1.Write the syntax for Declaring Variables and Exceptions in Embedded SQL.

ANSWER 1: Here’s an example of how to declare variables and exceptions in embedded SQL:

EXEC SQL BEGIN DECLARE SECTION;

int var1;

char var2[20];

EXCEPTION exception\_name;

EXEC SQL END DECLARE SECTION;

In this example, we declare an integer variable “var1” and a character array “var2” with a size of 20. We also declare an exception called “exception\_name”.

2.Create a schema Dept (Deptid, name, courses)

(a)Write the Embedded SQL statements for inserting new rows in to the above schema

(b)Write a simple C program for demonstrating the same.

ANSWER 2: Here’s an example of how to insert new rows into the “Dept” schema using Embedded SQL:

EXEC SQL INSERT INTO Dept (Deptid, name, courses) VALUES (1, 'Computer Science', 'CSE101');

This statement inserts a new row into the “Dept” table with a department ID of 1, a name of “Computer Science”, and a course of “CSE101”.

Here’s an example of a simple C program that demonstrates how to insert new rows into the “Dept” schema using Embedded SQL:

#include <stdio.h>

#include <stdlib.h>

#include <sqlca.h>

EXEC SQL BEGIN DECLARE SECTION;

int deptid;

char name[20];

char courses[20];

EXEC SQL END DECLARE SECTION;

int main() {

EXEC SQL CONNECT TO database\_name USER username USING password;

if (sqlca.sqlcode != 0) {

printf("Error connecting to database.\n");

exit(1);

}

deptid = 1;

strcpy(name, "Computer Science");

strcpy(courses, "CSE101");

EXEC SQL INSERT INTO Dept (Deptid, name, courses) VALUES (:deptid, :name, :courses);

if (sqlca.sqlcode != 0) {

printf("Error inserting row.\n");

exit(1);

}

EXEC SQL COMMIT WORK;

if (sqlca.sqlcode != 0) {

printf("Error committing transaction.\n");

exit(1);

}

EXEC SQL DISCONNECT database\_name;

return 0;

}

This program connects to the database using the specified username and password. It then inserts a new row into the “Dept” table with the specified values. If an error occurs during the insert or commit operation, the program prints an error message and exits.

3.Differentiate Static and Dynamic SQL.

ANSWER 3: Static SQL is SQL statements that are hard-coded into an application at compile time. The SQL statements are fixed and cannot be changed at runtime. Static SQL is typically used in applications where the SQL statements are known in advance and do not change frequently.

Dynamic SQL is SQL statements that are constructed at runtime. The SQL statements can be changed at runtime based on user input or other factors. Dynamic SQL is typically used in applications where the SQL statements are not known in advance or change frequently.

4.Write the syntax for Declaring the cursor.

ANSWER 4: Here’s an example of how to declare a cursor in SQL:

DECLARE cursor\_name CURSOR FOR SELECT column1, column2 FROM table\_name WHERE condition;

In this example, we declare a cursor called “cursor\_name” that selects the “column1” and “column2” columns from the “table\_name” table where the specified condition is true.