JDBC & ODBC

- Contents
  - Sample database
  - JDBC
  - ODBC
  - JDBC and ODBC API on-line Reference Manual

Sample DB 및 테이블

데이타베이스: university

테이블: student

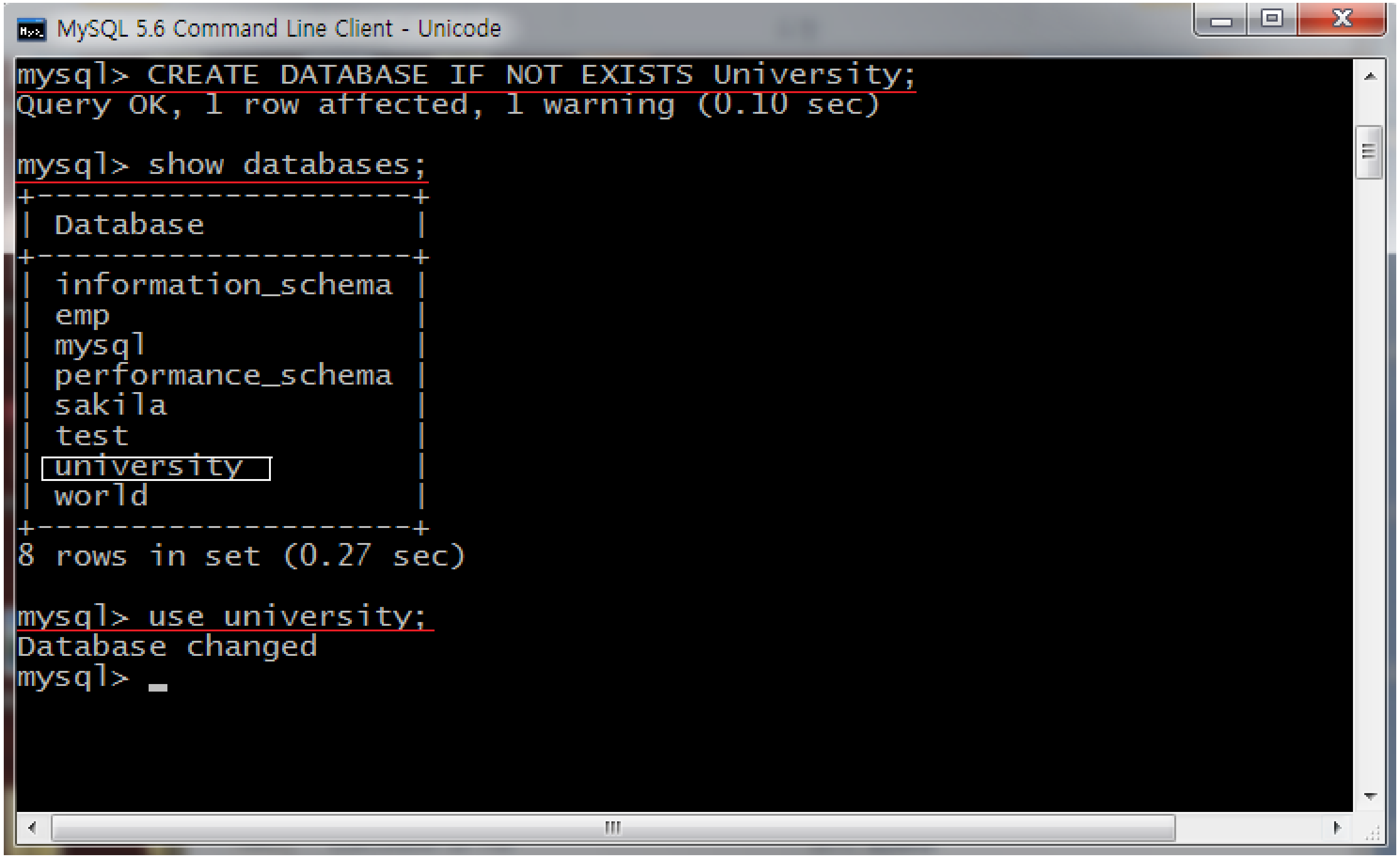
student

학번 (ID)	이름 (sname)	학과 (dept)	수강학점 (cred)
1000	Mary	컴퓨터	15
2001	Floredo	SW	18
1001	하늘	컴퓨터	21
200	Tom	전기	9
1005	바다	컴퓨터	12

- 스키마 출처: 교재

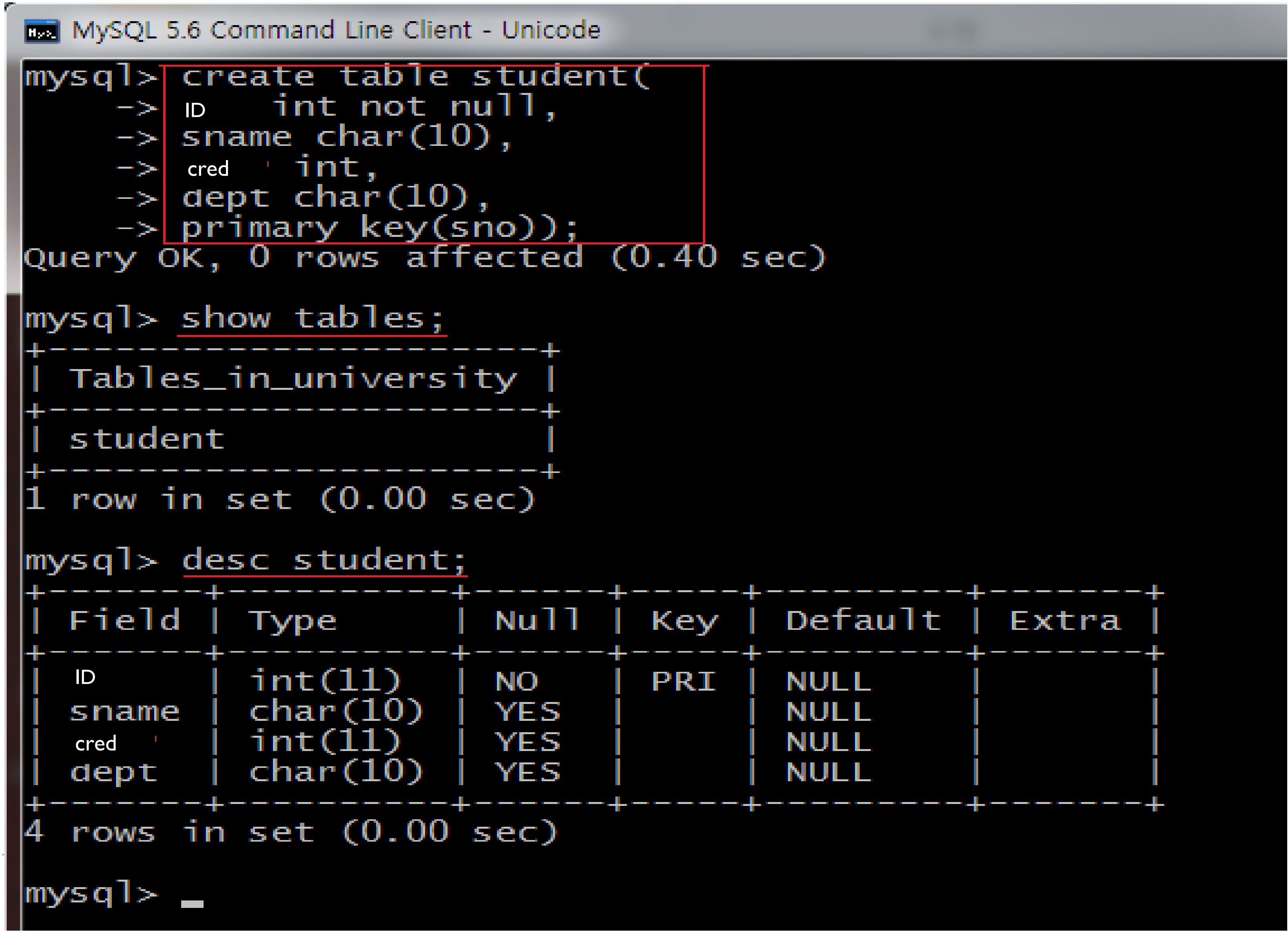
'University' 데이터베이스생성

- DBMS: MySQL 5.6



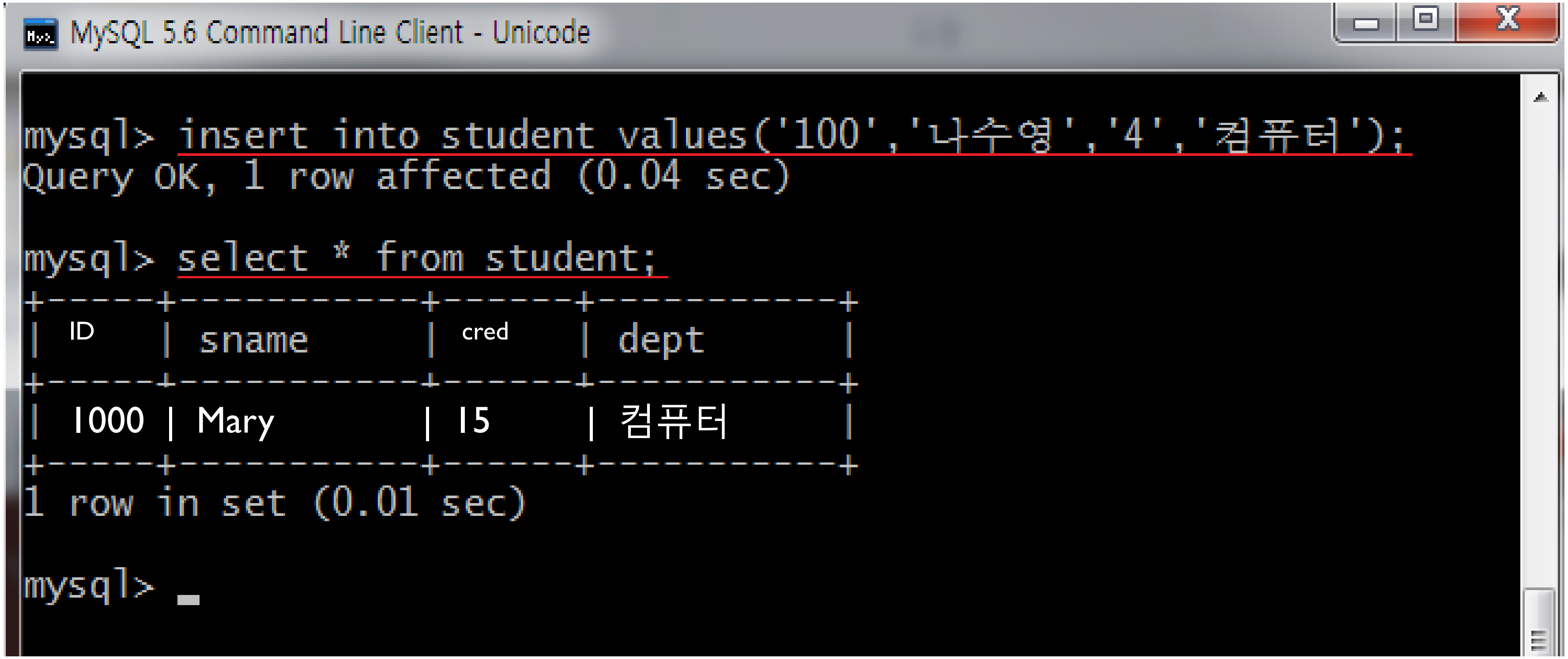
```
mysql> CREATE DATABASE IF NOT EXISTS University;
Query OK, 1 row affected, 1 warning (0.10 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| emp                |
| mysql              |
| performance_schema |
| sakila             |
| test               |
| university         |
| world              |
+--------------------+
8 rows in set (0.27 sec)

mysql> use university;
Database changed
mysql> _
```

‘university’ 데이터베이스에 ‘student’ 테이블 생성 및 확인



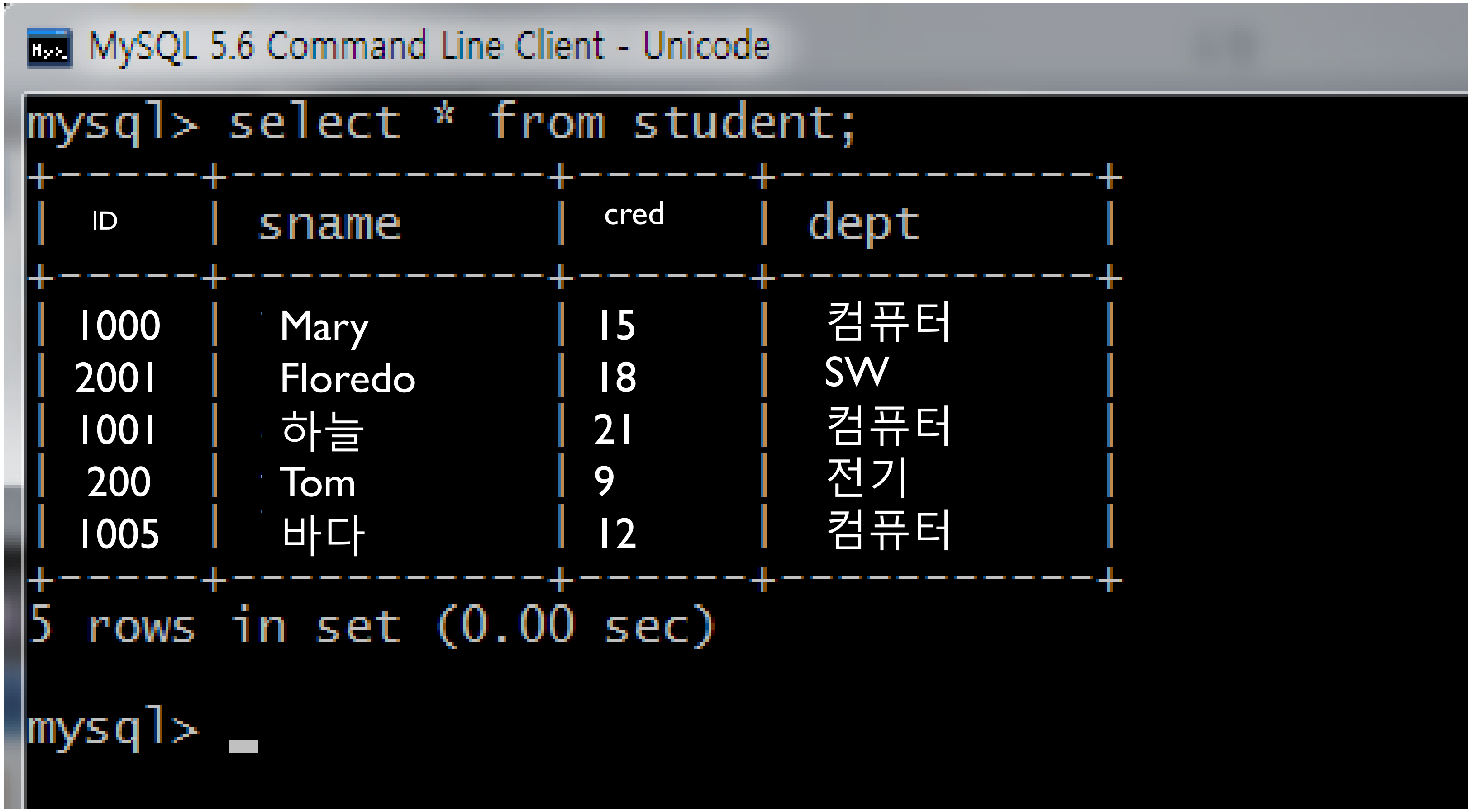
```
mysql> create table student(
    -> ID    int not null,
    -> sname char(10),
    -> cred    int,
    -> dept char(10),
    -> primary key(sno));
Query OK, 0 rows affected (0.40 sec)

mysql> show tables;
+-----------------------+
| Tables_in_university |
+-----------------------+
| student              |
+-----------------------+
1 row in set (0.00 sec)

mysql> desc student;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| ID    | int(11)  | NO   | PRI | NULL    |       |
| sname | char(10) | YES  |     | NULL    |       |
| cred  | int(11)  | YES  |     | NULL    |       |
| dept  | char(10) | YES  |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> _
```

‘student’ 테이블에 레코드 삽입 및 확인



```
mysql> insert into student values('100','나수영','4','컴퓨터');
Query OK, 1 row affected (0.04 sec)

mysql> select * from student;
+-----+-----------+------+-----------+
| ID  | sname     | cred | dept      |
+-----+-----------+------+-----------+
| 1000 | Mary     | 15   | 컴퓨터    |
+-----+-----------+------+-----------+
1 row in set (0.01 sec)

mysql> _
```

완성된 ‘student’ 테이블



```
mysql> select * from student;
+------+-------------+-------+------------+
|  ID  |  sname      |  cred |  dept      |
+------+-------------+-------+------------+
| 1000 |  Mary       |  15   |  컴퓨터    |
| 2001 |  Floredo    |  18   |  SW        |
| 1001 |  하늘       |  21   |  컴퓨터    |
|  200 |  Tom        |  9    |  전기      |
| 1005 |  바다       |  12   |  컴퓨터    |
+------+-------------+-------+------------+
5 rows in set (0.00 sec)

mysql> _
```

JDBC

JDBC 개발환경 구축

- 기본구축 준비사항
  - Eclipse(Java JDK)
  - MySQL 5.6

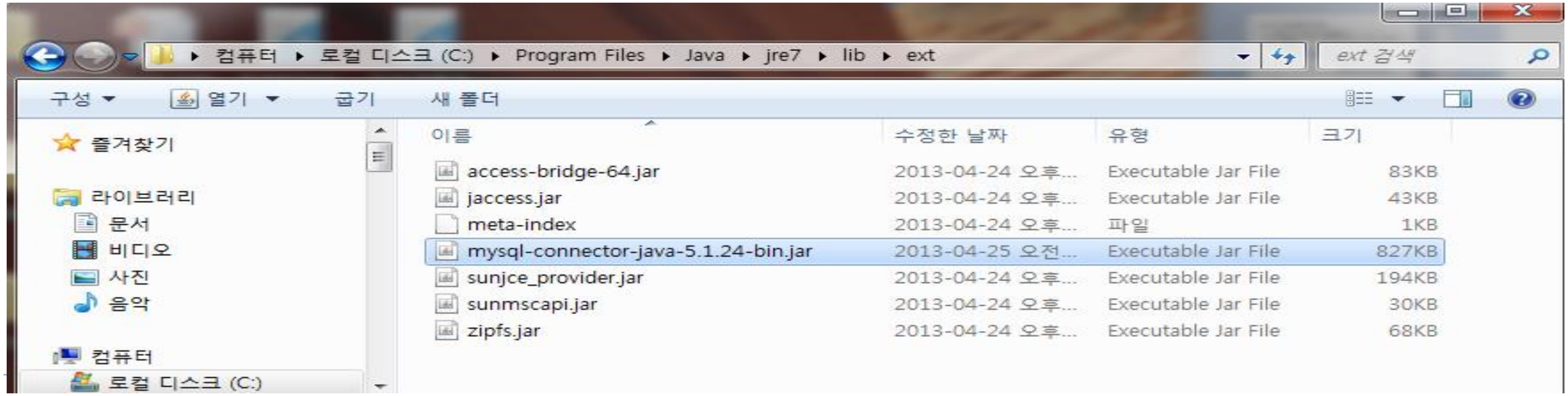
JDBC 개발환경 구축

JDBC 드라이버 설치 (MySQL기준)

- http://dev.mysql.com/downloads/connector/ 에서 "Connector/J" → Select Platform에서 Platform Independent 선택 후 다운
- 다운로드 받은 파일 압축을 풀고,

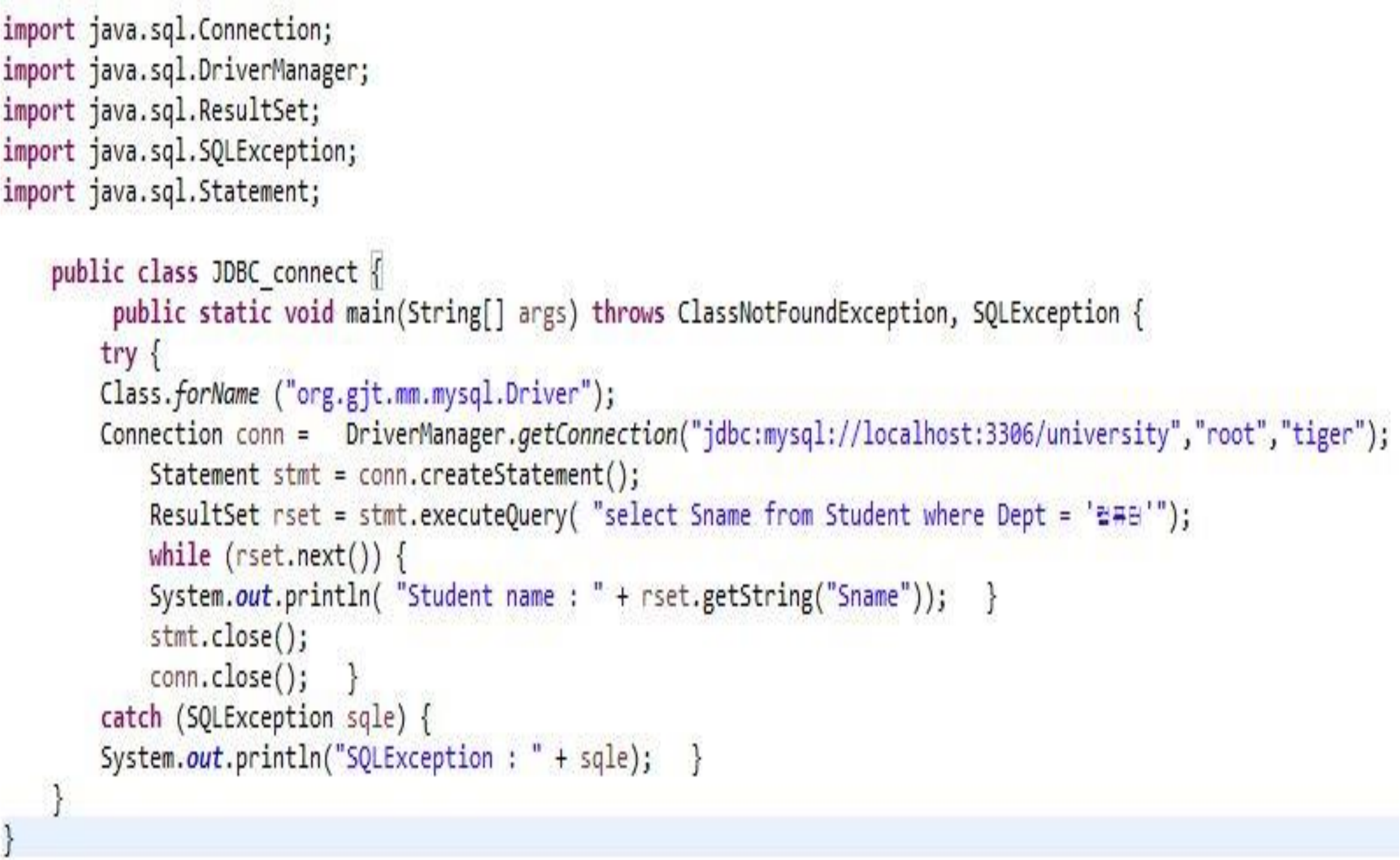
  Connector 파일 (예> mysql-connector-java-5.1.24-bin.jar) 을

  (내 컴퓨터의 자바가 설치된 디렉토리)\jre7\lib\ext\ 에 복사



이름	수정한 날짜	유형	크기
access-bridge-64.jar	2013-04-24 오후...	Executable Jar File	83KB
jaccess.jar	2013-04-24 오후...	Executable Jar File	43KB
meta-index	2013-04-24 오후...	파일	1KB
mysql-connector-java-5.1.24-bin.jar	2013-04-25 오전...	Executable Jar File	827KB
sunjce_provider.jar	2013-04-24 오후...	Executable Jar File	194KB
sunmscapi.jar	2013-04-24 오후...	Executable Jar File	30KB
zipfs.jar	2013-04-24 오후...	Executable Jar File	68KB

JDBC Code 예_1



```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

   public class JDBC_connect {
       public static void main(String[] args) throws ClassNotFoundException, SQLException {
      try {
      Class.forName ("org.gjt.mm.mysql.Driver");
      Connection conn =   DriverManager.getConnection("jdbc:mysql://localhost:3306/university","root","tiger");
         Statement stmt = conn.createStatement();
         ResultSet rset = stmt.executeQuery( "select Sname from Student where Dept = '컴퓨터'");
         while (rset.next()) {
         System.out.println( "Student name : " + rset.getString("Sname"));   }
         stmt.close();
         conn.close();   }
      catch (SQLException sqle) {
      System.out.println("SQLException : " + sqle);   }
   }
}
```

각종 JDBC 드라이버 설정정보

- Mysql
  - DRIVER : orj.gjt.mm.mysql.Driver
    URL : jdbc:mysql://localhost:3306/YOUR_DATABASE_NAME
- Oracle
  - DRIVER : oracle.jdbc.driver.OracleDriver
    URL : jdbc:oracle:thin:@localhost:1521:YOUR_SID
- Mssql
  - - JDBC-1.1 이상
    DRIVER : com.microsoft.sqlserver.jdbc.SQLServerDriver
    URL :
    jdbc:sqlserver://localhost:3433;databaseName=YOUR_DATABASE_NAME
  - - JDBC-1.0 이하
    DRIVER : com.microsoft.jdbc.sqlserver.SQLServerDriver
    URL :
    jdbc:microsoft:sqlserver://localhost:1433;databaseName=YOUR_DATABASE_NAME;selectMethod=cursor
  - - JK 드라이버
    DRIVER : com.jk.jdbc.Driver
    URL : jdbc:jk://localhost:1433/database=YOUR_DATABASE_NAME

각종 JDBC 드라이버 설정정보

DBMS별 설정코드정보

```
Class.forName (“org.gjt.mm.mysql.Driver”);
```

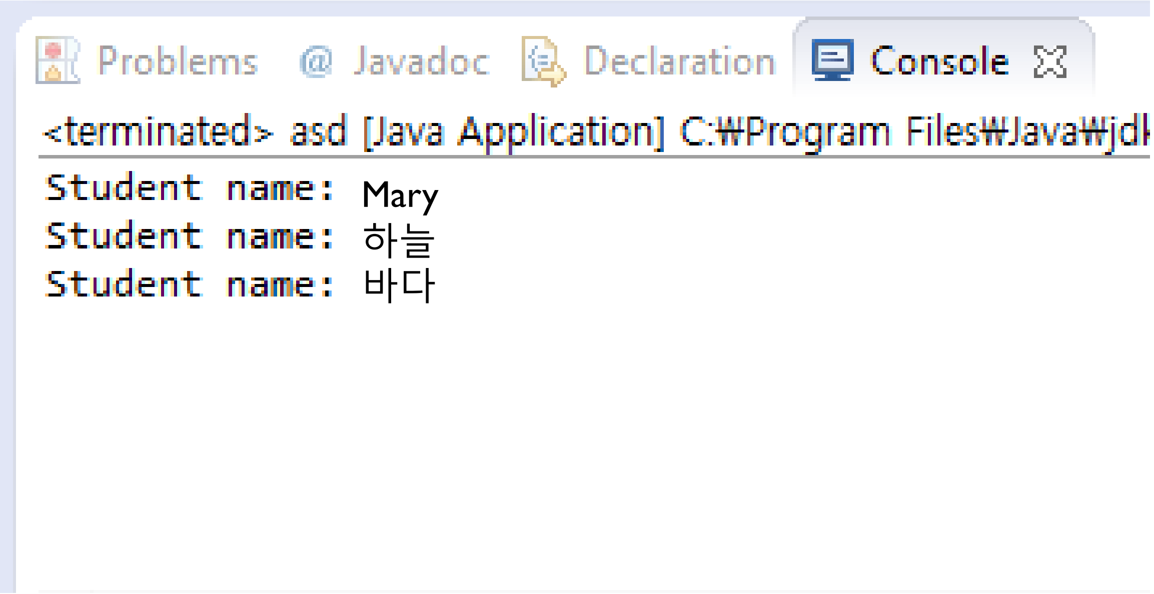
>>괄호 안 따옴표 부분에 DBMS의 DRIVER 정보를 넣어준다.

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/university",“root",“tiger");
```

>>괄호 안 따옴표 부분에 DBMS의 URL 정보를 넣어준다
>>(“URL”,”DBMS사용자아이디”,”DBMS사용자비밀번호”)

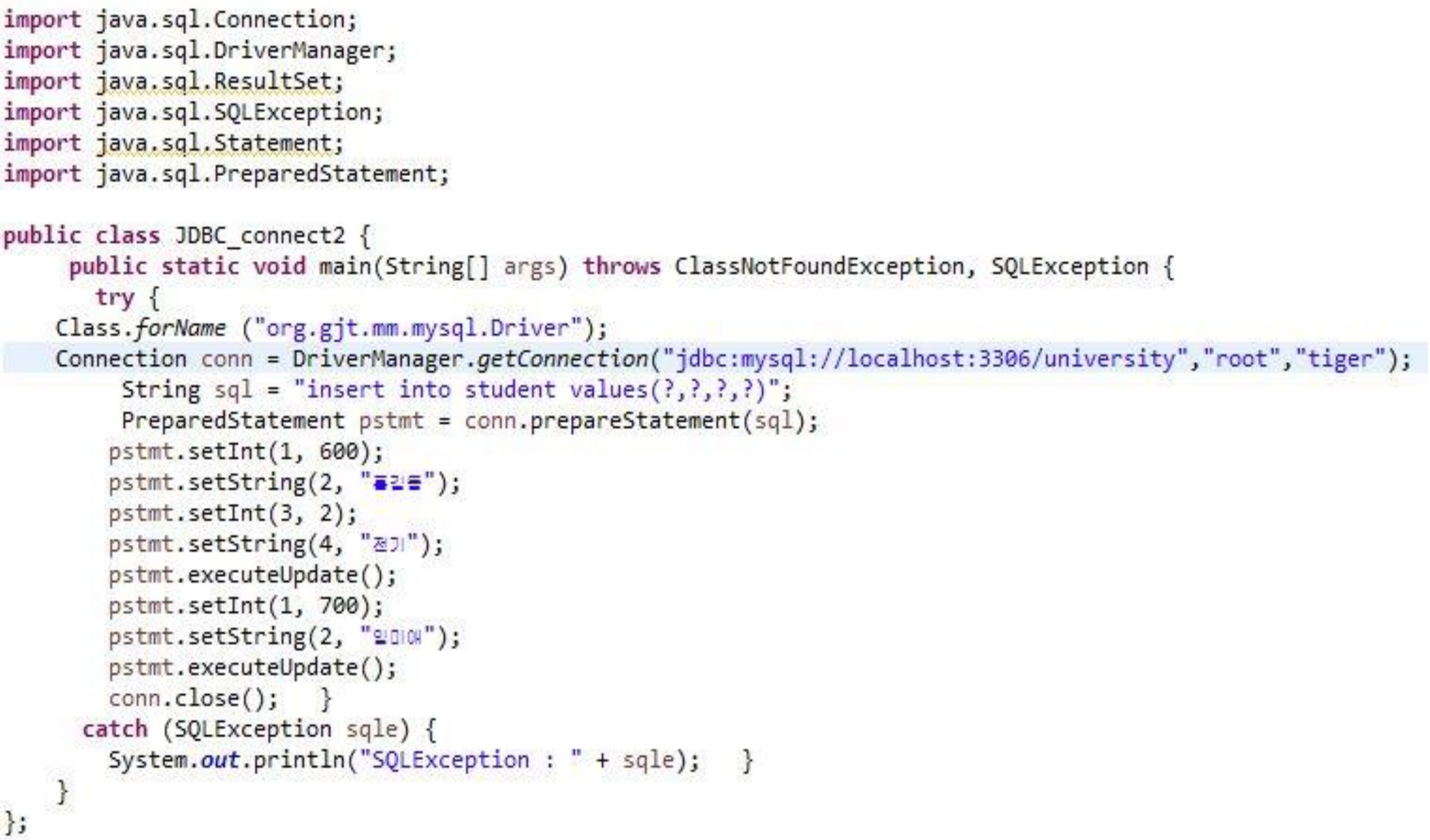
JDBC Code 예_1

- 코드 실행화면



```
<terminated> asd [Java Application] C:₩Program Files₩Java₩jdk
Student name: Mary
Student name: 하늘
Student name: 바다
```

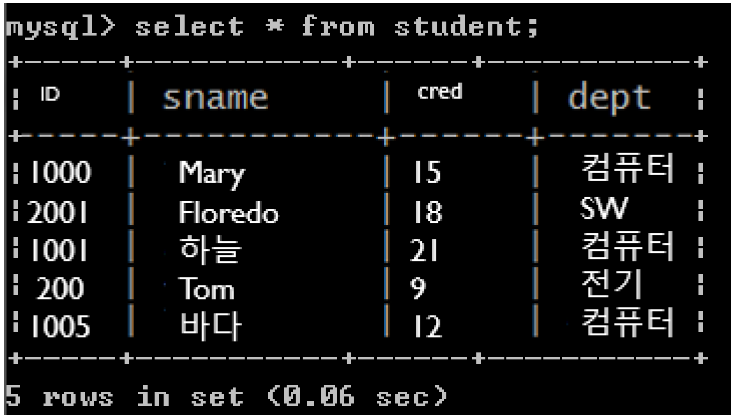
JDBC Code 예_2



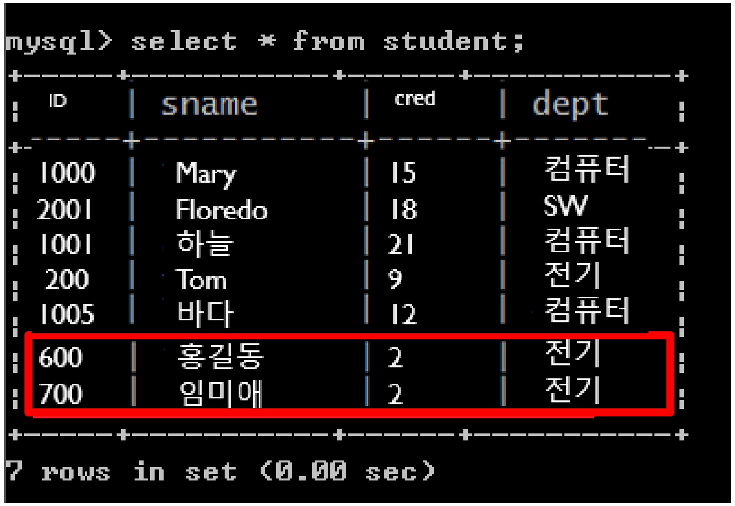
```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.PreparedStatement;

public class JDBC_connect2 {
     public static void main(String[] args) throws ClassNotFoundException, SQLException {
       try {
    Class.forName ("org.gjt.mm.mysql.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/university","root","tiger");
         String sql = "insert into student values(?,?,?,?)";
         PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, 600);
        pstmt.setString(2, "홍길동");
        pstmt.setInt(3, 2);
        pstmt.setString(4, "전기");
        pstmt.executeUpdate();
        pstmt.setInt(1, 700);
        pstmt.setString(2, "일미애");
        pstmt.executeUpdate();
        conn.close();    }
      catch (SQLException sqle) {
        System.out.println("SQLException : " + sqle);    }
    }
};
```

JDBC Code 예_2

- 코드 실행결과



```
mysql> select * from student;
+------+-------------+-------+-----------+
| ID   | sname       | cred  | dept      |
+------+-------------+-------+-----------+
| 1000 | Mary        | 15    | 컴퓨터    |
| 2001 | Floredo     | 18    | SW        |
| 1001 | 하늘        | 21    | 컴퓨터    |
| 200  | Tom         | 9     | 전기      |
| 1005 | 바다        | 12    | 컴퓨터    |
+------+-------------+-------+-----------+
5 rows in set (0.06 sec)
```



```
mysql> select * from student;
+------+-------------+-------+-----------+
| ID   | sname       | cred  | dept      |
+------+-------------+-------+-----------+
| 1000 | Mary        | 15    | 컴퓨터    |
| 2001 | Floredo     | 18    | SW        |
| 1001 | 하늘        | 21    | 컴퓨터    |
| 200  | Tom         | 9     | 전기      |
| 1005 | 바다        | 12    | 컴퓨터    |
| 600  | 홍길동      | 2     | 전기      |
| 700  | 임미애      | 2     | 전기      |
+------+-------------+-------+-----------+
7 rows in set (0.00 sec)
```

- Prepared statement를 통해 질의를 컴파일하여 저장한 후 인자 값을 바꾸어 가며 여러 번 수행 가능 (cf. dynamic SQL)

ODBC

ODBC 개발환경 구축

- 기본구축 준비사항
  - Microsoft Visual C++(32bit)
  - MySQL 5.6
  - MySQL Connector/ODBC

ODBC 개발환경 구축

1. MySQL Connector/ODBC 설치

- ODBC관리자는 32bit, 64bit 호환되지 않는다
- OS가 64bit이고 SQLConnect를 사용하는 프로그램 (Microsoft Visual C++)이 32bit로 build 되어 있다면 MySQL Connector/ODBC win32버전을 설치한다

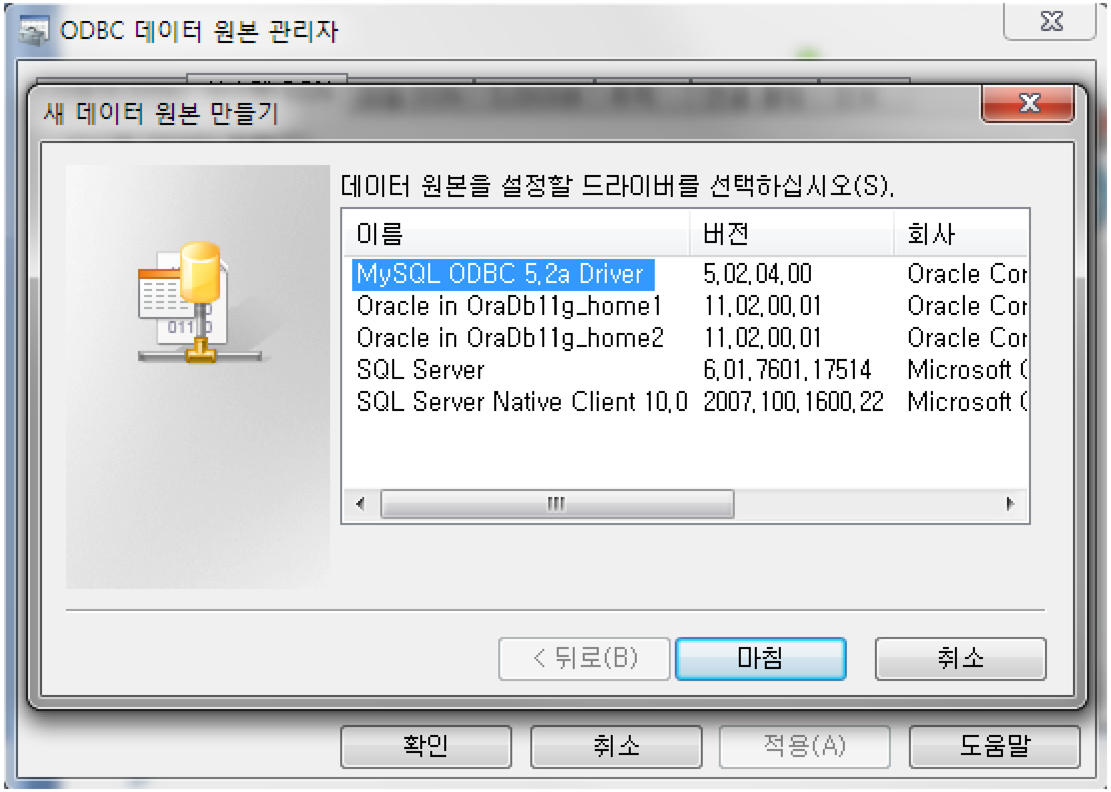
ODBC 개발환경 구축

2. 데이터 원본 (ODBC) 실행

- 32bit ODBC 관리자: C:\Windows\SysWOW64\odbcad32.exe
- 64bit ODBC 관리자: C:\Windows\System32\odbcad32.exe
- 64bit ODBC 관리자: 제어판-관리도구-데이타 원본(ODBC)

ODBC 개발환경 구축

3. 시스템 DSN - 추가 - MySQL ODBC Driver

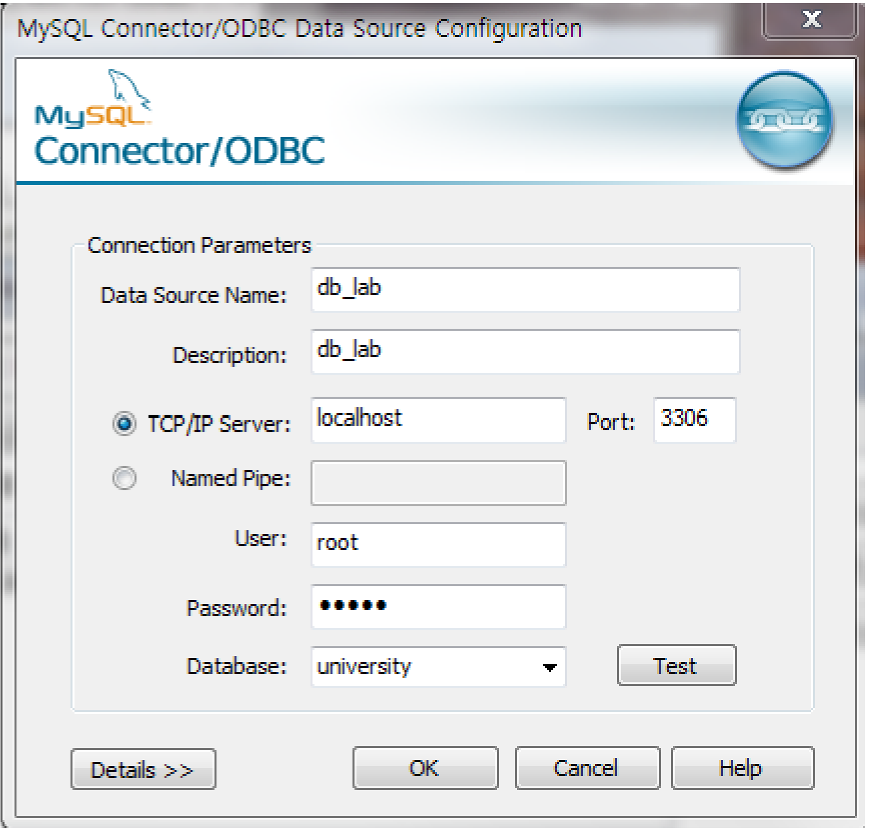


데이터 원본을 설정할 드라이버를 선택하십시오(S).

이름	버전	회사
MySQL ODBC 5.2a Driver	5.02.04.00	Oracle Cor
Oracle in OraDb11g_home1	11.02.00.01	Oracle Cor
Oracle in OraDb11g_home2	11.02.00.01	Oracle Cor
SQL Server	6.01.7601.17514	Microsoft (
SQL Server Native Client 10.0	2007.100.1600.22	Microsoft (

ODBC 개발환경 구축

- Data Source Name
  - MySQL DataBase 가 있는곳의 ip 기입
- Server
  - MySQL DataBase 가 있는곳의 ip 기입
- User
  - User 아이디
- Password
  - User 비밀번호
- Database
  - 접근 성공시 해당 MySQL에서 DB를 선택
  - Test버튼으로 연결 확인


MySQL Connector/ODBC Data Source Configuration

Connection Parameters

Data Source Name: db_lab

Description: db_lab

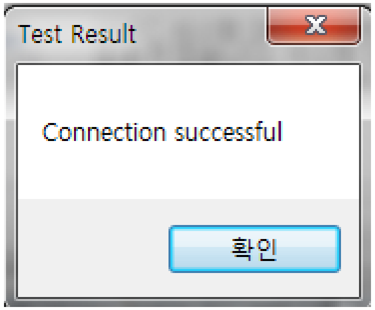
● TCP/IP Server: localhost Port: 3306

○ Named Pipe:

User: root

Password: •••••

Database: university


Test Result

Connection successful

ODBC Code 예

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <process.h>
#include <windows.h>

/* ODBC specific header file */
#include <sqlext.h>
#include <sql.h>

/* 핸들, 리턴코드 선언*/
SQLHENV henv;
SQLHDBC hdbc;
SQLHSTMT hstmt;
RETCODE _ret;

void main(){
SQLCHAR sname[50];
SQLINTEGER snamelen;

SQLAllocEnv (&henv);
SQLAllocConnect (henv, &hdbc);

_ret = SQLConnect(hdbc,(SQLCHAR *)“db_lab",SQL_NTS,(SQLCHAR *)“root",SQL_NTS,(SQLCHAR *)“tiger",SQL_NTS);
_ret = SQLAllocStmt(hdbc,&hstmt);
  {
   SQLCHAR select[]="SELECT sname FROM student WHERE dept = ‘컴퓨터’;";
_ret = SQLPrepare(hstmt,select,SQL_NTS);
_ret = SQLBindCol(hstmt,2,SQL_C_CHAR,(SQLPOINTER)sname,(SQLINTEGER)sizeof(sname),&snamelen);
_ret = SQLExecute(hstmt);
_ret = SQLFetch(hstmt);
printf("Sname\n");

while(_ret == SQL_SUCCESS){
_ret = SQLFetch(hstmt);
printf("%s\n ",sname);
  }
}
```

ODBC 개발환경 구축

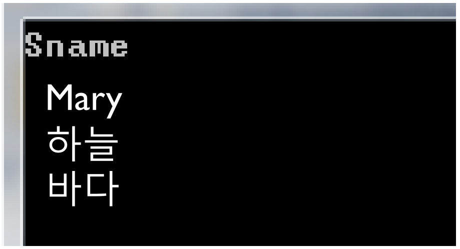
- DB계정 입력
  - _ret = SQLConnect(hdbc,(SQLCHAR*)“db_lab",SQL_NTS,(SQLCHAR*)“root",SQL_NTS,(SQLCHAR*)“tiger",SQL_NTS);

  - 소스코드 따옴표내의 빨간 글씨 부분에

    ODBD 환경 설정 때 입력한 Data Source Name, User, Password를 차례로 입력

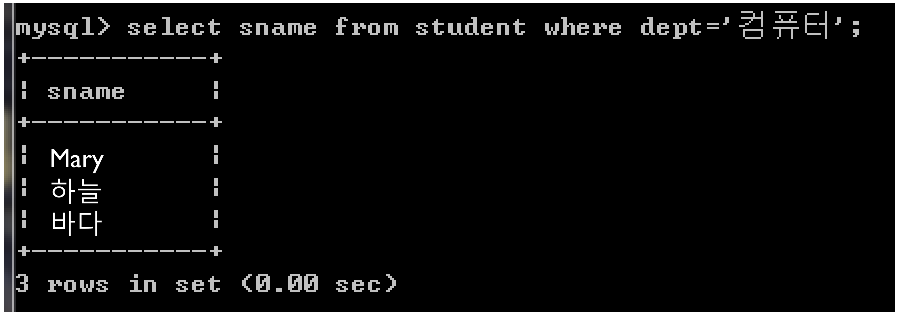
ODBC Code 예

- 코드 실행결과



```
Sname
 Mary
 하늘
 바다
```

- 대화식 SQL 도구에서 동일한 질의문실행결과

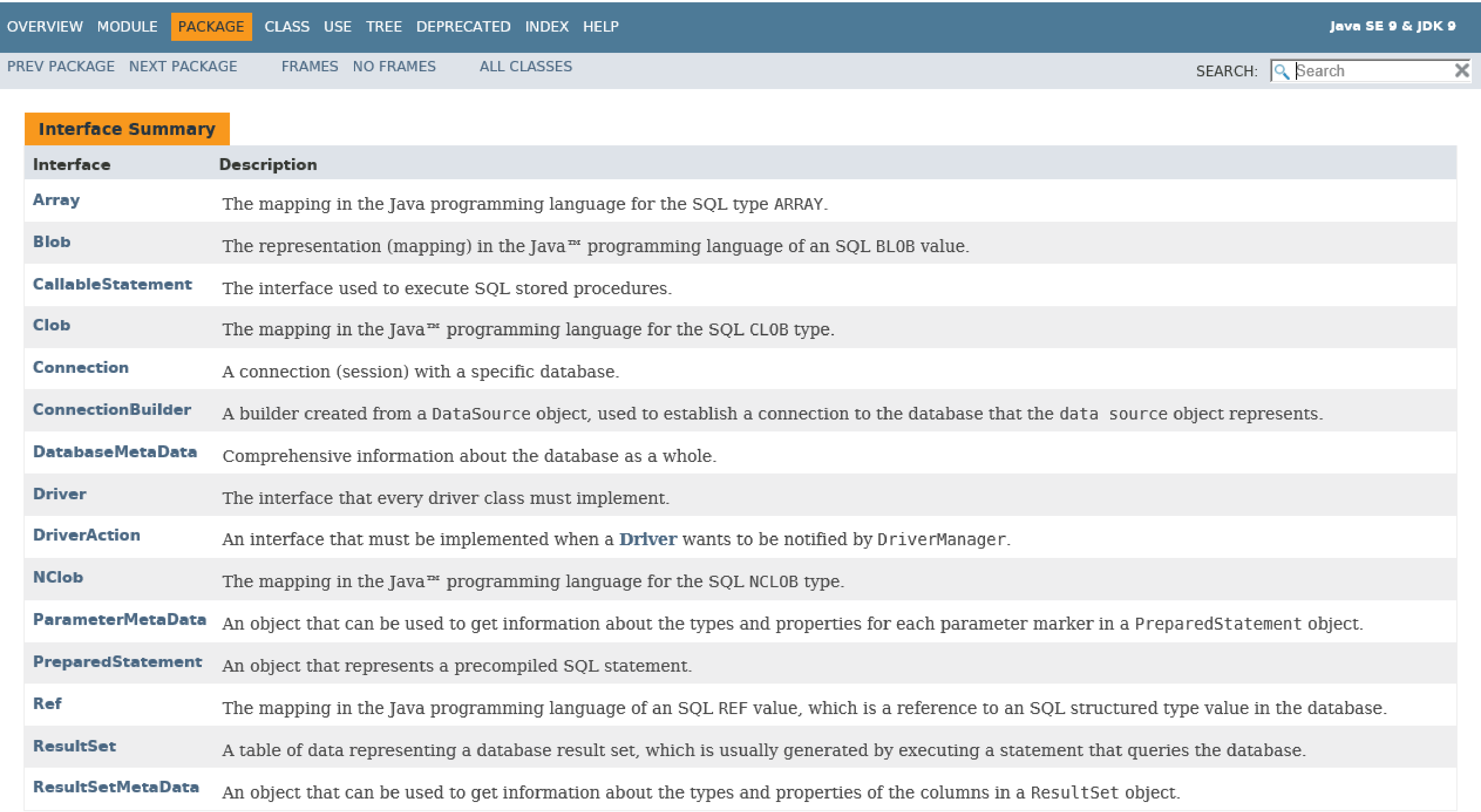


```
mysql> select sname from student where dept='컴퓨터';
+------------+
| sname      |
+------------+
| Mary       |
| 하늘       |
| 바다       |
+------------+
3 rows in set (0.00 sec)
```

JDBC and ODBC API On-line Reference Manual

JDBC 4.3 API : Programmers' Reference (Java SE 9)

- On-line Reference Manual (JDBC 4.3)
  - https://docs.oracle.com/javase/9/docs/api/java/sql/package-summary.html
  - interface summary: 찾고 싶은 API를 클릭
- JDBC Tutorial:
  - https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html
  - https://www.tutorialspoint.com/jdbc/index.htm

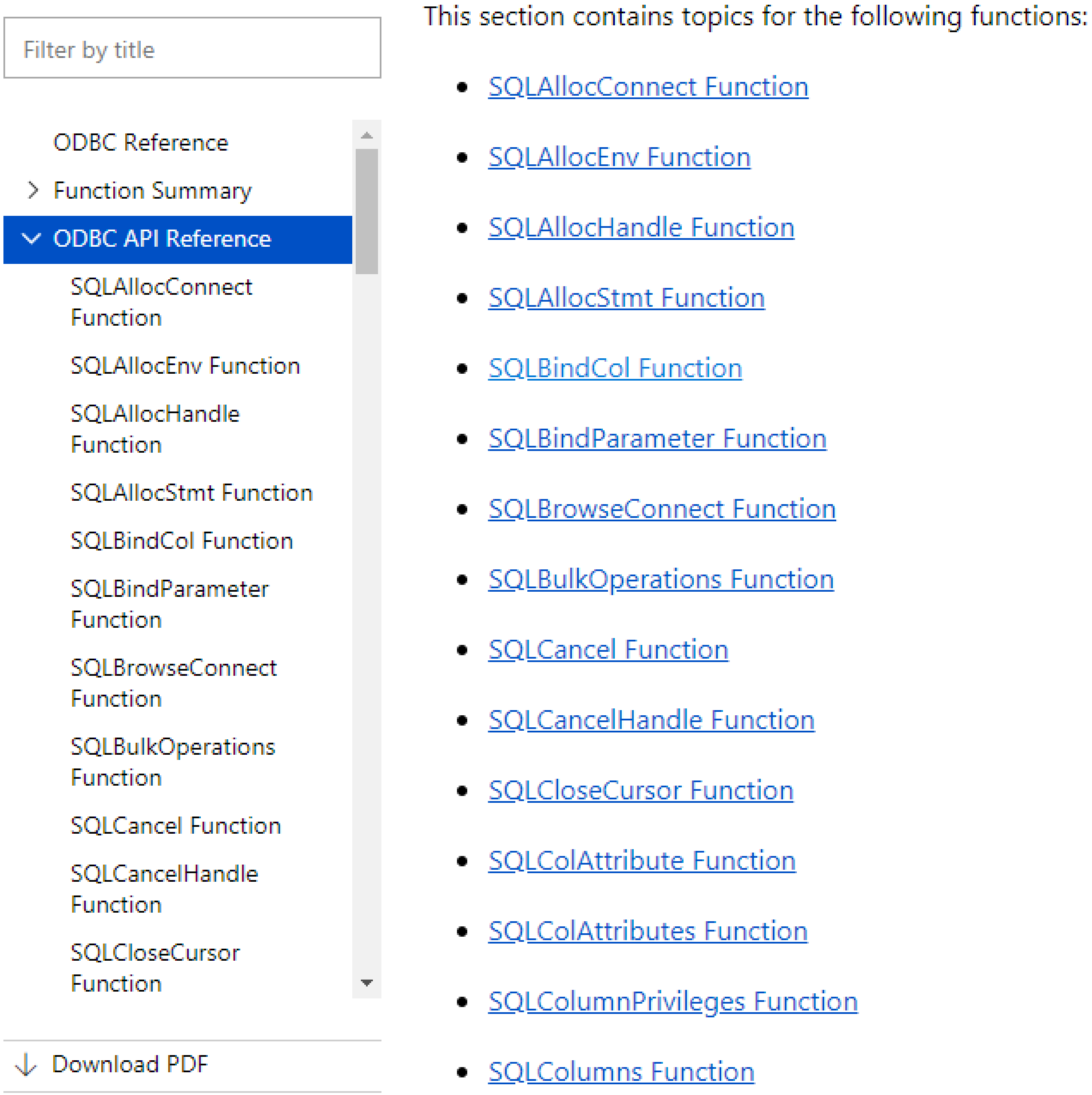


Interface Summary

Interface	Description
Array	The mapping in the Java programming language for the SQL type `ARRAY`.
Blob	The representation (mapping) in the Java™ programming language of an SQL `BLOB` value.
CallableStatement	The interface used to execute SQL stored procedures.
Clob	The mapping in the Java™ programming language for the SQL `CLOB` type.
Connection	A connection (session) with a specific database.
ConnectionBuilder	A builder created from a `DataSource` object, used to establish a connection to the database that the `data source` object represents.
DatabaseMetaData	Comprehensive information about the database as a whole.
Driver	The interface that every driver class must implement.
DriverAction	An interface that must be implemented when a **Driver** wants to be notified by `DriverManager`.
NClob	The mapping in the Java™ programming language for the SQL `NCLOB` type.
ParameterMetaData	An object that can be used to get information about the types and properties for each parameter marker in a `PreparedStatement` object.
PreparedStatement	An object that represents a precompiled SQL statement.
Ref	The mapping in the Java programming language of an SQL `REF` value, which is a reference to an SQL structured type value in the database.
ResultSet	A table of data representing a database result set, which is usually generated by executing a statement that queries the database.
ResultSetMetaData	An object that can be used to get information about the types and properties of the columns in a `ResultSet` object.

ODBC API 함수: Programmers' Reference Manual

- ODBC Programmer's Reference
  - https://docs.microsoft.com/en-us/sql/odbc/reference/odbc-programmer-s-reference
  - API reference 클릭
    - ODBC API reference 클릭
      - □ https://docs.microsoft.com/en-us/sql/odbc/reference/syntax/odbc-api-reference
      - □ 찾고 싶은 함수 클릭
- ODBC Tutorial
  - https://msdn.microsoft.com/en-us/library/thzzea08.aspx



This section contains topics for the following functions:

- SQLAllocConnect Function
- SQLAllocEnv Function
- SQLAllocHandle Function
- SQLAllocStmt Function
- SQLBindCol Function
- SQLBindParameter Function
- SQLBrowseConnect Function
- SQLBulkOperations Function
- SQLCancel Function
- SQLCancelHandle Function
- SQLCloseCursor Function
- SQLColAttribute Function
- SQLColAttributes Function
- SQLColumnPrivileges Function
- SQLColumns Function