Lecture 9 DB-Security

CS211 - Introduction to Database

DB Security

- To protect databases against compromises of their confidentiality, integrity and availability.

- DBMS security models
  - Authentication: a process by which you verify that someone is who they claim they are
  - Authorization: the process of establishing if the user (who is already authenticated), is permitted to have access to a resource. Authorization determines what a user is and is not allowed to do.
  - Access Control: the process of enforcing the required security for a particular resource.

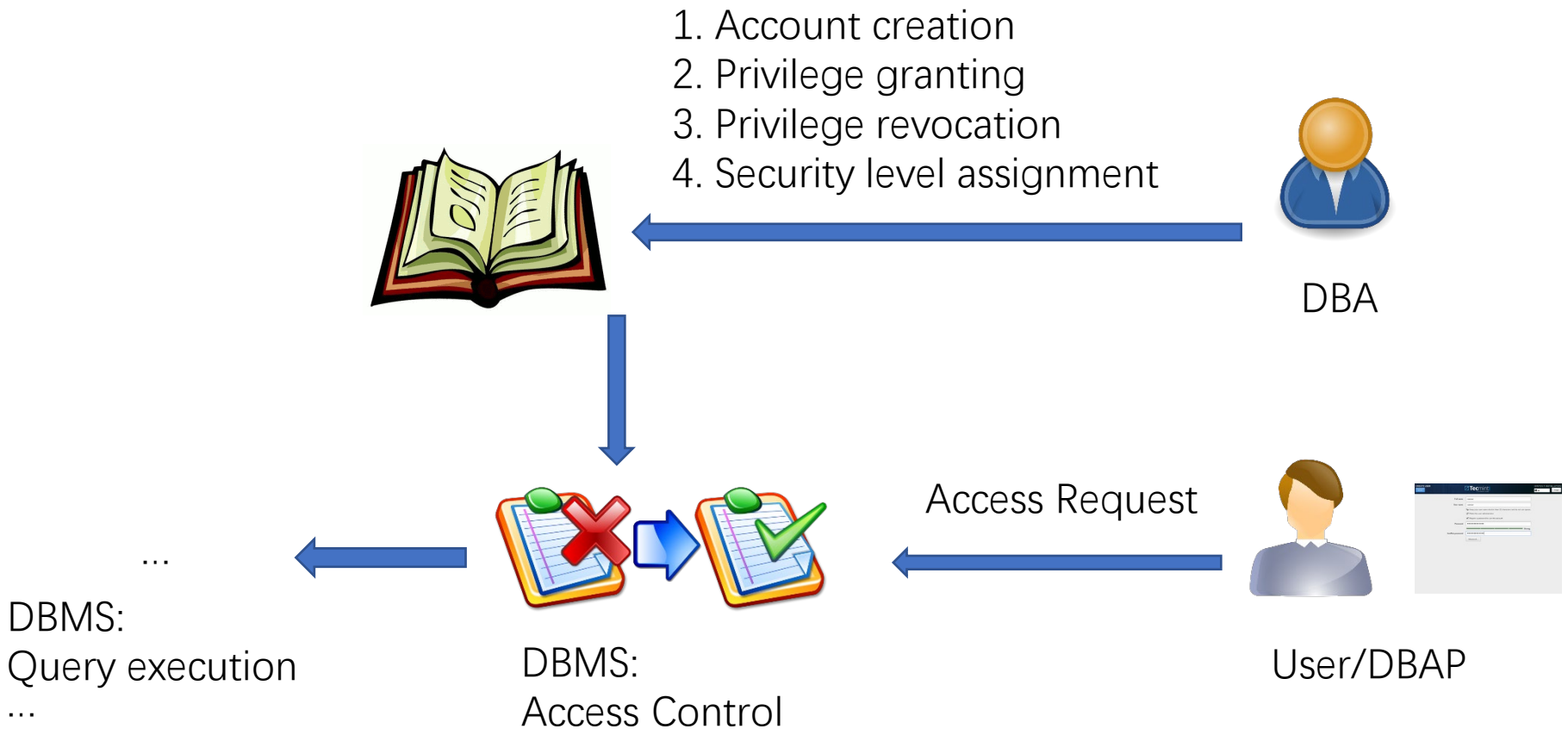
Discretionary Access Control

- **granting** and **revoking privileges**

- The DBA
  - granting privileges to users who need to use the system
  - classifying users and data in accordance with the policy of the organization

- Administrator account
  - 1. Account creation
  - 2. Privilege granting
  - 3. Privilege revocation
  - 4. Security level assignment

  - Action 1 is authentication, whereas 2 and 3 are discretionary authorization and 4 is used to control mandatory authorization


1. Account creation
2. Privilege granting
3. Privilege revocation
4. Security level assignment
DBA
Access Request
User/DBAP
DBMS:
Access Control
...
DBMS:
Query execution
...

AccessRule

- (S O t P)
  - Subject: user/user group, account
  - Object: data (granularity: database/table/attribute/tuple/data item)
  - t: right (create/update/insert/delete/alter/query)
  - Predicate: condition

  - S own right t on O when P is satisfied

  - {AccessRule} stored in data dictionary or sys directory

Example – security requirement

employee(eID, eName, age, gender, salary, dID, manager)

User group

HR: Access all information

General office: read eName, dID

An employee: his/her own record, read only

Manager: query employee in the department

Senior Manager: query all information, read only

Example - implementation

S	O	t	P
HR	Employee	Read, del, insert, update	
General office	Employee(eName, dID)	read	
Senior manager	Employee	read	
Employee	Employee	read	eID=:userID
Department manager	Employee	read	Manager=:userID

Example – Matrix - (S O: t)

	O1	O2	…	On
S1	All	All	…	All
S2	Read	-	…	-
…	…	…	…	…
Sn	-	Read	…	-

Given a matrix, and there is a query request: select-from-where

How do we validate the request?

List all required privileges and objects

```
UPDATE instructor SET workload=workload*1.1
        WHERE iID in (select iID from course where cID='211')
```

List all required privileges and objects

UPDATE instructor SET workload=workload*1.1

WHERE iID in (select iID from course where cID='211')

O	t
Instructor.workload	Update
Instructor.workload	Select
Instructor.iID	Select
Course.cID	Select
Course.iID	Select

List all privileges and objects

```
DELETE FROM student WHERE sID NOT IN
   (SELECT sID FROM rc)
```

List all privileges and objects

DELETE FROM student WHERE sID NOT IN
(SELECT sID FROM rc)

O	t
Student	Del
Student.sid	Select/read
rc.sID	Select

Which privilege on table instructor is NOT needed

instructor(iID, iName, dID, workload)

UPDATE instructor SET workload = workload+1 WHERE dID='03'

1. update(workload)
2. read(iName)
3. read(dID)
4. read(workload)

Example – View - (S O t P)

- Data item access control

```
CREATE VIEW empV1 AS select * from employee

CREATE VIEW empV2 AS select eName, dID from employee
(for general office)
```

- Predicate

```
CREATE VIEW empV3 AS
     select * from employee where eID = :userID

CREATE VIEW empV4 AS
    select * from employee where manager = :userID
```

Consider table student(sID,sName,GPA),

Is it possible to grant a user modification privileges only on those students whose GPA is above average? (by creating a view)

1. Yes
2. No

Consider table student(sID,sName,GPA),

Is it possible to grant a user modification privileges only on those students whose GPA is above average? (by creating a view)

1. Yes

2. No

```
CREATE VIEW stu_above_avg AS
    select * from student where GPA >
            ( select avg(GPA)  from student );
```

The view modification is not allowed

User levels

- User/Role levels: DBA → DBA developer → End User

- From right to left
- Level 1: select (read database/table/tuple/attribute…)
- Level 2: modify
  - Insert (tuple)
  - Update (data item)
  - Delete (tuple)
- Level 3: create
  - Create (database/table/index/view…)
  - Alter
  - Drop

1. Higher level has all privileges of lower level
2. Higher level has the right of granting

DBA – create a new user account

MariaDB/mySQL

- CREATE USER ‘test'@'localhost' IDENTIFIED BY ‘myPassword’;
- CREATE USER ‘test1'@'localhost' IDENTIFIED BY ‘myPassword’;

SQL - DCL

```
GRANT {privilege list}
ON {table/view}
TO {user/role list}
[with grant option]
```

- Privilege: select | insert | update | delete
- Only those privileges owned by the higher level user can be granted to lower level user
- With grant option: rights can be transited

```
GRANT all
ON proj1.*
TO test1@localhost
```

Example – security requirement

employee(eID, eName, age, gender, salary, dID, manager)

HR: Access all information

General office: read eName, dID

An employee: his/her own record, read only

Manager: query employee in the department

Senior Manager: query all information, read only

SQL DCL - Example

	eID
A Senior Manager	1001
A Department Manager	1021
A staff from HR	2001
A staff from General office	5001

```
CREATE VIEW empV1 AS select * from employee

CREATE VIEW empV2 AS select eName, dID from
employee (for general office)

CREATE VIEW empV3 AS
    select * from employee where eID = :userID

CREATE VIEW empV4 AS
   select * from employee where manager = :userID
```

```
GRANT all ON employee TO 2001;
GRANT select ON empV2 TO 5001;
GRANT select ON empV3 TO public;
GRANT select ON empV4 TO 1021;
```

SQL – DCL - revoke

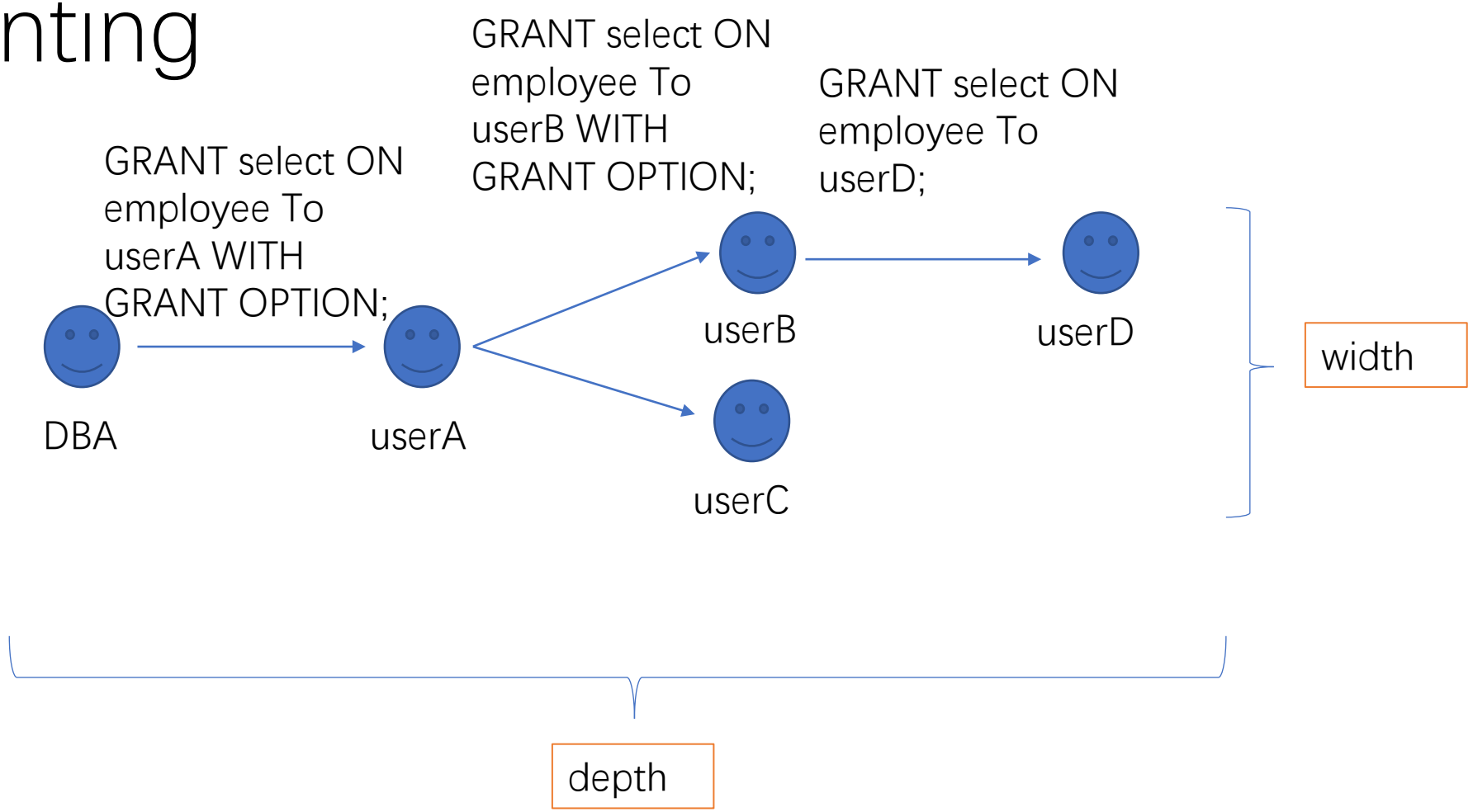
```
REVOKE {privilege list}
ON {table/view}
FROM {user/role list}
[cascade/restrict]
```

Cascade: Also revoke privileges granted from privileges being revoked (transitively), unless also granted from another source

Restrict: only revoke from directly related users(R)

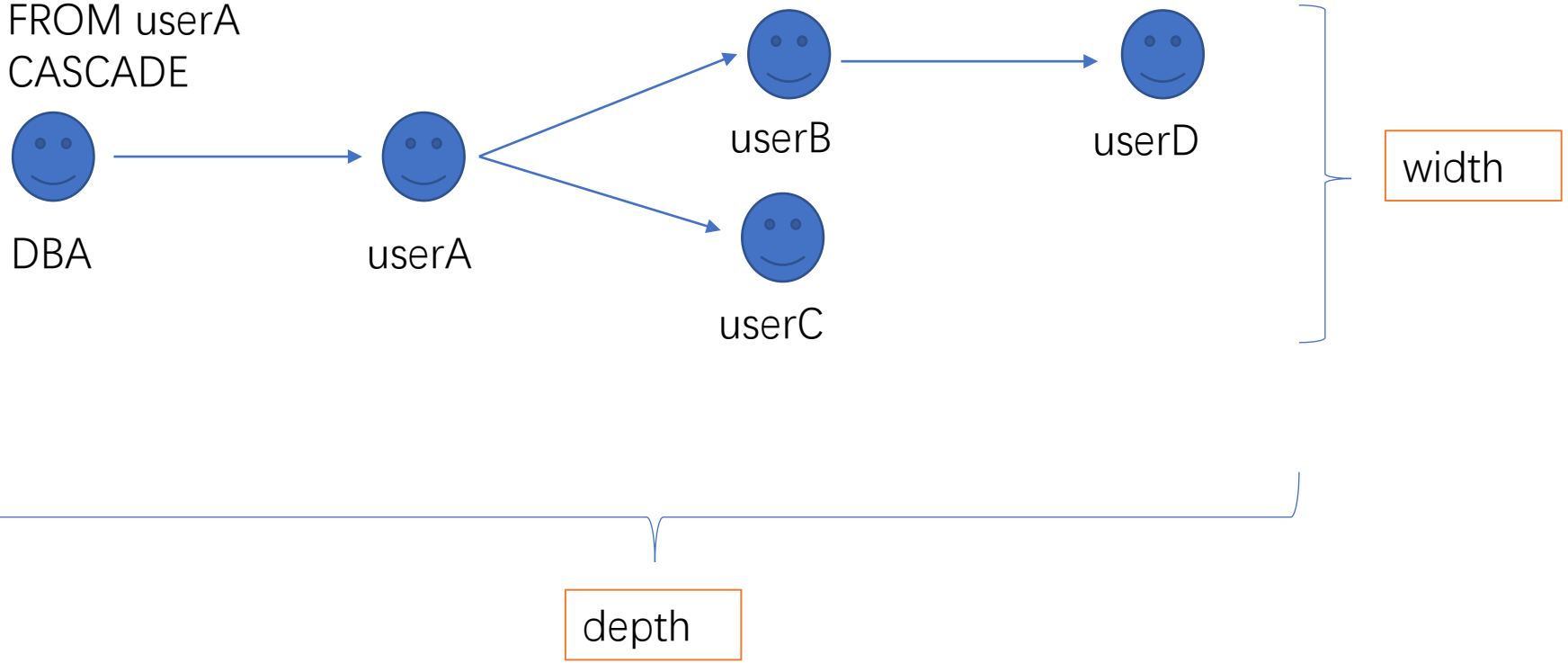
Both are not working in MariaDB/mySQL

Granting


GRANT select ON employee To userA WITH GRANT OPTION;
GRANT select ON employee To userB WITH GRANT OPTION;
GRANT select ON employee To userD;
DBA
userA
userB
userC
userD
width
depth

Revoking

REVOKE select
ON employee
FROM userA
CASCADE



Consider table *R* with owner A. Suppose we have the sequence:

(1) A grants select(*R*) to B with grant option;

(2) A grants select(R) to C with grant option;

(3) B grants select(R) to D;

(4) C grants select(R) to D;

(5) A revokes select(R) from B with restrict.

(6) D issue a select * from R; statement.

Does step (6) generate an error?

- Yes
- No

Consider table *R* with owner A. Suppose we have the sequence:

(1) A grants select(*R*) to B with grant option;

(2) A grants select(R) to C with grant option;

(3) B grants select(R) to D;

(4) C grants select(R) to D;

(5) A revokes select(R) from B with restrict.

(6) D issue a select * from R; statement.

Does step (6) generate an error?

- Yes
- No

Mandatory security

- Data object security level
  - Top secret, secret, confidential, unclassified
- User security level
  - Top secret, secret, confidential, unclassified

- User S is not permitted to read data object O unless Level(S)>=Level(O)
- User S is not permitted to write data object O when Level(S)>Level(O)
  - The level(O) is promoted as level(S) once O is written
  - If high level user write O, then low level user can not read it

Mandatory security – Implementation

- Extend schema definition

R(A1:D1, A2:D2, ⋯, An:Dn)

R(A1:D1,C1, A2:D2,C2, ⋯, An:Dn,Cn, TC)

C1⋯Cn: security level for attributes

TC: security level for tuple