UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SECD2523 DATABASE

SQL 5b | TCL

Innovating Solutions

LECTURE LEARNING OUTCOME

By the end of this lecture, students should be able to:

Construct SQL statements for Discretionary Access Control:

01 **Granting Privileges to Other Users (GRANT)**

02 **Revoking Privileges from Users (REVOKE)**

01 Granting Privileges to Other Users (GRANT)

02 Revoking Privileges from Users (REVOKE)

Before we continue this TCL,

we reuse the same database same tables created in DML3

mysql: **use db_dml3_dept_emp_loc;**

If failed to find the database and tables you created previously,

refer here to get our SQL statements to prepare all tables and sample data…
https://docs.google.com/document/d/1jC5daHE4WhP7N9U6ZxqRzBJcJAzqJxf1/edit?usp=sharing&ouid=112935562328060013817&rtpof=true&sd=true

Here, our tables with records

select * from locations;

```
mysql> select * from locations;
+-------------+----------------------------------------+-------------+--------------------+-------------------+------------+
| location_id | street_address                         | postal_code | city               | state_province    | country_id |
+-------------+----------------------------------------+-------------+--------------------+-------------------+------------+
|        1000 | 1297 Via Cola di Rie                   | 00989       | Roma               | NULL              | IT         |
|        1100 | 93091 Calle della Testa                | 10934       | Venice             | NULL              | IT         |
|        1200 | 2017 Shinjuku-ku                       | 1689        | Tokyo              | Tokyo Prefecture  | JP         |
|        1300 | 9450 Kamiya-cho                        | 6823        | Hiroshima          | NULL              | JP         |
|        1400 | 2014 Jabberwocky Rd                    | 26192       | Southlake          | Texas             | US         |
|        1500 | 2011 Interiors Blvd                    | 99236       | South San Francisco | California        | US         |
|        1600 | 2007 Zagora St                         | 50090       | South Brunswick    | New Jersey        | US         |
|        1700 | 2004 Charade Rd                        | 98199       | Seattle            | Washington        | US         |
|        1800 | 147 Spadina Ave                        | M5V 2L7     | Toronto            | Ontario           | CA         |
|        1900 | 6092 Boxwood St                        | YSW 9T2     | Whitehorse         | Yukon             | CA         |
|        2000 | 40-5-12 Laogianggen                    | 190518      | Beijing            | NULL              | CN         |
|        2100 | 1298 Vileparle (E)                     | 490231      | Bombay             | Maharashtra       | IN         |
|        2200 | 12-98 Victoria Street                  | 2901        | Sydney             | New South Wales   | AU         |
|        2300 | 198 Clementi North                     | 540198      | Singapore          | NULL              | SG         |
|        2400 | 8204 Arthur St                         | NULL        | London             | NULL              | UK         |
|        2500 | Magdalen Centre, The Oxford Science Park | OX9 9ZB     | Oxford             | Oxford            | UK         |
|        2600 | 9702 Chester Road                      | 09629850293 | Stretford          | Manchester        | UK         |
|        2700 | Schwanthalerstr. 7031                  | 80925       | Munich             | Bavaria           | DE         |
|        2800 | Rua Frei Caneca 1360                   | 01307-002   | Sao Paulo          | Sao Paulo         | BR         |
|        2900 | 20 Rue des Corps-Saints                | 1730        | Geneva             | Geneve            | CH         |
|        3000 | Murtenstrasse 921                      | 3095        | Bern               | BE                | CH         |
|        3100 | Pieter Breughelstraat 837              | 3029SK      | Utrecht            | Utrecht           | NL         |
|        3200 | Mariano Escobedo 9991                  | 11932       | Mexico City        | Distrito Federal, | MX         |
+-------------+----------------------------------------+-------------+--------------------+-------------------+------------+
23 rows in set (0.00 sec)
```

Here, our tables with records

select * from departments;

```
mysql> select * from departments;
+---------------+----------------------+------------+-------------+
| department_id | department_name      | manager_id | location_id |
+---------------+----------------------+------------+-------------+
|            10 | Administration       |        200 |        1700 |
|            20 | Marketing            |        201 |        1800 |
|            30 | Purchasing           |        114 |        1700 |
|            40 | Human Resources      |        203 |        2400 |
|            50 | Shipping             |        121 |        1500 |
|            60 | IT                   |        103 |        1400 |
|            70 | Public Relations     |        204 |        2700 |
|            80 | Sales                |        145 |        2500 |
|            90 | Executive            |        100 |        1700 |
|           100 | Finance              |        108 |        1700 |
|           110 | Accounting           |        205 |        1700 |
|           120 | Treasury             |       NULL |        1700 |
|           130 | Corporate Tax        |       NULL |        1700 |
|           140 | Control And Credit   |       NULL |        1700 |
|           150 | Shareholder Services |       NULL |        1700 |
|           160 | Benefits             |       NULL |        1700 |
|           170 | Manufacturing        |       NULL |        1700 |
|           180 | Construction         |       NULL |        1700 |
|           190 | Contracting          |       NULL |        1700 |
|           200 | Operations           |       NULL |        1700 |
|           210 | IT Support           |       NULL |        1700 |
|           220 | NOC                  |       NULL |        1700 |
|           230 | IT Helpdesk          |       NULL |        1700 |
|           240 | Government Sales     |       NULL |        1700 |
|           250 | Retail Sales         |       NULL |        1700 |
|           260 | Recruiting           |       NULL |        1700 |
|           270 | Payroll              |       NULL |        1700 |
+---------------+----------------------+------------+-------------+
27 rows in set (0.00 sec)
```

Here,
our tables with records

select * from employees;

```
mysql> select * from employees;
+-------------+------------+-------------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name   | email    | phone_number       | hire_date  | job_id     | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-------------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
|         100 | Steven     | King        | SKING    | 515.123.4567       | 1987-06-17 | AD_PRES    | 24000.00 |           NULL |       NULL |            90 |
|         101 | Neena      | Kochhar     | NKOCHHAR | 515.123.4568       | 1989-09-21 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         102 | Lex        | De Haan     | LDEHAAN  | 515.123.4569       | 1993-01-13 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         103 | Alexander  | Hunold      | AHUNOLD  | 590.423.4567       | 1990-01-03 | IT_PROG    |  9000.00 |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst       | BERNST   | 590.423.4568       | 1991-05-21 | IT_PROG    |  6000.00 |           NULL |        103 |            60 |
|         105 | David      | Austin      | DAUSTIN  | 590.423.4569       | 1997-06-25 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         106 | Valli      | Pataballa   | VPATABAL | 590.423.4560       | 1998-02-05 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         107 | Diana      | Lorentz     | DLORENTZ | 590.423.5567       | 1999-02-07 | IT_PROG    |  4200.00 |           NULL |        103 |            60 |
|         108 | Nancy      | Greenberg   | NGREENBE | 515.124.4569       | 1994-08-17 | FI_MGR     | 12000.00 |           NULL |        101 |           100 |
|         109 | Daniel     | Faviet      | DFAVIET  | 515.124.4169       | 1994-08-16 | FI_ACCOUNT |  9000.00 |           NULL |        108 |           100 |
|         110 | John       | Chen        | JCHEN    | 515.124.4269       | 1997-09-28 | FI_ACCOUNT |  8200.00 |           NULL |        108 |           100 |
|         111 | Ismael     | Sciarra     | ISCIARRA | 515.124.4369       | 1997-09-30 | FI_ACCOUNT |  7700.00 |           NULL |        108 |           100 |
|         112 | Jose Manuel| Urman       | JMURMAN  | 515.124.4469       | 1998-03-07 | FI_ACCOUNT |  7800.00 |           NULL |        108 |           100 |
|         113 | Luis       | Popp        | LPOPP    | 515.124.4567       | 1999-12-07 | FI_ACCOUNT |  6900.00 |           NULL |        108 |           100 |
|         114 | Den        | Raphaely    | DRAPHEAL | 515.127.4561       | 1994-12-07 | PU_MAN     | 11000.00 |           NULL |        100 |            30 |
|         115 | Alexander  | Khoo        | AKHOO    | 515.127.4562       | 1995-05-18 | PU_CLERK   |  3100.00 |           NULL |        114 |            30 |
|         116 | Shelli     | Baida       | SBAIDA   | 515.127.4563       | 1997-12-24 | PU_CLERK   |  2900.00 |           NULL |        114 |            30 |
|         117 | Sigal      | Tobias      | STOBIAS  | 515.127.4564       | 1997-07-24 | PU_CLERK   |  2800.00 |           NULL |        114 |            30 |
|         118 | Guy        | Himuro      | GHIMURO  | 515.127.4565       | 1998-11-15 | PU_CLERK   |  2600.00 |           NULL |        114 |            30 |
|         119 | Karen      | Colmenares  | KCOLMENA | 515.127.4566       | 1999-08-10 | PU_CLERK   |  2500.00 |           NULL |        114 |            30 |
|         120 | Matthew    | Weiss       | MWEISS   | 650.123.1234       | 1996-07-18 | ST_MAN     |  8000.00 |           NULL |        100 |            50 |
|         121 | Adam       | Fripp       | AFRIPP   | 650.123.2234       | 1997-04-10 | ST_MAN     |  8200.00 |           NULL |        100 |            50 |
|         122 | Payam      | Kaufling    | PKAUFLIN | 650.123.3234       | 1995-05-01 | ST_MAN     |  7900.00 |           NULL |        100 |            50 |
|         123 | Shanta     | Vollman     | SVOLLMAN | 650.123.4234       | 1997-10-10 | ST_MAN     |  6500.00 |           NULL |        100 |            50 |
|         124 | Kevin      | Mourgos     | KMOURGOS | 650.123.5234       | 1999-11-16 | ST_MAN     |  5800.00 |           NULL |        100 |            50 |
|         180 | Winston    | Taylor      | WTAYLOR  | 650.507.9876       | 1998-01-24 | SH_CLERK   |  3200.00 |           NULL |        120 |            50 |
|         181 | Jean       | Fleaur      | JFLEAUR  | 650.507.9877       | 1998-02-23 | SH_CLERK   |  3100.00 |           NULL |        120 |            50 |
|         182 | Martha     | Sullivan    | MSULLIVA | 650.507.9878       | 1999-06-21 | SH_CLERK   |  2500.00 |           NULL |        120 |            50 |
|         183 | Girard     | Geoni       | GGEONI   | 650.507.9879       | 2000-02-03 | SH_CLERK   |  2800.00 |           NULL |        120 |            50 |
|         184 | Nandita    | Sarchand    | NSARCHAN | 650.509.1876       | 1996-01-27 | SH_CLERK   |  4200.00 |           NULL |        121 |            50 |
|         185 | Alexis     | Bull        | ABULL    | 650.509.2876       | 1997-02-20 | SH_CLERK   |  4100.00 |           NULL |        121 |            50 |
|         186 | Julia      | Dellinger   | JDELLING | 650.509.3876       | 1998-06-24 | SH_CLERK   |  3400.00 |           NULL |        121 |            50 |
|         187 | Anthony    | Cabrio      | ACABRIO  | 650.509.4876       | 1999-02-07 | SH_CLERK   |  3000.00 |           NULL |        121 |            50 |
|         188 | Kelly      | Chung       | KCHUNG   | 650.505.1876       | 1997-06-14 | SH_CLERK   |  3800.00 |           NULL |        122 |            50 |
|         189 | Jennifer   | Dilly       | JDILLY   | 650.505.2876       | 1997-08-13 | SH_CLERK   |  3600.00 |           NULL |        122 |            50 |
|         190 | Timothy    | Gates       | TGATES   | 650.505.3876       | 1998-07-11 | SH_CLERK   |  2900.00 |           NULL |        122 |            50 |
|         191 | Randall    | Perkins     | RPERKINS | 650.505.4876       | 1999-12-19 | SH_CLERK   |  2500.00 |           NULL |        122 |            50 |
|         192 | Sarah      | Bell        | SBELL    | 650.501.1876       | 1996-02-04 | SH_CLERK   |  4000.00 |           NULL |        123 |            50 |
|         193 | Britney    | Everett     | BEVERETT | 650.501.2876       | 1997-03-03 | SH_CLERK   |  3900.00 |           NULL |        123 |            50 |
|         194 | Samuel     | McCain      | SMCCAIN  | 650.501.3876       | 1998-07-01 | SH_CLERK   |  3200.00 |           NULL |        123 |            50 |
|         195 | Vance      | Jones       | VJONES   | 650.501.4876       | 1999-03-17 | SH_CLERK   |  2800.00 |           NULL |        123 |            50 |
|         196 | Alana      | Walsh       | AWALSH   | 650.507.9811       | 1998-04-24 | SH_CLERK   |  3100.00 |           NULL |        124 |            50 |
|         197 | Kevin      | Feeney      | KFEENEY  | 650.507.9822       | 1998-05-23 | SH_CLERK   |  3000.00 |           NULL |        124 |            50 |
|         198 | Donald     | OConnell    | DOCONNEL | 650.507.9833       | 1999-06-21 | SH_CLERK   |  2600.00 |           NULL |        124 |            50 |
|         199 | Douglas    | Grant       | DGRANT   | 650.507.9844       | 2000-01-13 | SH_CLERK   |  2600.00 |           NULL |        124 |            50 |
|         200 | Jennifer   | Whalen      | JWHALEN  | 515.123.4444       | 1987-09-17 | AD_ASST    |  4400.00 |           NULL |        101 |            10 |
|         201 | Michael    | Hartstein   | MHARTSTE | 515.123.5555       | 1996-02-17 | MK_MAN     | 13000.00 |           NULL |        100 |            20 |
|         202 | Pat        | Fay         | PFAY     | 603.123.6666       | 1997-08-17 | MK_REP     |  6000.00 |           NULL |        201 |            20 |
|         203 | Susan      | Mavris      | SMAVRIS  | 515.123.7777       | 1994-06-07 | HR_REP     |  6500.00 |           NULL |        101 |            40 |
|         204 | Hermann    | Baer        | HBAER    | 515.123.8888       | 1994-06-07 | PR_REP     | 10000.00 |           NULL |        101 |            70 |
|         205 | Shelley    | Higgins     | SHIGGINS | 515.123.8080       | 1994-06-07 | AC_MGR     | 12000.00 |           NULL |        101 |           110 |
|         206 | William    | Gietz       | WGIETZ   | 51hr5.123.8181     | 1994-06-07 | AC_ACCOUNT |  8300.00 |           NULL |        205 |           110 |
+-------------+------------+-------------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
107 rows in set (0.00 sec)
```

Let’s start…

but, create the users first for manager, personnel, and director, use the following mysql query first:

```
CREATE USER 'manager' IDENTIFIED BY 'password';

CREATE USER 'personnel' IDENTIFIED BY 'password';

CREATE USER 'director' IDENTIFIED BY 'password';
```

```
mysql> CREATE USER 'manager' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE USER 'personnel' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE USER 'director' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.01 sec)
```

display the user list:

```
SELECT user, host FROM mysql.user;
```

```
mysql> SELECT user, host FROM mysql.user;
+------------------+-----------+
| user             | host      |
+------------------+-----------+
| director         | %         |
| manager          | %         |
| personnel        | %         |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
+------------------+-----------+
7 rows in set (0.00 sec)
```

Discretionary Access Control

Discretionary Access Control

- Each user is assigned specific permissions (called privileges) to access or use certain parts of a database.

  Privileges defined by the ISO standard are:

  - SELECT—the privilege to retrieve data from a table;
  - INSERT [(columnName)]—the privilege to insert new rows into a table;
  - UPDATE [(columnName)]—the privilege to modify rows of data in a table;
  - DELETE—the privilege to delete rows of data from a table;
  - REFERENCES [(columnName)]—the privilege to reference columns of a named table in integrity constraints (related to foreign keys);
  - USAGE—the privilege to use domains, collations, character sets, and translations.

Reference: https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html

Discretionary Access Control

- When a user creates something in the database, like a table, they automatically get permissions for it and can share it with others.

  Example:

  - When User A creates a table, they automatically have full permission to manage and use it.
  - Since User A created the table, they automatically receive permissions like SELECT, INSERT, UPDATE, DELETE, and more.
  - User A can share these permissions with User B by granting them specific access rights.
  - This gives User B permission to view the data in the table but not to modify it.
  - Now, User B can run queries to access the data.

Discretionary Access Control

- The system lets users share permissions, but this flexibility can be exploited by someone who tricks a user into sharing confidential information.

  Example:

  - Sensitive data exists in the table, like private salary details.
  - An unauthorized User C (someone who isn't supposed to access certain data) asks an authorized User B (someone with legitimate access) for "help" with a database task.
  - User C says, "Can you run this query and share the results with me? I can't access the data myself."
  - User B, trusting User C, runs the query and then shares the output, unknowingly revealing sensitive salary data to User C.
- SQL supports only discretionary access control through the GRANT and REVOKE statements.
- The mechanism is based on the concepts of authorization identifiers, ownership, and privileges

Granting Privileges to Other Users (GRANT)

Granting Privileges to Other Users (GRANT)

```
GRANT [PrivilegeList | ALL PRIVILEGES]
ON ObjectName
TO [AuthorizationIdList | PUBLIC]
[WITH GRANT OPTION];
```

- Grant All: Use ALL PRIVILEGES to give a user all permissions without listing them individually.
- Access for All: Use PUBLIC to allow everyone, now and in the future, to access an object.
- Object Types: Permissions can be given for tables, views, or other database objects.
- Sharing Privileges: WITH GRANT OPTION lets users pass their permissions to others.
- Restrict Sharing: Without WITH GRANT OPTION, users can't share their permissions.

Note: MySQL doesn't support PUBLIC.
But can still grant privileges to specific users (using % for any host).

GRANT all privileges

- Example 1:
  - Give the user with authorization identifier manager all privileges on employees table.

```
GRANT ALL PRIVILEGES
ON employees
TO manager
WITH GRANT OPTION;
```

```
mysql> GRANT ALL PRIVILEGES
    -> ON employees
    -> TO manager
    -> WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)
```

The user manager can now:

- view, **insert, update**, and **delete** data in employees table.
- use employees table and all employees columns in any table that he or she creates.
- because **WITH GRANT OPTION**, manager can share these privileges with other users.

GRANT specific privileges

- Example 2:
  - Give users personnel and director the privileges **SELECT** and **UPDATE** on column salary of the employees table.

```
GRANT SELECT, UPDATE (salary)
ON employees
TO personnel, director;
```

```
mysql> GRANT SELECT, UPDATE (salary)
    -> ON employees
    -> TO personnel, director;
Query OK, 0 rows affected (0.00 sec)
```

- The keyword omitted **WITH GRANT OPTION**, so that users personnel and director cannot share these privileges with other users.

GRANT specific privileges to PUBLIC

- Example 3:
  - Give all users the privilege **SELECT** on departments table.

```
GRANT SELECT
ON departments
TO PUBLIC;
```

- Using the keyword **PUBLIC** means that all users (now and in the future) can view all the data in departments table.
- There is no need to use **WITH GRANT OPTION** because every user has access to the table.

Note:
MySQL doesn't support PUBLIC.
But can still grant privileges to specific users (using % for any host).

```
GRANT SELECT
ON departments
TO 'root'@'localhost';
```

```
mysql> GRANT SELECT ON departments TO PUBLIC;
ERROR 1410 (42000): You are not allowed to create a user with GRANT
mysql> GRANT SELECT ON departments TO 'root'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Revoking Privileges from Users (REVOKE)

Revoking Privileges from Users (REVOKE)

```
REVOKE [GRANT OPTION FOR] {PrivilegeList | ALL PRIVILEGES}
ON ObjectName
FROM [AuthorizationIdList | PUBLIC]
[RESTRICT | CASCADE];
```

- **ALL PRIVILEGES** refers to all the permissions (like **SELECT, INSERT, DELETE**, etc.) a user has been given.
- The optional **GRANT OPTION FOR** clause lets you revoke the privileges sharing (given through **WITH GRANT OPTION**), without affecting the user's access to the table.
- If **REVOKE** a privilege and it causes an object (like a view) to no longer be needed, the command will fail unless you use the **CASCADE** keyword. If **CASCADE** is used, MySQL will automatically delete any unnecessary objects, like views or constraints, that are left behind after revoking the privilege.

Effects of REVOKE

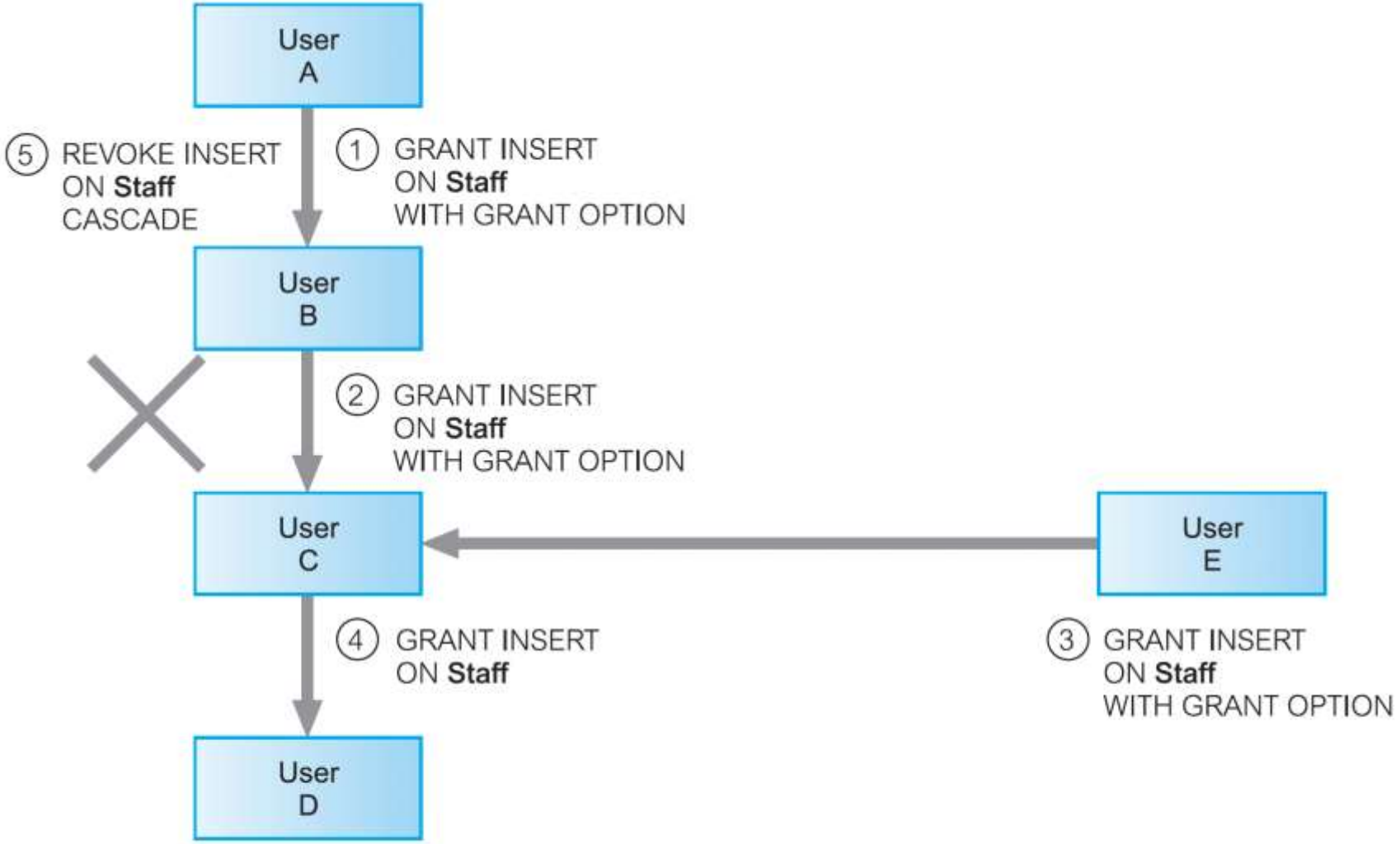

Figure 7.1 Effects of REVOKE.

- User A grants User B INSERT privilege on the Staff table WITH GRANT OPTION (step 1).
- User B passes this privilege on to User C (step 2).
- User C gets the same privilege from User E (step 3).
- User C then passes the privilege on to User D (step 4).
- When User A removes the INSERT privilege from User B (step 5), it doesn't affect User C because User C got the privilege from User E.
  If User E hadn't given User C the privilege, the removal would have also applied to User C and User D.

REVOKE specific privileges from PUBLIC

- Example 4:
  - Revoke the privilege **SELECT** on departments table from all users.

```
REVOKE SELECT
ON departments
FROM PUBLIC;
```

Note:
MySQL doesn't support PUBLIC.
But can still grant privileges to specific users (using % for any host).

```
REVOKE SELECT
ON departments
FROM 'root'@'localhost';
```

```
mysql> REVOKE SELECT
    -> ON departments
    -> FROM 'root'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

REVOKE specific privileges from named user

- Example 5:
  - Revoke all privileges you have given to director on the employees table.

```
REVOKE ALL PRIVILEGES
ON employees
FROM director;
```

```
mysql> REVOKE ALL PRIVILEGES
    -> ON employees
    -> FROM director;
Query OK, 0 rows affected (0.00 sec)
```

Checking privileges:

SHOW GRANTS FOR manager

SHOW GRANTS FOR director;

SHOW GRANTS FOR personne

```
mysql> SHOW GRANTS FOR manager;
+-----------------------------------------------------------------------------------------------------------+
| Grants for manager@%                                                                                      |
+-----------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `manager`@`%`                                                                       |
| GRANT ALL PRIVILEGES ON `db_dml3_dept_emp_loc`.`employees` TO `manager`@`%` WITH GRANT OPTION             |
+-----------------------------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)

mysql> SHOW GRANTS FOR director;
+-------------------------------------+
| Grants for director@%               |
+-------------------------------------+
| GRANT USAGE ON *.* TO `director`@`%` |
+-------------------------------------+
1 row in set (0.00 sec)

mysql> SHOW GRANTS FOR personnel;
+-----------------------------------------------------------------------------------------------------------+
| Grants for personnel@%                                                                                    |
+-----------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `personnel`@`%`                                                                     |
| GRANT SELECT, UPDATE (`salary`) ON `db_dml3_dept_emp_loc`.`employees` TO `personnel`@`%`                  |
+-----------------------------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Innovating Solutions
Menginovasi Penyelesaian