CSc 484
Database Management Systems

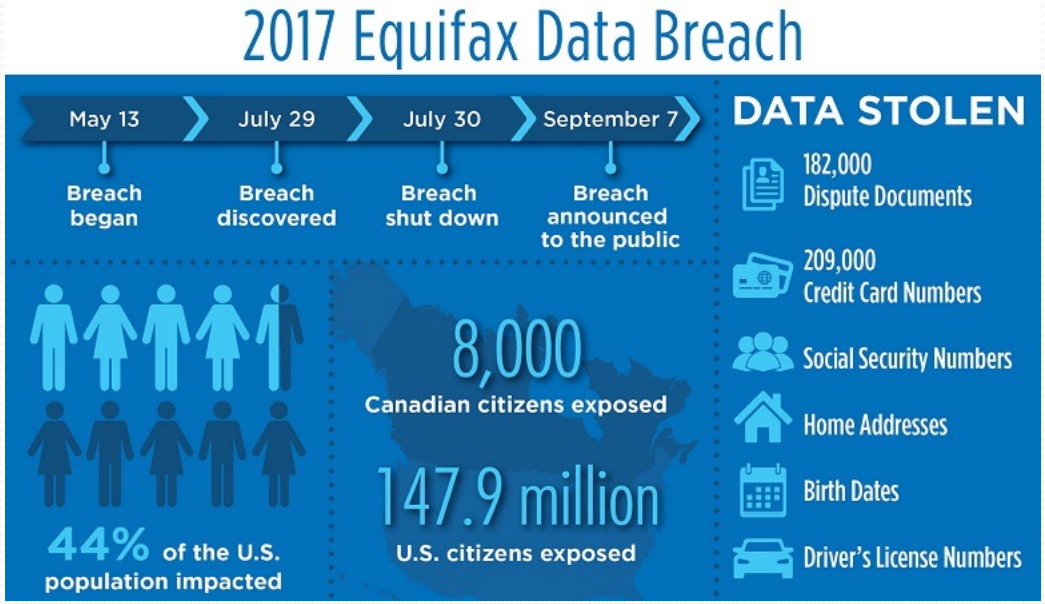
Ken Gamradt

Spring 2024

Security

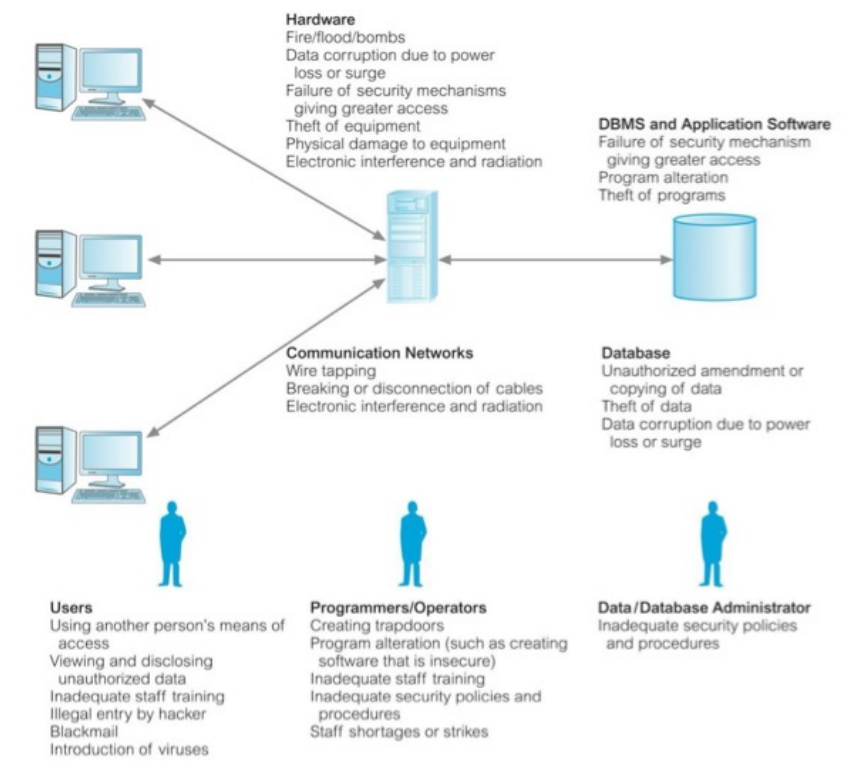
Database Security

- Data is the new wealth, and needs to be strictly controlled and managed
- Database security problem can be expensive



Database Security

- Potential threats to computer systems



Database Security

- Consider database security in relation to the following situation:
  - Theft and fraud
  - Loss of confidentiality (security)
  - Loss of privacy
  - Loss of integrity
  - Loss of availability

Counter Measures – computer-based controls

- Authorization
- Access controls
- Views
- Backup and recovery
- Integrity
- Encryption
- RAID technology

Authorization

- The granting of a right or privilege that enables a user or program to have legitimate access to a system or a system's object
  - A system's object represents a database
    - Relation
    - View
    - any other object

    that can be created within the system
- Authorization on data include:
  - Authorization on read data -- each is called a privilege
  - Authorization on insert new data --
  - Authorization on update data --
  - Authorization on delete data --
- Database administrator manages the users and the authorization of each user

Authorization

- When the SQL standard authorization mode is enabled, object owners can use the GRANT and REVOKE SQL statements to set the user privileges for specific database objects or for specific SQL actions

- The GRANT and REVOKE privileges are:
  - DELETE
  - EXECUTE
  - INSERT
  - SELECT
  - REFERENCES
  - TRIGGER
  - UPDATE

Authorization

- The GRANT statement is used to confer authorization

grant <privilege list>
on <relation name or view name>
to <user/role list>

grant select
on department
to Ken

- The user Ken can operate SELECT statements on the department relation

- PostgreSQL
  - https://www.postgresql.org/docs/current/sql-grant.html

Authorization

- The REVOKE statement is used to revoke an authorization

revoke <privilege list>
on <relation name or view name>
from <user/role list>

revoke select
on department
from Ken

- The SELECT privilege on the department relation is revoked from user Ken

- PostgreSQL
  - https://www.postgresql.org/docs/current/sql-revoke.html

Access Controls

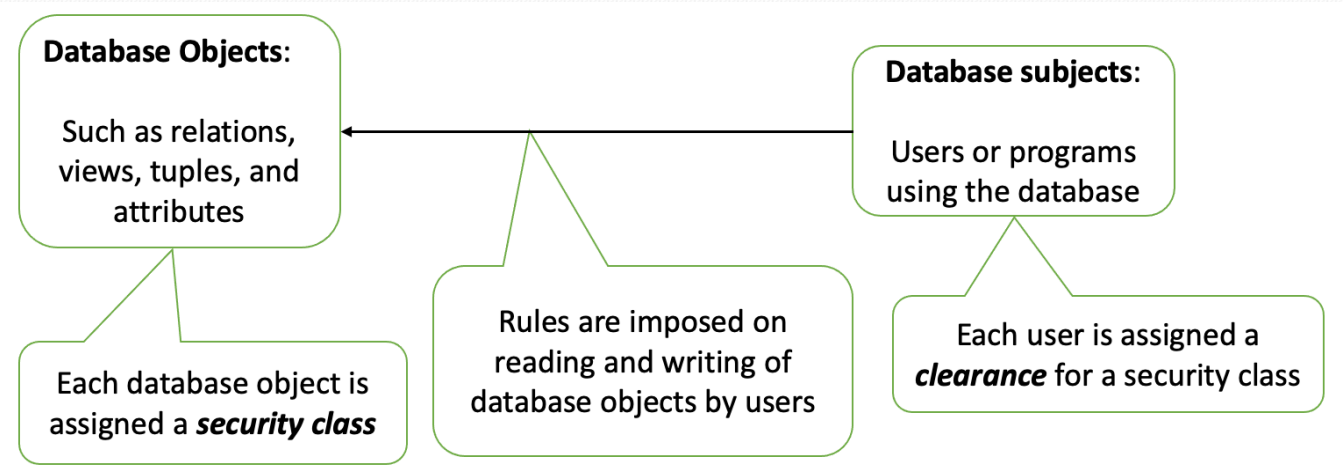
- Based on the granting and revoking of privileges
  - A privilege allows a user to create or access (read, write, or modify) some database object (relation, view, …) or to run certain DBMS utilities
- Approach 1: Discretionary Access Control (DAC)
  - SQL standards support it through GRANT and REVOKE

grant <privilege list>
on <relation name or view name>
to <user/role list>

revoke <privilege list>
on <relation name or view name>
from <user/role list>

Access Controls

- Approach 2: Mandatory Access Control (MAC)
  - Based on system-wide policies that cannot be changed by individual users
  - **NOT** supported by the SQL standard



Access Controls

- MAC
  - Security classes in a system are ordered, with a most secure class and a least secure class
- Example:
  - A system defines four security classes:
    - Tope Security (TS) -- Most secure class
    - Security (S)
    - Confidential (C)
    - Unclassified (U) -- Least secure class

  - Each object is assigned a security class
    - The relation Instructor is assigned in a Confidential security class
  - Each user is assigned a clearance for a security class
    - The student user is assigned with Unclassified clearance

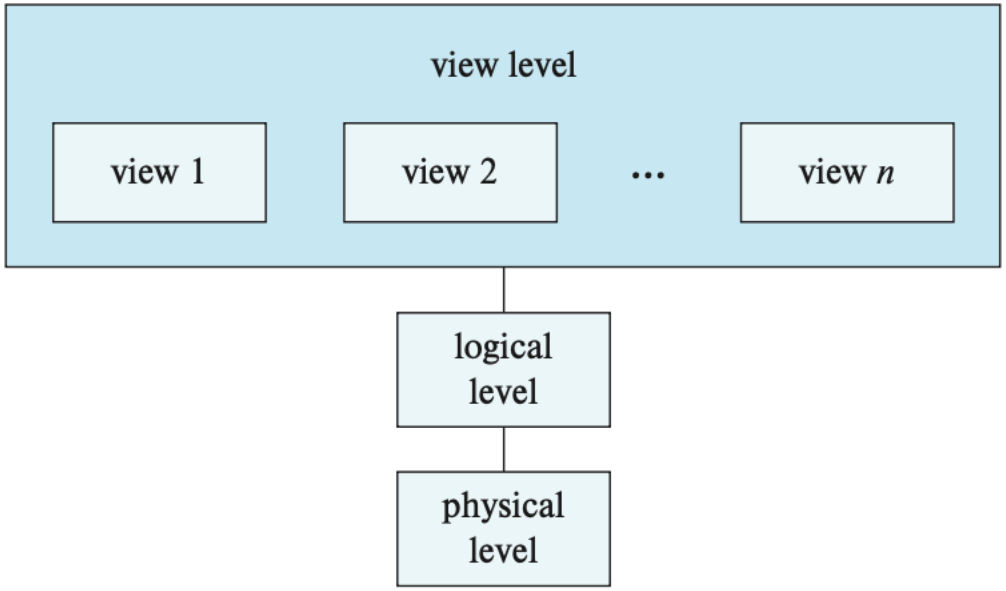
Access Controls

- Restrictions on all reads and writes of database objects
  - Subject S is allowed to read object O only if class(S) >= class(O)
    - E.g., a user with TS clearance can read a relation with C classification, but a user with C clearance cannot read a relation with TS classification
  - Subject S is allowed to write object O only if class(S) <= class(O)
    - E.g., a user with S clearance can only write objects with S or TS classification

- These rules seek to ensure that sensitive data can never be passed on to another user without the necessary clearance

Views

- An architecture of a database system



Views

- View:
  - The dynamic result of one or more relational operations operating on the base relations to produce another relation
  - A virtual relation that does not actually exist in the database, but is produced upon request by a particular user, at the time of request
- The view mechanism provides a powerful and flexible security mechanism by hiding parts of the database from certain users

Views

- A view is defined as query on one or more base relations or views
- Create a view:

```
create view viewName
as <query expression> ;          -- any legal query expression
```

Views

- E.g., staff from Computer Science only have the authorization to view the information of the instructors from the Computer Science department

Query Editor | Query History

```
1  select *
2  from instructor
3  where dept_name = 'Comp. Sci.'
```

Data Output

	id [PK] character varying (5)	name character varying (20)	dept_name character varying (20)	salary numeric (8,2)
1	10101	Srinivasan	Comp. Sci.	65000.00
2	45565	Katz	Comp. Sci.	75000.00
3	83821	Brandt	Comp. Sci.	92000.00

- There may be some concerns with this …

Views

- E.g., staff from Computer Science only have the authorization to view the information of the instructors from the Computer Science department
  - PostgreSQL – pgAdmin

Query Editor | Query History

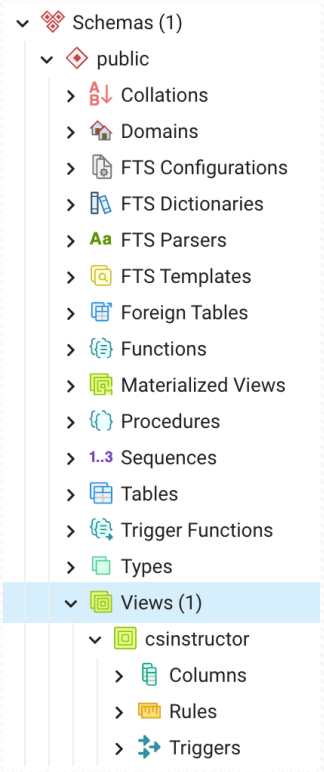
```
1  create view csInstructor
2  as select *
3      from instructor
4      where dept_name = 'Comp. Sci.'
```

Query Editor | Query History

```
1  select *
2  from csInstructor
```

Data Output

	id character varying (5)	name character varying (20)	dept_name character varying (20)	salary numeric (8,2)
1	10101	Srinivasan	Comp. Sci.	65000.00
2	45565	Katz	Comp. Sci.	75000.00
3	83821	Brandt	Comp. Sci.	92000.00



Views

- A view can be generated from multiple base relations
- E.g., create a view that list all course sections offered by the Physics dept in Fall 2017 with the building and room number of each section
  - PostgreSQL – pgAdmin

Query | Query History

```
1  create view physicsFall2017
2  as select C.course_id, sec_id, building, room_number
3      from course as C join section as S on (C.course_id = S.course_id)
4      where semester = 'Fall' and year = 2017 and dept_name = 'Physics';
```

Query | Query History

```
1  select *
2  from physicsFall2017
3
```

Data Output | Scratch Pad

	course_id character varying (8)	sec_id character varying (8)	building character varying (15)	room_number character varying (7)
1	PHY-101	1	Watson	100

Views

- The attribute names of a view can be specified explicitly

create view viewName **[(newAttributeName[,...])]**
as <query expression>

[(newAttributeName[,...])] -- Specify the attribute names explicitly
-- Very useful when the attributes come from
-- calculation or aggregate functions

Views

- Explicitly specify attribute names for a view
- E.g., create a view with all dept_name and total salary in each dept

Query Editor | Query History

```
1  create view dept_salary (dept_name, total_salary)
2  as select dept_name, sum(salary)
3      from instructor
4      group by dept_name
```

-- aggregate function expression
-- does not have a name

Query Editor | Query History

```
1  select dept_name, total_salary
2  from dept_salary
```

Data Output

	dept_name character varying (20)	total_salary numeric
1	Finance	170000.00
2	History	122000.00
3	Physics	182000.00
4	Music	40000.00
5	Comp. Sci.	232000.00
6	Biology	72000.00
7	Elec. Eng.	80000.00

Views

- Virtual relation that does not exist in the database
- How to handle queries based on views
  - Approach 1 (view resolution):
    - Database system stores the definition of the view
    - When a view is needed, the system looks up the definition and translates the request into an equivalent request against the source table of the view and then performs the equivalent request

Views

- Virtual relation that does not exist in the database
- How to handle queries based on views
  - Approach 2 (view materialization):
    - Store the view as a temporary relation in the database and maintain the currency of the view as the underlying base relations are updated
    - The update of view relations can be done immediately when any of the relations on which the view is defined is updated
    - Pro: fast response
    - Con: storage costs added

Views

- Remove a view
  - Delete the definition of view from database

drop view viewName **[RESTRICT | CASCADE]**

RESTRICT -- default - drop operation is rejected if there are any other
-- objects that depend on the existence of the
-- view being dropped

CASCADE -- all views defined on the dropped view are deleted

Query Editor | Query History

```
1  drop view physicsFall2017
```

Views

- Update on views can be difficult

Query Editor | Query History

```
1  create view instructor_info
2  as select ID, name, building
3      from instructor, department
4      where instructor.dept_name = department.dept_name
```

Query Editor | Query History

```
1  insert into instructor_info
2      values ('69987', 'White', 'Taylor')
```

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
69987	White	*null*	*null*

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Music	Packard	80000
Physics	Watson	70000
null	Taylor	*null*

- Updates on views actually happen on underlying base relations
  - No values for salary, dept_name, budget
- Problem: instructor_info view cannot get
  - (‘69987’, ‘White’, ‘Taylor’)

Views

- View updatability:
  - The FROM clause has only one database relation
  - The SELECT clause contains only attribute names of the relation and does not have any
    - Expressions
    - Aggregates
    - DISTINCT specification
  - Any attribute not listed in the select clause can be set to NULL if it
    - does not have a NOT NULL constraint
    - is not part of a primary key
  - The query does not have a GROUP BY or HAVING clause

Views

Advantages	Disadvantages
Data independence	Update restriction
Currency	Structure restriction
Improved security	Performance
Reduced Complexity	
Convenience	
Customization	
Data integrity	

Backup and Recovery

- Backup: the process of periodically copying the database and log file (possibly programs) to offline storage media
  - A DBMS should provide backup facilities to assist with the recovery of a database following failure
  - It is advised to make backup copies of the database and log file at regular intervals, as well as ensure that the copies are in a secure location
  - In the event of a failure that renders the database unusable, the backup copy and the details captured in the log file are used to restore the database to the latest possible consistent state

Integrity

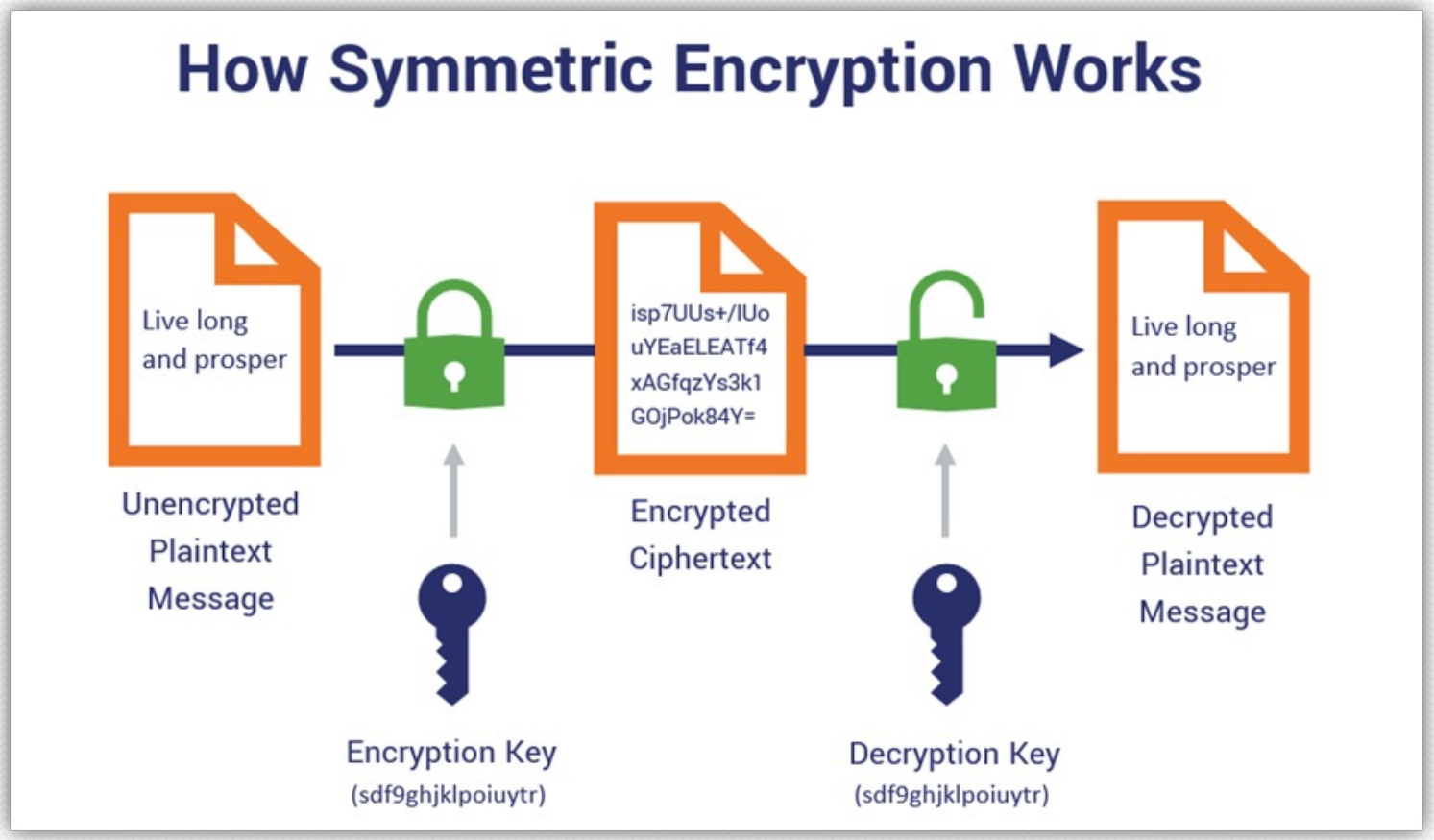
- Integrity constraints contribute to maintaining a secure database system by preventing data from becoming invalid, and hence giving misleading or incorrect results
  - Primary key constraint
  - Foreign key constraint
  - Domain constraint
  - NULL / NOT NULL
  - ….

Encryption

- The encoding of the data by a special algorithm that renders the data unreadable by any program without the decryption key

- Cryptosystem: to transmit data securely over insecure networks
  - An **encryption key** to encrypt the data (plaintext)
  - An **encryption algorithm**, along with the encryption key, transforms plaintext into **ciphertext**
  - A **decryption key** to decrypt the **ciphertext**
  - A **decryption algorithm**, along with the the decryption key, transforms **ciphertext** back into plaintext

Encryption


How Symmetric Encryption Works
Live long and prosper
Unencrypted Plaintext Message
Encryption Key
(sdf9ghjklpoiuytr)
isp7UUs+/lUo
uYEaELEATf4
xAGfqzYs3k1
GOjPok84Y=
Encrypted Ciphertext
Decryption Key
(sdf9ghjklpoiuytr)
Live long and prosper
Decrypted Plaintext Message

RAID

- Hardware that the DBMS is running on must be **fault-tolerant**
  - The DBMS should continue to operate even if one of the hardware components fails
  - Disk drives are the most vulnerable hardware component
- **RAID**
  - **Redundant Array of Independent Disks**
  - **Redundant Array of Inexpensive Disks**
  - Large disk array comprised of an arrangement of several independent disks that are organized to
    - improve reliability
    - increase performance

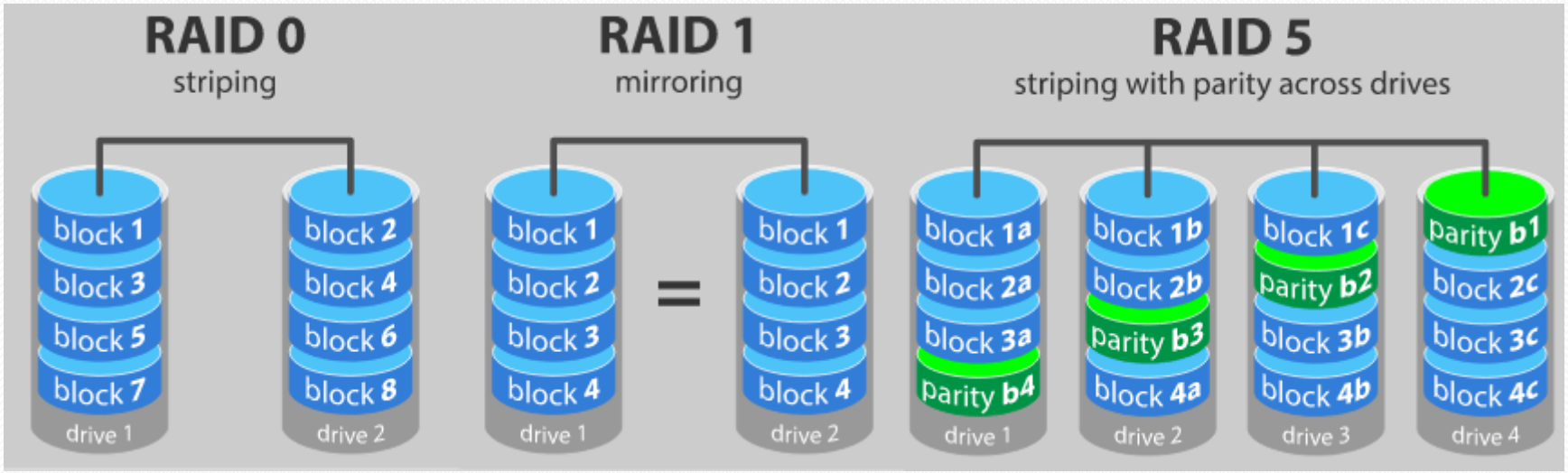
RAID

- Improve the reliability
  - Storing redundant information across the disks using a **parity** scheme or an **error-correcting** scheme
- Improve the performance through **data striping**
  - Data is segmented into equal-size partitions, which are transparently distributed across multiple disks

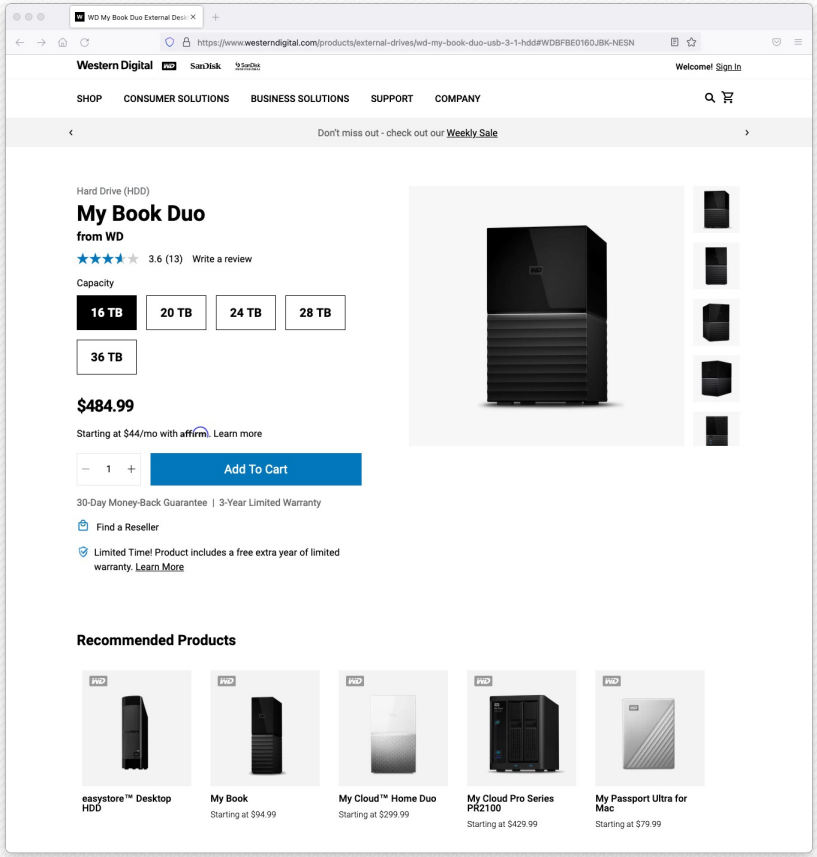
RAID

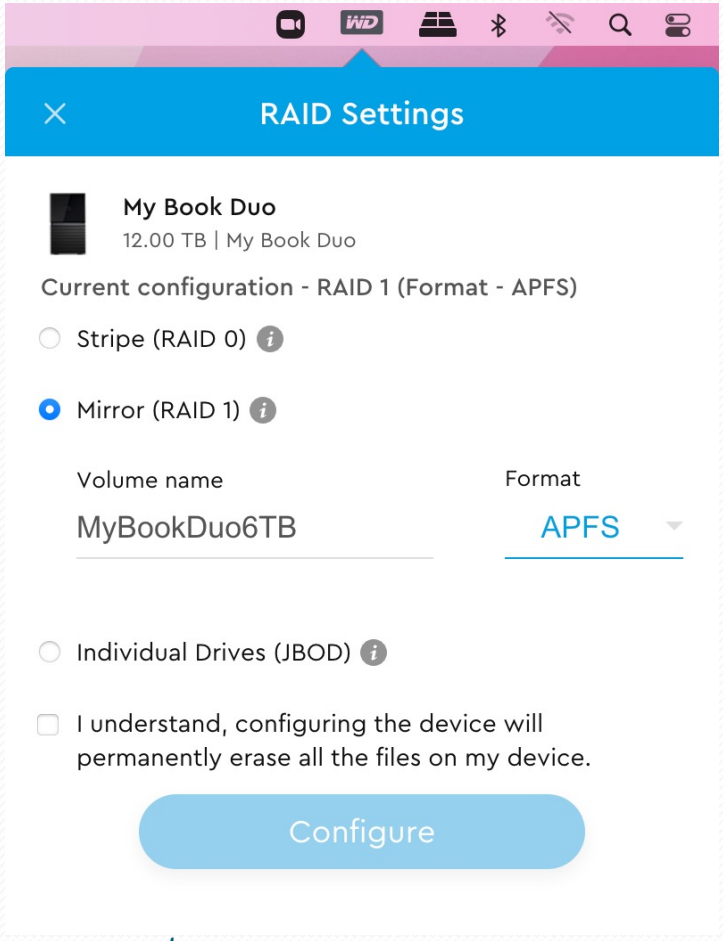
RAID Level	Description	Disk Number	Pros	Cons	Data Security
RAID 0	Stripped disks	Requires 2+ drives. Data are split into blocks that get written across all drives.	Large size; Superior I/O performance	No redundancy; Not fault-tolerant	Loss of any drive will cause total loss of data
RAID 1	Mirrored disks	Requires at least 2 drives. Data are stored twice by writing them to both drives.	Excellent read speed	Effective storage capacity is half of the total drive capacity	One drive fails, the other drive keeps working and with no data loss
RAID 5	Stripped disks + distributed parity	Requires 3+ drives. Data are split evenly on all drives.	Increased capacity; Improved read speed	Drive failures Influence throughput; Total capacity is reduced by parity	Single drive fails, and data will need to be rebuilt

RAID


RAID 0
striping
block 1
block 3
block 5
block 7
drive 1
block 2
block 4
block 6
block 8
drive 2
RAID 1
mirroring
block 1
block 2
block 3
block 4
drive 1
=
block 1
block 2
block 3
block 4
drive 2
RAID 5
striping with parity across drives
block 1a
block 2a
block 3a
parity b4
drive 1
block 1b
block 2b
parity b3
block 4a
drive 2
block 1c
parity b2
block 3b
block 4b
drive 3
parity b1
block 2c
block 3c
block 4c
drive 4

RAID


Western Digital
SanDisk
Welcome! Sign In
SHOP
CONSUMER SOLUTIONS
BUSINESS SOLUTIONS
SUPPORT
COMPANY
Don't miss out - check out our Weekly Sale
Hard Drive (HDD)
My Book Duo
from WD
3.6 (13) Write a review
Capacity
16 TB
20 TB
24 TB
28 TB
36 TB
$484.99
Starting at $44/mo with affirm. Learn more
Add To Cart
30-Day Money-Back Guarantee | 3-Year Limited Warranty
Find a Reseller
Limited Time! Product includes a free extra year of limited warranty. Learn More
Recommended Products
easystore™ Desktop HDD
My Book
Starting at $94.99
My Cloud™ Home Duo
Starting at $299.99
My Cloud Pro Series PR2100
Starting at $429.99
My Passport Ultra for Mac
Starting at $79.99


RAID Settings
My Book Duo
12.00 TB | My Book Duo
Current configuration - RAID 1 (Format - APFS)
Stripe (RAID 0)
Mirror (RAID 1)
Volume name
MyBookDuo6TB
Format
APFS
Individual Drives (JBOD)
I understand, configuring the device will permanently erase all the files on my device.
Configure

Acknowledgements

- Allied Solutions
  - https://www.alliedsolutions.net/resources/allied-insights/2017/10/03/3-ways-to-manage-equifax-breach
- Security Boulevard
  - https://securityboulevard.com/2020/11/symmetric-encryption-algorithms-live-long-encrypt/
- Wikipedia
  - https://en.wikipedia.org/wiki/RAID
- AOMEI Partition Assistant
  - https://www.diskpart.com/articles/raid-hard-drive.html
- Western Digital
  - https://www.westerndigital.com/solutions/raid