Real MySQL 8.0

3장 사용자 및 권한

MySQL이 다른 DBMS와 다른 점

- 사용자 계정 생성하는 법
  - 사용자의 아이디
  - **해당 사용자가 어느 IP에서 접속하고 있는지도 확인**
- 각 계정의 권한을 식별하는 법
  - 8.0 버전부터 롤Role(권한을 묶어서 관리하는 역할)의 도입
  - 각 사용자의 권한으로 미리 준비된 Role을 부여하는 게 가능

3.1 사용자 식별

- MySQL 사용자
  - 사용자의 계정
  - 사용자의 접속지점(클라이언트가 실행된 호스트명이나 도메인 또는 IP 주소)

→ 계정을 언급할 때, 아이디와 호스트를 함께 명시해야 함.

```
MySQL 서버가 가동 중인 로컬 호스트에서 svc_id라는 아이디로 접속할 때만 사용될 수 있는 계정
'svc_id'@'127.0.0.1'

모든 외부 컴퓨터에서 접속이 가능한 사용자 계정
'svc_id'@'%'
```

  - 만약, 서로 **동일한** 아이디가 있을 때

```
'svc_id'@'192.168.0.10' (비번은 123)
'svc_id'@'%' (비번은 abc)
```

권한이나 계정 정보에 대해, MySQL은 **범위가 가장 작은 것**을 항상 먼저 선택

그래서 IP 주소가 192.168.0.10인 PC에서 svc_id로 로그인 할 때,
비밀번호를 abc로 로그인하면 비밀번호가 일치하지 않는다는 이유로 접속 불가

3.2 사용자 계정 관리

관리 : 생성, 삭제, 변경

3.2.1 시스템 계정과 일반 계정

	대상	SYSTEM_USER 권한 소유	사용자를 위한 계정
시스템 계정 System Account	DB Server 관리자	O	O
일반 계정 Regular Account	응용 프로그램이나 개발자	X	O

- 일반 계정 : 시스템 계정 관리 불가능
- 시스템 계정 : 시스템 계정과 일반 계정 관리 **가능**,
  데이터베이스 서버 관리 관련 중요 작업

  - 계정 관리(계정 생성 및 삭제, 그리고 계정의 권한 부여 및 제거)
  - 다른 세션(Connection) 또는 그 세션에서 실행 중인 쿼리를 강제 종료
  - 스토어드 프로그램 생성 시 DEFINER를 타 사용자로 설정

- 내장된 계정

  - 'mysql.sys'@'localhost': MySQL 8.0부터 기본으로 내장된 sys 스키마의 객체(뷰나 함수, 그리고 프로시저)들의 DEFINER로 사용되는 계정
  - 'mysql.session'@'localhost': MySQL 플러그인이 서버로 접근할 때 사용되는 계정
  - 'mysql.infoschema'@'localhost': information_schema에 정의된 뷰의 DEFINER로 사용되는 계정

  → 내부적으로 각기 다른 목적을 가지고 있기에, 삭제되지 않도록 주의(애초에 잠겨있기는 함)

3.2.2 계정 생성

	명령
계정 생성	CREATE USER
권한 부여	GRANT

계정 생성할 때 주는 옵션들

3.2.2.1 IDENTIFIED WITH

- 사용자의 인증 방식과 비밀번호 설정
- 뒤에는 반드시 인증 방식(인증 플러그인 이름) 명시
  - 기본 인증 방식 : IDENTIFIED BY ‘password’
  - Native Pluggable Authentication
  - Caching SHA-2 Pluggable Authentication
  - PAM Pluggable Authentication
  - LDAP Pluggable Authentication
- CREATE USER, ALTER USER 명령으로,
  서버의 계정을 생성 또는 변경 시 연결 방식과 비밀번호 옵션, 자원 사용과 관련된 여러 옵션 설정 가능

3.2.2.2 REQUIRE

- MySQL 서버에 접속할 때,
암호화된 SSL/TLS 채널 사용할지 여부 설정 (아니면 비암호화된 채널로 연결하게 됨)

3.2.2.3 PASSWORD EXPIRE

- 비밀번호의 유효 기간을 설정하는 옵션
- 별도로 명시하지 않으면, default_password_lifetime 시스템 변수에 저장된 기간으로 설정됨
- 개발자나 DB 관리자의 비밀번호는 유효기간을 설정하는 것이 보안 상 안전
- 응용 프로그램 접속용 계정에 유효 기간을 설정하는 것은 위험할 수 있으니 주의!
- 설정 가능한 옵션

  - PASSWORD EXPIRE: 계정 생성과 동시에 비밀번호의 만료 처리
  - PASSWORD EXPIRE NEVER: 계정 비밀번호의 만료 기간 없음
  - PASSWORD EXPIRE DEFAULT: default_password_lifetime 시스템 변수에 저장된 기간으로 비밀번호의 유효 기간을 설정
  - PASSWORD EXPIRE INTERVAL n DAY: 비밀번호의 유효 기간을 오늘부터 n일자로 설정

3.2.2.4 PASSWORD HISTORY

- 한 번 사용했던 비밀번호를 재사용하지 못하게 설정하는 옵션
  - 이를 기억하기 위해, MySQL 서버는 mysql DB의 password_history 테이블 사용
- 설정 가능한 옵션

  - PASSWORD HISTORY DEFAULT: password_history 시스템 변수에 저장된 개수만큼 비밀번호의 이력을 저장하며, 저장된 이력에 남아있는 비밀번호는 재사용할 수 없다.
  - PASSWORD HISTORY n: 비밀번호의 이력을 최근 n개까지만 저장하며, 저장된 이력에 남아있는 비밀번호는 재사용할 수 없다.

3.2.2.5 PASSWORD REUSE INTERVAL

- 한 번 사용했던 비밀번호의 재사용 금지 기간을 설정하는 옵션
- 별도 명시하지 않으면, password_reuse_interval 시스템 변수에 저장된 기간으로 설정
- 설정 가능한 옵션

  - PASSWORD REUSE INTERVAL DEFAULT: password_reuse_interval 변수에 저장된 기간으로 설정
  - PASSWORD REUSE INTERVAL n DAY: n일자 이후에 비밀번호를 재사용할 수 있게 설정

3.2.2.6 PASSWORD REQUIRE

- 비밀번호가 만료되어 현재 사용하는 비밀번호를 필요로 할지 말지 결정하는 옵션
- 별도 명시하지 않으면, password_require_current 시스템 변수의 값으로 설정됨
- 설정 가능한 옵션

  - PASSWORD REQUIRE CURRENT: 비밀번호를 변경할 때 현재 비밀번호를 먼저 입력하도록 설정
  - PASSWORD REQUIRE OPTIONAL: 비밀번호를 변경할 때 현재 비밀번호를 입력하지 않아도 되도록 설정
  - PASSWORD REQUIRE DEFAULT: password_require_current 시스템 변수의 값으로 설정

3.2.2.8 ACCOUNT LOCK/UNLOCK

- 계정 생성 시 또는 ALTER USER 명령을 사용해 계정 정보를 변경할 때
  계정을 사용하지 못하게 잠글지 여부를 결정

- ACCOUNT LOCK: 계정을 사용하지 못하게 잠금
- ACCOUNT UNLOCK: 잠긴 계정을 다시 사용 가능 상태로 잠금 해제

3.3 비밀번호 관리

3.3.1 고수준 비밀번호

- 유효기간이나 이력 관리를 통한 재사용 금지 기능뿐만 아니라, 비밀번호를 쉽게 유추할 수 있는 단어들이 사용되지 않게 글자의 조합을 강제하거나 금칙어를 설정하는 기능도 있다.
- 유효성 체크 규칙 적용 : validate_password 컴포넌트 이용
- 비밀번호 정책 : 기본값은 MDEIUM으로 자동 설정

  - LOW: 비밀번호의 길이만 검증
  - MEDIUM: 비밀번호의 길이를 검증하며, 숫자와 대소문자, 그리고 특수문자의 배합을 검증
  - STRONG: MEDIUM 레벨의 검증을 모두 수행하며, 금칙어가 포함됐는지 여부까지 검증

- 사전에 명시되지 않은 단어들로, 즉 금칙어 제외한 단어들로 생성하게 하는 기능
  - validate_password.dictionary_file 시스템 변수에 금칙어들이 저장된 사전 파일을 등록
  - validate_password.policy 변수가 ‘STRONG’으로 설정된 경우에만 작동

3.3.2 이중 비밀번호(Dual Password)

- 계정의 비밀번호로 2개의 값을 동시에 사용 가능!

  (주의) 프라이머리와 세컨더리 2개 중 1개만 일치하면 됨

프라이머리 Primary	최근에 설정 된 비밀번호
세컨더리 Secondary	이전 비밀번호

- 기존 비밀번호 변경 구문에 RETAIN CURRENT PASSWORD 옵션 추가

```
-- // 비밀번호를 "ytrewq"로 설정
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'old_password';

-- // 비밀번호를 "qwerty"로 변경하면서 기존 비밀번호를 세컨더리 비밀번호로 설정
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password' RETAIN CURRENT PASSWORD;
```

- 계정의 보안을 위해, 사용 후 세컨더리 비밀번호는 삭제하는 게 좋음

3.4 권한(Privilege)

정적 권한	글로벌 권한	데이터베이스나 테이블 이외의 객체에 적용되는 권한	GANT 명령에서 특정 객체 명시 X
정적 권한	객체 권한	데이터베이스나 테이블을 제어하는데 필요한 권한	GANT 명령에서 특정 객체 명시 O
동적 권한		SUPER 권한이 쪼개져서 동적 권한으로 분할 됨	

→ 8.0 버전부터 동적 권한이 추가됨

→ 백업 관리자와 복제 관리자 개별로 꼭 필요한 권한만 부여할 수 있게 됨

표 3.1 정적 권한

구분	권한	Grant 테이블의 칼럼명	권한 범위
글로벌 권한	FILE	File_priv	파일
	CREATE ROLE	Create_role_priv	서버 관리
	CREATE TABLESPACE	Create_tablespace_priv	서버 관리
	CREATE USER	Create_user_priv	서버 관리
	DROP ROLE	Drop_role_priv	서버 관리
	PROCESS	Process_priv	서버 관리
글로벌 권한	PROXY	See proxies_priv table	서버 관리
	RELOAD	Reload_priv	서버 관리
	REPLICATION CLIENT	Repl_client_priv	서버 관리
	REPLICATION SLAVE	Repl_slave_priv	서버 관리
	SHOW DATABASES	Show_db_priv	서버 관리
	SHUTDOWN	Shutdown_priv	서버 관리
	SUPER	Super_priv	서버 관리
	USAGE	Synonym for "no privileges"	서버 관리

객체 권한	EVENT	Event_priv	데이터베이스
	LOCK TABLES	Lock_tables_priv	데이터베이스
	REFERENCES	References_priv	데이터베이스 & 테이블
	CREATE	Create_priv	데이터베이스 & 테이블 & 인덱스
	GRANT OPTION	Grant_priv	데이터베이스 & 테이블 & 스토어드 프로그램
	DROP	Drop_priv	데이터베이스 & 테이블, 뷰
	ALTER ROUTINE	Alter_routine_priv	스토어드 프로그램
	CREATE ROUTINE	Create_routine_priv	스토어드 프로그램
	EXECUTE	Execute_priv	스토어드 프로그램
	ALTER	Alter_priv	테이블
	CREATE TEMPORARY TABLES	Create_tmp_table_priv	테이블
	DELETE	Delete_priv	테이블
	INDEX	Index_priv	테이블
	TRIGGER	Trigger_priv	테이블
	INSERT	Insert_priv	테이블 & 칼럼
	SELECT	Select_priv	테이블 & 칼럼
	UPDATE	Update_priv	테이블 & 칼럼
	CREATE VIEW	Create_view_priv	뷰
	SHOW VIEW	Show_view_priv	뷰
객체 & 글로벌	ALL [PRIVILEGES]	Synonym for "all privileges"	서버 관리

표 3.2 동적 권한

권한	권한 범위
INNODB_REDO_LOG_ARCHIVE	리두 로그 관리
RESOURCE_GROUP_ADMIN	리소스 관리
RESOURCE_GROUP_USER	리소스 관리
BINLOG_ADMIN	백업 & 복제 관리
BINLOG_ENCRYPTION_ADMIN	백업 & 복제 관리
BACKUP_ADMIN	백업 관리
CLONE_ADMIN	백업 관리
GROUP_REPLICATION_ADMIN	복제 관리
REPLICATION_APPLIER	복제 관리
REPLICATION_SLAVE_ADMIN	복제 관리
CONNECTION_ADMIN	서버 관리
ENCRYPTION_KEY_ADMIN	서버 관리
PERSIST_RO_VARIABLES_ADMIN	서버 관리
ROLE_ADMIN	서버 관리
SESSION_VARIABLES_ADMIN	서버 관리
SET_USER_ID	서버 관리
SHOW_ROUTINE	서버 관리
SYSTEM_USER	서버 관리
SYSTEM_VARIABLES_ADMIN	서버 관리
TABLE_ENCRYPTION_ADMIN	서버 관리
VERSION_TOKEN_ADMIN	서버 관리
XA_RECOVER_ADMIN	서버 관리
APPLICATION_PASSWORD_ADMIN	이중 비밀번호 관리
AUDIT_ADMIN	Audit 로그 관리

- 권한 부여(GRANT 명령어)
  - 반드시 먼저 사용자를 생성하고, GRANT 명령으로 권한을 부여해야 함

```
GRANT privilege_list ON db.table TO 'user'@'host';
```

  - TO + 권한을 부여할 대상 사용자
  - ON + 어떤 오브젝트에 권한을 부여할지 결정
  - 권한의 범위에 따라 사용하는 방법이 달라짐

→ 글로벌 권한 : 특정 DB나 테이블에 부여될 수 없음

```
GRANT SUPER ON *.* TO 'user'@'localhost';
```

→ DB 권한 : 서버의 특정 DB나 모든 DB에 적용할 수 있음

```
GRANT EVENT ON *.* TO 'user'@'localhost';
GRANT EVENT ON employees.* TO 'user'@'localhost';
```

→ 테이블 권한 : 서버의 모든 DB에, 그리고 특정 DB의 오브젝트에 대해서만 권한 부여 가능

```
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'user'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE ON employees.* TO 'user'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE ON employees.department TO 'user'@'localhost';
```

→ 테이블의 특정 칼럼에 대해서만 권한 부여

```
GRANT SELECT, INSERT, UPDATE(dept_name) ON employees.department TO 'user'@'localhost';
```

- 전체적인 성능에 영향을 끼칠 수 있으므로, 테이블이나 칼럼 단위의 권한은 잘 사용하지 않음
  - 테이블에서 권한 허용하려는 칼럼만으로 별도의 VIEW 만들어서 사용
- 권한 관련 테이블

구분	저장소 테이블	설명
정적 권한	mysql.user	계정 정보 & 계정이나 역할에 부여된 글로벌 권한
	mysql.db	계정이나 역할에 DB 단위로 부여된 권한
	mysql.tables_priv	계정이나 역할에 테이블 단위로 부여된 권한
	mysql.columns_priv	계정이나 역할에 칼럼 단위로 부여된 권한
	mysql.procs_priv	계정이나 역할에 스토어드 프로그램 단위로 부여된 권한
동적 권한	mysql.global_grants	계정이나 역할에 부여되는 동적 글로벌 권한

3.5 역할(Role)

- MySQL 서버 내부적으로 계정과 역할(Role)은 똑같은 객체라 차이가 없음.

단지 하나의 사용자 계정에 다른 사용자 계정이 가진 권한을 병합해서 권한 제어가 가능해졌을 뿐임.

- CREATE USER 명령으로 계정 생성할 때는, 계정 이름과 호스트 부분을 함께 명시
  CREATE ROLE 명령으로 역할 생성할 때는, 호스트 부분을 별도 명시하지 않음

  → 계정과 역할의 차이 아니야??
  아니다. 호스트 부분을 명시하지 않은 경우에는 자동으로 '모든 호스트(%)'가 추가됨
  그래서 따로 명시하지 않았지만 뒤에 모든 호스트가 추가되어 있는 것

  → 같다는데 왜 명령어를 다르게 지정??

  데이터베이스의 관리의 직무를 분리할 수 있게 해서 보안을 강화하는 용도

- 역할의 호스트 부분은, 역할을 다른 계정에 부여하지 않고
  직접 로그인하는 용도로 쓸 때(실제 계정처럼 사용할 때) 그때는 호스트 부분이 중요함
- 권한 관련 테이블

저장소 테이블	설명
mysql.default_roles	계정별 기본 역할
mysql.role_edges	역할에 부여된 역할 관계 그래프