**-- 테이블 생성**

CREATE TABLE customer (

customer\_id INT PRIMARY KEY,

name VARCHAR(50),

age INT,

email VARCHAR(50),

country VARCHAR(30),

balance DECIMAL(10, 2),

created\_at DATE

);

**-- 데이터 삽입 예제**

-- 기본 데이터 삽입

INSERT INTO customer (customer\_id, name, age, email, country, balance, created\_at)

VALUES (1, 'Alice', 25, 'alice@example.com', 'USA', 100.00, '2024-01-01');

-- 여러 데이터 삽입

INSERT INTO customer (customer\_id, name, age, email, country, balance, created\_at)

VALUES

(2, 'Bob', 30, 'bob@example.com', 'Canada', 250.00, '2024-02-01'),

(3, 'Charlie', 35, 'charlie@example.com', 'UK', 500.00, '2024-03-01');

**-- 데이터 조회**

-- 전체 고객 조회

SELECT \* FROM customer;

-- 특정 컬럼 조회

SELECT name, email FROM customer;

-- 특정 조건으로 조회 (국가가 'USA'인 고객)

SELECT \* FROM customer WHERE country = 'USA';

-- 나이가 30 이상인 고객 조회

SELECT \* FROM customer WHERE age >= 30;

-- 특정 이름을 가진 고객 조회 ('Alice')

SELECT \* FROM customer WHERE name = 'Alice';

-- 잔액이 500 이상인 고객을 조회하는 쿼리

SELECT name, balance FROM customer WHERE balance >=500;

-- 잔액이 100 이상이고 나이가 25 미만인 고객 조회

SELECT \* FROM customer WHERE balance >= 100 AND age < 25;

-- 고객 이름 오름차순 정렬

SELECT \* FROM customer ORDER BY name ASC;

SELECT \* FROM customer ORDER BY name ASC LIMIT 1;

-- 고객 잔액 내림차순 정렬

SELECT \* FROM customer ORDER BY balance DESC;

SELECT \* FROM customer ORDER BY name DESC LIMIT 1;

-- 상위 5명의 잔액이 높은 고객을 조회하는 쿼리

SELECT \* FROM customer ORDER BY balance DESC LIMIT 5;

-- 이름에 ‘LEE’가 포함된 고객 목록을 조회하는 쿼리

SELECT \* FROM customer WHRER name LIKE ‘%Lee%’;

-- 이메일 주소가 example 도메인을 가진 고객을 조회하는 쿼리

SELECT \* FROM customer WHERE email LIKE ‘%@example.com’;

-- 특정 국가에 있는 고객들 중 잔액이 가장 높은 고객 조회  
SELECT \* FROM customer WHERE country=’USA’ ORDER BY balance DESC LIMIT 1;

-- 나이가 18세 이하인 고객을 나이 기준으로 정렬하여 조회하는 쿼리

SELECT \* FROM customer WHERE age<=18 ORDER BY age;

-- 데이터 갱신

-- 특정 고객의 나이 수정 (고객 ID가 1인 경우)

UPDATE customer SET age = 26 WHERE customer\_id = 1;

-- 여러 컬럼 갱신 (ID가 2인 고객의 나이와 잔액 수정)

UPDATE customer SET age = 31, balance = 300.00 WHERE customer\_id = 2;

-- 모든 고객의 잔액 10% 증가

UPDATE customer SET balance = balance \* 1.1;

-- 특정 국가의 고객 잔액 초기화 (캐나다 고객)

UPDATE customer SET balance = 0 WHERE country = 'Canada';

**-- 데이터 삭제**

-- 특정 고객 삭제 (고객 ID가 3인 경우)

DELETE FROM customer WHERE customer\_id = 3;

-- 잔액이 0인 고객 삭제

DELETE FROM customer WHERE balance = 0;

-- 나이가 40 이상인 고객 삭제

DELETE FROM customer WHERE age >= 40;

-- 특정 국가의 고객 삭제 ('UK')

DELETE FROM customer WHERE country = 'UK';

-- 모든 데이터 삭제

DELETE FROM customer;

**-- 집계 함수**

-- 전체 고객 수 조회

SELECT COUNT(\*) FROM customer;

-- 국가별 고객 수 조회

SELECT country, COUNT(\*) FROM customer GROUP BY country;

-- 전체 잔액 합계 조회

SELECT SUM(balance) FROM customer;

-- 평균 잔액 조회

SELECT AVG(balance) FROM customer;

-- 최대 잔액 조회

SELECT MAX(balance) FROM customer;

-- 최소 잔액 조회

SELECT MIN(balance) FROM customer;

-- 나이가 30 이상인 고객의 평균 잔액 조회

SELECT AVG(balance) FROM customer WHERE age >= 30;

-- 잔액이 500 이하인 고객 수 조회

SELECT COUNT(\*) FROM customer WHERE balance <= 500;

-- 고객의 이름과 잔액을 출력하면서 잔액 100 추가

SELECT name, balance + 100 AS updated\_balance FROM customer;

-- 2024년에 생성된 고객 조회

SELECT \* FROM customer WHERE YEAR(created\_at) = 2024;

-- 국가별 고객 수를 조회하는 쿼리  
SELECT country, COUNT(\*) AS customer\_count FROM customer GROUP BY country;

-- 국가별 평균 잔액을 조회하는 쿼리

SELECT country, AVG(balance) AS avg\_balance FROM customer GROUP BY country;

-- 연령대별(10대, 20대 등) 고객 수를 조회하는 쿼리

SELECT FLOOR(age / 10) \* 10 AS age\_group, COUNT(\*) AS customer\_count FROM customer GROUP BY age\_group;

-- 특정 연령대(30대) 고객의 평균 잔액을 조회하는 쿼리

SELECT AVG(balance) AS avg\_balance FROM customer WHERE age BETWEEN 30 AND 39;

**-------- HAVING 조건 사용------**

: GROUP BY 절에 생산된 결과 값 중 원하는 조건에 부합하는 데이터만 보고자 할 때 사용.

-- 고객 수가 5명 이상인 국가를 조회하는 쿼리

SELECT country, COUNT(\*) AS customer\_count FROM customer GROUP BY country HAVING customer\_count >= 5;

-- 총 잔액이 1000 이상인 국가를 조회하는 쿼리

SELECT country, SUM(balance) AS total\_balance FROM customer GROUP BY country HAVING total\_balance >= 1000;

-- 20대 고객의 평균 잔액이 300 이상인 연령대 조회

SELECT FLOOR(age / 10) \* 10 AS age\_group, AVG(balance) AS avg\_balance

FROM customer

GROUP BY age\_group

HAVING age\_group = 20 AND avg\_balance >= 300;

-- 고객 수가 10명 이하인 국가 조회

SELECT country, COUNT(\*) AS customer\_count FROM customer GROUP BY country HAVING customer\_count <= 10;

--10대 고객의 총 잔액이 500 이하인 연령대 조회

SELECT FLOOR(age / 10) \* 10 AS age\_group, SUM(balance) AS total\_balance

FROM customer

GROUP BY age\_group

HAVING age\_group = 10 AND total\_balance <= 500;

--특정 국가에서 잔액이 1000 이상인 고객 수를 조회

SELECT country, COUNT(\*) AS high\_balance\_count

FROM customer

WHERE balance >= 1000

GROUP BY country;

-- 30대 고객 중 잔액이 500 이상인 고객 수 조회

SELECT FLOOR(age / 10) \* 10 AS age\_group, COUNT(\*) AS customer\_count

FROM customer

WHERE balance >= 500

GROUP BY age\_group

HAVING age\_group = 30;

--가입일이 2023년 이후인 고객이 있는 국가 조회

SELECT country, COUNT(\*) AS recent\_customers

FROM customer

WHERE created\_at > '2023-01-01'

GROUP BY country

HAVING recent\_customers > 0;

-- 조건에 따른 데이터 변환

-- 이름이 'a'로 시작하는 고객 조회

SELECT \* FROM customer WHERE name LIKE 'a%';

-- 이메일에 'example'이 포함된 고객 조회

SELECT \* FROM customer WHERE email LIKE '%example%';

-- 잔액이 200 ~ 400 사이인 고객 조회

SELECT \* FROM customer WHERE balance BETWEEN 200 AND 400;

**-- 날짜 관련 쿼리**

-- 특정 날짜 이후에 가입한 고객 조회 (2024-01-01 이후)

SELECT \* FROM customer WHERE created\_at > '2024-01-01';

-- 특정 날짜 이전에 가입한 고객 조회 (2024-06-01 이전)

SELECT \* FROM customer WHERE created\_at < '2024-06-01';

-- 고객을 가입 연도별로 그룹화하여 수 조회

SELECT YEAR(created\_at) AS year, COUNT(\*) FROM customer GROUP BY year;

-- 고객의 연령대를 10살 단위로 나누어 고객 수 조회

SELECT FLOOR(age / 10) \* 10 AS age\_group, COUNT(\*) FROM customer GROUP BY age\_group;

-- 국가별 고객의 평균 나이 조회

SELECT country, AVG(age) AS avg\_age FROM customer GROUP BY country;

-- 고객의 이름과 잔액을 출력하고 잔액에서 50을 차감한 값 추가

SELECT name, balance, balance - 50 AS balance\_after\_deduction FROM customer;

-- 잔액이 가장 높은 고객의 이름과 잔액 조회

SELECT name, balance FROM customer WHERE balance = (SELECT MAX(balance) FROM customer);

SELECT name, balance

FROM customer

ORDER BY balance DESC

LIMIT 1;

SELECT name, balance

FROM customer

GROUP BY name, balance

HAVING balance = (SELECT MAX(balance) FROM customer);

SELECT name, balance

FROM customer

WHERE balance >= ALL (SELECT balance FROM customer);

-- 잔액이 가장 낮은 고객의 이름과 잔액 조회

SELECT name, balance FROM customer WHERE balance = (SELECT MIN(balance) FROM customer);

-- 나이가 30 이상인 고객의 총 잔액 조회

SELECT SUM(balance) AS total\_balance FROM customer WHERE age >= 30;

-- 고객을 이메일 도메인별로 그룹화하고 각 도메인별 고객 수 조회

SELECT SUBSTRING\_INDEX(email, '@', -1) AS domain, COUNT(\*) AS customer\_count FROM customer GROUP BY domain;

-- 가입일이 가장 최근인 고객 조회

SELECT \* FROM customer WHERE created\_at = (SELECT MAX(created\_at) FROM customer);

-- 국가별로 잔액이 200 이상인 고객 수 조회

SELECT country, COUNT(\*) AS customers\_with\_high\_balance FROM customer WHERE balance >= 200 GROUP BY country;

-- 전체 고객의 평균 나이 조회

SELECT AVG(age) AS avg\_age FROM customer;

-- 나이가 가장 많은 고객의 이름과 나이 조회

SELECT name, age FROM customer WHERE age = (SELECT MAX(age) FROM customer);

-- 잔액이 300 이상인 고객을 잔액 기준 내림차순으로 정렬하여 조회

SELECT \* FROM customer WHERE balance >= 300 ORDER BY balance DESC;

-- 모든 고객의 이름을 대문자로 변환하여 조회

SELECT UPPER(name) AS name\_upper FROM customer;

-- 고객의 이메일 주소에서 도메인 부분만 추출하여 조회

SELECT name, SUBSTRING\_INDEX(email, '@', -1) AS domain FROM customer;

-- 잔액이 500 이상인 고객의 수 조회

SELECT COUNT(\*) AS high\_balance\_customers FROM customer WHERE balance >= 500;

-- 평균 잔액보다 잔액이 높은 고객 조회

SELECT \* FROM customer WHERE balance > (SELECT AVG(balance) FROM customer);

-- 고객의 이름을 역순으로 정렬하여 조회

SELECT \* FROM customer ORDER BY name DESC;

-- 잔액이 0이 아닌 고객의 수 조회

SELECT COUNT(\*) AS non\_zero\_balance\_customers FROM customer WHERE balance > 0;

-- 잔액이 100 이하인 고객의 이름과 잔액을 조회

SELECT name, balance FROM customer WHERE balance <= 100;

-- 나이가 20 이상 30 이하인 고객의 이름과 나이를 조회

SELECT name, age FROM customer WHERE age BETWEEN 20 AND 30;

-- 국가별 고객의 최대 잔액을 조회

SELECT country, MAX(balance) AS max\_balance FROM customer GROUP BY country;

-- 고객을 연도별로 그룹화하고 각 연도별로 가입한 고객 수 조회

SELECT YEAR(created\_at) AS year, COUNT(\*) AS customer\_count FROM customer GROUP BY year;

-- 나이가 30 이상인 고객을 잔액 기준으로 내림차순 정렬하여 조회

SELECT \* FROM customer WHERE age >= 30 ORDER BY balance DESC;

-- 이메일이 example.com 도메인인 고객 수 조회

SELECT COUNT(\*) FROM customer WHERE email LIKE '%@example.com';

-- 국가별 고객의 평균 잔액 조회 (잔액이 100 이상인 경우만)

SELECT country, AVG(balance) AS avg\_balance FROM customer WHERE balance >= 100 GROUP BY country;

-- 가장 오래된 가입일을 가진 고객 조회

SELECT \* FROM customer WHERE created\_at = (SELECT MIN(created\_at) FROM customer);

-- 잔액이 100 이하인 고객의 이름과 잔액을 조회 (오름차순 정렬)

SELECT name, balance FROM customer WHERE balance <= 100 ORDER BY balance ASC;

-- 고객의 이름에 'e'가 포함된 고객 조회

SELECT \* FROM customer WHERE name LIKE '%e%';

-- 국가별 고객의 잔액 합계 조회

SELECT country, SUM(balance) AS total\_balance FROM customer GROUP BY country;

-- 고객 이름의 길이와 함께 이름을 조회

SELECT name, LENGTH(name) AS name\_length FROM customer;

-- 국가가 'USA'이거나 'Canada'인 고객 조회

SELECT \* FROM customer WHERE country IN ('USA', 'Canada');

-- 국가별로 가장 나이가 많은 고객 조회

SELECT country, MAX(age) AS max\_age FROM customer GROUP BY country;

-- 고객의 이름을 3글자로 잘라서 조회

SELECT name, LEFT(name, 3) AS short\_name FROM customer;

-- 이메일이 'gmail.com' 도메인인 고객 조회

SELECT \* FROM customer WHERE email LIKE '%@gmail.com';

-- 가입일이 현재 날짜와 차이가 30일 이하인 고객 조회

SELECT \* FROM customer WHERE DATEDIFF(CURDATE(), created\_at) <= 30;

-- 고객을 이름의 알파벳 순으로 그룹화하고 각 그룹의 고객 수 조회

SELECT LEFT(name, 1) AS name\_initial, COUNT(\*) FROM customer GROUP BY name\_initial;

-- 국가별 고객의 평균 잔액이 200 이상인 국가 조회

SELECT country FROM (SELECT country, AVG(balance) AS avg\_balance FROM customer GROUP BY country) AS country\_avg WHERE avg\_balance >= 200;

-- 국가별 고객의 최소 나이 조회

SELECT country, MIN(age) AS min\_age FROM customer GROUP BY country;

-- 전체 고객의 평균 가입일을 조회

SELECT DATE\_FORMAT(AVG(UNIX\_TIMESTAMP(created\_at)), '%Y-%m-%d') AS avg\_join\_date FROM customer;

-- 잔액이 높은 순으로 상위 5명의 고객 조회

SELECT \* FROM customer ORDER BY balance DESC LIMIT 5;

-- 고객의 이름을 소문자로 변환하여 조회

SELECT LOWER(name) AS name\_lower FROM customer;

-- 이메일이 'example.com'이 아닌 고객 조회

SELECT \* FROM customer WHERE email NOT LIKE '%@example.com';

-- 나이가 20대인 고객 조회

SELECT \* FROM customer WHERE age BETWEEN 20 AND 29;

-- 고객 수가 가장 많은 국가 조회

SELECT country FROM customer GROUP BY country ORDER BY COUNT(\*) DESC LIMIT 1;

-- 고객의 가입 년도별 평균 잔액 조회

SELECT YEAR(created\_at) AS join\_year, AVG(balance) AS avg\_balance FROM customer GROUP BY join\_year;

-- 이름에 공백이 포함된 고객 조회

SELECT \* FROM customer WHERE name LIKE '% %';

-- 전체 고객의 총 잔액과 평균 잔액을 동시에 조회

SELECT SUM(balance) AS total\_balance, AVG(balance) AS avg\_balance FROM customer;

-- 고객의 이메일 주소에서 사용자명만 추출하여 조회

SELECT SUBSTRING\_INDEX(email, '@', 1) AS username FROM customer;

-- 국가별 고객의 최대 나이와 최소 나이 조회

SELECT country, MAX(age) AS max\_age, MIN(age) AS min\_age FROM customer GROUP BY country;

-- 가장 자주 등장하는 국가 조회

SELECT country FROM customer GROUP BY country ORDER BY COUNT(\*) DESC LIMIT 1;

-- 고객의 잔액을 10단위로 나누어 그룹화하고 각 그룹별 고객 수 조회

SELECT FLOOR(balance / 10) \* 10 AS balance\_group, COUNT(\*) FROM customer GROUP BY balance\_group;

-- 나이와 잔액의 합이 100 이상인 고객 조회

SELECT \* FROM customer WHERE (age + balance) >= 100;

-- 국가별 고객의 평균 나이와 평균 잔액 조회

SELECT country, AVG(age) AS avg\_age, AVG(balance) AS avg\_balance FROM customer GROUP BY country;

-- 가입일이 가장 오래된 고객의 나이를 조회

SELECT age FROM customer WHERE created\_at = (SELECT MIN(created\_at) FROM customer);

-- 고객의 이름과 이름의 마지막 글자 조회

SELECT name, RIGHT(name, 1) AS last\_char FROM customer;

-- 고객의 평균 나이보다 나이가 많은 고객의 이름과 나이 조회

SELECT name, age FROM customer WHERE age > (SELECT AVG(age) FROM customer);

-- 잔액이 평균 잔액보다 높은 고객 수가 5명 이상인 국가 조회

SELECT country, COUNT(\*) AS above\_avg\_customers

FROM customer

WHERE balance > (SELECT AVG(balance) FROM customer)

GROUP BY country

HAVING above\_avg\_customers >= 5;

**-- 서브쿼리**

-- 전체 평균 잔액보다 잔액이 높은 고객 조회

SELECT \* FROM customer WHERE balance > (SELECT AVG(balance) FROM customer);

-- 잔액이 가장 높은 고객 조회

SELECT \* FROM customer WHERE balance = (SELECT MAX(balance) FROM customer);

-- 평균보다 나이가 많은 고객 조회

SELECT \* FROM customer WHERE age > (SELECT AVG(age) FROM customer);

**-- 그룹화와 집계**

-- 국가별 평균 잔액 조회

SELECT country, AVG(balance) FROM customer GROUP BY country;

-- 국가별 최대 잔액 조회

SELECT country, MAX(balance) FROM customer GROUP BY country;

-- 국가별 최소 잔액 조회

SELECT country, MIN(balance) FROM customer GROUP BY country;

-- 나이대별 고객 수 조회 (10살 단위)

SELECT FLOOR(age / 10) \* 10 AS age\_group, COUNT(\*) FROM customer GROUP BY age\_group;

-- 잔액이 100 이상인 고객의 국가별 수

SELECT country, COUNT(\*) FROM customer WHERE balance >= 100 GROUP BY country;

--고객 테이블에서 잔액이 전체 평균 잔액보다 높은 고객을 조회하는 쿼리

SELECT \* FROM customer

WHERE balance > (SELECT AVG(balance) FROM customer);

--각 고객의 잔액이 해당 국가의 평균 잔액보다 높은 고객을 조회하는 쿼리

SELECT \* FROM customer AS c

WHERE balance > (

SELECT AVG(balance) FROM customer WHERE country = c.country

);

-각 국가에서 잔액이 가장 높은 고객을 조회하는 쿼리

SELECT \* FROM customer

WHERE (country, balance) IN (

SELECT country, MAX(balance) FROM customer GROUP BY country

);

- 최근에 가입한 고객 중 잔액이 가장 높은 고객을 조회하는 쿼리

SELECT \* FROM customer

WHERE created\_at = (SELECT MAX(created\_at) FROM customer)

ORDER BY balance DESC

LIMIT 1;

- 30대 고객 중 잔액이 가장 높은 고객을 조회하는 쿼리

SELECT \* FROM customer

WHERE age BETWEEN 30 AND 39

ORDER BY balance DESC

LIMIT 1;

- 이메일 도메인(@example.com)을 기준으로 고객을 그룹화하고 각 도메인별 고객수를 조회하는 쿼리

SELECT SUBSTRING\_INDEX(email, '@', -1) AS domain, COUNT(\*) AS customer\_count

FROM customer

GROUP BY domain;

- 각 연령대별 최고 잔액을 가진 고객을 조회하는 쿼리(연령대를 10살 단위로 구분)

SELECT age DIV 10 \* 10 AS age\_group, MAX(balance) AS max\_balance

FROM customer

GROUP BY age\_group;

-각 국가에서 잔액이 두 번째로 높은 고객을 조회하는 쿼리

SELECT country, name, balance

FROM (

SELECT country, name, balance,

ROW\_NUMBER() OVER (PARTITION BY country ORDER BY balance DESC) AS rank

FROM customer

) AS ranked\_customers

WHERE rank = 2;

-특정 국가에서 평균 잔액이 전체 평균 잔액보다 높은 국가 목록 조회

SELECT country

FROM customer

GROUP BY country

HAVING AVG(balance) > (SELECT AVG(balance) FROM customer);

-고객별 나이가 평균 나이 이상인 경우 ‘성인’, 미만인 경우 ‘청소년’으로 분류하는 쿼리

SELECT name, age,

CASE WHEN age >= (SELECT AVG(age) FROM customer) THEN '성인'

ELSE '청소년' END AS age\_group

FROM customer;

(윈도우 함수)

-각 고객의 잔액이 고객 목록에서 몇 번째로 높은지 순위를 매기는 쿼리

SELECT name, balance,

RANK() OVER (ORDER BY balance DESC) AS balance\_rank

FROM customer;

-각 국가별로 나이가 가장 많은 고객의 이름과 나이를 조회하는 쿼리

SELECT country, name, age

FROM (

SELECT country, name, age,

RANK() OVER (PARTITION BY country ORDER BY age DESC) AS age\_rank

FROM customer

) AS ranked\_customers

WHERE age\_rank = 1;

-각 고객의 총 잔액 대비 비율을 계산하는 쿼리

SELECT name, balance,

ROUND(balance / SUM(balance) OVER () \* 100, 2) AS balance\_percentage

FROM customer;

-고객 목록에서 각 고객의 잔액이 전체 평균에서 얼마나 벗어낫는지 계산하는 쿼리

SELECT name, balance,

balance - AVG(balance) OVER () AS deviation\_from\_avg

FROM customer;

- 각 연령대별 누적 잔액을 계산하는 쿼리(연령대를 10살 단위로 구분)

SELECT age DIV 10 \* 10 AS age\_group, name, balance,

SUM(balance) OVER (PARTITION BY age DIV 10 \* 10 ORDER BY age) AS cumulative\_balance

FROM customer;

-각 고객의 이전 고객과의 잔액 차이를 계산하는 쿼리

SELECT name, balance,

LAG(balance) OVER (ORDER BY customer\_id) AS previous\_balance,

balance - LAG(balance) OVER (ORDER BY customer\_id) AS balance\_difference

FROM customer;

-고객 목록에서 현재 고객의 잔액이 최근 3명의 고객의 평균 잔액보다 높은지 여부를 계산하는 쿼리

SELECT name, balance,

CASE WHEN balance > AVG(balance) OVER (ORDER BY customer\_id ROWS BETWEEN 2 PRECEDING AND CURRENT ROW)

THEN 'Above Average'

ELSE 'Below Average' END AS balance\_comparison

FROM customer;

- 국가별로 가입일이 가장 오래된 고객의 가입일을 조회하는 쿼리

SELECT country, MIN(created\_at) OVER (PARTITION BY country) AS oldest\_customer\_date

FROM customer;

-고객의 잔액이 해당 국가의 전체 잔액에서 차지하는 비율을 계산하는 쿼리

SELECT country, name, balance,

ROUND(balance / SUM(balance) OVER (PARTITION BY country) \* 100, 2) AS country\_balance\_percentage

FROM customer;

-국가별 평균 잔액보다 높은 고객 목록 조회

SELECT country, name, balance

FROM customer

WHERE balance > AVG(balance) OVER (PARTITION BY country);

(트랜잭션 및 데이터 수정)

- 트랜잭션을 사용하여 특정 고객의 잔액을 증가시키고, 잔액 증가 후 조건에 맞지 않으면 롤백하는 쿼리

START TRANSACTION;

UPDATE customer SET balance = balance + 100 WHERE customer\_id = 1;

-- 조건 확인

IF (SELECT balance FROM customer WHERE customer\_id = 1) > 1000 THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

- 모든 고객의 이메일 도메인을 일괄 변경하는 쿼리

UPDATE customer

SET email = CONCAT(SUBSTRING\_INDEX(email, '@', 1), '@newdomain.com');

- 잔액이 500 이상인 고객에게 5%의 보너스를 지급하는 쿼리

UPDATE customer

SET balance = balance \* 1.05

WHERE balance >= 500;

-특정 국가의 모든 고객 잔액을 초기화하는 쿼리

UPDATE customer

SET balance = 0

WHERE country = 'Canada';

-30대 고객을 삭제하는 쿼리

DELETE FROM customer

WHERE age BETWEEN 30 AND 39;

-customer 테이블에서 이름이 동일한 고객 중 잔액이 가장 낮은 고객만 남기고 삭제하는 쿼리

DELETE FROM customer

WHERE customer\_id NOT IN (

SELECT customer\_id FROM (

SELECT customer\_id, MIN(balance)

FROM customer

GROUP BY name

) AS min\_balance\_customers

);

-잔액이 없는 고객을 삭제하는 쿼리

DELETE FROM customer WHERE balance = 0;

- 특정 국가의 고객 잔액을 전체 평균 잔액으로 업데이트하는 쿼리

UPDATE customer

SET balance = (SELECT AVG(balance) FROM customer)

WHERE country = 'USA';

- 모든 고객의 이름을 대문자로 변경하는 쿼리

UPDATE customer

SET name = UPPER(name);

특정 고객 ID에 해당하는 고객의 모든 정보를 복사하여 새 레코드로 추가하는 쿼리

INSERT INTO customer (name, age, email, country, balance, created\_at)

SELECT name, age, email, country, balance, created\_at

FROM customer

WHERE customer\_id = 1;

--고급 쿼리--

(CTE와 윈도우 함수)

- 최근 가입한 상위 5명의 고객을 조회하고, 가입일 순으로 순위를 매겨서 보여주는 쿼리

WITH Recent\_Customers AS (

SELECT \*, RANK() OVER (ORDER BY created\_at DESC) AS rank

FROM customer

)

SELECT \* FROM Recent\_Customers

WHERE rank <= 5;

- 각 국가별 평균 나이와 평균 잔액을 계산하고, 전체 평균과 비교하는 쿼리

WITH Country\_Averages AS (

SELECT country, AVG(age) AS avg\_age, AVG(balance) AS avg\_balance

FROM customer

GROUP BY country

)

SELECT c.country, c.avg\_age, c.avg\_balance,

(c.avg\_age - (SELECT AVG(age) FROM customer)) AS age\_deviation,

(c.avg\_balance - (SELECT AVG(balance) FROM customer)) AS balance\_deviation

FROM Country\_Averages AS c;

각 고객의 잔액이 자신의 연령대에서 몇 번째로 높은지 순위를 매기는 쿼리

SELECT name, age, balance,

RANK() OVER (PARTITION BY age DIV 10 ORDER BY balance DESC) AS age\_group\_rank

FROM customer;

- 국가별 잔액 상위 3명의 고객을 조회하는 쿼리

WITH Country\_Rank AS (

SELECT country, name, balance,

RANK() OVER (PARTITION BY country ORDER BY balance DESC) AS balance\_rank

FROM customer

)

SELECT \* FROM Country\_Rank

WHERE balance\_rank <= 3;

각 국가별 평균 잔액보다 높은 잔액을 가진 고객 목록 조회

WITH Country\_Average\_Balance AS (

SELECT country, AVG(balance) AS avg\_balance

FROM customer

GROUP BY country

)

SELECT c.name, c.country, c.balance

FROM customer AS c

JOIN Country\_Average\_Balance AS cab ON c.country = cab.country

WHERE c.balance > cab.avg\_balance;

최근 6개월 이내에 가입한 고객들의 총 잔액과 평균 잔액을 계산하는 쿼리

SELECT SUM(balance) AS total\_balance, AVG(balance) AS avg\_balance

FROM customer

WHERE created\_at >= DATE\_SUB(CURDATE(), INTERVAL 6 MONTH);

각 국가에서 가입 후 평균 잔액이 증가한 고객 수 조회

WITH Customer\_Initial\_Balance AS (

SELECT customer\_id, balance AS initial\_balance, created\_at, country

FROM customer

WHERE created\_at = (SELECT MIN(created\_at) FROM customer)

),

Balance\_Increase AS (

SELECT c.customer\_id, c.country, c.balance, cib.initial\_balance

FROM customer AS c

JOIN Customer\_Initial\_Balance AS cib ON c.customer\_id = cib.customer\_id

WHERE c.balance > cib.initial\_balance

)

SELECT country, COUNT(\*) AS balance\_increase\_count

FROM Balance\_Increase

GROUP BY country;

고객의 이름과 이메일에서 공통 문자열을 기준으로 상관 관계를 계산하는 쿼리

SELECT c1.name AS name1, c2.name AS name2,

LENGTH(SUBSTRING\_INDEX(c1.email, '@', 1)) - LENGTH(REPLACE(c1.email, SUBSTRING\_INDEX(c2.email, '@', 1), '')) AS common\_chars

FROM customer c1

JOIN customer c2 ON c1.customer\_id < c2.customer\_id;

국가별 잔액이 가장 높은 고객과 가장 낮은 고객을 동시에 조회하는 쿼리

SELECT country, MAX(balance) AS max\_balance, MIN(balance) AS min\_balance

FROM customer

GROUP BY country;

각 연령대별 평균 잔액을 계산하고, 전체 평균 잔액보다 낮은 연령대만 조회하는 쿼리

WITH Age\_Group\_Avg AS (

SELECT age DIV 10 \* 10 AS age\_group, AVG(balance) AS avg\_balance

FROM customer

GROUP BY age\_group

)

SELECT age\_group, avg\_balance

FROM Age\_Group\_Avg

WHERE avg\_balance < (SELECT AVG(balance) FROM customer);  
  
(트랜잭션 및 데이터 수정)

특정 국가의 고객 잔액을 증가시키되, 조건을 만족하지 않으면 트랜잭션을 롤백하는 쿼리

START TRANSACTION;

UPDATE customer

SET balance = balance + 100

WHERE country = 'USA';

-- 조건을 확인합니다.

IF (SELECT COUNT(\*) FROM customer WHERE balance > 1000 AND country = 'USA') = 0 THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

이메일 도메인을 기준으로 고객 목록을 분류하여 각 도메인별 잔액 총합을 계산하는 쿼리

SELECT SUBSTRING\_INDEX(email, '@', -1) AS domain, SUM(balance) AS total\_balance FROM customer GROUP BY domain;

모든 고객의 나이를 연령대별로 변경하여 새 테이블에 삽입하는 트랜잭션 쿼리

START TRANSACTION; CREATE TABLE customer\_age\_group AS SELECT name, age DIV 10 \* 10 AS age\_group, balance, country FROM customer; COMMIT;

특정 국가의 모든 고객을 삭제하는 트랜잭션 쿼리 (예: UK)

START TRANSACTION;

DELETE FROM customer WHERE country = 'UK';

COMMIT;

특정 고객의 잔액을 복사하여 새 고객으로 추가하는 쿼리

INSERT INTO customer (name, age, email, country, balance, created\_at)

SELECT name, age, CONCAT(email, '.new'), country, balance, created\_at

FROM customer

WHERE customer\_id = 1;