

# Node.js MySQL 연결

박준서

# 목차

- MySQL 데이터 생성
- Javascript 파일 생성
- 실행 결과
- 추가

# MySQL 데이터 생성

```
mysql> show tables;
+-----+
| Tables_in_seminar |
+-----+
| member             |
+-----+
1 row in set (0.01 sec)

mysql> desc member;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name  | varchar(10)   | NO   | PRI | NULL    |       |
| grade | int           | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

```
mysql> select * from member;
+-----+-----+
| name      | grade |
+-----+-----+
| 박준서    | 4     |
| 손중영    | 4     |
| 이성복    | 4     |
+-----+-----+
3 rows in set (0.00 sec)
```

- 테이블을 생성하고, 데이터 삽입
- select로 데이터가 잘 들어갔는지 확인
- utf-8을 사용할 경우 한글은 3바이트

# Javascript 파일 생성

```
const mysql = require('mysql');
require('dotenv').config()

const conn = mysql.createConnection({
  host: 'localhost',
  user: process.env.USER,
  password: process.env.PASSWORD,
  database: 'seminar'
});

conn.connect();

conn.query('select * from member;', (error, rows, cols) => {
  console.log(rows);
});

conn.end();
```

- Javascript의 mysql 모듈을 이용
- Host 와 user name, password, database 이름을 입력
- MySQL에 연결 후 쿼리 실행

# 실행 결과

```
aaa@ubuntu:~/js-mysql$ sudo node index.js
[
  RowDataPacket { name: '박준서', grade: 4 },
  RowDataPacket { name: '손중영', grade: 4 },
  RowDataPacket { name: '이성복', grade: 4 }
]
```

- MySQL에서 생성한 테이블의 정보를 불러올 수 있었음
- Row는 dictionary 형태
- 따라서 쿼리를 조작하지 않아도 원하는 정보 추출 가능

# 추가

```
conn.query('insert into member (name, grade) values ("준서", 4);');  
conn.query('select * from member;', (error, rows, cols) => {  
  console.log(rows);  
});
```

```
aaa@ubuntu:~/js-mysql$ sudo node index.js  
[  
  RowDataPacket { name: '박준서', grade: 4 },  
  RowDataPacket { name: '손종영', grade: 4 },  
  RowDataPacket { name: '이성복', grade: 4 },  
  RowDataPacket { name: '준서', grade: 4 }  
]
```

- 당연히 추가, 수정 등의 쿼리도 가능함
- 테이블 명, 데이터 등을 변수로 쓰면 사용자 입력도 가능
- mysql2 모듈이 더 좋다고 함

QnA