

데이터베이스 과제 1

고려대학교 통계학과
이재승

◆ Exercises of Chapter 1

#2. What four main types of actions involve databases? Briefly discuss each.

- 데이터베이스를 이루는 4가지의 메인 타입은 크게 Defining, Constructing, Manipulating, Sharing이라고 할 수 있다.

- Defining : 데이터베이스에 저장될 데이터의 Data Types, Structures, Constraints 등을 정의한다.

- Constructing : DBMS에 의해 컨트롤되는 것인 Storage Medium에 데이터를 저장(Store)하는 과정이다.

- Manipulating : 데이터베이스에 대하여 Query하고 Update하는 것을 말한다. 또한 Data로부터 리포트를 만들어낸다.

- Sharing : 다수의 유저와 프로그램들에 대해 동시다발적으로 데이터베이스에 접근할 수 있도록 허용해주는 것을 말한다.

#4. What are the responsibilities of the DBA and the database designers?

- DBA는 Database Administrators의 약자로 DB 시스템을 관장하는 사람이라고 보면 된다. 데이터베이스의 쓰임새에 대해서 모니터링하며, 데이터베이스 운영에 필요한 HW와 SW 리소스를 관리하기도 한다. 또한 Security Breach나 Poor Performance에 대해 책임을 져야 하는 사람이다. DBA가 져야 하는 가장 큰 책임은 Security Breach(보안 문제)와 Poor Performance(성능 저하)가 있겠다.

- Database Designers는 말 그대로 데이터베이스를 디자인하는 사람으로서 쉽게 말하면 Database의 핵심 구성요소인 Schema를 디자인하는 업무를 한다. 데이터가 어디에 저장할지 정의하며, 적합한 데이터 구조를 택하기도 한다. 많은 유저 그룹들로부터 요구사항들을 모으고, Data와 Processing 요구사항에 대해 View들을 정의한다. 데이터베이스를 사용하려는 유저들의 요구사항에 따라 Schema를 디자인하고 데이터 저장 위치와 데이터 구조 등을 정의하는 업무를 하는 사람이다.

#7. Discuss the differences between database systems and information retrieval systems.

- Database Systems은 Structured Data를 다룬다. Schema에 기반하여 Query Model이 조직되어 있다. Rich Metadata Operations가 있으며 Query가 Data를 뽑아내는 구조다. Exact Matching에 기반하여 결과를 도출해내기 때문에 모두 Correct한 결과를 뽑아낸다.

- Information Retrieval System은 Database System보다 단순한 Data Model을 사용한다. 문

서들의 모음으로 이루어진 정보로, 문서는 구조화되지 않았으며(Unstructured data) 딱 고정된 Schema가 없으며 다양한 데이터 모델들이 있다. Query Model은 Free-form이다. 또한 문서에 대하여 검색을 할 때 returns list나 pointers를 request하기도 한다. Approximate Matching에 기반하여 Result를 뽑아낸다. 그렇기 때문에 imprecise하다.

#10. Specify all the relationships among the records of the database shown in Figure 1.2.

- STUDENT의 Major와 COURSE의 Department가 서로 이어진다.
- COURSE의 Course_number와 SECTION의 Course_number는 서로 같은 것을 의미한다.
- SECTION의 Section_identifier와 GRADE_REPORT의 Section_identifier는 서로 같은 것을 의미한다.
- GRADE_REPORT의 경우, STUDENT의 Student_number와 GRADE_REPORT의 Student_number가 서로 같은 것을 의미하며, SECTION의 Section_identifier와 GRADE_REPORT의 Section_identifier가 서로 같은 것을 의미한다.
- PREREQUISITE의 경우 말 그대로 선행 과목을 의미하며, Course_number에 해당하는 과목을 수강하기 이전에 Prerequisite_number의 과목을 먼저 수강해야 한다는 것을 의미한다.
- 각각의 SECTION 기록은 COURSE와 관련이 있다.
- 각각의 GRADE REPORT 기록은 하나의 STUDENT 기록과 하나의 SECTION 기록과 관련이 있다.

#12. Cite some examples of integrity constraints that you think can apply to the database shown in Figure 1.2.

- STUDENT의 Student_number는 unique해야 한다. 학생 1인에게 주어지는 Student_number는 그 자체로 고유한 값만 있어야 한다.
- COURSE의 Course_number는 unique해야 한다. 과목 하나에게 주어지는 Course_number는 그 자체로 고유한 값만 있어야 한다.
- SECTION에 있는 Course_number는 COURSE에 있는 Course_number와 동일해야 한다. 즉, SECTION에 있는 Course_number가 COURSE에 있는 Course_number에 똑같이 존재해야 한다는 것을 의미한다.
- GRADE_REPORT에 있는 Student_number가 STUDENT에 있는 Student_number와 동일해야 한다. 즉, GRADE_REPORT에 있는 Student_number가 STUDENT에 있는 Student_number에 똑같이 존재해야 한다는 것을 의미한다.
- 각각의 학생에게는 반드시 Student_number가 부여되어야 한다. STUDENT에 있는 Student_number에서 해당 Student_number를 표현한다.
- 각각의 과목에게는 반드시 Course_number가 부여되어야 한다. COURSE에 있는 Course_number에서 해당 Course_number를 표현한다.

◆ Exercises of Chapter 2

#3. What is the difference between a database schema and a database state?

- Database Schema는 데이터베이스에 있는 데이터를 묘사한다. 즉, 데이터에 대한 데이터를 의미한다. 이를 Meta-data라고 한다. Database의 Schema를 디자인하며, 그렇게 많이 바뀌지는 않는 요소이기도 하다. 전체 데이터베이스의 Skeleton Structure이다.
- Database State는 특정 Moment에서의 DB에 있는 데이터를 의미한다. Insert, Update, Delete 등의 조치로 인해 State가 변경되기도 한다.

#4. Describe the three-schema architecture. Why do we need mappings between schema levels? How do different schema definition languages support this architecture?

- Three-Schema Architecture는 Program과 Data 사이의 독립성, Multiple User Views, DB Description을 저장하기 위한 Catalog의 사용이라는 특성들을 실현할 수 있기 위해서 활용한다.
 - 즉, Physical Database로부터 User Applications를 분리하기 위해서 활용한다.
- 1) Internal Level에서는 Internal Schema를 활용한다.
 - Physical Data Model을 활용한다.
 - Data Storage와 Access Path의 디테일함을 묘사한다.
 - 2) Concept Level에서는 Conceptual Schema를 활용한다.
 - 전체 DB에 대한 구조를 묘사한다.
 - Physical Storage Structures의 디테일함을 숨긴다.
 - Entities, Data types, Relationships user operations, constraints에 포커스를 둔다.
 - 3) External Level에서의 External Schema를 활용한다.
 - 각각의 다른 User Groups들에게 적합한 View를 묘사한다.

#5. What is the difference between logical data independence and physical data independence? Which one is harder to achieve? Why?

◎ Logical Data Independence

- External Schema를 변경할 필요 없이 Conceptual Schema를 변경하는 것을 말한다.
- Record Type이나 Data Item을 추가하거나 제거한다.

◎ Physical Data Independence

- Conceptual Schema나 External Schema나 Application Program을 변경할 필요 없이 Internal Schema를 변경하는 것을 말한다.
- Extra access structure와 같은 physical한 file을 재조직한다.

→ Logical Data Independence가 더 어렵다. 왜냐하면 Application Program에 영향을 주지

않은 채 Structural가 Constraint 변화를 주기 때문이다.

#6. What is the difference between procedural and nonprocedural DMLs?

- 2개를 서로 비교하자면 아래와 같다.

◎ Nonprocedural DML

- High-level DML로서 set-oriented이다.
- Casual end users가 주로 이용한다.

.

◎ Procedural DML

- Low-level DML로서 record-oriented이다.
- 반드시 General-purpose programming language에 embed되어야 한다.
- Programmer들이 주로 이용한다.

#13. Choose a database application with which you are familiar. Design a schema and show a sample database for that application, using the notation of Figures 1.2 and 2.1. What types of additional information and constraints would you like to represent in the schema? Think of several users of your database, and design a view for each.

- 축구 리그를 운영하는 리그의 입장에서 DB를 구축해보았다.

◆ 구단

| | | | | | |
|-----|-----|------|------|----|----------------|
| 구단명 | 소재지 | 경기장명 | 구단주명 | 감독 | 직전 시즌 리그 순위 |
|-----|-----|------|------|----|----------------|

◆ 감독

| | | | | | |
|-----|-------|----|----|-------|----------------|
| 감독명 | 담당 구단 | 국적 | 나이 | 계약 기간 | 직전 시즌 리그 순위 |
|-----|-------|----|----|-------|----------------|

◆ 선수

| | | | | | |
|-----|-------|-----|----|----|----|
| 선수명 | 소속 구단 | 포지션 | 나이 | 연봉 | 국적 |
|-----|-------|-----|----|----|----|

◆ 리그 기록

| | | | | | |
|----|-----|-------|--------|-------|-------|
| 일시 | 홈 팀 | 어웨이 팀 | 경기장 장소 | 경기 결과 | 득점 기록 |
|----|-----|-------|--------|-------|-------|

◆ 축구경기 중계 방송 송출

| | | | |
|-----|--------|-------|-------|
| 방송사 | 방송사 국적 | 계약 기간 | 계약 금액 |
|-----|--------|-------|-------|

- 글씨를 진하게 표현한 구단 Table에서의 구단명, 감독 Table에서의 감독명, 선수 Table에서의 선수명, 축구경기 중계 방송 송출 Table에서의 방송사명이 각 Table에서의 ID로 기능한다. Database의 표현 간략화를 위해 이 Attribute들에 속한 Value들에 대해 Index를 부여해주는 것도 하나의 방법이겠다.
 - 구단 Table에서의 직전 시즌 리그 순위, 감독 Table에서의 직전 시즌 리그 순위는 정수값으로 나타내어야 한다. 총 20팀이 축구 리그에 참여한다고 가정하면 직전 시즌 리그 순위의 값은 1~20 안에 반드시 포함되어야 한다.
 - 감독 Table에서의 나이와 선수 Table에서의 나이 역시 정수값으로 나타내어야 한다. 이 나이는 만 나이를 기준으로 한다.
 - 구단 Table의 구단명, 감독 Table의 담당 구단, 선수 Table의 소속 구단은 서로 관련이 있다. 주어진 구단 리스트 범위 안에서 모든 Value들이 결정된다.
 - 구단 Table의 경기장명과 리그 기록 Table의 경기장 장소는 서로 관련이 있다.
 - 축구 리그에 참여하는 총 구단의 수가 20개라고 가정하면, 한 해 축구 리그가 종료된다면 구단 Table에 속하게 되는 총 Tuple의 수는 20개, 감독 Table에 속하게 되는 총 Tuple의 수도 20개, 리그 기록에 속하게 되는 총 Tuple의 수는 380개가 된다. Home / Away 방식으로 각 팀당 38경기를 치르게 되기 때문이다.
-
- 축구 리그를 운영하는 리그의 입장에서 이 모든 자료를 Database에 정리함으로써 리그의 원활한 운영을 촉진할 수 있을 것이다.
 - 감독의 입장에서 선수 Table을 본다면, 같은 리그 내에서 우수한 성과를 보이는 선수의 정보를 파악함으로써 본인 구단으로 해당 선수를 영입할 수 있을 것이다.
 - 반대로 선수의 입장에서 감독 Table을 본다면, 훌륭한 감독 아래서 본인의 선수 커리어를 쌓고 싶다는 마음에 감독님께 오히려 역컨택을 할 수도 있을 것이다.
 - 축구 관중의 입장에서는 모든 Table에 흥미를 갖게 될 것이다. 구단 Table을 통해 올 한 해 축구 리그에 참여하는 구단들의 정보를 파악할 수 있다. 감독 Table을 통해 해당 구단의 사령탑이 어떤 사람인지 파악할 수 있으며, 선수 Table을 통해 어떤 선수들이 축구 리그에서 활약을 펼칠 것인지에 대해 생각해볼 수 있다. 또한 리그 기록 Table을 통해 축구 리그가 어떻게 운영되었는지 파악할 수 있으며, 축구 경기 직관이 아닌 TV 중계로 축구 경기를 시청할 경우 축구경기 중계 방송 송출 Table이 도움이 될 것이다.