use webinfinity\_noon;

-- to search parent

SELECT category\_name

FROM parent\_category

WHERE sub\_category = (SELECT category\_id FROM parent\_category WHERE category\_name = 'Electronics');

SELECT category\_id

FROM parent\_category

WHERE sub\_category = (SELECT category\_id FROM parent\_category WHERE sub\_category = 0);

-- to search subcategory of parent

SELECT category\_name

FROM parent\_category

WHERE sub\_category = (SELECT category\_id FROM parent\_category WHERE category\_name = 'Mobiles & Accessories');

-- to search sub sub category of subcategory

SELECT category\_name

FROM parent\_category

WHERE sub\_category = (SELECT category\_id FROM parent\_category WHERE category\_name = 'iPhones');

-- to search sub sub category of sub sub category

SELECT category\_name

FROM parent\_category

WHERE sub\_sub\_category = (SELECT category\_id FROM parent\_category WHERE category\_name = 'iPhone 16');

SELECT

p1.category\_name AS main\_category,

p2.category\_name AS sub\_category,

p3.category\_name AS sub\_sub\_category,

p1.category\_id AS parent\_id

FROM

parent\_category p1

LEFT JOIN

parent\_category p2 ON p2.sub\_category = p1.category\_id

LEFT JOIN

parent\_category p3 ON p3.sub\_category = p2.category\_id

WHERE

p1.category\_name = "Women's Fashion";

-- show full hierarchy

WITH RECURSIVE category\_hierarchy AS (

-- Base Case (Starting Point)

SELECT

category\_id,

category\_name,

sub\_category,

sort\_order,

1 AS level,

CAST(category\_name AS CHAR(255)) AS full\_path

FROM parent\_category

WHERE sub\_category = 0 -- Root categories like Electronics, Women's Fashion

UNION ALL

-- Recursive Case (Jo sub-categories ko connect karega)

SELECT

c.category\_id,

c.category\_name,

c.sub\_category,

c.sort\_order,

ch.level + 1,

CONCAT(ch.full\_path, ' → ', c.category\_name) AS full\_path

FROM parent\_category c

JOIN category\_hierarchy ch

ON c.sub\_category = ch.category\_id

)

-- Final Output

SELECT

category\_id,

full\_path AS category\_hierarchy,

level,

sort\_order

FROM category\_hierarchy

ORDER BY full\_path,sort\_order;

------------------------------------------------------------------------------

create table status(

status\_id int primary key auto\_increment,

status\_name varchar(255) unique

);

insert into status(status\_name) values ('active'),('not active'),('out of stock'),('pending');

select \* from parent\_category;

CREATE TABLE parent\_category (

category\_id INT PRIMARY KEY AUTO\_INCREMENT,

category\_name VARCHAR(255) NOT NULL UNIQUE,

sub\_category INT,

sub\_sub\_category INT,

sort\_order TINYINT,

status\_id INT,

category\_url VARCHAR(255) UNIQUE,

parent\_category\_img\_url VARCHAR(255),

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE

);

INSERT INTO parent\_category (category\_id, category\_name, sub\_category, sub\_sub\_category, sort\_order, status\_id, category\_url)

VALUE

(NULL, 'Electronics', 0, 0, 1, 1, 'electronics');

INSERT INTO parent\_category (category\_id, category\_name, sub\_category, sub\_sub\_category, sort\_order, status\_id, category\_url)

VALUE

(NULL, 'Mobiles & Accessories', 1, 0, 1, 1, 'mobiles-&-accessories');

INSERT INTO parent\_category (category\_id, category\_name, sub\_category, sub\_sub\_category, sort\_order, status\_id, category\_url)

VALUE

(NULL, 'iPhones', 2, 0, 2, 1, 'iphones');

INSERT INTO parent\_category (category\_id, category\_name, sub\_category, sub\_sub\_category, sort\_order, status\_id, category\_url)

VALUE

(NULL, 'iPhone 16 Series', 3, 3, 1, 1, 'iphone-16-series');

INSERT INTO parent\_category (category\_id, category\_name, sub\_category, sub\_sub\_category, sort\_order, status\_id, category\_url)

VALUE

(NULL, 'iPhone 16 Pro', 4, 4, 2, 1, 'iphone-16-pro');

---------------------------------------------------------------------------------------------------------------------------

select \* from parent\_category;

INSERT INTO parent\_category (category\_id, category\_name, sub\_category, sub\_sub\_category, sort\_order, status\_id, category\_url)

VALUE

(NULL, "Women's Fashion", 0, 0, 3, 1, "women's-fashion");

INSERT INTO parent\_category (category\_id, category\_name, sub\_category, sub\_sub\_category, sort\_order, status\_id, category\_url)

VALUE

(NULL, 'Clothing', 6, 0, 2, 1, 'clothing');

INSERT INTO parent\_category (category\_id, category\_name, sub\_category, sub\_sub\_category, sort\_order, status\_id, category\_url)

VALUE

(NULL, 'Dresses', 7, 0, 2, 1, 'dresses');

INSERT INTO parent\_category (category\_id, category\_name, sub\_category, sub\_sub\_category, sort\_order, status\_id, category\_url)

VALUE

(NULL, 'Work Dresses', 8, 8, 4, 1, 'work-dresses');

-- Brands Table

CREATE TABLE brands (

brand\_id INT PRIMARY KEY AUTO\_INCREMENT,

brand\_name VARCHAR(255) NOT NULL UNIQUE,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE

);

INSERT INTO brands (brand\_name,status\_id) VALUES

('Apple',1),

('Threadz By Ajooni',1);

-- Products Table

CREATE TABLE products (

product\_id INT PRIMARY KEY AUTO\_INCREMENT,

product\_name VARCHAR(255) NOT NULL,

seo\_url VARCHAR(255) NOT NULL,

description TEXT,

category\_id INT,

brand\_id INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (category\_id) REFERENCES parent\_category(category\_id) ON DELETE CASCADE,

FOREIGN KEY (brand\_id) REFERENCES brands(brand\_id) ON DELETE CASCADE

);

INSERT INTO products (product\_name, seo\_url, description, category\_id, brand\_id, status\_id) VALUES

('iPhone 16 Pro 512GB White Titanium 5G', 'iphone-16-pro-512gb-white-titanium-5g', 'Middle East Version with FaceTime', 5, 1, 1),

('Button Detail Belted Dress', 'button-detail-belted-dress', 'Elegant white dress with detailed belt design', 9, 2, 1);

-- Pricing Table

CREATE TABLE pricing (

pricing\_id INT PRIMARY KEY AUTO\_INCREMENT,

product\_id INT,

price DECIMAL(10,2),

original\_price DECIMAL(10,2),

discount DECIMAL(10,2),

shipping\_fee DECIMAL(10,2),

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

INSERT INTO pricing (product\_id, price, original\_price, discount, shipping\_fee, status\_id) VALUES

(1, 5015.00, 5549.00, 534.00, 0.00, 1),

(2, 104.00, 709.00, 605.00, 0.00, 1);

-- Images Table

CREATE TABLE images (

image\_id INT PRIMARY KEY AUTO\_INCREMENT,

product\_id INT,

image\_url VARCHAR(255),

sort\_order INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

INSERT INTO images (product\_id, image\_url, sort\_order, status\_id) VALUES

(1, 'iphone\_image1.jpg', 1, 1),

(1, 'iphone\_image2.jpg', 2, 1),

(2, 'dress\_image1.jpg', 1, 1),

(2, 'dress\_image2.jpg', 2, 1);

-- Reviews Table

CREATE TABLE reviews (

review\_id INT PRIMARY KEY AUTO\_INCREMENT,

product\_id INT,

review\_count INT,

review\_rating DECIMAL(3,1),

review\_ranking VARCHAR(255),

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

INSERT INTO reviews (product\_id, review\_count, review\_rating, review\_ranking, status\_id) VALUES

(1, 4, 5.0, 'Top Rated', 1),

(2, 5, 4.0, 'Best Seller', 1);

CREATE TABLE customer\_reviews (

review\_id INT PRIMARY KEY AUTO\_INCREMENT,

product\_id INT,

user\_id INT,

rating DECIMAL(3,1) CHECK (rating BETWEEN 1 AND 5),

review\_text TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

INSERT INTO customer\_reviews (product\_id, user\_id, rating, review\_text, status\_id) VALUES

(1, 101, 5.0, 'Amazing product!', 1),

(1, 102, 4.5, 'Very good, but can be improved.', 1),

(1, 103, 5.0, 'Loved it!', 1),

(1, 104, 4.0, 'Good product!', 1),

(2, 105, 4.0, 'Nice quality!', 1),

(2, 106, 3.5, 'Could be better.', 1),

(2, 107, 4.5, 'Good performance!', 1),

(2, 108, 4.0, 'Worth it.', 1),

(2, 109, 4.0, 'Solid purchase.', 1);

-- Ranking Table

CREATE TABLE ranking (

ranking\_id INT PRIMARY KEY AUTO\_INCREMENT,

product\_id INT,

ranking\_value VARCHAR(255),

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

INSERT INTO ranking (product\_id, ranking\_value, status\_id) VALUES

(1, 'Top 5', 1),

(2, 'Top 10', 1);

-- Tags Table

CREATE TABLE tags (

tag\_id INT PRIMARY KEY AUTO\_INCREMENT,

tag\_name VARCHAR(255) UNIQUE NOT NULL,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE

);

INSERT INTO tags (tag\_name,status\_id) VALUES ('Best Seller',1),('Top Rated',1);

-- Product Tags Relationship Table

CREATE TABLE product\_tags (

product\_id INT,

tag\_id INT,

PRIMARY KEY (product\_id, tag\_id),

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE,

FOREIGN KEY (tag\_id) REFERENCES tags(tag\_id) ON DELETE CASCADE

);

INSERT INTO product\_tags (product\_id, tag\_id, status\_id) VALUES

(1, 1, 1),

(2, 2, 1);

CREATE TABLE delivery\_tag (

delivery\_tag\_id INT PRIMARY KEY AUTO\_INCREMENT,

delivery\_tag\_name VARCHAR(255),

delivery\_day\_name VARCHAR(255),

category\_id INT,

product\_id INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (category\_id) REFERENCES parent\_category(category\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

insert into delivery\_tag (delivery\_tag\_name,delivery\_day\_name,product\_id,status\_id)

values ('express','Get it Tomorrow',1,1),('express','Get it Tomorrow',2,1);

create table moving\_tag (

moving\_tag\_id int primary key auto\_increment,

moving\_tag\_icon varchar(255),

moving\_tag\_name varchar(255) not null,

category\_id INT,

product\_id INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (category\_id) REFERENCES parent\_category(category\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

insert into moving\_tag (moving\_tag\_icon,moving\_tag\_name,category\_id,product\_id,status\_id)

values ('https://cdn-icons-png.flaticon.com/128/5637/5637217.png','Free Delivery',5,1,1),('https://cdn-icons-png.flaticon.com/128/11712/11712454.png','Selling out fast',9,2,1);

-- Sellers Table

CREATE TABLE sellers (

seller\_id INT PRIMARY KEY AUTO\_INCREMENT,

seller\_name VARCHAR(255) NOT NULL,

rating DECIMAL(3, 1),

positive\_rating INT,

partner\_since DATE,

item\_as\_described INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE

);

INSERT INTO sellers (seller\_name, rating, positive\_rating, partner\_since,item\_as\_described,status\_id) VALUES

('Apple Deals', 4.8, 98, '2025-03-31', 100.00,1),

('Noon Fashion Group', 4.2, 74, '2025-03-31', 80.00,1);

-- Product-Sellers Relationship Table

CREATE TABLE product\_sellers (

product\_id INT,

seller\_id INT,

PRIMARY KEY (product\_id, seller\_id),

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE,

FOREIGN KEY (seller\_id) REFERENCES sellers(seller\_id) ON DELETE CASCADE

);

INSERT INTO product\_sellers (product\_id, seller\_id, status\_id) VALUES

(1, 1, 1),

(2, 2, 1);

-- Banks Table

CREATE TABLE banks (

bank\_id INT PRIMARY KEY AUTO\_INCREMENT,

bank\_name VARCHAR(255) NOT NULL UNIQUE,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE

);

INSERT INTO banks (bank\_name,status\_id) VALUES

('Mashreq Bank',1),

('ADCB Bank',1);

-- Offers Table

CREATE TABLE offers (

offer\_id INT PRIMARY KEY AUTO\_INCREMENT,

bank\_id INT,

category\_id INT,

product\_id INT,

offer\_details VARCHAR(255),

discount\_percentage DECIMAL(5,2),

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (bank\_id) REFERENCES banks(bank\_id) ON DELETE CASCADE,

FOREIGN KEY (category\_id) REFERENCES parent\_category(category\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

-- (Linking Products with Banks)

INSERT INTO offers (bank\_id, product\_id, offer\_details, discount\_percentage, status\_id) VALUES

(1, 1, 'Earn 5% cashback with the Mashreq noon Credit Card', 5.00, 1),

(2, 2, 'Earn 15% cashback with the Mashreq noon Credit Card', 15.00, 1);

-- Promotions Table

CREATE TABLE promotions (

promo\_id INT PRIMARY KEY AUTO\_INCREMENT,

promo\_code VARCHAR(50) NOT NULL UNIQUE,

promo\_details VARCHAR(255),

discount\_percentage DECIMAL(5,2),

valid\_from DATE,

valid\_till DATE,

category\_id INT,

product\_id INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (category\_id) REFERENCES parent\_category(category\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

INSERT INTO promotions (promo\_code, promo\_details, discount\_percentage, valid\_from, valid\_till, product\_id, status\_id)

VALUES

('SAVEBIG', 'Extra 10% off!', 10.00, '2025-04-01', '2025-05-30', 1, 1),

('ADCBONE', 'Extra 20% off!', 20.00, '2025-04-01', '2025-05-30', 2, 1);

create table badgeList(

badgeList\_id int primary key auto\_increment,

badgeList\_icon varchar(255),

badgeList\_name varchar(255),

category\_id INT,

product\_id INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (category\_id) REFERENCES parent\_category(category\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

insert into badgeList(badgeList\_icon,badgeList\_name,category\_id,product\_id,status\_id)

values ('https://cdn-icons-png.flaticon.com/128/3954/3954386.png','High Rated Seller',1,1,1),('https://cdn-icons-png.flaticon.com/128/5359/5359689.png','Cash on Delivery',1,1,1),('https://cdn-icons-png.flaticon.com/128/9578/9578794.png','Secure Transaction',1,1,1);

create table variant\_colour(

variant\_colour\_id int primary key auto\_increment,

variant\_colour\_name varchar(255),

variant\_colour\_img\_url varchar(255),

product\_id INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

insert into variant\_colour(variant\_colour\_name,product\_id,status\_id)

values ('Desert Titanium',1,1),('Black Titanium',1,1),('Natural Titanium',1,1),('White Titanium',1,1);

create table variant\_memory(

variant\_memory\_id int primary key auto\_increment,

variant\_memory\_name varchar(255),

product\_id INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

insert into variant\_memory(variant\_memory\_name,product\_id,status\_id)

values ('1 TB',1,1),('128 GB',1,1),('256 GB',1,1),('512 GB',1,1);

create table variant\_version(

variant\_version\_id int primary key auto\_increment,

variant\_version\_name varchar(255),

product\_id INT,

status\_id INT,

FOREIGN KEY (status\_id) REFERENCES status(status\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

insert into variant\_version(variant\_version\_name,product\_id,status\_id)

values ('International Version',1,1),('Middle East Version',1,1),('USA Version',1,1);

next file

/\*\* CREATE VIEW product\_view AS

SELECT

p.product\_id,

p.product\_name,

p.seo\_url,

p.description,

c1.category\_name AS category\_name,

b.brand\_name,

pr.price,

pr.original\_price,

pr.discount,

pr.shipping\_fee,

i.image\_url,

r.review\_count,

r.review\_rating,

r.review\_ranking,

rk.ranking\_value,

t.tag\_name,

dt.delivery\_tag\_name,

dt.delivery\_day\_name,

mt.moving\_tag\_icon,

mt.moving\_tag\_name,

s.seller\_name,

s.rating AS seller\_rating,

s.positive\_rating,

s.partner\_since,

s.item\_as\_described,

bn.bank\_name,

o.offer\_details,

o.discount\_percentage AS offer\_discount,

promo.promo\_code,

promo.promo\_details,

promo.discount\_percentage AS promo\_discount,

promo.valid\_from,

promo.valid\_till,

bi.badgeList\_icon,

bi.badgeList\_name,

v.variant\_colour\_name,

v.variant\_colour\_img\_url,

vm.variant\_memory\_name,

vv.variant\_version\_name,

st.status\_name

FROM products p

LEFT JOIN parent\_category c1 ON p.category\_id = c1.category\_id

LEFT JOIN brands b ON p.brand\_id = b.brand\_id

LEFT JOIN pricing pr ON p.product\_id = pr.product\_id

LEFT JOIN images i ON p.product\_id = i.product\_id

LEFT JOIN reviews r ON p.product\_id = r.product\_id

LEFT JOIN ranking rk ON p.product\_id = rk.product\_id

LEFT JOIN product\_tags pt ON p.product\_id = pt.product\_id

LEFT JOIN tags t ON pt.tag\_id = t.tag\_id

LEFT JOIN delivery\_tag dt ON p.product\_id = dt.product\_id

LEFT JOIN moving\_tag mt ON p.product\_id = mt.product\_id

LEFT JOIN product\_sellers ps ON p.product\_id = ps.product\_id

LEFT JOIN sellers s ON ps.seller\_id = s.seller\_id

LEFT JOIN offers o ON p.product\_id = o.product\_id

LEFT JOIN banks bn ON o.bank\_id = bn.bank\_id

LEFT JOIN promotions promo ON p.product\_id = promo.product\_id

LEFT JOIN badgeList bi ON p.product\_id = bi.product\_id

LEFT JOIN variant\_colour v ON p.product\_id = v.product\_id

LEFT JOIN variant\_memory vm ON p.product\_id = vm.product\_id

LEFT JOIN variant\_version vv ON p.product\_id = vv.product\_id

LEFT JOIN status st ON p.status\_id = st.status\_id;

select \* from product\_view;

SELECT product\_id, COUNT(\*) FROM product\_view GROUP BY product\_id;

SELECT product\_id, COUNT(\*) FROM products GROUP BY status\_id;

DROP VIEW product\_view;

select \* from products;

SELECT p.product\_id, p.product\_name, s.status\_id, s.status\_name

FROM products p

LEFT JOIN status s ON p.status\_id = s.status\_id

WHERE p.product\_id = 1; \*\*/

/\*\* create or replce view dated 020042025 \*\*/

CREATE OR REPLACE VIEW product\_view AS

WITH

first\_image AS (

SELECT \* FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY product\_id ORDER BY image\_url) AS rn

FROM images

) img WHERE rn = 1

),

first\_tag AS (

SELECT pt.product\_id, t.tag\_name FROM product\_tags pt

JOIN tags t ON pt.tag\_id = t.tag\_id

GROUP BY pt.product\_id

),

first\_delivery\_tag AS (

SELECT \* FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY product\_id ORDER BY delivery\_tag\_name) AS rn

FROM delivery\_tag

) dt WHERE rn = 1

),

first\_moving\_tag AS (

SELECT \* FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY product\_id ORDER BY moving\_tag\_name) AS rn

FROM moving\_tag

) mt WHERE rn = 1

),

first\_seller AS (

SELECT \* FROM (

SELECT ps.product\_id, s.\*

FROM product\_sellers ps

JOIN sellers s ON ps.seller\_id = s.seller\_id

WHERE ps.seller\_id IS NOT NULL

) seller

GROUP BY seller.product\_id

),

first\_offer AS (

SELECT \* FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY product\_id ORDER BY discount\_percentage DESC) AS rn

FROM offers

) o WHERE rn = 1

),

first\_promo AS (

SELECT \* FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY product\_id ORDER BY discount\_percentage DESC) AS rn

FROM promotions

) promo WHERE rn = 1

),

first\_badge AS (

SELECT \* FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY product\_id ORDER BY badgeList\_name) AS rn

FROM badgeList

) bi WHERE rn = 1

),

first\_variant\_colour AS (

SELECT \* FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY product\_id ORDER BY variant\_colour\_name) AS rn

FROM variant\_colour

) v WHERE rn = 1

),

first\_variant\_memory AS (

SELECT \* FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY product\_id ORDER BY variant\_memory\_name) AS rn

FROM variant\_memory

) vm WHERE rn = 1

),

first\_variant\_version AS (

SELECT \* FROM (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY product\_id ORDER BY variant\_version\_name) AS rn

FROM variant\_version

) vv WHERE rn = 1

)

SELECT

p.product\_id,

p.product\_name,

p.seo\_url,

p.description,

c1.category\_name,

b.brand\_name,

pr.price,

pr.original\_price,

pr.discount,

pr.shipping\_fee,

fi.image\_url,

r.review\_count,

r.review\_rating,

r.review\_ranking,

rk.ranking\_value,

ft.tag\_name,

fdt.delivery\_tag\_name,

fdt.delivery\_day\_name,

fmt.moving\_tag\_icon,

fmt.moving\_tag\_name,

fs.seller\_name,

fs.rating AS seller\_rating,

fs.positive\_rating,

fs.partner\_since,

fs.item\_as\_described,

bn.bank\_name,

fo.offer\_details,

fo.discount\_percentage AS offer\_discount,

fp.promo\_code,

fp.promo\_details,

fp.discount\_percentage AS promo\_discount,

fp.valid\_from,

fp.valid\_till,

fb.badgeList\_icon,

fb.badgeList\_name,

fvc.variant\_colour\_name,

fvc.variant\_colour\_img\_url,

fvm.variant\_memory\_name,

fvv.variant\_version\_name,

st.status\_name

FROM products p

LEFT JOIN parent\_category c1 ON p.category\_id = c1.category\_id

LEFT JOIN brands b ON p.brand\_id = b.brand\_id

LEFT JOIN pricing pr ON p.product\_id = pr.product\_id

LEFT JOIN first\_image fi ON p.product\_id = fi.product\_id

LEFT JOIN reviews r ON p.product\_id = r.product\_id

LEFT JOIN ranking rk ON p.product\_id = rk.product\_id

LEFT JOIN first\_tag ft ON p.product\_id = ft.product\_id

LEFT JOIN first\_delivery\_tag fdt ON p.product\_id = fdt.product\_id

LEFT JOIN first\_moving\_tag fmt ON p.product\_id = fmt.product\_id

LEFT JOIN first\_seller fs ON p.product\_id = fs.product\_id

LEFT JOIN first\_offer fo ON p.product\_id = fo.product\_id

LEFT JOIN banks bn ON fo.bank\_id = bn.bank\_id

LEFT JOIN first\_promo fp ON p.product\_id = fp.product\_id

LEFT JOIN first\_badge fb ON p.product\_id = fb.product\_id

LEFT JOIN first\_variant\_colour fvc ON p.product\_id = fvc.product\_id

LEFT JOIN first\_variant\_memory fvm ON p.product\_id = fvm.product\_id

LEFT JOIN first\_variant\_version fvv ON p.product\_id = fvv.product\_id

LEFT JOIN status st ON p.status\_id = st.status\_id;

select \* from product\_view;

next file  
  
use webinfinity\_noon; -- for header section

create table header(

header\_id int primary key auto\_increment,

header\_name varchar(255) not null,

header\_icon varchar(255),

header\_url varchar(500),

sort\_order TINYINT,

created\_at timestamp default current\_timestamp,

status\_id int,

foreign key (status\_id) references status(status\_id) on delete cascade

);

insert into header(header\_name,header\_icon,header\_url,sort\_order,status\_id) values

('noon',null,'logo',1,1),

('Delivery Location','uae\_flag.png','delivery-location',2,1),

('Search',null,'search',3,1),

('Language',null,'language',4,1),

('Log in','https://cdn-icons-png.flaticon.com/128/4929/4929629.png','log-in',5,1),

('Wishlist','wishlist\_icon.png','wishlist',6,1),

('Cart','cart\_icon.png','cart',7,1);

CREATE TABLE header\_settings (

id INT PRIMARY KEY CHECK (id = 1),

background\_color VARCHAR(7) DEFAULT '#ffcc00',

text\_color VARCHAR(7) DEFAULT '#000000',

hover\_color VARCHAR(7) DEFAULT '#f0f0f0',

search\_placeholder VARCHAR(255) DEFAULT 'What are you looking for?',

delivery\_location VARCHAR(255) DEFAULT 'Dubai',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

INSERT INTO header\_settings (id, background\_color, text\_color, hover\_color, search\_placeholder, delivery\_location) VALUES

(1, '#ffcc00', '#000000', '#f0f0f0', 'What are you looking for?', 'Dubai');

Next file

-- phpMyAdmin SQL Dump

-- version 5.0.4

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: Oct 25, 2022 at 12:04 PM

-- Server version: 10.4.17-MariaDB

-- PHP Version: 8.0.2

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `one-place`

--

-- --------------------------------------------------------

--

-- Table structure for table `accounts`

--

use webinfinity\_noon;

CREATE TABLE `accounts` (

`id` int(11) NOT NULL,

`username` text DEFAULT NULL,

`password` text DEFAULT NULL,

`role` text DEFAULT NULL,

`dateCreated` date DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

select \* from accounts;

--

-- Dumping data for table `accounts`

--

INSERT INTO `accounts` (`id`, `username`, `password`, `role`, `dateCreated`) VALUES

(1, 'admin', '$2b$10$SKf41MV0GAtdNZ3yRa0UBeCKWi9j1adh1KKHk6cI9JxrMwAHTa5SS', 'admin', '2025-04-03');

SET SQL\_SAFE\_UPDATES = 0;

UPDATE accounts

SET password = '$2b$10$SKf41MV0GAtdNZ3yRa0UBeCKWi9j1adh1KKHk6cI9JxrMwAHTa5SS'

WHERE username = 'admin';

-- --------------------------------------------------------

--

-- Table structure for table `calendar`

--

CREATE TABLE `calendar` (

`id` int(10) NOT NULL,

`title` text DEFAULT NULL,

`details` text DEFAULT NULL,

`deadlineDate` date DEFAULT NULL,

`hours` text DEFAULT NULL,

`addDate` date DEFAULT NULL,

`worker` text DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--

-- Table structure for table `clients`

--

CREATE TABLE `clients` (

`client\_id` int(15) NOT NULL,

`client` text DEFAULT NULL,

`clientDetails` text DEFAULT NULL,

`phone` text DEFAULT NULL,

`country` text DEFAULT NULL,

`street` text DEFAULT NULL,

`city` text DEFAULT NULL,

`postalCode` text DEFAULT NULL,

`clientDateCreated` date DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--

-- Table structure for table `orders`

--

CREATE TABLE `orders` (

`id` int(15) NOT NULL,

`client\_id` int(15) NOT NULL,

`date` date DEFAULT NULL,

`price` float DEFAULT NULL,

`status` text DEFAULT NULL,

`workerName` text DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--

-- Table structure for table `products`

--

CREATE TABLE `products` (

`id` int(15) NOT NULL,

`order\_id` int(15) DEFAULT NULL,

`productName` text DEFAULT NULL,

`amount` int(5) DEFAULT NULL,

`itemPrice` float DEFAULT NULL,

`totalPrice` float DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Indexes for dumped tables

--

--

-- Indexes for table `accounts`

--

ALTER TABLE `accounts`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `calendar`

--

ALTER TABLE `calendar`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `clients`

--

ALTER TABLE `clients`

ADD PRIMARY KEY (`client\_id`);

--

-- Indexes for table `orders`

--

ALTER TABLE `orders`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `products`

--

ALTER TABLE `products`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `accounts`

--

ALTER TABLE `accounts`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT for table `calendar`

--

ALTER TABLE `calendar`

MODIFY `id` int(10) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT for table `clients`

--

ALTER TABLE `clients`

MODIFY `client\_id` int(15) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT for table `orders`

--

ALTER TABLE `orders`

MODIFY `id` int(15) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT for table `products`

--

ALTER TABLE `products`

MODIFY `id` int(15) NOT NULL AUTO\_INCREMENT;

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;