Database Security – Part 1

Introduction

Introduction

- **Why database security?**
  - Databases often store **data that are sensitive** in nature.
  - Databases need to preserve **data integrity**.
  - . . .

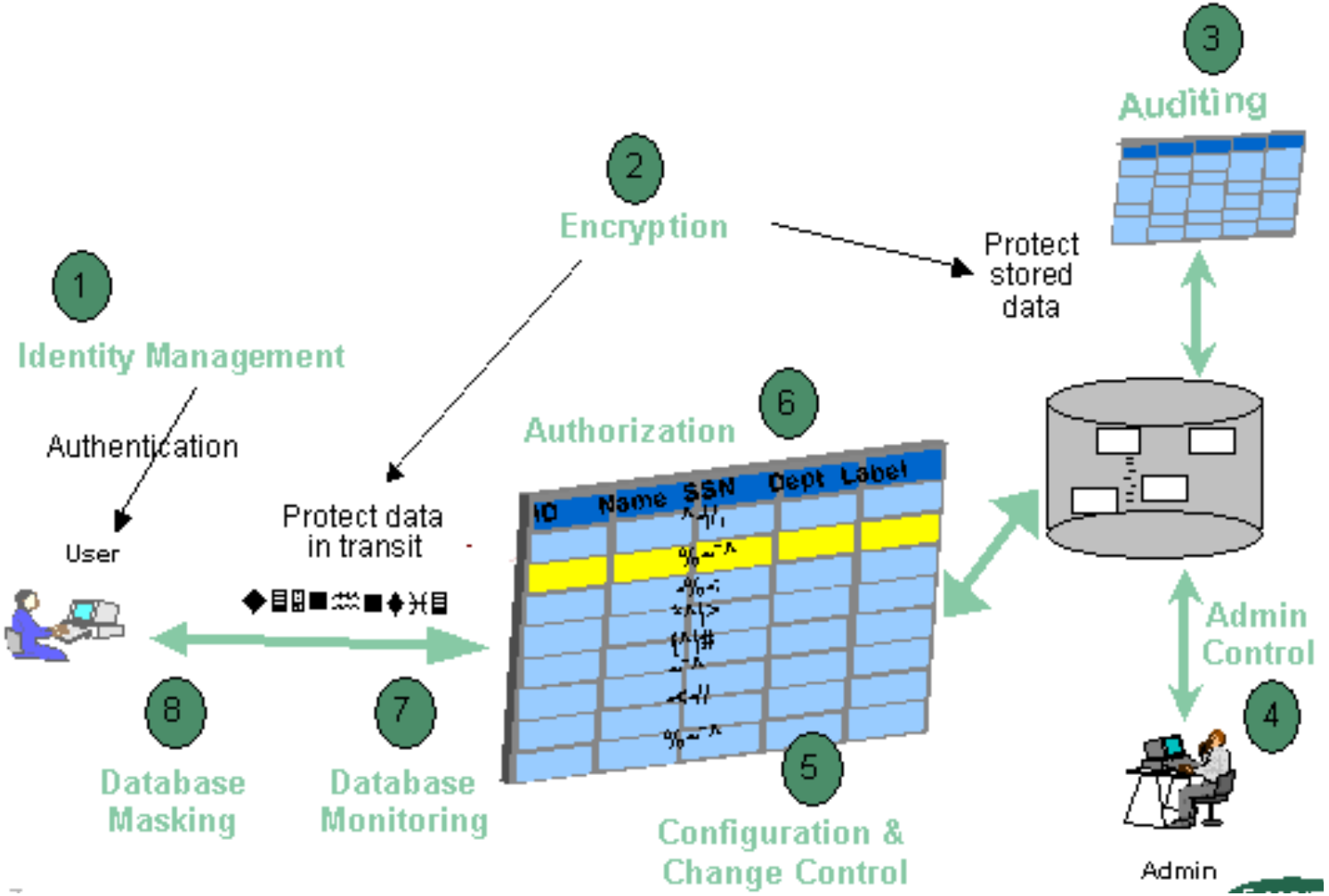
Example: Consider a payroll database, it must be ensured that:

- Salaries may not be disclosed to arbitrary users of the database;
- Salaries can only be modified by users that are properly authorized.

Introduction

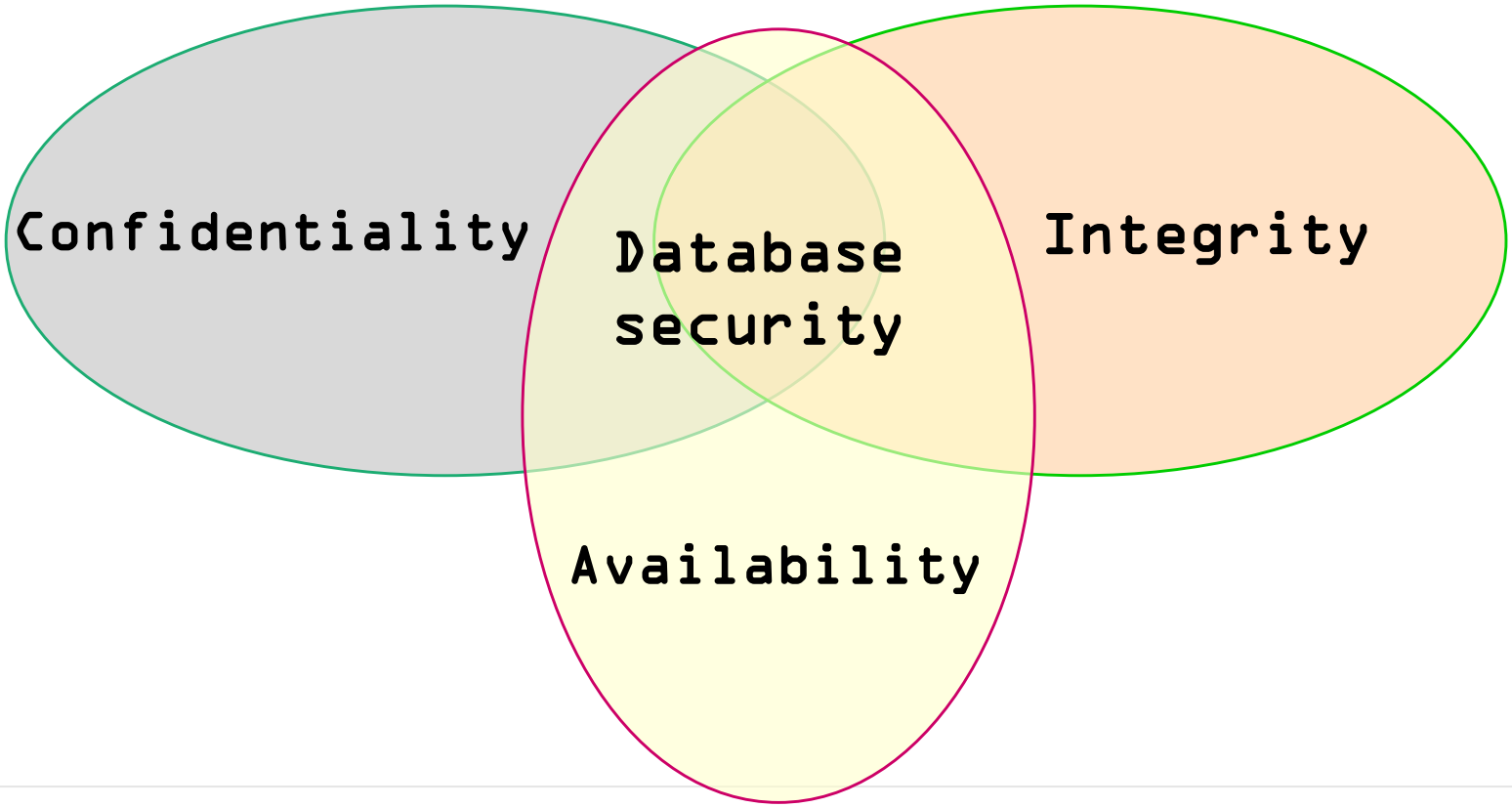
- The protection which database security gives is usually directed **against two cases**:
  - Stop users without database access from having any access;
  - Stop users with database access from performing actions on the database which are not required to perform their duties.

Comprehensive Database Security[1]



[1] Oracle database 11g security: data masking, Jonathan Penn, Forrester Research

Main Objectives of Database Security



Threats to Databases

- A potential **breach of security** that, if successful, will have a certain impact on databases.

Threats to Databases

- A potential **breach of security** that, if successful, will have a certain impact on databases.
  - **Loss of confidentiality**: data should not be accessible to those who do not have legitimate access rights,

    e.g., *a student is not allowed to view grades of other students.*

Threats to Databases

- A potential **breach of security** that, if successful, will have a certain impact on databases.
  - **Loss of confidentiality**: data should not be accessible to those who do not have legitimate access rights,

    e.g., *a student is not allowed to view grades of other students.*
  - **Loss of integrity**: data should not be corrupted, through intentional or accidental acts,

    e.g., *students are allowed to see their grades, yet not allowed (obviously) to modify them.*

Threats to Databases

- A potential **breach of security** that, if successful, will have a certain impact on databases.
  - **Loss of confidentiality**: data should not be accessible to those who do not have legitimate access rights,

    e.g., *a student is not allowed to view grades of other students.*
  - **Loss of integrity**: data should not be corrupted, through intentional or accidental acts,

    e.g., *students are allowed to see their grades, yet not allowed (obviously) to modify them.*
  - **Loss of availability**: data should remain accessible to those who have legitimate access rights,

    e.g., *a lecturer is allowed to change grades of students.*

Control Measures

1. **Access control**
   - Restrict access to the database system,

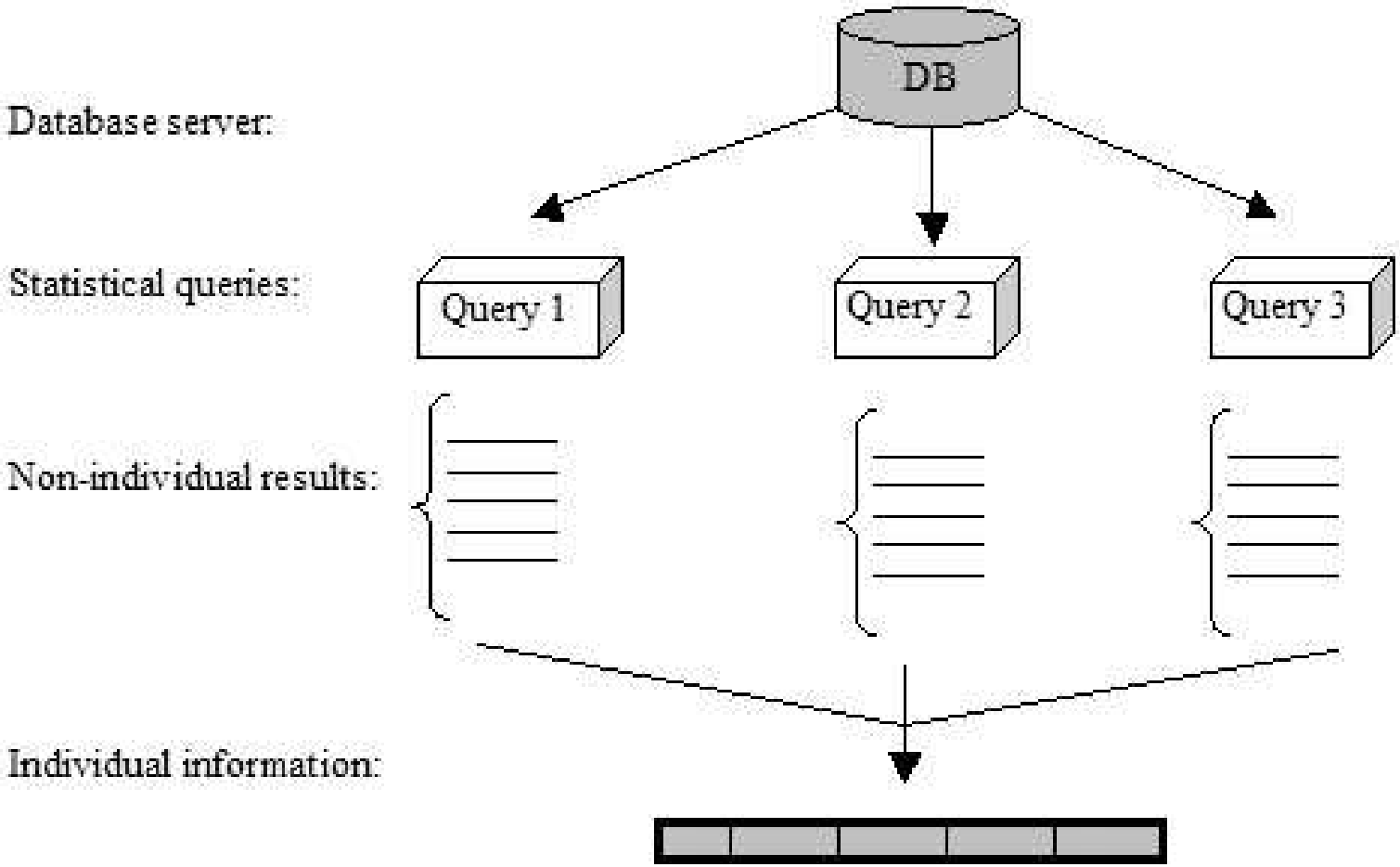
     e.g., *user accounts and passwords*.
2. **Inference control**
   - Ensure that data that users are not authorized to access cannot be inferred from statistical or summary data,

     e.g., *know the average salary of a department, but don't know the salary of a particular person*.
3. **Flow control**
   - Prevent data to flow into unauthorized users,

     e.g., *avoid covert channels*.
4. **Data encryption**
   - Protect sensitive data during storage and transmission,

     e.g., *passwords and credit card information*.

Inference Attack[2]



[2] Inference Attacks to Statistical Databases: Data Suppression, Concealing Controls and Other Security Trends, 2000

Inference Attack - Examples

- An extensive data re-identification experiment run in 1990 by the United States Government:

  87% of 248 million US citizens could be uniquely identified based on the combination of gender, date of birth and a five-digit ZIP code.

Inference Attack - Examples

- Suppose that we have a database which contains information of employees, including names, ages and salaries, and only allows aggregation queries. If we happen to know that Peter is the oldest employee in the company, can we infer the salary of Peter through aggregation queries?

Inference Attack - Examples

- Suppose that we have a database which contains information of employees, including names, ages and salaries, and only allows aggregation queries. If we happen to know that Peter is the oldest employee in the company, can we infer the salary of Peter through aggregation queries?

  (1) We could repeatedly ask: "*How many employees are there whose age is greater than X?*" until the answer is 1

  (2) Then we could ask: "*what is the average salary of all employees whose age is greater than X?*".