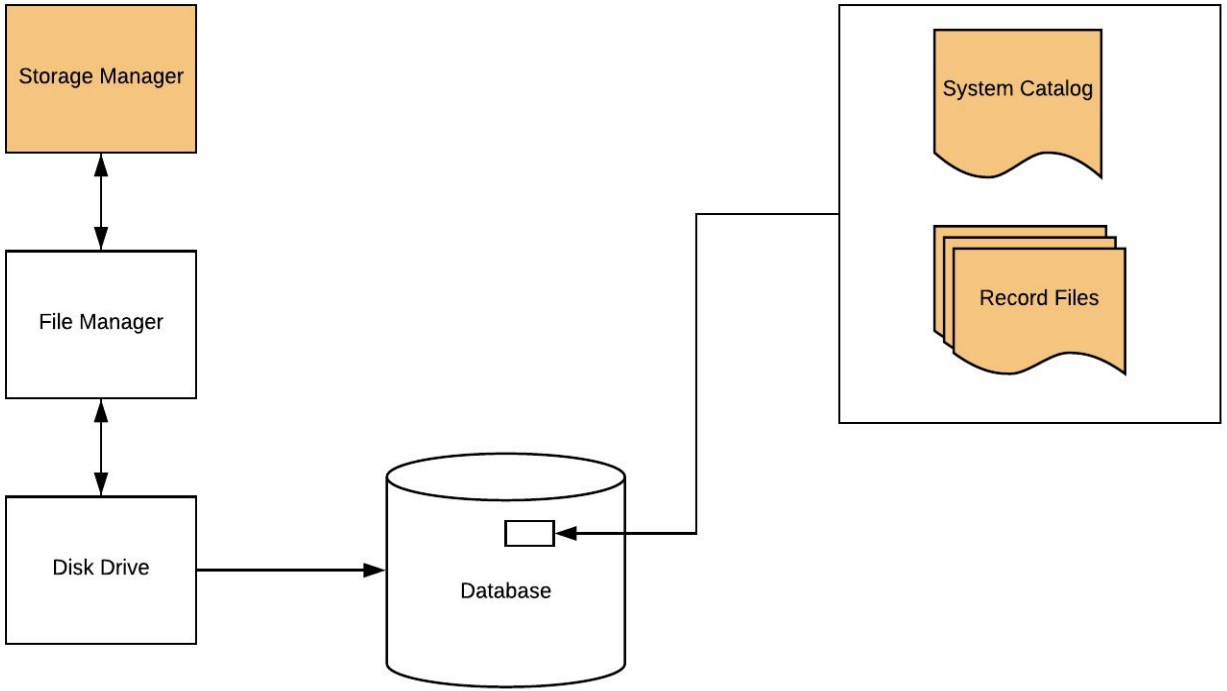
CMPLE 321 - 2018 Spring

ASSIGNMENT 1

By Özlem Şimşek

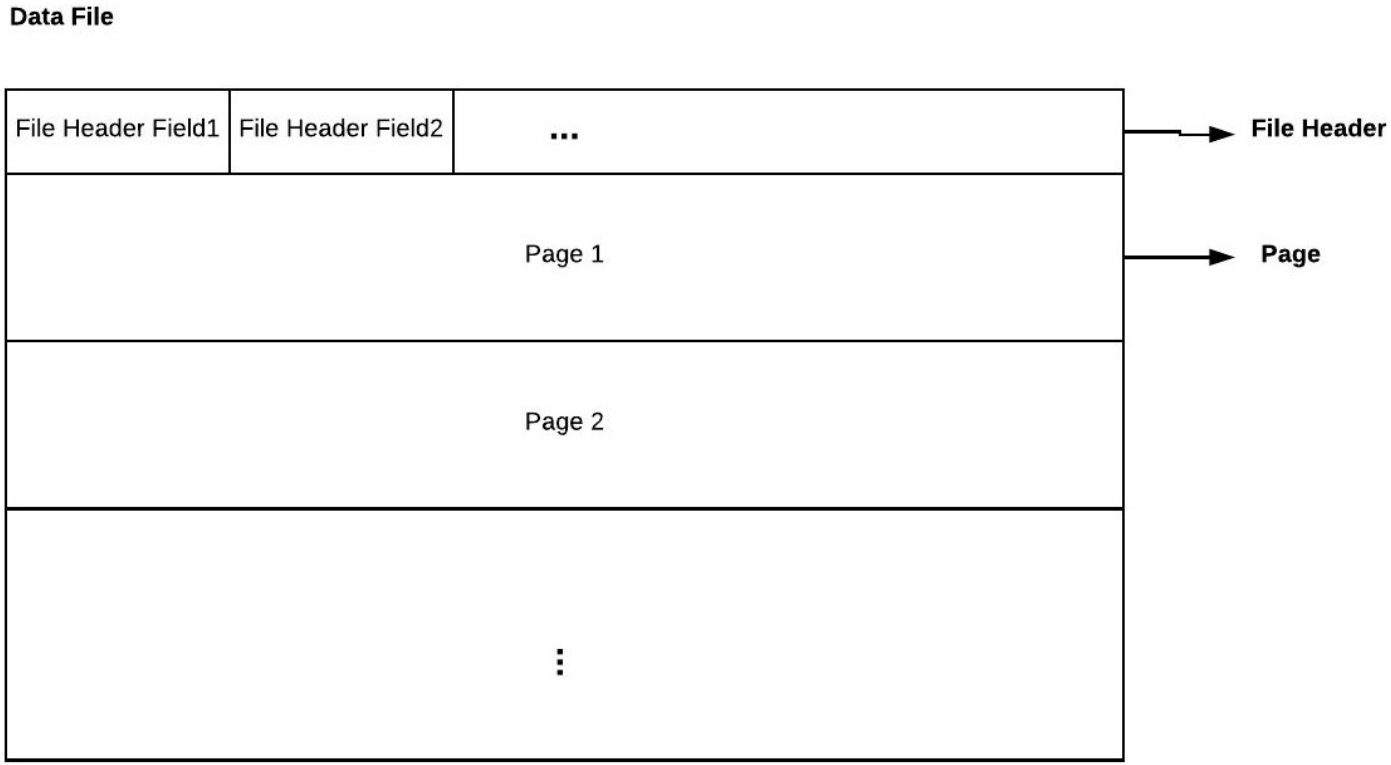
General System



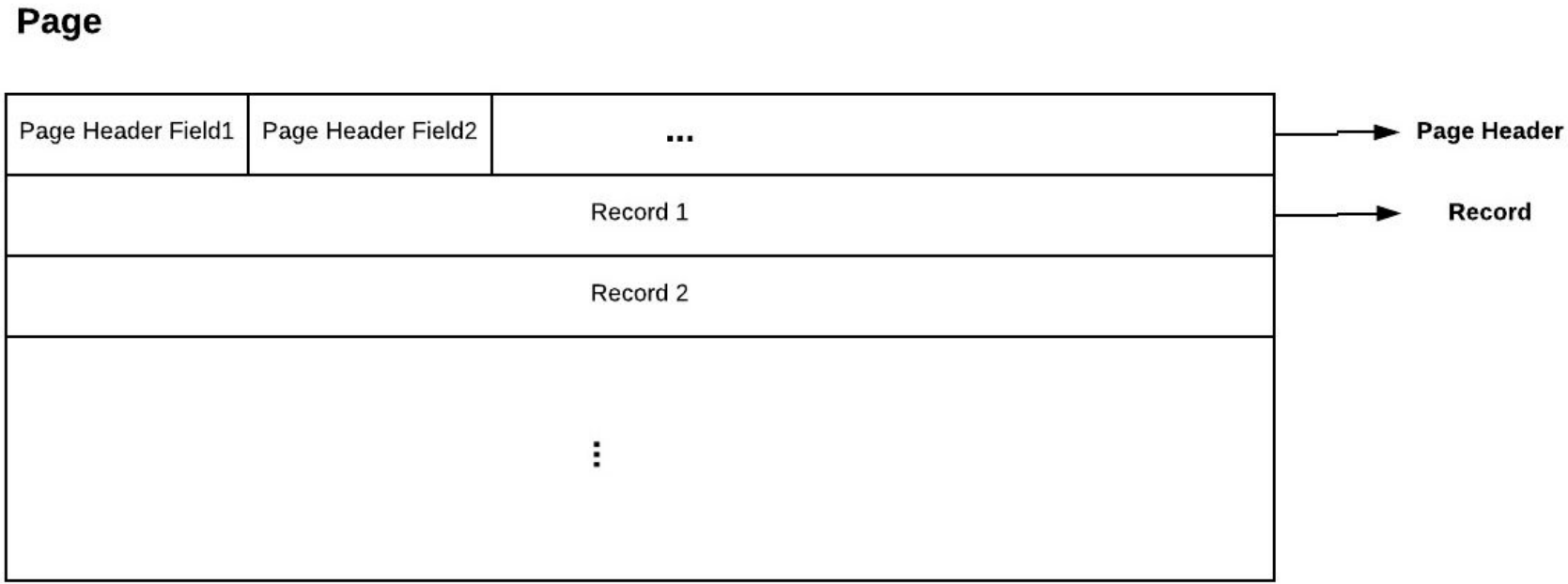
System Elements

- **System Catalog**: Stores metadata of the data system:
  - Record types
  - Files
  - Pages
  - ...
- **Files**: Stores pages of records. A file can store multiple pages. Each file has a file header.
- **Pages**: Stores actual data records. A page can store multiple records. Each page has a page header.
- **Records**: Representation of data. A record is composed of one or more data fields. (i.e. student) Each record has a record header.
- **Fields**: Representation of an attribute. (i.e. student id)
- Records and fields are specified with types.

Files and Pages



Pages and Records

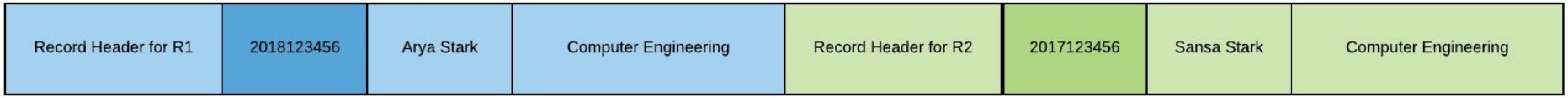


Records and Fields

Record

Example Record : Student Type

```
struct Student{
    int student_id;
    char name[30];
    char department[20];
};
```



Assignment 1 - Due 12.03.2018 23:59

In this assignment, a storage manager system will be designed.

- The design shall have a system catalog which stores metadata, and multiple data files that store the actual data.
- The system needs to support common data handling operations of DDL (Data Definition Language) and DML (Data Manipulation Language).

General Assumptions

- No error check
- Define your assumptions and constraints clearly.
- Write pseudo codes in well format.
  - Use indentation
  - Do not write very long pseudo code
  - Refer to your definitions (i.e. Refer to x field in page header as pageheader.x)

Pseudo Code Example

Algorithm 1 Compute sum of integers in array

```
1: procedure ArraySum(A)
2:     sum = 0
3:     for each integer i in A do
4:         sum = sum + i
5:     end for
6:     Return sum
7: end procedure
```

```
Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while not at end of this document do
    read current;
    if understand then
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end
```

Algorithm 1: How to write algorithms