CSU34041

Database Security

Yvette Graham
ygraham@tcd.ie

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Student Online Teaching Advice Notice

The materials and content presented within this session are intended solely for use in a context of teaching and learning at Trinity.

Any session recorded for subsequent review is made available solely for the purpose of enhancing student learning.

Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at Trinity.

Please be mindful of your physical environment and conscious of what may be captured by the device camera and microphone during videoconferencing calls.

Recorded materials will be handled in compliance with Trinity's statutory duties under the Universities Act, 1997 and in accordance with the University's policies and procedures.

Further information on data protection and best practice when using videoconferencing software is available at https://www.tcd.ie/info_compliance/data-protection/.

Overview

- Integrity v Security
- Access Control
- Discretionary Access Control
- Mandatory Access Control
- Discretionary v Mandatory Access Control
- Role-based Access Control

Recall from last lecture ...

Integrity vs Security


Integrity and **Security** are related but they are not the same

- Integrity is concerned with *accidental* corruption
- Security is concerned with *deliberate* corruption

Integrity

- Integrity Constraints

Security

- Security Policies
- Access Control

Access Control

Access Control

- Prevent unauthorised persons from accessing the system
  - to obtain information
  - to make malicious changes
- DBMS can restrict access to the DB
  - User Accounts
  - Privileges
  - Security Levels

Access Control

Database Administrator (DBA) is responsible for:

- User Account Creation
  - Encrypted table maintained by the DBMS
- Privilege Granting and Revocation
  - Discretionary Access Control
- Security Level Assignment
  - Mandatory Access Control
- Role-Based Access

Discretionary Access Control

Privileges

Access privileges can be specified at two levels:

– Account Level

- DBA can specify the privileges that each account holds independently of the relations in the database

– Relation Level

- DBA can control privilege to access each individual relation or view in the database

Account Level Privileges

These privileges apply to the capabilities provided to an account

- Examples of privileges include:
  - CREATE SCHEMA
  - CREATE TABLE
  - CREATE VIEW
  - ALTER
  - DROP

Relation Level Privileges

- Can be specified on *entire relations* or *specific attributes*
  - determine what operations can be performed
- Each relation has an "owner"
  - Typically the account that *created the table*
  - This account then controls the granting and revoking of privileges to other accounts for that table

Relation Level Privileges

Privilege types are:

- Read Privilege
  - gives an account the ability to use SELECT to retrieve rows from this relation
- Modification Privilege
  - gives an account the ability to use INSERT, UPDATE and DELETE to modify rows in this relation
- Reference Privilege
  - gives an account the ability to refer to this relation when specifying integrity constraints

Views

Views are an important *discretionary authorisation mechanism*

- Allow the owner of a relation(s) to grant partial access to the information contained in that relation
  - Access to a restricted set of attributes
  - Access to a restricted set of rows
- A *view* acts as a new relation in the DB

Views

A *view* is created from a query using CREATE VIEW

```
CREATE VIEW PopularBooks AS SELECT ISBN, Title, Author, PublishDate
FROM Books  WHERE IsPopular = 1
```

Granting Privileges

- Privileges are allocated to users using the GRANT command in SQL
  - GRANT privilege TO user;
  - GRANT privilege ON relation TO user;
- The owner of a relation
  - automatically has all the relation privileges granted to them
  - can use the GRANT command to specify user privileges for that relation

Revoking Privileges

- It is often desirable to remove a privilege from a particular user
  - temporary access
  - abuse of privilege
- In SQL the REVOKE command is used to cancel privileges
  - REVOKE privilege FROM user;
  - REVOKE privilege ON relation FROM user;

Propagation of Privileges

- Whenever the owner A of a relation R grants privileges on R to another user B, the privilege can be given with or without the GRANT OPTION
  - If the GRANT OPTION is given, then B can also grant that privilege on R to other users
- Command Syntax
  - GRANT privilege ON relation TO user WITH GRANT OPTION;

Dangers of Propagation

A is the owner of relation R

- A grants B the DELETE privilege on R, with GRANT OPTION
- B grants C the DELETE privilege on R, also with GRANT OPTION
- In this way, privileges can propagate without the knowledge of the relation owner
- If A revokes the privilege granted to B, all the privileges that B propagated should automatically be revoked by the DBMS

Dangers of Propagation

A is the owner of relation R

- A grants B the DELETE privilege on R, with GRANT OPTION
- A grants C the DELETE privilege on R, also with GRANT OPTION
- B and C both grant D the DELETE privilege on R
- B later revokes the DELETE privilege from D
- However, D continues to have the DELETE privilege, as it was also granted from C

Example

A DBA creates four user accounts

- James, Victoria, Henry and George

- The DBA only wants James to be able to create relations in the DB
  - GRANT CREATE TABLE TO James;

- James now has the ability to create tables
  - He does not have the ability to grant CREATE TABLE to other users

Example

- James creates two tables:

Land

ID_num	Name	Kingdom_ID	Family_ID	Area	Value

Lords

Tax_num	Name	Address	Age	Decoration	Salary

- James is the owner of these tables
- He automatically has all relation privileges on each of these tables

Example

James wants to grant Henry the ability to insert, retrieve and delete rows in both of these tables

- However, he doesn't want Henry to be able to pass this ability on to other users

- James issues the following command:

```
GRANT INSERT, SELECT, DELETE ON Land, Lords TO Henry;
```

Example

- James wants to grant Victoria the ability to retrieve information from either of the tables
- He also trusts her to pass on this ability to other users of the database
- James issues the following command:

GRANT SELECT ON Land, Lords TO Victoria WITH GRANT OPTION;

Example

- Victoria can now propagate this privilege to other user accounts using the GRANT command
- She wants to grant George the ability to retrieve information from the Lords table
- Victoria issues the following command:

```
GRANT SELECT ON Lords TO George;
```

Example

- James decides to revoke the SELECT privilege on Lords from Victoria
- James issues the following command:

```
REVOKE SELECT ON Lords FROM Victoria
```

- The DBMS must now automatically revoke the SELECT privilege from George as it was granted to him by Victoria, who no longer has the privilege

Example

- James feels a bit bad, and wants to give Victoria back the ability to see the Lords information
- He also wants Victoria to be able to propagate this privilege again
- However, he only wants her to be able to see:
  - name, age and address
  - Lords who received the “Victoria Cross”

- How does he achieve this?

Example

- James creates a View on the Lords table:

```
CREATE VIEW Lords_Restricted AS
SELECT Name, Age, Address
FROM Lords
WHERE Decoration = “Victoria Cross”;
```

- After the view is created, James grants SELECT to Victoria as follows

```
GRANT SELECT ON Lords_Restricted TO Victoria WITH GRANT
    OPTION
```

Example

- Finally, James wants to grant Henry the ability to update the Salary field in the Lords table
- James issues the following command:

```
GRANT UPDATE (Salary) ON Lords TO Henry
```

- UPDATE and INSERT are examples of privileges that can be specified on attribute(s)
  - DELETE and SELECT are not attribute specific
- That functionality is handled using Views

Mandatory Access Control

Mandatory Access Control

- Mandatory Access Control (MAC) classifies data and users based upon *security levels*
  - can be combined with discretionary access control
  - desirable in government, military and intelligence
- Not commonly available in Commercial DBMS
  - Some companies, for instance Oracle, have released special versions of DBMS for government which include MAC

Mandatory Access Control

- Most simple example of security levels are:

  – Top Secret, Secret, Confidential, Unclassified

  – TS ≥ S ≥ C ≥ U

- Each *subject* and *object* are given a security level
  - A subject is said to be given access to an object.

  – **Subject** (User Account, Application Program…)

  – **Object** (Relation, Tuple, Attribute, View, Operation…)

- The security level of the subject is compared with that of the object for the DBMS to decide if the action is permitted

Discretionary v Mandatory Access Control

Access Control Comparison

Discretionary  Mandatory

- Discretionary Access Control
  - Flexible
  - Complex to manage
  - Can be vulnerable to malicious attacks
- Mandatory Access Control
  - Rigid
  - Very secure
- Trade-off between Security and Applicability

Role-based Access Control

Role-based Access Control

- Privileges and other permissions are associated with organisational roles rather than individual user accounts
- Users are then assigned to appropriate roles
- Roles can be created in SQL using
  - CREATE ROLE
  - DESTROY ROLE

Role-based Access Control

- GRANT and REVOKE are then used to allocate privileges to the created roles
- Users are allocated to roles
  - GRANT role TO user1
  - Multiple individuals can be assigned to each role
  - Any individual assigned to a role automatically has the privileges associated with that role
- An individual can be assigned to multiple roles

Security Databases

- Ensuring security for large DBs is an important and difficult task
- Many different issues involved

  – legal, social, ethical etc.

- Most countries have General Data Protection Regulations (GDPR)

  – requires holders of personal information to take reasonable precautions to ensure that there is no unauthorised access to the data

Summary

- Integrity and Security are related but they are not the same thing
  - Integrity is concerned with *accidental* corruption
  - Security is concerned with *deliberate* corruption
- Integrity
  - Integrity Constraints
- Security
  - Privilege Granting and Revocation
  - Security Level Assignment

CSU34041

Database Security

Yvette Graham
ygraham@tcd.ie

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin