

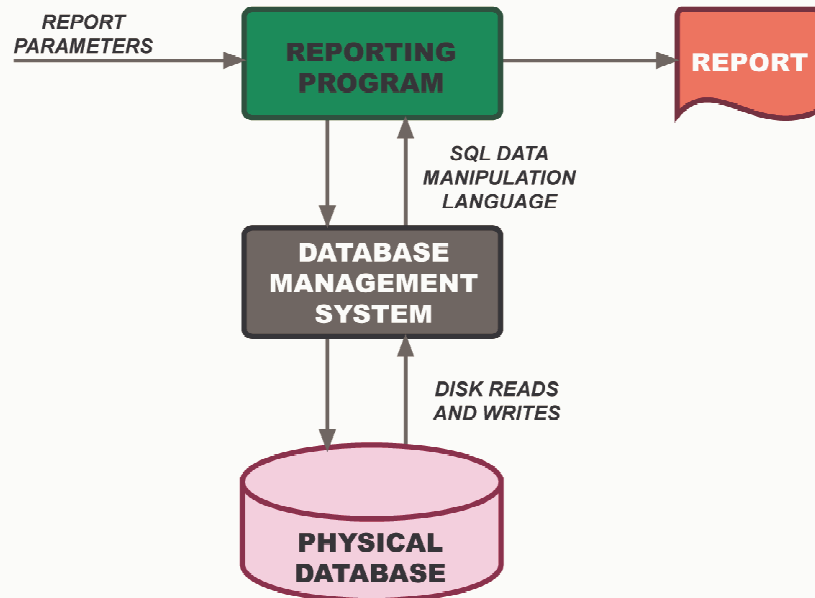
# E-Genting Programming Competition 2004

Pre-Competition Workshop, Week 1  
21 September 2004

## Workshop Outline

<b>Week</b>	<b>Material</b>
<b>1</b>	<b>Writing reporting programs using embedded SQL.</b>
<b>2</b>	<b>Estimating the execution time of an SQL query.</b>
<b>3</b>	<b>Designing systems (dataflow, pseudo-code, state transition).</b>
<b>4</b>	<b>Implementing processes (dataflow to working program).</b>
<b>5</b>	<b>Miscellaneous programming techniques</b>

## Embedded SQL Data Flows



## Interactive Query

**Entered query:**

```
select custName from custFile  
    where custId = 100127
```

**Displayed result:**

```
custName  
JOE BLOGS
```

## Embedded SQL Statement

```
exec sql begin declare section;
    char  nameVar[41];    // Customer name variable
    long  idVar;         // Customer id variable
exec sql end declare section;
// ...

idVar = 100127;
exec sql select  custName
                into      :nameVar
                from      custFile
                where     custId = :idVar;
if (SQLCODE != 0)
    { /* process row not found error */ }
cout << "Customer name is " << nameVar << '\n';
```

## JDBC Method Calls

```
Statement      stmt;        // JDBC statement
ResultSet      rs;         // JDBC result set
int            idVar;      // Customer id variable
String         nameVar;    // Customer name variable
//...

idVar = 100127;
rs = stmt.executeQuery (
    "select custName from custFile" +
    "where custId = " + idVar
);
if ( ! rs.next()) { /* process row not found error */ }
nameVar = rs.getString ("custName");
System.out.println ("Customer name is " + nameVar);
```

**Errors that an embedded SQL pre-processor can detect at compile time that raw JDBC calls cannot detect until run time:**

- **incorrectly spelled reserved words such as 'select' and 'where';**
- **syntax errors in SQL statements;**
- **misspelled column names;**
- **ambiguous column names (i.e. a reference to an unqualified column name that is a valid column name in two tables in the query);**
- **incompatible database column and host variable data types.**

## Cursor Select

```
exec sql declare custCur cursor for
        select  custId, custName
        from    custFile;
exec sql open custCur;
for (;;) {
    exec sql fetch  custCur
        into       :idVar, :nameVar;
    if (SQLCODE != 0) break;
    cout << idVar << " " << nameVar << '\n';
}
exec sql close custCur;
```

## Formalities

```
exec sql include sqlca;  
//...  
int main ()  
{  
    //...  
    exec sql connect to 'custDb';  
    //...  
}
```

**Alternatively, in the competition:**

```
// Insert statement to connect to the  
// database here.
```