERD**

