



2020-하계 계절학기 해외취업역량 (직무)

1DAY - 4
MY SQL 살펴보기

MY SQL 실행



가장 정확

서비스

서비스(로컬)

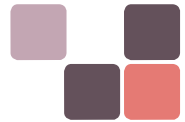
서비스(로컬)

MySQL80

이름	설명	상태	시작 유형	다음 사용자로 로그인
McAfee Security Scan Comp...	McAf...		수동	Local System
MessagingService_2ecc85	문자 ...		수동(트리...	Local System
Microsoft (R) 진단 허브 표...	진단 ...		수동	Local System
Microsoft Account Sign-in ...	사용...		수동(트리...	Local System
Microsoft Edge Elevation Se...	Keep...		수동	Local System
Microsoft iSCSI Initiator Ser...	이 컴...		수동	Local System
Microsoft Passport	사용...		수동(트리...	Local System
Microsoft Passport Container	ID 공...		수동(트리...	Local Service
Microsoft SharePoint Works...			수동	Local Service
Microsoft Software Shadow...	볼륨 ...		수동	Local System
Microsoft Storage Spaces S...	Micr...		수동	Network Service
Microsoft Store 설치 서비스	Micr...	실행 ...	수동	Local System
Microsoft Windows SMS 라...	적절...		수동(트리...	Local Service
Mozilla Maintenance Service	Mozil...		수동	Local System
MyFirewall 4.0 Service	MyFi...		수동	Local System
MySQL80		실행 ...	자동	Network Service
Net.Tcp Port Sharing Service	net.tc...		사용 안 함	Local Service
Netlogon	사용...		수동	Local System
Network Connected Devices...	네트...		수동(트리...	Local Service
Network Connection Broker	Wind...	실행 ...	수동(트리...	Local System
Network Connections	네트...		수동	Local System
Network Connectivity Assist...	UI 구...		수동(트리...	Local System

확장 표준

MY SQL 중지



보기(V) 도움말(H)



서비스(로컬)

MySQL80

서비스 [중지](#)
서비스 [일시 중지](#)
서비스 [다시 시작](#)

이름	설명	상태	시작 유형	다음 사용자로 로그인
McAfee Security Scan Comp...	McAf...		수동	Local System
MessagingService_2ecc85	문자 ...		수동(트리...	Local System
Microsoft (R) 진단 허브 표...	진단 ...		수동	Local System
Microsoft Account Sign-in ...	사용...		수동(트리...	Local System
Microsoft Edge Elevation Se...	Keep...		수동	Local System
Microsoft iSCSI Initiator Ser...	이 컴...		수동	Local System
Microsoft Passport	사용...		수동(트리...	Local System
Microsoft Passport Container	ID 공...		수동(트리...	Local Service
Microsoft SharePoint Works...			수동	Local Service
Microsoft Software Shadow...	볼륨 ...		수동	Local System
Microsoft Storage Spaces S...	Micr...		수동	Network Service
Microsoft Store 설치 서비스	Micr...	실행 ...	수동	Local System
Microsoft Windows SMS 라...	적절...		수동(트리...	Local Service
Mozilla Maintenance Service	Mozil...		수동	Local System
MyFirewall 4.0 Service	MyFi...		수동	Local System
MySQL80		실행 ...	자동	Network Service
Net.Tcp Port			사용 안 함	Local Service
Netlogon			수동	Local System
Network Cor			수동(트리...	Local Service
Network Cor		행 ...	수동(트리...	Local System
Network Cor			수동	Local System
Network Cor			수동(트리...	Local System

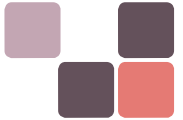
확장 / 표준

QL80 서비스를 중지합니다.

시작(S)
중지(O)
일시 중지(U)
계속(M)
다시 시작(E)
모든 작업(K)
새로 고침(F)

중지(T) | 일시 중지(P) | 계속(R)

MY SQL 중지



- ❖ 속성 창 - "중지"버튼 클릭
- ❖ 윈도우가 실행될 때 자동으로 실행되길 원하지 않는다면, 시작 유형 "자동"을 "수동"으로 변경
- ❖ 이 경우 유형 창에서 "시작"버튼을 매번 눌러줘야 MySQL서버가 실행됨

MySQL80 속성(로컬 컴퓨터)

일반 로그인 복구 종속성

서비스 이름: MySQL80

표시 이름: MySQL80

설명:

실행 파일 경로:
"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe" --defaults-file

시작 유형(E): 자동

서비스 상태: 중지됨

시작(S) 중지(T) 일시 중지(P) 계속(R)

서비스를 시작할 때 적용할 시작 매개 변수를 지정할 수 있습니다.

시작 매개 변수(M):

확인 취소 적용(A)

MY SQL 버전, 날짜, 특징 알아보기



```
mysql> select version(), current_date;
```

```
+-----+-----+  
| version() | current_date |  
+-----+-----+  
| 8.0.20    | 2020-07-13   |  
+-----+-----+  
1 row in set (0.00 sec)
```

쿼리를 이용해서 계산식의 결과도 구할 수 있다.

함수 및 수식 사용 예제

```
1  mysql> SELECT SIN(PI()/4), (4+1)*5;
```

```
2  +-----+-----+
```

```
3  | SIN(PI()/4) | (4+1)*5 |
```

```
4  +-----+-----+
```

```
5  | 0.707107 | 25 |
```

```
6  +-----+-----+
```

MY SQL 버전, 날짜, 특징 알아보기



하나의 SQL은 여러 줄로 입력가능하다.

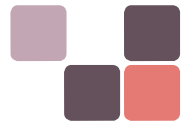
MySQL은 문장의 끝을 라인으로 구분하는 것이 아니라 semicolon(;)으로 구분하기 때문에 여러 줄에 걸쳐 문장을 쓰는 것도 가능합니다.

```
1  mysql> SELECT
2      -> USER()
3      -> ,
4      -> CURRENT_DATE;
5  +-----+-----+
6  | USER()      | CURRENT_DATE |
7  +-----+-----+
8  | joesmith@localhost | 1999-03-18 |
9  +-----+-----+
```

SQL을 입력하는 도중에 취소할 수 있다.

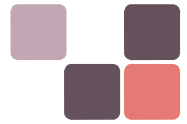
긴 쿼리를 작성하다가 중간에 취소해야 하는 경우에는 즉시 `^\c`를 붙여주면 됩니다.

```
1  mysql> SELECT
2
3      -> USER()
4
5      -> ^\c
6
```



❖ DBMS에 존재하는 데이터베이스 확인하기

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sakila |  
| sys |  
| world |  
+-----+  
6 rows in set (0.01 sec)
```



❖ 데이터를 저장하는 공간 테이블(Table)

- 데이터베이스는 엑셀의 표와 유사한 테이블을 가질 수 있음
- 엑셀과 다른 점은 데이터베이스를 생성해도 테이블은 존재하지 않음
- 테이블을 사용하려면 테이블을 생성하는 SQL을 사용해야 한다.
- 테이블에 값을 저장하려면 저장하기 위한 SQL을 사용해야 한다.

❖ 테이블(table)의 구성요소

table Student

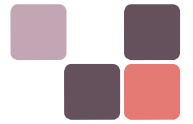
column(key)

student_id	name	grade	dept
1	정성진	1	컴퓨터
2	박현진	2	수학
3	홍길동	4	물리
...

row(record)

field

field



❖ 테이블(table)의 구성요소

- 테이블 : RDBMS(관계형 dbms)의 기본적 저장구조에서 한 개 이상의 column과 0개 이상의 row로 구성
- 열(Column) : 테이블 상에서의 단일 종류의 데이터를 나타냄. 특정 데이터 타입 및 크기를 가지고 있음
- 행(Row) : Column들의 값의 조합. 레코드라고 불림. 기본키(PK)에 의해 구분. 기본키는 중복을 허용하지 않으며 없어서는 안됨
- Field : Row와 Column의 교차점으로 Field는 데이터를 포함할 수 있고 없을 때는 NULL 값을 가지고 있음

번호	이름	생일	성별	필드 레코드
1	김미우	20yy-mm-dd	여	
2	김유나	20yy-mm-dd	여	레코드
3	김민후	20yy-mm-dd	남	
4	김해윤	20yy-mm-dd	여	

데이터베이스 생성



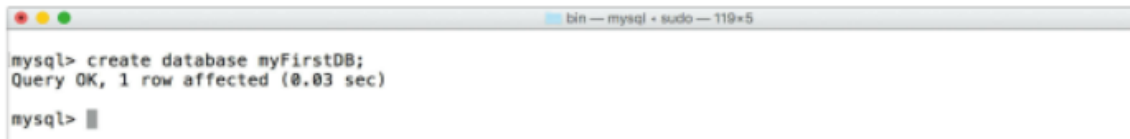
생성할 데이터베이스의 이름은 myFirstDB입니다. myFirstDB를 만들기 위한 명령문은 다음과 같습니다.

[코드 6-1] myFirstDB 데이터베이스 생성

● 3-1_createDatabase.sql

```
create database myFirstDB;
```

앞의 데이터베이스 생성 명령문을 입력한 후 **Enter**를 누르면 myFirstDB 데이터베이스가 생성됩니다.



```
bin — mysql • sudo — 119x5
mysql> create database myFirstDB;
Query OK, 1 row affected (0.03 sec)
mysql> █
```

그림 6-11 코드 6-1 쿼리문 실행 결과

데이터베이스가 잘 생성되었는지 확인하려면 다음과 같이 데이터베이스를 볼 수 있는 명령문을 사용합니다.

[코드 6-2] 데이터베이스 보기

● 3-2_showDatabases.sql

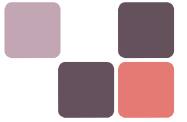
```
show databases;
```



```
bin — mysql • sudo — 119x16
mysql> create database myFirstDB;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| myFirstDB |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)

mysql> █
```



다음은 myFirstDB 데이터베이스를 사용하는 명령문입니다.

[코드 6-3] 데이터베이스 선택

● 3-3_useDatabase.sql

```
use myFirstDB;
```

```
bin — mysql • sudo — 119x6
mysql> use myFirstDB;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> █
```

그림 6-13 코드 6-3 쿼리문 실행 결과

한 개의 데이터베이스에서는 여러 가지 테이블을 만들어 사용할 수 있습니다. 테이블은 보통 하나의 기능에 대해 하나의 테이블을 생성해 사용합니다. 예를 들어, 회원정보를 갖고 있는 테이블, 작성한 게시글을 갖고 있는 테이블, 업로드한 이미지의 정보를 갖고 있는 테이블 등으로 구성될 수 있습니다.

필드의 데이터 크기와 데이터형 지정하기



5-1. 숫자형 데이터

수를 집어넣을 때의 데이터형에 대해 알아보겠습니다. 다음 표에서 저장 가능한 수는 음의 정수를 포함한 범위입니다. 해당 필드에 무조건 양의 정수만 넣는다고 가정했을 때 양의 정수만 입력 가능하게 하는 방법이 있으며, 이 방법을 사용하면 음의 정수 저장 공간은 필요하지 않으므로 그 수만큼 양의 정수 저장 공간이 늘어납니다.

데이터형	byte	저장 가능 수	양의 정수 전환 시 저장 가능 수(unsigned)
tinyint	1byte	-128 ~ 127	0 ~ 255
smallint	2byte	-32768 ~ 32767	0 ~ 65535
mediumint	3byte	-8388608 ~ 8388607	0 ~ 16777215
int 또는 integer	4byte	-2147483648 ~ 2147483647	0 ~ 4294967295
bigint	8byte	-9223372036854775808 ~ 9223372036854775807	0 ~ 18446744073709551615

실수를 사용하려면 float 또는 double을 사용해야 합니다.

필드의 데이터 크기와 데이터형 지정하기



5-2. 문자형 데이터

문자형 데이터는 테이블에 텍스트를 저장하기 위한 데이터형으로, 주소, 이름, 이메일 등의 정보를 테이블에 저장하기 위해 사용합니다.

데이터형	저장 가능 글자 수
char	255
varchar	255
tinytext	255
text	65535
mediumtext	16777215
longtext	4294967295
enum	지정된 값 중 한 가지 값만 사용 가능
set	지정된 값 중 여러 가지 값을 사용 가능

문자열의 데이터형을 사용하려면 데이터형을 적고 괄호를 연 후 사용할 글자 수를 지정해주어야 합니다.

필드의 데이터 크기와 데이터형 지정하기



char 데이터형을 이용하여 네 글자까지 저장 공간을 지정했을 때 세 글자를 입력하면 한 글자를 입력할 공간이 불필요하게 남게 됩니다. 하지만 varchar(4)를 적용한 후 세 글자를 입력하면 자동으로 남는 공간을 없애줍니다. 즉, 불필요한 공간이 없어지므로 효율적입니다. 그 대신 varchar은 char에 비해 속도 처리가 느리다는 단점이 있습니다.

• char(4)를 지정한 경우			
김	태	영	쓰지 않는 불필요 공간 발생
• varchar(4)를 지정한 경우			
김	태	영	쓰지 않는 공간 삭제하여 유용

그림 6-17 varchar과 char의 차이

tinytext, text, mediumtext, longtext는 지정된 크기에 한하여 사용할 수 있는 데이터형입니다. char이 너무 작아서 사용하지 못한다면 tinytext를 사용하고, tinytext가 너무 작아서 사용하지 못한다면 text를 사용하면 됩니다.

앞에서 enum은 지정된 값 중 하나만 사용할 수 있다고 했습니다. 예를 들어, 성별의 정보를 받는다고 가정하면 남성 또는 여성으로만 값이 정해져 있기 때문에 이러한 값을 받는 필드를 만들 때 유용합니다.

char을 사용하여 남성이라는 정보를 받거나 여성이라는 정보를 받으면 된다고 생각할 수 있지만, enum은 받는 값이 정해져 있기 때문에 해당 값을 좀 더 간단하게 저장합니다. 그러므로 효율성면에서 좋습니다. 또한 다른 값이 들어가는 사태를 원천적으로 막을 수 있다는 이점이 있습니다.

필드의 데이터 크기와 데이터형 지정하기

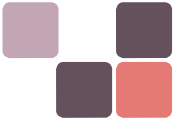


5-3. 날짜형 데이터

날짜형 데이터는 어떤 글이 언제 작성 완료되었는지, 어떤 회원이 언제 가입했는지, 누가 몇 시에 웹사이트에 로그인했는지 등과 같은 정보를 저장할 때 사용하는 데이터형입니다.

데이터형	저장 범위	표시 형식	크기
date	1000-01-01 ~ 9999-12-31	YYYY-MM-DD	3byte
datetime	1000-01-01 00:00:00 ~ 9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS	8byte
timestamp	1970-01-01 00:00:00 ~ 2038-01-19 03:14:07	YYYYMMDDHHMMSS	4byte
time	-838:59:59 ~ 838:59:59	HH:MM:SS	3byte
year	1901~2155	YYYY	1byte

date는 년도, 월, 일만 저장하는 데이터형이며, datetime은 년도, 월, 일, 시, 분, 초까지 저장하는 데이터형입니다. timestamp는 리눅스에서 사용하는 시간인데, 리눅스에서는 1970년 1월 1일 0시 0분 0초부터 시간을 세기 시작했습니다. 그러므로 저장 범위가 1970년 1월 1일 0시 0분 0초부터 시작합니다. time은 시 분 초만 저장하는 데이터형이며 year는 연도만 저장하는 데이터형입니다.



Thank You