Database

Database system

- It's everywhere in our daily life
  - 清華大學校務系統
  - 便利商店收銀員系統
  - 銀行、提款機、存款、轉帳、…
  - 手機通訊記錄
  - 醫療紀錄
  - 水電帳單
  - …

校務資訊系統
Academic Information Systems
https://www.ccxp.nthu.edu.tw/ccxp/INQUIRE/

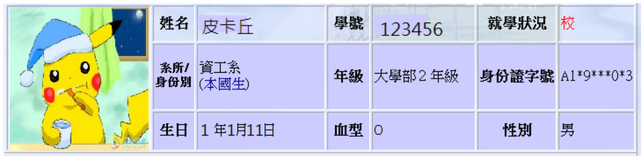
Why not file systems?

1. What we need is not only data, but also the relations among them, which can be very complex.
   - The relations are also data
   - Also need data to describe data (metadata)
2. Common data operations are easier to perform using DataBase Management System (DBMS)
   - Search: retrieve data from the database
   - Update: update existing data
   - Insertion: insert new data
   - Deletion: remove existing data

- Some DBMS uses files to store data, and some file systems use database to manage files.

Case study:清華大學校務系統

- Every student has a record
  - Consisted of many attributes: Name, Student ID, Status, Major, Grade, ID, Birthday, Blood type, Gender, Photo, …



姓名	皮卡丘	學號	123456	就學狀況	校
系所/身份別	資工系 (本國生)	年級	大學部 2 年級	身份證字號	A1*9***0*3
生日	1 年1月11日	血型	O	性別	男

- Every attribute has a data type
  - Name: character(80)
  - Status: {校，復，退，休}
  - Photo: image(160x160)

Table

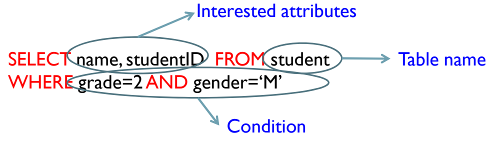
- In a ***relational database*** (the most commonly used one), records are organized using tables
  - Columns for attributes; rows for records

Name	StudentID	Status	Major	Grade	B-day	Gender	…
皮卡丘	123456	校	資工系	二年級	1-1-11	M	…
可達鴨	789012	休	中文系	二年級	2-2-22	F	…
…	…	…	…	…			…

- Primary key: one (or multiple) attributes that can be used to uniquely identify each row in a table
  - Ex: Student ID is the primary key of the student table
    - Why not use name? and why not use ID (身分證字號)?
  - Multiple attributes as the primary key: ex: name+address

Query

- Using SQL (Structured Query Language)
- Ex:



SELECT name, studentID FROM student
WHERE grade=2 AND gender='M'

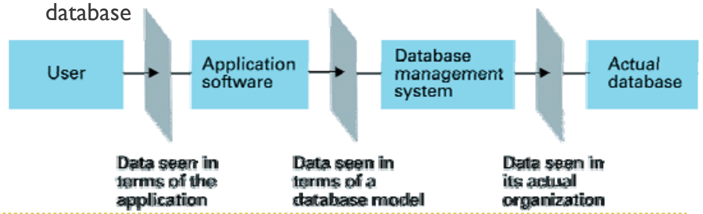
- Ex: query all attributes

SELECT * FROM student WHERE StudentID=123456

- Indices for the attributes in conditions should be pre-built to speedup queries
  - Primary key is always indexed

Multi-tier architecture

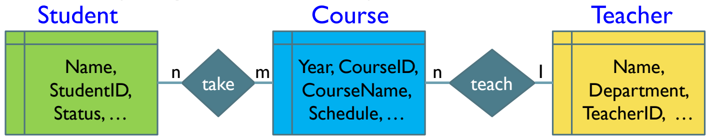
- How to turn the queried results to the form you see?
  - The major may be coded: 01 核工系，02 數學系，…
  - Image is a bit stream or a file name
- Need application software to present the data to user.
  - There may be more than one tiers in between.
  - Different applications (user interfaces) can access the same database



User
Application software
Database management system
Actual database
Data seen in terms of the application
Data seen in terms of a database model
Data seen in its actual organization

Relations

- The relations among objects are described by the ER model (Entity-Relation model)



Student
Course
Teacher
Name, StudentID, Status, …
n
take
m
Year, CourseID, CourseName, Schedule, …
n
teach
1
Name, Department, TeacherID, …

- Relations are also organized as tables

CourseTaking

StudentID	CourseID	Grade	Status
123456	990110	-1	Normal
234567	990221	-1	Dropped

CourseTeaching

TeacherID	CourseID
888999	990110
777666	990221

- What should be the primary key?

Query from different tables (Join)

- Suppose you want to know your course names, schedule, and instructors' names.

科目名稱	上課時間	授課老師
英文	M3M4W3	小瑤
微積分	T3T4H3H4	小剛，小智
體育	F5F6	小智
…	…	…

Your student ID

```
SELECT Course.Name, Course.Schedule, Teacher.Name
FROM   Course, CourseTaking, CourseTeaching, Student, Teacher
WHERE Student.StudentID='123456' AND
       Student.StudentID=CourseTaking.StudentID AND
       CourseTaking.CourseID=Course.CourseID AND
       CourseTeaching.CourseID=Course.CourseID AND
       CourseTeaching.TeacherID=Teacher.TeacherID.
```

Usually will build a "view" to speedup common queries

Transaction (update, insertion, deletion)

- Suppose you want to add a course for next semester.
  - Using SQL insertion command

```
INSERT INTO CourseTaking VALUES ('123456','990110');
```

- But a transaction is more than just an insertion like that.

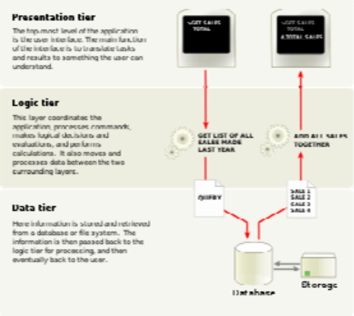
  A sequence of operations that must all happen together
  - Before insertion
    - The system needs to check if there is a schedule confliction
    - Also, the capacity of the class, the pre-requirement, …
  - After insertion
    - Suppose there is an attribute in Course, called "NoStudent", that records the total number of students taking this course.

```
UPDATE Course SET NoStudent=ns+1 WHERE CourseID='990110';
```

ns is a pre-queried number for Course.NoStudent

Three-tier architecture

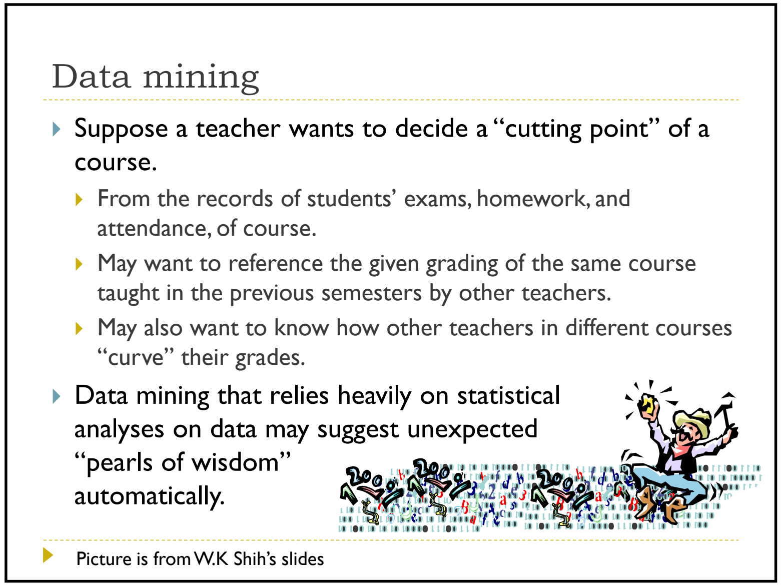
- There is a logic tier, between user interface and DBMS, that packs all the actions of a transaction together.
  - Query the schedule of courses that the student had taken.
  - Query the schedule of the course to take.
  - Check if there is any time confliction.
  - If no confliction, do the next checking. Otherwise, report an error to user-interface (presentation tier).


Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.
>GET SALES TOTAL
>GET SALES TOTAL
4 TOTAL SALES
GET LIST OF ALL SALES MADE LAST YEAR
ADD ALL SALES TOGETHER
QUERY
SALE 1
SALE 2
SALE 3
SALE 4
Database
Storage

From wikipedia: http://en.wikipedia.org/wiki/Multitier_architecture

Maintaining database integrity

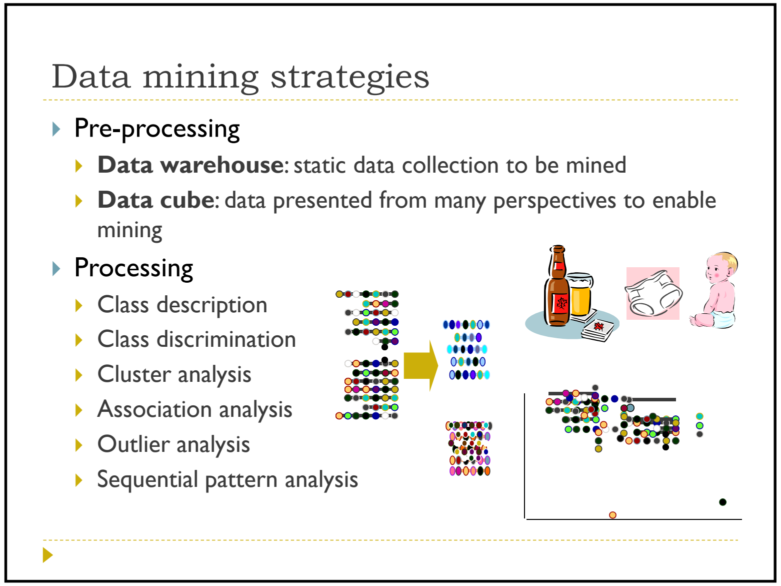
- Suppose two students add the same course simultaneously.
  - They both pass all the checking, and have done the insertion.
  - Both of them got ns=30, so when they updated the database, the attribute "NoStudent" is set 31. But there are 32 students on the course list.
- DBMS need to maintain database integrity
  - **Transaction log:** non-volatile record of each transaction's activities, built before the transaction is allowed to happen.
  - **Locking:** preventing others from accessing data being used by a transaction.
  - **Roll-back:** procedure to undo a failed, partially completed transaction



Data mining

- Suppose a teacher wants to decide a "cutting point" of a course.
  - From the records of students' exams, homework, and attendance, of course.
  - May want to reference the given grading of the same course taught in the previous semesters by other teachers.
  - May also want to know how other teachers in different courses "curve" their grades.
- Data mining that relies heavily on statistical analyses on data may suggest unexpected "pearls of wisdom" automatically.

- Picture is from W.K Shih's slides



Data mining strategies

- Pre-processing
  - **Data warehouse**: static data collection to be mined
  - **Data cube**: data presented from many perspectives to enable mining
- Processing
  - Class description
  - Class discrimination
  - Cluster analysis
  - Association analysis
  - Outlier analysis
  - Sequential pattern analysis