

캡스톤 디자인

데이터베이스

설계 계획서 초안

Kim Gyeongseon

2022-7-25

목차

1. 업무 기능도
2. 요구사항 명세서
3. 업무 흐름도
4. 논리적 데이터 모델(ERD)
5. 물리적 데이터 모델(ERD-IE)
6. 사용 용어
7. 도메인 기술서
8. Table 기술서
9. MySQL 코드

1. 업무 기능도



2. 요구사항 명세서

단위업무	회원
<ol style="list-style-type: none"> 1) 회원은 학번(PK), 비밀번호, 이름, 성별, 학과의 정보를 가진다. 2) 회원은 누구든 파티 모집을 할 수 있다. 3) 학번은 8자리의 숫자로 이루어지며 정수형으로 저장된다. 4) 비밀번호는 영어와 숫자의 조합으로 이루어지며 문자열로 저장된다. 5) 이름은 문자열로 저장된다. 6) 성별은 남/여로 지정할 수 있으며 DB 상에 문자열로 저장된다. 7) 학과는 전기전자공학과/스마트전기전자공학과/기계공학과/스마트기계공학과/스마트소프트웨어학과로 지정할 수 있으며 DB 상에 문자열로 저장된다. 	

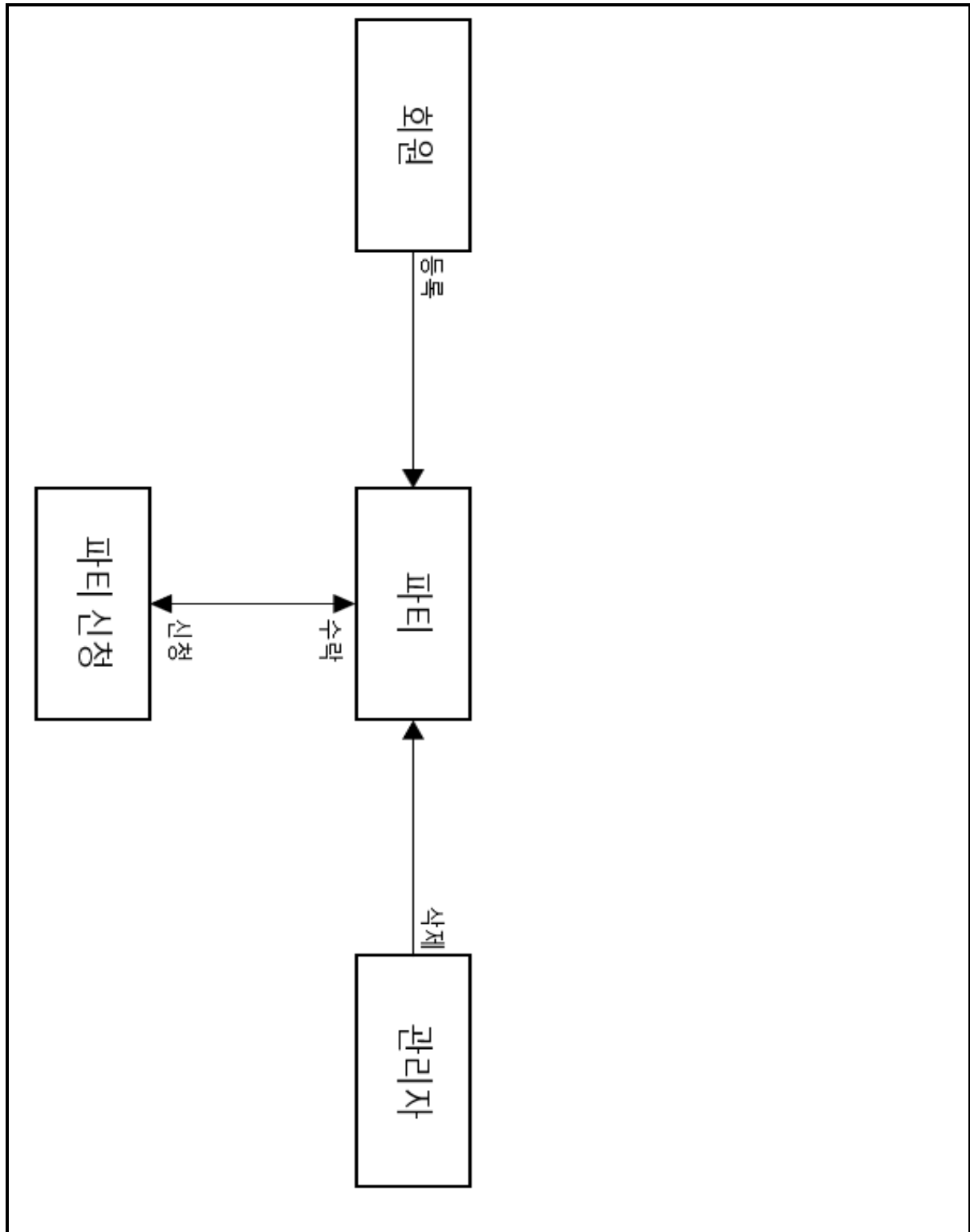
단위업무	파티
<ol style="list-style-type: none"> 1) 파티는 글 번호(PK), 파티 이름, 카테고리, 매장 이름, 배달 플랫폼, 먹을 날짜, 먹을 시간, 먹을 방식, 배달 위치, 결제 방식, 모집 인원, 현재 모집 인원, 채팅 링크, 작성자(FK), 완료 여부의 정보를 가진다. 2) 파티 테이블은 앱 상에서 게시물의 형태로 보여진다. 3) 파티 테이블의 데이터는 작성자(FK)가 게시물을 작성하면 저장이 되는데 작성자가 적어야 하는 데이터들은 파티 이름, 카테고리, 매장 이름, 배달 플랫폼, 먹을 날짜, 먹을 시간, 먹을 방식, 배달 위치, 결제 방식, 모집 인원, 채팅 링크이다. 4) 게시물이 올라가면 회원 테이블에 저장 되어있는 회원들이 게시물을 보고 신청을 할 수 있다. 5) 신청을 게시물 작성자가 수락하면 현재 모집 인원이 1씩 증가하고 현재 모집 인원은 모집 인원까지만 증가한다. 6) 신청 회원 중 빠지는 회원이 생긴다면 빠진 회원만큼 현재 모집 인원이 감소된다. 7) 모집 인원만큼 현재 모집 인원이 찼다면 완료 버튼이 활성화가 되고 작성자가 완료 버튼을 누르면 신청한 회원들에게 채팅 링크가 공유된다. 8) 글 번호는 첫 게시물을 기준으로 정수형 1부터 시작해서 1씩 증가하며 자동으로 저장된다. 9) 파티 이름은 영어, 한글, 숫자, 특수기호의 조합으로 이루어지며 150글자까지 지정할 수 있고 문자열로 저장된다. 10) 카테고리는 고기/도시락/양식/일식/중식/치킨/카페/피자/패스트푸드/한식/분식/기타로 지정할 수 있으며 DB 상에 문자열로 저장된다. 11) 매장 이름은 영어, 한글, 숫자, 특수기호의 조합으로 이루어지며 10글자까지 지정할 수 있고 문자열로 저장된다. 12) 배달 플랫폼은 배달의 민족/요기요/쿠팡 이츠로 지정할 수 있으며 DB 상에 문자열로 저장된다. 	

- 13) 먹을 날짜는 오늘/내일로 지정할 수 있으며 DB 상에는 해당 날짜가 DATE(YYYY-MM-DD)로 저장된다.
- 14) 먹을 시간은 배달을 시킬 시간으로 TIME(HH:MM:SS)로 저장된다.
- 15) 먹을 방식은 같이 먹기/따로 먹기로 지정할 수 있으며 DB 상에 문자열로 저장된다.
- 16) 배달 위치는 배달을 받을 위치로 혹은 배달을 나눌 위치로 산학 협동관/미래관/본관 건물/식당으로 지정할 수 있고 DB 상에 문자열로 저장된다.
- 17) 결제 방식은 모여서 후결제/파티장 선결제/모아서 선결제로 지정할 수 있으며 DB 상에 문자열로 저장된다.
- 18) 모집 인원은 게시물을 작성한 사람이 1 ~ 5까지 지정할 수 있으며 정수형으로 저장된다.
- 19) 현재 모집 인원은 게시물에 신청을 한 인원을 작성자가 수락하면 1씩 증가하며 모집 인원만큼까지 1부터 올라가고 정수형으로 저장된다.
- 20) 채팅 링크는 게시물을 작성한 사람이 대화를 나눌 채팅방을 만든 후 채팅방에 들어갈 수 있는 링크를 올리면 해당 링크가 문자열 형태로 저장된다.
- 21) 작성자는 게시물을 작성한 사람으로 회원 테이블에 저장된 회원 중 한명이 되며 해당 회원의 학번이 저장된다.
- 22) 완료 여부는 미완료/완료로 DB 상에 문자열로 저장된다.
- 23) 게시물의 작성자는 게시물을 수정할 수 있으며 게시물을 수정하면 해당 데이터가 수정된다.
- 24) 게시물의 작성자는 게시물을 삭제할 수 있으며 게시물을 삭제하면 해당 데이터가 삭제된다.
- 25) 게시물의 작성자가 회원 탈퇴를 하면 해당 작성자가 작성했던 게시물들은 작성자(FK) 값이 NULL로 바뀌고 유지된다.

단위업무	파티 신청
<ol style="list-style-type: none"> 1) 파티 신청은 글 번호(PK, FK), 신청자(PK, FK), 신청 시간, 수락 여부의 정보를 가진다. 2) 회원이 특정 파티에 신청을 할 때마다 파티 신청에 그 신청 정보가 저장된다. 3) 게시물의 작성자는 신청 정보를 보고 신청을 수락/거절할 수 있다. 4) 게시물의 작성자는 모집 인원의 수만큼만 신청을 수락할 수 있다. 5) 글 번호는 파티의 글 번호를 가진다. 6) 신청자는 회원의 학번을 가진다. 7) 신청 시간은 신청자가 파티에 신청을 한 시간으로 DATETIME 정보를 저장한다. 8) 수락 여부는 게시물 작성자가 신청자를 수락했는지 안 했는지의 여부로 미확인/거절/수락의 값을 가지고 DB 상에 문자열로 저장된다. 	

단위업무	관리자
	<ol style="list-style-type: none"> 1) 관리자는 아이디(PK), 비밀번호, 이름, 전화번호의 정보를 가진다. 2) 관리자는 게시물(파티)을 삭제할 수 있는 권한을 가진다. 3) 아이디는 정수형으로 저장된다. 4) 비밀번호는 영어와 숫자의 조합으로 이루어지며 문자열로 저장된다.. 5) 이름은 한글로 이루어지며 문자열로 저장된다. 6) 전화번호는 숫자로 이루어지며 문자열로 저장된다.

3. 업무 흐름도



```

erDiagram
    회원 ||--o{ "파티 신청" : ""
    "파티 신청" ||--|| "수탁" : ""
    "수탁" ||--o{ 파티 : ""
    파티 ||--o{ 관리자 : ""

    회원 {
        string 학번
        string 비밀번호
        string 학과
        string 이름
        string 성별
    }

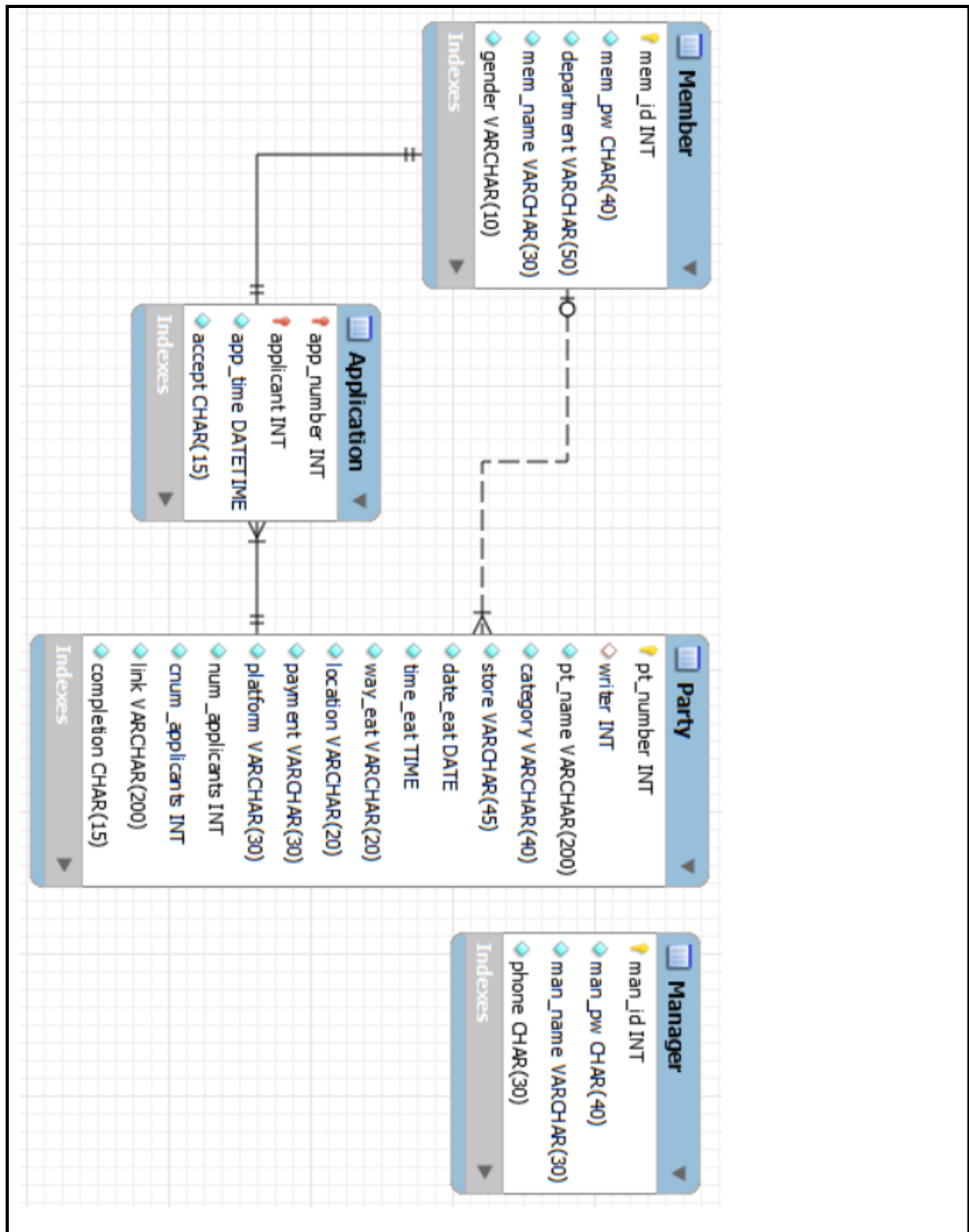
    "파티 신청" {
        string 신청자
        string 신청_시간
        bool 수탁_여부
        string 글_번호
    }

    "수탁" {
        string 예약_번호
        string 작성자
        string 파티_이름
        string 카테고리
        string 매장_이름
        string 먹을_날짜
        string 먹을_시간
        string 먹을_방식
        string 배달_위치
        string 배달_플랫폼
        int 모집_인원
        int 잔제_모집_인원
        string 채팅_링크
        bool 완료_여부
    }

    파티 {
        string 관리자
        string 아이디
        string 비밀번호
        string 이름
        string 전화번호
        string 글_번호
        string 작성자
        string 파티_이름
        string 카테고리
        string 매장_이름
        string 먹을_날짜
        string 먹을_시간
        string 먹을_방식
        string 배달_위치
        string 배달_플랫폼
        int 모집_인원
        int 잔제_모집_인원
        string 채팅_링크
        bool 완료_여부
    }

    관리자 {
        string 아이디
        string 비밀번호
        string 이름
        string 전화번호
    }
  
```


5. 물리적 데이터 모델(ERD-IE)



6. 사용 용어

테이블	논리명	물리명	약어
회원 (Member)	학번	member_id	mem_id
	비밀번호	member_password	mem_pw
	학과	department	department
	이름	member_name	mem_name
	성별	gender	gender
파티 (Party)	글 번호	party_number	pt_number
	작성자	writer	writer
	파티 이름	party_name	pt_name
	카테고리	category	category
	매장 이름	store	store
	먹을 날짜	date_to_eat	date_eat
	먹을 시간	time_to_eat	time_eat
	먹을 방식	way_of_eating	way_eat
	배달 위치	location	location
	결제 방식	payment	payment
	배달 플랫폼	platform	platform
	모집 인원	number_of_applicants	num_applicants
	현재 모집 인원	current_number_of_applicants	cnum_applicants
	채팅 링크	link	link
	완료 여부	completion	completion
파티 신청 (Application)	글 번호	application_number	app_number
	신청자	applicant	applicant
	신청 시간	application_time	app_time
	수락 여부	acceptance_status	accept
관리자 (Manager)	아이디	manager_id	man_id
	비밀번호	manager_password	man_pw
	이름	manager_name	man_name
	전화번호	phone	phone

7. 도메인 기술서

테이블	물리명	약어	데이터 저장 타입
회원 (Member)	member_id	mem_id	INT
	member_password	mem_pw	CHAR(40)
	department	department	VARCHAR(50)
	member_name	mem_name	VARCHAR(30)
	gender	gender	VARCHAR(10)
파티 (Party)	party_number	pt_number	INT
	writer	writer	INT
	party_name	pt_name	VARCHAR(200)
	category	category	VARCHAR(40)
	store	store	VARCHAR(45)
	date_to_eat	date_eat	DATE
	time_to_eat	time_eat	TIME
	way_of_eating	way_eat	VARCHAR(20)
	location	location	VARCHAR(20)
	payment	payment	VARCHAR(30)
	platform	platform	VARCHAR(30)
	number_of_applicants	num_applicants	INT
	current_number_of_applicants	cnum_applicants	INT
	link	link	VARCHAR(200)
	completion	completion	CHAR(15)
파티 신청 (Application)	application_number	app_number	INT
	applicant	applicant	INT
	application_time	app_time	DATETIME
	acceptance_status	accept	CHAR(15)
관리자 (Manager)	manager_id	man_id	INT
	manager_password	man_pw	CHAR(40)
	manager_name	man_name	VARCHAR(30)
	phone	phone	CHAR(30)

8. 테이블 기술서

테이블명		Member				
기능		회원가입을 한 회원들의 정보가 저장되는 테이블				
No	Attribute	Domain	NN	K.T.	Default	Description
1	mem_id	INT	Y	PK		회원의 학번 저장, ID
2	mem_pw	CHAR(40)	Y			회원의 비밀번호 저장
3	department	VARCHAR(50)	Y			회원의 학과 저장
4	mem_name	VARCHAR(30)	Y			회원의 이름 저장
5	gender	VARCHAR(10)	Y			회원의 성별 저장
기타 참고 사항						

테이블명		Party				
기능		회원이 배달 시킬 사람을 구하는 게시물을 올릴 시 관련 정보가 저장되는 테이블				
No	Attribute	Domain	NN	K.T.	Default	Description
1	pt_number	INT	Y	PK		글 번호 저장
2	writer	INT		FK		작성자 저장 - Member의 mem_id
3	pt_name	VARCHAR(200)	Y			게시물의 제목 저장
4	category	VARCHAR(40)	Y			배달 음식 유형 저장
5	store	VARCHAR(45)	Y			배달 매장 이름 저장
6	date_eat	DATE	Y			먹을 날짜 저장 - YYYY-MM-DD 형태
7	time_eat	TIME	Y			먹을 시간 저장 - HH:MM:SS 형태
8	way_eat	VARCHAR(20)	Y			먹을 방법 저장
9	location	VARCHAR(20)	Y			모일 장소 저장
10	payment	VARCHAR(30)	Y			결제 방법 저장
11	platform	VARCHAR(30)	Y			배달 시킬 플랫폼 저장
12	num_applicants	INT	Y			모집할 인원 저장
13	cnum_applicants	INT	Y		0	현재 모인 인원 저장
14	link	VARCHAR(200)	Y			신청자들이 들어갈 채팅 링크 저장
15	completion	CHAR(15)	Y		'미완료'	모집 완료 여부 저장
기타 참고 사항						

pt_number는 AUTO_INCREMENT로 자동으로 1부터 값이 1씩 증가하며 저장된다.

테이블명	Application					
기능	회원들이 Party를 보고 신청 시 관련 정보가 저장되는 테이블					
No	Attribute	Domain	NN	K.T.	Default	Description
1	app_number	INT	Y	PFK		신청한 글 번호 저장 - Party의 pt_number
2	applicant	INT	Y	PFK		신청자 저장 - Member의 mem_id
3	app_time	DATETIME	Y			신청한 시간 저장
4	accept	CHAR(15)	Y		'미확인'	작성자가 신청자를 받았는지 여부 저장
기타 참고 사항						

테이블명	Manager					
기능	회원, 게시물을 관리할 수 있는 관리자의 정보가 저장되는 테이블					
No	Attribute	Domain	NN	K.T.	Default	Description
1	man_id	INT	Y	PK		관리자 아이디 저장
2	man_pw	CHAR(40)	Y			관리자 비밀번호 저장
3	man_name	VARCHAR(30)	Y			관리자 이름 저장
4	phone	CHAR(30)	Y			관리자 전화번호 저장
기타 참고 사항						

9. MySQL 코드

데이터베이스, 사용자 생성 코드

```
CREATE DATABASE Together;
```

```
CREATE user 'together_user'@'localhost' IDENTIFIED BY 'together';
```

```
GRANT ALL PRIVILEGES ON Together.* TO 'together_user'@'localhost' WITH GRANT OPTION;
```

```
FLUSH PRIVILEGES;
```

테이블 생성 코드

```
USE NNMJ;
```

```
DROP TABLE IF EXISTS Member;
```

```
CREATE TABLE Member(
```

```
    mem_id          INT                NOT NULL,
```

```
    mem_pw          CHAR(40)          NOT NULL,
```

```
    department     VARCHAR(50)       NOT NULL,
```

```
    mem_name       VARCHAR(30)       NOT NULL,
```

```
    gender         VARCHAR(10)       NOT NULL,
```

```
    PRIMARY KEY(mem_id)
```

```
) ENGINE = InnoDB      DEFAULT CHARSET = utf8mb4;
```

```
DROP TABLE IF EXISTS Party;
```

```
CREATE TABLE Party(
```

```
    pt_number       INT                NOT          NULL
```

```
    AUTO_INCREMENT,
```

```
    writer          INT,
```

```
    pt_name         VARCHAR(200)      NOT NULL,
```

```
    category        VARCHAR(40)       NOT NULL,
```

```
    store           VARCHAR(45)       NOT NULL,
```

```
    date_eat        DATE              NOT NULL,
```

```
    time_eat        TIME              NOT NULL,
```

```
    way_eat         VARCHAR(20)       NOT NULL,
```

```
    location        VARCHAR(20)       NOT NULL,
```

```
    payment         VARCHAR(30)       NOT NULL,
```

```
    platform        VARCHAR(30)       NOT NULL,
```

```

num_applicants    INT                                NOT NULL,
cnum_applicants    INT                                NOT NULL    DEFAULT 0,
link               VARCHAR(200)    NOT NULL,
completion         CHAR(15)                NOT NULL    DEFAULT '미완료',
PRIMARY KEY(pt_number),
FOREIGN KEY(writer) REFERENCES Member(mem_id) ON UPDATE CASCADE ON DELETE SET
NULL
) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4;

DROP TABLE IF EXISTS Application;
CREATE TABLE Application(
    app_number    INT                                NOT NULL,
    applicant     INT                                NOT NULL,
    app_time      DATETIME    NOT NULL,
    accept        CHAR(15)                NOT NULL    DEFAULT '미확인',
    PRIMARY KEY(app_number, applicant),
    FOREIGN KEY(app_number) REFERENCES Party(pt_number) ON UPDATE CASCADE ON
DELETE CASCADE,
    FOREIGN KEY(applicant) REFERENCES Member(mem_id) ON UPDATE CASCADE ON DELETE
CASCADE
) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4;

DROP TABLE IF EXISTS Manager;
CREATE TABLE Manager(
    man_id        INT                                NOT NULL,
    man_pw        CHAR(40)                NOT NULL,
    man_name      VARCHAR(30)    NOT NULL,
    phone         CHAR(30)                NOT NULL,
    PRIMARY KEY(man_id)
) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4;

```

insert문 예제

```
USE NNMJ;
```

```
# 학번, 비밀번호, 과, 이름, 성별 순으로 넣기
```

```
INSERT INTO Member(mem_id, mem_pw, department, mem_name, gender)
```

```
VALUES(22160007, 'password', '스마트소프트웨어학과', '김경선', '여자');
```

```
select * from member;
```

```
# 작성자(FK, Member의 mem_id 중 하나 넣어야 함), 파티 이름, 카테고리, 매장 이름, 먹을 날짜, 먹을 시간, 먹을 방법, 장소, 결제 방법, 배달 플랫폼, 모을 인원, 채팅 링크 순으로 넣기
```

```
# pt_number(글 번호)는 AUTO_INCREMENT로 넣을 필요 없음
```

```
# cnum_applicants(현재 모집 인원)은 DEFAULT 값(0)이 있으니 넣을 필요 없음
```

```
# completion(완료 여부)는 DEFAULT 값('미완료')이 있으니 넣을 필요 없음
```

```
INSERT INTO Party(writer, pt_name, category, store, date_eat, time_eat, way_eat, location, payment, platform, num_applicants, link)
```

```
VALUES(22160007, '파티 이름', '카테고리', '매장 이름', '2022-12-03', '23:26:00', '같이먹자', '산학협동관', '만나서후결제', '배달의민족', 5, 'https://kakao.???');
```

```
select * from party;
```

```
# 글 번호(FK, Party의 pt_number 중 하나 넣어야 함), 신청자(FK, Member의 mem_id 중 하나 넣어야 함), 신청 시간 순으로 넣기
```

```
# accept(수락 여부)는 DEFAULT 값('미확인')이 있으니 넣을 필요 없음
```

```
# now()로 현재 시간 넣음
```

```
INSERT INTO Application(app_number, applicant, app_time)
```

```
VALUES(1, 22160007, now());
```

```
select * from application;
```

```
# 아이디, 비밀번호, 이름, 전화번호 순으로 넣기
```

```
INSERT INTO Manager(man_id, man_pw, man_name, phone)
```

```
VALUES(12345, 'password', '관리자', '01012345678');
```

```
select * from manager;
```