# PDA기반 단일사용자 데이터베이스 관리 시스템 설계 및 구현

노승기°, 이정준, 이상호, 오용철
한국산업기술대학교 컴퓨터공학과
{ Mesia°, jjlee, shlee, oh }@kpu.ac.kr

# Design and Implementation of Single User DBMS based on PDA

Seung-Ki Noh°, Jeong-Joon Lee, Sang-Ho Lee, Yong-Chul Oh
Department of Computer Engineering
Korea Polytechnic University

요 약

최근의 PDA 및 모바일 장치의 데이터 저장 용량 및 CPU 성능이 향상됨에 따라 이를 이용한 다양한 서비스가 가능해질 것으로 기대되고 있다. 이러한 서비스는 대용량의 데이터 저장과 빈번한 데이터의 처리, 그리고 신뢰성을 요구하고 있어 그에 맞는 데이터베이스 관리 시스템 개발이 필요하다. 본 연구 목적은 1) PDA 장치의 특성 및 서비스 형태를 고려한 데이터베이스 관리 시스템 설계 안을 제시하고, 2) 구현 환경 및 결과를 설명하여, 3) PDA 기반의 모바일 데이터베이스 관리 시스템의 국산화에 기여하는 것이다.

## 1. 서 론

사용자에게 휴대성의 매력을 주는 PDA(Personal Digital Assistant)는 방송의 디지털화(DMB)와 홈 네트워크 등 새로운 기술과 융합되어 다양한 형태의 멀티미디어 콘텐츠 서비스를 제공할 것으로 기대되고 있다. 최근에 PDA의 데이터 저장 용량 및 CPU 속도의 향상은 그 가능성을 뒷받침 한다. 그러나 기존의 데이터 처리 방식으로는 사진 및 동영상 등 멀티미디어 콘텐츠 서비스를 위한 대용량의 데이터 저장과 신속한 처리가 사실상 어렵다. 따라서 데이터베이스를 활용한 서비스가 필요하다. 또한 최근 기업에서는 PDA 또는 이와 유사한 장치를 통해서 모바일 SFA(영업지원시스템), WMS(창고관리), CRM(고객관리), TMS(차량 관리), 주문관리 등 다양한 산업 분야에 걸쳐서 데이터베이스를 필수적인 업무 시스템으로 요구하고 있다.

위와 같은 필요성에 따라 PDA 용 데이터베이스 관리 시스템은 그 업무와 서비스, 그리고 PDA 장치의 특성 등을 고려하여 설계되어야 한다. 예를 들어, 실시간 데이터 처리가 중요한 서비스에 사용되는 PDA용 데이터베이스 관리 시스템의 경우, 다양한 기능을 제공하기 위해서 범용 관계 연산을 추가하기 보다는 PDA 장치의 성능을 고려하여 자주 쓰이지 않는 연산들을 찾아 제외시키거나 전체 시스템에 큰 부하를 줄 수 있는 요인들을 찾아서 좀 더 단순하게 설계 할 필요가 있다. 이렇게 함으로써 개발 시간을 단축하고 목적에 맞는 데이터베이스 관리 시스템을 개발할 수 있다.

본 논문에서는 이와 관련한 내용으로 PDA 장치의 특성 및 서비스 형태를 고려한 데이터베이스 관리 시스템 설계 안을 제시하고, 구현 환경 및 결과를 설명하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 PDA 장치의 특성 및 서비스 시나리오를 알아본다. 그리고 3장에서는 PDA 기반 데이터베이스 관리 시스템 아키텍처를 설명한다. 4장에서는 Query Processor 설계 안을 제시하고, 5장에서는 구현 환경 및 결과를 설명하여 결론을 맺는다.

## 2. PDA 장치의 특성 및 서비스 시나리오

### 2.1. PDA 장치의 특성

PDA와 같은 모바일 장치에서는 Desktop PC와는 달리, 하드 디스크 장치가 없으므로 데이터를 영구적으로 저장하기 위해서는 플래시 메모리 영역에 데이터를 저장해야 한다. 또한 64MB 이하의 작은 메인 메모리(RAM)가 탑재되어 있으므로 메모리 사용을 최소화하여 효율적인 메모리 접근을 해야 한다.

PDA 장치는 제품을 출시한 회사나 모델에 따라서 장착한 CPU 계열이 다를 수 있다. 지금까지는 OS의 기능 한계로 인해 CPU 계열에 따른 프로그램 개발이 필요하다. 그리고 전원 공급에 있어서, 사용시간에 제약이 있는 배터리를 이용하기 때문에 전원 공급의 갑작스런 중단이 발생 할 수 있다. 따라서 프로그램 개발자는 이러한 PDA 장치의 특성에 유념할 필요가 있다.

### 2.2. 서비스 시나리오

모바일 환경에서 데이터베이스를 활용한 비즈니스로 이동 업무가 많은 운송 · 물류 · 영업 분야를 들 수 있다. 이러한 분야에는 현장근무요원이 주요 구성원으로 조직되어 있으며, 현장에서 필요한 정보의 실시간 접근과 신속한 업무 처리가 요구된다.

그림 2-1은 PDA를 이용한 영업지원 시스템 시나리오를 보여준다. 영업 사원이 현장에서 고객과 상담을 하면서 PDA 장치를 이용하여 고객에 대한 정보를 조회하거나 신규입력을 한다. 영업 사원은 업무가 끝나면 회사에 직접 가지 않고 회사 내에 있는 SFA DB Server와 모바일 데이터베이스를 동기화 하여 업무 보고를 끝내고 귀가한다.

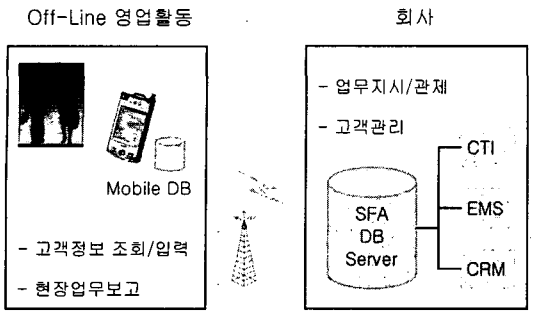


그림 2-1 영업지원시스템(SFA) 시나리오

## 3. 모바일 데이터베이스 관리 시스템(DBMS) 아키텍처

기본적으로 데이터베이스 관리 시스템은 그림 3-1 과 같이 크게 질의 처리기(Query Processor)와 저장 시스템(Storage System)으로 이루어진다. [1]

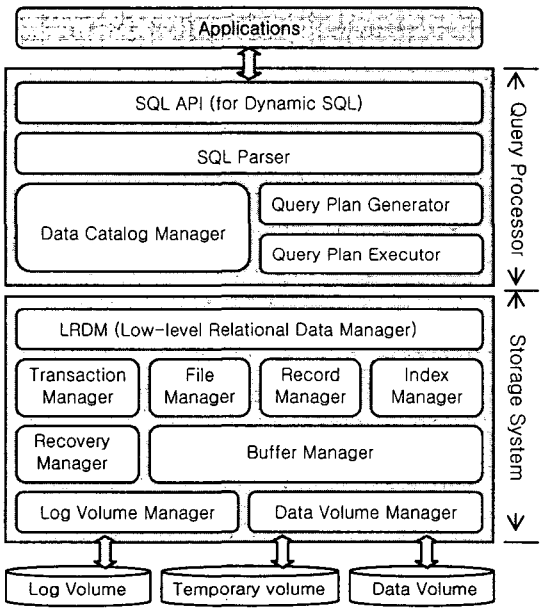


그림 3-1 DBMS의 구성도

질의 처리기(Query Processor)는 사용자의 SQL질의를 입력 받아 그것을 분석하고 구체적인 실행 계획을 세워 실행하도록 저장 시스템에 명령하는 모듈이다. 따라서 질의 처리기 내부에는 기능별로 SQL질의를 입력 받는 SQL API와 데이터베이스 스키마의 생성 및 조회를 수행하는 Data Catalog Manager, 그리고 기본적인 SQL 문법 및 사용된 스키마의 적절성을 체크하여 AST(Abstract Syntax Tree)[2]를 생성하는 SQL Parser, 이와 같이 생성된 AST(Abstract Syntax Tree)를 토대로 하여 질의 계획(Query Plan)을 생성하는 Query Plan Generator, 마지막으로 질의 계획(Query Plan)에 따라 실행을 하는 Query Plan Executor로 나누어 볼 수 있다.

저장 시스템(Storage System)은 실질적으로 데이터를 디스크에 저장하면서 파손회복, 트랜잭션 관리 등의 기능을 수행한다. 저장 시스템의 구조는 레코드 단위로 데이터를 관리/검색하는 Record Manager 와 File Manager, 효율적인 검색을 위해 인덱스 사용을 제어하는 Index Manager, 갑작스런 정전과 같은 상황에 대비해 데이터베이스를 일관된 상태로 복귀시키는 Transaction Manager와 Recovery Manager, 데이터 볼륨에서 페이지 단위로 데이터를 가져오는 Data Volume Manager, 그리고 로그볼륨에서 데이터를 가져오는 Log Volume Manager, 마지막으로 이들을 버퍼링하는 Buffer Manager로 구성된다.

## 4. Query Processor 설계 방법

먼저, SQL 질의 처리 흐름[3]을 알아보고 SQL 지원 범위와 주요 모듈의 설계 방법을 논의하도록 할 것이다.

### 4.1. SQL 질의 처리 흐름

SQL 질의는 그림 4-1과 같은 처리흐름을 갖는다. SQL 질의가 들어오면, 정의된 문법(Grammar)에 따라 Parser 에서 기본적인 문법 체크를 하여 AST(Abstract Syntax Tree)를 생성한다. 이후에 Query Plan Generator가 Catalog를 참조하여 생성된 AST에 대한 의미검사(Semantic Check)를 수행하고 질의계획(Query Plan)을 작성한다. 질의 계획에는 질의 처리 순서와 사용할 인덱스를 구체적으로 기술하고 있다. 따라서 Query Plan Executor는 이 질의 계획을 그대로 수행하여 결과를 반환하게 된다.

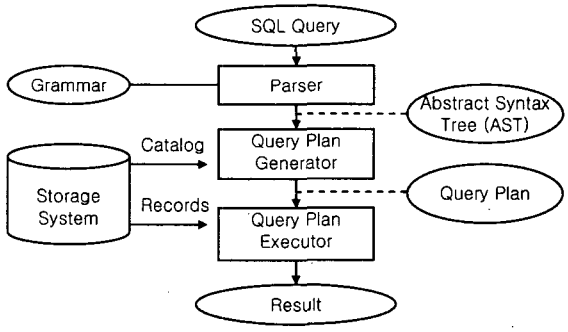


그림 4-1 SQL 질의 처리 흐름도

### 4.2. 주요 모듈의 설계

Query Plan Generator 는 Parser를 통해 생성된 AST를 이용하여 Query Plan을 생성하는 모듈이다. Query Plan 은 질의를 수행하는 방법을 트리 형태로 구성한 데이터 구조이다. Query Plan 내에서의 트리 노드는 레코드 반환을 요구하는 질의와 레코드 반환 없이 수행만을 요구하는 질의에 따라서 반복노드(iterative node)와 비반복노드(non-iterative node)로 분류되어 형성된다. 한 개 이상의 레코드를 반환하는 노드로서 Selection, Projection, Join, Order-by, Aggregate 등이 있으며, 레코드의 반환 없이 단순히 처리만을 하는 노드로서 create_table, drop_table, deletion, update 등이 있다. 그림 4-2 에서 보여준 예제[4]를 응용하여 Query Plan을 작성한다면, Qeury Plan Generator 모듈을 만들 수 있다.

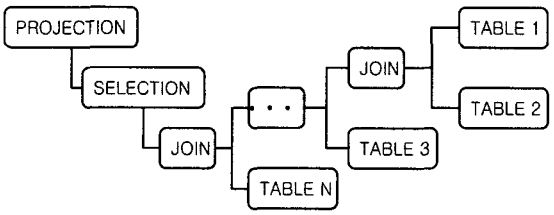


그림 4-2 SELECT 문의 Query Plan 예제

Query Plan Executor 는 생성된 Query Plan 을 그대로 실행하는 모듈이다. 그림 4-2 를 예로 들어 설명하자면, TABLE1 노드에서 레코드를 하나 얻어, TABLE2에서 얻은 레코드와 조인되는 레코드를 하나 추출하고, 이를 TABLE3과 TABLE N 노드와 조인하여 하나의 결과를 가져오면, SELECTION 노드에서 사용자가 요구한 조건과 일치하는지 검사하여 PROJECTION 노드에서 해당 컬럼을 출력한다. 이 작업을 테이블에서 가져올 레코드가 더 이상 없을 때까지 반복 한다.

### 4.3. SQL 지원 범위

2장에서 언급한 서비스 시나리오에 따라 이러한 비즈니스업무는 비교적 간단한 질의처리만을 요구한다. 따라서 ANSI SQL92[5]의 기능을 대부분 지원하되, 거의 사용하지 않는 기능을 일부 축소시켜 실행프로그램의 크기를 줄인 다면 프로그램 효율성을 증대시킬 수 있다.

구체적인 SQL 질의 범위는 표 4-1과 같이 제안한다.

표 4-1 SQL 질의 범위

| | SQL 질의 문 | 의미 |
|---|---|---|
| DDL | Create Table | 테이블 생성 |
| | Create Index | 인덱스 생성 |
| | Drop Table | 테이블 삭제 |
| | Drop Index | 인덱스 삭제 |
| DML | SELECT FROM WHERE GROUP BY HAVING ORDER BY | 질의 검색 |
| | Insert Into Values | 테이블에 레코드 삽입 |
| | Insert Into Select From Where Group By Having Order By | 질의 검색 결과를 테이블에 삽입 |
| | Delete From Where | 테이블의 레코드 삭제 |
| | Update Set Where | 테이블의 레코드 수정 |

## 5. 개발 환경 및 결과

본 논문에서 기술한 내용을 바탕으로 한국산업기술대학교(Korea Polytechnic University)에서 개발한 *SubDBMS*의 개발환경을 설명 하고자 한다. *SubDBMS*는 모바일 데이터베이스 관리 시스템이며 윈도우 XP 운영체제를 기반으로 Embedded Visual C++ 4.0 개발 툴에서 대부분 개발되었다. 그러나 Query Processor 모듈 만은 Python 언어로 작성되었는데, 이는 개발 시간 단축과 성능을 참작한 선택이었다. Storage System 모듈을 C 언어로 개발하여 빠른 수행 속도를 모색하였고, Query Processor 모듈은 C언어 대신 Python 언어로 작성하여 6개월간의 개발 시간을 단축하게 되었다.

이러한 단축에도 불구하고 다른 모바일 상용 DBMS와 성능을 비교하면 성능에는 차이가 거의 없다.

개발된 *subDBMS*는 약 1.6MB 용량을 차지하며 ARM 계열의 PDA 에서 테스트되었다. 이동환경의 제약을 고려한 용량축소와 PocketPC2003에 맞추어 ARM 계열 CPU 에서 구동 되도록 개발되었다.

그림 5-1은 개발 언어간의 계층 관계를 보여준다. Query Processor 모듈이 Python 언어로 개발되었기 때문에 다른 모듈과의 연결을 위해 특별히 Binding모듈[6]이 추가된 모습이다. D-Parser 모듈은 Query Processor 모듈에 속하나 C언어로 개발되어, 분리된 형태를 보여주고 있다.

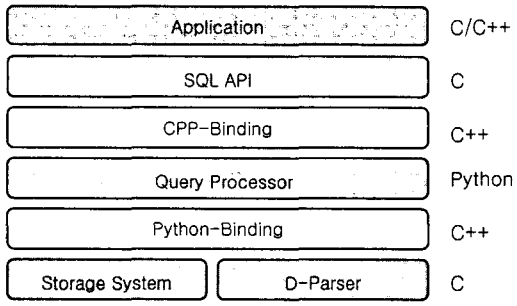


그림 5-1 언어별 DBMS 설계 구조

## 6. 결 론

PDA 장치 및 모바일의 성능 향상으로 인해 모바일 비즈니스가 활성화되고 있는 시점에서, PDA기반 모바일 데이터베이스 시장에 우리나라 제품의 경쟁력 확보가 중요하게 되었다. 이러한 경쟁력은 PDA 장치의 특성과 서비스 형태에 적합한 설계를 토대로 구현할 때 갖출 수 있다. 향후에는 Python 실행 모듈 최적화를 통한 용량 축소, 동기화기( Synchronizer)개발을 예정하고 있다.

## 참고 문헌

[1] Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database System Implementation" , Prentice Hall book , 2000

[2] Robert E. Noonan, "An Algorithm for Generating Abstract Syntax Trees", Comput. Lang. 10(3/4): 225-236 (1985)

[3] E. Wong and K. Youssefi, "Decomposition - a strategy for query processing," ACM Trans. on Database Systems 1:3 (1976), pp. 223-241

[4] L.R Gotlieb, "Computing joins of relations," Proc. ACM SIGMOD Intl. Conf. on Management of Data (1975), pp. 55-63

[5] http://www.programmar.com/ex_sql2.htm

[6] Mark Lutz "Programming Python" , O'Reilly book , March 2001