



제5장

데이터 형식과 내장함수-2

1. JSON 데이터와 대용량 데이터 저장

■ JSON(JavaScript Object Notation)

- 웹 환경이나 모바일 응용 프로그램 등에서 데이터를 교환하기 위해 만든 개방형 표준 포맷
- JSON으로 작성한 데이터는 속성(key)과 값(value)의 쌍으로 구성
- JSON은 자바스크립트 언어에서 파생되었지만 특정한 프로그래밍 언어에 종속되지 않은 독립적인 데이터 포맷

(중요) JSON은 언어에 종속되지 않고 교환을 할 수 있는 데이터 포맷형식!!
포맷이 단순하고 공개 되어 있기에 거의 대부분의 프로그래밍 언어에서 쉽게
읽거나 쓸 수 있도록 프로그래밍 가능하다.

1. JSON 데이터와 대용량 데이터 저장

- 한 명의 사용자를 JSON 데이터로 표현한 예

```
{
  "아이디": "KHD",
  "이름": "강호동",
  "생년": 1970,
  "지역": "경북",
  "국번": "011",
  "전화번호": "22222222",
  "키": 182,
  "가입일": "2017.7.7"
}
```

	userName	height
▶	강호동	182
	이휘재	180
	남희석	180
	박수홍	183

MySQL 테이블

JSON_OBJECT()
JSON_ARRAY()

```
{
  "name": "강호동", "height": 182
},
{
  "name": "이휘재", "height": 180
},
{
  "name": "남희석", "height": 180
},
{
  "name": "박수홍", "height": 183
}
```

JSON 데이터

그림 7-10 테이블의 데이터를 JSON 데이터로 변환

1. JSON 데이터와 대용량 데이터 저장

- 키가 180cm 이상인 사람의 이름과 키를 JSON 데이터로 변환

```
USE sqldb;  
SELECT JSON_OBJECT('name', userName, 'height', height) AS 'JSON 값'  
FROM userTBL  
WHERE height >= 180;
```

	JSON 값
▶	{"name": "강호동", "height": 182}
	{"name": "이휘재", "height": 180}
	{"name": "남희석", "height": 180}
	{"name": "박수홍", "height": 183}

JSON_OBJECT는 쿼리문의 결과를 JSON형태로 바꿔주는 함수이다.

1. JSON 데이터와 대용량 데이터 저장

- @json 변수에 JSON 데이터를 대입하면서 테이블의 이름을 userTBL로 지정

```
1 SET @json='{ "userTBL":  
2 [  
3   {"name":"강호동", "height":182},  
4   {"name":"이휘재", "height":180},  
5   {"name":"남희석", "height":180},  
6   {"name":"박수홍", "height":183}  
7 ]  
8 }';
```

```
9 SELECT JSON_VALID(@json) AS JSON_VALID;  
10 SELECT JSON_SEARCH(@json, 'one', '남희석') AS JSON_SEARCH;  
11 SELECT JSON_EXTRACT(@json, '$.userTBL[2].name') AS JSON_EXTRACT;  
12 SELECT JSON_INSERT(@json, '$.userTBL[0].mDate', '2019-09-09') AS JSON_INSERT;  
13 SELECT JSON_REPLACE(@json, '$.userTBL[0].name', '토마스') AS JSON_REPLACE;  
14 SELECT JSON_REMOVE(@json, '$.userTBL[0]') AS JSON_REMOVE;
```

1. JSON_VALID()는 대입된 문자열이 JSON형태인지 확인함(맞으면 1, 틀리면 0을 리턴)
2. JSON_SEARCH()는 '남희석'의 인덱스를 리턴하는 함수인데, JSON은 배열 형태로 되어 있어 2를 리턴한다. 인자값 중 one이 있는데 이것은 all로 해도 상관없다. 그냥 문법이다.
3. JSON_EXTRACT()는 인덱스의 값을 추출하는 것이다. 여기서는 남희석을 가져올 것이다.
4. JSON_INSERT()는 해당 인덱스의 값에 삽입을 해라는 함수이다.
5. JSON_REPLACE()는 해당 인덱스의 값을 치환하라는 함수이다.
6. JSON_REMOVE()는 해당 인덱스의 값을 삭제하라는 함수이다.

JSON_VALID	JSON_SEARCH	JSON_EXTRACT
1	"\$.userTBL[2].name"	"남희석"

JSON_INSERT
{ "userTBL": [{ "name": "강호동", "mDate": "2019-09-09", "height": 182 }, { "name": "이휘재", "height": 180 }, { "name": "남희석", "height": 180 }, { "name": "박수홍", "height": 183 }] }

JSON_REPLACE
{ "userTBL": [{ "name": "토마스", "height": 182 }, { "name": "이휘재", "height": 180 }, { "name": "남희석", "height": 180 }, { "name": "박수홍", "height": 183 }] }

JSON_REMOVE
{ "userTBL": [{ "name": "이휘재", "height": 180 }, { "name": "남희석", "height": 180 }, { "name": "박수홍", "height": 183 }] }

감사합니다.

