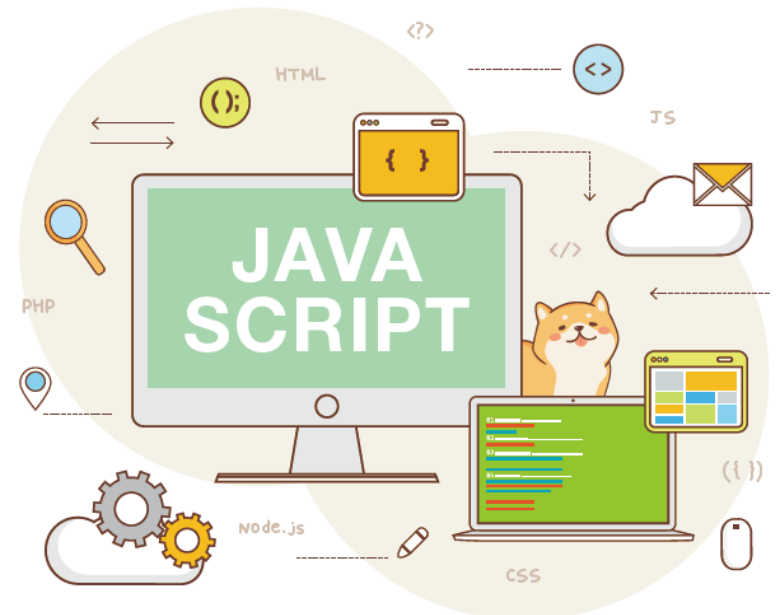


서버 통신 자바스크립트 (Chapter 11)

스마트헬스 케어 – 11주차

학습 내용

1. REST API 만들기
2. MySQL 설치
3. 기본 쿼리 실습

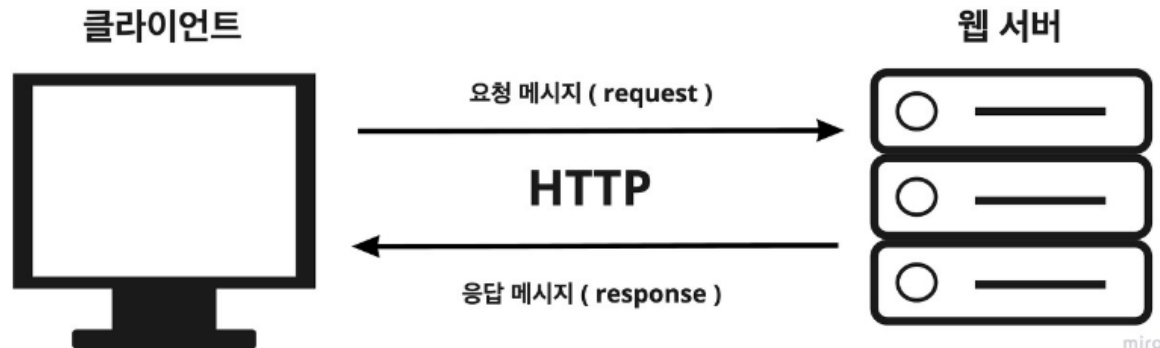


이론 및 예제 실습

1_REST API 개념

1. REST API : REST 기반으로 API를 구현하는 것

- HTTP 표준을 기반으로 애플리케이션에서 호출할 수 있는 인터페이스를 만든 것
- JSON이나 XML 응답 형식이 대부분
- REST(Representational State Transfer) : 네트워크상에서 Client와 Server 사이에서 HTTP로 통신하는 것
- API(Application Programming Interface) : 클라이언트가 리소스를 요청할 수 있도록 서버측에서 제공된 인터페이스



1_REST API 호출 (웹페이지에서 호출 - AJAX)

```
<script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
```

```
$.ajax({
```

호출 Method

```
type : "GET",
```

서버 호출 URL

```
url : "http://test.soxcorp.co.kr:31880/testapi",
```

서버로 전달할 데이터

```
data : { name: "soxcorp", age: 4 },
```

받는 데이터 형식

```
dataType : "text",
```

서버로 부터 데이터를
받으면 호출되는 함수
(이벤트 핸들러)

```
success : function (data) {  
    document.write(data);  
    console.log(JSON.parse(data));  
},
```

서버가 보낸 데이터는
data에 담긴다

통신 에러시 호출되는
함수
(이벤트 핸들러)

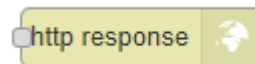
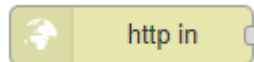
```
error : function () {  
    alert("통신실패!!!!");  
}
```

```
});
```

1_REST API 만들기

생성 프로세스

1. 'http in' 노드로 호출할 인터페이스 정의 (호출방식, 호출 이름)
2. 함수 노드로 클라이언트에게 전송할 데이터 (JSON) 가공
3. 'http response' 노드로 2번에서 만든 데이터를 내보냄



- 접속 URL을 설정함
 - 접속 주소 = Node_red 접속URL + URL
- ex) `http://127.0.0.1:1880/mytest`

A configuration panel for the 'http in' node. It has a title bar with a gear icon and two document icons. Below the title bar, there are three rows: '메소드' (Method) with a dropdown menu showing 'GET', 'URL' with a text input field containing '/mytest', and '이름' (Name) with a text input field containing '이름'.

- 전달할 데이터 생성

```
var value = 30
msg.payload = {
  titleCaption: "" + value ,
  titleColor: '#ff0000',
  fontSize: '50px'
}
return msg;
```

* 전송할 데이터는 msg.payload에 담는다.

1_REST API 만들기

REST API 만들기



속성

메소드: GET

URL: /mytest

이름: 이름

- msg.payload에 클라이언트가 보낸 요청 파라미터 데이터가 있음

속성

이름: 이름

Setup On Start 코드 On Stop

```
1 var param = msg.payload;
2
3 console.log(param);
4
5 msg.payload = {
6   test :{
7     node : "test"
8   },
9   obj : param
10 };
11 return msg;
```

확인 → <http://127.0.0.1:1880/mytest>

1_REST API 호출 (클라이언트 AJAX)

```
<script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
```

```
<script>
    $(document).ready(function () {
        myfunc();
    });

    function myfunc() {
        $.ajax({
            type : "GET",
            url  : "http://127.0.0.1:1880/mytest",
            data : { name: "홍길동", age: 24 },
            dataType : "json",
            success : function (data) {
                console.log(data);
            },
            error : function () {
                alert("통신실패!!!!");
            }
        });
    }
</script>
```


1_REST API -function 노드

function 노드 구성의 공식

1. 클라이언트가 보낸 parameter 데이터 받기
2. 클라이언트에 응답으로 보낼 데이터 만들기
3. 보낼 데이터를 msg.payload에 입력하기

```
let param = msg.payload;    → 클라이언트가 보낸 파라미터 데이터  
console.log(param);
```

```
let senddata = {            → 응답으로 보낼 데이터 생성  
  test :{ node : "test" }  
};
```

```
msg.payload = senddata;    → msg.payload에 입력  
return msg;
```

1_REST API 만들기

REST API 만들기

{ msg : "24세 홍길동님 환영합니다." } 로 보내내는 function으로 수정해 보시오



속성

메소드: GET

URL: /mytest

이름: 이름

속성

이름: 이름

Setup On Start 코드 On Stop

```
1 let param = msg.payload;
2 console.log(param);
3
4 let senddata = {
5   | msg: `${param.age}세 ${param.name}님 환영합니다.`
6 };
7
8 msg.payload = senddata;
9 return msg;
```

2_Mysql 설치


1. mysql 설치

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

- 버전은 5.7.xx 로 선택

MySQL Product Archives

MySQL Installer (Archived Versions)

 Please note that these are old versions. New releases will have recent bug fixes and features!
To download the latest release of MySQL Installer, please visit MySQL Downloads.

Product Version:

Operating System:

Windows (x86, 32-bit), MSI Installer

Apr 6, 2022

(mysql-installer-web-community-5.7.38.0.msi)

Windows (x86, 32-bit), MSI Installer

Apr 6, 2022

(mysql-installer-community-5.7.38.0.msi)

 We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.


2. workbanch 설치

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

- 버전은 6.3.xx 로 선택

MySQL Product Archives

MySQL Workbench (Archived Versions)

 Please note that these are old versions. New releases will have recent bug fixes and features!
To download the latest release of MySQL Workbench, please visit MySQL Downloads.

Product Version:

Operating System:

Windows (x86, 32-bit), MSI Installer

Feb 15, 2013

(mysql-workbench-gpl-5.2.47-win32.msi)

Windows (x86, 32-bit), ZIP Archive

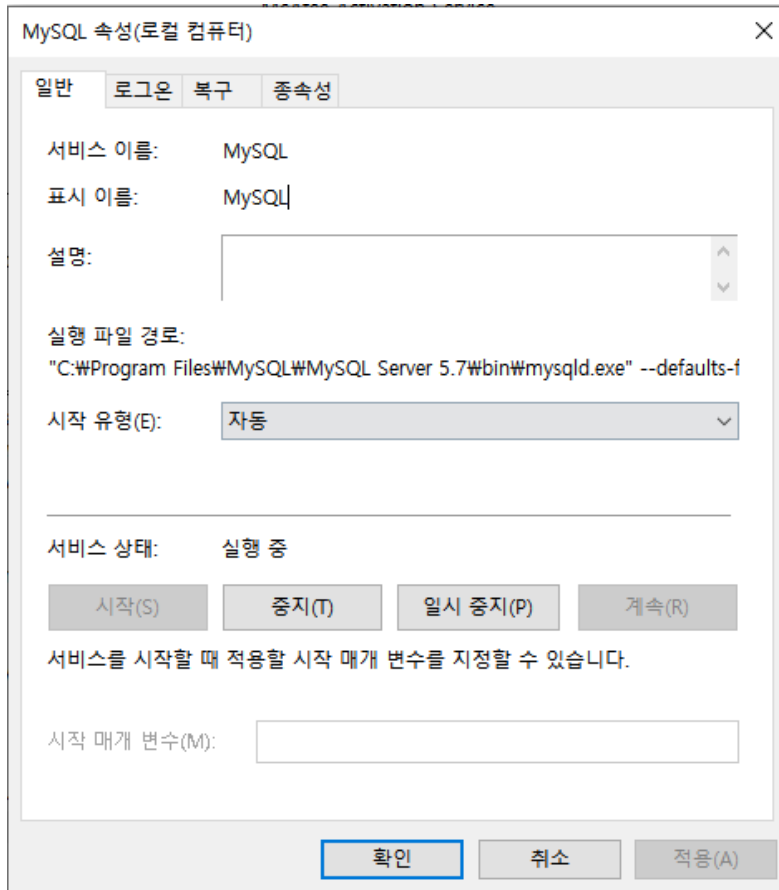
Feb 15, 2013

(mysql-workbench-gpl-5.2.47-win32-noinstall.zip)

 We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

2_Mysql 설치

mysql 실행 : [제어판] - [관리도구] - [서비스]에서 실행



2_Mysql 설치

사용자 만들기

1. 커맨드 창 띄우기

2. mysql 실행

```
cd C:\Program Files\MySQL\MySQL Server 5.7\bin
```

```
mysql -u root -p
```

[암호 입력]

3. 유저 생성

```
use mysql
```

```
CREATE USER 'user1'@'localhost' IDENTIFIED BY 'user123';
```

```
CREATE USER 'user1'@'%' IDENTIFIED BY 'user123';
```

```
grant all privileges on *.* to user1@'%' identified by 'user123';
```

```
grant all privileges on *.* to user1@'localhost' identified by 'user123';
```

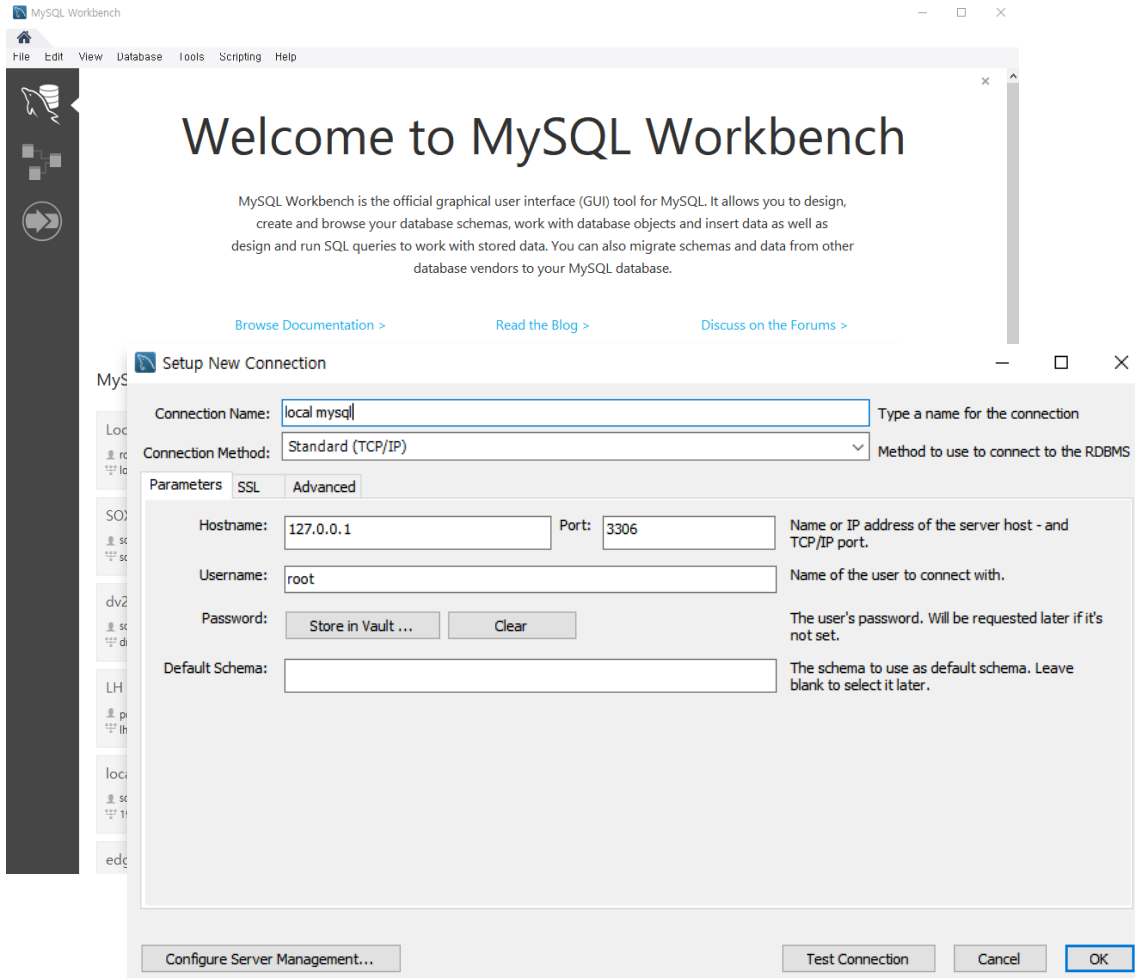
```
flush privileges;
```

```
select user, host from user;
```

2_Mysql Workbench로 접속

workbench 실행

: 접속 정보 입력 후 더블클릭



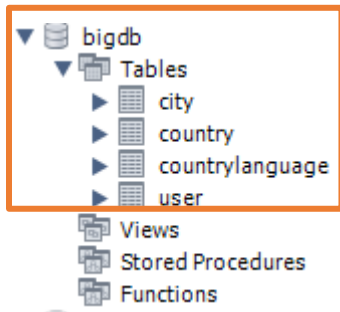
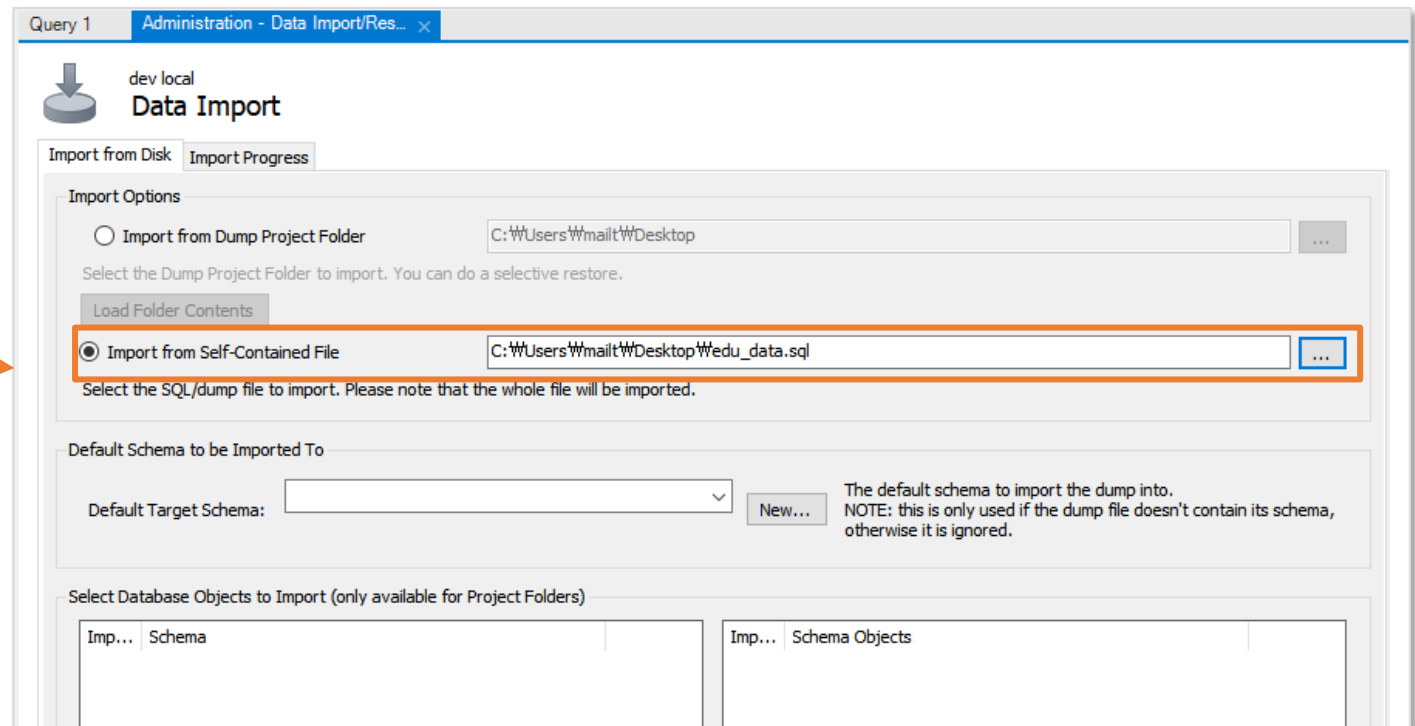
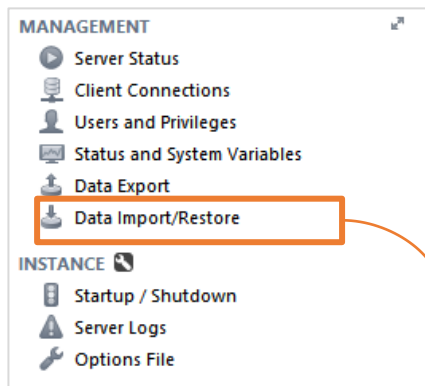
접속 정보 입력

- Connection Name : 이름
- Username : 접속 아이디 (user1)
- Password : 암호 (user123)

2_Mysql 설치

실습 데이터 임포트 하기 : 'edu_data.sql '

→ bigdb 스키마 생성



3_기본 쿼리 실습

■ Workbench 사용

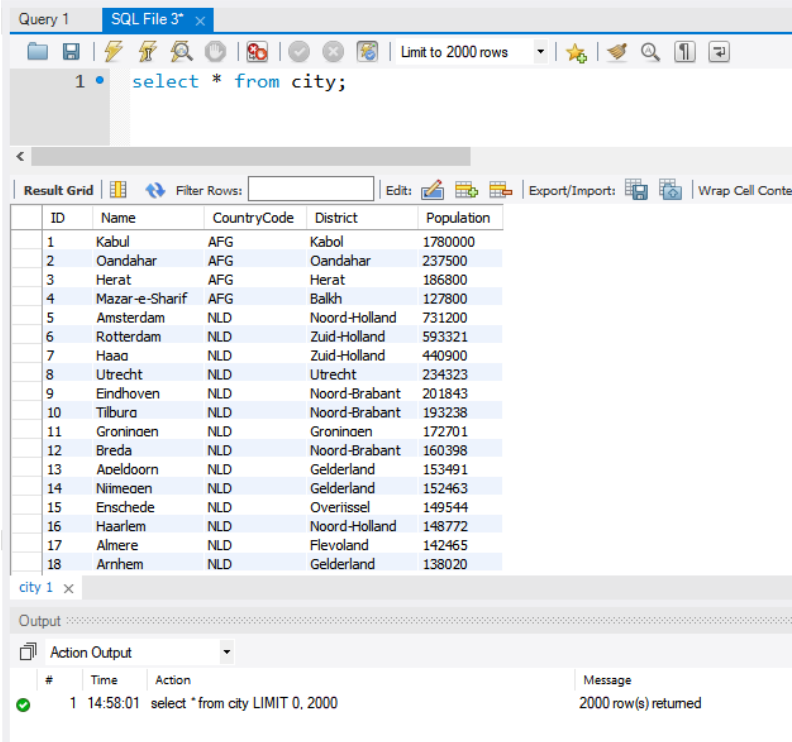
The screenshot displays the MySQL Workbench interface. The top pane shows a list of SQL queries. The second query, `SELECT * FROM city;`, is highlighted with an orange box. Below the queries, the 'Result Grid' shows the output of this query, also highlighted with an orange box. The grid contains columns for ID, Name, CountryCode, District, and Population, with 100 rows of data. The bottom pane shows the 'Action Output' tab, which lists the execution of the query and the number of rows returned (100 row(s) returned).

ID	Name	CountryCode	District	Population
4064	Odessa	USA	Texas	89293
4065	Carson	USA	California	89089
4066	Charleston	USA	South Carolina	89063
4067	Charlotte Amalie	VIR	St Thomas	13000
4068	Harare	ZWE	Harare	1410000
4069	Bulawavo	ZWE	Bulawavo	621742
4070	Chitungwiza	ZWE	Harare	274912
4071	Mount Darwin	ZWE	Harare	164362
4072	Mutare	ZWE	Manicaland	131367
4073	Gweru	ZWE	Midlands	128037
4074	Gaza	PSE	Gaza	353632
4075	Khan Yunis	PSE	Khan Yunis	123175
4076	Hebron	PSE	Hebron	119401
4077	Jabalva	PSE	North Gaza	113901
4078	Nablus	PSE	Nablus	100231
4079	Rafah	PSE	Rafah	92020
4080	Seoul	KOR	Seoul	10000000
NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message
12	21:30:22	SELECT * FROM city LIMIT 0, 100	100 row(s) returned
13	21:30:49	SELECT DISTINCT CountryCode FROM city LIMIT 0, 100	100 row(s) returned
14	21:31:12	SELECT count(DISTINCT CountryCode) FROM city LIMIT 0, 100	1 row(s) returned
15	21:33:36	SELECT * FROM city LIMIT 0, 100	100 row(s) returned
16	21:34:40	SELECT COUNT(*) as 'CityCount' CountryCode FROM city GROUP BY CountryCode LIMIT 0, 100	100 row(s) returned

3_기본 쿼리 실습

- SELECT는 테이블의 데이터를 읽어 출력
- SELECT 필드목록
FROM 테이블
WHERE 조건
[ORDER BY 정렬기준]
- SELECT와 FROM 사이의 필드 목록에 출력할 필드의 이름을 지정하되 * 기호는 모든 필드 출력
 - SELECT * FROM city;
- WHERE 절은 읽을 레코드의 조건을 지정한다.
 - 필드와 특정값을 비교하는 조건문 형식으로 작성한다.
 - SELECT * FROM city WHERE Population > 700000;
 - 특정 레코드의 특정 필드만 표시
 - SELECT Name, Population FROM city WHERE Population > 700000;
 - 문자열이나 날짜는 작은 따옴표로 감싼다.
 - AND는 두 조건이 모두 참인 레코드를 검색하며 OR는 두 조건 중 하나라도 참인 레코드를 검사
 - SELECT * FROM city WHERE Population >= 100000 AND Population >= 700000;



The screenshot shows a SQL query editor with a query window and a results grid. The query is `select * from city;`. The results grid displays 18 rows of data with columns: ID, Name, CountryCode, District, and Population. The data includes cities from Afghanistan (AFG) and the Netherlands (NLD).

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
2	Oandahar	AFG	Oandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843
10	Tilburg	NLD	Noord-Brabant	193238
11	Groningen	NLD	Groningen	172701
12	Breda	NLD	Noord-Brabant	160398
13	Apeldoorn	NLD	Gelderland	153491
14	Niimegen	NLD	Gelderland	152463
15	Enschede	NLD	Overijssel	149544
16	Haarlem	NLD	Noord-Holland	148772
17	Almere	NLD	Flevoland	142465
18	Arnhem	NLD	Gelderland	138020

Below the results grid, the 'Action Output' section shows a message: '2000 row(s) returned'.

3_기본 쿼리 실습

- 레코드를 추가하는 명령은 INSERT-INTO-VALUES이다.
 - INSERT INTO 테이블 (필드목록) VALUES (값목록)
 - INSERT INTO city (Name, CountryCode, Population) VALUES ('Seoul', 'KOR', 10000000);
- 명령문과 필드 목록은 딱 한 번만 밝히고 실제 삽입할 데이터만 나열할 수 있다.
 - INSERT INTO city (Name, CountryCode, Population) VALUES ('Busan', 'KOR', 5000000), ('Daejeon', 'KOR', 4000000);
- 레코드를 삭제할 때는 DELETE-FROM-WHERE 명령을 사용한다.
 - DELETE FROM 테이블 WHERE 조건
 - DELETE FROM city WHERE Name = 'Busan';
- 레코드 값을 변경할 때는 UPDATE-SET-WHERE 명령을 사용한다.
SET 키워드 뒤에 필드에 값을 대입하는 대입문이 옴 (복수개의 필드를 한꺼번에 변경가능)
 - UPDATE 테이블 SET 필드=값 [,필드=값] WHERE 조건
 - 서울의 인구를 1100만명으로 변경
UPDATE city SET Population = 11000000 WHERE Name = 'Seoul';

3_기본 쿼리 실습

```
SELECT * FROM city WHERE Population > 700000;
```

```
SELECT Name, Population FROM city WHERE Population > 700000;
```

```
SELECT * FROM city WHERE Population >= 100000 AND Population >= 700000 ;
```

```
SELECT * FROM city WHERE name LIKE 'Over%';
```

```
SELECT * FROM city ORDER BY population DESC LIMIT 4;
```

```
INSERT INTO city (Name, CountryCode, Population) VALUES ('Seoul', 'KOR', 10000000);
```

```
INSERT INTO city (Name, CountryCode, Population) VALUES  
('Busan', 'KOR', 5000000),  
('Daejeon', 'KOR', 4000000);
```

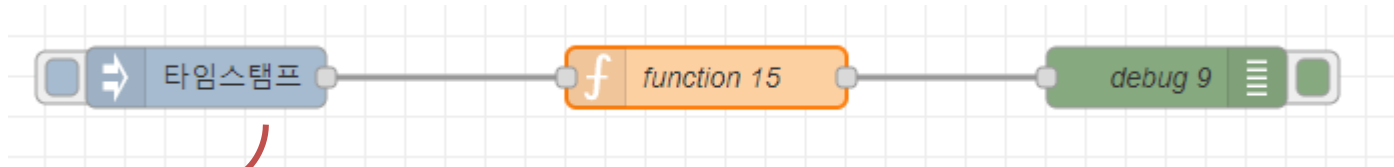
```
DELETE FROM city WHERE Name = 'Busan';
```

```
UPDATE city SET Population = 11000000 WHERE Name = 'Seoul';
```

응용 예제

실습-1

실습 문제 : inject 노드에서 2개의 숫자를 받고 function 노드에서 두 수를 더한 결과를 debug 노드에서 출력하도록 노드를 구성하시오.



The configuration window for the '타임스탬프' node is shown. It has a title bar with '속성' (Properties) and icons for settings, save, and close. Below the title bar is a label '이름' (Name) with a text input field containing '이름'. The main area contains three rows of configuration, each with a label, an equals sign, a dropdown menu, and a text input field. The first row is 'msg. payload' = 'timestamp'. The second row is 'msg. num1' = '0_9 100'. The third row is 'msg. num2' = '0_9 200'. A red box highlights the second and third rows.

실습-2

실습 문제 : 클라이언트(웹페이지)에서 2개의 숫자를 받고 두 수를 더한 결과를 돌려주는 REST API 노드를 구성하시오.



속성

메소드

GET

URL

/plus

이름

이름