Embedded SQL and the CLI

Chapter 7.7

© 1997 UW CSE

SQL is very high-level, but...

- Too low level for end-users
- Requires detailed knowledge of DB schema
- Unsatisfactory as a user interface
- Solution: *End user sees only an interface written by a DB programmer*
  - "host language" such as COBOL, C, VB, etc.
  - program contains the SQL, hard-coded or dynamically generated

"Impedance" Mismatch

- Host language doesn't understand SQL
- Host language data types vs SQL data types
- Conventional imperative languages don't match declarative outlook of SQL
  - Whole table operations
  - Say "what" not "how"
- Must somehow connect to the DBMS

Solutions

- "Cursor" concept
  - expose one row at a time
  - process a result table like a sequential file
- Embedded SQL
  - Standard SQL statements in host source, translated by preprocessor
- DBMS via an API (Application Programming interface)
  - Calls use host language conventions

CLI: Call-level Interface

- DB APIs have been around for decades
  - relational and non-relational
- "The" CLI is a recently standardized API
  - Calls to SQL in host language syntax
  - Added to the SQL Standard in 1995
  - Based on Microsoft ODBC (Open Database Connectivity)
  - Program declares a struct (class) for each row type (i.e. for each table)
    - "data exchange" supports data type conversion between DB format and host (e.g. C++)