

윈도우 SAS 시스템에서 SAS/ACCESS 소프트웨어를 이용한 데이터베이스 연결

강 형 창¹⁾ 김 철 수²⁾

요 약

본 논문에서는 윈도우 SAS 시스템을 이용하여 윈도우 운영체제, 리눅스 운영체제, 유닉스 운영체제에 각각 저장되어 있는 원격 데이터베이스를 액세스하고 저장하는 방법에 대해서 설명한다. 원격 데이터베이스를 액세스하거나 저장하기 위해서는 윈도우 SAS 시스템이 설치되어 있는 로컬 컴퓨터에 ODBC에 DSN이 미리 등록되어 있어야 하며, 윈도우 SAS 시스템에서 SAS/ACCESS 소프트웨어를 이용하여 원격 데이터베이스를 액세스하거나 저장할 수 있다.

주요용어: ODBC DSN SAS/ACCESS

1. 서론

대용량 데이터베이스 자료를 분석할 기회가 많아지고 있고, 데이터 마이닝과 같은 새로운 기법의 출현으로 데이터베이스 자료를 직접 액세스하여 처리할 필요가 있다. 대부분의 데이터베이스는 로컬 컴퓨터에 존재하는 것이 아니라, 원격 컴퓨터에 존재하게 된다. 이에 본 연구에서는 원격 데이터베이스를 윈도우 SAS 시스템이 설치되어 있는 로컬 컴퓨터에서 연결하기 위한 방법에 대해서 설명한다. 그리고 SAS 시스템을 이용하여 다른 데이터베이스 간에 테이블을 저장하는 내용에 대해서 다룬다.

윈도우 SAS 시스템을 이용하여 원격 데이터베이스를 연결하기 위해서는 SAS/ACCESS 소프트웨어 또는 SAS/ACCESS ODBC 인터페이스가 필요하다. SAS/ACCESS 소프트웨어를 이용하면 데이터베이스의 자료를 액세스하고 SAS 프로그램에서 사용할 수 있다. SAS/ACCESS ODBC 인터페이스는 SAS LIBNAME 문장, SQL 프로시저, DBLOAD 프로시저를 이용하여 데이터베이스에 연결하는데 이용된다. 이 중에서 SQL 프로시저는 SAS/ACCESS 소프트웨어와 같이 사용되어 데이터베이스와 SAS 프로그램 사이에서 데이터베이스의 데이터를 SAS 프로그램으로 보내거나, SAS 프로그램의 데이터 셋을 데이터베이스의 데이터로 보내는 경우에 사용되고, SAS LIBNAME 문장은 데이터베이스 연결에 관계되며, DBLOAD 프로시저는 데이터베이스에 저장하고자 하는 경우에 사용된다.

이러한 SAS/ACCESS 소프트웨어를 이용하면 데이터베이스를 직접 분석할 수 있고, 서로 다른 데이터베이스 간에도 데이터를 서로 주고받을 수 있게된다. 본 논문의 구성은 다음과 같다. 2장에서는 데이터베이스 연결을 위한 요소들에 대해서 설명하며 3장에서는 원격 데이터베이스를 로컬 윈도우 SAS 시스템에서 연결하는 과정과 데이터베이스 간에 테이블 저장에 대한 내용을 다루고 4장에서는 결론 및 제언을 다룬다.

1) 제주대학교 전산통계학과 박사과정 hchkang@cheju.ac.kr

2) 제주대학교 전산통계학과 교수 cskim@cheju.ac.kr

2. 데이터베이스 연결을 위한 요소

SAS/ACCESS 소프트웨어를 이용하여 원격의 데이터베이스를 연결하기 위해서는 ODBC 데이터 원본 관리자에서 DSN을 미리 등록한 후, SAS/ACCESS 소프트웨어를 이용하면 ODBC 연결을 통하여 데이터베이스를 연결하는 인터페이스를 제공한다.

본 논문에서는 윈도우 운영체제 MSSQL 데이터베이스와 리눅스 운영체제 MYSQL 데이터베이스, 그리고 유닉스 운영체제 ORACLE 데이터베이스를 연결하기 위해 다음과 같이 DSN을 등록하였다.

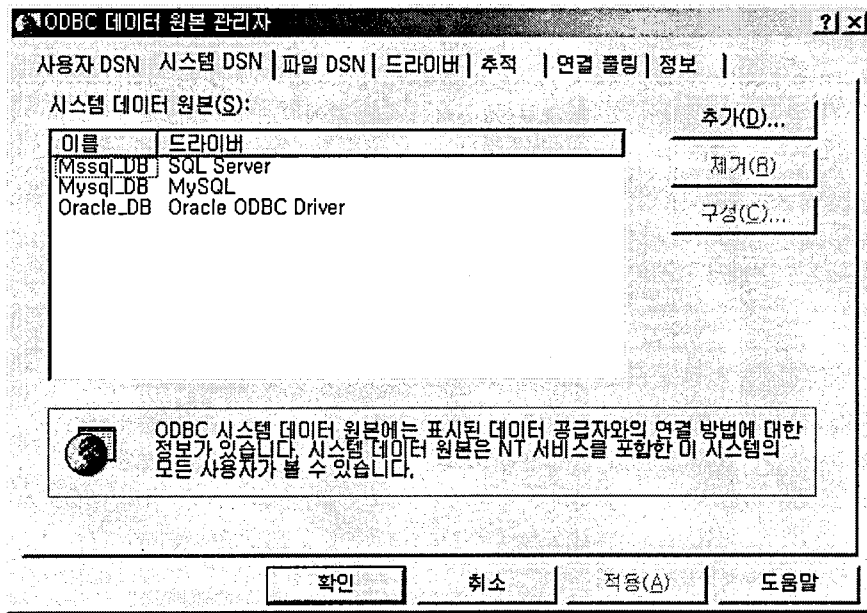


그림 1 ODBC DSN 등록

SAS에서 ODBC는 다음과 같은 인터페이스를 제공한다.

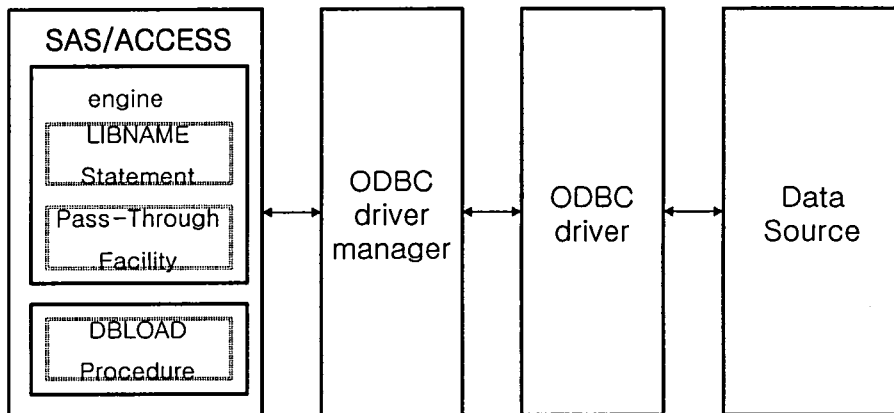


그림 2 ODBC 인터페이스 SAS 시스템

ODBC 인터페이스는 클라이언트 인터페이스, ODBC 드라이버 관리자, 그리고 ODBC 드라이버 세 가지로 구성된다. 클라이언트 인터페이스는 SAS/ACCESS 인터페이스 ODBC에서 제공하고, ODBC 드라이버 관리자는 클라이언트 인터페이스와 ODBC 드라이버를 상호 연결하며, ODBC 드라이버는 외부 데이터 또는 특정 데이터베이스를 다룰 수 있도록 연결시켜 준다.

ODBC가 로컬 컴퓨터에 등록이 된 다음에 SAS 시스템에서 SAS/ACCESS 소프트웨어를 이용하여 원격 데이터베이스를 연결한다.

3. 윈도우 SAS 시스템에서의 데이터베이스 연결

윈도우 SAS 시스템에서 SAS/ACCESS 소프트웨어를 이용하여 데이터베이스를 연결하는 방법은 크게 두 가지로 나눌 수 있다.

하나는 SAS/ACCESS 인터페이스로 데이터베이스를 연결하는 방법이고, 다른 하나는 SAS/ACCESS 인터페이스 ODBC로 ODBC에 등록되어 있는 DSN을 이용하여 데이터베이스를 연결하는 방법이 있다. 이 두 가지 방법 모두 ODBC에 등록되어 있는 DSN을 사용한다. 본 논문에서는 SAS 시스템에서 데이터베이스 연결 과정을 다음의 두 부분으로 나누어 설명하는데 첫째, SAS/ACCESS 인터페이스로 ORACLE 데이터베이스를 연결하는 하는 과정을 설명하고 둘째, SAS/ACCESS ODBC 인터페이스로 MSSQL 및 MYSQL 데이터베이스를 연결하는 과정에 대하여 설명한다.

3.1 SAS/ACCESS 인터페이스를 이용한 데이터베이스 연결

SAS/ACCESS 인터페이스를 이용하여 데이터베이스를 연결하는 방법은 데이터베이스를 SAS 시스템으로 읽어오는 것과 SAS 시스템의 데이터 셋을 데이터베이스에 저장하는 것으로 나눌 수 있다. 먼저, 데이터베이스를 SAS 시스템으로 읽어오는 방법을 보면 첫 번째, ACCESS 프로시저를 이용하여 데이터베이스를 액세스하는 방법과 LIBNAME 문장을 이용하여 데이터베이스 액세스하는 방법이 있다. 다음 [그림 3]은 ACCESS 프로시저로 ORACLE 데이터베이스를 연결하거나 LIBNAME 문장을 이용하여 ORACLE 데이터베이스를 연결한 예이다.

```
Proc Access DBMS=oracle;
Create work.ex1.access;
Table=dept;
User="scott"; Orapw="tiger"; Path="@Oracle_DB";
Assign=yes; List All;
Create work.ex1.view;
Select All;
run;
-----
Libname OraLib oracle user=scott password=tiger path=Oracle DB;
```

그림 3 ORACLE 데이터베이스를 SAS 시스템으로 불러오기
- ACCESS 프로시저 또는 LIBNAME 문장 -

두 번째 방법은 SQL 프로시저를 이용하여 데이터베이스를 액세스하는 방법이 있다. 다음 [그림 4]는 SQL 프로시저를 이용하여 ORACLE 데이터베이스를 연결한 예이다.

윈도우 SAS 시스템에서 SAS/ACCESS 소프트웨어를 이용한 데이터베이스 연결

```
Proc Sql;  
Connect To oracle(user="scott" orapw="tiger" path="Oracle_DB");  
Create Table work.ex2  
As Select * From connection to oracle(Select * From dept);  
Disconnect From oracle;  
Quit;
```

그림 4 ORACLE 데이터베이스를 SAS 시스템으로 불러오기
- SQL 프로시저 -

다음으로 SAS 시스템의 데이터 셋을 데이터베이스에 저장하는 방법은 첫 번째, DBLOAD 프로시저를 이용하여 SAS 데이터 셋을 데이터베이스에 저장하는 방법이다. 다음 [그림 5]는 DBLOAD 프로시저를 이용하여 SAS 데이터 셋을 ORACLE 데이터베이스에 저장하는 예이다.

```
Proc Dbload DBMS=oracle DATA=work.ex2;  
Orapw="tiger"; User="scott"; Path="Oracle_DB";  
Table=sas_ex1; Label;Reset All;Load;  
Run;
```

그림 5 SAS 데이터 셋을 ORACLE 데이터베이스에 저장하기
- DBLOAD 프로시저 -

두 번째, SQL 프로시저를 이용하여 SAS 데이터 셋을 데이터베이스에 저장하는 방법이 있다. 다음 [그림 6]은 SQL 프로시저를 이용하여 SAS 데이터 셋을 ORACLE 데이터베이스에 저장하는 예이다.

```
Proc Sql;  
Connect To oracle(user="scott" orapw="tiger" path="Oracle_DB");  
Create Table oracle.sas_ex2 As select * From  
connection to oracle(Select * From dept);  
Disconnect From oracle;  
Quit;
```

그림 6 SAS 데이터 셋을 ORACLE 데이터베이스에 저장하기
- SQL 프로시저 -

3.2 SAS/ACCESS ODBC 인터페이스를 이용한 데이터베이스 연결

SAS/ACCESS ODBC 인터페이스를 이용하는 방법은 SAS/ACCESS 인터페이스를 이용한 데이터베이스 연결과 마찬가지로 데이터베이스를 SAS 시스템으로 자료를 불러오는 것과 SAS 데이터 셋을 데이터베이스로 저장하는 것으로 나눌 수 있다. 먼저 데이터베이스를 SAS 시스템으로 읽어오는 방법을 보면 SQL 프로시저를 이용하여 데이터베이스를 액세스하는 방법과 LIBNAME 문장을 이용하여 데이터베이스 액세스하는 방법이 있다. 다음 [그림 7]은 SQL 프로시저로 MSSQL 데이터베이스를 연결하거나 LIBNAME 문장을 이용하여 MSSQL 데이터베이스를 연결한 예이다.

```

Proc Sql;
Connect To odbc (dsn=Mssql_DB uid=JSP pwd=mssql);
Create Table work.mssql_ex
  As Select * From connection to odbc(Select * From stu_list);
Disconnect from odbc;
Quit;
-----
Libname odbc_ms odbc dsn="Mssql_DB" uid=JSP pwd=mssql;

```

그림 7 MSSQL 데이터베이스를 SAS 시스템으로 불러오기
- SQL 프로시저 또는 LIBNAME 문장 -

다음으로 SAS 시스템의 데이터 셋을 데이터베이스에 저장하는 방법은 DBLOAD 프로시저를 이용하여 SAS 데이터 셋을 MSSQL 데이터베이스에 저장하는 것으로 다음은 그 예이다.

```

Proc Dbload DBMS=odbc Data=work.ex2; dsn="Mssql_DB";
uid="JSP"; password="mssql";
Table="Mssql.upload";
Limit=0; Load;
Run;

```

그림 8 SAS 데이터 셋을 MSSQL 데이터베이스로 저장하기
- DBLOAD 프로시저 -

MSSQL 데이터베이스를 연결한 것과 마찬가지로 방법으로 MYSQL 데이터베이스를 연결하여 데이터베이스를 SAS 시스템에 불러올 수 있으며, SAS 데이터 셋을 MYSQL 데이터베이스에 저장할 수 있다. 다음은 SAS window 문장을 이용하여 MYSQL 데이터베이스의 테이블을 SAS 데이터 셋으로 불러오는 예이다.

```

Enter Data Source Name(DSN):      Mysql_DB
Enter Useid:                      cskim
Enter Password:                   _____
Enter Database Table Name        postzip

```

그림 9 MYSQL 데이터베이스를 SAS 시스템으로 불러오기
- SAS window 문장을 이용한 SQL 프로시저 -

다음은 SAS window 문장을 이용하여 SAS 데이터 셋을 MYSQL 데이터베이스의 테이블로 저장하는 예이다.

```

Enter SAS Data Set Name          work.postzip
Enter Table Name                  upload
Enter Data Source Name(DSN):     Mysql_DB
Enter Useid:                      cskim
Enter Password:                   _____

```

그림 10 SAS 데이터 셋을 MYSQL 데이터베이스로 저장하기
- SAS window 문장을 이용한 DBLOAD 프로시저 -

지금까지 다룬 ORACLE 데이터베이스, MSSQL 데이터베이스, 그리고 MYSQL 데이터베이스를 SAS 시스템으로 연결하는 과정과 SAS 데이터 셋을 데이터베이스에 저장하는 과정을 이용하면 데이터베이스 간의 테이블 읽기 및 저장이 가능하다. 다음 [그림 11]은 ORACLE 데이터베이스와 MYSQL 데이터베이스를 연결하여 ORACLE 데이터베이스에 MYSQL 데이터베이스의 테이블을 저장하는 예이다.

| | |
|--|-------------------|
| Enter Data Source Name(Open Database) | <u>Oracle_DB</u> |
| Enter Useid(Open Database) | <u>scott</u> |
| Enter Password(Open Database) | _____ |
| Enter Table Name(Open Database) | <u>dept</u> |
| Enter Data Source Name(Save Database): | <u>Mysql_DB</u> |
| Enter Useid(Save Database) | <u>cskim</u> |
| Enter Password(Save Database) | _____ |
| Enter Table Name(Save Database) | <u>ora_upload</u> |

그림 11 ORACLE 데이터베이스 테이블을 MYSQL 데이터베이스 테이블로 저장하기
- SAS window 문장과 SQL 프로시저 -

4. 결론

본 논문에서는 원격 데이터베이스를 연결하기 위해 ODBC와 SAS/ACCESS 소프트웨어를 이용하여 SAS 시스템에서 데이터베이스를 불러오는 방법과 저장하는 방법에 대해 설명하였다. 이는 윈도우 SAS 시스템에서 SAS/ACCESS 인터페이스, 또는 SAS/ACCESS ODBC 인터페이스를 이용하면 윈도우 운영체제의 MSSQL 데이터베이스를 리눅스 운영체제의 MYSQL 데이터베이스, 또는 유닉스 운영체제의 ORACLE 데이터베이스로 변환할 수 있으며, 같은 방법을 이용하여 리눅스, 또는 유닉스 운영체제의 데이터베이스를 윈도우 운영체제의 데이터베이스로 변환할 수 있다는 것을 알 수 있다.

그러나 원격 데이터 베이스를 연결하여 데이터베이스의 자료를 SAS 시스템으로 불러오거나 SAS 데이터 셋의 내용을 데이터베이스로 저장하기 위해서는 ODBC에 DSN이 미리 등록되어 있어야 하고, SAS/ACCESS 소프트웨어를 사용할 수 있는 라이선스 및 SAS/ACCESS ODBC 인터페이스를 사용할 수 있는 라이선스가 필요하다.

앞으로 연구될 방향은 JDBC를 이용한 원격 데이터베이스 연결 후 SAS 시스템에서 분석·처리 가능한 데이터 파일로 변환하는 방법에 관한 것이다.

참고문헌

- SAS/ACCESS Software for Relational Databases: Reference8
- SAS/ACCESS Software for Relational Databases: Reference8(ODBC Chapter)
- SAS/ACCESS Software for Relational Databases: Reference8(ORACLE Chapter)