

SQL 기초 문법

함수와 조건문

SQL 기초 문법 | SQL 코딩 테스트 준비하기

강사 나동빈

SQL 기초 문법

함수와 조건문

SQL 기초 문법 함수와 조건문

집계 함수

SQL. 기초 다지기

- MySQL에서는 다양한 집계 함수를 제공한다.
- GROUP BY와 함께 사용되는 경우가 많다.

함수	설명
COUNT(컬럼명)	NULL 값이 아닌 레코드의 수 계산
SUM(컬럼명)	해당 컬럼의 레코드 값의 합을 계산
MIN(컬럼명)	해당 컬럼의 레코드 값의 최솟값을 계산
MAX(컬럼명)	해당 컬럼의 레코드 값의 최댓값을 계산
AVG(컬럼명)	해당 컬럼의 레코드 값의 평균값을 계산

SQL 기초 문법 함수와 조건문

기본적인 수학 함수

SQL.
기초 다지기

- MySQL에서는 기본적인 수학 함수를 제공한다.

함수	설명
ABS(수)	절댓값 출력
CEILING(수)	소수점 이하 올림
FLOOR(수)	소수점 이하 내림
TRUNCATE(수, 자릿수)	소수점 이하 특정 자릿수에서 버림
ROUND(숫자, 자릿수)	소수점 이하 특정 자릿수에서 반올림

```
SELECT
    ROUND(723.58123, -1),
    ROUND(723.58123, 0),
    ROUND(723.58123, 1);
```

SQL 기초 문법 함수와 조건문

기본적인 수학 함수

SQL. 기초 다지기

- MySQL에서는 기본적인 수학 함수를 제공한다.

The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
1 • SELECT
2     ROUND(723.58123, -1),
3     ROUND(723.58123, 0),
4     ROUND(723.58123, 1);
```

The result grid displays the output of the query:

	ROUND(723.58123, -1)	ROUND(723.58123, 0)	ROUND(723.58123, 1)
▶	720	724	723.6

SQL 기초 문법 함수와 조건문

기타 수학 함수

SQL. 기초 다지기

- MySQL에서는 기타 수학 함수를 제공한다.

함수	설명
SQRT(수)	루트 계산
POW(수, 지수)	제곱 계산
EXP(지수)	e의 지수 제곱
LOG(수)	자연 로그 계산
SIN(수), COS(수), TAN(수)	삼각 함수

- MySQL에서는 문자열 함수를 제공한다.

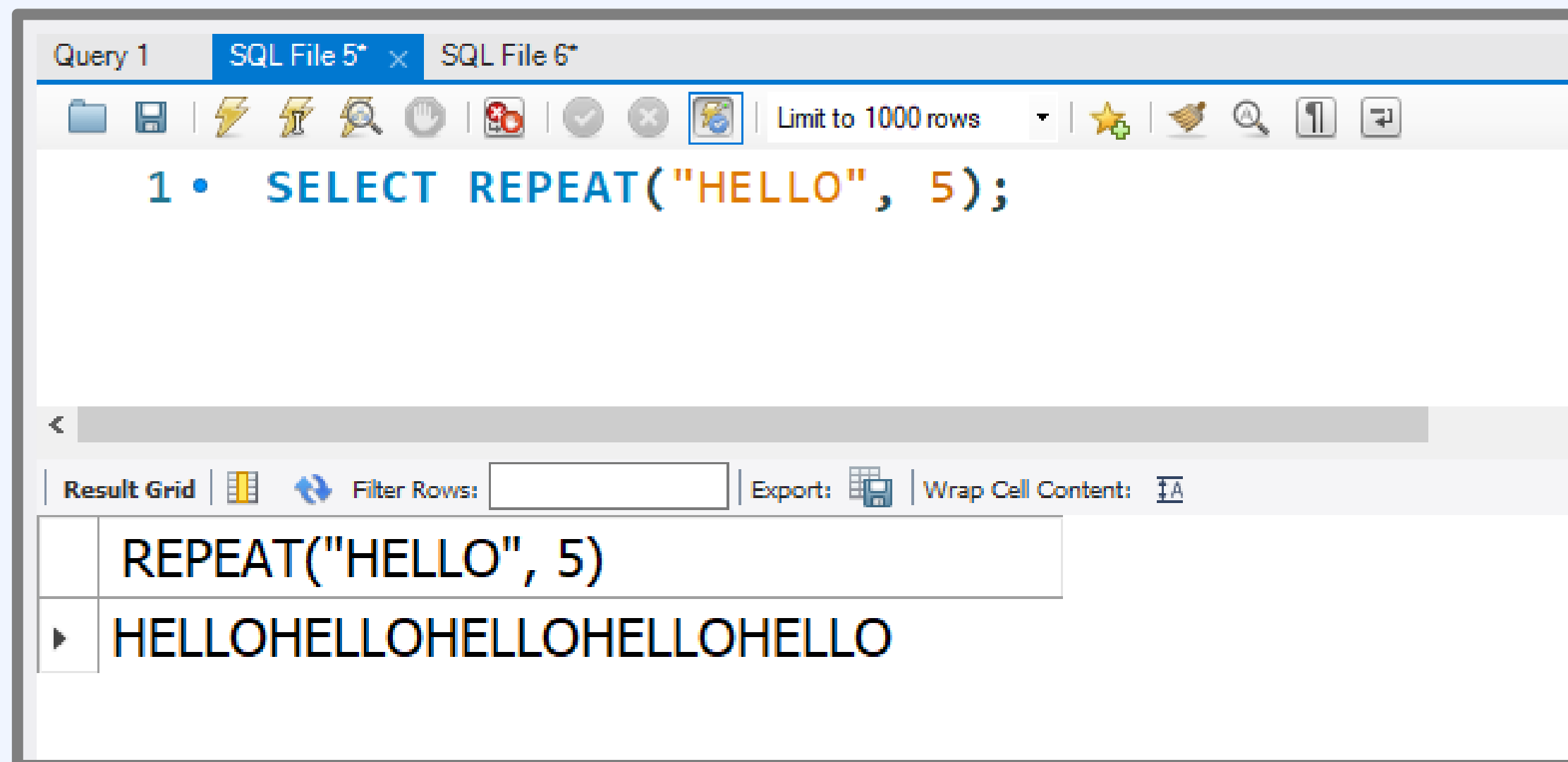
함수	설명
REPEAT(문자열, 반복 횟수)	문자열 반복
REVERSE(문자열)	문자열 뒤집기
LOWER(문자열)	소문자로 변경
UPPER(문자열)	대문자로 변경
SUBSTR(문자열, 인덱스)	인덱스부터 문자열 추출
SUBSTR(문자열, 인덱스, 길이)	인덱스부터 특정 길이만큼 문자열 추출

SQL 기초 문법 함수와 조건문

문자열 함수

SQL. 기초 다지기

- MySQL에서는 문자열 함수를 제공한다.



SQL 기초 문법 함수와 조건문

날짜 및 시간 함수

SQL.
기초 다지기

- MySQL에서는 날짜 및 시간과 관련한 함수를 제공한다.
- 게시글이 작성될 때를 기준으로 현재 시각을 기록하기 위해 NOW()를 많이 사용한다.

함수	설명
NOW()	현재 날짜와 시간 계산
CURDATE()	현재 날짜 출력
CURTIME()	현재 시간 출력
YEAR(날짜)	연도만 반환
MONTH(날짜)	월만 반환

- MySQL에서는 CASE를 사용할 수 있다.

```
CASE {변수}
  WHEN {비교할 값 1} THEN {반환 값 1}
  WHEN {비교할 값 2} THEN {반환 값 2}
  ELSE {위 조건에 해당되지 않는 경우 반환할 값}
END
```

SQL 기초 문법 함수와 조건문

CASE 문법 2) 변수를 사용하지 않는 방식

SQL.
기초 다지기

- MySQL에서는 CASE를 사용할 수 있다.

```
CASE
  WHEN {조건문 1} THEN {반환 값 1}
  WHEN {조건문 2} THEN {반환 값 2}
  ELSE {위 조건에 해당되지 않는 경우 반환할 값}
END
```

SQL 기초 문법 함수와 조건문

CASE 문법 2) 변수를 사용하지 않는 방식

SQL.
기초 다지기

- MySQL에서는 CASE를 이용해 필드(Field)를 추가할 수 있다.

```
SELECT name,
(
CASE
WHEN age < 30 THEN 'young'
ELSE 'old'
END
) AS age
FROM student;
```

SQL 기초 문법

함수와 조건문

IF() 함수

SQL.

기초 다지기

- IF() 함수는 다음과 같은 형태로 사용할 수 있다.

```
IF({조건문}, {참일 경우 반환 값}, {거짓일 경우 반환 값})
```

- 사용 예시는 다음과 같다.

```
SELECT IF(5 > 3, 'good', 'bad');
```

```
SELECT name, IF(age < 30, 'young', 'old') AS age  
FROM student;
```

- IFNULL() 함수는 다음과 같은 형태로 사용할 수 있다.

```
IFNULL({컬럼명}, {대체할 값})
```

- 기본적으로 해당 레코드를 출력하되, 값이 NULL이라면 대체할 값을 출력한다.

- '=' 연산자가 SET과 함께 사용될 때는 대입 연산자의 의미를 가진다.

```
SET @{변수명} = {값};
```

- 사용자 정의 변수는 다음과 같이 사용할 수 있다.

```
SET @start = 75, @end = 90;  
SELECT * FROM student WHERE score BETWEEN @start and @end;
```