

● 축산관리플랫폼

입사지원자 : 이창수

연락처 : 010-2982-4852

E-mail : ptg_kensin@naver.com

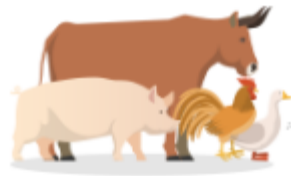
github.com/cesclee/intflowtest



가축 전염병 예방



생산성 개선



동물 복지 실현

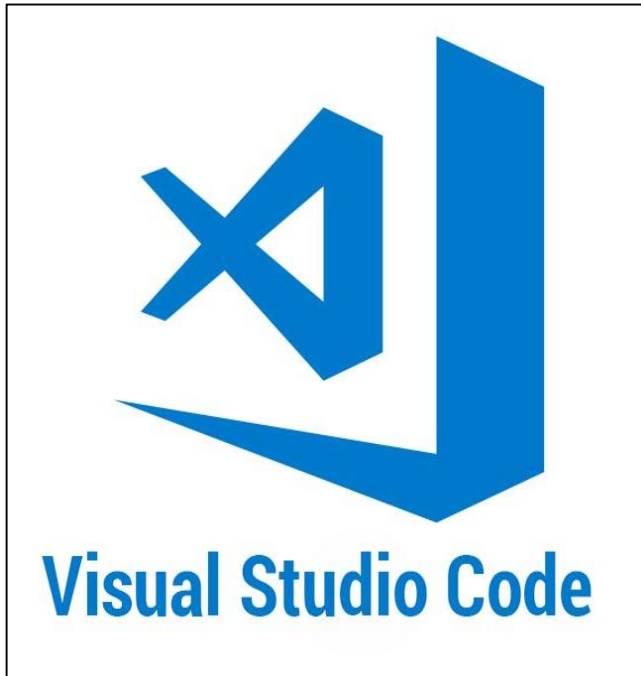




2021-02-10

Contents

- ❶ 요구사항정의
- ❷ Task 1
 - 사용 IDE, 응용프로그램
 - ERD (perter-chan)초안 제작
 - 정규화과정
 - 릴레이션 스키마 및 테이블명세서
 - 초안 수정 및 ERD생성
- ❸ Task 2
 - DB연동 테스트
 - 함수 5가지 수행여부 및 결과



DB 연동 및 함수제작 목적



ERD제작 및 DB구축 목적



정규화과정 및 테이블명세서 출력 목적


Task 1

[요구사항정의]

- 축산 관리 플랫폼에 **회원**으로 가입하려면 회원아이디, 비밀번호, 농장이름, 이메일, 전화번호 정보를 입력해야 한다.
- 로그인 페이지에서는 회원아이디, 비밀번호를 입력해야 한다.
- 농장 페이지에서는 사료재고를 확인할 수 있다.
- 농장 페이지에서는 **일정표**를 관리할 수 있는데, 세부적으로는 일정내용, 시작일시, 종료일시 정보를 저장, 삭제할 수 있다.
- 농장 페이지에서는 농장 내의 이상이 발생한 **개체 로그**를 확인할 수 있는데, 세부적으로는 이상징후, 발생일시, 개체정보가 있다.
- 관제 페이지에서는 농장 내의 모든 **카메라** 정보를 확인할 수 있다.
- 건물 페이지에서는 **건물** 내의 실시간 센서값을 확인할 수 있는데, 세부적으로는 온도, 습도, 풍량, 가스 정보를 시계열 그래프로 나타낸다.
- 구역 페이지에서는 구역내의 모든 카메라 정보를 확인할 수 있다.
- 구역 페이지에서 특정 카메라를 선택하면, 카메라 내의 모든 개체 정보를 확인할 수 있다.
- 구역 페이지에서 특정 개체를 선택하면, 해당 개체의 실시간 행동데이터를 확인할 수 있는데, 세부적으로는 발정, 이상행위, 활동성, 식사량, 음수량 정보를 시계열 그래프로 나타낸다.

※ 농장의 구조

- 농장 1:N 건물
- 건물 1:N 구역
- 구역 1:N 카메라
- 카메라 1:N 개체

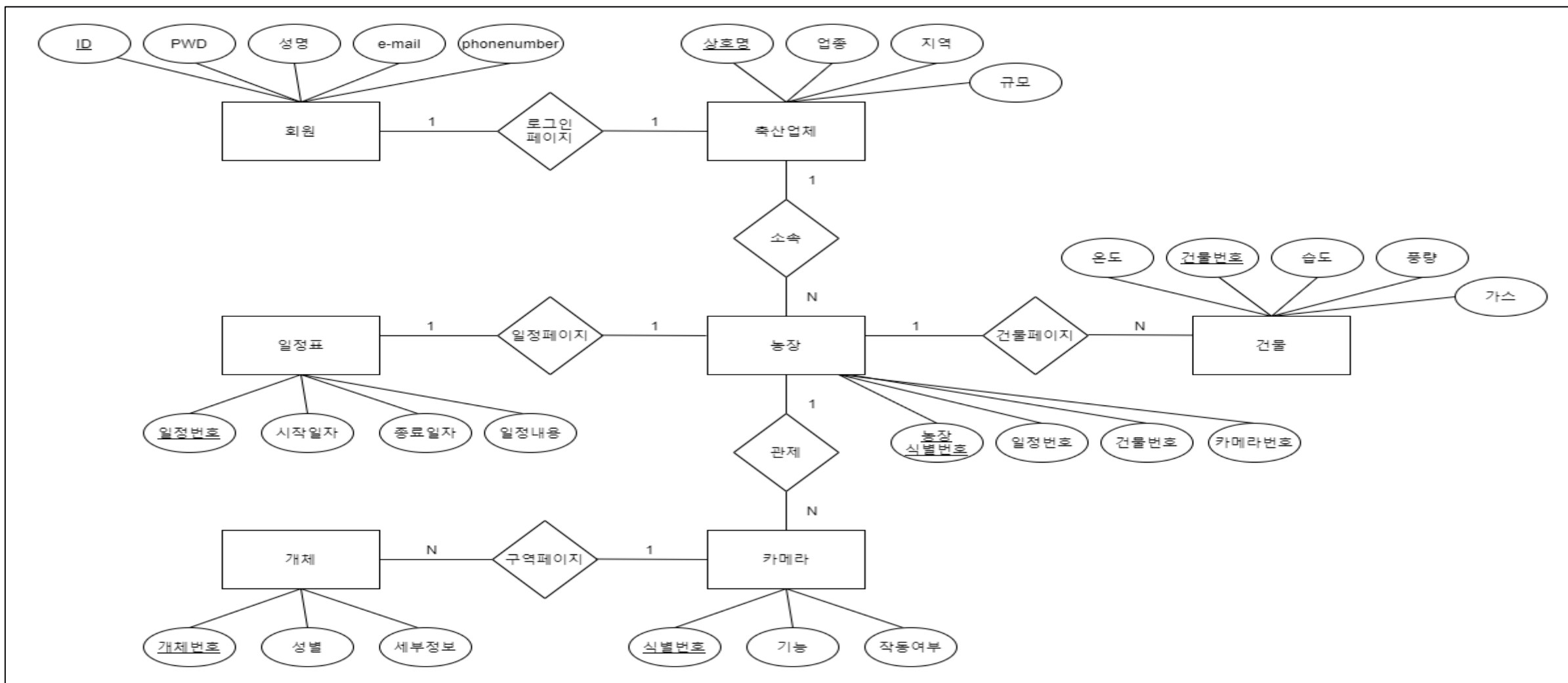
 테이블 선정 컬럼 선정 테이블 관계 설정

수행업무

1. ERD생성
2. 릴레이션 스키마 및 테이블 명세서
3. 정규화과정

수행방식

요구사항정의를 통한 도메인집합
정리와 피터찬방식의 ERD제작을
바탕으로 초안의 틀을 잡고
정규화과정을 거쳐 수정작업을 진행



ERD초안-notation:peter-chan

Task 1

요구사항분석

도메인정리

정규화과정

릴레이션 스키마 및 테이블명세서

초안 수정 및 ERD 생성

축산관리플랫폼_릴레이션스키마/테이블명세서																
회원								col_number	tablename	fieldname	data TYPE	data LENG	KEY	nullable	AutoIncrea	default
<u>ID</u>	PWD	name	e-mail	phone	상호명			1	animal	animalNo	int	int	PRI	NO		NULL
					FK			2	animal	sex	varchar	varchar(45)		YES		NULL
업체								3	animal	memo	longtext	longtext		YES		NULL
<u>상호명</u>	업종	region	규모(등급)					1	building	buildingNo	int	int	PRI	NO		NULL
								2	building	temperatur	float	float		YES		NULL
일정표								3	building	humidity	float	float		YES		NULL
<u>번호</u>	시작	종료	내용					4	building	wind	float	float		YES		NULL
								5	building	gas	float	float		YES		NULL
카메라								1	company	companyNo	int	int	PRI	NO		NULL
<u>카메라번호</u>	기능설명	작동여부						2	company	field	varchar	varchar(45)		YES		NULL
								3	company	location	varchar	varchar(45)		YES		NULL
건물								4	company	scale	int	int		YES		NULL
<u>건물번호</u>	온도	습도	풍량	가스				5	company	companyco	varchar	varchar(45)		YES		NULL
								1	controller_1	farm_num	int	int	PRI	NO		NULL
개체								2	controller_1	schedule_n	int	int		NO		NULL
<u>개체번호</u>	성별	세부정보						3	controller_1	building_n	int	int		NO		NULL
								4	controller_1	camera_nu	int	int		NO		NULL
농장								1	member	ID	int	int	PRI	NO	auto_incre	NULL
<u>식별번호</u>	<u>일정번호</u>	<u>건물번호</u>	<u>카메라번호</u>					2	member	PWD	varchar	varchar(45)		NO		NULL
								3	member	NAME	varchar	varchar(45)		YES		NULL
								4	member	E-MAIL	varchar	varchar(100)		YES		NULL
								5	member	PHONENU	varchar	varchar(50)		YES		NULL
								6	member	memberco	varchar	varchar(45)		YES		NULL
								1	schedule	schduleNu	int	int	PRI	NO		NULL
								2	schedule	start	varchar	varchar(45)		YES		NULL
								3	schedule	end	varchar	varchar(45)		YES		NULL
								4	schedule	memo	varchar	varchar(45)		YES		NULL

ERD초안에 근거한 RDB스키마/테이블명세서

요구사항분석

도메인정리

정규화과정

릴레이션 스키마 및 테이블명세서

초안 수정 및 ERD 생성

축산관리플랫폼_릴레이션스키마/테이블명세서																
회원							col_number	tablename	fieldname	data TYPE	data LENGTH	KEY	nullable	AutoIncrease	default	
ID	PWD	name	e-mail	phone	상호명			1 animal	animalNo	int	int	PRI	NO		NULL	
					FK			2 animal	sex	varchar	varchar(45)		YES		NULL	
업체								3 animal	memo	longtext	longtext		YES		NULL	
상호명	업종	region	규모(등급)					1 building	buildingNo	int	int	PRI	NO		NULL	
								2 building	temperature	float	float		YES		NULL	
일정표								3 building	humidity	float	float		YES		NULL	
번호	시작	종료	내용					4 building	wind	float	float		YES		NULL	
								5 building	gas	float	float		YES		NULL	
카메라								1 company	companyname	int	int	PRI	NO		NULL	
카메라번호	기능설명	작동여부						2 company	field	varchar	varchar(45)		YES		NULL	
								3 company	location	varchar	varchar(45)		YES		NULL	
건물								4 company	scale	int	int		YES		NULL	
건물번호	온도	습도	풍량	가스				5 company	companycol	varchar	varchar(45)		YES		NULL	
								1 controller_farm	farm_number	int	int	PRI	NO		NULL	
개체								2 controller_farm	schedule_number	int	int		NO		NULL	
개체번호	성별	세부정보						3 controller_farm	building_number	int	int		NO		NULL	
								4 controller_farm	camera_number	int	int		NO		NULL	
농장								1 member	ID	int	int	PRI	NO	auto_increment	NULL	
식별번호	일정번호	건물번호	카메라번호					2 member	PWD	varchar	varchar(45)		NO		NULL	
								3 member	NAME	varchar	varchar(45)		YES		NULL	
								4 member	E-MAIL	varchar	varchar(100)		YES		NULL	
								5 member	PHONENUMBER	varchar	varchar(50)		YES		NULL	
								6 member	membercol	varchar	varchar(45)		YES		NULL	
								1 schedule	schduleNumber	int	int	PRI	NO		NULL	
								2 schedule	start	varchar	varchar(45)		YES		NULL	
								3 schedule	end	varchar	varchar(45)		YES		NULL	
								4 schedule	memo	varchar	varchar(45)		YES		NULL	

-요구사항정의를 바탕으로 데이터베이스로 생성할 릴레이션 스키마 및 테이블명세서의 초안제작-



정규화 과정을 위한 도메인 정리 및 중복여부확인

회원아이디	회원비밀번호	회원이름	회원이메일	회원전화번호	회사명	회사지역	회사주소	일정내용	일정시작	일정종료	건물번호	센서번호	건물온도	건물습도	건물풍량	건물가스	카메라번호	개체정보	카메라정보	농장사료재고
abcd	1234	cesclee	ptg_kensin@naver.com	123-1234	ㅇㅇ축산	담양	담양군	xxx aaa	1 2	5 7	1 1	1 2								

데이터예시를 위한 임의의 값 선정

정규화가 필요한 테이블선정 및 도메인분할

회원											
회원아이디	회원비밀번호	회원이름	회원이메일	회원전화번호	회사명	회사지역	회사주소	일정번호	일정내용	일정시작	일정종료
abcd	1234	cesclee	ptg_kensin@naver.com	123-1234	ㅇㅇ축산	담양	담양군	1	xxx	1	5
건물							개체				
건물번호	카메라번호	건물습도	건물풍량	건물가스	농장사료재고		개체번호	개체상태	식사량	음수량	활동성
1	1						1	발정	부족	부족	y
1	2						1	발정	부족	부족	n
							1	발정	부족	부족	y

Task 1

요구사항분석	도메인정리	정규화과정	릴레이션 스키마 및 테이블명세서	초안 수정 및 ERD 생성
--------	-------	-------	-------------------	----------------

도메인정리																			
축산관리플랫폼																			
회원아이디	회원비밀번호	회원이름	회원이메일	회원전화번호	회사명	회사지역	회사주소	일정번호	일정내용	일정시작	일정종료	건물번호	센서번호	건물온도	건물습도	건물풍량	건물가스	카메라번호	개체정보
abcd	1234	cesclee	ptg_kensin@naver.com	123-1234	o o 축산	담양	담양군		xxx	1	5	1	1						
									aaa	2	7	1	2						
1NF																			
정규화가 필요한 테이블 : 축산관리플랫폼																			
회원																			
회원아이디	일정번호	회원비밀번호	회원이름	회원이메일	회원전화번호	회사명	회사지역	회사주소	일정내용	일정시작	일정종료								
abcd	1	1234	cesclee	kensin@naver.	123-1234	o o 축산	담양	담양군	xxx	1	5								
회원_건물_관계(1:N)																			
회원아이디	건물번호																		
abcd	1																		
건물																			
건물번호	센서번호	카메라번호	개체번호	건물습도	건물풍량	건물가스	농장사료재고	개체상태	식사량	음수량	활동성								
1	1	1	1					발정	부족	부족	y								
1	2	1	1					발정	부족	부족	n								

도메인 집합 정리 및 중복 값 제거에 따른 제1정규화

Task 1

요구사항분석

도메인정리

정규화과정

릴레이션 스키마 및 테이블명세서

초안 수정 및 ERD 생성

1NF														
정규화가 필요한 테이블 : 회원														
회원 : 하위테이블로 일정 테이블을 갖고 요구사항의 농장테이블은 건물테이블로 편입														
회원아이디	일정번호	회원비밀번호	회원이름	회원이메일	회원전화번호	회사명	회사지역	회사주소	일정내용	일정시작	일정종료			
abcd	1	1234	cesclee	kensin@naver.	123-1234	ㅇㅇ축산	담양	담양군	xxx	1	5			
회원_건물_관계														
회원아이디	건물번호													
abcd	1													
건물														
건물번호	카메라번호	개체번호	센서번호	건물습도	건물풍량	건물가스	농장사료재고	개체상태	식사량	음수량	활동성			
1	1	1	1					발정	부족	부족	y			
1	1	1	2					발정	부족	부족	n			

2NF																
2정규화가필요한 테이블: 회원, 건물																
회원									회원_일정_관계		일정					
회원아이디	회원비밀번호	회원이름	회원이메일	회원전화번호	회사명	회사지역	회사주소		회원아이디	일정번호		회원아이디	일정번호	일정내용	일정시작	일정종료
abcd	1234	cesclee	ptg_kensin@naver.com	123-1234	ㅇㅇ축산	담양	담양군		abcd	1		abcd	1	xxx	1	5
회원_건물_관계																
회원아이디	건물번호															
abcd	1															
건물																
건물번호	센서번호	건물습도	건물풍량	건물가스	농장사료재고	건물_카메라_관계		카메라		개체상태	식사량	음수량	활동성			
1	1						1	1		발정	부족	부족	y			
1	2									발정	부족	부족	n			

부분 함수 종속 제거에 따른 제2정규화

Task 1

요구사항분석	도메인정리	정규화과정	릴레이션 스키마 및 테이블명세서	초안 수정 및 ERD 생성
--------	-------	-------	-------------------	----------------

2NF																							
2정규화가필요한 테이블: 회원, 건물																							
회원								회원_일정_관계(1:N)				일정											
<u>회원아이디</u>	회원비밀번호	회원이름	회원이메일	회원전화번호	회사명	회사지역	회사주소	<u>회원아이디</u>	<u>일정번호</u>		<u>회원아이디</u>	<u>일정번호</u>	일정내용	일정시작	일정종료								
abcd	1234	cesclee	ptg_kensin@naver.com	123-1234	ㅇㅇ축산	담양	담양군	abcd	1		abcd	1	xxx	1	5								
회원_건물_관계(1:N)																							
<u>회원아이디</u>	<u>건물번호</u>																						
abcd	1																						
건물								건물_카메라_관계(1:N)				카메라											
<u>건물번호</u>	센서번호	농장사료재고	건물습도	건물풍량	건물가스			<u>건물번호</u>	<u>카메라번호</u>		<u>카메라번호</u>	<u>개체번호</u>	개체상태	식사량	음수량	활동성							
1	1							1	1		1	1	발정	부족	부족	y							
1	2										1	1	발정	부족	부족	n							
3NF																							
화가 필요한 테이블 : 건물																							
건물								건물_카메라_관계(1:N)				카메라											
<u>건물번호</u>	센서번호	농장사료재고	건물습도	건물풍량	건물가스			<u>건물번호</u>	<u>카메라번호</u>		<u>카메라번호</u>	<u>개체번호</u>	개체상태	식사량	음수량	활동성							
1	1							1	1		1	1	발정	부족	부족	y							
1	2										1	1	발정	부족	부족	n							
건물_센서_관계(1:N)																							
<u>건물번호</u>	<u>센서번호</u>																						
1	1																						
1	2																						
센서																							
<u>센서번호</u>	건물습도	건물풍량	건물가스																				
1																							

이행적 함수 종속 제거에 따른 제3정규화

릴레이션스키마															
회원								회원_일정_관계(1:N)		일정					
회원아이디	회원비밀번호	회원이름	회원이메일	회원전화번호	회사명	회사지역	회사주소	회원아이디	일정번호	회원아이디	일정번호	일정내용	일정시작	일정종료	
abcd	1234	cesclee	ptg_kensin@naver.com	123-1234	ㅇㅇ축산	담양	담양군	abcd	1	abcd	1	xxx	1	5	
회원_건물_관계(1:N)															
회원아이디	건물번호														
abcd	1														
건물								건물_카메라_관계(1:N)		카메라					
건물번호	센서번호	농장사료재고	건물습도	건물풍량	건물가스			건물번호	카메라번호	카메라번호	개체번호	개체상태	식사량	음수량	활동성
1	1							1	1	1	1	발정	부족	부족	y
1	2									1	1	발정	부족	부족	n
건물_센서_관계(1:N)															
건물번호	센서번호														
1	1														
1	2														
센서															
센서번호	건물습도	건물풍량	건물가스												
1															
2															

1. 회원테이블은 하위테이블로 일정테이블을 갖는다.(1:N)
2. 건물은 하위테이블로 센서,카메라 테이블을 갖는다.(1:N)

Task 1

요구사항분석

도메인정리

정규화과정

릴레이션 스키마 및 테이블명세서

초안 수정 및 ERD 생성

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

```
-- Table `mydb`.`member`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`member` (
  `mID` VARCHAR(45) NOT NULL,
  `mPWD` VARCHAR(45) NULL,
  `mNAME` VARCHAR(45) NULL,
  `E-MAIL` VARCHAR(45) NULL,
  `PHONE` VARCHAR(45) NULL,
  `mCOMPANY` VARCHAR(45) NULL,
  `mREGION` VARCHAR(45) NULL,
  `mLOCATION` VARCHAR(45) NULL,
  PRIMARY KEY (`mID`))
ENGINE = InnoDB;
```

```
-- Table `mydb`.`schedule`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`schedule` (
  `scheduleNo` INT NOT NULL,
  `start` DATE NULL,
  `end` DATE NULL,
  `memo` VARCHAR(100) NULL,
  `member_mID` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`scheduleNo`),
  INDEX `fk_schedule_member_idx` (`member_mID` ASC) VISIBLE,
  CONSTRAINT `fk_schedule_member`
    FOREIGN KEY (`member_mID`)
      REFERENCES `mydb`.`member` (`mID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`building`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`building` (
  `buildingNo` INT NOT NULL,
  `sensorNo` INT NULL,
  `feeds` FLOAT NULL,
  `member_mID` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`buildingNo`, `member_mID`),
  INDEX `fk_building_member1_idx` (`member_mID` ASC) VISIBLE,
  CONSTRAINT `fk_building_member1`
    FOREIGN KEY (`member_mID`)
      REFERENCES `mydb`.`member` (`mID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`camera`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`camera` (
  `cameraNo` INT NOT NULL,
  `animal` INT NULL,
  `states` VARCHAR(45) NULL,
  `foods` VARCHAR(45) NULL,
  `healthy` VARCHAR(45) NULL,
  `moves` VARCHAR(45) NULL,
  `water` VARCHAR(45) NULL,
  `building_buildingNo` INT NOT NULL,
  `building_member_mID` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`cameraNo`, `building_buildingNo`, `building_member_mID`),
  INDEX `fk_camera_building1_idx` (`building_buildingNo` ASC, `building_member_mID` ASC) VISIBLE,
  CONSTRAINT `fk_camera_building1`
    FOREIGN KEY (`building_buildingNo`, `building_member_mID`)
      REFERENCES `mydb`.`building` (`buildingNo`, `member_mID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`sensor`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`sensor` (
  `sensorNo` INT NOT NULL,
  `temperature` FLOAT NULL,
  `humidity` FLOAT NULL,
  `wind` FLOAT NULL,
  `gas` FLOAT NULL,
  `building_buildingNo1` INT NOT NULL,
  PRIMARY KEY (`sensorNo`, `building_buildingNo1`),
  INDEX `fk_sensor_building2_idx` (`building_buildingNo1` ASC) VISIBLE,
  CONSTRAINT `fk_sensor_building2`
    FOREIGN KEY (`building_buildingNo1`)
      REFERENCES `mydb`.`building` (`buildingNo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

ERD 제작 후 forward engineering 작업

Task 1

요구사항분석	도메인정리	정규화과정	릴레이션 스키마 및 테이블명세서	초안 수정 및 ERD 생성
--------	-------	-------	-------------------	----------------

테이블 구분 명세서

회원

count_number	tablename	fieldname	data TYPE	data LENGTH	KEY	nullable	AutoIncras	default
4	member	E-MAIL	varchar	varchar(45)		YES		NULL
6	member	mCOMPANY	varchar	varchar(45)		YES		NULL
1	member	mID	varchar	varchar(45)	PRI	NO		NULL
8	member	mLOCATION	varchar	varchar(45)		YES		NULL
3	member	mNAME	varchar	varchar(45)		YES		NULL
2	member	mPWD	varchar	varchar(45)		YES		NULL
7	member	mREGION	varchar	varchar(45)		YES		NULL
5	member	PHONE	varchar	varchar(45)		YES		NULL

일정

count_number	tablename	fieldname	data TYPE	data LENGTH	KEY	nullable	AutoIncras	default
3	schedule	end	date	date		YES		NULL
5	schedule	member_mID	varchar	varchar(45)	MUL	NO		NULL
4	schedule	memo	varchar	varchar(100)		YES		NULL
1	schedule	scheduleNo	int	int	PRI	NO		NULL
2	schedule	start	date	date		YES		NULL

건물

count_number	tablename	fieldname	data TYPE	data LENGTH	KEY	nullable	AutoIncras	default
1	building	buildingNo	int	int	PRI	NO		NULL
3	building	feeds	float	float		YES		NULL
4	building	member_mID	varchar	varchar(45)	PRI	NO		NULL
2	building	sensorNo	int	int		YES		NULL

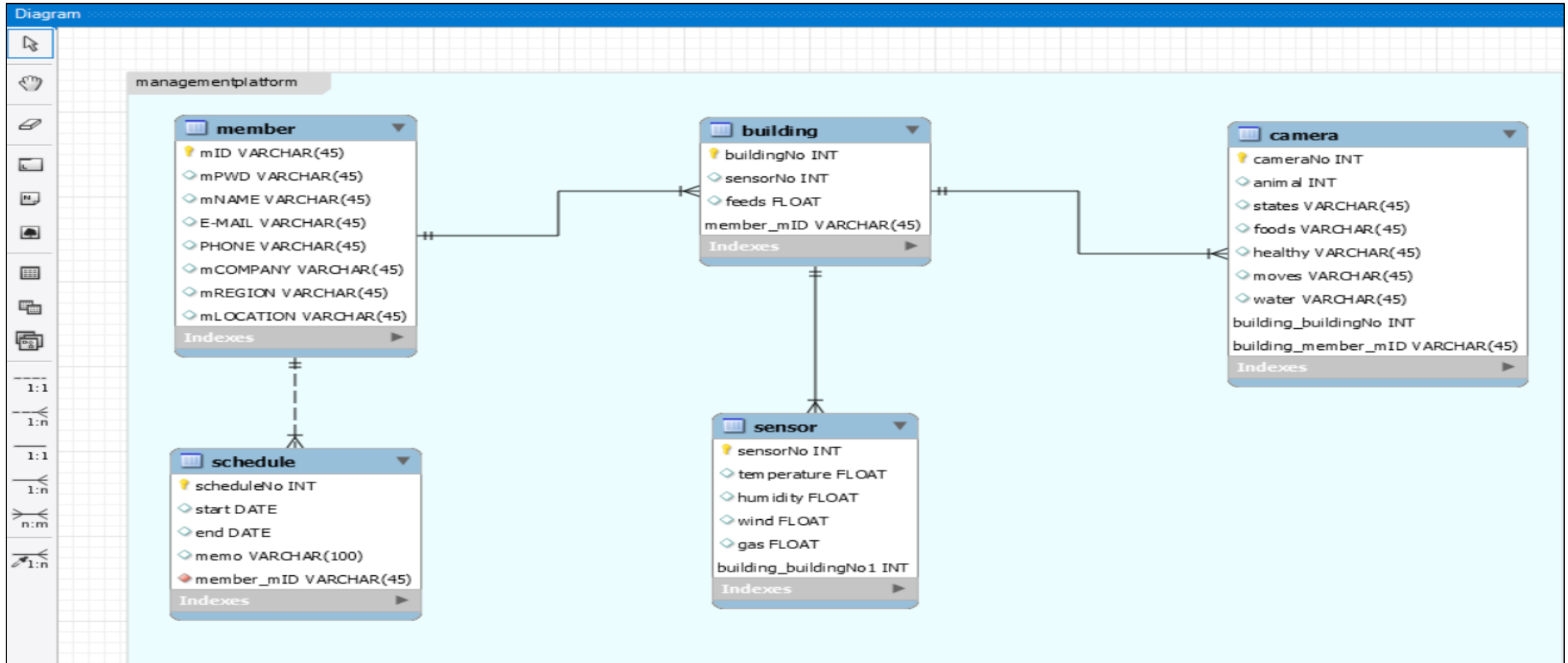
센서

count_number	tablename	fieldname	data TYPE	data LENGTH	KEY	nullable	AutoIncras	default
6	sensor	building_building	int	int	PRI	NO		NULL
5	sensor	gas	float	float		YES		NULL
3	sensor	humidity	float	float		YES		NULL
1	sensor	sensorNo	int	int	PRI	NO		NULL
2	sensor	temperature	float	float		YES		NULL
4	sensor	wind	float	float		YES		NULL

카메라

count_number	tablename	fieldname	data TYPE	data LENGTH	KEY	nullable	AutoIncras	default
2	camera	animal	int	int		YES		NULL
8	camera	building_building	int	int	PRI	NO		NULL
9	camera	building_membe	varchar	varchar(45)	PRI	NO		NULL
1	camera	cameraNo	int	int	PRI	NO		NULL
4	camera	foods	varchar	varchar(45)		YES		NULL
5	camera	healthy	varchar	varchar(45)		YES		NULL
6	camera	moves	varchar	varchar(45)		YES		NULL
3	camera	states	varchar	varchar(45)		YES		NULL
7	camera	water	varchar	varchar(45)		YES		NULL

쿼리문 사용하여 테이블명세서 출력 후 엑셀로 export



Mysql workbench에서 ERD 수정작업진행

Task 2

python으로 위에서 생성한 데이터베이스에 접근하고 아래 기능을 함수로 구현해주세요.

1. 1분마다 실시간 센서값을 삽입하는 함수
2. 1분마다 개체의 실시간 행동데이터 삽입하는 함수
3. 온도가 가장 높은 건물에 있는 개체들의 평균 활동성을 출력하는 함수
4. 통신이상으로 특정 구역의 데이터가 삽입되고 있지 않았을 경우를 가정하여
10분 동안 행동데이터의 삽입이 없는 구역을 가지고 있는 건물을 출력하는 함수
5. 최근 10분 내에 이상행위를 한 개체가 있는 건물을 출력하는 함수

※ 데이터가 필요한 경우, 생성한 데이터베이스에 임의의 값을 넣으셔도 됩니다.

Import Pymysql

Import threading

수행업무

Vscode에서 생성한 DB 접근 후
5가지 기능의 함수 구현

수행방식

DB연동을 위한 pymysql,
시간 간격으로 삽입, 출력을 위한
threading 패키지 Import

```

1  #링크테스트 pymysql 패키지연동
2  import pymysql
3  #db연동테스트를 위해 임의 멤버 값 1개 지정 후 테스트
4  conn=pymysql.connect
5  conn=pymysql.connect(host='localhost', user='root',password='rhtmxhq12@L', db="mydb",charset='utf8')
6  curs=conn.cursor()
7  sql="select * from member"
8  curs.execute(sql)
9  rows = curs.fetchall()
10
11 print(rows)
12 conn.close()
13
14 |

```

```

bugpy\launcher' '14819' '--' 'c:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts\linktest01.py'
(('1234', '1234', None, None, None, None, None, None),)
PS C:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts> c:: cd 'c:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts'; & 'python' 'c:\Users\이창수\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\p
ython\debugpy\launcher' '14830' '--' 'c:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts\linktest01.py'
(('1234', '1234', None, None, None, None, None, None),)
PS C:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts> c:: cd 'c:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts'; & 'python' 'c:\Users\이창수\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\p
ython\debugpy\launcher' '14845' '--' 'c:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts\linktest01.py'
(('1234', '1234', None, None, None, None, None, None),)
PS C:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts> c:: cd 'c:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts'; & 'python' 'c:\Users\이창수\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\p
ython\debugpy\launcher' '14858' '--' 'c:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts\linktest01.py'
(('1234', '1234', None, None, None, None, None, None),)
PS C:\Users\이창수\Documents\TASK01_INTFLOW\python_scripts>

```

PK 값 임의 설정 후 DB연동 테스트진행

1분마다 실시간 센서 값을 삽입하는 함수 / 개체의 실시간 행동데이터 삽입하는 함수

```
import threading
import pymysql

#60초간격으로 sql 문을 통한 호출
def time60s():
    threading.Timer(60,time60s).start()
    #pymysql패키지를 통한 db연동
    conn=pymysql.connect(host='localhost', user='root',password='rhtmxhq12@L', db="mydb",charset='utf8')
    curs=conn.cursor()

    #함수 1번 :센서값을 센서컬럼에 업데이트 / 삽입시 insert 테이블 into (컬럼) values();
    sql="update mydb.sensor set temperature=35 where sensorNo='1';"
    #함수 2번 : 개체의 정보를 1분마다 입력
    sql2="update camera set moves='moving' where foods is not null;"
    curs.execute(sql)
    curs.execute(sql2)
    conn.commit()
    conn.close()
    print('삽입완료')

time60s()
```

Sql : 함수 1번내용
Sql2 : 함수 2번내용

DB 연동 후, Sql문 호출을 통한 명령실행 후 연결종료

1분마다 실시간 센서 값을 삽입하는 함수 / 개체의 실시간 행동데이터 삽입하는 함수

1 • SELECT * FROM mydb.sensor;

sensorNo	temperature	humidity	wind	gas	building_buildingNo 1
1	NULL	NULL	NULL	NULL	1
2	12	5	3	2	2
3	43	5	3	2	3
4	45	NULL	NULL	NULL	2
NULL	NULL	NULL	NULL	NULL	NULL

함수1

1 • SELECT * FROM mydb.sensor;

sensorNo	temperature	humidity	wind	gas	building_buildingNo 1
1	35	NULL	NULL	NULL	1
2	12	5	3	2	2
3	43	5	3	2	3
4	45	NULL	NULL	NULL	2
NULL	NULL	NULL	NULL	NULL	NULL

1 • SELECT * FROM mydb.camera;

cameraNo	animal	states	foods	healthy	moves	water	building_buildingNo	building_member_mID
1	2	NULL	enough	NULL	NULL	NULL	1	1234
11	12	NULL	enough	NULL	NULL	NULL	1	1234
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

함수2

1 • SELECT * FROM mydb.camera;

cameraNo	animal	states	foods	healthy	moves	water	building_buildingNo	building_member_mID
1	2	NULL	enough	NULL	moving	NULL	1	1234
11	12	NULL	enough	NULL	moving	NULL	1	1234
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

실행 전

실행 후

온도가 가장 높은 건물에 있는 개체들의 평균 활동성을 출력하는 함수

```
import threading
import pymysql
#pymysql패키지를 통한 db연동
conn=pymysql.connect(host='localhost', user='root',password='rhtmxhq12@L', db="mydb",charset='utf8')
curs=conn.cursor()
#건물번호출력_빌딩테이블 중 일정 온도 이상인 건물번호만
sql="select building_buildingNo1 from sensor where temperature>=34"
curs.execute(sql)
conn.commit()
rows=curs.fetchall()
print(rows)
conn.close()
```

```
PS C:\Users\이창수\Documents\TASK01_INTFLOW\task02> C:; cd C:\Users\이창수\Documents\TASK01_INTFLOW\task02 ; & py
uncher' '5747' '--' 'c:\Users\이창수\Documents\TASK01_INTFLOW\task02\function03.py'
((1,), (2,))
PS C:\Users\이창수\Documents\TASK01_INTFLOW\task02> █
```

```
1 • SELECT * FROM mydb.sensor;
```

	sensorNo	temperature	humidity	wind	gas	building_buildingNo1
	1	35	NULL	NULL	NULL	1
	2	12	5	3	2	2
	3	43	5	3	2	3
	4	45	NULL	NULL		2
*	NULL	NULL	NULL	NULL	NULL	NULL

PK 값 임의 설정 후 DB연동 테스트진행

통신이상으로 특정 구역의 데이터가 삽입되고 있지 않았을 경우를 가정하여
10분 동안 행동데이터의 삽입이 없는 구역을 가지고 있는 건물을 출력하는 함수

```
import threading
import pymysql
#600초간격으로 sql 문을 통한 호출
def time60s():
    threading.Timer(600,time60s).start()
    #pymysql패키지를 통한 db연동
    conn=pymysql.connect(host='localhost', user='root',password='rhtmxhq12@L', db="mydb",charset='utf8')
    curs=conn.cursor()
    #건물번호 출력_조건 값들이 null 일 경우
    sql="SELECT building_buildingNo1 from sensor where humidity is null and wind is null and gas is null;"
    curs.execute(sql)
    conn.commit()
    rows=curs.fetchall()
    print("건물번호 : ",rows)
    conn.close()
time60s()
```

1 • SELECT * from sensor;

2

	sensorNo	temperature	humidity	wind	gas	building_buildingNo1
1	1	35	NULL	NULL	NULL	1
2	2	12	5	3	2	2
3	3	43	5	3	2	3
4	4	45	NULL	NULL	NULL	2
5	NULL	NULL	NULL	NULL	NULL	NULL

1 • SELECT building_buildingNo1 from sensor where humidity is null and wind is null and gas is null;

2

	building_buildingNo1
1	1
2	2

PK 값 임의 설정 후 DB연동 테스트진행

10분 내에 이상행위를 한 개체가 있는 건물을 출력하는 함수

```

> 이창수 > Documents > TASK01_INTFLOW > task02 > function05.py > time60s
import threading
import pymysql
import datetime
#60초간격으로 sql 문을 통한 호출
def time60s():
    threading.Timer(60,time60s).start()
    #pymysql패키지를 통한 db연동
    conn=pymysql.connect(host='localhost', user='root',password='rhtmxhq12@L', db="mydb",charset='utf8')
    curs=conn.cursor()
    #sql문을 통한 조건입력 후 10분간 움직임 없는 개체의 건물번호 출력
    sql="SELECT building_buildingNo FROM camera where date_sub(CURDATE(),INTERVAL -10 minute) and moves ='moving';"
    curs.execute(sql)
    conn.commit()
    rows=curs.fetchall()
    print(rows)

    conn.close()

time60s()

```

```

PS C:\Users\이창수\Documents\TASK01_INTFLOW\task02> cd 'c:\Users\이창수\Documents\TASK01_INTFLOW\task02' & python 'c:\Users\이창수\Documents\TASK01_INTFLOW\task02\function05.py'
((1,),(1,))
PS C:\Users\이창수\Documents\TASK01_INTFLOW\task02> cd 'c:\Users\이창수\Documents\TASK01_INTFLOW\task02' & 'python' 'c:\Users\이창수\Documents\TASK01_INTFLOW\task02\function05.py'
((1,),(1,))
PS C:\Users\이창수\Documents\TASK01_INTFLOW\task02>

```

1 • SELECT * FROM mydb.camera;

cameraNo	animal	states	foods	healthy	moves	water	building_buildingNo	building_member_mID
1	NULL	NULL	enough	NULL	moving	NULL	1	1234
11	NULL	NULL	enough	NULL	moving	NULL	1	1234
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

1 • SELECT building_buildingNo FROM camera where moves ='moving';

building_buildingNo
1
1

PK 값 임의 설정 후 DB연동 테스트진행