Computer Security: Principles and Practice

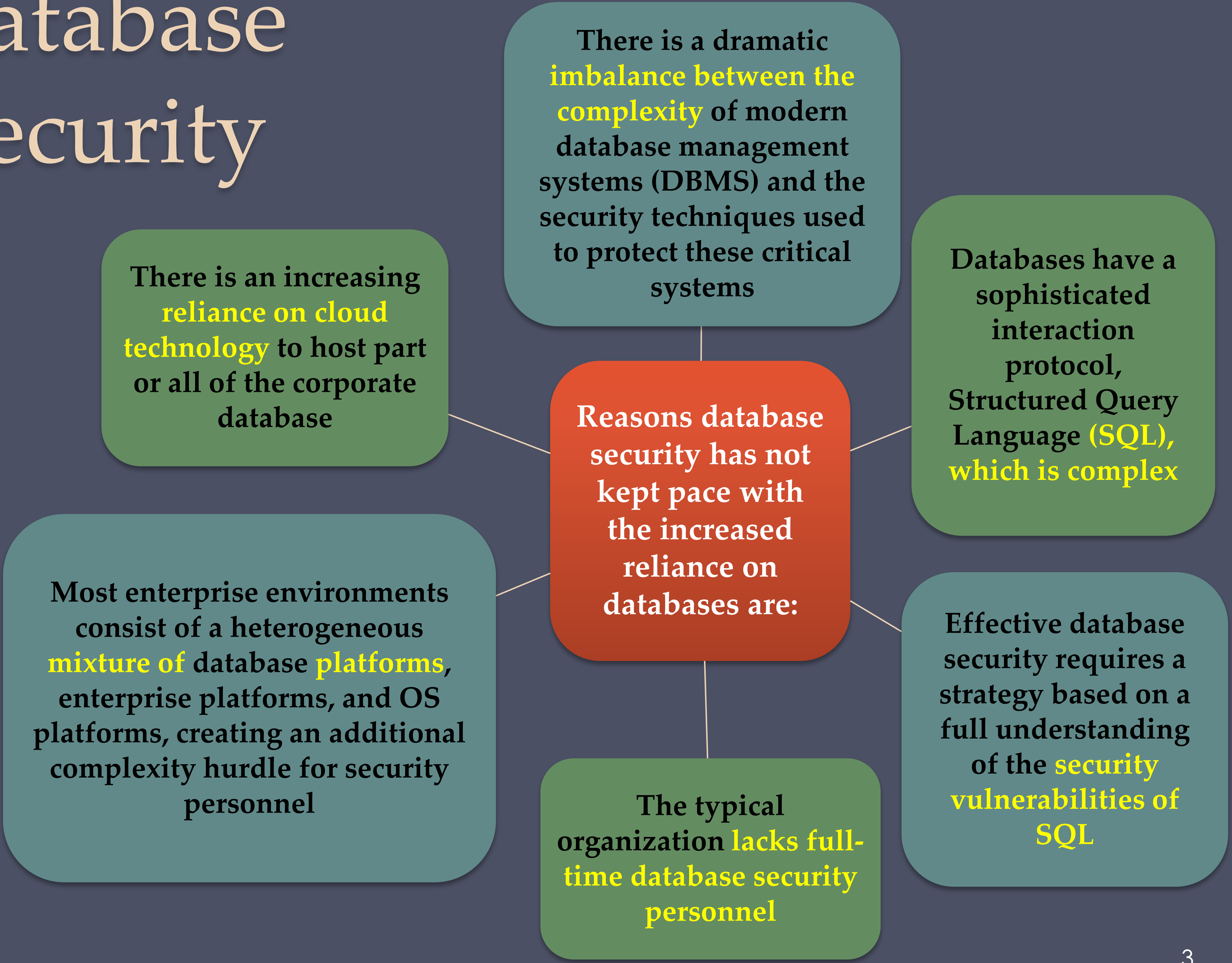
Fourth Edition

By: William Stallings and Lawrie Brown

Chapter 5

Database and Data Center Security

Database Security



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

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

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

The typical organization lacks full-time database security personnel

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

- The next few slides (up to slide 13) contain background on databases.
- Will not cover it in detail in class
- Students need to know the background

Databases

- Structured collection of data stored for use by one or more applications

- Contains the relationships between data items and groups of data items

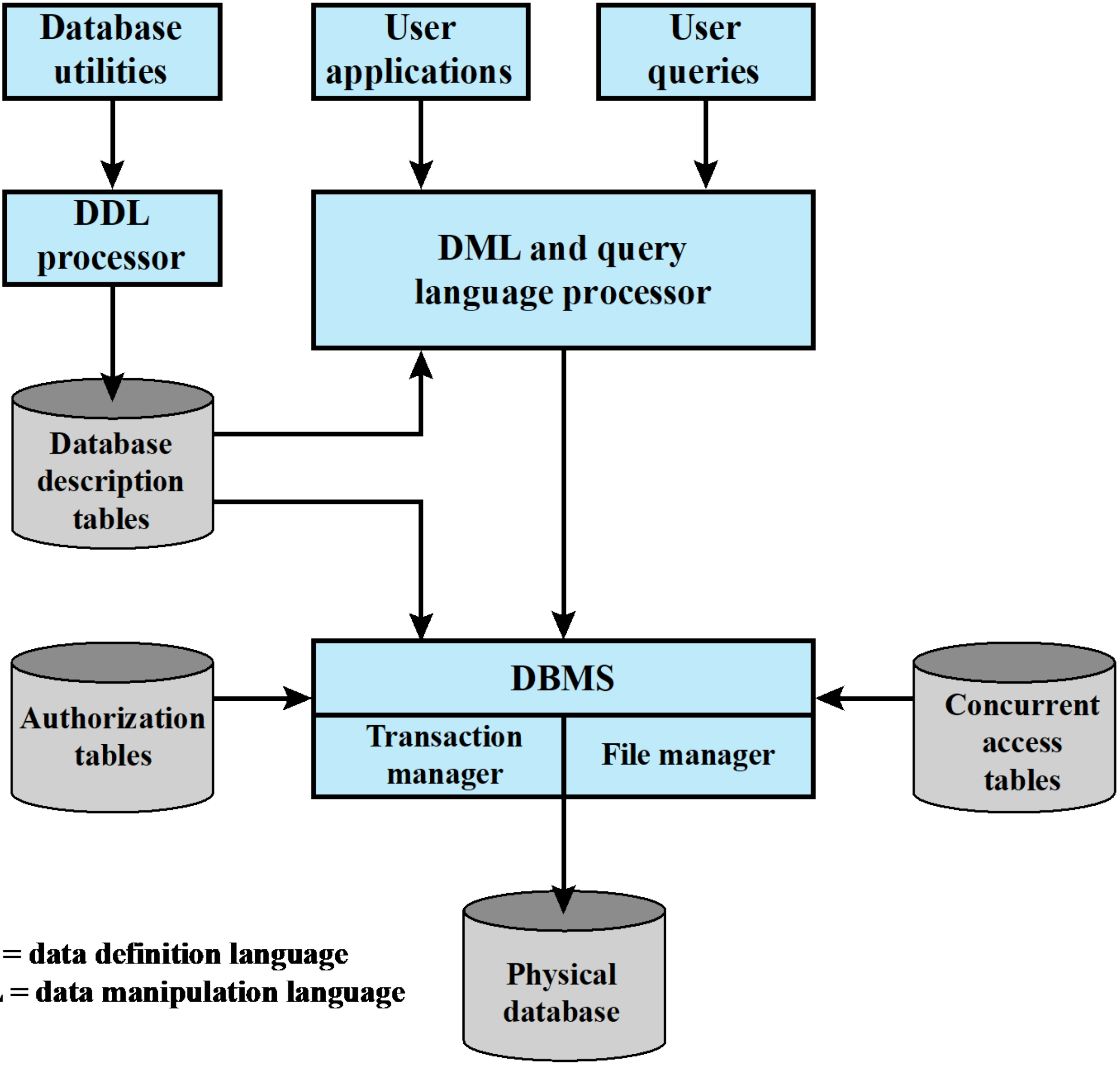
- Can sometimes contain sensitive data that needs to be secured

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

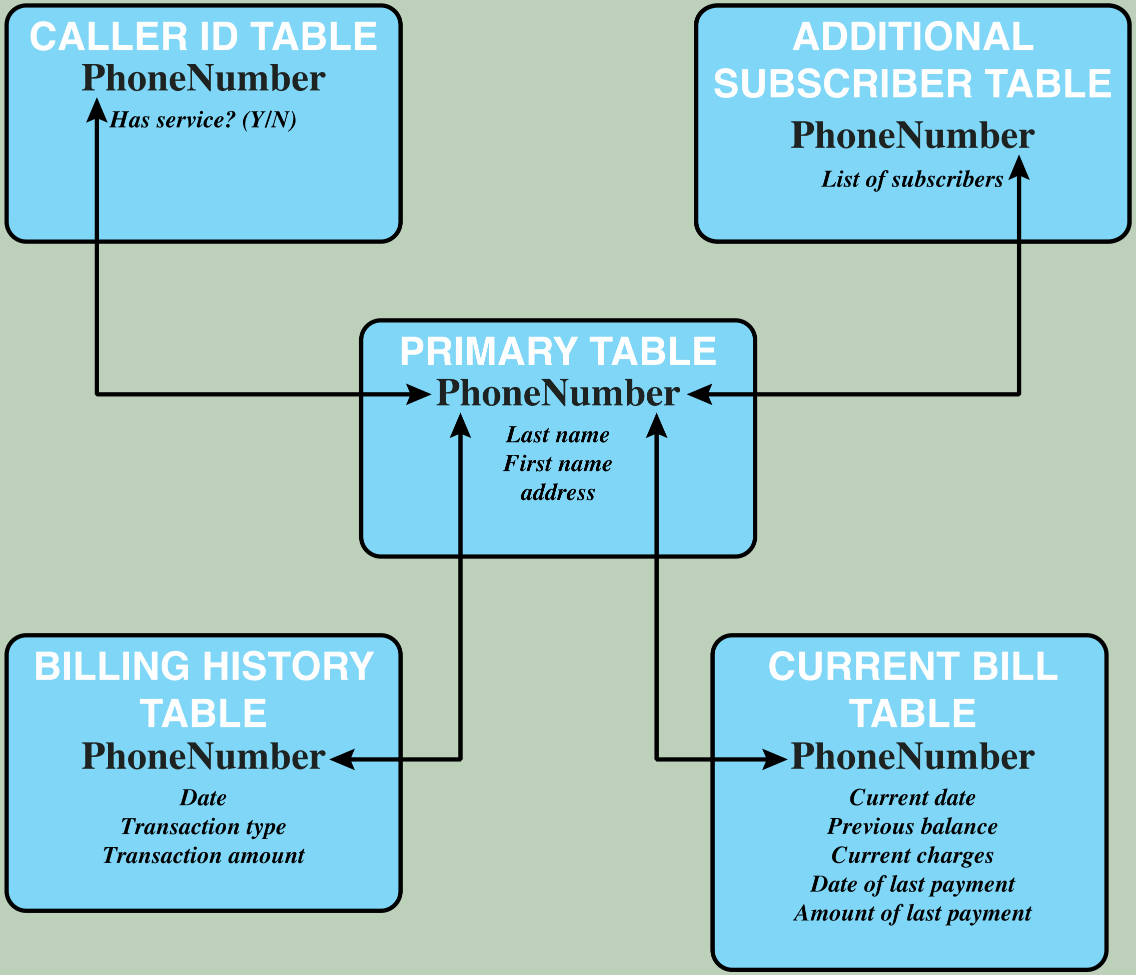


DDL = data definition language
DML = data manipulation language

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



CALLER ID TABLE
PhoneNumber
Has service? (Y/N)

ADDITIONAL SUBSCRIBER TABLE
PhoneNumber
List of subscribers

PRIMARY TABLE
PhoneNumber
Last name
First name
address

BILLING HISTORY TABLE
PhoneNumber
Date
Transaction type
Transaction amount

CURRENT BILL TABLE
PhoneNumber
Current date
Previous balance
Current charges
Date of last payment
Amount of last payment

Figure 5.2 Example Relational Database Model. A relational database uses multiple tables related to one another by a designated key; in this case the key is the PhoneNumber field.

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

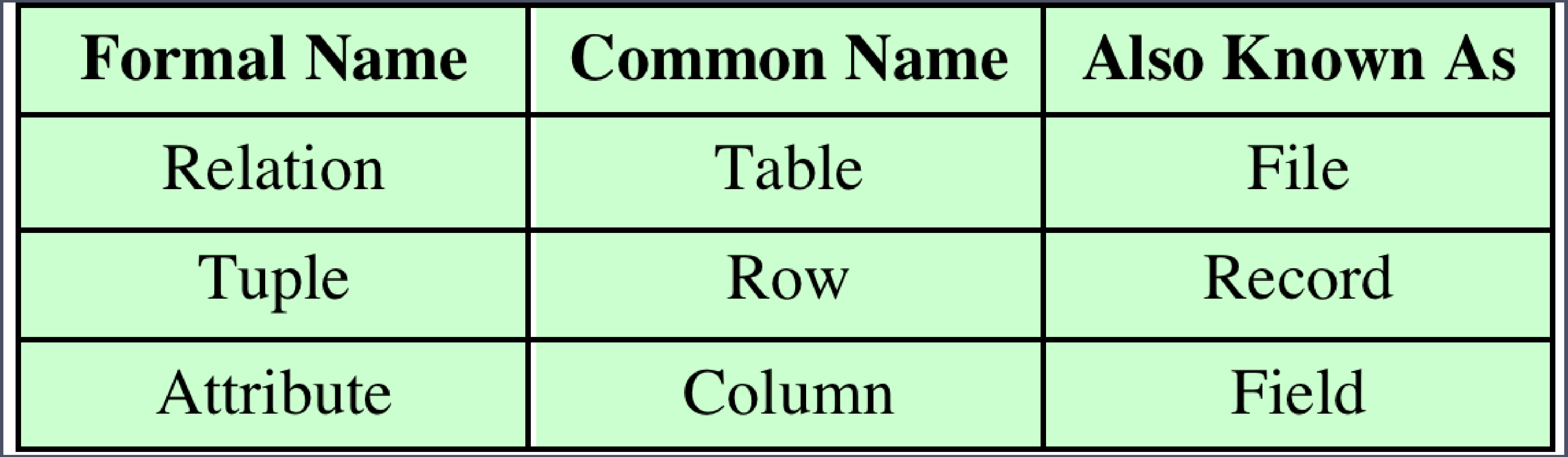
- Links one table to attributes in another

View/virtual table

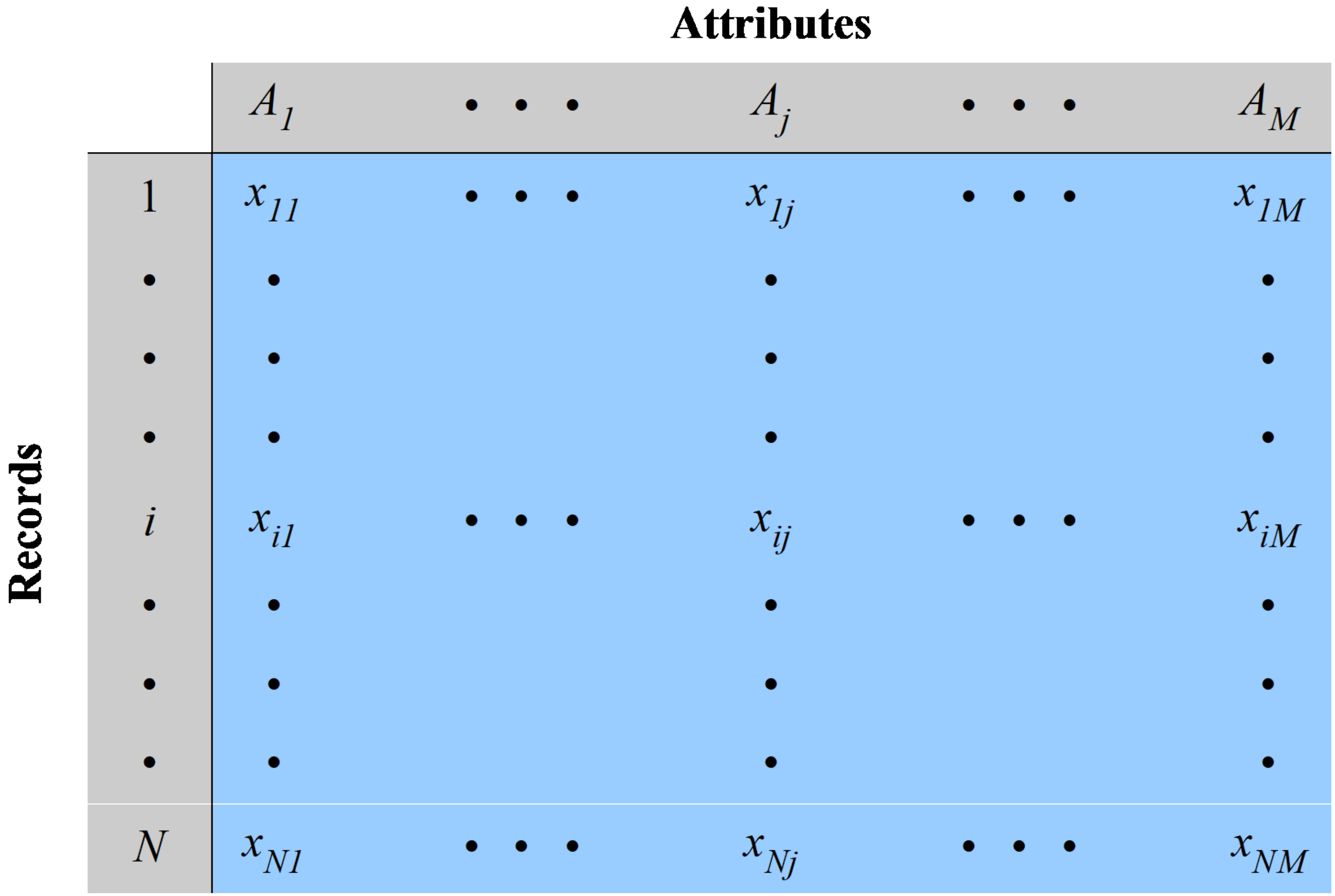
- Result of a query that returns selected rows and columns from one or more tables
- Views are often used for security purposes

Table 5.1

Basic Terminology for Relational Databases



Formal Name	Common Name	Also Known As
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

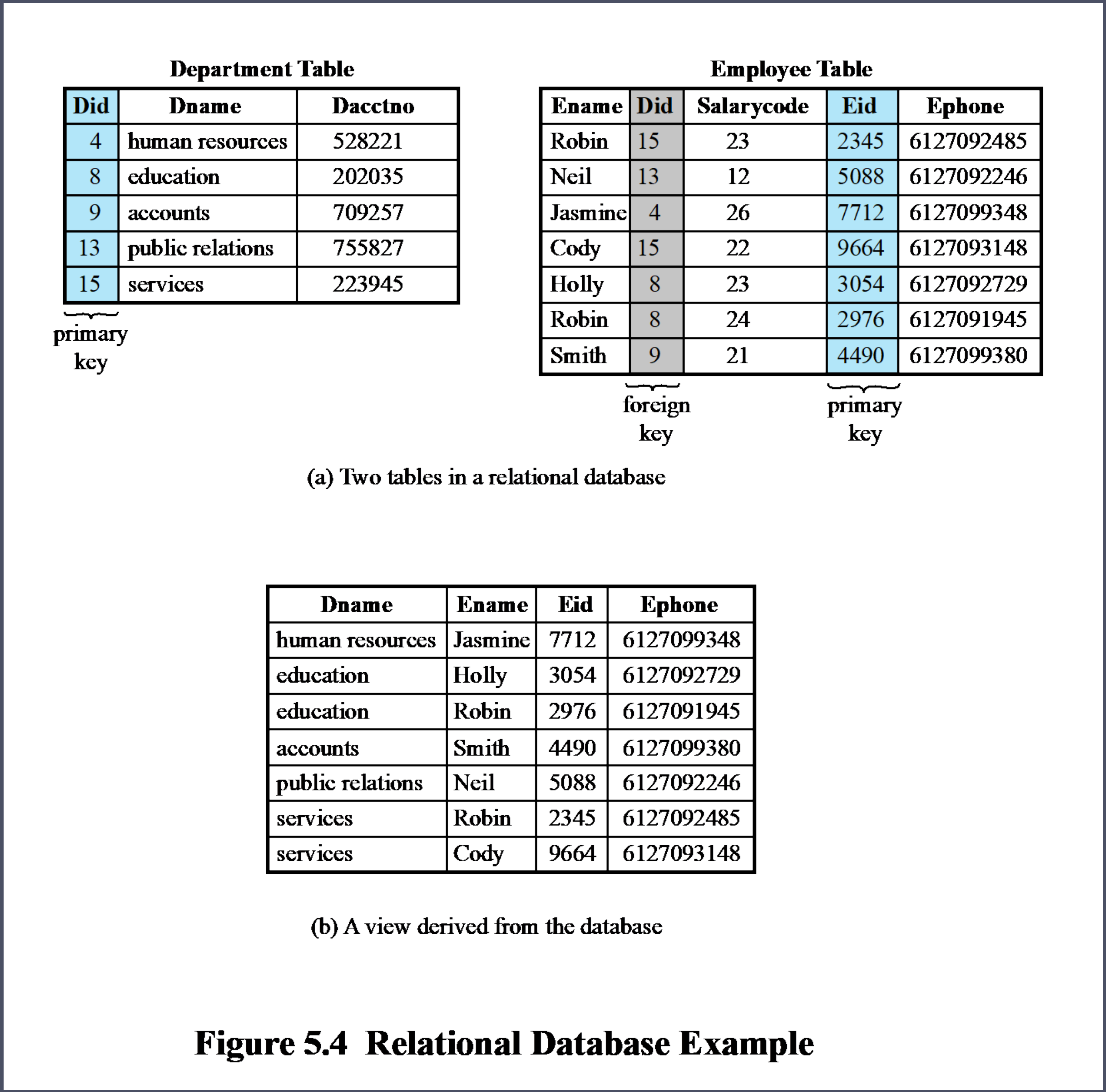


Attributes

Records

	A_1	• • •	A_j	• • •	A_M
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

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

primary key

Employee Table

Ename	Did	Salarycode	Eid	Ephone
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

(a) Two tables in a relational database

Dname	Ename	Eid	Ephone
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

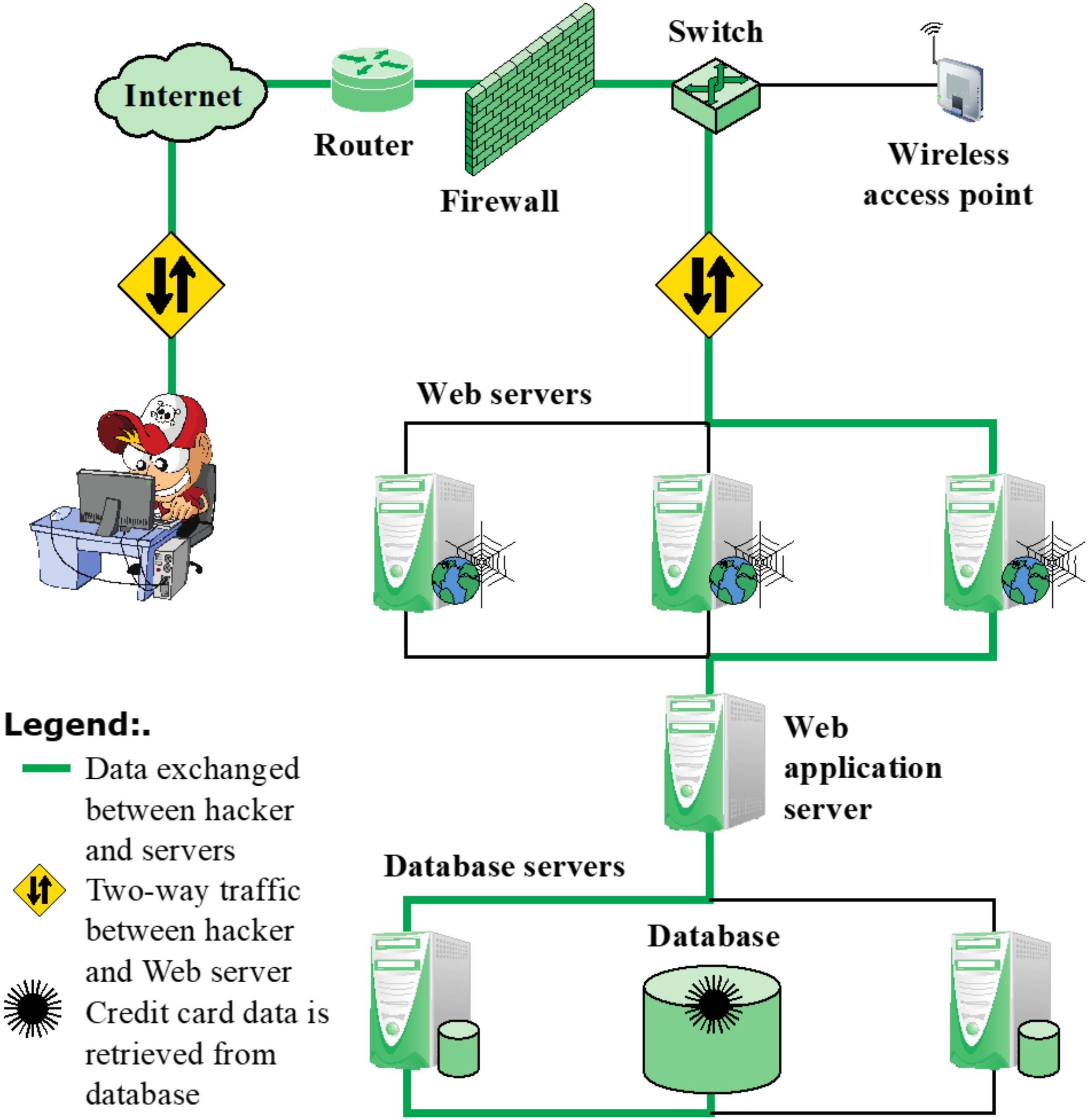
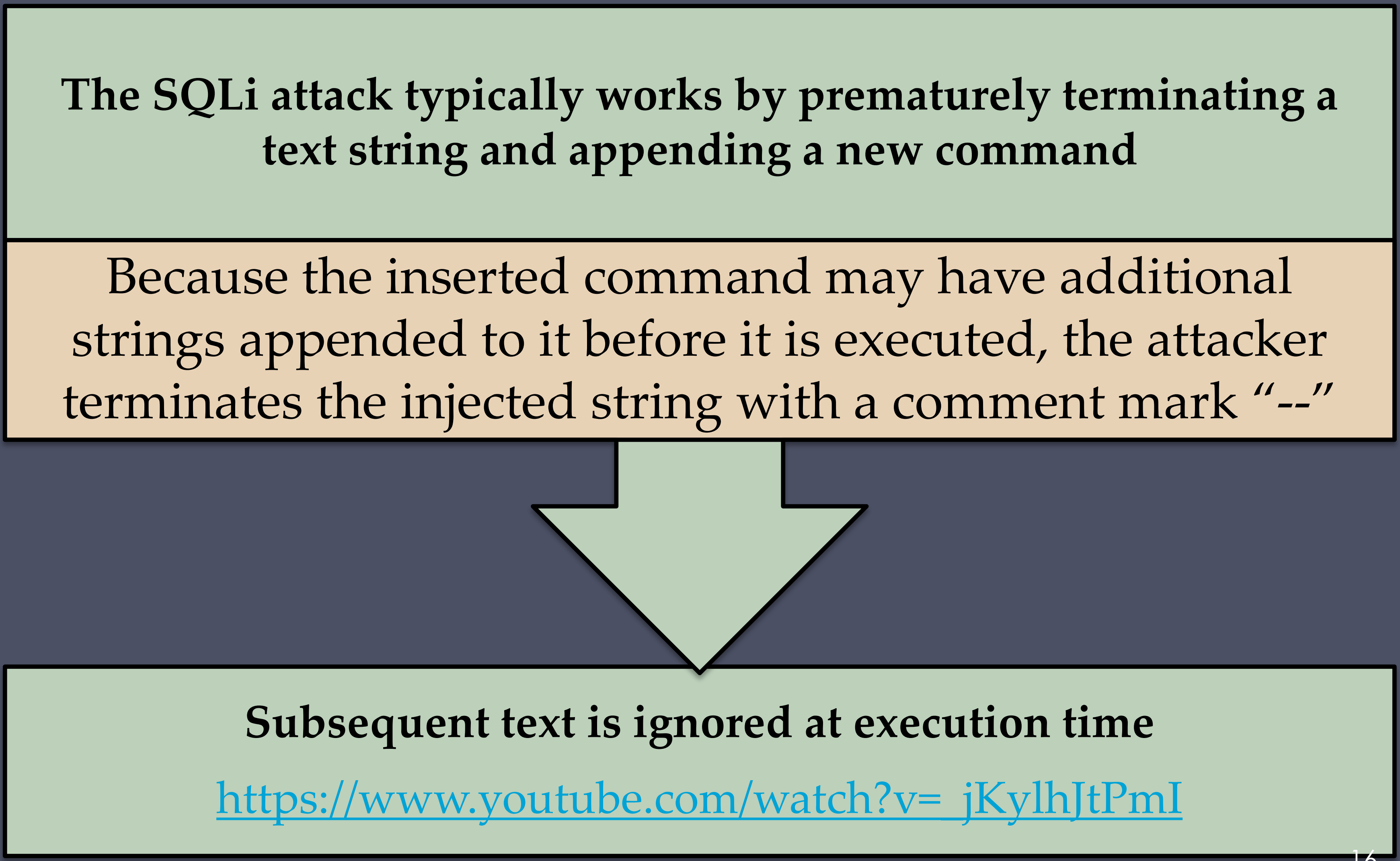

Internet
Router
Firewall
Switch
Wireless access point
Web servers
Web application server
Database servers
Database
Legend:.
Data exchanged between hacker and servers
Two-way traffic between hacker and Web server
Credit card data is retrieved from database

Figure 5.5 Typical SQL Injection Attack

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

https://www.youtube.com/watch?v=_jKylhJtPmI

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 in 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

SQLi Attack Avenues

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

SQL Attacks

- Three main types of SQL attacks:
  - In-band
  - Inferential
  - Out-of-band

Inband SQL Attacks

- Uses the same communication channel for injecting SQL code and retrieving results
- The retrieved data are presented directly in the application Web page
- Examples of attacks 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 SQL 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
- Examples of attacks 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 SQL Attack

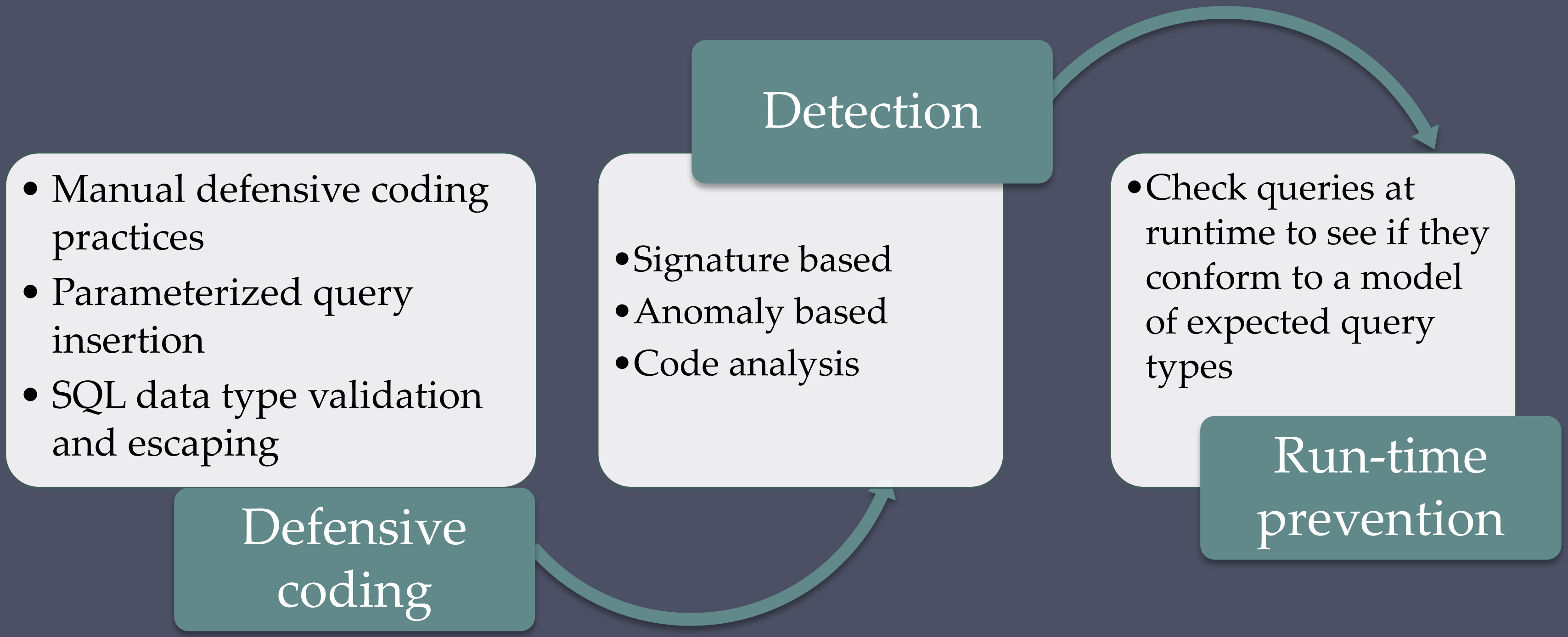
- Data are retrieved using a different channel

- E.g., an email with the results of the query is generated and sent to the tester

- This can be used when there are limitations on information retrieval, but outbound connectivity from the database server is lax

SQLi Countermeasures

- Three types:



Defensive coding

- Manual defensive coding practices
- Parameterized query insertion
- SQL data type validation and escaping

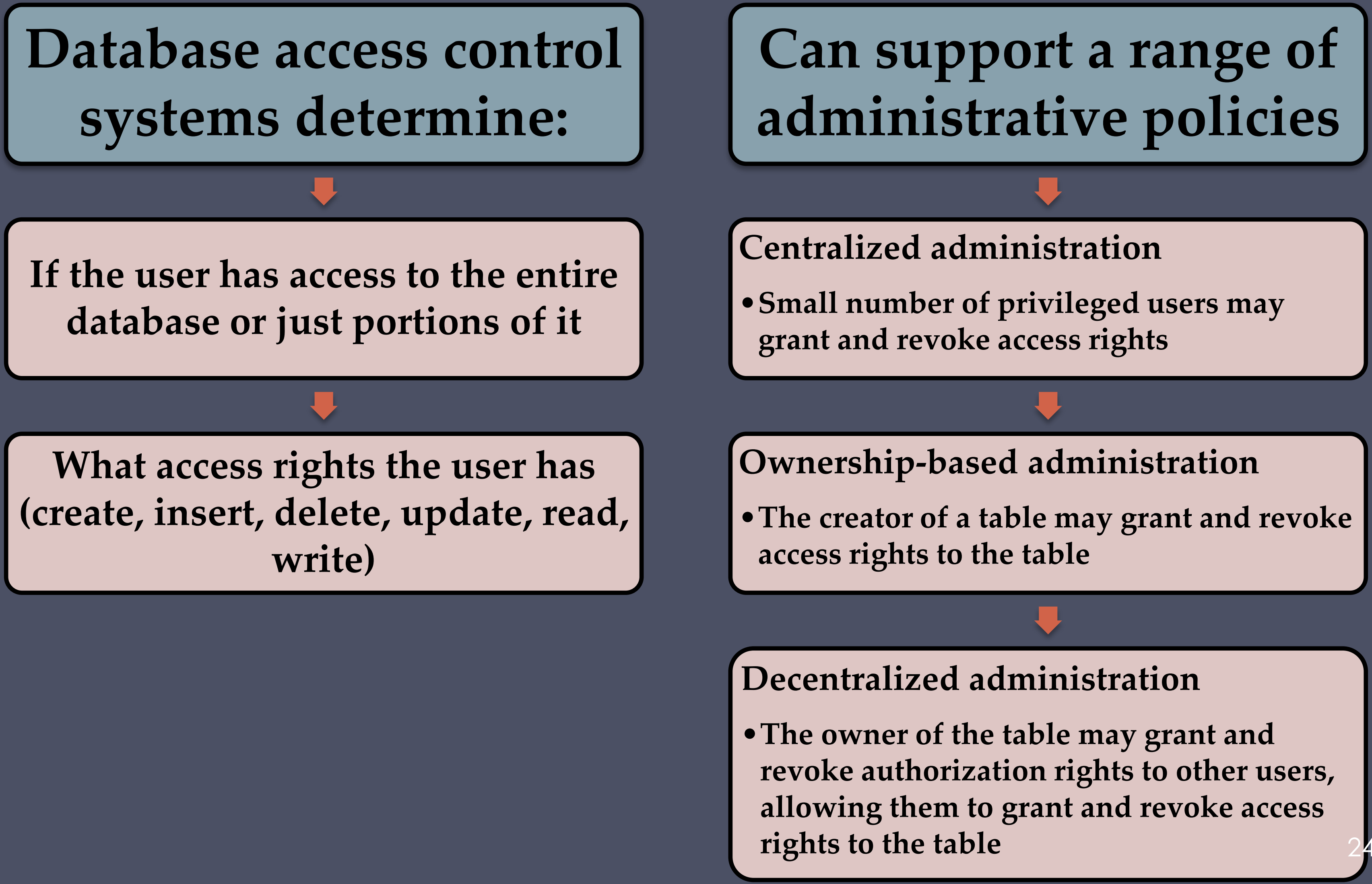
Detection

- Signature based
- Anomaly based
- Code analysis

Run-time prevention

- Check queries at runtime to see if they conform to a model of expected query types

Database Access Control



Database access control systems determine:

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

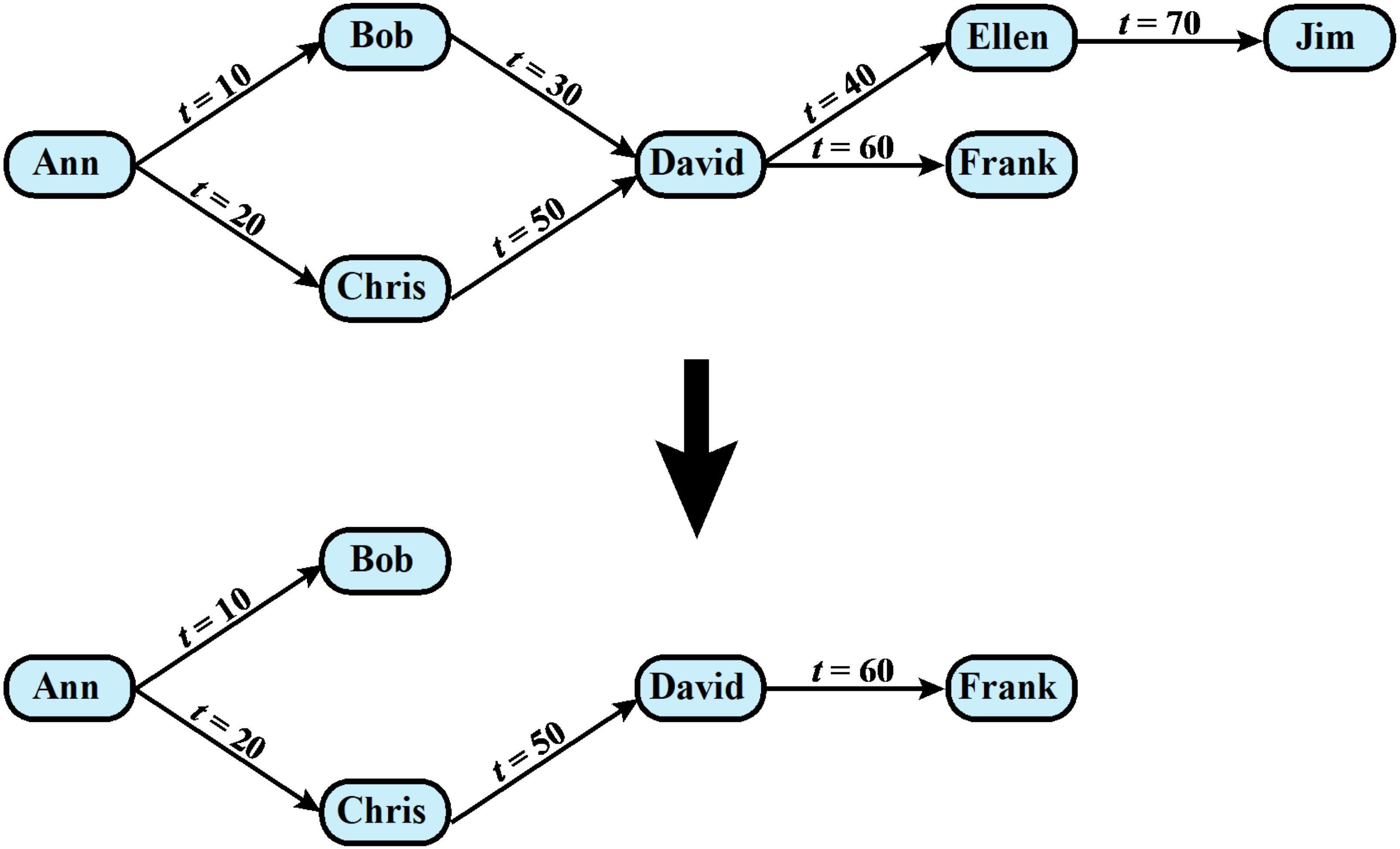


Figure 5.6 Bob Revokes Privilege from David

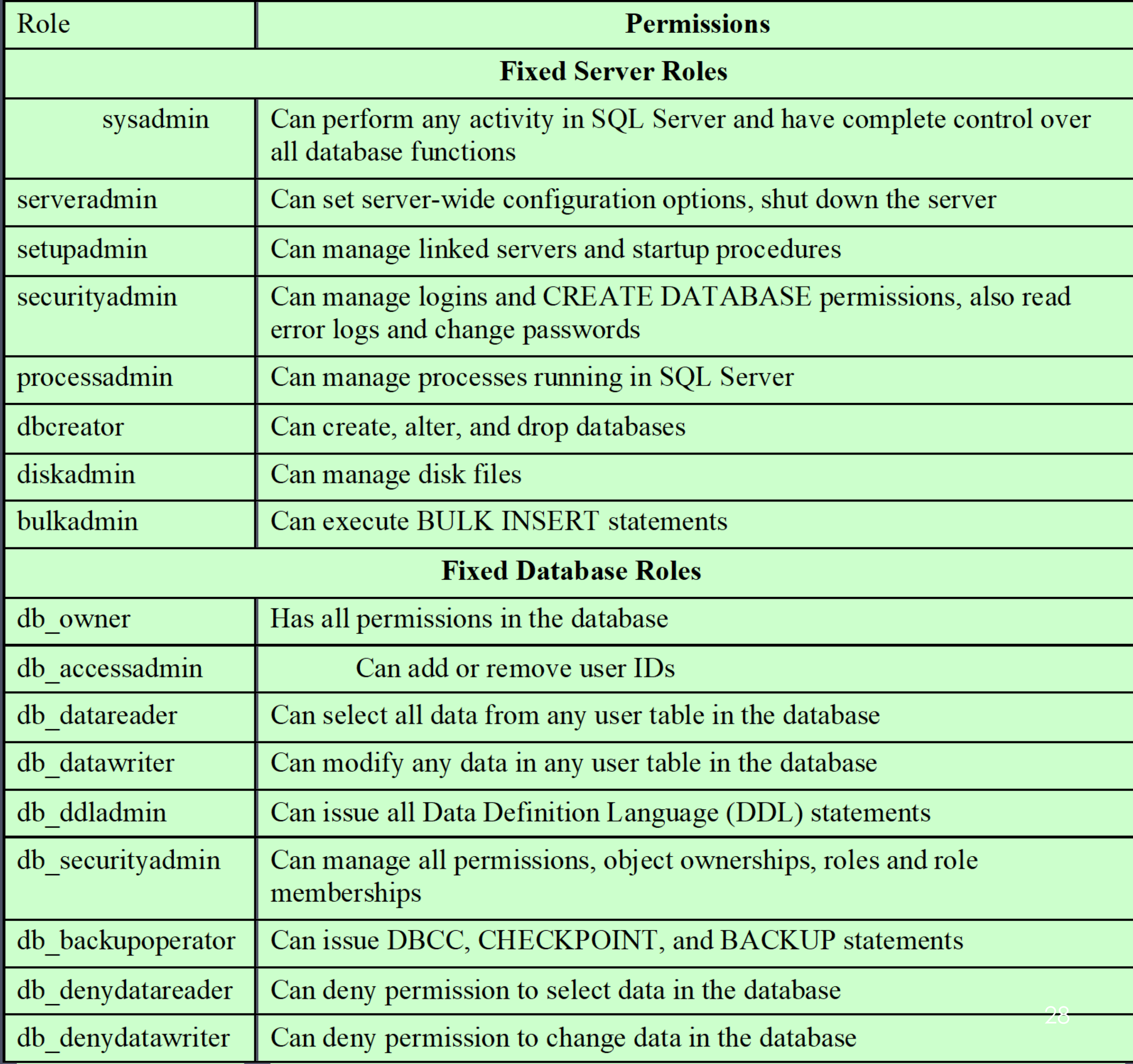
Role-Based Access Control (RBAC) in databases

- 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	End user	Administrator
• An end user who owns database objects as part of an application	• An end user who operates on database objects via a particular application but does not own any of the database objects	• User who has administrative responsibility for part or all of the database

Table 5.2

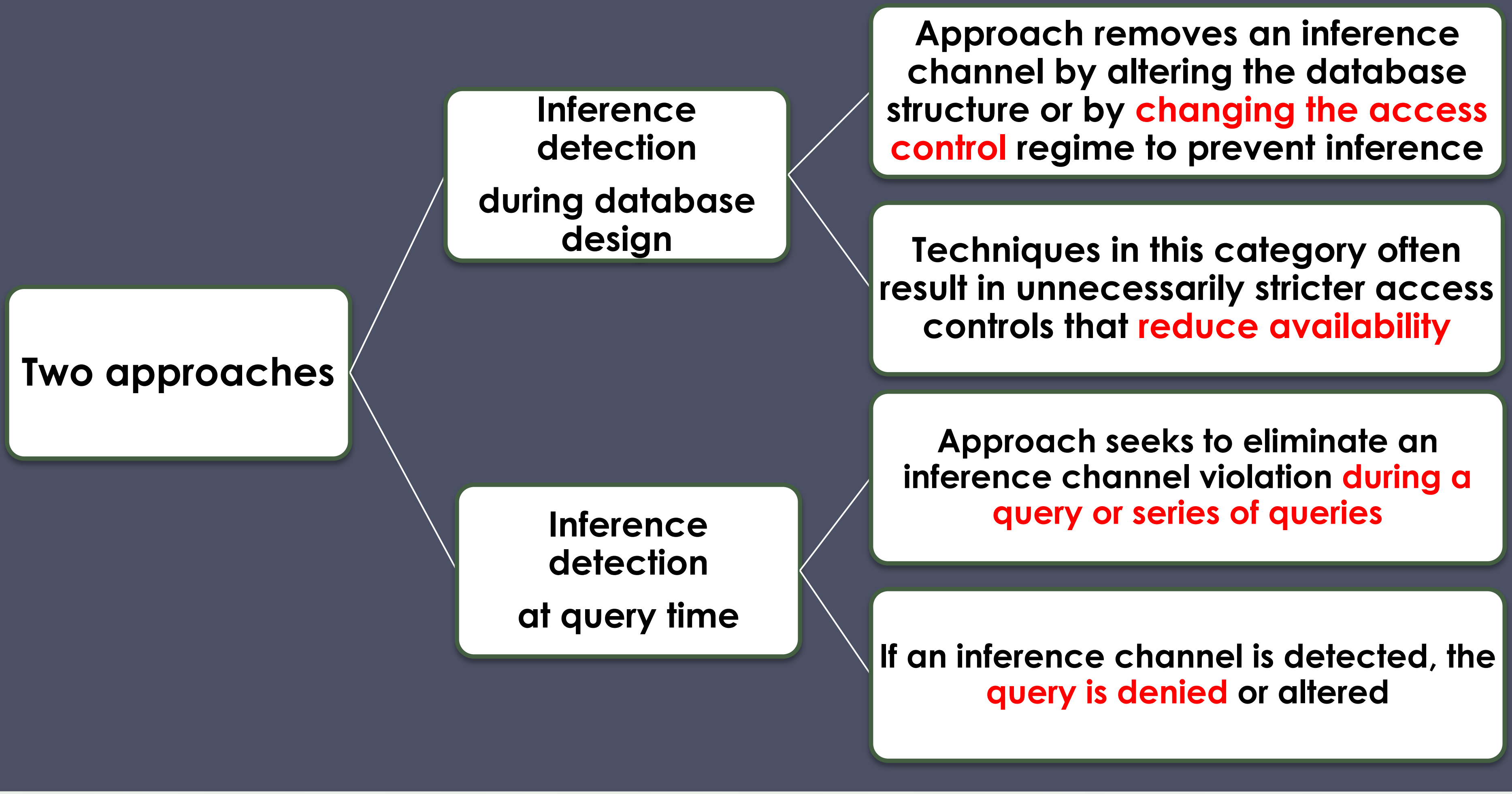
Fixed Roles in Microsoft SQL Server



Role	Permissions
Fixed Server Roles	
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
Fixed Database Roles	
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

(Table is on page 165 in the textbook)

Inference Detection



Two approaches

Inference detection

during database design

Approach removes an inference channel by altering the database structure or by changing the access control regime to prevent inference

Techniques in this category often result in unnecessarily stricter access controls that reduce availability

Inference detection

at query time

Approach seeks to eliminate an inference channel violation during a query or series of queries

If an inference channel is detected, the query is denied or altered

- 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

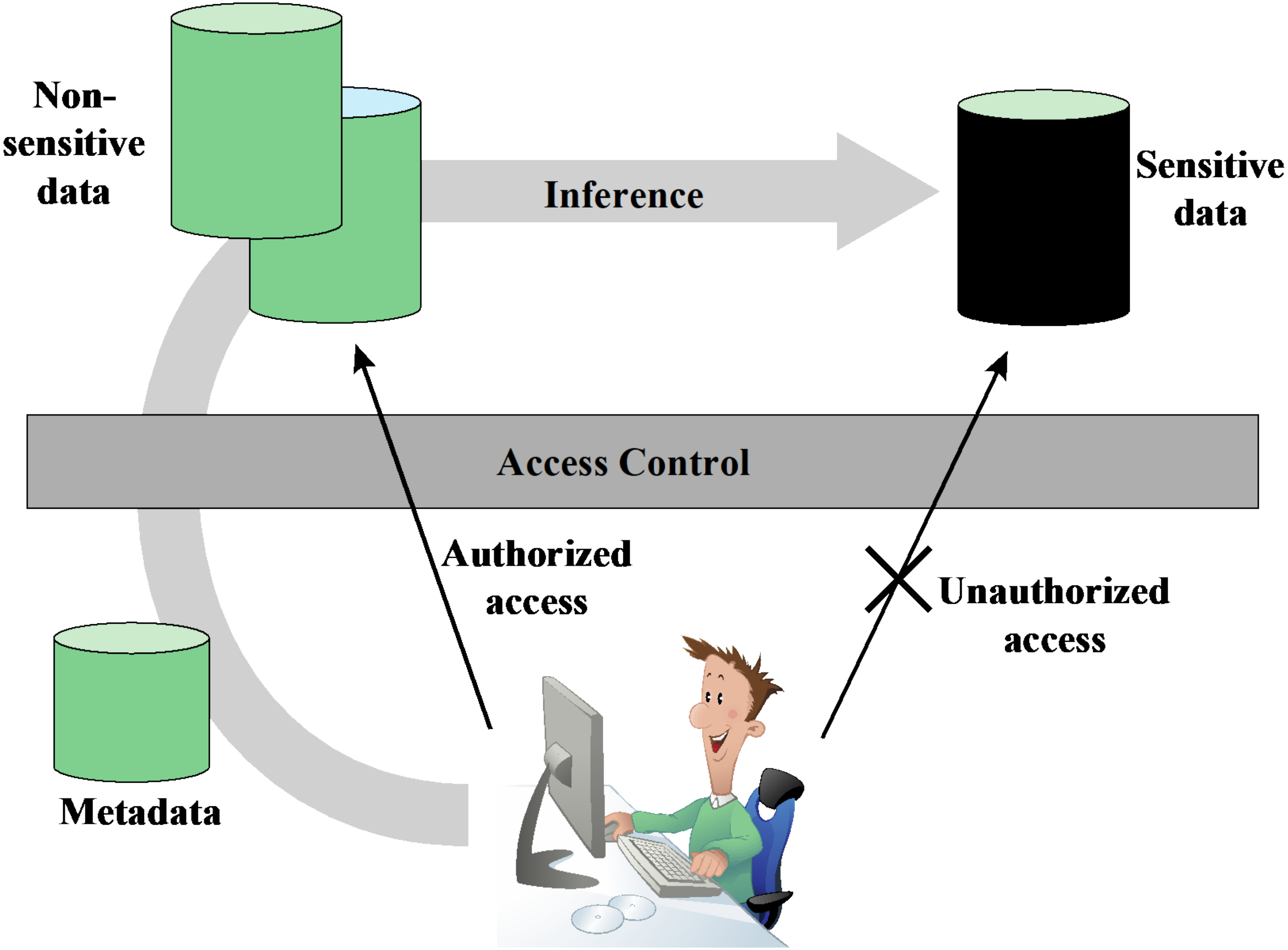


Figure 5.7 Indirect Information Access Via Inference Channel

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

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

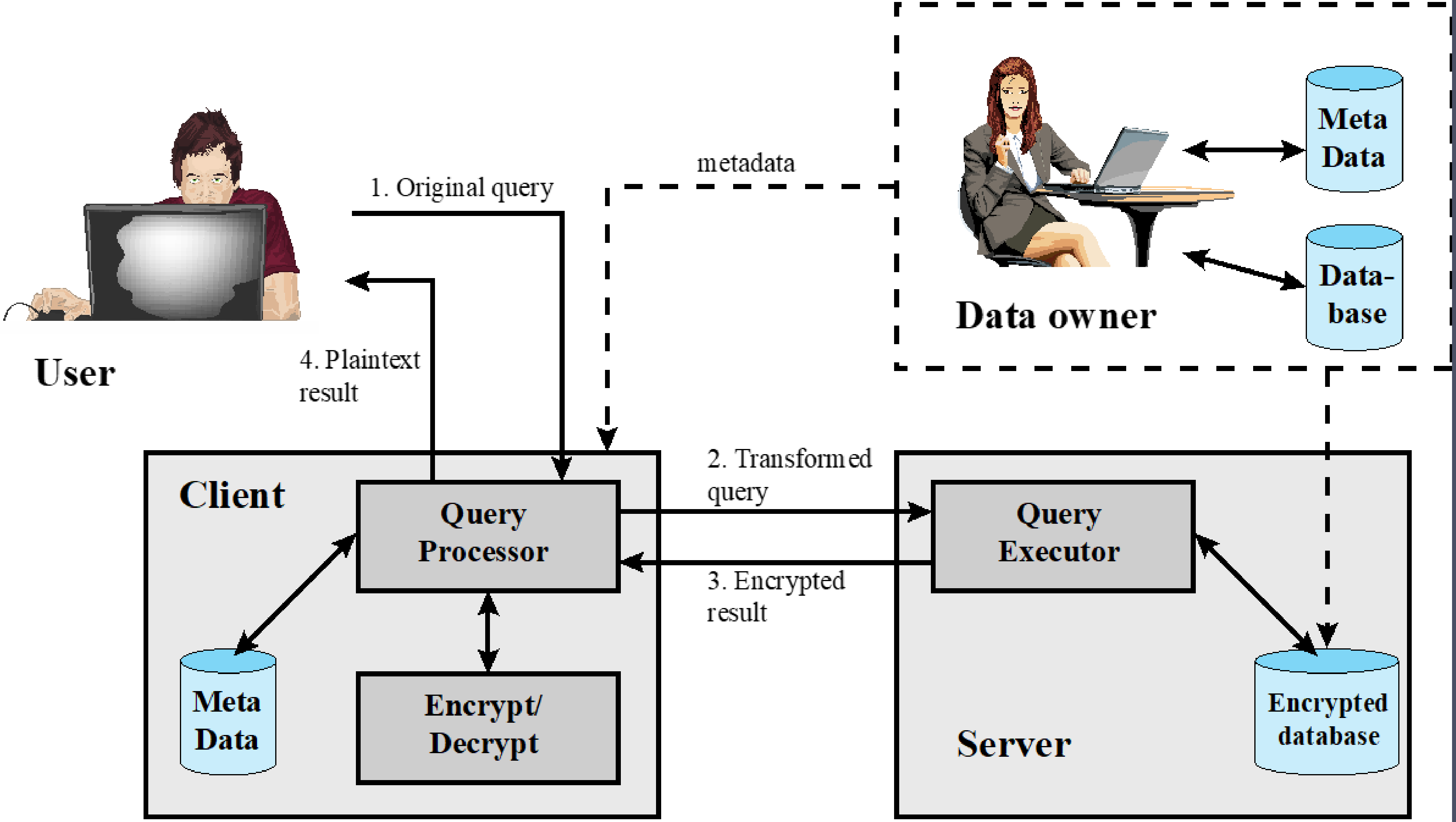


Figure 5.9 A Database Encryption Scheme

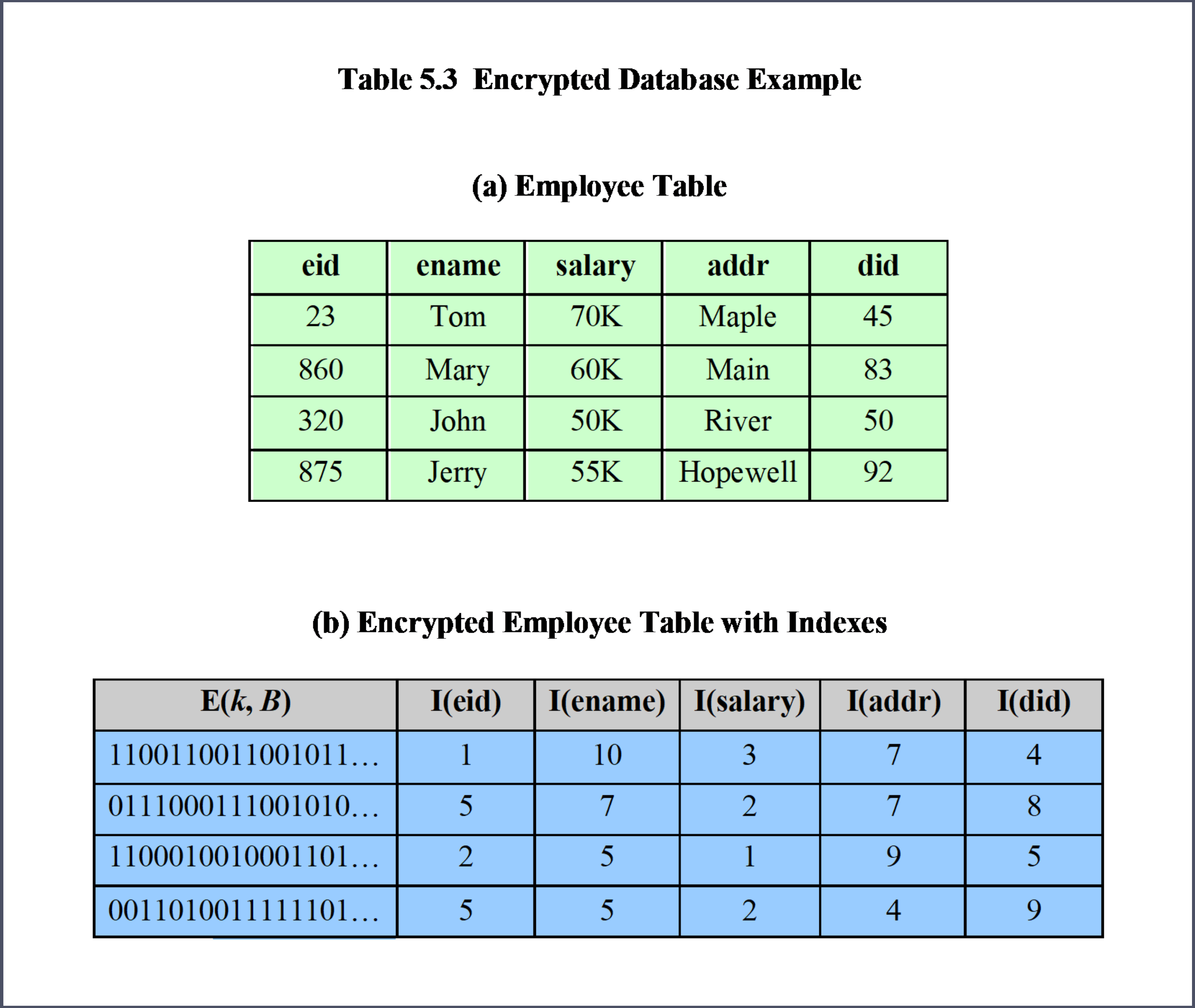


Table 5.3 Encrypted Database Example

(a) Employee Table

eid	ename	salary	addr	did
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

E(*k*, *B*)	I(eid)	I(ename)	I(salary)	I(addr)	I(did)
1100110011001011…	1	10	3	7	4
0111000111001010…	5	7	2	7	8
1100010010001101…	2	5	1	9	5
0011010011111101…	5	5	2	4	9

Data Center Security

- Data center:
  - An enterprise facility that houses a large number of servers, storage devices, and network switches and equipment
  - The number of servers and storage devices can run into the tens of thousands in one facility
  - Generally includes redundant or backup power supplies, redundant network connections, environmental controls, and various security devices
  - Can occupy one room of a building, one or more floors, or an entire building
- Examples of uses include:
  - Cloud service providers
  - Search engines
  - Large scientific research facilities
  - IT facilities for large enterprises

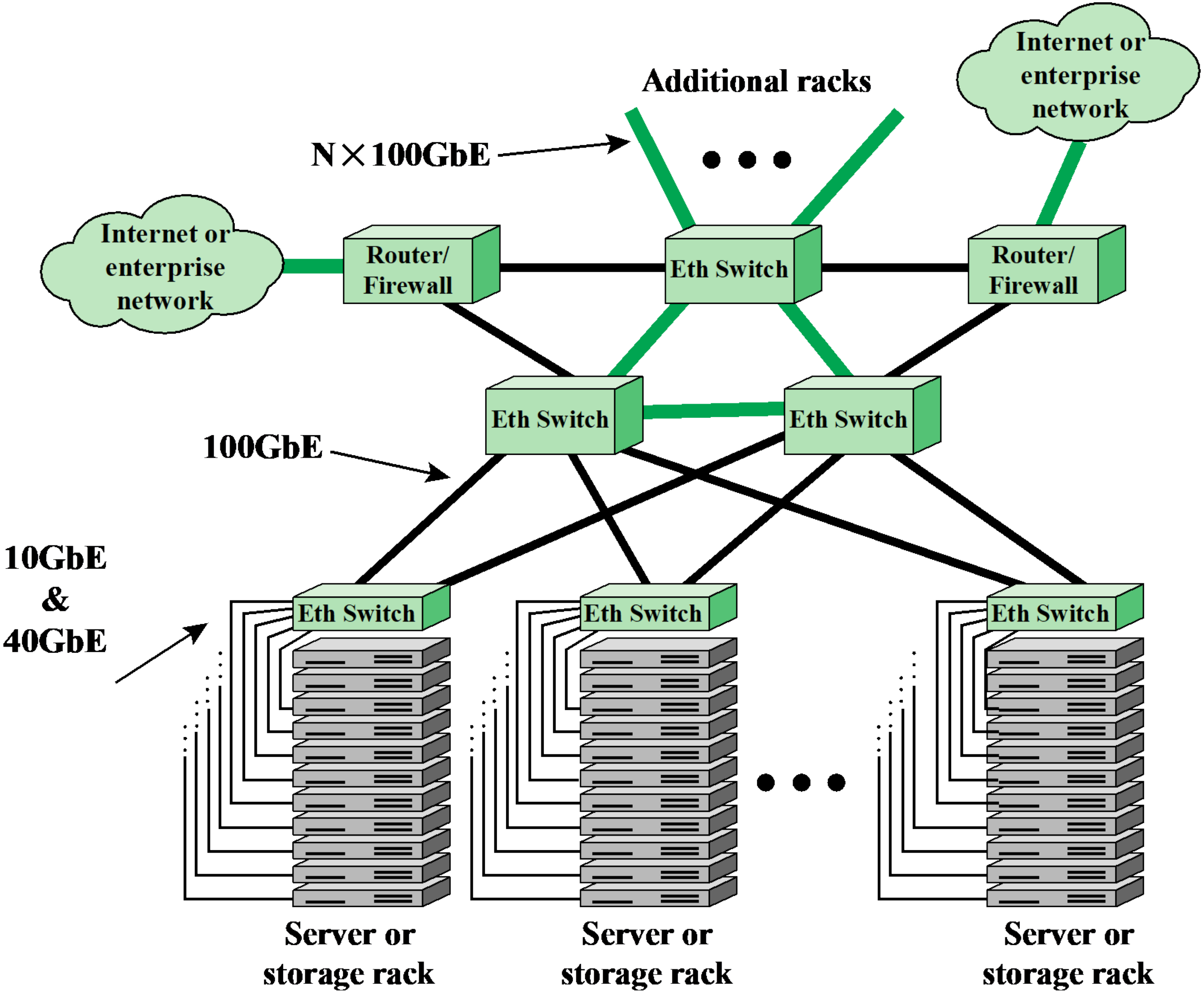
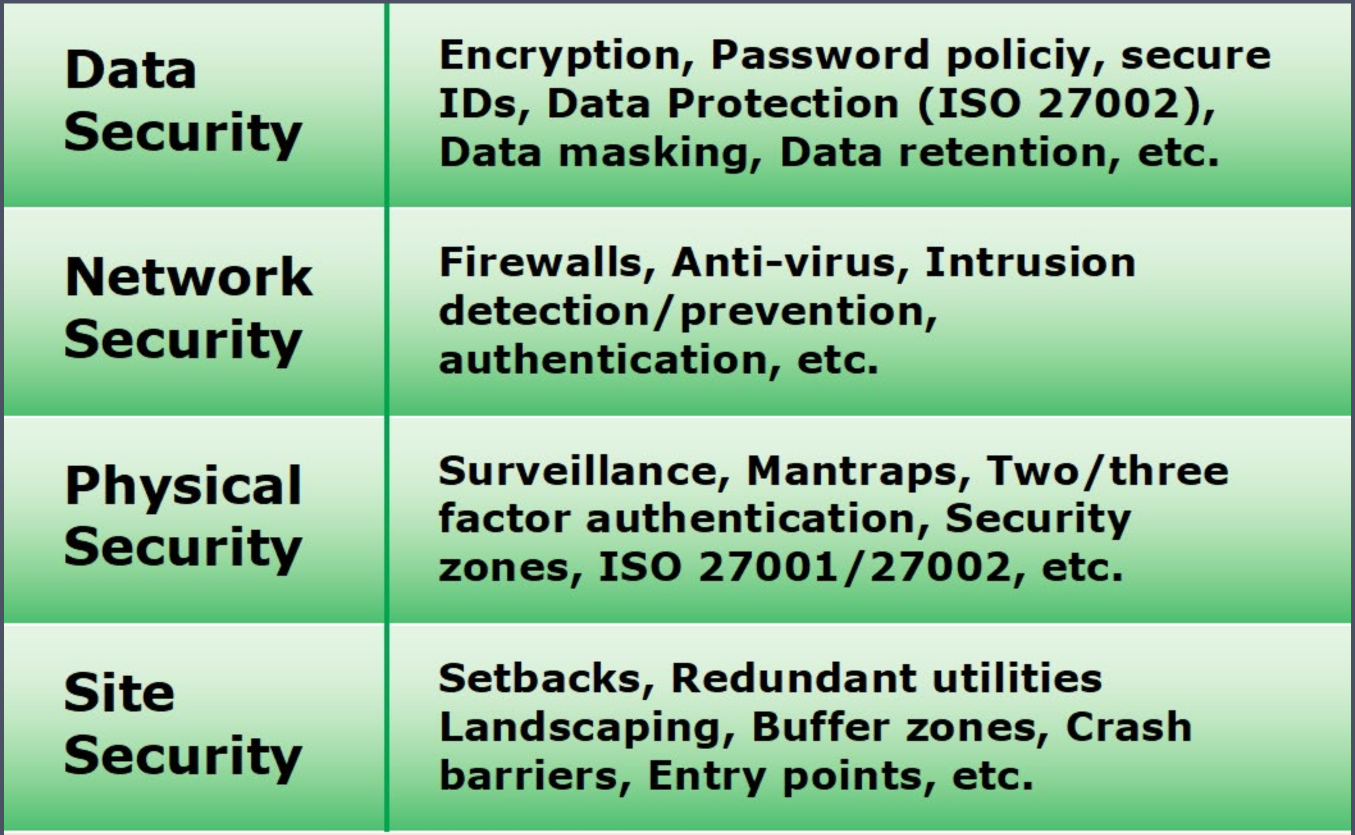


Figure 5.11 Key Data Center Elements



Data Security	Encryption, Password policiy, secure IDs, Data Protection (ISO 27002), Data masking, Data retention, etc.
Network Security	Firewalls, Anti-virus, Intrusion detection/prevention, authentication, etc.
Physical Security	Surveillance, Mantraps, Two/three factor authentication, Security zones, ISO 27001/27002, etc.
Site Security	Setbacks, Redundant utilities Landscaping, Buffer zones, Crash barriers, Entry points, etc.

Figure 5.12 Data Center Security Model

TIA-492

- The Telecommunications Industry Association (TIA)
- TIA-492 (*Telecommunications Infrastructure Standard for Data Centers*) specifies the minimum requirements for telecommunications infrastructure of data centers
- Includes topics such as:
  - Network architecture
  - Electrical design
  - File storage, backup, and archiving
  - System redundancy
  - Network access control and security
  - Database management
  - Web hosting
  - Application hosting
  - Content distribution
  - Environmental control
  - Protection against physical hazards
  - Power management

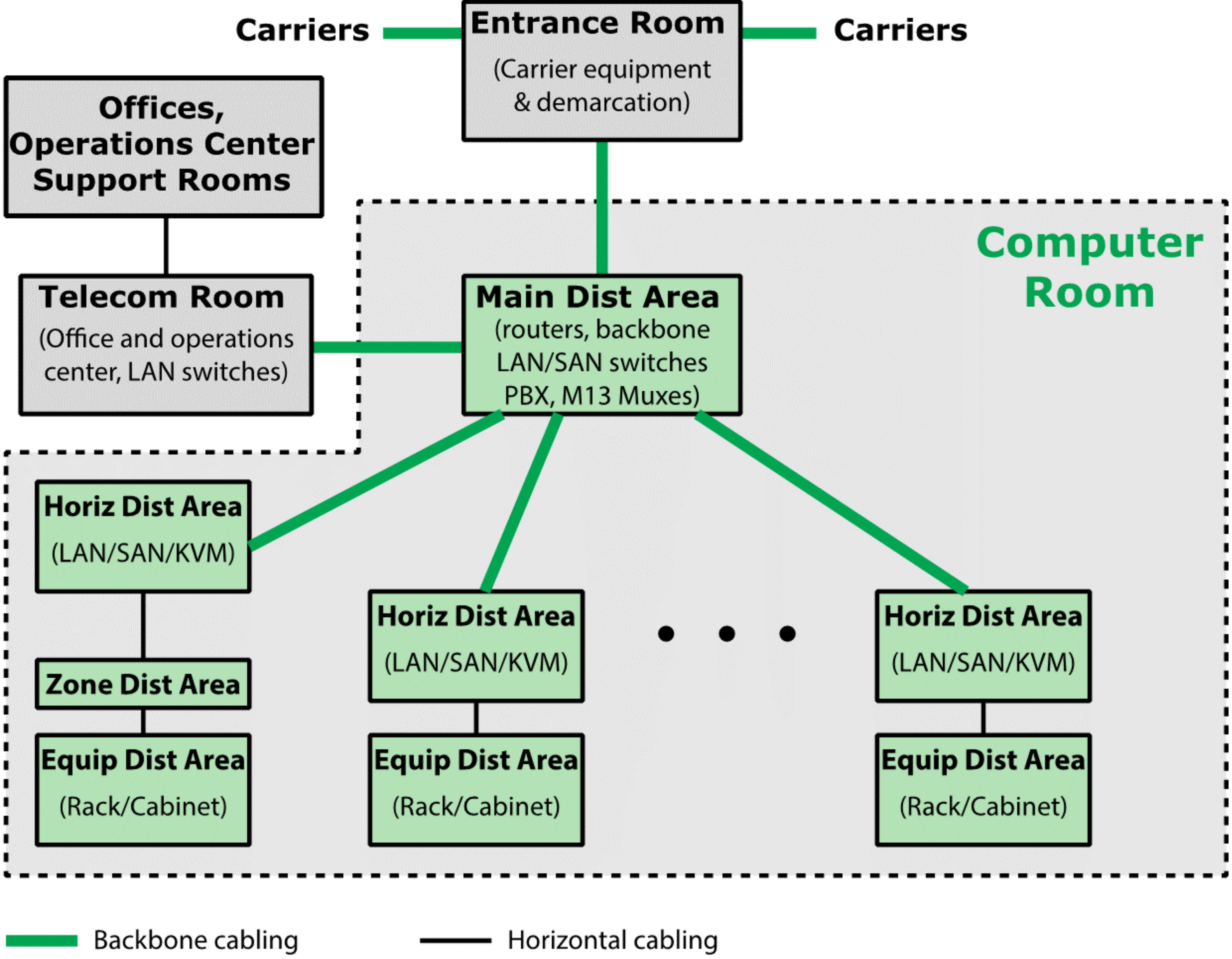
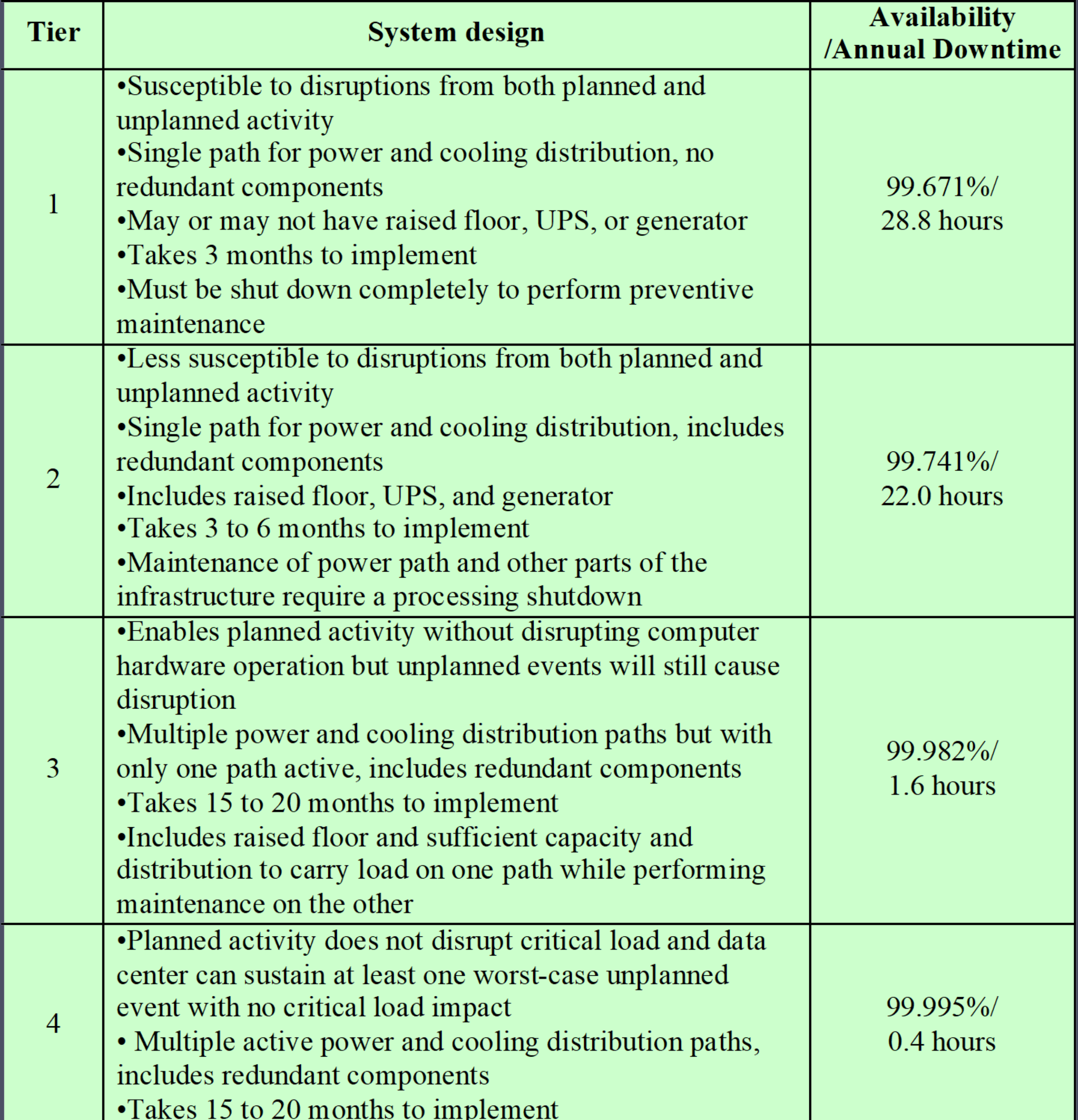


Figure 5.13 TIA-942 Compliant Data Center Showing Key Functional Areas

Table 5.4

Data Center Tiers Defined in TIA-942



Tier	System design	Availability /Annual Downtime
1	•Susceptible to disruptions from both planned and unplanned activity •Single path for power and cooling distribution, no redundant components •May or may not have raised floor, UPS, or generator •Takes 3 months to implement •Must be shut down completely to perform preventive maintenance	99.671%/ 28.8 hours
2	•Less susceptible to disruptions from both planned and unplanned activity •Single path for power and cooling distribution, includes redundant components •Includes raised floor, UPS, and generator •Takes 3 to 6 months to implement •Maintenance of power path and other parts of the infrastructure require a processing shutdown	99.741%/ 22.0 hours
3	•Enables planned activity without disrupting computer hardware operation but unplanned events will still cause disruption •Multiple power and cooling distribution paths but with only one path active, includes redundant components •Takes 15 to 20 months to implement •Includes raised floor and sufficient capacity and distribution to carry load on one path while performing maintenance on the other	99.982%/ 1.6 hours
4	•Planned activity does not disrupt critical load and data center can sustain at least one worst-case unplanned event with no critical load impact • Multiple active power and cooling distribution paths, includes redundant components •Takes 15 to 20 months to implement	99.995%/ 0.4 hours

(Table is on page 177 in textbook)

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

Practical 4

- Complete Problems 5.2, 5.8 and 5.9 in your textbook.
- Submit your work in PDF online on the course website.
- Be warned about plagiarism!
- Submission deadline: Wednesday 28 August 2024, 9:00am
- Absolutely no late submissions will be allowed. No excuse of slow server, load shedding, etc.
- No email submissions will be accepted.

Practical 4

5.2 The following table provides information on students of a computer programming club.

Student-ID	Name	Skill Level	Age
99	Jimmy	Beginner	20
36	David	Experienced	22
82	Oliver	Medium	21
23	Alice	Experienced	21

Practical 4

5.2 Continued…

The primary key is *Student-ID*. Explain whether or not each of the following rows can be added to the table.

Student-ID	Name	Skill Level	Age
91	Tom	Experienced	22
36	Dave	Experienced	21
	Bob	Beginner	20

Practical 4

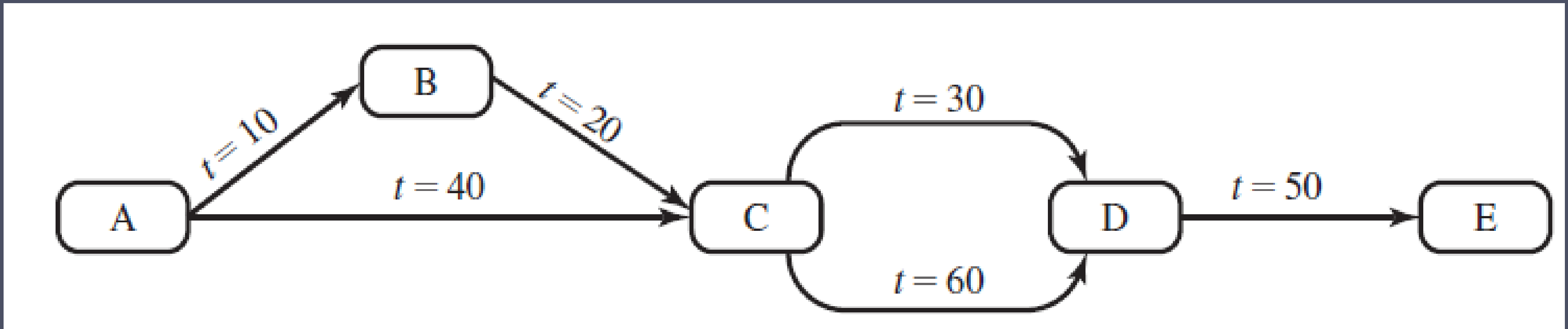
5.8 Assume A, B, and C grant certain privileges on the employee table to X, who in turn grants them to Y, as shown in the following table, with the numerical entries indicating the time of granting:

UserID	Table	Grantor	READ	INSERT	DELETE
X	Employee	A	15	15	—
X	Employee	B	20	—	20
Y	Employee	X	25	25	25
X	Employee	C	30	—	30

At time *t* = 35, B issues the command REVOKE ALL RIGHTS ON Employee FROM X. Which access rights, if any, of Y must be revoked, using the conventions defined in Section 5.2?

Practical 4

5.9 The figure below shows a sequence of grant operations for a specific access right on a table. Assume at *t* = 70, B revokes the access right from C. Using the conventions defined in Section 5.2, show the resulting diagram of access right dependencies.


A
B
C
D
E
t = 10
t = 20
t = 40
t = 30
t = 60
t = 50