Fakultas Informatika
School of Computing
Telkom University

Database System

10 | Database Security

Tahun Ajar Ganjil 2024/2025

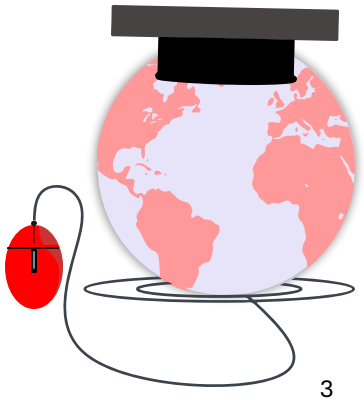
Oleh:
Tim Dosen

GOALS OF MEETING

1. Students knows various database issues and threats
2. Students understand how to securing databases
3. Students understand various access control policy

OUTLINES

- Introduction to Database Security
- Database Issues and Threats
- Control Measures
- Access Control
- Access Control Policy



INTRODUCTION TO DATABASE SECURITY

- WHAT IS DATABASE SECURITY
- IMPORTANCE DATA
- WHY DATABASE SECURITY IS NEEDED
- DATABASE ISSUES

WHAT IS DATABASE SECURITY

- Database security is the practice of keeping data/database protected from corruption and/or unauthorized access. The focus behind database security is to ensure priaccy while protecting personal or corporate data.
- It is a means of putting in place the different form of information security controls to protect database against compromise of their confidentiality, integrity, and availability
- Database security is protection of database data against accidental or intentional loss, destruction, or misuse.
- Database security is protection from malicious attempts to steal (view) or modify data.

IMPORTANCE DATA

- Bank accounts
- Credit card, salary, income tax data
- Resident Data
- University admissions, marks/grades

- Data ➔ crown jewels for organization
- Recent headlines:

- PDN Terkena ransom ware -> Data terkunci, harus bayar untuk bisa men-decript

WHY DATABASE SECURITY IS NEEDED

- The goal of database security is to protect data from accidental or intentional threats to their integrity and access.
- Database environments can grow more complex, with distributed databases residing on client/server architectures and personal computers as well as on mainframes.
- Access to data has become more open via the Internet and corporate intranets and from mobile computing devices.
- As a result, effectively managing data security has become more difficult and time-consuming.
- So organizations must be aware of security threats and take action to protect data.

DATABASE SECURITY ISSUES

Database Security is a broad ara that addresses many issues, including:

- **Legal and ethical issues**, regarding the right to access certain information
- **Policy issues**, what kinds of information should not be made publicly available (at the governmental, institutional, or corporate level regarding)
- **System levels issues**, which various security functions should be enforced (ex: should be handled at the physical hardware level, the operating system level, or the DBMS level)
- **Security Levels issues**, The need in some organizations to identify multiple security levels and to categorize the data and users based on these classifications (ex: top secret, secret, confidential, and unclassified)

THREATS TO DATABASES

- **Loss of Integrity.** Database integrity refers to the requirement that information be protected from improper modification. Modification of data includes creating, inserting, and updating data; changing the status of data; and deleting data. Integrity is lost if unauthorized changes are made to the data by either intentional or accidental acts.
- **Loss of availability.** Database availability refers to making objects available to a human user or a program who/which has a legitimate right to those data objects. Loss of availability occurs when the user or program cannot access these objects.

THREATS TO DATABASES (CONT.)

- **Loss of confidentiality**. Database confidentiality refers to the protection of data from unauthorized disclosure. Unauthorized, unanticipated, or unintentional disclosure could result in loss of public confidence, embarrassment, or legal action against the organization.
- SQL Injection.

CONTROL MEASURES

- DATABASE ENCRYPTION
- ACCESS CONTROL (DAC, MAC, RBAC)

CONTROL MEASURES

To protect databases against the threats discussed above, it is common to implement four kinds of control measures:

- access control,
- inference control,
- flow control, and
- encryption.

DATABASE ENCRYPTION

- E.g. What if a laptop/disk/USB key with critical data is lost?
- Partial solution: encrypt the database at storage level, transparent to application
  - Whole database/file/relation
    - Unit of encryption: page
  - Column encryption
  - Main issue: key management
    - E.g. user provides decryption key (password) when database is started up
  - Supported by many database systems
    - Standard practice now to encrypt credit card information, and other sensitive information

DATABASE ENCRYPTION ALGORITHM

Encryption consists of applying an **encryption algorithm to data using some prespecified encryption key.** The resulting data must be **decrypted** using a **decryption key** to recover the original data.

- Data Encryption and Advanced Encryption Standards
- Symmetric Key Algorithms
- Public (Asymmetric) Key Encription
- Digital Signatures
- Digital Certificates

ACCESS CONTROL

- Subject: active entity that requests access to an object
  - e.g., user, account or program
- Object: passive entity accessed by a subject
  - e.g., database, record, tuple, column, relation, file, view
- Access right (privileges): how a subject is allowed to access an object
  - e.g., subject *s* can read object *o*

RELATION-LEVEL GRANULARITY

[*Confidential*] Relation Employees

Name	Salary	Job-Performance
Smith	40000	Fair
Brown	80000	Good

TUPLE-LEVEL GRANULARITY

Relation Employees

Name	Salary	Job-Performance	
Smith	40000	Fair	[Confidential]
Brown	80000	Good	[Public]

ATTRIBUTE-LEVEL GRANULARITY

Relation Employees

Name [Public]	Salary [Secret]	Job-Performance [Confidential]
Smith	40000	Fair
Brown	80000	Good

CELL-LEVEL GRANULARITY

Relation Employees

Name	Salary	Job-Performance
Smith [P]	40000 [S]	Fair [C]
Brown [C]	80000 [C]	Good [S]

P : Public; C: Confidential; S: Secret

ACCESS CONTROL POLICY

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Role-Based Access Control (RBAC)

DISCRETIONARY ACCESS CONTROL (DAC)

- For ***each subject*** access right to the objects are defined
  - (subject, object, +/- access mode)
  - (Black, Employee-relation, read)
- User based
- Grant and Revoke
- Problems:
  - Propagation of access rights
  - Revocation of propagated access rights

ACCESS CONTROL MECHANISM

- Security through Views
- Stored Procedures
- Grant and Revoke

SECURITY THROUGH VIEWS

- Assign rights to access predefined views

```
CREATE VIEW Outstanding-Student
AS SELECT NAME, COURSE, GRADE
FROM Student
WHERE GRADE > B
```

Problem:

Difficult to maintain updates.

STORED PROCEDURES

- Assign rights to execute compiled programs

```
GRANT RUN ON <program> TO <user>
```

Problem:

Programs may access resources for which the user who runs the program does not have permission.

GRANT AND REVOKE

- **GRANT** <privilege> **ON** <relation> **TO** <user> **[WITH GRANT OPTION]**

```
GRANT INSERT ON Student TO Matthews

GRANT INSERT, UPDATE(GRADE) ON Student TO FARKAS

GRANT INSERT(NAME) ON Student TO Brown
```

Notes:

- When granting INSERT at the column level, you must include all the not null columns in the row
- GRANT command applies to base relations as well as views

GRANT AND REVOKE (CONT.)

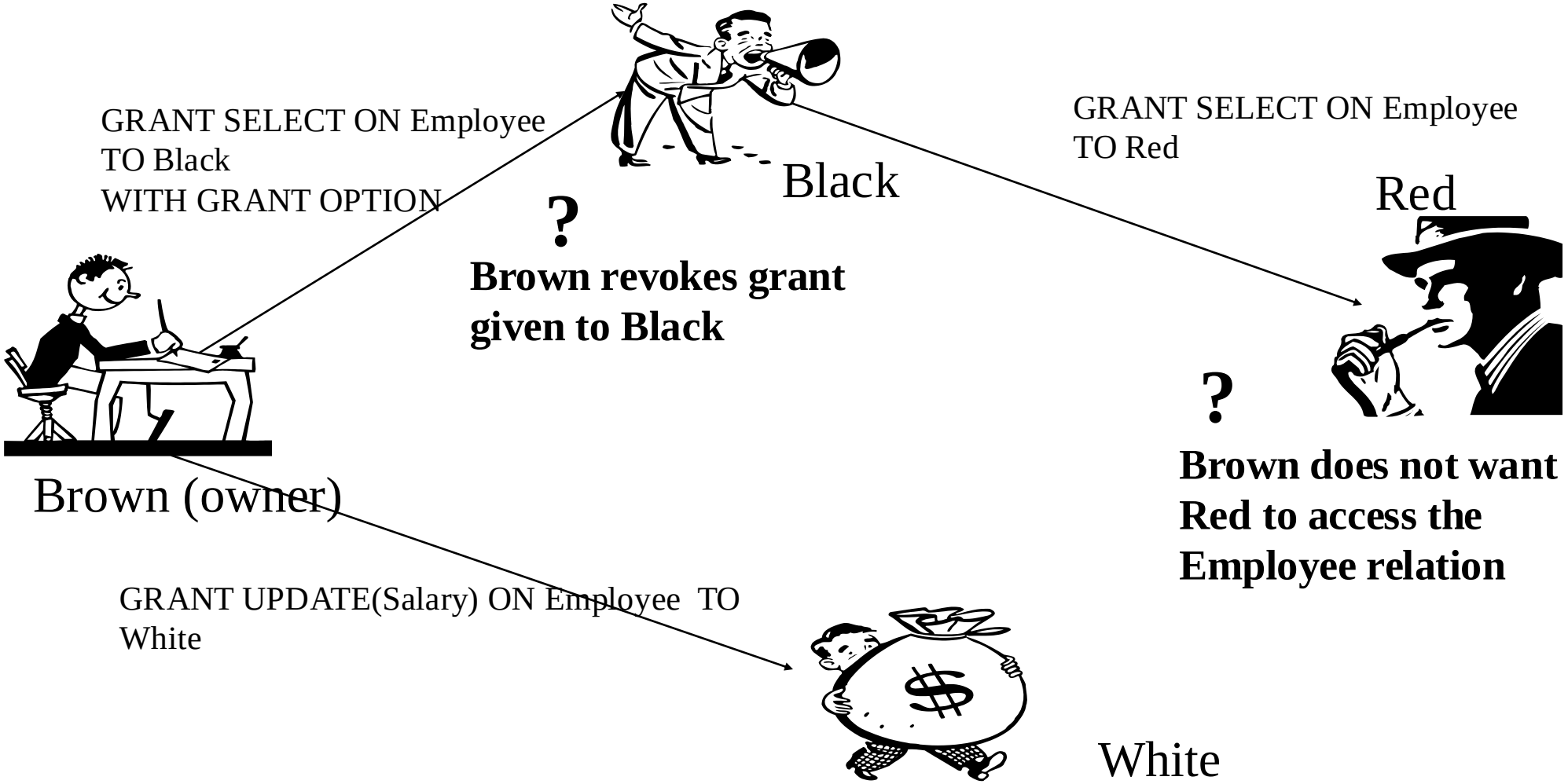
- **REVOKE** <privileges> **[ON <relation>] FROM** <user> [**restrict** | **cascade**]

```
REVOKE SELECT ON Student FROM Blue

REVOKE UPDATE ON Student FROM Black

REVOKE INSERT(NAME) ON Student FROM Brown
```

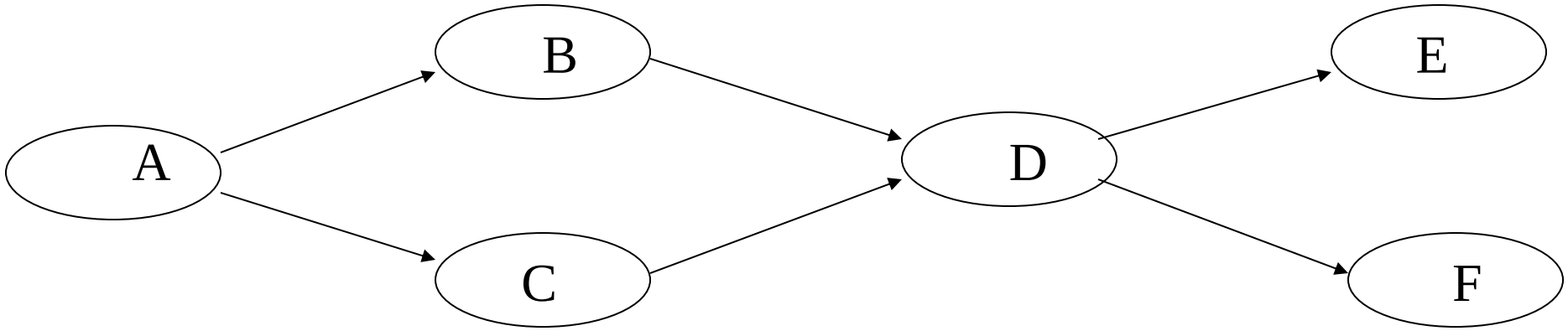
Notes:

- Revocation of a privilege from a user may cause other users also to lose that privilege; referred to as **cascading** of the revoke
- We can prevent cascading by specifying **restrict**
- With **restrict**, the **revoke** command fails if cascading revokes are required

DAC BY GRANT AND REVOKE



Cascading Revoke



A revokes D's privileges??
A revokes B's privileges??

MANDATORY ACCESS CONTROL (MAC)

- Security classes
  - Top-Secret, Secret, Confidential, Public
- Objects: security classification
  - File 1 is Secret, File 2 is Public
- Subjects: security clearances
  - Brown is cleared to Secret, Black is cleared to Public
- Dominance ($\geq$)
  - Top-Secret $\geq$ Secret $\geq$ Public

MAC – BELL-LAPADULA (BLP) MODEL

- Single security property: a subject S is allowed a read access to an object O only if label(S) dominates label(O)
- Star-property: a subject S is allowed a write access to an object O only if label(O) dominates label(S)

No direct flow of information from high security objects to low security objects!

MULTILEVEL SECURITY

- Multilevel security users at different security level, see different versions of the database
- Problem: different versions need to be kept consistent and coherent without downward signaling channel (covert channel)

MULTILEVEL RELATION

- Schema R(A1,C1,...,An,Cn,Tc)
  - R: relation name
  - Ai: attribute name
  - Ci: security classes
  - Tc: Tuple security classes
- Instantiation of relation: sets of tuples of the form <a1,c1,...,an,cn,tc>
  - ai: attribute value
  - ci: attribute classification label
  - tc: tuple classification label

MULTILEVEL RELATION EXAMPLE

SSN	λ(SSN)	Course	λ(Course)	Grade	λ(Grade)
111-22-3333	S	CSCE 786	S	A	TS
444-55-6666	S	CSCE 567	S	C	TS

Top-secret user sees all data

Secret user sees Secret-View:

SSN	λ(SSN)	Course	λ(Course)	Grade	λ(Grade)
111-22-3333	S	CSCE 786	S	null	S
444-55-6666	S	CSCE 567	S	null	S

ROLE BASED ACCESS CONTROL

- Roles permit common privileges for a class of users can be specified just once by creating a corresponding “role”
- Roles can be created using the CREATE ROLE and DESTROY ROLE commands.
- Privileges can be granted to or revoked from roles
- Roles can be assigned to users, and even to other roles
- RBAC can be used with traditional discretionary and mandatory access controls; it ensures that only authorized users in their specified roles are given access to certain data or resources.

GRANT AUTHORIZATION IN SQL

- **create role** teller
  create role manager

- **grant select on** branch **to** teller
  grant update (balance) **on** account **to** teller
  grant all privileges on account **to** manager

  grant teller **to** manager

  grant teller **to** alice, bob
  grant manager **to** avi

REVOKING AUTHORIZATION IN SQL

- <privilege-list> may be **all to** revoke all privileges the revokee may hold.
- If <revokee-list> includes **public** all users lose the privilege except those granted it explicitly.
- If the same privilege was granted twice to the same user by different grantees, the user may retain the privilege after the revocation.
- All privileges that depend on the privilege being revoked are also revoked.

EXERCISE #1

Eclaire as an owner of HR schema, want to give view and insert access of employees table to Froyo.

Froyo can also give view access of Employees table to Gingerbread.

Is It possible to do this??

Eclaire:

```
GRANT SELECT, INSERT
ON Employees
TO Froyo
WITH GRANT OPTION
```

Froyo:

```
GRANT SELECT ON Movies
TO Gingerbread
```

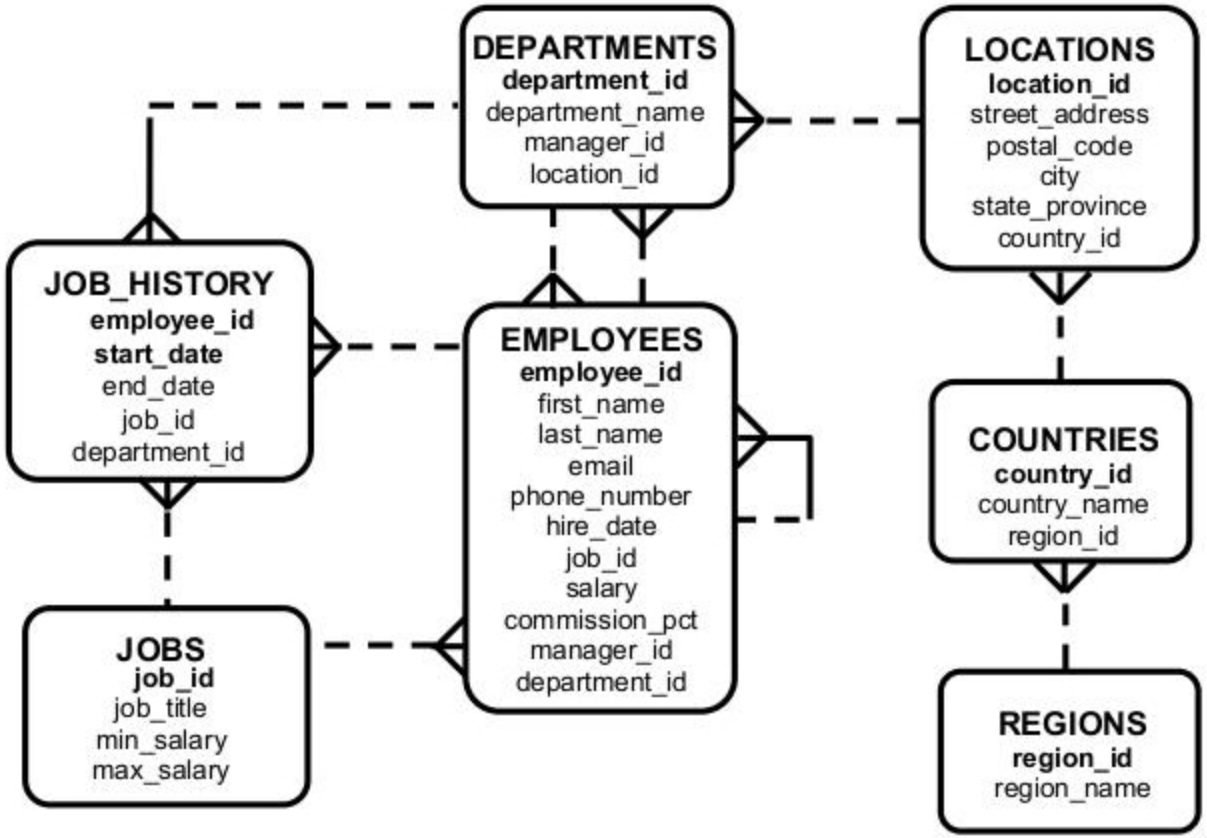
EXERCISE #2

Froyo also want to give update access of employees table to Gingerbread.

Is It possible to do this??

No. Because He doesn't have the access!!

EXERCISE #3



If Eclaire wants to give limited view access to user Honeycomb, specifically for this information: Employee name, department name, and job title which located in Brisbane

how to do it?

```
CREATE VIEW Employee_Data AS
       SELECT CONCATE(first_name,'
       ',last_name), department_name,
       job_title
       FROM employees JOIN jobs USING
       (job_id)
       JOIN departments USING
       (department_id)
       JOIN locations USING
       (location_id) WHERE
       city='Brisbane'
```

REFERENSI

- Database Security: Definition and Explanation, Access Control and Encryption - ClassNotes.ng
- Readings in Database Systems, Peter Bailis (2015); Elmasri, Navathe, “Fundamental of Database Systems”, Seventh Edition, Pearson, 2016,
- Hoffer, Jeffrey A., et.al., "Modern Database Management", Twelfth Edition, Pearson, 2016.

ANY QUESTIONS?