

# HW9 - DB Application Programming

마감일: 2024.12.15.

## 1. 숙제 목표

- 모델에서 데이터베이스 구현
- 요구사항으로 부터 유효한 트랜잭션 작성
- 트랜잭션 워크로드 테스트
- 응용 프로그램 설계 및 구현

## 2. 배경

HW8에서 호텔 관리 시스템 데이터베이스를 설계하였다. HW8에서는 비즈니스에 필요한 질의, 트랜잭션과 데이터베이스를 구현한다. 애플리케이션 구현을 위하여 C, C++ 또는 Java 프로그래밍 언어를 사용한다(**Python 사용 안함**). 또한 SQL 실습을 위해 설치하였던 PostgreSQL을 기본 데이터베이스로 활용한다.

실제 사용되는 응용 프로그램은 웹이나 앱 형태의 GUI를 제공하여 사용자 친화적인 프로그램으로 구현되어 있다. 하지만, 이 숙제에서 학생들은 데이터베이스에 프로그래밍에 집중하기 위하여 명령형 인터페이스(command line interface)와 같은 단순한 상호 작용 방식으로 프로그램을 구현한다.

## 3. 숙제 요구 사항

응용 프로그램에 대해 두 개의 데이터베이스 클라이언트 (관리자 클라이언트와 고객 클라이언트) 기능을 구현한다. 각 기능은 한정된 질의 및 트랜잭션 셋을 지원한다.

(1) 두 클라이언트 모드를 실행하기 위한 로그인 기능 제공

응용 프로그램을 실행하기 위해서 로그인 기능이 필요하다. 관리자 모드 로그인과 일반 사용자 로그인의 두 가지 옵션을 제공한다. 일반 사용자의 경우 고객 ID (또는 데이터베이스 설계에 사용한 고객에 대한 유일한 식별자, 기본 키 등)를 이용한다.

관리자모드: login('manager')

사용자모드: login('customer', unique\_id)

이 숙제에서 인증에 대해 걱정할 필요는 없다. 각 사용자가 실행할 수 있는 제한된 유형의 질의로 인하여 각자 다른 권한을 부여받는 것이다.

(2) 각 클라이언트 모드별 명령 구현

HW8에 설명한 대로 시나리오에 대한 질의 및 트랜잭션을 구현한다. 데이터베이스 디자인에 따라 각 질의/트랜잭션에 대한 적절한 인자(argument)를 결정한다. 예를 들어 호텔에서

**check\_in**은 예약 번호를 하나의 인자(argument)로 가질 수 있다.

(a) 지원 질의

- 관리자 모드용 질의

a. **rooms\_occupied**: view the currently occupied rooms

b. **housekeeping**: list house-keeping assignments

- 일반 사용자 모드용 질의:

a. **rooms\_available**: view the room types and costs that are still available

b. **cost\_at\_checkout**: calculate the total cost for the guest at checkout time

c. **my\_reservations**: list future reservations for the guest

(b) 지원 트랜잭션

- 관리자 모드용 트랜잭션

a. **check\_in**: check-in a guest (must be day of reservation or fail)

b. **check\_out**: check-out a guest (if the checkout is early then calculate the refund to the guest)

c. **mark\_serviced**: record that a room has been serviced for the day; no room should be serviced twice in the same day

- 일반 사용자용 트랜잭션:

a. **reserve**: make a reservation (denied if no more rooms of requested type)

b. **cancel**: cancel a reservation (denied if same day)

(3) 테스트 프로그램 및 예제 상호 작용

HW9은 프로그래밍 인터페이스를 구축하면 된다. 그러나 클라이언트를 작성하고 상호 작용을 시뮬레이션하기 위해 일련의 함수 호출을 발행하는 여러 테스트 프로그램을 제공한다. 흥미로운 상황을 다루기에 충분한 프로그램을 작성한다. 테스트 프로그램은 HW8에 나열된 상황에서 작성해야 하지만 부가적인 기능을 추가하는 것도 가능하다. 보고서에 잘 설명하면 추가 점수를 부여한다.

## 4. 구현 참고 사항

(1) C++/Java인 경우 메소드가 다른 질의 및 트랜잭션인 클래스 (예 : ManagerClient 및 CustomerClient)로 두 가지 기능 모드를 구현한다. C인 경우 적절한 함수로 구현을 한다.

(2) C++ 또는 Java에 대한 쿼리 및 트랜잭션을 발행하는 방법은 예제 애플리케이션을 참조한다.

(3) 트랜잭션 isolation level을 'serializable'로 설정하는 것을 권장한다. 예제 애플리케이션이나 강의 교재를 참고하면 설정 방법을 알 수 있다.

(4) 트랜잭션은 실제 응용 프로그램과 일치하는 방식으로 작동해야 한다. PostgreSQL에 교착 상태가 표시되어 트랜잭션이 실패하면 트랜잭션을 다시 시도한다. 자원이 고갈되어 (예 : 객실, 돈) 거래가 실패하면 사용자에게 오류를 반환한다.

- (5) 필요에 따라 추가 질의나 트랜잭션등의 기능을 자유롭게 추가하여 구현할 수 있다.
- (6) 프로그래밍 방식 이외의 인터페이스 (예 : 클래스 및 메서드)를 만들 필요는 없다. 모든 메소드들은 유용한 메시지를 사용자에게 출력해야 한다.
- (7) 코드는 잘 문서화되어야 한다. 각 함수에 대하여 동작 기능, 매개 변수 및 반환 값을 잘 설명한다. 필요한 경우 주석 코드를 작성한다.

## 5. 제출 내용 및 시연 방법 안내

- (1) 소스 코드와 pdf로 변환한 보고서 파일을 묶어서 하나의 zip 파일 생성
- (2) Plato를 이용하여 zip 파일
- (3) 시연 방법
  - Docker 형태로 데이터베이스 자체를 같이 올리지 않는다면, 프로그램은 컴파일되지만 실제 동작을 확인할 수는 없습니다.
  - 추후 제공하는 구글 캘린더 링크를 통하여 DataLab 방문 시간이나 줌 데모 시연 날짜를 잡음

- 프로그램 시연은 12월 23일, 24일 DataLab을 방문하거나 줌을 통하여 최대 10분동안 시연함

## 6. 기타사항

- (1) 채점 기준 (100점 환산 기준)
- (2) 숙제 중 가장 많은 비중을 차지함
- (3) 12월 20일 23:59이후에는 0점 처리함
- (4) Copy가 발견되었을 경우 해당하는 모든 사람을 0점 처리함