# Individual Assignment 1 & 2

DB 설계 및 디자인 2, 2023학년도 2학기

* 제출시한: 10월 8일 (일요일), 11:59 PM, (LMS를 통해 제출).
* 제출하는 파일명은 “**숙제1&2\_홍길동.docx**”와 같이 자신의 이름을 마지막에 붙이시오.
* 설명을 요하는 문제의 답을 작성시에는 **자신의 이해를 충분히 보일 수 있는 자신의 언어**로 설명하시오. **설명이 미진하거나 슬라이드를 단순히 베끼는 등 학생의 이해여부를 확인할 수 없거나 성의가 없으면 감점임.**
* 모델링의 경우 펜을 사용하는 등 수기로 작성하면 20% 이상의 패널티가 있으니 수기작성하지 마시오. 아래에 설명한대로 반드시 자신이 선택한 소프트웨어를 사용하여 수업시간에 배운대로 깔끔하게 작성하시오.
* 질문 있을 시 ijkim@mju.ac.kr으로 질문하시오.

Weights (비중): 4% of total grades.

1. List and discuss three disadvantages of file processing systems and briefly discuss them. (file professing systems의 단점을 세가지 적고 이를 설명하시오.) (20 points).

File processing systems의 단점은 크게 데이터의 복제, 데이터 공유의 제한, 데이터 의존성이다. 우선, 데이터의 복제는 데이터의 중복과도 관련이 있는데, 데이터를 수정할 때마다 데이터 파일이 달라지기 때문에 수많은 복제본이 생기고, 이를 통해 여러 파일들 사이의 데이터 값이 달라지며 무결성 법칙을 위배한다. 또한, 여러 컴퓨터에 있는 파일들 중에 단 하나라도 값이 update된다면 이를 다른 컴퓨터들도 모두 다 다운받아야 하기 때문에, memory 낭비 등 여러 문제를 일으킨다. 따라서 데이터의 중복은 file processing system의 치명적인 단점이라고 할 수 있다.

두번째로, 데이터 공유의 제한인데, 이는 앞서 말한 데이터의 복제와 연관이 깊다. 데이터를 update했을 경우, 이를 공유해야 하는데, 공유하려면 file processing system에서는 file을 모두 복제해서 email이나 다른 수단을 통해 update 이전 파일을 가지고 있던 컴퓨터에 직접 file을 복제해 공유해야 한다. 이는 당연하게 업무의 효율을 떨어뜨리고, 시간 낭비를 촉구한다.

마지막으로 데이터 의존성이다. 데이터 의존성은 프로그램의 구조가 데이터의 구조에 영향을 받는 것을 뜻한다. File processing system에서는 각 응용 프로그램마다 데이터를 개별적으로 관리하기 때문에, 데이터의 중복 문제를 일으킬 수 있다. 또한, 각 응용 프로그램이 데이터파일의 메타 데이터 즉, 데이터의 구조 및 설명을 처리해야 해서, 파일 구조가 변화하면 프로그램 코드 또한 모두 수정해야 할 수도 있다. Dbms와 달리 data read, insert, update, delete기능이 없기 때문에 이 또한 프로그램에서 코드를 작성해야 한다.

1. Research the Internet for a business success story of using database or database related technologies (e.g., business intelligence). Discuss how the application was implemented using databases (or relevant technologies) (데이터베이스 혹은 관련 테크놀로지를 사용한 성공사례를 조사해 보시오. 어떠한 사례인지, 어떻게 데이터베이스나 관련 기술을 사용하여 어플리케이션이 구현되고 사용되었는지 소개하시오.) (30 points)

Airbnb는 숙박 공유 플랫폼으로, 호스트가 자신의 집이나 숙소를 빌려주고, 여행객이 이를 대여하는 서비스를 제공한다. Airbnb는 호스트 및 여행객 사용자 정보와 프로필을 관리하는 기능과 숙소를 검색하고 예약할 수 있는 기능, 호스트와 메시지를 전송하고 수신할 수 있는 기능, 예약 및 결제 처리 기능, 리뷰 작성 및 평가 종합 기능, 데이터 분석 및 운영 기능, 클라이언트-서버 통신 기능을 제공한다.

우선, Airbnb는 사용자의 정보와 프로필 데이터를 저장하고 관리하는 데에 MySQL 데이터베이스를 사용하고, 사용자 인증 및 보안을 통해 암호화와 접근 제어 기술을 사용한다. 이는 정형 데이터를 처리할 때 가장 중요한 데이터 일관성과 무결성을 MySQL이 보장하기 때문이다. 또한, 사용자 관리와 관련된 작업은 트랜잭션이 종종 필요한데, MySQL은 ACID 트랜잭션 속성을 지원하기 때문에, 데이터의 일관성 및 안정성을 보장한다. 이외에도, 데이터베이스의 수직, 수평 확장 지원 및 신뢰성, 안전성 등의 이유로 Airbnb는 프로필 저장 기능 외에도 리뷰 작성 및 평가 기능에도 MySQL을 자주 활용한다.

숙소 검색 및 예약 기능은 MySQL과 Elasticsearch를 사용한다. 호스팅 목록과 예약정보는 MySQL을 통해 저장하고 빠른 숙소 검색 및 검색 결과 랭킹 제공 기능은 Elasticsearch를 통해 제공한다. Elasticsearch는 전문적인 텍스트 검색 및 분석을 위해 설계된 오픈소스 검색 엔진으로, 검색 및 랭킹을 빠르게 처리하는 데에 최적화 되어있어서 검색 및 랭킹 기능에 자주 사용된다. 이외에도 Elasticsearch는 검색기능에 필터, 정렬, 집계옵션을 활용할 수 있고, 다양한 데이터 형식과 복잡한 쿼리를 처리하여 유연성이 좋다. 또한 수평확장이 가능하여 대량의 데이터와 사용자 요청을 처리할 수 있어 대규모 트래픽을 처리하는 서비스에 이점을 제공한다.

호스트와 메시지를 전송, 수신할 수 있는 기능은 MySQL과 Redis를 사용하는데, Redis는 메시지 큐 및 세션 관리를 위해 활용한다. Redis는 메모리 기반의 키-값 저장소로, 빠른 읽기와 쓰기 작업을 지원하기 때문에 실시간 메시징 및 활동 알림 기능에 사용한다. 따라서 성능향상 및 세션관리를 위해 Redis를 사용한다.

예약 및 결제 처리 기능은 Apache Kafka를 사용하여 실시간 예약 및 결제 처리 기능을 제공하고, MySQL을 통해 결제 정보와 예약 정보를 안전하게 저장한다. Apache Kafka는 이벤트 스트림 처리를 위한 분산 메시징 시스템으로, 예약 및 결제와 같은 이벤트를 생성하고 처리하기 위해 사용하며, 이벤트의 비동기 처리로 인해 작업의 독립성과 서비스와 확장성을 향상시킬 수 있다. 또한, Apache Kafka는 분산 아키텍처를 기반으로 고가용성 및 내결함성을 제공하는데 이를 통해 안전한 예약 및 결제 이벤트를 처리하고 데이터의 손실을 방지한다.

데이터 분석 및 운영 기능은 대부분 Hadoop 및 Apache Spark를 활용하며, Aerospike와 같은 NoSQL 데이터베이스를 사용하여 일부 데이터를 저장하고 검색한다. Hadoop과 Apache Spark는 대규모 데이터 처리를 위한 분산 데이터 처리 프레임 워크이고, 배치 처리 및 실시간 스트리밍 처리를 모두 지원한다. 또한, 데이터 과학자와 분석가가 데이터를 분석하고 예측 모델을 개발하는데 사용되어 예약 예측 모델을 개발한다. NoSQL은 비정형 데이터를 저장하고 검색하는 데에 활용되어 지리적 데이터와 같은 데이터를 관리한다.

클라이언트와 서버 통신 기능은 GraphQL을 사용하여 효율적인 데이터 통신을 구현한다. GraphQL은 사용자가 필요로 하는 데이터를 지정하여 요청하고 응답하는 데에 사용된다. GraphQL은 클라이언트가 필요로 하는 데이터를 명시적으로 요청할 수 있는 쿼리언어이다. 이는 오버 페칭이나 언더 페칭과 같은 문제를 방지하고, 불필요한 데이터의 전송을 주려 네트워크 트래픽을 절약하는데에 도움이 된다. 또한, GraphQL을 사용하면 클라이언트는 단일 요청에서 여러 데이터 소스로부터 데이터를 가져온다. 따라서 다수의 API 호출이 필요하지 않고, 데이터를 효율적으로 수집하여 빠르고 효율적인 요청 및 응답을 가능하게 한다. 이외에도 단일 엔드포인트 사용으로 API 버전 관리 단순화 등을 돕는다.

1. Explain the differences between user views (external schema), a conceptual schema, and an internal schema as different views of the same database. And discuss the advantages of the three-level architecture. (같은 데이터베이스를 바라보는 서로 다른 관점으로서의 user views (external schema), conceptual schema, 그리고 internal schema(의 차이점)을 설명하시오. 그리고 three-level architecture의 장점을 논하시오.) (20 points).

External schema는 user와 가장 가까운 level의 schema이다. 사용자의 권한, 원하는 기능에 따라 database 중 일부분만 담는 view와 같은 역할이다. Conceptual schema는 database의 content를 modeling하는 schema이다. 따라서 database의 전체적인 내용을 담고 있다고 볼 수 있다. Conceptual schema의 대표적인 예시로 ERD(Entity-Relationship diagram)을 들 수 있다. Internal schema는 physical적인 저장소 구조를 나타낸다. 실질적인 db를 구현하려 하는 관점인 schema이다. logical schema, physical schema등의 예시로 들 수 있다. Logical schema에서는 정규화를 통해 database를 디자인하고, physical schema에서는 데이터 공간의 할당, 암호화 등의 내용을 나타낸다. Conceptual schema와 Internal schema을 비교하자면, 본격적으로 db를 구현하기 전에, 큰 그림을 그리고 내용을 구성하는 관점과 db를 본격적으로 구현해 나가는 관점으로 나누어 볼 수 있다. Internal schema에서는 conceptual schema에서 고려하지 않는 성능, 유지, 보수 등을 신경 쓴다.

Three-level architecture의 주요 목표는 데이터 독립성 제공이다. 우선, external level에서 external schema는 conceptual level의 conceptual schema를 변경하더라도 영향을 받지 않는다. 이를 logical data independence라고 한다. 또한, conceptual schema는Internal schema의 데이터 저장이나 관리 세부 정보를 변경하더라도 이에 영향을 받지 않는다. 이를 physical data independence라고 한다. 이러한 데이터 독립성은 three-level로 architecture를 구분했기 때문이다. 또한 하나의 사용자 view가 다른 사용자 view에 영향을 주지 않는다. 즉, 한 사용자가 database schema나 응용프로그램을 변경해도 영향을 주지 않는다. DBA가 conceptual 구조를 변경하더라도, 다른 사용자나 응용 프로그램에 영향을 미치지 않는다.

1. Describe and explain the three design stage of database development process, and the input/output of each design stages. (수업시간에 배운 **데이터베이스 개발 프로세스**의 **세가지 디자인 stage**가 무엇인지와 그 **input, output**을 적고 이를 **설명하시오**.) Please be specific with your answers as needed to show your understanding (자신의 이해를 보일 수 있을 정도로 자세히 설명하시오). (30 points)

Database development process의 design stage는 conceptual database design, logical database design, physical database design으로 나뉜다. Conceptual database design 단계에서는 data requirement를 input으로 받아 비즈니스 룰에 맞게 ERD(Entity Relationship diagram)을 그리며 전체적인 database의 내용을 채워 넣는다. Logical database design 단계에서는 앞서 그린 ERD를 conceptual schema로 input 받고, relational model로 변경하여 logical schema를 output한다. 이 과정에서 정규화 과정이 일어나고, 비즈니스 룰을 강화하기 위한 여러 제약을 추가한다. Physical database design 단계에서는 앞서 그린 relational model을 logical schema로 input받고, 성능 및 index, data placement 등을 결정해 불필요한 자원을 최소화하여 Internal schema 즉, physical schema를 output 한다.

1. (100 pts) 다음은 CPV라고 하는 DVD 대여점을 위해 DB를 설계하는 케이스이다. 수업시간에 배운 ER (그리고 Enhanced ER) notations 을 사용하여, 다음에 주어진 user requirements 에 기반한 ER diagram 을 그리시오. 모든 user requirements를 빠짐없이 참고하여 그리시오. 수업시간에 사용한 ER diagram notation을 사용하여, 자신이 사용할 수 있고 쓰기 편한 프로그램(예: diagramming tools (e.g., Visio, EA, 등), Powerpoint, MS Word 혹은 그 외의 툴들)을 사용하여 그리고, 그려진 ERD를 현재의 document에 답안으로 붙여 넣어 제출하시오. 수기작성하여 사진을 찍어 올리는 등의 행위는 20% 이상의 감점이 적용됨.

Continental Palms Video (CPV) is considering creating a database to keep track of various aspects of the store. Impressed by your database knowledge and skills, the management of CPV has decided to hire you to design a conceptual data model.

CPV has various genres (Classic, Horror, Action, Comedy, Foreign, Educational, etc.) of Films for rent. Each film is uniquely identified by a Film ID. Each film is also described by a title, a rate, and director(s). A film may have several directors. For some special genres of films, more information is recorded. Each foreign film has a spoken language and a subtitle language. Each educational film has a recommended age group. A film may be both foreign film and educational film.

To allow customers to search for films based on their preferred performers, CPV also records the information about performers: ID (identifier) and name. A film may have zero to many performers. Each performer performs a role in a film and CPV also records the role of each performer in a film. A performer may perform in many films. To be recorded in the database, a performer must perform in at least one film.

CPV carries one or more copies of each film. Each film must have at least one copy. Each copy belongs to one film. Each copy has a copy number that is unique among the copies of the same film. But multiple copies of different film may have the same copy number. Other information about a copy includes its condition (e.g., broken). 도표, 스케치, 그림, 텍스트이(가) 표시된 사진

자동 생성된 설명