

OCAML RDBMS Initial Project Charter

Sal Razzaq

Version 1.0: July 6, 2006

Contents

1	Abstract	3
2	Description	3
3	System Components	3
4	Scope	4
5	Demonstration	4
6	Schedule	5
7	Members	5
8	Website	5

1 Abstract

This project models an In-Memory Relational Database Management System using facilities of a Functional Language. In particular, we use a flavor of ML (Meta Language) called OCAML (Objective Caml - <http://caml.inria.fr/>) to implement a working Relational Database Management System.

2 Description

Relational Database Management Systems (RDBMS) store data as a bag of tuples. A bag of tuples is called a *Table* in RDBMS and each tuple in the table is called a *Row*.

Relational Database Management Systems use SQL, a query language, as an interface to the data. By issuing declarative statements written in SQL, users can retrieve data they desire. One of the key SQL phrases is the WHERE clause in a SELECT statement. The WHERE clause allows a user to filter rows of data contained in a table based on a predicate.

OCAML supports user-defined data types, tuples as well as lists of tuples. These features of OCAML can be used as building blocks for a RDBMS. OCAML also provides elegant ways of creating and applying High Order Functions (HOFs) for performing tasks such as filtering or folding based on a predicate.

For an implementation of a RDBMS in OCAML, we also need to interpret SQL. OCAML provides rich functionality to lex (ocamllex tool) and parse (ocamlyacc tool) statements.

3 System Components

1. A lexer to tokenize SQL statements
2. A parser to synthesize tokens into an Abstract Syntax Tree

3. An interpreter called "mssql" to execute the SQL Commands
4. A library of OCAML functions called "mssqlib.cmo" to support execution of the SQL Commands

4 Scope

The project will implement the following subset of SQL commands:

- CREATE DATABASE
- CREATE TABLE
- DROP TABLE
- INSERT
- DELETE
- UPDATE
- SELECT

In order to cover all the basic SQL statements needed for an RDBMS to be functional, limited number of options will be implemented for each of the above mentioned SQL constructs. For example, the initial implementation will support aggregation (GROUP BY) in SQL SELECT Statements to demonstrate folding recursion but SELECT will be limited to operations on a single table in a given SQL statement.

5 Demonstration

1. Deploy mssqlib.cmo to your OCAML system
2. Open the mssqlib in the OCAML interpreter
3. Issue SQL commands such as:

- `let dbms = msql dbms "CREATE TABLE mytable (name string, score int)";;`
- `let dbms = msql dbms "INSERT mytable (name, score) VALUES ('joe', 99)";;`
- `let mytable = msql dbms "SELECT * FROM mytable WHERE name = 'joe'";;`

6 Schedule

June 23, 06 Initial Project Proposal

July 17, 06 Email Status Report

August 4, 06 Project Submission/Demo

7 Members

Sal Razzaq

8 Website

<http://ml-dbms.sourceforge.net/>