

Quinn Aho - hm4719

Muhammed Eliwat - hk8198

<https://drive.google.com/file/d/12DndgA18eiCjO-hmMxHynPwKWw2yGLO2/view?usp=sharing>

**SQL.txt:**

-- 1. Frequent clients - List the clients who completed the most service orders.

```
SELECT c.client_id,  
       c.first_name,  
       c.last_name,  
       c.email,  
       COUNT(*) AS completed_orders  
FROM clients c  
JOIN orders o ON c.client_id = o.client_id  
WHERE o.status = 'completed'  
GROUP BY c.client_id  
ORDER BY completed_orders DESC;
```

-- 2. Uncommitted clients - List clients who submitted 3 or more requests but never completed an order.

```
SELECT c.client_id,  
       c.first_name,  
       c.last_name,  
       c.email,  
       COUNT(sr.request_id) AS request_count  
FROM clients c  
JOIN service_requests sr ON c.client_id = sr.client_id  
LEFT JOIN orders o ON sr.request_id = o.request_id AND o.status = 'completed'  
GROUP BY c.client_id  
HAVING COUNT(sr.request_id) >= 3 AND SUM(CASE WHEN o.order_id IS NOT NULL THEN  
1 ELSE 0 END) = 0;
```

-- 3. This month's accepted quotes - List all quotes agreed upon in a given month.

```
SELECT q.quote_id,  
       q.request_id,  
       q.quoted_price,  
       q.scheduled_date,  
       q.scheduled_start_time,  
       q.scheduled_end_time,  
       q.notes,  
       q.created_at,  
       c.first_name,  
       c.last_name  
FROM quotes q
```

```
JOIN service_requests sr ON q.request_id = sr.request_id
JOIN clients c ON sr.client_id = c.client_id
WHERE q.status = 'accepted'
      AND MONTH(q.created_at) = ? -- provide numeric month
      AND YEAR(q.created_at) = ?; -- provide numeric year
```

-- 4. Prospective clients - List clients who registered but never submitted any request.

```
SELECT c.client_id,
       c.first_name,
       c.last_name,
       c.email,
       c.created_at
FROM clients c
LEFT JOIN service_requests sr ON c.client_id = sr.client_id
WHERE sr.request_id IS NULL;
```

-- 5. Largest job - List the service requests with the largest number of rooms ever completed.

```
SELECT sr.request_id,
       sr.client_id,
       sr.number_of_rooms,
       sr.service_address,
       sr.cleaning_type,
       sr.preferred_date,
       sr.preferred_time,
       c.first_name,
       c.last_name
FROM service_requests sr
JOIN orders o ON sr.request_id = o.request_id AND o.status = 'completed'
JOIN clients c ON sr.client_id = c.client_id
WHERE sr.number_of_rooms = (
    SELECT MAX(sr2.number_of_rooms)
    FROM service_requests sr2
    JOIN orders o2 ON sr2.request_id = o2.request_id AND o2.status = 'completed'
);
```

-- 6. Overdue bills - List all unpaid bills older than one week.

```
SELECT b.bill_id,
       b.order_id,
       b.client_id,
       b.amount,
       b.discount,
       b.final_amount,
       b.status,
       b.due_date,
```

```

        b.created_at,
        c.first_name,
        c.last_name
FROM bills b
JOIN clients c ON b.client_id = c.client_id
WHERE b.status <> 'paid'
      AND b.due_date < DATE_SUB(CURDATE(), INTERVAL 7 DAY);

```

-- 7. Bad clients - List clients who never paid any overdue bill.

```

SELECT c.client_id,
       c.first_name,
       c.last_name,
       c.email
FROM clients c
JOIN bills b ON c.client_id = b.client_id
LEFT JOIN payments p ON b.bill_id = p.bill_id
WHERE b.status <> 'paid'
      AND b.due_date < DATE_SUB(CURDATE(), INTERVAL 7 DAY)
GROUP BY c.client_id
HAVING SUM(CASE WHEN p.payment_id IS NOT NULL THEN 1 ELSE 0 END) = 0;

```

-- 8. Good clients - List clients who always paid their bills within 24 hours of being generated.

```

WITH first_payments AS (
  SELECT bill_id, MIN(paid_at) AS first_paid_at
  FROM payments
  GROUP BY bill_id
)
SELECT c.client_id,
       c.first_name,
       c.last_name,
       c.email
FROM clients c
JOIN bills b ON c.client_id = b.client_id
JOIN first_payments fp ON b.bill_id = fp.bill_id
GROUP BY c.client_id
HAVING SUM(
  CASE
    WHEN TIMESTAMPDIFF(HOUR, b.created_at, fp.first_paid_at) <= 24 THEN 0
    ELSE 1
  END
) = 0;

```