

08 피벗 / JSON 내장함수



Part 1.

피벗



01 피벗


피벗

```
select userid,
sum(if(prodname='모니터',amount,0)) as '모니터',
sum(if(prodname='청바지',amount,0)) as '청바지',
sum(if(prodname='메모리',amount,0)) as '메모리',
sum(if(prodname='운동화',amount,0)) as '운동화',
sum(if(prodname='책',amount,0)) as '책',
sum(amount) as '합계' from buytbl group by userid;
```

	userid	모 니 터	청 바 지	메 모 리	운 동 화	책	합계
▶	BBK	5	0	10	4	0	19
	EJW	0	1	0	0	3	4
	JYP	1	0	0	0	0	1
	KBS	0	3	0	2	0	6
	SSK	0	0	0	0	5	5

00 중간문제

1 usertbl의 addr(주소)를 이용하여 다음과 같은 피벗 테이블을 만들어 보세요

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap				
	서울	경남	경북	경기
▶	4	2	1	2

```
select
sum(if(addr='서울',1,0)) as '서울',
sum(if(addr='경남',1,0)) as '경남',
sum(if(addr='경북',1,0)) as '경북',
sum(if(addr='경기',1,0)) as '경기'
from usertbl;
```

00 중간문제

1 usertbl의 addr(주소)를 이용하여 다음과 같은 피벗 테이블을 만들어 보세요

	userid	전자	의류	서적	null
▶	BBK	2	0	0	0
	EJW	0	1	2	0
	JYP	1	0	0	0
	KBS	1	1	0	0
	SSK	0	0	1	0

```
select userid,sum(if(addr='서울',1,0)) as '서울',sum(if(addr='경남',1,0)) as '경남',sum(if(addr='경북',1,0)) as '경북',sum(if(addr='경기',1,0)) as '경기' from usertbl group by userid;
```

Part 2.

JSON 관련 내장함수



01 JSON

JSON

JavaScript Object Notation

웹과 모바일 응용 프로그램등과 데이터를 교환하기 위한 개방형 표준 포맷
속성(Key)와 값(Value)로 쌍을 이루며 구성되어 있다
특정 프로그래밍 언어에 종속되어 있지 않은 독립적인 데이터 포맷이다

JSON_OBJECT() - object저장

```
use sqldb;  
select JSON_OBJECT('name',name,'height',height) as 'JSON값'  
from usertbl where height >=180;
```

	JSON값
▶	{"name": "임재범", "height": 182}
	{"name": "이승기", "height": 182}
	{"name": "성시경", "height": 186}

01 JSON

문제

-- buytbl에서 userID와 ProdName과 amount를 각각 key로-- 지정해서 value을 JSON로 출력해보세요
(amount 3이상인 행만 적용)

답

```
select JSON_OBJECT('ID',userID,'PN',prodName,'Amount',amount)from buytbl where amount>=3;
```


01 JSON

JSON 변수 지정

```
SET @json= '{"usertbl" :  
[  
    {"name":"임재범","height":182},  
    {"name":"이승기","height":182},  
    {"name":"성시경","height":186}  
]}';
```

JSON 내장함수 사용

```
SELECT JSON_VALID(@json) as JSON_VALID; -- 문자열 형식 일치여부  
SELECT JSON_SEARCH(@json,'one','성시경') as JSON_SEARCH; -- 내용찾기 one :검색되는 1번째, all:모든  
SELECT JSON_EXTRACT(@json, '$.usertbl[2].name') AS JSON_EXTRACT; -- 값 추출  
SELECT JSON_INSERT(@json, '$.usertbl[0].mDate','2020-01-01') AS JSON_INSERT; -- 값 삽입  
SELECT JSON_REPLACE(@json, '$.usertbl[0].name','홍길동') AS JSON_REPLACE; -- 값 변경  
SELECT JSON_REMOVE(@json,'$.usertbl[0]') AS JSON_REMOVE; -- 요소 삭제
```

01 문제

문제

```
set @Acc='{ "IDPW" :  
[  
    { "ID": "aaaa", "PW": 1111 },  
    { "ID": "bbbb", "PW": 2222 },  
    { "ID": "cccc", "PW": 1111 },  
    { "ID": "dddd", "PW": 4444 },  
    { "ID": "eeee", "PW": 1111 }  
]  
}';
```

1. ID가 'bbbb'인 object 위치 찾기 (JSON_SEARCH)
2. 3번째 idx의 ID 추출(JSON_EXTRACT)
3. 0번째 idx에 name 키와 'JungWoo' value 값 Insert해서 출력(JSON_INSERT)
4. 1번째 idx의 ID를 'wow'로 변경해서 출력(JSON_REPLACE)
5. 2번째 idx를 Remove해서 출력(JSON_REMOVE)

01 답

답

1. ID가 'bbbb'인 object 위치 찾기 (JSON_SEARCH)
2. 3번째 idx의 ID 추출(JSON_EXTRACT)
3. 0번째 idx에 name 키와 'JungWoo' value 값 Insert해서 출력(JSON_INSERT)
4. 1번째 idx의 ID를 'wow'로 변경해서 출력(JSON_REPLACE)
5. 2번째 idx를 Remove해서 출력(JSON_REMOVE)

```
SELECT JSON_SEARCH(@Acc, 'all','bbbb') as JSON_SEARCH;  
SELECT JSON_EXTRACT(@Acc, '$.IDPW[3].ID') AS JSON_EXTRACT;  
SELECT JSON_INSERT(@Acc, '$.IDPW[0].name','JungWoo') AS JSON_INSERT;  
SELECT JSON_REPLACE(@Acc, '$.IDPW[1].ID','wow') AS JSON_REPLACE;  
SELECT JSON_REMOVE(@Acc,'$.IDPW[2]') AS JSON_REMOVE;
```

01 JSON

JSON 자료형 테이블 생성

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

Object 단위 삽입

```
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" }');  
select * from employees;
```

01 JSON

배열형[] 으로 입력

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

내장함수 사용삽입 JSON_OBJECT()

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

```
select * from employees;
```

01 JSON

내장함수 사용삽입 JSON_ARRAY()

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

01 JSON

JSON_EXTRACT()

```
select id,name,JSON_EXTRACT(profile,'$. "age"')from employees where  
JSON_EXTRACT(profile,'$. "age"')>=30;
```

JSON_EXTRACT()

```
select id,name,JSON_EXTRACT(profile,'$. "부서"')from employees;
```

01 JSON

JSON_EXTRACT() - 배열형 조회

```
select id,name,JSON_EXTRACT(profile,'$[0]') from employees where name='김갑수';  
select id,name,JSON_EXTRACT(profile,'$[1]') from employees where name='김갑수';  
select id,name,JSON_EXTRACT(profile,'$[2]') from employees where name='김갑수';
```

JSON_EXTRACT() - 배열형 조회

```
select id,name,JSON_EXTRACT(profile,'$.자격증[0]') from employees where name='은연수';  
select id,name,JSON_EXTRACT(profile,'$.자격증[1]') from employees where name='은연수';  
select id,name,JSON_EXTRACT(profile,'$.자격증[2]') from employees where name='은연수';
```


01 JSON

값 조회 - JSON_EXTRACT()

```
select id,name,json_extract(profile, '$."부서"')  
from employees where json_extract(profile, '$."부서"') = '개발';
```

01 JSON

포함문자 검색- JSON_EXTRACT()

[오류]

```
select id,name,json_extract(profile, '$."부서"')  
from employees where json_extract(profile, '$."부서"') like '개%';
```

-- 밑줄 쿼리를 실행하면 0이 나오며, 그 이유는 json_extract가 추출 결과에 ""를 붙이기 때문입니다.

-- 해서 구문을 평가하는 결과가 0이 나오게됩니다.(Mysql 은 문자열은 숫자로 0)

```
select '"개발"' like '개%';
```

[정정]

MySQL의 "추출 된 결과를 ->> 연산자를 사용하여 "" 를 제거해 줍니다.

```
select id,name, profile ->> '$."부서"', json_extract(profile, '$."부서"')  
from employees where profile ->> '$."부서"' like '개%';
```

01 JSON

조회- JSON_EXTRACT()

-- 종업원의 이름과 나이를 표시

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

-- 조건절 사용 , 나이가 35이상인 종업원의 이름과 나이를 표시

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

01 JSON

조회- JSON_REPLACE()

-- age 필드의 값을 30으로 변경

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

-- age와 성별 필드 값을 변경

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

-- 모든 직원의 나이 필드 값에 1을 더해 조회

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

END.

고생하셨습니다.

