

2024 Spring Database

PROJECT 1-3 INTRODUCTION

Project 1-3

- Goal

- Project 1-1 및 1-2 코드에 기반하여
- 데이터 삽입, 삭제, 검색 기능 구현

- Prerequisites

- 테이블이 정상적으로 생성되는 지 확인

- Implementing Queries

- INSERT, DELETE, SELECT문을 구현
- 메시지 정의문서(2024_1-3_Messages.docx) 참조

Prerequisites

- 프로젝트 1-2에서 구현한 CREATE 기능
- 구문 4가지를 차례대로 입력 시, 정상적으로 4개의 스키마가 생성되어야 함.

<pre>create table students (id char (10) not null, name char (20), primary key (id));</pre>	<pre>create table lectures (id int not null, name char (20), capacity int, primary key (id));</pre>
<pre>create table ref (id int, foreign key (id) references lectures (id));</pre>	<pre>create table apply (s_id char (10) not null, l_id int not null, apply_date date, primary key (s_id, l_id), foreign key (s_id) references students (id), foreign key (l_id) references lectures (id));</pre>

실행 예시 - 1

■ INSERT

```
DB_2023-12345> insert into account values(9732, 'Perryridge');  
DB_2023-12345> 1 row inserted
```

- 올바른 INSERT문의 실행
- 쿼리 오류에 대한 종류별 메시지 출력
- PK/FK constraint에 위배되지 않는 쿼리만 Test Case로 주어짐을 보장

■ DELETE

```
DB_2023-12345> delete from account where branch_name = 'Perryridge';  
DB_2023-12345> 5 row(s) deleted
```

- 올바른 DELETE문의 실행
- 쿼리 오류에 대한 종류별 메시지 출력
- PK/FK constraint에 위배되지 않는 쿼리만 Test Case로 주어짐을 보장

실행 예시 - 2

■ SELECT

- 출력 시 테두리, 필드 공백 제한 없음
- 출력되는 column name과 data를 기준으로 채점
- 출력 column name 앞에 table name을 붙여도 무방
e.g., 아래 예시에서, 출력 column명으로
borrower.loan_number, loan_number 모두 허용

```
DB_2023-12345> select * from account;
+-----+-----+-----+
| account_number | branch_name | balance |
+-----+-----+-----+
| A-101          | Downtown   | 500     |
| A-102          | Perryridge | 400     |
| A-201          | Brighton   | 900     |
| A-215          | Mianus     | 700     |
| A-217          | Brighton   | 750     |
| A-222          | Redwood    | 700     |
| A-305          | Round Hill | 350     |
+-----+-----+-----+

DB_2023-12345> select customer_name, borrower.loan_number, amount
from borrower, loan
where borrower.loan_number = loan.loan_number and branch_name = 'Perryridge';
+-----+-----+-----+
| customer_name | loan_number | amount |
+-----+-----+-----+
| Adams         | L-16        | 1300   |
| Hayes         | L-15        | 1500   |
+-----+-----+-----+
```

Where clause

- where절의 조건의 개수에 대한 규칙
- where절을 구성하는 condition은 최대 2개이며,
- 괄호로 묶인 복잡한 condition은 고려하지 않는다.
 - condition이 하나인 경우
where condition1
 - condition이 두개인 경우
where condition1 and condition2
where condition1 or condition2

Where clause 내 비교연산자

- where절 안에 비교연산자는 아래와 같은 조건에서만 비교가 가능하다.
 - ▣ char 타입의 값은 길이와 상관없이 다른 char 타입의 값과 비교(=, !=)할 수 있다.
 - ▣ int 타입의 값은 다른 int 타입의 값과 비교(>, <, =, !=, >=, <=)할 수 있다.
 - ▣ date 타입의 값은 다른 date 타입의 값과 비교(>, <, =, !=, >=, <=)할 수 있다.
 - ▣ null 은 다른 모든 타입의 값과 비교(is, is not)할 수 있다.

Data Type

- char(n)
 - 고정길이 문자열을 나타내는 데이터 타입
 - E.g., 'Word', '30', ...
- int
 - 정수형 데이터 타입
 - 본 프로젝트에서는 0과 양의 정수만 포함
 - e.g., 0, 1, 2, ...
- date
 - YYYY-MM-DD형식의 날짜를 나타내는 데이터 타입
 - 본 프로젝트에서는 아래의 범위 내의 날짜만 고려하면 된다.
 - YYYY: 2000~2030
 - MM: 01~12
 - DD: 01~31
 - e.g., 2023-10-20
- 위 세가지 데이터 타입 모두 value로 null을 허용
 - 데이터가 존재하지 않음을 나타냄
 - e.g., null

Additional Assignment (Extra Credit)

- 문서의 **2.3.Primary key/Foreign key Constraints**를 구현한 학생들에게 프로젝트 1-1 및 1-2 성적을 만회할 수 있는 가산점을 최대10점 제공
- INSERT/DELETE 구문에서 PK/FK constraint를 구현
 - InsertDuplicatePrimaryKeyError
 - InsertReferentialIntegrityError
 - DeleteReferentialIntegrityPassed

(주의)

- Optional 추가 구현으로 인해 본래 INSERT/DELETE 구문 기능 코드에 오류가 발생하여 본 점수에서 감점되지 않도록 주의
- Optional 문제 관련 Q&A는 진행하지 않음

FAQ

- Q. INSERT/DELETE에서 추가 구현사항을(PK/FK constraint) 구현하지 않아도 감점이 없나요?
 - 2.1.Prerequisites와 2.2.SQL Queries를 모두 구현한 경우, 감점되지 않습니다.
- Q. 이번 프로젝트에서 만점을 받아야 가산점을 받을 수 있나요?
 - 아닙니다. 본 프로젝트 결과와 관계 없이 Optional 문제 구현을 완성하면 그에 대한 점수를 받을 수 있습니다.
- Q. Optional 문제를 구현했는데 이미 1-1, 1-2에서 만점이면 어떻게 되나요?
 - 이미 두 프로젝트 모두 만점이거나 가산점 일부를 받아 모두 만점이 되는 경우 잔여 점수를 추가해 드리지는 않습니다.

FAQ

- Query sequence가 들어올 경우 어떻게 처리하나요?
 - query1;query2;query3; 과 같이 들어온 경우, query1, query2, query3 을 순차적으로 처리.
 - ex) queryK(K번째 쿼리)에 에러가 있을 경우,
 - query1부터 queryK-1까지는 정상적으로 처리하고,
 - queryK 관련 에러 메시지를 출력

제출 관련 안내

- **Due: 5/21(화) 23:59pm**
- 자세한 설명은 **프로젝트 문서**를 참고
- 프로젝트 문서에 정의되지 않은 요소를 추가로 구현하고 싶다면,
 - **MySQL** 기준으로 구현
 - 해당 내용을 리포트에 적어서 제출
- 프로젝트 관련 문의: **5/17 (금)까지**, ETL 게시판 사용
- 개별 코드 구현에 대한 문의는 형평성에 따라 불가합니다.