Persistence and Databases

Week 10 Presentation 1

Rehash of Character Streams

- What can character streams do that byte streams cannot?
- Read and write line by line
  - When buffered
- Change encodings
- Can use a scanner and parse text input

File and Directory Operations

- Check the File class…
  - https://docs.oracle.com/javase/8/docs/api/java/io/File.html

- Copying and deleting files
- Listing and searching directories
- and more

You can do a lot of things with files other than reading and writing them. There are constants in the File class that take care of differences between Linux, Mac, and Windows.

Files USED to be very important

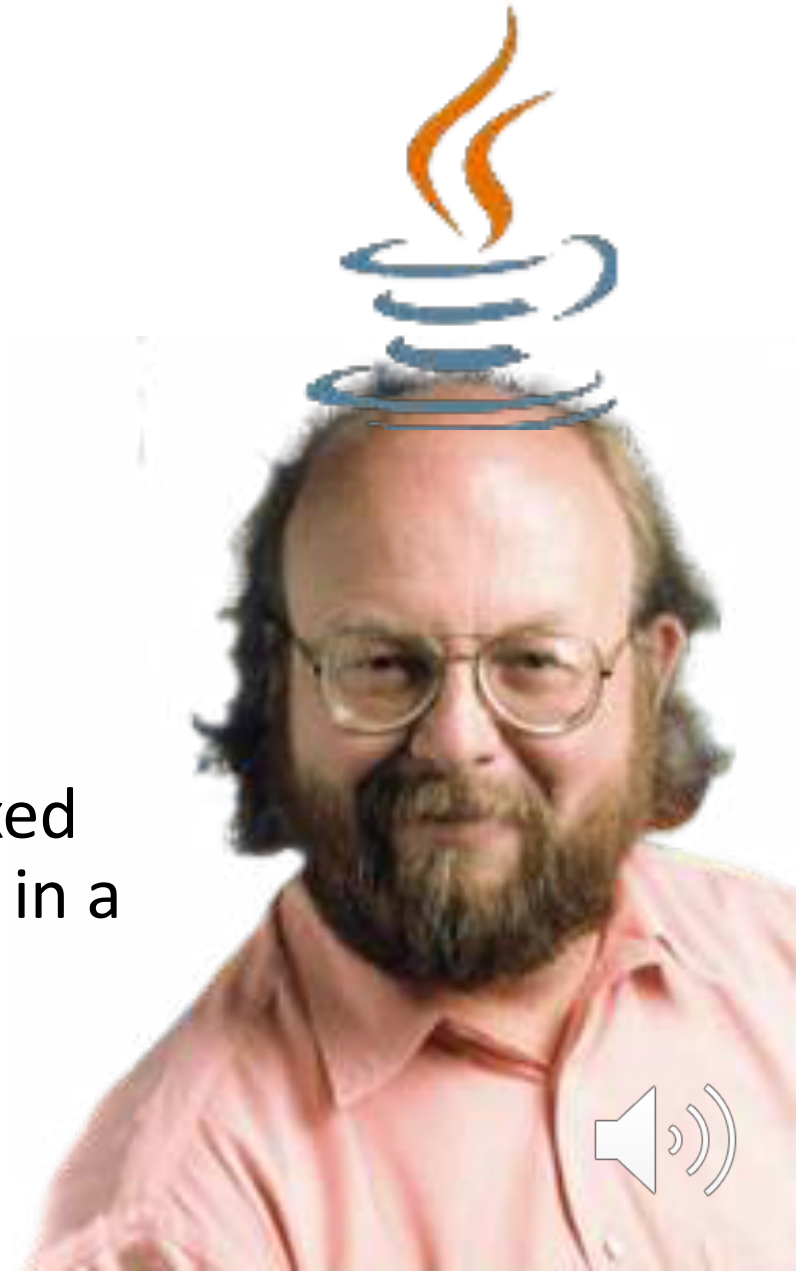
- They still are, to an extent, for specialized applications, and they are still the backbone of persistence.
- However, as an application developer, you are increasingly isolated from files.
- We saw with Image and Media you can add Properties, and we will see that in a similar way databases act as a layer between programs and files.
- A lot of data comes from networks as well.
- Every application, 40 years ago, used to open and close a lot of files, this is no longer the case. You open files mostly to load data in memory, and work there. All this is related to the cost of memory – if you are directly working with files the memory footprint of a program can be reduced…

In 1970 1 Mb cost ???

- When Dennis Ritchie created C memory was very very expensive.

- Programmers needed to be very careful to release memory as soon as it was no longer needed to try and save a few bytes…

In 1990 1 Mb cost ???



Twenty years later, James Gosling could take a more relaxed approach, have a garbage collector freeing memory once in a while, and afford the luxury of a 2- byte char.

2020: $0.10

- And today? Memory costs next to nothing. Just one problem: as the cost of memory was decreasing, applications were using more and more of it. And computers were supporting more and more users. You'd be wrong to believe that you no longer have to worry about memory. In some languages and environments (I'm thinking of Web servers running PHP) memory-per-user is limited to keep everything under control.
- But you don't need to fear if you sometimes load quite a lot of data in memory.

The result? A common way of working:

- Because memory is so cheap these days many people load everything to memory perform the work, and finally save everything to disk when done – or they use databases (which requires a different approach)

1. Load all data required initially
2. Work in memory (e.g. using Collections)
3. Save everything at the end
4. Use databases when data needs to be shared or it is in very large volumes beyond memory size

IMPORTANT

- When working with databases, the logic is very different from when you work with files - few Java developers understand it!

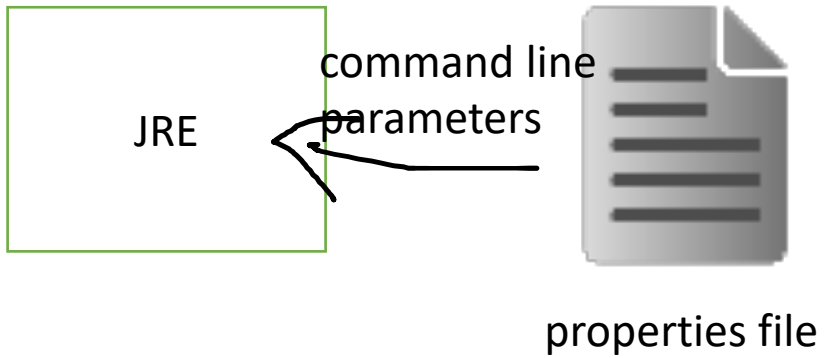
- The issue is that is the correct way of working with files is the wrong way when with databases
- This is because databases are **systems** that are optimized for data retrieval and processing and work much faster than anything you can do in Java

How are files largely used in industry?

- Multimedia documents are handled by specialized routines
- Parameters

We saw Media and Images in JavaFx as well as Property files. In many cases all the file reading is performed by a method in a specialized object and the developer doesn’t directly work with File I/O.

Parameter files



One important use of parameter files is when we would like a user to be able to modify important settings when they just have a .class file

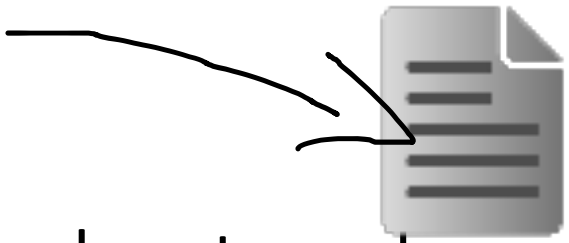
- Specify directories
- Remote connections

JDK

This occurs when the user does not need and probably cannot use a JDK (ie they are not a java developer)

Parameter Files – Usage Pattern

- First you should assign some reasonable default values
- Then as required replace values and save in a parameter file

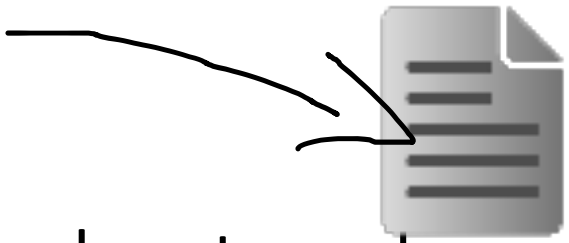


- The values to replace may be specified on the command line

Locations for resources should always be specified in property files

Parameter Files – Usage Pattern

- First you should assign some reasonable default values
- Then as required replace values and save in a parameter file



- The values to replace may be specified on the command line

Locations for resources should always be specified in property files

java.util.properties

- loading key/value pairs into a Properties object from a stream,
- retrieving a value from its key,
- listing the keys and their values,
- enumerating over the keys, and
- saving the properties to a stream.

https://docs.oracle.com/javase/8/docs/api/java/util/Properties.html

Setting up properties object

```
. . .
// create and load default properties
Properties defaultProps = new Properties();
FileInputStream in = new FileInputStream("defaultProperties");
defaultProps.load(in);
in.close();

// create application properties with default
Properties applicationProps = new Properties(defaultProps);

// now load properties
// from last invocation
in = new FileInputStream("appProperties");
applicationProps.load(in);
in.close();
. . .
```

Saving Properties

```
FileOutputStream out = new FileOutputStream("appProperties");
applicationProps.store(out, "---No Comment---");
out.close();
```

Getting property information

- contains(Object value) and containsKey(Object key)
  Returns true if the value or the key is in the Properties object. Properties inherits these methods from Hashtable. Thus they accept Object arguments, but only String values should be used.
- getProperty(String key) and getProperty(String key, String default)
  Returns the value for the specified property. The second version provides for a default value. If the key is not found, the default is returned.
- list(PrintStream s) and list(PrintWriter w)
  Writes all of the properties to the specified stream or writer. This is useful for debugging.
- elements(), keys(), and propertyNames()
  Returns an Enumeration containing the keys or values (as indicated by the method name) contained in the Properties object. The keys method only returns the keys for the object itself; the propertyNames method returns the keys for default properties as well.
- stringPropertyNames()
  Like propertyNames, but returns a Set<String>, and only returns names of properties where both key and value are strings. Note that the Set object is not backed by the Properties object, so changes in one do not affect the other.
- size()
  Returns the current number of key/value pairs.

Setting property values

- setProperty(String key, String value)
  Puts the key/value pair in the Properties object.
- remove(Object key)
  Removes the key/value pair associated with key.

How are files largely used in industry?

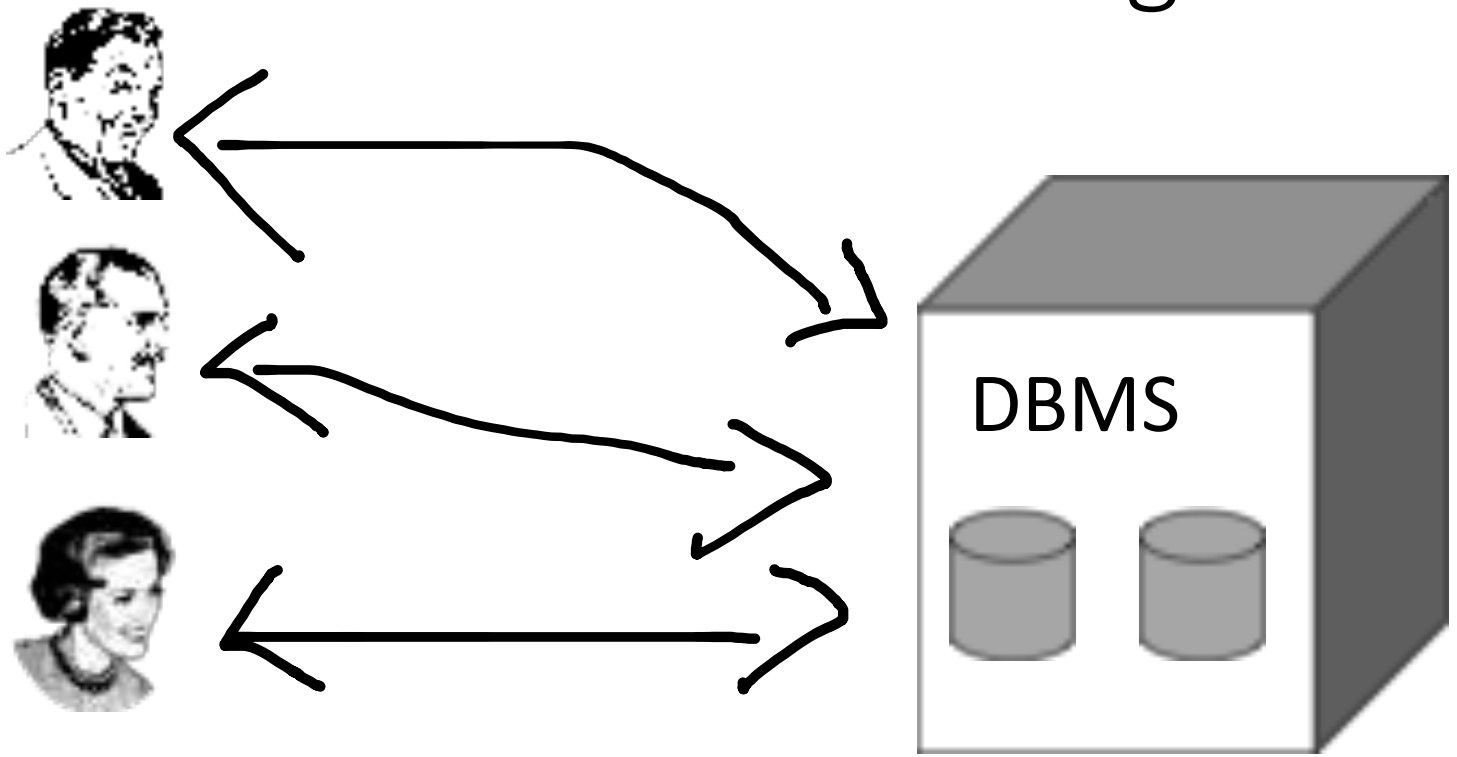
- Multimedia documents are handled by specialized routines
- Parameters

- Data Collection and Exchange

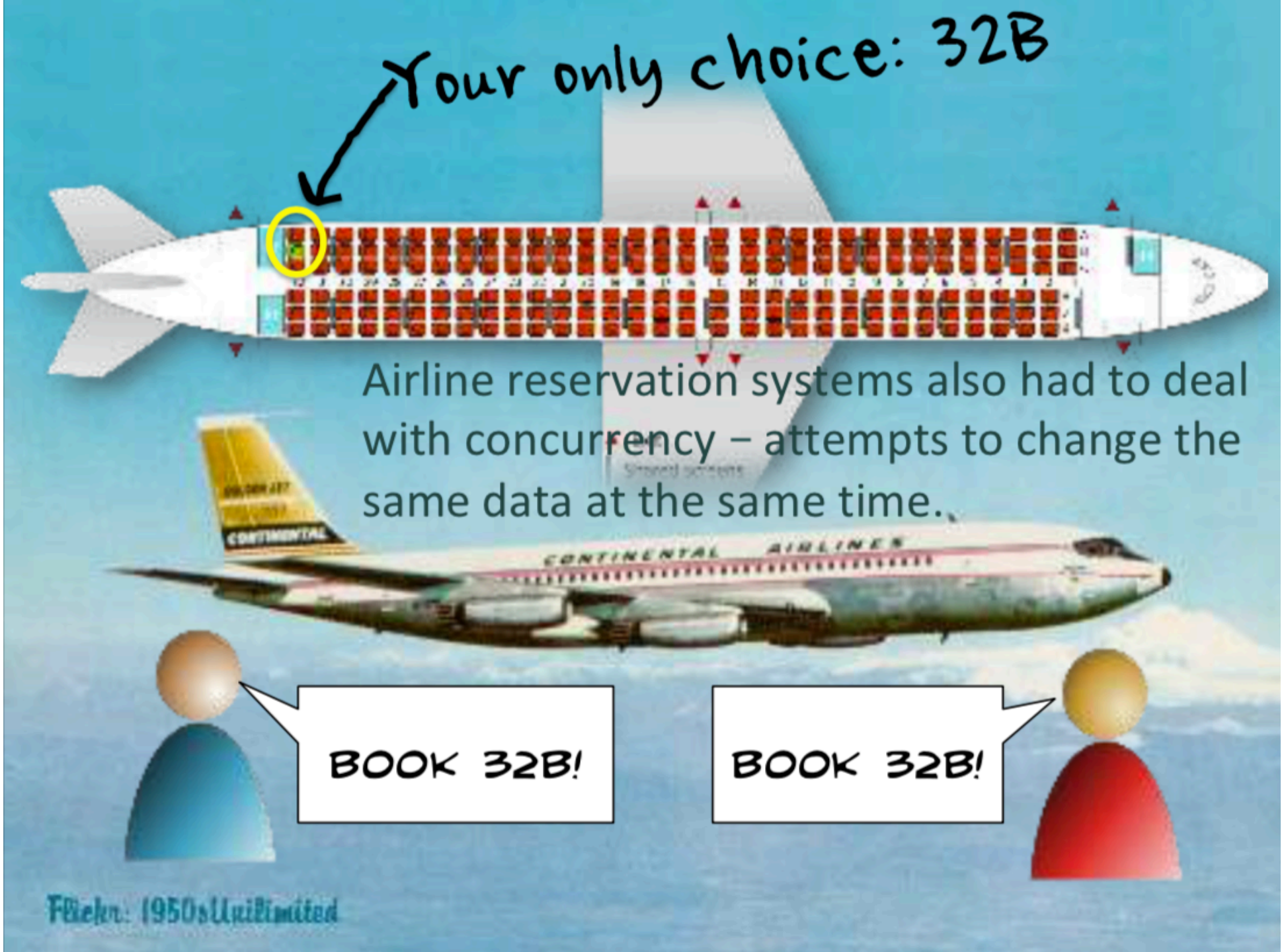
You mostly read and write data for exchanging data between systems – which include database systems.

DATABASES

DBMS – DataBase Management System



Very early systems were developed systems for managing files and retrieving data to abstract away “lower” software layers that deal with things like disk access and data retrieval… To avoid having to bother with lower software layers.


Your only choice: 32B
Airline reservation systems also had to deal with concurrency – attempts to change the same data at the same time.
CONTINENTAL AIRLINES
BOOK 32B!
BOOK 32B!
Flickr: 1950sUnlimited

CONTROL changes

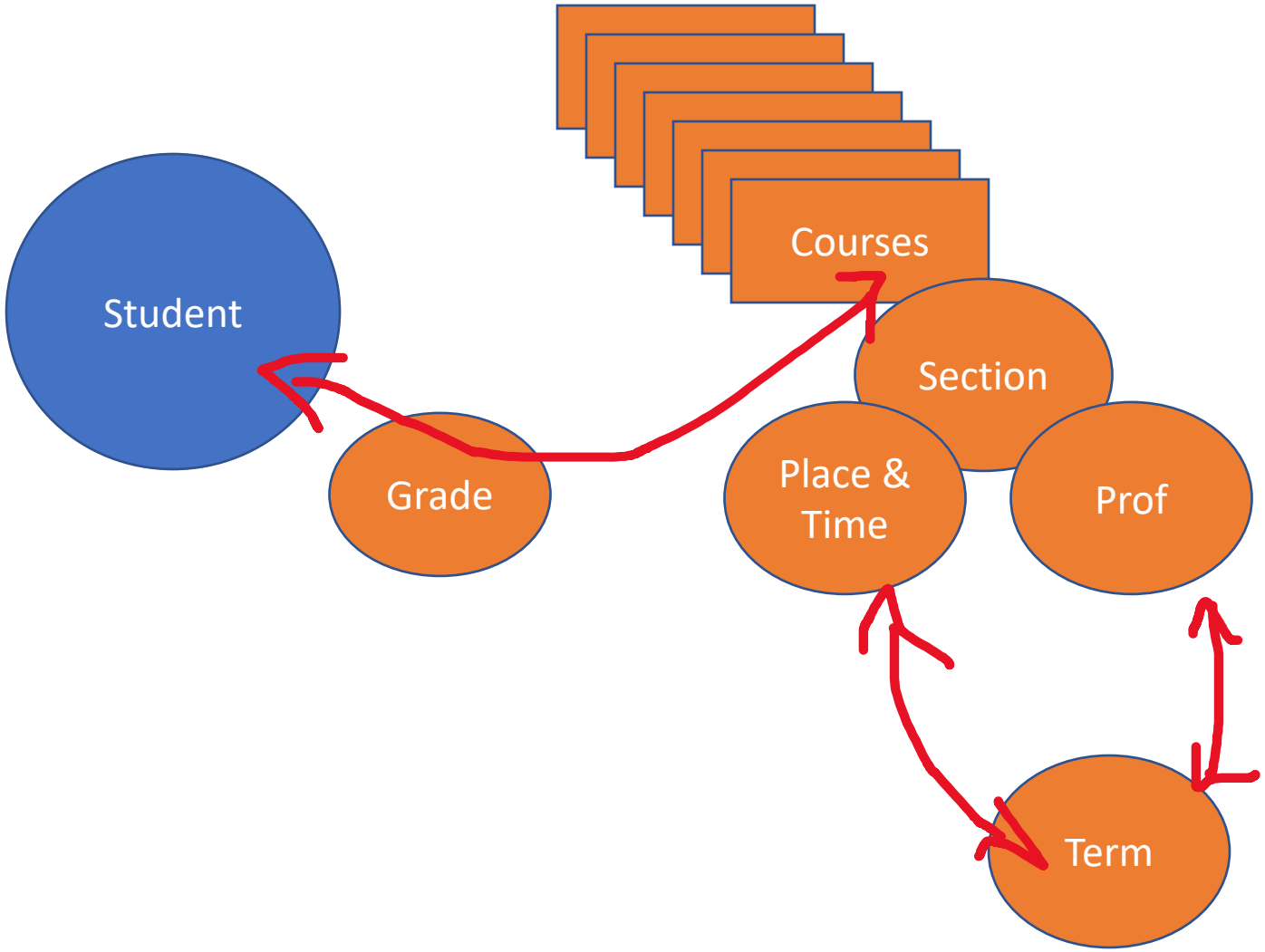
RETRIEVE data

This is what database systems where designed for: checking that changes don't lead to an inconsistent state (two people in the same seat, reservation for a non-existing flight), and also to retrieve data as fast as possible, using a "high-level" language (find this that satisfies those conditions ...) instead of looping on file records and checking each one.

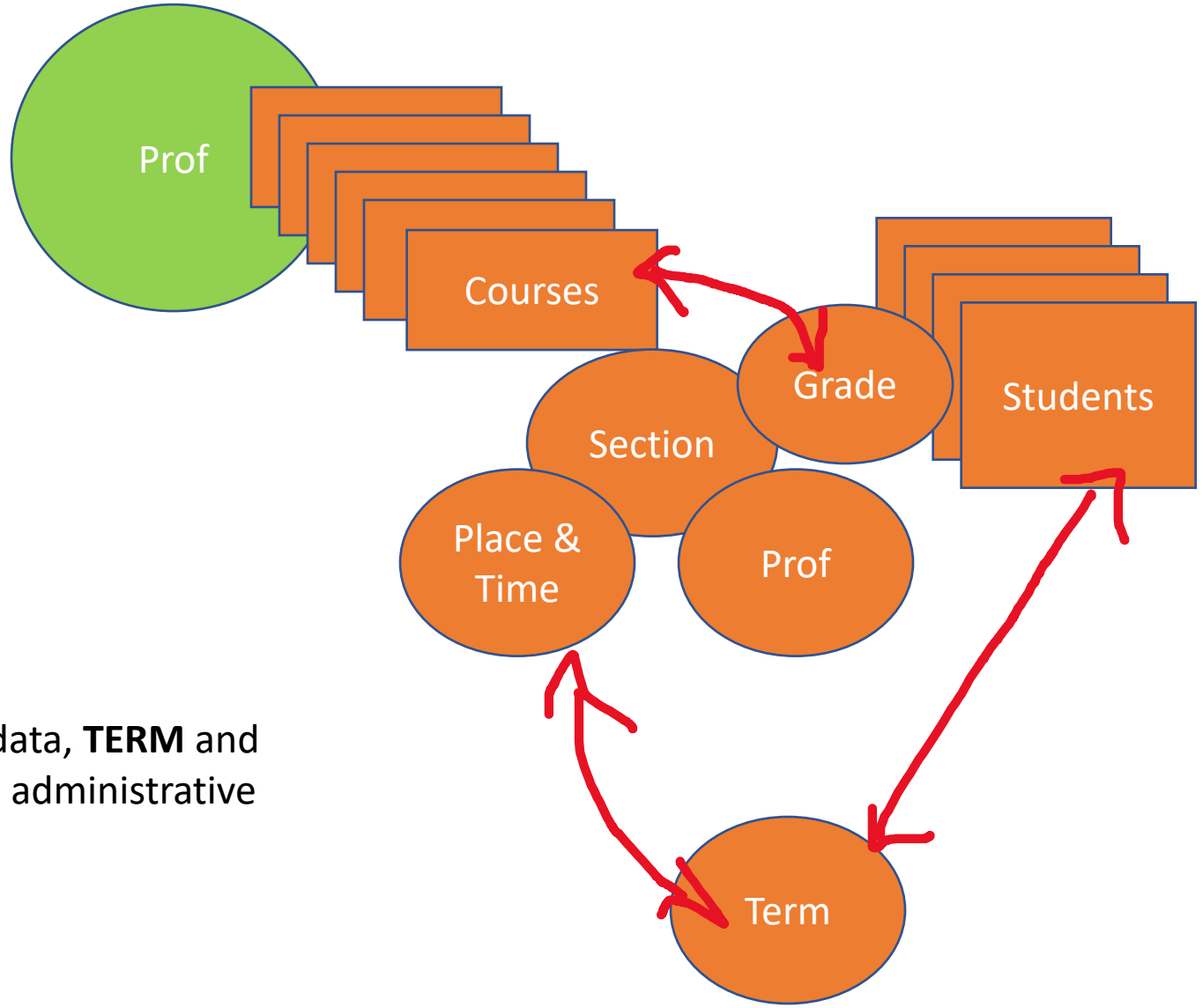
DIFFERENT VIEWS

A key problem with shared data is that people have different views of the same things. When you design an object for a single application, it's relatively easy.

For instance a Student object might contain a collection of Course objects, and the professor is an attribute of the course.



For a prof, each course is associated with a collection of students enrolled in the course.
Same data, presented differently.

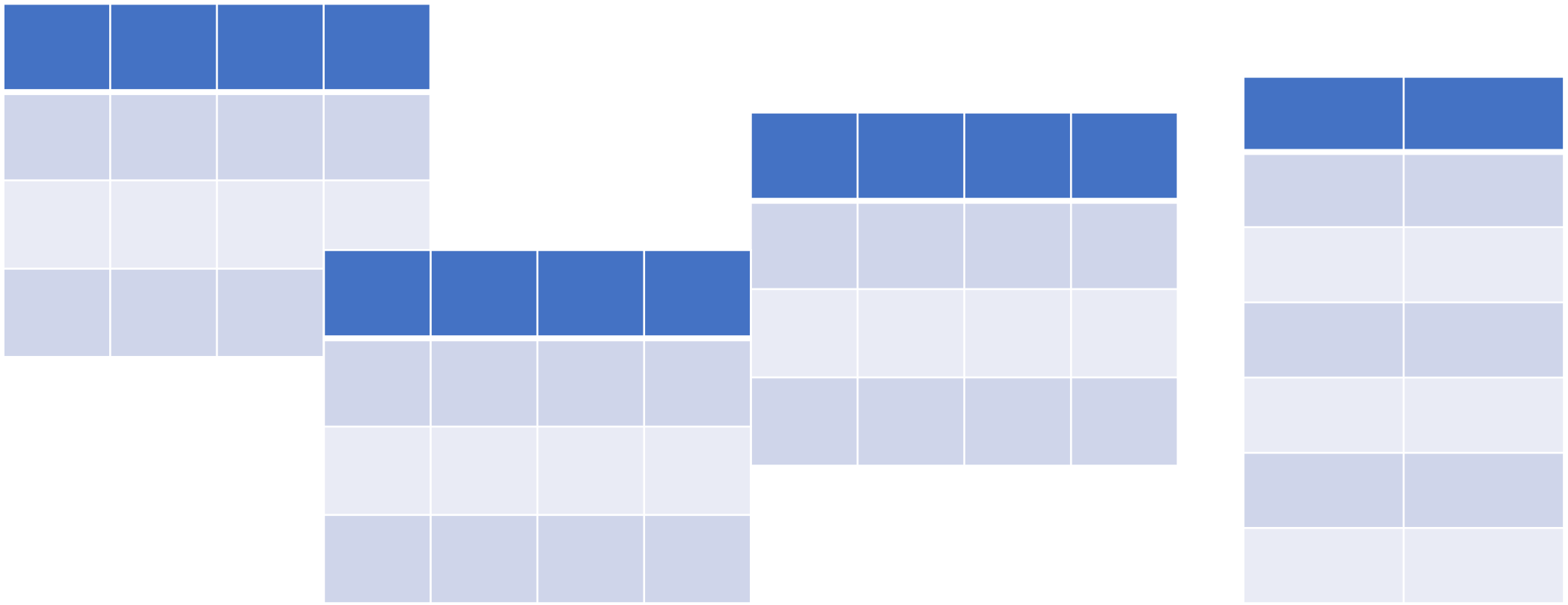


A database must
allow both views of the data, **TERM** and
possibly more views (the administrative
one ...)

Relational Databases

In 1970, a (British) IBM computer scientist described a way to store data in tables in a database, derived from the mathematical set theory. It was the start of a revolution in data management, and most databases in use today are based on his ideas.

Edgar F. Codd
1923 - 2003



Codd said that the database should give a tabular ("as tables") view of data, which is quite natural. Each table contains a set.

Sets

- Sets are unordered. If columns are ordered differently, the information isn't changed. Same with rows.

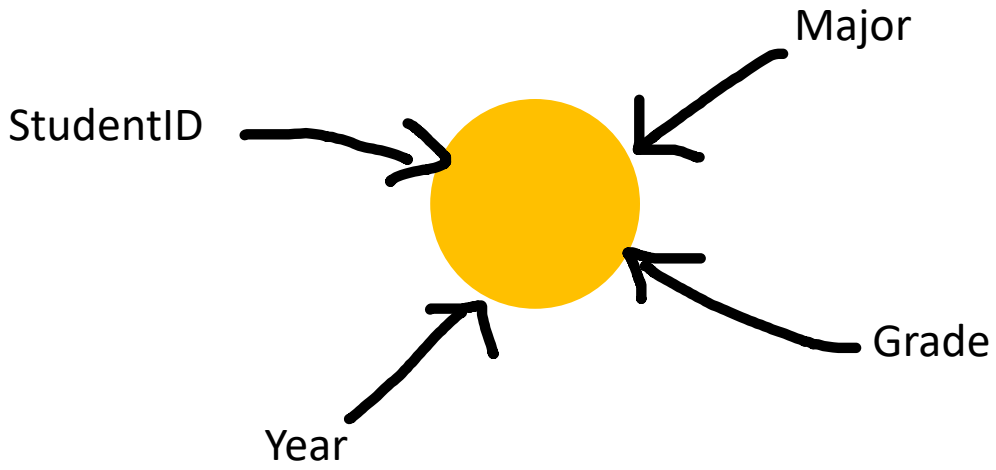
- And Sets don't allow duplicates. It's important that each different row is stored only once – this is done by defining "keys" that allow to specify one particular row.

Stduent ID	Major	Grade	Year

Relational Databases

Because all the attributes are related to a single event, the table is also known as a "relation", hence "relational theory of databases" and "relational database".

A Relation



Coming up next…

Relational Databases
JDBC – Java Database Connectivity API