ĐẠI HỌC
BÁCH KHOA

25 YEARS ANNIVERSARY
SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

IT3090E - Databases

Chapter 4: Structured Query Language
part 4

Muriel VISANI
murielv@soict.hust.edu.vn

Contents

- Chapter 1: Introduction
- Chapter 2: Relational databases
- Chapter 3: Relational algebra
- **Chapter 4: Structured Query Language (SQL)**
- Chapter 5: Database Design
- Chapter 6: Indexing
- Chapter 7: Query processing and optimization
- Chapter 8: Constraints, rules and triggers
- Chapter 9: Security
- *(Optional) Chapter 10: Transactions: concurrency and recovery*

Outline of Chapter 4

1. Data Definition and Data Manipulation SQL languages
2. Creating and managing views
3. Privileges and User Management in SQL

Global Outline of Chapter 4

- Chapter 4 - Part 1:
  - 1 - Introduction to SQL
  - 2 – Definition of a Relational Schema (DDL)
  - 3 – Data Manipulation: 3.1.-3.3. Insertion, deletion, updates
- Chapter 4 - Part 2:
  - 3.4. Data Manipulation Language for Querying (simple queries)
- Chapter 4 - Part 3:
  - 3.4. Data Manipulation Language for Querying (complex queries)
- Chapter 4 - Part 4:
  - 4. Privileges and User Management in SQL

Learning objective of Chapter 4 - part 4

- Have experience with a DBMS: **manage user account and database access permissions**

Keywords of Chapter 4

Keyword	Description
Query	A request (SQL statement) for information from a database
Subquery	A subquery (inner query, nested query) is a query within another (SQL) query.
Privileges	Database access permissions
View	A view is the result set of a stored query on the data, which the database users can query just as they would in a persistent database collection object.

Privileges and User Management in SQL

1. Privileges
2. Creating users
3. Granting privileges
4. Revoking privileges
5. Roles

3.1. Privileges

- There are two types of **privileges**:
  - System Privileges: This indicate user to CREATE, ALTER, or DROP database elements.
  - Object Privileges: This allows user to EXECUTE, SELECT, INSERT, or DELETE data from database objects to which the privileges apply.
- **Roles** are the collection of privileges or access rights.

3.1. Privileges – system privileges

- CREATE object: allows users to create the specified object in their own schema.
- CREATE ANY object: allows users to create the specified object in any schema.
- The same rules apply for the ALTER and DROP system privileges

3.1. Privileges – object privileges

- SELECT, INSERT, DELETE, UPDATE: privileges on table/view
- REFERENCES: privilege on a relation: right to refer to that relation in an integrity constraint
- USAGE: the right to use that element in one’s own declarations
  - Synonym for “no privileges”
- TRIGGER: privilege on a relation; the right to define triggers on that relation
- EXECUTE: the right to execute a piece of code, such as a procedure or function

3.2. Creating users

- Syntax: variations in different database platforms
  - Creating an user in Oracle, MySQL:

    ```
    CREATE USER username IDENTIFIED BY password;
    ```

  - Creating an user in PostgreSQL:

    ```
    CREATE USER username
    [[WITH] options] PASSWORD password;
    ```

  - Deleting:

    ```
    DROP USER username [CASCADE];
    ```

    CASCADE will remove all schema objects of the user before deleting the user

- Example:

  ```
  CREATE USER toto IDENTIFIED BY pwdtoto
  ```

3.3. Granting privileges

- Syntax:

  GRANT <privilege list> ON <database element> TO <user list>

  [WITH GRANT OPTION] ;

  - <privilege list> : INSERT, SELECT, …, ALL PRIVILEGES
  - <database element>: a table, a view
  - WITH GRANT OPTION:
    - the user is allowed to grant the privilege to other users

- Example:

  GRANT SELECT, INSERT ON student TO tom WITH GRANT OPTION;

3.3. Granting privileges

- A user is referred to by authorization ID, typically their login name
- There is an authorization ID called PUBLIC.
  - Granting a privilege to PUBLIC makes it available to any authorization ID.
- A user has all possible privileges on the objects (such as relations) that they create.
  - The object owner may grant privileges to other users (authorization ID’s), including PUBLIC.
  - The object owner may also grant privileges WITH GRANT OPTION

3.4. Revoking privileges

- Syntax (for revoking object privileges):

  REVOKE <privilege list> ON <database element> FROM <user list>

  [CASCADE| RESTRICT] ;

  - CASCADE : revokes the privileges in <privilege_list>, plus all privileges that depend on the privileges being revoked.
    - It removes the revoked rights from all users that have been granted the rights by the user revoked
    - If you want those other users to retain the rights granted by the user with the GRANT OPTION, you must then manually assign those rights explicitly to those other users.
  - RESTRICT: does not to revoke the specified privilege if there are any dependent privileges.

3.4. Revoking privileges

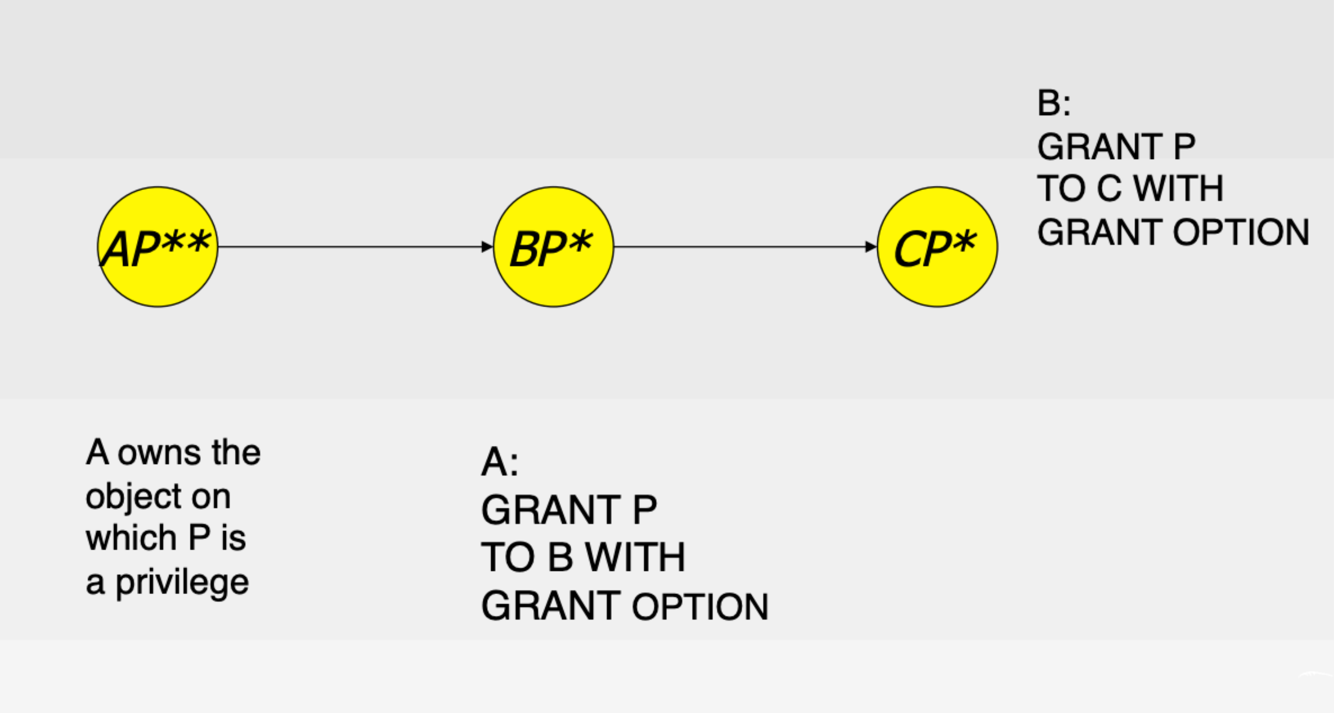
- Syntax (for revoking grant privileges):

REVOKE GRANT OPTION FOR ….. [CASCADE] ; : remove the grant option

- Examples:

```
REVOKE  INSERT ON student FROM tom CASCADE;
REVOKE  GRANT OPTION FOR student;
```

3.4. GRANT diagrams

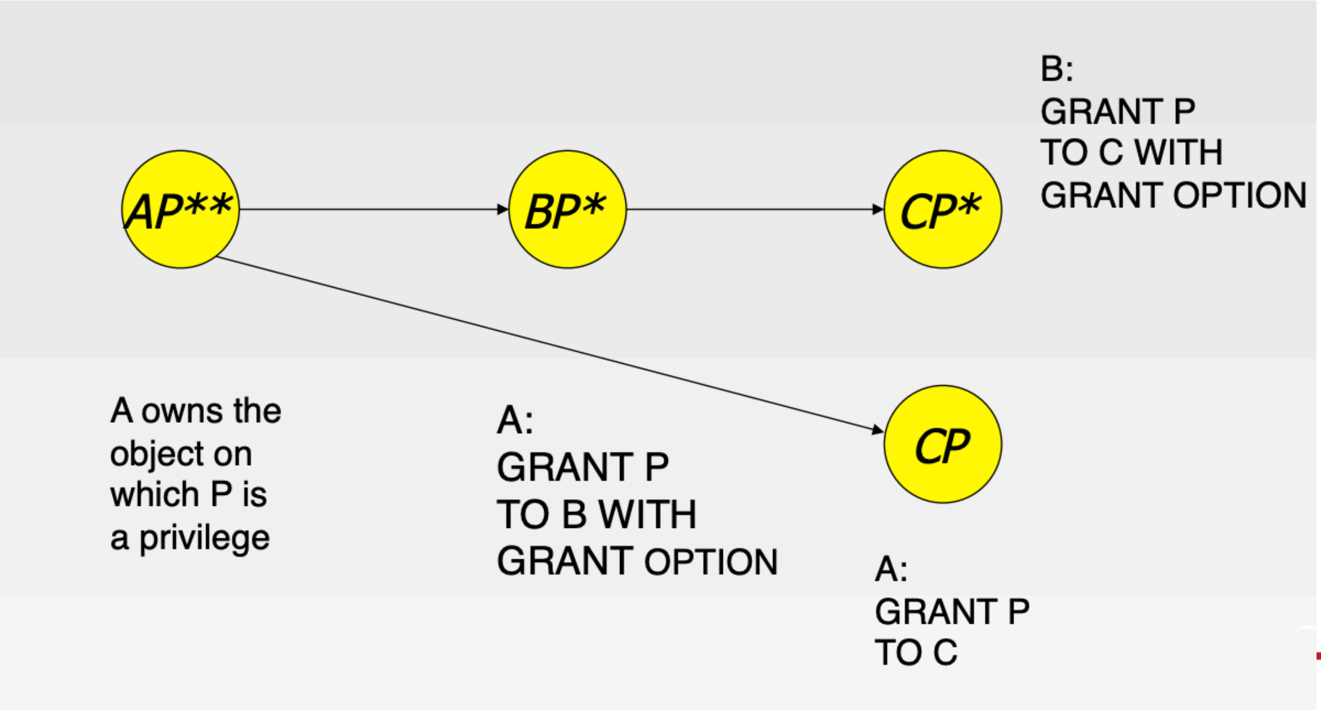


B:
GRANT P
TO C WITH
GRANT OPTION

A owns the object on which P is a privilege

A:
GRANT P
TO B WITH
GRANT OPTION

3.4. GRANT diagrams



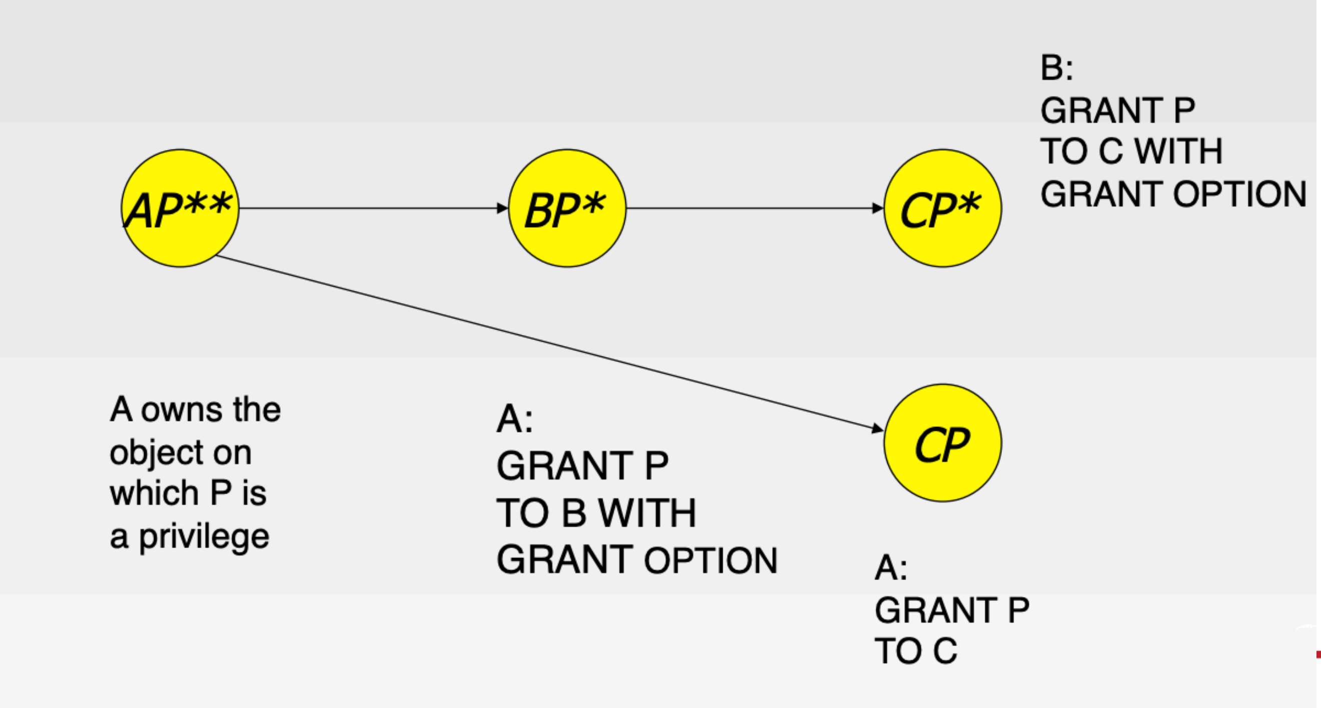
B:
GRANT P
TO C WITH
GRANT OPTION

A owns the
object on
which P is
a privilege

A:
GRANT P
TO B WITH
GRANT OPTION

A:
GRANT P
TO C

3.4. GRANT diagrams



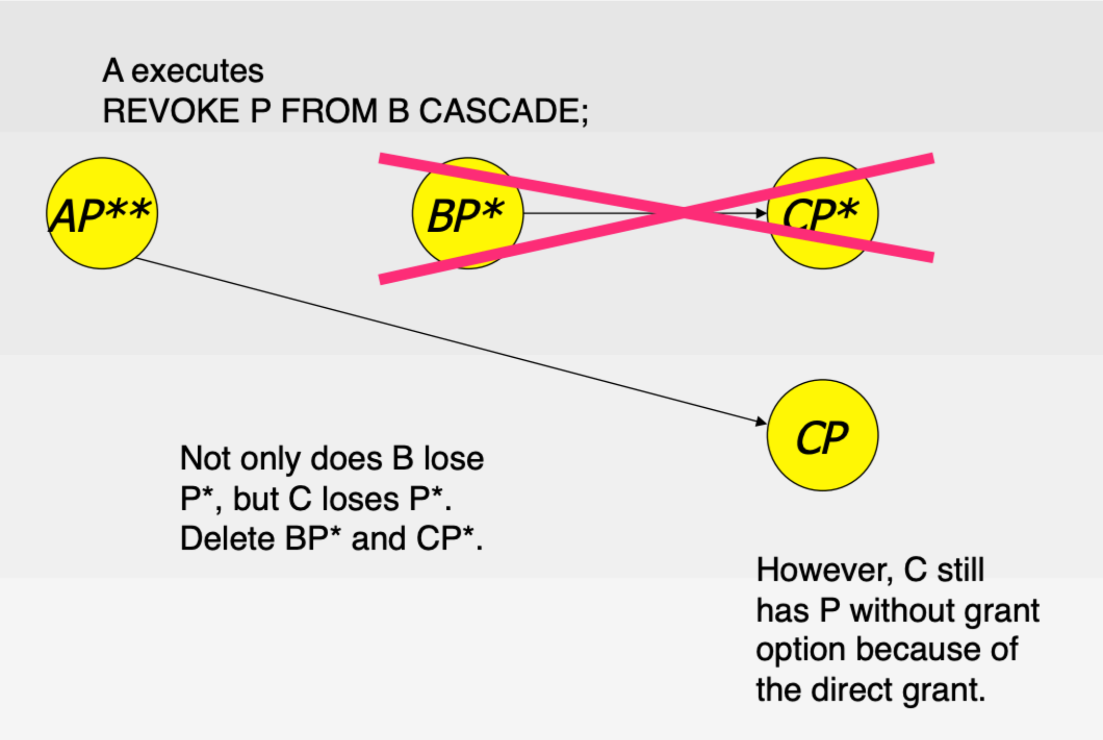
B:
GRANT P
TO C WITH
GRANT OPTION

A owns the object on which P is a privilege

A:
GRANT P
TO B WITH
GRANT OPTION

A:
GRANT P
TO C

3.4. GRANT diagrams

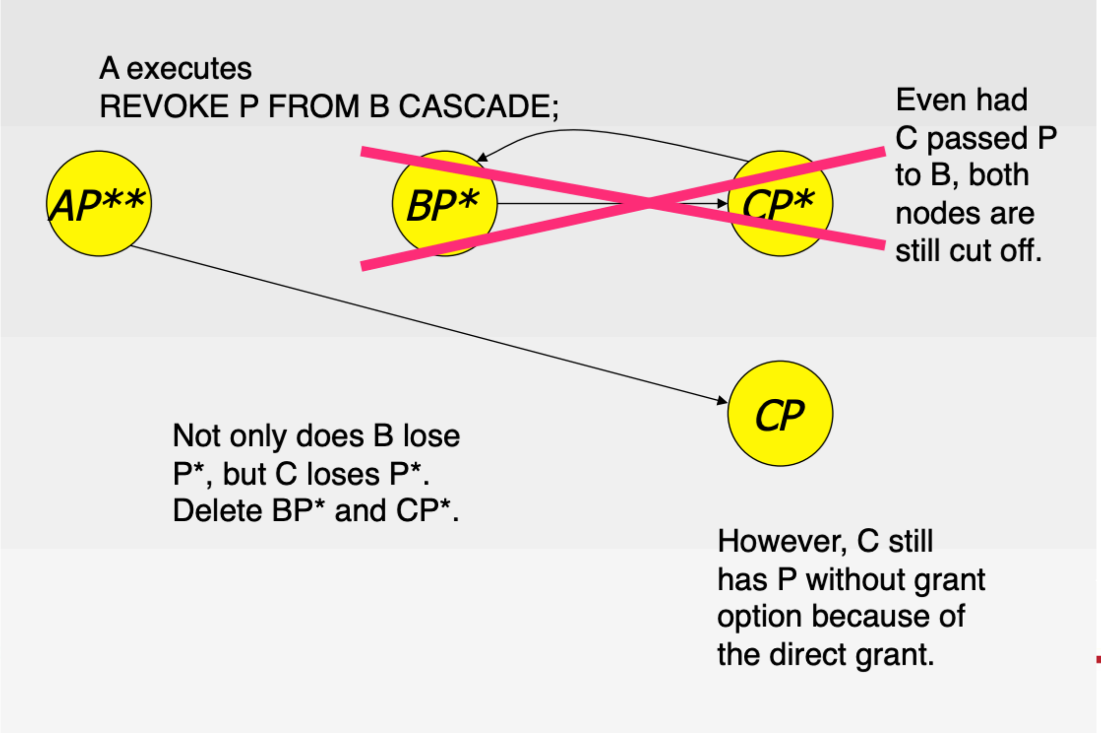


A executes
REVOKE P FROM B CASCADE;

Not only does B lose
P*, but C loses P*.
Delete BP* and CP*.

However, C still
has P without grant
option because of
the direct grant.

3.4. GRANT diagrams



A executes
REVOKE P FROM B CASCADE;

Even had C passed P to B, both nodes are still cut off.

Not only does B lose P*, but C loses P*. Delete BP* and CP*.

However, C still has P without grant option because of the direct grant.

3.5. Roles in SQL Server

- Roles are a part of the tiered security model:
  - Login security: Connecting to the server
  - Database security: Getting access to the database
  - Database objects: Getting access to individual database objects and data

- Server roles are maintained by the database administrator (DBA) and apply to the entire server, not an individual database file.

- Database roles are applied to an individual database.

3.5. Roles in SQL Server – server roles

- The PUBLIC role sets the basic default permissions for all users.
  - Every user that's added to SQL Server is automatically assigned to the public role—you don't need to do anything
  - The public server role is granted VIEW ANY DATABASE permission and the CONNECT permission on the default endpoints.
- The PUBLIC server role is not a fixed server role, because the permissions can be changed

3.5. Roles in SQL Server – server roles

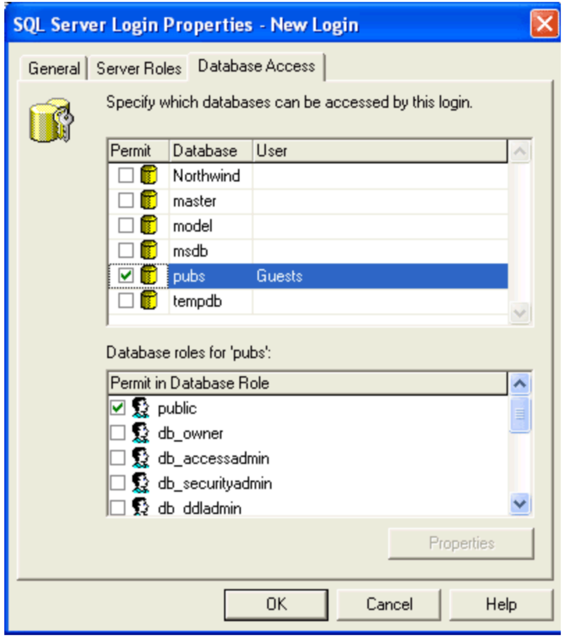
- The fixed server roles are applied serverwide, and there are several predefined server roles:
    - SysAdmin: Any member can perform any action on the server.
    - ServerAdmin: Any member can set configuration options on the server.
    - SetupAdmin: Any member can manage linked servers and SQL Server startup options and tasks.
    - Security Admin: Any member can manage server security.
    - ProcessAdmin: Any member can kill processes running on SQL Server.
    - DbCreator: Any member can create, alter, drop, and restore databases.
    - DiskAdmin: Any member can manage SQL Server disk files.
    - BulkAdmin: Any member can run the bulk insert command.
    - .

3.5. Roles in SQL Server – database roles

- You may need to create your own, but you have access to several predefined database roles:
  - db_owner: Members have full access.
  - db_accessadmin: Members can manage Windows groups and SQL Server logins
  - db_datareader: Members can read all data.
  - db_datawriter: Members can add, delete, or modify data in the tables.
  - db_ddladmin: Members can run dynamic-link library (DLL) statements.
  - db_securityadmin: Members can modify role membership and manage permissions.
  - db_bckupoperator: Members can back up the database.
  - db_denydatareader: Members can't view data within the database.
  - db_denydatawriter: Members can't change or delete data in tables or views.

3.5. Roles in SQL Server

- In SQL Server, you can change the role of a user (by default PUBLIC) through the SQL Server interface



Specify which databases can be accessed by this login.

Permit	Database	User
□	Northwind	
□	master	
□	model	
□	msdb	
☑	pubs	Guests
□	tempdb	

Database roles for 'pubs':

Permit in Database Role	
☑	public
□	db_owner
□	db_accessadmin
□	db_securityadmin
□	db_ddladmin

Summary

- Privileges and User Managements
  - Privileges
  - Creating user
  - Granting / Revoking privileges

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!

soict.hust.edu.vn/ fb.com/groups/soict