SQL 고급

MariaDB의 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	$-10^{38}+1$~$+10^{38}-1$	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

MariaDB의 데이터 형식

❖ 문자 데이터 형식

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 65535까지 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

MariaDB의 데이터 형식

❖ 날짜와 시간 데이터 형식

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' 형식으로 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 시간대로 변환하여 저장
YEAR	1	1901~2155까지 저장. 'YYYY' 형식으로 사용

```
SELECT CAST('2020-10-19 12:35:29.123' AS DATE) AS 'DATE' ;
SELECT CAST('2020-10-19 12:35:29.123' AS TIME) AS 'TIME' ;
SELECT CAST('2020-10-19 12:35:29.123' AS DATETIME) AS 'DATETIME' ;
```

MariaDB의 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

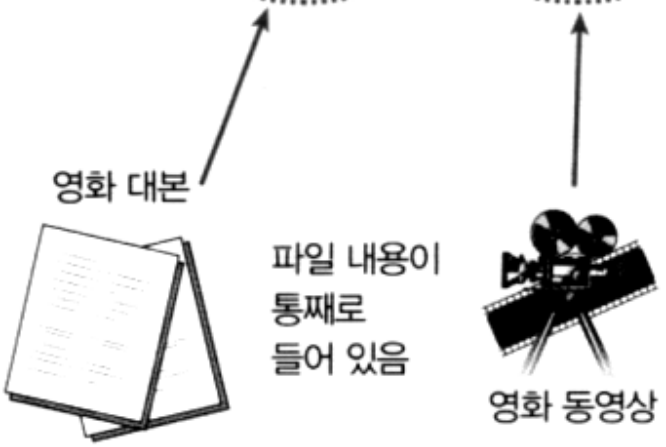
MariaDB의 데이터 형식

❖ LONGTEXT, LONGBLOB

- LOB: Large Object
- 4GB 크기의 파일을 하나의 데이터로 저장 가능
- LONGBLOG : 이미지, 동영상 데이터
- LONGTEXT : 소설, 게시판의 글 내용

영화 테이블

영화id	영화 제목	감독	주연배우	LONGTEXT 영화 대본	LONGBLOB 영화 동영상
0001	쉰들러리스트	스필버그	리암 니슨	#####	#####
0002	쇼생크탈출	프랭크다라본트	팀 로빈스	#####	#####
0003	라스트모히칸	마이클 만	다니엘 데이 루이스	#####	#####



MariaDB의 데이터 형식

❖ 데이터 형식과 형 변환

- CAST(표현식 AS 데이터형식 [(길이)])
- CONVERT(표현식, 데이터형식 [(길이)])

```
USE sqlDB ;
SELECT AVG(amount) AS '평균 구매 개수' FROM buyTBL ;

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

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

```
SELECT CAST('2022$12$12' AS DATE);
SELECT CAST('2022/12/12' AS DATE);
SELECT CAST('2022%12%12' AS DATE);
SELECT CAST('2022@12@12' AS DATE);
```

MariaDB의 데이터 형식

❖ 데이터 형식과 형 변환

```
SELECT
  num,
  CONCAT(CAST(price AS CHAR(10)), 'X', CAST(amount AS CHAR(4)) ,'=' )
    AS '단가X수량',
  price*amount AS '구매액'
FROM buyTBL ;
```

MariaDB의 데이터 형식

❖ 암시적 형변환

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

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- 제어 흐름 함수
  - IF(조건, 참, 거짓)

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

  - IFNULL(수식1, 수식2)
    - 수식1이 NULL이 아니면 수식1 리턴, NULL이면 수식2 리턴

```
SELECT IFNULL(NULL, '널이군요'), IFNULL(100, '널이군요');
```

  - NULLIF(수식1, 수식2)
    - 수식1과 수식2가 같으면 NULL 반환, 다르면 수식1 반환

```
SELECT NULLIF(100,100), NULLIF(200,100);
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- 제어 흐름 함수
  - CASE ~ WHEN ~ ELSE ~ END

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

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 문자열 함수
  - ▪ ASCII(아스키코드), CHAR(숫자)

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

  - ▪ BIT_LENGTH(문자열), CHAR_LENGTH(문자열), LENGTH(문자열)

```
SELECT BIT_LENGTH('abc'),  CHAR_LENGTH('abc'), LENGTH('abc');
SELECT BIT_LENGTH('가나다'),  CHAR_LENGTH('가나다'), LENGTH('가나다');
```

  - ▪ CONCAT(문자열1, 문자열2, …), CONCAT_WS(구분자, 문자열1, 문자열2, …),

```
SELECT CONCAT_WS('/', '2022', '01', '01');
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 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)

```
SELECT
  ELT(2, '하나', '둘', '셋'),          -- 둘
  FIELD('둘', '하나', '둘', '셋'),     -- 2
  FIND_IN_SET('둘', '하나,둘,셋'),     -- 2
  INSTR('하나둘셋', '둘'),             -- 3
  LOCATE('둘', '하나둘셋');            -- 3
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 문자열 함수
  - FORMAT(숫자, 소수점 자리수)

```
SELECT FORMAT(123456.123456, 4);
```

  - BIN(숫자), HEX(숫자), OCT(숫자)

```
SELECT BIN(31), HEX(31), OCT(31);
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 문자열 함수
  - ▪ INSERT(기준 문자열, 위치, 길이, 삽입할 문자열)

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

  - ▪ LEFT(문자열, 길이), RIGHT(문자열, 길이)

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

  - ▪ UPPER(문자열), LOWER(문자열)

```
SELECT LOWER('abcdEFGH'), UPPER('abcdEFGH');
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 문자열 함수
  - LPAD(문자열, 길이), RPAD(문자열, 길이)

```
SELECT LPAD('이것이', 5, '##'), RPAD('이것이', 5, '##');
```

  - LTRIM(문자열), RTRIM(문자열)

```
SELECT LTRIM('   이것이'), RTRIM('이것이   ');
```

  - TRIM(문자열), TRIM(방향 자를_문자열 FROM 문자열)

```
SELECT TRIM('   이것이   '), TRIM(BOTH 'ㅋ' FROM 'ㅋㅋㅋ재밌어요.ㅋㅋㅋ');
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- 문자열 함수
  - REPEAT(문자열, 횟수)

```
SELECT REPEAT('이것이', 3);
```

  - REPLACE(문자열, 원래문자열, 바꿀문자열)

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

  - REVERSE(문자열)

```
SELECT REVERSE ('MariaDB');
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 문자열 함수
  - SPACE(길이)

```
SELECT CONCAT('이것이', SPACE(10), 'MariaDB다');
```

  - SUBSTRING(문자열, 시작위치, 길이),
    SUBSTRING(문자열 FROM 시작위치 FOR 길이)

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

  - SUBSTRING_INDEX(문자열, 구분자, 횟수)

```
SELECT
  SUBSTRING_INDEX('cafe.naver.com', '.', 2),  -- cafe.naver
  SUBSTRING_INDEX('cafe.naver.com', '.', -2); -- naver.com
```

구분자가 횟수만큼 나온 이후 오른쪽 버림
음수일경우 왼쪽 버림

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 수학 함수
  - ABS(숫수)

```
SELECT REPEAT('이것이', 3);
```

  - SIN(숫수), COS(숫수), TAN(숫수)
  - ACOS (숫수), ASIN(숫수), ATAN(숫수), ATAN2(숫수1, 숫자2)

  - CEILING(숫자), FLOOR(숫자), ROUND(숫자)

```
SELECT CEILING(4.7), FLOOR(4.7), ROUND(4.7);
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 숫자 함수
  - CONVERT(숫자, 원래 진수, 변환할 진수)

```
SELECT CONV('AA', 16, 2), CONV(100, 10, 8);
```

  - DEGREES(숫자), RADIANS(숫자), PI()

```
SELECT DEGREES(PI()), RADIANS(180);
```

  - MOD(숫자1, 숫자2) 또는 숫자1 % 숫자2 또는 숫자1 MOD 숫자2

```
SELECT MOD(157, 10), 157 % 10, 157 MOD 10;
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 숫자 함수
  - POW(숫자1, 숫자2), SQRT(숫자)

```
SELECT POW(2,3), SQRT(9);
```

  - RAND()

```
SELECT RAND(), FLOOR(1 + (RAND() * (7-1)) );
```

  - EXP(X), LN(숫자), LOG(숫자), LOG(밑수, 숫자), LOG2(숫자), LOG10(숫자)

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 숫자 함수
  - SIGN(숫자)

```
SELECT SIGN(100), SIGN(0), SIGN(-100.123);
```

  - TRUNCATE(숫자, 정수)

```
SELECT TRUNCATE(12345.12345, 2), TRUNCATE(12345.12345, -2);
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- 날짜 및 시간 함수
  - ADDDATE(날짜, 차이), SUBDATE(날짜, 차이)

```
SELECT
  ADDDATE('2022-01-01', INTERVAL 31 DAY),
  ADDDATE('2022-01-01', INTERVAL 1 MONTH);
SELECT
  SUBDATE('2022-01-01', INTERVAL 31 DAY),
  SUBDATE('2022-01-01', INTERVAL 1 MONTH);
```

  - ADDTIME(날짜/시간, 시간), SUBTIME(날짜/시간, 시간)

```
SELECT
  ADDTIME('2022-01-01 23:59:59', '1:1:1'),
  ADDTIME('15:00:00', '2:10:10');
SELECT
  SUBTIME('2022-01-01 23:59:59', '1:1:1'),
  SUBTIME('15:00:00', '2:10:10');
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- 날짜 및 시간 함수
  - CURDATE(), CURTIME(), NOW(), SYSDATE()
  - YEAR(날짜), MONTH(날짜), DAY(날짜)
  - HOUR(시간), MINUTE(시간), SECOND(시간), MICROSECOND(시간)

```
SELECT YEAR(CURDATE()), MONTH(CURRENT_DATE()), DAYOFMONTH(CURRENT_DATE);
SELECT HOUR(CURTIME()), MINUTE(CURRENT_TIME()), SECOND(CURRENT_TIME),
       MICROSECOND(CURRENT_TIME);
```

  - DATE(), TIME()

```
SELECT DATE(NOW()), TIME(NOW());
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ○ 날짜 및 시간 함수
  - DATEDIF(날짜1, 날짜2), TIMEDIFF(날짜1 또는 시간1, 날짜2 또는 시간2)

```
SELECT DATEDIFF('2022-01-01', NOW()), TIMEDIFF('23:23:59', '12:11:10');
```

  - DAYOFWEEK(날짜), MONTHNAME(), DAYOFYEAR(날짜)

```
SELECT DAYOFWEEK(CURDATE()), MONTHNAME(CURDATE()), DAYOFYEAR(CURDATE());
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- 날짜 및 시간 함수
  - LAST_DAY(날짜)

```
SELECT LAST_DAY('2022-02-01');
```

  - MAKEDATE(연도, 정수)

```
SELECT MAKEDATE(2022, 32);
```

  - MAKETIME(시,분,초)

```
SELECT MAKETIME(12, 11, 10);
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- 날짜 및 시간 함수
  - PERIOD_ADD(연월, 개월수), PERIOD_DIFF(연월1, 연월2)

```
SELECT PERIOD_ADD(202201, 11), PERIOD_DIFF(202201, 201812);
```

  - QUARTER(날짜)

```
SELECT QUARTER('2022-07-07');
```

  - TIME_TO_SEC(시간)

```
SELECT TIME_TO_SEC('12:11:10');
```

MariaDB의 내장 함수와 윈도 함수

❖ MariaDB의 내장 함수

- ㅇ 시스템 정보 함수
  - CURRENT_USER(), DATABASE()

```
SELECT CURRENT_USER(), DATABASE();
```

  - FOUND_ROWS():바로 앞의 SELECT 호출 결과 행의 수 리턴

```
USE sqlDB;
SELECT * FROM userTBL;
SELECT FOUND_ROWS();
```

  - ROW_COUNT(): INSERT, UPDATE, DELETE 호출 결과 영향 받은 행의 수 리턴

```
USE sqlDB;
UPDATE buyTBL SET price=price*2;
SELECT ROW_COUNT();
```

❖ 파일을 이용한 데이터 조작

- 테이블 → 파일

```
USE sqlDB;
SELECT * INTO OUTFILE 'C:/temp/userTBL.txt' FROM userTBL;
```

  - 컬럼 구분자 : 탭

- 파일 → 테이블

```
CREATE TABLE memberTBL LIKE userTBL; -- 테이블 구조만 복사

LOAD DATA LOCAL INFILE 'C:/temp/userTBL.txt' INTO TABLE memberTBL;

SELECT * FROM memberTBL;
```