과제 7: 저장 프로시저, 커서, 트리거, 보안 및 실기시험 준비

이번 과제는 저장 프로시저, 커서, 그리고 트리거, 보안 등의 개념들 이해하고 사용할 수 있는 지를 확인하는 내용으로 실기시험 형식으로 진행합니다. 과제 6에서 사용한 데이터베이스를 전제로 다음 문제에 답하시오.

1. 학사DB와 유사한 데이터베이스를 생성하는데 데이터베이스 이름은 자신 이름과 학번의 마지막 숫자를 사용한다. 가령 홍길동 학생의 학번이 3219\*\*\*3이라면 데이터베이스 이름은 '홍길동3'으로 정한다. '홍길동3' 데이터베이스에는 다음의 테이블들이 존재한다.

홍길동\_student3 (sid3 char(10), sname3 char(10), major3 char(10), gpa3 float)

홍길동\_course3(cid3 char(10), cname3 char(20), instructor3 char(10))

홍길동\_course\_taken3(sid3 char(10), cid3 char(10), grade3 float, year\_taken3 int)

홍길동\_student3의 major3은 **전공 이름**으로 구성되며 student 데이터 중 학번 마지막 수가 3인 경우는 제외한다. 홍길동\_course3에서 instructor3는 교수 이름으로 구성된다. 홍길동\_course\_taken3에서 기본 키는 밑줄 친 부분으로 수정하고 외래키 설정은 학사DB의 course\_taken의 정의대로 한다.

(a) 위에서 제시한 대로 테이블들을 정의하시오.

(b) 위에서 요구한 대로 학사DB로 부터 데이터를 입력하시오.

(c) 다음을 실행하시오.

select \*

from 홍길동\_student3 natural join 홍길동\_course\_taken3 natural join 홍길동\_course3

order by sid3;

2. 수강 테이블(홍길동\_course\_taken3)에 학생의 수강 과목과 성적을 입력하는 저장 프로시저 '홍길동\_AddCourseGrade3'를 구현하고 이의 사용 예를 보이시오. 단 이는 매개변수로 학생 이름(pStud\_name3), 과목 이름(pCo\_name3), 그리고 성적(pGrade3), 그리고 pIsError3라는 int 변수를 입력 받고 프로시저 바디에서 현재의 날짜로부터 year 정보를 입력할 수 있도록 한다. pIsError는 식별할 수 없는 학생 이름이 입력되는 경우는 1을, 식별할 수 없는 과목 이름의 경우는 2를, 이 둘의 경우는 3, 그리고 그 이외의 경우는 4를 반환하도록 한다. 구현에 에러 핸들러는 반드시 포함해야 합니다.

(a) 구현 내용을 보이시오.

(b) 정상 실행 경우와 각각의 에러가 발생한 경우를 보이시오.

3. 학생들의 평점(GPA)을 수강 내역으로부터 계산하는 저장 프로시저(홍길동\_ComputeGPA3)를 **커서**를 통해 구현하려고 한다. MySQL에서는 read only 커서만을 제공하므로 갱신은 update문을 사용해야 한다. 구현 내용과 실행 결과를 보이시오.

4. 수강 내역 테이블 즉 홍길동\_course\_taken3은 학생들의 수강 과목의 학점들이 포함되는 중요한 정보다. 따라서 수강 내역 테이블에 대한 audit 정보를 담는 홍길동\_course\_taken\_Audit3 테이블을 정의하고 trigger들을 통해 audit 정보를 입력하려고 한다. 홍길동\_Course\_taken\_Audit3 테이블의 필드들은 일련번호, 홍길동\_course\_taken3의 기본 키, 사용자 정보, 실행 연산, 그리고 수정 시간으로 구성된다. 사용자 정보는 user() 함수의 값을 사용하도록 하는데 User()는 현 세션의 사용자를 반환한다.

(a) insert, delete, update시에 사용될 Trigger들의 정의를 보이시오.

(b) 위의 trigger들이 정상 작동함을 보이는 예제들을 보이시오.

5. 장학생(홍길동\_ScholarshipStudent3) 테이블을 view로 정의하려고 한다.

(a) 이를 정의하고 실행 결과를 보이시오.

(b) MySQL의 Check Option의 의미를 설명하고 이를 장학생 뷰에 적용했을 때 어떤 변화를 보이는 지를 보이시오.

(제외) 6. 학사DB의 사용을 위한 사용자 계정로 student, professor, academicManager를 생성한다. academicManager는 학사DB 전체에 대한 모든 접근 권한과 이에 권한 부여 능력을 갖는다. 한편, student는 모든 테이블들에 대한 select 권한만을, professor는 모든 테이블들에 대한 select 권한, 그리고 course\_taken에 대해 insert, update 권한만을 갖도록 한다. Root 사용자 세션에서 실행했을 때, 위의 작업이 가능한 script 파일을 생성하고 show grants for 문을 통해 각 사용자의 권한을 확인하도록 한다. 스크립트 파일의 내용과 실행화면을 출력하여 제출하시오.