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

**Query 6: non-trivial SQL query (must use at least two tables in FROM clause)**

*Goal: Show staff members that have the most positive reviews from customers*

*Display: List of staff* *members that have the most numbers of maximum rating in the review*

SELECT staff.staff\_id, first\_name, last\_name, COUNT(\*) AS five\_star\_reviews

FROM review, staff

WHERE rating = 5

AND review.staff\_id = staff.staff\_id

GROUP BY staff\_id

HAVING COUNT(\*) = (

SELECT MAX(staff\_review\_count)

FROM (

SELECT COUNT(\*) AS staff\_review\_count

FROM review

WHERE rating = 5

GROUP BY staff\_id

) AS counts

);

A screenshot of a computer

AI-generated content may be incorrect.

**Query 7: non-trivial SQL query (must use at least two tables in FROM clause)**

*Goal: Show the most favorable staff member of the nail spa*

*Display: List of staff* *members that have the greatest number of appointments with customers*

SELECT staff.staff\_id, first\_name, last\_name, COUNT(\*) AS appointments

FROM appointment, staff

WHERE appointment.staff\_id = staff.staff\_id

GROUP BY staff\_id

HAVING COUNT(\*) = (

SELECT MAX(appointment\_count)

FROM (

SELECT COUNT(\*) AS appointment\_count

FROM appointment

GROUP BY staff\_id

) AS counts

);

A screenshot of a computer

AI-generated content may be incorrect.

**Query 8: non-trivial SQL query (must use at least two tables in FROM clause)**

*Goal: Display customers who gave very negative feedback about the nail salon's service.*

*Display: List of customers who gave a rating below 3 in the reviews and the staff members who associated with those reviews.*

SELECT r.appointment\_id, r.rating, c.customer\_id, c.first\_name, c.last\_name, s.staff\_id, s.first\_name, s.last\_name

FROM review as r, customer as c, staff as s

WHERE r.customer\_id = c.customer\_id

AND r.staff\_id = s.staff\_id

AND rating < 3

ORDER BY rating ASC;

A screenshot of a computer

AI-generated content may be incorrect.

**Query 9: non-trivial SQL query (must use at least three tables in FROM clause)**

*Goal: Display the customer who brings the highest profit to the nail salon.*

*Display: List of customers who gave the most benefits to the nail salon and the amount of the money they contributed to the business.*

SELECT c.customer\_id, c.first\_name, c.last\_name, SUM(p.amount + p.tip\_amount) AS total\_payment

FROM customer as c, appointment as a, payment as p

WHERE c.customer\_id = a.customer\_id

AND a.appointment\_id = p.appointment\_id

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

HAVING SUM(p.amount + p.tip\_amount) = (

SELECT MAX(total\_amount) FROM (

SELECT SUM(p1.amount + p1.tip\_amount) AS total\_amount

FROM appointment as a1, payment as p1

WHERE a1.appointment\_id = p1.appointment\_id

GROUP BY a1.customer\_id

) AS totals

);

A screenshot of a computer

AI-generated content may be incorrect.

**Query 10: Non-trivial SQL query**

**o must use at least three tables in FROM clause**

**o must use aliasing or renaming at least once**

**throughout SQL query**

*Goal: Display the most important services of the nail salon.*

*Display: List of the services which bring the most benefit to the nail salon and the total money that they contributed to the business.*

SELECT s.service\_name, SUM(p.amount + p.tip\_amount) AS total\_payment

FROM service as s, appointment\_service as aps, appointment as a, payment as p

WHERE s.service\_id = aps.service\_id

AND aps.appointment\_id = a.appointment\_id

AND a.appointment\_id = p.appointment\_id

GROUP BY s.service\_name

HAVING SUM(p.amount + p.tip\_amount) = (

SELECT MAX(service\_total\_payment) FROM (

SELECT SUM(p1.amount + p1.tip\_amount) AS service\_total\_payment

FROM service as s1, appointment\_service as aps1, appointment as a1, payment as p1

WHERE s1.service\_id = aps1.service\_id

AND aps1.appointment\_id = a1.appointment\_id

AND p1.appointment\_id = a1.appointment\_id

GROUP BY s1.service\_name

) AS totals

);

A screenshot of a computer

AI-generated content may be incorrect.