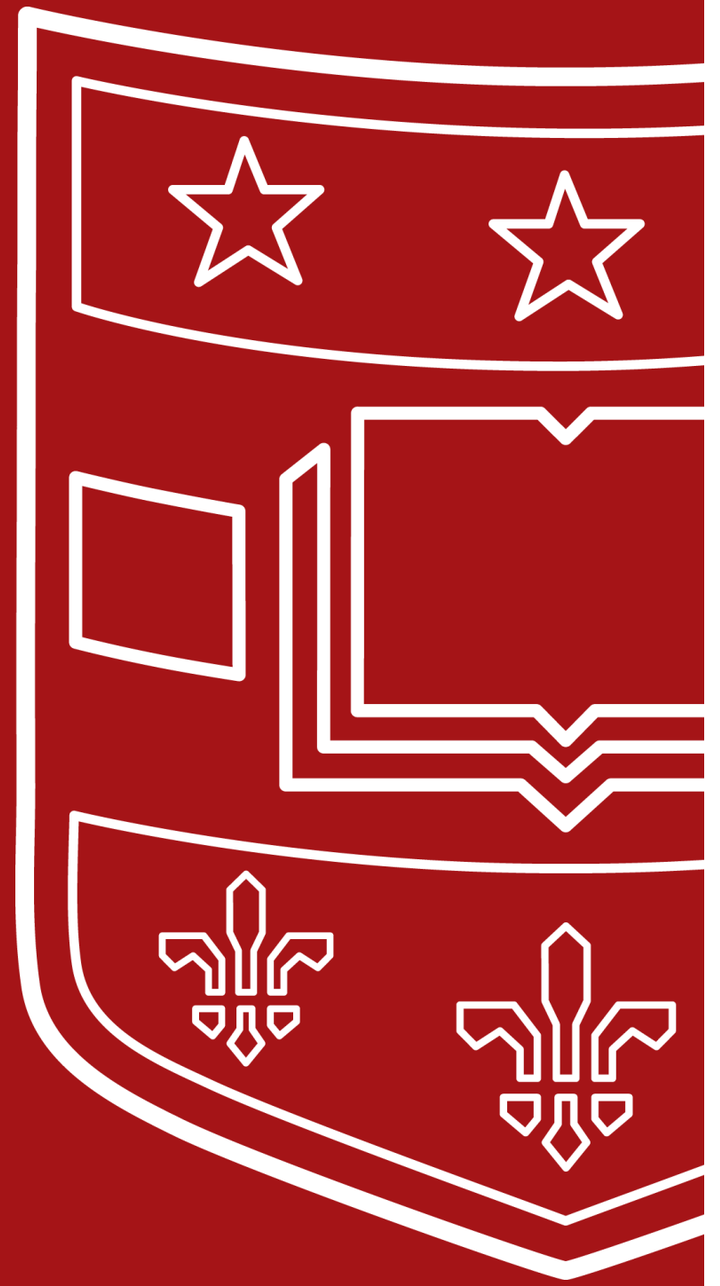
Database Management Systems



- Front End

Washington University in St. Louis

Front End Access

- Think about your users
  - Do they want to use a database?
  - Do they want to write SQL queries?

- How to access the database....
  - From a website?
  - From an application?

Basic Setup

- Install drivers
- Create a connection to the DB
- Prepare your query
- Execute the query
- Process results

PHP Examples

- Simple query
- Query with user input

Java Examples

- Simple Query
- Query with user input

Object Relational Mapping

- Maps table instances to objects
  - Minimizes the amount of SQL necessary
- Example

ORMs

- So, should you use an ORM?
  - Advantages?
  - Disadvantages?

SQL Injection

- Our applications must accept input from users
  - What if that input isn't what we expect?

- SELECT * FROM users WHERE username = ‘jake’ and PASSWORD = ‘jakespasswd’

- SELECT * FROM users WHERE username = ‘jake’ and (PASSWORD = ‘jakespasswd’ or ‘x’ = ‘x’)

SQL Injection

- Code Injection

- Function Call Injection
  - SELECT TRANSLATE (‘user input’, ‘from_string’, ‘to_string’) FROM dual;

  - SELECT TRANSLATE (“ || UTL_HTTP.REQUEST (‘http://129.107.2.1/’) || ”, '98765432’, ‘9876’) FROM dual;

SQL Injection Risks

- Database Fingerprinting
- Denial of Service
- Bypassing Authentication
- Identifying Injectable Parameters
- Executing Remote Commands
- Performing Privilege Escalation

Protecting Against SQL Injection

- Binding Parameters
  - PreparedStatement stmt = conn.prepareStatement(“SELECT * FROM EMPLOYEE WHERE EMPLOYEE_ID=? AND PASSWORD=?”);
    stmt.setString(1, employee_id);
    stmt.setString(2, password);
- Input Validation
- Function Security

SQL Injection Exercise

- http://sqlzoo.net/hack/