**Databases**

- Used everywhere
  - Development (debugging)
  - General accounting
  - Web applications
    - Uses MySQL with a PHP3 web interface

**SQL (Structured Query Language)**

- Pretty much a standard (ANSI)
  - Many SQL servers have special features that are not standard (Microsoft, mSQL, MySQL, etc)

**JDBC (Java Database Connectivity)**

- An API which enables SQL calls to be made from Java, and also translates SQL data types to Java data types
- Advantages:
  - Very portable (Java runs on practically all platforms, and the API allows access to many databases without recompilation/relink, provided you have the drivers)
- Disadvantages:
  - Performance
    - Java is slow
    - Going through a translation driver slows the program down. If you want speed, it might be better to use a native API

---

**JDBC-How it works**

**JDBC-Code sample**
JDBD-Code sample (part 2)

JDBD-Code sample (part 3)

ODBC (Open Database Connectivity)
- API that interfaces with various databases (SQL, Oracle, Access, dBASE, etc.)
- Interfaces through a driver
- Advantages (mostly the same as those of JDBC)
  - A single API for all the databases supported
  - No recompiling or relinking necessary when adding a new database type. Simply install the driver

You have ODBC (Windoze users)

JDBC ⇔ ODBC Bridge
- The bridge is a JDBC driver which translates JDBC operations to ODBC operations
- There may not be a direct JDBC driver written for a particular database type of interest
- Obviously, there is greater overhead in using this bridge (but slower access is better than no access!!)