**SVKM’s NMIMS**

**School of Technology Management & Engineering, Chandigarh**

A.Y. 2023 - 24

**Course: Database Management Systems**

**Project Report**

|  |  |  |
| --- | --- | --- |
| Program | B.tech Computer Engineering | |
| Semester | 4 | |
| Name of the Project: | E-commerce Stationery DataBase | |
|  | | |
| Details of Project Members |  |  |
| Batch | Roll No. | Name |
| B1 & B2 | A069, A123 &. A126 | Kusum Panamgipalli, Anshu Mehta, Vaishnavi Awasthi |
| Date of Submission: 02-04-2024 | | |

**Contribution of each project Members:**

|  |  |  |
| --- | --- | --- |
| Roll No. | Name: | Contribution |
| A069 | Kusum Panamgipalli | All the SQL table values |
| A123 | Anshu Mehta | All the 20 Queries of SQL |
| A126 | Vaishnavi Awasthi | Creation of E-R diagram and Relationship table |

**Github link of your project:**

**Note:**

1. Create a readme file if you have multiple files
2. All files must be properly named (Example:R004\_DBMSProject)
3. Submit all relevant files of your work ( Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

**Rubrics for the Project evaluation:**

|  |  |
| --- | --- |
| First phase of evaluation:  Innovative Ideas (5 Marks)  Design and Partial implementation (5 Marks) | 10 marks |
| Final phase of evaluation  Implementation, presentation and viva, Self-Learning and Learning Beyond classroom | 10 marks |

**Table of Contents**

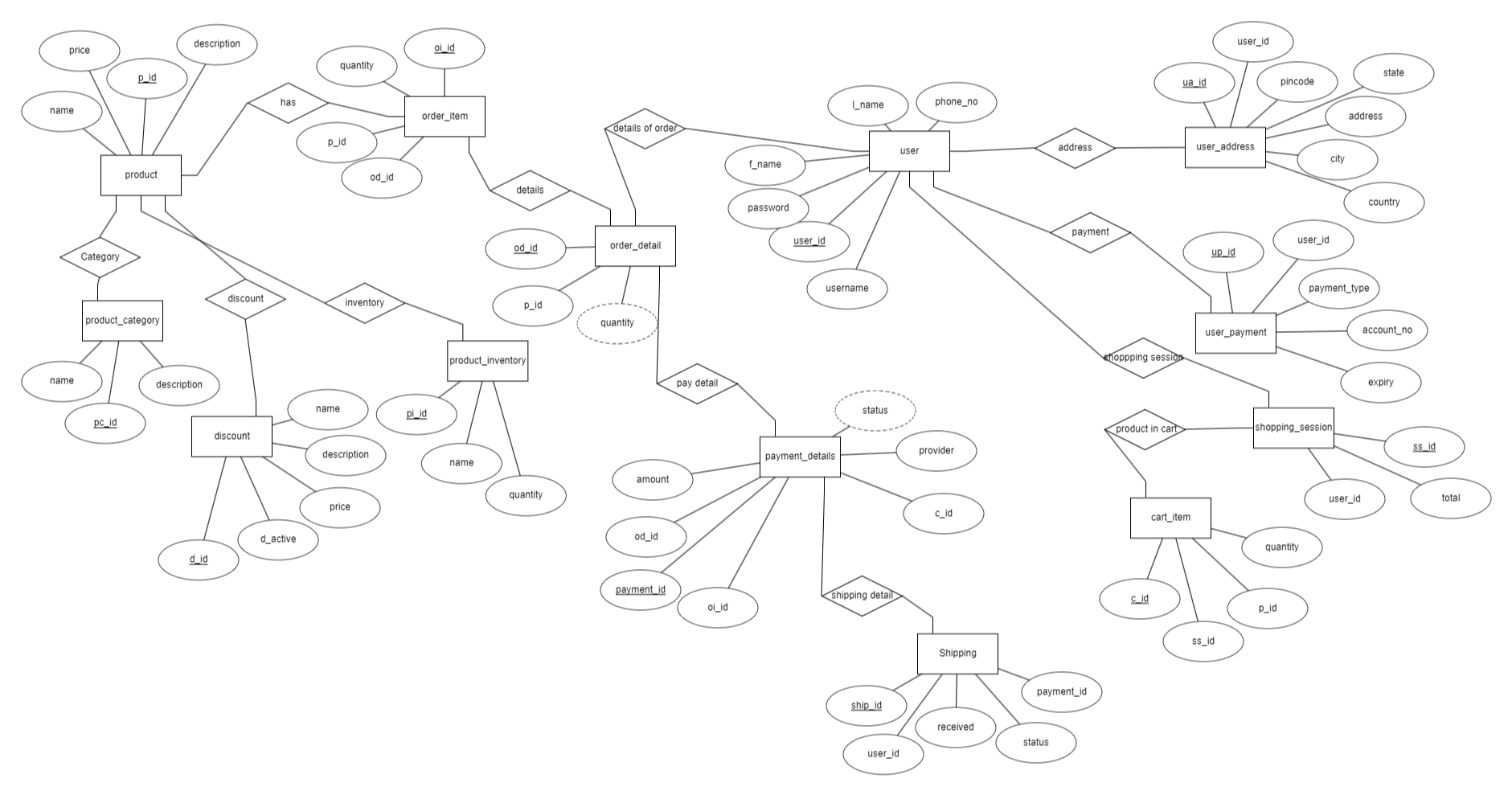
|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no.** |
| **1** | Storyline | 3 |
| **2** | Components of Database Design | 3 |
| **3** | Entity Relationship Diagram | 4 |
| **4** | Relational Model | 5 |
| **5** | Normalization | 5 |
| **6** | SQL Queries | 6 |
| **7** | Learning from the Project | 20 |
| 8 | Self-learning beyond classroom | 20 |
| 9 | Conclusion | 21 |

* 1. **Storyline**

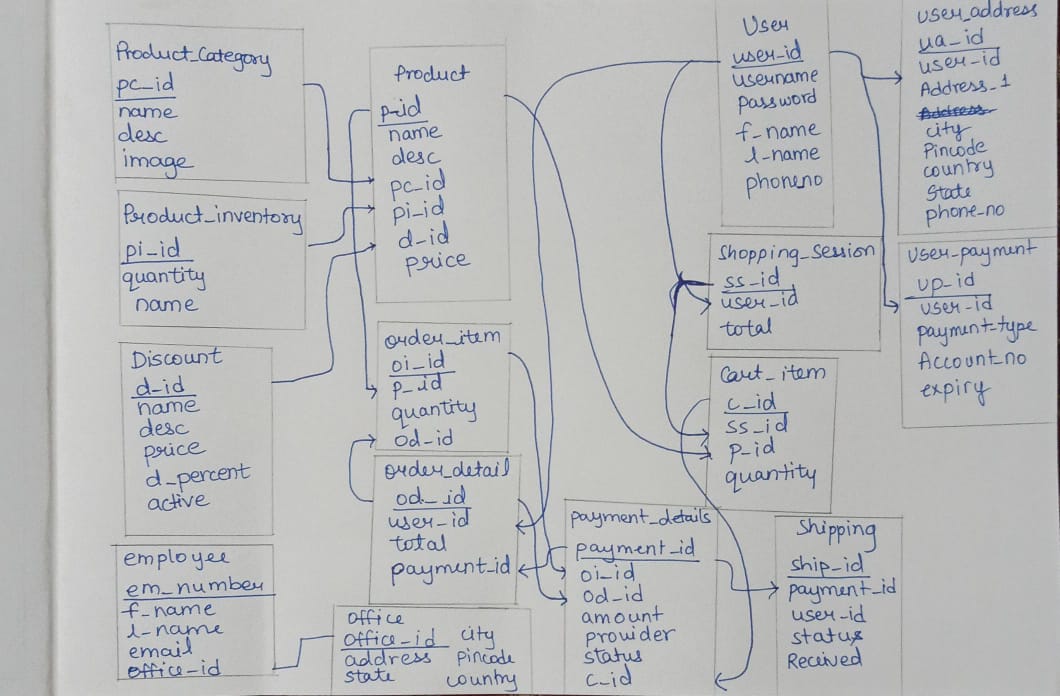
Welcome to Giggly Bits, your one-stop destination for all your stationery needs! At Giggly Bits, we are committed to providing you with a vast selection of high-quality stationery items, from the essential pens and notebooks to the creative and personalized stationery that adds a touch of elegance to your daily life. Whether you're a student looking for the perfect notebook for your notes, a professional seeking a sleek pen for your meetings, or someone in need of customized stationery for special occasions, we have got you covered. With our competitive prices, easy checkout process, and a wide range of products catering to both individual and business needs, Giggly Bits is your go-to for all your stationery requirements. Experience the convenience and variety of shopping with Giggly Bits. Book your order today and let the creativity begin. The system, named Giggly Bits Management System, will efficiently manage inventory, customer orders, and various administrative tasks, ensuring a seamless shopping experience for our customers.

**II. Components of Database**

**III. Entity Relationship Diagram**



**IV. Relational Model**



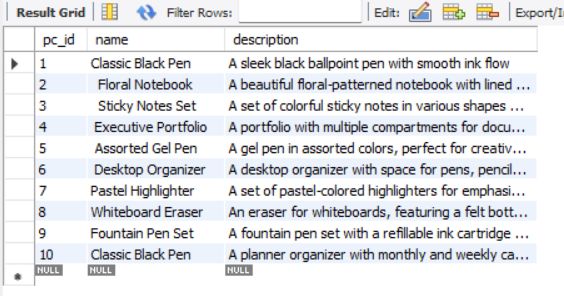
**V. Normalization**

All the tables were in 1NF and 2NF because they all had atomicity which means no multivalued attributes were present. All were in 2NF as they was no partial dependencies and LHS were all prime attributes and RHS was non-prime attributes.

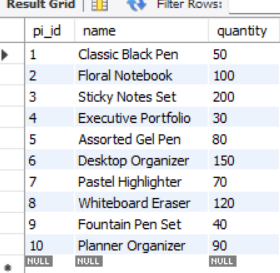
**VI. SQL Queries**

**Output of Tables:**

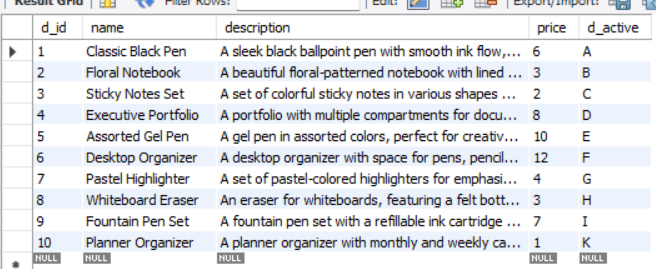
1)Product Category



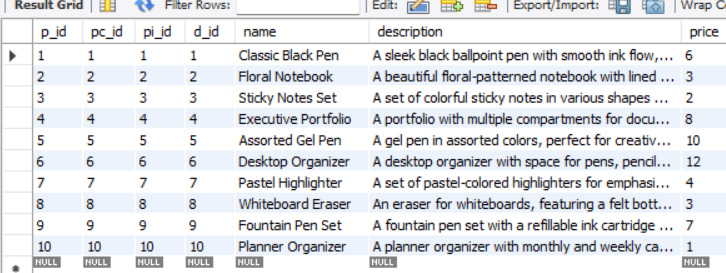
2)Product Inventory



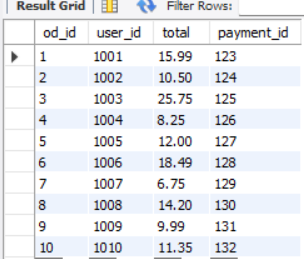
3) Discount



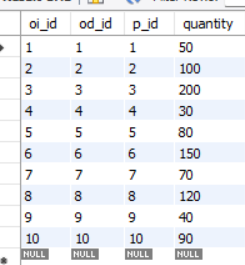
4)Product



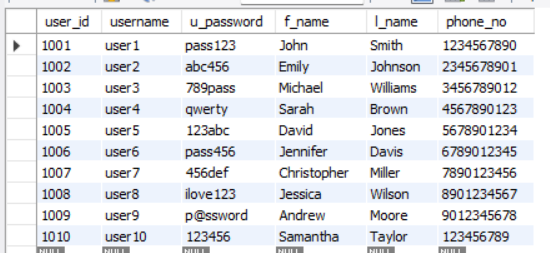
5)Oder Details



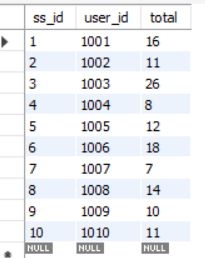
6)Order Item



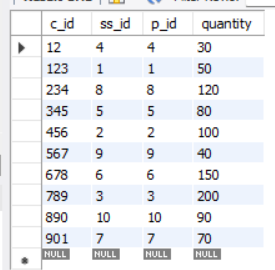
7)User



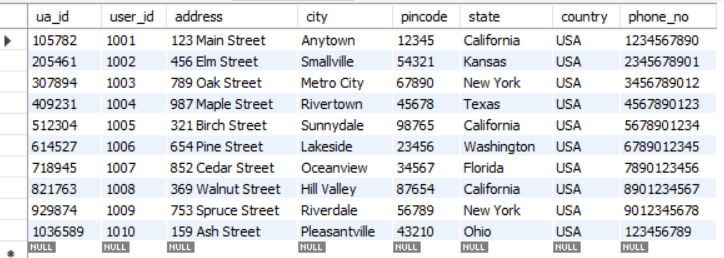
8)Shopping Session



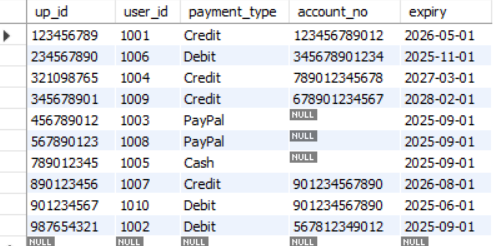
9)Cart Item



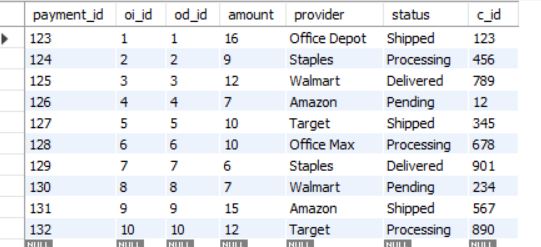
10) User Address



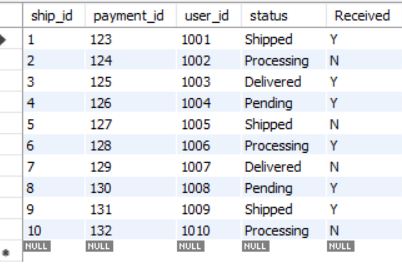
11)User Payment



12)Payment Details



13)Shipping



**SQL Queries:**

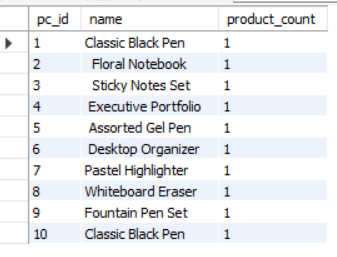
**#Query 1: Count the number of products in each category.**

SELECT product\_category.pc\_id, product\_category.name, COUNT(\*) AS product\_count

FROM product\_category

JOIN product ON product\_category.pc\_id = product.pc\_id

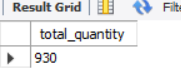
GROUP BY product\_category.pc\_id;



**#Query 2: Calculate the total quantity of products in inventory**

SELECT SUM(quantity) AS total\_quantity

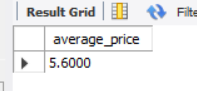
FROM product\_inventory;



**#Query 3: Find the average price of products.**

SELECT AVG(price) AS average\_price

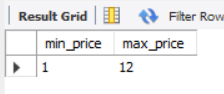
FROM product;



**#Query 4: Find the minimum and maximum price of products.**

SELECT MIN(price) AS min\_price, MAX(price) AS max\_price

FROM product;

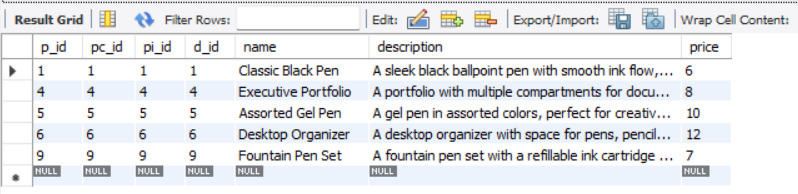


**#Query 5: Find all products with a price greater than the average price.(Nested Query)**

SELECT \*

FROM product

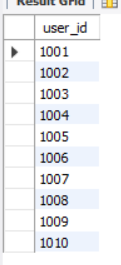
WHERE price > (SELECT AVG(price) FROM product);



**#Query 6: Find all users who have placed orders.**

SELECT DISTINCT user\_id

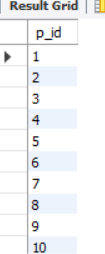
FROM order\_detail;



**#Query 7: Find all products that have been ordered.**

SELECT DISTINCT p\_id

FROM order\_item;

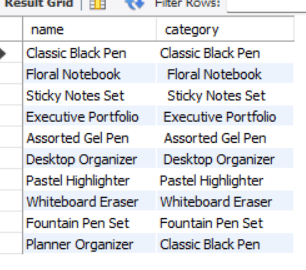


**#Query 8: List all products and their categories.**

SELECT product.name, product\_category.name AS category

FROM product

JOIN product\_category ON product.pc\_id = product\_category.pc\_id;

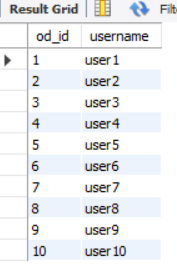


**#Query 9: List all orders and their corresponding users.**

SELECT order\_detail.od\_id, user.username

FROM order\_detail

JOIN user ON order\_detail.user\_id = user.user\_id;

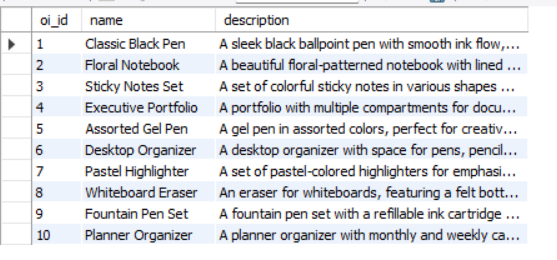


**#Query 10: List all products in an order, including product details.**

SELECT order\_item.oi\_id, product.name, product.description

FROM order\_item

JOIN product ON order\_item.p\_id = product.p\_id;



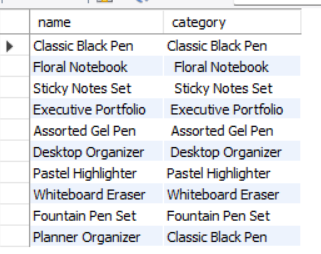
**#Query 11: Create a view to list all products with their categories.**

CREATE VIEW product\_category\_view AS

SELECT product.name, product\_category.name AS category

FROM product

JOIN product\_category ON product.pc\_id = product\_category.pc\_id;

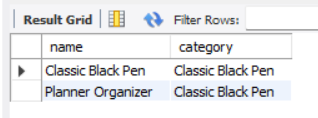


**#Query 12: Use the view to find all products in the "Classic Black Pen" category.**

SELECT \*

FROM product\_category\_view

WHERE category = 'Classic Black Pen';



**#Query 13: List all users and their addresses, including users without addresses.**

SELECT user.user\_id, user.username, user\_address.address

FROM user

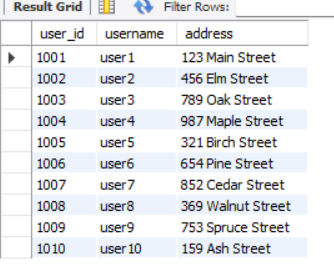
LEFT JOIN user\_address ON user.user\_id = user\_address.user\_id

UNION

SELECT user.user\_id, user.username, user\_address.address

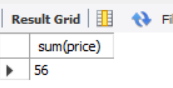
FROM user

RIGHT JOIN user\_address ON user.user\_id = user\_address.user\_id;



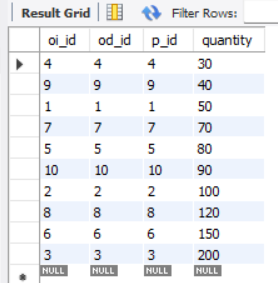
**#14) Give the total od the price when the discount was given;**

select sum(price) from Discount;



**#15) Arrange all order item based on their quantities in ascending order.**

select\*from order\_item order by quantity ASC;

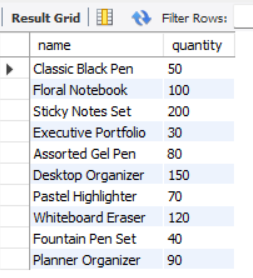


**#Query 16: List all products and their inventory details, including products without inventory.**

SELECT product.name, product\_inventory.quantity

FROM product

LEFT JOIN product\_inventory ON product.pi\_id = product\_inventory.pi\_id;

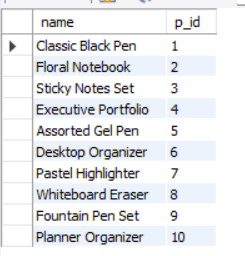


**#Query 17: List all inventory items and their corresponding products, including inventory items without products.**

SELECT product\_inventory.name, product.p\_id

FROM product\_inventory

RIGHT JOIN product ON product\_inventory.pi\_id = product.pi\_id;



**#Query 18: List all users and their addresses, including users without addresses.**

SELECT user.user\_id, user.username, user\_address.address

FROM user

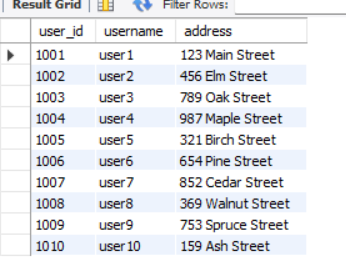
LEFT JOIN user\_address ON user.user\_id = user\_address.user\_id

UNION

SELECT user.user\_id, user.username, user\_address.address

FROM user

RIGHT JOIN user\_address ON user.user\_id = user\_address.user\_id;

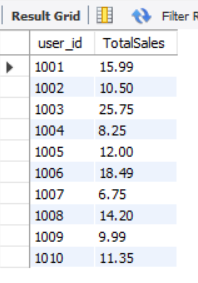


**#Query 19: To find total sales amount for each user**

SELECT user\_id, SUM(total) AS TotalSales

FROM order\_detail

GROUP BY user\_id;

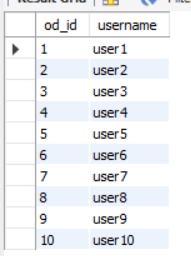


**#Query 20: List all orders and their corresponding users.**

SELECT order\_detail.od\_id, user.username

FROM order\_detail

INNER JOIN user ON order\_detail.user\_id = user.user\_id;



**VII. Self -Learning beyond classroom**:

* We learned what goes behind the Logic of any Business
* We learned how to design an E-R model of a business which plays very crucial role in building aspect of business.
* As business these days prefers to choose cloud database, we got an insight into Cloud Database creation through Google and Geeks for Geeks websites.

**VIII. Learning from the Project**

We learned how to Design an E-R model which is base of any database to be created. This project also helped us gain insight into what goes behind the websites that uses database.

We also learned that we can connect SQL and with file that helps create any websites.

We also learned that how SQL plays crucial role in Security aspect.

**X. Conclusion**

 We would like to conclude that this DBMS Mini Project is helped us understand the benefits of improvements it brings to the Stationery Shop’s operations. It increases the efficiency in Inventory Management. Technology gets involved into it which helps improve the business and leads to successful outputs. It enhances the Customer Experience, as nothing is more fruitful than the satisfaction a customer gets after purchasing the desired items, for any Entrepreneur.