



DATABASE PROJECT INVENTORY

CONTENT

01

OUR TEAM

02

PROGRAM INTRODUCTION

03

ERD

04

USE CASE DIAGRAM

05

DATABASE SCHEMA

06

POPULATING DATA

07

QUERY

OUR TEAM



Al Azhar Rizqi
Rifa'i Firdaus



Ichsan Ali
Darmawan



Sri Kresna
Maha Dewa



Steven Christian
Susanto

INTRO



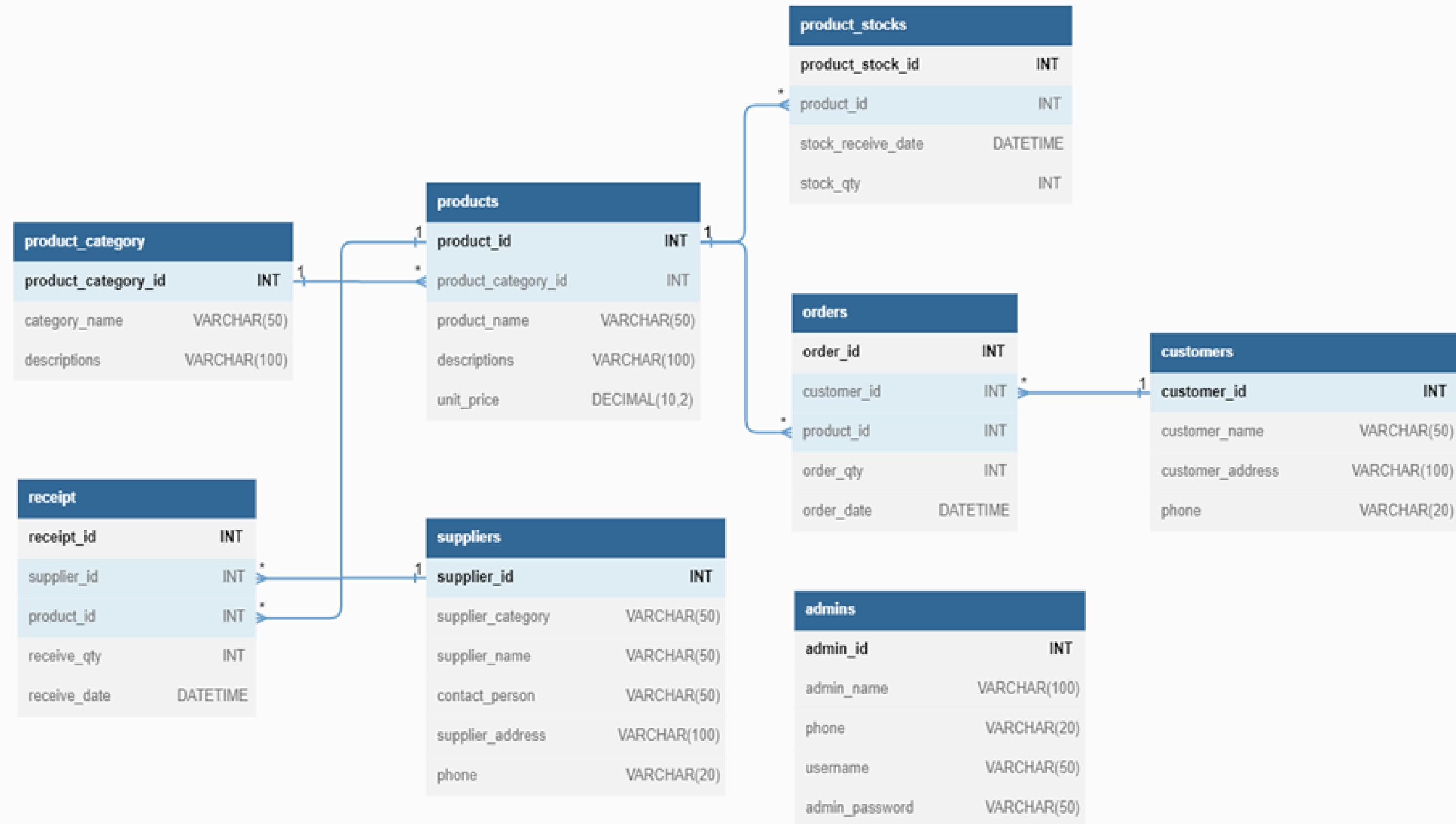
Streamlined Efficiency: Our system offers barcode scanning, automated replenishment, sales forecasting, reporting, and analytics to optimize inventory management processes, save time, and reduce costs.



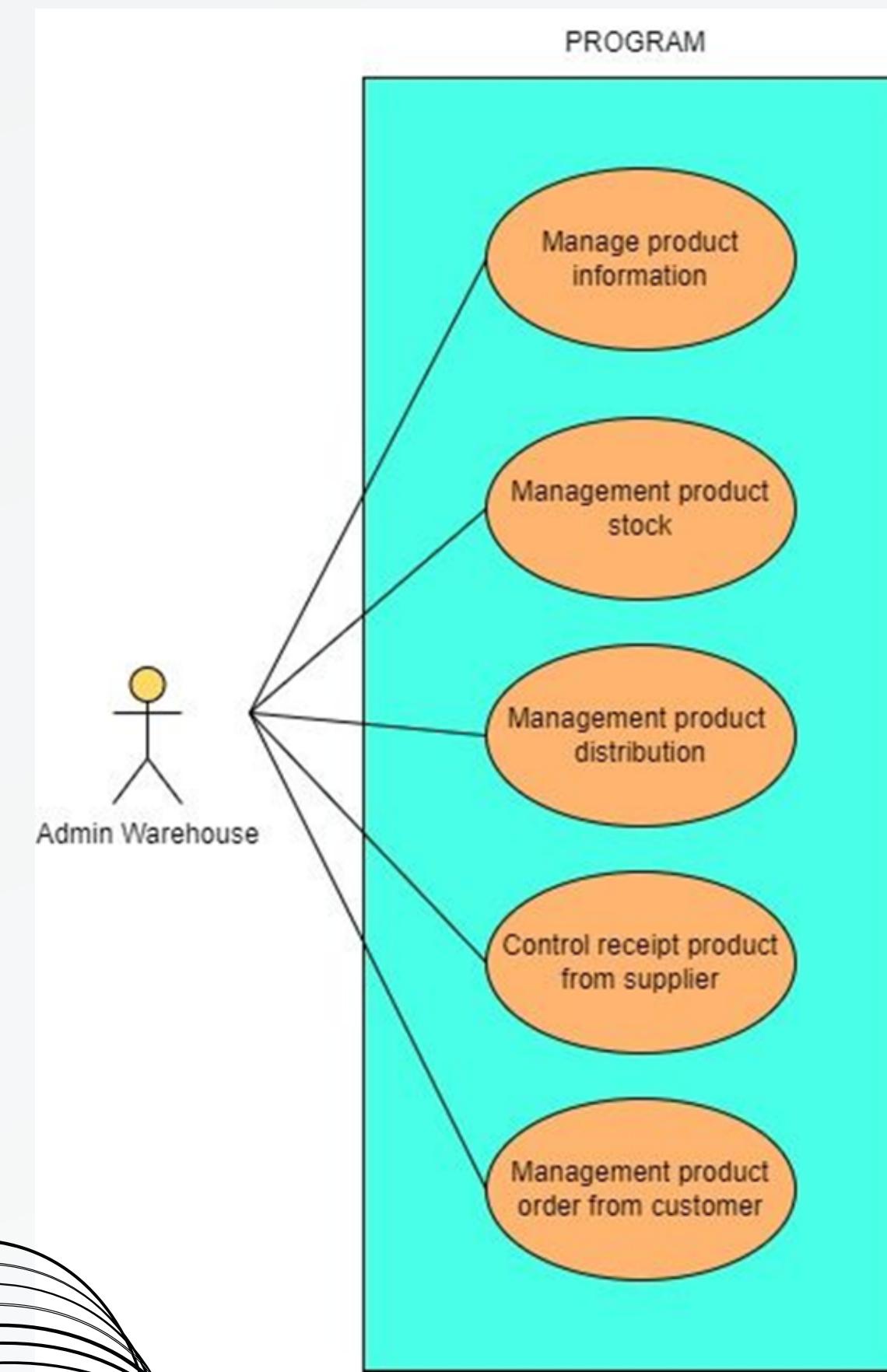
Enhanced Productivity and Customer Satisfaction: Real-time inventory tracking prevents waste and stockouts, improving customer satisfaction. The system ensures compliance, enabling informed decisions and prompt response to customer demands.

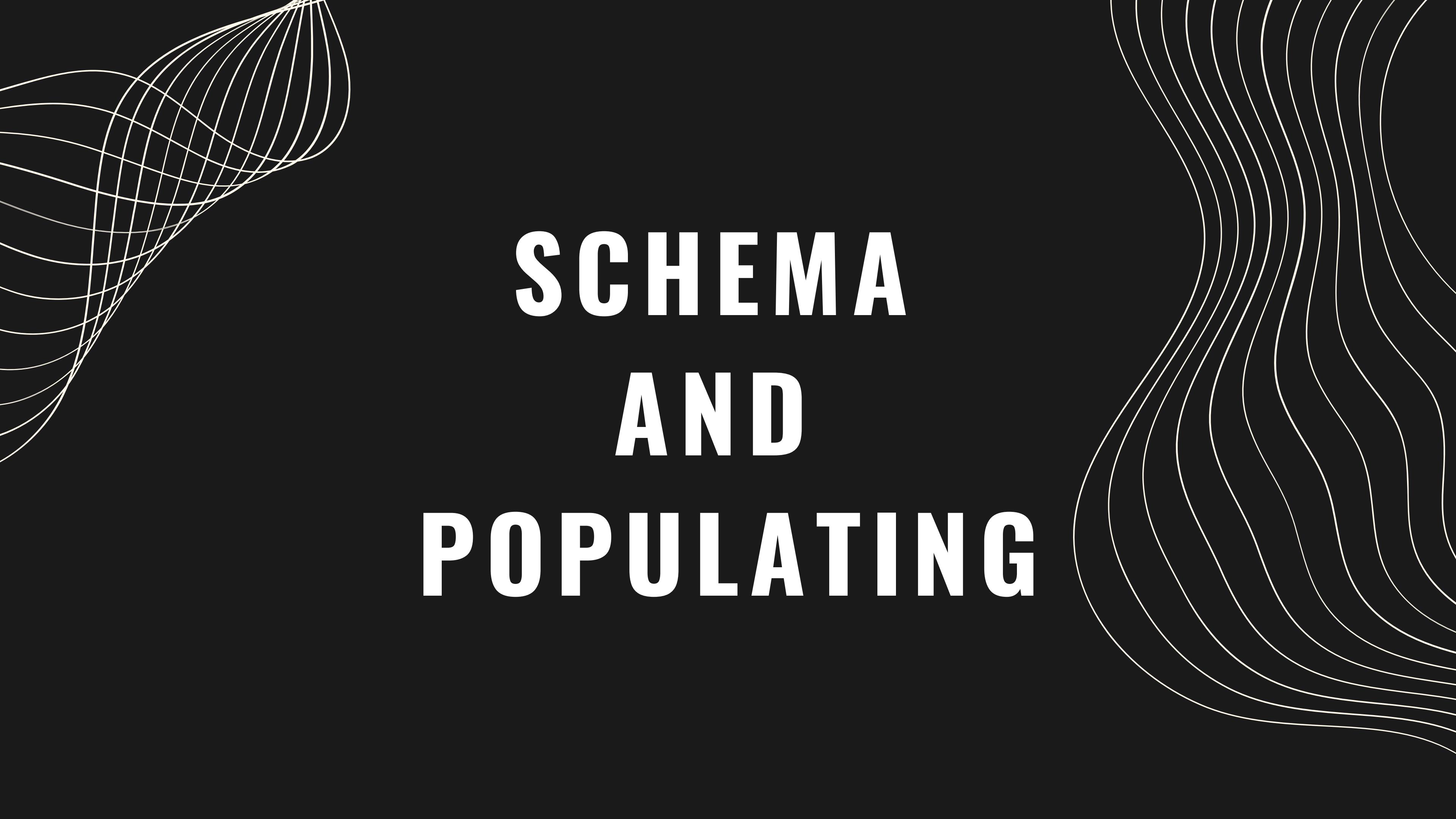


ERD



USE CASE DIAGRAM





S C H E M A A N D P O P U L A T I N G

ADMINS

```
CREATE TABLE admins (
    admin_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    admin_name VARCHAR(100) NOT NULL,
    phone VARCHAR(20) NOT NULL,
    username VARCHAR(50) NOT NULL,
    admin_password VARCHAR(50) NOT NULL -- encrypt using bcrypt and utf8
);
```

```
INSERT INTO admins (admin_id, admin_name, phone, username, admin_password) VALUES
-- real password for admin1 is "password1" it's has been encrypt using bcrypt gensalt and hash with utf8
(1, 'Admin1', '123-456-7890', 'admin1', '$2b$12$j5abRn3PCE0fYItzgYOaVec0tr5Ukq8Dw.vGOcRZNS6Dd09j6/2Lq'),
-- real password for admin2 is "password2" it's has been encrypt using bcrypt gensalt and hash with utf8
(2, 'Admin2', '987-654-3210', 'admin2', '$2b$12$GdR9iSPq4DqU54.AuU1hUuOvAyM3i58IMhvoxCrLB76v.3tnk.RNe');
```

	admin_id	admin_name	phone	username	admin_password
	1	Admin1	123-456-7890	admin1	\$2b\$12\$j5abRn3PCE0fYItzgYOaVec0tr5Ukq8Dw.vGOcRZNS6
▶ *	2	Admin2	987-654-3210	admin2	\$2b\$12\$GdR9iSPq4DqU54.AuU1hUuOvAyM3i58IMhvoxCrLB76

PRODUCTS

```
CREATE TABLE products (
    product_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    product_category_id INT NOT NULL,
    product_name VARCHAR(50) NOT NULL,
    descriptions VARCHAR(100) NOT NULL,
    unit_price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (product_category_id) REFERENCES product_category(product_category_id)
);
```

```
-- INSERT INTO products
INSERT INTO products (product_category_id, product_name, descriptions, unit_price) VALUES
(1, 'Smartphone', 'High-end smartphone with advanced features', 999.99),
(1, 'Laptop', 'Powerful laptop for professional use', 1499.99),
(2, 'T-Shirt', 'Casual cotton t-shirt', 19.99),
(2, 'Jeans', 'Denim jeans for men', 49.99),
(3, 'Blender', 'Kitchen blender for smoothies and food preparation', 39.99),
(4, 'Shampoo', 'Moisturizing shampoo for all hair types', 9.99),
(4, 'Face Cream', 'Anti-aging face cream with SPF 50', 29.99),
(5, 'The Great Gatsby', 'Classic novel by F. Scott Fitzgerald', 12.99),
(5, 'To Kill a Mockingbird', 'Pulitzer Prize-winning novel by Harper Lee', 10.99),
(6, 'Yoga Mat', 'Non-slip yoga mat for comfortable workouts', 24.99),
(6, 'Dumbbell Set', 'Set of adjustable dumbbells for strength training', 59.99);
```

PRODUCTS

	product_id	product_category_id	product_name	descriptions	unit_price
▶	1	1	Smartphone	High-end smartphone with advanced features	999.99
	2	1	Laptop	Powerful laptop for professional use	1499.99
	3	2	T-Shirt	Casual cotton t-shirt	19.99
	4	2	Jeans	Denim jeans for men	49.99
	5	3	Blender	Kitchen blender for smoothies and food prepara...	39.99
	6	4	Shampoo	Moisturizing shampoo for all hair types	9.99
	7	4	Face Cream	Anti-aging face cream with SPF 50	29.99
	8	5	The Great Gatsby	Classic novel by F. Scott Fitzgerald	12.99
	9	5	To Kill a Mockingbird	Pulitzer Prize-winning novel by Harper Lee	10.99
	10	6	Yoga Mat	Non-slip yoga mat for comfortable workouts	24.99
	11	6	Dumbbell Set	Set of adjustable dumbbells for strength training	59.99
●	NULl	NULl	NULl	NULl	NULl

PRODUCT_CATEGORY

```
CREATE TABLE product_category (
    product_category_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    category_name VARCHAR(50) NOT NULL,
    descriptions VARCHAR(100) NOT NULL
);
```

```
-- INSERT INTO product_category
INSERT INTO product_category (category_name, descriptions) VALUES
('Electronics', 'Electronic devices and components'),
('Clothing', 'Clothing and apparel'),
('Home and Kitchen', 'Home appliances and kitchenware'),
('Beauty', 'Beauty and personal care'),
('Books', 'Books and literature'),
('Sports', 'Sports and fitness');
```

	product_category_id	category_name	descriptions
▶	1	Electronics	Electronic devices and components
	2	Clothing	Clothing and apparel
	3	Home and Kitchen	Home appliances and kitchenware
	4	Beauty	Beauty and personal care
	5	Books	Books and literature
	6	Sports	Sports and fitness
*	NONE	NONE	NONE

PRODUCT_STOCKS

```
CREATE TABLE product_stocks (
    product_stock_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    product_id INT NOT NULL,
    stock_receive_date DATETIME NOT NULL,
    stock_qty INT NOT NULL,
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

```
-- INSERT INTO product_stocks
INSERT INTO product_stocks (product_id, stock_receive_date, stock_qty) VALUES
(1, '2023-06-01 09:00:00', 10),
(1, '2023-06-02 14:30:00', 5),
(1, '2023-06-03 11:45:00', 20),
(2, '2023-06-04 08:15:00', 15),
(2, '2023-06-05 16:20:00', 8),
(3, '2023-06-06 09:30:00', 12),
(4, '2023-06-07 13:00:00', 25),
(5, '2023-06-08 11:15:00', 18),
(5, '2023-06-09 08:45:00', 30),
(6, '2023-06-10 16:30:00', 6),
(6, '2023-06-11 10:00:00', 15),
(6, '2023-06-12 12:30:00', 10);
```

	product_stock_id	product_id	stock_receive_date	stock_qty
▶	1	1	2023-06-01 09:00:00	10
	2	1	2023-06-02 14:30:00	5
	3	1	2023-06-03 11:45:00	20
	4	2	2023-06-04 08:15:00	15
	5	2	2023-06-05 16:20:00	8
	6	3	2023-06-06 09:30:00	12
	7	4	2023-06-07 13:00:00	25
	8	5	2023-06-08 11:15:00	18
	9	5	2023-06-09 08:45:00	30
	10	6	2023-06-10 16:30:00	6
	11	6	2023-06-11 10:00:00	15
	12	6	2023-06-12 12:30:00	10
	NUL	NUL	NUL	NUL

SUPPLIERS

```
CREATE TABLE suppliers (
    supplier_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    supplier_category VARCHAR(50) NOT NULL,
    supplier_name VARCHAR(50) NOT NULL,
    contact_person VARCHAR(50) NOT NULL,
    supplier_address VARCHAR(100) NOT NULL,
    phone VARCHAR(20) NOT NULL
);
```

```
-- INSERT INTO suppliers
INSERT INTO suppliers (supplier_category, supplier_name, contact_person, supplier_address, phone) VALUES
('Electronics', 'ABC Electronics', 'John Smith', '123 Main Street, Anytown', '123-456-7890'),
('Clothing', 'XYZ Clothing', 'Jane Doe', '456 Elm Street, Othertown', '987-654-3210'),
('Home and Kitchen', '123 Home Appliances', 'Mike Johnson', '789 Oak Avenue, Hometown', '555-123-4567'),
('Beauty', 'Beauty Supplies Inc.', 'Sarah Thompson', '321 Maple Drive, Cityville', '555-987-6543'),
('Books', 'Book World', 'Robert Davis', '654 Cedar Lane, Booktown', '111-222-3333'),
('Sports', 'Sports Unlimited', 'Chris Roberts', '888 Pine Street, Athletica', '444-555-6666');
```

	supplier_id	supplier_category	supplier_name	contact_person	supplier_address	phone
▶	1	Electronics	ABC Electronics	John Smith	123 Main Street, Anytown	123-456-7890
	2	Clothing	XYZ Clothing	Jane Doe	456 Elm Street, Othertown	987-654-3210
	3	Home and Kitchen	123 Home Appliances	Mike Johnson	789 Oak Avenue, Hometown	555-123-4567
	4	Beauty	Beauty Supplies Inc.	Sarah Thompson	321 Maple Drive, Cityville	555-987-6543
	5	Books	Book World	Robert Davis	654 Cedar Lane, Booktown	111-222-3333
	6	Sports	Sports Unlimited	Chris Roberts	888 Pine Street, Athletica	444-555-6666
●	HULL	HULL	HULL	HULL	HULL	HULL

RECEIPT

```
CREATE TABLE receipt (
    receipt_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    supplier_id INT NOT NULL,
    product_id INT NOT NULL,
    receive_qty INT NOT NULL,
    receive_date DATETIME NOT NULL,
    FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

```
-- INSERT INTO receipt
INSERT INTO receipt (supplier_id, product_id, receive_qty, receive_date) VALUES
(1, 1, 5, '2023-06-06 10:30:00'),
(1, 2, 3, '2023-06-07 13:45:00'),
(2, 3, 10, '2023-06-08 16:00:00'),
(2, 4, 7, '2023-06-09 09:15:00'),
(1, 5, 4, '2023-06-10 11:30:00'),
(3, 6, 8, '2023-06-11 14:00:00'),
(4, 1, 15, '2023-06-12 16:30:00'),
(5, 2, 20, '2023-06-13 10:45:00'),
(5, 3, 12, '2023-06-14 12:15:00'),
(6, 4, 5, '2023-06-15 15:45:00'),
(6, 6, 10, '2023-06-16 09:30:00'),
(4, 1, 6, '2023-06-17 11:00:00');
```

	receipt_id	supplier_id	product_id	receive_qty	receive_date
▶	1	1	1	5	2023-06-06 10:30:00
	2	1	2	3	2023-06-07 13:45:00
	3	2	3	10	2023-06-08 16:00:00
	4	2	4	7	2023-06-09 09:15:00
	5	1	5	4	2023-06-10 11:30:00
	6	3	6	8	2023-06-11 14:00:00
	7	4	1	15	2023-06-12 16:30:00
	8	5	2	20	2023-06-13 10:45:00
	9	5	3	12	2023-06-14 12:15:00
	10	6	4	5	2023-06-15 15:45:00
	11	6	6	10	2023-06-16 09:30:00
	12	4	1	6	2023-06-17 11:00:00
	NUL	NUL	NUL	NUL	NUL

CUSTOMER

```
CREATE TABLE customers (
    customer_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    customer_name VARCHAR(50) NOT NULL,
    customer_address VARCHAR(100) NOT NULL,
    phone VARCHAR(20) NOT NULL
);
```

```
-- INSERT INTO customers
INSERT INTO customers (customer_name, customer_address, phone) VALUES
('John Doe', '789 Oak Avenue, Another Town', '555-123-4567'),
('Jane Smith', '321 Maple Drive, Somewhere', '555-987-6543'),
('Robert Johnson', '456 Elm Street, Anytown', '555-555-5555'),
('Sarah Davis', '123 Main Street, Hometown', '555-111-2222');
```

	customer_id	customer_name	customer_address	phone
▶	1	John Doe	789 Oak Avenue, Another Town	555-123-4567
	2	Jane Smith	321 Maple Drive, Somewhere	555-987-6543
	3	Robert Johnson	456 Elm Street, Anytown	555-555-5555
	4	Sarah Davis	123 Main Street, Hometown	555-111-2222
	NUL	NUL	NUL	NUL

ORDERS

```
CREATE TABLE orders (
    order_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    customer_id INT NOT NULL,
    product_id INT NOT NULL,
    order_qty INT NOT NULL,
    order_date DATETIME NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

```
-- INSERT INTO orders
INSERT INTO orders (customer_id, product_id, order_qty, order_date) VALUES
(1, 1, 2, '2023-06-11 14:30:00'),
(1, 3, 5, '2023-06-12 09:45:00'),
(2, 2, 1, '2023-06-13 12:00:00'),
(2, 4, 3, '2023-06-14 15:15:00'),
(3, 5, 4, '2023-06-15 10:30:00'),
(3, 6, 2, '2023-06-16 13:45:00'),
(4, 2, 1, '2023-06-17 17:00:00'),
(4, 5, 3, '2023-06-18 10:15:00'),
(4, 1, 2, '2023-06-19 12:30:00'),
(1, 5, 4, '2023-06-20 14:45:00'),
(2, 6, 1, '2023-06-21 17:00:00'),
(3, 3, 2, '2023-06-22 10:30:00');
```

	order_id	customer_id	product_id	order_qty	order_date
▶	1	1	1	2	2023-06-11 14:30:00
	2	1	3	5	2023-06-12 09:45:00
	3	2	2	1	2023-06-13 12:00:00
	4	2	4	3	2023-06-14 15:15:00
	5	3	5	4	2023-06-15 10:30:00
	6	3	6	2	2023-06-16 13:45:00
	7	4	2	1	2023-06-17 17:00:00
	8	4	5	3	2023-06-18 10:15:00
	9	4	1	2	2023-06-19 12:30:00
	10	1	5	4	2023-06-20 14:45:00
	11	2	6	1	2023-06-21 17:00:00
	12	3	3	2	2023-06-22 10:30:00
*	NULL	NULL	NULL	NULL	NULL

QUERIES

PRODUCT OP

```
-- Display all product and related category
```

```
SELECT p.product_name, c.category_name  
FROM products p  
JOIN product_category c ON p.product_category_id = c.product_category_id;
```

	product_name	category_name
▶	Smartphone	Electronics
	Laptop	Electronics
	T-Shirt	Clothing
	Jeans	Clothing
	Blender	Home and Kitchen
	Shampoo	Beauty
	Face Cream	Beauty
	The Great Gatsby	Books
	To Kill a Mockingbird	Books
	Yoga Mat	Sports
	Dumbbell Set	Sports

```
-- Display product with available stock
```

```
SELECT p.product_name, ps.stock_qty  
FROM products p  
JOIN product_stocks ps ON p.product_id = ps.product_id  
WHERE ps.stock_qty > 0;
```

	product_name	stock_qty
▶	Smartphone	10
	Smartphone	5
	Smartphone	20
	Laptop	15
	Laptop	8
	T-Shirt	12
	Jeans	25
	Blender	18
	Blender	30
	Shampoo	6
	Shampoo	15
	Shampoo	10

PRODUCT OP

```
-- Get a list of all product categories along with their descriptions  
SELECT category_name, descriptions FROM product_category;
```

	category_name	descriptions
▶	Electronics	Electronic devices and components
	Clothing	Clothing and apparel
	Home and Kitchen	Home appliances and kitchenware
	Beauty	Beauty and personal care
	Books	Books and literature
	Sports	Sports and fitness

```
-- Get stock of certain products based on product ID  
SELECT ps.stock_qty  
FROM product_stocks ps  
JOIN products p ON ps.product_id = p.product_id  
WHERE p.product_id = 6;
```

	stock_qty
▶	6
	15
	10

```
-- Get a list of products with prices above a certain value  
SELECT product_name, unit_price  
FROM products  
WHERE unit_price > 50.99;
```

	product_name	unit_price
▶	Smartphone	999.99
	Laptop	1499.99
	Dumbbell Set	59.99

RECEIPT TOP

```
-- Get a list of goods received within a certain date range
SELECT r.receipt_id, r.receive_date, p.product_name, r.receive_qty
FROM receipt r
JOIN products p ON r.product_id = p.product_id
WHERE r.receive_date BETWEEN '2023-06-06 00:00:00' AND '2023-06-08 23:59:59';
```

	receipt_id	receive_date	product_name	receive_qty
▶	1	2023-06-06 10:30:00	Smartphone	5
	2	2023-06-07 13:45:00	Laptop	3
	3	2023-06-08 16:00:00	T-Shirt	10

SUPPLIER OP

```
-- Get a list of all goods received by a particular supplier
SELECT r.receipt_id, r.receive_date, p.product_name, r.receive_qty
FROM receipt r
JOIN products p ON r.product_id = p.product_id
WHERE r.supplier_id = 2;
```

	receipt_id	receive_date	product_name	receive_qty
▶	3	2023-06-08 16:00:00	T-Shirt	10
	4	2023-06-09 09:15:00	Jeans	7

ORDER OP

```
-- Get a list of products that have been ordered by a particular customer
SELECT o.order_id, p.product_name, o.order_qty
FROM orders o
JOIN products p ON o.product_id = p.product_id
JOIN customers c ON o.customer_id = c.customer_id
WHERE c.customer_id = 1;
```

	order_id	product_name	order_qty
▶	1	Smartphone	2
	2	T-Shirt	5
	10	Blender	4

```
-- Counting the number of orders made each month
SELECT MONTH(order_date) AS bulan, YEAR(order_date) AS tahun, COUNT(*) AS jumlah_pesanan
FROM orders
GROUP BY YEAR(order_date), MONTH(order_date)
ORDER BY tahun, bulan;
```

	bulan	tahun	jumlah_pesanan
▶	6	2023	12

THANK YOU

