

데이터베이스설계_03 기말 프로젝트 DGU Event Apply System(DEAS)

2020112144 윤진수

1. 프로그램 기획

DGU Event Apply System(DEAS)는 교내 행사들을 한눈에 확인하고, 지원할 수 있는 시스템이다. ‘컴퓨터·AI학과 대학원 FAIR’, ‘학기제현장실습 사전설명회’, ‘동아리 신입부원 모집’ 등 기존에 구글폼으로 모집되던 모든 행사들이 대상이다. 이 프로그램을 통해 학생은 여러 행사들을 한눈에 확인하고 편리하게 지원할 수 있고, 교수 및 교직원등 행사 관계자는 여러 행사를 등록하고 설문조사를 설정하여 행사 전 필요한 데이터를 쉽게 수집할 수 있다.

2. 요구사항분석

데이터베이스 설계의 첫 단계로 요구사항 분석을 진행하였다. 사용자의 요구 사항을 수집하고 분석하여 개발할 데이터베이스의 용도를 파악한다.

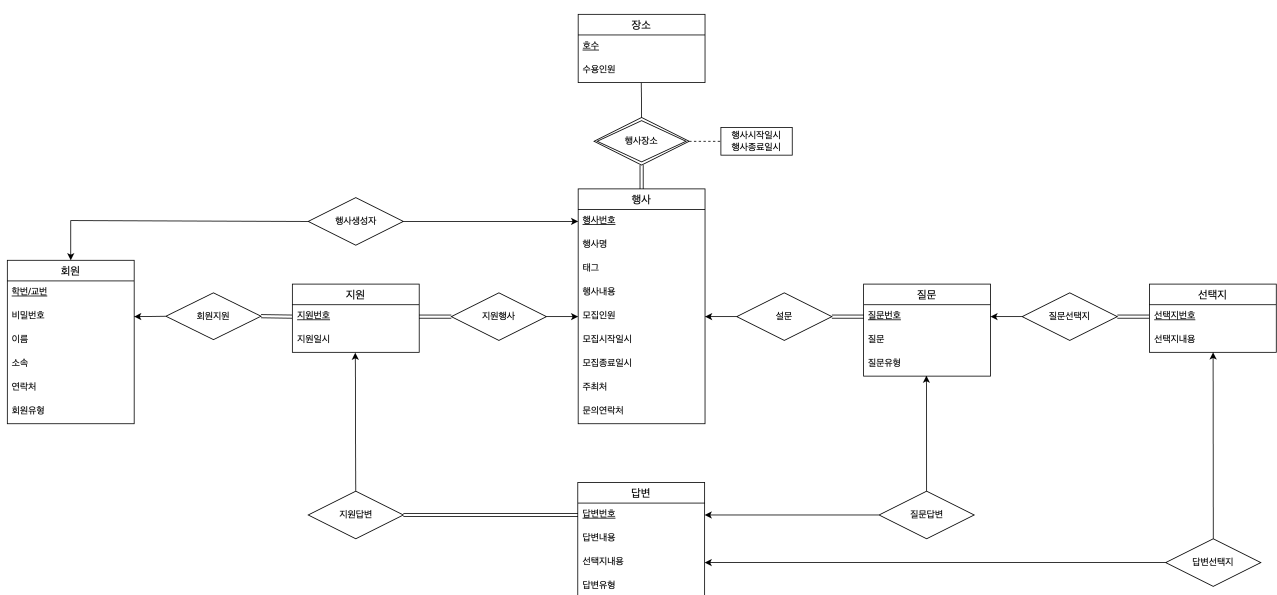
- 요구사항명세서

1. 시스템에 회원으로 가입하려면 학번/교번, 비밀번호, 이름, 소속, 연락처, 회원유형(학생/교직원/관리자)을 입력해야 한다.
2. 회원은 학번/교번으로 식별한다.
3. 행사를 생성하려면 행사명, 태그, 행사내용, 설문조사, 모집인원, 행사일자, 시작시간, 종료시간, 행사장소, 주최처, 문의연락처 정보를 입력해야 한다.
4. 행사장소는 수용인원 정보를 유지해야 하고, 행사의 모집인원은 행사장소의 수용인원을 초과할 수 없다.
5. 행사는 하나의 행사장소에서 진행될 수 있고, 행사장소는 같은 시간에 하나의 행사만 진행된다.
6. 행사는 행사번호로 식별하고, 행사번호, 참가자 시작/종료일시 정보를 유지해야 한다.
7. 회원은 여러 행사를 지원할 수 있고, 하나의 행사를 여러 회원이 지원할 수 있다.
8. 회원이 행사에 지원하면 지원에 대한 설문조사 정보를 유지해야 한다.
9. 회원은 지원한 행사 목록을 확인할 수 있다.
10. 회원은 생성한 행사의 지원자 목록과 설문조사 결과를 확인할 수 있다.

3. 개념적 설계

앞서 정의한 요구사항을 바탕으로 DBMS에 독립적인 개념적 스키마를 설계한다.

- E-R 다이어그램



회원, 행사, 장소, 질문, 선택지, 답변, 지원의 Entity가 존재한다. 각 Entity들은 회원지원, 지원행사, 지원답변, 행사장소, 설문, 질문답변, 질문선택지, 답변선택지 의 Relation으로 표현된다.

각 Relation에서 화살표를 통해 Cardinality를 표현하였다. 화살표는 One을, 일반적인 선은 Many를 의미한다. 예를 들어, 행사는 여러 장소에서 진행될 수 있고, 장소도 시간에 따라 여러 행사가 진행될 수 있다. 그리고, 행사는 여러 질문을 설문으로 가질 수 있고, 질문은 특정 행사 하나에서만 존재한다.

Participation 정보는 Total Participation은 이중선으로, Partial Participation은 일반적인 선으로 표현하였다. 예를 들면, 행사가 진행되려면 반드시 장소가 있어야 한다. 이 때, 장소 Entity에는 ‘온라인’ 튜플이 존재한다고 가정한다. 그리고 어떤 장소에서는 행사가 진행되지 않을 수도 있다.

행사장소 Relation(관계)은 논리적 설계 단계에서 릴레이션(테이블)으로 변환된다. 행사번호와 장소호수를 기본키로 가지는 Weak Entity Set으로, 이중 마름모로 표현되었다.

4. 논리적 설계

E-R 다이어그램을 논리적 데이터 모델을 이용해 DBMS에 적합한 논리적 스키마로 변환한다.

- 릴레이션 스키마

member(id, password, name, affiliation, contact, type)
 apply(id, member_id, event_id, apply_time)
 event(id, creator_id, name, tag, description, max_participant, status, apply_start_time, apply_end_time, organizer, contact)
 question(id, event_id, content, type)
 option(id, question_id, content)
 answer(id, apply_id, question_id, type, option_id, answer)
 location(id, capacity)
 event_location(event_id, location_id, start_time, end_time)



먼저, E-R 다이어그램의 모든 개체를 릴레이션으로 변환한다. 각 릴레이션의 기본키는 밑줄로 나타낸다. 다대다 관계인 ‘행사장소’ 관계는 외래키인 행사번호와 장소의 호수를 기본키로 하는 행사장소 릴레이션으로 표현된다. Relation의 속성인 행사 시작, 종료 일시는 이 릴레이션의 속성으로 표현된다.

지원답변, 질문답변 등의 일대다, 일대일 관계는 외래기로 표현하였다.

5. 물리적 설계

위 설계를 바탕으로 DDL을 작성하고 제약조건을 설정한다. 분량상 보고서에는 일부 테이블만 작성하였다.

```
CREATE TABLE event
(
  id INT NOT NULL AUTO_INCREMENT COMMENT '행사번호',
  creator_id VARCHAR(20) NOT NULL COMMENT '학번/교번',
  name VARCHAR(40) NOT NULL COMMENT '행사명',
  tag ENUM('SESSION', 'SEMINAR', 'CLUB', 'MISC') NULL COMMENT '태그',
  description TEXT NOT NULL COMMENT '행사내용',
  max_participant INT NOT NULL DEFAULT 0 COMMENT '모집인원',
  status ENUM('PENDING', 'IN_PROGRESS', 'COMPLETE') NOT NULL DEFAULT 'PENDING' COMMENT '모집상태',
  apply_start_time TIMESTAMP NOT NULL COMMENT '모집시작일시',
  apply_end_time TIMESTAMP NOT NULL COMMENT '모집종료일시',
  organizer VARCHAR(40) NOT NULL COMMENT '주최처',
  contact VARCHAR(20) NOT NULL COMMENT '문의연락처',
  PRIMARY KEY(id)
) COMMENT '행사';

ALTER TABLE event ADD CONSTRAINT chk_apply_time CHECK(apply_start_time < apply_end_time);
```

```
CREATE TABLE answer
(
  id INT NOT NULL AUTO_INCREMENT COMMENT '답변번호',
  apply_id INT NOT NULL COMMENT '지원번호',
  question_id INT NOT NULL COMMENT '질문번호',
  type ENUM('CHOICE', 'ESSAY') NOT NULL COMMENT '답변유형',
  choice_id INT NULL COMMENT '선택지번호',
  answer TEXT NULL COMMENT '답변내용',
  PRIMARY KEY(id)
) COMMENT '답변';

ALTER TABLE answer ADD CONSTRAINT UQ_apply_id_question_id UNIQUE(apply_id, question_id);

ALTER TABLE answer ADD CONSTRAINT FK_question_T0_answer
FOREIGN KEY(question_id) REFERENCES question (id) ON DELETE CASCADE;

ALTER TABLE answer ADD CONSTRAINT FK_choice_T0_answer
FOREIGN KEY(choice_id) REFERENCES choice (id) ON DELETE CASCADE;

ALTER TABLE answer ADD CONSTRAINT FK_apply_T0_answer
FOREIGN KEY(apply_id) REFERENCES apply(id) ON DELETE CASCADE;
```

event DDL은 행사 정보를 저장하는 테이블이다. 행사번호는 AUTO_INCREMENT 속성을 통해 자동으로 하나씩 증가하도록 설정하였다. 행사명, 주최처, 문의연락처는 입력될 내용이 길지 않으므로 VARCHAR 타입에 적은 최대값을 주었고, 행사내용은 긴 글이 들어갈 수 있도록 TEXT 타입으로 설정하였다. 태그와 모집상태 등 고정된 값들은 가지는 속성은 ENUM 타입으로 지정하였다. 애플리케이션 로직에서 처리되지 못한 예외에 대비해 모집시작일시가 모집종료일시보다 먼저가 되도록 CHECK 제약 조건을 추가하였다.

answer DDL은 행사 지원시 답변을 저장하는 테이블이다. 행사 테이블과 마찬가지로, 답변 내용은 TEXT

타입을 사용하고, 답변유형은 ENUM으로 설정하였다. 특정 지원의 하나의 질문에 여러 답변이 생기는 것을 방지하기 위해 UNIQUE 제약 조건을 추가하였다. 답변 정보의 지원, 질문, 선택지 정보는 외래키를 통해 추가 되었다. 외래키에는 ON DELETE CASCADE 옵션을 통해 답변이 삭제될 때, 답변과 관련된

6. 구현

MySQL을 설치하고 DDL을 통해 데이터베이스를 설정한다. 데이터베이스와의 연계를 위한 프로그램은 웹 기반 시스템으로, Spring Boot와 Spring MVC를 사용해 Model, Controller를 구성하고, Thymeleaf 템플릿 엔진을 활용해 View를 구성한다.

```
@Entity
@Getter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class Location {
    @Id
    private String id;

    private Integer capacity;

    @OneToMany(mappedBy = "location")
    private List<EventLocation> eventLocations
        = new ArrayList<>();
}

@Repository
public interface ApplyRepository extends JpaRepository<Apply, Integer> {

    List<Apply> findByMember(Member member);

    Integer countByEvent(Event event);

    List<Apply> findByEvent(Event event);
}
```

Spring Data JPA를 사용하여, class와 테이블을 매핑하고 메소드를 통해 데이터베이스를 조작한다.

location 테이블의 기본키인 호수는 String 타입으로 정의되고 @Id 어노테이션으로 기본 키임을 명시한다. eventLocations 리스트는 event_location 테이블과의 일대다 관계를 나타낸다.

기본적인 CRUD 동작은 JPA가 제공한

```
@Repository
public interface LocationRepository extends JpaRepository<Location, String> {

    @Query("""
        SELECT l FROM Location l
        LEFT JOIN EventLocation el ON el.location.id = l.id
        AND el.startTime < :end
        AND el.startTime > :start
        WHERE el.event.id IS NULL
        """)
    List<Location> findAvailableLocations(@Param("start") LocalDateTime start, @Param("end") LocalDateTime end);
}
```

다. findByMember() 메소드는 “SELECT * FROM apply WHERE member_id = ?”의 쿼리를 생성한다.

하지만 복잡한 쿼리는 직접 효율을 위해 직접 작성하였다. @Query 어노테이션을 통해 JPQL(Java Persistence Query Language)로 복잡한 쿼리를 작성할 수 있다. 해당 쿼리는 특정 시간대(start ~ end) 사이에 event_location 테이블에 저장되지 않은 모든 location 정보를 반환한다. 이는 행사를 생성 중 시간을 고르고 장소를 고를 때 작동하여 이미 행사가 예정되지 않은 장소 리스트를 보여주는데 사용된다.

```
<tr
th:each="eventItem : ${createdEvents}"
th:onclick="location.href='/event/' + [[${eventItem.event.id}]]"
>
```

Thymeleaf는 Spring에서 model 객체를 통해 전달한 데이터를 동적으로 렌더링할 수 있도록 한다.

Spring이 전달한 createdEvents 리스트를 순회하며 각각에 대한 <tr>을 만들고 클릭 시 해당 이벤트 페이지로 이동하도록 하는 예시이다.

행사 시작일 2024. 12. 17. 오후 06:00

행사 종료일 2024. 12. 19. 오후 10:00

신청 시작일 2024. 12. 03. 오전 12:00

신청 종료일 2024. 12. 18. 오전 12:00

장소 ☒ 장소를 선택하세요

주최처 ☐ 101-272 (30)

문의 연락처 ☐ 101-275 (30)

☐ 101-277 (30)

☐ 101-279 (30)

☐ 101-281 (30)

☐ 101-286 (70)

WHERE	ORDER BY
event_id	location_id
start_time	end_time
1	1 101-189
	2024-12-19 16:00:00
	2024-12-19 21:00:00

101-189 강의실에 12/19 16:00~21:00 에 행사가 예약되어있다.

이 경우 행사 시간이 기존 등록된 행사장소와 겹치는 경우 신청 가능한 장소에 표시되지 않는다.

행사 생성 시 설문조사를 함께 등록할 수 있고, 지원 시 이에 답변할 수 있다.

설문조사

질문 1 질문하실 교수님

☐ 단답형 ☒ 객관식

선택지

질문 2 교수님께 궁금한 점

☒ 단답형 ☐ 객관식

행사 지원

질문하실 교수님

- ☐ 윤병운 교수님
- ☒ 염세경 교수님
- ☐ 손영두 교수님

교수님께 궁금한 점

교수님께서 어떤 연구를 진행하시나요?

지원 정보 페이지에서 지원한 행사와 생성한 정보의 현황을 확인할 수 있고,

내가 생성한 행사 정보

행사명	모집상태	지원자 수	지원정보 보기
동국대학교 산업시스템공학과 대학원 설명회	모집 중	1	지원정보 보기

내가 지원한 행사 정보

행사명	모집상태	마감일
동국대학교 산업시스템공학과 대학원 설명회	모집 중	2024-12-19T00:00

내가 생성한 행사 정보

행사명	모집상태	지원자 수	지원정보 보기
-----	------	-------	---------

내가 지원한 행사 정보

행사명	모집상태	마감일
동국대학교 산업시스템공학과 대학원 설명회	모집 중	2024-12-19T00:00

지원정보 보기 버튼을 통해 지원한 회원의 정보와 답변을 확인할 수 있다.

지원 정보

행사명	학번/교번	이름	소속	연락처	질문	답변
동국대학교 산업시스템공학과 대학원 설명회	2020112144	윤진수	컴퓨터공학전공	010-8636-2572	질문하실 교수님	염세경 교수님
					교수님께 궁금한 점	교수님께서 어떤 연구를 진행하시나요?