

다음과 같은 Table을 Flash Memory에 저장하고 관리하는 저장시스템을 만들려고 한다.

Student(INT id, CHAR*30 name, CHAR*20 department, FLOAT score, IMAGE photo)
 Department(CHAR*20 department, INT numberOfStudents, INT numberOfProfessors)
 Professor(INT id, CHAR*30 name, CHAR*20 department, CHAR*30 email)

1. 이 저장시스템이 지원하는 연산은 다음과 같다. 단 각 연산자는 API로 제공되어야 한다.

1) ErrorCode ConnectStorageSystem(char* pathName)

필요한 초기화를 한다.

2) Insert

```
ErrorCode    Insert_Student (Student *student);
ErrorCode    Insert_Professor(Professor *professor);
```

* Department는 따로 입력되는 것이 아니라, 해당 Department의 최초 Record가 나오면 생성하고, number_students와 num_professors를 적절하게 증가시킨다.

3) Delete: 해당 Tuple이 삭제되어야 한다.

```
ErrorCode    Delete_Student(int id);
ErrorCode    Delete_Professor(int id);
```

* Department는 number_students==0 && num_professors==0 이면 삭제한다.

4) List: 해당 Table의 모든 Record를 전달한다.

```
ErrorCode    List_Table(int table_type, Results *results);
```

5) Search: search의 결과로 해당 Tuple의 값이 GUI로 표시되어야 한다.

```
ErrorCode    Search_Student_Id(int id, Results *results);
ErrorCode    Search_Student_Name(char* name, Results *results);
ErrorCode    Search_Student_Department(char *department, Results *results);
ErrorCode    Search_Student_Score(float in, float max, Results *results);
ErrorCode    Search_Department_Id(char *name, Results *results);
ErrorCode    Search_Professor_Id(int id, Results *results);
```

```
ErrorCode Search_Professor_Name(char* name, Results *results);
ErrorCode Search_Professor_Department(char *department, Results *results);
```

6) ErrorCode DisconnectStorageSystem(char* pathName)

프로그램을 종료한다. 필요한 초기화를 한다.

2. 여기서 사용되는 Type은 아래와 같다.

```
#define ErrorCode unsigned int
// unsigned int 로 정의된 Error Code
// 0x00000000 (=E_OK)은 이상 없이 수행되었다는 것을 의미한다.
class Results {
    unsigned int numberOfResults;
    unsigned int typeOfResult; // 1: student, 2: department, 3: professor
    union {
        Student      *students;
        Department   *departments;
        Professor     *professors;
    };
};

class Student {
    unsigned int    id;
    char           name[31];
    char           department[21];
    float          score;
    IMAGE         *image;
};

class Department {
    char           department[21];
    unsigned int   number_students;
    unsigned int   number_professors;
};
```

Class Professor

```
    unsigned int    id;
    char            name[31];
    char            department[21];
    char            e-mail[31];
};
```

* Due data: 2007년 6월 25일

* 제출물: Compile 가능한 C++ Source 프로그램

* 검사: Oral Test

* 주의: Test 용 프로그램으로 수행을 하여 수행시간이 가장 빠른 조에게는 중간, 기말, 숙제의 총합에 대한 10%의 보너스 점수를 주며 소정의 상품도 줄 것임.

* 기타 질문사항은 FP 수업 게시판에 올릴 것.