

이 SQL 쿼리들은 다양한 조건에 따라 데이터를 그룹화하고 집계하는 방법을 보여주고 있습니다. 각 쿼리에 대해 설명을 추가하겠습니다.

1. 회원 테이블에서 전체 행 개수를 구하는 쿼리

```
SELECT COUNT(*),  
       COUNT(cust_id),  
       COUNT(city),  
       COUNT(region)  
FROM customer;
```

- `COUNT(*)` : 테이블의 모든 행을 계산.
- `COUNT(cust_id)` : `cust_id` 값이 `NULL`이 아닌 행만 계산.
- `COUNT(city)` , `COUNT(region)` : 각각 `city` 와 `region` 컬럼이 `NULL`이 아닌 행만 계산.

2. 회원 테이블에서 마일리지 총합, 평균, 최소, 최대 구하기

```
SELECT SUM(mileage),  
       AVG(mileage),  
       MIN(mileage),  
       MAX(mileage)  
FROM customer;
```

- `SUM(mileage)` : 모든 회원의 마일리지 총합.
- `AVG(mileage)` : 마일리지 평균.
- `MIN(mileage)` , `MAX(mileage)` : 각각 마일리지의 최소, 최대 값.

3. 도시가 서울인 사람들의 마일리지 총합, 평균, 최소, 최대 구하기

```
SELECT SUM(mileage),  
       AVG(mileage),  
       MIN(mileage),  
       MAX(mileage)  
FROM customer  
WHERE city LIKE '서울%';
```

- `city` 가 '서울'로 시작하는 고객만 필터링.
- 서울에 해당하는 고객들의 마일리지에 대한 집계.

4. 도시 별로 회원 수와 평균 마일리지 구하기

```
SELECT city,
       COUNT(*) AS customer_count,
       AVG(mileage) AS average_mileage
FROM customer
GROUP BY city;
```

- GROUP BY city: 도시 별로 그룹화.
- 각 도시별 고객 수와 마일리지 평균을 계산.

5. 직급별, 도시별로 회원 수와 평균 마일리지 구하기

```
SELECT contact_position,
       city,
       COUNT(*) AS customer_count,
       AVG(mileage) AS average_mileage
FROM customer
GROUP BY contact_position, city
ORDER BY contact_position, city;
```

- contact_position 과 city 별로 고객 수와 마일리지 평균을 그룹화.
- 결과는 contact_position 과 city 로 정렬되어 출력.

6. 도시별로 전체 고객 수 및 평균 마일리지 구하기 (고객 수가 10명 이상인 경우)

```
SELECT city,
       COUNT(*) AS customer_count,
       AVG(mileage) AS average_mileage
FROM customer
GROUP BY city
HAVING COUNT(*) >= 10;
```

- HAVING COUNT(*) >= 10 : 고객 수가 10명 이상인 도시만 선택.
- 각 도시별로 고객 수와 마일리지 평균을 계산.

7. 도시별로 마일리지 총합 구하기 (회사 이름에 'T'가 포함되고, 마일리지 총합이 100 이상인 경우)

```
SELECT city,
       SUM(mileage) AS total_mileage
FROM customer
WHERE company_name LIKE '%T%'
GROUP BY city
HAVING SUM(mileage) >= 100;
```

- company_name 에 'T'가 포함된 고객만 필터링.
- 각 도시별로 마일리지 총합이 100 이상인 경우만 선택.

8. 지역(region)이 NULL이 아닌 고객의 도시별 고객 수 및 평균 마일리지 구하기

```
SELECT city,
       COUNT(*) AS customer_count,
       AVG(mileage) AS average_mileage
FROM customer
WHERE region IS NOT NULL
GROUP BY city;
```

- region 이 NULL이 아닌 경우만 필터링.
- 각 도시별 고객 수와 마일리지 평균을 계산.

9. 직급이 '대표'인 경우의 직급별 도시별 고객 수 구하기

```
SELECT contact_position,
       city,
       COUNT(*) AS customer_count
FROM customer
WHERE contact_position LIKE '%대표%'
GROUP BY contact_position, city;
```

- 직급이 '대표'인 고객을 필터링하여 직급별 도시별로 고객 수를 계산.

10. 직급이 '대표'인 경우의 직급별 지역별 고객 수 구하기

```
SELECT region,
       COUNT(*) AS customer_count
FROM customer
WHERE contact_position LIKE '%대표%'
GROUP BY region;
```

- 직급이 '대표'인 고객을 필터링하고, 지역별로 고객 수를 계산.

11. 직원 이름을 콤마(,)로 연결하여 출력하기

```
SELECT GROUP_CONCAT(name, ', ') AS names
FROM employee;
```

- 모든 사원의 이름을 하나의 문자열로 연결하여 출력.

12. 회원 지역을 콤마(,)로 연결하여 출력하기

```
SELECT GROUP_CONCAT(DISTINCT region, ', ') AS regions
FROM customer;
```

- DISTINCT 를 사용하여 중복된 지역을 제거하고 콤마로 연결.

13. 도시별로 회사 이름을 콤마(,)로 연결하여 출력하기

```
SELECT city,
       GROUP_CONCAT(company_name, ', ') AS customer_companies
FROM customer
GROUP BY city;
```

- 각 도시별로 회사 이름을 콤마로 연결하여 출력.

14. 도시의 총 갯수와 중복을 제거한 도시 개수 출력하기

```
SELECT COUNT(city) AS total_city_count,
       COUNT(DISTINCT city) AS distinct_city_count
FROM customer;
```

- 모든 도시의 총 갯수와 중복 제거된 도시의 갯수를 출력.

15. 년도별 주문 갯수 출력하기

```
SELECT STRFTIME('%Y', order_date) AS order_year,
       COUNT(*) AS order_count
FROM orders
GROUP BY order_year;
```

- 주문 날짜를 기준으로 연도별 주문 수를 출력.

16. 년도별, 분기별 주문 갯수 출력하기

```
SELECT STRFTIME('%Y', order_date) AS order_year,
       (CAST((STRFTIME('%m', order_date) - 1) / 3 + 1 AS INTEGER)) AS
       quarter,
       COUNT(*) AS order_count
FROM orders
GROUP BY order_year, quarter
ORDER BY order_year, quarter;
```

- 연도 및 분기별로 주문 수를 계산하고 결과를 정렬.

17. 월별로 주문 갯수 구하기 (조건: required_date 가 shipped_date 보다 작은 경우)

```
SELECT STRFTIME('%m', order_date) AS order_month,
       COUNT(*) AS order_count
FROM orders
WHERE required_date < shipped_date
GROUP BY order_month
ORDER BY order_month;
```

- 주문이 출하된 월별로 주문 수를 계산 (출하일보다 요청일이 이전인 경우만 필터링).

18. 제품 중 '우유'가 포함된 상품의 재고 개수 구하기

```
SELECT product_name,  
       SUM(stock) AS total_stock  
FROM product  
WHERE product_name LIKE '%우유%'  
GROUP BY product_name;
```

- '우유'가 포함된 제품의 총 재고를 계산.

19. 마일리지가 100 이상인 VIP 고객과 일반 고객으로 나누어 회원 수와 평균 마일리지 구하기

```
SELECT CASE  
       WHEN mileage >= 100 THEN 'VIP고객'  
       ELSE '일반고객'  
END AS customer_type,  
COUNT(*) AS customer_count,  
AVG(mileage) AS average_mileage  
FROM customer  
GROUP BY customer_type;
```

- 마일리지를 기준으로 고객을 'VIP'와 '일반'으로 나누어 회원 수와 평균 마일리지를 출력.

이러한 쿼리는 데이터 집계, 필터링, 그룹화의 여러 기능을 SQLite에서 활용하는 방법을 보여줍니다.