Lecture 1.

Introduction

COMP3278B

Introduction to Database Management Systems

Dr. Ping Luo

Email : pluo@cs.hku.hk

Department of Computer Science, The University of Hong Kong

Acknowledgement: **Dr. Chui Chun Kit**

We are going to learn…

- **Getting started…**
  - What is a database management system (DBMS)?
  - Why is a DBMS important?
  - Why don’t we simply use a file to store all the data?
- **A brief introduction to a DBMS**

Section 1.1

What is a Database?

What is a database?

Three Concepts

- **Data**
  - Some values referring to real world facts.
  - May be in various formats, e.g., text, image, audio file, video file, etc.
- **Database**
  - A large collection of inter-related data.
- **Database management system (DBMS)**
  - DBMS = database(s) + a set of programs that store and access the data.

Database applications

- **Many daily applications involve databases.**
  - Banking

What data are stored in the backend database of an ATM machine?

Customers records | Login Password | Transactions | Account balance | ...

DBMS

Database applications

- **Many daily applications involve databases.**
  - Banking
  - Airline

What data are stored in the backend database of an airline booking system?

Airport information

Aircraft

Flight (date, time, origin, destination…)

Customer record

Booking

DBMS

Database applications

- **Many daily applications involve databases.**
  - Banking
  - Airlines
  - Social network

Member information

Friend list

Friend request

News feed

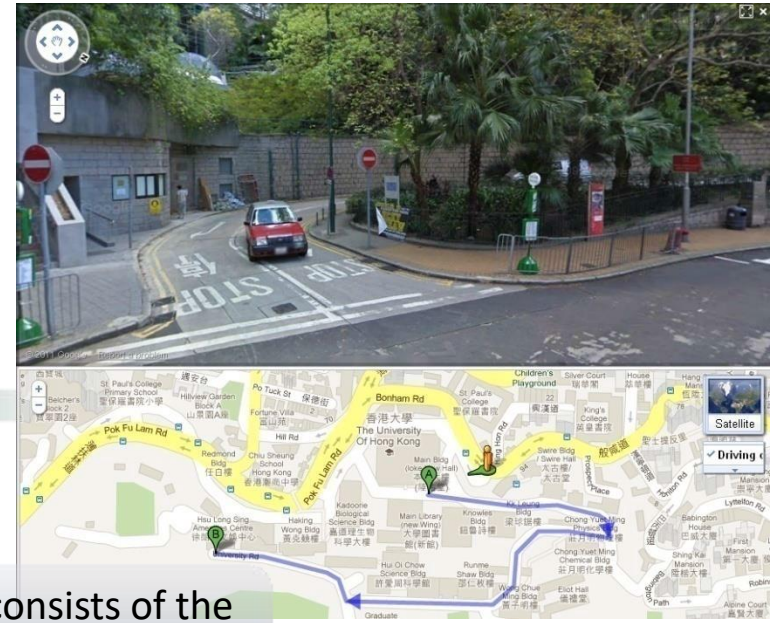
Inbox

Check-in locations

DBMS

Database applications

- **Many daily applications involve databases.**
  - Banking
  - Airlines
  - Social network
  - Online maps



Google Street View

Application program

DBMS

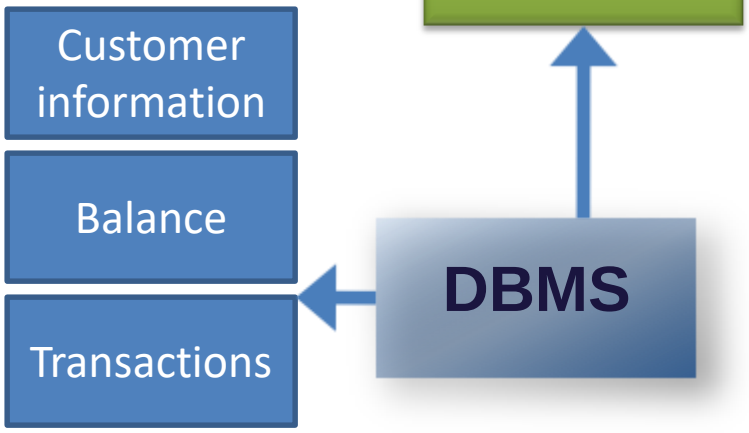
Locations | Roads | Images

An information system can consists of the **data layer** and the **application layer**.

Database applications

- **Many daily applications involve databases.**
  - Banking
  - Airlines
  - Universities
  - Social network
  - Online maps
  - Smart card systems

One database may support many different applications simultaneously.



File system v.s. DBMS

- **In the early days, application programs were built on top of file systems. (e.g., DOS, Windows file explorer)**

Why don't we simply use **a file system** to manage our data?

So that I don't need to spend resources (e.g., $$$) on maintaining a database/ hiring a database administrator…

Boss, let us use a database to organize our data ☺!

File system v.s. DBMS

- **6 drawbacks of storing data in a file system:**
  - **1. Difficulty in accessing data** - May be **inefficient** to locate a piece of information.

⋮

Ping; 12345670; CB326, HKU, Pokfulam Rd; account balance : $100

Jolly; 91234567; CYC 311; HKU, Pokfulam Rd; account balance: $2,000

Yvonne; 31234567; CB 415, HKU, Pokfulam Rd; account balance: $1,000

Savings-account records

If we keep all the savings account records in a text file, I need to scan the file once to retrieve any particular record.

File system v.s. DBMS

- **6 drawbacks of storing DB in file systems:**
  - **2. Data redundancy and inconsistency** - Data got updated in one place but forget to update the other one.

⋮

Ping; 12345670; CB326, HKU, Pokfulam Rd; account balance : $100

Jolly; 91234567; CYC 311; HKU, Pokfulam Rd; account balance: $2,000

Yvonne; 31234567; CB 415, HKU, Pokfulam Rd; account balance: $1,000

Savings-account records

⋮

Ben/ account balance: $100,000/ CYC 314, HKU, Pokfulam Rd/ 21234567/

Ping/ account balance: $4,000/ CB326, The University of Hong Kong, Pokfulam Rd, Hong Kong/ 12345670 /

Kevin/ account balance $20,000/ 2nd Floor … Central/ 62234567/

Checking-account records

Address information duplicated among different files!

File system v.s. DBMS

- **6 drawbacks of storing DB in file systems:**
  - **3. Data isolation** – Because data are scattered in different files, and files may be in **different formats**, writing new programs to retrieve the appropriate data is difficult.

⋮

Ping; 12345670; CB326, HKU, Pokfulam Rd; account balance : $100

Jolly; 91234567; CYC 311; HKU, Pokfulam Rd; account balance: $2,000

Yvonne; 31234567; CB 415, HKU, Pokfulam Rd; account balance: $1,000

Savings-account records

⋮

Ben/ account balance: $100,000/ CYC 314, HKU, Pokfulam Rd/ 21234567/

Ping/ account balance: $4,000/ CB326, The University of Hong Kong, Pokfulam Rd, Hong Kong/ 12345670 /

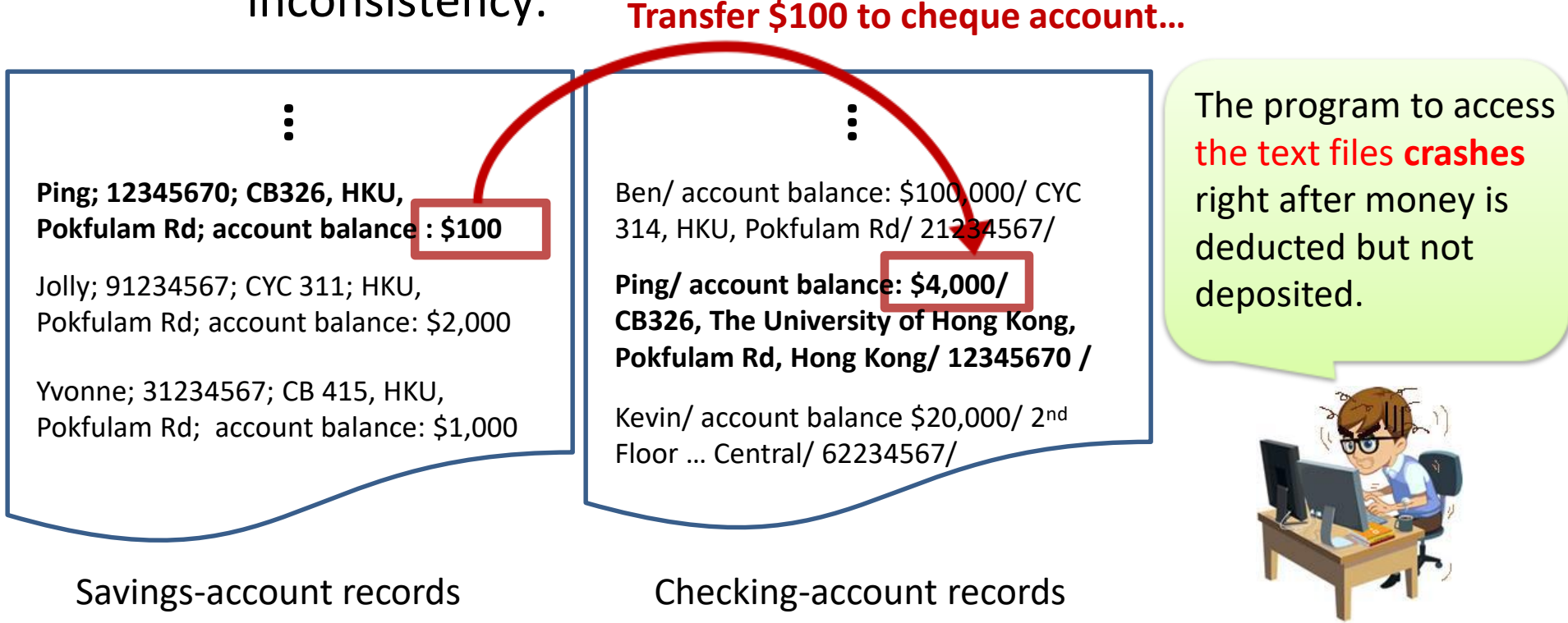
Kevin/ account balance $20,000/ 2nd Floor … Central/ 62234567/

Checking-account records

One file uses ";" to separate fields while the other uses "/", also, the fields are in different orders!

File system v.s. DBMS

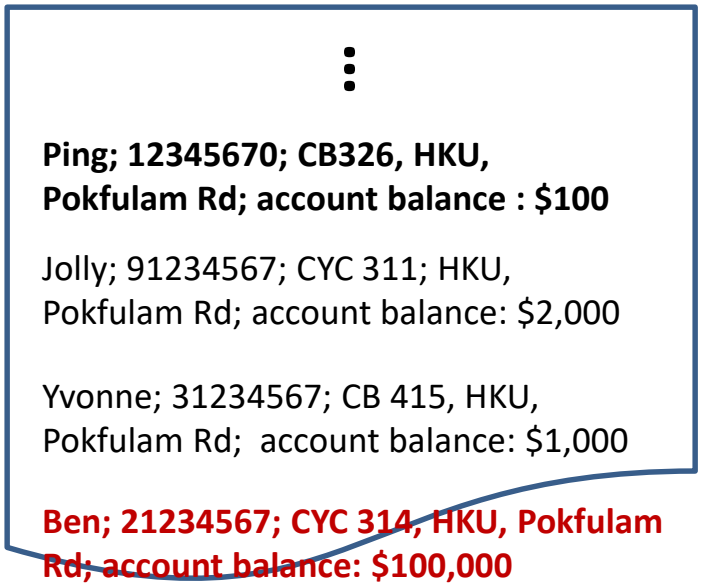
- **6 drawbacks of storing DB in file systems:**
  - **4. Atomicity problems** – It's difficult to ensure that a transfer of money is done completely, leading to failure and inconsistency.



The program to access the text files **crashes** right after money is deducted but not deposited.

File system v.s. DBMS

- **6 drawbacks of storing DB in file systems:**
  - **5. Concurrent access problems** – Inconsistency can occur, e.g., two customers reading and updating a balance at the same time.



Savings-account records

10:00 Ping opens the file.

10:15 Boss opens the same file.

11:00 Boss added one more customer "Ben" to the file and saved the file.

12:00 Ping saves the file and overwrite the file boss saved.

13:00 Boss open the file and found that **"Ben" is not in the file**!

File system v.s. DBMS

- **6 drawbacks of storing DB in file systems:**
  - **6. Integrity problem** – Programmers need to enforce consistency constraints by adding code in many application programs, which is hard to design and manage.

⋮

Ping; 12345670; CB326, HKU, Pokfulam Rd; account balance : $100

Jolly; 91234567; CYC 311; HKU, Pokfulam Rd; account balance: $2,000

Yvonne; 31234567; CB 415, HKU, Pokfulam Rd; account balance: $1,000

Savings-account records

The savings account records should have at least $200 in the account balance.

A checker program has to be added before every withdraw and transfer transactions.

Database Management System

- Big DBMS vendors: Oracle, IBM DB2, Microsoft, SAP Sybase etc.

- Open source DBMS: PostgreSQL, MySQL.

PostgreSQL

To Recap

- **What is a database? Data, Database, and DBMS**
  - Data: some values referring to real world facts.
  - Database: a large collection of inter-related data.
  - DBMS = database(s) + a set of programs that store and access the data.
- **File vs. Database**

1. **Difficulty in accessing data** - May be **inefficient** to locate a piece of information.
2. **Data redundancy and inconsistency** - Data got updated in one place but forget to update the other one.
3. **Data isolation** – Because data are scattered in different files, and files may be in **different formats**, writing new programs to retrieve the appropriate data is difficult.
4. **Atomicity problems** – It's difficult to ensure that a transfer of money is done completely, leading to failure and inconsistency.
5. **Concurrent access problems** – Inconsistency can occur, e.g., two customers reading and updating a balance at the same time.
6. **Integrity problem** – Programmers need to enforce consistency constraints by adding code in many application programs, which is hard to design and manage.

Database Management System

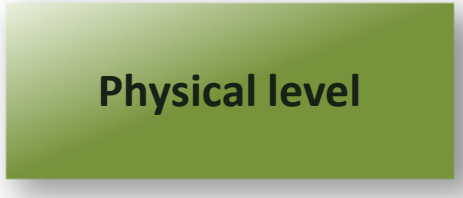
- **A good DBMS aims at handling all problems related to large DB management.**
- **DBMS provides an environment that is both convenient and efficient to use.**
- **Three powerful concepts supported by a DBMS:**
  - 1. Data abstraction.
  - 2. Data modeling.
  - 3. Database languages.

1. Data abstraction

- **Three levels of data abstraction.**

1. Physical level

- Describes **HOW** the data are actually stored in the computer system.
- Describes complex low-level data structures in detail.
- E.g., Whether the data are **compressed** or not? With **what data compression methods**?

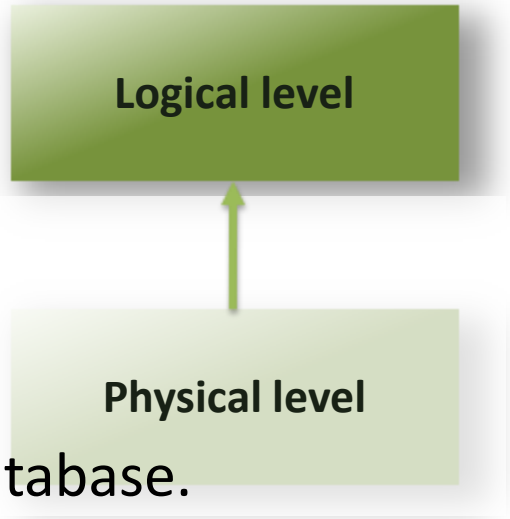


1. Data abstraction

- **Three levels of data abstraction.**

2. Logical level

- Describes **WHAT** data are stored.
- Describes the **relationship** among data.
- **It's independent to physical data!**
  - Adding an index to speed up the access won't affect the data stored in the database.
  - Compressing the data in the physical level also won't affect application programs or user interfaces.
- We use the logical level of abstraction to model the relational information in the database.

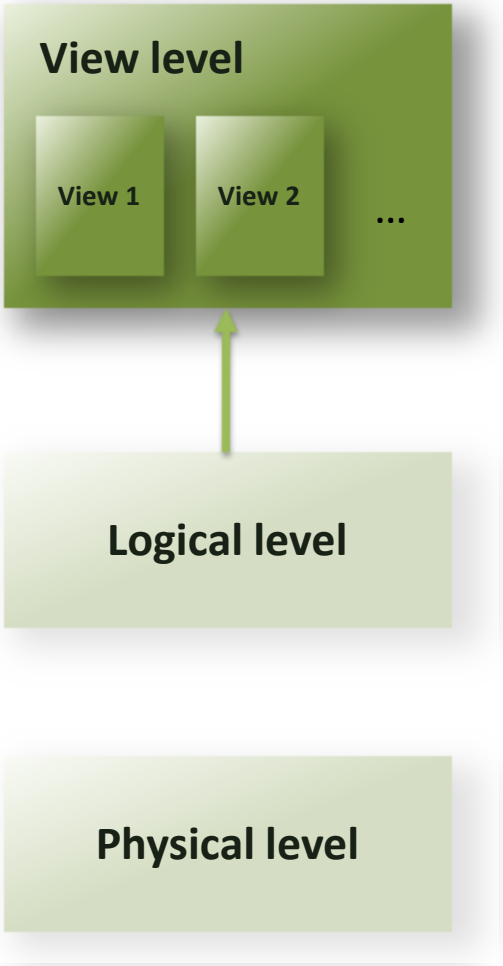


1. Data abstraction

- **Three levels of data abstraction.**

3. View level

- Describes only **part of** the entire DB.
- Many users of the database system do not need to access all information in the database. i.e. different users should have different **view** to the database.
- It simplify users' interaction with the database system.



1. Data abstraction

- **Important concept of Schema and Instance**
  - A **Schema** describes the structure of the database.
  - An **Instance** is the actual data of the database at a particular time.

Schema and **instance** are analog to **type** and **value** in programming.

```
customer {
    string customer_name;
    string customer_address;
}
```

A **schema** of customer

Logical level

Name	Address
Ping	CB326, University of Hong Kong …
Ben	CB314, University of Hong Kong …

An **instance** of customers

Name	Address
Ping	CB326, University of Hong Kong …
Jolly	CB311, University of Hong Kong …

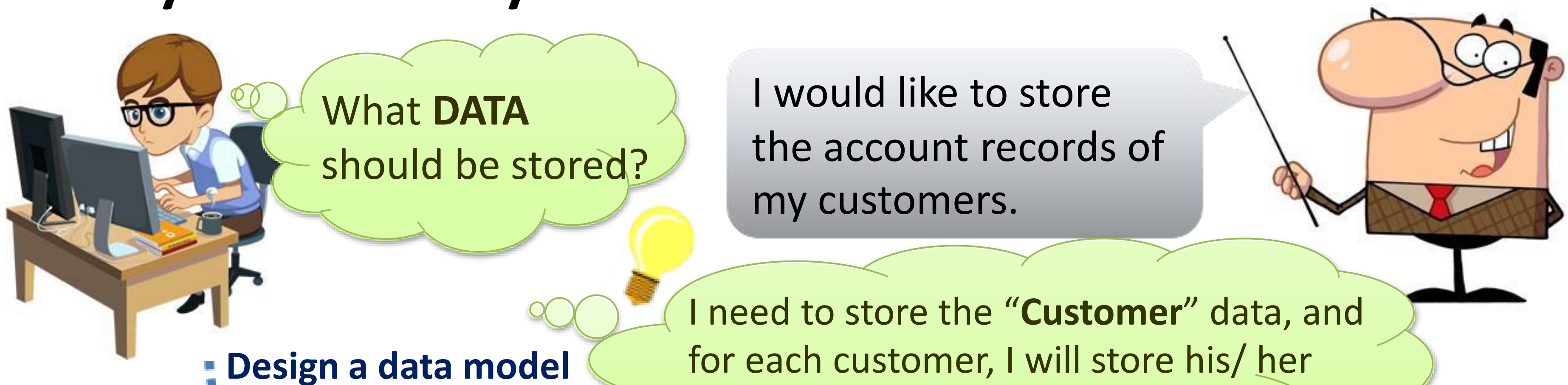
Another **instance** of customers

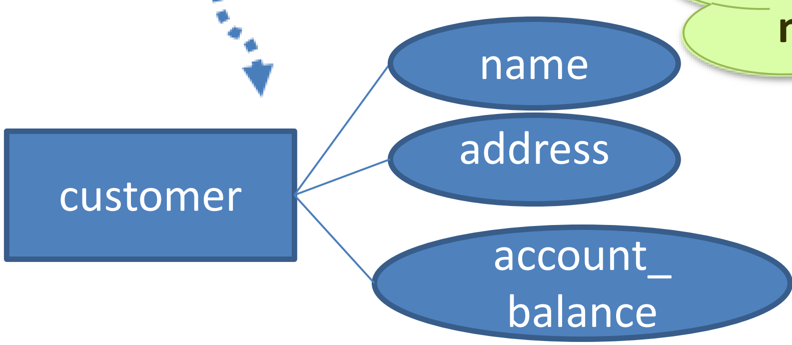
Database Management System

- **Three powerful concepts supported by a DBMS:**
  - 1. Data abstraction.
  - 2. Data modeling.
  - 3. Database languages.

2. Data modeling

- **Data modeling is used to describe data in a systematic way.**



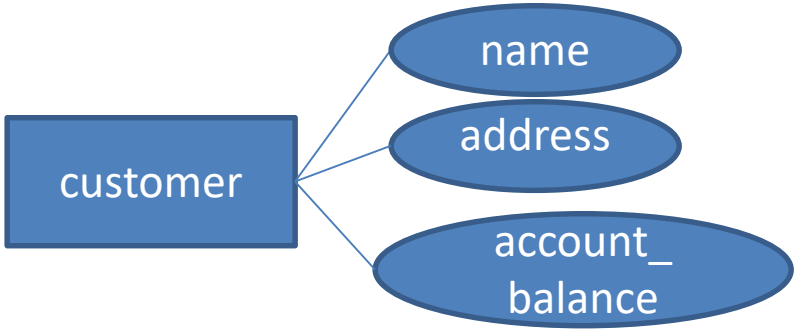


Name	Address	Account balance
Ping	HKU…	$100
Ben	…	$2,000

The **data model** → The **data instance**

2. Data modeling

- **Data modeling is used to describe data in a systematic way.**



The **data model**

Name	Address	Account balance
Ping	HKU…	$100
Ping	…	$2,000

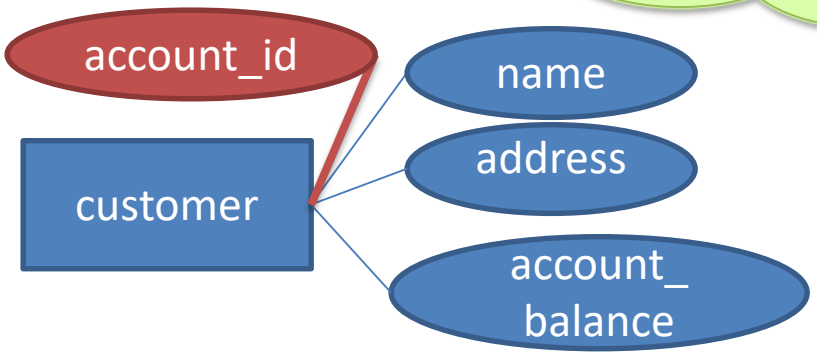
The **data instance**

2. Data modeling

- **Data modeling is used to describe data in a systematic way.**

Easy ☺! I add a **unique account ID** to distinguish different account of the same customer!

A customer can have more than one account!!!

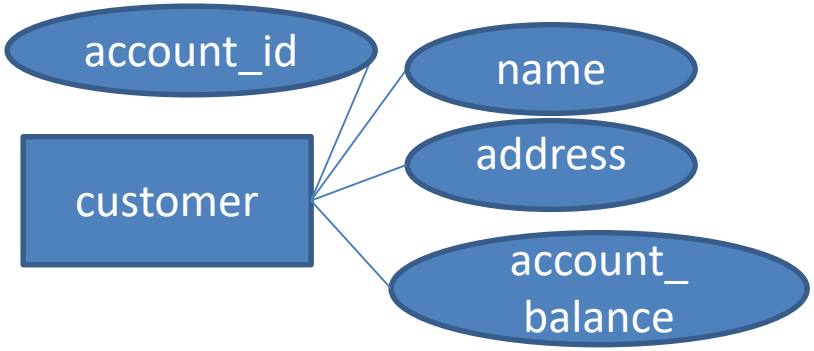


Account ID	Name	Address	Account balance
1	Ping	HKU…	$100
2	Ping	HKU…	$2,000

The **data model** ⇢ The **data instance**

2. Data modeling

- **Data modeling is used to describe data in a systematic way.**



The **data model**

Data redundancy

Account ID	Name	Address	Account balance
1	Ping	HKU…	$100
2	Ping	HKU…	$2,000

The **data instance**

2. Data modeling

- **Data modeling is used to describe data in a systematic way.**

Let me **separate** the customer information with the account information.

Customer ID	Name	Address
1	Ping	HKU…
2	…	…

Account ID	Balance
1	$100
2	$2,000

Account ID	Name	Address	Account balance
1	Ping	HKU…	$100
2	Ping	HKU…	$2,000

The **data instance**

2. Data modeling

- **Data modeling is used to describe data in a systematic way.**

An extra information is needed to represent the **relationship** of the **customers** with the **accounts**, then one customer can have more than one account, and the customer info will not be duplicated ☺!

Customer ID	Account ID
1	1
1	2

Customer ID	Name	Address
1	Ping	HKU…
2	…	…

Account ID	Balance
1	$100
2	$2,000

2. Data modeling

- **Data modeling is used to describe data in a systematic way.**

An extra information is needed to represent the **relationship** of the **customers** with the **accounts**, then one customer can have more than one account, and the customer info will not be duplicated ☺!

Customer ID	Account ID
1	1
1	2

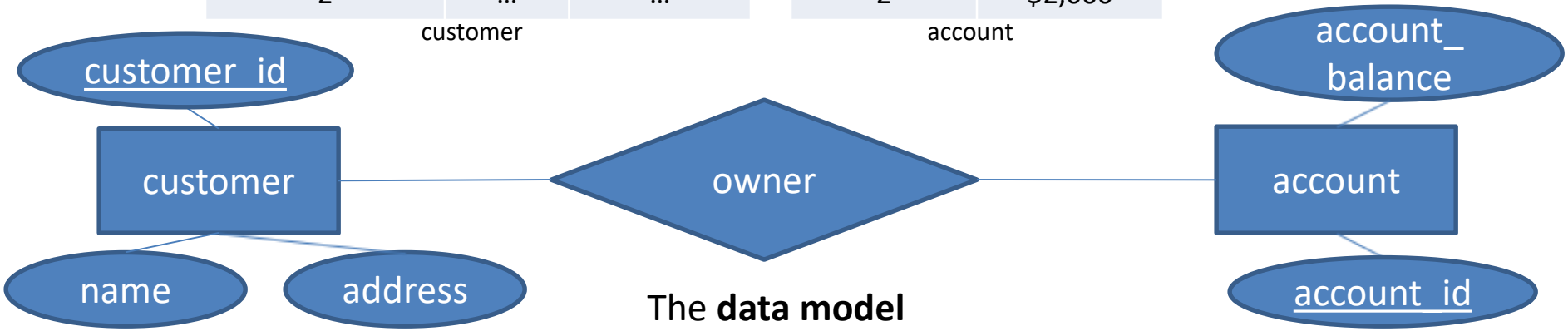
owner

Customer ID	Name	Address
1	Ping	HKU…
2	…	…

customer

Account ID	Balance
1	$100
2	$2,000

account



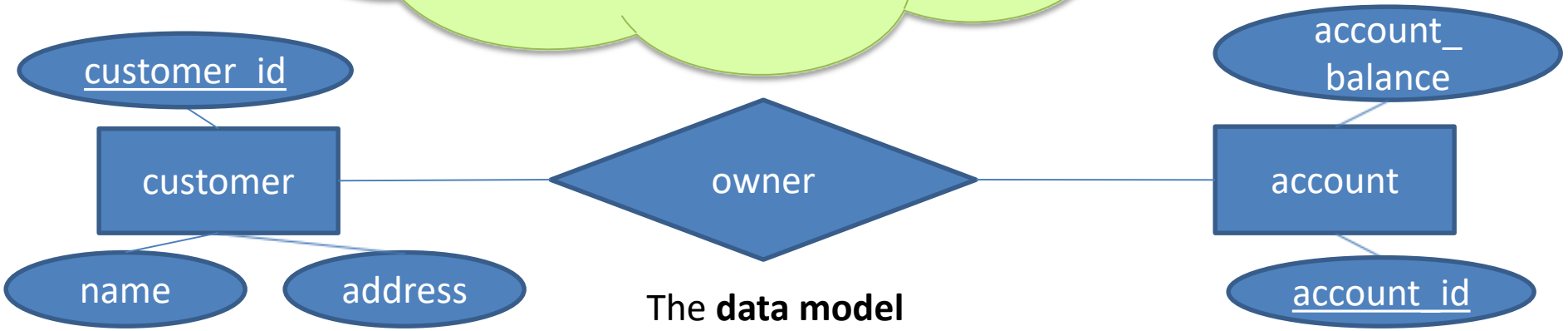
The **data model**

2. Data modeling

- **Data modeling is used to describe data in a systematic way.**

We need to **design the data model** carefully so that it can:

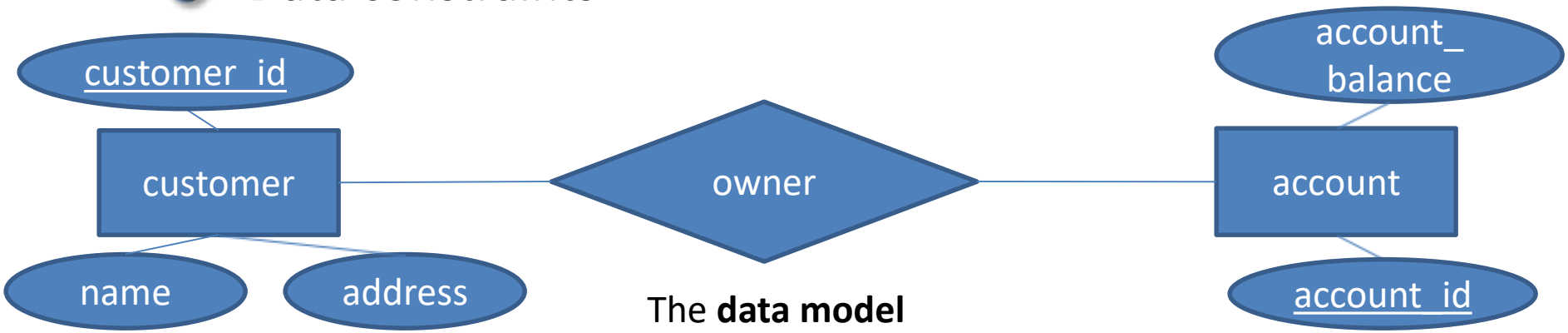
1. **Capture user requirements.**
2. **Store/ access data efficiently.**



The **data model**

2. Data modeling

- **Data modeling is used to describe data in a systematic way.**
- **A collective tool for describing**
  - Data
  - Data relationships
  - Data semantics
  - Data constraints



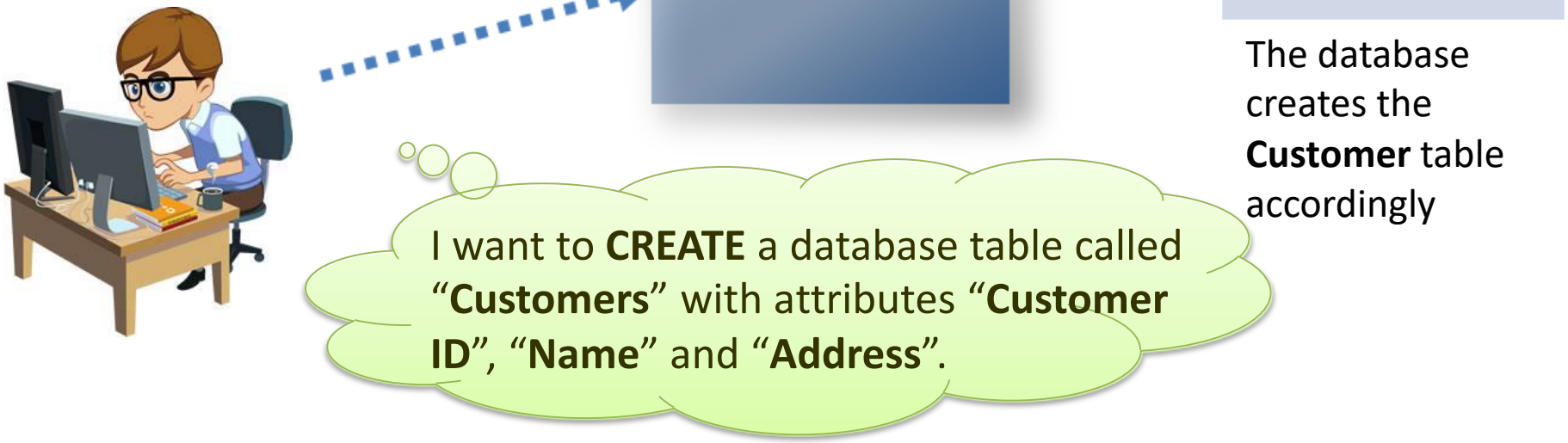
The **data model**

Database Management System

- **Three powerful concepts supported by a DBMS:**
  - 1. Data abstraction.
  - 2. Data modeling.
  - 3. Database languages.

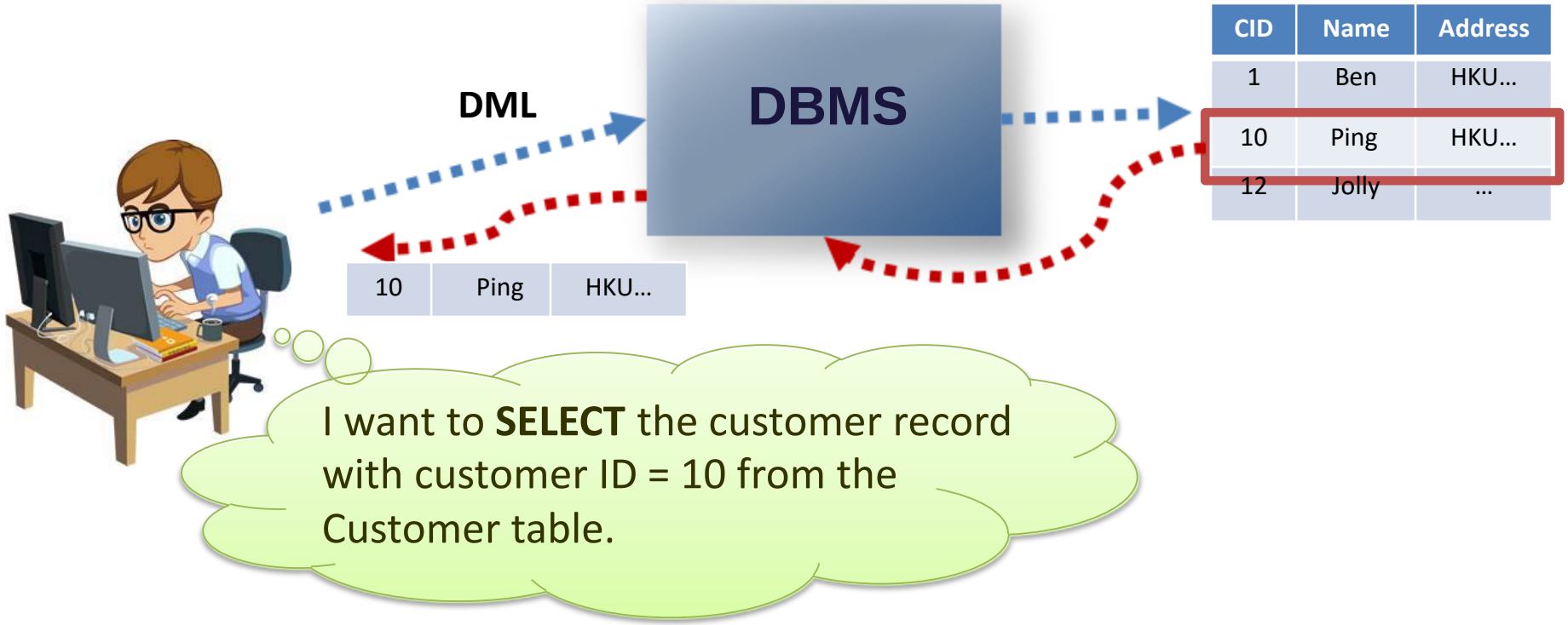
3. Database languages

- **DBMS provide different tools for managing data.**
  - Data definition language (DDL)



3. Database languages

- **DBMS provide different tools for managing data.**
  - Data definition language (DDL)
  - Data manipulation language (DML)



To Recap

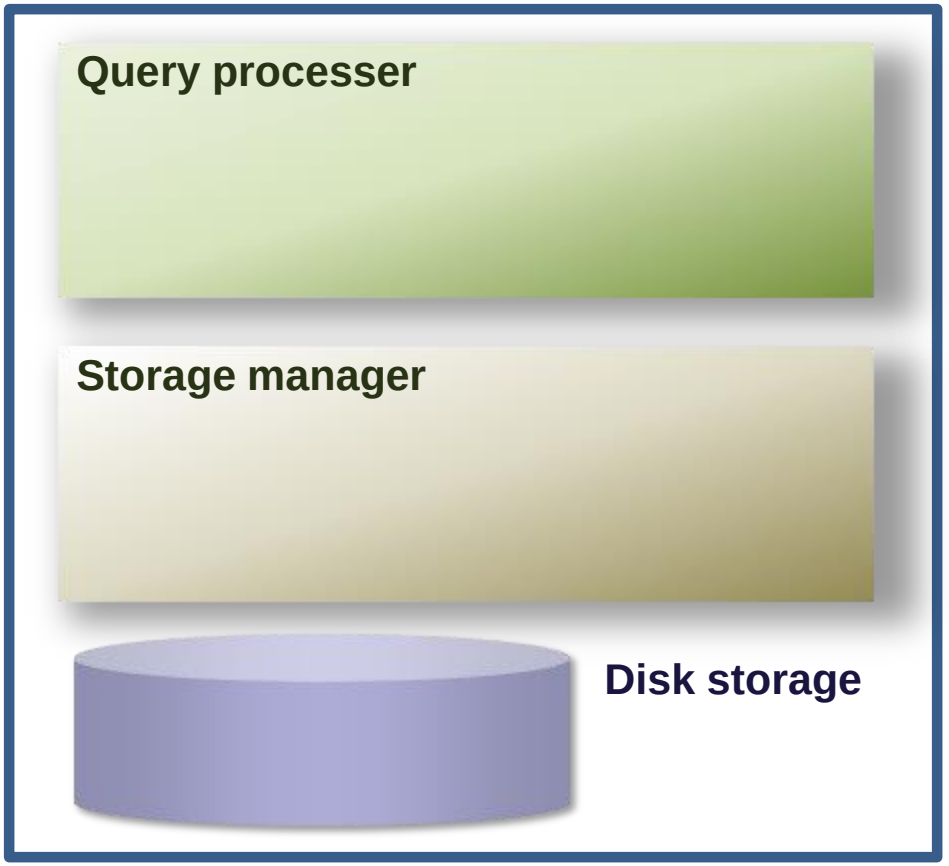
- **What is a database? Data, Database, and DBMS**
- **File vs. Database**
  - 6 drawbacks of file systems
- **Three concepts supported by a DBMS**
  1. Data Abstraction (Physical/Logical/View)
  2. Data Modeling (Table/Attributes/Relationships)
  3. Database Language (DDL & DML)

Section 1.2

Database Overview

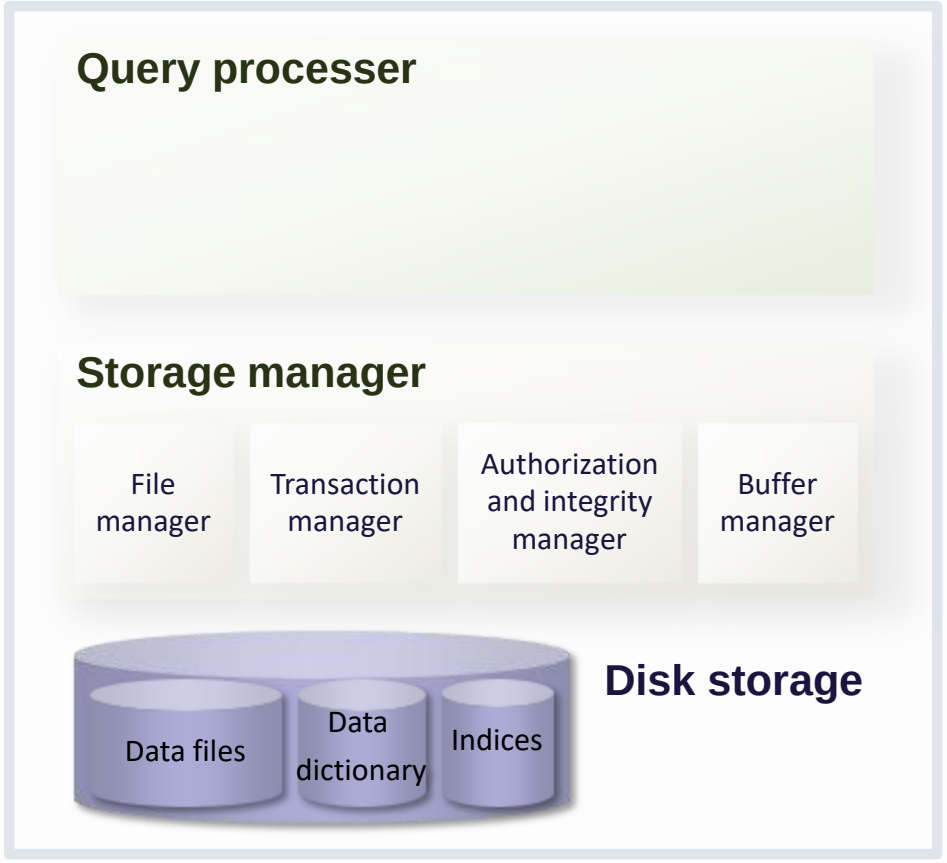
Database Overview

- **A DBMS is a complex system partitioned into modules that deal with each of the responsibilities of the overall system.**
- **The functional components can be broadly divided into**
  - Disk Storage
  - Storage manager
  - Query processer



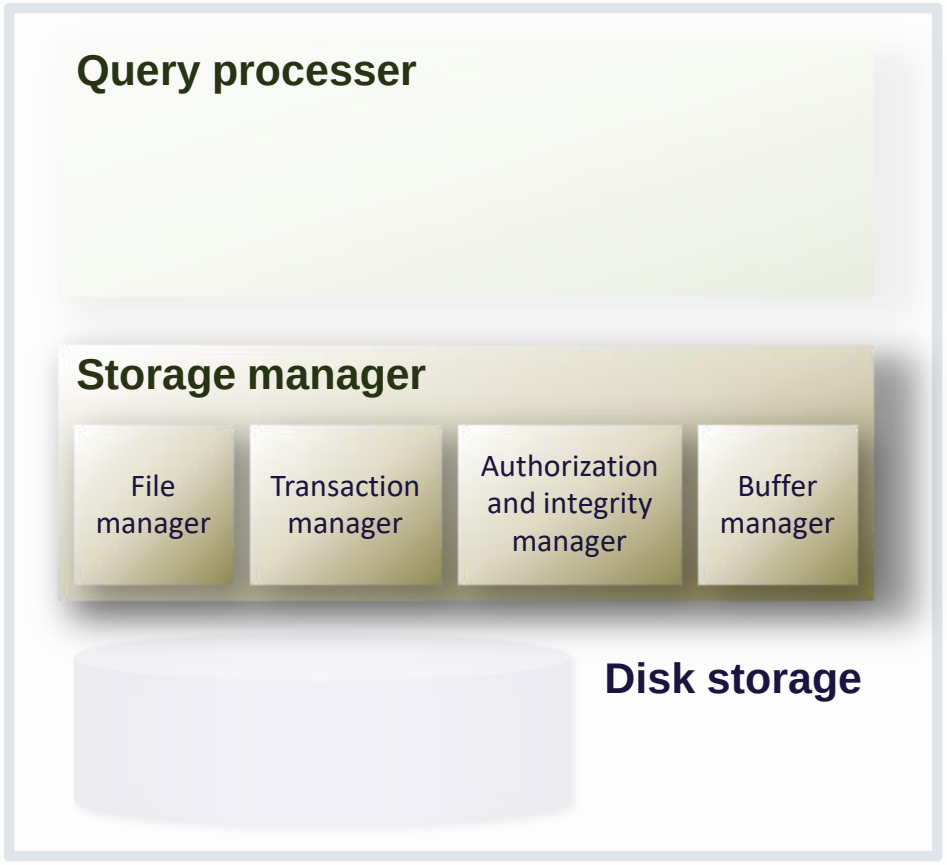
Disk storage

- **Disk storage consists of**
  - **Data files** – stores the database itself.
  - **Data dictionary** – stores metadata about the structure of the database, in particular the schema of the database.
  - **Indices** – provide fast access to data records.



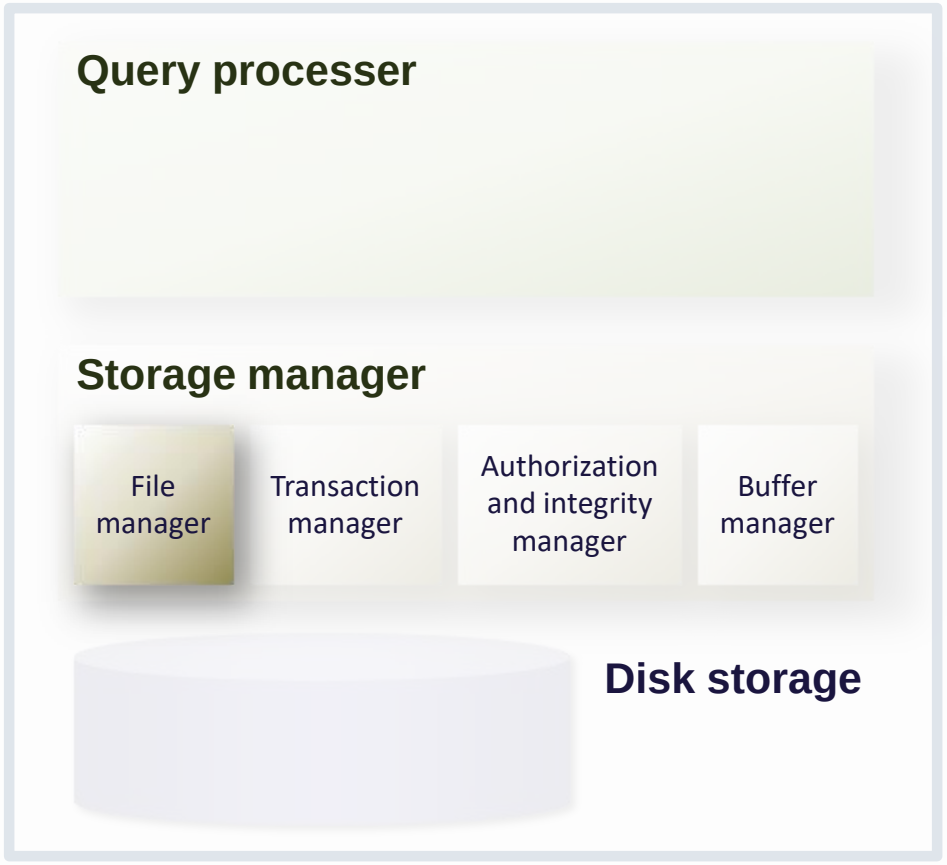
Storage manager

- **A program module in DBMS that provides interface between the low-level data and the application programs/queries.**
- **Consists of**
  - File manager
  - Transaction manager
  - Authorization and integrity manager
  - Buffer manager



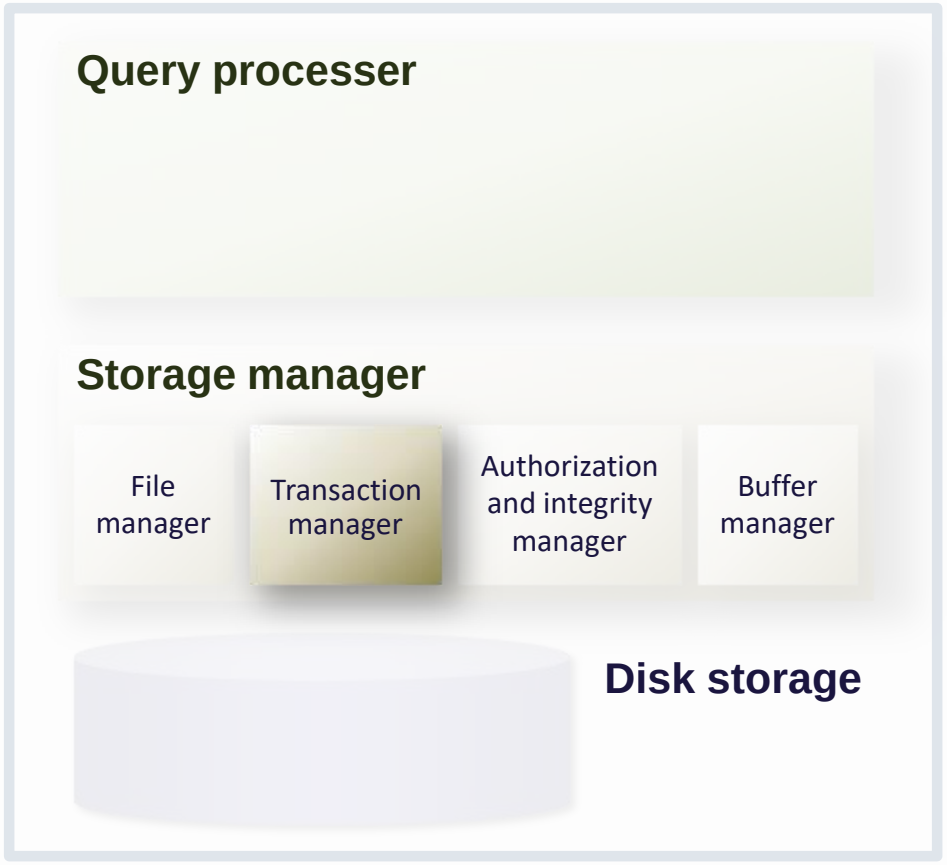
File manager

- Manages the allocation of space on disk storage and the data structures used to represent information stored on disk.



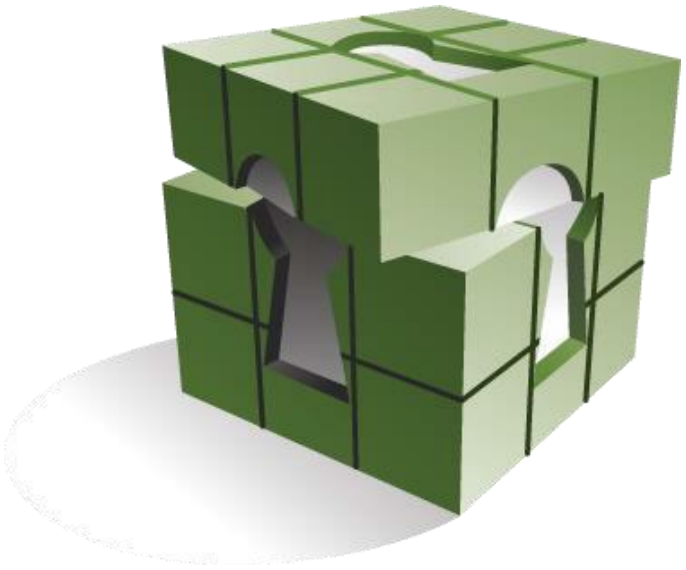
Transaction manager

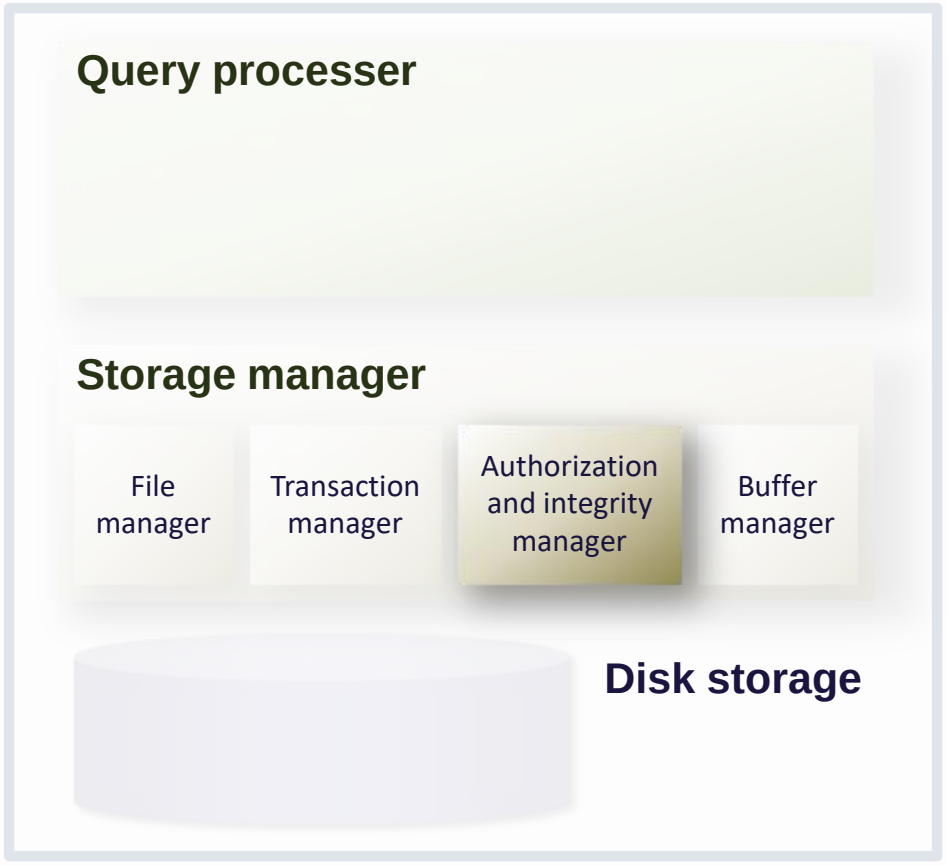
- Ensures that the database remains in a consistent (correct) state despite system failures.
- Ensures interaction among all concurrent transactions does not make the DB inconsistent.



Authorization and integrity manager

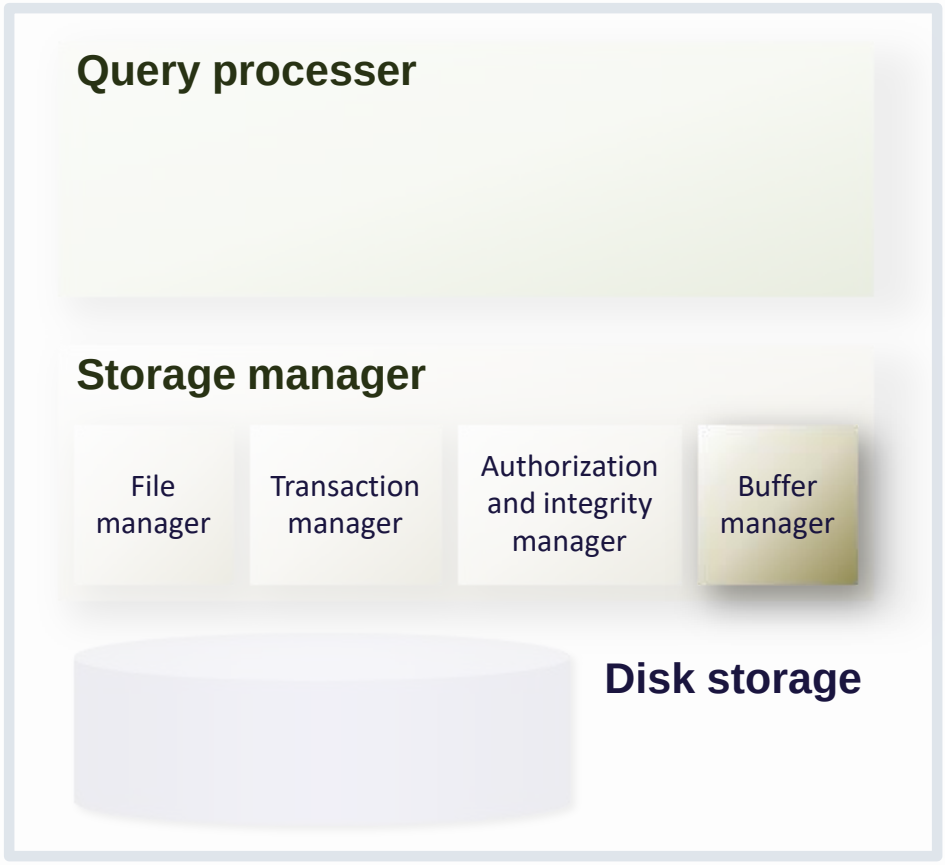
- Tests for the satisfaction of integrity constraints and checks the authority of users to access data.





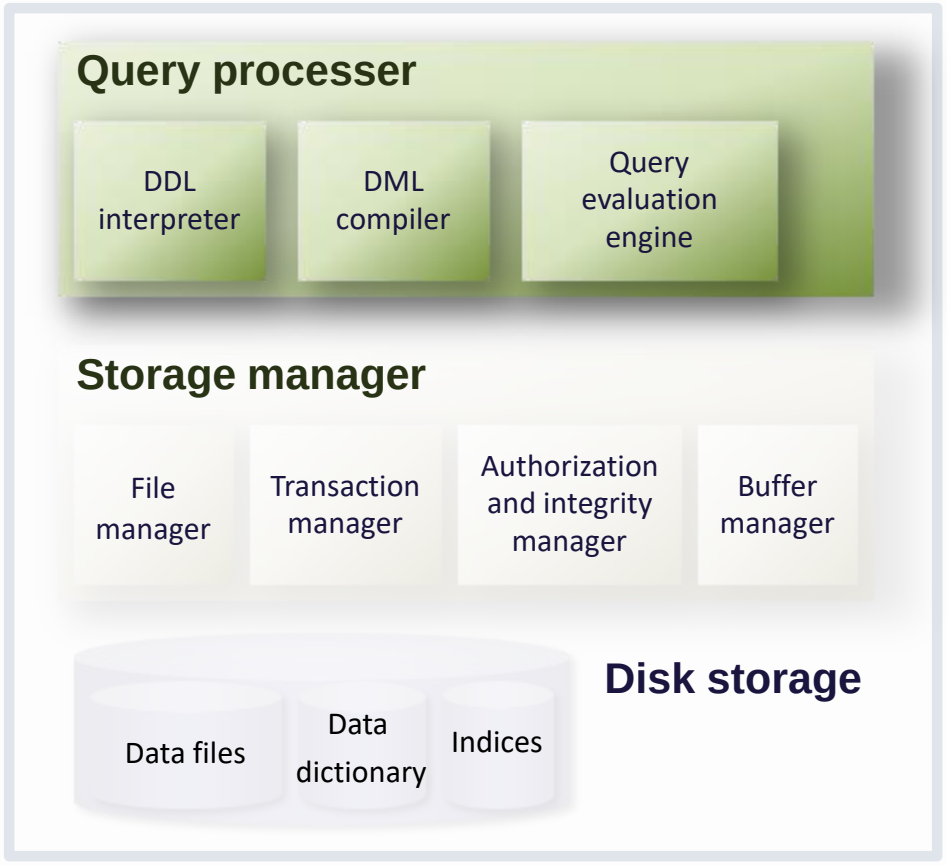
Buffer manager

- Responsible for fetching data from disk storage into main memory.
- Decide what data to cache in main memory.



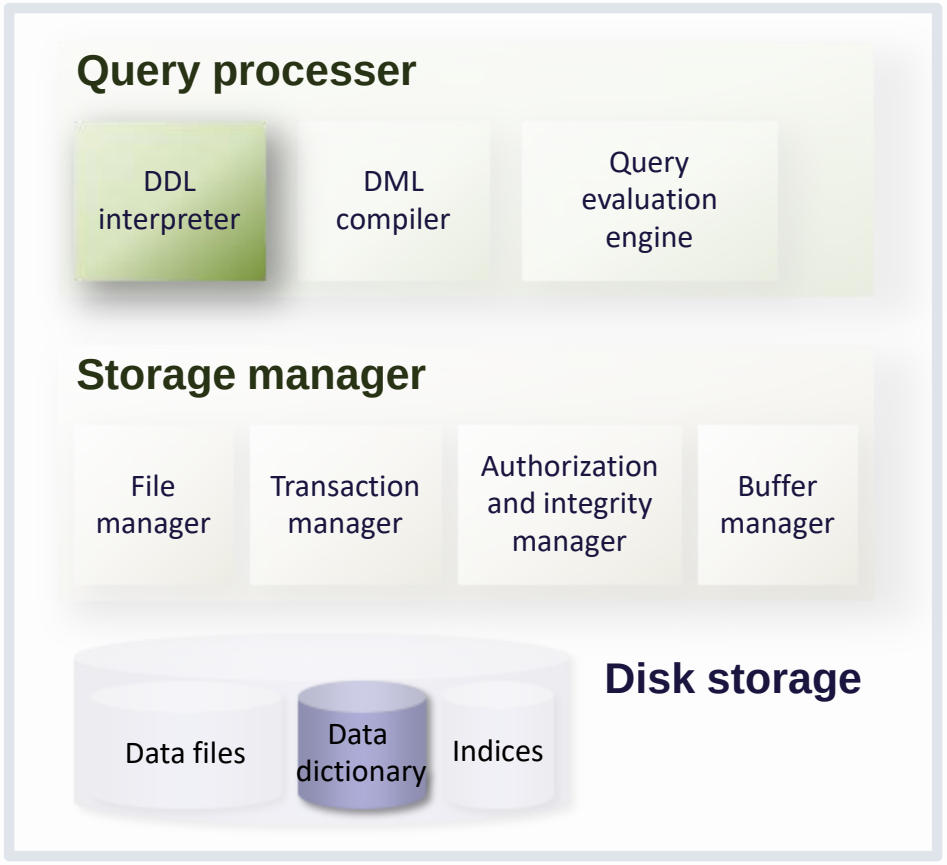
Query processer

- **A program module in DBMS that converts high-level user requests to efficient low-level commands to DB.**
- **Consists of**
  - DDL interpreter
  - DML compiler
  - Query evaluation engine



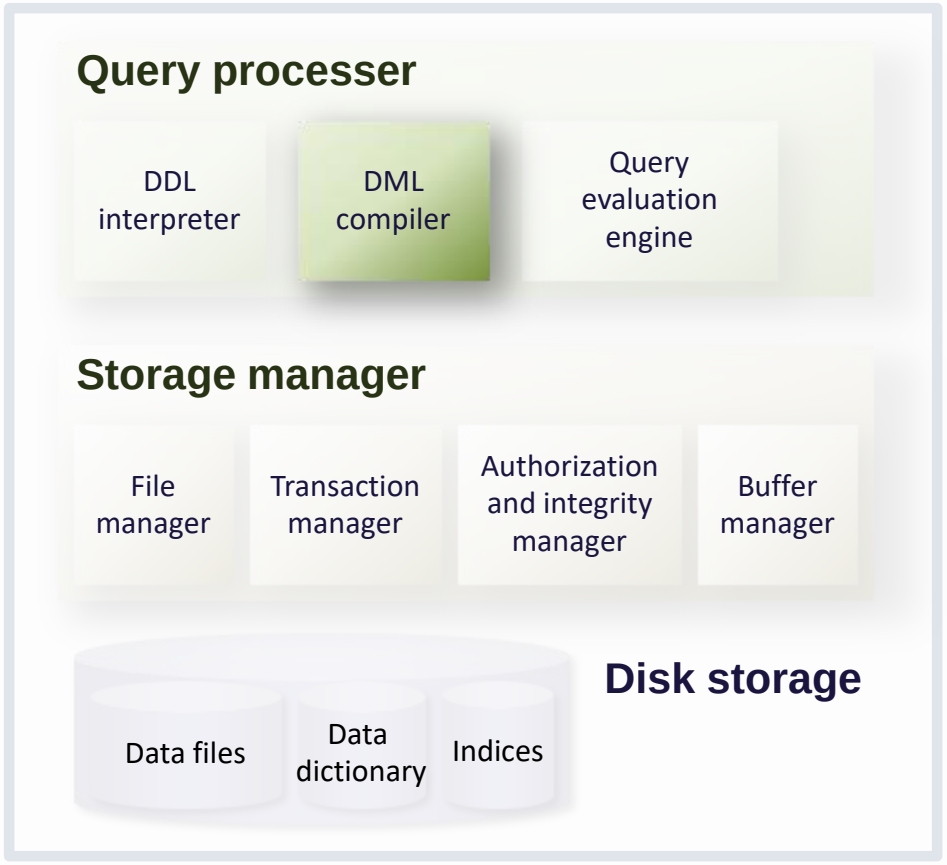
DDL interpreter

- **Interprets the DDL statements and records the definition in the data dictionary (i.e. structure of the DB)**



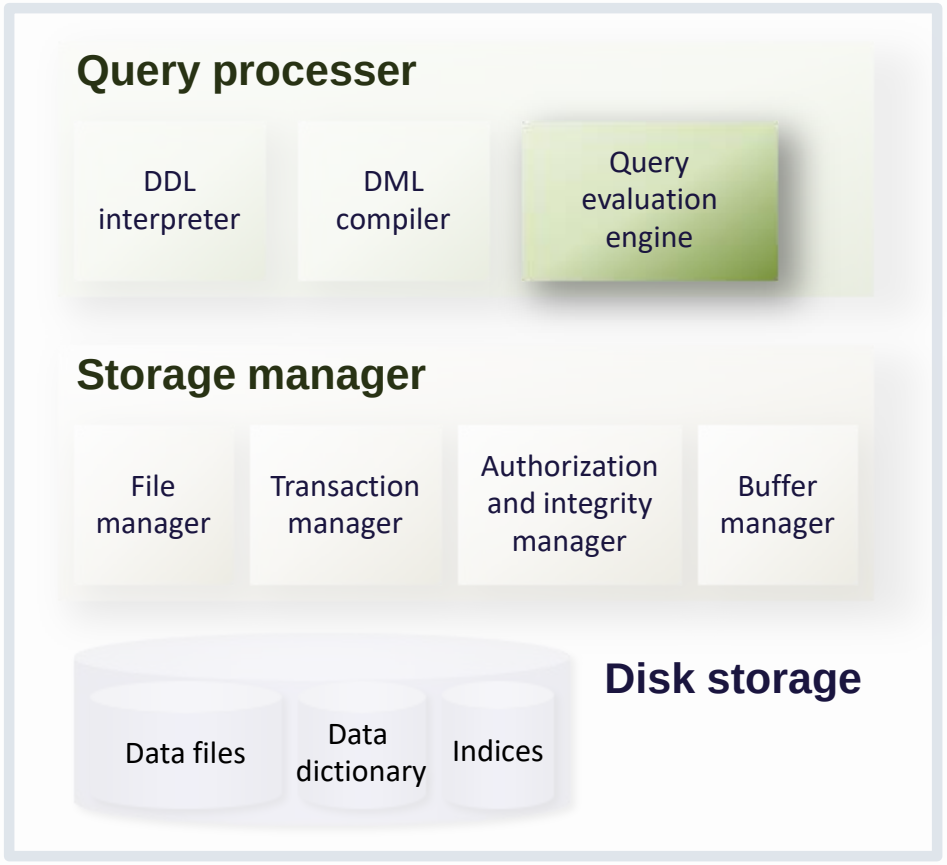
DML compiler

- **Translates DML statements in a query language into an evaluation plan.**
  - **Evaluation plan** - low-level instructions that the query evaluation engine understand.
  - **Query optimization** – Picks the lowest cost evaluation from among the alternative plans.



Query evaluation engine

- **Executes low-level instructions generated by the DML compiler.**



Summary

- Simple file systems are inadequate for enterprise needs – the 6 drawbacks.
- **Three powerful concepts supported by a DBMS:**
  - Data abstraction - 3 levels of abstraction, schema and instances.
  - Data modeling.
  - Database languages – DDL and DML.

Summary

- **Three functional components of a DBMS**
  - Disk Storage: data files, data dictionary, indexes
  - Storage manager: File manager, Transaction manager, Authorization and integrity manager, Buffer manager
  - Query processer: DDL interpreter, DML complier, Query evaluation engine

Lecture 1.

END

COMP3278B

Introduction to Database Management Systems

Dr. Ping Luo

Email : pluo@cs.hku.hk

Department of Computer Science, The University of Hong Kong

Course Project

Project Descriptions. You are invited to develop a **facial login component** of an Intelligent Course Management System (ICMS). The face login component may have the following functions.

- When a student login with his/her face, his/her information such as name, login time, and welcome message will be presented in the graphics user interface (GUI).
- If the student has class within one hour, the corresponding course information, classroom address, teacher's message, links of Zoom, tutorial/lecture notes, other course materials and so on and so forth will be presented in the GUI. The student could click the links to redirect to Zoom or other materials. The GUI should also allow the student to send the above information to his/her email address by email.
- If the student does not have class at the moment, the GUI could present a personal class timetable for the student.
- The system should record the latest behaviour of the student, such as when he/she logins the system, how long the student stays in the system, etc.

Requirements.

- **Group**: 1-5 students as a group.
- **GUI**: Each group may design GUI based on the understanding of the above user requirement (You could make your own design choice, because in real project, clients typically don't know what they really want).
- **Database**: your database should have at least five tables. How to design the tables is your design choice.

Course Project

Development Tools.

- **Face Recognition:** Python + OpenCV (full codes provided.)
- **GUI**: Python GUI or Qt. (code not provided.)
- **Database**: Python + MySQL (sample codes provided.)
- **Other**: You can use any other Python packages if you see fit. However, the main programming language should be Python and the DBMS should be MySQL.

Marks (course project 20% of the final mark).

- 10% for software development. (4% GUI + 6% database)
- Other 10% for 10-minute presentation, including but not limited to development plan, milestones, contribution of each group member, video recording of demo, software design, database design (ER Diagram, tables), difficulties you encountered and how to solve them, etc.
- Live demo is allowed, but please make sure your program works well and stably in order to save time in presentation.
- Creative GUI design, creative software functions or creative DB design will have bonus points.