



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. 여기서 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 }] }		

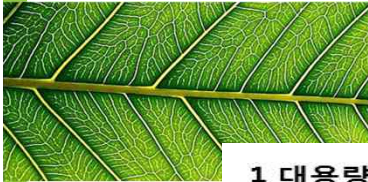
2. 대용량 데이터 저장

LONG TEXT, LONGBLOB

- MySQL은 대용량 데이터(Large Object, LOB)를 저장하기 위해 LONGTEXT, LONGBLOB 데이터 형식을 지원
- 이 데이터 형식을 이용하면 약 4GB의 파일을 하나의 데이터로 저장할 수 있음



그림 7-13 LONGTEXT, LONGBLOB 데이터 형식의 활용 예



2. 대용량 데이터 저장(실습)

1 대용량 텍스트 파일과 동영상 파일 준비하기

1-1 영화 대본 파일 3개와 영화 동영상 파일 3개 준비

2 영화 데이터베이스 만들기

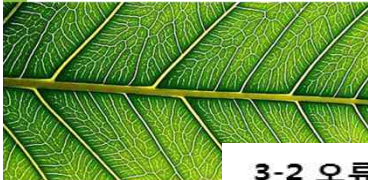
2-1 영화 데이터베이스(movieDB)와 영화 테이블(movieTBL) 만들기

```
CREATE DATABASE movieDB;  
  
USE movieDB;  
CREATE TABLE movieTBL  
( movie_id INT,  
  movie_title VARCHAR(30),  
  movie_director VARCHAR(20),  
  movie_star VARCHAR(20),  
  movie_script LONGTEXT,  
  movie_film LONGBLOB )  
DEFAULT CHARSET=utf8mb4;
```

3 대용량 데이터 입력하기

3-1 파일을 데이터로 입력

```
INSERT INTO movieTBL VALUES (1, '쥘리엔 리스트', '스필버그', '리암 니슨',  
LOAD_FILE('C:\WWW\SQL\WWW\Movies\WWW\Schindler.txt'), LOAD_FILE('C:\WWW\SQL\WWW\Movies\WWW\chindler.mp4'));
```

2. 대용량 데이터 저장(실습)

3-2 오류 없이 입력되었으면 조회

```
SELECT * FROM movieTBL;
```

	movie_id	movie_title	movie_director	movie_star	movie_script	movie_film
▶	1	원들러 리스트	스필버그	리암 니슨	NULL	NULL

3-3 max_allowed_packet 값 수정

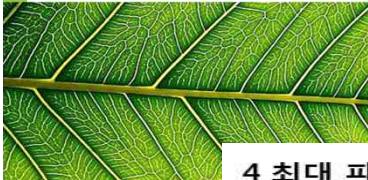
```
SHOW variables LIKE 'max_allowed_packet';
```

	Variable_name	Value
▶	max_allowed_packet	4194304

3-4 파일을 업로드하거나 다운로드할 폴더 경로를 별도로 허용

```
SHOW variables LIKE 'secure_file_priv';
```

	Variable_name	Value
▶	secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\



2. 대용량 데이터 저장(실습)

4 최대 파일 크기와 허용된 파일의 경로 추가하기

4-1 명령 프롬프트를 관리자 모드로 열기

4-2 my.ini 파일 확인

```
CD %PROGRAMDATA%  
CD MySQL  
CD "MySQL Server 8.0"  
DIR
```

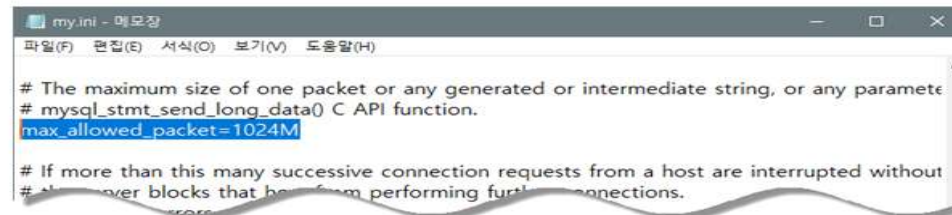
```
관리자: Windows PowerShell  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
PS C:\WINDOWS\system32> CD  
Microsoft Windows [Version 10.0.17134.228]  
(c) 2018 Microsoft Corporation. All rights reserved.  
C:\WINDOWS\system32> CD %PROGRAMDATA%  
C:\ProgramData> CD MySQL  
C:\ProgramData\MySQL> CD "MySQL Server 8.0"  
C:\ProgramData\MySQL\MySQL Server 8.0> DIR  
C:\ProgramData\MySQL\MySQL Server 8.0 디렉터리  
2018-07-06 오후 04:36 <DIR> .  
2018-07-06 오후 04:36 <DIR> ..  
2018-09-06 오전 11:18 <DIR> Data  
2018-07-06 오후 04:36 433 installer_config.xml  
2018-07-06 오후 04:36 13,250 my.ini  
2018-07-06 오후 04:36 <DIR> tmpspace  
2개 파일 13,683 바이트  
4개 디렉터리 44,309,147,648 바이트 남음  
C:\ProgramData\MySQL\MySQL Server 8.0>
```



2. 대용량 데이터 저장(실습)

4-3 max_allowed_packet 값을 1024M(1GB)으로 변경

```
max_allowed_packet=1024M
```



4-4 secure-file-priv 옵션 아래에 한 줄을 추가

```
secure-file-priv="C:/ProgramData/MySQL/MySQL Server 8.0/Uploads" secure-file-priv="C:/SQL/Movies"
```





2. 대용량 데이터 저장(실습)

4-5 MySQL 서버 재시작

```
NET STOP MySQL
NET START MySQL
```

```
관리자: Windows PowerShell
C:\ProgramData\MySQL\MySQL Server 8.0> NET STOP MySQL
MySQL 서비스를 멈춥니다.
MySQL 서비스가 잘 멈추었습니다.

C:\ProgramData\MySQL\MySQL Server 8.0> NET START MySQL
MySQL 서비스를 시작합니다.
MySQL 서비스가 잘 시작되었습니다.
```

5 다시 대용량 데이터 입력하기

5-1 다시 대용량 데이터 입력

```
USE movieDB;
TRUNCATE movieTBL;

INSERT INTO movieTBL VALUES ( 1, '쉘들러 리스트', '스필버그', '리암 니슨',
    LOAD_FILE('C:\WWSQL\WWMovies\WWSchindler.txt'), LOAD_FILE('C:\WWSQL\WWMovies\WWSchindler.mp4') );
INSERT INTO movieTBL VALUES ( 2, '쇼생크 탈출', '프랭크 다라본트', '팀 로빈스',
    LOAD_FILE('C:\WWSQL\WWMovies\WWSchawshank.txt'), LOAD_FILE('C:\WWSQL\WWMovies\WWSchawshank.mp4') );
INSERT INTO movieTBL VALUES ( 3, '라스트 모히칸', '마이클 만', '다니엘 데이 루이스',
    LOAD_FILE('C:\WWSQL\WWMovies\WWMohican.txt'), LOAD_FILE('C:\WWSQL\WWMovies\WWMohican.mp4') );

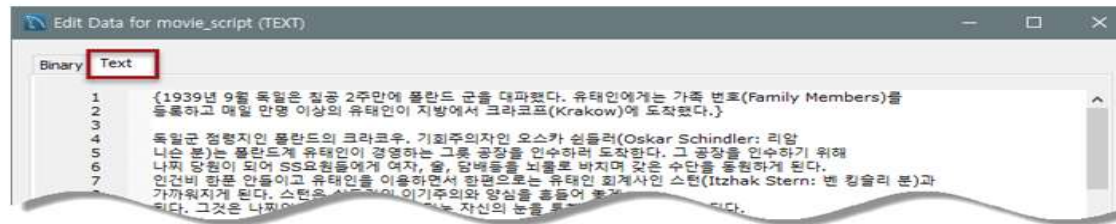
SELECT * FROM movieTBL;
```

	movie_id	movie_title	movie_director	movie_star	movie_script	movie_film
	1	쉘들러 리스트	스필버그	리암 니슨	{1939년 9월 독일은 침공 2주만에 폴란드 군을 대...	BLOB
	2	쇼생크 탈출	프랭크 다라본트	팀 로빈스	은행원인 앤디 듀플레인은 아내와 프로골퍼인 그의...	BLOB
	3	라스트 모히칸	마이클 만	다니엘 데이 루이스	마이클 만Michael Mann이 영화화하려 했던 토머스 ...	BLOB



2. 대용량 데이터 저장(실습)

5-2 대본 전체 확인

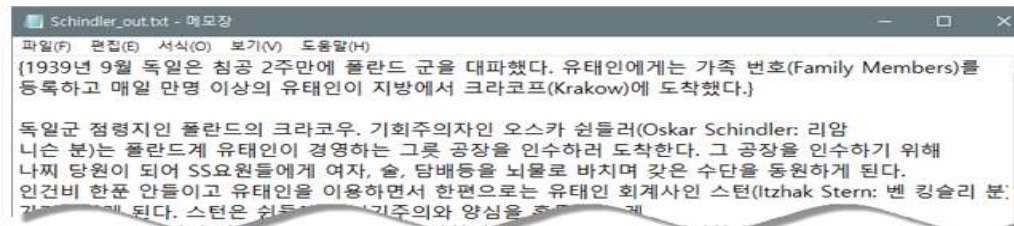


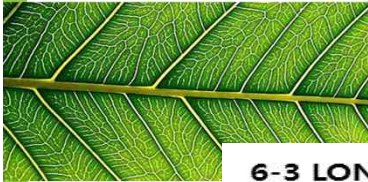
6 입력된 데이터를 파일로 내려받기

6-1 영화 아이디가 1인 윈들러 리스트의 영화 대본(movie_script)을 Schindler_out.txt 파일로 내려받기

```
SELECT movie_script FROM movieTBL WHERE movie_id=1  
INTO OUTFILE 'C:/SQL/Movies/Schindler_out.txt'  
LINES TERMINATED BY 'WWN';
```

6-2 내려받은 파일 열기

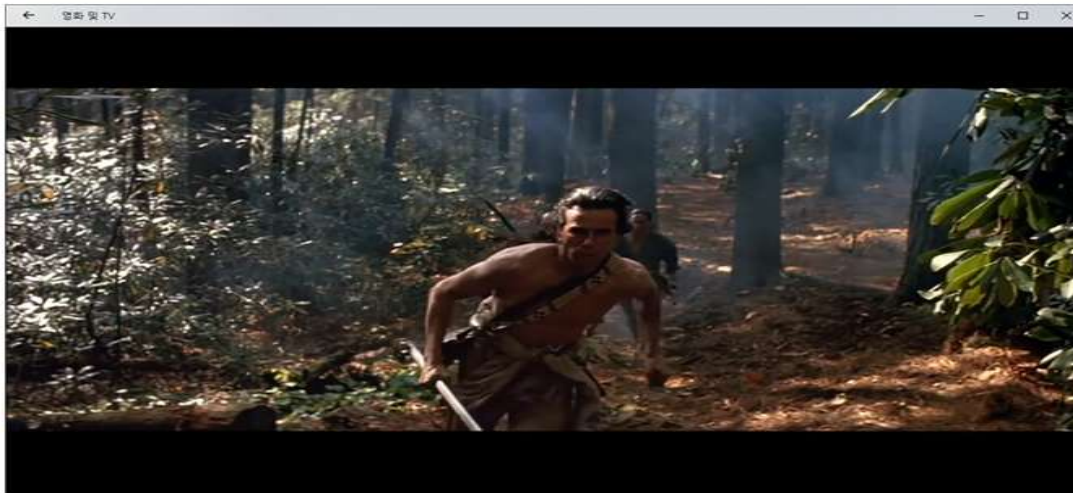




2. 대용량 데이터 저장(실습)

6-3 LONGBLOB 형식인 영화 동영상(movie_film)은 INTO DUMPFILE 문을 사용하면 바이너리 파일로 내려 받을 수 있음

```
SELECT movie_film FROM movieTBL WHERE movie_id=3  
      INTO DUMPFILE 'C:\WWWSQLWWW\Movies\WWWMohican_out.mp4';
```



A close-up photograph of two overlapping green leaves. The top leaf is a lighter shade of green and shows a detailed network of veins. The bottom leaf is a darker shade of green and also shows vein structure. A white text overlay is centered on the image.

감사합니다.