BealtyNaitNet SQL queries

Phase ll

Team: Linda & Bao Thinh Diep

Part C

**Query 1: JOIN of at least 3 tables (use JOIN ON)**

*Goal: Display customer appointments with assigned staff and service details*

*Display: Customer names, appointment dates, staff names, and service information*

SELECT c.first\_name AS customer\_first\_name,

c.last\_name AS customer\_last\_name,

a.appointment\_date,

a.start\_time,

s.first\_name AS staff\_first\_name,

s.last\_name AS staff\_last\_name,

s.position,

a.status,

a.total\_amount

FROM CUSTOMER c

JOIN APPOINTMENT a ON c.customer\_id = a.customer\_id

JOIN STAFF s ON a.staff\_id = s.staff\_id

ORDER BY a.appointment\_date, a.start\_time;

A screenshot of a computer

Description automatically generated

**Query 2: Nested queries with IN/ANY/ALL + GROUP BY**

*Goal: Find customers who have appointments with staff that have commission rates above average*

*Display: Customer names who worked with high-commission staff members*

SELECT c.first\_name, c.last\_name, COUNT(a.appointment\_id) AS appointment\_count FROM CUSTOMER c

JOIN APPOINTMENT a

ON c.customer\_id = a.customer\_id

WHERE a.staff\_id IN (

SELECT s.staff\_id

FROM STAFF s

WHERE s.commission\_rate > (SELECT AVG(commission\_rate) FROM STAFF)

)

GROUP BY c.customer\_id, c.first\_name, c.last\_name

HAVING COUNT(a.appointment\_id) >= 1

ORDER BY appointment\_count DESC;

A screenshot of a computer

Description automatically generated

**Query 3: Correlated nested query with aliasing**

*Goal: Find staff members who have appointments scheduled (correlated with outer query) Display: Staff names who have at least one appointment.*

SELECT s.first\_name, s.last\_name, s.position,

(SELECT COUNT(\*) FROM APPOINTMENT a1

WHERE a1.staff\_id = s.staff\_id) as appointment\_count

FROM STAFF s

WHERE EXISTS (SELECT 1

FROM APPOINTMENT a2

WHERE a2.staff\_id = s.staff\_id)

ORDER BY appointment\_count DESC;

**Query 4: FULL OUTER JOIN**

*Goal: Show all customers and all staff, including those without appointments*

*Diaplay: Complete list showing everyone (customers and staff) regardless of appointment status*

SELECT 'Customer' as person\_type, c.first\_name, c.last\_name, a.appointment\_date, a.start\_time, a.status

FROM CUSTOMER c

LEFT JOIN APPOINTMENT a ON c.customer\_id = a.customer\_id

--above is part one shows all customers that include have and have not appointment. (no appointment show customer + NULL)

UNION -- combine part 1 and part 3

SELECT 'Staff' as person\_type, s.first\_name, s.last\_name, a.appointment\_date, a.start\_time, a.status

FROM STAFF s

LEFT JOIN APPOINTMENT a ON s.staff\_id = a.staff\_id WHERE s.staff\_id

--above is part 2 that left join for stadd who do not have any appointment

NOT IN (SELECT DISTINCT staff\_id FROM APPOINTMENT)

ORDER BY person\_type, first\_name;

A screenshot of a computer

Description automatically generated

**Query 5: Set operations (UNION/EXCEPT/INTERSECT)**

*Goal: Combine active customers and active staff into one contact list*

*Display: Complete contact list showing all active people (customers and staff)*

SELECT 'Customer' as contact\_type, first\_name, last\_name, phone, email, 'N/A' as position FROM CUSTOMER WHERE is\_active = 1

UNION

SELECT 'Staff' as contact\_type, first\_name, last\_name, phone, email, position

FROM STAFF WHERE is\_active = 1

ORDER BY contact\_type, last\_name;

A screenshot of a computer

Description automatically generated