	University: Assiut **Faculty: FCI** **Dept.: IS**	**ADB** **Midterm Exam 2** **IS411**	**15/12/2016** **Level 4** **Time: 90 Min.**	

1- The following query selects those continents that have countries in which more than 50% of the population speak English. Is this an example of using a correlated subquery? Why or why not?

```
SELECT DISTINCT Continent FROM Country
WHERE Code IN (SELECT CountryCode FROM CountryLanguage
               WHERE Language='English' AND Percentage>50);
```

The example shown is *not* an example of a correlated subquery because the subquery can be resolved completely without regard to the outer query. In a correlated subquery, the inner `SELECT` is dependent on the outer query.

2- What is the effect of executing the following query?

```
SELECT Continent, Name FROM Country c1
WHERE Population >= ALL (SELECT Population FROM Country c2
                         WHERE c1.Continent=c2.Continent);
```

The query returns, for each continent, the country whose population is greater than or equal to the population of every country on the same continent. In other words, it returns the country with the greatest population on each continent.

3- Why is the following use of a subquery in the FROM clause not correct?

```
SELECT Name, Language FROM Country AS c, (SELECT Language FROM CountryLanguage
                                          WHERE CountryCode = c.Code) AS tmp
```

The server returns `ERROR 1109 (42S02): Unknown table 'c' in where clause` if you try to execute this query. Subqueries in the `FROM` clause of a query cannot be correlated with the outer query.

4- Ahmed wants to create a table of country capitals. He creates the Capitals table by copying the structure and data of the City table, but then by mistake copies all of the data from the City table into the Capitals table:

```
mysql> CREATE TABLE Capitals LIKE City;
Query OK, 0 rows affected (0.01 sec)
mysql> INSERT INTO Capitals SELECT * FROM City;
Query OK, 4079 rows affected (0.08 sec)
Records: 4079 Duplicates: 0 Warnings: 0
```

The city ID of a country's capital is stored in the Capital field of the Country table. Using a subquery, how can Joe remove all the non-capital cities from the Capitals table?

The subquery in the following statement searches the `City` table to identify the city IDs of all capital
cities. The `IS NOT NULL` clause is needed because a few countries don't have a capital. The outer
statement deletes all rows in the table `Capitals` that are not found by the subquery:

DELETE FROM Capitals WHERE ID NOT IN (SELECT Capital FROM Country WHERE Capital IS NOT NULL);

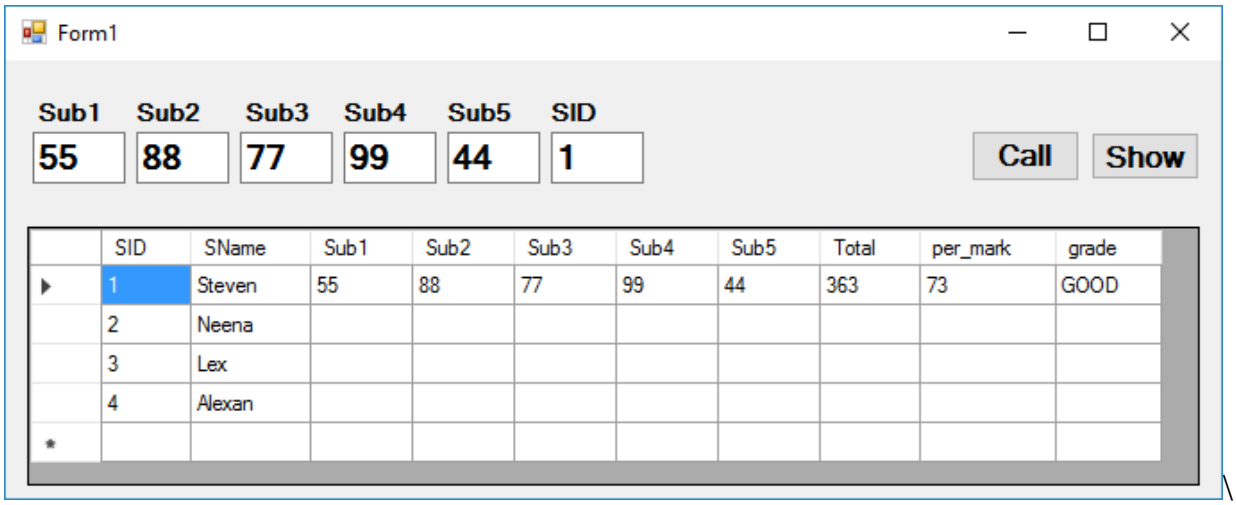
5- We have a table student_marks . There are data only in STUDENT ID and NAME columns.

SID	NAME	SUB1	SUB2	SUB3	SUB4	SUB5	TOTAL	PER_MARKS	GRADE
1	Steven	0	0	0	0	0	0	0.00	
2	Neena	0	0	0	0	0	0	0.00	
3	Lex	0	0	0	0	0	0	0.00	
4	Alexan	0	0	0	0	0	0	0.00	

Write sql code when you received all subject marks, then total marks of all subject, the percentage of total marks and grade will be automatically calculated.
Hint: Grade (will be stored GRADE column) :
- If PER_MARKS>=90 -> 'EXCELLENT'
- If PER_MARKS>=75 AND PER_MARKS<90 -> 'VERY GOOD'
- If PER_MARKS>=60 AND PER_MARKS<75 -> 'GOOD'
- If PER_MARKS>=40 AND PER_MARKS<60 -> 'AVERAGE'
- If PER_MARKS<40-> 'NOT PROMOTED'

```
delimiter //
CREATE TRIGGER student_marks_update
BEFORE UPDATE ON student_marks
FOR EACH ROW
BEGIN
SET NEW.TOTAL = NEW.SUB1 + NEW.SUB2 + NEW.SUB3 + NEW.SUB4 + NEW.SUB5;
SET NEW.per_mark = NEW.TOTAL/5;
IF NEW.per_mark >=90 THEN SET NEW.GRADE = 'EXCELLENT';
ELSEIF NEW.per_mark>=75 AND NEW.per_mark<90 THEN SET NEW.GRADE = 'VERY
GOOD';
ELSEIF NEW.PER_MARK>=60 AND NEW.PER_MARK<75 THEN SET NEW.GRADE =
'GOOD';
ELSEIF NEW.per_mark>=40 AND NEW.per_mark<60 THEN SET NEW.GRADE =
'AVERAGE';
ELSE SET NEW.GRADE = 'NOT PROMOTED';
END IF;
END;//
delimiter ;
```

6- Write a stored procedure that take 5 subject score and student id then update student_marks table. Write a c# code that implement the following:
Call button: call a procedure and update student data.
Show button: show student data.




```
Create procedure student_update(id int,s1 int,s2 int,s3 int,s4 int,s5 int)
UPDATE STUDENT_MARKS SET SUB1 = s1, SUB2 = s2, SUB3 = s3, SUB4 = s4, SUB5 =
s5 WHERE SID = id;

private void Call_Click(object sender, EventArgs e)
        {
            MySqlConnection con = new
MySqlConnection("server=localhost;database=world;uid=root;pwd=root");
            con.Open();
            MySqlCommand cmd = new MySqlCommand("student_update", con);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@s1", textBox1.Text);
            cmd.Parameters.AddWithValue("@s2", textBox2.Text);
            cmd.Parameters.AddWithValue("@s3", textBox3.Text);
            cmd.Parameters.AddWithValue("@s4", textBox4.Text);
            cmd.Parameters.AddWithValue("@s5", textBox5.Text);
            cmd.Parameters.AddWithValue("@id", textBox6.Text);
            cmd.ExecuteNonQuery();
            con.Close();
        }
        private void Show_Click(object sender, EventArgs e)
        {
            MySqlDataAdapter da = new MySqlDataAdapter("select * from
student_marks;", "server=localhost;database=world;uid=root;pwd=root");
            DataTable dt = new DataTable();
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }
```