

1. 문자열 길이 세기

```
SELECT LENGTH('HELLO') AS length_hello
      ,LENGTH('HELLO') AS char_length_hello
      ,LENGTH('안녕') AS length_annyung
      ,LENGTH('안녕') AS char_length_annyung;
```

- LENGTH 함수는 문자열의 길이를 계산하는 함수입니다.
 - 'HELLO'는 5자로 구성되어 있으며, 한글 '안녕'은 UTF-8에서 6바이트로 계산됩니다.

2. 문자열 붙이기

```
SELECT 'DREAMS' || 'COME' || 'TRUE' AS concatenated
      , '2023' || '-' || '01' || '-' || '29' AS date_with_dash;
```

- 문자열을 이어 붙으려면 || 연산자를 사용합니다.
 - DREAMS, COME, TRUE 를 합쳐서 DREAMSCOMETRUE 로 만듭니다.
 - 2023, 01, 29 를 '-'로 구분하여 날짜 형식으로 출력합니다.

3. 문자열 자르기

```
SELECT SUBSTR('SQL 수업', 1, 3) AS left_string
      ,SUBSTR('SQL 수업', -4) AS right_string
      ,SUBSTR('SQL 수업', 2, 5) AS substr_2_5
      ,SUBSTR('SQL 수업', 2) AS substr_2;
```

- SUBSTR 함수는 문자열을 특정 위치에서 자릅니다.
 - SUBSTR('SQL 수업', 1, 3) 은 첫 번째 문자부터 3글자를 잘라냅니다.
 - SUBSTR('SQL 수업', -4) 는 뒤에서 네 번째부터 끝까지 잘라냅니다.

4. 문자열 패딩하기

```
SELECT printf('%10s', 'SQL') AS lpad_string
      ,printf('%-5s', 'SQL') || '*' AS rpad_string;
```

- SQLite에서는 문자열을 패딩할 수 있는 직접적인 함수가 없지만, printf 함수를 사용하여 자리를 맞추고 패딩 효과를 줄 수 있습니다.
 - %10s 는 오른쪽 정렬, %-5s 는 왼쪽 정렬을 의미합니다.

5. 문자열 트림하기

```
SELECT LENGTH(LTRIM(' SQL ')) AS length_trim_leading
,LENGTH(RTRIM(' SQL ')) AS length_trim_trailing
,LENGTH(TRIM(' SQL ')) AS length_trim_both;
```

- LTRIM, RTRIM, TRIM 함수는 각각 왼쪽, 오른쪽, 양쪽에서 공백을 제거하는 함수입니다.
 - 이 예제는 공백을 제거한 후의 문자열 길이를 확인합니다.

6. 문자열 위치 찾기

```
SELECT INSTR('SQL JAVA RUST', 'RUST') AS field_position
,INSTR('SQL,JAVA,RUST', 'RUST') AS find_in_set
,INSTR('너의 인생을 살아라', '인생') AS instr_position;
```

- INSTR 함수는 특정 문자열이 처음 나타나는 위치를 반환합니다.
 - 'RUST' 는 첫 번째 예제에서 9번째에 위치하고, 두 번째 예제에서도 9번째입니다.
 - '인생'은 세 번째 문자열에서 4번째 위치에 나타납니다.

7. 숫자 올림, 내림, 반올림, 자르기

```
SELECT CEIL(123.56) AS ceiling_value
,FLOOR(123.56) AS floor_value
,ROUND(123.56) AS rounded_value
,ROUND(123.56, 1) AS rounded_value_1;
```

- CEIL 은 올림, FLOOR 는 내림, ROUND 는 반올림을 수행합니다.
 - ROUND(123.56, 1) 은 소수점 첫째 자리에서 반올림합니다.

8. 현재 날짜 및 시간

```
SELECT DATETIME('now') AS current_datetime
,DATE('now') AS current_date
,TIME('now') AS current_time;
```

- SQLite에서 DATETIME('now') 는 현재 날짜와 시간을 출력하며, DATE('now') 는 현재 날짜만, TIME('now') 는 현재 시간만 출력합니다.

9. CASE WHEN

```
SELECT CASE WHEN 12500 * 450 > 5000000 THEN '초과달성' ELSE '미달성' END AS
achievement_status;
```

- CASE WHEN 은 조건에 따라 서로 다른 결과를 출력하는 SQL 구문입니다.

- 이 경우, $12500 * 450$ 이 5000000을 초과하면 '초과달성', 그렇지 않으면 '미달성'을 출력합니다.

10. NULLIF, COALESCE

```
SELECT COALESCE(1, 0) AS ifnull_1
      , COALESCE(NULL, 0) AS ifnull_null;
```

- COALESCE 함수는 인수 중 첫 번째로 NULL 이 아닌 값을 반환합니다.
 - COALESCE(1, 0) 은 1을 반환하고, COALESCE(NULL, 0) 은 0을 반환합니다.

문제 1

```
SELECT company_name
      , SUBSTR(company_name, 3) AS cust_company2
      , phone
      , REPLACE(SUBSTR(phone, 2), '(', '-') AS phone_number2
FROM customer;
```

- 고객사의 회사명에서 세 번째 글자부터 추출하고, 전화번호에서 괄호를 하이픈으로 대체하여 출력합니다.

문제 2

```
SELECT order_id
      , unit_price * quantity AS order_amount
      , ROUND((unit_price * quantity * discount), 0) AS discount_amount
      , (unit_price * quantity) - ROUND((unit_price * quantity * discount),
0) AS final_order_amount
FROM order_details;
```

- 주문 내역에서 총액을 계산하고, 할인 금액과 최종 금액을 계산합니다.

문제 3

```
SELECT name
      , birth_date
      , STRFTIME('%Y', 'now') - STRFTIME('%Y', birth_date) AS age
      , hire_date
      , JULIANDAY('now') - JULIANDAY(hire_date) AS days_since_hired
      , DATE(hire_date, '+500 days') AS days_after_500
FROM employee;
```

- 직원의 나이와 고용된 날짜로부터 경과한 일수 및 500일 후의 날짜를 계산합니다.

문제 4

```
SELECT contact_name
      ,company_name
      ,city
      ,CASE WHEN city LIKE '%특별시' OR city LIKE '%광역시' THEN '대도시' ELSE
'도시' END AS city_type
      ,mileage
      ,CASE WHEN mileage >= 200 THEN 'VVIP 고객'
              WHEN mileage >= 100 THEN 'VIP 고객'
              ELSE '일반 고객'
      END AS mileage_category
FROM customer;
```

- 고객이 대도시에 거주하는지 여부와 마일리지를 기준으로 고객 등급을 나눕니다.

문제 5

```
SELECT order_id
      ,cust_id
      ,order_date
      ,STRFTIME('%Y', order_date) AS order_year
      ,(CAST(STRFTIME('%m', order_date) AS INTEGER) - 1) / 3 + 1 AS
order_quarter
      ,STRFTIME('%m', order_date) AS order_month
      ,STRFTIME('%d', order_date) AS order_day
      ,STRFTIME('%w', order_date) AS order_weekday
      ,CASE STRFTIME('%w', order_date) WHEN '0' THEN 'Sunday'
                                         WHEN '1' THEN 'Monday'
                                         WHEN '2' THEN 'Tuesday'
                                         WHEN '3' THEN 'Wednesday'
                                         WHEN '4' THEN 'Thursday'
                                         WHEN '5' THEN 'Friday'
                                         ELSE 'Saturday' END AS order_week
FROM orders;
```

- 주문 날짜에서 연도, 분기, 월, 일, 요일을 추출하며, 요일을 이름으로 출력합니다.