WEEK 8

DATABASE SECURITY – PART 1

STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
  - Create a user in mysql and assign the user a password
  - Using the mysql database, list the users
  - Grant the user some privileges
  - List at least 5 privileges in MySQL
  - Revoke privileges from a user
  - Using the system catalog, list all the privileges

DATABASE SECURITY

- **Legal/Ethical Issues** (right to privacy of information, …)
- **Policy Issues** (what does your company make available; who within the company should be updating what)
- **System Issues** (where and how should security be handled? Physical hardware level, operating system level, DBMS level) →we will look at this one!

- 2 Types of Database Security Mechanisms:
  - **Discretionary:** grant privileges to users on tables, records, fields, etc. based on the discretion of the owner of the table (what you're used to in Unix)
  - **Mandatory:** Multiple security levels, categorize the data and the users on the levels; decisions of who gets to see what are based only on relative levels/security labels (example: Top Secret, Secret, Confidential, Unclassified)

DATABASE SECURITY AND THE DBA

- DBA has a system account (like a superuser). The DBA account performs the following actions
  - Account Creation (sets account name, password)
  - Privilege Granting
  - Privilege Revocation
  - Security Level Assignments (set user account to appropriate security classification)
- User must log on to DBMS using an account name and password
- All actions by the user are monitored in a log file
- If anything suspicious occurs a database audit is performed to check actions and accounts

DISCRETIONARY ACCESS CONTROL

- Grant and revoke privileges from users, 2 levels
  - **Account Level:** giving the user access to the database in general
    - CREATE SCHEMA
    - CREATE TABLE
    - CREATE VIEW
    - ALTER, DELETE, MODIFY, SELECT

- **Relation Level:** giving the user access to each relation in the database
  - Each relation R is assigned an owner account (usually the account that created the relation in the first place). The owner account is given all privileges on that relation. This owner can pass on privileges to other users by granting privileges to their accounts
  - SELECT privilege on R → Account can only retrieve from R
  - MODIFY privilege on R (divided into UPDATE, DELETE and INSERT privileges). With the Insert and Update you can restrict to just certain attributes
  - References privilege on R → Used for creating integrity constraints

Table 13.6 Permissible Static Privileges for GRANT and REVOKE

Privilege	Meaning and Grantable Levels
ALL [PRIVILEGES]	Grant all privileges at specified access level except GRANT OPTION and PROXY.
ALTER	Enable use of ALTER TABLE. Levels: Global, database, table.
ALTER ROUTINE	Enable stored routines to be altered or dropped. Levels: Global, database, routine.
CREATE	Enable database and table creation. Levels: Global, database, table.
CREATE ROUTINE	Enable stored routine creation. Levels: Global, database.
CREATE TABLESPACE	Enable tablespaces and log file groups to be created, altered, or dropped. Level: Global.
CREATE TEMPORARY TABLES	Enable use of CREATE TEMPORARY TABLE. Levels: Global, database.
CREATE USER	Enable use of CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES. Level: Global.
CREATE VIEW	Enable views to be created or altered. Levels: Global, database, table.
DELETE	Enable use of DELETE. Level: Global, database, table.
DROP	Enable databases, tables, and views to be dropped. Levels: Global, database, table.
EVENT	Enable use of events for the Event Scheduler. Levels: Global, database.
EXECUTE	Enable the user to execute stored routines. Levels: Global, database, routine.
FILE	Enable the user to cause the server to read or write files. Level: Global.
GRANT OPTION	Enable privileges to be granted to or removed from other accounts. Levels: Global, database, table, routine, proxy.
INDEX	Enable indexes to be created or dropped. Levels: Global, database, table.
INSERT	Enable use of INSERT. Levels: Global, database, table, column.
LOCK TABLES	Enable use of LOCK TABLES on tables for which you have the SELECT privilege. Levels: Global, database.
PROCESS	Enable the user to see all processes with SHOW PROCESSLIST. Level: Global.
PROXY	Enable user proxying. Level: From user to user.
REFERENCES	Enable foreign key creation. Levels: Global, database, table, column.
RELOAD	Enable use of FLUSH operations. Level: Global.
REPLICATION CLIENT	Enable the user to ask where master or slave servers are. Level: Global.
REPLICATION SLAVE	Enable replication slaves to read binary log events from the master. Level: Global.
SELECT	Enable use of SELECT. Levels: Global, database, table, column.
SHOW DATABASES	Enable SHOW DATABASES to show all databases. Level: Global.
SHOW VIEW	Enable use of SHOW CREATE VIEW. Levels: Global, database, table.
SHUTDOWN	Enable use of **mysqladmin shutdown**. Level: Global.
SUPER	Enable use of other administrative operations such as CHANGE MASTER TO, KILL, PURGE BINARY LOGS, SET GLOBAL, and **mysqladmin debug** command. Level: Global.
TRIGGER	Enable trigger operations. Levels: Global, database, table.
UPDATE	Enable use of UPDATE. Levels: Global, database, table, column.
USAGE	Synonym for "no privileges"

CREATING USERS

OR (for a user with no password) → WEAKEST

```
CREATE USER 'jeffrey'@'localhost';
```

OR → WEAK

```
CREATE USER 'lreid'@'localhost' identified by 'mypass';
```

OR → STRONGEST

```
CREATE USER 'jeffrey'@'localhost' IDENTIFIED BY PASSWORD
  '*90E462C37378CED12064BB3388827D2BA3A9B689';
```

(This one is because if you do the first one, the actual password will go into the log file and it can be viewed)

To remove a user that you have created use the DROP command:

```
DROP USER 'jeffery'@'localhost';
```

DB PRIVILEGES SQL COMMANDS

- **REVOKE** → removes privileges
- **GRANT** → adds privileges
- **WITH GRANT OPTION** → allows one user to give another user the ability to give other users privileges

QUESTION: CAN YOU FIGURE OUT WHAT THESE COMMANDS MIGHT DO?

GRANT ALL ON mydb.* TO 'someuser'@'somehost';

GRANT SELECT, INSERT ON mydb.* TO 'someuser'@'somehost';

GRANT SELECT, INSERT ON mydb.mytbl TO 'someuser'@'somehost';

GRANT SELECT (col1), INSERT (col1,col2) ON mydb.mytbl TO 'someuser'@'somehost';

- We did something like this in assignment 2 already:

```
DROP USER 'ta'@'localhost';
CREATE USER 'ta'@'localhost' IDENTIFIED BY ‘somepassword';
GRANT ALL PRIVILEGES ON yourwesternuseridassign2db.* TO 'ta'@'localhost';
FLUSH PRIVILEGES;
```

EXAMPLES:

- Assume that:
  - *EMPLOYEE*, *PROJECT* are my tables
  - *lreid*, *sskinner*, *nflanders*, *hsimpson*, *mburns* are users
  - *BIRTHDATE*, *SALARY* and *NAME* are some of the columns from the table *EMPLOYEE*

- Suppose that *nflanders* does the following:

GRANT INSERT,DELETE ON employee, project TO lreid;

- Suppose then that *nflanders* does this command:

GRANT SELECT ON employee, project TO hsimpson WITH GRANT OPTION;

- Suppose then that *hsimpson* does this command

GRANT SELECT ON employee TO mburns;

QUESTION: What do you think happens when this command is executed:

REVOKE SELECT on EMPLOYEE from hsimpson;

- Suppose we only want ***sskinner*** to see the birthdate and name of employees but not salaries or other info for the Payroll department (then we would do the following:)

CREATE VIEW vlimitemp as SELECT name, bdate FROM employee WHERE DeptNo = '5';

QUESTION: Then what would we do?

ANSWER:
GRANT SELECT ON vlimitemp TO sskinner;

QUESTION: What do you think this statement does:

GRANT UPDATE ON Employee(Salary) TO bgumbel;

ANSWER:
bgumbel can only update the salaries but no other fields!

- Can also create VIEWS → if you want a user to only see column A from table AAA and column B from table BBB you can create a view and give the user access to the view

EXAMPLE FROM THE SYSTEM TABLES:

mysql> select * from USER_PRIVILEGES;

```
+-------------------+---------------+----------------+--------------+
| GRANTEE           | TABLE_CATALOG | PRIVILEGE_TYPE | IS_GRANTABLE |
+-------------------+---------------+----------------+--------------+
| 'root'@'localhost`| def           | SELECT         | YES          |
| 'root'@'localhost`| def           | INSERT         | YES          |
| 'root'@'localhost`| def           | UPDATE         | YES          |
| 'root'@'localhost'| def           | DELETE         | YES          |
| 'root'@'localhost'| def           | CREATE         | YES          |
```

…

mysql> select * from TABLE_PRIVILEGES;

```
+----------------+---------------+--------------+------------+----------------+--------------+
| GRANTEE        | TABLE_CATALOG | TABLE_SCHEMA | TABLE_NAME | PRIVILEGE_TYPE | IS_GRANTABLE |
+----------------+---------------+--------------+------------+----------------+--------------+
| ''@'localhost' | def           | vetdb        | owner      | SELECT         | NO           |
| ''@'localhost' | def           | vetdb        | owner      | INSERT         | NO           |
| ''@'localhost' | def           | vetdb        | owner      | UPDATE         | NO           |
| ''@'localhost' | def           | vetdb        | owner      | DELETE         | NO           |
| ''@'localhost' | def           | vetdb        | owner      | CREATE         | NO           |
| ''@'localhost' | def           | vetdb        | owner      | DROP           | NO           |
| ''@'localhost' | def           | vetdb        | owner      | REFERENCES     | NO           |
```