

SQL 기초와 데이터 분석

하홍석

4. 다양한 SQL 함수 다루기

4. 다양한 SQL 함수 다루기

1. 타임스탬프 함수
2. 타입 변환
3. 조건문
4. 그 외 유용한 함수

타임스탬프 함수

타임스탬프 함수 : SQL에서 날짜와 시간 다루기

- 데이터타입
 - STRING : 'yyyy-mm-dd', 'yyyy-mm-dd HH:MM:SS'
 - DATE : yyyy-mm-dd
 - DATETIME : YYYY-MM-DD HH:MM:SS
 - TIMESTAMP : YYYY-MM-DD HH:MM:SS UTC

타임스탬프 함수 : 현재 시간

- NOW() : UTC 기준으로 현재 시간을 가져오는 함수
 - CURRENT_TIMESTAMP()
 - CURTIME() : 현재 시간 반환
 - CURRENT_DATE() = CURDATE() : yyyy-mm-dd 형식으로 반환
- SYSDATE() : 함수가 호출된 시간을 반환
- YEAR() : 날짜에서 연도 추출 / MONTH() : 날짜에서 월 추출 / DAY()
- HOUR() / MINUTE() / SECOND()
- WEEKDAY()
- MONTHNAME() / DAYNAME()

타임스탬프 함수 : 날짜 형식화

- STR_TO_DATE : 문자열 타입을 날짜 타입으로 변경
- DATE_FORMAT : 지정된 형식으로 날짜를 출력
 - %Y : 연도 (2023)
 - %y : 연도 (23)
 - %m : 월 (11)
 - %d : 일 (20)
 - %H : 시 (05), 24시간 형태
 - %T : hh:mm:ss
 - %s : 초
- 출처 : <https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>

타임스탬프 함수 : 날짜 연산

- ADDDATE() : 특정 interval 만큼 시간을 더함
 - DATE_ADD()
- SUBDATE() : 특정 interval 만큼 시간을 뺌
 - DATE_SUB()
- CONVERT_TZ() : 타임존 변경하여 출력
- DATEDIFF() : 두 날짜 간의 차이를 반환
- TIMEDIFF() : 두 시간 간의 차이를 반환
- TIME_TO_SEC() : 시간을 초 단위로 반환

타입 변환

타입 변환

- 필요성
 - 데이터 타입 불일치로 인한 연산/비교 오류를 피하기 위함
 - 다양한 데이터 소스 간의 호환성을 유지하기 위함
- CAST
- CONVERT

조건문

조건문 : IF

products			
product_id	category	name	price
0	키즈	어린이칫솔	1500
1	스포츠	손목보호대	10000
2	주방용품	밥그릇	2000
3	디지털	마우스	15000

RDBMS

Query

```
SELECT price,  
IF(price >= 10000, '고가', '저가') AS '가격구분'  
FROM product
```

price	가격구분
1500	저가
10000	고가
2000	저가
15000	고가

조건문 : IFNULL

orders_v3			
odr_index	product_id	user_name	date
0	3	철수	20231101
1	0	민수	20231103
2	3	갑동	20231103
3	2	갑순	20231104
4	1	갑순	NULL

RDBMS

Query

```
SELECT user_name, IFNULL(date, 20231104)  
as date  
FROM orders_v3
```

user_name	date
철수	20231101
민수	20231103
갑동	20231103
갑순	20231104
갑순	20231104

조건문 : CASE WHEN

products_B			
product_id	category	name	price
33	키즈	어린이칫솔	2000
34	스포츠	무릎보호대	13000
35	주방용품	국자	5000

RDBMS

Query

```
SELECT price,  
       CASE WHEN price > 10000 THEN '고가'  
            WHEN (price <= 10000 AND price >4000) THEN '중가'  
            ELSE '저가'  
       END AS price_class  
FROM products_B
```

price	price_class
2000	저가
13000	고가
5000	중가

조건문 : CASE WHEN

clicks

clk_index	product_id	user_name	date
0	0	영희	20231014
1	0	철수	20231014
2	0	영희	20231015
3	0	길순	20231016

RDBMS



Query

```
SELECT date,
CASE date
WHEN '20231014' THEN '첫째날'
WHEN '20231015' THEN '둘째날'
ELSE '기타'
END AS date_group
FROM clicks
```



date	date_group
20231014	첫째날
20231014	첫째날
20231015	둘째날
20231016	기타

그 외 유용한 함수

그 외 유용한 함수 : RANK()

- rank()
- dense_rank()
- percent_rank()

```
select age,  
rank() over (order by age) as asc_rank,  
rank() over (order by age desc) as desc_rank,  
dense_rank() over (order by age) as dense_rank,  
percent_rank() over (order by age) as percent_rank  
from Customers
```

- <https://www.programiz.com/sql/online-compiler/>

그 외 유용한 함수 : LEAD()

- lead()

```
SELECT
    clk_index,
    user_name,
    date,
    LEAD(clk_index, 1) OVER (
        PARTITION BY user_name, date
        ORDER BY clk_index )
    next_click
FROM
    clicks
ORDER BY 3, 1
```

그 외 유용한 함수 : LAG()

- lag()

```
SELECT
    clk_index,
    user_name,
    date,
    LAG(clk_index, 1) OVER (
        PARTITION BY user_name, date
        ORDER BY clk_index )
    previous_clk
FROM
    clicks
ORDER BY 3, 1
```

- <https://dev.mysql.com/doc/refman/8.0/en/window-function-descriptions.html>

End of Document