

MySQL & Maria DB

JSON 데이터

목차

- ▶ JSON 데이터
- ▶ JSON 데이터 입력
- ▶ JSON 함수
- ▶ JSON 가져오기

JSON 데이터

▶ JSON 데이터

- ▶ JSON 데이터는 웹 환경이나 모바일 응용 프로그램 등에서 데이터를 교환하기 위해 만든 개방형 표준 포맷입니다.
- ▶ JSON으로 작성한 데이터는 속성(key)과 값(value)로 구성되어 있습니다.
- ▶ JSON은 자바스크립트 언어에서 파생되었지만 특정한 프로그래밍 언어에 종속되지 않은 독립적인 데이터 포맷입니다.

```
{ "users": [
  {
    "firstName": "Ray",
    "lastName": "Villalobos",
    "joined": {
      "month": "January",
      "day": 12,
      "year": 2012
    }
  },
  {
    "firstName": "John",
    "lastName": "Jones",
    "joined": {
      "month": "April",
      "day": 28,
      "year": 2010
    }
  }
]
}
```

JSON 데이터 입력

▶ JSON 데이터 입력

- ▶ 아래 테이블처럼 JSON 데이터를 입력할 수 있는 컬럼을 만들고 JSON 데이터가 들어 있는 샘플 데이터를 삽입해 보겠습니다.

```
CREATE TABLE employees (  
    id integer AUTO_INCREMENT primary key,  
    name VARCHAR(200),  
    profile JSON  
);
```

```
insert into employees(name, profile) values('홍길동', '{ "age" : 30,  
"gender" : "man", "부서": "개발" }');  
insert into employees(name, profile) values('김철수', '{ "age" : 43,  
"gender" : "man", "부서": "재무" }');  
insert into employees(name, profile) values('박영희', '{ "age" : 37,  
"gender" : "woman", "부서": "회계" }');
```

JSON 데이터 입력

- ▶ JSON 데이터 입력

- ▶ JSON 데이터는 아래와 같이 배열로 입력 또한 가능합니다.

```
insert into employees(name, profile) values('김갑수', '[35, "man", "인사"]');
```

- ▶ 하지만 문자열로 JSON 을 입력할 경우 실수할 여지가 많으므로 MySQL/MariaDB 가 제공하는 JSON 관련 함수를 사용하여 처리하는 게 좋습니다.

JSON 함수

▶ JSON_OBJECT

- ▶ JSON Object 입력을 쉽게 해주는 함수입니다.
- ▶ JSON_OBJECT 함수의 파라미터로 name1, value1, name2, value2 처럼 key: value 쌍을 맞춰서 호출하며 가독성을 위해 파라미터 쌍마다 개행을 하는 것을 추천합니다.

```
insert into employees(name, profile) values('신상일', json_object(  
    'age', 28,  
    'gender', 'man',  
    '부서', '연구'  
));
```

JSON 함수

▶ JSON_ARRAY

- ▶ 배열 입력 시 유용하며 다음은 종업원 정보 중에 소지한 자격증을 배열로 입력하는 예제입니다.

```
insert into employees(name, profile) values('은연수', json_object(  
    'age', 29,  
    'gender', 'woman',  
    '부서', '개발',  
    '자격증', json_array('CISA', 'PMP', 'CISSP')  
));
```

JSON 함수

▶ JSON_Extract

- ▶ 컬럼에서 JSON 데이터를 추출하며 JSON Path 문법을 사용합니다.
- ▶ 아래는 종업원의 이름과 부서를 표시하는 예제입니다.

예 `select id, name, json_extract(profile, '$.부서') from employees;`

id	name	json_extract(profile, '\$.부서')
1	홍길동	"개발"
2	김철수	"재무"
3	박영희	"회계"
4	김갑수	<null>
5	신상일	"연구"
6	은연수	"개발"

JSON 함수

▶ JSON_Extract

- ▶ 다음은 나이가 35 이상인 종업원의 이름과 나이를 표시합니다.

예

```
select id, name, json_extract(profile, '$.age')  
from employees  
where json_extract(profile, '$.age') >= 35;
```

	id	name	json_extract(profile, '\$.age')
1	2	김철수	43
2	3	박영희	37

JSON 함수

▶ JSON_Replace

- ▶ JSON 컬럼에서 값을 치환해서 리턴하는 함수입니다..
- ▶ 다음은 age 필드의 값을 30으로 변경하는 예제입니다.

예

```
select id, name, json_replace(profile, '$.age', 30)
from employees;
```

	id	name	json_replace(profile, '\$.age', 30)
1	1	홍길동	{"age": 30, "gender": "man", "부서": "개발"}
2	2	김철수	{"age": 30, "gender": "man", "부서": "재무"}
3	3	박영희	{"age": 30, "gender": "woman", "부서": "회계"}
4	4	김갑수	[35, "man", "인사"]
5	5	신상일	{"age": 30, "gender": "man", "부서": "연구"}
6	6	은연수	{"age": 30, "gender": "woman", "부서": "개발", "자격증": ["CISA", "PMP", "CISSP"]}

JSON 함수

▶ JSON_Replace

- ▶ `son_replace` 도 여러 컬럼을 수정할 수 있으며 **key, value** 쌍으로 파라미터를 넘기면 됩니다. 가독성을 위해 `json_object` 처럼 파라미터 쌍마다 개행을 하는 것이 좋습니다.
- ▶ 다음은 `age`와 `gender` 필드의 값을 변경해서 표시하는 예제입니다.

예

```
select id, name, json_replace(profile,  
    '$.age', 30,  
    '$.gender', '남녀')  
from employees;
```

	id	name	json_replace(profile, '\$.age', 30, '\$.gender', '남녀')
1	1	홍길동	{"age": 30, "gender": "남녀", "부서": "개발"}
2	2	김철수	{"age": 30, "gender": "남녀", "부서": "재무"}
3	3	박영희	{"age": 30, "gender": "남녀", "부서": "회계"}
4	4	김갑수	[35, "man", "인사"]
5	5	신상일	{"age": 30, "gender": "남녀", "부서": "연구"}
6	6	은연수	{"age": 30, "gender": "남녀", "부서": "개발", "자격증": ["CISA", "PMP", "CISSP"]}

JSON 함수

▶ JSON_Replace

- ▶ 모든 컬럼값을 동일하게 변경하는 것은 보통 유용하지가 않으므로 기존 컬럼 값을 수정할 경우 `json_extract` 와 `json_replace` 를 같이 사용해서 값을 변경합니다.
- ▶ 다음은 모든 직원의 `age` 필드 값에 1을 더해서 가져옵니다.

예

```
select id, name,  
       json_replace(profile, '$.age', json_extract(profile, '$.age') + 1)  
from employees;
```

	id	name	json_replace(profile, '\$.age', json_extract(profile, '\$.age') + 1)
1	1	홍길동	{"age": 31, "gender": "man", "부서": "개발"}
2	2	김철수	{"age": 44, "gender": "man", "부서": "재무"}
3	3	박영희	{"age": 38, "gender": "woman", "부서": "회계"}
4	4	김갑수	[35, "man", "인사"]
5	5	신상일	{"age": 29, "gender": "man", "부서": "연구"}
6	6	은연수	{"age": 30, "gender": "woman", "부서": "개발", "자격증": ["CISA", "PMP", "CISSP"]}