Lecture 9: Problems with Statement

Dynamic Values Problem:

```
// Variables with dynamic values
int sid = 101;
String sname = "Max";
int marks = 48;

// This WON'T work - variables not substituted
String sql = "INSERT INTO student VALUES (sid, sname, marks)";

// This works but is problematic
String sql = "INSERT INTO student VALUES (" + sid + ", '" + sname + "', " + marks + ")";
```

Major Problems with Statement:

1. SQL Injection Vulnerability:

```
// If sname comes from user input: "Max'; DROP TABLE student; --"
String sname = "Max'; DROP TABLE student; --";
String sql = "INSERT INTO student VALUES (" + sid + ", '" + sname + "', " + marks + ")";
// Results in: INSERT INTO student VALUES (101, 'Max'; DROP TABLE student; --', 48)

// This could delete your entire table!
```

2. String Concatenation Complexity:

```
// Complex and error-prone
String sql = "INSERT INTO student VALUES (" + sid + ", '" + sname + "', " + marks + ")";

// Must handle quotes, escaping, data types manually
```

3. Performance Issues:

- SQL is compiled every time
- No query plan reuse
- Database has to parse SQL each execution

4. Data Type Handling:

```
// Date/Time values are complex to handle
Date joinDate = new Date();
String sql = "INSERT INTO employee VALUES (" + id + ", '" + name + "', '" + joinDate + "')";

// Date formatting issues across databases
```