jinyPHP 2권 - 029. 권환설정

**29. 권한설정**

====================

MYSQL은 설치시 기본적으로 mysql 데이터베이스명 하나 존재합니다. mysql 데이터베이스명 안에는 MYSQL 사용자 계정, 패스워드, 권한, DB 구조등 다양한 설정정보들이 들어가 있습니다.

이러한 설정정보에 사용자와 권한을 추가하여 다수의 사용자가 하나의 MYSQL 서버에 접속하여 작업을 할 수 있습니다.

**29.1 사용자 추가**

====================

1개의 서버에 하나의 서비스만 동작을 할때는 사용자 라는 개념이 없을 수 있습니다. MYSQL 서버에는 다수의 서비스를 운영할 수도 있는데, 보안상 하나의 계정정보를 공동으로 이용하는 것은 권장하지는 않습니다.

MYSQL은 root 사용자 계정외에 별도의 사용자 계정을 추가할 수 있습니다. 호스팅처럼 여러 사용자에게 MYSQL 서버의 서비스를 오픈해야 하는 경우라면 사용자를 추가하는 기능은 매우 중요합니다.

**29.1.1 CREATE USER**

====================

create user 명령은 쉽게 사용자를 추가할 수 있습니다. 아이디와 @ 그리고 접속 서버를 설정하면 됩니다.

|  |
| --- |
| mysql> create user hojin@localhost identified by '!@abcd';  Query OK, 0 rows affected (0.01 sec) |

localhost는 MYSQL 서버를 자체 서버 내에서만 접속을 할 수 있습니다. 만일 외부서버에서 접속을 허용하고 싶다면 localhost 대신에 %를 사용합니다. %는 임의의 모든 IP 를 의미합니다.

|  |
| --- |
| mysql> create user jiny@'%' identified by '!@abcd';  Query OK, 0 rows affected (0.01 sec) |

**29.1.2 직접 등록**

====================

insert, update 명령을 이용하여 사용자를 추가할 수도 있습니다. insert 방식으로 사용자를 추가할 때는 별도의 DB 갱신이 필요합니다.

|  |
| --- |
| mysql> insert into user (Host, User,Password) values('localhost','info',password('!@abcd'));  Query OK, 1 row affected, 3 warnings (0.00 sec)  mysql> flush privileges;  Query OK, 0 rows affected (0.01 sec)  mysql> select Host, User from user;  +-----------+-------+  | Host | User |  +-----------+-------+  | % | jiny |  | 127.0.0.1 | root |  | ::1 | root |  | localhost | hojin |  | localhost | info |  | localhost | root |  +-----------+-------+  6 rows in set (0.00 sec) |

**29.1.3 변경사항 적용**

====================

FLUSH 명령은 MYSQL의 내부 캐쉬를 지우고 새로운 캐쉬를 로드하는 명령어 입니다. flush 명령을 사용하기 위해서는 MYSQL 계정 사용자는 reload 권환이 있어야 합니다.

* **PRIVILEGES**

옵션은 사용자 권한테이블을 다시 읽어 들이는 작업을 합니다. 만일 INSERT 형태로 MYSQL 회원을 추가하거나 변경을 했을 경우에 이를 적용을 해주어야 합니다.

이때 적용부분을 갱신하는 명령은 plush privileges 입니다.

|  |
| --- |
| mysql> plush prilvileges; |

flush privileges 명령은 grant 테이블을 재로드 함으로 변경사항을 바로 적용이 가능하도록 하는 기능입니다. 하지만 insert, update 문을 사용하지 않고 grant 명령을 통하여 작업을 한 경우에는 이명령으로 갱신을 하지 않아도 됩니다.

또는 쉘명령에서

]# mysqladmin reload

형태로 갱신을 할 수 있습니다.

* LOGS

현재의 로그파일을 지우고 새롭게 생성을 합니다. 로그파일명을 지정하지 않으면 기존 이름에 +1 을 증가된 파일명이 생성이 됩니다.

|  |
| --- |
| mysql> flush logs;  Query OK, 0 rows affected (0.00 sec) |

* TABLES

열려진 모든 테이블을 닫습니다. MYSQL에서 테이블을 오픈하여 사용하는 것은 캐시 크기와 상관이 있습니다. 많은 접속 스레드의 생성과 테이블을 오픈할때 캐시의 용량부족으로 정상적인 처리가 되지 않을 수 있습니다.

|  |
| --- |
| mysql> flush tables;  Query OK, 0 rows affected (0.05 sec) |

* STATUS

모든 상태변수를 0으로 초기화 합니다.

|  |
| --- |
| mysql> flush status;  Query OK, 0 rows affected (0.00 sec) |

* HOSTS

HOST 테이블을 다시 읽어 들입니다. 만일 접속 호스트의 IP 주소가 변경이 될경우 사용하면 좋은 옵션입니다.

|  |
| --- |
| mysql> flush hosts;  Query OK, 0 rows affected (0.00 sec) |

**29.2 권한확인**

====================

MYSQL에서 추가한 사용자 각각의 권한은 SHOW GRANTS 명령으로 확인을 할 수 있습니다.

|  |
| --- |
| mysql> show grants;  +-----------------------------------------------------------------------------------------------------+  | Grants for root@localhost |  +-----------------------------------------------------------------------------------------------------+  | GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost' IDENTIFIED BY PASSWORD '\*822DBCD38A3CD19B2C3404B458CD48525B846E22' WITH GRANT OPTION |  | GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION |  +-----------------------------------------------------------------------------------------------------+  2 rows in set (0.00 sec) |

권한을 설정하는 방법은 grant 명령을 이용하여 처리할 수 있습니다. 만일 권한을 설정과 비밀번호를 같이 할경우에는 뒤에 **identified by '비밀번호'** 를 추가하시면 됩니다.

특정 계정에 대한 권한만을 확인하고자 할때는 SHOW GRANTS 뒤에 for 계정 정보를 넣어 주시면 됩니다.

|  |
| --- |
| mysql> show grants for hojin@localhost;  +-----------------------------------------------------------------------------------------------------+  | Grants for hojin@localhost |  +-----------------------------------------------------------------------------------------------------+  | GRANT USAGE ON \*.\* TO 'hojin'@'localhost' IDENTIFIED BY PASSWORD '\*A51F6BE471B9F2B7CB6F8A21ED1A4EA61846A50A' |  +-----------------------------------------------------------------------------------------------------+  1 row in set (0.00 sec) |

**29.2.1 사용자 DB권한**

====================

호스팅과 같이 하나의 서버에서 다수의 MYSQL 사용자와 데이터베이스가 있을 경우 기존 insert 방식으로 사용자나 DB 를 추가하면 다른 사용자의 DB가 접속되어 목록으로 볼 수 있습니다.

하지만, grant 명령을 통하여 사용자 DB 권한을 설정하면 해당사용자에 한하여 DB접속 권한 및 목록을 출력합니다.

|  |
| --- |
| mysql> grant all privileges on DB명.\* to '유저명'@'localhost' identified by '비밀번호'; |

위의 예는 기존에 등록된 MYSQL 계정의 권한과 비밀번호를 변경합니다.

**29.2.2 일부권한**

====================

MYSQL 사용자 계정에게 일부 기능에 대한 권한만 부여할 수도 있습니다.

|  |
| --- |
| mysql> grant select, insert, update on DB명.\* to '유저명'@'localhost' identified by '비밀번호'; |

위의 예는 해당 계정은 select, insert, update 만 가능하다는 의미입니다.

**29.2.3 권한삭제**

====================

권한을 제거할 때는 REVOKE 명령을 사용합니다.

|  |
| --- |
| mysql> revoke all on DB명.테이블 from 사용자ID  mysql> revoke all on test.\* from hojin@localhost; |

**29.3 계정삭제**

====================

MYSQL에 등록한 사용자를 삭제할 수 있습니다.

**]# drop user 사용자ID**

|  |
| --- |
| mysql> drop user hojin@localhost;  Query OK, 0 rows affected (0.00 sec) |

**29.4 접속관리**

====================

MYSQL의 각각의 접속은 독립적인 쓰레드로 동작을 합니다. 또한 멀티 쓰레드를 생성할 수 있습니다. 멀티 쓰레드를 통하여 동시에 같은 테이블에 여러 쿼리를 처리할 수 있습니다.

모든 쓰레드에서 인덱스 파일은 공유합니다.

**29.4.1 ProcessList**

====================

현재 접속하여 동작하는 쓰레드를 확인할 수 있습니다.

|  |
| --- |
| mysql> show processlist;  +----+------+-----------------+------+---------+------+-------+------------------+  | Id | User | Host | db | Command | Time | State | Info |  +----+------+-----------------+------+---------+------+-------+------------------+  | 79 | root | localhost:62222 | jiny | Query | 0 | init | show processlist |  | 80 | root | localhost:62401 | jiny | Sleep | 1168 | | NULL |  +----+------+-----------------+------+---------+------+-------+------------------+  2 rows in set (0.01 sec) |

쓰레드 접속 확인은 쉘 명령창에서도 확인을 할 수 있습니다.

**]# mysqladmin processlist**

|  |
| --- |
| C:\Bitnami\wampstack-5.6.30-0\mysql\bin>mysqladmin -u root -p processlist  Enter password: \*\*\*\*\*\*\*\*  +----+------+-----------------+------+---------+------+-------+------------------+  | Id | User | Host | db | Command | Time | State | Info |  +----+------+-----------------+------+---------+------+-------+------------------+  | 79 | root | localhost:62222 | jiny | Sleep | 152 | | |  | 85 | root | localhost:63007 | | Query | 0 | init | show processlist |  +----+------+-----------------+------+---------+------+-------+------------------+ |

**29.4.2 접속해제**

====================

KILL 명령은 사용하지 않는 스레드를 강제로 종료할 수 있습니다.

|  |
| --- |
| mysql> KILL 80;  Query OK, 0 rows affected (0.00 sec)  mysql> show processlist;  +----+------+-----------------+------+---------+------+-------+------------------+  | Id | User | Host | db | Command | Time | State | Info |  +----+------+-----------------+------+---------+------+-------+------------------+  | 79 | root | localhost:62222 | jiny | Query | 0 | init | show processlist |  +----+------+-----------------+------+---------+------+-------+------------------+  1 row in set (0.00 sec) |

사용하지 않은 스레드를 정리함으로써 MYSQL의 접속 및 성능을 놓여 충분한 자원 관리를 할 수 있습니다.