

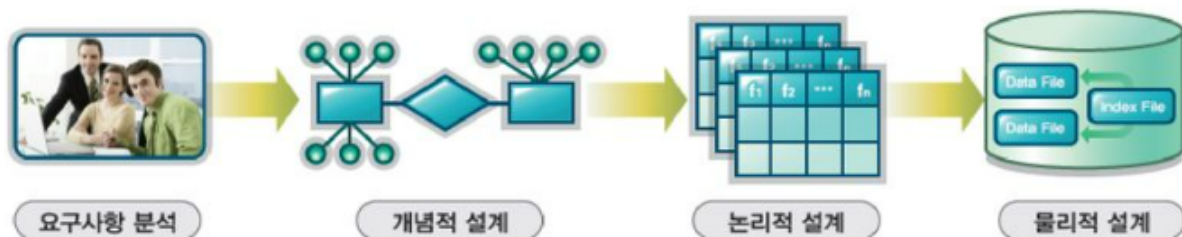


## [미니프로젝트] 3조 ❤️

### 🌴 프로젝트 소개

- 일상적인 업무와 생활에서의 스트레스와 피로로부터 벗어나기 위해 여행은 훌륭한 수단입니다. 하지만, 바쁜 일상과 업무에 쫓기는 현대인들에게 멀리 여행을 가는 것은 부담스러운 일이 될 수 있습니다. 호캉스는 그러한 일상에서 벗어나 휴식을 취하고 안정을 찾을 수 있는 기회를 제공합니다. 이러한 이유로 호캉스는 현대의 사람들에게 매력적인 선택지입니다.
- 호텔 예약 시스템에 대한 데이터베이스 설계를 수행했습니다. 이 프로젝트는 기존 호텔 예약 웹 사이트의 플로우를 참고하여 새로운 데이터베이스를 디자인하고 구현하는 것이 목표였습니다.

### 데이터베이스 설계 단계



---

## 요구사항 분석

- 기존 사이트의 플로우를 활용하여 사용자가 호텔을 검색하고 예약하는 프로세스를 지원하는 데이터베이스를 개발했습니다.

---

## 데이터베이스 개념적 설계



개념적 설계는 요구 사항 분석 결과물을 개념적 데이터 모델(ER모델 등)을 사용해, 개념적 구조로 표현

### 작업 과정

1. 개체 추출, 각 개체의 주요 속성과 키 속성 선별
2. 개체 간의 관계 설정
3. E-R 다이어그램으로 표현

## ER 모델( Entity Relation Model )



ER 모델은 요구사항으로부터 얻어낸 정보들을 개체(Entity), 애트리뷰트(Attribute), 관계성(Relation)으로 기술하는 데이터 모델을 말합니다.

- **개체의 유형( Entity Type )**
  - 개체란 단독으로 존재하는 객체를 의미하며, 동일한 객체는 존재하지 않습니다.
    - 회원 한 명
  - 개체들의 집합을 Entity Type이라고 합니다.
    - 개체가 회원 한 명이면, 개체들의 회원이 됩니다.
  - ER 다이어그램(ERD)에서 Entity Type은 네모로 표현합니다.
- **애트리뷰트, 속성( Attribute )**

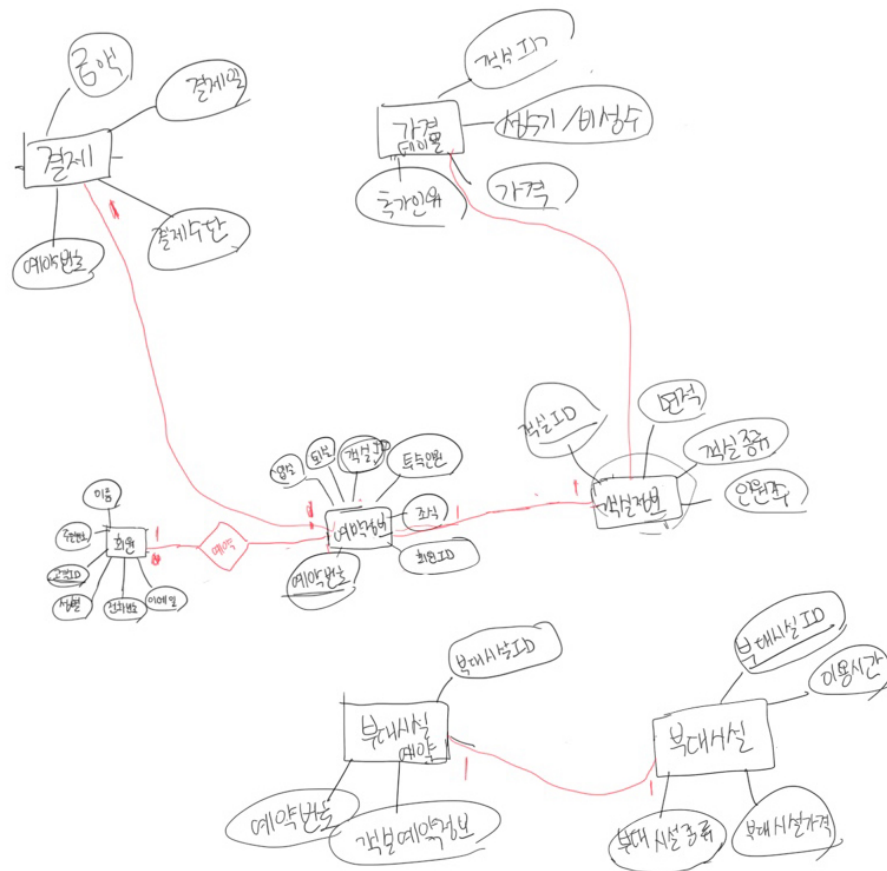
- 개체들의 집합을 Entity Type이라고 합니다.
- 예를 들면 회원의 이름, 나이, 성별이 있습니다.
- ER 다이어그램(ERD)에서 Attribute는 원으로 표현합니다.

## • 관계 ( Relation )

- Entity Type간의 관계를 의미합니다.
- 회원과 호텔의 객실은 “예약”이라는 관계를 갖습니다.
- ER 다이어그램(ERD)에서 Relation은 마름모로 표현합니다.

## 개체관계 다이어그램(ERD) 러프

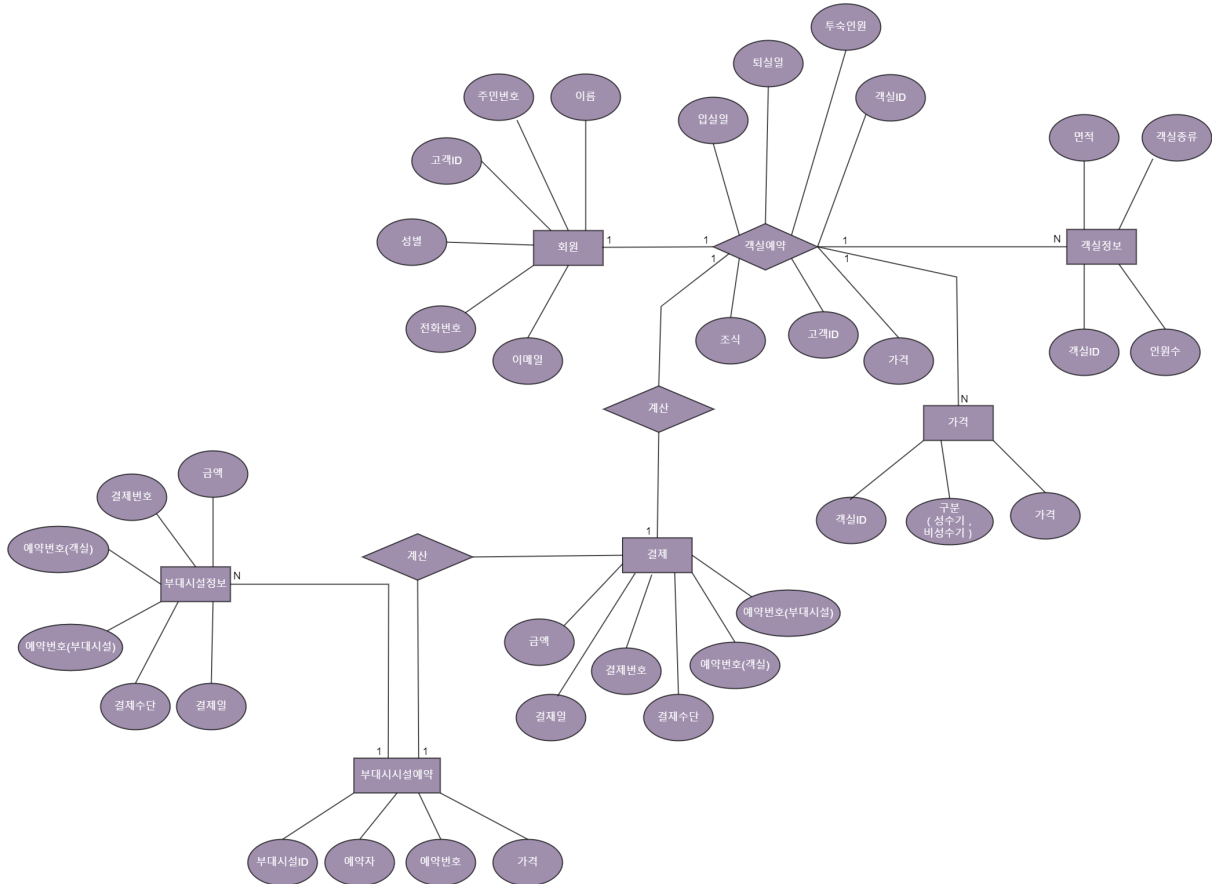
- 각 개체의 주요 속성과 키 속성을 러프하게 손으로 작성



## 개체관계 다이어그램(ERD)

- E-R다이어그램을 프로그램을 사용해 작성해보았습니다.

- 사용한 프로그램: [draw.io](https://draw.io)



## 데이터베이스 논리적 설계



개념적 스키마를 논리적 데이터 모델을 이용해 논리적 구조로 표현  
(일반적으로 관계 데이터 모델을 많이 사용)

### 작업 과정

1. 개념적 설계 단계의 결과물인 E-R 다이어그램을 릴레이션 스키마로 변환
2. 속성의 데이터 타입, 길이, 널 값 허용 여부, 기본 값 등을 세부적으로 결정하고 결과를 문서화 시킴

## 1차 설계

- 사용한 툴: AQueryTool (5개 생성 제한)
- E-R다이어그램의 개체를 릴레이션으로 변환한다.

User						
회원						
PK	AI	FK Null	Logical Name	Name	Type	
		+	고객 I D	user_id	VARCHAR(50)	
		+	이름	user_name	VARCHAR(50)	
		+	주민번호	user_rn	VARCHAR(50)	
		+	성별	user_gender	VARCHAR(50)	
		+	전화번호	user_tel	VARCHAR(50)	
		+	이메일	user_email	VARCHAR(50)	

RoomInfo						
객실정보						
PK	AI	FK Null	Logical Name	Name	Type	
		+	객실 I D	room_id	INT UNSIGNED	
		+	객실종류	room_type	VARCHAR(50)	
		+	면적	room_area	FLOAT	
		+	인원수	room_peoplenum	INT UNSIGNED	

Price						
객실가격						
PK	AI	FK Null	Logical Name	Name	Type	
		+	객실 I D	room_id	INT UNSIGNED	
		+	구분 ( 성수기 , t	price_sep	VARCHAR(50)	
		+	추가인원	price_addpeople	INT UNSIGNED	
		+	가격	price_price	INT UNSIGNED	

Reservation						
객실예약						
PK	AI	FK Null	Logical Name	Name	Type	
		+	고객 I D	user_id	VARCHAR(50)	
		+	예약번호	res_number	INT UNSIGNED	
		+	입실일	res_checkin	DATE	
		+	퇴실일	res_checkout	DATE	
		+	투숙인원	res_people	INT UNSIGNED	
		+	객실 I D	res_roomid	INT UNSIGNED	
		+	가격	res_roomtotal	INT UNSIGNED	

Facilities_Info						
부대시설정보						
PK	AI	FK Null	Logical Name	Name	Type	
		+	부대시설ID	facility_id	INT UNSIGNED	
		+	부대시설종류	facility_type	VARCHAR(50)	
		+	오픈시간	facility_open	TIME	
		+	마감시간	facility_close	TIME	
		+	가격	facility_price	INT UNSIGNED	

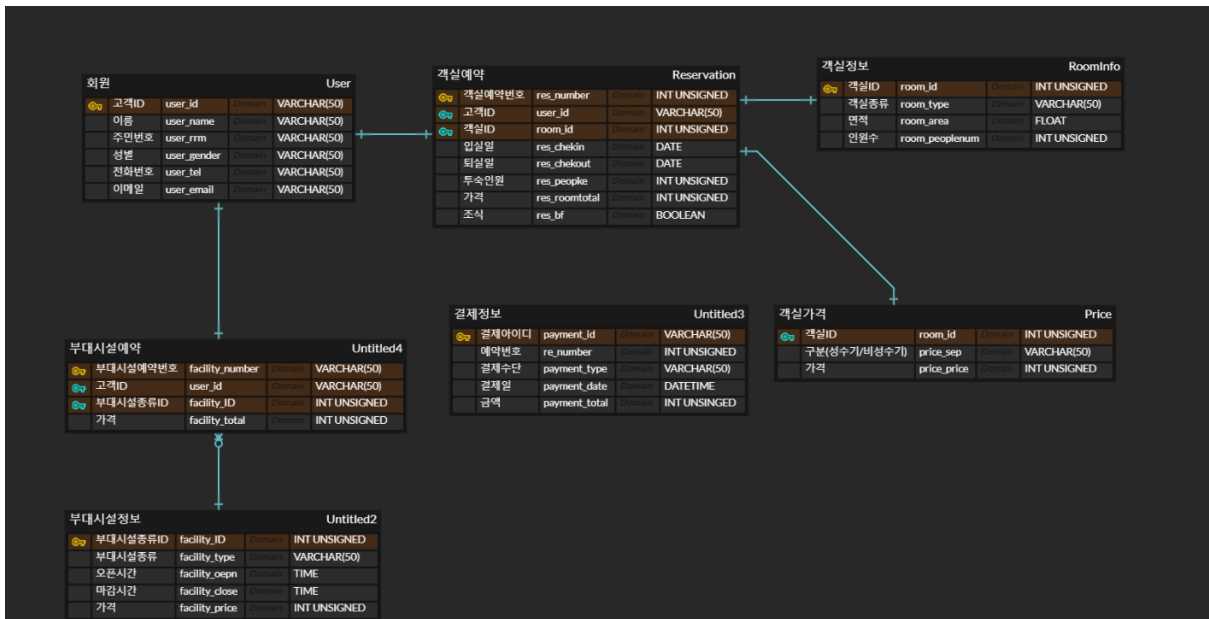
Facilities						
부대시설예약						
PK	AI	FK Null	Logical Name	Name	Type	
		+	부대시설ID	facility_ID	INT UNSIGNED	
		+	예약자	facility_user	VARCHAR(50)	
		+	예약번호	facility_number	VARCHAR(50)	
		+	가격	facility_total	INT UNSIGNED	

Payment						
결제정보						
PK	AI	FK Null	Logical Name	Name	Type	
		+	결제번호	payment_id	VARCHAR(50)	
		+	예약번호(객실)	rse_number	INT UNSIGNED	
		+	예약번호(부대시	facility_number	INT UNSIGNED	
		+	결제수단	payment_type	VARCHAR(50)	
		+	결제일	payment_date	DATETIME	
		+	금액	payment_total	INT UNSIGNED	

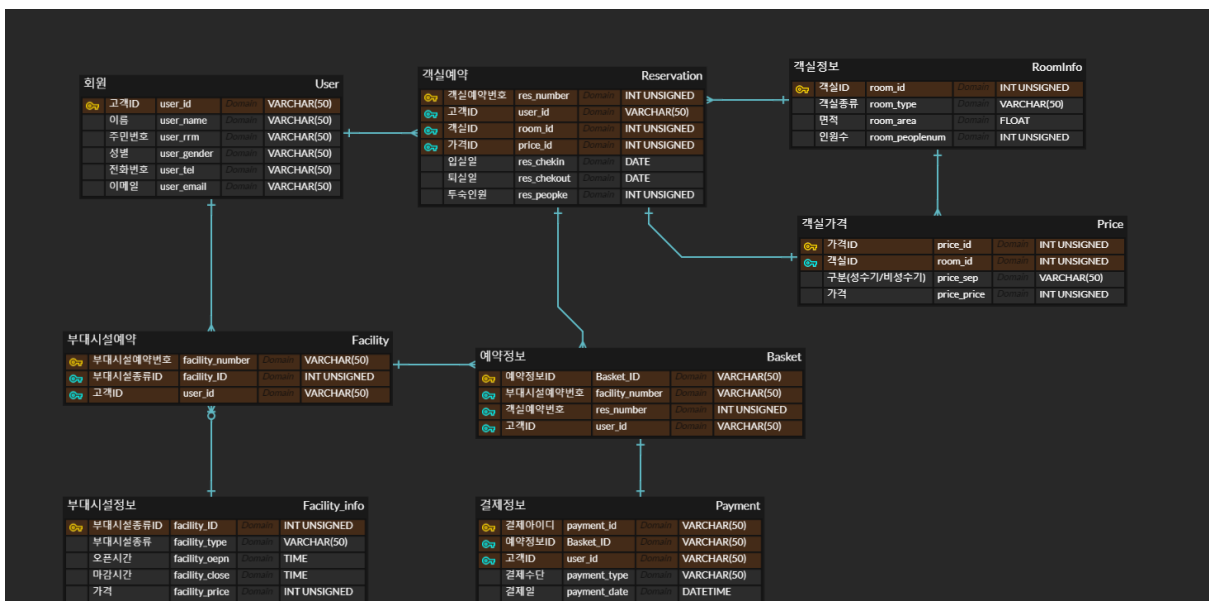
## 2차 설계

- 사용한 툴: erdcloud

- 1차 설계를 기반으로, 릴레이션간의 관계를 추가하고, 기본키와 외래키 참조 설정을 했습니다.



## 설계 완성



## 테이블 명세서 작성

- 위의 설계를 토대로 테이블 명세서 작성.

- 테이블 명세서는 각 테이블의 속성과, 해당 속성의 데이터 유형, 제약 조건 등을 기술한 문서입니다.

테이블 이름			회원(User)			
속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
고객ID	varchar(50)	N		PK		
이름	varchar(50)	N				
주민등록번호	varchar(50)	N				
성별	boolean	N				
전화번호	varchar(50)	N				
이메일	varchar(50)	Y				

테이블 이름			객실예약(Reservation)			
속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
객실예약번호	int unsigned	N		PK		
고객ID	varchar(50)	N			FK	
객실ID	int unsigned	N			FK	
가격ID	int unsigned	N			FK	
입실일	date	N				
퇴실일	date	N				
투숙인원	int unsigned	N	0			1이상

테이블 이름			객실정보(RoomInfo)			
속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
객실ID	int unsigned	N		PK		
객실종류	varchar(50)	N				
면적	float	N				
인원수	int unsigned	N				1이상

테이블 이름			객실가격(Price)			
속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
객실가격	int unsigned	N		PK		
객실ID	int unsigned	N			FK	
구분(성수기/비성수기)	varchar(50)	N				
가격	int unsigned	N				

테이블 이름			부대시설예약(Facility)			
속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
부대시설예약번호	varchar(50)	N		PK		
고객ID	varchar(50)	N			FK	
부대시설종류ID	int unsigned	N			FK	

테이블 이름			부대시설정보(Facility_Info)			
속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
부대시설종류ID	int unsigned	N		PK		
부대시설종류	varchar(50)	N				
오픈시간	time	N				
마감시간	time	N				
가격	int unsigned	N				

테이블 이름			예약정보(Basket)			
속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
예약정보ID	varchar(50)	N		PK		
부대시설예약번호	varchar(50)	N			FK	
객실예약번호	int unsigned	N			FK	

테이블 이름			결제(Payment)			
속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
결제ID	varchar(50)	N		PK		
예약정보ID	varchar(50)	N			FK	
결제수단	varchar(50)	N				
결제일	datetime	N				

## 데이터베이스 물리적 설계



논리적 설계에서 나온 테이블 및 인덱스를 실제 데이터베이스 시스템에 구현하기 위한 단계

### 샤그니엘 예약 시스템.sql

```
CREATE TABLE `User` (
  `user_id` VARCHAR(50) NOT NULL,
  `user_name` VARCHAR(50) NOT NULL,
  `user_rrm` VARCHAR(50) NOT NULL,
  `user_gender` VARCHAR(50) NOT NULL,
```

```

`user_tel` VARCHAR(50) NOT NULL,
`user_email` VARCHAR(50) NULL
);

CREATE TABLE `RoomInfo` (
  `room_id` INT UNSIGNED NOT NULL,
  `room_type` VARCHAR(50) NOT NULL,
  `room_area` FLOAT NOT NULL,
  `room_peoplenum` INT UNSIGNED NULL
);

CREATE TABLE `Facility_info` (
  `facility_ID` INT UNSIGNED NOT NULL,
  `facility_type` VARCHAR(50) NOT NULL,
  `facility_oepn` TIME NOT NULL,
  `facility_close` TIME NOT NULL,
  `facility_price` INT UNSIGNED NOT NULL
);

CREATE TABLE `Facility` (
  `facility_number` VARCHAR(50) NOT NULL,
  `facility_ID` INT UNSIGNED NOT NULL,
  `user_id` VARCHAR(50) NOT NULL
);

CREATE TABLE `Price` (
  `price_id` INT UNSIGNED NOT NULL,
  `room_id` INT UNSIGNED NOT NULL,
  `price_sep` VARCHAR(50) NOT NULL,
  `price_price` INT UNSIGNED NOT NULL
);

CREATE TABLE `Payment` (
  `payment_id` VARCHAR(50) NOT NULL,
  `Basket_ID` VARCHAR(50) NOT NULL,
  `user_id` VARCHAR(50) NOT NULL,
  `payment_type` VARCHAR(50) NOT NULL,
  `payment_date` DATETIME NOT NULL
);

CREATE TABLE `Reservation` (
  `res_number` INT UNSIGNED NOT NULL,
  `user_id` VARCHAR(50) NOT NULL,
  `room_id` INT UNSIGNED NOT NULL,
  `price_id` INT UNSIGNED NOT NULL,
  `res_chekin` DATE NOT NULL,
  `res_chekout` DATE NOT NULL,
  `res_peopke` INT UNSIGNED NOT NULL
);

CREATE TABLE `Basket` (
  `Basket_ID` VARCHAR(50) NOT NULL,
  `facility_number` VARCHAR(50) NOT NULL,
  `res_number` INT UNSIGNED NOT NULL,
  `user_id` VARCHAR(50) NOT NULL
);

```



```

ALTER TABLE `User` ADD CONSTRAINT `PK_USER` PRIMARY KEY (
    `user_id`
);

ALTER TABLE `RoomInfo` ADD CONSTRAINT `PK_ROOMINFO` PRIMARY KEY (
    `room_id`
);

ALTER TABLE `Facility_info` ADD CONSTRAINT `PK_FACILITY_INFO` PRIMARY KEY (
    `facility_ID`
);

ALTER TABLE `Facility` ADD CONSTRAINT `PK_FACILITY` PRIMARY KEY (
    `facility_ID`
);

ALTER TABLE `Price` ADD CONSTRAINT `PK_PRICE` PRIMARY KEY (
    `price_id`,
    `room_id`
);

ALTER TABLE `Payment` ADD CONSTRAINT `PK_PAYMENT` PRIMARY KEY (
    `payment_id`,
    `Basket_ID`,
    `user_id`
);

ALTER TABLE `Reservation` ADD CONSTRAINT `PK_RESERVATION` PRIMARY KEY (
    `res_number`
);

ALTER TABLE `Basket` ADD CONSTRAINT `PK_BASKET` PRIMARY KEY (
    `Basket_ID`
);

ALTER TABLE `Facility` ADD CONSTRAINT `FK_Facility_info_TO_Facility_1` FOREIGN KEY (
    `facility_ID`
)
REFERENCES `Facility_info` (
    `facility_ID`
);

ALTER TABLE `Facility` ADD CONSTRAINT `FK_User_TO_Facility_1` FOREIGN KEY (
    `user_id`
)
REFERENCES `User` (
    `user_id`
);

ALTER TABLE `Price` ADD CONSTRAINT `FK_RoomInfo_TO_Price_1` FOREIGN KEY (
    `room_id`
)
REFERENCES `RoomInfo` (
    `room_id`
);

```

```

ALTER TABLE `Payment` ADD CONSTRAINT `FK_Basket_TO_Payment_1` FOREIGN KEY (
    `Basket_ID`
)
REFERENCES `Basket` (
    `Basket_ID`
);

ALTER TABLE `Payment` ADD CONSTRAINT `FK_Basket_TO_Payment_2` FOREIGN KEY (
    `user_id`
)
REFERENCES `Basket` (
    `user_id`
);

ALTER TABLE `Reservation` ADD CONSTRAINT `FK_User_TO_Reservation_1` FOREIGN KEY (
    `user_id`
)
REFERENCES `User` (
    `user_id`
);

ALTER TABLE `Reservation` ADD CONSTRAINT `FK_RoomInfo_TO_Reservation_1` FOREIGN KEY (
    `room_id`
)
REFERENCES `RoomInfo` (
    `room_id`
);

ALTER TABLE `Reservation` ADD CONSTRAINT `FK_Price_TO_Reservation_1` FOREIGN KEY (
    `price_id`
)
REFERENCES `Price` (
    `price_id`
);

ALTER TABLE `Basket` ADD CONSTRAINT `FK_Facility_TO_Basket_1` FOREIGN KEY (
    `facility_number`
)
REFERENCES `Facility` (
    `facility_number`
);

ALTER TABLE `Basket` ADD CONSTRAINT `FK_Facility_TO_Basket_2` FOREIGN KEY (
    `user_id`
)
REFERENCES `Facility` (
    `user_id`
);

ALTER TABLE `Basket` ADD CONSTRAINT `FK_Reservation_TO_Basket_1` FOREIGN KEY (
    `res_number`
)
REFERENCES `Reservation` (

```

```
`res_number`  
);
```

## ★ 프로젝트를 하면서 느낀 점 ★



**SWCAMP 1기\_안우용** 오후 12:25

DB 설계가 예상보다 쉽지 않다는 것을 깨닫게 되었습니다. DB 설계는 프로젝트의 핵심 단계로, 수업에서 배운 내용만으로는 실제 업무에서 활용하기에는 부족하다는 것을 깨달았습니다. 특히, 정규화의 중요성을 인지하고, 프로젝트를 진행할 때 어떻게 접근해야 하는지에 대한 고민을 먼저 하는 습관을 키워야 한다는 것을 알게 되었습니다.



**SWCAMP 1기\_고라희** 오후 12:30

DB 설계 작업을 처음 경험하면서 설계의 중요성을 알게 되었습니다. 팀원들과의 원활한 소통과 협업으로 진행되는 과정에서 나아가는 데 뿌듯함을 느꼈습니다. 그러나 처음부터 명확한 주제 선정과 구현까지 세부적으로 진행했다면 결과물을 더욱 효과적으로 완성할 수 있었을 것이라는 아쉬움이 남습니다. 이번 경험을 토대로 다음 프로젝트에서는 보다 효율적으로 작업을 진행하여 좋은 결과물로 나타났으면 좋겠습니다.



**SWCAMP 1기\_김단아** 오후 12:37

주제를 선정하고 그에 대한 DB 설계를 단계별로 진행해보면서, DB 설계가 생각보다도 더 많은 시행착오 끝에 구현된다는 사실을 깨닫게 되었습니다. 문제점과 보완점을 찾아가는 과정이 쉽지만은 않았지만 팀원들과 소통하고 해결해가며 무사히 완성할 수 있었습니다. 이론만이 아닌 실제로 DB 설계를 경험하고 배우게 되어 좋았다고 생각합니다.



**SWCAMP 1기\_정수민** 오후 12:38

DB설계를 상상으로 했던 것과 다르게, 단계별로 설계를 거치면서 수정되는 것들이 많았습니다. 수정에 수정을 거듭하면서 설계 단계의 중요성에 대해서 뼈저리게 느끼게 되었습니다. 이번의 경험을 바탕으로 다음에 설계를 할 때에는 각 설계의 단계에 대한 공부를 좀 더 해서, 요구사항 분석부터 순서대로 1정규화부터 3정규화를 거쳐, 구현까지 좀 더 발전한 결과물을 만들고 싶습니다.