학습목표

1. 권한 설정
2. 역할 설정

학습내용

- 권한 설정에 대해 배울 수 있습니다.
- 역할의 개념에 대해 배울 수 있습니다.

사전퀴즈

1. 사용자를 추가하기 위해서는 mysql.user테이블에 레코드를 추가하면 된다. (insert into)
정답 : X
해설 : MariaDB에서는 사용자를 추가하려면 CREATE USER 명령어를 사용해야 한다.

2. MariaDB의 원격접속포트는 3306번이다.
정답 : O
해설 : MariaDB은 서버/클라이언트 구조로 되어 있고 3306번을 통해 원격접속이 가능하다.

수업

1. 권한설정

접근권한(Access Control) 설정

- DCL(Data Control Language)
  · 권한 및 역할 설정하는 언어
  · 특정 테이블에 대한 CRUD(Create/Retrieve/Update/Delete) 권한 설정
  * 권한부여(GRANT)/권한회수(REVOKE)로 나뉨
  * 주로 DBA(DataBase Adminitor)가 설정

1) 로컬시스템/외부유저 등록

(현재방식)
create user '사용자명'@'호스트' identified by '비밀번호' ;
* 호스트
사용자 접속 주소/위치를 뜻한다. 예) 로컬접속/원격접속/LAN 접속 등
사용자 접속 주소/위치에 맞게 사용자들이 각각 등록되어야 한다.
즉, 같은 사용자라고 하더라도 어느 주소/위치에서 접속하느냐에 따라서 다른 권한/역할을 가질 수 있다.

(예전방식)
insert into user(host, user, password, ssl_cipher, x509_issuer, x509_subject)
values('호스트','사용자명',password('비밀번호'),'','','') ;

2) 사용자 삭제

drop user 사용자명@호스트 ;

3) 반영하기

flush privileges ;

새로운 사용자 레코드를 user 테이블에 추가했다고 해서 시스템에 바로 반영된 것이 아님

사용자가 실제로 권한을 부여받고 테이블 생성 등의 작업을 하려면 사용자 등록을 시스템에 반영해야 함

4) 사용자 권한 확인

show grants for '사용자명'@'호스트' ;

예) show grants for 'sampleUser'@'localhost' ;

동일 시스템에서 접속하는 sampleUser 사용자의 모든 권한 확인

5) 사용자에게 권한 부여

grant [all/CRUD(테이블의 컬럼명)] on [*/DB명].[*/테이블명] to '사용자명'@'호스트' identified by '비밀번호' ;

호스트가

localhost인 경우 → localhost

로컬랜인 경우 → 대역대 (예: 192.168.0.%)

인터넷 전체인 경우 → %

6) 사용자의 권한 삭제

revoke [all/CRUD] on [*/DB명].[*/테이블명] from '사용자명'@'호스트' ;

실제 예제 참고

https://blog.naver.com/jihye_sally_yoon/221407838657

예제 1-1

사용자 sampleUser를 새로 하나 추가/삭제하시오

```
use mysql
;

-- 로컬시스템/외부유저등록
create user 'sampleUser'@'localhost' identified by 'sampleUser';

-- 반영하기
flush privileges
;

-- 유저 확인!
select user, host
from user
;
```

user (2×6)

user	host
root	
root	%
root	127.0.0.1
root	::1
root	localhost
sampleUser	localhost

```
-- 사용자삭제
drop user 'sampleUser'@'localhost';

-- 유저 확인
select user, host
from user
;
```

user (2×5)

user	host
root	
root	%
root	127.0.0.1
root	::1
root	localhost

예제 1-2

sampleUser(localhost/로컬랜/인터넷전체)의 접속을 허용하시오.

1) localhost

```
grant all on world.* to 'sampleUser'@'localhost' identified by 'sampleUser';
```

→ world DB에 있는 모든 테이블 접속/CRUD 권한을 동일시스템에서 접속하는 sampleUser 사용자에게 부여한다.

2) 로컬랜

```
36
37 grant all on *.* to 'sampleUser'@'192.168.0.%' identified by 'sampleUser';
38
```

3) 인터넷전체

```
40
41 grant all on world.* to 'sampleUser'@'%' identified by 'sampleUser';
42
```

예제 1-3

localhost의 sampleUser에게 worldDB의 검색/추가권한을 부여하시오.

```
44
45 grant select,insert on world.* to 'sampleUser'@'localhost' identified by 'sampleUser';
46
```

예제 1-4

localhost의 sampleUser에게 worldDB의 city 테이블의 도시명(Name)의 업데이트 권한을 부여하시오.

```
48
49 grant update(name) on world.city to 'sampleUser'@'localhost' identified by 'sampleUser';
50
```

예제 1-5

localhost의 sampleUser의 모든 권한을 삭제하시오.

```
52
53 revoke all on *.* from 'sampleUser'@'localhost';
54
```

2. 역할설정

- 개별 테이블에 대한 CRUD 권한을 사용자별로 설정하면 경우의 수가 테이블 수 X 사용자 수의 조합이 생김
- 이런 문제점을 개선하기 위해서 롤(역할)을 정하고 역할별 권한 설정을 하고 사용자에게 역할을 부여하는 형태로 사용
- 사용자가 여러 개의 역할을 가지는 것이 가능함 (관리자, 사용자 등)

역할 생성

CREATE ROLE 역할명;

역할에 대해 권한 설정

GRANT CRUD ON 테이블명 TO 역할명;

사용자에게 역할 부여

GRANT 역할 TO 사용자명;

예제 2-1

world DB를 다루는 world_admin 역할을 생성하시오.

```
55
56 create role world_admin;
57
```

예제 2-2

world_admin 역할에 city 테이블 추가/삭제 권한을 부여하시오.

```
58
59 grant insert,delete on world.city to world_admin;
60
```

예제 2-3

sampleUser에게 world_admin 역할을 부여하시오.

```
62
63 grant world_admin to sampleUser;
64
```

3. MariaDB 원격접속 설정

- MariaDB을 동일시스템 외에 접근 가능하도록 설정
- 사용자를 원격사용자로 등록

  GRANT ALL PRIVILEGES ON DB명.테이블명 TO '사용자명' @'%' IDENTIFIED BY '비밀번호' ;
- my.ini 수정 (bind-address 부분 주석처리)

  # Instead of skip-networking the default is now to listen only on

  # localhost which is more compatible and is not less secure

  # bind-address = 127.0.0.1

  나는 윈도우용 MariaDB 10.3을 default 설정으로 설정했다.

  그래서 my.ini 파일 위치는 C:₩Program Files₩MariaDB 10.3₩data₩my.ini

  my.ini 파일을 열어보니 bind-address 설정 변수 자체가 처음부터 없었다.

- MariaDB 서버 재시작
- 방화벽 3306 포트 열기

역할(ROLE)이 있는 경우와 없는 경우의 차이점

전문가의견

DCL(Data Control Language)은 테이블/DB 별로 특정 사용자의 접근 권한을 지정합니다.

그리고 CRUD(Create, Retrieve, Update, Delete) 권한에 대해서도 접근권한을 지정하고 컬럼별로 접근권한을 지정할 수도 있습니다.

만약 ROLE이 지원되지 않으면(MySQL의 경우) 사용자 수 X DB 수 X 테이블 수 X CRUD X 컬럼별로 값들을 세부적으로 지정해줘야 합니다.
ROLE이 지원되면 사용자에게는 ROLE을 부여하고 ROLE의 접근권한을 명시하면 됩니다. 복잡도가 줄어들면서 개별 사용자의 역할을 지정하는 것이 명쾌합니다. 예를 들면, 특정 사용자가 관리자이면서 동시에 일반사용자라면 ROLE이 있는 경우는 간단하지만 없는 경우는 복잡한 경우의 수가 발생하게 됩니다.