

• DB구현 •

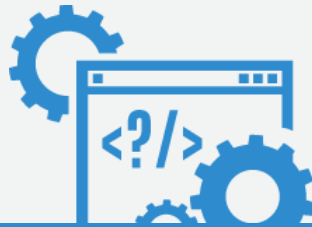
권한과 사용자 정의





학습내용

- ❖ 권한 시스템
- ❖ DCL 구문을 이용한 사용자와 권한 관리
- ❖ GUI 방식을 이용한 사용자와 권한 관리



학습목표

- ❖ 권한의 개념과 필요성에 대해 설명할 수 있다.
- ❖ DCL 구문을 이용한 사용자와 권한 관리를 할 수 있다.
- ❖ GUI 방식을 이용한 사용자와 권한 관리를 할 수 있다.

권한 시스템

권한의 개념

➤ 권한의 정의

권한

명령어 실행 또는 데이터를 접근하거나 사용하는 허가 여부

➤ MySQL의 권한

- 허가 되지 않은 사용자가 DBMS에 접속하거나 사용자의 무분별한 데이터 접근을 제어
- 사용자들의 모든 접속 시도 및 질의 실행 동작에 대한 접근 제어 리스트(ACL)기반의 보안정책을 운용

➤ MySQL 권한 시스템의 기능

접속 권한

테이블
접근 권한

명령어
실행 권한

권한 시스템

권한 시스템의 구성요소

➤ 시스템 테이블

- MySQL 스키마에 위치
- 사용자 및 시스템 관리를 위한 데이터 저장

| 시스템 테이블 | 역할 |
|---------|-----------------------------------|
| 권한 | 사용자 계정 및 권한과 관련된 정보를 저장 |
| 객체 정보 | 저장 프로그램, 사용자 정의 함수 등의 DB 객체 정보 저장 |
| 로그 | 로그에 필요한 정보를 저장 |
| 표준 시간대 | 시간대(time zone)과 관련된 정보를 저장 |
| 최적화 | 최적화기 수행에 필요한 정보를 저장 |
| 기타 | DB 감사, 방화벽, DB 엔진 등에 필요한 정보를 저장 |

➤ 권한 시스템을 위한 시스템 테이블

User 테이블

DB 테이블

Tables_priv
테이블

Columns_priv
테이블

권한 시스템

권한 시스템의 구성요소

➤ 권한 시스템을 위한 시스템 테이블

User 테이블

호스트, 사용자 아이디, 비밀번호,
명령어 별 실행권한 관리

| Field | Type | Null | Key | Default |
|-------------|---------------|------|-----|---------|
| Host | char(60) | NO | PRI | |
| User | char(16) | NO | PRI | |
| Password | char(41) | NO | | |
| Select_priv | enum('N','Y') | NO | | N |
| Insert_priv | enum('N','Y') | NO | | N |
| Update_priv | enum('N','Y') | NO | | N |
| Delete_priv | enum('N','Y') | NO | | N |
| Create_priv | enum('N','Y') | NO | | N |
| Drop_priv | enum('N','Y') | NO | | N |

User 테이블

DB 테이블

스키마 내에서
명령어 별 실행권한 관리

| Field | Type | Null | Key | Default |
|-------------|---------------|------|-----|---------|
| Host | char(60) | NO | PRI | |
| Db | char(64) | NO | PRI | |
| User | char(16) | NO | PRI | |
| Select_priv | enum('N','Y') | NO | | N |
| Insert_priv | enum('N','Y') | NO | | N |
| Update_priv | enum('N','Y') | NO | | N |
| Delete_priv | enum('N','Y') | NO | | N |
| Create_priv | enum('N','Y') | NO | | N |
| Drop_priv | enum('N','Y') | NO | | N |

DB 테이블

권한 시스템

권한 시스템의 구성요소

➤ 권한 시스템을 위한 시스템 테이블

Tables_priv
테이블

테이블 별 명령어 실행권한 관리

| Field | Type | Null | Key | Default |
|-------------|-------------------------------|------|-----|---------|
| Host | char(60) | NO | PRI | |
| Db | char(64) | NO | PRI | |
| User | char(16) | NO | PRI | |
| Table_name | char(64) | NO | PRI | |
| Grantor | char(77) | NO | MUL | |
| Timestamp | timestamp | NO | | CUR... |
| Table_priv | set('Select','Insert','Upd... | NO | | |
| Column_priv | set('Select','Insert','Upd... | NO | | |

Tables_priv 테이블

Columns_priv
테이블

컬럼 별 명령어 실행권한 관리

| Field | Type | Null | Key | Default |
|-------------|-------------------------------|------|-----|---------|
| Host | char(60) | NO | PRI | |
| Db | char(64) | NO | PRI | |
| User | char(16) | NO | PRI | |
| Table_name | char(64) | NO | PRI | |
| Column_name | char(64) | NO | PRI | |
| Timestamp | timestamp | NO | | CUR... |
| Column_priv | set('Select','Insert','Upd... | NO | | |

Columns_priv 테이블

권한 시스템

권한 시스템의 동작과정

➤ 사용자 접속 인증

- 사용자의 호스트와 사용자명을 확인
- User 테이블에서 Host, User, Password 컬럼값을 검사하여 접속 여부 결정

| Host | User | 대상 |
|-------------------|------------|--|
| 'koreatech.ac.kr' | 'student1' | koreatech.ac.kr로부터 접속한 student1이라는 사용자 |
| 'koreatech.ac.kr' | " | koreatech.ac.kr로부터 접속한 모든 사용자 |
| '%' | " | 모든 호스트의 사용자 |

DCL 구문을 이용한 사용자와 권한 관리

사용자 추가/삭제

➤ 사용자 추가

- 시스템 테이블인 User 테이블에 INSERT INTO 문을 사용하여 사용자 추가가 가능하나 INSERT 문의 오류로 시스템 관리에 치명적 문제를 발생시킬 수 있음
- 사용자와 사용자의 권한을 CREATE USER 문 과 GRANT 문 을 사용하여 관리하는 방식을 권장

➤ CREATE USER 구문형식

구문형식

```
CREATE USER <사용자명>  
IDENTIFIED BY '패스워드'
```

➤ 다수 사용자 요청 처리

| 표기 | 의미 |
|------------------------|---------------------------------------|
| cjh. 'localhost' | localhost에서 접속하는 cjh 사용자만 접속 가능 |
| cjh. 'koreatech.ac.kr' | koreatech.ac.kr에서 접속하는 cjh 사용자만 접속 가능 |
| cjh. '%' | 모든 호스트에서 접속하는 cjh 사용자 접속 가능 |

DCL 구문을 이용한 사용자와 권한 관리

사용자 추가/삭제

➤ 사용자 삭제

- User 테이블에 DELETE 문을 사용하여 사용자 삭제가 가능하나 오류로 문제가 발생할 소지가 있음
- DROP USER 문을 사용하여 사용자를 삭제하는 방식을 권장

➤ DROP USER 구문 형식

구문형식

```
DROP USER <사용자명>
```

DCL 구문을 이용한 사용자와 권한 관리

권한 부여/회수

➤ 권한 부여

- 직접 시스템 테이블을 조작은 매우 위험한 방법
- 사용자에게 권한을 부여 시 GRANT 명령어를 사용

➤ GRANT 명령어 구문형식

구문형식

```
GRANT <권한>
    [(<컬럼 리스트>)] ON <데이터베이스 혹은 테이블>
    TO <사용자명>@<호스트명>
    [IDENTIFIED BY '<비밀번호>']
    [WITH GRANT OPTION]
```

➤ 데이터베이스 및 테이블 표기방법

| 표기 | 의미 |
|--------------|--------------------------|
| koreatech.강의 | koreatech 데이터베이스의 강의 테이블 |
| koreatech.* | koreatech 데이터베이스의 모든 테이블 |
| *.* | 모든 데이터베이스의 모든 테이블 |

➤ GRANT 문의 예

- localhost에서 접속하며 student라는 사용자에게 koreatech 스키마에 대한 모든 권한을 부여하시오.

명령

```
GRANT ALL PRIVILEGES ON koreatech.*
    TO student@'localhost'
```

DCL 구문을 이용한 사용자와 권한 관리

권한 부여/회수

➤ GRANT 문의 예

- localhost에서 접속하며 student라는 사용자에게 koreatech 스키마의 강의 테이블에 대한 SELECT 권한을 부여하시오.

명령

```
GRANT SELECT ON koreatech.강의  
TO student@'localhost'
```

➤ 권한 회수

- 권한 부여와 마찬가지로 시스템 테이블을 직접 조작하는 것은 매우 위험
- 사용자에게 권한을 회수 시 REVOKE 명령어를 사용

➤ REVOKE 명령어 구문형식

구문형식

```
REVOKE <권한>  
[(<컬럼 리스트>)] ON <데이터베이스 혹은 테이블>  
FROM <사용자명>@'<호스트명>'
```

DCL 구문을 이용한 사용자와 권한 관리

권한 부여/회수

➤ REVOKE 문의 예

- localhost의 student2에게 부여된 koreatech 스키마의 교수 테이블의 교수이름 컬럼에 대한 SELECT 권한을 제거하시오



DCL 구문을 이용한 사용자와 권한 관리

The screenshot shows the MySQL Workbench interface. In the 'Query' editor, the following SQL statement is entered:

```
1 CREATE USER 'student'@'localhost' IDENTIFIED BY '12345'
```

The 'Output' pane at the bottom displays the execution results:

| Time | Action | Message | Duration / Fetch |
|-------------|---|--|-----------------------|
| 7 15:43:26 | SELECT * FROM 사원 LIMIT 0, 1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 8 15:49:34 | START TRANSACTION | 0 row(s) affected | 0.000 sec |
| 9 15:51:03 | LOCK TABLES 사원 WRITE | 0 row(s) affected | 0.000 sec |
| 10 15:52:23 | UPDATE 사원 SET 생년월일 = '1995-12-10' WHERE 사원번호 = 'EMP001' | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 | 0.000 sec |
| 11 15:52:53 | UNLOCK TABLES | 0 row(s) affected | 0.000 sec |
| 12 15:53:33 | COMMIT | 0 row(s) affected | 0.000 sec |

A blue banner at the bottom of the screenshot contains the text: **CREATE USER를 이용하여 사용자 이름, 암호 등을 사용**

The screenshot shows the MySQL Workbench interface. In the 'Query' editor, the following SQL statements are entered:

```
1 CREATE USER 'student'@'localhost' IDENTIFIED BY '12345';  
2  
3 GRANT INSERT ON company.* TO 'student'
```

The 'Output' pane at the bottom displays the execution results:

| Time | Action | Message | Duration / Fetch |
|-------------|---|--|------------------|
| 8 15:49:34 | START TRANSACTION | 0 row(s) affected | 0.000 sec |
| 9 15:51:03 | LOCK TABLES 사원 WRITE | 0 row(s) affected | 0.000 sec |
| 10 15:52:23 | UPDATE 사원 SET 생년월일 = '1995-12-10' WHERE 사원번호 = 'EMP001' | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 | 0.000 sec |
| 11 15:52:53 | UNLOCK TABLES | 0 row(s) affected | 0.000 sec |
| 12 15:53:33 | COMMIT | 0 row(s) affected | 0.000 sec |

A blue banner at the bottom of the screenshot contains the text: **GRANT (권한)을 사용하여 권한 부여**



DCL 구문을 이용한 사용자와 권한 관리

The screenshot displays the MySQL Workbench interface with the following SQL queries in the query editor:

```
1 CREATE USER 'student'@'localhost' IDENTIFIED BY '12345';
2
3 GRANT INSERT ON company.* TO 'student'@'localhost';
4
5 GRANT SELECT(사원번호, 사원이름, 전화번호) ON company.사원
6   TO 'student'@'localhost';
7
8 REVOKE SELECT(전화번호) ON company.사원
9   FROM 'student'@'localhost';
10
```

The Output window shows the execution results of these queries:

| Time | Action | Message | Duration / Fetch |
|-------------|---|--|------------------|
| 10 15:52:23 | UPDATE 사원 SET 생년월일 = '1985-12-10' WHERE 사원번호 = 'EMP001' | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 | 0.000 sec |
| 11 15:52:59 | UNLOCK TABLES | 0 row(s) affected | 0.000 sec |
| 12 15:53:33 | COMMIT | 0 row(s) affected | 0.000 sec |
| 13 15:53:35 | CREATE USER 'student'@'localhost' IDENTIFIED BY '12345' | 0 row(s) affected | 0.000 sec |
| 14 16:00:56 | GRANT INSERT ON company.* TO 'student'@'localhost' | 0 row(s) affected | 0.000 sec |
| 15 16:01:23 | GRANT SELECT(사원번호, 사원이름, 전화번호) ON company.사원 TO 'student'@'localhost' | 0 row(s) affected | 0.000 sec |
| 16 16:01:23 | REVOKE SELECT(전화번호) ON company.사원 FROM 'student'@'localhost' | 0 row(s) affected | 0.000 sec |

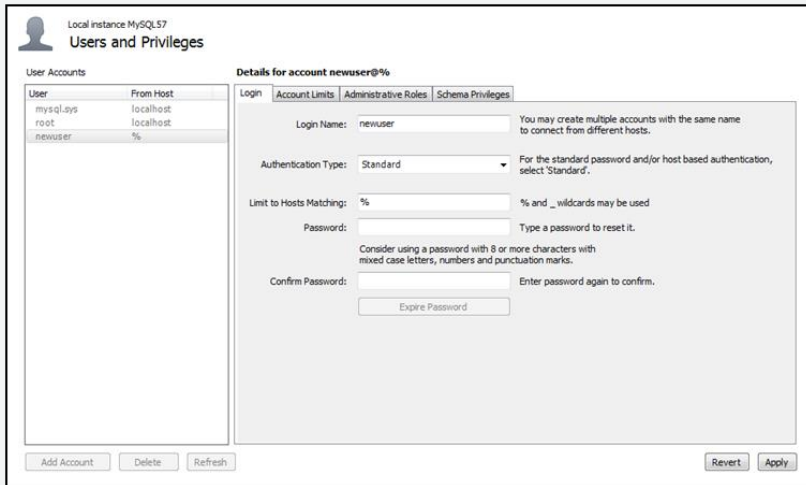
At the bottom of the screenshot, a text overlay reads: **REVOKE <권한>을 사용하여 권한 회수**

GUI 방식을 이용한 사용자와 권한 관리

Users and Privileges 메뉴

➤ Users and Privileges

- 사용자 추가 및 제거를 비롯하여 사용자의 권한을 GUI 방식으로 관리할 수 있는 메뉴



➤ Users and Privileges의 기능

Login 탭

사용자의 추가, 변경 또는 삭제 기능을 담당

Account Limits 탭

사용자가 SQL 문을 요청할 수 있는 횟수, 연결할 수 있는 횟수 등을 결정

Administrative Roles 탭

역할을 생성하여 일괄적으로 권한을 부여

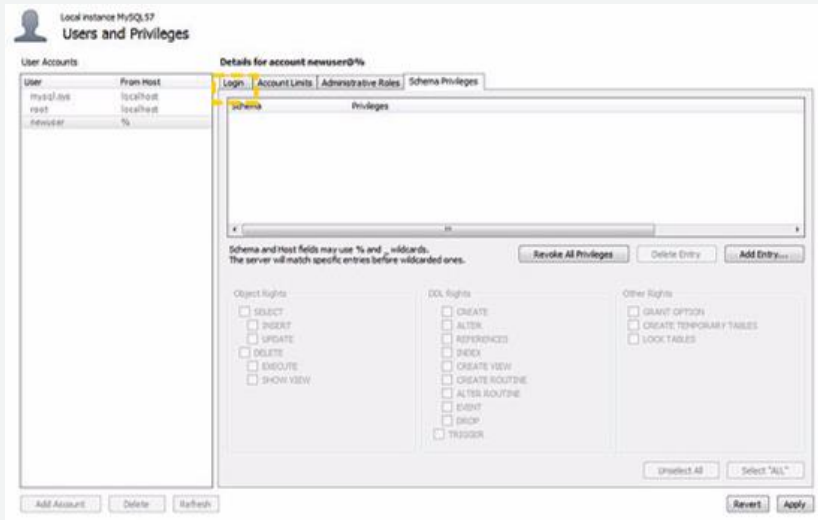
Schema Privileges 탭

사용자 별 권한을 세세하게 부여

GUI 방식을 이용한 사용자와 권한 관리

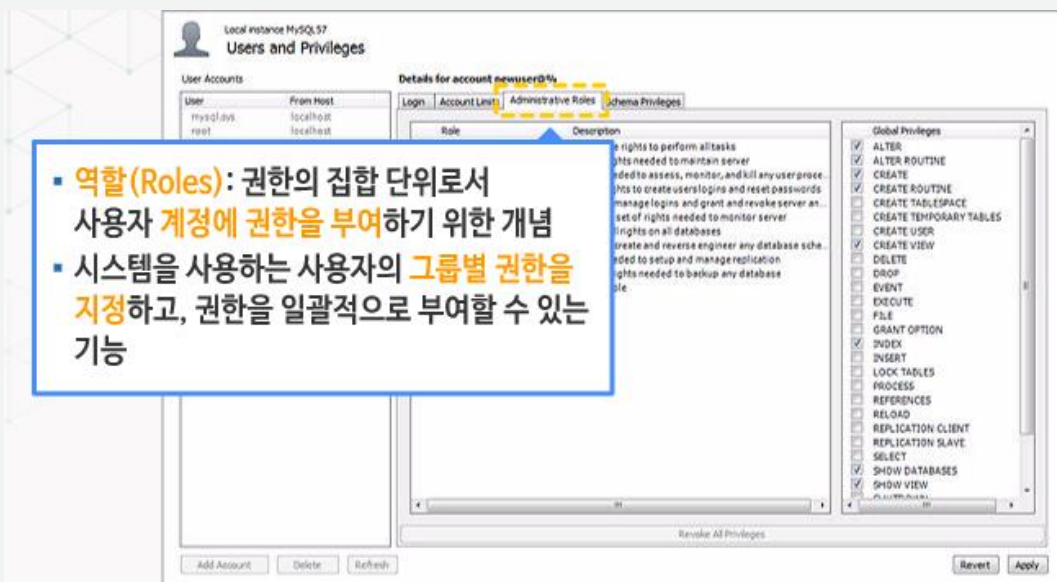
사용자 추가/삭제

➤ Login 탭



- Login 탭에서는 사용자를 추가하거나 삭제할 수 있습니다

➤ Administrative Roles 탭

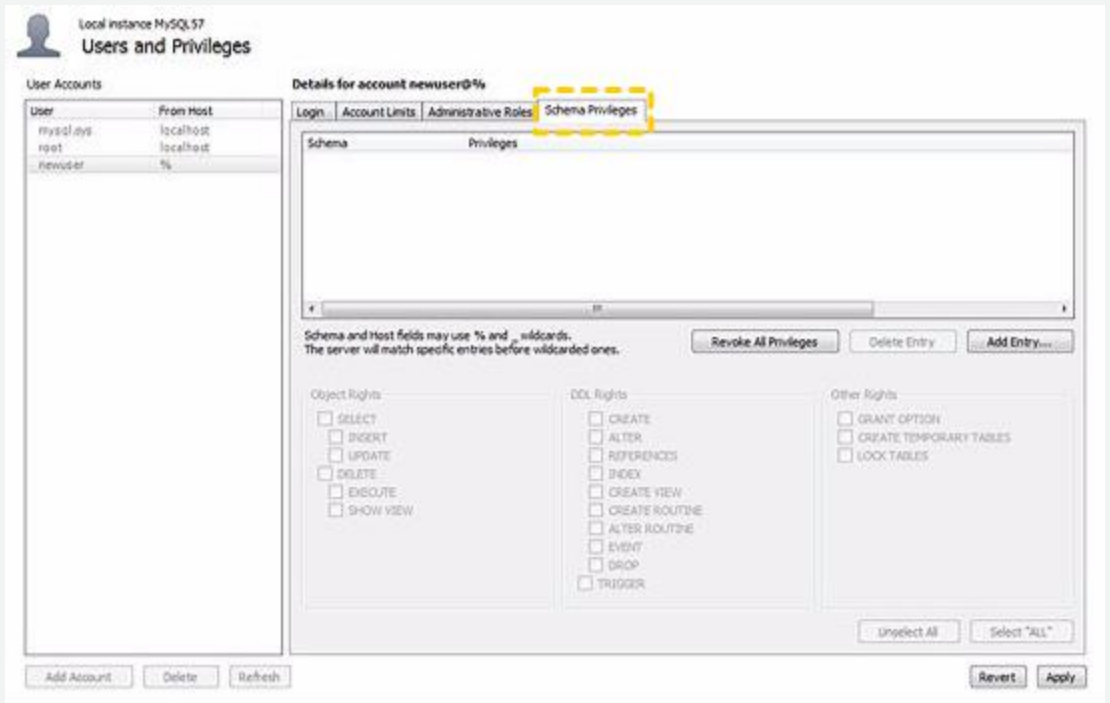


- Administrative Roles 탭은 Role이란 ‘역할’을 나타내며, 집합 단위로 사용자의 계정에 권한을 부여할 수 있습니다.

GUI 방식을 이용한 사용자와 권한 관리

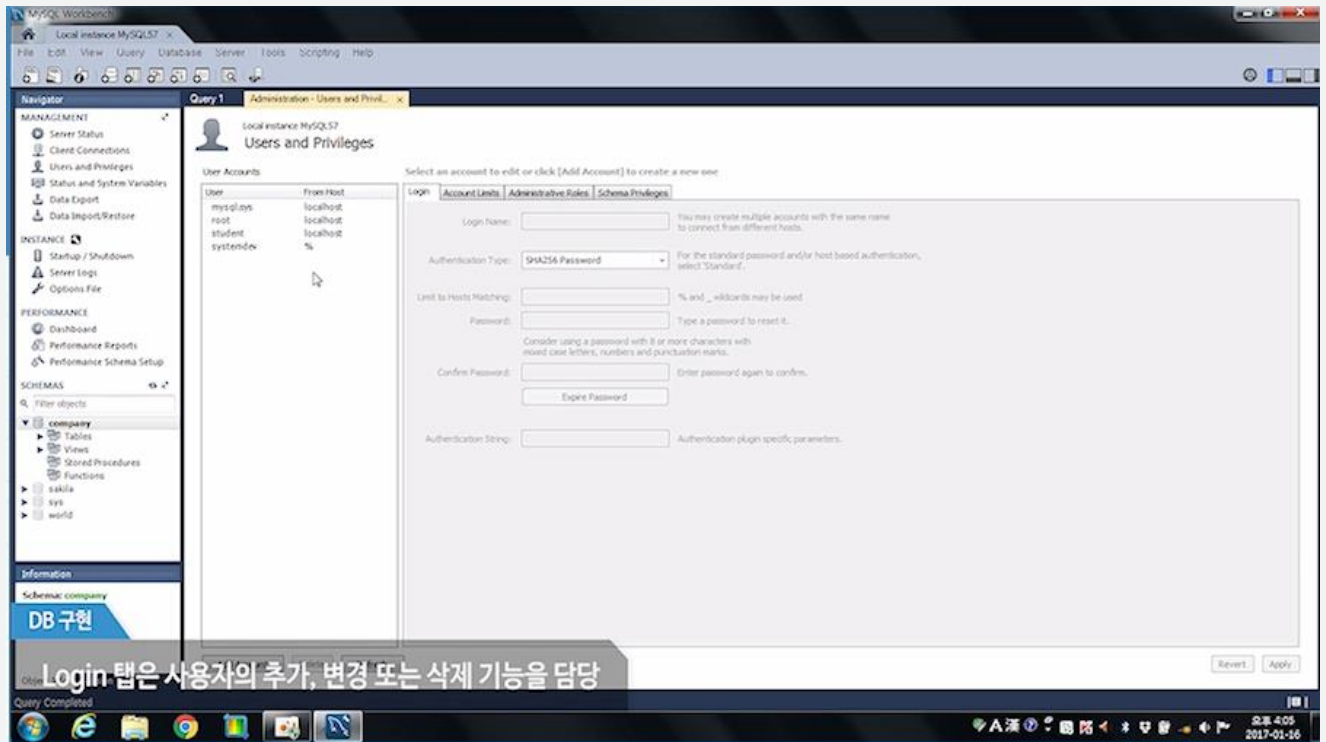
사용자 추가/삭제

➤ Schema Privileges 탭

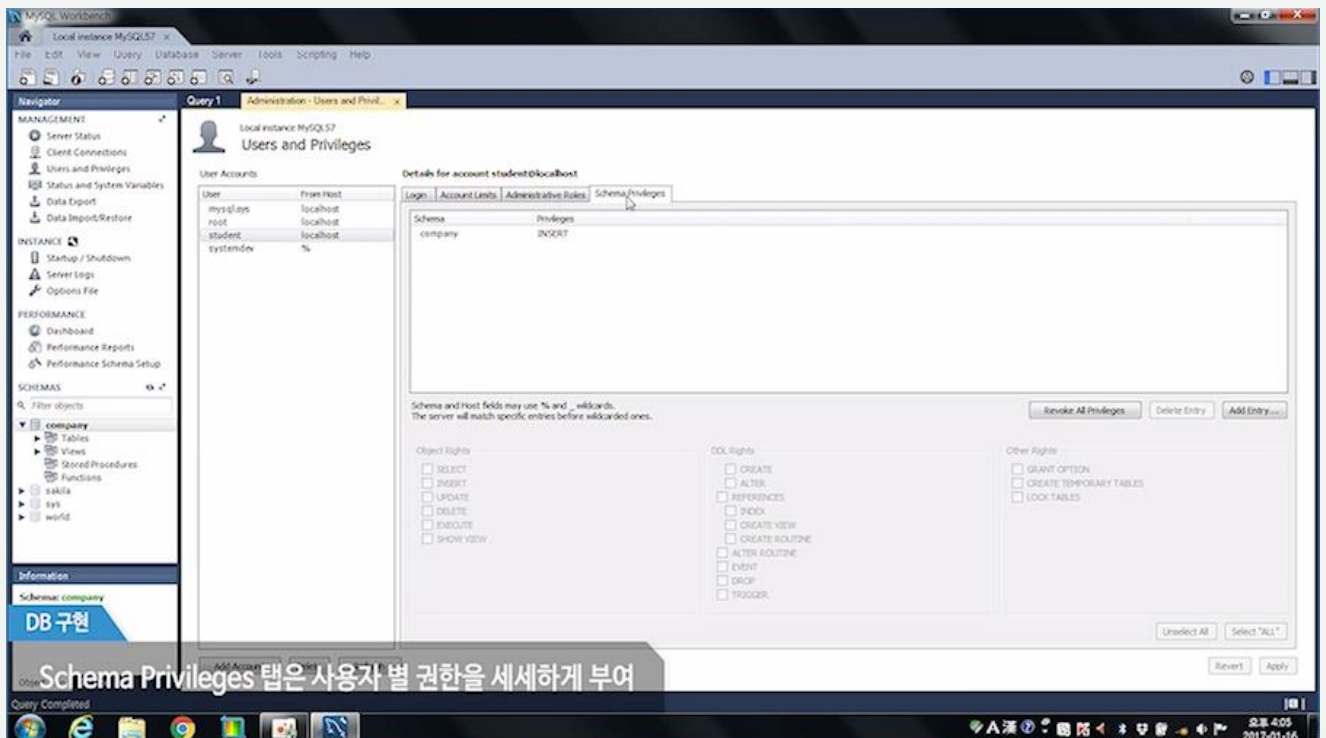


- 시스템을 사용하는 사용자를 그룹화하여 각 그룹별 권한을 지정하고 사용자가 포함되는 그룹에 맞춰 권한을 일괄적으로 부여할 수 있는 기능도 수행할 수 있습니다.

GUI 방식을 이용한 사용자와 권한 관리



Login 탭은 사용자의 추가, 변경 또는 삭제 기능을 담당



Schema Privileges 탭은 사용자 별 권한을 세세하게 부여



핵심요약

권한 시스템

- ❖ MySQL의 권한 시스템은 허가 되지 않은 사용자가 DBMS에 접속하거나 사용자의 무분별한 데이터 접근을 제어하는 기능
 - MySQL 서버로의 접속 권한
 - 데이터베이스에서의 SQL 및 각종 명령의 실행 권한
 - 테이블 접근 권한

| 시스템 테이블 | 기능 |
|------------------|----------------------------------|
| User 테이블 | 호스트, 사용자 ID, 패스워드, 명령어 별 실행권한 관리 |
| DB 테이블 | 스키마 내에서 명령어 별 실행권한 관리 |
| Tables_priv 테이블 | 테이블 별 명령어 실행권한 관리 |
| Columns_priv 테이블 | 컬럼 별 명령어 실행권한 관리 |



핵심요약

DCL 구문을 이용한 사용자와 권한 관리

- ❖ GRANT 문
 - 사용자의 추가 및 권한 부여
 - 특정 데이터베이스의 테이블의 컬럼에 대한 권한까지 상세히 설정할 수 있음
- ❖ REVOKE 문
 - 사용자의 권한을 제거할 때 사용함
- ❖ DELETE 문
 - 사용자를 완전히 삭제하고자 할 때 사용함



핵심요약

GUI 방식을 이용한 사용자와 권한 관리

❖ Users and Privileges

- 사용자 추가 및 제거를 비롯하여 사용자의 권한을 GUI 방식으로 관리할 수 있는 메뉴

| Users and Privileges 메뉴 | 기능 |
|-------------------------|--|
| Login 탭 | 사용자의 추가, 변경 또는 삭제 기능을 담당 |
| Account Limits 탭 | 사용자가 SQL 문을 요청할 수 있는 횟수, 연결할 수 있는 횟수 등을 결정 |
| Administrative Roles 탭 | 역할을 생성하여 일괄적으로 권한을 부여 |
| Schema Privileges 탭 | 사용자 별 권한을 세세하게 부여 |