Using SQL in an Application

- Database Applications
  - Can SQL do everything required from an application?
  - Can we use SQL together with other programming languages?
    - Host language provides features not included in SQL
      - **Example:** Control mechanisms, assignment statements, error handling

- How can we use SQL together with other programming languages?
Using SQL in an Application

• SQL constructs can be included in an application program in two different ways
  • Statement-level Interface (SLI)
    Program mixture of statements in two languages (host language, SQL)
    • Embedded SQL
    • Dynamic SQL
  • Call-level Interface (CLI)
    Application entirely in host language
    • Also decides the string variables with SQL at run time
    • Example: ODBC & JDBC
Embedded SQL

- Extend host language with SQL syntax
  - Schema to be known when program written
    Programmer knows names of tables, attributes,...
  - SQL available for preprocessing
EXEC SQL BEGIN DECLARE SECTION;
  unsigned long num_enrolled;
  char *crs_code, *semester;
EXEC SQL END DECLARE SECTION;

other host language declarations and statements

statements to set the variables semester and crs_code

EXEC SQL SELECT C.Enrollment
    INTO :num_enrolled
FROM Class C
WHERE C.CrsCode = :crs_code
    AND C.Semester = :semester;

the rest of the host language program
Dynamic SQL

• SQL statements constructed as values of strings variables by the host language at run time

• SQL statement might not be ready for the pre-compiler

• To make parameter information available at compile time
  • SQLDA (Descriptor Area)
  • USING/INTO
Example

printf(“Which column of CLASS would you like to see?”);
scanf(“%s”, column); // get user input(Enrollment or Room)

// Incorporate user input into SQL statement

```c
sprintf(my_sql_stmt,
   "SELECT C.%s FROM CLASS C \
       WHERE C.CrsCode = ? AND C.Semester = ?",
   column);
EXEC SQL PREPARE st1 FROM :my_sql_stmt1
EXEC SQL EXECUTE st1 INTO :some_string_var
   USING :crs_code, :semester;
```
Call-Level Interface (DB API)

- Single interface to possibly multiple DBMS engines
- Connect to a database
- Map objects between host language and DBMS
- Manage query results
ODBC & JDBC

• ODBC (Open DataBase Connectivity)
  ODBC was originally developed by Microsoft and Simba Technologies

• JDBC (Java DataBase Connectivity)
  • Sun developed as set of Java interfaces
    • javax.sql.*
Application Architecture

• Single-Tier
  • Historically, data intensive applications ran on a single tier which contained
    • The DBMS
    • Application logic and business rules
    • User interface

• Two-Tier
  • Client-server architecture

• Three-Tier (and multi-tier)
  • Used for many web systems
  • Very scalable
Two-Tier Architecture

- Client/server architecture
  - The server implements the business logic and data management
- Separate presentation from the rest of the application

Figure From Slides, CMPT 354: Database Systems I (Jiannan Wang - SFU), Fall 2018
Presentation Layer

• Responsible for handling the user interactions with the middle tier

• One application may have multiple versions that correspond to different interfaces
  • Web browsers, mobile phones,…
  • Style sheets can assist in controlling versions
Three-Tier Architecture

- Separate presentation from the rest of the application
- Separate the application logic from the data management
Business logic Layer

• The middle layer is responsible for running the business logic of the application which controls
  • What data is required before an action is performed
  • The control flow of multi-stage actions
  • Access to the database layer

• Multi-stage actions performed by the middle tier may require database access
  • But will not usually make permanent changes until the end of the process
    • E.g. adding items to a shopping basket in an Internet shopping site
Data Management Layer

• The data management tier contains one, or more databases
  • Which may be running on different DBMSs

• Data needs to be exchanged between the middle tier and the database servers
  • This task is not required if a single data source is used but,
  • May be required if multiple data sources are to be integrated
  • XML is a language which can be used as a data exchange format between database servers and the middle tier
Three-Tier Architecture & The Web

• In the domain of web applications three tier architecture usually refers to
  • Web server
  • Application server
  • Database server
Acknowledgements

I have used materials from the following resources in preparation of this course:

• Database Systems: The Complete Book

• **Database Systems: An Application Oriented Approach** (Kifer, Bernstein, Lewis)

• Database System Concepts: [https://www.db-book.com](https://www.db-book.com)

• Course offerings
  • CMPT 354 (Jiannan Wang - SFU): [https://sfu-db.github.io/cmpt354/](https://sfu-db.github.io/cmpt354/)
  • W 4111 (Eugene Wu - Columbia): [https://w4111.github.io/](https://w4111.github.io/)
  • CS 186 (Joe Hellerstein - Berkeley): [https://sites.google.com/site/cs186fall17/](https://sites.google.com/site/cs186fall17/)