

## 데이터 입력/출력



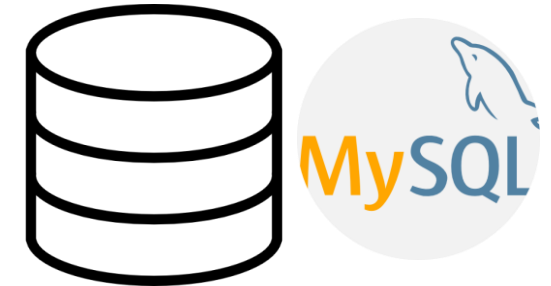
## 라즈베리파이

### 제어 컨트롤러



### 데이터저장

데이터저장



# 실습 과제

1) 라즈베리파이 라즈비안 설치

라즈베리파이

- IP : 166.104.81.31
- DNS : hyu-iot.hanyang.ac.kr

2) 도커 설치

3) 도커를 이용하여 Node-red, Mysql 설치

## A팀

- Node-red (name : nodered\_1881, port : 1881)
- MySQL (name : mysql\_3307, port : 3307, , PW: iot1004)

## B팀

- Node-red (name : nodered\_1882, port : 1882)
- MySQL (name : mysql\_3308, port : 3308, PW: iot1004)

# Container Application in Docker

<https://www.docker.com/products/docker-hub/>



## Node-Red 서버

```
$ docker run -itd -p 1880:1880 -v /home/iot/nodered_1880:/data --name  
nodered_1880 nodered/node-red:latest
```



## Mysql 서버

<https://dev.mysql.com/downloads/installer/>

```
$ docker run -itd -p 3306:3306 --name mysql_3306 -e  
MYSQL_ROOT_PASSWORD=iot1004 mysql:latest
```



mongoDB®

## mongoDB 서버

```
$ docker run -itd -p 27017:27017 --name mongodb mongodb/mongodb-  
community-server:latest
```



## MQTT 서버

```
$ docker run -it -p 1883:1883 -p 9001:9001 -v  
mosquitto.conf:/mosquitto/config/mosquitto.conf eclipse-mosquitto
```

# 실습 과제



IoT 보드 (솔파)

- Broker : test.mosquitto.org
- Mac address : D4:8A:FC:B5:34:64
- Topic out : i2r/a@gmail.com/out
- Topic in : i2r/a@gmail.com/in



IoT 보드 (주찬)

- Broker : test.mosquitto.org
- Mac address : D4:8A:FC:B7:1D:84
- Topic out : i2r/b@gmail.com/out
- Topic in : i2r/b@gmail.com/in



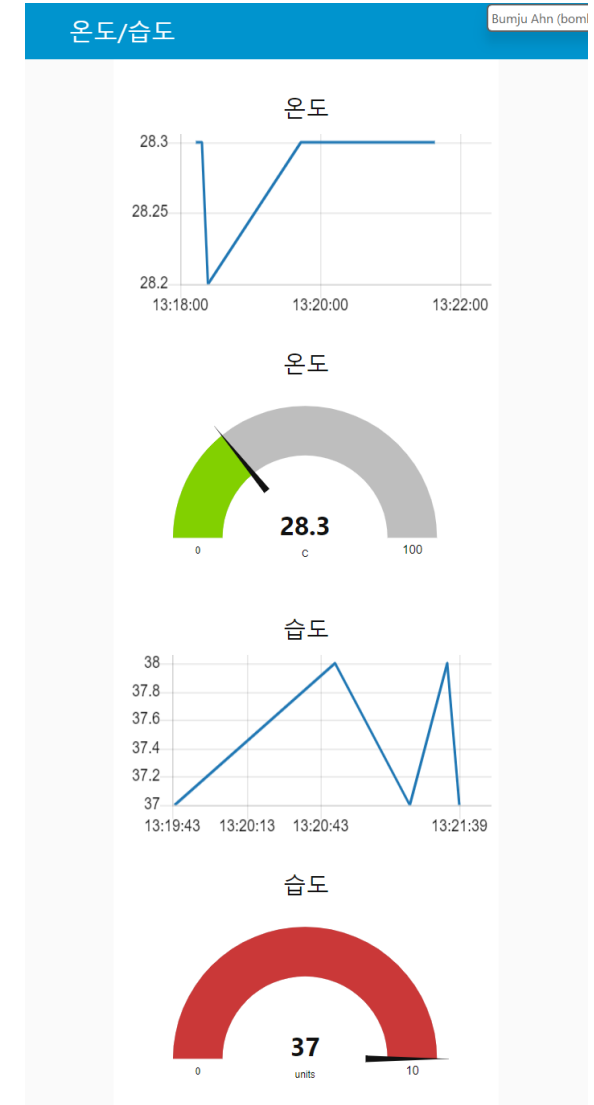
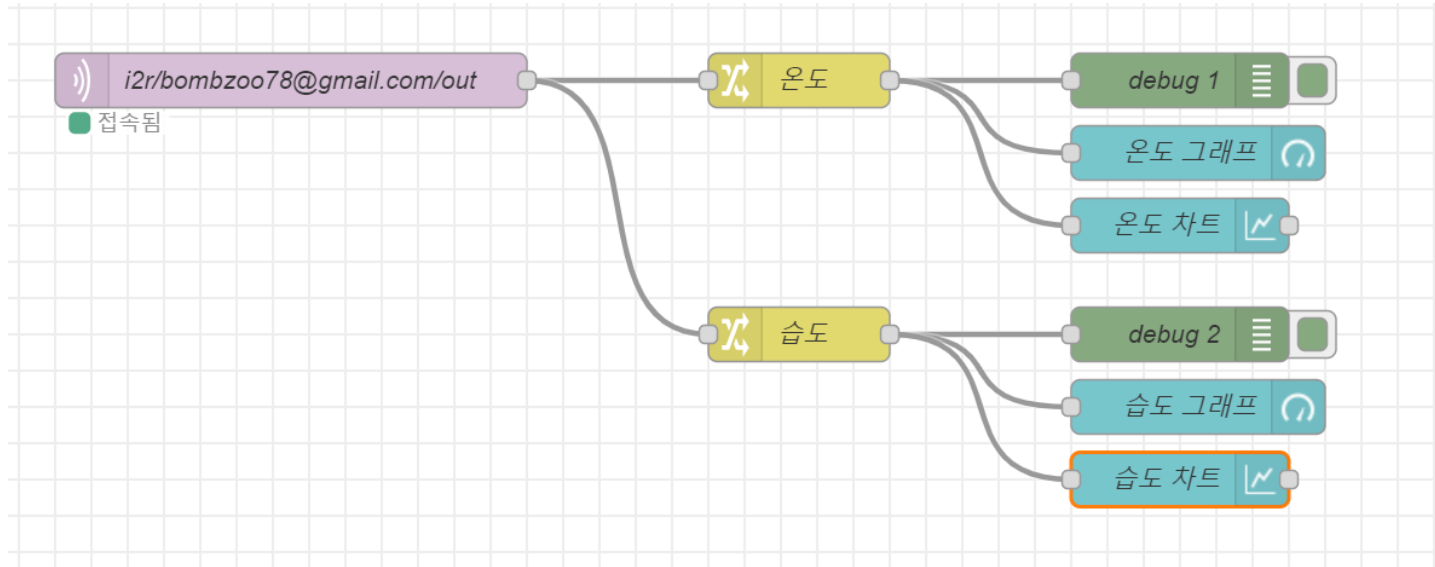
IoT 보드 (범주)

- Broker : ai.doowon.ac.kr
- Mac address : 08:B6:1F:75:11:FE
- Topic out : i2r/bombzoo78@gmail.com/out
- Topic in : i2r/bombzoo78@gmail.com/in

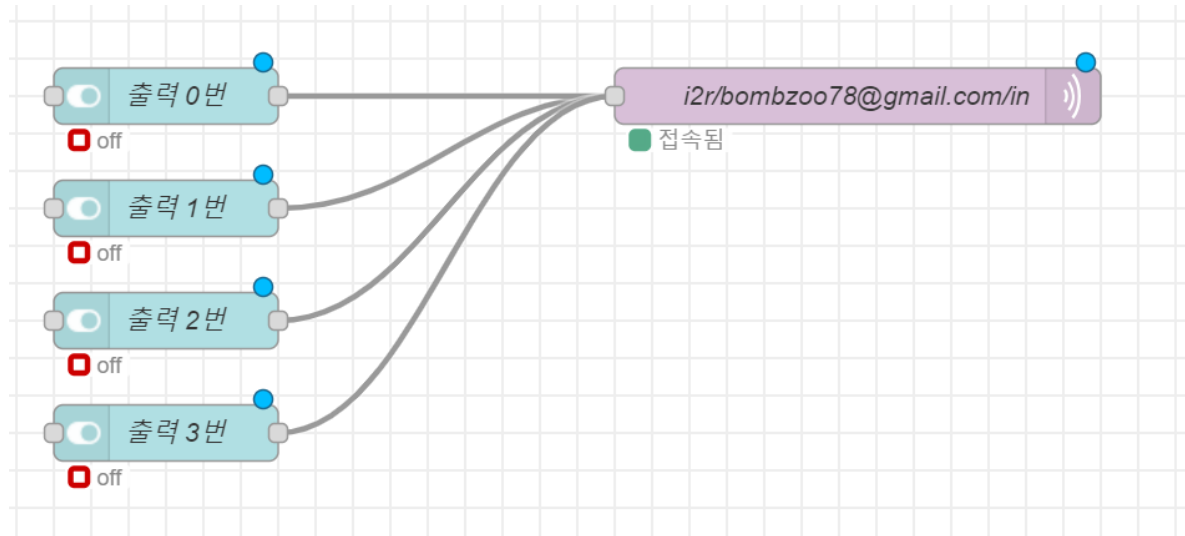
## 해결 과제 (노드레드)

- 1) IoT보드의 온도, 습도값을 그래프와 차트로 웹에 표시하여라
- 2) IoT보드 각각의 릴레이 출력을 제어하는 (ON/OFF) 스위치 dashboard를 만들어라
- 3) 습도 60이상시 2번 릴레이를 ON하여라
- 4) 온도 25이상시 1번 릴레이를 ON하여라.
- 5) 습도 60이상시 IOT보드(범주)의 3번 릴레이를 ON하여라
- 6) 정해진 시간에 3번 릴레이를 ON하고 1분뒤 OFF 하여라 (스케줄러)
- 7) IoT보드 출력의 상태를 표시하는 LED표시등을 만들어라

# 1) IoT보드의 온도, 습도값을 그래프와 차트로 웹에 표시하여라



## 2) IoT보드 각각의 릴레이 출력을 제어하는 (ON/OFF) 스위치 dashboard를 만들어라



### ≡ 스위치

#### 스위치

출력 1번



출력 2번



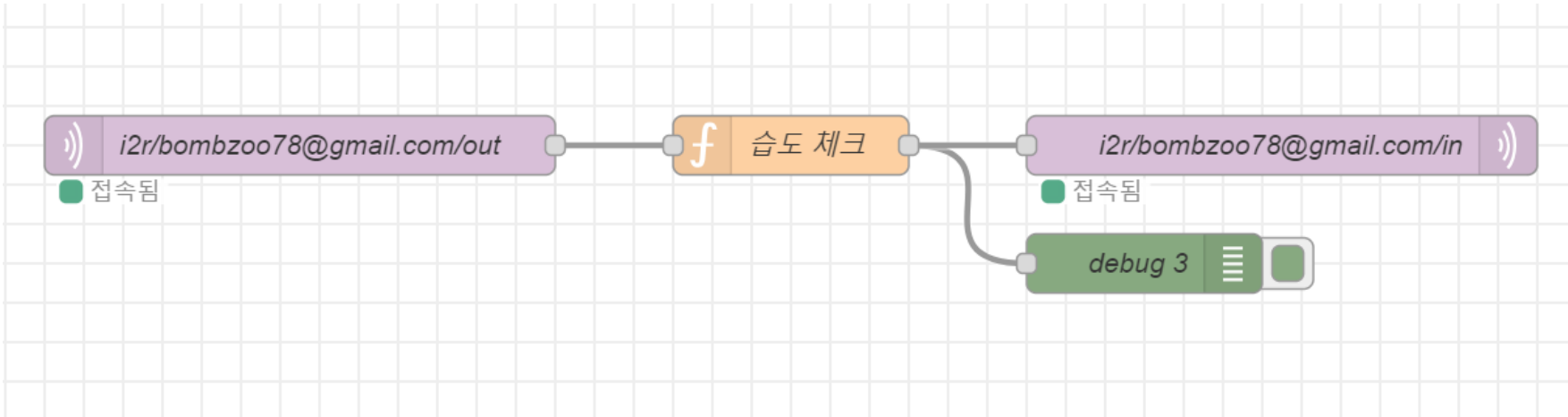
출력 3번



출력 2번



### 3) 습도 60이상시 2번 릴레이를 ON하여라



```
1  if (msg.payload.humi >= 60){
2      msg.payload = {
3          "mac": "08:B6:1F:75:11:FE",
4          "order": 2,
5          "no": 1,
6          "value": true
7      }
8      return msg;
9  }
10 else {
11     msg.payload = {
12         "mac": "08:B6:1F:75:11:FE",
13         "order": 2,
14         "no": 1,
15         "value": false
16     }
17     return msg;
18 }
```

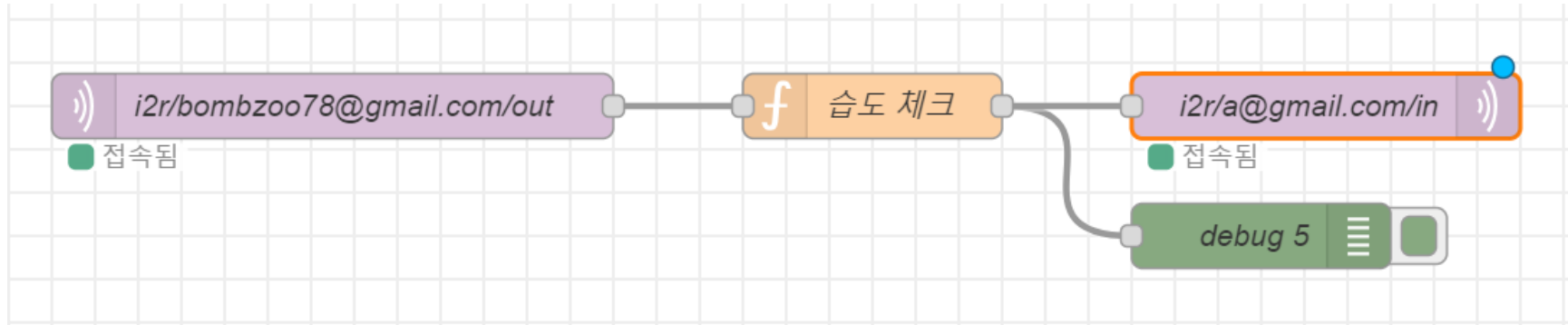


4) 온도 25이상시 1번 릴레이를 ON하여라.



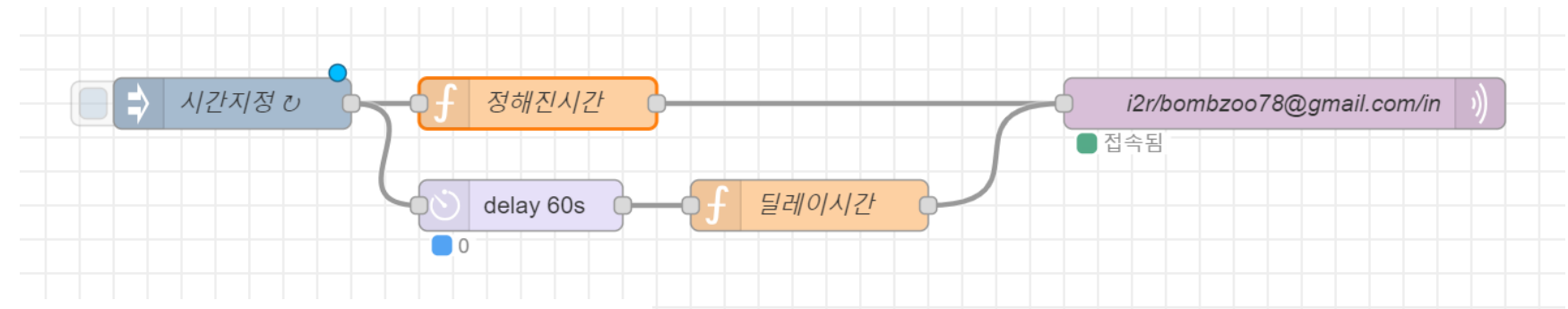
```
1  if (msg.payload.temp >= 25){
2      msg.payload = {
3          "mac": "08:B6:1F:75:11:FE",
4          "order": 2,
5          "no": 1,
6          "value": true
7      }
8      return msg;
9  }
10 else {
11     msg.payload = {
12         "mac": "08:B6:1F:75:11:FE",
13         "order": 2,
14         "no": 1,
15         "value": false
16     }
17     return msg;
18 }
```

## 5) 습도 60이상시 IOT보드(범주)의 3번 릴레이를 ON하여라



```
1  if (msg.payload.humi >= 60){
2      msg.payload = {
3          "mac": "D4:8A:FC:B5:34:64",
4          "order": 2,
5          "no": 1,
6          "value": true
7      }
8      return msg;
9  }
10 else {
11     msg.payload = {
12         "mac": "D4:8A:FC:B5:34:64",
13         "order": 2,
14         "no": 1,
15         "value": false
16     }
17     return msg;
18 }
```

6) 정해진 시간에 3번 릴레이를 ON하고 1분뒤 OFF 하여라 (스케줄러)



이름 시간지정

msg.payload

=

true

x

msg.topic

=

a\_z

x

+ 추가

inject now

```
1  if (msg.payload){
2      msg.payload = {
3          "mac": "08:B6:1F:75:11:FE",
4          "order": 2,
5          "no": 1,
6          "value": true
7      }
8      return msg;
9  }
10
```

☐ Node-RED시작의 0.1 초 후, 아래를 시행

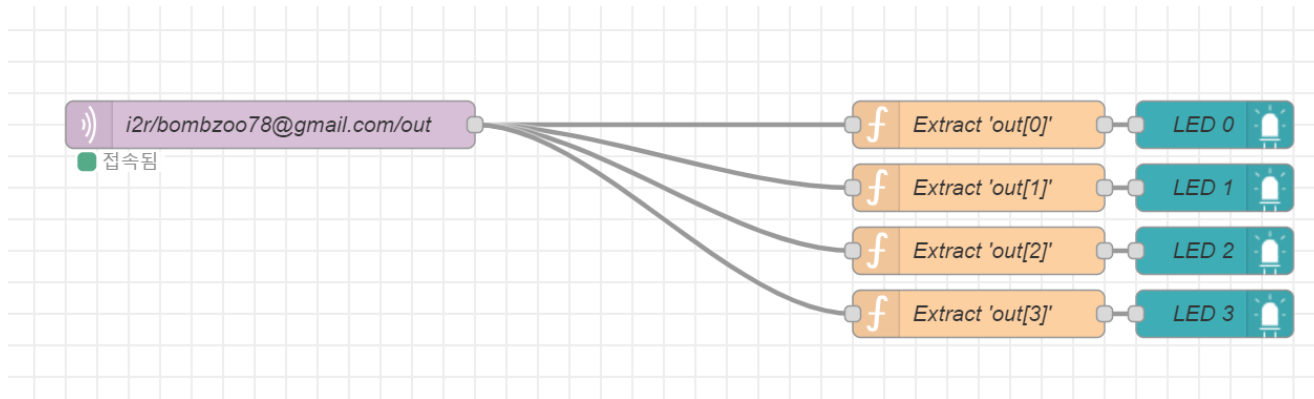
반복 지정한 일시

시각 05:35

요일	<input checked="" type="checkbox"/> 월요일	<input checked="" type="checkbox"/> 화요일	<input checked="" type="checkbox"/> 수요일
	<input checked="" type="checkbox"/> 목요일	<input checked="" type="checkbox"/> 금요일	<input checked="" type="checkbox"/> 토요일
	<input checked="" type="checkbox"/> 일요일		

## 7) IoT보드 출력의 상태를 표시하는 LED표시등을 만들어라

node-red-contrib-ui-led 설치 필요



```
1  var payload = msg.payload;
2  var inValue = payload.out[0];
3
4  msg.payload = inValue === 1;
5
6  return msg;
```

### ≡ 스위치

스위치



## 데이터 입력/출력



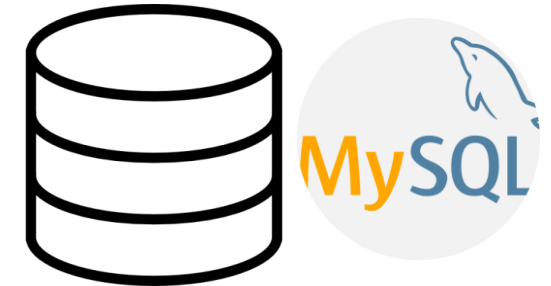
## 라즈베리파이

### 제어 컨트롤러



### 데이터저장

데이터저장



## 해결 과제 (MySQL)

- 1) MySQL 컨테이너 접속하여 DB (iot)와 테이블(status)을 생성하라 (콘솔, SQL문 사용)
- 2) status 테이블에 IoT의 온습도, 입/출력 정보를 저장할수 있는 스키마를 생성하여라
- 3) 노드레드를 IoT보드 상태 정보를 실시간으로 DB에 저장하여라

## MySQL 컨테이너 접속

컨테이너  
라지미안

```
iot@HYU-IOT:~ $ docker ps -a
CONTAINER ID   IMAGE             COMMAND                  CREATED        STATUS
NAMES
0734fcb762e0   mysql:latest      "docker-entrypoint.s..." 10 hours ago   Up 7 hours
mysql_3306
7f0c0f92c3f4   nodered/node-red  "./entrypoint.sh"        10 hours ago   Up 7 hours (healt
nodered_1880
iot@HYU-IOT:~ $ docker exec -it mysql_3306 /bin/bash
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 151
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

mysql

# IOT DB 만들기



# IOT DB 만들기

## IOT 보드 실시간 정보

# 데이터 보기

SELECT \* FROM status;

"type":3,  
"email": "bombzoo78@gmail.com",  
"mac": "EC:64:C9:43:E8:B8",  
"temp":28.4,  
"humi":38,  
"in":[0,0,0,0],  
"out":[0,0,0,0]



"timestamp " :1727170478,  
"email": "bombzoo78@gmail.com",  
"mac": "EC:64:C9:43:E8:B8",  
"temp":28.4,  
"humi":38,  
"in":[0,0,0,0],  
"out":[0,0,0,0]

mysql> select \* from status;

timestamp	email	mac	temp	humi	in00	in01	in02	in03	out00	out01	out02	out03
1727174982	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	40	0	0	0	0	0	0	0	0
1727174984	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	39	0	0	0	0	0	0	0	0
1727174990	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.1	39	0	0	0	0	0	0	0	0
1727175003	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	39	0	0	0	0	0	0	0	0
1727175006	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.2	40	0	0	0	0	0	0	0	0
1727175015	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.3	40	0	0	0	0	0	0	0	0
1727175020	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.8	99	0	0	0	0	0	0	0	0
1727175025	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	51	0	0	0	0	0	0	1	0
1727175030	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	47	0	0	0	0	0	0	1	0
1727175035	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	45	0	0	0	0	0	0	0	0
1727175040	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	44	0	0	0	0	0	0	0	0
1727175047	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	43	0	0	0	0	0	0	0	0
1727175050	bombzoo78@gmail.com	08:B6:1F:75:11:FE	29.4	42	0	0	0	0	0	0	0	0

# [ IOT DB 만들기 ]

iot DB



## # 데이터베이스 만들기

```
CREATE DATABASE iot;
```

## # 데이터베이스 사용하기

```
USE iot;
```

status TABLE

-	-	-
-	-	-
-	-	-
-	-	-

## # 테이블 보기

```
SHOW TABLES;
```

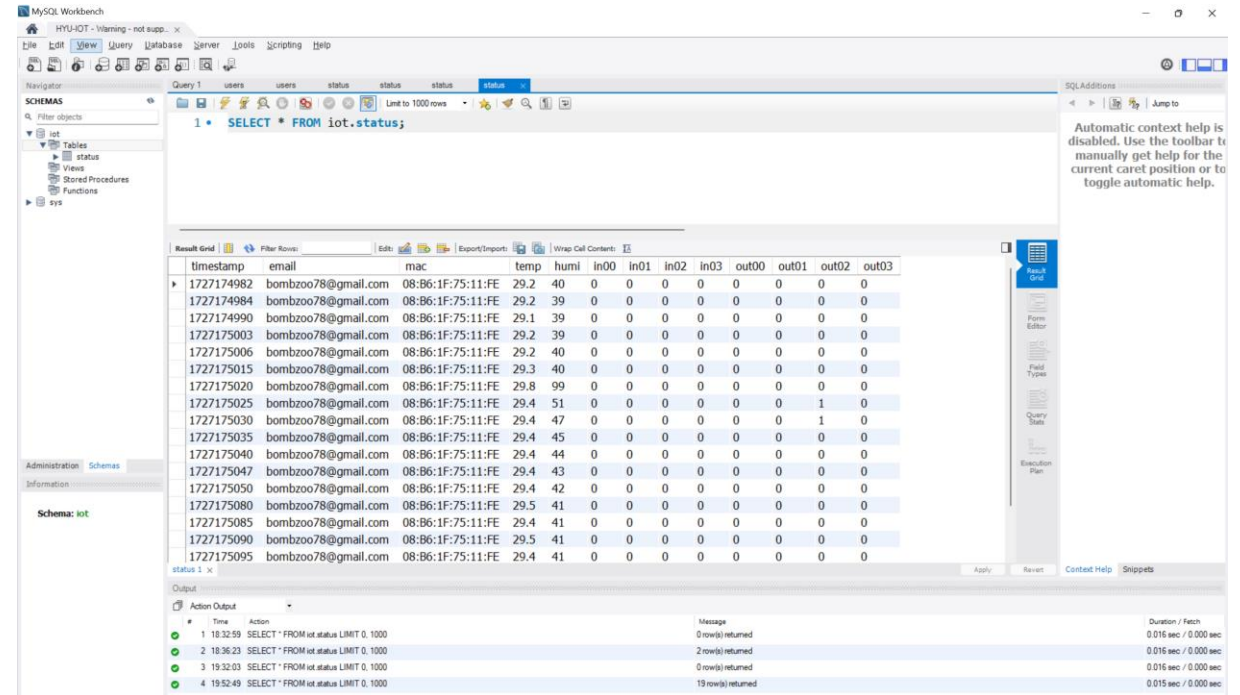
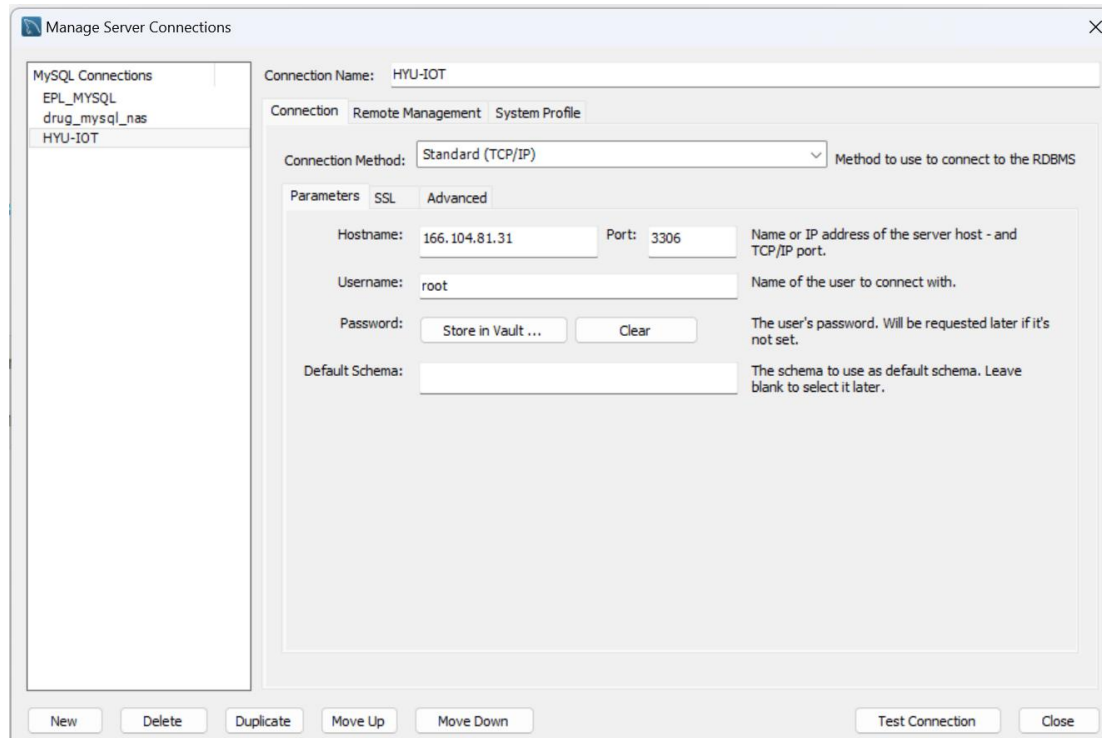
## # 테이블 만들기

```
CREATE TABLE status (  
    timestamp VARCHAR(45) PRIMARY KEY,  
    email VARCHAR(45) NOT NULL,  
    mac VARCHAR(45) NOT NULL,  
    temp FLOAT NOT NULL,  
    humi FLOAT NOT NULL,  
    in00 INT NOT NULL,  
    in01 INT NOT NULL,  
    in02 INT NOT NULL,  
    in03 INT NOT NULL,  
    out00 INT NOT NULL,  
    out01 INT NOT NULL,  
    out02 INT NOT NULL,  
    out03 INT NOT NULL  
);
```

# MySQL Workbench

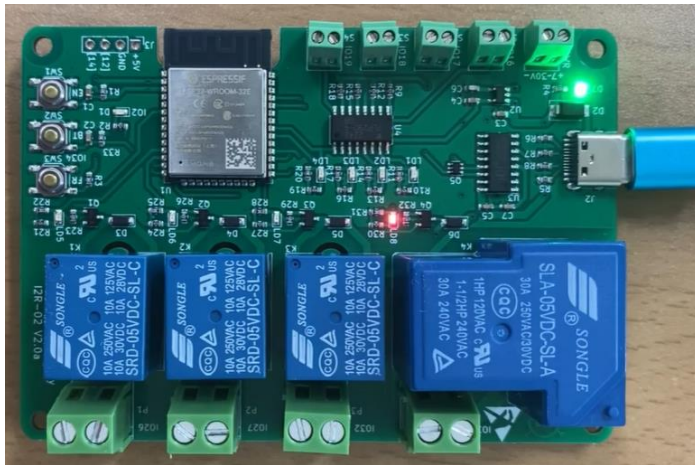
# 홈페이지

<https://dev.mysql.com/downloads/workbench/>



# 노드레드 - MySQL 연결

데이터 입력/출력



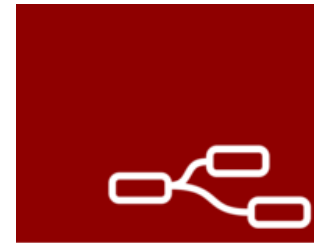
양방향 통신(IOT)



MQTT

라즈베리파이

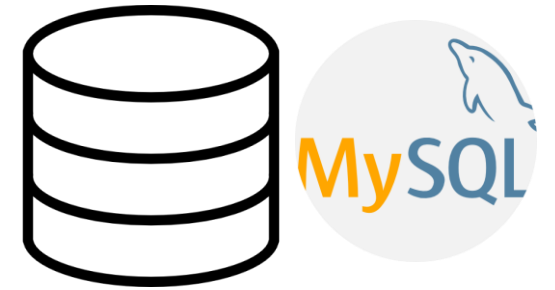
제어 컨트롤러



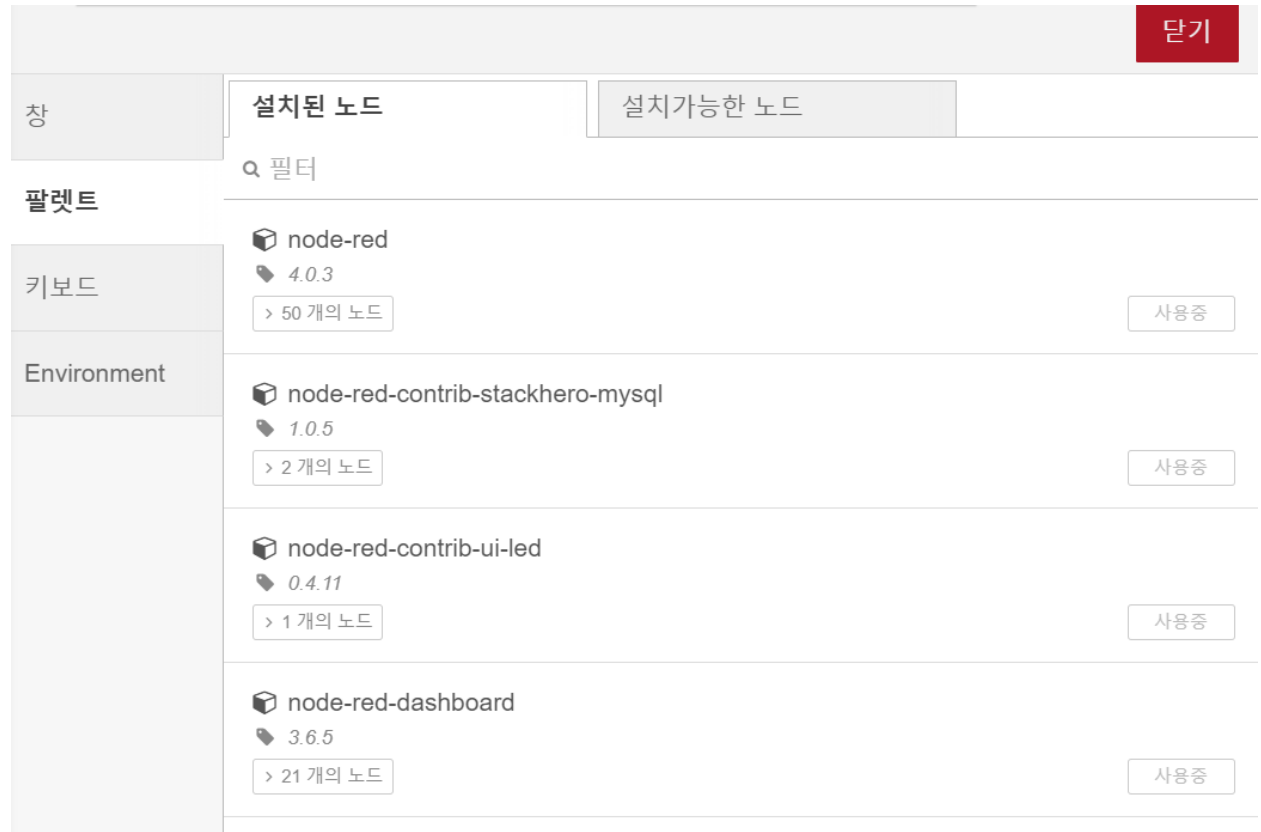
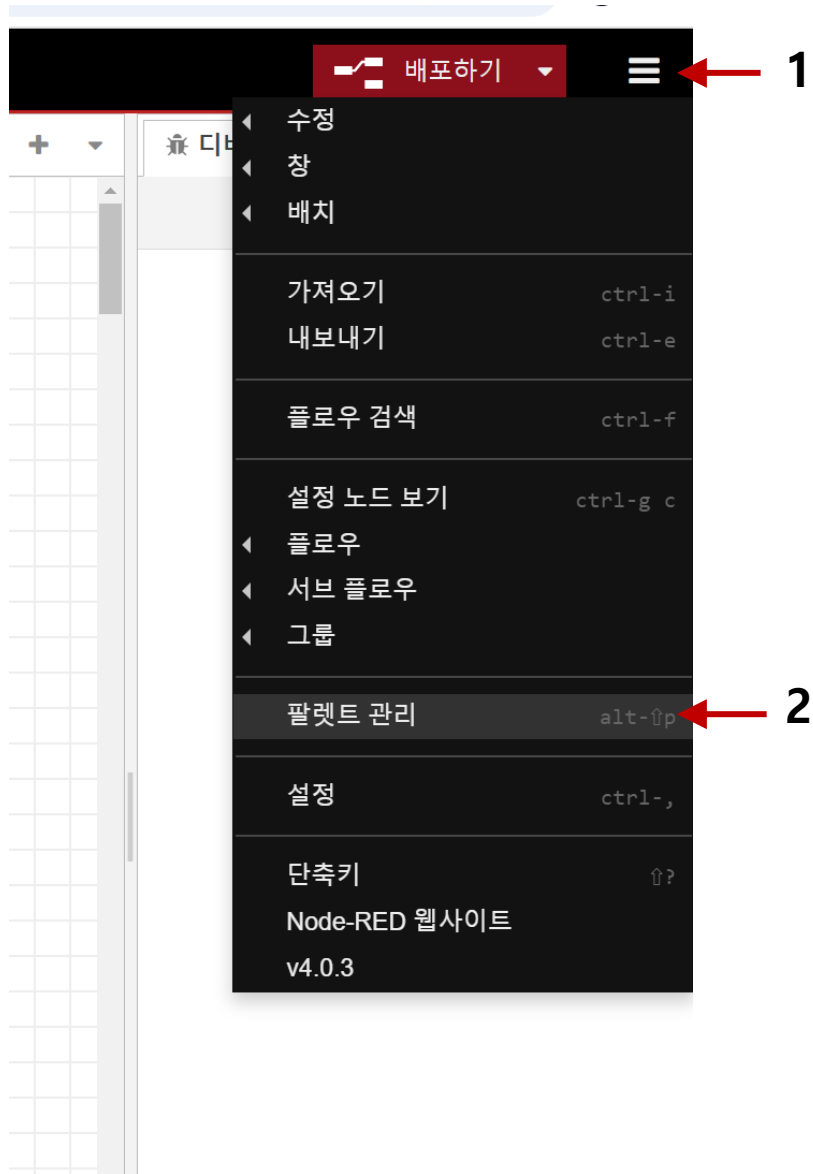
Node-RED

데이터저장

데이터저장



## Node-Red 팔레트 설치

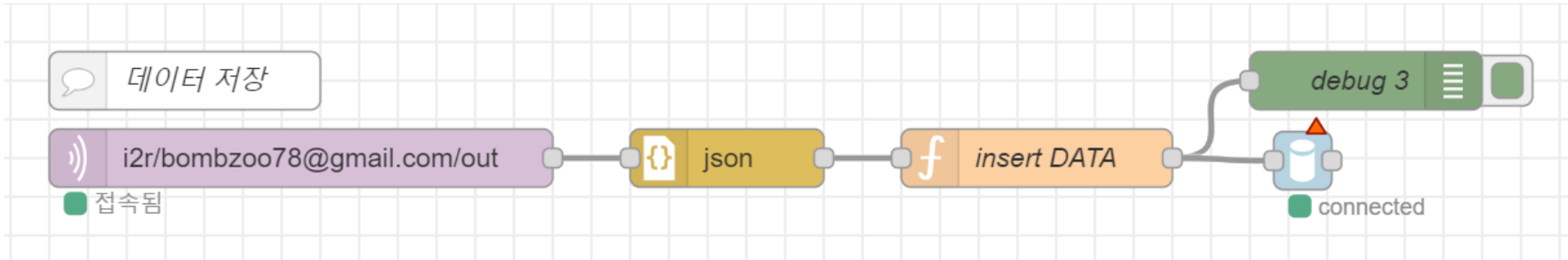


1. node-red-dashboard

2. node-red-contrib-ui-led

3. node-red-contrib-stackhero-mysql

# 노드레드 - MySQL 연결

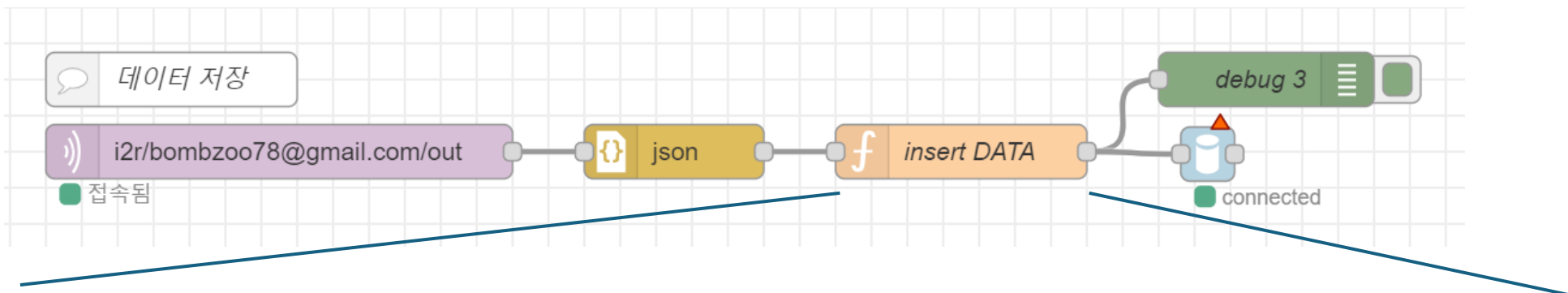


## IOT 보드 실시간 정보

"type":3,  
"email": "bombzoo78@gmail.com",  
"mac": "EC:64:C9:43:E8:B8",  
"temp":28.4,  
"humi":38,  
"in":[0,0,0,0],  
"out":[0,0,0,0]



"timestamp " :1727170478,  
"email": "bombzoo78@gmail.com",  
"mac": "EC:64:C9:43:E8:B8",  
"temp":28.4,  
"humi":38,  
"in":[0,0,0,0],  
"out":[0,0,0,0]



// 메시지에서부터 JSON 데이터를 추출합니다.

```
const jsonData = msg.payload;
```

```
// const timestamp = new Date().toLocaleString('ko-KR', { timeZone: 'Asia/Seoul', hour12: false });
```

```
const timestamp = Math.floor(new Date().getTime() / 1000);
```

```
const email = jsonData.email;
```

```
const mac = jsonData.mac;
```

```
const temp = jsonData.temp;
```

```
const humi = jsonData.humi;
```

```
const in00 = jsonData.in[0];
```

```
const in01 = jsonData.in[1];
```

```
const in02 = jsonData.in[2];
```

```
const in03 = jsonData.in[3];
```

```
const out00 = jsonData.out[0];
```

```
const out01 = jsonData.out[1];
```

```
const out02 = jsonData.out[2];
```

```
const out03 = jsonData.out[3];
```

// MySQL 쿼리를 생성합니다. 문자열 값은 작은따옴표로 감싸야 하고, 숫자 값은 그대로 넣습니다.

```
let sql = `INSERT INTO iot.status (timestamp, email, mac, temp, humi, in00, in01, in02, in03, out00, out01, out02, out03)
VALUES ('${timestamp}', '${email}', '${mac}', ${temp}, ${humi}, ${in00}, ${in01}, ${in02}, ${in03}, ${out00}, ${out01},
${out02}, ${out03})`;
```

// 메시지에 SQL 쿼리 추가

```
msg.topic = sql;
```

// 메시지 전달

```
return msg;
```

