Cairo University
Faculty of Engineering
Computer Engineering Dept.
CMP N202

Fall 2015

LAB #5
Accessing DB from C# Application - part2
More GUI controls

Objectives

After this lab, the student should be able to:

- Create and fill database using SQL statements
- Connect C# application to a database
- Execute SQL queries on the database by calling direct SQL statements in C# application.

Database Creation using SQL statements

Create and Fill the Database

Script **CompanyDB.sql** does the following:

1- Creates a new DB called CompanyLab2 ➔ ***Create Database*** statement
2- Creates DB tables: Employee, Department, Dept_Locations, Project, and Works_On tables ➔ ***Create Table*** statement
3- Creates primary and foreign keys for each table ➔ ***Create/Alter Table*** statement

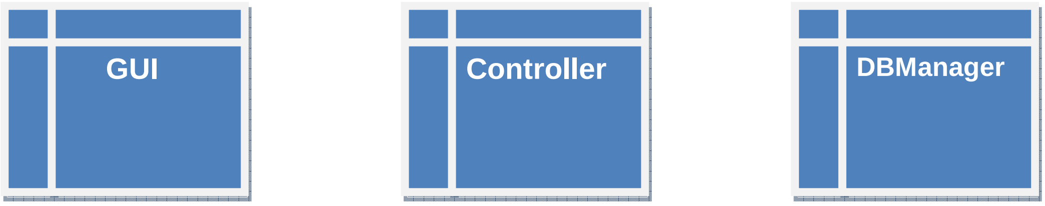
�Why do we use Alter table to create Dno foreign key in table Employee?

4- Fills database tables with some sample data ➔ ***Insert Into*** statement
5- Completes tables filling ➔ ***Update*** statement

�What are those nulls inserted in table Employee at first?

Windows Application Example

The application you are given consists of the following three modules:

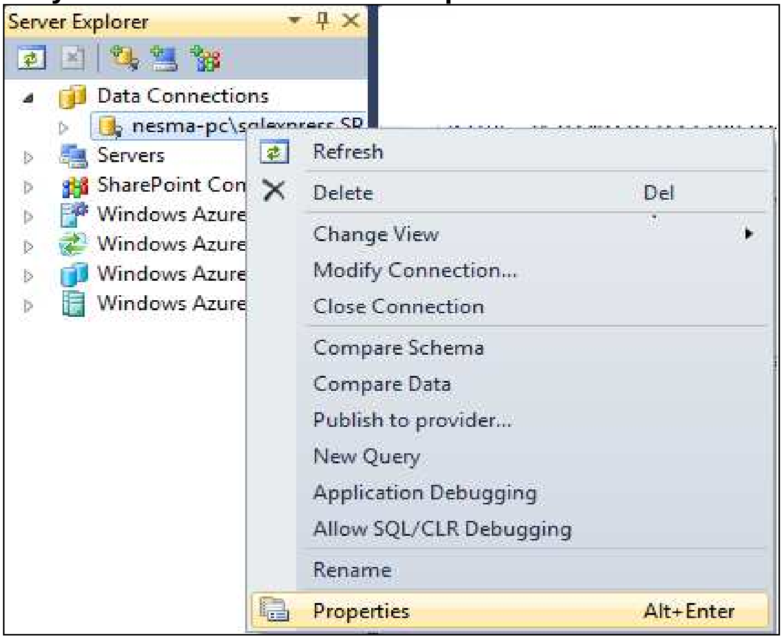


- ❑ User deals with the GUI to specify which functionality he wants to use.
- ❑ Controller class is responsible for controlling the flow of the program, therefore, the GUI deals with the controller class in order to specify what action to be done when the user for example clicks on a certain button.
- ❑ DBManager class is responsible for dealing with the database. Controller deals with this class when it finds any action involving DB.
- ❑ **<u>Note</u>**: Controller and DBManager and user-defined classes not library classes.

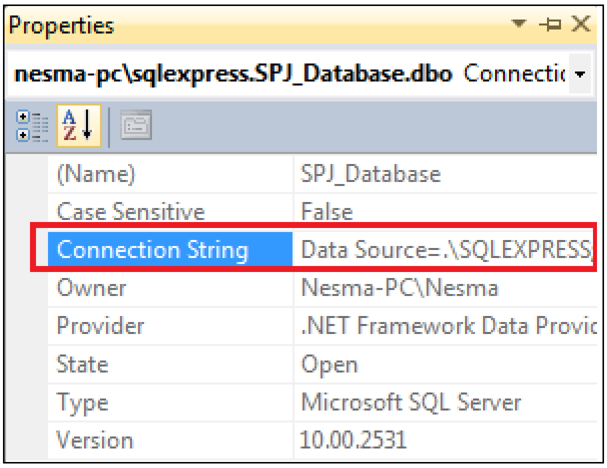
Connecting to the database (revision)

Steps

1. **In Server Explorer/Database Explorer, right click on the database connection you created and choose Properties**



2. **In the Properties list you will find a property called "Connection String", copy its value to use it in the c# code**



nesma-pc\sqlexpress.SPJ_Database.dbo Connecti

(Name)	SPJ_Database
Case Sensitive	False
Connection String	Data Source=.\SQLEXPRESS
Owner	Nesma-PC\Nesma
Provider	.NET Framework Data Provic
State	Open
Type	Microsoft SQL Server
Version	10.00.2531

3. **Use the following code to connect to the database**

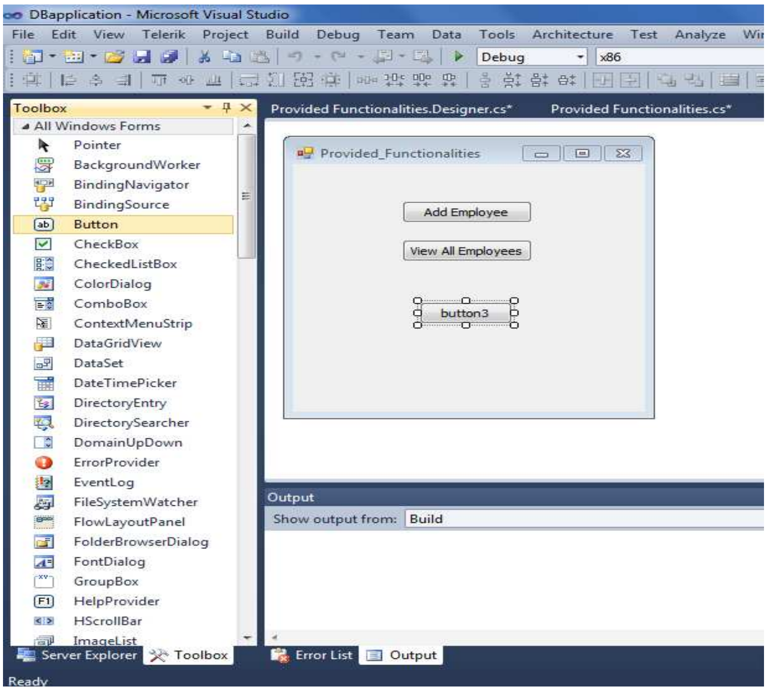
```
SqlConnection myConnection;
myConnection = new SqlConnection(DB_Connection_String);
try
{
     myConnection.Open();
     Console.WriteLine("The DB connection is opened successfully");
}
catch (Exception e)
{
     Console.WriteLine("The DB connection is failed");
}
```

Executing complex queries through the application

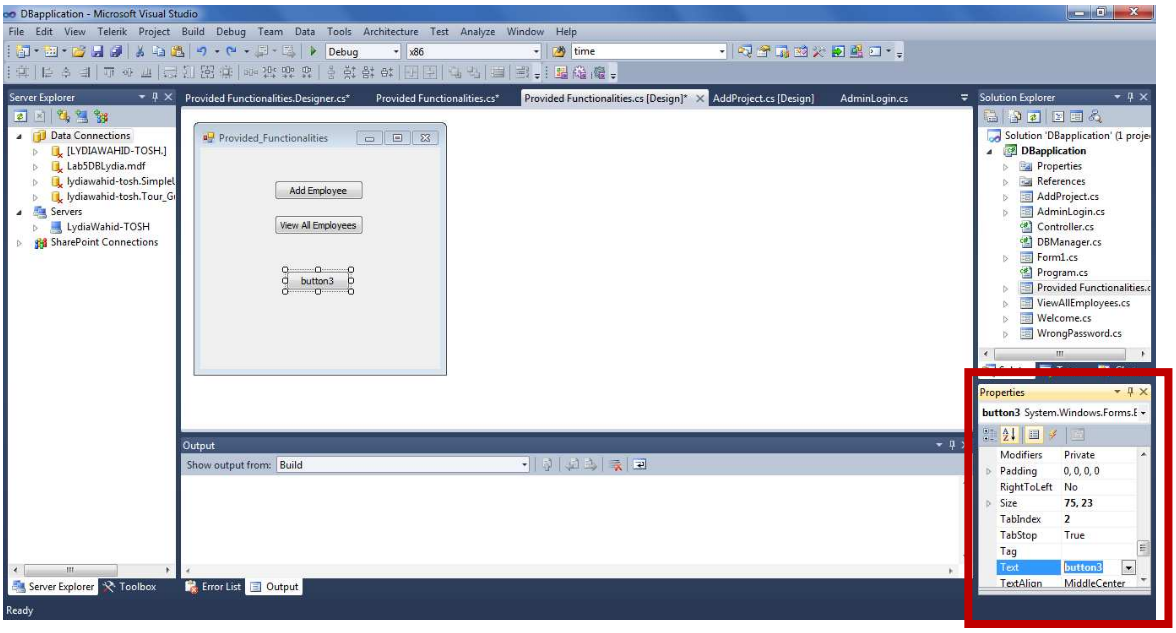
Example: If we want to add the following functionality to our application:
“Select project name and department name for all projects that are in departments located in a certain location” (location will be specified by the user):

Steps:

1. Add a button for this functionality in the “Provided_Functionalities” form

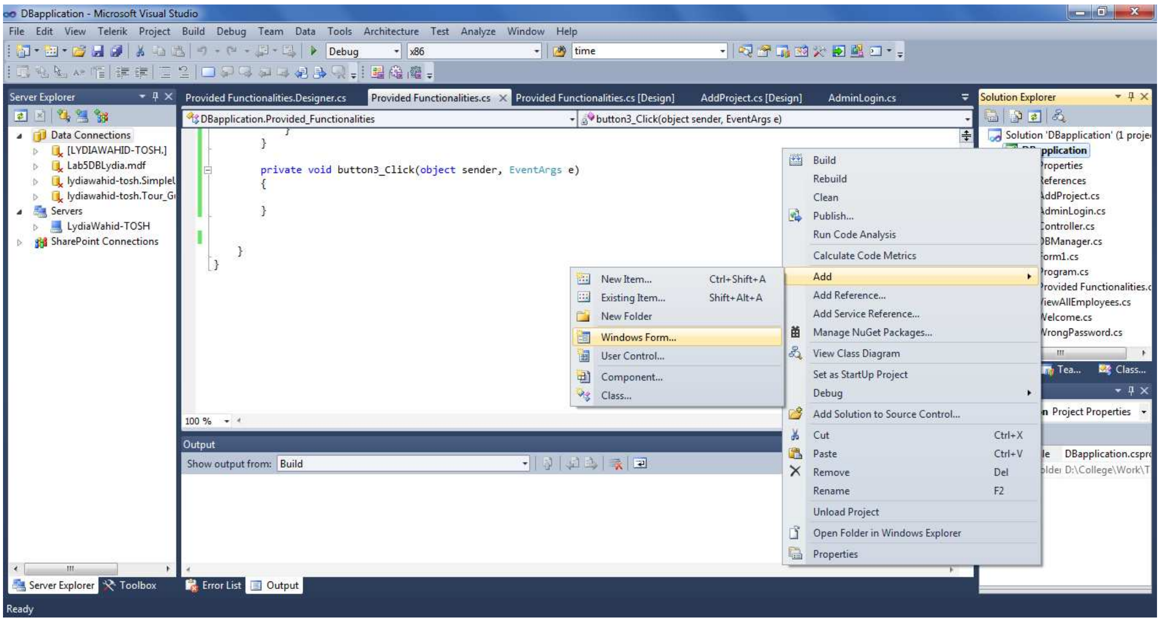


2. Rename the button by selecting it and then changing the “Text” in the properties window that is found at the bottom right. Notice that this changes only the text that is displayed on the button, not its name.



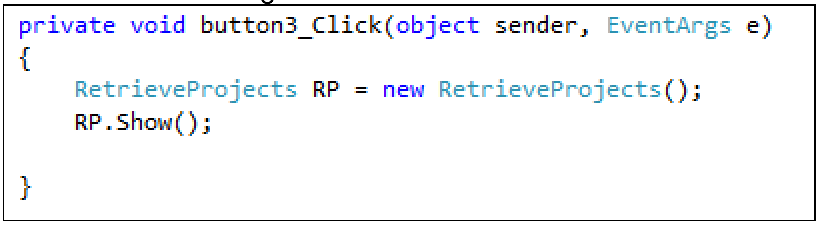
Modifiers	Private
Padding	0, 0, 0, 0
RightToLeft	No
Size	75, 23
TabIndex	2
TabStop	True
Tag	
Text	button3
TextAlign	MiddleCenter

3. Add new windows form and name it “RetrieveProjects”.



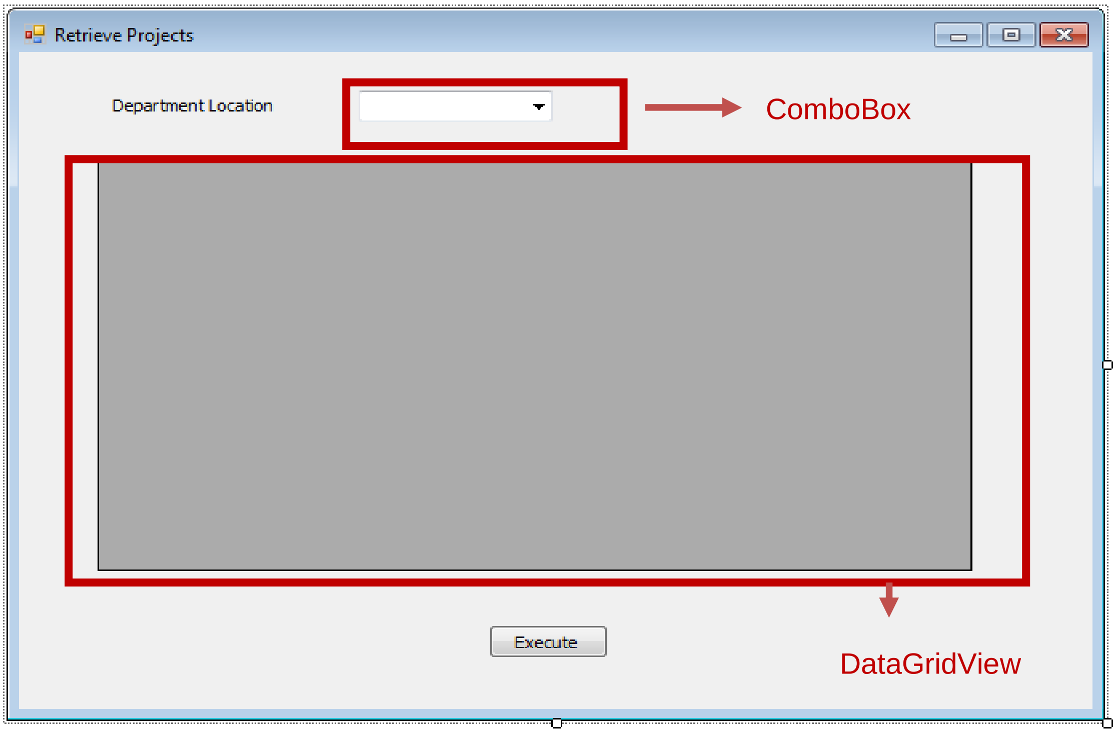
```
        }

        private void button3_Click(object sender, EventArgs e)
        {

        }

    }
}
```

4. Double click the button to write the code that will be executed when the button click event happens. Write the following code in order to show the new window added.

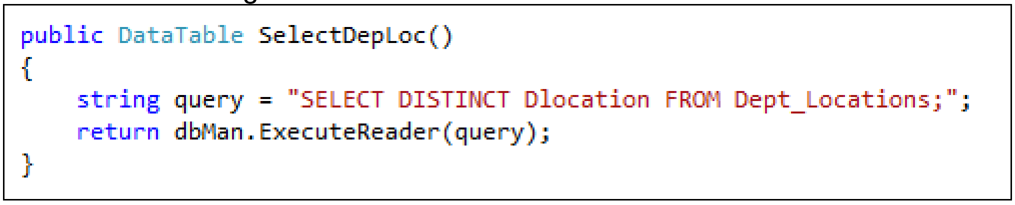


```
private void button3_Click(object sender, EventArgs e)
{
    RetrieveProjects RP = new RetrieveProjects();
    RP.Show();

}
```

5. In the “RetrieveProjects” windows form add a “combobox” and “dataGridView” from the toolbox. ComboBox will be used to select from it the department location that we want; it will get its data from the “Dlocation” colomn found in “Dept_locations” table. The “dataGridView” will be used to view the required information that will be retrieved from the database.

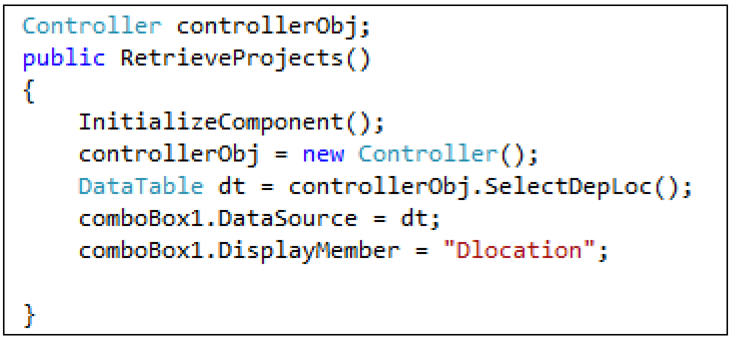


6. In order to let the comboBox get its data from the the “Dlocation” column found in “Dept_locations” table, we will perform the following steps:
   - ❑ Define the following member function in the controller class



```
public DataTable SelectDepLoc()
{
    string query = "SELECT DISTINCT Dlocation FROM Dept_Locations;";
    return dbMan.ExecuteReader(query);
}
```

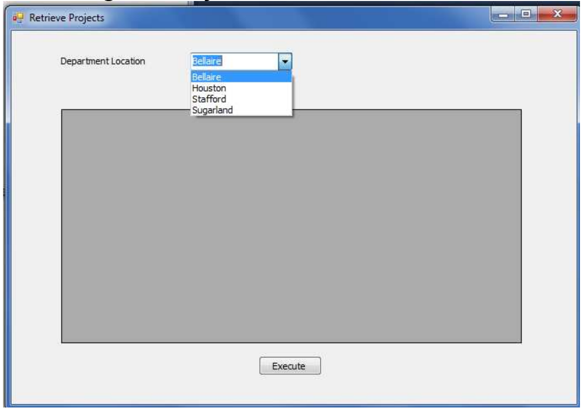
   Notice the use of the word “DISTINCT” in the query to remove duplicate tuples.
   - ❑ Then in the “RetrieveProjects” form code, add a “Controller” object and specify the datasource of the comboBox as shown below in the form constrcutor:



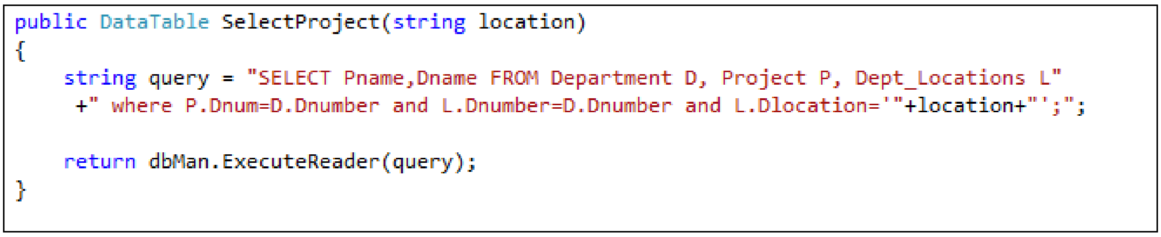
```
Controller controllerObj;
public RetrieveProjects()
{
    InitializeComponent();
    controllerObj = new Controller();
    DataTable dt = controllerObj.SelectDepLoc();
    comboBox1.DataSource = dt;
    comboBox1.DisplayMember = "Dlocation";

}
```

   Where the “comboBox1.DisplayMember” is the column in the database that the comboBox will get its data from.

   - ❑ When you run the project and go to the “RetrieveProjects” form, you will see the following when you click the arrow of the combo box.

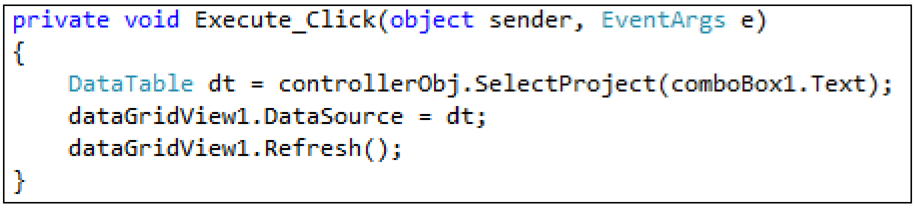


7. In order to execute the query and see the results, do the following:
   - ❑ Add a dataGridView to display the resulting table
   - ❑ Add the following function in the “Controller” class in order to form the query that we want to execute:



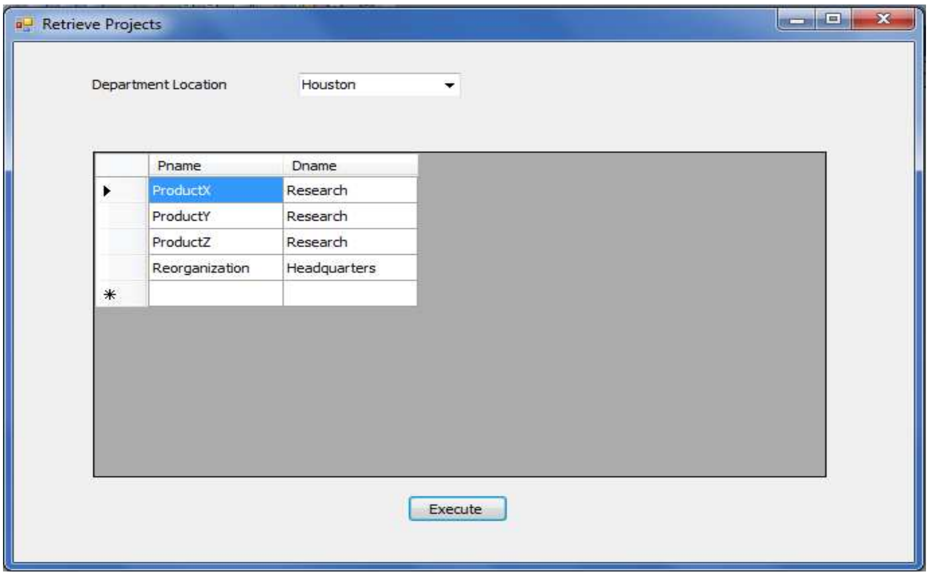
```
public DataTable SelectProject(string location)
{
    string query = "SELECT Pname,Dname FROM Department D, Project P, Dept_Locations L"
     +" where P.Dnum=D.Dnumber and L.Dnumber=D.Dnumber and L.Dlocation='"+location+"';";

    return dbMan.ExecuteReader(query);
}
```

   - ❑ Then in the “RetrieveProjects” form, in the event handler of “Execute” button click, will write the following code:



```
private void Execute_Click(object sender, EventArgs e)
{
    DataTable dt = controllerObj.SelectProject(comboBox1.Text);
    dataGridView1.DataSource = dt;
    dataGridView1.Refresh();
}
```

❑ When you run the project and choose “Houston” from the comboBox and click “Execute” button, the dataGridView will contain the result of the query as shown:



Department Location: Houston

Pname	Dname
ProductX	Research
ProductY	Research
ProductZ	Research
Reorganization	Headquarters

Requirements (To be delivered next lab)

Extend the given code and add these functionalities to your application. (10 marks)
(You may add more than one functionality in one form).

1- Get SSN and address for all employees with salary less than 40000. (1 mark)
2- Get all female employees who work in "Administration" department. (1 mark)
3- Get departments names for all departments located at "Houston" (1 mark)
4- Get employees names and salaries for all employees who work in a project located at "Stafford" or in "Houston" and work less than 35 hours. (1 marks)
5- Insert a new department. (1 mark)
6- Allow user to update salary of an employee of a certain SSN. (1 mark)
7- Get the last names of department managers and their salaries. (1 mark)
8- Get Name and SSN for all employees working in "Research" department or working on projects controlled by "Research" department. (2 marks)
9- Get maximum, minimum and average salary for employees (1 mark)