

# SQL – DB 프로그래밍

## - *Database Basic* -

2014.11

이 익 훈

E-mail: ihlee90@nhn.com

ihlee90@gmail.com

# 학습 목표

- I. DB 응용 프로그래밍을 할 수 있다.
  - C++이용해서 MySQL DB를 조회, 조작할 수 있다.
  - ODBC를 이용한 환경설정 및 DB 프로그래밍을 할 수 있다.

## ◆ 시나리오 (나 = 이창업)

MySQL DBMS 사용도 좀 할 수 있고, 데이터베이스 설계도 알게 되었어요.

이제 MySQL DB를 이용하는 프로그래밍 방법을 알아야 겠어요. C++로 DB 조작하는 방법을 공부해야 겠네요.

(추가) 인터넷 쇼핑몰 개발을 위해 닷넷 등으로 MySQL을 이용하는 웹프로그래밍 공부가 필요해요.

## ◆ 실습 환경

### 1. 서버: Windows (실습 PC)

- OS: Windows 7
- Java: JDK 1.6 이상
- DBMS: Mysql 5.6 (db 계정: nextuser )

### 2. 클라이언트 : 학생 실습용 노트북

- OS: Windows7
- MySQL Workbench
- IDE: Visual C++



# C++를 이용한 DB 프로그래밍 (MySQL)

## ◆ 실습 순서

1. MySQL 서버 정보 확인 및 접속
2. DB 사용자 정보 확인 및 DB 생성
3. ODBC 드라이버 연결
4. SELECT
5. UPDATE / INSERT
6. Prepared statement

환경  
설정

프로그래밍

## ◆ 실습을 위해 사용하는 데이터

- ◆ 데이터베이스 : 원하는이름\_bookshop
- ◆ 테이블

```
CREATE TABLE book (  
    id      INT,  
    title VARCHAR(100),  
    author  VARCHAR(50),  
    publication  VARCHAR(30),  
    pub_year  INT,  
    isbn  CHAR(13),  
    price  INT,  
    num_inventory  INT  
);
```

- NEXT 도서목록 자료 사용

## 1. 서버 정보 확인 및 접속

- ◆ 서버 도메인 주소: 실습 서버  
(교수의 실습 PC)

- ◆ OS: Windows7

- \* 원격지의 윈도우 서버라면 원격데스크탑으로 접속해서 확인

- \* Unix, Linux 서버라면, Putty 등의 터미널 프로그램을 설치하고 ssh로 터미널 접속

- .



## 2. DB 사용자 정보확인 및 DB 생성

- ◆DB 사용자 : nextuser

- Terminal에서 서버로 직접 접속  
또는 클라이언트의 Workbench로 접속

- ◆개인 DB 생성하세요.

원하는 이름의 **DB 생성...** *학번\_book*

nextuser에게 생성된 DB에 대한 **권한 추가**  
대량 **데이터 삽입**

### 3. ODBC 드라이버 연결

- ◆ MySQL Connector 다운로드

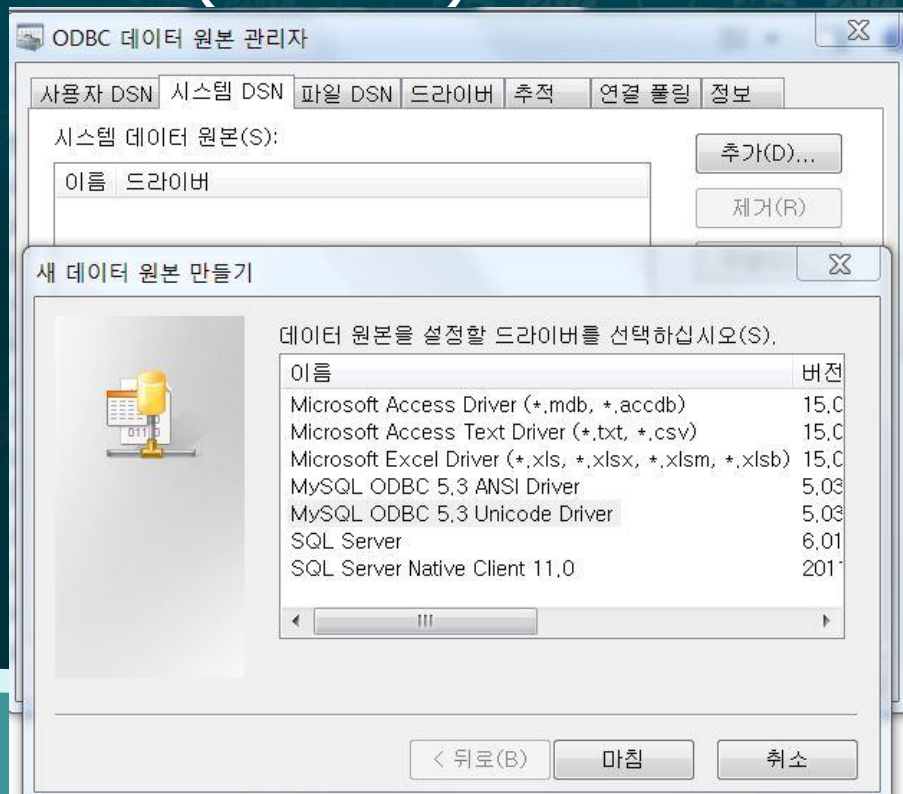
<http://dev.mysql.com/downloads/connector/odbc/>

- ◆ ODBC 설치 파일을 다운로드 받아서 실행

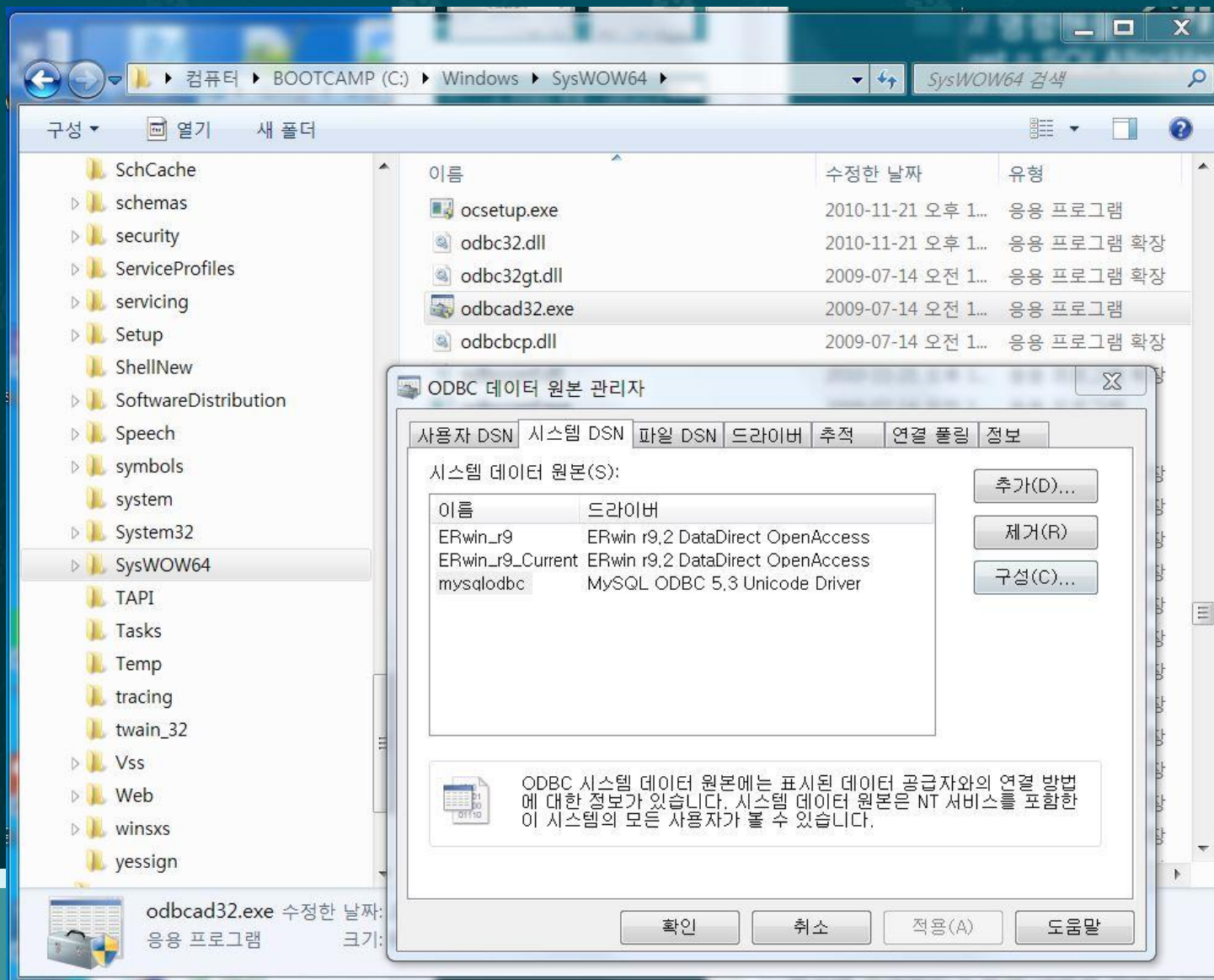
\* OS 에 맞게 32비트 또는 64비트 설치

### 3. ODBC 드라이버 연결 (계속)

#### ◆ 윈도우 제어판의 “관리도구 > 데이터원본 (ODBC)” 설정



# ODBC 32 bit 설정 (64비트 OS에서)





## 4. ODBC 연결 - SELECT

### [MySQL 연동하는 프로그램 예]

```
#include "stdafx.h"  
#include "Windows.h"  
#include "iostream"  
  
using namespace std;  
#include "sql.h"  
#include "sqlext.h"  
int _tmain(int argc, _TCHAR* argv[]) {  
    int ret;  
    SQLHENV hEnv;  
    SQLHDBC hDbc;  
    SQLHSTMT hStmt;
```

//SQL과 연결

```
ret = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &hEnv);  
ret = SQLSetEnvAttr(hEnv, SQL_ATTR_ODBC_VERSION,  
    (SQLPOINTER)SQL_OV_ODBC3, SQL_IS_INTEGER);  
ret = SQLAllocHandle(SQL_HANDLE_DBC, hEnv, &hDbc);
```

```
ret = SQLConnect(hDbc, (SQLWCHAR*)L"mysqlodbc64", SQL_NTS,  
    (SQLWCHAR*)L"nextuser", SQL_NTS,  
    (SQLWCHAR*)L"dbgood", SQL_NTS);
```

```
if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO)  
    printf("SQL Connect Ok \n");  
else { // display error code when we get error... SQLGetDiagRec 참고  
    printf("sqlconnect error .\n");    return(1);  
}
```

// 명령행들

```
ret = SQLAllocHandle(SQL_HANDLE_STMT, hDbc, &hStmt);  
ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT author, title, price FROM  
    book", SQL_NTS);
```

```
if (ret == SQL_SUCCESS) {  
    int iCount = 0;  
    SQLLEN iAuthorLen, iTitleLen, iPriceLen, iPrice;  
    char strAuthor[200], strTitle[200];
```

Insert,  
Update도  
가능

```

while (TRUE) {
    ret = SQLFetch(hStmt); // 값을 얻어올때
    if (ret == SQL_ERROR || ret == SQL_SUCCESS_WITH_INFO)
        printf("An error occured\n");
    if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
        SQLGetData(hStmt, 1, SQL_C_WCHAR, strAuthor, 200, &iAuthorLen);
        SQLGetData(hStmt, 2, SQL_C_WCHAR, strTitle, 200, &iTitleLen);
        SQLGetData(hStmt, 3, SQL_C_ULONG, &iPrice, 0, &iPriceLen);

        /* Print the row of data */
        _tprintf(_T(" Row  %d : "), iCount++);
        _tprintf(_T(" %ls  %ls  %d \n"), strAuthor, strTitle, iPrice);
    }
}

else
    printf("An error occured during excuting query!! \n");

//접속 종료 및 반환
if (hStmt) SQLFreeHandle(SQL_HANDLE_STMT, hStmt);
if (hDbc) SQLDisconnect(hDbc);
if (hDbc) SQLFreeHandle(SQL_HANDLE_DBC, hDbc);
if (hEnv) SQLFreeHandle(SQL_HANDLE_ENV, hEnv);

return 0;
}

```

한글 주의  
 -DB가 utf8  
 -ODBC가 utf8  
 -SQLGetData  
 가 wchar



# 참고: Prepared statement

- ◆ 다수의 Insert, Update, Select 시에 SQL에 입력되는 값을 변수화함
  - ◆ 코드의 생산성, 속도 측면에서 유리한 방법
  - ◆ SQLPrepare(), SQLExecute() 사용





**NHN INSTITUTE FOR THE  
NEXT NETWORK**