# RKY Cohort 11.0 SQL Assignment – Queries with Results

## Task 1: Agents’ names and phone numbers

**Query:**

SELECT agent\_name, phone\_number  
FROM agents;

**Result:**

|  |  |
| --- | --- |
| agent\_name | phone\_number |
| Andrew | 07635971237 |
| Helena | 07436432564 |
| Rickey | 07436437644 |
| Melvin | 07871432564 |
| Kris | 07436432564 |

## Task 2: Customer names where working\_area = 'Leeds'

**Query:**

SELECT cust\_name  
FROM customer  
WHERE working\_area = 'Leeds';

**Result:**

|  |
| --- |
| cust\_name |
| Angelo |
| Claire |

## Task 3: Order number & amount where order\_description = 'shoes'

**Query:**

SELECT order\_num, order\_amount  
FROM orders  
WHERE order\_description = 'shoes';

**Result:**

|  |
| --- |
| No rows returned |

## Task 4: agent\_code, order\_amount, order\_description where order\_amount = 3000 AND order\_description = 'clothes'

**Query:**

SELECT agent\_code, order\_amount, order\_description  
FROM orders  
WHERE order\_amount = 3000  
 AND order\_description = 'clothes';

**Result:**

|  |
| --- |
| No rows returned |

## Task 5: All columns from orders where order\_amount > 2000

**Query:**

SELECT \*  
FROM orders  
WHERE order\_amount > 2000;

**Result:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| order\_num | order\_amount | advance\_amount | order\_date | cust\_code | agent\_code | order\_description |
| 2 | 3000 | 2000 | 2021-08-10 | 2 | A01 | shirt |

## Task 6: Customer name & phone where the name contains the letter 'e'

**Query:**

SELECT cust\_name, phone\_number  
FROM customer  
WHERE cust\_name LIKE '%e%';

**Result:**

|  |  |
| --- | --- |
| cust\_name | phone\_number |
| Pedro | 07425686433 |
| Werner | 07135646752 |
| Claire | 071356467525 |

## Task 7: Row(s) in orders with the minimum order\_amount

**Query:**

SELECT \*  
FROM orders  
WHERE order\_amount = (SELECT MIN(order\_amount) FROM orders);

**Result:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| order\_num | order\_amount | advance\_amount | order\_date | cust\_code | agent\_code | order\_description |
| 1 | 1000 | 500 | 2021-10-04 | 1 | A04 | shoe |

## Task 8: Total order amount from orders

**Query:**

SELECT SUM(order\_amount) AS total\_order\_amount  
FROM orders;

**Result:**

|  |
| --- |
| total\_order\_amount |
| 4000 |

## Task 9: Number of distinct agents that currently have orders (from orders)

**Query:**

SELECT COUNT(DISTINCT agent\_code) AS agent\_count\_with\_orders  
FROM orders;

**Result:**

|  |
| --- |
| agent\_count\_with\_orders |
| 2 |

## Task 10: Highest purchase (order) amount per customer and overall highest

**Query:**

Per-customer max  
SELECT cust\_code, MAX(order\_amount) AS max\_order\_amount  
FROM orders  
GROUP BY cust\_code;  
  
Overall max  
SELECT MAX(order\_amount) AS highest\_order\_amount  
FROM orders;

**Result:**

|  |  |
| --- | --- |
| cust\_code | max\_order\_amount |
| 1 | 1000 |
| 2 | 3000 |
| Overall highest | 3000 |

## Task 11: Highest order amount on '2022-07-13' with agent\_code (from orders)

**Query:**

SELECT TOP 1 agent\_code, order\_amount  
FROM orders  
WHERE order\_date = '2022-07-13'  
ORDER BY order\_amount DESC;

**Result:**

|  |
| --- |
| No rows returned |

## Task 12: Names & customer city where customers and their agents work in the same city

**Query:**

SELECT c.cust\_name, c.cust\_city  
FROM customer AS c  
JOIN agents AS a ON a.agent\_code = c.agent\_code  
WHERE c.working\_area = a.working\_area;

**Result:**

|  |
| --- |
| No rows returned |

## Task 13: All customer names with the agent names who work for them

**Query:**

SELECT c.cust\_name, a.agent\_name  
FROM customer AS c  
JOIN agents AS a ON a.agent\_code = c.agent\_code  
ORDER BY c.cust\_name;

**Result:**

|  |  |
| --- | --- |
| cust\_name | agent\_name |
| Angelo | Kris |
| Claire | Kris |
| Curt | Helena |
| Pedro | Melvin |
| Werner | Andrew |

## Task 14: Orders where the customer is not located in the same city as the agent’s working area

**Query:**

SELECT o.order\_num, o.order\_amount, o.order\_date, o.order\_description,  
 c.cust\_name, c.cust\_city, a.agent\_name, a.working\_area  
FROM orders AS o  
JOIN customer AS c ON c.cust\_code = o.cust\_code  
JOIN agents AS a ON a.agent\_code = o.agent\_code  
WHERE c.cust\_city <> a.working\_area;

**Result:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| order\_num | order\_amount | order\_date | order\_description | cust\_name | cust\_city | agent\_name | working\_area |
| 1 | 1000 | 2021-10-04 | shoe | Pedro | York | Melvin | Birmingham |
| 2 | 3000 | 2021-08-10 | shirt | Curt | Wales | Andrew | London |

## Task 15: All orders issued by agent 'Rickey' (from orders)

**Query:**

SELECT \*  
FROM orders  
WHERE agent\_code = (  
 SELECT agent\_code FROM agents WHERE agent\_name = 'Rickey'  
);

**Result:**

|  |
| --- |
| No rows returned |