Embedded SQL

What is SQL?

- Structured Query Language (SQL) is a standardized language used to manipulate database objects and the data they contain.
- It comprised of several different statements that are used manipulate data values.
- SQL being a nonprocedural is not a general-purpose programming language.

```
UPDATE EMPLOYEE SET LASTNAME = 'Jones' WHERE
EMPID = '001'
```

What is Embedded SQL?

- As a result, database applications are usually developed by combining capabilities of a high-level programming language with SQL.
- The simplest approach is to embed SQL statements directly into the source code file(s) that will be used to create an application. This technique is referred to as embedded SQL programming.
- sqlca.h – header file to be included.

Embedded SQL

- High-level programming language compilers cannot interpret, SQL statements.
- Hence source code files containing embedded SQL statements must be preprocessed before compiling.
- Thus each SQL statement coded in a high-level programming language source code file must be prefixed with the keywords **EXEC SQL** and terminated with either a semicolon or the keywords **END_EXEC**.

Embedded SQL

- Likewise, the Database Manager cannot work directly with high-level programming language variables.
- Instead, it must use special variables known as **host variables** to move data between an application and a database.
- Two types of Host variables:-
  1. Input Host Variables – Transfer data to database
  2. Output Host Variables – receives data from database

Embedded SQL

- Host variables are ordinary programming language variables.
- To be set apart, they must be defined within a special section known as a **declare section**.

```
EXEC SQL BEGIN DECLARE SECTION
char EmployeeID[7];
double Salary;
EXEC SQL END DECLARE SECTION
```

- Each host variable must be assigned a unique name even though declared in different declaration section.

Embedded SQL

```
main() {
 EXEC SQL BEGIN DECLARE SECTION;
 int OrderID, CustID;
 char SalesPerson[10], Status[6];
 EXEC SQL END DECLARE SECTION;

 printf ("Enter order number: ");
 scanf ("%d", &OrderID);

 EXEC SQL SELECT CustID, SalesPerson, Status FROM
Orders WHERE OrderID = :OrderID INTO :CustID,
:SalesPerson, :Status;

 printf ("Customer number: %d \n", CustID);
 printf ("Salesperson: %s \n", SalesPerson);
 printf ("Status: %s \n", Status);
}
```

Embedded SQL

Connecting to Database using embedded sql

```
EXEC SQL CONNECT :userid IDENTIFIED BY :passwd;
EXEC SQL CREATE TABLE Test (a int);
EXEC SQL INSERT INTO Test VALUES (1);
EXEC SQL SELECT MAX (a) INTO :value from R;
printf (“Max value=%d\n”,value);
```

Cursor

- Can declare a cursor on a query statement which generates a relation.
- Can open a cursor, repeatedly fetch a tuple, move the cursor, until all tuples have been retrieved.
- Control order: ORDER BY, in queries that are accessed through a cursor
- Can also modify/delete tuple pointed to by cursor.
- Must close cursor at end.

Cursor

```
EXEC SQL DECLARE myCursor CURSOR FOR SELECT bid
from Reservations;

EXEC SQL OPEN myCursor;
EXEC SQL WHENEVER NOT FOUND DO break;
while (1) {
     EXEC SQL FETCH myCursor INTO :num;
}

EXEC SQL CLOSE myCursor;
```