The Korean Communications in Statistics Vol. 15, No. 1, 2008, pp. 51–64

R 언어를 통한 데이터베이스 접근†

심송용[1)], 강희모[2)], 이윤환[3)]

요 약

일반적으로 R 언어는 작은 크기의 자료분석에 적당하다. 이 연구에서 데이터베이스에 저장된 대용량의 자료를 R 언어를 통해서 접근하는 방법에 대해서 알아보았다. 실제 자료를 사용하여 MySQL, Oracle 또는 PostgreSQL 데이터베이스 서버에 저장된 자료에 접근하는 예를 제공하였다.

주요용어: DBMS; MySQL; ODBC; SQL; Oracle; PostgreSQL.

1. 서론

현대사회에서 컴퓨터와 네트워크의 발달은 많은 양의 정보를 발생시키고 있으며, 이러한 정보는 사회 각 분야에서 정책을 결정할 때 중요한 도구로 사용되고 있다. 많은 양의 정보는 거의 대부분 데이터베이스 서버에 저장되어 있기 때문에, 서버에서 정보를 이용하려면 데이터베이스 서버에 접근이 가능해야 한다.

R은 작은 자료를 처리하는 경우에는 효율적이나 대용량의 자료를 R의 내부에서 처리하는 것은 효율적이지 않다. R 언어에 대한 자세한 내용은 R Development Core Team (2007a), R 그래픽은 심송용 (2005), R을 이용한 통계분석은 Everitt와 Hothorn (2006), Dalgaard (2002) 등에 자세히 소개되어 있으나 여러 가지의 데이터베이스 서버들에서 관리되는 대용량 자료에 대한 접근에 대한 소개는 드물다.

본 연구에서는 R에서 데이터베이스에 저장된 자료를 처리하는 방법을 알아보기로 한다.

대부분의 응용프로그램에서 데이터베이스 엔진에 접속하기 위해서는 ODBC (Open Database Connectivity)를 사용한다 (그림 1.1 참조). ODBC는 데이터베

† 이 연구는 2007년 한림대학교 교비 연구비 HRF-2007-029에 의해 수행되었음.
1) (200-702) 강원도 춘천시 옥천동 1번지 한림대학교 정보통계학과, 교수.
Correspondence: sysim@hallym.ac.kr
2) (200-702) 강원도 춘천시 옥천동 1번지 한림대학교 정보통계학과, 전임강사.
3) (200-702) 강원도 춘천시 옥천동 1번지 한림대학교 정보통계학과, 박사과정.

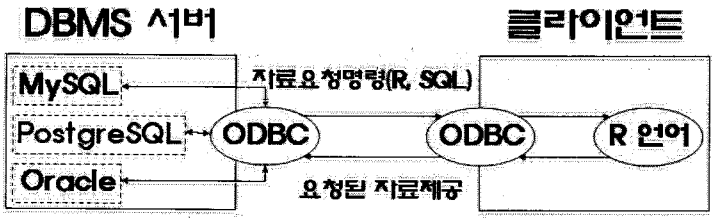


그림 1.1: ODBC를 이용한 DBMS 서버와 클라이언트

이스의 종류에 상관없이 데이터베이스에 접속하기 위한 application programming interface(API)를 제공해 주는 것으로 데이터베이스의 자료에 접근하기 위하여 프로그램을 작성할 때 데이터베이스와 독립적으로 작업할 수 있는 기능을 제공한다.

R에서도 데이터베이스 서버에 접속하려면 해당 데이터베이스 서버용 ODBC 드라이버를 설치해야 되며 그 과정은 각각의 데이터베이스 서버에 따라 조금씩 달라지므로 각 DBMS(Database Management System)를 R 언어에서 접근하는 방법을 소개할 때 설정 방법에 대해 논하기로 한다.

R은 버전 2.0.0 부터 RODBC 패키지를 제공하고 이 패키지를 이용하여 데이터베이스에 접근할 수 있을 뿐만 아니라 원격 데이터에도 접근이 가능하다.

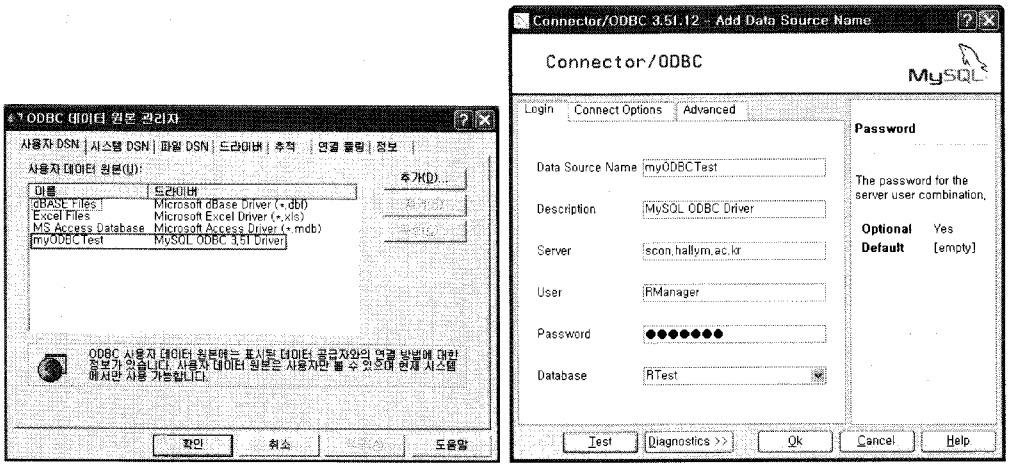
일반적으로 서버급 컴퓨터의 경우 유닉스 계열의 운영체계를 많이 사용하고 클라이언트 컴퓨터는 마이크로소프트의 윈도우 계열인 경우가 많으므로 본 연구에서는 DBMS 는 유닉스 계열의 서버 컴퓨터에서 운영되고, R은 윈도우 XP에 설치된 것을 가정하였다.

2. R과 MySQL

가장 많이 사용하는 오픈소스 계열의 데이터베이스 서버인 MySQL 서버에 저장된 자료를 R에서 사용하는 방법을 알아보기로 하자. MySQL은 멀티쓰레드, 멀티유저 SQL DBMS로서 전세계에서 11,000,000개 이상의 서버가 운영중이다 (http://www.mysql.com /why-mysql/).

2.1. 요구되는 프로그램

클라이언트 컴퓨터에 R이 설치되어 있고 DBMS로 원격지에 MySQL 서버가 설치되어 있는 환경을 가정한다. R에서 MySQL의 자료를 처리하려면 MySQL 서버 컴퓨터에 데이터베이스 서버(http://dev.mysql.com/)가 있어야 하며, MySQL



(a) ODBC 데이터 원본 관리자 (b) 사용자 DSN 추가

그림 2.1: ODBC 드라이버 설정

에 접속할 ODBC 드라이버가 클라이언트 컴퓨터에 설치되어야 하고(http://dev.mysql.com/) R에 RODBC 패키지가 설치되어야 한다 (R Development Core Team, 2007d). 이들에 대한 설치 및 설정에 대해서 알아보자.

2.2. 윈도우 XP에서 MySQL ODBC 드라이버 설치와 환경설정

ODBC 드라이버는 `http://dev.mysql.com/`에서 내려받은 압축 파일을 실행하면 자동으로 설치되고, 설치가 올바르게 되었는지는 제어판에서 **성능 및 유지관리** → **관리도구** → **데이터원본(ODBC)** 아이콘을 선택하면 그림 2.1(a)와 같은 ODBC 데이터 원본 관리자 창이 보이며 이 창의 **드라이버** 탭에서 MySQL ODBC 3.51 Driver 항목이 보이면 설치가 올바르게 된 것이다. 제어판이 **클래식 보기**인 경우에는 **성능 및 유지관리** 단계 없이 바로 **관리도구** 단계에서 시작한다.

드라이버의 설정은 그림 2.1(a)의 **사용자 DSN**(Data Source Name) 탭에서 **추가** 버튼을 클릭하면 **새 데이터 원본 만들기** 창이 생성되고, 여기서 추가하려는 드라이버 `MySQL ODBC 3.51 Driver`를 선택한 후 **마침**을 클릭한다. 그러면 그림 2.1(b)의 `Connector /ODBC` 창이 나타나고, 이 창에 데이터베이스 서버 접속에 필요한 여러 가지 정보를 올바르게 입력하면 새로운 **사용자 DSN**이 생성된다. 그림 2.1(b)의 첫 번째 `Login` 탭은 접속할 데이터베이스의 접속에 대한 정보를 입력하는 것으로 `Data Source Name`에는 적당한 이름을, `Description`에는 위 DSN의

설명을, Server에는 MySQL 서버가 설치된 컴퓨터 이름을, User에는 MySQL 서버에 등록된 사용자 ID를, Password에는 해당 ID의 암호를 입력한다. 생성된 DSN은 그림 2.1(a)의 사각형 안에서 확인할 수 있다.

두 번째 탭 Connection Options은 연결에 필요한 추가 옵션을 주는 곳으로 MySQL 포트(Port)나 소켓(Socket) 등을 설정하는데 기본값을 사용하는 경우 별도의 설정이 필요하지 않다. MySQL의 기본 포트 번호는 3306인데, 이외의 포트를 사용한 경우에는 Port Option에 해당 포트를 입력한다. 접속 설정에 대한 확인은 Test 버튼을 클릭하여 설정사항이 이상없으면 성공 메세지가 보이며 추가한 DSN으로 ODBC 접속에 사용할 수 있다.

2.3. MySQL에 저장된 자료 접근하기

ODBC 드라이버가 로컬 컴퓨터에 정상적으로 설치/설정되면 다음과 같이 R 언어에서 RODBC 패키지를 설치한다.

```
R> install.packages("RODBC")
R> library(RODBC)
```

이제 R에서 데이터 베이스에 연결할 수 있으며, R에서 MySQL에 접속하기 위한 함수는 odbcConnect(ODBC로 데이터베이스에 연결하는 함수), odbcClose(접속된 데이터베이스와 연결을 해제할 때 사용), odbcGetInfo(접속된 데이터베이스의 접속 정보 출력), odbcQuery(쿼리를 접속된 데이터베이스에 전송), sqlGetResults(odbcQuery 함수 결과를 DataFrame 객체로 가져옴), sqlQuery(odbcQuery 함수를 수행하고 sqlGetResults 함수를 사용하여 결과를 DataFrame 객체로 가져옴), odbcSetAutoCommit(연결하는 DBMS 의 autoCommit을 설정), sqlFetch(데이터베이스의 테이블인을 읽어 R의 DataFrame 객체로 가져옴), sqlCopy(쿼리문 수행 결과 값을 원격 테이블에 복사) 및 sqlSave(Data-Frame의 자료를 테이블명에 저장) 함수 등이 있다. 이들 함수의 자세한 사용법은 R Development Core Team (2007c)를 참조하자.

2.4. ODBC 연결을 위한 MySQL 서버 설정

다른 컴퓨터에서 ODBC로 데이터베이스 서버에 접속하려는 경우 보안문제 때문에 접속을 거부하도록 서버가 기본적으로 설정되어 있는 경우가 많다. 따라서 R에서 MySQL 서버에 접속하기 위한 MySQL과 방화벽 설정을 설정해야 한다.

먼저 MySQL이 다른 컴퓨터에서 접속할 수 있도록 접근을 허용하도록 한다. 접속 허용은 관리테이블인 db와 user에서 host 필드에 접속을 허가하는 데이터베

이스 사용자 ID와 데이터베이스 명을 '%'로 설정한다. 두 개의 관리테이블에서 host 명이 '%'이면 데이터베이스 사용자와 데이터베이스는 모든 컴퓨터에서 데이터베이스 서버로 접근할 수 있다. 다음은 MySQL에서 관리자로 접속 후 UPDATE 문을 사용하여 host 명을 변경하는 MySQL 명령문이다.

```
MySQL> UPDATE user SET host='%' WHERE user='사용자명';
MySQL> UPDATE db SET host='%' WHERE db='접속할 DB명';
MySQL> FLUSH PRIVILEGES;
```

다른 방법으로 GRANT 문을 사용하여 사용자를 추가할 수 있다.

다음으로 로컬 컴퓨터에서 데이터베이스 서버에 접속할 수 있도록 설정해야 될 사항은 방화벽을 해제해야 한다. 대부분의 운영체제는 보안 때문에 대부분의 포트를 외부에서 접속할 수 없도록 방화벽이 설정되어 있다. 따라서 접속을 원하는 DBMS의 포트에 대하여 방화벽 설정을 해제해야 데이터베이스 서버에 접속이 가능하다.

2.5. 적용사례

DBI, RODBC 패키지의 함수를 이용하여 적용사례를 실행해 보자. 적용사례에 이용한 자료는 Knowledge Discovery and Data Mining Conference 2004에서 공개된 자료 중 양자 물리학(quantum physics) 자료의 검증 자료(test data)이다. 이 자료는

http://kodiak.cs.cornell.edu/kddcup/

에서 사용자 등록 후 내려 받을 수 있다. 이 자료는 ID변수 한 개와 분류 변수 한 개, 일반 변수 78 개로 구성되었고, 자료의 개수는 100,000 개, 텍스트 파일에서 크기는 98.1MB이다. DBMS가 설치된 서버의 사양은 CPU가 펜티엄 Ⅲ 450MHz dual, 메모리 512MB, 운영체제는 Fedora 5 이고, 클라이언트 컴퓨터는 CPU가 펜티엄 Ⅳ 3GHz, 메모리 1GB, 운영체제는 Windows XP Professional 이다.

```
R> library(DBI); library(RODBC) ; hdb <- odbcConnect("myODBCTest")
R> unix.time(res <- sqlQuery(hdb, "select avg(attribute5),
   avg(attribute6) from rtest02"))
   user  system elapsed
   0.00    0.00    2.07
R> unix.time(res <- sqlQuery(hdb, "select avg(attribute5),
```

```
   avg(attribute6) from rtest02"))
   user  system elapsed
   0.00    0.00    0.45
R> res
  avg(attribute5) avg(attribute6)
1       0.1266546      0.05042849
```

위에서 1 줄의 myODBCTest는 그림 2에서 추가한 DSN이고, 2–3 줄과 6–7 줄은 데이터베이스 서버에 접속하여 R에서 결과가 출력되기까지 소요시간(unix.time)을 계산하였다. 최초 실행시간은 2.07 초(5 줄)로 나타났고, 그 이후 DBMS에서 동일한 명령을 실행하면 그 내용이 캐시(cache) 메모리에 저장되어 있기 때문에 실행시간이 9 줄과 같이 0.45 초로 현저하게 줄어든다. 적용사례에 사용한 MySQL 버전은 5.0.27이고, 이 DBMS에 설정된 캐시의 용량은 기본값이 1MB이다.

3. R과 PostgreSQL

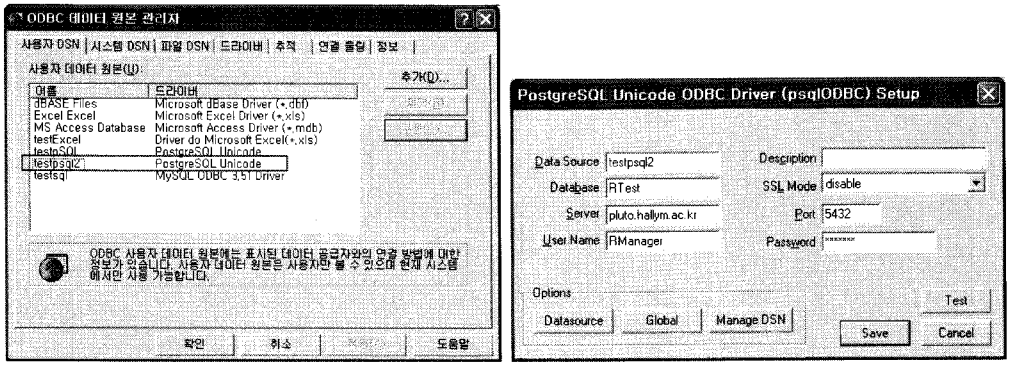
R에서 PostgreSQL 서버의 자료를 처리하는 방법에 대해서 알아보자. PostgreSQL은 PostgreSQL Global Development Group (2007)에서 제공하고 있으며 MySQL과 마찬가지로 무료로 사용할 수 있으며 최신 버전은 8.2.4이다.

3.1. 요구되는 프로그램

R에서 PostgreSQL 데이터베이스 서버에 접속하려면 MySQL 데이터베이스 서버에 접속할 때와 같이 PostgreSQL 데이터베이스 서버(http://www.postgresql.org/ftp/ win32/), R에서 DBI(R Development Core Team, 2007b)와 RODBC(R Development Core Team, 2007c) 패키지, 클라이언트 컴퓨터에 ODBC 드라이버가 설치(http://postgresql. org/ftp/odbc/versions/msi)가 요구된다.

3.2. 윈도우 XP에서 PostgreSQL ODBC 설치와 환경설정

윈도우 XP용 PostgreSQL ODBC 드라이버는 http://postgresql.org/ftp/odbc/versions/msi에서 psqlodbc-*version*.zip으로 되어 있는 압축파일 중에서 PostgreSQL 서버의 버전과 *version*이 일치되는 파일의 압축을 해제하여 psqlodbc.msi 파일을 실행하면 자동으로 설치된다. ODBC 드라이버 설치 확인은 2.2 절과 같이 새 데이터 원본 만들기 창에서 PostgreSQL Unicode와 Postgre SQL ANSI를 목록에서 확인한다. PostgreSQL DSN 설정은 그림 3.1(a)에서 추가 버튼



(a) ODBC 데이터 원본 관리자 (b) 사용자 DSN 추가

그림 3.1: PostgreSQL ODBC 드라이버 설정

을 클릭하면 새 데이터 원본 만들기 창이 열리고, 여기에 보이는 목록 중 추가하려는 드라이버 PostgreSQL Unicode와 PostgreSQL ANSI에서 하나를 선택한다. 그림 3.1(b)에서 Data Source에는 DSN 입력을, Description에는 DSN에 대한 간단한 설명을, Database에는 데이터베이스 이름을, Server에는 데이터베이스가 설치된 서버 컴퓨터 이름 또는 IP 주소를, Port에는 데이터베이스에 접속할 포트로 기본값은 5432를, User Name에는 데이터베이스 사용자 ID를, Password에는 해당 ID의 암호를 입력하면 DSN이 추가되며 그림 3.1(a)의 사각형에는 이와 같은 방법으로 추가된 DSN 목록인 testpsql2가 있음을 볼 수 있다.

3.3. ODBC 연결을 위한 PostgreSQL 서버 설정

데이터베이스 서버가 설치된 리눅스 운영체제는 외부에서 불특정 다수의 접근을 허락하지 않기 위하여 방화벽을 설정하였으며 데이터베이스 서버 자체적으로 외부에서 접근에 대한 설정파일이 존재한다. 클라이언트 컴퓨터에서 PostgreSQL에 접속을 허가하려면 방화벽 설정을 해제하고, 두 개의 설정파일 내용을 변경해야 한다. 설정 파일은 다음과 같다.

- pg_hba.conf 설정 파일은 서버로의 모든 접근을 관리하는 파일로 기본값이 외부 컴퓨터에서 접속할 수 없도록 설정되어 있다. 따라서 외부에서 접속하려면 다음과 같은 IPv4 local connection 설정

```
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD
# IPv4 local connections:
```

```
host    all         all         127.0.0.1/32          trust
```

의 CIDR-ADDRESS를

```
host    all         all         0.0.0.0/0             trust
```

로 변경하면 모든 클라이언트 컴퓨터에서 접속을 할 수 있다. 여기서 각 설정은 TYPE는 접속 타입 설정, DATABASE는 데이터베이스 이름 설정, USER는 사용자 이름 설정, CIDR-ADDRESS는 접속을 허가하는 IP 설정, METHOD는 패스워드를 전송할 때 암호화 형태 설정이다.

- postgresql.conf 파일의 설정: pg_hba.conf 파일은 외부에서 접속을 할 수 있도록 설정하는 것이고, postgresql.conf 파일은 PostgreSQL 서버 자체의 접속에 대한 설정으로 외부 컴퓨터로부터 요청을 받을 수 있도록 한다. 변경할 부분은 listen_addresses의 값을 특정한 호스트로 바꾸거나 모든 호스트(*)로 바꾸는 것이다. 이 값은 최초에 'localhost'로 설정되어 있다. 이 값을 변경하면 외부의 모든 컴퓨터에서 접속 요청을 받아 서버에 접속할 수 있다. 만일 특정한 컴퓨터에서만 접속을 허가한다면 그 컴퓨터의 IP 주소를 입력하고, 설정할 IP가 여러 개이면 쉼표(,)로 구분하여 입력한다.

3.4. 적용사례

R에서 DBI, RODBC 패키지의 함수를 이용하여 PostgreSQL 서버에 접속하는 적용사례를 실행해 보자. 적용사례에 사용한 자료와 서버와 클라이언트 컴퓨터의 사양은 MySQL에서 제시한 것과 동일하다.

```
R> library("DBI"); library("RODBC"); con <- odbcConnect("testpSQL2")
R> unix.time(res <- sqlQuery(hdb, "select avg(attribute5),
   avg(attribute6) from rtest02"))
   user  system elapsed
   0.02    0.01    5.81
R> unix.time(res <- sqlQuery(hdb, "select avg(attribute5),
   avg(attribute6) from rtest02"))
   user  system elapsed
   0.02    0.01    1.50
R> res
        avg        avg
1 0.1266546 0.05042849
```

위에서 1 줄의 testpSQL2는 클라이언트에 설정된 DSN이고, 2 – 3 줄과 6 – 7 줄은 MySQL의 적용사례와 동일하게 DBMS에 접속하여 명령문 실행결과가 나오기까지 소요시간을 unix.time으로 출력하였다. 출력결과 처음 명령문 소요시간은 5.81 초(5 줄)이고, 이후 다시 실행한 경우(9 줄) 캐시와 관련있기 때문에 실행시간은 1.50 초가 소요되었다. 명령문 실행결과는 res(2, 6줄)에 저장하였고, 그 결과는 10 – 12 줄에 출력하였다. 여기서 사용한 DBMS는 postgresql-8.1.8-1이다.

4. R과 Oracle

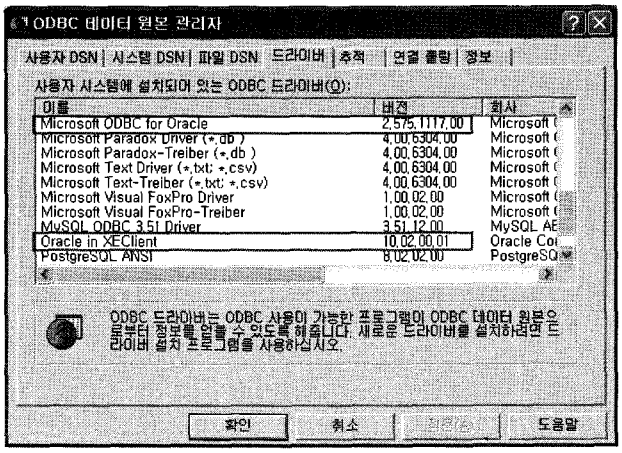
오라클(Oracle)은 미국 오라클사의 관계형 데이터베이스 관리 시스템(RDBS)의 이름으로 현재 가장 널리 사용되는 대표적인 제품의 하나이다. 현재 오라클의 최신 버전은 Oracle Database 10g로 그 종류는 네 종류가 있으며, 각 종류마다 사용할 수 있는 기능에 차이가 있다.

4.1. 요구되는 프로그램

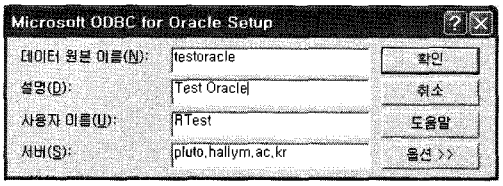
R에서 Oracle 데이터베이스 서버에 접속하려면 MySQL 데이터베이스 서버에 접속할 때와 같이 Oracle 데이터베이스 서버(http://oracle.com), R에 DBI(R Development Core Team, 2007b) 와 RODBC(R Development Core Team, 2007c) 패키지, 클라이언트 컴퓨터에 ODBC 드라이버의 설치 (http://www.oracle.com/technology/software/tech/windows/odbc/index.html, http://www.oracle.com/technology/software/products/database/xe/htdocs/102xewinsoft.html)가 요구된다.

4.2. 윈도우 XP에서 Oracle ODBC 드라이버 설치와 설정

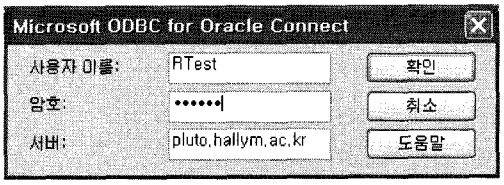
오라클 ODBC 드라이버는 http://www.oracle.com/technology/software/tech/windows/odbc/index.html에서 ODBC 드라이버를 얻을 수 있는데 이 드라이버는 오라클 데이터베이스 서버 standard edition 이상의 설치 CD로 설치할 수 있다. ODBC 드라이버만 설치할 경우 http://www.oracle.com/technology/software/products/database/xe/htdocs/102xewinsoft.html에서 OracleXEClient.exe을 실행하면 드라이버가 자동으로 설치된다. 이 경우 두 종류의 드라이버가 설치되는데 하나는 Microsoft ODBC for Oracle이며 나머지 하나는 Oracle in XEClient이다. 이 오라클 ODBC 드라이버들은 그림 4.1(a)의 목록에서 확인할 수 있으며, 모두 Oracle 접속에 사용할 수 있다.



(a) ODBC 데이터 원본 관리자



(b) 사용자 DSN 추가



(c) 사용자 DSN 정보 확인

그림 4.1: 오라클 ODBC 연결

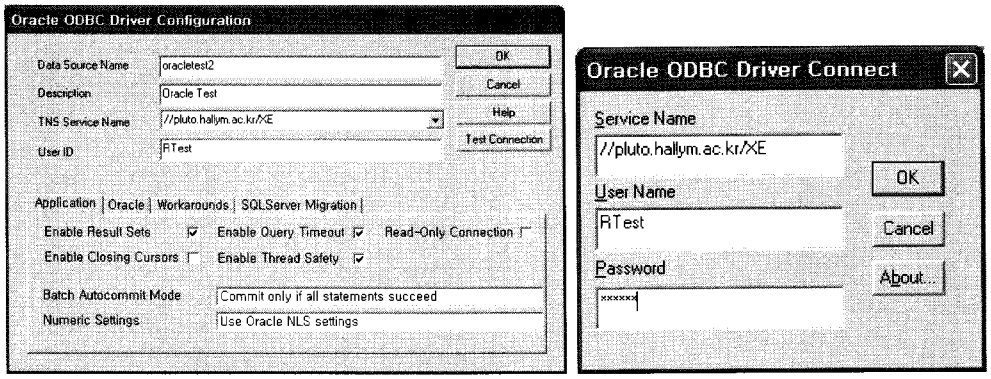
4.2.1. Microsoft ODBC DSN 설정

Microsoft ODBC for Oracle 드라이버의 DSN 생성은 그림 4.1(a)의 **사용자 DSN** 탭 또는 **시스템 DSN** 탭에서 **추가** 버튼을 클릭하면 데이터 원본을 설정할 드라이버 목록이 **새 데이터 원본 만들기** 창에 보인다. 이 목록들 중 Mirosoft ODBC for Oracle을 선택하고 **마침** 버튼을 클릭하면 사용자 DSN의 정보를 입력하는 그림 4.1(b) 창이 나타난다. 이 창에서 입력되는 값으로 **데이터 원본 이름**에는 DSN을, **설명**에는 DSN의 간략한 설명을, **사용자 이름**에는 접속할 오라클 서버의 사용자 ID를, **서버**에는 접속할 오라클 서버를 입력한다.

DSN이 설정되었으면 R을 실행하여 오라클 서버의 데이터에 접근하자. 먼저 R을 실행하고 다음과 같이 입력하면

```
R> library("DBI"); library("RODBC"); con <- odbcConnect("testoracle")
```

그림 4.1(c)의 대화창이 생성되면서 접속할 Oracle 서버가 설치된 호스트와 DBMS의 ID에 해당하는 암호를 요구한다. 이 창에는 그림 4.1(b)에서 설정한 암호를 입



(a) 오라클 ODBC DSN 추가 (b) 오라클 ODBC 연결 설정

그림 4.2: 오라클 ODBC 연결

력한다. 요구되는 정보가 올바르게 입력되었으면 **확인** 버튼을 클릭할 때, R 콘솔 화면에 아무런 메세지가 출력되지 않으며, R에서 오라클 데이터베이스 서버에 접속이 성공한 것이다. 접속이 실패할 경우 에러 메세지가 출력된다.

4.2.2. XEClient DSN 설정

Oracle in XEClient 드라이버의 DSN 생성은 그림 4.1(a)의 **사용자 DSN** 탭 또는 **시스템 DSN** 탭에서 **추가** 버튼을 클릭하여 생성한다. 이 그림에서 보는 것과 같이 데이터 원본을 설정할 드라이버 목록이 **새 데이터 원본 만들기** 창에 나타나는데 이 목록에서 Oracle in XEClient를 선택하고 **마침** 버튼을 클릭하면 사용자 DSN의 정보를 입력하는 그림 4.2(a) 창이 나타난다. 이 창에서 입력되는 값으로 Data Source Name에는 DSN 명을, Description에는 DSN의 간략한 설명을, TNS Service Name에는 접속할 오라클 서버와 데이터베이스를, User ID에는 접속할 오라클 서버의 사용자 ID를 입력한다. 설정이 올바르게 되었는지 확인은 Test Connection 버튼을 클릭하면, 연결설정 성공화면이 나타난다.

DSN이 설정되었으면 R을 실행하여 오라클 서버의 데이터에 접근하자. 먼저 R을 실행하고 다음과 같이 입력하면

```
R> library("DBI"); library("RODBC"); con <- odbcConnect("testoracle")
```

그림 4.2(b)의 창이 나타나며 **암호**를 제외하고 그림 4.2(a)에서 설정된 값을 가져온다. 여기에 암호를 입력하고 **확인** 버튼을 클릭할 때, 정보가 올바르게 입력하였

으면 R 콘솔 화면에 아무런 메세지가 출력되지 않으며, R에서 오라클 데이터베이스 서버에 접속이 성공한 것이다.

4.3. 적용사례

R에서 DBI, RODBC 두 패키지의 함수를 사용하여 Oracle에 대한 적용사례를 실행해 보자. 적용사례에서 사용하는 자료와 서버와 클라이언트 컴퓨터의 사양은 MySQL과 PostgreSQL에서 제시한 것과 동일하다.

```
R> library("DBI"); library("RODBC"); con <- odbcConnect("testoracle")
R> unix.time(res <- sqlQuery(hdb, "select avg(attribute5),
   avg(attribute6) from rtest02"))
   user  system elapsed
   0.00    0.00    2.45
R> unix.time(res <- sqlQuery(hdb, "select avg(attribute5),
   avg(attribute6) from rtest02"))
   user  system elapsed
   0.00    0.00    0.68
R> res
  AVG(ATTRIBUTE5) AVG(ATTRIBUTE6)
1       0.1266546      0.05042849
```

1 줄에서 con은 클라이언트에 설정된 DSN으로 DBMS에 접속을 시도하고, 2 – 3 줄과 6 – 7 줄은 R에서 DBMS에 접속하여 명령문이 모두 실행될때까지 소요시간을 R의 unix.time로 출력하였다. 초기 실행시간은 2.45 초(5 줄)가 소요되었고, 이후 동일한 명령문을 DBMS에서 실행한 경우 0.68 초(9 줄)가 소요되었다. 위의 세 가지 DBMS로 특정한 명령어를 실행한 결과 소요시간은 MySQL, Oracle, PostgreSQL 순으로 빠르게 나타났다. res(2, 6줄)는 명령문 실행결과를 저장하였고, 그 결과는 10 – 12 줄에서 확인할 수 있다. 적용사례에 사용한 DBMS는 oracle-xe-univ-10.2.0.1-1.0이다.

5. 결언

R은 선형 비선형 모델링, 시계열 분석, 분류, 다변량분석 등의 다양한 통계기법, 화려한 그래픽 기법 그리고 고도의 확장성을 제공할 뿐 아니라 최신 통계이론이 R로 구현되어 제공된다. 그리고 R은 무료 소프트웨어이기 때문에 누구나 자유롭게 사용할 수 있으며, 다양한 운영체제에서도 사용할 수 있는 장점을 가지고 있다. 또한 다양한 아스키, 바이너리 데이터 파일을 자유롭게 읽을 수 있고, 특히 얼

마전까지도 제공하지 않았던 데이터베이스 서버에 접속하는 기능이 추가되어 대용량 자료에 대한 처리가 가능해졌다. 이와 같이 R 언어는 SAS, SPSS 등과 같은 상업용 통계프로그램의 주된 기능을 대부분 수행할 수 있을 뿐 아니라 상용 프로그램이 제공하지 못하는 최신 이론을 접할 수 있기 때문에 교육용, 연구용 뿐만 아니라 기업용 통계분석 프로그램으로 사용하는데도 어려움이 없다.

참고문헌

심송용 (2005). <통계그래픽스>, 교우사, 서울.

Dalgaard, P. (2002). *Introductory Statistics with R*, Springer, New York.

Everitt, B. S. and Hothorn, T. (2006). *A Handbook of Statistical Analysis Using R*, Chapman & Hall/CRC, New York.

PostgreSQL Global Development Group (2007), `http://www.postgresql.org/`.

R Development Core Team (2007a). *R: A Language and Environment for Statistical Computing.* ISBN 3-900051-07-0, Vienna, Austria, `http://www.R-project.org`.

R Development Core Team (2007b). *R: A Language and Environment for Statistical Computing.* ISBN 3-900051-07-0, Vienna, Austria, `http://cran.r-proect.org/doc/packages/DBI.pdf`.

R Development Core Team (2007c). *R: A Language and Environment for Statistical Computing.* ISBN 3-900051-07-0, Vienna, Austria, `http://cran.r-project.org/doc/packages/RODBC.pdf`.

R Development Core Team (2007d). *R: A Language and Environment for Statistical Computing.* ISBN 3-900051-07-0, Vienna, Austria, `http://cran.r-project.org/src /contrib/Descriptions/RODBC.html`.

`http://dev.mysql.com/`.

`http://kodiak.cs.cornell.edu/kddcup/`.

`http://www.mysql.com/why-mysql/`.

`http://www.postgresql.org/ftp/odbc/versions/msi/`.

`http://www.postgresql.org/ftp/win32/`.

`http://www.oracle.com`.

`http://www.oracle.com/technology/software/tech/windows/odbc/index.html`.

`http://www.oracle.com/technology/software/products/database/xe/htdocs/102xewinsoft.html`.

[2007년 7월 접수, 2007년 10월 채택]

Access to Databases through the R-Language†

Songyong Sim[1)], HeeMo Kang[2)], YoonHwan Lee[3)]

Abstract

In general, R is useful for small size data. We study how to access a large data set in a database by R-language. We provide real examples to access data sets stored in database servers MySQL, Oracle or PostgreSQL.

Keywords: DBMS; MySQL; ODBC; SQL; Oracle; PostgreSQL.

† This work was supported by Hallym University Research Fund HRF-2007-029.

1) Professor, Department of Statistics, Hallym University, 1 Ockcheon-Dong, Chucheon, Kangwon-do 200-702, Korea. Correspondence: sysim@hallym.ac.kr

2) Full time instructor, Hallym University, 1 Ockcheon-Dong, Chucheon, Kangwon-do 200-702, Korea.

3) Part time instructor, Hallym University, 1 Ockcheon-Dong, Chucheon, Kangwon-do 200-702, Korea.