

# 데이터베이스 SQL 문 변환기 실험 연구

나아름, 양진영, 박지민, 박경모  
가톨릭대학교 컴퓨터정보공학부

e-mail : { kpark, jyy0103, a90907, bongbbi }@catholic.ac.kr

## An Experiment Study on Database SQL Statement Converter

A-reum Na, Jinyoung Yang, Ji-Min Park, Kyeongmo Park

Computer Science and Information Engineering School, The Catholic University of Korea

### 요 약

데이터베이스는 자료의 검색과 갱신의 효율화를 위해 논리적으로 연관된 여러 자료의 집합체로 만드는 것을 의미한다. 인터넷과 하드웨어의 발달로 관리해야 할 자료의 양이 많아 지고 있으며, 이를 쉽게 관리 하기위해 데이터베이스는 나날이 발전하고 있다. 본 논문에서는 데이터베이스의 역사, 종류, 데이터 베이스 언어 현황과 그중 SQL 언어의 명령어와 이를 쉽게 사용 할 수 있도록 돕는 변환 프로그램 실험 결과를 보고한다.

### 1. 서 론

2000 년대에 들어서 인터넷과 하드웨어의 발달로 생산되는 데이터의 양이 급증하였다. 다양한 기업, 기관, 단체 등은 막대한 양의 데이터를 보다 쉽게 관리 하기 위해 컴퓨터를 이용하여 데이터 베이스화를 이용하기 시작했다. 이에 따라 데이터 베이스의 중요함은 나날이 높아지고 있다.

데이터 베이스는 논리적으로 연관된 여러 자료의 모음으로 구성시켜 검색과 갱신의 효율화를 꾀한 것이다. 즉, 여러 개의 자료 파일을 조직적으로 통합하여 자료 항목의 중복을 없애고 자료를 구조화하여 기억시켜 한 조직의 여러 응용 프로그램이 공동으로 소유, 유지, 이용하는 자료의 집합체라고 할 수 있다. 데이터의 종속성과 중복성의 문제를 해결하기 위해 데이터 베이스를 공유 할 수 있도록 관리 할 수 있는 시스템이다.

데이터 베이스의 특성은 질의에 대한 실시간 처리 및 응답 할 수 있는 실시간 접근성(real-time accessibilities), 동적 특성인 갱신, 삽입, 삭제를 할 수 있는 지속적인 변화(continuous evolution), 여러 사용자가 동시에 사용하는 동시 공유(concurrent sharing), 위치나 주소가 아닌 값에 따라 참조하는 내용에 의한 참조(content reference), 지속성(persistent), 상호관련성(inter-related) 으로 이루어진다 [1].

본 논문의 2 장에서 이러한 데이터베이스를 구현하는 언어와 이의 역사, 종류, 현황을 알아보고, 3 장에서 데이터베이스의 분류와 쓰임에 대한 간단한 예시를 알아본다. 4 장에서 이를 이용해 간단한 프로그램을 구현하여 실험하며 마지막으로 5 장에서 결론 및 앞으로의 향후 과제에 관해 논한다.

### 2. 데이터 베이스

#### 2.1 역사[1]

데이터 베이스란 논리적으로 연관된 하나 이상의 자료의 모음으로 검색과 갱신의 효율화를 추구 한다. 이 용어가 처음 사용된 것은 1963 년 6 월 제 1 차 미국 SDC(System Development Corporation)가 산타모니카에서 개최한 '컴퓨터 중심의 데이터베이스 개발과 관리' 라는 심포지엄 제목에서 Data Base Systems 용어를 공식적으로 처음 사용되었다. 또한 1965.9 제 2 차 SDC 심포지엄의 주제인 컴퓨터 중심의 데이터 베이스 시스템에서 Data Base Systems 라는 용어를 처음 사용하였다. 그 후 아래 표와 같이 데이터베이스의 용어가 발전해 왔다.

연도	역사
1963	데이터베이스라는 용어가 'Development and Management of Computer Center Data Bases'라는 심포지엄에서 처음 사용
1963	최초의 범용 데이터베이스 관리시스템(DBMS) 설계: GE사에서 개발한 'Integrated Data Store'
1970	에드거 프랭크 코드(Edgar Frank Codd)는 관계형 데이터베이스 모델 제안
1976	피터 첸(Peter Chen, 陳品山, 천핀산)이 개체 관계(ER) 모델 제안
1980	개인용 컴퓨터를 위한 데이터베이스 관리시스템(DBMS) 개발(dBASE, Paradox 등)
1983	상용 관계 데이터베이스 관리시스템(DBMS) 등장(DB2, Oracle, Sybase 등)
1986	데이터베이스를 다루는 언어인 SQL(Structured Query Language)이 관계형 데이터베이스 관리시스템의 표준 언어로 채택
1990년대	상용 객체지향 데이터베이스 관리시스템(DBMS), SQL(Structured Query Language), <b>병렬 데이터베이스</b> 등장
2000년대	XML(eXtensible Markup Language)과 연관된 <b>질의 언어</b> 인 XQuery가 등장

(그림 1) 데이터베이스의 발전 과정

또한 시장에서 DBMS 의 발전 방향은 대략적으로 다음과 같다 [3.]

1. 확장 가능한 프로그래밍 환경 포함
2. 인터넷 및 XML 지원 강화
3. 개발 및 관리 환경의 통합
4. 대용량/동시접속 처리, 무 정지 시스템 지원
5. 보안 강화

## 2.2 SQL 과 NoSQL

SQL 이란 관계 데이터 베이스 표준 언어이다. 데이터 베이스를 사용할 때 데이터 베이스에 접근 할 수 있는 데이터베이스 하부 언어이다.

### 2.2.1 SQL [4]

SQL(Structured Query Language) : IBM 에서 1970 년대 초반 도널드 D. 챔벌린과 레이먼드 F. 보이스가 처음 개발 하였다 [3]. 관계형 데이터베이스 관리 시스템(RDBMS : Relational DataBase Management System)의 데이터를 관리하기 위해 설계된 프로그래밍 언어 이다. 관계형 데이터베이스 관리 시스템에서 자료의 검색과 관리, 데이터베이스 스키마 생성과 수정, 데이터베이스 객체 접근 조정 관리를 위해 고안 되었다.

### 2.2.2 NoSQL

NoSQL 데이터베이스는 전통적인 관계형 데이터베이스 보다 덜 제한적인 일관성 모델을 이용하는 데이터의 저장 및 검색을 위한 매커니즘을 제공한다. 이러한 접근에 대한 동기에는 디자인의 단순화, 수평적 확장성, 세세한 통제를 포함한다. NoSQL 데이터베이스는 단순 검색 및 추가 작업을 위한 매우 최적화된 키 값 저장 공간으로, 레이턴시와 스루풋과 관련하여 상당한 성능 이익을 내는 것이 목적이다. NoSQL 데이터베이스는 빅데이터와 실시간 웹 애플리케이션의 상업적 이용에 널리 쓰인다. 또한 NoSQL 시스템은 SQL 계열 쿼리 언어를 사용할 수 있다는 사실을 강조한다는 면에서 "Not only SQL"로 불리기도 한다.

## 2.3 DBMS 의 종류[5]

- 계층형/네트워크 DBMS
  - 1970 년대부터 1980 년대 중반까지 Mainframe DBMS 시장을 주도
  - 데이터의 집합체는 트리 또는 네트워크 형태로 표현
  - 특정한 검색에 대해 매우 빠른 처리 성능
  - 비교적으로 H/W Resource 적게 소요
  - 내용의 변화가 있을 경우 유지보수 어려움.

## - 관계형 DBMS

- 1980 년 후반부터 점진적 관계형으로 이행
- 데이터에 특정 형태를 가지지 않고 자연스러운 검색 가능.
- 조작성과 유연성이 뛰어난
- 비교적으로 H/W Resource 많이 차지

## - 객체지향 DBMS

- 데이터를 일정한 형태로 저장할 필요 없어 갱신이 용이
- 데이터 처리가 용이
- 관계형에 비해 처리속도가 우수
- CAD/CAM 분야나 멀티미디어 DB 구축을 위한 도구로 주목

## 2.4 데이터 베이스 언어 현황

334 systems in ranking, September 2017

Rank			DBMS	Database Model	Score		
Sep 2017	Aug 2017	Sep 2016			Sep 2017	Aug 2017	Sep 2016
1.	1.	1.	Oracle 📈📈	Relational DBMS	1359.09	-8.78	-66.47
2.	2.	2.	MySQL 📈📈	Relational DBMS	1312.61	-27.69	-41.41
3.	3.	3.	Microsoft SQL Server 📈📈	Relational DBMS	1212.54	-12.93	+0.99
4.	4.	4.	PostgreSQL 📈📈	Relational DBMS	372.36	+2.60	+56.01
5.	5.	5.	MongoDB 📈📈	Document store	332.73	+2.24	+16.74
6.	6.	6.	DB2 📈	Relational DBMS	198.34	+0.87	+17.15
7.	7.	8.	Microsoft Access 📈	Relational DBMS	128.81	+1.78	+5.50
8.	8.	7.	Cassandra 📈	Wide column store	126.20	-0.52	-4.29
9.	9.	10.	Redis 📈	Key-value store	120.41	-1.49	+12.61
10.	10.	11.	Elasticsearch 📈	Search engine	120.00	+2.35	+23.52

(그림 2) 데이터베이스 언어 선호도 [6]

위 표를 보면 현재 사용되고 있는 DBMS 언어의 빈도를 알 수 있다. 그 중 상위 3 개의 언어는 압도적으로 선호되어 많이 쓰이고 있으며, 세 언어 모두 관계형 DBMS 이다. 이에 따라 우리는 상위 2 개에 해당하는 SQL 언어 Oracle 과 MySQL 을 이용하여 간단한 기능을 지닌 SQL 문 생성 프로그램을 구현하고자 한다.

## 3. SQL 명령어 분류와 문법

정의어(DDL: Data Definition Language)는 정의 기능으로서 다양한 형태의 요구를 지원 할 수 있도록 가장 적절한 데이터 베이스 구조를 정의 할 수 있는 기능이다. 데이터 구조 정의에 고려해야 할 사항들은 목표 DBMS 가 지원하는 데이터 모델에 맞게 기술해야 하며, 물리적 저장 장치에 저장하는데 필요한 명세를 포함 해야 한다. 또한, 데이터의

논리적 구조와 물리적 구조 사이의 사상(mapping)을 명세 한다.

정의어 (DDL : Data Definition Language)의 예시로는 CREATE, ALTER, DROP 이 존재한다. CREATE 은 TABLE, INDEX, VIEW 를 생성하며, ALTER 는 TABLE 의 변경, DROP 은 TABLE, INDEX, VIEW 의 제거에 사용된다.

조작어(DML:Data Manipulation Language)는 사용자와 데이터베이스 사이의 인터페이스를 위한 수단을 제공한다. 사용자의 요구에 따라 체계적으로 데이터베이스를 접근하고 조작 가능해야 한다. 조작 기능에 고려해야 할 사항들은 쉽고 자연스러운 조작 방법과 명확하고 완전한 데이터 사이의 명세가 가능해야 하며, 효율적인 데이터 접근, 처리가 가능하도록 고려해야 한다.

조작어(DML : Data Manipulation Language)의 예시로는 SELECT, UPDATE, INSERT, DELETE 가 존재한다. SELECT 는 데이터 베이스의 데이터 검색, UPDATE 는 데이터베이스의 데이터 갱신, INSERT 는 데이터베이스의 데이터 삽입, DELETE 는 데이터베이스의 데이터 삭제에 사용된다.

제어어(DCL : Data Control Language)는 제어 기능으로서 데이터의 정확성과 보안성을 유지하는 기능이다. 제어 기능에 고려해야 할 사항들은 무결성(integrity) 유지, 보안(security), 권한(authority) 검사, 병행 제어(concurrency control), 회복(recovery)을 고려해야 한다.

제어어(DCL : Data Control Language)의 예시로는 GRANT, REVOKE, COMMIT, ROLLBACK 이 존재한다. GRANT 는 사용자에게 권한을 부여하고, REVOKE 는 사용자의 권한을 해제하고, COMMIT 은 데이터를 데이터 베이스에 저장하고 트랜잭션을 성공적으로 종료하며, ROLLBACK 은 데이터의 변경 사항을 취소하고 원상태로 복귀한 후 트랜잭션을 종료한다.

## 4. SQL 생성문 구현

### 4.1 SQL 변환기 필요성

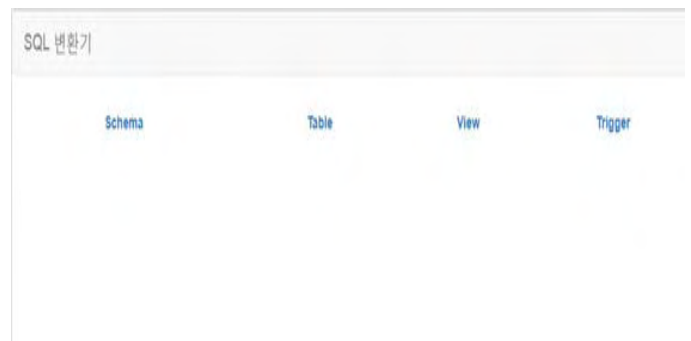
MySQL 의 경우 Workbench 라는 MySQL GUI 툴이 존재하며 Oracle DB 의 경우 SQL Developer 라는 GUI 툴을 제공한다. 이와 같은 다양한 GUI 툴 제공으로 인해 많은 프로그래머들은 실제 프로그램 개발 시 SQL 문을 사용하지 않게 되는 경우가 많이 발생한다. GUI 는 Query 문 작성 없이 간단한 클릭과 텍스트 입력으로 데이터베이스 설계부터 개발 관리를 도와주는 장점을 가진다. 하지만 Query 문에 대한 사용이 급격히 줄어들기 때문에 간단한 SQL 문조차 잊어버리게 되는 경우가 발생하게 된다.

이와 같은 상황을 해결하고자 SQL 변환기를 통해 가장 많이 쓰이는 MySQL 과 Oracle DB 의 SQL 문을 원하는 질의 형태로 만들어주는 도구가 필요할 것으로 예상된다.

### 4.2 모델링

먼저 사용자가 변환할 정보를 입력할 수 있도록 웹 페이지 구현한다. 그 다음 사용자가 변환할 정보 입력 시 그에 맞는 SQL 쿼리 문법이 나올 수 있도록 웹 페이지 구현한다. 그리고 사용자가 변환할 정보를 입력할 경우 그에 맞는 SQL 쿼리문이 나오도록 틀을 만들고 구현한다. 최종적으로 정보를 입력하면 SQL 쿼리 문법이 출력된다

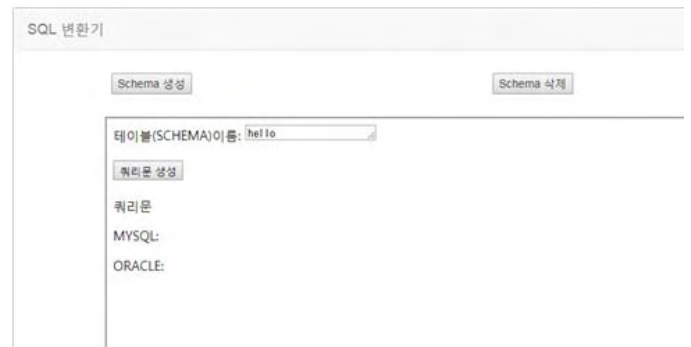
### 4.3 구현



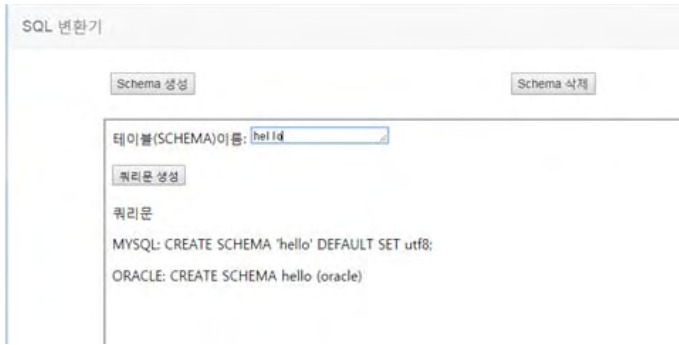
(그림 3)생성할 SQL 문 종류 선택을 선택한다.



(그림 4)선택 후 화면이 나오면 입력하는 칸이 있다.



(그림 5) 생성할 테이블(SCHEMA) 이름 입력한다.



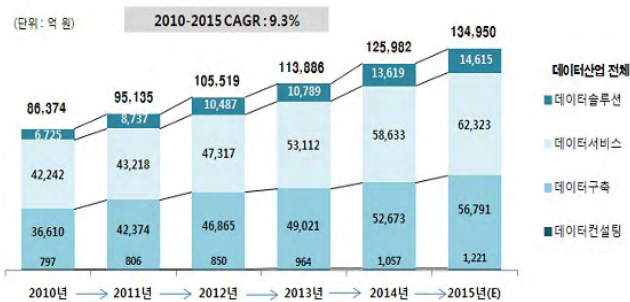
(그림 6) 입력한 값에 따라 코드가 생성된다.

이 아닌 비전문가들도 쉽게 다룰 수 있도록 방법을 지속적으로 연구 해야 할 필요가 있다.

## 참고 문헌

- [1] 이석호, 데이터베이스론, 정익사, 2010
- [3] 손호성, SQL SERVER 2005 완벽가이드, 영진닷컴, 2007
- [4] Chamberlin, Donald D, Boyce, Raymond F, SEQUEL : A Structured English Query Language, IBM Research Laboratory, 1974
- [5] 한국데이터베이스진흥센터, 데이터베이스월드 1993년 10월호, 한국데이터베이스진흥원, 1993
- [6] <https://db-engines.com/en/ranking>, 2017년 9월

## 5. 결론



(그림 7) 데이터산업 시장 규모

출처: 데이터 전문가 지실 포털

‘그림 7’에 보면 국내 데이터 시장 규모는 2014년 전년대비 6.3% 성장한 12조 5,982억 원이며, 2015년에는 13조 4,950억 원으로 전년 대비 7.1% 성장할 것으로 예상하였다. 2010년 이후 연평균 성장률은 9.3%로 매년 성장세를 유지하고 있는 것으로 나타났다. 또한 빅 데이터로 인해 향후에도 꾸준히 성장할 것으로 전망된다.

이에 따라 수집된 데이터들에 대한 관리가 더욱 중요해지는데 이에 대한 관리가 SQL을 통해 이루어지는 점을 바라볼 때 단순한 GUI에서 이루어지는 작업보다 내부에서 이루어지는 Query 문 자체에 대한 이해도가 뒷받침되어야 수 천억 개에 달하는 데이터들에 대한 관리가 제대로 이루어 질 것이다.

따라서, Query 문에 대해 쉽게 접근할 수 있도록 SQL 변환기를 제공 함으로써 일일이 전체 코드를 입력하지 않고, 필요한 몇몇 코드만을 입력해 얻고자 하는 Query 문을 얻을 수 있다.

향후 SQL 변환기에 대한 추가적인 개발과 유지보수가 이루어 진다면 Query 문에 대한 교육 및 기존 개발자들의 참고자료로 활용될 것이다.

현재 세계적으로 컴퓨터 교육산업이 성장하고 있는 추세이기 때문에, 데이터베이스 분야도 많은 사람들의 수요가 생길 것으로 예상된다. 하지만 데이터 베이스화는 현재 비전문가들에겐 매우 어려운 분야라고 할 수 있다. 따라서 향후 데이터베이스화를 전문가만