

2025년 상반기 K-디지털 트레이닝

SQL 고급 - 데이터타입, 내장함수

[KB] IT's Your Life

1 MySQL의 데이터 형식

✓ MySQL에서 지원하는 데이터 형식의 종류

- Data Type으로 표현
 - 데이터 형식, 데이터형, 자료형, 데이터 타입등 다양하게 불림
- 데이터 형식에 대한 이해가 필요한 이유
 - SELECT문 더욱 잘 활용
 - 테이블의 생성 효율적으로 하기 위해 필요
- MySQL에서 데이터 형식의 종류는 30개 정도
 - 중요하고 자주 쓰는 형식에 대해 중점 학습

1 MySQL의 데이터 형식

✔ 숫자 데이터 형식

데이터 형식	바이트 수	숫자 범위	설명
BIT(N)	N/8		1~64bit를 표현. b'0000' 형식으로 표현
TINYINT	1	-128~127	정수
★SMALLINT	2	-32,768~32,767	정수
MEDIUMINT	3	-8,388,608~8,388,607	정수
★INT INTEGER	4	약 -21억 ~ +21억	정수
★BIGINT	8	약 -900경 ~ +900경	정수
★FLOAT	4	-3.40E+38~-1.17E-38	소수점 아래 7자리까지 표현
DOUBLE REAL	8	-1.22E-308~1.79E+308	소수점 아래 15자리까지 표현
★DECIMAL(m,[d]) NUMERIC(m,[d])	5~17	$-10^{38}+1 \sim +10^{38}-1$	전체 자릿수(m)와 소수점 이하 자릿수(d)를 가진 숫자형 예) decimal(5,2)는 전체 자릿수를 5자리로 하되, 그 중 소수점 이하를 2자리로 하겠다는 의미

1 MySQL의 데이터 형식

☑ 문자 데이터 형식

문자 데이터는
캐릭터셋을 신경써야한다

데이터 형식		바이트 수	설명
★CHAR(n)		1~255	고정길이 문자형. n을 1부터 255까지 지정. character의 약자 그냥 CHAR만 쓰면 CHAR(1)과 동일
★VARCHAR(n)		1~65535	가변길이 문자형. n을 사용하면 1부터 65535 까지 지정. Variable character의 약자
BINARY(n)		1~255	고정길이의 이진 데이터 값
VARBINARY(n)		1~255	가변길이의 이진 데이터 값
TEXT 형식	TINYTEXT	1~255	255 크기의 TEXT 데이터 값
	TEXT	1~65535	N 크기의 TEXT 데이터 값
	MEDIUMTEXT	1~16777215	16777215 크기의 TEXT 데이터 값
	★LONGTEXT	1~4294967295	최대 4GB 크기의 TEXT 데이터 값
BLOB 형식	TINYBLOB	1~255	255 크기의 BLOB 데이터 값
	BLOB	1~65535	N 크기의 BLOB 데이터 값
	MEDIUMBLOB	1~16777215	16777215 크기의 BLOB 데이터 값
	★LONGBLOB	1~4294967295	최대 4GB 크기의 BLOB 데이터 값
ENUM(값들...)		1 또는 2	최대 65535개의 열거형 데이터 값
SET(값들...)		1, 2, 3, 4, 8	최대 64개의 서로 다른 데이터 값

1 MySQL의 데이터 형식

✓ 날짜와 시간 데이터 형식

시간 데이터를 다룰 때는 타임존이 어떻게 되느냐가 따짐
우리나라는 다행히 하나의 타임존만을 사용함

데이터 형식	바이트 수	설명
★DATE	3	날짜는 1001-01-01 ~ 9999-12-31까지 저장되며 날짜 형식만 사용 'YYYY-MM-DD' 형식으로 사용됨.
TIME	3	-838:59:59.000000 ~ 838:59:59.000000까지 저장되며, 'HH:MM:SS' 형식으 로 사용
★DATETIME	8	날짜는 1001-01-01 00:00:00 ~ 9999-12-31 23:59:59까지 저장되며 형식은 'YYYY-MM-DD HH:MM:SS' 형식으로 사용
TIMESTAMP	4	날짜는 1001-01-01 00:00:00 ~ 9999-12-31 23:59:59까지 저장되며 형식은 'YYYY-MM-DD HH:MM:SS' 형식으로 사용. time_zone 시스템 변수와 관련이 있으며 UTC 시간대 변환하여 저장
YEAR	1	1901 ~ 2155까지 저장. 'YYYY' 형식으로 사용

타임존과 무관한 타임스탬프를 이용해도 됨.

1 MySQL의 데이터 형식

✓ 기타 데이터 형식

데이터 형식	바이트 수	설명
★GEOMETRY	N/A	공간 데이터 형식으로 선, 점 및 다각형 같은 공간 데이터 개체를 저장하고 조작
★JSON	8	JSON(JavaScript Object Notation) 문서를 저장

✓ LONGTEXT, LONGBLOB

- LOB(Large Object, 대량의 데이터)을 저장하기 위해 LONGTEXT, LONGBLOB 데이터 형식 지원
- 지원되는 데이터 크기는 약 4GB의 파일을 하나의 데이터로 저장 가능
- LONGTEXT
 - ex) 장편소설과 같은 큰 텍스트 파일
- LONGBLOB
 - ex) 동영상 파일과 같은 큰 바이너리 파일

1 MySQL의 데이터 형식

✓ 데이터 형식과 형 변환

○ 데이터 형식 변환 함수

- CAST(), CONVERT() 함수를 가장 일반적으로 사용
- 데이터 형식 중에서 가능한 것은 BINARY, CHAR, DATE, DATETIME, DECIMAL, JSON, SIGNED INTEGER, TIME, UNSIGNED INTEGER
- 함수 사용법

형식:

CAST (expression AS 데이터형식 [(길이)])

CONVERT (expression , 데이터형식 [(길이)])

ex) sqlDB의 구매 테이블(buyTbl)에서 평균 구매 개수를 구하는 쿼리문

SELECT CAST(AVG(amount) AS SIGNED INTEGER) AS '평균 구매 개수' FROM buytbl ;

또는

SELECT CONVERT(AVG(amount) , SIGNED INTEGER) AS '평균 구매 개수' FROM buytbl ;



	평균 구매 개수
▶	3

1 MySQL의 데이터 형식

✓ 데이터 형식과 형 변환

○ 암시적인 형 변환

- CAST()나 CONVERT() 함수를 사용하지 않고 형이 변환되는 것

boolean 결과값 경우의수는
true false unknown

null에 대해서 모든 연산의
결과는 unknown이 나오거나
처리가 안됨

```
SELECT '100' + '200';      -- 문자와 문자를 더함(정수로 변환되서 연산됨)
SELECT CONCAT('100', '200'); -- 문자와 문자를 연결(문자로 처리)
SELECT CONCAT(100, '200');  -- 정수와 문자를 연결(정수가 문자로 변환되서 처리)
```

```
SELECT 1 > '2mega';        -- 정수인 2로 변환되어서 비교
SELECT 3 > '2MEGA';        -- 정수인 2로 변환되어서 비교
SELECT 0 = 'mega2';        -- 문자는 0으로 변환됨
```

'100' + '200'	CONCAT('100', '200')	CONCAT(100, '200')	1 > '2mega'	3 > '2MEGA'	0 = 'mega2'
▶ 300	▶ 100200	▶ 100200	▶ 0	▶ 1	▶ 1

1 MySQL 내장 함수

✓ 내장 함수

- 흐름 함수, 문자열 함수, 수학 함수, 날짜/시간 함수, 전체 텍스트 검색 함수, 형 변환 함수, XML 함수, 비트 함수, 보안/압축 함수, 정보 함수, 공간 분석 함수, 기타 함수 등

✓ 제어 흐름 함수 if문이 아닌 if함수다.

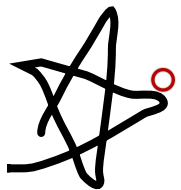
- 프로그램의 흐름 제어

- IF (수식, 참, 거짓)

- 수식이 참 또는 거짓인지 결과에 따라서 2중 분기

`SELECT IF (100>200, '참이다', '거짓이다');` 삼항연산자가 생각하는 문장

오라클에서는 NVL이죠?



- **IFNULL(수식1, 수식2)**

- 수식1이 NULL이 아니면 수식1이 반환되고 수식1이 NULL이면 수식2가 반환

수식은 보통 칼럼명

변수=변수?변수:다른변수;

1 MySQL 내장 함수

✓ 제어 흐름 함수

○ NULLIF(수식1, 수식2)

- 수식1과 수식2가 같으면 NULL을 반환, 다르면 수식1을 반환

○ CASE ~ WHEN ~ ELSE ~ END

- CASE는 내장 함수는 아니며 연산자(Operator)로 분류
- 다중 분기에 사용되므로 내장함수와 함께 알아두자

스위치문이 생각나는 함수

```
SELECT CASE 10
      WHEN 1 THEN '일'
      WHEN 5 THEN '오'
      WHEN 10 THEN '십'
      END AS 'CASE연습';
```

```
case 보통칼러명
when 값1 then 들어갈값1
when 값2 then 들어갈값2
when 값3 then 들어갈값3
when 값4 then 들어갈값4
else 들어갈값5
end as 칼러명별칭
```

- CASE 뒤의 값이 10이므로 세 번째 WHEN이 수행되어 '십' 반환
- 만약, 해당하는 사항이 없다면 ELSE 부분이 반환

1 MySQL 내장 함수

✓ 문자열 함수

- 문자열 조작, 활용도 높음
- ASCII (아스키 코드),
 - 문자의 아스키 코드값 반환
- CHAR(숫자)
 - 숫자의 아스키 코드값에 해당하는 문자 반환

```
SELECT ASCII('A'), CHAR(65);
```

- BIT_LENGTH(문자열), CHAR_LENGTH(문자열), LENGTH(문자열)
 - 할당된 Bit 크기 또는 문자 크기 반환
 - CHAR_LENGTH()는 문자의 개수 반환
 - LENGTH()는 할당된 Byte 수 반환

1 MySQL 내장 함수

✓ 문자열 함수

○ CONCAT(문자열1, 문자열2,...), CONCAT_WS(구분자, 문자열1, 문자열2,...)

- CONCAT() : 문자열을 이어줌
- CONCAT_WS() : 구분자와 함께 문자열을 이어주는 역할 ✓

`SELECT CONCAT_WS('/', '2025', '01', '01');`

- 2025/01/01 반환

1 MySQL 내장 함수

✓ 문자열 함수

○ GROUP_CONCAT(문자열 SEPERATOR ',')

- GROUPBY 항목의 개별 내용을 하나의 문자열로 연결
- 연결 시 지정한 구분자로 연결

```
SELECT u.userID, u.name, GROUP_CONCAT(b.prodName SEPARATOR ',') AS '상품목록'
FROM usertbl u
    LEFT OUTER JOIN buytbl b
    ON u.userID = b.userID
GROUP BY u.userID, u.name
ORDER BY u.userID;
```

	userID	name	상품목록
▶	BBK	바비킴	모니터,메모리,운동화,운동화
	EJW	은지원	책,청바지,책
	JKW	조관우	NULL
	JYP	조용필	모니터
	KBS	김범수	운동화,노트북,청바지
	KKH	김경호	NULL
	LJB	임재범	NULL
	LSG	이승기	NULL
	SSK	성시경	책
	YJS	윤종신	NULL

1 MySQL 내장 함수

✓ 문자열 함수

○ **ELT(위치, 문자열1, 문자열2, ...), FIELD(찾을 문자열, 문자열1, 문자열2, ...), FIND_IN_SET (찾을 문자열, 문자열 리스트), INSTR(기준 문자열, 부분 문자열), LOCATE(부분 문자열, 기준 문자열)**

- ✎ ▪ **ELT() : 위치 번째에 해당하는 문자열 반환**
- ✎ ▪ **FIELD() : 찾을 문자열의 위치를 찾아 반환, 없으면 0**
- ✎ ▪ **FIND_IN_SET() : 찾을 문자열을 문자열 리스트에서 찾아 위치 반환**
 - 문자열 리스트는 콤마(,)로 구분되어 있고 공백이 없어야 함
- **INSTR()는 기준 문자열에서 부분 문자열 찾아 그 시작 위치 반환**
- **LOCATE()는 INSTR()와 동일하지만 파라미터의 순서가 반대**

1 MySQL 내장 함수

☑ 문자열 함수



○ FORMAT(숫자, 소수점 자릿수)

- 숫자를 소수점 아래 자릿수까지 표현, 1,000단위마다 콤마 표시해 줌

○ BIN(숫자), HEX(숫자), OCT(숫자)

- 2진수, 16진수, 8진수의 값을 반환

○ INSERT(기준 문자열, 위치, 길이, 삽입할 문자열)

- 기준 문자열의 위치부터 길이만큼 지우고 삽입할 문자열 끼워 넣음

```
SELECT INSERT('abcdefghi', 3, 4, '@@@'), INSERT('abcdefghi', 3, 2, '@@@@');
```

- 'ab@@@@ghi'와 'ab@@@@efghi' 반환

1 MySQL 내장 함수

✓ 문자열 함수

- LEFT(문자열, 길이), RIGHT(문자열, 길이)
 - 왼쪽 또는 오른쪽에서 문자열의 길이만큼 반환

```
SELECT LEFT('abcdefghi', 3), RIGHT('abcdefghi', 3);
```

- 'abc'와 'ghi' 반환

- UPPER(문자열), LOWER(문자열)
 - 소문자를 대문자로, 대문자를 소문자로 변경
- LPAD(문자열, 길이, 채울 문자열), RPAD(문자열, 길이, 채울 문자열)
 - 문자열을 길이만큼 늘린 후에 빈 곳을 채울 문자열로 채움

1 MySQL 내장 함수

✓ 문자열 함수

○ LTRIM(문자열), RTRIM(문자열)

- 문자열의 왼쪽/오른쪽 공백을 제거,中间的 공백은 제거되지 않음

○ TRIM(문자열), TRIM(방향 자를_문자열 FROM 문자열) ✓

- TRIM(문자열)은 문자열의 앞뒤 공백을 모두 없앴
- TRIM(방향 자를_문자열 FROM 문자열) 에서 방향은 LEADING(앞), BOTH(양쪽), TRAILING(뒤) 으로 표시

```
SELECT TRIM('   이것이   '), TRIM(BOTH 'ㅋ' FROM 'ㅋㅋㅋ재밋어요.ㅋㅋㅋㅋ');
```

- '이것이'와 '재밋어요.' 반환

1 MySQL 내장 함수

✓ 문자열 함수

○ REPEAT(문자열, 횟수) ✓

- 문자열을 횟수만큼 반복

○ REPLACE(문자열, 원래 문자열, 바꿀 문자열) ✓

- 문자열에서 원래 문자열을 찾아서 바꿀 문자열로 바꿈

```
SELECT REPLACE('이것이 MySQL이다', '이것이', 'This is');
```

- 'This is MySQL이다' 반환

○ REVERSE(문자열)

- 문자열의 순서를 거꾸로 바꿈

1 MySQL 내장 함수

✓ 문자열 함수

○ SPACE(길이)

- 길이만큼의 공백을 반환

○ SUBSTRING(문자열, 시작위치, 길이) 또는 SUBSTRING(문자열 FROM 시작위치 FOR 길이)

- 시작위치부터 길이만큼 문자를 반환, 길이가 생략되면 문자열의 끝까지 반환

```
SELECT SUBSTRING('대한민국만세', 3, 2);
```

데이터베이스에서는 인덱스는 1로 시작함

|

- '민국' 반환

날짜 타입을 다룰때 sub string을 활용한다

○ SUBSTRING_INDEX(문자열, 구분자, 횟수)

- 문자열에서 구분자가 왼쪽부터 횟수 번째까지 나오면 그 이후의 오른쪽은 버림
- 횟수가 음수면 오른쪽부터 세고 왼쪽을 버림

1 MySQL 내장 함수

✓ 수학 함수

- ABS(숫자)
 - 숫자의 절댓값 계산
- ACOS(숫자), ASIN(숫자), ATAN(숫자), ATAN2(숫자1, 숫자2), SIN(숫자), COS(숫자), TAN(숫자)
 - 삼각 함수와 관련된 함수 제공
- CEILING(숫자), FLOOR(숫자), ROUND(숫자)
 - 올림, 내림, 반올림 계산
- CONV(숫자, 원래 진수, 변환할 진수)
 - 숫자를 원래 진수에서 변환할 진수로 계산
- DEGREES(숫자), RADIANS(숫자), PI ()
 - 라디안 값을 각도값으로, 각도값을 라디안 값으로 변환, PI()는 3.141592 반환
- EXP(X), LN(숫자), LOG(숫자), LOG(밑수, 숫자), LOG2(숫자), LOG10(숫자)
 - 지수, 로그와 관련된 함수 제공

1 MySQL 내장 함수

✓ 수학 함수

- MOD(숫자1, 숫자2) 또는 숫자1 % 숫자2 또는 숫자1 MOD 숫자2
 - 숫자1을 숫자2로 나눈 나머지 값을 구함
- POW(숫자1, 숫자2), SQRT(숫자)
 - 거듭제곱값 및 제곱근을 구함
- RAND()
 - RAND()는 0 이상 1 미만의 실수 구함
 - 'm ≤ 임의의 정수 < n'를 구하고 싶다면 FLOOR(m + (RAND() * (n-m))) 사용
- SIGN(숫자)
 - 숫자가 양수, 0, 음수인지 판별, 결과는 1, 0, -1 셋 중에 하나 반환
- TRUNCATE(숫자, 정수)
 - 숫자를 소수점을 기준으로 정수 위치까지 구하고 나머지는 버림

1 MySQL 내장 함수

☑ 날짜 및 시간 함수

- ADDDATE(날짜, 차이), SUBDATE(날짜, 차이)
 - 날짜를 기준으로 차이를 더하거나 뺀 날짜 구함
- ADDTIME(날짜/시간, 시간), SUBTIME(날짜/시간, 시간)
 - 날짜/시간을 기준으로 시간을 더하거나 뺀 결과를 구함
- ✓ ○ CURDATE(), CURTIME(), NOW(), SYSDATE() ☆
 - CURDATE() : 현재 연-월-일
 - CURTIME() : 현재 시 : 분 : 초
 - NOW(), SYSDATE() : 현재 '연-월-일 시 : 분 : 초'
- ✓ ○ YEAR(날짜), MONTH(날짜), DAY(날짜), HOUR(시간), MINUTE(시간), SECOND(시간), MICROSECOND(시간)
 - 날짜 또는 시간에서 연, 월, 일, 시, 분, 초, 밀리초 구함

ADDDATE('1998-01-02',INTERVAL 42 DAY);

ADDDATE('1998-01-02',INTERVAL 4 MONTH);

ADDTIME(now(),'1:00:00')

dayofweek('1998-07-29') 요일 1,,7

1 MySQL 내장 함수

✓ 날짜 및 시간 함수

○ DATE(), TIME()

- DATETIME 형식에서 연-월-일 및 시 : 분 : 초만 추출

TIMEDIFF('1998-07-29 19:00:02',
'2025-05-31 21:00:00')도 가능

○ DATEDIFF(날짜1, 날짜2), TIMEDIFF(날짜1 또는 시간1, 날짜1 또는 시간2)

- DATEDIFF()는 날짜1-날짜2의 일수를 결과로 구함

○ DAYOFWEEK(날짜), MONTHNAME(), DAYOFYEAR(날짜)

dayofweek('1998-07-29') 요일 1,,,7

- 요일(1:일, 2:월~7:토) 및 1년 중 몇 번째 날짜인지 구함

○ LAST_DAY(날짜)

- 주어진 날짜의 마지막 날짜를 구함 LAST_DAY('2018-09-01 1:00:02') => '2018-09-30'

1 MySQL 내장 함수

☑ 날짜 및 시간 함수

- **MAKEDATE(연도, 정수)** makedate(2020,32) => '2021-02-01'
 - 연도에서 정수만큼 지난 날짜 구함
- **MAKETIME(시, 분, 초)**
 - 시, 분, 초를 이용해서 '시 : 분 : 초'의 TIME 형식 만듦
- **PERIOD_ADD(연월, 개월수), PERIOD_DIFF(연월1, 연월2)**
 - PERIOD_ADD()는 연월에서 개월만큼의 개월이 지난 연월 구함 period_Add(201703,15)=>201806
 - PERIOD_DIFF()는 연월1-연월2의 개월수 구함
- **QUARTER(날짜)**
 - 날짜가 4분기 중에서 몇 분기인지를 구함
- **TIME_TO_SEC(시간)**
 - 시간을 초 단위로 구함