Embedded SQL

Extract from C program:

```
EXEC SQL BEGIN DECLARE SECTION;
short DEPT, DNO;
char  NAME[26];
EXEC SQL END DECLARE SECTION;

main()
{
    GetInput(&DEPT);
    EXEC SQL DECLARE C1 CURSOR FOR
        SELECT ENAME, DEPTNO FROM EMP WHERE DEPTNO = :DEPT;
    EXEC SQL OPEN C1
    while (SQLCODE == 0) {
        EXEC SQL FETCH C1 INTO :NAME, :DNO
    }
    EXEC SQL CLOSE C1
}
```

Precompilers from different vendors will produce different code.

Precompiling Embedded SQL

The precompiler and DBMS interact in several ways:

- The DBMS parser checks the syntax of the SQL statements.
- The DMBS checks the semantics of the SQL statements (table and column names are correct, data types of host language variables are correct, etc.)
- If the SQL statements are OK, optimise them to produce an access plan, which is stored in the database.
- Relevant sections of the access plan are stored as arguments to function calls generated by the precompiler.

Call level interface (CLI)

An application programming interface (API), i.e. it provides a set of function calls.

In the SQL standards world, “CLI” means “not Embedded SQL” (which is also referred to as and API).

- a clean separation between the application program and SQL.
- building a program is simpler; no preprocessor (although calls to the DBMS runtime library are the same).
- debugging is simpler; you debug your own code rather than code generated by a precompiler.

Programming models for relational systems

What if we want to change the DBMS that our application uses?

Embedded SQL

Requires compiling application program using new DBMS vendor’s precompiler. We might not have the source code (e.g. “shrink-wrapped” applications).

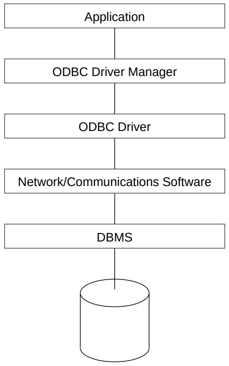
Vendor-specific API

Requires changing the source code to use the new DBMS vendor’s API, and recompiling with the new DBMS vendor’s libraries. Again, we might not have the source code.

ODBC

Edit the “.odbc.ini” file to refer to a new ODBC driver. Do not change the source code. Do not recompile. ODBC driver manager will pick up the new driver at runtime.

ODBC architecture



Application

ODBC Driver Manager

ODBC Driver

Network/Communications Software

DBMS

ODBC SQL execution models

ExecDirect

Combine all steps in a single function call (SQLExecDirect).

Prepare/Execute

Separate steps 1, 2 and 3 from 4 and 5. Good when same SQL statements (possibly with parameters) are executed repeatedly; only needs to be prepared once, and access plan will be execute immediately each time. Access plan is deleted from DBMS after program ends.

Stored procedures

Like Prepare/Execute, but preparation step can be independent from application program. Stored procedure persist in database beyond the runtime of the application. Application doesn’t need to perform preparation phase.

Executing SQL statements

1. Formulate SQL statement
2. Send SQL statement to DBMS
3. Parse and optimise the SQL statement to produce an access plan.
4. Execute the access plan. Authorisation checks done here (although these could be done in step 3).
5. Client and server interact to send status information and data to the client as requested by the client.