

Assignment 1:

Given the id of a user, fetch all orders (Id, Order Date, Order Total) of that user which are in shipped state. Orders should be sorted by order date column in chronological order.

(assume that a order can be in shipped state if all its order_item are in the shipped state)

```
SELECT
    o.order_id,
    o.order_date,
    o.cart_value
FROM
    Users u
INNER JOIN
    Orders o ON o.user_id = u.user_id
WHERE
    u.user_id = "1"
    AND NOT EXISTS (
        SELECT *
        FROM Order_Item oi
        LEFT JOIN
            Order_status os ON os.order_item_id = oi.order_item_id
        WHERE oi.order_id = o.order_id
            AND os.status != 'shipped'
    )
ORDER BY
    o.order_date;
```

User_table:

user_id	email_id	password	phone	type	created_at
1	john.doe@example.com	password123	9876543210	SHOPPER	2025-03-24 17:28:18
2	jane.smith@example.com	securePass1	9123456789	ADMIN	2025-03-24 17:28:18
3	alice.jones@example.com	alicePass99	9987654321	SHOPPER	2025-03-24 17:28:18
4	bob.brown@example.com	brownie456	9785641230	SHOPPER	2025-03-24 17:28:18
5	charlie.davis@example.com	charlie789	9432167890	SHOPPER	2025-03-24 17:28:18
6	emily.wilson@example.com	wilson123	9551234567	SHOPPER	2025-03-24 17:28:18
7	daniel.lee@example.com	leePass555	9987412365	ADMIN	2025-03-24 17:28:18
8	olivia.moore@example.com	oliviaPass	9911223344	SHOPPER	2025-03-24 17:28:18
9	henry.taylor@example.com	henryPass	9109876543	SHOPPER	2025-03-24 17:28:18
10	sophia.anderson@example.com	sophia123	8899776655	SHOPPER	2025-03-24 17:28:18

Orders table:

order_id	user_id	cart_value	order_date	address_id
1	1	250.75	2025-03-24 17:28:32	1
2	2	1250.00	2025-03-24 17:28:32	2
3	3	99.99	2025-03-24 17:28:32	3
4	4	320.50	2025-03-24 17:28:32	4
5	5	45.90	2025-03-24 17:28:32	5
6	6	560.00	2025-03-24 17:28:32	6
7	7	670.45	2025-03-24 17:28:32	7
8	8	149.99	2025-03-24 17:28:32	8
9	9	240.10	2025-03-24 17:28:32	9
10	10	299.80	2025-03-24 17:28:32	10
19	1	1500.50	2025-03-18 17:54:11	111
20	2	2500.75	2025-03-08 17:54:11	112
21	3	1200.00	2025-02-11 17:54:11	113
22	4	500.25	2025-01-27 17:54:11	114
23	5	800.40	2024-12-28 17:54:11	115
24	1	400.15	2024-11-28 17:54:11	111
25	3	1000.25	2024-10-29 17:54:11	113
26	2	300.75	2024-09-29 17:54:11	112
27	1	1500.50	2024-11-28 18:01:21	111
28	2	2500.75	2024-10-29 18:01:21	112
29	3	800.40	2024-09-29 18:01:21	113
30	4	1000.25	2024-09-09 18:01:21	114

Order_item:

order_item_id	order_id	quantity_added	total	product_id
1	1	2	2000	1
2	2	5	7500	2
3	3	1	20	3
4	4	2	100	4
5	5	3	135	5
6	6	4	40	6
8	8	3	90	8
9	9	2	200	9
10	10	1	200	10
11	10	5	333	1
1001	24	2	3000	2
1002	25	1	2501	3
1003	26	3	2401	4
1004	27	1	1000	5

Order_status:

order_item_id	status	updated_at
1	SHIPPED	2025-03-24 17:29:09
2	CANCELED	2025-03-24 17:29:09
3	RETURNED	2025-03-24 17:29:09
4	SHIPPED	2025-03-24 17:29:09
5	SHIPPED	2025-03-24 17:29:09
6	SHIPPED	2025-03-24 17:29:09
8	RETURNED	2025-03-24 17:29:09
9	SHIPPED	2025-03-24 17:29:09
10	SHIPPED	2025-03-24 17:29:09
11	SHIPPED	2025-03-31 00:38:26
1001	SHIPPED	2025-03-31 00:38:26
1002	SHIPPED	2025-03-31 00:38:26
1003	SHIPPED	2025-03-31 00:38:26
1004	SHIPPED	2025-03-31 00:38:26

Sub-query:

product_id	status
2	CANCELED
3	RETURNED
8	RETURNED

// result 🍌

order_id	order_date	cart_value
24	2024-11-28 17:54:11	400.15
27	2024-11-28 18:01:21	1500.50
19	2025-03-18 17:54:11	1500.50
1	2025-03-24 17:28:32	250.75

Assignment 2:

Insert five or more images of a product using batch insert technique.

```
Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase", "root", "root");
PreparedStatement ps = connection.prepareStatement("INSERT INTO Users (user_id, name,
email) VALUES (?, ?, ?)");

ps.setInt(1, 1);
ps.setString(2, "John");
ps.setString(3, "john@example.com");
ps.addBatch();

ps.setInt(1, 2);
ps.setString(2, "Jane");
ps.setString(3, "jane@example.com");
ps.addBatch();

ps.setInt(1, 3);
ps.setString(2, "Alice");
ps.setString(3, "alice@example.com");
ps.addBatch();

ps.executeBatch(); // Executes all the statements in a single batch
connection.close();
```

Assignment 3:

Delete all those products which were not ordered by any Shopper in last 1 year. Return the number of products deleted.

```
DELETE FROM Product
WHERE product_id NOT IN (
    SELECT DISTINCT oi.product_id
    FROM Orders o
    INNER JOIN
        Order_Item oi ON oi.order_id = o.order_id
    INNER JOIN
        Users u ON u.user_id = o.user_id
    WHERE
        o.order_date >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR) AND u.type =
"SHOPPER"
);
```

// sub-query

product_id
1
3
4
5
6
8
9
10
2

Assignment 4:

Select and display the category title of all top parent categories sorted alphabetically and the count of their child categories.

```
SELECT
  c.category_name AS TITLE,
  COUNT(c.category_id)
FROM Category c
LEFT JOIN Child_category cc
  ON cc.category_id = c.category_id
GROUP BY TITLE
ORDER BY TITLE;
```

child-category table:

child_category_name	category_id
Laptops	1
Mobile Phones	1
Men's Wear	2
Women's Wear	2
Kitchen Appliances	3
Fiction	4
Non-fiction	4
Fitness	5
Outdoor	5
Study Tables	7

Category table:

category_id	category_name
9	Beauty
4	Books
2	Clothing
1	Electronics
7	Furniture
10	Groceries
3	Home Appliances
5	Sports
8	Stationery
6	Toys

// result:

TITLE	COUNT(c.category_id)
Beauty	1
Books	2
Clothing	2
Electronics	2
Furniture	1
Groceries	1
Home Appliances	1
Sports	2
Stationery	1
Toys	1