

# Information Security (CS3002)

**Instructor:** Dr. Muhammad Usama

**Email:** [usama.khanzada@nu.edu.pk](mailto:usama.khanzada@nu.edu.pk)

**Book:** Computer Security - Principles and Practice (Chapter 5)

Database and Data Center  
Security

# Database Security

The increasing reliance on cloud technology to host part or all of the corporate database

Most enterprise environments consist of a heterogeneous mixture of database platforms, enterprise platforms, and OS platforms, creating an additional complexity hurdle for security personnel

There is a dramatic imbalance between the complexity of modern database management systems (DBMS) and the security technique used to protect these critical systems

Reasons database security has not kept pace with the increased reliance on databases are:

The typical organization lacks full-time database security personnel

Databases have a sophisticated interaction protocol, Structured Query Language (SQL), which is complex

Effective database security requires a strategy based on a full understanding of the security vulnerabilities of SQL

# Databases

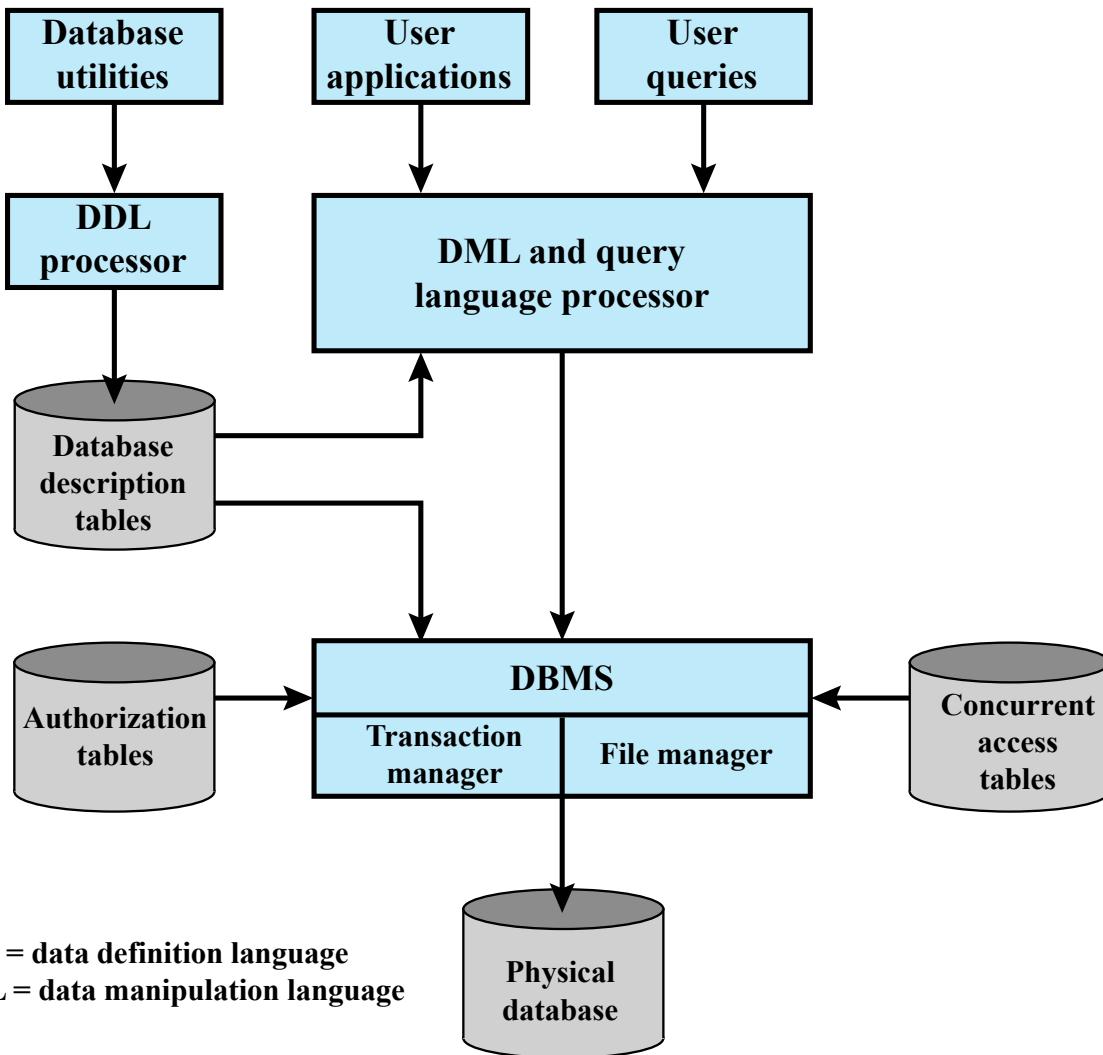
- Structured collection of data stored for use by one or more applications
- Contains the relationships between data items and groups of data items
- Can sometimes contain sensitive data that needs to be secured

## Query language

- Provides a uniform interface to the database for users and applications

Database management system (DBMS)

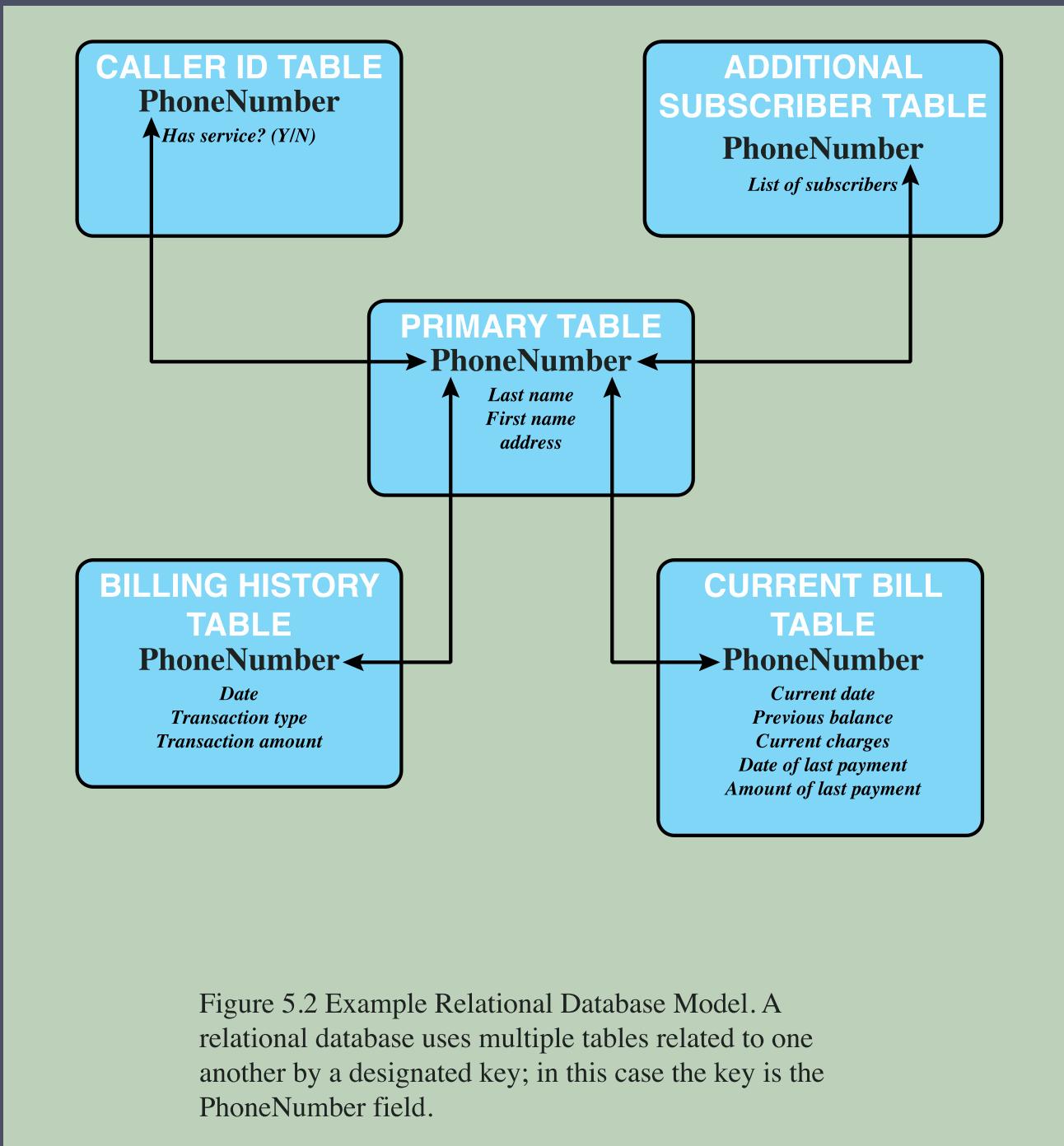
- Suite of programs for constructing and maintaining the database
- Offers ad hoc query facilities to multiple users and applications



**Figure 5.1 DBMS Architecture**

# Relational Databases

- Table of data consisting of rows and columns
  - Each column holds a particular type of data
  - Each row contains a specific value for each column
  - Ideally has one column where all values are unique, forming an identifier/key for that row
- Enables the creation of multiple tables linked together by a unique identifier that is present in all tables
- Use a relational query language to access the database
  - Allows the user to request data that fit a given set of criteria



# Relational Database Elements

- Relation
  - Table/file
- Tuple
  - Row/record
- Attribute
  - Column/field

## Primary key

- Uniquely identifies a row
- Consists of one or more column names

## Foreign key

- Links one table to attributes in another

## View/virtual table

- Result of a query that returns selected rows and columns from one or more tables
- Views are often used for security purposes

# Table 5.1

## Basic Terminology for Relational Databases

<b>Formal Name</b>	<b>Common Name</b>	<b>Also Known As</b>
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

		Attributes								
		$A_1$	•	•	•	$A_j$	•	•	•	$A_M$
Records	1	$x_{11}$	•	•	•	$x_{1j}$	•	•	•	$x_{1M}$
	•	•				•			•	
	•	•				•			•	
	•	•				•			•	
	$i$	$x_{i1}$	•	•	•	$x_{ij}$	•	•	•	$x_{iM}$
	•	•				•			•	
	•	•				•			•	
	•	•				•			•	
	$N$	$x_{N1}$	•	•	•	$x_{Nj}$	•	•	•	$x_{NM}$

**Figure 5.3 Abstract Model of a Relational Database**

**Department Table**

<b>Did</b>	<b>Dname</b>	<b>Dacctno</b>
4	human resources	528221
8	education	202035
9	accounts	709257
13	public relations	755827
15	services	223945

primary key

**Employee Table**

<b>Ename</b>	<b>Did</b>	<b>Salarycode</b>	<b>Eid</b>	<b>Ephone</b>
Robin	15	23	2345	6127092485
Neil	13	12	5088	6127092246
Jasmine	4	26	7712	6127099348
Cody	15	22	9664	6127093148
Holly	8	23	3054	6127092729
Robin	8	24	2976	6127091945
Smith	9	21	4490	6127099380

foreign key

primary key

(a) Two tables in a relational database

<b>Dname</b>	<b>Ename</b>	<b>Eid</b>	<b>Ephone</b>
human resources	Jasmine	7712	6127099348
education	Holly	3054	6127092729
education	Robin	2976	6127091945
accounts	Smith	4490	6127099380
public relations	Neil	5088	6127092246
services	Robin	2345	6127092485
services	Cody	9664	6127093148

(b) A view derived from the database

**Figure 5.4 Relational Database Example**

# Structured Query Language (SQL)

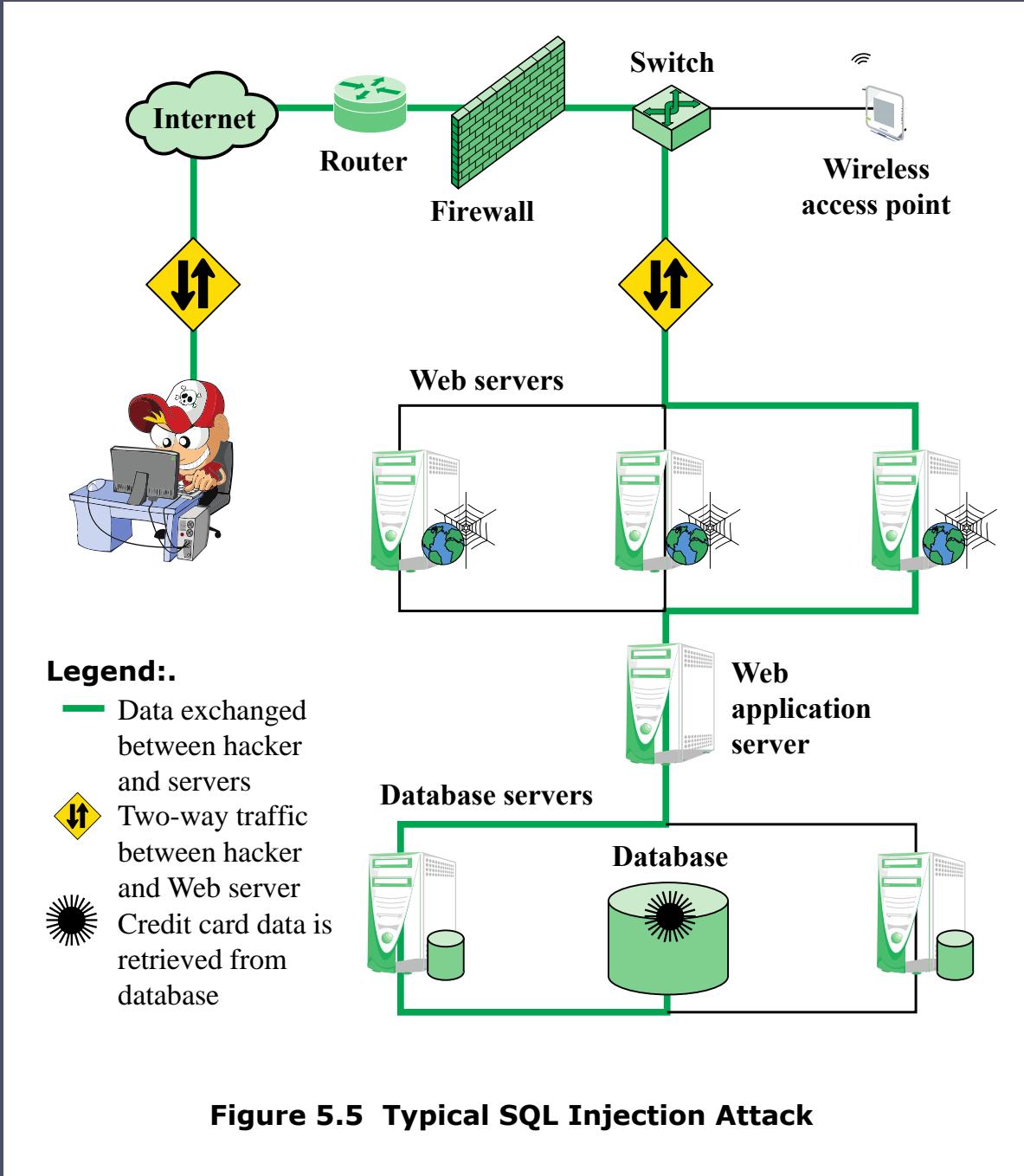
- Standardized language to define schema, manipulate, and query data in a relational database
- Several similar versions of ANSI/ISO standard
- All follow the same basic syntax and semantics

**SQL statements can be used to:**

- Create tables
- Insert and delete data in tables
- Create views
- Retrieve data with query statements

# SQL Injection Attacks (SQLi)

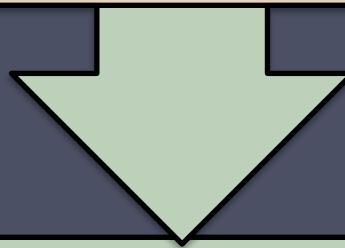
- One of the most prevalent and dangerous network-based security threats
- Designed to exploit the nature of Web application pages
- Sends malicious SQL commands to the database server
  - Most common attack goal is bulk extraction of data
  - Depending on the environment SQL injection can also be exploited to:
    - Modify or delete data
    - Execute arbitrary operating system commands
    - Launch denial-of-service (Dos) attacks



# Injection Technique

**The SQLi attack typically works by prematurely terminating a text string and appending a new command**

Because the inserted command may have additional strings appended to it before it is executed the attacker terminates the injected string with a comment mark “- -”



**Subsequent text is ignored at execution time**

# SQLi Attack Avenues

## User input

- Attackers inject SQL commands by providing suitable crafted user input

## Server variables

- Attackers can forge the values that are placed in HTTP and network headers and exploit this vulnerability by placing data directly into the headers

## Second-order injection

- A malicious user could rely on data already present in the system or database to trigger an SQL injection attack, so when the attack occurs, the input that modifies the query to cause an attack does not come from the user, but from within the system itself

## Cookies

- An attacker could alter cookies such that when the application server builds an SQL query based on the cookie's content, the structure and function of the query is modified

## Physical user input

- Applying user input that constructs an attack outside the realm of web requests

# Inband Attacks

- Uses the same communication channel for injecting SQL code and retrieving results
- The retrieved data are presented directly in application Web page
- Include:

## Tautology

This form of attack injects code in one or more conditional statements so that they always evaluate to true

## End-of-line comment

After injecting code into a particular field, legitimate code that follows are nullified through usage of end of line comments

## Piggybacked queries

The attacker adds additional queries beyond the intended query, piggy-backing the attack on top of a legitimate request

# Inferential Attack

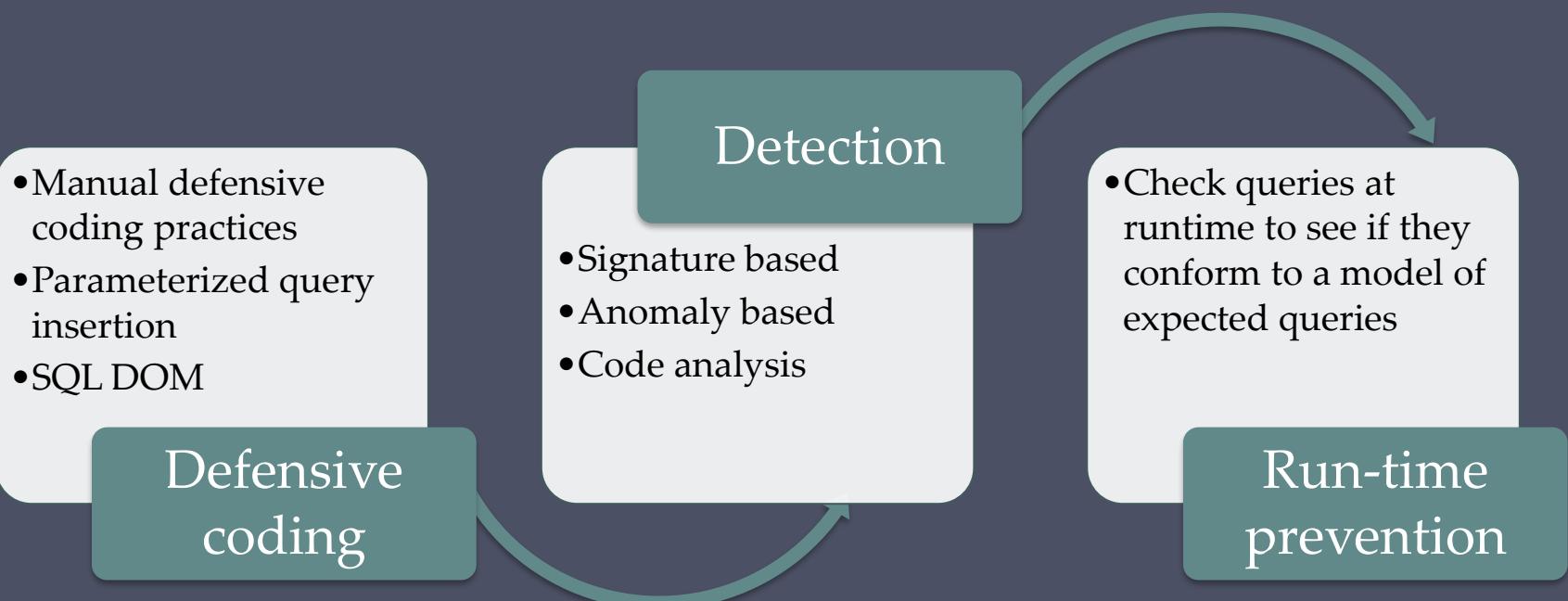
- There is no actual transfer of data, but the attacker is able to reconstruct the information by sending particular requests and observing the resulting behavior of the Website/database server
- Include:
  - Illegal/logically incorrect queries
    - This attack lets an attacker gather important information about the type and structure of the backend database of a Web application
    - The attack is considered a preliminary, information-gathering step for other attacks
  - Blind SQL injection
    - Allows attackers to infer the data present in a database system even when the system is sufficiently secure to not display any erroneous information back to the attacker

# Out-of-Band Attack

- Data are retrieved using a different channel
- This can be used when there are limitations on information retrieval, but outbound connectivity from the database server is lax

# SQLi Countermeasures

- Three types:



# Database Access Control

**Database access control system determines:**

If the user has access to the entire database or just portions of it

What access rights the user has (create, insert, delete, update, read, write)

**Can support a range of administrative policies**

**Centralized administration**

- Small number of privileged users may grant and revoke access rights

**Ownership-based administration**

- The creator of a table may grant and revoke access rights to the table

**Decentralized administration**

- The owner of the table may grant and revoke authorization rights to other users, allowing them to grant and revoke access rights to the table

# SQL Access Controls

- Two commands for managing access rights:
  - Grant
    - Used to grant one or more access rights or can be used to assign a user to a role
  - Revoke
    - Revokes the access rights
- Typical access rights are:
  - Select
  - Insert
  - Update
  - Delete
  - References

# Cascading Authorizations

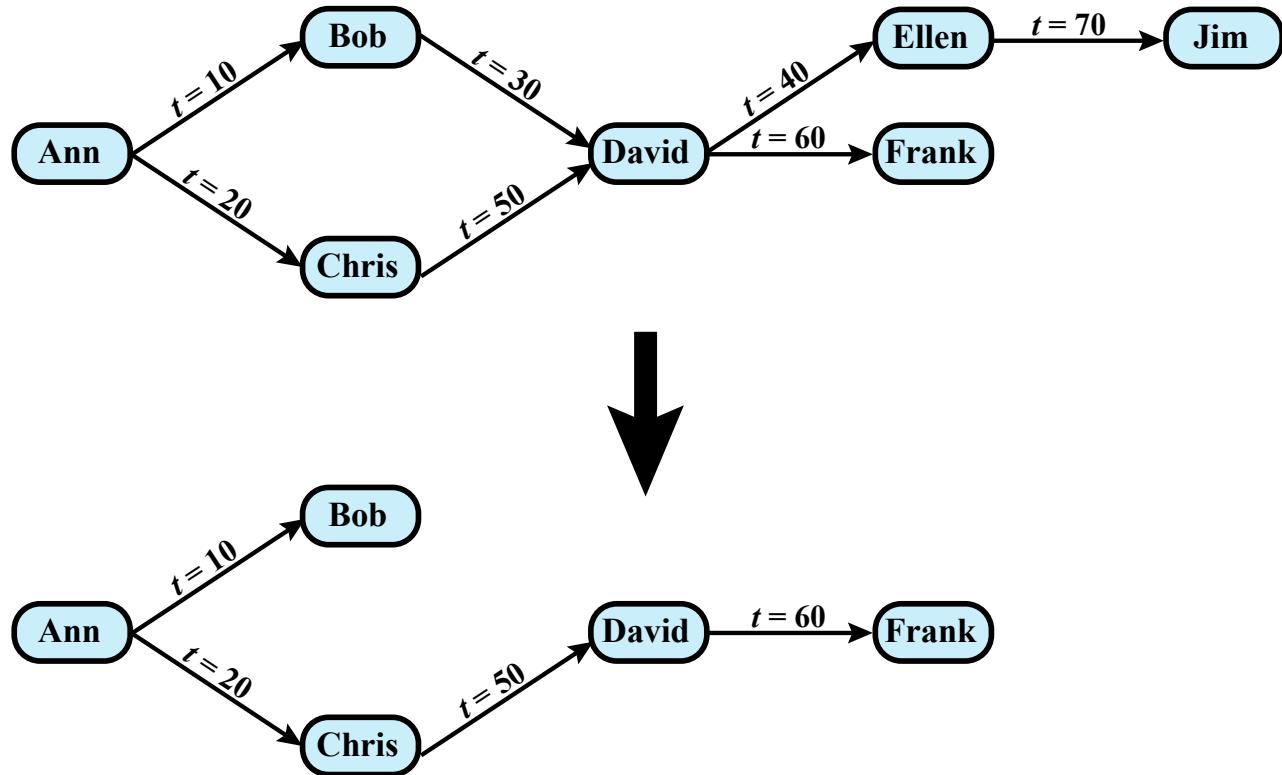


Figure 5.6 Bob Revokes Privilege from David

# Role-Based Access Control (RBAC)

- Role-based access control eases administrative burden and improves security
- A database RBAC needs to provide the following capabilities:
  - Create and delete roles
  - Define permissions for a role
  - Assign and cancel assignment of users to roles
- Categories of database users:

## Application owner

- An end user who owns database objects as part of an application

## End user

- An end user who operates on database objects via a particular application but does not own any of the database objects

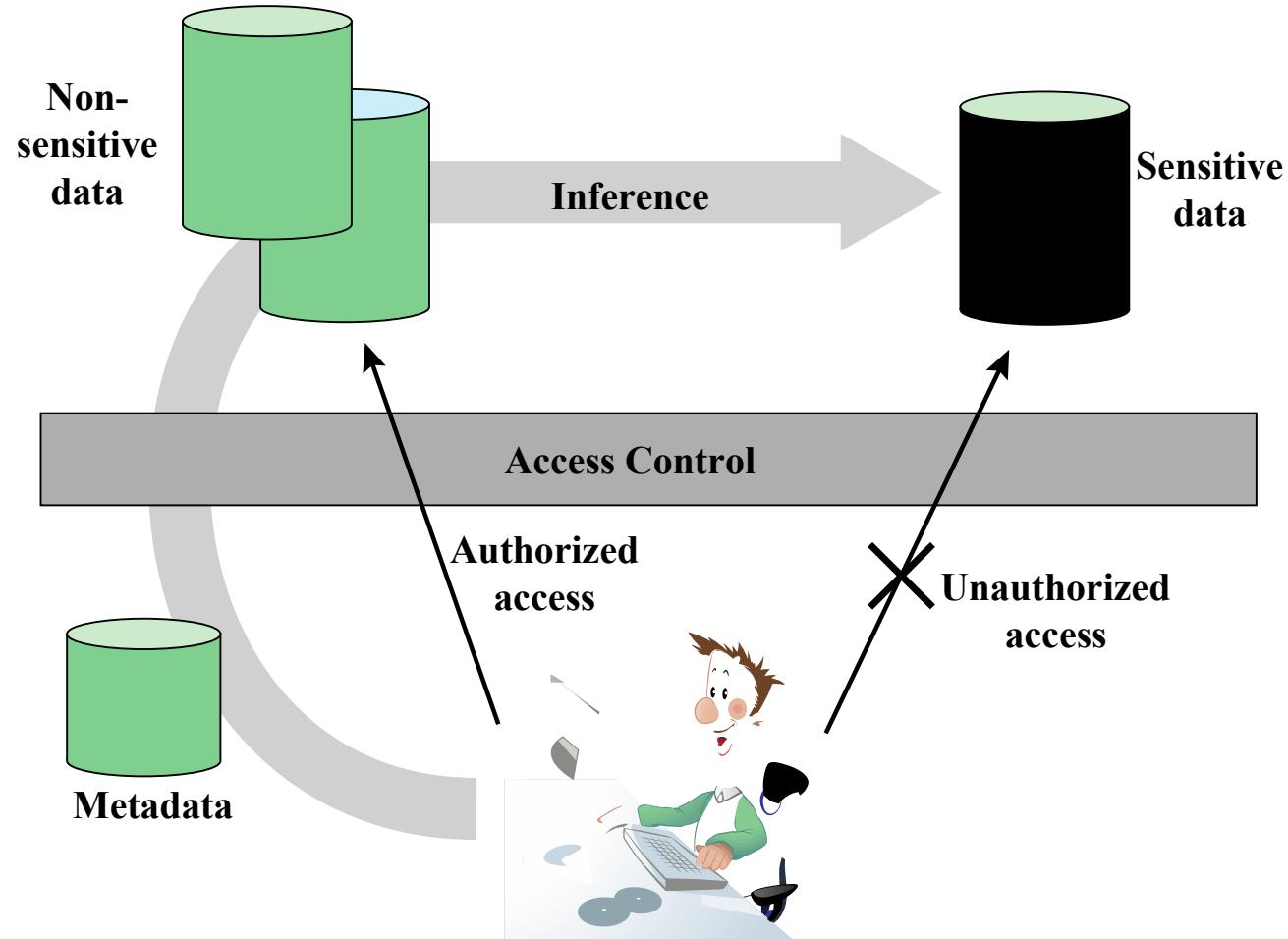
## Administrator

- User who has administrative responsibility for part or all of the database

# Table 5.2

## Fixed Roles in Microsoft SQL Server

Role	Permissions
<b>Fixed Server Roles</b>	
sysadmin	Can perform any activity in SQL Server and have complete control over all database functions
serveradmin	Can set server-wide configuration options, shut down the server
setupadmin	Can manage linked servers and startup procedures
securityadmin	Can manage logins and CREATE DATABASE permissions, also read error logs and change passwords
processadmin	Can manage processes running in SQL Server
dbcreator	Can create, alter, and drop databases
diskadmin	Can manage disk files
bulkadmin	Can execute BULK INSERT statements
<b>Fixed Database Roles</b>	
db_owner	Has all permissions in the database
db_accessadmin	Can add or remove user IDs
db_datareader	Can select all data from any user table in the database
db_datawriter	Can modify any data in any user table in the database
db_ddladmin	Can issue all Data Definition Language (DDL) statements
db_securityadmin	Can manage all permissions, object ownerships, roles and role memberships
db_backupoperator	Can issue DBCC, CHECKPOINT, and BACKUP statements
db_denydatareader	Can deny permission to select data in the database
db_denydatawriter	Can deny permission to change data in the database



**Figure 5.7 Indirect Information Access Via Inference Channel**

Item	Availability	Cost (\$)	Department
Shelf support	in-store/online	7.99	hardware
Lid support	online only	5.49	hardware
Decorative chain	in-store/online	104.99	hardware
Cake pan	online only	12.99	housewares
Shower/tub cleaner	in-store/online	11.99	housewares
Rolling pin	in-store/online	10.99	housewares

(a) Inventory table

Availability	Cost (\$)
in-store/online	7.99
online only	5.49
in-store/online	104.99

Item	Department
Shelf support	hardware
Lid support	hardware
Decorative chain	hardware

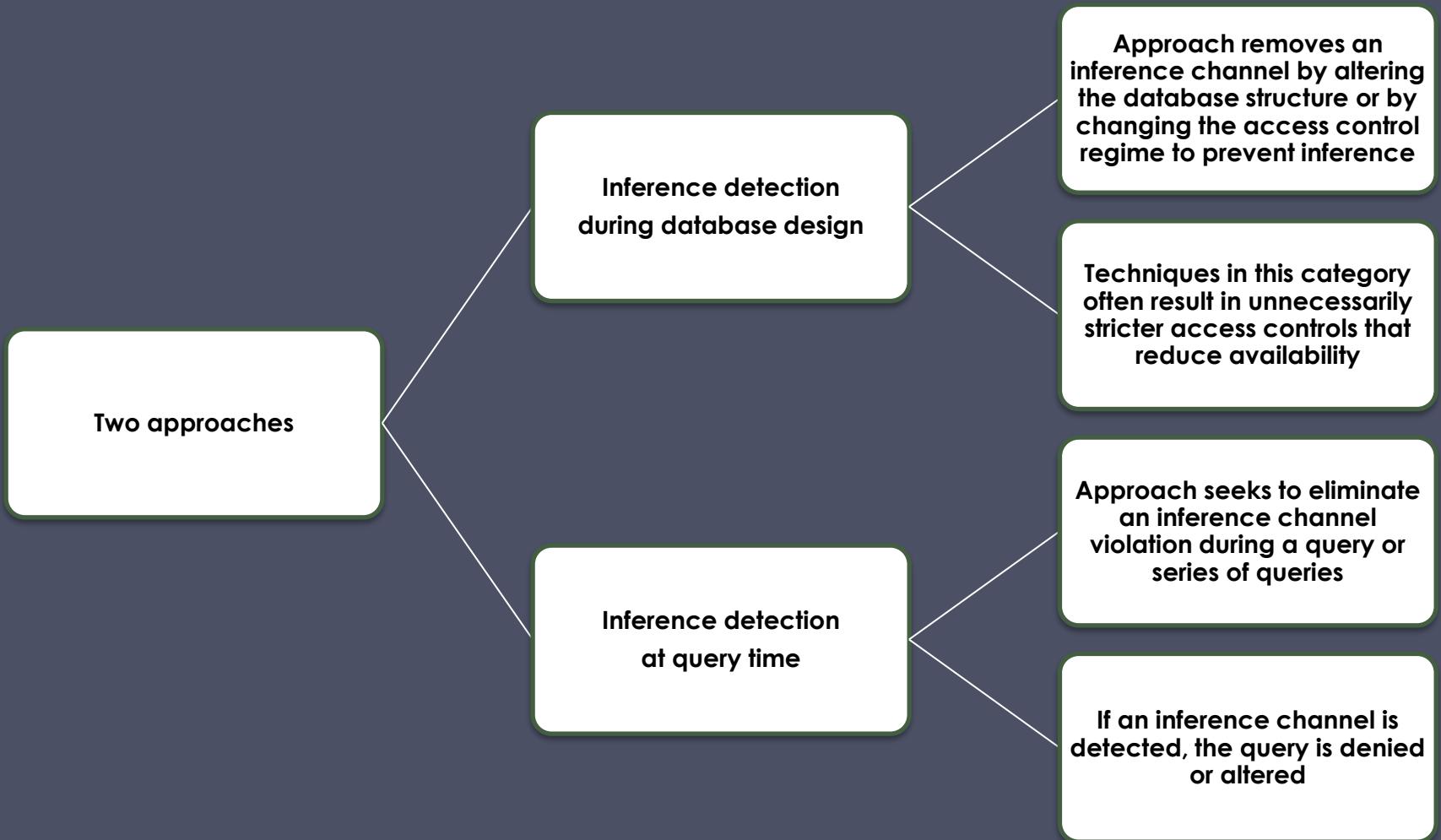
(b) Two views

Item	Availability	Cost (\$)	Department
Shelf support	in-store/online	7.99	hardware
Lid support	online only	5.49	hardware
Decorative chain	in-store/online	104.99	hardware

(c) Table derived from combining query answers

**Figure 5.8 Inference Example**

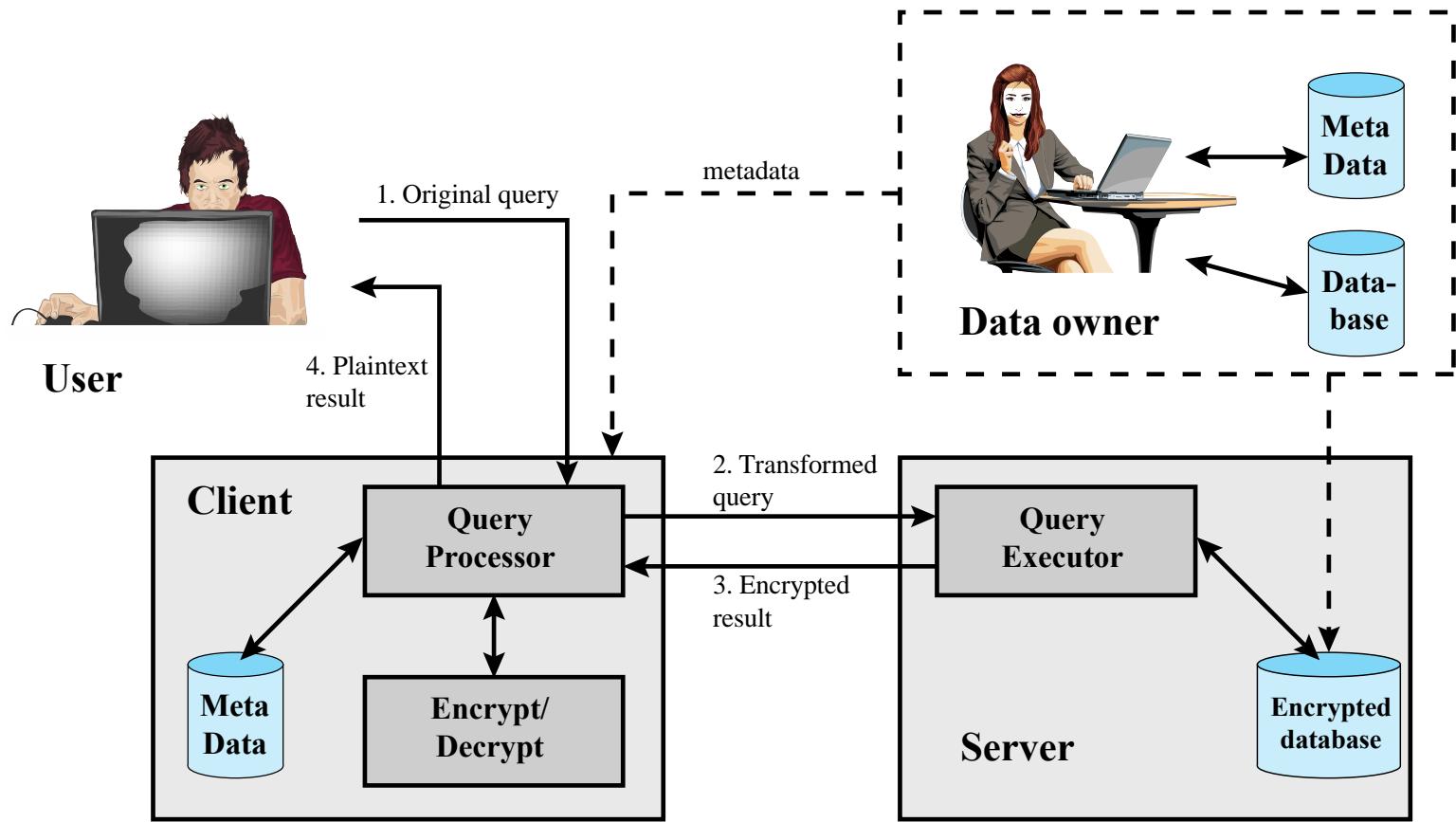
# Inference Detection



- Some inference detection algorithm is needed for either of these approaches
- Progress has been made in devising specific inference detection techniques for multilevel secure databases and statistical databases

# Database Encryption

- The database is typically the most valuable information resource for any organization
  - Protected by multiple layers of security
    - Firewalls, authentication, general access control systems, DB access control systems, database encryption
    - Encryption becomes the last line of defense in database security
  - Can be applied to the entire database, at the record level, the attribute level, or level of the individual field
- Disadvantages to encryption:
  - Key management
    - Authorized users must have access to the decryption key for the data for which they have access
  - Inflexibility
    - When part or all of the database is encrypted it becomes more difficult to perform record searching



**Figure 5.9 A Database Encryption Scheme**

$E(k, B_1)$	$I_{11}$	$\bullet \bullet \bullet$	$I_{1j}$	$\bullet \bullet \bullet$	$I_{1M}$
•	•		•		•
•	•		•		•
•	•		•		•
$E(k, B_i)$	$I_{i1}$	$\bullet \bullet \bullet$	$I_{ij}$	$\bullet \bullet \bullet$	$I_{iM}$
•	•		•		•
•	•		•		•
•	•		•		•
$E(k, B_N)$	$I_{N1}$	$\bullet \bullet \bullet$	$I_{Nj}$	$\bullet \bullet \bullet$	$I_{NM}$

$$B_i = (x_{i1} \| x_{i2} \| \dots \| x_{iM})$$

**Figure 5.10 Encryption Scheme for Database of Figure 5.3**

**Table 5.3 Encrypted Database Example**

**(a) Employee Table**

<b>eid</b>	<b>ename</b>	<b>salary</b>	<b>addr</b>	<b>did</b>
23	Tom	70K	Maple	45
860	Mary	60K	Main	83
320	John	50K	River	50
875	Jerry	55K	Hopewell	92

**(b) Encrypted Employee Table with Indexes**

<b>E(<math>k, B</math>)</b>	<b>I(eid)</b>	<b>I(ename)</b>	<b>I(salary)</b>	<b>I(addr)</b>	<b>I(did)</b>
1100110011001011...	1	10	3	7	4
0111000111001010...	5	7	2	7	8
1100010010001101...	2	5	1	9	5
0011010011111101...	5	5	2	4	9

# Summary

- The need for database security
- Database management systems
- Relational databases
  - Elements of a relational database system
  - Structured Query Language
- SQL injection attacks
  - A typical SQLi attack
  - The injection technique
  - SQLi attack avenues and types
  - SQLi countermeasures
- Database access control
  - SQL-based access definition
  - Cascading authorizations
  - Role-based access control
- Inference
- Database encryption
- Data center security
  - Data center elements
  - Data center security considerations
  - TIA-492