MySQL Task-2

**Jash Patel**

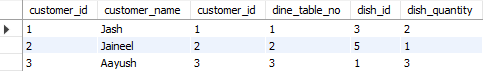
**Demonstration of Joins which includes only two table.**

Inner Join:

SELECT \* FROM orders

INNER JOIN customer

ON orders.customer\_id=customer.customer\_id;

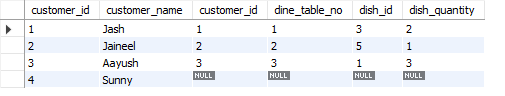


Left Join:

SELECT \* FROM customer

LEFT JOIN orders

ON customer.customer\_id=orders.customer\_id;



**Problem statements for Inner Joins including more than two tables.**

**Q. Join orders, customer and dish tables using inner join to retrieve customer name, table number, dish name, dish quantity.**

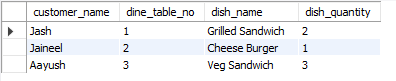
Ans.

SELECT customer\_name, dine\_table\_no, dish\_name, dish\_quantity

FROM orders

INNER JOIN (customer, dish)

ON (orders.customer\_id=customer.customer\_id AND orders.dish\_id=dish.dish\_id);



**Q. Join tables using inner join to determine which customer is sitting on which table and which staff is incharge of that table.**

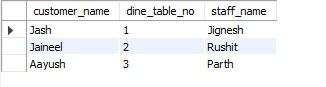
Ans.

SELECT customer.customer\_name, orders.dine\_table\_no, dine\_table.staff\_name

FROM orders

INNER JOIN (customer, dine\_table)

ON (orders.customer\_id=customer.customer\_id AND orders.dine\_table\_no=dine\_table.dine\_table\_no);



**Problem statements to demonstrate left join on more than two tables:**

**Q. Retrieve all customers, their orders (if any), the corresponding dish details, and the staff assigned to the dine table. If any detail is missing (like no order or no staff assigned), show NULL.**

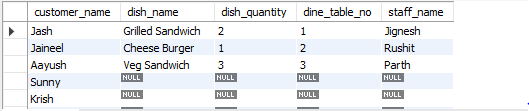
Ans.

SELECT customer.customer\_name, dish.dish\_name, orders.dish\_quantity, orders.dine\_table\_no, dine\_table.staff\_name

FROM (customer)

LEFT JOIN (orders, dish, dine\_table)

ON (orders.customer\_id=customer.customer\_id AND orders.dish\_id=dish.dish\_id AND orders.dine\_table\_no=dine\_table.dine\_table\_no);



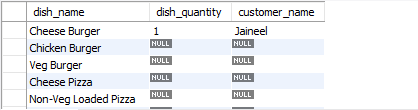
**Q. List all dishes with the total quantity ordered and the customer who placed the order. If a dish hasn't been ordered, show NULL for customer details.**

SELECT dish.dish\_name, orders.dish\_quantity, customer.customer\_name

FROM dish

LEFT JOIN (orders, customer)

ON (orders.customer\_id=customer.customer\_id AND orders.dish\_id=dish.dish\_id);



**Problem statements to demonstrate right join on more than two tables:**

**Q. List all staff, their dine tables, and the orders placed. If no orders exist, show NULL for the order details**.

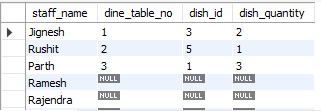
Ans.

SELECT staff.staff\_name, dine\_table.dine\_table\_no, orders.dish\_id, orders.dish\_quantity

FROM (dine\_table, orders)

RIGHT JOIN staff

ON (staff.staff\_id=dine\_table.staff\_id AND dine\_table.dine\_table\_no=orders.dine\_table\_no);



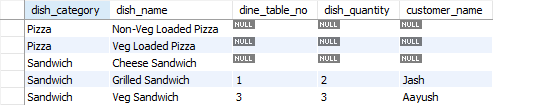
**Q. List all dishes, including those that have not been ordered, along with any existing order and customer information.**

SELECT dish.dish\_category, dish.dish\_name, orders.dine\_table\_no, orders.dish\_quantity, customer.customer\_name

FROM (orders, customer)

RIGHT JOIN dish

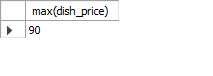
ON (dish.dish\_id=orders.dish\_id AND orders.customer\_id=customer.customer\_id);



**Demonstration of Sub Queries.**

**Q. Using Sub-Queries, find the second highest price of all the dishes from the dish table.**

SELECT max(dish\_price) FROM dish WHERE dish\_price < (SELECT max(dish\_price) FROM dish);



**Q. Using sub-queries, retrieve all the details of the dish from the dish table which have been ordered.**

SELECT \* FROM dish

WHERE dish\_id IN (SELECT dish\_id FROM orders);



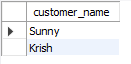
**Q. List the names of customers who have ordered the most expensive dish in the menu.**

ANS.

SELECT customer\_name FROM customer

WHERE customer\_id IN (SELECT customer\_id FROM orders

WHERE dish\_id=(SELECT dish\_id FROM dish WHERE dish\_price=(SELECT max(dish\_price) FROM dish)));



**Q. Find the names of staff members whose salary is higher than the average staff salary**

SELECT staff\_name FROM staff

WHERE staff\_salary>(SELECT AVG(staff\_salary) FROM staff);



Q. Retrieve the dine table numbers that have not received any orders.

Q. List customers who have ordered more than the average quantity of dishes per order.

Q. Identify dishes from the dish table that have never been ordered.

Q. Determine the staff member who has handled the highest number of orders.