```
CREATE TABLE authors (
    author_id INT AUTO_INCREMENT PRIMARY KEY,
    author_name VARCHAR(255) NOT NULL
);
CREATE TABLE genres (
    genre_id INT AUTO_INCREMENT PRIMARY KEY,
    genre_name VARCHAR(255) NOT NULL
);
CREATE TABLE books (
    book_id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    publication_year YEAR NOT NULL,
    author_id INT,
    genre_id INT,
    FOREIGN KEY (author_id) REFERENCES authors(author_id),
    FOREIGN KEY (genre_id) REFERENCES genres(genre_id)
);
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL
);
CREATE TABLE borrowed_books (
    borrow_id INT AUTO_INCREMENT PRIMARY KEY,
    book_id INT,
    user_id INT,
    borrow_date DATE NOT NULL,
    return_date DATE,
    FOREIGN KEY (book_id) REFERENCES books(book_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
INSERT INTO authors (author_name)
VALUES
    ('J.K. Rowling'),
    ('George Orwell');
INSERT INTO genres (genre_name)
VALUES
    ('Fantasy'),
    ('Dystopian');
INSERT INTO books (title, publication_year, author_id, genre_id)
VALUES
    ('Harry Potter and the Philosopher\'s Stone', 1997, 1, 1),
    ('1984', 1949, 2, 2);
INSERT INTO users (username, email)
VALUES
    ('john_doe', 'john_doe@example.com'),
    ('jane_smith', 'jane_smith@example.com');
INSERT INTO borrowed_books (book_id, user_id, borrow_date, return_date)
```

```
VALUES
    (1, 1, '2024-11-01', '2024-11-15'),
    (2, 2, '2024-11-05', NULL);

SELECT
    o.id AS order_id,
    c.name AS customer_name,
    c.country AS customer_country,
    e.first_name AS employee_first_name,
    e.last_name AS employee_last_name,
    od.product_id,
    p.name AS product_name,
    p.price AS product_price,
    cat.name AS category_name,
    s.name AS supplier_name,
    sh.name AS shipper_name,
    o.date AS order_date,
    od.quantity AS quantity_ordered
FROM
    dataset_db.orders o
INNER JOIN dataset_db.customers c ON o.customer_id = c.id
INNER JOIN dataset_db.employees e ON o.employee_id = e.employee_id
INNER JOIN dataset_db.order_details od ON o.id = od.order_id
INNER JOIN dataset_db.products p ON od.product_id = p.id
INNER JOIN dataset_db.categories cat ON p.category_id = cat.id
INNER JOIN dataset_db.suppliers s ON p.supplier_id = s.id
INNER JOIN dataset_db.shippers sh ON o.shipper_id = sh.id;

SELECT COUNT(*) AS row_count
FROM (
    SELECT
        o.id AS order_id,
        c.name AS customer_name,
        c.country AS customer_country,
        e.first_name AS employee_first_name,
        e.last_name AS employee_last_name,
        od.product_id,
        p.name AS product_name,
        p.price AS product_price,
        cat.name AS category_name,
        s.name AS supplier_name,
        sh.name AS shipper_name,
        o.date AS order_date,
        od.quantity AS quantity_ordered
    FROM
        dataset_db.orders o
    INNER JOIN dataset_db.customers c ON o.customer_id = c.id
    INNER JOIN dataset_db.employees e ON o.employee_id = e.employee_id
    INNER JOIN dataset_db.order_details od ON o.id = od.order_id
    INNER JOIN dataset_db.products p ON od.product_id = p.id
    INNER JOIN dataset_db.categories cat ON p.category_id = cat.id
    INNER JOIN dataset_db.suppliers s ON p.supplier_id = s.id
    INNER JOIN dataset_db.shippers sh ON o.shipper_id = sh.id
) AS combined_data;

SELECT COUNT(*) AS row_count_with_left_join
FROM (
    SELECT
        o.id AS order_id,
        c.name AS customer_name,
        c.country AS customer_country,
        e.first_name AS employee_first_name,
        e.last_name AS employee_last_name,
        od.product_id,
        p.name AS product_name,
        p.price AS product_price,
        cat.name AS category_name,
        s.name AS supplier_name,
        sh.name AS shipper_name,
        o.date AS order_date,
        od.quantity AS quantity_ordered
    FROM
        dataset_db.orders o
    LEFT JOIN dataset_db.customers c ON o.customer_id = c.id
    LEFT JOIN dataset_db.employees e ON o.employee_id = e.employee_id
    INNER JOIN dataset_db.order_details od ON o.id = od.order_id
    INNER JOIN dataset_db.products p ON od.product_id = p.id
    LEFT JOIN dataset_db.categories cat ON p.category_id = cat.id
    LEFT JOIN dataset_db.suppliers s ON p.supplier_id = s.id
    LEFT JOIN dataset_db.shippers sh ON o.shipper_id = sh.id
) AS combined_data;

SELECT
    o.id AS order_id,
    c.name AS customer_name,
    e.first_name AS employee_first_name,
    e.last_name AS employee_last_name,
    od.product_id,
    p.name AS product_name,
    p.price AS product_price,
    cat.name AS category_name,
    s.name AS supplier_name,
    sh.name AS shipper_name,
    o.date AS order_date,
    od.quantity AS quantity_ordered
FROM
```

```
    dataset_db.orders o
INNER JOIN dataset_db.customers c ON o.customer_id = c.id
INNER JOIN dataset_db.employees e ON o.employee_id = e.employee_id
INNER JOIN dataset_db.order_details od ON o.id = od.order_id
INNER JOIN dataset_db.products p ON od.product_id = p.id
INNER JOIN dataset_db.categories cat ON p.category_id = cat.id
INNER JOIN dataset_db.suppliers s ON p.supplier_id = s.id
INNER JOIN dataset_db.shippers sh ON o.shipper_id = sh.id
WHERE e.employee_id > 3 AND e.employee_id <= 10;

SELECT
    cat.name AS category_name,
    COUNT(*) AS row_count,
    AVG(od.quantity) AS avg_quantity
FROM
    dataset_db.order_details od
INNER JOIN dataset_db.products p ON od.product_id = p.id
INNER JOIN dataset_db.categories cat ON p.category_id = cat.id
GROUP BY
    cat.name;

SELECT
    cat.name AS category_name,
    COUNT(*) AS row_count,
    AVG(od.quantity) AS avg_quantity
FROM
    dataset_db.order_details od
INNER JOIN dataset_db.products p ON od.product_id = p.id
INNER JOIN dataset_db.categories cat ON p.category_id = cat.id
GROUP BY
    cat.name
HAVING
    AVG(od.quantity) > 21;

SELECT
    cat.name AS category_name,
    COUNT(*) AS row_count,
    AVG(od.quantity) AS avg_quantity
FROM
    dataset_db.order_details od
INNER JOIN dataset_db.products p ON od.product_id = p.id
INNER JOIN dataset_db.categories cat ON p.category_id = cat.id
GROUP BY
    cat.name
HAVING
    AVG(od.quantity) > 21
ORDER BY
    row_count DESC;
```

```
SELECT
    cat.name AS category_name,
    COUNT(*) AS row_count,
    AVG(od.quantity) AS avg_quantity
FROM
    dataset_db.order_details od
INNER JOIN dataset_db.products p ON od.product_id = p.id
INNER JOIN dataset_db.categories cat ON p.category_id = cat.id
GROUP BY
    cat.name
HAVING
    AVG(od.quantity) > 21
ORDER BY
    row_count DESC
LIMIT 4 OFFSET 1;
```