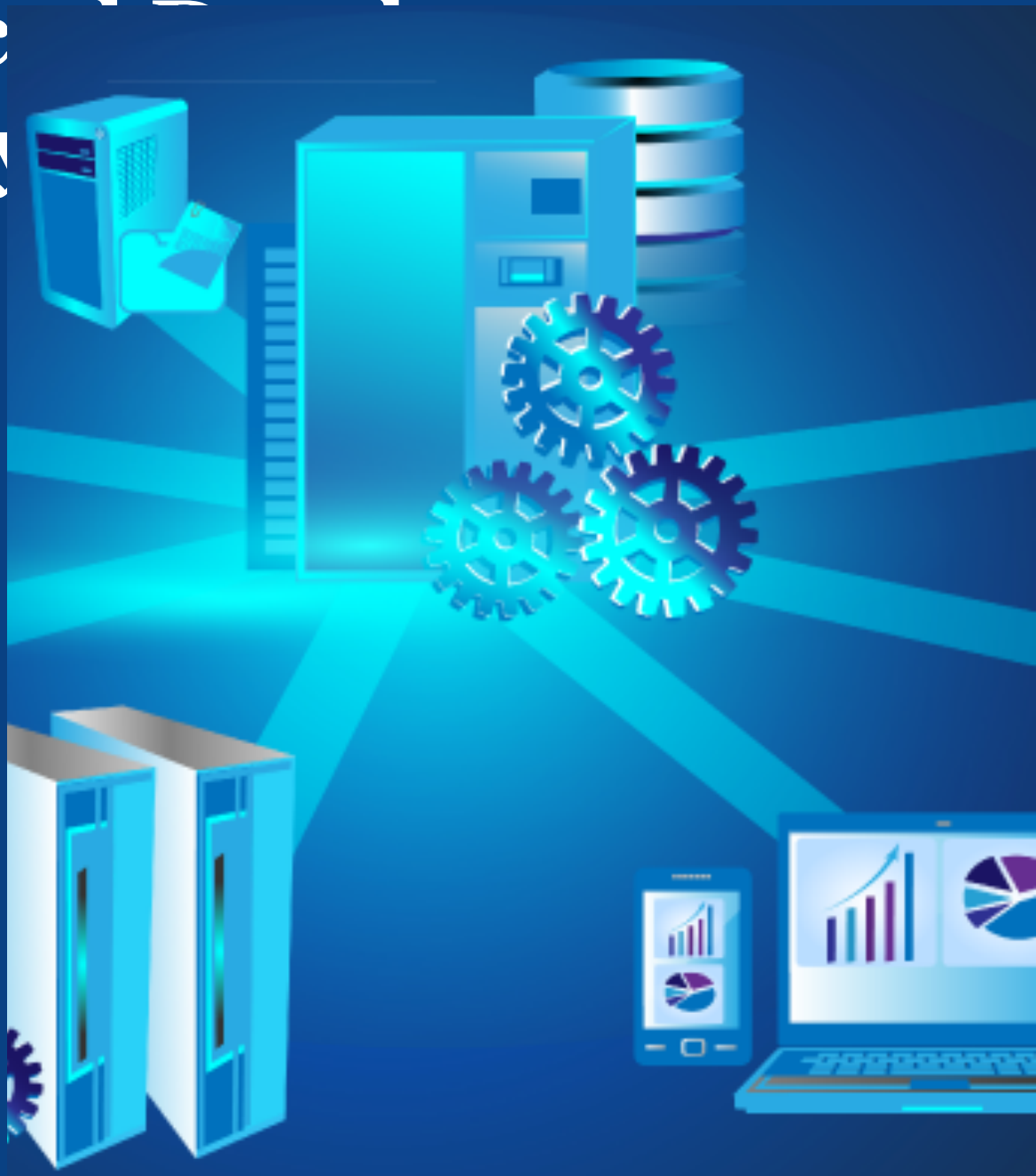
COMP90050: Advanc… Sy…

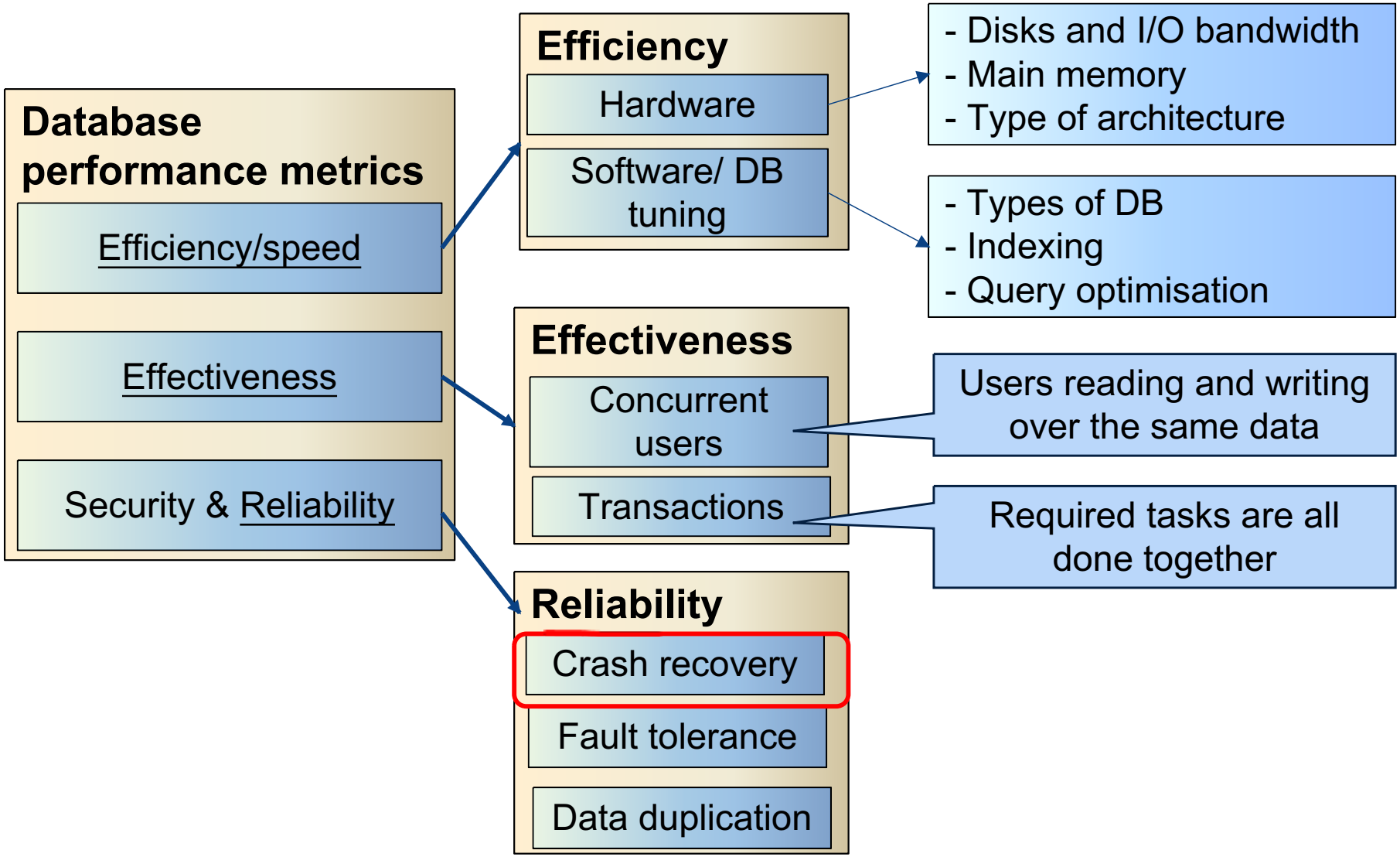


Lecturer: Farhana Choudhury (PhD)

Backups

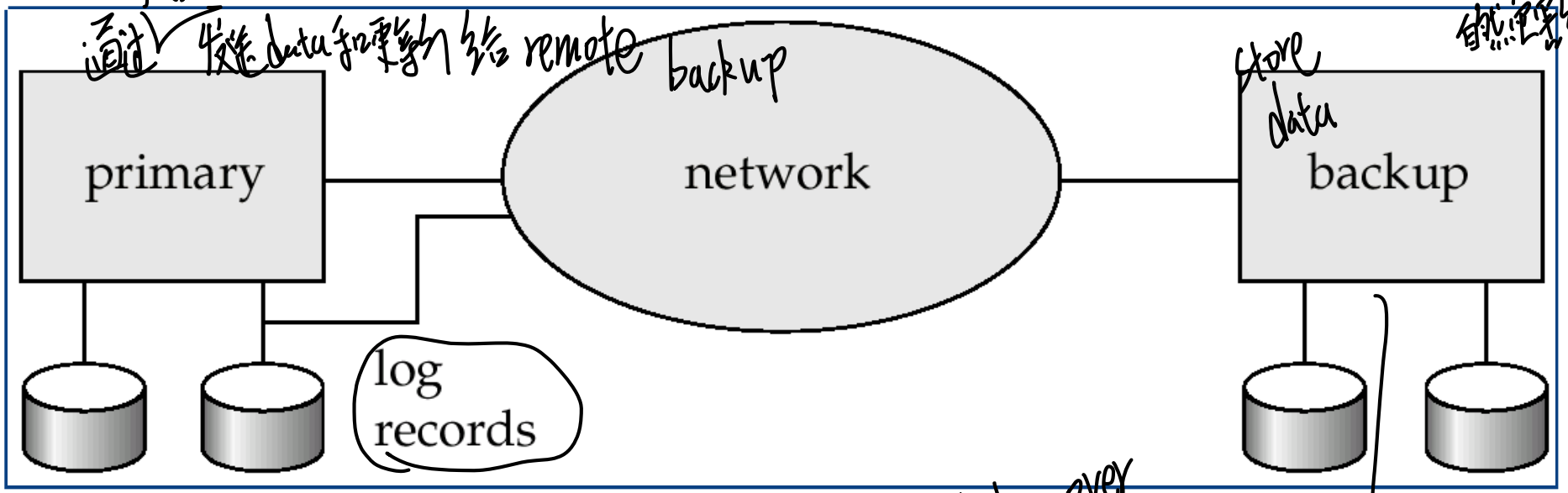
Week 12 part 1

Core Concepts of Database management system



Other Considerations: Remote Backup Systems

Remote backup systems provide high availability by allowing transaction processing to continue even if the primary site is destroyed



when backup need to take over: primary failed and cannot recover from crash.

Detection phase → If there's no communiction between primary & backup

Two potential reasons ① The network got interrupted

② primary failed (how can we distinguish from ①?)

We set network redundency → more than one network links between these two

communication (heart beat) → communicate regularly in a frequency/interval

↓

all new changes. data, logs.

if heartbeat communication 中断 ⇒ 认为 primary failed ⇒ time for backup to take control.

phase: Transfer of control → 1. perform recovery (用 backup 中存储的 log, data 来 redo committed transactions and roll back the incomplete transactions (make seamless transfer of control) 2. backup → new primary (all will be performed using this).

☆How remote backup, takeover process can be performed effeciently?

① process the log records (less things to recovery, will be faster) ⇒☆redo the transcation☆

☆but not undo ⇒ 有可能primary site 正在做这些没(commited) that has not commited

的transaction. ⇒ 要是 backup undo ⇒ primary site finish ⇒ 刻log ⇒ the transaction is undone correctly ☆

② Create checkpoints ⇒ backup creat its own checkpoints ⇒ 把things need to be down record. for recovery phase ⇒ (reduce the things need to recover)

③ Hot spell confederation

① Continous redo ⇒ Whenever the redo log arrives, the backup will apply that update.

② roll back only when failure detected.

unfinished transaction

ensure high durability of ~~copy~~ data using backup.

① ⇒ primary site can delay commits of transaction till the update is logged also at the backup

⇒ send log to backup, although cause delay in transaction commit.

可选不同的setting ⇒ 對应用需求. ① One-safe: The transaction can commit as soon as its commit log report is written at the primary site.

pro: fast ⇒ 无需等 backups (log to be stored)

cons: If crash happens, log may not reach backup at that time ⇒ possible lost update.

② two-very-safe: transaction can commit when log is written both at primary and backup. pros: more safe. durable. robust to any failure. (not lost any)

cons slower (each committed transaction has to wait two committed

logs to be stored

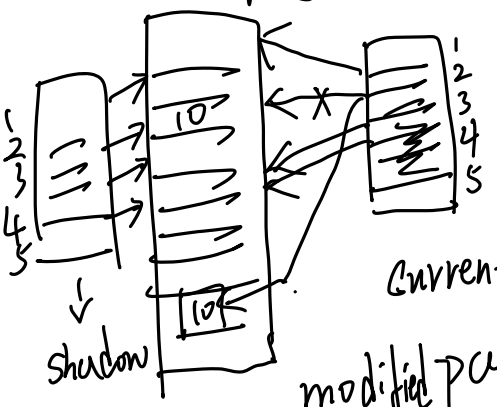
③ Two-safe: 3.1 when both active → two-very safe (frequent communication, don't have to wait much long time)
sides

3.2 when only primary is active ⇒ one safe

---

next strategy for recovery: Shadow paging: maintain 2 page table during the life time of transactions

page on disk



if modify page, make a copy page on disk

Change copy value only during execution

Then at commit time: current PT go to disk.

modified pages go to disk., Shadow PT point to the checkpoint (before what page is transcation commited) current PT → now becomes shadow PT.

finish the transaction commit.

→ Before the commits of transaction, crash happens. Shadow page table maintained on all status and info before the transaction start.

pros: Shadow PT can be used directly to recover before crash recovery

No overhead of writing any log records

Cons: every time copy entire PT as shadow PT is expensive. when PT is (large table)

high commit overhead each commit time ⇒ the updated page needs to be flushed to disk.

Remote Backup Systems Contd.

Detection of failure: Backup site **must detect when primary site has failed**

- To distinguish primary site failure from link failure, **maintain several communication links between the primary and the remote** backup
- **Use heart-beat messages**

Transfer of control:

- To take over control, **backup site first perform recovery using its copy of the database and all the log records** it has received from primary
  - Thus, completed transactions are redone and incomplete transactions are rolled back
- When the backup site takes over processing **it becomes the new primary**

Remote Backup Systems Contd.

Time to recover:

- To reduce delay in takeover, **backup site periodically processes the redo log** records
- In effect, it **performs a checkpoint, and can then delete earlier parts of the log**

Hot-Spare configuration permits very fast takeover:

- **Backup continually processes redo log record as they arrive** , applying the updates locally
- When failure of the primary is detected the backup rolls back incomplete transactions, and is **ready to process new transactions**

Remote Backup Systems Contd.

To ensure durability of updates - delay transaction commit until update is logged at backup

But we can avoid this delay by permitting lower degrees of durability

One-safe: commit as soon as transaction's commit log record is written at primary

- Problem: updates may not arrive at backup before it takes over.

Two-very-safe: commit when transaction's commit log record is written at primary and backup

- Reduces availability since transactions cannot commit if either site fails.

Two-safe: proceed as in two-very-safe if both primary and backup are active. If only the primary is active, the transaction commits as soon as is commit log record is written at the primary

- Better availability than two-very-safe; avoids problem of lost transactions in one -safe.

Alternative to Logs: Shadow Paging

Shadow paging is an alternative to log-based recovery

Idea: **maintain *two* pageTables during the lifetime of a transaction** –the **current page table**, and the **shadow page table**

Store the **shadow page table in nonvolatile storage** , such that state of the database prior to transaction execution may be recovered

- Shadow page table is never modified during execution

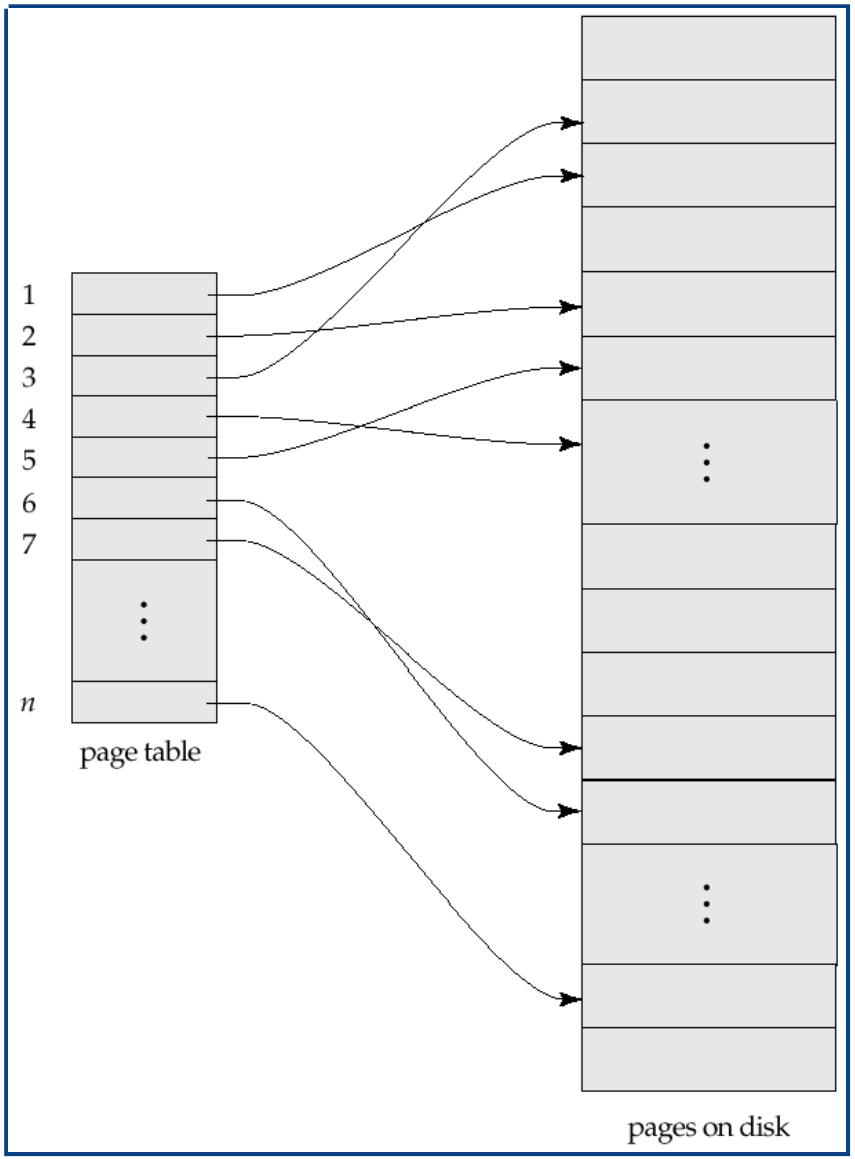
Alternative to Logs: Shadow Paging

To start with, both the page tables are identical. **Only the current page table is used for data item accesses** during execution of the transaction

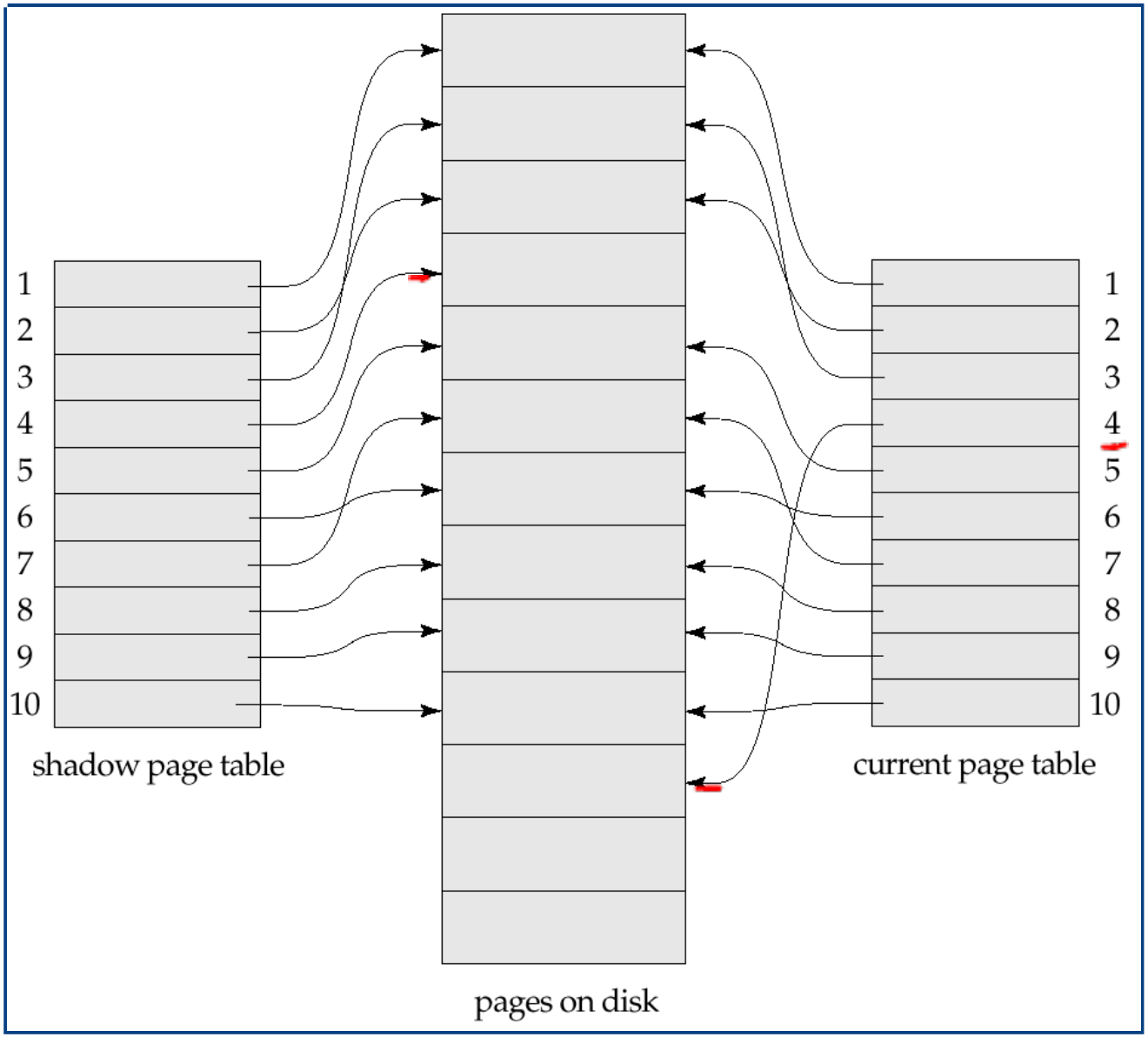
Whenever **any page is about to be written** :

- A **copy of this page is made onto an unused page**
- The **current page table is then made to point to the copy**
- The **update is performed on the copy**

Sample Page Table



Example of Shadow Paging



Shadow and current page tables after write to page 4

Shadow Paging Contd.

To commit a transaction：

1. **Flush all modified pages in main memory to disk**
2. **Output current page table to disk**
3. **Make the current page table the new shadow page table**，as follows:
   - keep a pointer to the shadow page table at a fixed (known) location on disk.
   - to make the current page table the new shadow page table, simply update the pointer to point to current page table on disk.

Once pointer to shadow page table has been written, transaction is committed.

Shadow Paging Contd.

No recovery is needed after a crash — new transactions can start right away, using the shadow page table.

Advantages of shadow-paging over log-based schemes:

- No overhead of writing log records
- Recovery is trivial

Disadvantages:

- Copying the entire page table is very expensive when the page table is large
- Pages not pointed to from current/shadow page table should be freed (garbage collected)
- Commit overhead is high - flush every updated page, and page table
- Data gets fragmented (related pages get separated on disk)
- Hard to extend algorithm to allow transactions to run concurrently

Backups and crash recovery in practice

Strategy plan based on:

- Goals and requirement of your organization/task
- The nature of your data and usage pattern
- Constraint on resources

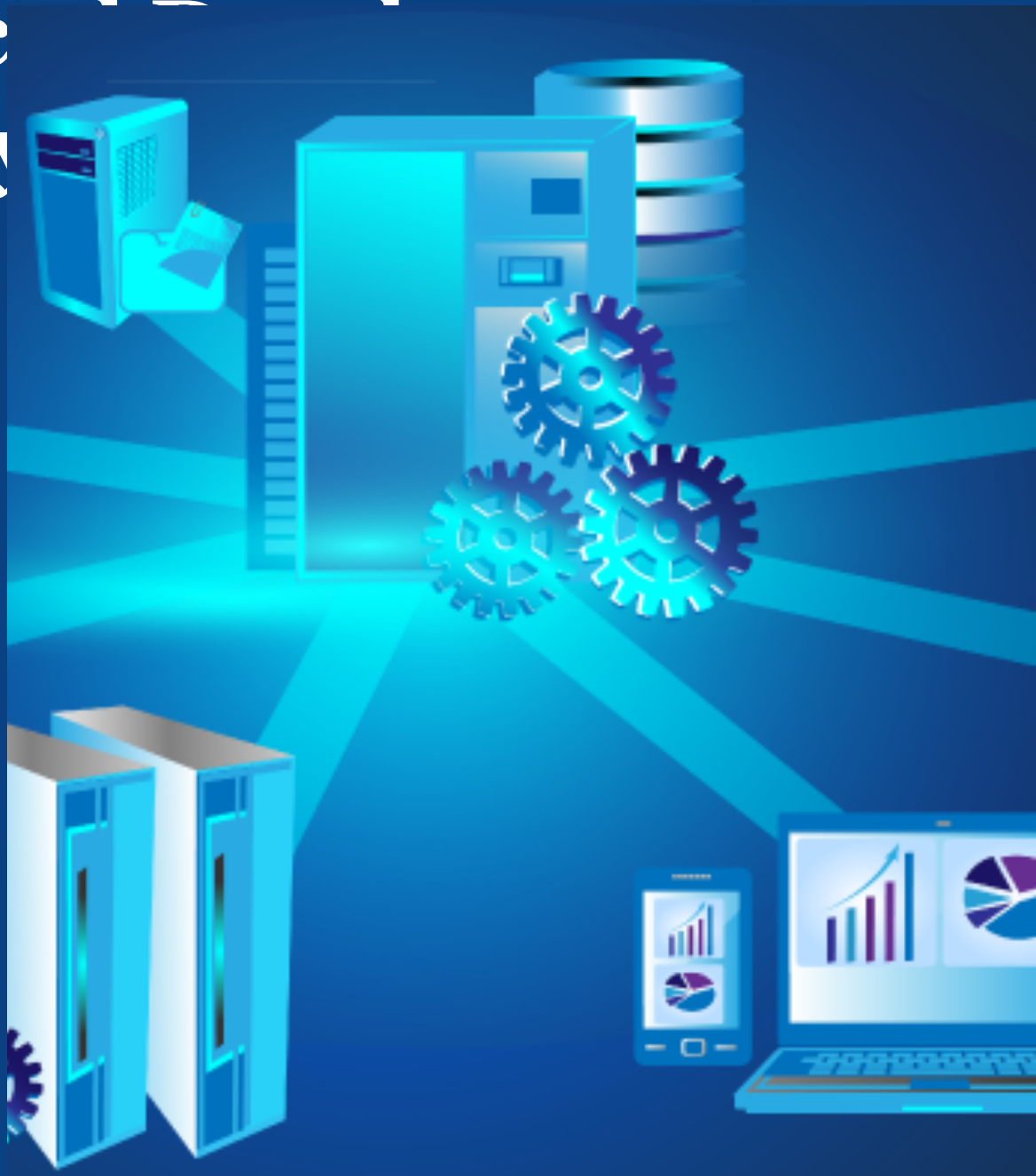
Design backup strategy:

- Full disk backup vs partial - Are changes likely to occur in only a small part of the database or in a large part of the database?
- How frequently data changes
  - If frequent: use differential backup that captures only the changes since the last full database backup 越少越省耗时
- Space requirement of the backups – depends on the resource
- Multiple past instances of backup – useful if point-in-time recovery is needed

backup 多个时间点的 log、data

Resource: https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/back-up-and-restore-of-sql-server-databases?view=sql-server-ver16

COMP90050: Advanc… S…



Lecturer: Farhana Choudhury (PhD)

Data warehousing

Week 12 part 2

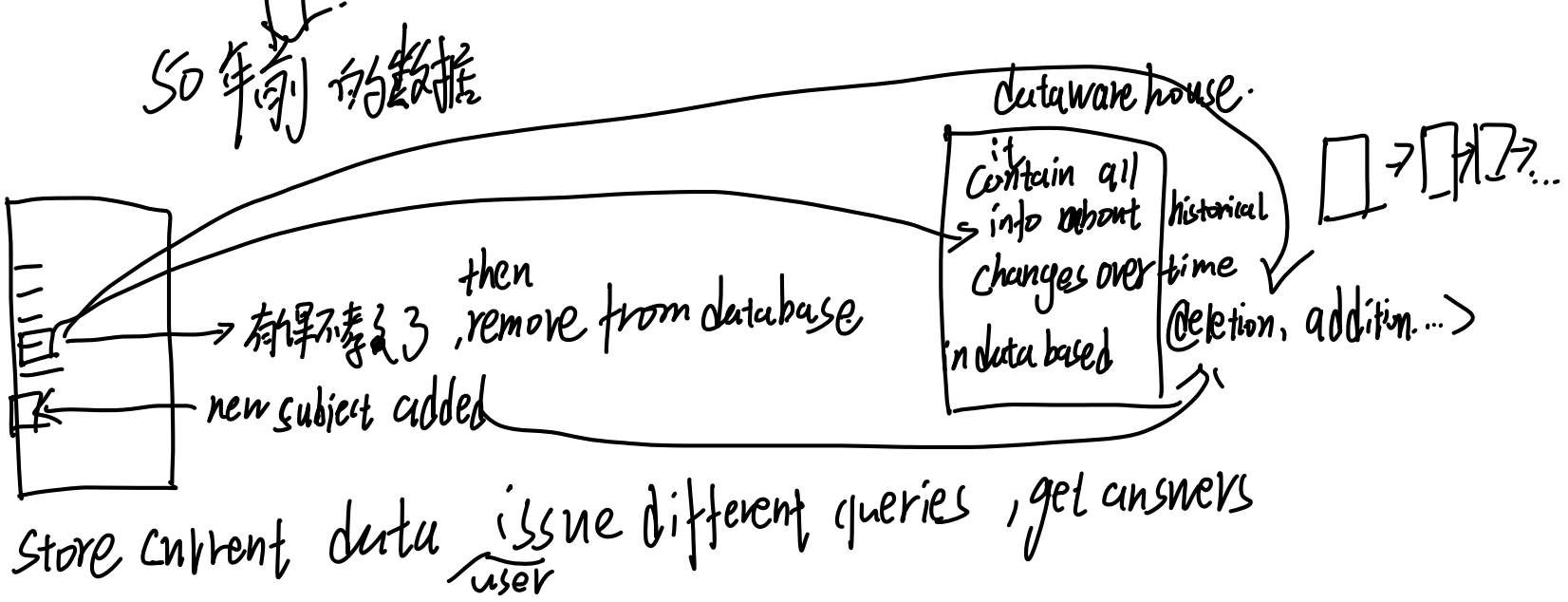
Specialised databases: Data Warehousing

Data Warehousing

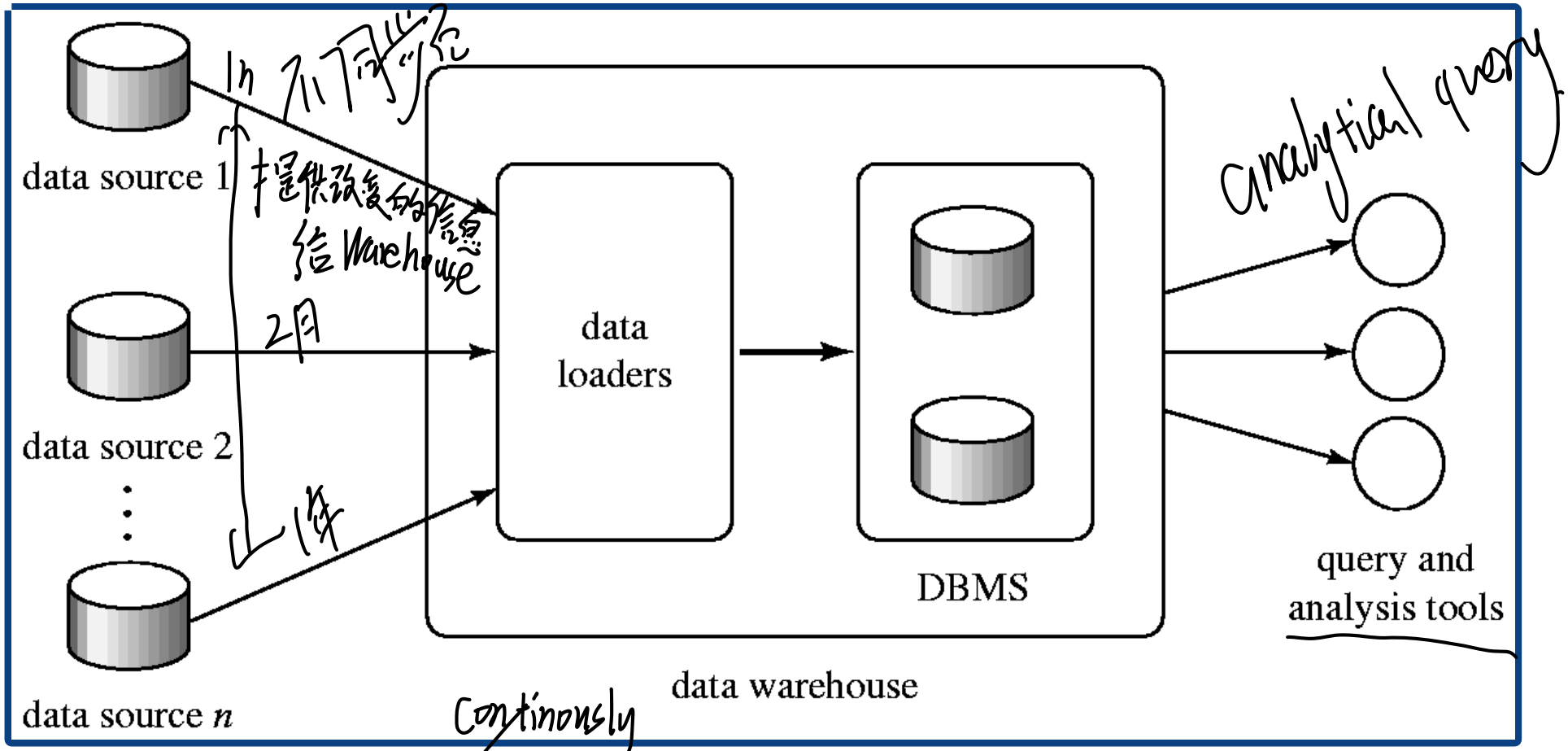
Corporate **decision making requires** a unified view of all organizational data, including **historical data**

A **data warehouse** is a repository (archive) of information gathered from multiple sources, stored under a unified schema, for analytics and reporting purposes

- Greatly simplifies querying, permits study of historical trends
- Shifts decision support query load away from transaction processing systems



Data Warehousing



① source periodically provide data to warehouse. Continously

② warehouse periodically ask for new data from data sources. Continously

Design Issues

When and how to gather data:

- **Source driven architecture**: data sources transmit new information to warehouse, either continuously or periodically (e.g. at night)
- **Destination driven architecture**: warehouse periodically requests new information from data sources

- Keeping warehouse exactly synchronized with data sources (e.g., **using two-phase commit) is too expensive**
  - Usually **OK to have slightly out-of-date** data at warehouse
  - Data/updates are periodically downloaded form online transaction processing (**OLTP**) systems (most of the DBMS work we have seen so far)

More Warehouse Design Issues

What schema to use

- Depends on purpose
- Schema integration

Data cleansing

- E.g. correct mistakes in addresses (misspellings, zip code errors)
- *Merge* address lists from different sources and *purge* duplicates

How to propagate updates

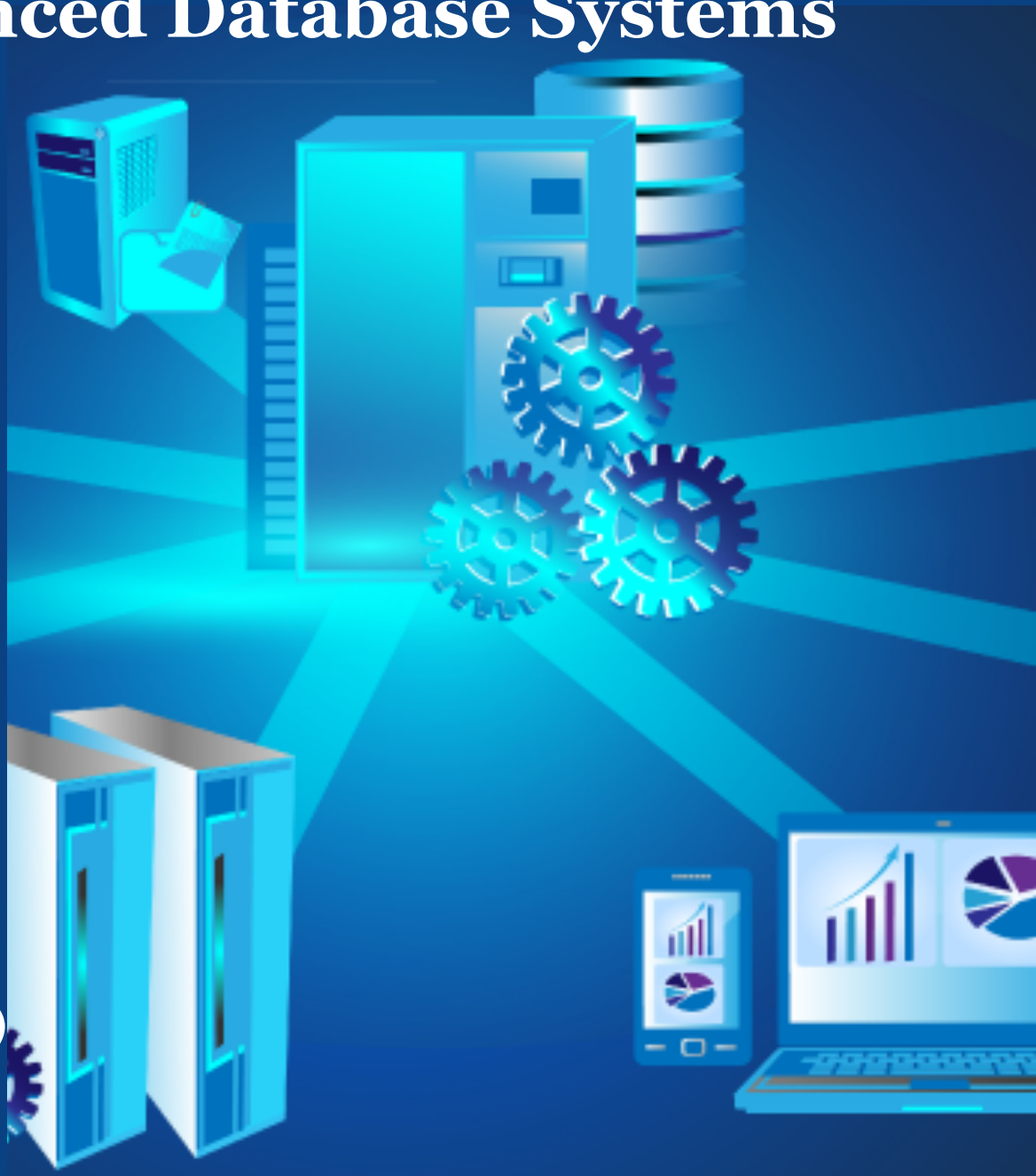
- The data stored in a data warehouse is documented with an element of time, either explicitly or implicitly

What data to summarize

- Raw data may be too large to store
- *Aggregate values (totals/subtotals) often suffice*
- Queries on raw data can often be transformed by query optimizer to use aggregate values

THE UNIVERSITY OF MELBOURNE

COMP90050 Advanced Database Systems



Semester 2, 2024

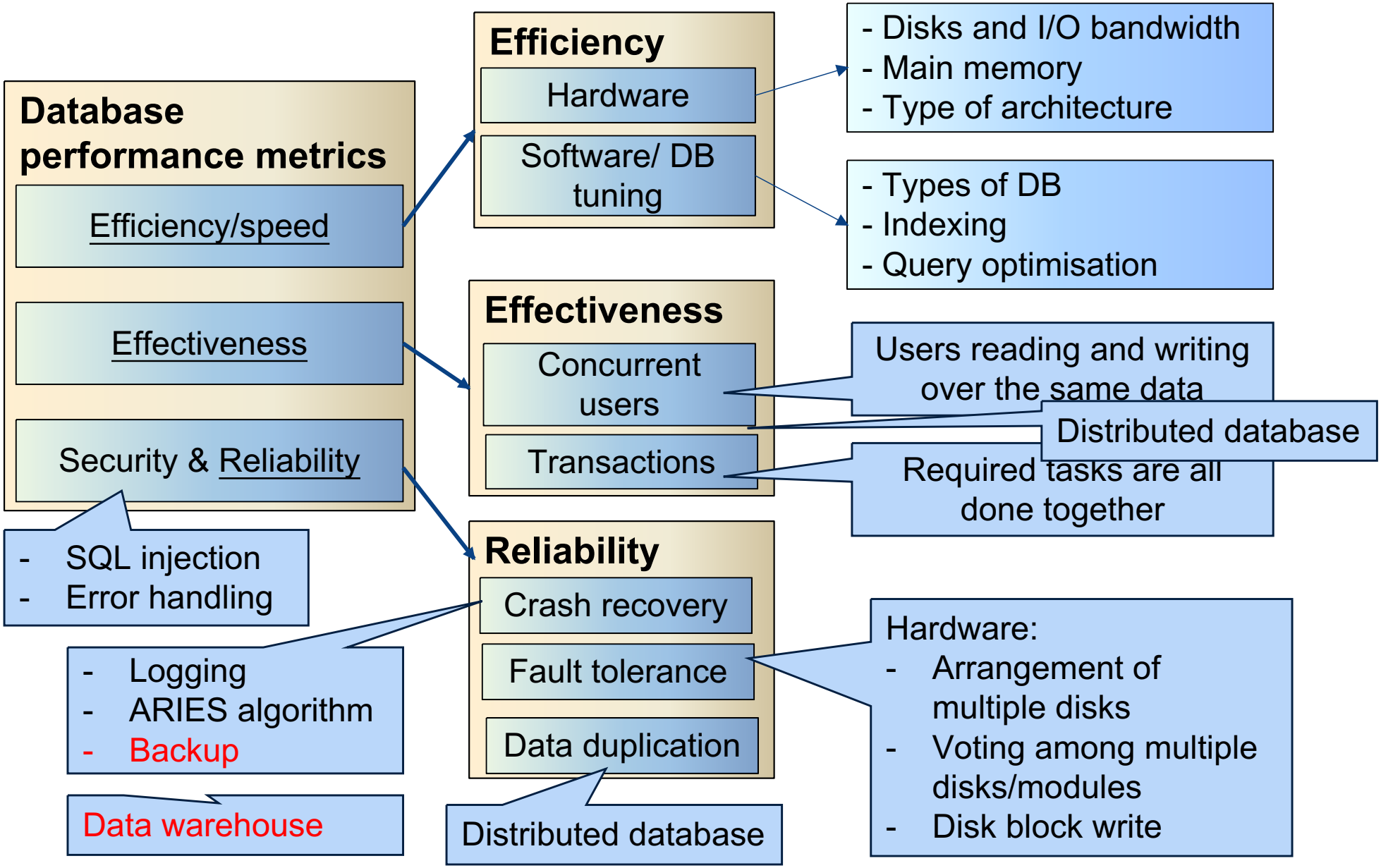
Lecturer: Farhana Choudhury (PhD)

Live lecture – Week 12

Today's topics

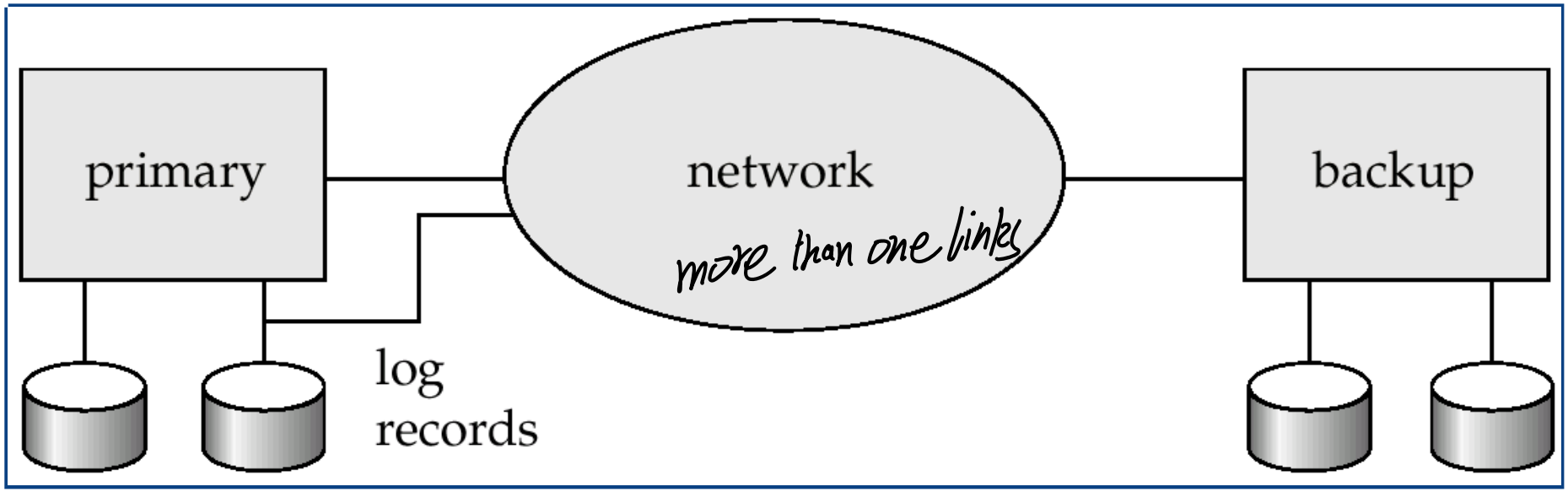
- Review this week's topics
- Recap of the subject
- Final exam preparation

Core Concepts of Database management system



Remote Backup Systems

Remote backup systems provide high availability by allowing transaction processing to continue even if the primary site is destroyed



network redundancy is important to minimize the risk of failure which prevents the communication between primary and backup in network;

We have also seen backups and recovery in practice

Strategy plan based on:

- Goals and requirement of your organization/task
- The nature of your data and usage pattern
- Constraint on resources

reduce overhead

⇓

Design backup strategy:

- Full disk backup vs partial
- How frequently data changes
  - If frequent: use differential backup

if data has frequently changed part(其他不变)
just copy changes. not all part.

- Space requirement of the backups – depends on the resource
- Multiple past instances of backup – useful if point-in-time recovery is needed

只copy 和上周
只copy 和上月
→ 很容易 revert back to point-in-time
What the state was at that point

Crash recovery model (including backups) in practice

Choose the right recovery model for your application

Types of recovery models in MS SQL server:

a. Simple: No logs, but has backups. Recovery is done from the last backup

b. Full: Uses logs plus backups, regular checkpoints

c. Bulk logged: Logs are not maintained for each individual writes, but for multiple writes together* (reduce the number of logs generate every time) over head.

*For some operations

Crash recovery model (including backups) in practice

Choose the right recovery model for your application

1. Find the current recovery model

```
SELECT name, recovery_model_desc
FROM sys.databases
WHERE name = 'model' ;
```

checkpoint 1min V.S checkpoint 1h

Tradeoff → frequent checkpoints come with some overhead but safe. long interval checkpoint: crash recovery will cost longer time (how much time we want to spend on recovery if crash happens)

2. Change the model if needed

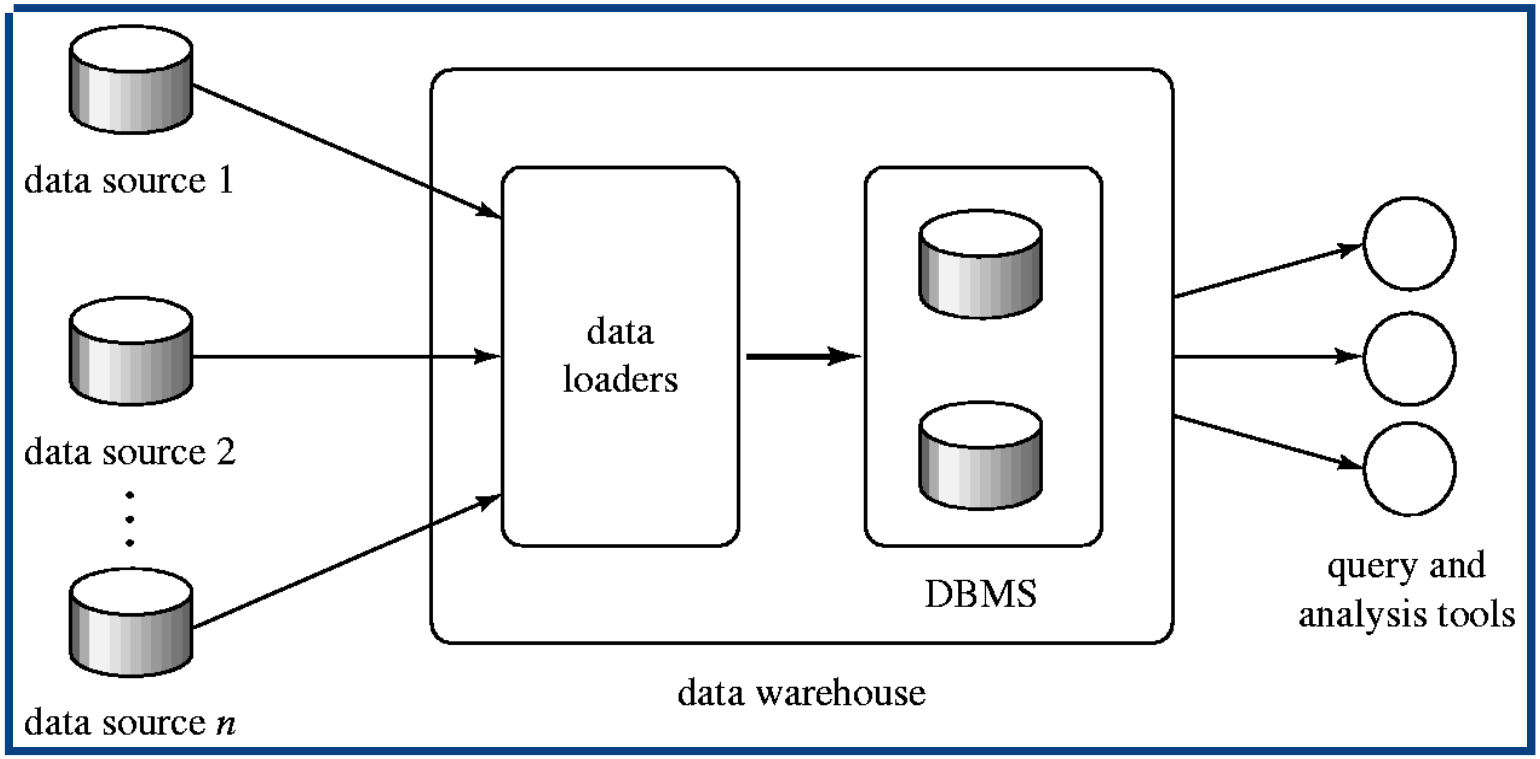
more frequent checkpoint → faster recovery

3. **Checkpoint interval:** Specify target recovery interval, for example, if a crash happens, recover within 1 min. The system will then determine how often it needs to create checkpoints based on that value.

Data warehouse design issues

When and how to gather data: two ways data sources → Warehouse.

- **Source driven architecture**: data sources transmit new information to warehouse, either continuously or periodically (e.g. at night) (frequencies 可不同)
- **Destination driven architecture**: warehouse periodically requests new information from data sources



Data warehouse design issues

In a data warehouse, when should the data sources transmit new information to warehouse frequently, instead of periodically (e.g., at night)?

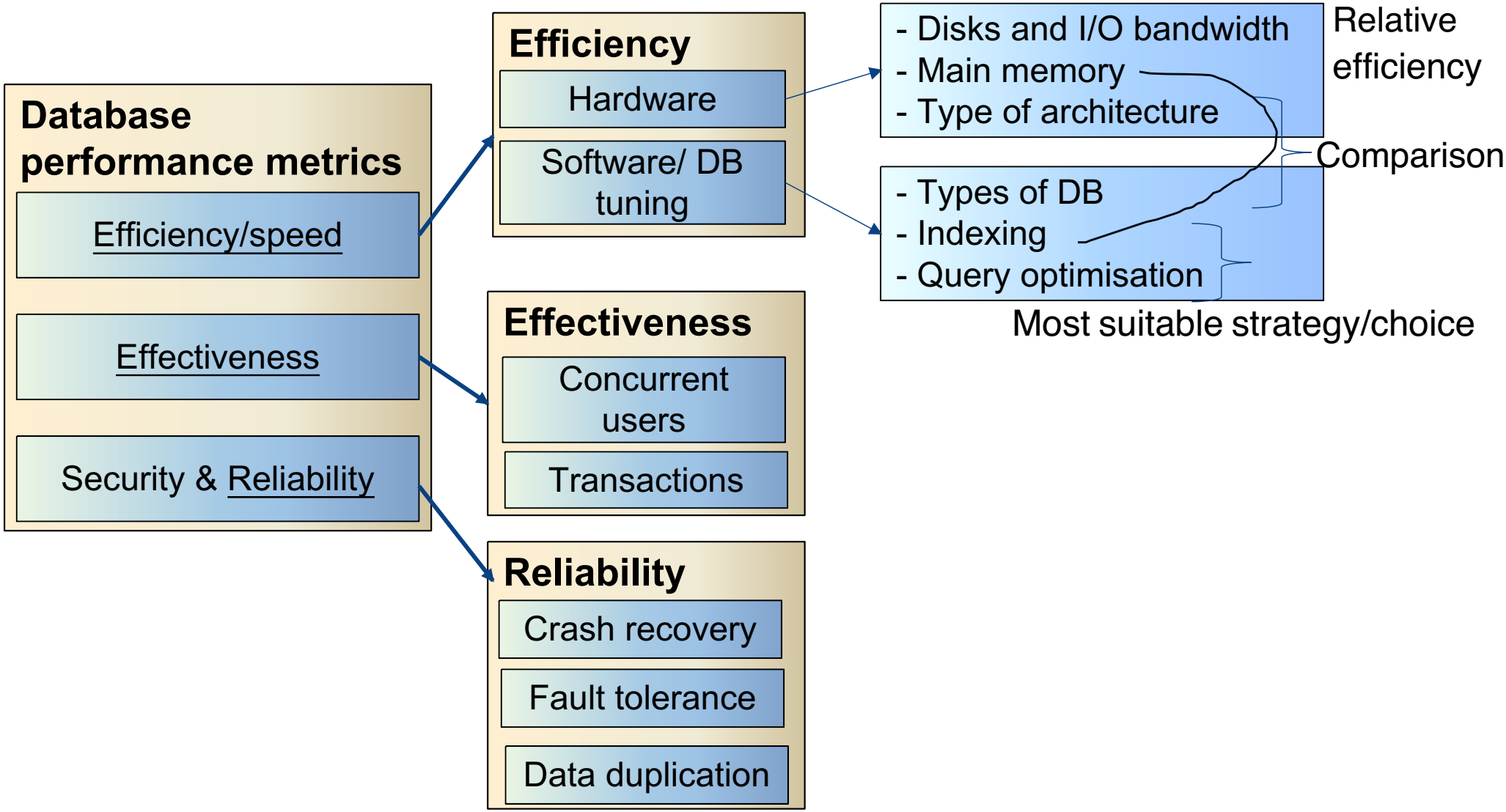
Time for a poll –
Pollev.com/farhanachoud585

Real-time Inventory Management: In e-commerce or retail, real-time inventory data is crucial for operation.

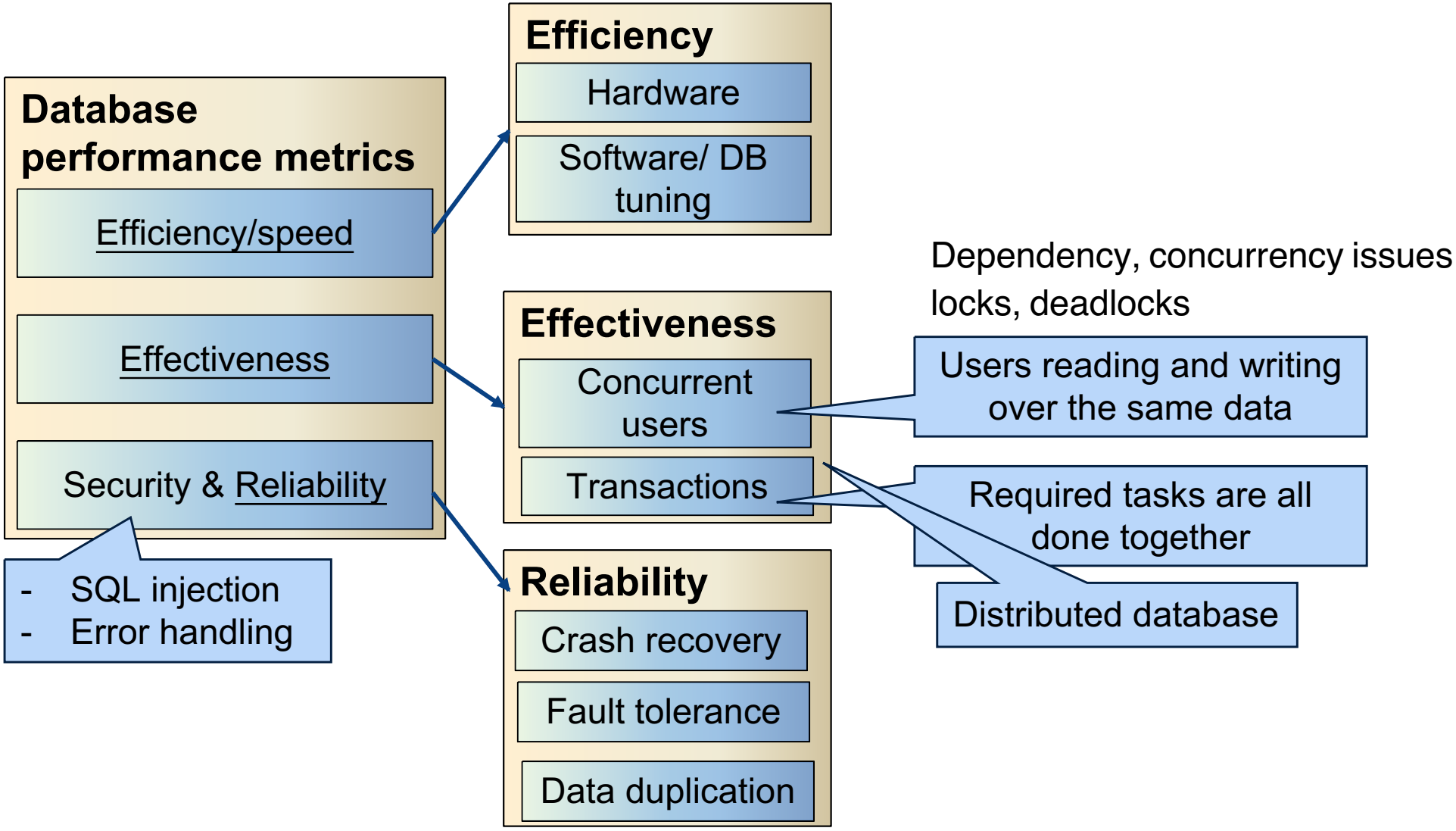
We need to analyze frequently updated inventory information also combined with historical orders to make decision instantly, to prevent oversell or backlog of products. (Instantly / promptly) on increase or decrease inventory

(especially during crisis such as Double 11 or black friday)

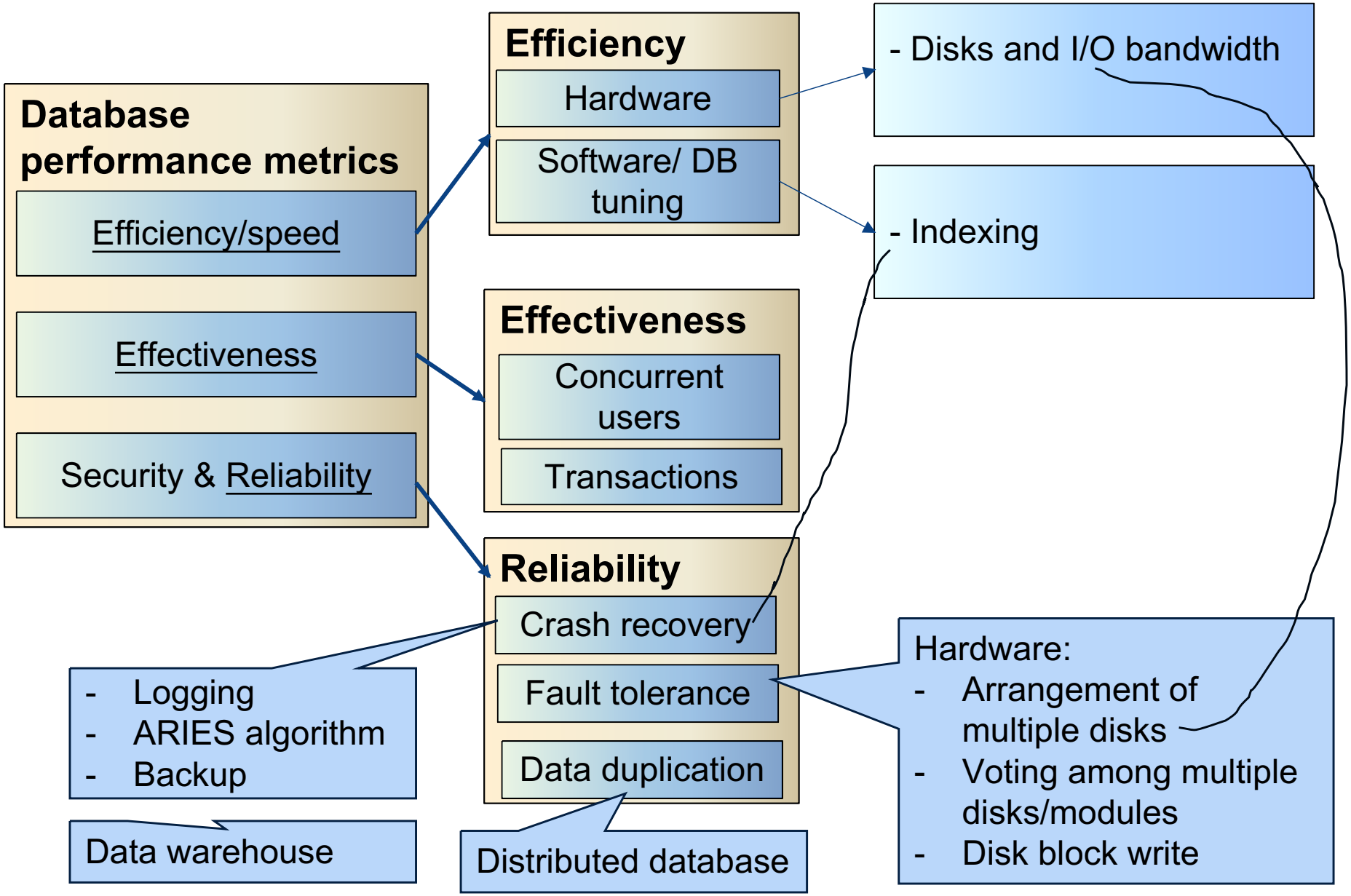
Recap of the subject



Recap of the subject



Recap of the subject



Final exam discussion

- Covering all topics (except lab materials)
- The exam is 2 hours long
- It is worth 50% of your total mark
- It’s on-campus written exam
- It’s open book - Student notes, printed materials, dictionaries - all are permitted
- No need for calculator
- You can use pencil to draw diagrams

Final exam discussion

- Read the questions!
- A question can have multiple parts (a, b, c …)
- Different questions have different marks – answer accordingly

ESS now open

Use the following link to directly access the system: https://unimelb.bluera.com/unimelb

THANK YOU!!!!!