

# 국제 표준 한글 부호계를 지원하는 관계형 데이터베이스 한글 사용자 질의 환경 개발

유 영호, 손 영성, 정 준영, 김 경석  
부산대학교 전자계산학과

A Development of User Environment for Querying on RDBMS  
supporting Hangul Code in ISO 10646-1

Youngho You, Youngsung Son, Junyoung Jung, Kyongsok Kim  
Dept. of Computer Science, Pusan National University

## 요약

국제 표준화 기구 (ISO) 에서 1992 년 ISO 10646-1 을 확정하였다. ISO 10646-1 에 들어 있는 국제 표준 한글 부호계에는 현재까지 발견된 옛한글 글자가 모두 들어 있다. 이전에는 옛한글에 대한 표준 부호계가 없어서 옛한글 자료에 대한 전산화에 큰 걸림돌이 되었으나, 국제 표준 한글 부호계의 등장으로 이러한 문제가 해결되었다고 할 수 있다. 본 논문에서는 국제 표준 한글 부호계를 유닉스의 X-window 환경에서 지원하는 방안이 대해 살펴 보고, 국제 표준 한글 부호계를 지원하는 관계형 데이터베이스 한글 사용자 질의 환경을 개발하므로써, 지원 방안의 실증과 함께 옛한글 자료를 데이터베이스화 하기위한 바탕을 마련하였다.

## I. 들머리

국제 표준 글자 부호계 ISO/IEC 10646-1 (본 논문에서는 ISO/IEC 10646-1을 10646-1 또는 국제 표준 글자 부호계라 부르고, 여기에 있는 한글 부호계를 국제 표준 한글 부호계라 부르겠다) 이 1992 년 7 월에 국제 표준으로 확정되고, 이어 1993 년 5 월에 표준 문서가 배포되었다.

국제 표준 글자 부호계는 전 세계의 주요한 글자를 모두 수용하고 있기 때문에, 앞으로 국내 및 국제 소프트웨어 개발 및 판매에 큰 영향을 미칠 것으로 보인다. 그 전에는 영어 위주로 개발된 프로그램을 다시 각 나라의 글자에 맞추어 고치는 작업 (localization) 을 해야하는 등의 문제가 있었으나, 10646-1 을 지원하는 프로그램은 한글화 작업이 간단해지므로, 국내 및 국제 소프트웨어 시장에 아주 큰 영향이 있게 된다. 소프트웨어 기술의 국제 경쟁력 강화를 위해 국제 표준 한글 부호계에 대한 기반 기술 축적이 필요할 때다.

국내에서 개발되어 이미 시판되고 있는 국산 주전산기 II (일명 타이콤) 및 현재 개발 중에 있는 국산 주전산기 III 의 운영 체제가 모두 유닉스이고, 또한 통신 환경의 개선으로 유닉스 사용자가 늘어나고 있는 추세이다. 본 연구에서는 유닉스의 X 윈도 환경에서 국제 표준 한글 부호

계를 지원할 수 있도록 하였다.

옛한글 자료를 데이터베이스화하기 위해서는 크게, 입력 (자판), 부호계 (코드), 그리고 출력 (글꼴) 의 세 가지 문제를 풀어야 한다. 이 중에서 옛한글 처리에 걸림돌이 되었던 부호계의 문제가 국제 표준 한글 부호계의 등장으로 해결되었다. 폰트는 개발에 크게 어려움이 없고, 자판 문제는 [김 경석 89 가, 89 나, KimK 90b, 92c] 등에서 제안한 바 있다. 더욱이, 이 네 논문에서 제안한 옛한글 자판을 옛한글 문서 편집기에서 써 본 결과 아무런 문제없이 쓸 수 있음이 이미 확인된 바 있다. [KimK 94a]

위에서 살펴본 대로, 한글 자료를 데이터베이스화하기 위한 기본 요건들은 모두 해결되었다고 할 수 있다. 본 연구에서는 이러한 요건들을 바탕으로, 한글 자료를 데이터베이스화하기 위해 꼭 필요한 한글 입출력을 제공하는 한글 사용자 질의 환경을 개발하였다.

## II. 국제 표준 한글 부호계의 지원 방안

### 1. 유닉스 시스템에서의 국제 표준 한글 부호계의 처리 방안

유닉스 환경에서 여러 나라의 글자를 지원하는 방안으로 EUC (Extended UNIX Code) 를 쓰는데, EUC 는 기본

적으로 ISO 2022 방식을 따른다. ISO 10646-1 은 ISO 2022 방식을 따르지 않으므로, EUC 방식으로 국제 표준 한글 부호계를 정상적으로 지원할 수는 없다. [김 경석 93 나]

유닉스 시스템에서 국제 표준 한글 부호계를 지원하는 방법은 크게 보아 다음과 같이 세 가지로 나눌 수 있다.

1) 운영 체제 (OS) 수준에서 한글을 처리하는 방법

운영 체제 수준에서 한글 도스와 같이 국제 표준 한글 부호계를 지원하는 한글 유닉스 시스템을 만들어 지원하는 방법이다. 이는 기존의 유닉스 시스템을 고쳐서 여기에 한글 처리 루틴을 추가하는 방법으로 구현할 수 있다. 그러나, 이 방법은 작업량이 너무 방대하다는 단점이 있다.

2) 새로운 한글 X 터미널을 만드는 방법

현재까지 대부분의 유닉스 시스템에서는 유닉스 시스템 그래픽 사용자 환경인 X 윈도우 시스템에서 한글 X 터미널을 만들어 한글을 사용할 수 있도록 하였다. 이 방법은 X 윈도우 환경에서 수행할 수 있는 대부분의 사용자 프로그램에서 한글을 사용할 수 있다는 장점이 있다.

3) 사용자 프로그램 (Application Program) 수준에서 한글을 지원

사용자 (응용) 프로그램 수준에서 국제 표준 한글 부호계를 지원하는 루틴을 내장시켜서 한글을 지원하는 방법이다. 이 방법에서는 한글 X 터미널과 같은 별도의 한글 시스템의 지원 없이도 한글을 쓸 수 있다. 또한 한글 지원이 운영 체제와는 따로 이루어지기 때문에, 이전까지 유닉스 시스템에서 쓸 수 없었던 각종 조합 방식 한글 부호계를 쓸 수 있다는 장점이 있다.

본 연구에서는 세 번째 방법을 써서 국제 표준 한글 부호계를 지원하였다. 완성형 부호계만을 지원하는 기존의 한글 Motif 의 레이블 위젯과 텍스트 위젯에 대응하는 국제 표준 한글 부호계를 지원 (옛한글까지 모두 지원함) 하는 레이블 위젯과 텍스트 위젯을 개발하여 한글 입력력을 처리하였다. 위젯 수준으로 한글을 지원할 경우 다른 응용 프로그램에서도 이 위젯들을 사용하여 쉽게 국제 표준 한글 부호계를 지원할 수 있는 장점이 있다.

2. 국제 표준 한글 부호계를 위한 자판

옛한글을 위한 자판에 대해서 이미 몇 가지 연구가 나와 있다. 먼저 체신부에서 지원하여 한국 표준 연구소에서 수행한 연구 보고서 [체신부]에서는 옛한글 글자를 넣기 위해서 특수 문자 입력 방식을 빌어서 쓰는데, 이는 부호값을 바로 넣거나 ("코드 입력 방식") 아니면 "메뉴 선택 방법"을 쓰는 것이다. [체신부] 이것은 요즘 한글 입력과 아주 다른 방식이라 쓰기에 불편할 뿐만 아니라, 입력 속도도 느리다. 한편, 옛소리마디를 입력하기 위해서는 "한자의 독음 입력 방식" 처럼 "색인음에 의한 메뉴 선택 방법"을 쓴다. 이 방식 또한 요즘 한글 입력 방식과 아주 다른 방식이라 쓰기에 불편할 뿐만 아니라, 입력 속도도 느리다.

본 연구에서 사용한 자판의 글쇠 배당은 [그림 1] 과

같다.

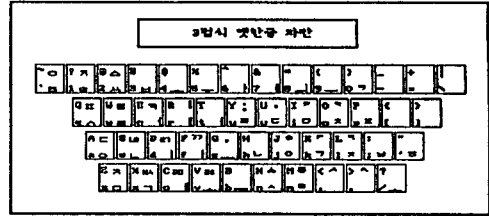


그림 1. 3 벌식 옛한글 자판

글쇠의 배당은 세벌식 390 자판을 기본으로 하여, 옛 흘글자를 모양이나 성격이 비슷한 요즘 한글 글쇠의 가까이 배당하여 쉽게 외울 수 있도록 하였고, 입력 속도는 고려하지 않았다. 옛 겹글자의 입력은 그 겹글자를 이루는 흘글자를 하나씩 입력하도록 하여 입력 오토마타에서 겹글자 테이블을 두어 겹글자가 가능한 입력에 대해 테이블을 검사하여 처리하였다. 보기를 들어, 'ㅅㅈ' 은 자판에서 'ㅅ' + 'ㅈ' 과 같이 입력하면 입력 오토마타에서 겹글자를 조합한다. 입력 오토마타에 대한 설명은 III 장의 2.1 에 있다.

3. X 윈도우에서 사용할 수 있는 글꼴 연구와 개발

국제 표준 한글 부호계에는 현재까지 발견된 옛한글 자모가 모두 포함되어 있다. 그러나, 현재 개발되어 있는 글꼴 중, 옛한글을 일부 지원하기는 하지만 완전히 지원하는 글꼴은 아직 없다. 본 연구에서는 국제 표준 한글 부호계에 들어 있는 첫소리 글자, 가운뎃 소리 글자, 끝소리 글자에 대한 글꼴을 각각 한 벌씩 만들었다.

X 윈도우 시스템에서 사용되어지는 글꼴 파일은 Bitmap Distribution Format 2.1 에 따라 기술되어 있다. 이 파일을 bdf 파일이라 부르고 확장자 bdf 를 붙여서 표시한다. X 서버가 직접 사용하는 글꼴 파일은 snf (server natural format) 파일로 확장자 snf 를 붙여서 표시한다. snf 파일은 bdf 파일을 bdf2snf 라는 변환 프로그램을 사용해서 X 서버가 사용할 수 있도록 한 것이다. snf 파일의 포맷은 만든 회사에 따라 다르므로, 호환하려면 bdf 파일을 가져가서 다시 컴파일해야 한다.

본 연구에서 개발하여 사용한 한글 글꼴의 크기는, 일반적으로 16×16 을 사용하지만, 옛한글의 세겹글자까지 표현하기 위하여 글꼴의 크기를 24×24 로 하였다. 영문 글꼴로는 기존의 X 윈도우용 영문 글꼴 중에서 24×24 한글 글꼴과 크기가 어울리도록 12×24 글꼴을 사용하였다.

III. 한글 사용자 질의 환경의 구현

본 연구에서는 유닉스 환경에서 국제 표준 한글 부호계를 쓰는 한글 입출력 환경을 구현하고, 이를 데이터베이스 관리 시스템의 질의 입력으로 쓸 수 있도록 하였다.

유닉스 환경에서 한글은 일반 텍스트 모드에서는 쓸 수 없고 그래픽 환경에서만 쓸 수 있다. 그래서, 본 시스템에서도 한글의 입력과 출력을 유닉스의 표준 그래픽 환경인 X 윈도우 시스템상에서 구현하였다.

## 1. 시스템의 구성

본 시스템은 국제 표준 한글 부호계를 사용하여 사용자가 입력한 질의를 받아들이고, 최종 결과를 화면에 보여주는 한글 입출력 모듈, 메뉴상에서 여러가지 기능을 수행할 수 있는 메뉴 관리 모듈, 한글 입출력 모듈에서 입력된 질의를 적절하게 변환하여 DBMS 에 전달하고 원하는 결과를 얻어 내는 질의 처리 모듈, 그리고 질의 처리 결과를 화면에 보내기 위하여 준비하는 결과 처리 모듈로 크게 나누어 볼 수 있다. 시스템의 전체적인 구성은 [그림 2] 와 같다.

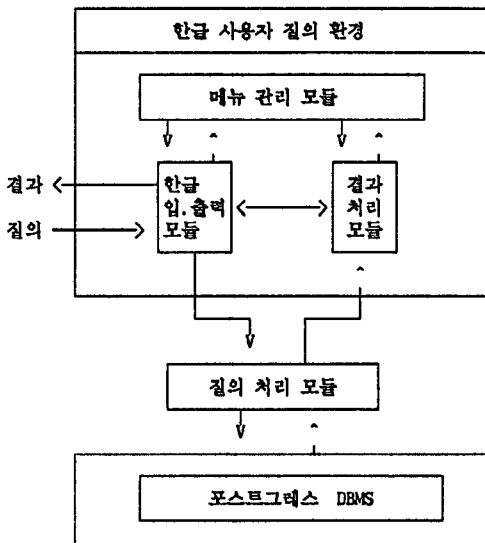


그림 2. 시스템 구성도

질의 처리 모듈을 따로 분리하여 구성한 이유는 시스템 확장과 유지보수를 효과적으로 할 수 있도록 하기 위해서다. 보기를 들어, 포스트그레스가 아닌 다른 데이터베이스 관리 시스템을 쓰고자 할 때, 질의 처리 모듈만을 새로 작성함으로써 다른 시스템과도 연결하여 사용할 수 있다. 또한, 원격지 데이터베이스 접근 시스템으로 확장하고자 할 때에도 유용한데, 이를 위해서는 [그림 2] 의 구조에서 한글 사용자 질의 환경을 클라이언트로, 질의 처리 모듈과 데이터베이스 시스템을 서버로 나누고, 여기에 클라이언트와 서버사이의 데이터 전송을 담당할 통신 부분을 추가함으로써

서 간단히 구현할 수 있다.

## 2. 한글 입출력 모듈의 구현

본 연구에서는 OSF/Motif 의 문자열 출력을 위한 레이블 위젯과 텍스트 자료 입력을 위한 텍스트 위젯에 대응하는 국제 표준 한글 부호계용 위젯을 개발하여 한글 입출력을 지원한다는 것은 앞서 말하였다. 국제 표준 한글 부호계 방식의 한글을 위한 구조체를 기존 영문 위젯에 추가함으로써 이를 구현하였다.

### 2.1 한글 입력기의 구현

본 연구에서는 세벌식 자판을 사용하므로, 첫소리 글자와 끝소리 글자의 글쇠 위치가 달라서 입력기의 설계가 아주 간단하다. 사용자가 글쇠를 입력할 때의 입력 모드를 검사하여, 영문 모드일 경우는 글쇠에 해당하는 영문 코드를 생성하고, 한글 모드일 경우는 한글 처리 루틴에서 한글 입력 오토마타의 상태와 입력에 따라 오토마타의 상태를 전이시키고 한글 코드를 생성한다. 특히, 한글 입력 모드에서 입력된 글쇠에 대해서는 접글자를 이룰 수 있는지에 대한 검사를 하여야 하는데, 이는 별도의 루틴에서 준비된 접글자 테이블을 검사하여 처리하도록 하였다. 접글자 테이블은 아래와 같이 구성하였다.

기본 글자	접글자가 가능한 글자	접글자의 코드
-------	-------------	---------

입력되는 소리마디가 “첫소리 글자 + 가운뎃소리 글자” 또는 “첫소리 글자 + 가운뎃소리 글자 + 끝소리 글자”로 이루어지는 소리마디를 완전한 소리마디라고 하고 그 밖의 것을 불완전한 소리마디로 정의한다. 한글 입력기에서 이러한 불완전한 소리마디에 대해서도 처리해 주어야 한다. 본 연구에서는 국제 표준 한글 부호계에 정의되어 있는 첫소리 채움 글자와 가운뎃소리 채움 글자를 한글 오토마타에서 자동적으로 삽입되도록 하였다.

### 2.2 한글 출력기의 구현

출력 루틴의 설계에서 가장 중요한 부분은, 유닉스 시스템에서 일반적으로 사용하는 비트맵 글꼴을 화면에 나타내도록 하는 부분이다.

X 라이브러리엔 X 윈도우용 글꼴을 화면에 출력해 주는 XDrawString, XDrawImageString 등의 함수가 있다. XDrawString 함수는 오로지 포그라운드 픽셀만 그리고, XDrawImageString 함수는 문자의 사각 테두리 안의 포그라운드와 백그라운드 픽셀을 그린다. XDrawString 함수의 구조는 다음과 같다.

```
XDrawString(XtDisplay(da), XtWindow(da),
            gc, x, y, s, strlen(s));
```

여기서 x 와 y 는 문자열이 출력될 위치를 나타내고

strlen(s) 는 문자열의 길이를 나타낸다. 그리고, 출력할 문자열의 폰트는 gc (graphic context) 에 있는 폰트 필드로 결정된다. XtDisplay(da) 와 XtWindow(da) 는 각각 문자열을 출력할 화면과 윈도우를 지정한다. XDrawImageString 은 XDrawString 와 같은 파라미터를 갖는다.

ASCII 문자의 출력을 위해서는 XDrawString 와 XDrawImageString 을 사용하면 되지만, 한글 폰트 파일에서의 폰트 위치를 한 바이트로는 모두 나타내지 못하기 때문에, 한글 출력을 위해서는 16 비트 코드를 지원하는 XDrawString16 함수와 XDrawImageString16 함수를 사용한다. 이 함수들은 XDrawString 와 같은 파라미터를 가지는데, s 의 자료형이 XChar2b \* 라는 두 바이트 구조체의 형태를 가지는 점이 다르다. 구조체 XChar2b 의 구성은 아래와 같다.

```
typedef struct {
    unsigned char byte1;
    unsigned char byte2;
} XChar2b;
```

한글 한 소리마디를 출력할 때, 해당되는 각 글자의 폰트 파일에서의 위치를 XChar2b 에 넣어서 XDrawString16 나 XDrawImageString16 함수를 호출하면 각 글자에 해당하는 글꼴이 조합되어 화면에 출력된다.

국제 표준 한글 부호계는 한 글자를 2 바이트로 나타내며, 한 소리마디를 4 또는 6 바이트로 나타내기 때문에 문자열을 출력하는데 있어서 한 소리마디가 가변 길이임을 고려해야 한다. KS C 5601 완성형 또는 두 바이트 조합형과 같은 2 바이트 한글 부호계를 쓰면, 한글 한 소리마디가 두 바이트를 차지하므로, 한 바이트를 차지하는 ASCII 문자의 배수가 되기 때문에 출력 루틴 작성이 쉽다. 즉, 주어지는 문자열에서 현재 출력할 글자가 한글인지를 검사하여 한글일 경우는 다음 바이트를 강제로 읽어서 두 바이트로 소리마디를 구성하여 출력하면 된다.

그러나, 국제 표준 한글 부호계를 사용할 경우, 한글 한 소리마디가 끝소리 글자가 있느냐 없느냐에 따라, 6 또는 4 바이트를 차지하여, 영어 한 글자의 일정한 배수가 되지 않기 때문에 두 바이트 한글 부호계에서 쓰는 방법을 쓸 수가 없다. 따라서 문자열을 출력할 때 한글이 나오면, 먼저 끝소리 글자가 있는지를 검사해서, 6 또는 4 바이트를 읽어 한 소리마디를 구성하여 출력해야 한다.

### 3. 메뉴 관리 모듈

본 시스템의 메뉴 관리 모듈은 사용자와의 인터페이스를 담당한다. 전체적인 화면의 구성은 [그림 3] 과 같고, 각 부분의 기능은 다음과 같다.

#### (1) 메뉴

1. Open - 이미 저장되어 있는 데이터베이스를 연다.
2. Close - 현재 사용중인 데이터베이스를 저장하고 닫는다.

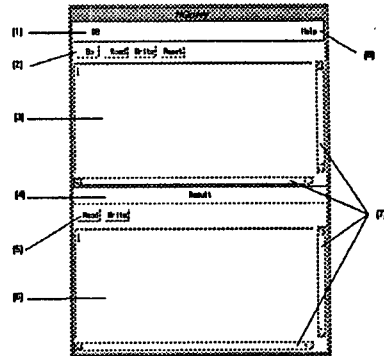


그림 3. 메뉴 관리 모듈 구성도

3. Create - 새로이 데이터베이스를 만든다.
  4. Destroy - 현재 존재하는 데이터베이스를 삭제한다.
  5. Quit - 본 프로그램을 종료한다.
- (2) 입력창 버튼
1. Go - 입력창에 기술되어 있는 질의 (Query) 를 실행한다.
  2. Read - 이미 저장되어 있는 질의를 읽어온다.
  3. Write - 입력창에 기술된 질의를 저장한다.
  4. Reset - 입력창을 리셋한다.
- (3) 입력창  
입력 중인 사용자 질의를 보여준다.
- (4) 결과표시 레이블
- (5) 결과창 버튼
1. Read - 이미 저장되어 있는 질의의 결과를 읽어온다.
  2. Write - 현재 결과창에 나타나 있는 질의의 결과를 저장한다.
- (6) 결과창  
입력창에 기술되어 있는 질의의 결과가 나타난다.
- (7) 스크롤 바  
입력창과 결과창의 내용이 한 화면을 넘어갈 때, 스크롤 바를 이용하여 아래, 위, 좌, 우로 움직여 화면을 볼 수 있다.
- (8) 도움말
1. About - 본 프로그램에 대한 간단한 소개이다.
  2. Help - 본 프로그램의 도움말이다.
4. 질의 처리 모듈의 설계 및 구현과 포스트그래스와의 접속 방법 연구

본 연구에서 사용자 질의 환경을 구성할 때 수행 특성에 따라 여러 모듈로 나누었는데, 그 중에서 질의 처리 모듈은 사용자 환경, 특히 입출력 모듈에서 구성된 질의를 데

이더베이스 관리 시스템으로 전달하여 실제 질의를 수행하는 역할을 한다.

앞서 말한바와 같이 질의 처리 모듈을 분리한 이유는, 차후에 이 시스템의 확장과 유지 보수를 효과적으로 하기 위함이었다. 특히 포스트그레스가 아닌 다른 데이터베이스 관리 시스템에 쓰고자 할 때 프로그램 전체를 다시 만들지 않고 질의 처리 모듈만을 새로 작성하여 다른 시스템과도 연결하여 사용할 수 있게 하고자 하는 것이 목적이었다. 본 연구에서는 국제 표준 한글 부호계 지원이라는 또 하나의 목표가 있으므로, 국제 표준 한글 부호계로 입력된 사용자의 질의를 포스트그레스에서 처리할 수 있도록 하는데 중점을 두었다.

질의 처리 모듈에서 질의 처리는 다시 전처리, 질의 처리, 결과 분석의 세 가지 단계로 나누어서 질의를 수행하게 된다. 질의 처리 모듈의 전체 구성은 [그림 4]와 같다.

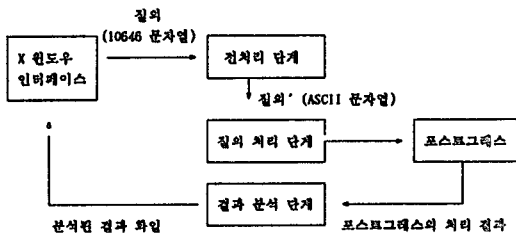


그림 4. 질의 처리 모듈 구성도

#### 4.1 전처리 단계

전처리 단계에서는 국제 표준 한글 부호계로 입력된 사용자의 질의를 포스트그레스에 적합한 형태로 바꾸는 작업을 한다. 포스트그레스는 ASCII 코드 형태의 질의만을 처리하므로 사용자의 국제 표준 한글 부호계 형식의 입력을 포스트그레스에 적합한 형태로 바꾸는 작업이 필요하다. 이 작업은 "iso-10646" 이라는 새로운 클래스를 정의하고, 새 클래스에 대한 입출력 형식을 정의하여 구현하였다.

##### 가. 입력 함수 (iso-in)

자판에서 한글을 입력하면, 한글 입출력 모듈에서 글자에 해당하는 부호값을 ASCII 문자열로 질의 처리 모듈에 넘겨준다. 입력 함수에서는 이를 십육진수로 고쳐서 데이터베이스에 넣는다. 보기를 들어, "가" 를 입력하면 실제로 데이터베이스에는 "0x11001161"이 들어가게 된다.

##### 나. 출력 함수 (iso-out)

입력 함수와 반대되는 일을 하는 것으로, 데이터베이스로부터 십육진수로 표현된 국제 표준 한글 부호계의 부호값을 읽어와서 이를 ASCII 문자열로 만들어 결과 처리 모듈로 넘겨준다.

#### 4.2 질의 처리 단계

질의 처리 단계에서는 포스트그레스의 LIBPQ 라이브러리를 사용하여 사용자의 질의를 포스트그레스에 요청하거나 직접 처리한다. 이미 존재하는 데이터베이스의 삭제, 다른 데이터베이스로의 이동은 포스트그레스에 요청하지 않고도 처리할 수 있다. 포스트그레스에 대한 요청은 LIBPQ 라이브러리 중의 PQexec() 함수를 이용하게 되는데, 이 함수의 결과값을 분석하여 결과 분석 단계에서 X 윈도우 인터페이스로 제공될 결과가 생성된다.

#### 4.3 결과 분석 단계

사용자의 질의에 대한 포스트그레스의 결과를 X 윈도우 인터페이스가 처리하기 좋은 형태로 가공하여 파일 형태로 제공한다. 파일은 질의가 성공하였는가 실패하였는가에 대한 답을 제시하고 그 뒤에 처리 결과를 가지고 있는 구조이다. X 윈도우 인터페이스로 보내질 파일의 구조는 [그림 5]와 같다.

성공	TYPE1	명령어			
	TYPE2	그룹의 수	$n(t) \times n(A)$	A_List...	t_List...
			$n(t) \times n(A)$	A_List...	t_List...
실패	TYPE1	명령어	에러 메시지		

n(t) : 튜플의 수  
 n(A) : 에트리뷰트의 수  
 t\_List : 튜플 리스트  
 A\_List : 에트리뷰트 리스트

그림 5. 결과 파일의 구조

#### 5. 결과 처리 모듈

질의 처리 모듈에서 전송 받은 결과는 튜플의 스트림으로 되어 있는데, 이를 결과 처리 모듈에서 사용자가 알아볼 수 있는 형태, 즉, 테이블 형태로 재구성한다. 재구성된 결과는 한글 입출력 모듈로 전송되어 사용자가 이를 받아볼 수 있다.

#### IV. 맺음말

본 논문에서는, 국제 표준 한글 부호계 방식의 한글 입출력을 지원하는 한글 사용자 질의 환경에 대해 살펴보았다. 본 연구의 의의는 X 윈도우와 관계형 DBMS (Database Management System) 환경에서 국제 표준 한글 부호계 방식의 한글 자료 (옛한글 포함) 처리 및 입출력을 지원하는 프로그램을 직접 개발함으로써, 관계형 데이터베이스에서 한글 입출력을 원활히 할 수 있게 되어, 이전에는 할 수 없었던 옛한글 자료의 데이터베이스화를 가능하게

했다는데 있다.

앞으로의 연구는 포스트그레스가 아닌 다른 데이터베이스로 시스템을 확장하고, 더 나아가서는 원격지 데이터베이스 접근 시스템으로 확장하는 방향으로 하여 한글 자료의 데이터베이스화에 더 많은 기여를 할 수 있게 되기를 바란다.

## V. 참고 문헌

[김 경석 89 가] "전자계산기에서 옛글 (옛한글) 을 치기에 좋은 자판에 대해서 생각해 봄: 세벌식과 두벌식을 중심으로". 김 경석. 한글 새 소식 제 198 호 (1989.2). 8-11 쪽. 한글 학회.

[김 경석 89 나] "한글 사전 편찬 전산화를 위한 터뉘이: 옛한글 가나다 순 및 옛한글 자판". 김 경석. 교육 한글 제 2 호 (1989). 5-50 쪽. 한글 학회.

[김 경석 93 나] 고속 중형 컴퓨터 (주전산기 III) S/W 환경에서의 표준 한글 처리 연구. 김 경석 (책임 연구원). 부산대학교 정보 통신 연구소. 1993.1 - 1993.12. 145 쪽.

[채신부] "표준 한글 한자 코드 확장 및 처리 방식 연구 별책: 정보 교환용 부호 확장 시안, 확장 한글 처리 방식 제안서". 채신부. 연구 기관: 재단 법인 한글 표준 연구소. 104 쪽.

[ISO 93] ISO/IEC International Standard 10646-1, Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane. May 1993. ISO (the International Organization for Standardization).

[KimK 90b] "Databases supporting Hangeul (or Korean Script)", Kyongsok Kim. 1990 IEEE International Conference on Systems, Man and Cybernetics. pp. 485-490, Nov. 4-7, 1990. Universal City, CA., U.S.A.

[KimK 92c] "A common Approach to Designing the Hangeul Code and Keyboard", Kyongsok Kim. Computer Standards & Interfaces, Vol. 14, No. 4, pp. 297-325, 1992.

[Post] Postgres Reference Manual. University of California at Berkeley.

[KimK 94a] "An implementation of an Old-Hangeul Editor supporting Syllable-Initial-Peak-Final Hangeul Code in ISO/IEC 10646-1". Kyongsok Kim and Junghwa Lee. Proceedings of 1994 International Conference on Computer Processing of Oriental Languages. May 10 - 13, 1994.

[KimK 94b] "Relational Databases supporting International Standard Character Code ISO/IEC 10646-1". Kyongsok Kim, Hongwon Yun, Yongho Kim. Proceedings of 1994 International Conference on Computer Processing of Oriental Languages. May 10 - 13, 1994.