**Day 10: Assignments 1&2:**

**Assignment 1:**

**Craft a query using an INNER JOIN to combine 'orders' and 'customers' tables for customers in a specified region, and a LEFT JOIN to display all customers including those without orders.**

>SELECT

c.customer\_id,

c.customer\_name,

c.region,

o.order\_id,

o.order\_date,

FROM

customer c

INNER JOIN

order o ON c.customer\_id = o.customer\_id

WHERE

c.region = ‘Europe’

ORDER BY

c.customer\_id, o.order\_date;

SELECT

c.customer\_id,

c.customer\_name,

c.region,

o.order\_id,

o.order\_date,

FROM

customer c

LEFT JOIN

order o ON c.customer\_id = o.customer\_id

ORDER BY

c.customer\_id, o.order\_date;

**Assignment 2:**

**Utilize a subquery to find customers who have placed orders above the average order value, and write a UNION query to combine two SELECT statements with the same number of columns.**

**Subquery to find customers with orders above average:**

>SELECT c.customer\_id, c.customer\_name, o.order\_id, o.order\_amount

FROM customer AS c

INNER JOIN order AS o ON c.customer\_id = o.customer\_id

WHERE o.order\_amount > (

SELECT AVG(order\_amount)

FROM order

);

**UNION query to combine two SELECT statements with the same number of columns.**

>SELECT customer\_id, customer\_name, NULL AS order\_id, NULL AS order\_amount

FROM customer

WHERE customer\_id NOT IN (

SELECT c.customer\_id

FROM customer AS c

INNER JOIN order AS o ON c.customer\_id = o.customer\_id

WHERE o.order\_amount > (

SELECT AVG(order\_amount)

FROM order

)

)

>UNION

>SELECT c.customer\_id, c.customer\_name, o.order\_id, o.order\_amount

FROM customer AS c

INNER JOIN order AS o ON c.customer\_id = o.customer\_id

WHERE o.order\_amount > (

SELECT AVG(order\_amount)

FROM order

);