



CSE 311L(Database Management System)

LAB-Week 08 (Part A)

Controlling User Access

Topics:

- ▶ Creating Users
- ▶ Granting System Privileges
- ▶ What is a role?
- ▶ Creating and Granting Privileges to a Role
- ▶ Changing Password
- ▶ Granting Object Privileges
- ▶ Using WITH GRANT OPTION and PUBLIC key

Creating Users

```
CREATE USER user_name  
IDENTIFIED BY password;
```

- **local access** -> **username@localhost**
- **remote access** -> **username**
- **particular domain** -> **username@'%.domain_name.com'**

Granting Privileges

```
GRANT ALL  
ON *.* _name  
TO user_name;
```

```
GRANT create session, create table, create view  
ON database_name.table_name  
TO user_name;
```

View Grants

```
SHOW GRANTS FOR user_name;
```

Showing list of users

```
SELECT * FROM mysql.user;
```

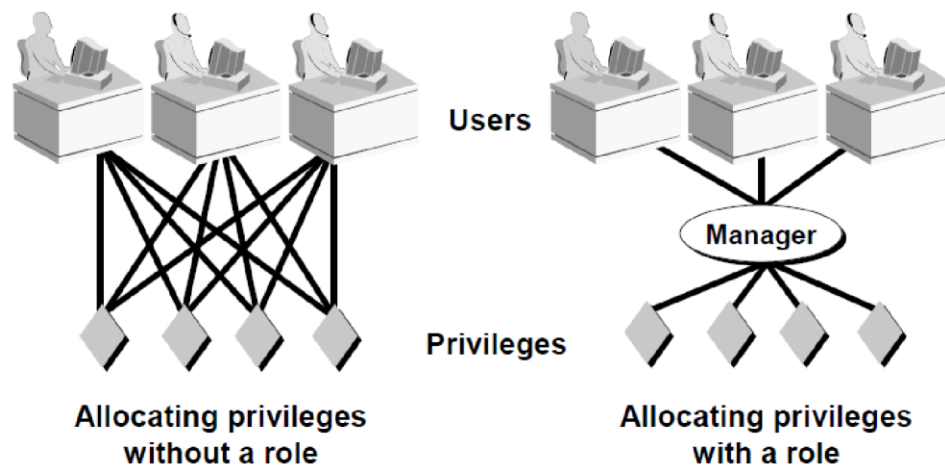
Revoke Privileges

```
REVOKE ALL  
ON database_name.table_name  
FROM user_name;
```

Rename User

```
RENAME user user_name  
TO user_name_new;
```

What is a role?



Creating and Granting Privileges to a Role

```
CREATE ROLE manager;  
  
GRANT create table, create view  
TO manager;  
  
GRANT manager  
TO user_name1, user_name2;
```

Changing Password

```
ALTER USER user_name  
IDENTIFIED BY dhaka;
```

Granting Object Privileges

```
GRANT  select
ON     employees
TO     sue, rich;
```

- Grant privileges to update specific columns to users and roles.

```
GRANT  update (department_name, location_id)
ON     departments
TO     scott, manager;
Grant succeeded.
```

Using WITH GRANT OPTION and PUBLIC key

- Give a user authority to pass along privileges.

```
GRANT  select, insert
ON     departments
TO     scott
WITH   GRANT OPTION;
Grant succeeded.
```

- Allow all users on the system to query data from Alice's DEPARTMENTS table.

```
GRANT  select
ON     alice.departments
TO     PUBLIC;
Grant succeeded.
```



CSE 311L(Database Management System)

LAB-Week 08 (Part B)

Triggers/Stored Procedure Examples & Demo

Trigger Syntax

```
CREATE TRIGGER trigger_name
Trigger_time trigger_event ON table_name
FOR EACH ROW
trigger_body;
```

Trigger Event Keywords

Trigger Event	OLD	NEW
INSERT	NO	YES
UPDATE	YES	YES
DELETE	YES	NO

Trigger Example

```
CREATE TRIGGER employee_update
BEFORE INSERT ON employees
FOR EACH ROW
SET NEW.First_name = UPPER(NEW.First_name)
```

Stored Procedure Example

```
CREATE PROCEDURE Employee()
BEGIN
    SELECT
        First_name,
        Last_name,
        Job_ID,
        Employee_ID
    FROM
        employees
    ORDER BY Employee_ID;
END;
```

Calling a procedure

```
CALL Employee();
```