

# 10강

## 시간/날짜 및 기타함수





## 1. 시간/날짜 관련 함수들

함수	설명
<code>CURRENT_DATE()</code> , <code>CURDATE()</code>	현재 날짜 반환
<code>CURRENT_TIME()</code> , <code>CURTIME()</code>	현재 시간 반환
<code>CURRENT_TIMESTAMP()</code> , <code>NOW()</code>	현재 시간 날짜 반환

※ `CURDATE()` 현재 년-월-일을, `CURTIME()`는 현재 시:분:초를 구한다. `NOW()`는 현재 년-월-일 시:분:초를 구한다. 생성된 테이블에 자료를 삽입할 때 기본값을 현재 날짜나 현재시간으로 지정하여야 할 때 사용하면 유용하다.

### 코드 예제

```
public void insertReviewBoard(ReviewDTO rdto) {
    getConnect();
    try {
        String sql = "insert into review values(null,?,?,?,?,?,current_date(),0,?)";
```

	r_code	r_pw	p_code	r_title	r_content	m_name	r_date	r_readcount	m_id
▶	1	1234	37	생각보다 많이 가벼움	생각보다 많이 가벼움	오승현	2024-02-14	22	kjb1045
	3	1234	34	만족	조립도 어렵지 않고 플라스틱임에도...	홍길동	2024-02-14	10	kjb1030
	4	1234	56	만족	제품이 좋습니다. 잘 쓰고 있습니다.	홍길동	2024-02-14	9	kjb1030
	5	1234	83	만족	생각보다 사용하기 편리하네요	진달래	2024-02-14	16	tom9867
	6	1234	71	만족	서랍장 편리하네요	홍길동	2024-02-15	5	kjb1030
	7	1234	17	생각보다 무거워요	생각했던 것 보다 무거워서 사용하...	오승현	2024-02-16	16	kjb1045
	9	1234	87	너무 무거워요	생각 보다 무거워서 사용하기 불편 ...	진달래	2024-02-21	10	tom9867



## JSP - SQL 코드 예제


```
public void insertReviewBoard(ReviewDTO rdto) {
    getConnect();
    try {
        String sql = "insert into review values(null,?,?,?,?,?,current_Date(),0,?)";
        pstmt = con.prepareStatement(sql);
        pstmt.setString(1, rdto.getR_pw());
        pstmt.setInt(2, rdto.getP_code());
        pstmt.setString(3, rdto.getR_title());
        pstmt.setString(4, rdto.getR_content());
        pstmt.setString(5, rdto.getM_name());
        pstmt.setString(6, rdto.getM_id());
        pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if(con!=null) con.close();
            if(pstmt!=null) pstmt.close();
            if(rs!=null) rs.close();
        } catch (SQLException se){
            se.printStackTrace();
        }
    }
}
```



## UI 적용된 화면

## 상품 리뷰

상품 사용후기입니다.

번호	상품정보	제목	작성자	작성일	조회
5	 블랙서랍장 3단 35_두칸형 만족		홍길동	2024-02-15	9
4	 블렌딩 분리수거함 (스윙형 1단) 만족		진달래	2024-02-14	22
3	 파스텔 리빙박스 32L_블루 만족		홍길동	2024-02-14	10
2	 [모음전] 조립식 서랍장 (4층) 만족		홍길동	2024-02-14	12
1	 [모음전] 퓨어 서랍장 (8층) 생각보다 많이 가벼움		오승현	2024-02-14	27



```
select current_date(),curdate(),current_time(),curtime(),now() ,current_timestamp()
```



current_date()	curdate()	current_time()	curtime()	now()	current_timestamp()
2024-11-17	2024-11-17	11:40:17	11:40:17	2024-11-17 11:40:17	2024-11-17 11:40:17

함수	설명
<b>DATE(datetime표현식)</b>	DATETIME 표현식에서 년월일 정보만 추출하여 반환
<b>TIME(datetime표현식)</b>	DATETIME 표현식에서 시간 정보만 추출하여 반환

※ DATE( ), TIME( ) 함수 안에 DATETIME 표현식을 직접 입력 할 때는 반드시 문자형식으로 '2023-11-10' 또는 '20:10:10'으로 작성함에 주의하자!!

```
select date(now()),date('2025-11-21');
```

	date(now())	date('2025-11-21')
▶	2025-11-21	2025-11-21

```
select time(now()),time('18:10:10');
```

	time(now())	time('18:10:10')
▶	02:45:52	18:10:10



## 👉 여기서 잠깐 !!

greendb 스키마안의 <orders> 테이블의 order\_date의 날짜가 2024-01-7 ~ 2024-01-09 사이의 모든 데이터가 출력되도록 SQL문을 작성하여 아래 화면과 같은 결과를 도출하세요

(단, DATE, BETWEEN ~ AND 함수를 이용)

### 출력 결과

	order_id	customer_id	order_date	status
▶	7	7	2024-01-07	주문완료
	8	8	2024-01-08	배송완료
	9	9	2024-01-09	주문완료
*	NULL	NULL	NULL	NULL



함수	설명
<b>YEAR(datetime)</b>	주어진 DATETIME값의 <b>년도</b> 반환
<b>MONTHNAME(datetime)</b>	주어진 DATETIME값의 <b>월(영문)</b> 반환
<b>MONTH(datetime)</b>	주어진 DATETIME값의 <b>월</b> 반환
<b>WEEKDAY(datetime)</b>	주어진 DATETIME값의 <b>요일값</b> 반환( <b>월요일:0</b> )
<b>WEEK(datetime)</b>	주어진 DATETIME값의 <b>주수를</b> 반환
<b>DAYNAME(datetime)</b>	주어진 DATETIME값의 <b>요일명</b> 반환
<b>DAYOFMONTH, DAY(datetime)</b>	주어진 DATETIME값의 <b>날짜(일)</b> 반환

## SELECT

```

order_date,
YEAR(order_date) AS YEAR,
MONTHNAME(order_date) AS MONTHNAME,
MONTH(order_date) AS MONTH,
WEEKDAY(order_date) AS WEEKDAY,
DAYNAME(order_date) AS DAYNAME,
DAY(order_date) AS DAY
FROM orders;

```

	order_date	YEAR	MONTHNAME	MONTH	WEEKDAY	DAYNAME	DAY
▶	2024-01-01	2024	January	1	0	Monday	1
	2024-01-02	2024	January	1	1	Tuesday	2
	2024-01-03	2024	January	1	2	Wednesday	3
	2024-01-04	2024	January	1	3	Thursday	4
	2024-01-05	2024	January	1	4	Friday	5
	2024-01-06	2024	January	1	5	Saturday	6
	2024-01-07	2024	January	1	6	Sunday	7
	2024-01-08	2024	January	1	0	Monday	8
	2024-01-09	2024	January	1	1	Tuesday	9
	2024-01-10	2024	January	1	2	Wednesday	10
	2024-01-11	2024	January	1	3	Thursday	11
	2024-01-12	2024	January	1	4	Friday	12
	2024-01-13	2024	January	1	5	Saturday	13
	2024-01-14	2024	January	1	6	Sunday	14
	2024-01-15	2024	January	1	0	Monday	15



## 👉 여기서 잠깐 !!

<orders> 테이블의 order\_date의 날짜 형식을 '년/월/일 요일(영문)'의 형식으로 변경하여 NewDate 필드를 아래 출력 결과 화면과 같이 출력되도록 SQL문을 작성 하시오.

(단, YEAR, MONTH, DAY, DAYNAME, CONCAT, CONCAT\_WS, UPPER, LEFT 함수를 모두이용)

### 출력 결과

	order_id	NewDate
▶	1	2024/1/1 MON
	2	2024/1/2 TUE
	3	2024/1/3 WED
	4	2024/1/4 THU
	5	2024/1/5 FRI
	6	2024/1/6 SAT
	7	2024/1/7 SUN
	8	2024/1/8 MON
	9	2024/1/9 TUE
	10	2024/1/10 WED
	11	2024/1/11 THU
	12	2024/1/12 FRI
	13	2024/1/13 SAT
	14	2024/1/14 SUN
	15	2024/1/15 MON





함수	설명
<b>hour(datetime)</b>	주어진 DATETIME의 시 반환
<b>minute(datetime)</b>	주어진 DATETIME의 분 반환
<b>second(datetime)</b>	주어진 DATETIME의 초 반환

```
SELECT  
hour(now()), minute(now()), second(now());
```



Number of Records: 1		
hour(now())	minute(now())	second(now())
5	57	11



함수	설명
<b>ADDDATE(날짜, 차이), DATE_ADD</b>	날짜를 기준으로 차이를 더한 날짜를 구한다.
<b>SUBDATE(날짜, 차이), DATE_SUB</b>	날짜를 기준으로 차이를 뺀 날짜를 구한다.

SELECT

```
ADDDATE('2021-06-20', INTERVAL 1 YEAR),
ADDDATE('2021-06-20', INTERVAL -2 MONTH),
ADDDATE('2021-06-20', INTERVAL 3 WEEK),
ADDDATE('2021-06-20', INTERVAL -4 DAY),
ADDDATE('2021-06-20', INTERVAL -5 MINUTE),
ADDDATE('2021-06-20 13:01:12', INTERVAL 6 SECOND);
```

INTERVAL 범위의 간격을 의미 한다.



ADDDATE('2021-06-20', INTERVAL 1 YEAR)	ADDDATE('2021-06-20', INTERVAL -2 MONTH)	ADDDATE('2021-06-20', INTERVAL 3 WEEK)	ADDDATE('2021-06-20', INTERVAL -4 DAY)	ADDDATE('2021-06-20', INTERVAL -5 MINUTE)	ADDDATE('2021-06-20 13:01:12', INTERVAL 6 SECOND)
▶ 2022-06-20	2021-04-20	2021-07-11	2021-06-16	2021-06-19 23:55:00	2021-06-20 13:01:18



함수	설명
<b>DATEDIFF(날짜1, 날짜2), DATE_DIFF</b>	날짜1 - 날짜2의 일수를 결과로 구한다.
<b>TIMEDIFF(날짜1 또는 시간1, 날짜1 또는 시간2), TIME_DIFF</b>	시간1 - 시간2의 결과를 구한다.
<b>TIMESTAMPDIFF(단위, 날짜1, 날짜2)</b>	차이를 연,분기,월, 주,일,시,분,초를 지정하여 가져올 때

※ DATEDIFF (date1, date2)는 date1 - date2 의 '일수 차이'를 계산한다. 따라서 date1이 더 미래면 양수, date1이 더 과거면 음수가 나온다.

TIMEDIFF()는 시간1-시간2의 결과를 구한다. 이 외에도 차이를 연,분기,월,주,일,시,분,초 를 지정하여 가져올 때 사용하는 함수가 TIMESTAMPDIFF() 함수이다.

- SECOND : 초
- MINUTE : 분
- HOUR : 시
- DAY : 일
- WEEK : 주
- MONTH : 월
- QUARTER : 분기
- YEAR : 년

```
SELECT DATEDIFF('2024-12-29',NOW()),
TIMEDIFF('16:27:00','06:00:00'),
TIMESTAMPDIFF(QUARTER,'2024-1-1','2025-1-1'),
TIMESTAMPDIFF(MONTH,'2024-1-1','2025-1-1')
```



DATEDIFF('2024-12-29',NOW())	TIMEDIFF('16:27:00','06:00:00')	TIMESTAMPDIFF(QUARTER,'2024-1-1','2025-1-1')	TIMESTAMPDIFF(MONTH,'2024-1-1','2025-1-1')
29	10:27:00	4	12



```
SELECT
    shipment_id,
    tracking_number,
    DATEDIFF(shipped_date,current_date()) AS '오늘까지 일수'
FROM shipment;
```



shipped\_date가 미래일 때만 양수

	shipment_id	tracking_number	오늘까지 일수
▶	1	TRK-001	-689
	2	TRK-002	-688
	3	TRK-003	-687
	4	TRK-004	-686
	5	TRK-005	-685
	6	TRK-006	-684
	7	TRK-007	-683
	8	TRK-008	-682
	9	TRK-009	-681
	10	TRK-010	-680
	11	TRK-011	-679
	12	TRK-012	-678
	13	TRK-013	-677
	14	TRK-014	-676
	15	TRK-015	-675



함수	설명
<b>LAST_DAY(날짜)</b>	주어진 날짜의 마지막 날짜를 구한다.

※ LAST\_DAY()는 주어진 날짜의 마지막 날짜를 구한다. 주로 그 달이 몇 일까지 있는지 확인할 때 사용한다.

```
select order_id, order_date, last_day(order_date) from orders;
```



	order_id	order_date	last_day(order_date)
▶	1	2024-01-01	2024-01-31
	2	2024-01-02	2024-01-31
	3	2024-01-03	2024-01-31
	4	2024-01-04	2024-01-31
	5	2024-01-05	2024-01-31
	6	2024-01-06	2024-01-31
	7	2024-01-07	2024-01-31
	8	2024-01-08	2024-01-31
	9	2024-01-09	2024-01-31
	10	2024-01-10	2024-01-31
	11	2024-01-11	2024-01-31
	12	2024-01-12	2024-01-31
	13	2024-01-13	2024-01-31
	14	2024-01-14	2024-01-31
	15	2024-01-15	2024-01-31



## 2. 기타 함수

함수	설명
<code>IF(조건, TRUE, FALSE)</code>	조건이 참이라면 TRUE, 거짓이면 FALSE를 반환

```
SELECT if(100>200, '참이다','거짓이다')
```



```
if(100>200, '참이다','거짓이다')
거짓이다
```

```
select business_name as '가게명',
if(can_takeout=1, '포장가능', '포장불가') as '포장유무'
from businesses;
```

가게명	포장유무
▶ 화룡각	포장가능
철구분식	포장가능
알코렐라	포장가능
바른떡볶이	포장가능
북극냉면	포장불가
보쌈마니아	포장가능
에그사라다	포장가능
달다방	포장가능
마카오마카롱	포장가능
김밥마라	포장가능
소소스윗	포장가능
사사서서소소스시	포장가능
린민짬뽕	포장가능
파시조아	포장가능
할매장국	포장불가
노선이탈리아	포장가능
커피앤코드	포장가능
신림등백순대	포장가능



## ※ 보다 복잡한 조건은 **CASE(CASE ~ WHEN ~ ELSE ~ END)**문을 사용한다.

- WHEN - THEN은 항상 같이 사용되어야 한다.
- WHEN - THEN은 여러 개 사용이 가능하다.
- ELSE가 존재하면 모든 WHEN - THEN 조건이 TRUE(참)이 아닌 경우 ELSE의 결과값을 반환한다.
- ELSE가 없고, 조건이 TRUE(참)이 아니면 NULL을 반환한다.
- 반드시 END로 종료한다.

```
CASE WHEN 조건1 THEN 결과값1
      WHEN 조건2 THEN 결과값2
      WHEN 조건N THEN 결과값N
      ELSE 결과값
END
```

```
SELECT
CASE 10
  WHEN 1 THEN '일'
  WHEN 5 THEN '오'
  WHEN 10 THEN '십'
  ELSE '모름'
END AS PRINT;
```

※ CASE 뒤의 값이 10이므로 세 번째 WHEN이 수행되어 '십' 이 반환된다, 만약, 해당사항이 없다면 FALSE 부분이 반환된다.



**PRINT**

십



※ 보다 복잡한 조건은 **CASE**문을 사용하면 유용하다.

```
select business_name as '가게명',  
case status  
  when 'OPN' then '운영중'  
  when 'RMD' then '휴무'  
  when 'CLS' then '폐업'  
  else '점검'  
end as '영업상태'  
from businesses;
```



가게명	영업상태
화룡각	운영중
철구분식	운영중
알코렐라	휴무
바른떡볶이	운영중
북극냉면	운영중
보쌈마니아	운영중
에그사라다	점검
달다방	운영중
마카오마카롱	운영중
김밥마라	운영중
소소스윗	운영중
사사서셔소쇼스시	점검
린민짬뽕	폐업
파시조아	운영중
할매장국	폐업
노선이탈리아	운영중
커피앤코드	운영중
신림동백순대	점검





## 👉 여기서 잠깐 !!

greendb 스키마안의 < businesses > 테이블의 status, can\_takeout을 이용해  
다음 조건을 모두 적용하여 가게 상태를 표시하시오.

status = 'OPN' 이고 can\_takeout = 1 이면 '정상영업+포장가능'

status = 'OPN' 이고 can\_takeout = 0 이면 '정상영업+포장불가'

status = 'CLS' 이면 '폐업'

기타 상태는 '기타상태'

출력 결과 화면과 같이 출력되도록 SQL문을 작성하시오.

(단, CASE, IF 문을 이용)

### 출력 결과

	business_name	status	can_takeout	business_condition
▶	화룡각	OPN	1	정상영업+포장가능
	철구분식	OPN	1	정상영업+포장가능
	알코헬라	RMD	1	기타상태
	바른떡볶이	OPN	1	정상영업+포장가능
	북극냉면	OPN	0	정상영업+포장불가
	보쌈마니아	OPN	1	정상영업+포장가능
	에그사라다	VCT	1	기타상태
	달다방	OPN	1	정상영업+포장가능
	마카오마카롱	OPN	1	정상영업+포장가능
	김밥마라	OPN	1	정상영업+포장가능
	소소스윗	OPN	1	정상영업+포장가능
	사사서서소소소시	VCT	1	기타상태
	린민짬뽕	CLS	1	폐업
	파시조아	OPN	1	정상영업+포장가능
	할매장국	CLS	0	폐업
	노선이탈리아	OPN	1	정상영업+포장가능
	커피앤코드	OPN	1	정상영업+포장가능
	신림동백순대	VCT	1	기타상태



함수	설명
<b>IFNULL(수식1, 수식2)</b>	수식1이 NULL일 때 수식2 출력

※ IFNULL()은 수식1이 NULL이 아니면 수식1이 반환되고, 수식1이 NULL이면 수식2가 반환된다.

UI 적용된 화면

```
SELECT
  IFNULL('A', 'B'),
  IFNULL(NULL, 'B');
```



Number of Records: 1

IFNULL('A', 'B')	IFNULL(NULL, 'B')
A	B

## 구매 정보

고객님 상품구매가 완료 되었습니다.

수령인 오승현  
 구매일자 2024-11-30  
 구매코드 2024-11-30\_4

주문내역 조회

홈으로 이동

o_date	o_code	p_code	o_qty	o_total	m_id
2024-11-30	1	124	4	67,600	kjb1045
2024-11-30	2	124	1	16,900	kjb1045
2024-11-30	3	20	1	22,900	kjb1045
2024-11-30	4	55	3	50,700	kjb1045

코드 예제

```
public int codeaddSelect() {
    getConnect();
    int maxcode = 0;
    try {
        String sql = "select ifnull(max(o_code),0)+1 from orders where o_date = current_date()";
```

# 10강

## 연습문제





1. greendb 스키마안의 menus 테이블에서 칼로리 (kilocalories)가 500 이상이면 '고칼로리', 300 이상이면 '중칼로리', 그 외는 '저칼로리'라고 표시하여 메뉴 이름과 함께 출력되록 SQL문을 작성하시오. (단, if문 사용)

### <결과>

	menu_name	calorie_level
	아메리카노	저칼로리
	고르곤졸라피자	고칼로리
	보쌈	고칼로리
	장국	중칼로리
	까르보나라	고칼로리
	바닐라마카롱	저칼로리
	백순대	고칼로리
	마늘보쌈	고칼로리
	양념순대볶음	고칼로리
	단팥빵	저칼로리
	간짜장	고칼로리
	뚝카롱	저칼로리
	비빔냉면	고칼로리
	참치김밥	고칼로리
	치즈떡볶이	고칼로리
	플레인와플	저칼로리
	chap-sal-sundae	중칼로리
	yuk-gae-jang	중칼로리
	gukmul-tteokbokki	중칼로리
	don-gas-gimbap	고칼로리
	sam-seon-jjambong	고칼로리



1. greendb 스키마안의 모든 메뉴(menu)에 대해, 가격(price)에 따라 아래 조건으로 메뉴의 가격 등급을 표시되도록 SQL문을 작성하시오.

가격이 5,000원 미만이면 → '저가'

5,000 ~ 9,999원이면 → '중간'

10,000원 이상이면 → '고가'

(단, CASE WHEN 사용)

### <결과>

	menu_name	price	price_grade
▶	물냉면	9000	중간
	아메리카노	4500	저가
	고르곤졸라피자	12000	고가
	보쌈	14000	고가
	장국	8500	중간
	까르보나라	9000	중간
	바닐라마카롱	1500	저가
	백순대	11000	고가
	마늘보쌈	16000	고가
	양념순대볶음	12000	고가
	단팥빵	1500	저가
	간짜장	7000	중간
	똥카롱	2000	저가
	비빔냉면	9000	중간
	참치김밥	3000	저가
	치즈떡볶이	5500	중간
	플레인와플	6500	중간
	찹쌀순대	3000	저가