Lecture 9:
• Dynamic SQL (cont)

Announcements:
• HW-2 out

Result Sets

Features vary from one DBMS to another
• Move forward only
• Move forward or backward one row at a time
• Move to arbitrary locations
• Modify or delete retrieved records
Q: What are some advantages/disadvantages?

Result Set placement
• Placed before first row of the result
• Getting next result (conceptually) “pulls” the value from the DBMS
• Some form of indication that last row was reached
Dynamic SQL

Our plan ...

- Briefly look at basic dynamic SQL APIs in three languages
- Start with Java and JDBC
- Then Python and C++
- Then basic web approaches (some simple PHP and Node.JS)

We’ll only cover the basics ...

- enough to give you a feel for how the API’s work
- which can be helpful for building simple apps
- and also even when using frameworks

Java and JDBC

Java Database Connectivity (JDBC) API

- A Java API for accessing RDBMSs (RDBMS independent)
- Each specific DBMS implements a JDBC “driver” (i.e., the API)
- Similar to MS ODBC
- Many languages today have DBMS-specific libraries
- ... but most follow JDBC/ODBC style

Must install the specific JDBC driver for your system

- Connector/J for MySQL
- This is just a jar file ...
- The JDBC API comes standard in Java (but not the Driver)
import java.sql.*;

public class MySQLQuery {
    public static void main(String[] args) throws Exception {
        String url = "jdbc:mysql://cps-database.gonzaga.edu/cpsc321";
        Connection cn = DriverManager.getConnection(url, "user", "password");
        Statement st = cn.createStatement();
        String query = "SELECT * FROM pet ORDER BY name";
        ResultSet rs = st.executeQuery(query);
        while (rs.next()) {
            String name = rs.getString("name");
            System.out.println("name = " + name);
        }
        rs.close();
        st.close();
        cn.close();
    }
}

Note: examples take shortcuts to fit on a slide

- no comments, exception handling (try-catch blocks), error checking, etc.

Java and JDBC: A Simple Example (cont)

To compile and run the program

$ javac MySQLQuery.java
$ java MySQLQuery
name = babe
name = bill
name = fido
name = hobbes
name = toto

Above works on ada ... on your system may need to include the jar:

    javac MySQLQuery.java
    java -cp .:mysql-connector-java-8.0.30.jar MySQLQuery

Note the MySQL connected can be downloaded from:


other bindings: https://dev.mysql.com/downloads/ (e.g., for C++)