MySQL Queries - Group 61

1. Create Database

```
CREATE DATABASE sales_system_db;
```

2. Use Database

```
USE sales_system_db;
```

3. Create Tables

Users Table

```
CREATE TABLE Users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    role ENUM('user', 'admin', 'seller') NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Sellers Table

```
CREATE TABLE seller (
    seller_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    seller_name VARCHAR(255) NOT NULL,
    business_name VARCHAR(255),
    seller_email VARCHAR(255),
    seller_phone VARCHAR(15),
    seller_image LONGTEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(id) ON DELETE CASCADE
);
```

Products Table

```
CREATE TABLE Products (
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```sql
    name VARCHAR(100) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    stock_qty INT NOT NULL,
    seller_id INT,
    product_image LONGTEXT,
    category ENUM(
        'Electronics',
        'Clothing',
        'Home Appliances',
        'Books',
        'Sports',
        'Beauty & Health',
        'Toys',
        'Automotive',
        'Furniture',
        'Groceries'
    ) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (seller_id) REFERENCES seller(seller_id) ON DELETE SET NULL
);
```

Orders Table

```sql
CREATE TABLE Orders (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT NOT NULL,
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    total DECIMAL(10, 2) NOT NULL,
    seller_status ENUM('pending', 'shipped') DEFAULT 'pending',
    buyer_status ENUM('active', 'cancelled', 'delivered') DEFAULT 'active',
    FOREIGN KEY (user_id) REFERENCES Users(id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES Products(id) ON DELETE CASCADE
);
```

Feedback Table

```sql
CREATE TABLE feedback (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) NOT NULL,
    description TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    status ENUM('pending', 'reviewed', 'resolved') DEFAULT 'pending'
);
```

4. Data Retrieval Queries

Get All Users with Role 'user'

```
SELECT * FROM Users WHERE role = 'user' ORDER BY created_at DESC;
```

Count Total Users

```
SELECT COUNT(*) AS total_users FROM Users;
```

Count Total Products

```
SELECT COUNT(*) AS total_products FROM Products;
```

Count Total Orders

```
SELECT COUNT(*) AS total_orders FROM Orders;
```

Count Total Feedback Entries

```
SELECT COUNT(*) AS total_feedback FROM feedback;
```

Get Recent 5 Orders

```
SELECT
    o.id AS order_id,
    u.name AS user_name,
    p.name AS product_name,
    o.quantity,
    o.total,
    o.order_date,
    o.seller_status,
```

```sql
    o.buyer_status
FROM Orders o
JOIN Users u ON o.user_id = u.id
JOIN Products p ON o.product_id = p.id
ORDER BY o.order_date DESC
LIMIT 5;
```

Get All Feedback Entries

```sql
SELECT * FROM feedback ORDER BY created_at DESC;
```

Get Detailed Order Information

```sql
SELECT
    o.id AS order_id,
    u.name AS user_name,
    p.name AS product_name,
    s.seller_name,
    o.quantity,
    o.total,
    o.order_date,
    o.seller_status,
    o.buyer_status,
    p.product_image,
    s.seller_image
FROM Orders o
JOIN Users u ON o.user_id = u.id
JOIN Products p ON o.product_id = p.id
JOIN seller s ON p.seller_id = s.seller_id
ORDER BY o.order_date DESC;
```

Get Products with Seller Name

```sql
SELECT
    p.id,
    p.name,
    p.description,
    p.price,
    p.stock_qty,
    p.category,
    p.product_image,
    p.created_at,
    s.seller_name
FROM Products p
LEFT JOIN seller s ON p.seller_id = s.seller_id
ORDER BY p.created_at DESC;
```

Get All Sellers

```
SELECT * FROM seller ORDER BY created_at DESC;
```

5. Update Queries

Update Feedback Status

```
UPDATE feedback SET status = ? WHERE id = ?;
```

Get Seller by User ID

```
SELECT * FROM seller WHERE user_id = ?;
```

Get Total Sales for Seller

```
SELECT SUM(o.total) AS total_sales
FROM Orders o
JOIN Products p ON o.product_id = p.id
WHERE p.seller_id = ?
  AND o.seller_status = 'shipped';
```

Get Pending Orders for Seller

```
SELECT COUNT(*) AS orders_pending
FROM Orders o
JOIN Products p ON o.product_id = p.id
WHERE p.seller_id = ?
  AND o.seller_status = 'pending';
```

Get Products in Stock for Seller

```sql
SELECT COUNT(*) AS products_in_stock
FROM Products
WHERE seller_id = ?;
```

Get Monthly Revenue for Seller

```sql
SELECT SUM(o.total) AS monthly_revenue
FROM Orders o
JOIN Products p ON o.product_id = p.id
WHERE p.seller_id = ?
  AND o.seller_status = 'shipped'
  AND MONTH(o.order_date) = MONTH(CURRENT_DATE())
  AND YEAR(o.order_date) = YEAR(CURRENT_DATE());
```

Get Recent Orders for Seller

```sql
SELECT
    o.id,
    o.order_date,
    u.name AS customer_name,
    o.seller_status,
    o.total
FROM Orders o
JOIN Products p ON o.product_id = p.id
JOIN Users u ON o.user_id = u.id
WHERE p.seller_id = ?
ORDER BY o.order_date DESC
LIMIT 5;
```

Get Inventory Status for Seller

```sql
SELECT
    name,
    category,
    stock_qty,
    price
FROM Products
WHERE seller_id = ?
ORDER BY stock_qty ASC
LIMIT 5;
```

Get Specific Product

```
SELECT
    id,
    name,
    description,
    price,
    stock_qty,
    category,
    product_image
FROM Products
WHERE id = ?;
```

Delete Product by ID

```
DELETE FROM Products WHERE id = ?;
```

Get Seller Name and Image by User ID

```
SELECT
    seller_name,
    seller_image
FROM seller
WHERE user_id = ?;
```

Get User and Seller Details by User ID

```
SELECT
    u.name,
    u.email,
    s.seller_name,
    s.business_name,
    s.seller_email,
    s.seller_phone,
    s.seller_image
FROM Users u
LEFT JOIN seller s ON u.id = s.user_id
WHERE u.id = ?;
```

Update Seller Details

```sql
UPDATE seller
SET
    seller_name = ?,
    business_name = ?,
    seller_email = ?,
    seller_phone = ?,
    seller_image = ?
WHERE user_id = ?;
```

Update Order Status

```sql
UPDATE Orders
SET seller_status = ?
WHERE id = ?;
```

Insert New User

```sql
INSERT INTO Users (name, email, password, role) VALUES (?, ?, ?, ?);
```

Insert New Seller

```sql
INSERT INTO seller (user_id, seller_name, seller_email) VALUES (?, ?, ?);
```

Insert New Feedback

```sql
INSERT INTO feedback (email, description) VALUES (?, ?);
```

Insert New Order

```sql
INSERT INTO Orders (user_id, product_id, total, quantity, seller_status,
buyer_status) VALUES (?, ?, ?, ?, 'pending', 'active');
```

Update Product Stock Quantity

```sql
UPDATE Products SET stock_qty = ? WHERE id = ?;
```

Get Latest Order ID

```sql
SELECT id FROM Orders ORDER BY id DESC LIMIT 1;
```

Update Buyer Status

```sql
UPDATE Orders SET buyer_status = ? WHERE id = ?;
```

6. Additional Queries from PHP Scripts

User Authentication: Select User by Email

```sql
SELECT * FROM Users WHERE email = ?;
```

Register New User

```sql
INSERT INTO Users (name, email, password, role) VALUES (?, ?, ?, ?);
```

Register New Seller

```sql
INSERT INTO seller (user_id, seller_name, seller_email) VALUES (?, ?, ?);
```

Delete Product

```sql
DELETE FROM Products WHERE id = ?;
```

Update Product Details

```sql
-- If updating with a new image
UPDATE Products
SET name = ?, description = ?, price = ?, stock_qty = ?, category = ?,
product_image = ?
WHERE id = ?;

-- If updating without a new image
UPDATE Products
SET name = ?, description = ?, price = ?, stock_qty = ?, category = ?
WHERE id = ?;
```

Insert New Order During Transaction

```
INSERT INTO Orders (user_id, product_id, total, quantity, seller_status,
buyer_status) VALUES (?, ?, ?, ?, 'pending', 'active');
```

Update Stock Quantity After Order

```
UPDATE Products SET stock_qty = ? WHERE id = ?;
```

User Management

Select User by Email

```
SELECT * FROM Users WHERE email = ?;
```

Insert New User

```
INSERT INTO Users (name, email, password, role) VALUES (?, ?, ?, ?);
```

Insert New Seller

```
INSERT INTO seller (user_id, seller_name, seller_email) VALUES (?, ?, ?);
```

Get User and Seller Details by User ID

```
SELECT
      u.name,
      u.email,
      s.seller_name,
      s.business_name,
      s.seller_email,
      s.seller_phone,
      s.seller_image
  FROM Users u
  LEFT JOIN seller s ON u.id = s.user_id
  WHERE u.id = ?;
```

Product Management

Insert New Product

```
  INSERT INTO Products (name, description, price, stock_qty, category,
seller_id, product_image) VALUES (?, ?, ?, ?, ?, ?, ?);
```

Update Product Details

```
 -- With Image
  UPDATE Products
  SET name = ?, description = ?, price = ?, stock_qty = ?, category = ?,
product_image = ?
  WHERE id = ?;

  -- Without Image
  UPDATE Products
  SET name = ?, description = ?, price = ?, stock_qty = ?, category = ?
  WHERE id = ?;
```

Delete Product by ID

```
 DELETE FROM Products WHERE id = ?;
```

Get Specific Product

```
SELECT
      id,
      name,
      description,
      price,
      stock_qty,
      category,
      product_image
  FROM Products
  WHERE id = ?;
```

Get Products with Seller Name

```
SELECT
      p.id,
      p.name,
      p.description,
      p.price,
      p.stock_qty,
      p.category,
      p.product_image,
      p.created_at,
      s.seller_name
  FROM Products p
  LEFT JOIN seller s ON p.seller_id = s.seller_id
  ORDER BY p.created_at DESC;
```

Order Management

Insert New Order

```
  INSERT INTO Orders (user_id, product_id, total, quantity, seller_status,
buyer_status) VALUES (?, ?, ?, ?, 'pending', 'active');
```

Update Order Status

```
 UPDATE Orders
  SET seller_status = ?
  WHERE id = ?;
```

Update Buyer Status

```
  UPDATE Orders
  SET buyer_status = ?
  WHERE id = ?;
```

Get Recent Orders for Seller

```
 SELECT
      o.id,
      o.order_date,
      u.name AS customer_name,
      o.seller_status,
      o.total
  FROM Orders o
  JOIN Products p ON o.product_id = p.id
  JOIN Users u ON o.user_id = u.id
  WHERE p.seller_id = ?
  ORDER BY o.order_date DESC
  LIMIT 5;
```

Get Detailed Order Information

```
SELECT
      o.id AS order_id,
      u.name AS user_name,
      p.name AS product_name,
      s.seller_name,
      o.quantity,
      o.total,
      o.order_date,
      o.seller_status,
      o.buyer_status,
      p.product_image,
      s.seller_image
  FROM Orders o
  JOIN Users u ON o.user_id = u.id
  JOIN Products p ON o.product_id = p.id
  JOIN seller s ON p.seller_id = s.seller_id
  ORDER BY o.order_date DESC;
```

Get Recent 5 Orders

```
  SELECT
      o.id AS order_id,
      u.name AS user_name,
      p.name AS product_name,
      o.quantity,
      o.total,
      o.order_date,
      o.seller_status,
      o.buyer_status
  FROM Orders o
  JOIN Users u ON o.user_id = u.id
  JOIN Products p ON o.product_id = p.id
  ORDER BY o.order_date DESC
  LIMIT 5;
```

Get All Orders with Detailed Information

```
SELECT
    o.id AS order_id,
    u.name AS user_name,
    p.name AS product_name,
    s.seller_name,
    o.quantity,
    o.total,
    o.order_date,
    o.seller_status,
    o.buyer_status,
    p.product_image,
    s.seller_image
FROM Orders o
JOIN Users u ON o.user_id = u.id
JOIN Products p ON o.product_id = p.id
JOIN seller s ON p.seller_id = s.seller_id
ORDER BY o.order_date DESC;
```

Feedback Management

Insert New Feedback

```
INSERT INTO feedback (email, description) VALUES (?, ?);
```

Update Feedback Status

```
UPDATE feedback SET status = ? WHERE id = ?;
```

Registration and User Roles

Insert New User

```
INSERT INTO Users (name, email, password, role) VALUES (?, ?, ?, ?);
```

Insert New Seller

```
INSERT INTO seller (user_id, seller_name, seller_email) VALUES (?, ?, ?);
```

Miscellaneous

Get Latest Order ID

```
  SELECT id FROM Orders ORDER BY id DESC LIMIT 1;
```

Get Seller Name and Image by User ID

```
  SELECT
      seller_name,
      seller_image
  FROM seller
  WHERE user_id = ?;
```

Get User and Seller Details by User ID

```
SELECT
      u.name,
      u.email,
      s.seller_name,
      s.business_name,
      s.seller_email,
      s.seller_phone,
      s.seller_image
  FROM Users u
  LEFT JOIN seller s ON u.id = s.user_id
  WHERE u.id = ?;
```

Update Seller Details

```
UPDATE seller
  SET
      seller_name = ?,
      business_name = ?,
      seller_email = ?,
      seller_phone = ?,
      seller_image = ?
  WHERE user_id = ?;
```