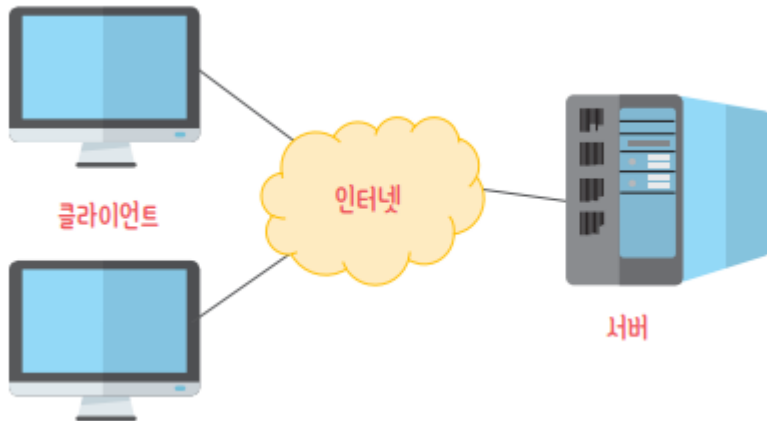


네트워크 및 데이터베이스 프로그래밍

예제 소스 코드는 파일과 연결되어 있습니다.
editplus(유료), notepad++(무료)와 같은 편집 도구를
미리 설치하여 PPT를 슬라이드 쇼로 진행할 때 소스
파일과 연결하여 보면 강의하실 때 편리합니다.

네트워크 기초

■ 서버와 클라이언트

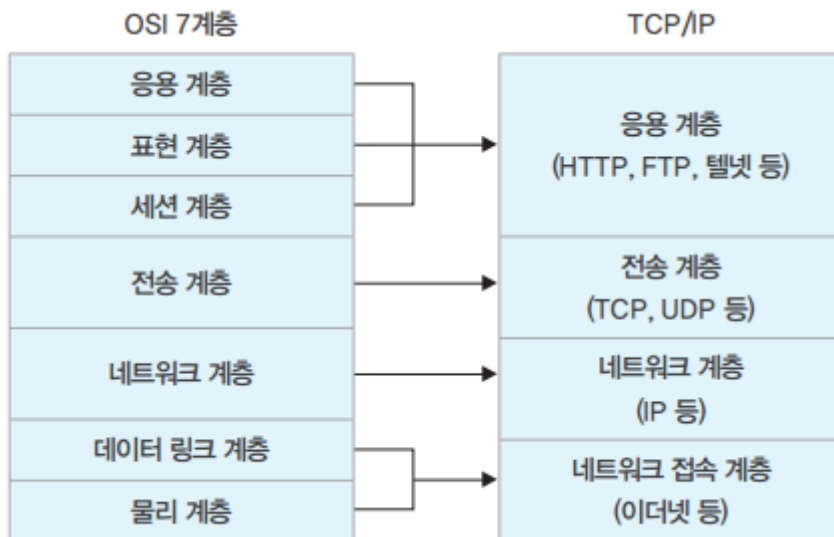


- 서버는 클라이언트보다 미리 실행되어 클라이언트의 요청을 대기
- 클라이언트는 서비스를 요청하기 전에 먼저 서버와 연결 시도
- 클라이언트가 연결을 요청하면 서버는 수용 혹은 거부
- 클라이언트와 서버가 서로 연결되면 클라이언트의 요청을 서버가 처리해서 클라이언트에 응답

네트워크 기초

■ TCP와 UDP

- OSI 모델과 TCP/IP



네트워크 기초

■ TCP와 UDP

● TCP

- 전화와 유사한 연결 지향 프로토콜
- 데이터 손실이 없고 데이터의 전달 순서가 보장
- 연결 설정과 해제에 따른 시간적 부담이 발생

● UDP

- 편지와 유사한 비연결 지향 프로토콜
- 데이터 전달 속도가 빠름
- 데이터 손실 발생 가능성이 있고 데이터의 전달 순서를 보장 않음

네트워크 기초

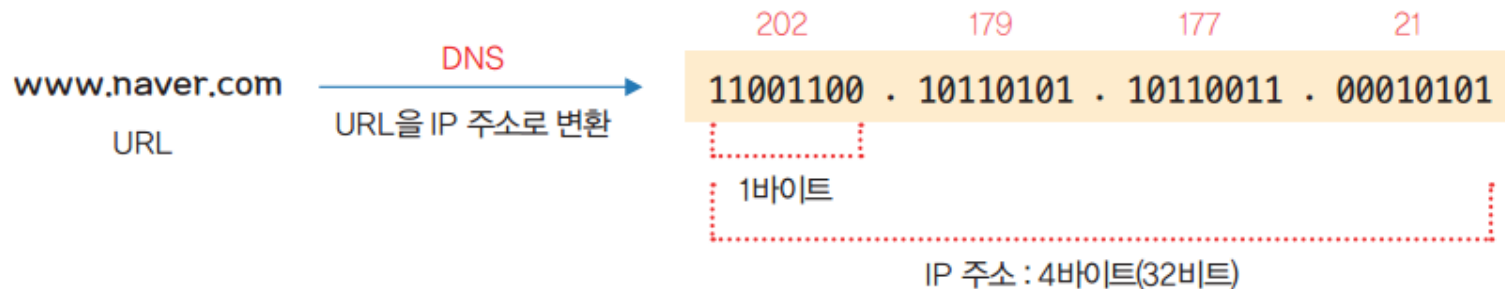
■ IP 주소와 DNS 서버

● IP 주소

- 컴퓨터에 부여된 고유 주소. 즉 인터넷 주소
- 네트워크 어댑터마다 할당
- 32비트로 구성되며, xxx.xxx.xxx.xxx 형식으로 표현

● DNS 서버

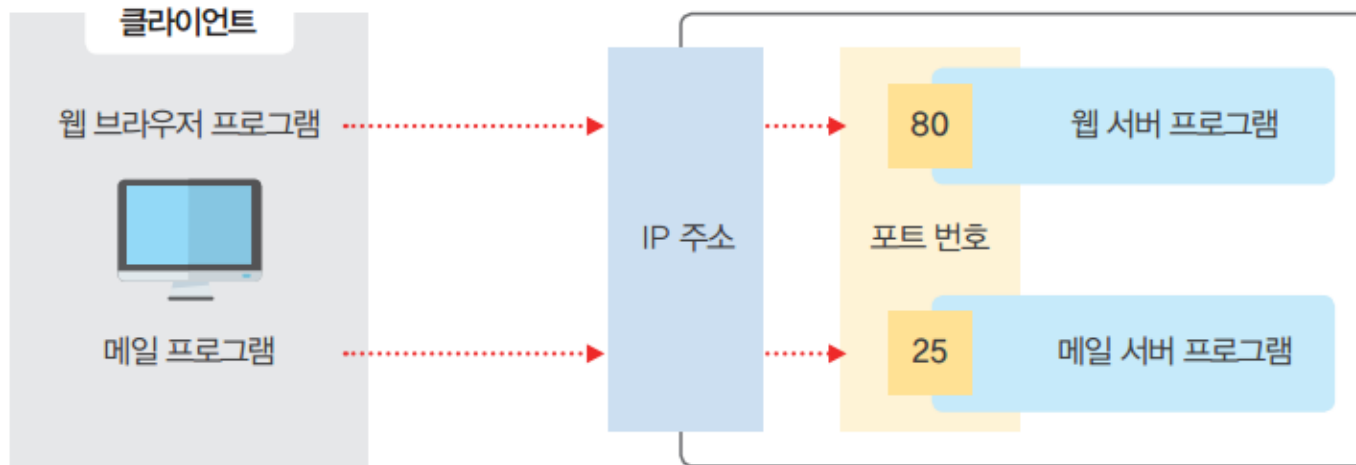
- 114 전화국과 유사하며 URL을 IP 주소로 변환하는 서버
- 숫자보다 문자열이 편하기 때문에 IP 주소보다는 www.naver.com처럼 URL을 사용



네트워크 기초

■ 포트

- 논리적인 통신 연결 번호를 의미하며, 0~65,535사이의 번호를 사용
- 0~1,023은 인터넷주소관리기구가 특정 프로그램에 예약
- 서버 프로그램은 고정된 포트 번호를 사용. 예를 들어 파일 서버는 21번, 메일 서버는 25번, 웹 서버는 80번 포트를 사용
- 클라이언트 컴퓨터에서 웹 브라우저로 웹 서버에 접속하려면 해당 웹 서버의 IP주소 외에 80번 포트로 연결을 요청



IP 주소 얻기

■ InetAddress

- IP 주소를 나타내는 클래스
- java.net 패키지에 있으며 자바 프로그램에서 IP번호와 URL을 조사할 때 사용
- 생성자

```
static InetAddress[] getAllByName(String host)
static InetAddress getByAddress(byte[] addr)
static InetAddress getByAddress(String host, byte[] addr)
static InetAddress getByName(String host)
static InetAddress getLocalHost()
```

- 여기서 addr은 다음과 같이 사용

```
byte[] addr = new byte[4];
addr[0] = (byte)202;
addr[1] = (byte)179;
addr[2] = (byte)177;
addr[3] = (byte)21;
```

IP 주소 얻기

■ InetAddress

- InetAddress 클래스가 제공하는 주요 메서드

메서드	설명
byte[] getAddress()	IP 주소를 배열 타입으로 변환한다.
String getHostAddress()	IP 주소를 String 타입으로 변환한다.
String getHostName()	호스트 이름을 String 타입으로 변환한다.

- 예제 : [sec01/InetAddressDemo](#)

```
호스트 이름을 입력하시오 : www.hanbit.co.kr
www.hanbit.co.kr의 IP 주소 : 218.38.58.195
로컬 IP 주소 : 192.168.0.000
```


소켓과 클라이언트-서버 통신

■ 소켓

- 서버와 클라이언트 간에 통신할 수 있도록 추상화한 연결 통로, 즉 두 컴퓨터 간에 통신하는 끝부분을 의미
- 자바 클라이언트와 자바 서버는 `java.net.Socket` 클래스를 사용해 상호 데이터를 전송
- 클라이언트는 `Socket` 생성자로 소켓을 생성하지만, 서버는 `ServerSocket` 클래스의 생성자로 서버 소켓을 생성한 후 그것을 사용해 `Socket` 객체를 생성
- `Socket` 클래스의 주요 생성자

```
Socket()
```

```
Socket(InetAddress address, int port)
```

```
Socket(String host, int port)
```

소켓과 클라이언트-서버 통신

■ 소켓

- Socket 클래스가 제공하는 주요 메서드

메서드	설명
<code>void close()</code>	소켓을 닫는다.
<code>void connect(SocketAddress endpoint)</code>	소켓을 서버와 연결한다.
<code>InetAddress getInetAddress()</code>	원격 컴퓨터의 <code>InetAddress</code> 객체를 가져온다.
<code>InetAddress getLocalAddress()</code>	로컬 컴퓨터의 <code>InetAddress</code> 객체를 가져온다.
<code>InputStream getInputStream()</code>	소켓에서 <code>InputStream</code> 객체를 가져온다.
<code>OutputStream getOutputStream()</code>	소켓에서 <code>OutputStream</code> 객체를 가져온다.
<code>int getLocalPort()</code>	로컬 컴퓨터의 포트 번호를 가져온다.
<code>int getPort()</code>	원격 컴퓨터의 포트 번호를 가져온다.

소켓과 클라이언트-서버 통신

■ 소켓

- 서버는 다수의 클라이언트와 상대하므로 클라이언트와 달리 연결 요청을 처리할 수 있는 ServerSocket 클래스를 사용
- ServerSocket은 클라이언트가 연결을 요청하면 대응하는 Socket 객체를 생성하는 역할
- 사용하는 ServerSocket클래스의 생성자

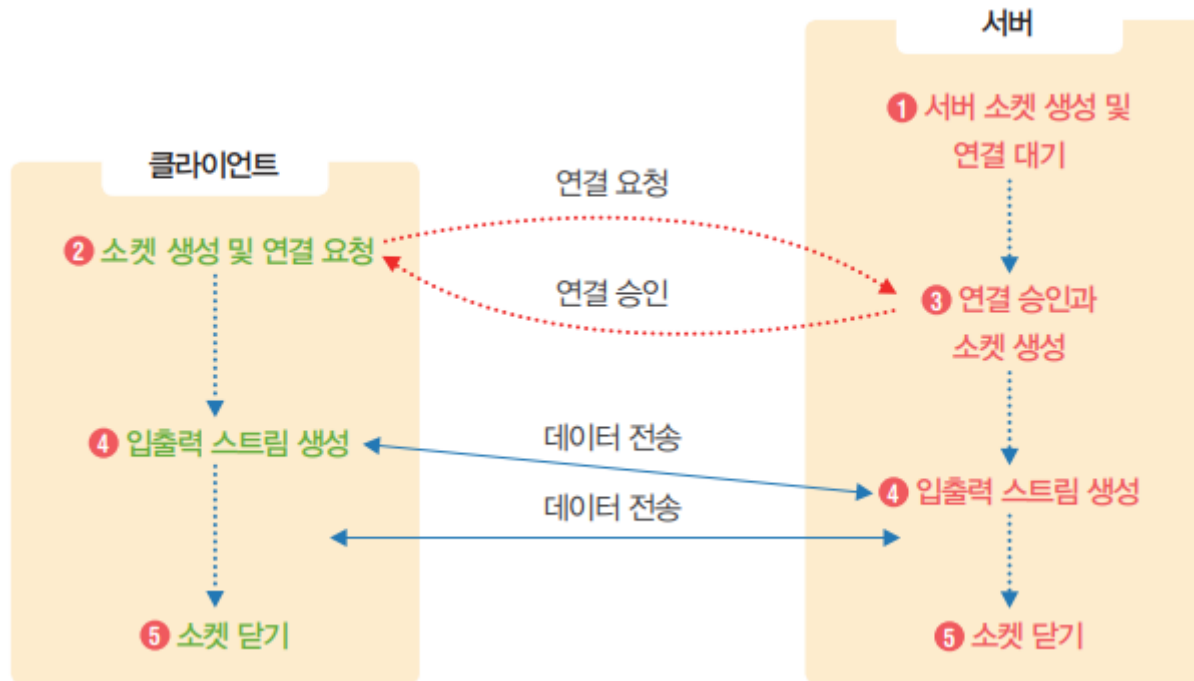
```
ServerSocket(int port)
```

- ServerSocket 클래스가 제공하는 주요 메서드

메서드	설명
Socket accept()	클라이언트의 연결 요청을 받아 Socket 객체를 생성한다.
void close()	서버 소켓을 닫는다.
public InetAddress getInetAddress()	소켓에 연결된 인터넷 주소를 가져온다.

소켓과 클라이언트-서버 통신

■ 소켓을 이용한 클라이언트-서버 통신 과정



소켓과 클라이언트-서버 통신

■ 예제

- 문자열을 받는 간단한 서버 : [sec02/SimpleServer](#)
- 문자열을 보내는 간단한 클라이언트 : [sec02/SimpleClient](#)
- 명령 창에서 서버와 클라이언트의 통신 과정

서버

```
C:\ 명령 프롬프트
1 D:\workspace\chap17\bin>java sec02. SimpleServer
3 받은 문자열 = 안녕, 단순 서버야.

D:\workspace\chap17\bin>
```

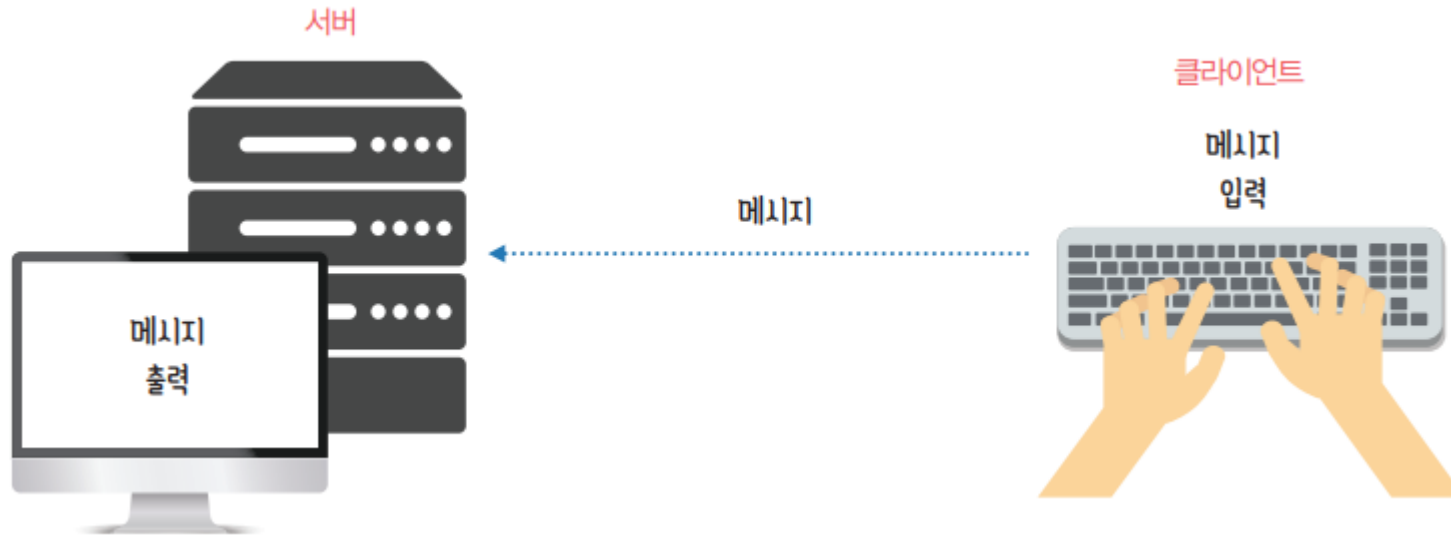
클라이언트

```
C:\ 명령 프롬프트
2 D:\workspace\chap17\bin>java sec02. SimpleServer
받은 문자열 = 안녕, 단순 서버야.

D:\workspace\chap17\bin>
```

클라이언트-서버의 소켓 통신 응용

■ 에코 프로그램의 동작 과정



클라이언트-서버의 소켓 통신 응용

■ 예제 : 일대일 서버 및 클라이언트

- 일대일 에코 서버 : [sec03/Echo1Server](#)
- 일대일 에코 클라이언트 : [sec03/EchoClient](#)

- 서버

```
D:\workspace\chap18\bin> java sec03.Echo1Server
```

연결 대기 중... ①

메시지를 기다리는 중... ③

읽은 메시지 메아리 : 안녕하세요. ⑤

읽은 메시지 메아리 : 좋습니다. ⑦

서버 종료 ⑨

- 클라이언트

```
D:\workspace\chap18\bin> java sec03.EchoClient
```

에코 서버와 연결 시도... ②

에코 서버와 연결 성공...

보낼 메시지가 있나요? 안녕하세요. ④

보낼 메시지가 더 있나요? 좋습니다. ⑥

보낼 메시지가 더 있나요? 끝. ⑧

클라이언트 종료

클라이언트-서버의 소켓 통신 응용

■ 예제 : 일대다 서버 및 클라이언트

- 일대다 에코 서버 : [sec03/Echo2Server](#)
- 일대다 클라이언트 : 일대일 클라이언트와 동일

- 서버

```
D:\workspace\chap17\bin> java sec03.Echo2Server
```

서버 소켓 생성

연결 대기 중...

1

연결 대기 중...

클라이언트와 통신을 위한 새로운 스레드 생성

3

읽은 메시지 메아리 : 나야, 봉이 김선달

5

연결 대기 중...

클라이언트와 통신을 위한 새로운 스레드 생성

7

읽은 메시지 메아리 : 나야, 의적 임꺽정

9

- 첫 번째 클라이언트

에코 서버와 연결 시도...

에코 서버와 연결 성공...

2

보낼 메시지가 있나요? 나야, 봉이 김선달

4

보낼 메시지가 더 있나요?

- 두 번째 클라이언트

에코 서버와 연결 시도...

에코 서버와 연결 성공...

6

보낼 메시지가 있나요? 나야, 의적 임꺽정

8

보낼 메시지가 더 있나요?

데이터베이스 기초

■ 데이터베이스?

- 데이터베이스 관리 시스템(DBMS)은 크게 세 가지 종류인 계층형, 네트워크형, 관계형으로 구분
- 최근에는 관계형 DBMS가 주류
- 관계형 데이터베이스는 일련의 정형화된 테이블로 구현.
- 관계형 데이터베이스에서는 테이블을 재구성하지 않더라도 데이터를 다양한 방법으로 접근하거나 조합 가능
- 관계형 데이터베이스는 제작과 이용이 비교적 쉬울 뿐만 아니라 확장도 용이

데이터베이스 기초

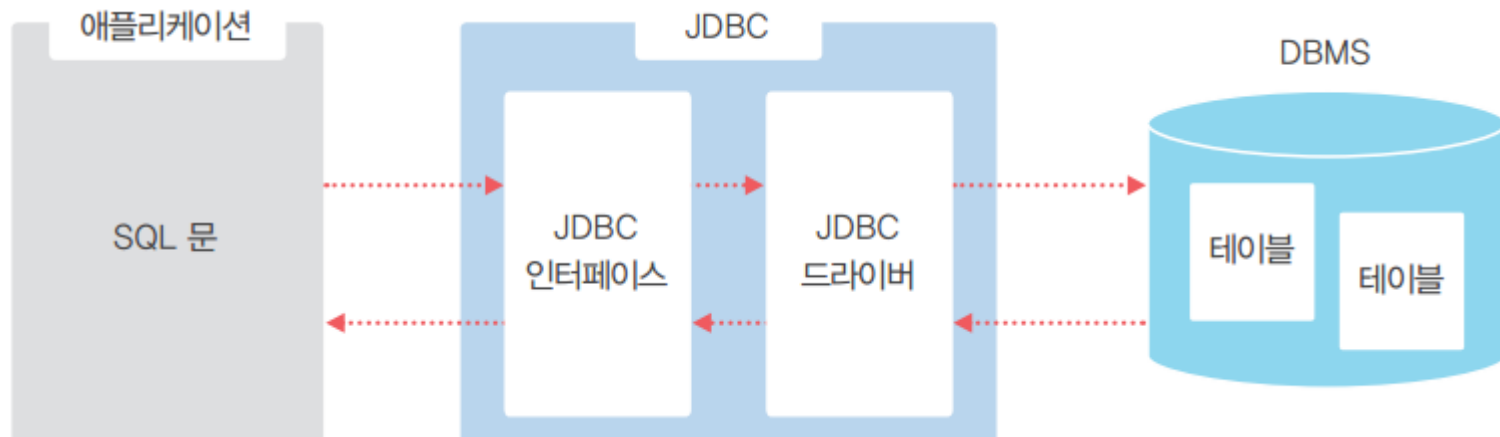
■ 관계형 데이터베이스 테이블의 예



데이터베이스 기초

■ JDBC 드라이버

- DBMS는 종류가 다양하며, 그 구조와 특징도 다 다르다. 따라서 자바 표준 API를 확장해 DBMS에 접근한다는 것은 자바의 기본 개념에 맞지 않을 뿐만 아니라 개발자에게도 고통
- 자바는 모든 DBMS에서 공통으로 사용할 수 있는 인터페이스와 클래스로 구성된 JDBC를 제공
- 각 DBMS 벤더는 JDBC를 구현한 JDBC드라이버를 개발자에게 제공
- JDBC가 데이터베이스에 접근하는 과정



데이터베이스 기초

■ SQL 명령어

- SQL 명령어는 반드시 세미콜론(;)으로 종료

SQL 명령어	설명	
DDL	CREATE	테이블을 생성한다.
	DROP	테이블을 삭제한다.
	ALTER	테이블 구조를 변경한다.
	USE	사용할 데이터베이스를 지정한다.
DML	INSERT	데이터를 추가한다.
	UPDATE	데이터를 갱신한다.
	DELETE	데이터를 삭제한다.
	SELECT	데이터를 조회한다.
DCL	GRANT	테이블에 권한을 부여한다.

MySQL 설치 및 응용

■ MySQL 다운로드 및 설치

- MySQL 다운로드하기(<http://dev.mysql.com/downloads/mysql/>)

General Availability (GA) Releases Archives

MySQL Community Server 8.0.19

Select Operating System:
Microsoft Windows

Looking for previous GA versions?

Operating System	Version	Size	Download
Windows (x86, 32-bit), MSI Installer	8.0.19	18.6M	Download
<small>(mysql-installer-web-community-8.0.19.0.msi) MD5: 32943776cb2239db451dda86dc0ad51 Signature</small>			
Windows (x86, 32-bit), MSI Installer	8.0.19	398.9M	Download
<small>(mysql-installer-community-8.0.19.0.msi) MD5: 1a882015da71b93f20c4717e63b6817c Signature</small>			

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

Windows (x86, 32 & 64-bit), MySQL Installer MSI

[Go to Download Page >](#)

[Login »](#)
using my Oracle Web account

[Sign Up »](#)
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

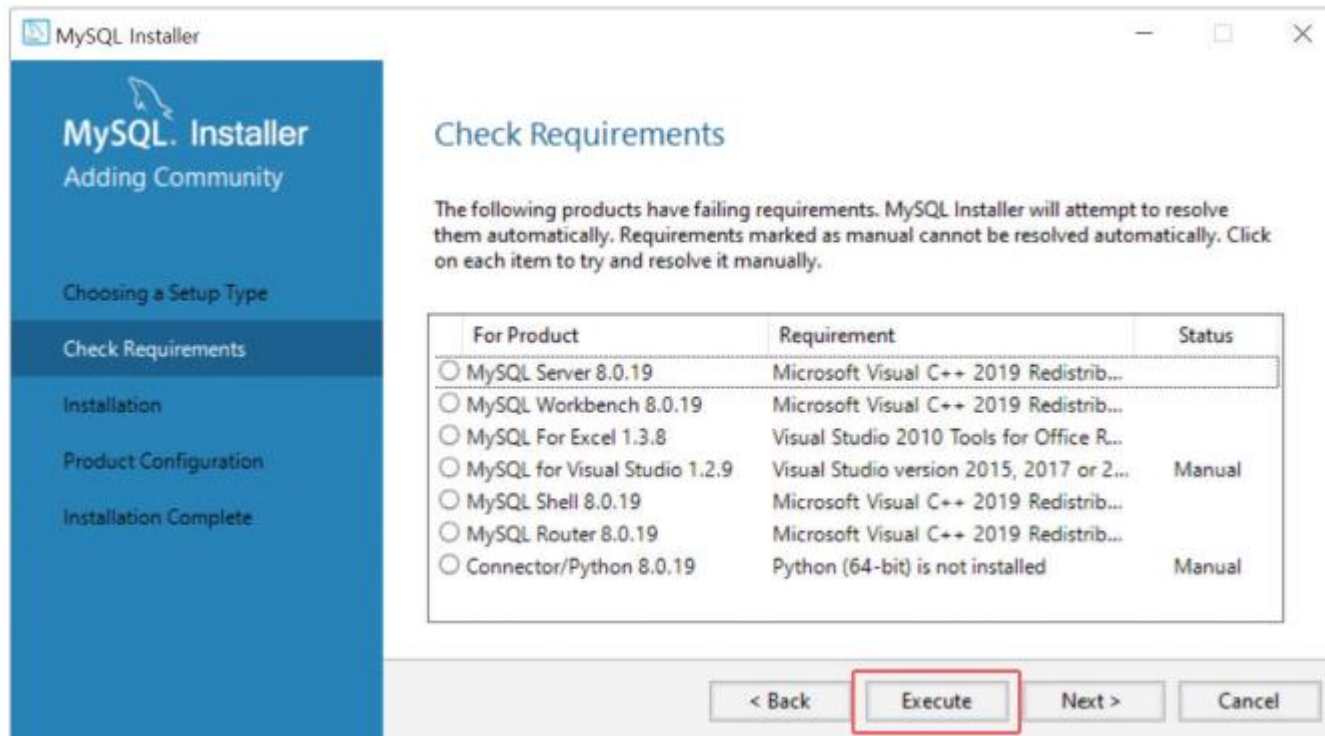
[No thanks, just start my download.](#)

- Microsoft Visual C++ 20xx redistributable이 설치 필요

MySQL 설치 및 응용

■ MySQL 다운로드 및 설치

- MySQL 설치하기 : 다양한 설정이 있지만, 여기서는 라이선스에 동의한 후 기본 설정을 그대로 두고 [Execute] 버튼이나 [Next] 버튼을 클릭해 설치하면 큰 무리가 없다. 특히, 다음 화면이 나타나면 [Execute] 버튼을 클릭하여 MySQL 설치에 필요한 소프트웨어 설치 필요



MySQL 설치 및 응용

■ MySQL 다운로드 및 설치

- 사용자 계정 설정 및 설치 완료 : 루트 비밀번호(편의상 'password')를 입력. 계속해서 단계별 [Finish] 과정을 여러 번 반복하게 되며 마지막으로 MySQL Workbench 환영 창이 나타나면 설치가 완료

■ MySQL 드라이버 추출

- 정상적으로 설치하면 'C:\Program Files (x86)\MySQL\Connector J 버전' 폴더에 MySQL 드라이버((mysql-connector-java-버전번호.jar)가 있음
- MySQL 드라이버 파일을 임의 폴더(여기서는 C:\Program Files\MySQL\로 이동

MySQL 설치 및 응용

■ MySQL 드라이버 설정하기

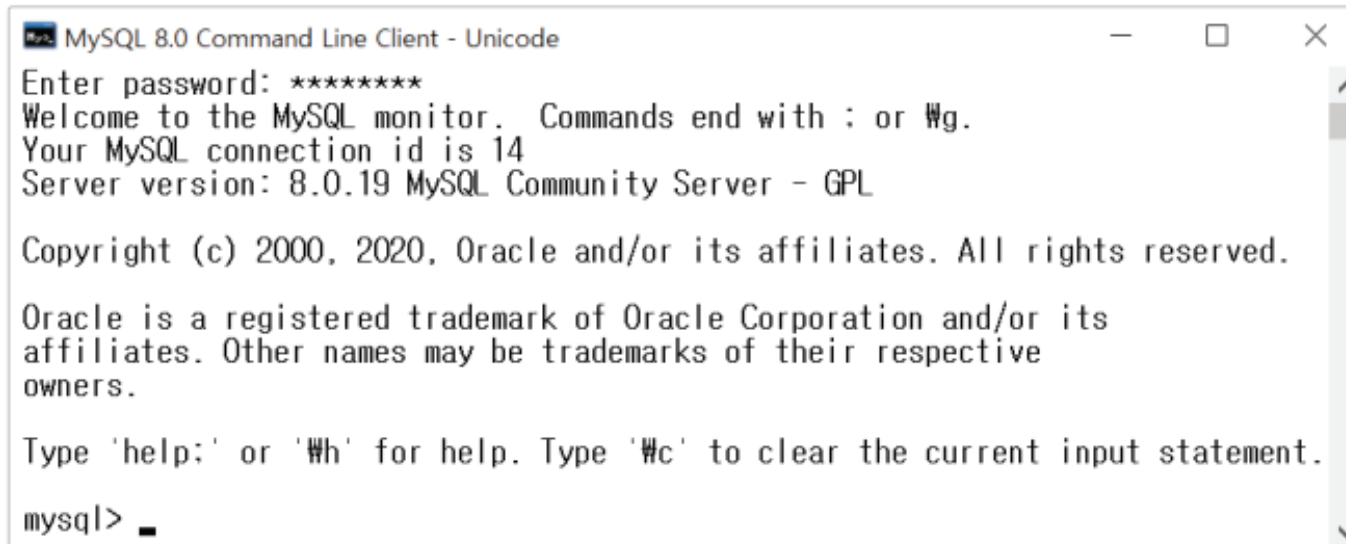
The following steps illustrate the process of adding the MySQL driver to the Eclipse IDE:

1. In the **Properties for chap18** dialog, select **Java Build Path** from the left-hand list.
2. In the **Java Build Path** tab, click the **Libraries** sub-tab.
3. Click the **Add External JARs...** button.
4. In the **JAR Selection** dialog, click the **Add External JARs...** button.
5. In the **JAR Selection** dialog, select **mysql-connector-java-8.0.19** from the file list.
6. In the **Properties for chap18** dialog, confirm the addition of **mysql-connector-java-8.0.19.jar** to the **Modulepath**.
7. Click **Apply and Close** in the **Properties for chap18** dialog.
8. In the **Package Explorer**, verify that **mysql-connector-java-8.0.19.jar** is listed under **Referenced Libraries**.

MySQL 설치 및 응용

■ SQL 명령어

- 대부분의 DBMS는 이런 SQL명령어를 직접 실행할 수 있는 명령 창을 제공



```
MySQL 8.0 Command Line Client - Unicode
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.19 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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 설치 및 응용

■ SQL 명령어

```
CREATE DATABASE 데이터베이스_이름;
```

```
CREATE TABLE 테이블_이름 (열_이름, 열_타입, ... )
```

```
CREATE DATABASE contacts;
```

contacts 데이터베이스를 생성한다.

```
USE contacts;
```

사용할 데이터베이스로 변경한다.

```
CREATE TABLE person (  
  name VARCHAR(20),  
  phone VARCHAR(20),  
  email VARCHAR(20),  
  PRIMARY KEY(phone));
```

contacts 데이터베이스에
person 테이블을 생성한다.
person 테이블은 name,
phone, email 열로 구성되며
phone을 기본키로 사용한다.

MySQL 설치 및 응용

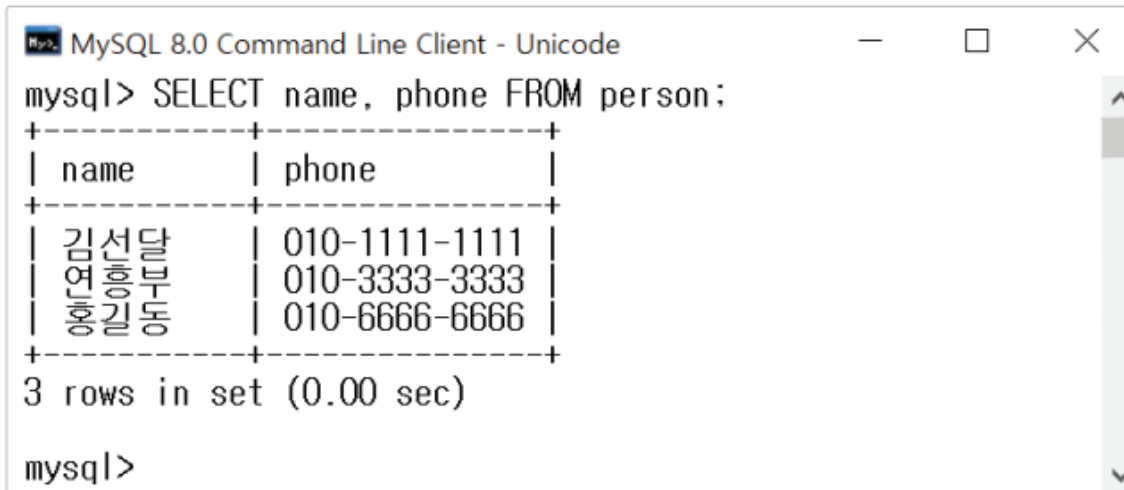
■ SQL 명령어

```
INSERT INTO 테이블_이름 [(열_이름, ...)] VALUES(값, ...);
```

```
INSERT INTO person (name, phone, email) VALUES ('김선달', '010-1111-1111',  
'kim@one.com');
```

```
SELECT 열_이름, ... FROM 테이블_이름 [WHERE 조건] [ORDER BY 정렬 방식];
```

```
SELECT name, phone FROM person;
```



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client - Unicode". The user has entered the command `mysql> SELECT name, phone FROM person;`. The output is a table with two columns: `name` and `phone`. The table contains three rows of data: `김선달` with phone `010-1111-1111`, `연홍부` with phone `010-3333-3333`, and `홍길동` with phone `010-6666-6666`. Below the table, it says "3 rows in set (0.00 sec)". The prompt `mysql>` is visible at the bottom.

```
mysql> SELECT name, phone FROM person;
```

name	phone
김선달	010-1111-1111
연홍부	010-3333-3333
홍길동	010-6666-6666

```
3 rows in set (0.00 sec)
```

```
mysql>
```

MySQL 설치 및 응용

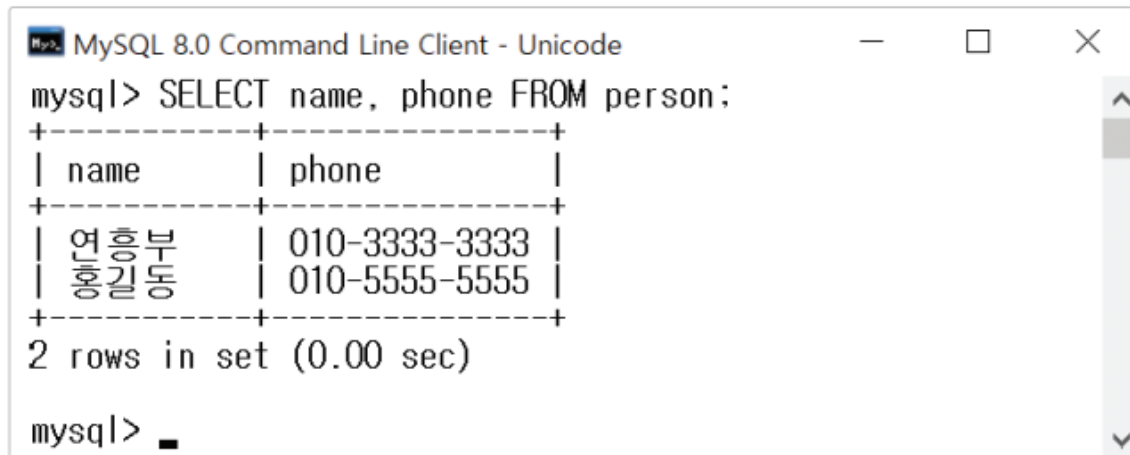
■ SQL 명령어

DELETE FROM 테이블_이름 [WHERE 조건];

DELETE FROM person WHERE name='김선달';

UPDATE 테이블_이름 SET 열_이름=값, ... [WHERE 조건];

UPDATE person SET phone='010-5555-5555' WHERE name='홍길동';



MySQL 8.0 Command Line Client - Unicode

```
mysql> SELECT name, phone FROM person;
```

name	phone
연홍부	010-3333-3333
홍길동	010-5555-5555

2 rows in set (0.00 sec)

```
mysql> _
```

■ JDBC 프로그래밍 과정

- ① JDBC드라이버 로드 및 데이터베이스를 연결한다.
- ② SQL문을 실행한다.
- ③ ResultSet 객체를 처리한다.
- ④ 자원 객체를 닫는다.

JDBC 프로그래밍

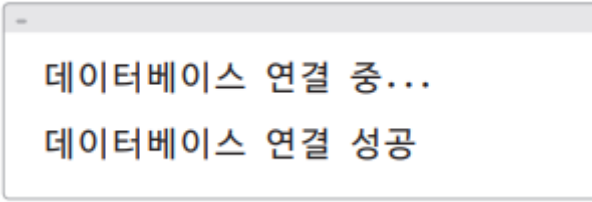
■ JDBC드라이버 로드 및 데이터베이스 연결

- MySQL 드라이버의 경우 다음 실행문으로 드라이버를 메모리에 로드
mysql-connector-java 6.0이전: **Class.forName("com.mysql.jdbc.Driver");**
mysql-connector-java 6.0부터: **Class.forName("com.mysql.cj.jdbc.Driver");**
- 데이터베이스와 연결해 Connection 객체 생성

```
Connection con = DriverManager.getConnection(url, user_id, passwd);
```

- 여기서 url은 다음 형식: jdbc:subprotocol:subname
- mysql-connector-java 6.0부터는 서버 타임 존이 필요
jdbc:mysql://IP주소/DB이름 (mysql-connector-java 6.0이전)
jdbc:mysql://IP주소/DB이름?serverTimezone=Asia/Seoul (mysql-connector-java 6.0부터)

- 예제 : [sec05/DB1Demo](#)



데이터베이스 연결 중...

데이터베이스 연결 성공

JDBC 프로그래밍

■ SQL 문 실행

- SQL문을 DBMS에 전송할 수 있는 Statement 객체 생성

```
Statement stmt = con.createStatement();
```

- Statement 인터페이스가 제공하는 주요 메서드

메서드	설명
boolean execute(String sql)	SQL 문을 수행한다. 실행 결과가 ResultSet 객체라면 true, 아니면 false를 반환한다.
ResultSet executeQuery(String sql)	SQL 문을 수행한다. ResultSet 객체를 반환한다.
int executeUpdate(String sql)	SQL 문을 수행한다. 영향을 받은 레코드의 개수를 반환한다.

- 예제 : [sec05/DB2Demo](#)

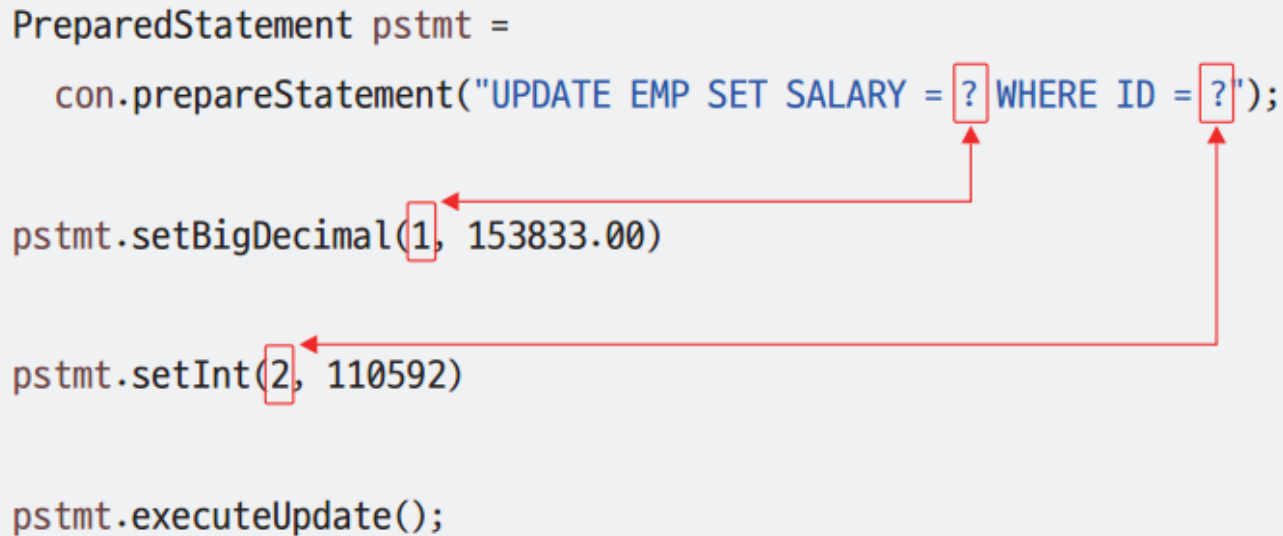
```
데이터베이스 연결 중...
데이터베이스 연결 성공
레코드 추가 성공
```

JDBC 프로그래밍

■ SQL 문 실행

- PreparedStatement 인터페이스
 - Statement의 자식 인터페이스
 - SQL문을 미리 만들어 두고 사용할 수 있기 때문에 효율성이나 유지 보수 면에서 유리

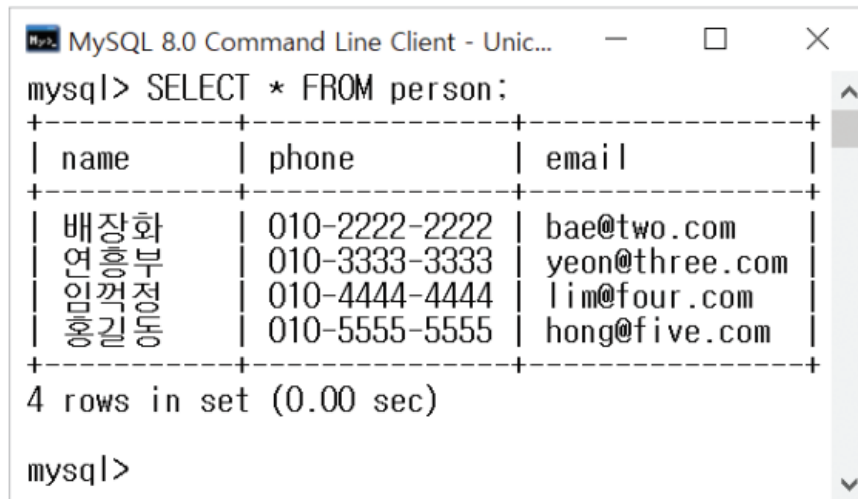
```
PreparedStatement pstmt =  
    con.prepareStatement("UPDATE EMP SET SALARY = ? WHERE ID = ?");  
  
pstmt.setBigDecimal(1, 153833.00)  
  
pstmt.setInt(2, 110592)  
  
pstmt.executeUpdate();
```



JDBC 프로그래밍

■ SQL 문 실행

- PreparedStatement 인터페이스
- 예제 : [sec05/DB3Demo](#)



The screenshot shows a MySQL 8.0 Command Line Client window. The user has entered the command `mysql> SELECT * FROM person;`. The output is a table with 4 rows and 3 columns: `name`, `phone`, and `email`. The data is as follows:

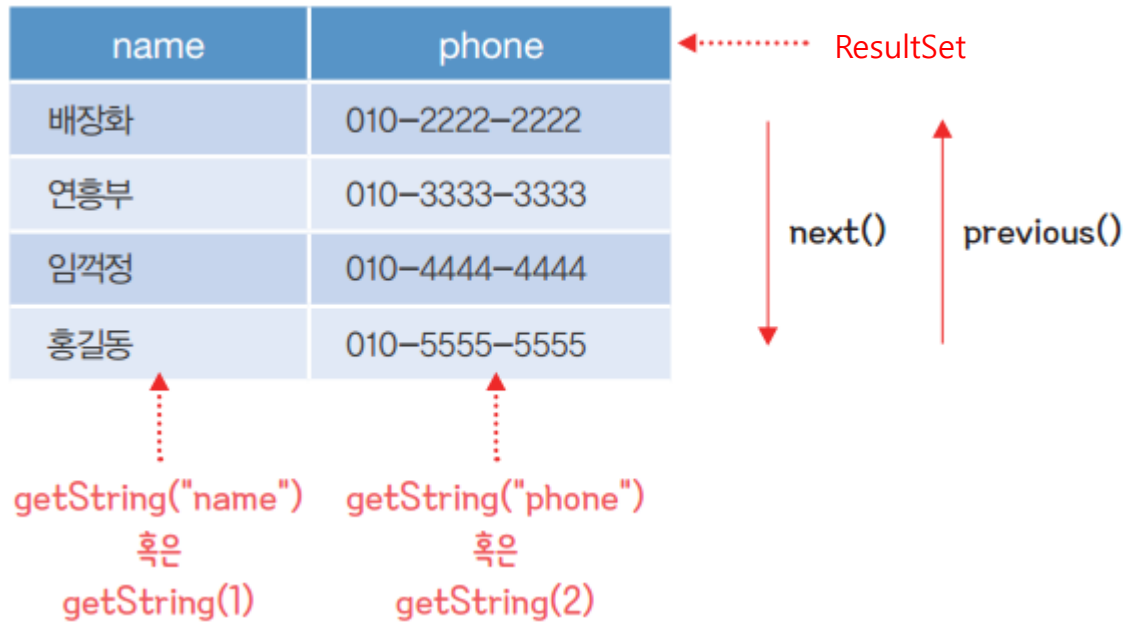
name	phone	email
배장화	010-2222-2222	bae@two.com
연홍부	010-3333-3333	yeon@three.com
임꺼정	010-4444-4444	lim@four.com
홍길동	010-5555-5555	hong@five.com

Below the table, it says "4 rows in set (0.00 sec)". The prompt `mysql>` is visible at the bottom.

JDBC 프로그래밍

■ 결과집합의 처리

- ResultSet은 SQL질의문의 결과물을 추상화한 인터페이스
- 예를 들어 SELECT name, phone FROM person;의 결과집합



JDBC 프로그래밍

■ 결과집합의 처리

- ResultSet 인터페이스가 제공하는 주요 메서드

메서드	설명
void afterLast()	커서를 마지막 행 다음으로 이동한다.
void beforeFirst()	커서를 첫 행 이전으로 이동한다.
void close()	ResultSet 객체를 닫는다.
boolean first()	커서를 첫 행으로 이동한다.
int getInt(int columnIndex) long getLong(String columnLabel)	현재 레코드에서 주어진 열의 값을 int 혹은 long 타입으로 반환한다.
int getRow()	현재 레코드 번호를 반환한다.
String getString(int columnIndex) String getString(String columnLabel)	현재 레코드에서 주어진 열의 값을 String 타입으로 반환한다.
boolean last()	커서를 마지막 행으로 이동한다.
boolean next()	커서를 다음 행으로 이동한다.
boolean previous()	커서를 이전 행으로 이동한다.

JDBC 프로그래밍

■ 결과집합의 처리

- 예제 : [sec05/DB4Demo](#)

데이터베이스 연결 중...

데이터베이스 연결 성공

name : 배장화 phone : 010-2222-2222

name : 연흥부 phone : 010-3333-3333

name : 임꺽정 phone : 010-4444-4444

name : 홍길동 phone : 010-5555-5555