

SOFTWARE QUALITY ENGINEERING

Submitted to: Ma'am Irum Matloob

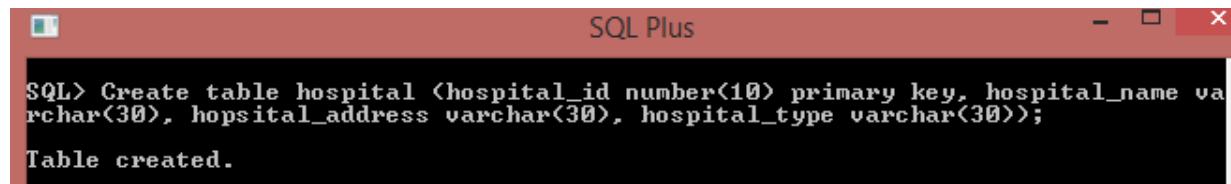
Submitted by:
Aqsa Tabasum (002)
Arshiya Saleem (004)
Shamsa Kanwal (028)

**Title: Hospital Management System
(DESIGN+CODING)**

CREATE QUERIES:

First we will create the table, in which the record will be stored. We have created seven different tables which are hospital, department, ward, room, staff, patient and doctor.

HOSPITAL TABLE:



```
SQL> Create table hospital (hospital_id number(10) primary key, hospital_name varchar(30), hospital_address varchar(30), hospital_type varchar(30));  
Table created.
```

This table has hospital_id which has the data type number which means that integer value will be stored in hospital id, Moreover it has been declared as a primary key which will uniquely identify the record. Hospital_name has the data type varchar which will store the variables. Similarly hospital_address and hospital_type has varchar data type and the query create will create the table hospital.

DEPARTMENT TABLE:



```
SQL> Create table Department (department_id number(10) primary key, department_name varchar(30), department_type varchar(30));  
Table created.
```

The table department has department_id of integer data type and is a primary key, department_name and department_type has varchar datatype. This table is used to specify the different departments of hospital.

ROOM TABLE:



```
SQL> Create table Room (room_id number(10) primary key, no_of_patients number(30), patient_name varchar(30), room_floor varchar(30));  
Table created.
```

The table room has room_id of integer data type and is a primary key, no of patients has also integer data type patient_name and room_floor has varchar datatype. This table is used to specify the number of rooms that are assigned to different patients.

WARD TABLE:

```
SQL> Create table Ward (Ward_id number(10) primary key, no_of_patients number(30), patients_name varchar(30) , room_floor varchar(30));  
Table created.
```

The table ward has ward_id of integer data type and is a primary key, no of patients has also integer data type patient_name and room_floor has varchar datatype. This table is used to specify the number of wards that are assigned to different patients.

STAFF TABLE:

```
SQL> Create table Staff (staff_id number(10) primary key, staff_name varchar(30) , staff_job varchar(30),department_id number(10));  
Table created.
```

DOCTOR TABLE:

```
SQL> Create table Doctor (doctor_id number(10) primary key, doctor_name varchar(30) , specialization varchar(30),doctor_address varchar(30),department_id number(10));  
Table created.
```

PATIENT TABLE:

```
SQL> Create table Patient (patient_id number(10) primary key, patient_name varchar(30) , gender varchar(30),patient_address varchar(30));  
Table created.  
SQL>
```

INSERT AND DISPLAY AND DELETE AND ALTER QUERIES:

INSERT AND DISPLAY QUERYFOR HOSPITAL:

```
SQL> Insert into Hospital values (1, 'AQSA_hospital' , 'Islamabad', 'Private');  
1 row created.  
SQL> Insert into Hospital values (2, 'Shamsa_hospital' , 'Rawalpindi', 'Government');  
1 row created.
```

This query is used to insert the data into the hospital table. Single inverted commas are used to store the value for varchar data type. So, we have created two data records for AQSA and SHAMSA hospital.

```

SQL> select * from Hospital;
HOSPITAL_ID HOSPITAL_NAME          HOSPITAL_ADDRESS
HOSPITAL_TYPE
-----  -----
Private      1 AQSA_hospital        Islamabad
Government   2 Shamsa_hospital     Rawalpindi
SQL> 

```

Now this select query is used to display the record that is stored, as we have stored two records so these records are displayed on the screen by executing the above query.

INSERT, DELETE AND DISPLAY QUERY FOR DEPARTMENT:

```

SQL> Insert into Department values (1 , 'Psychology_AQSA' , 'MICU');
1 row created.

SQL> Insert into Department values ( 2 , 'Cardiology' , 'ICU');
1 row created.

SQL> delete from Department where department_id= 2;
1 row deleted.

SQL> Insert into Department values ( 2 , 'Cardiology_shamsa' , 'ICU');
1 row created.

SQL> select * from Department;
DEPARTMENT_ID DEPARTMENT_NAME          DEPARTMENT_TYPE
-----  -----
1 Psychology_AQSA                    MICU
2 Cardiology_shamsa                  ICU
SQL> 

```

Now in this table first we have stored the two records of data in the department table. Then we have run the delete query to delete the second row in the department table, now again the insert query is used to store the data and finally the select query is used to display the record that is stored in department table.

INSERT AND DISPLAY QUERY FOR ROOM:

```

SQL> Insert into Room values (1 ,1, 'AQSA','First');

1 row created.

SQL> Insert into Room values (2 ,1, 'Shamsa','ground');

1 row created.

SQL> select * from Room;
   ROOM_ID NO_OF_PATIENTS PATIENT_NAME
   ROOM_FLOOR
   First          1           AQSA
   ground         2           Shamsa

```

We have used the insert query here to insert the two records of data in the room table and then the select query to display the record.

INSERT, ALTER RENAME AND DISPLAY QUERY FOR WARD:

```

SQL> alter table Ward rename column ROOM_FLOOR to Ward_floor;
Table altered.

SQL> Insert into Ward values (3 ,2, 'AQSA_TABASSUM','ground');

1 row created.

```

We have used the alter rename query to rename the tuple room_floor to ward_floor, then we have used the insert query to insert the record into the table.

```

SQL> select * from Ward;
   WARD_ID NO_OF_PATIENTS PATIENTS_NAME
   WARD_FLOOR
   ground          3           2 AQSA_TABASSUM

```

The select query to display the record of ward.

INSERT AND DISPLAY QUERY FOR STAFF:

```

SQL> Insert into Staff values (1 , 'AQSA' , 'Technician' , 1);
1 row created.

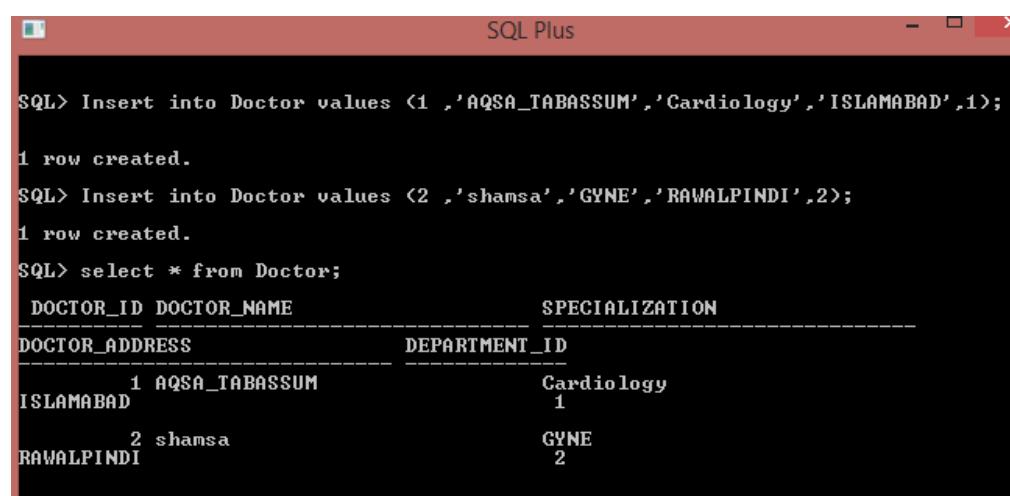
SQL> Insert into Staff values (2 , 'shamsa' , 'physician' , 2);
1 row created.

SQL> select * from Staff;
      STAFF_ID STAFF_NAME                      STAFF_JOB
DEPARTMENT_ID
      1 AQSA                           Technician,
      1
      2 shamsa                         physician,
      2

```

We have used the insert query here to insert the two records of data in the staff table and then the select query to display the record.

INSERT AND DISPLAY QUERY FOR DOCTOR:



```

SQL> Insert into Doctor values (1 , 'AQSA_TABASSUM' , 'Cardiology' , 'ISLAMABAD' , 1);
1 row created.

SQL> Insert into Doctor values (2 , 'shamsa' , 'GYNE' , 'RAWALPINDI' , 2);
1 row created.

SQL> select * from Doctor;
      DOCTOR_ID DOCTOR_NAME                      SPECIALIZATION
DOCTOR_ADDRESS          DEPARTMENT_ID
      1 AQSA_TABASSUM                           Cardiology
ISLAMABAD                  1
      2 shamsa                             GYNE
RAWALPINDI                  2

```

We have used the insert query here to insert the two records of data in the doctor table and then the select query to display the record.

INSERT AND DISPLAY QUERY FOR PATIENT:

SAVEPOINT AND COMMIT:

```

SQL> Insert into Patient values (1 , 'AQSA' , 'Female' , 'RAWALPINDI');
1 row created.

SQL> select * from Patient;
PATIENT_ID PATIENT_NAME          GENDER
PATIENT_ADDRESS
-----      -----
1 AQSA           Female
RAWALPINDI

SQL> savepoint insert_done;
Savepoint created.

SQL> commit;
Commit complete.

SQL>

```

We have used the insert query here to insert the two records of data in the room table and then the select query to display the record.

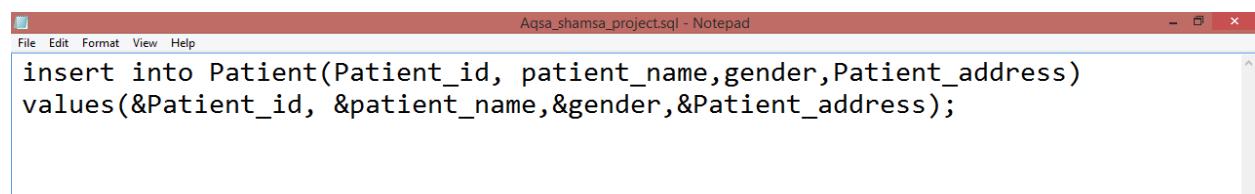
Then used the save insert_done which will discard the most recent delete operation without discarding the earlier insert statement

Then we have used the commit statement to permanently save the record.

SCRIPT QUERY:

This query will be used to enter the data from the user. I have used the script query for patients:

Script:



A screenshot of a Windows Notepad window titled "Aqsa_shamsa_project.sql - Notepad". The window contains the following SQL script:

```

insert into Patient(Patient_id, patient_name,gender,Patient_address)
values(&Patient_id, &patient_name,&gender,&Patient_address);

```

Output:

```

SQL> @d:\AQSA_shamsa_project.sql;
Enter value for patient_id: 02
Enter value for patient_name: 'AQSA TABASSUM'
Enter value for gender: 'FEMALE'
Enter value for patient_address: 'ISLAMABAD'
old   2: values(&Patient_id, &patient_name,&gender,&Patient_address)
new   2: values(02, 'AQSA TABASSUM','FEMALE','ISLAMABAD')

1 row created.

SQL> @d:\AQSA_shamsa_project.sql;
Enter value for patient_id: 08
Enter value for patient_name: 'SHAMSA'
Enter value for gender: 'FEMALE'
Enter value for patient_address: 'SIALKOT'
old   2: values(&Patient_id, &patient_name,&gender,&Patient_address)
new   2: values(08, 'SHAMSA','FEMALE','SIALKOT')

1 row created.

SQL> @d:\AQSA_shamsa_project.sql;
Enter value for patient_id: 03
Enter value for patient_name: 'AMNA'
Enter value for gender: 'FEMALE'
Enter value for patient_address: 'MURREE'
old   2: values(&Patient_id, &patient_name,&gender,&Patient_address)
new   2: values(03, 'AMNA','FEMALE','MURREE')

1 row created.

```

Display patient table:



The screenshot shows the SQL Plus interface with a red header bar containing the title "SQL Plus". The main area displays the results of a SQL query:

```

SQL> select *from patient;
PATIENT_ID PATIENT_NAME          GENDER
----- -----
PATIENT_ADDRESS
      2 AQSA TABASSUM           FEMALE
ISLAMABAD
      8 SHAMSA                 FEMALE
SIALKOT
      3 AMNA                  FEMALE
MURREE

PATIENT_ID PATIENT_NAME          GENDER
----- -----
PATIENT_ADDRESS
      1 AQSA                  Female
RAWALPINDI

SQL>

```

UPDATE QUERY:

```
SQL> update Doctor set specialization='Nurology' where doctor_id=1;
1 row updated.
```

```
SQL> select * from Doctor;
```

DOCTOR_ID	DOCTOR_NAME	SPECIALIZATION
1	AQSA_TABASSUM ISLAMABAD	Nurology 1
2	shamsa RAWALPINDI	GYN 2

```
SQL>
```

This query will update the record by changing the specialization column and replacing the cardiology with neurology where id =1. In this record Aqsa tabassum has the id 1 so its record has updated.

DELETE QUERY:

```
SQL> delete from staff where staff_name='shamsa';
1 row deleted.
```

```
SQL> select * from Staff;
```

STAFF_ID	STAFF_NAME	STAFF_JOB
1	AQSA 1	Technician,

```
SQL>
```

Now this query will delete the whole record where staff name= shamsa.

SHOW CREATED TABLES:

```
SQL> select table_name from user_tables;
TABLE_NAME
-----
DEPT
EMP
BONUS
SALGRADE
DEPARTMENT_DETAIL
EMPLOYEE_DETAIL
MY_EMPLOYEE
MY_EMPLOYEE_AQSA
HOSPITAL
DEPARTMENT
ROOM

TABLE_NAME
-----
WARD
STAFF
DOCTOR
PATIENT
15 rows selected.

SQL>
```

This query will display all tables that are created by the user.

RENAME TABLE:

```
SQL> rename Room to Hospital_ROOMS;
Table renamed.

SQL> desc Hospital_ROOMS;
Name                                Null?    Type
-----                            -----
ROOM_ID                           NOT NULL NUMBER<10>
NO_OF_PATIENTS                   NUMBER<30>
PATIENT_NAME                      VARCHAR2<30>
ROOM_FLOOR                         VARCHAR2<30>

SQL> _
```

This query will rename the table room with the name Hospital_rooms. And then the desc statement is used to display the contents of Hospital_rooms table.

COMMENTS:

```
SQL> comment on table Hospital is 'THIS IS AQSA TABASSUM 02 AND SHAMSA KANWAL ';

Comment created.

SQL> select comments from user_tab_comments;

COMMENTS
-----
THIS IS AQSA TABASSUM 02 AND SHAMSA KANWAL
AQSA TABASSUM 02

COMMENTS
-----
AQSA TABASSUM 02

15 rows selected.
```

This comment query is used to add the comments to the specific table and to show that comments.

DROP THE NO OF PATIENTS COLUMN FROM THE WARD TABLE.

```
SQL> alter table Ward drop column no_of_patients;

Table altered.

SQL> desc ward;
 Name          Null?    Type
 ----          -----   -----
 WARD_ID      NOT NULL NUMBER(10)
 PATIENTS_NAME VARCHAR2(30)
 WARD_FLOOR   VARCHAR2(30)
```

Now to delete a column from the table we use the above query that is the drop column query. Here it will delete the column name no of patient from the Patients table.

CONSTRAINT OF FOREIGN KEY:

```

SQL> alter table Ward add <Dept_id number<10>,constraint Aqsa_Ward_Dept_id_fk f
oreign key<Dept_id> references Department<Department_id>>;
Table altered.

SQL> desc Ward;
Name                                         Null?    Type
-----                                     -----
WARD_ID                                     NOT NULL NUMBER<10>
PATIENTS_NAME                               VARCHAR2<30>
WARD_FLOOR                                   VARCHAR2<30>
DEPT_ID                                     NUMBER<10>

```

This query will create the department_id of Debarment table as the foreign key for Ward table.

```

SQL> select constraint_name,constraint_type from user_constraints;
CONSTRAINT_NAME          C
COMM_EMP_MIN              C
SYS_C0011630               C
SYS_C0011631               C
AQSA_WARD_DEPT_ID_FK      R
MY_EMP_DEPT_ID_FK         R
FK_DEPTNO                 R
PK_DEPT                   P
PK_EMP                    P
MY_EMP_ID                  P
MY_DEPTID_PK                P
BIN$YhZWpUWMSjmTGDTxLOZrCA==$0 P

CONSTRAINT_NAME          C
SYS_C0011746               P
SYS_C0011747               P
BIN$4kGI0rnZQZK8+GAI9e+Ulkw==$0 P
SYS_C0011749               P
SYS_C0011750               P
SYS_C0011751               P
SYS_C0011752               P
SYS_C0011753               P

19 rows selected.

```

Activate Window
Go to PC settings

This query will show all the constraints that were applied to different tables.

DISPLAY DOCTOR NAME, SPECIALIZATION AND DEPARTMENT_NAME FROM THE DATABASE

```
SQL> select department_name, doctor_name,specialization from department, doctor
where doctor.department_id= department.department_id;
```

DEPARTMENT_NAME	DOCTOR_NAME
SPECIALIZATION	
Psychology_AQSA	AQSA_TABASSUM
Nurology	
Cardiology_shamsa	shamsa
GYNE	

Activate Window

This query will display the department name, doctor name and specialization from the doctor and department table at the same place.

DISPLAY THE NAMES OF DOCTOR WHO BELONG TO DEPARTMENT =1 AND SPECIALIZATION ="NURIOLOGY":

```
SQL> select doctor_name, Doctor_address, doctor_id from doctor where department_id =1 and specialization= 'Nurology';
```

DOCTOR_NAME	DOCTOR_ADDRESS	DOCTOR_ID
AQSA_TABASSUM	ISLAMABAD	1

As AQSA TABASSUM is the name of the doctor where department id =1 and specialization =neurology, so it will be displayed.

DISPLAY THE WARD DETAILS WHERE DEPT_ID =2

```
SQL> Insert into Ward values (1 , 'AQSA','ground',2);
1 row created.

SQL> select ward_id, patients_name, ward_floor from ward where dept_id=2;
```

WARD_ID	PATIENTS_NAME	WARD_FLOOR
1	AQSA	ground

DISCARD THE MOST RECENT DELETE OPERATION WITHOUT DISCARDING THE EARLIER INSERT STATEMENT

```
SQL> Rollback to savepoint Insert_done;
Rollback complete.
```

VISUAL STUDIO CODE:

```

// aqsa_code.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;
class Patient
{
private:
    struct patient_node
    {
        int id;
    char name[30];
    int age;
    char gender[30];
    char phone_no[30];
    char disease[30];
    char medicine[30];
    char address[30];
    int room_no;
    int ward_no;
    patient_node *next;
    patient_node *prev;
    };
public:
    struct patient_node *head, *tail;

Patient()
{head=NULL;
tail=NULL;
}
bool emptyList()
{if(head==NULL )
{return true;
}
else
{return false;
}
}

void insert_end_patient(int i,char Nam[30] ,int a,char g[30],char phone[30],char dis[30],char med[30],char add[30],int r
,int w )
{patient_node *temp;
temp=new patient_node;
strcpy(temp->name,Nam);
temp->id=i;
temp->age=a;
strcpy(temp->gender,g);
strcpy(temp->phone_no,phone);
strcpy(temp->disease,dis);
strcpy(temp->medicine,med);
strcpy(temp->address,add);
temp->room_no=r;
temp->ward_no=w;
patient_node *ptr;
ptr=head;
if(emptyList())
{tail=head=temp;
}
else
{ptr->next=temp;
temp->prev=ptr;
ptr=ptr->next;
}
}
}

```

```

    }
else
{
tail->next=temp;
temp->prev=tail;
tail=temp;
temp->next=head;
head->prev=temp;
}
}
void delete_record()
{int i;
cout<<"ENTER THE NODE YOU WANT TO DELETE ";
cin>>i;
patient_node *s1;
patient_node*s2;
s1=NULL;
s2=head;
while(s2->next!=head && s2->id!=i)
{s1=s2;
s2=s2->next;
}
if(s2->id!=i)
{cout<<endl;
cout<<"THIS VALUE DONOT EXIST "<<endl;
cout<<endl;
cout<<"SO THE LIST WILL BE "<<endl;
}
else if(s2==head)
{
s1=head->prev;
head=head->next;
tail->next=s2->next;
s1->next=head;
s2->next=NULL;
s2->prev=NULL;
delete s2;
}
else if(s2->next==head)
{s1->next =head;
head->prev=s1;
s2->prev=NULL;
s2->next=NULL;
delete s2;
}
else
{
s1->next=s2->next;
s2->next->prev=s1;
s2->prev=NULL;
s2->next=NULL;
delete s2;}
}
void insert()
{int n,i,a,r , w;
char Nam[30], g[30], phone[30], dis[30],med[30], add[30];
patient_node *temp;
temp=new patient_node;
cout<<"ENTER THE ID YOU WANT TO INSERT"<<endl;

```

```

cin>>i;
temp->id=i;
cout<<"ENTER THE NAME OF THE PATIENT YOU WANT TO INSERT"<<endl;
cin>>Nam;
strcpy(temp->name,Nam);
cout<<"ENTER THE AGE OF THE PATIENT YOU WANT TO INSERT"<<endl;
cin>>a;
temp->age=a;
cout<<"ENTER THE GENDER OF THE PATIENT YOU WANT TO INSERT"<<endl;
cin>>g;
strcpy(temp->gender,g);
cout<<"ENTER THE PHONE OF THE PATIENT YOU WANT TO INSERT"<<endl;
cin>> phone;
strcpy(temp->phone_no,phone);
cout<<"ENTER THE DISEASES OF THE PATIENT YOU WANT TO INSERT"<<endl;
cin>>dis;
strcpy(temp->disease,dis);
cout<<"ENTER THE ADDRESS OF THE PATIENT YOU WANT TO INSERT"<<endl;
cin>>add;
strcpy(temp->address,add);
cout<<"ENTER THE MEDICINE OF THE PATIENT YOU WANT TO INSERT"<<endl;
cin>>med;
strcpy(temp->medicine,med);
cout<<"ENTER THE ROOM OF THE PATIENT YOU WANT TO INSERT"<<endl;
cin>>r;
temp->room_no=r;
cout<<"ENTER THE WARD OF THE PATIENT YOU WANT TO INSERT"<<endl;
cin>>w;
temp->ward_no=w;
int done=0;
if(head==NULL)
{
    head=temp;
    temp->id=i;
    temp->next=NULL;
    done=1;
}
else if(head->id>=i)
{
    temp->id=i;
    temp->next=head;
    temp->next=tail->next;
    temp->prev=tail;
    tail->next->prev=temp;
    tail->next=temp;
    head=temp;
    done=1;
}
else
{
    if(tail->id<i)
    {
        temp->id=i;
        tail->next=temp;
        temp->prev=tail;
        tail=temp;
        temp->next=head;
        head->prev=temp;
        done=1;
    }
}

```

```

        }

        if(done==0)
        {temp->id=i;

patient_node *s1;
patient_node *s2;
s1=head;
s2=s1->next;
while(s2!=head)
{if(s1->id<i && s2->id>i)
{temp->next=s2;
s1->next=temp;
temp->prev=s1;
s2->prev=temp;
done=1;
break;
}
s1=s1->next;
s2=s2->next;}

        }

    }

void traverse_pateint()
{patient_node *ptr;
ptr=head;
if(emptyList())
{cout<<"INFORMATION ABOUT THE PATEINT IS "<<endl;;
do
{
cout<<"NAME: "<<ptr->name<<endl;
cout<<"ID: "<<ptr->id<<endl;
cout<<"AGE: "<<ptr->age<<endl;
cout<<"GENDER: "<<ptr->gender<<endl;
cout<<"PHONE NUMBER: "<<ptr->phone_no<<endl;
cout<<"DISEASE: "<<ptr->disease<<endl;
cout<<"MEDICINE: "<<ptr->medicine<<endl;
cout<<"ADDRESS: "<<ptr->address<<endl;
cout<<"ROOM NUMBER: "<<ptr->room_no<<endl;
cout<<"WARD NUMBER: "<<ptr->ward_no<<endl;
cout<<endl;
ptr=ptr->next;
}
while(ptr!=head);
}
}

void search()
{int i;
cout<<"ENTER THE ID OF THE PATIENT WHOM YOU WANT TO SEARCH"<<endl;
cin>>i;
patient_node *ptr;
ptr=head;
if(emptyList())
{do
{
if(ptr->id==i)

```

```

{cout<<"NAME: "<<ptr->name<<endl;
cout<<"ID: "<<ptr->id<<endl;
cout<<"AGE: "<<ptr->age<<endl;
cout<<"GENDER: "<<ptr->gender<<endl;
cout<<"PHONE NUMBER: "<<ptr->phone_no<<endl;
cout<<"DISEASE: "<<ptr->disease<<endl;
cout<<"MEDICINE: "<<ptr->medicine<<endl;
cout<<"ADDRESS: "<<ptr->address<<endl;
cout<<"ROOM NUMBER: "<<ptr->room_no<<endl;
cout<<"WARD NUMBER: "<<ptr->ward_no<<endl;
}
ptr=ptr->next;
}
while(ptr!=head);
}
}
void update()
{int r, w,i;
cout<<"ENTER THE ID WHOM YOU WANT TO UPDATE THE RECORD"<<endl;
cin>>i;
patient_node *ptr;
ptr=head;
if(emptyList())
{do
{
if(ptr->id==i)
{cout<<"ENTER THE NEW ROOM NUMBER"<<endl;
cin>>r;
ptr->room_no= r;
cout<<"ENTER THE NEW WARD NUMBER"<<endl;
cin>>w;
ptr->ward_no= w;
ptr->id=ptr->id;
strcpy(ptr->name,ptr->name);
ptr->age=ptr->age;
strcpy(ptr->gender,ptr->gender);
strcpy(ptr->phone_no,ptr->phone_no);
strcpy(ptr->disease,ptr->disease);
strcpy(ptr->address,ptr->address);
strcpy(ptr->medicine,ptr->medicine);
}
ptr=ptr->next;
}while(ptr!=head);
}
}
class Room
{
private:
struct room_node
{int roomno;
int no_of_beds;
bool ac;
int occupied_bed;
room_node *next;
room_node *prev;
};
struct room_node *head,*tail;
public:
Room()

```

```

{head=NULL;
tail=NULL;
}
bool empty()
{if(head==NULL)
{return true;
}
else
{
    return false;
}

void insert_end_room(int rn, int nob,bool AC, int ob)
{
room_node *temp1;
temp1=new room_node;
temp1->roomno=rn;
temp1->no_of_beds=nob;
temp1->ac=AC;
temp1->occupied_bed=ob;
room_node *ptr;
ptr=head;
if(empty())
{tail=head=temp1;
}
else
{
tail->next=temp1;
temp1->prev=tail;
tail=temp1;
temp1->next=head;
head->prev=temp1;
}
}
void traverse_room()
{room_node *pointer;
pointer=head;
if(!empty())
{
cout<<"INFORMATION ABOUT THE ROOOM IS "<<endl;
do
{
cout<<"ROOM NUMBER: "<<pointer->roomno<<endl;
cout<<"NUMBER OF BEDS: "<<pointer->no_of_beds<<endl;
cout<<"AC: "<<pointer->ac<<endl;
cout<<"OCCUPIED BED: "<<pointer->occupied_bed<<endl;
cout<<endl;
pointer=pointer->next;
}
while(pointer!=head);
}}
void searchRoom()
{int r;
cout<<"ENTER THE ROOM NUMBER WHOM YOU WANT TO SEARCH"<<endl;
cin>>r;
room_node *pointer;
pointer=head;
if(!empty())
{do
{

```

```

if(pointer->roomno==r)
{cout<<"ROOM NUMBER: "<<pointer->roomno<<endl;
cout<<"NUMBER OF BEDS: "<<pointer->no_of_beds<<endl;
cout<<"AC: "<<pointer->ac<<endl;
cout<<"OCCUPIED BED: "<<pointer->occupied_bed<<endl;
cout<<endl;
}
pointer=pointer->next;
}
while(pointer!=head);

}

}

};

class Ward
{
private:
struct ward_node
{int ward_no;
int no_of_beds;
bool ac;
int occupied_bed;
ward_node *next;
ward_node *prev;
};
struct ward_node *head,*tail;
public:
    Ward()
{head=NULL;
tail=NULL;
}
bool empty()
{if(head==NULL)
{return true;
}
else
    return false;
}
void insert_end_room(int wn, int nob,bool AC, int ob)
{
ward_node *temp1;
temp1=new ward_node;
temp1->ward_no=wn;
temp1->no_of_beds=nob;
temp1->ac=AC;
temp1->occupied_bed=ob;
ward_node *ptr;
ptr=head;
if(empty())
{tail=head=temp1;
}
else
{
tail->next=temp1;
temp1->prev=tail;
tail=temp1;
temp1->next=head;
head->prev=temp1;
}
}
}

```

```

void traverse_ward()
{ward_node *pointer;
pointer=head;
if(!empty())
{
cout<<"INFORMATION ABOUT THE WARD IS "<<endl;
do
{
cout<<"WARD NUMBER: "<<pointer->ward_no<<endl;
cout<<"NUMBER OF BEDS: "<<pointer->no_of_beds<<endl;
cout<<"AC: "<<pointer->ac<<endl;
cout<<"OCCUPIED BED: "<<pointer->occupied_bed<<endl;
cout<<endl;
pointer=pointer->next;
}
while(pointer!=head);
}}
void searchWard()
{int w;
cout<<"ENTER THE WARD NUMBER WHOM YOU WANT TO SEARCH"<<endl;
cin>>w;
ward_node *pointer;
pointer=head;
if(!empty())
{do
{
if(pointer->ward_no==w)
{cout<<"WARD NUMBER: "<<pointer->ward_no<<endl;
cout<<"NUMBER OF BEDS: "<<pointer->no_of_beds<<endl;
cout<<"AC: "<<pointer->ac<<endl;
cout<<"OCCUPIED BED: "<<pointer->occupied_bed<<endl;
cout<<endl;
}
pointer=pointer->next;
}
while(pointer!=head);

}
}
int main()
{
Patient l;
Room r;
Ward w;
char choice;
char n;
do
{cout<<"*****"<<endl;
cout<<"WELCOME TO HOSPITAL MANAGEMENT SYSTEM "<<endl;
cout<<"THIS PROJECT IS CREATED BY AQSA TABASSUM AND SHAMSA KANWAL"<<endl;
cout<<"PRESS 1 TO SEE THE LIST OF THE PATEINTS "<<endl;
cout<<"PRESS 2 TO SEE THE LIST OF THE ROOMS "<<endl;
cout<<"PRESS 3 TO SEE THE LIST OF THEWARDS "<<endl;
cout<<"PRESS 4 TO DELETE THE INFORMATION ABOUT THE PATEINT"<<endl;
cout<<"PRESS 5 TO INSERT THE INFORMATION OF THE PATIENT ANYWHERE YOU WANT"<<endl;
cout<<"PRESS 6 TO SEARCH THE THE PATIENT "<<endl;
cout<<"PRESS 7 TO SEARCH THE THE ROOM "<<endl;
cout<<"PRESS 8 TO SEARCH THE THE WARD "<<endl;
}

```

```

cout<<"PRESS 9 TO UPDATE THE THE RECORD "<<endl;
cout<<"*****"*<<endl;
cin>>choice;
if(choice=='1')
{
l.insert_end_pateint(1,"AQSA",18,"FEMALE","0345","MALARIA","RIZEK","RAWALPINDI",3,4);
l.insert_end_pateint(2,"ANUM",17,"FEMALE","034005","CANCER","GLUCOPHAGE","ISLAMBAD",3,5);
l.insert_end_pateint(3,"ALI",18,"MALE","0345","BLOOD PRESSURE","CO-EASY","RAWALPINDI",3,4);
l.insert_end_pateint(4,"SADIA",17,"FEMALE","034005","BRAIN
HAMERAGE","PANADOL","LAHORE",3,5);
l.insert_end_pateint(5,"HIRA",18,"FEMALE","0345","MALARIA","RIZEK","RAWALPINDI",3,4);
l.insert_end_pateint(7,"AMNA",17,"FEMALE","034005","HEART ATTACK","CALPOL","ISLAMBAD",3,5);
l.traverse_pateint();
}

else if(choice=='2')
{r.insert_end_room(1,4,true,2);
r.insert_end_room(2,4,false,2);
r.insert_end_room(3,4,true,2);
r.insert_end_room(4,4,false,2);
r.insert_end_room(5,4,false,2);
r.traverse_room();
}
else if(choice=='3')
{w.insert_end_room(1,3,true,2);
w.insert_end_room(2,3,true,2);
w.insert_end_room(3,3,false,1);
w.insert_end_room(4,4,true,3);
w.insert_end_room(5,5,false,1);
w.traverse_ward();
}
else if(choice=='4')
{l.delete_record();
l.traverse_pateint();
}
else if(choice=='5')
{l.insert();
l.traverse_pateint();
}
else if(choice=='6')
{l.search();
}
else if(choice=='7')
{r.searchRoom();
}
else if(choice=='8')
{w.searchWard();
}
else if(choice=='9')
{l.update();
l.traverse_pateint();
}
else
{
    cout<<"INCORRECT NUMBER"<<endl;
    cout<<"DO YOU WANT TO CONTINUE(Y/N)";
    cin>>n;
}

}while(n=='Y' ||n=='y');

```

```

        system("pause");
    return 0;
}

```

OUTPUT:

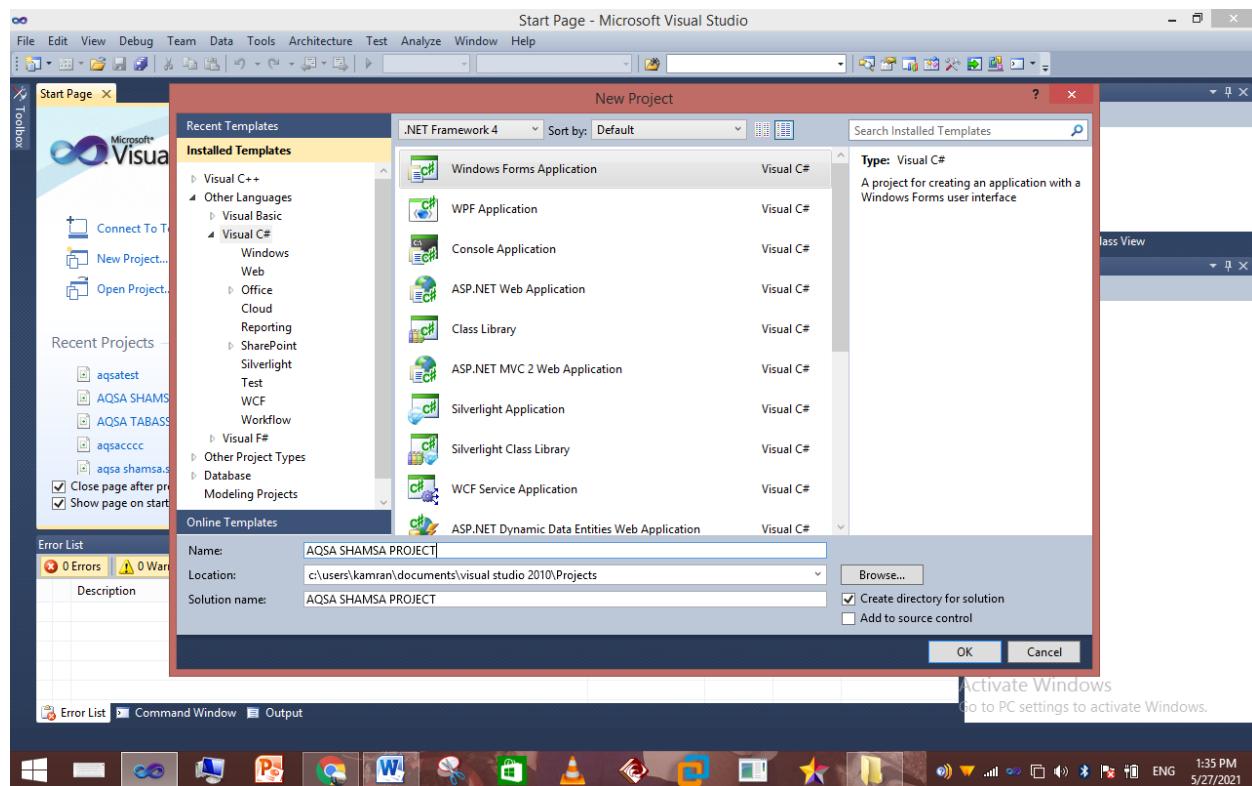
```

*****
WELCOME TO HOSPITAL MANAGEMENT SYSTEM
THIS PROJECT IS CREATED BY AQSA TABASSUM AND SHAMSA KANWAL
PRESS 1 TO SEE THE LIST OF THE PATEINTS
PRESS 2 TO SEE THE LIST OF THE ROOMS
PRESS 3 TO SEE THE LIST OF THE WARDS
PRESS 4 TO DELETE THE INFORMATION ABOUT THE PATEINT
PRESS 5 TO INSERT THE INFORMATION OF THE PATIENT ANYWHERE YOU WANT
PRESS 6 TO SEARCH THE THE PATIENT
PRESS 7 TO SEARCH THE THE ROOM
PRESS 8 TO SEARCH THE THE WARD
PRESS 9 TO UPDATE THE THE RECORD
*****

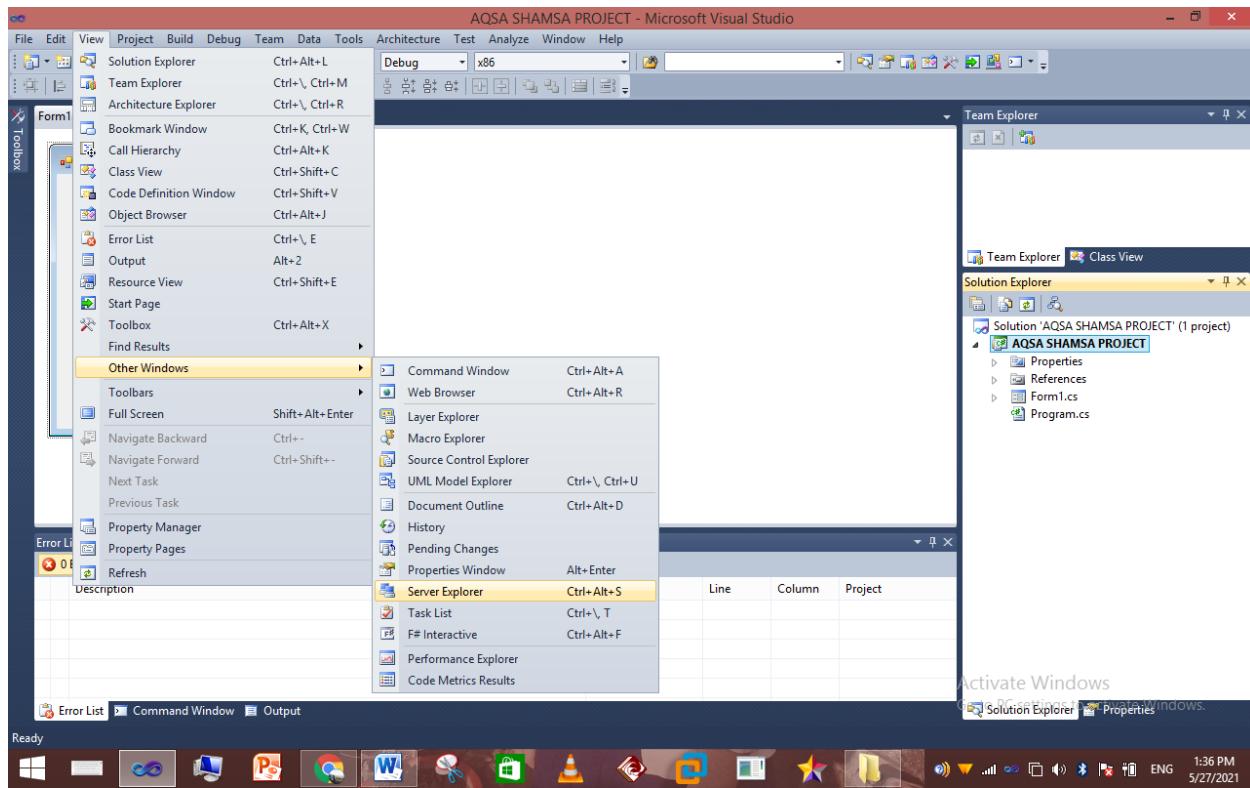
```

CONNECTION OF SQL WITH MICROSOFT VISUAL

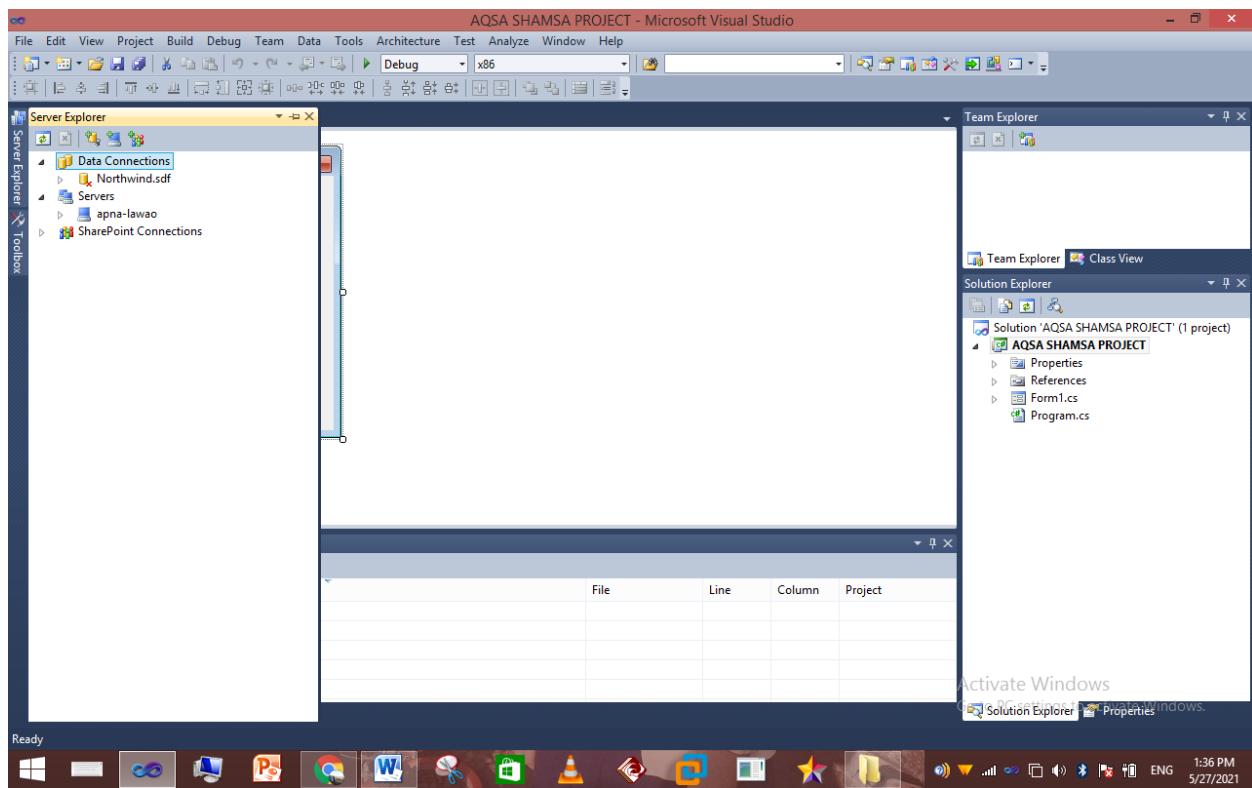
STUDIO:First open the visual studio window, then open the Window Form application and name the file.



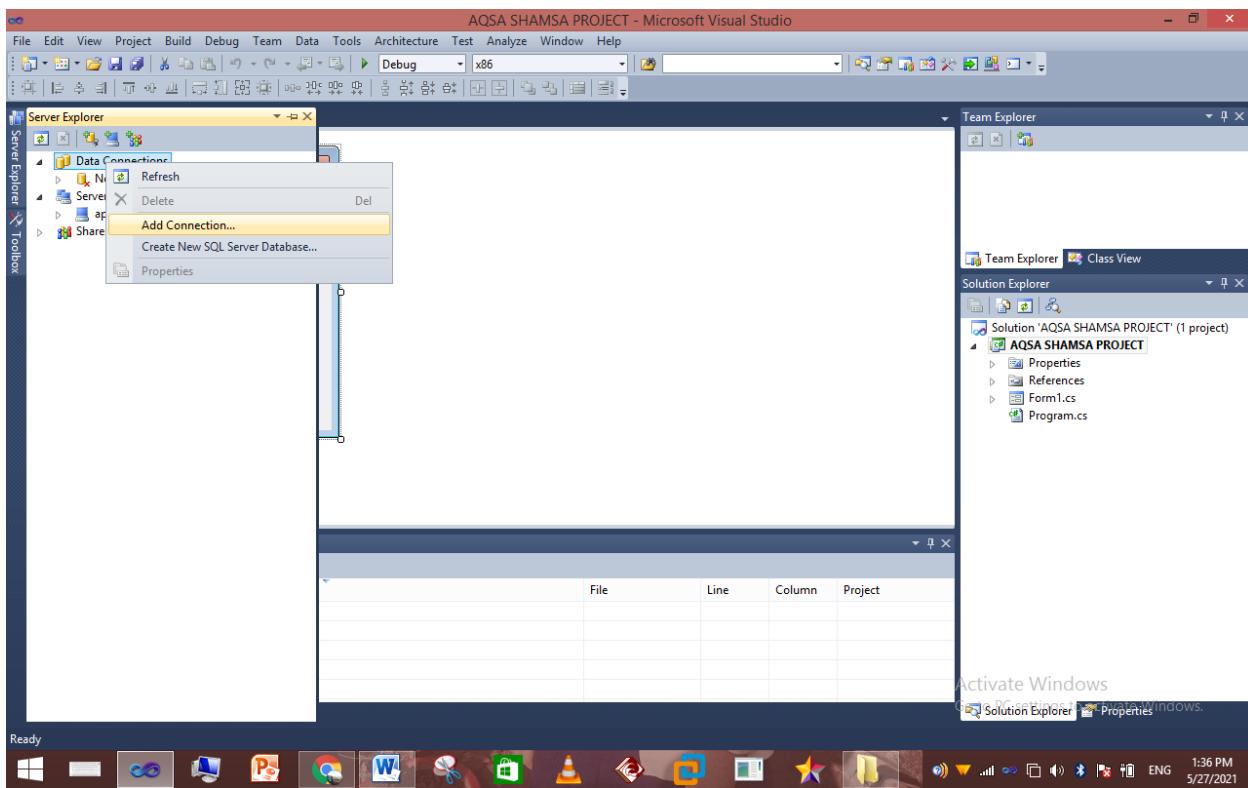
Then open the view tab and open server explorer from other windows option.



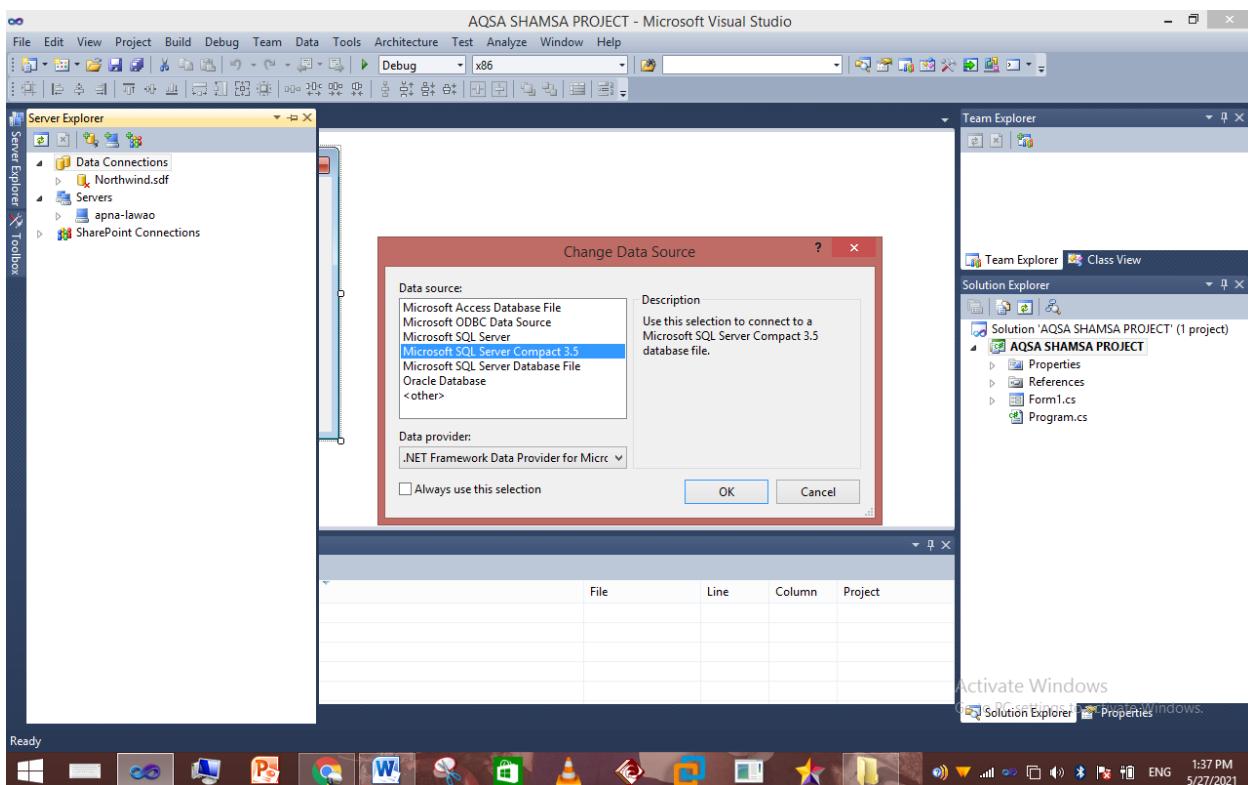
Now here you will make the connection.

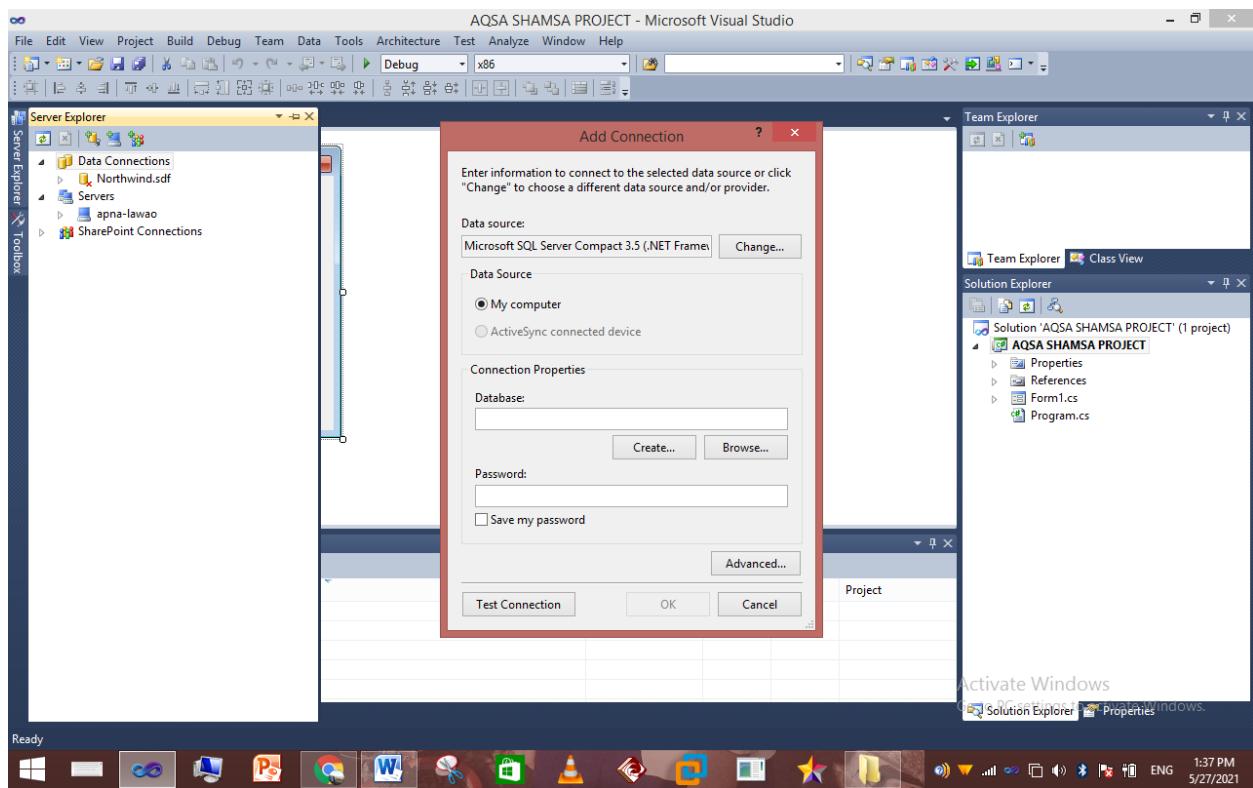


Now you will select add connection by right clicking the data connection.

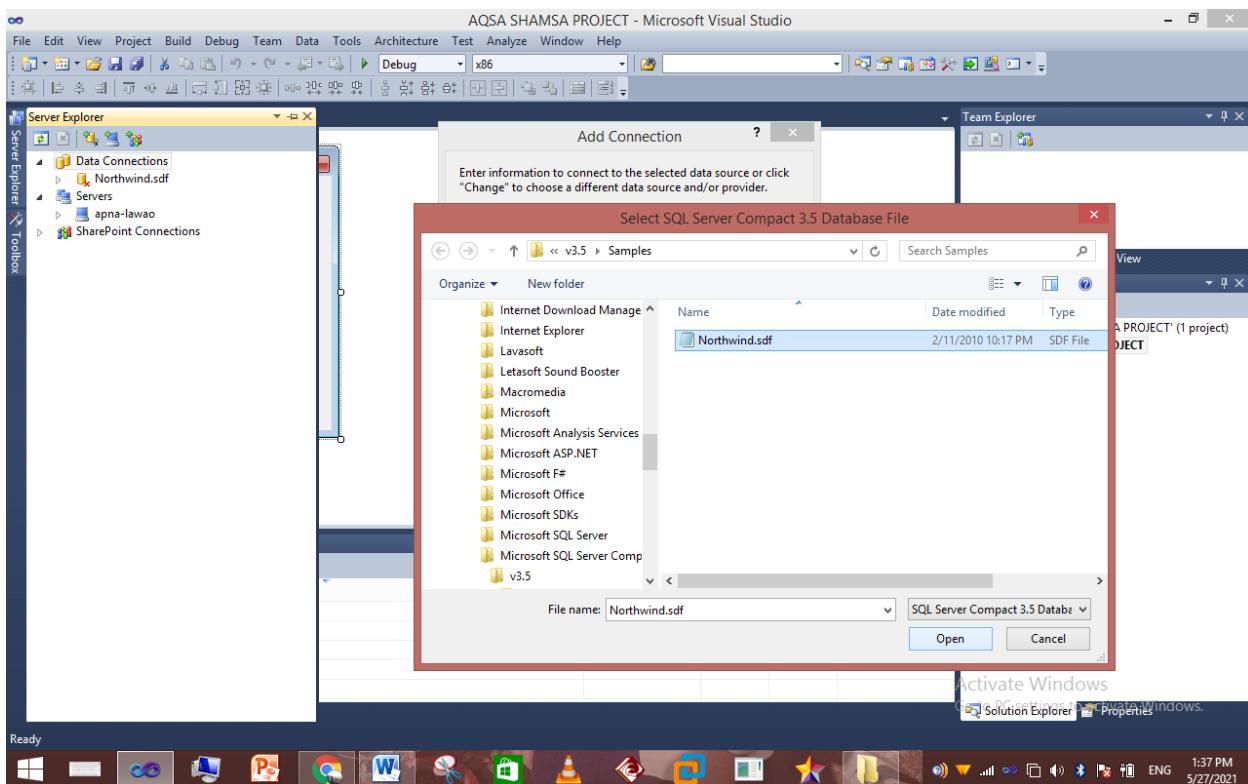


Now selection Microsoft sql server compact 3.5.

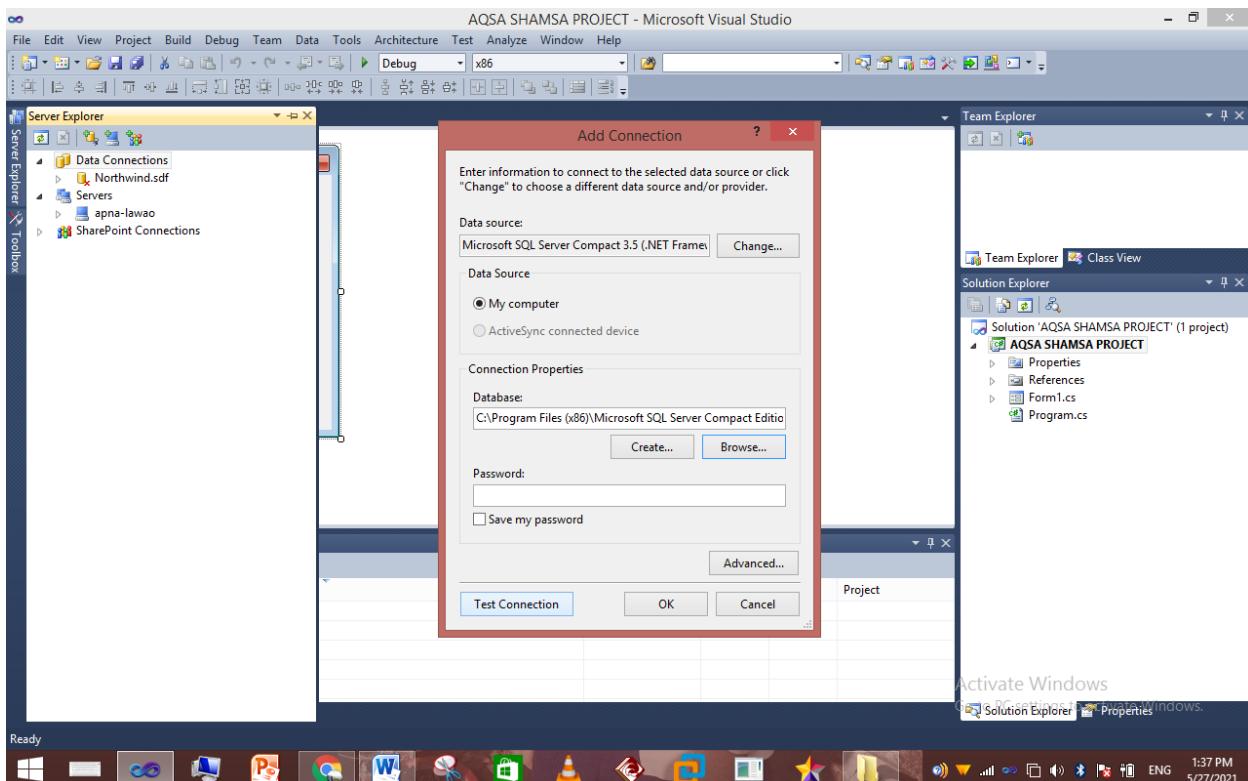




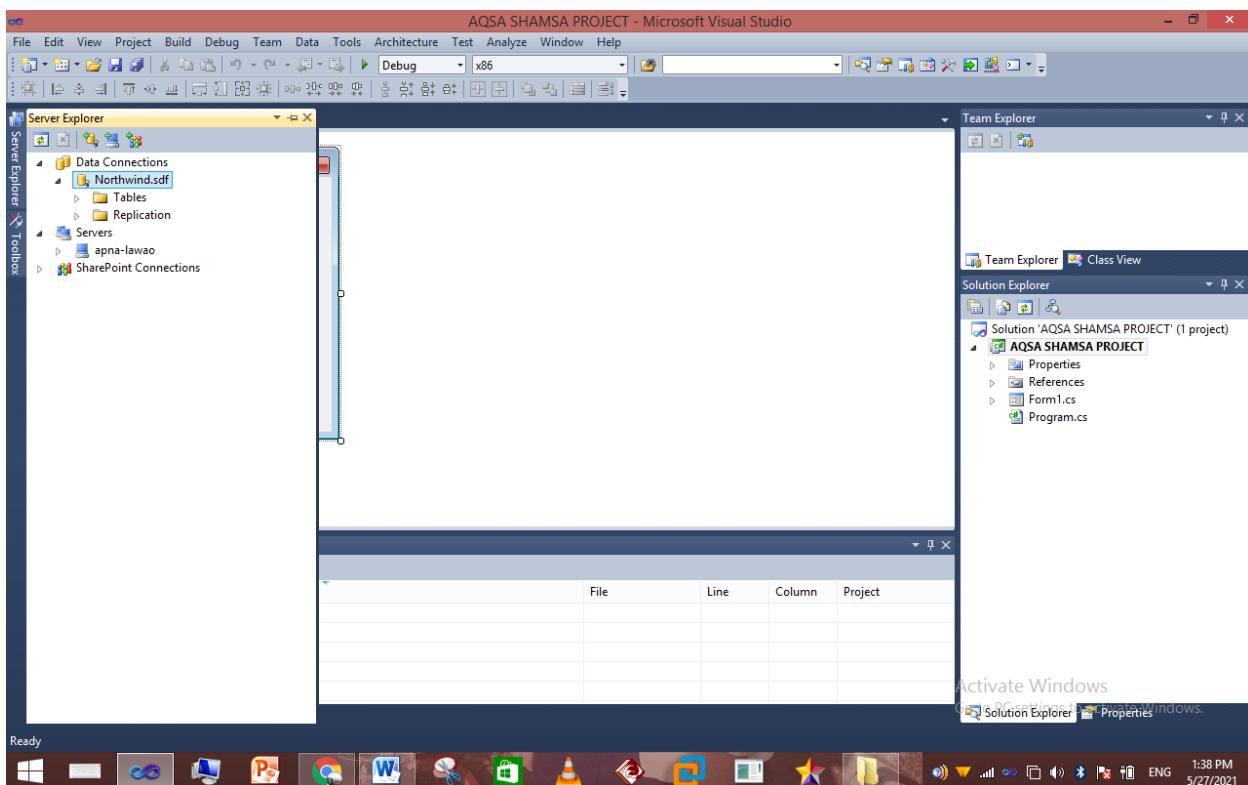
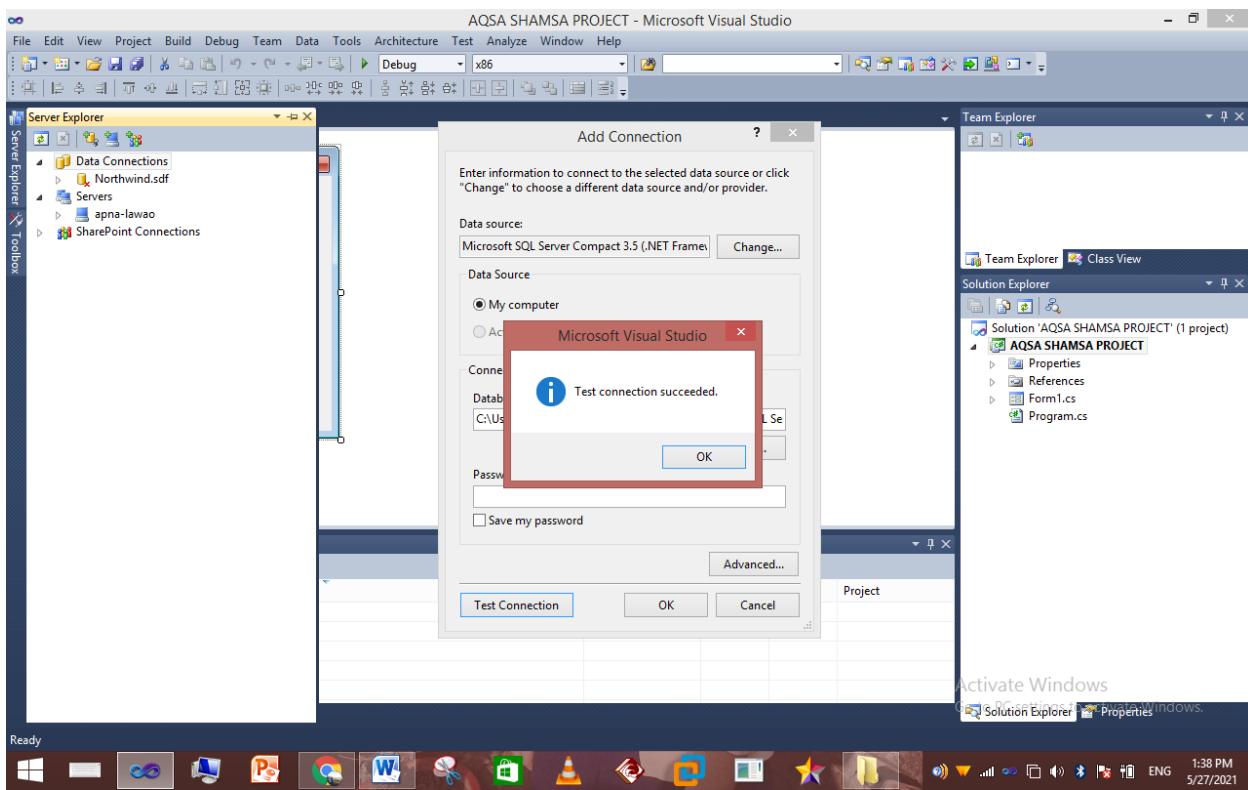
Click on browse and select northwindow.sdf .



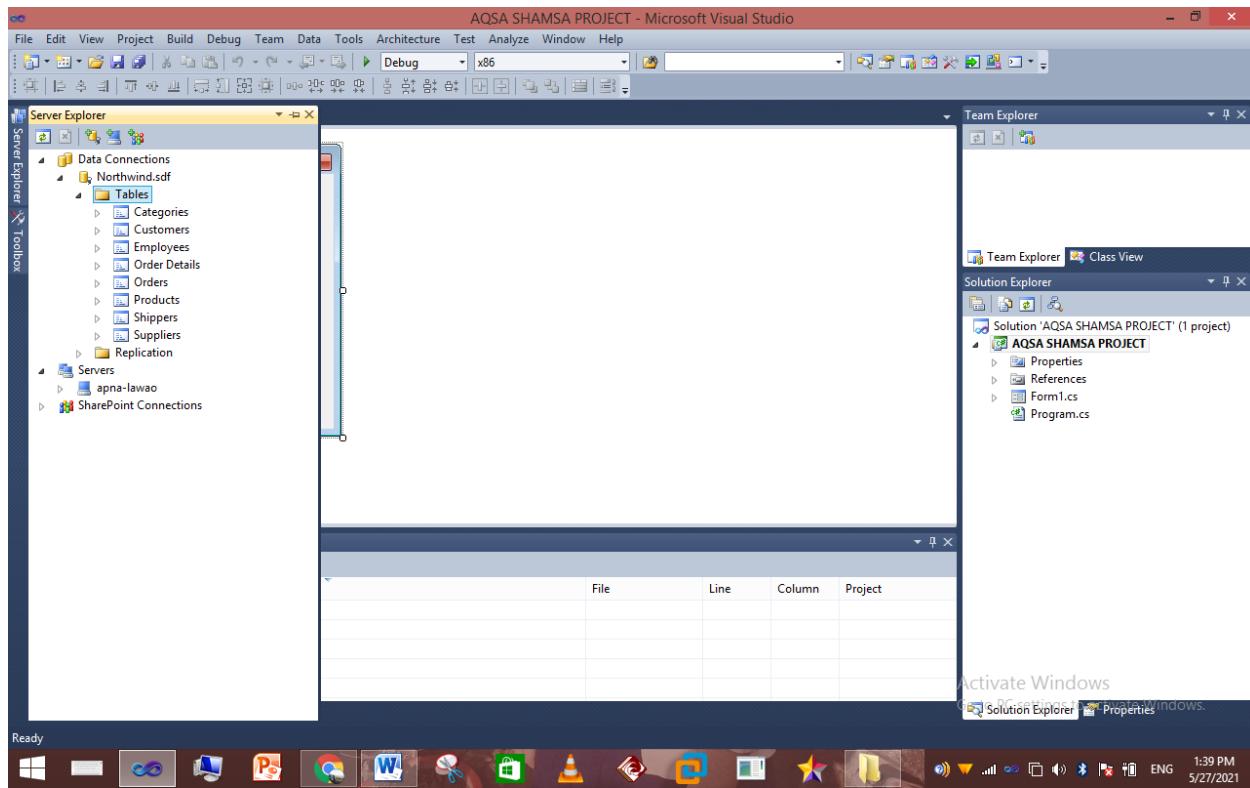
Click on test connection.



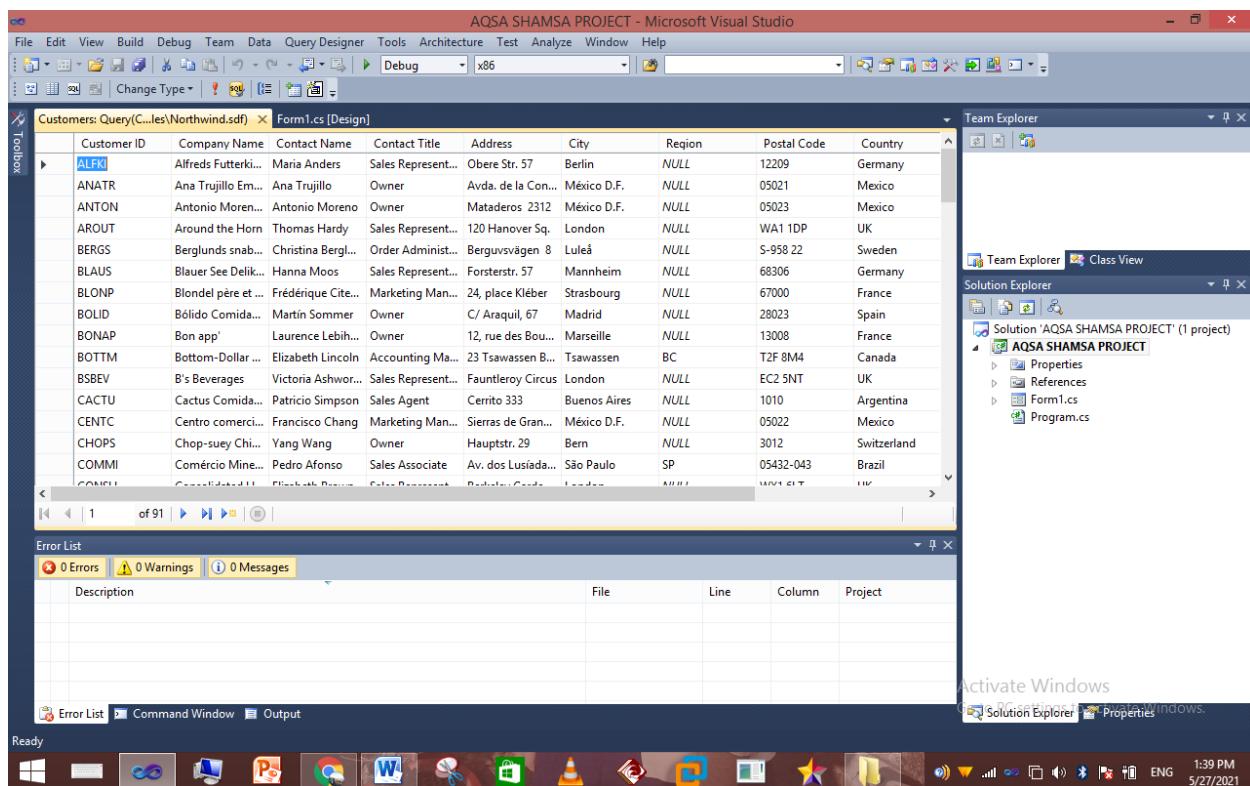
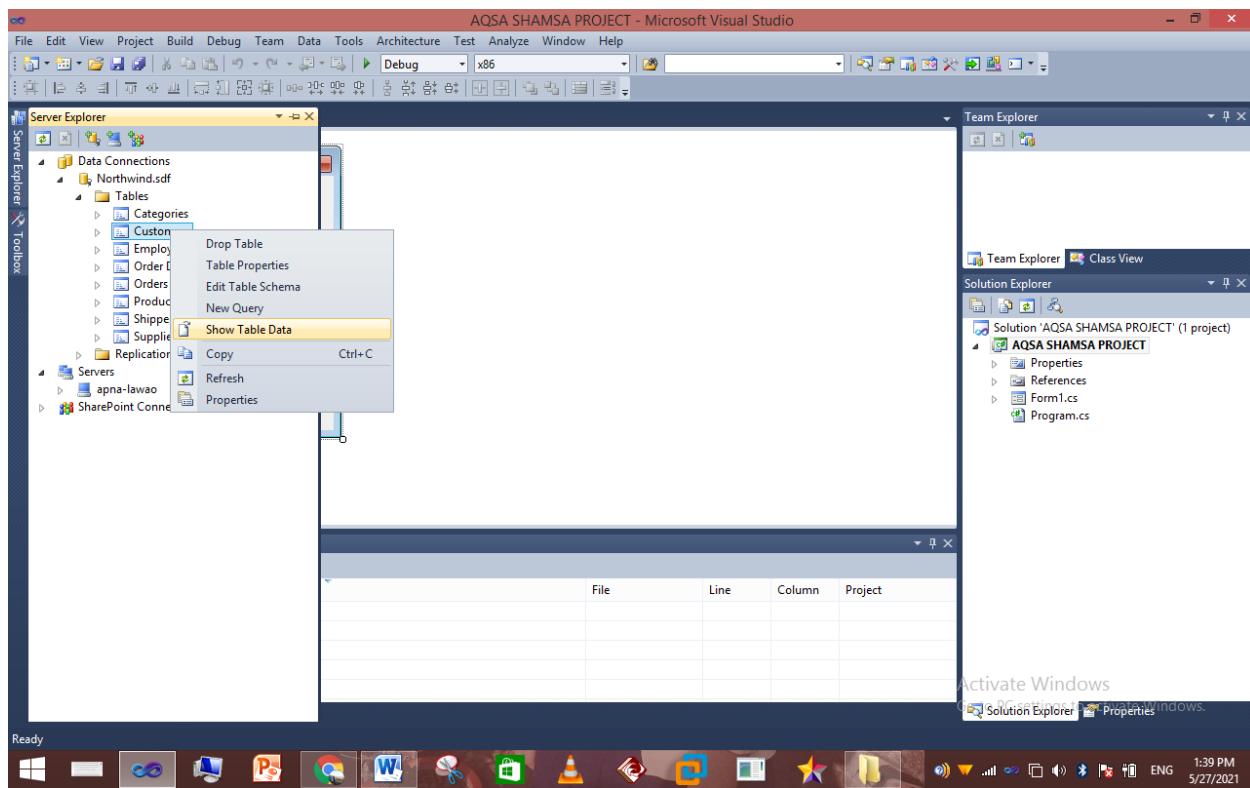
Connection is succeeded.



You can see the list of tables:

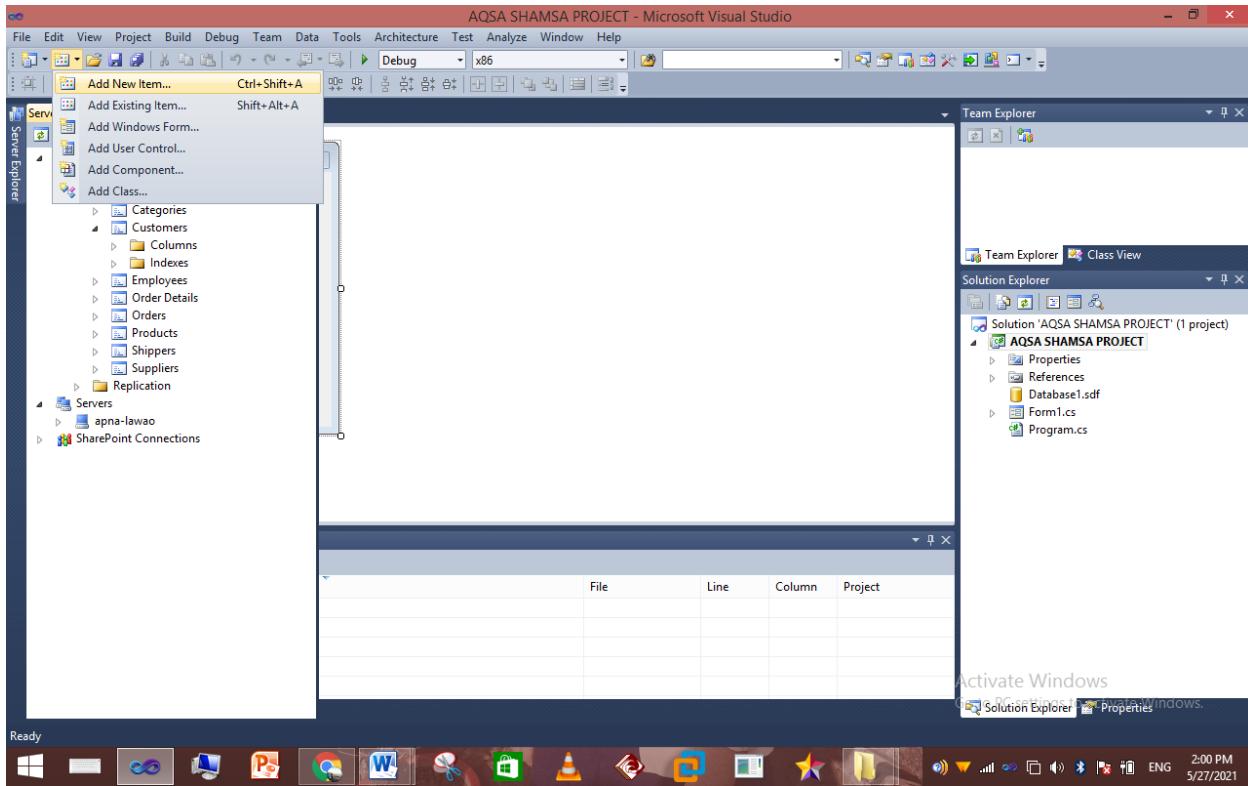


Click on show table data to see the record of table:

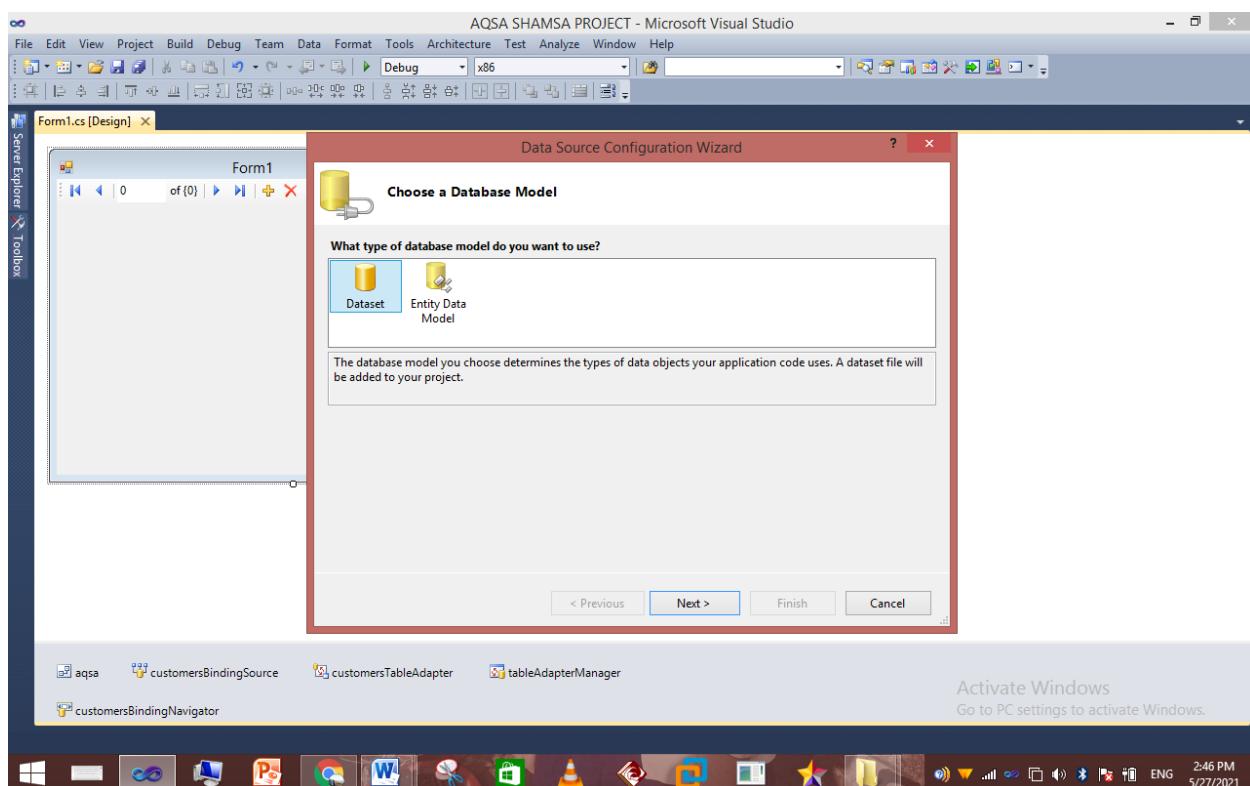
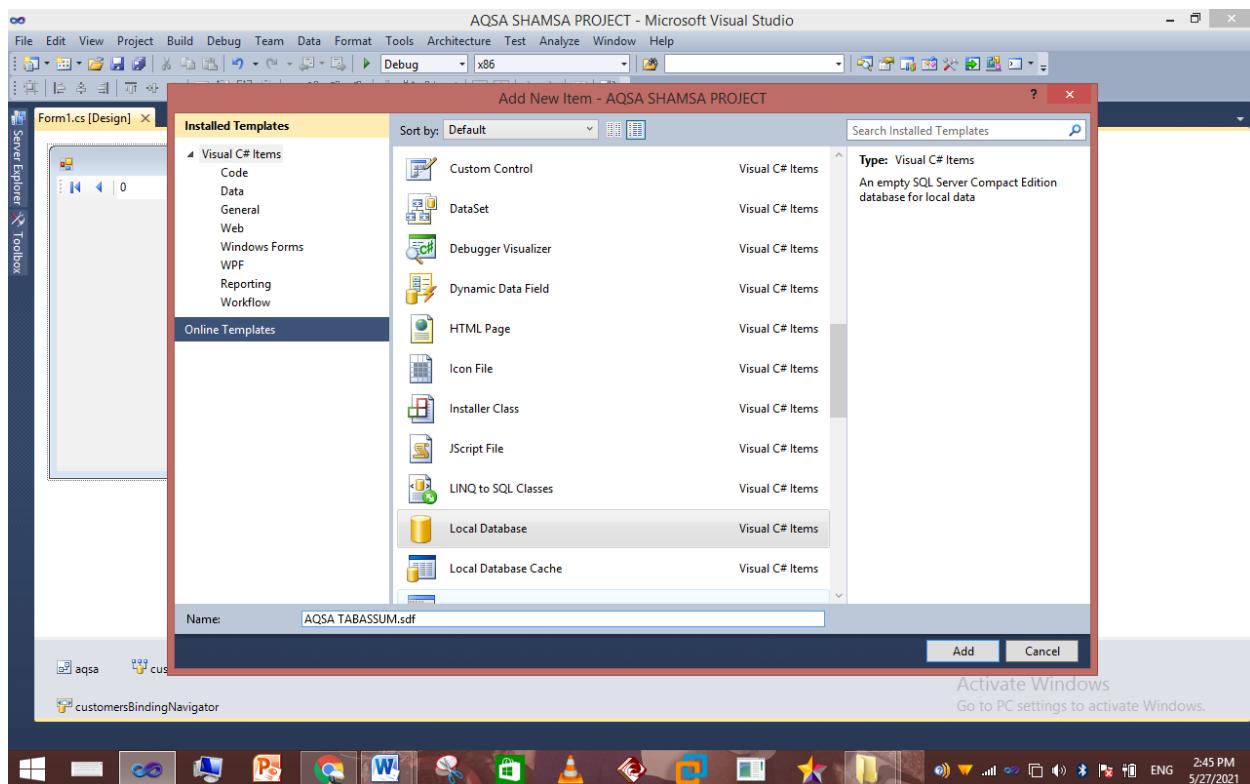


NOW TO PLACE TABLES IN THE FORM:

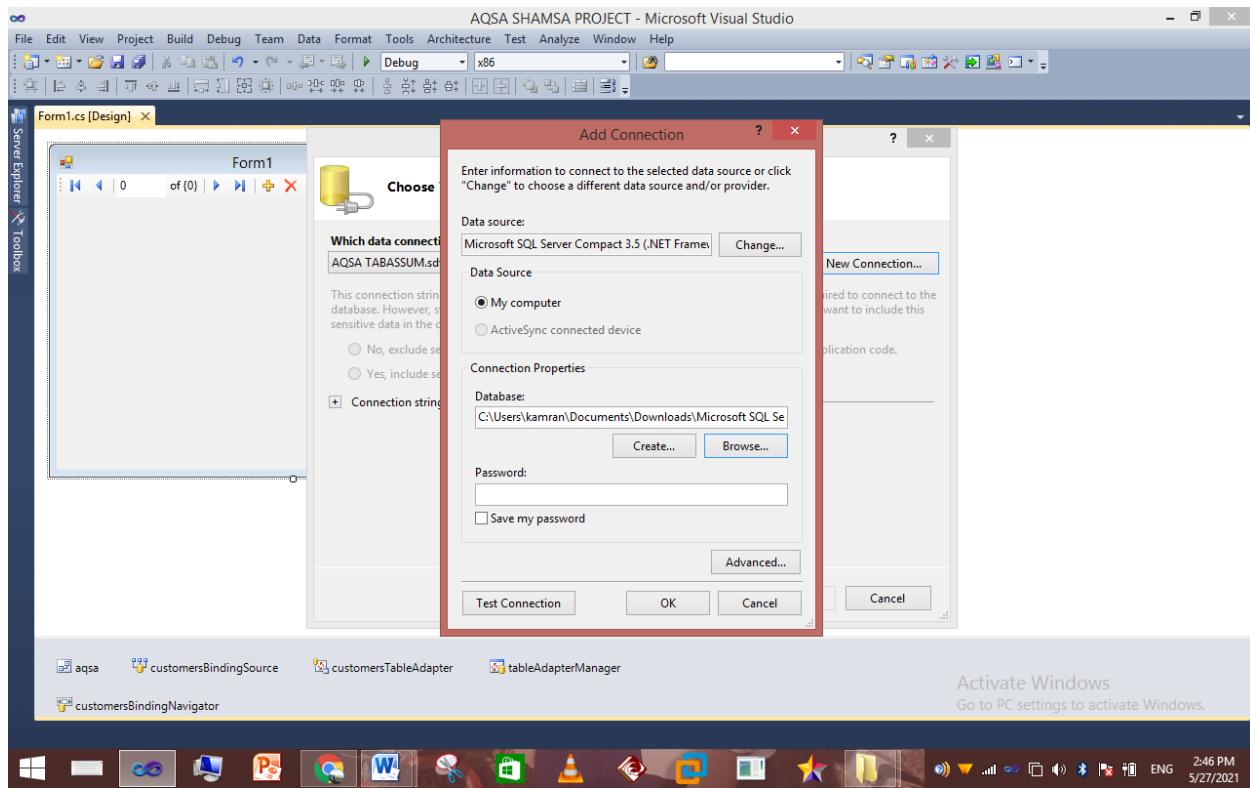
First click on Add new item option:



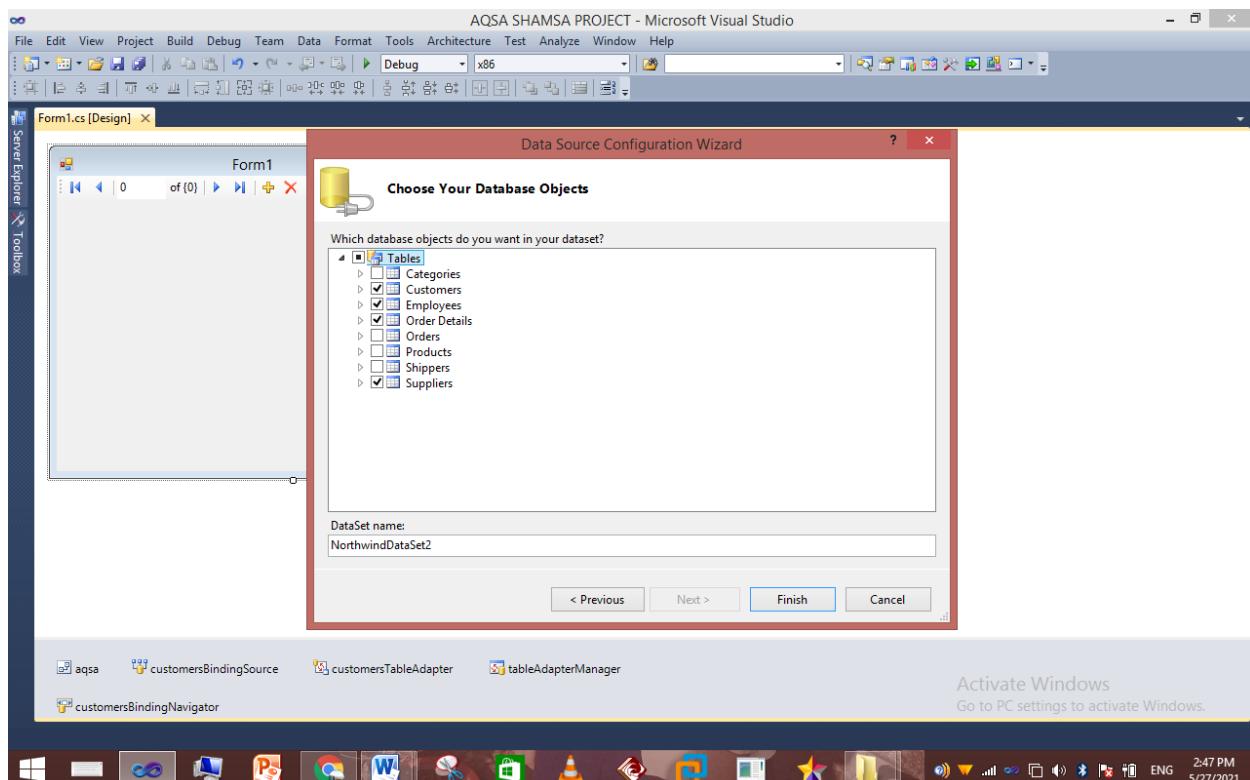
Now select the local database option.



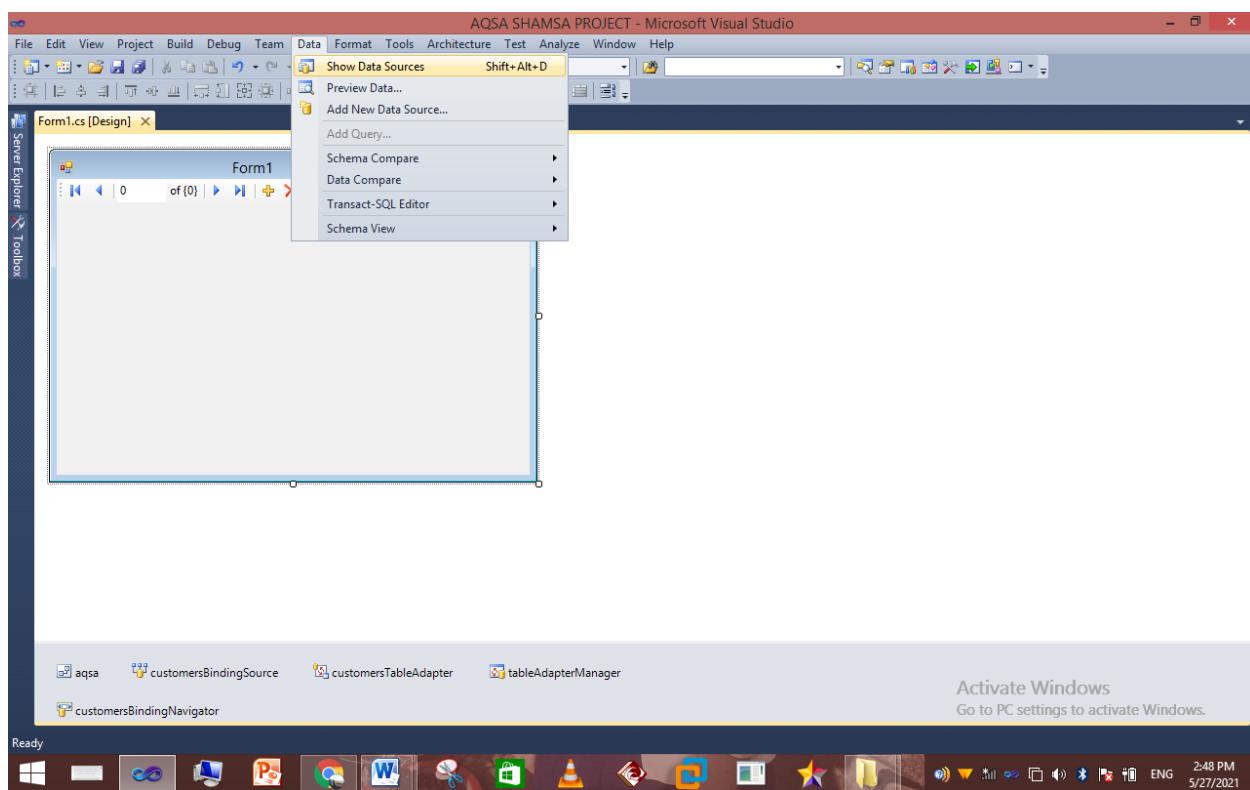
Make the connection:

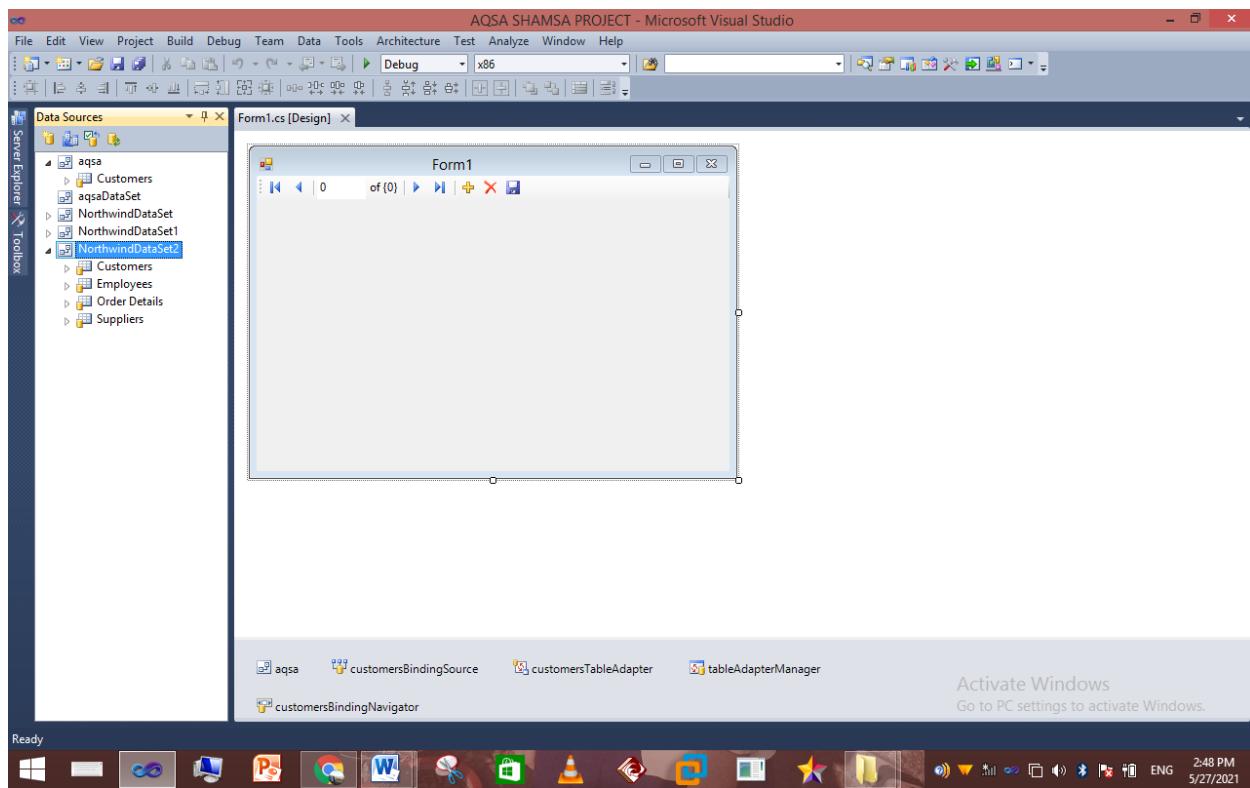


Now, you will select the tables , which you want on the forms. You can also select all the tables.

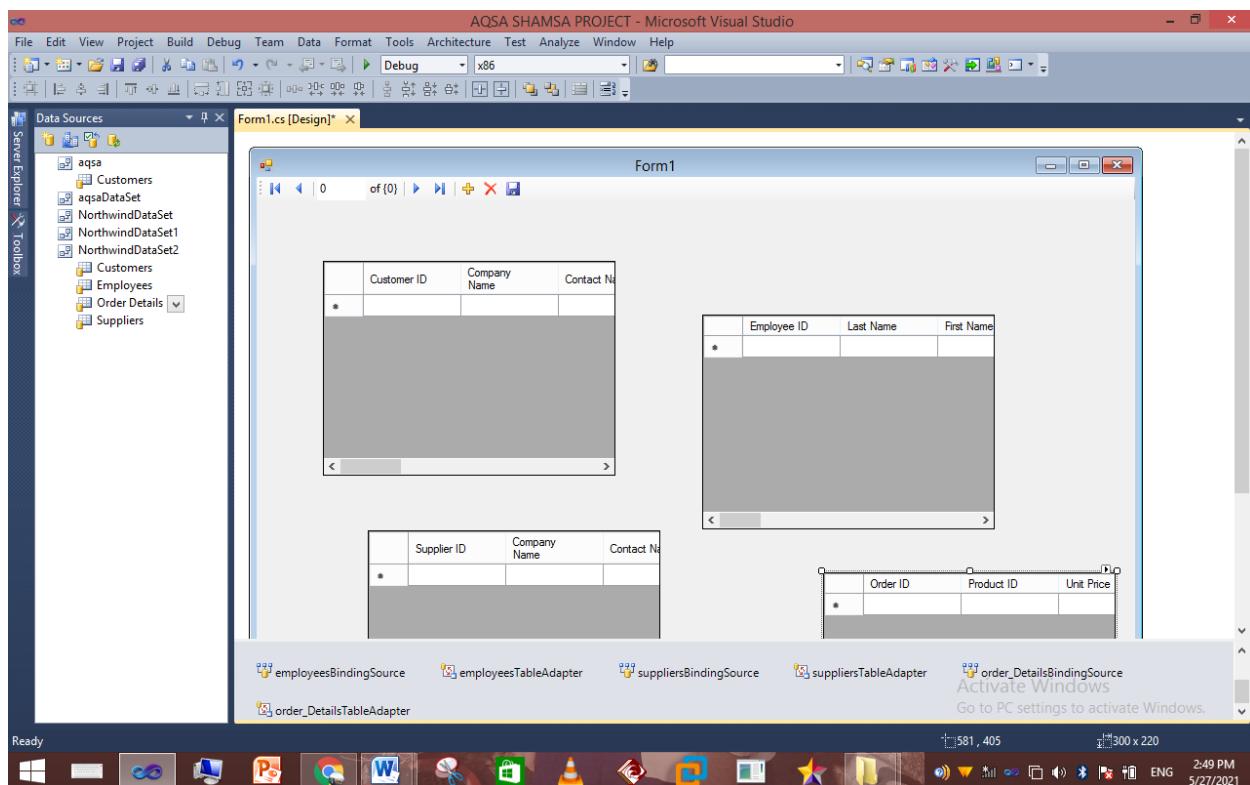


Now select the show data sources to view the tables selected:

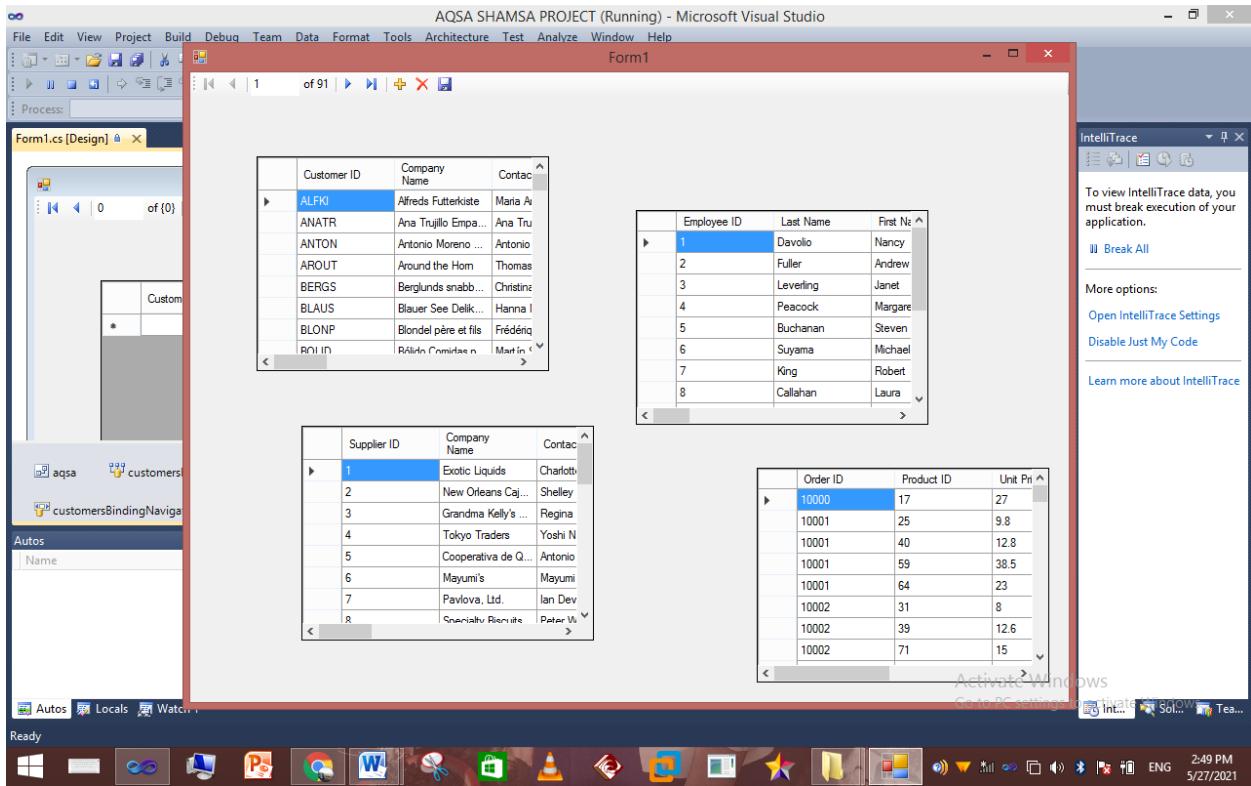




Now drag the tables from the list to the form:

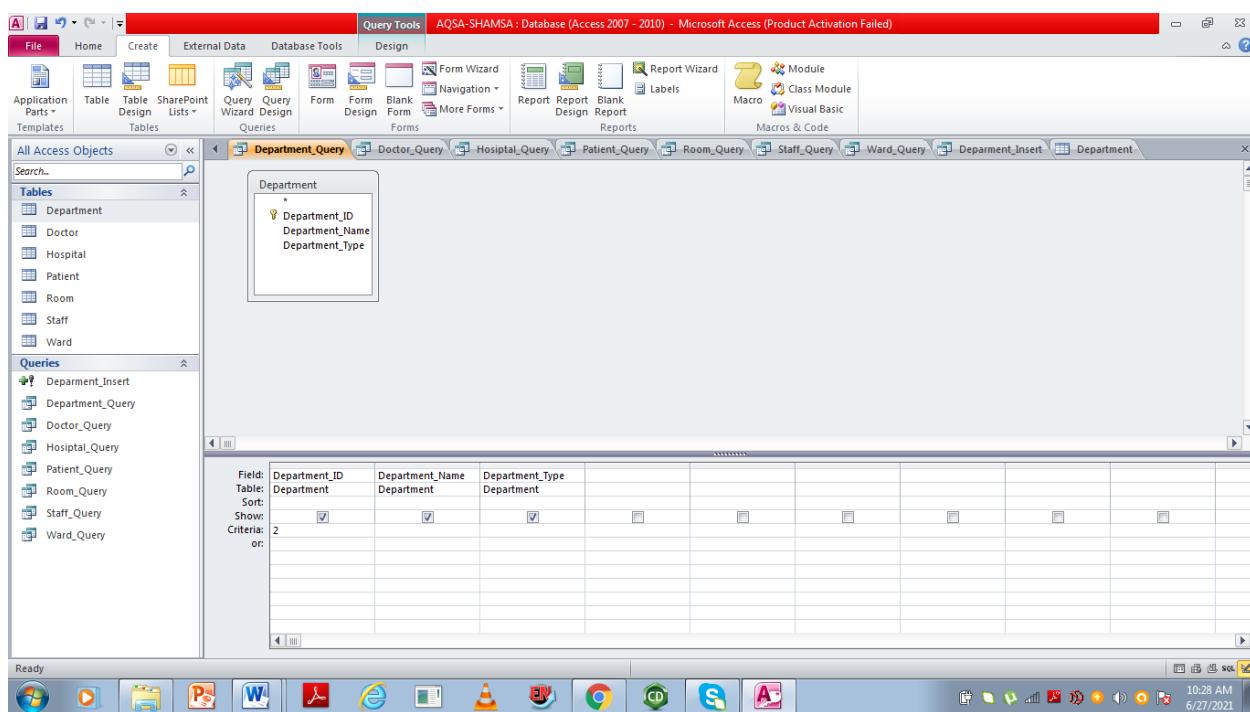
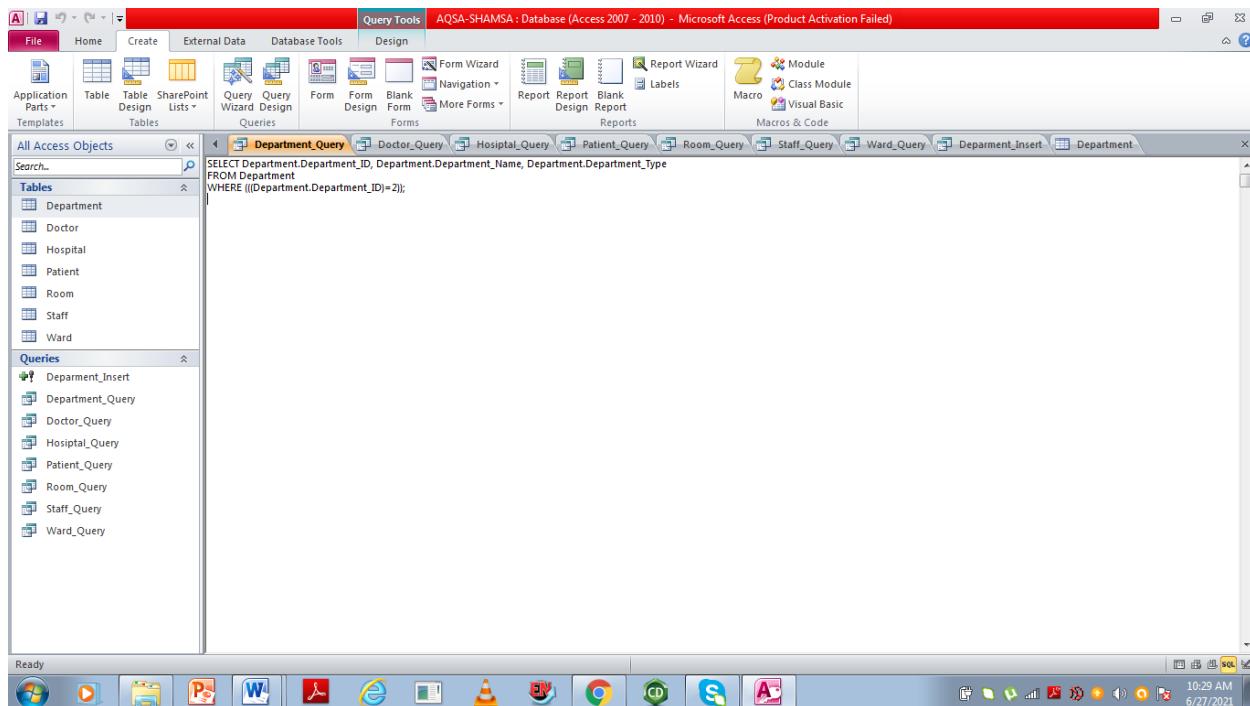


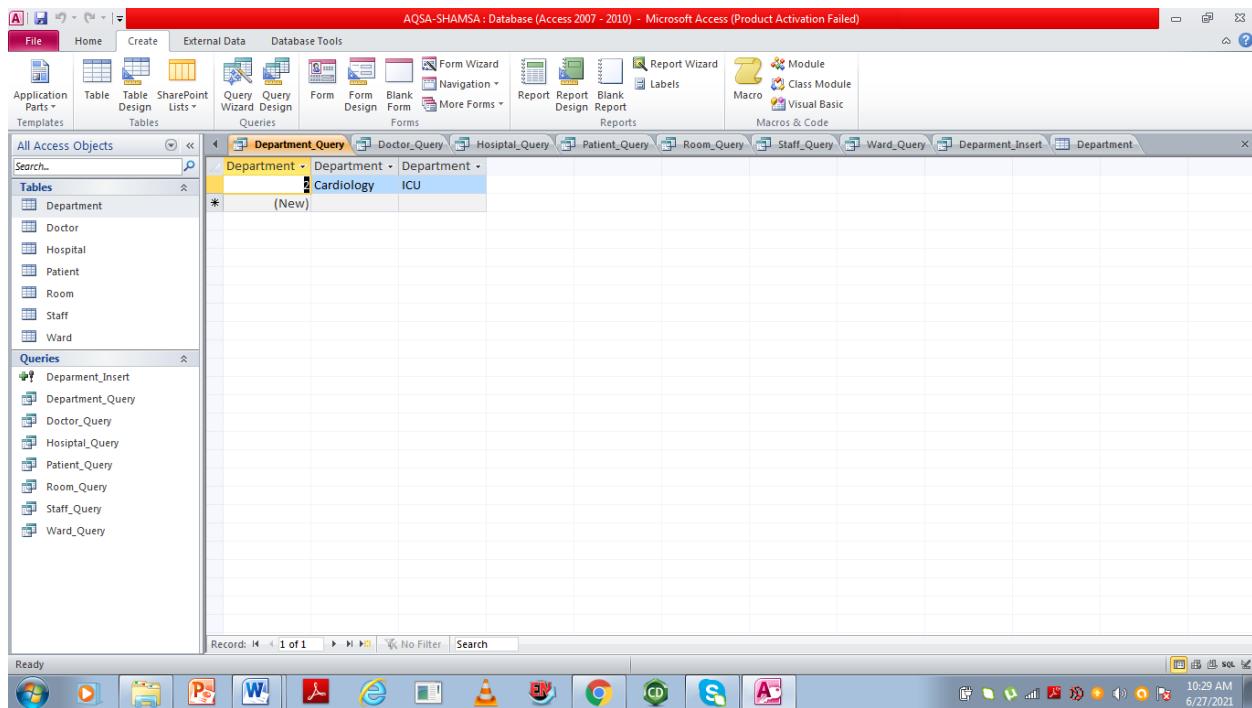
Run the program:



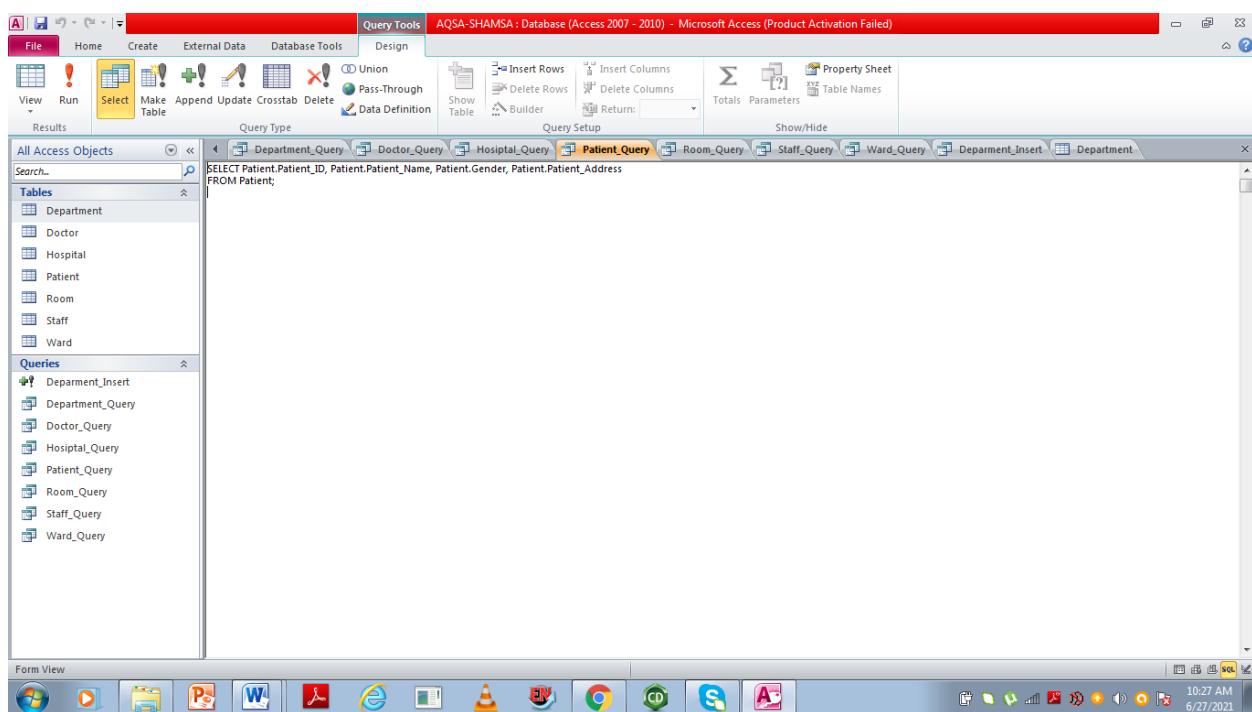
MICROSOFT ACCESS FILE:

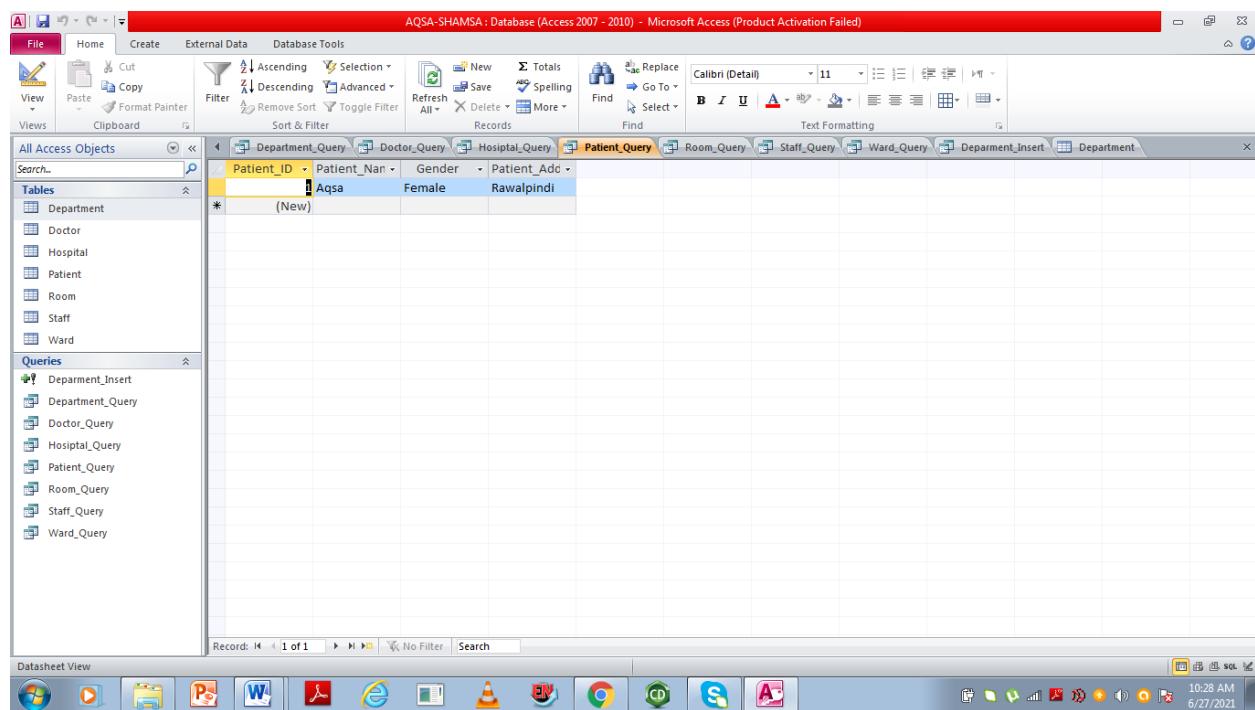
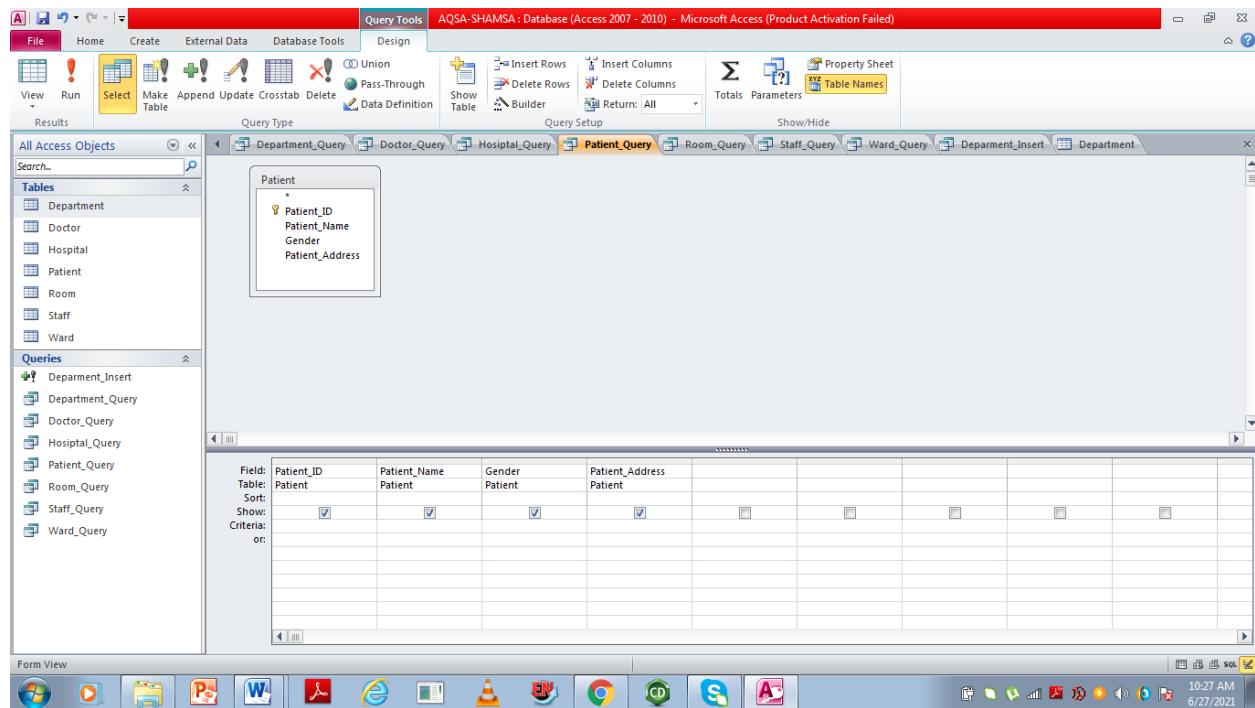
Select Query for Department in MS access:



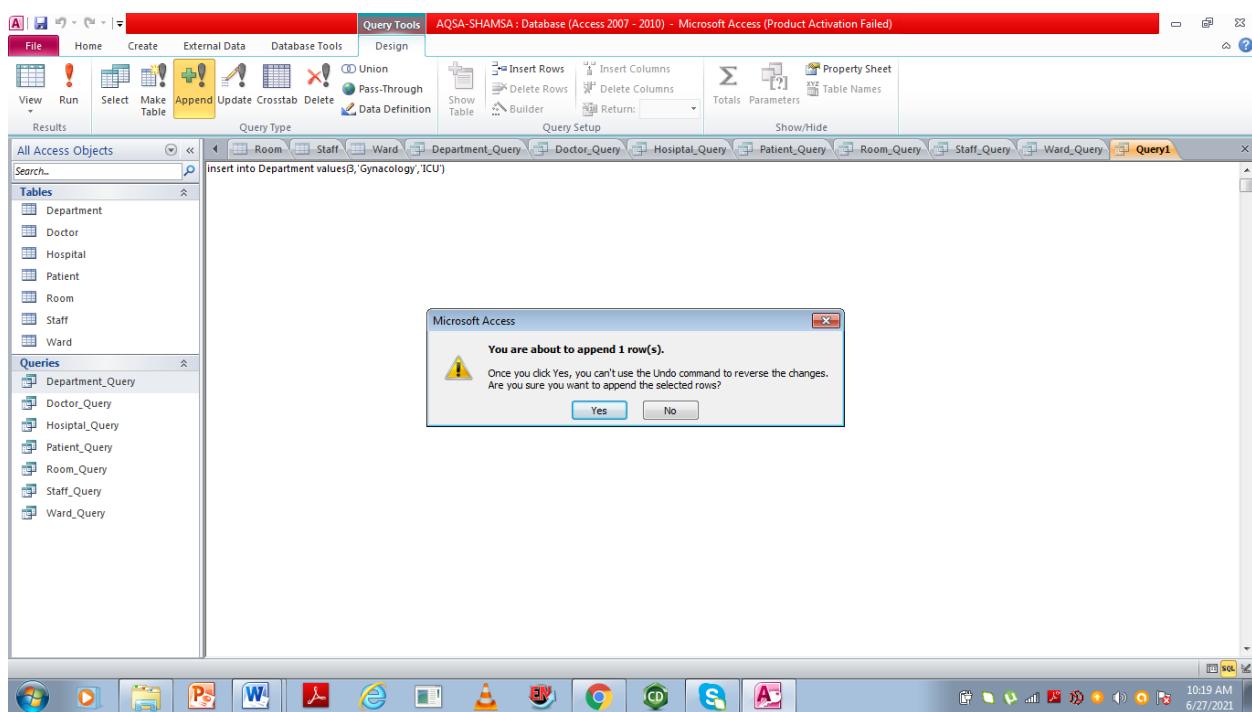
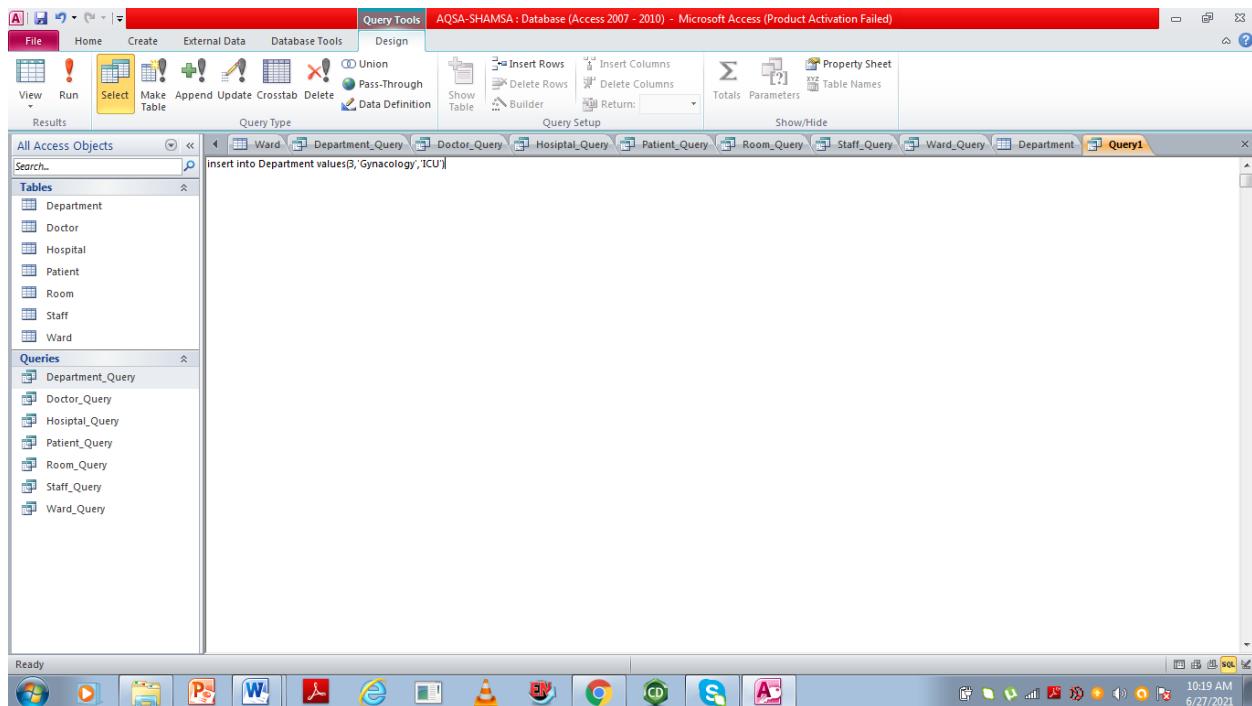


Select Query for Patient in MS access:





Insert Query for Department in MS access:



A screenshot of Microsoft Access showing the 'Table Tools' ribbon tab selected. The main area displays a table named 'Department_Query' with three records:

Department	Department	Department
1 Psychology_AC MICU		
2 Cardiology	ICU	
3 Gynecology	ICU	

The left sidebar shows the 'Tables' and 'Queries' sections. The status bar at the bottom indicates 'Record: 1 of 3'.

Delete Query For Doctor Table:

A screenshot of Microsoft Access showing the 'Query Tools' ribbon tab selected. A delete query is being run against the 'Doctor' table. A confirmation dialog box is displayed:

Microsoft Access

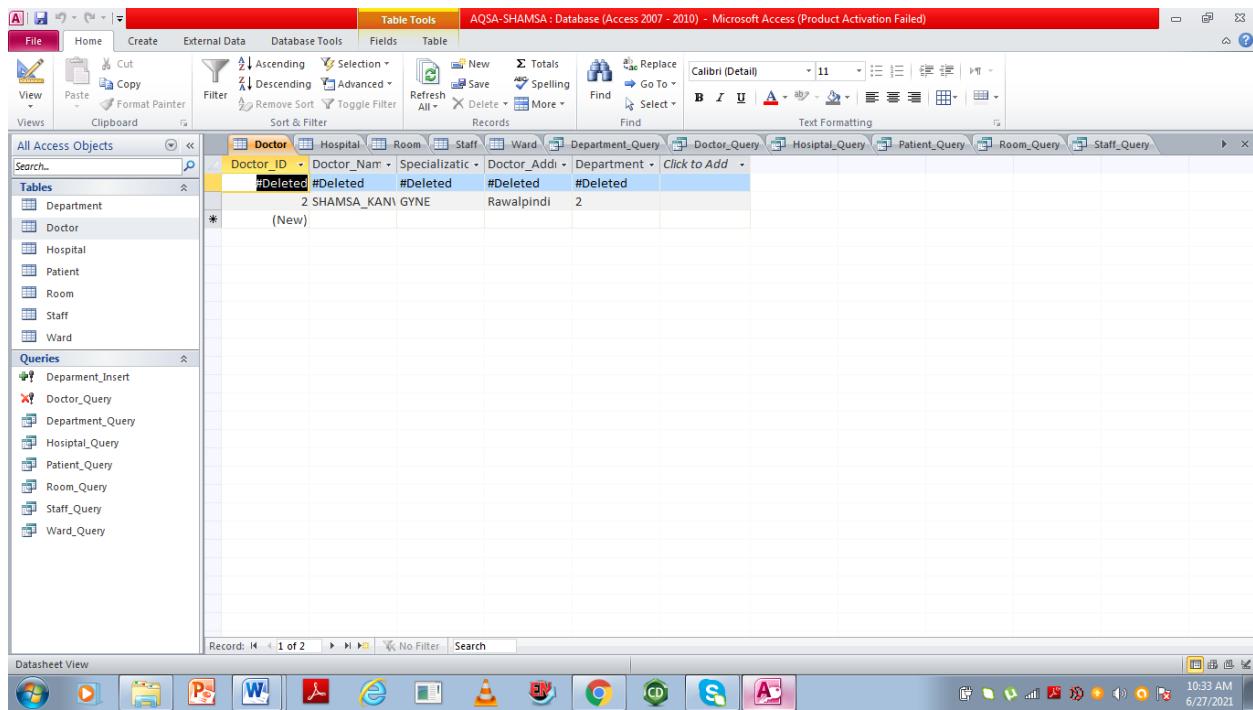
You are about to delete 1 row(s) from the specified table.
Once you click Yes, you can't use the Undo command to reverse the changes.
Are you sure you want to delete the selected records?

Buttons: Yes, No

The query builder interface shows the following SQL query:

```
Field: Doctor_ID
Table: Doctor
Delete: Doctor
Where: Where
Criteria: or:
or:
```

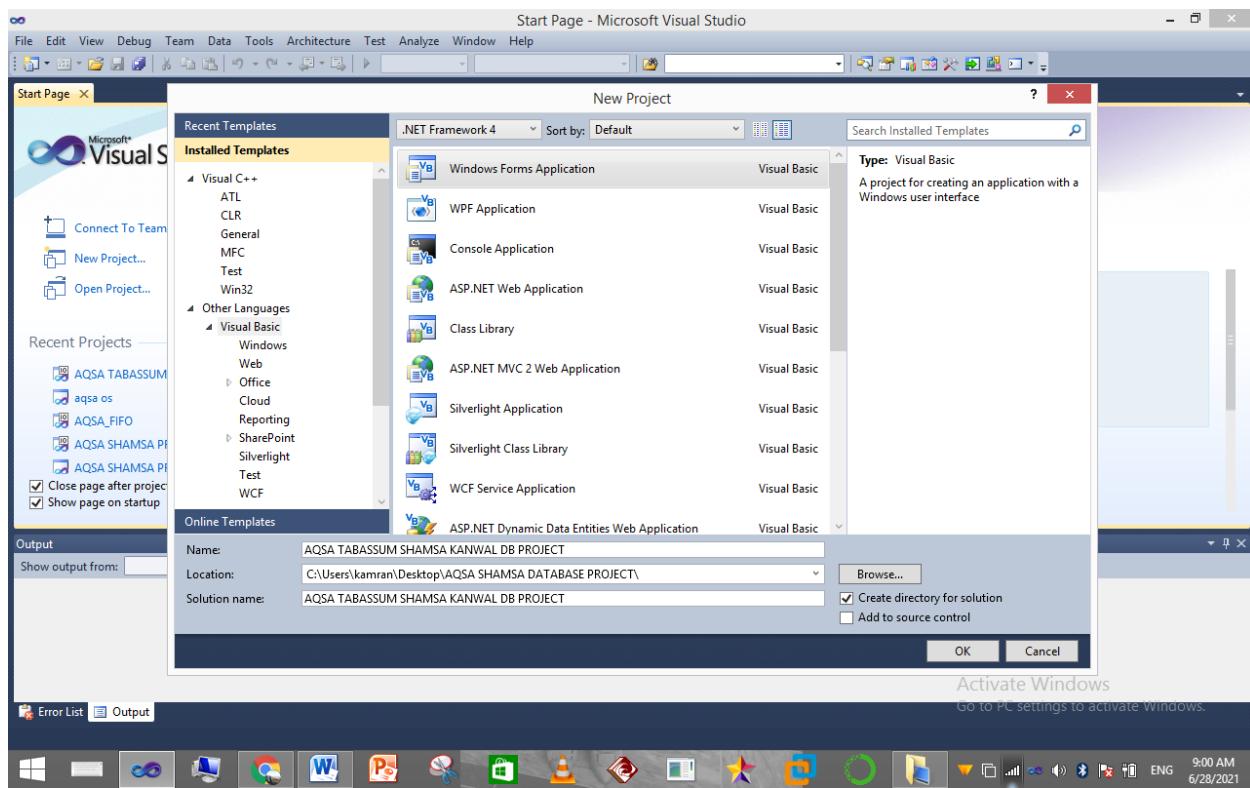
The status bar at the bottom indicates '10:33 AM 6/27/2021'.



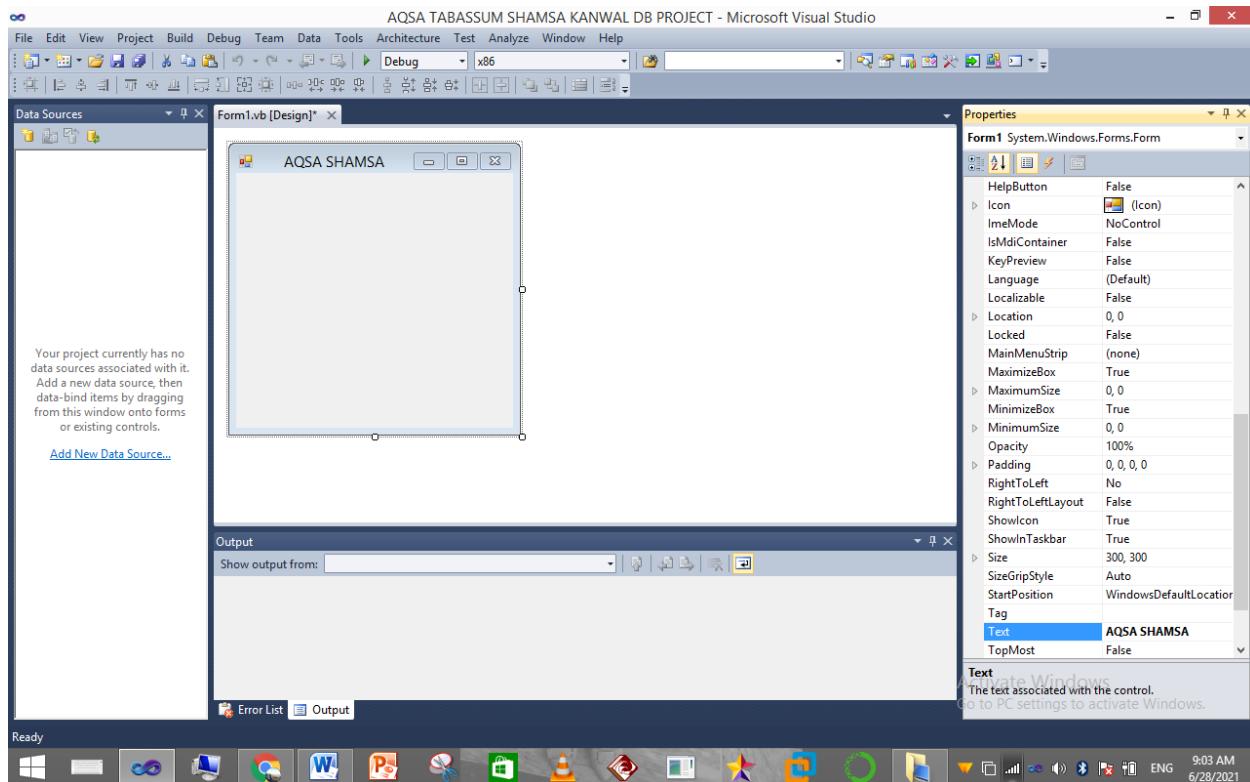
CONNECTION OF MS ACCESS WITH VISUAL STUDIO

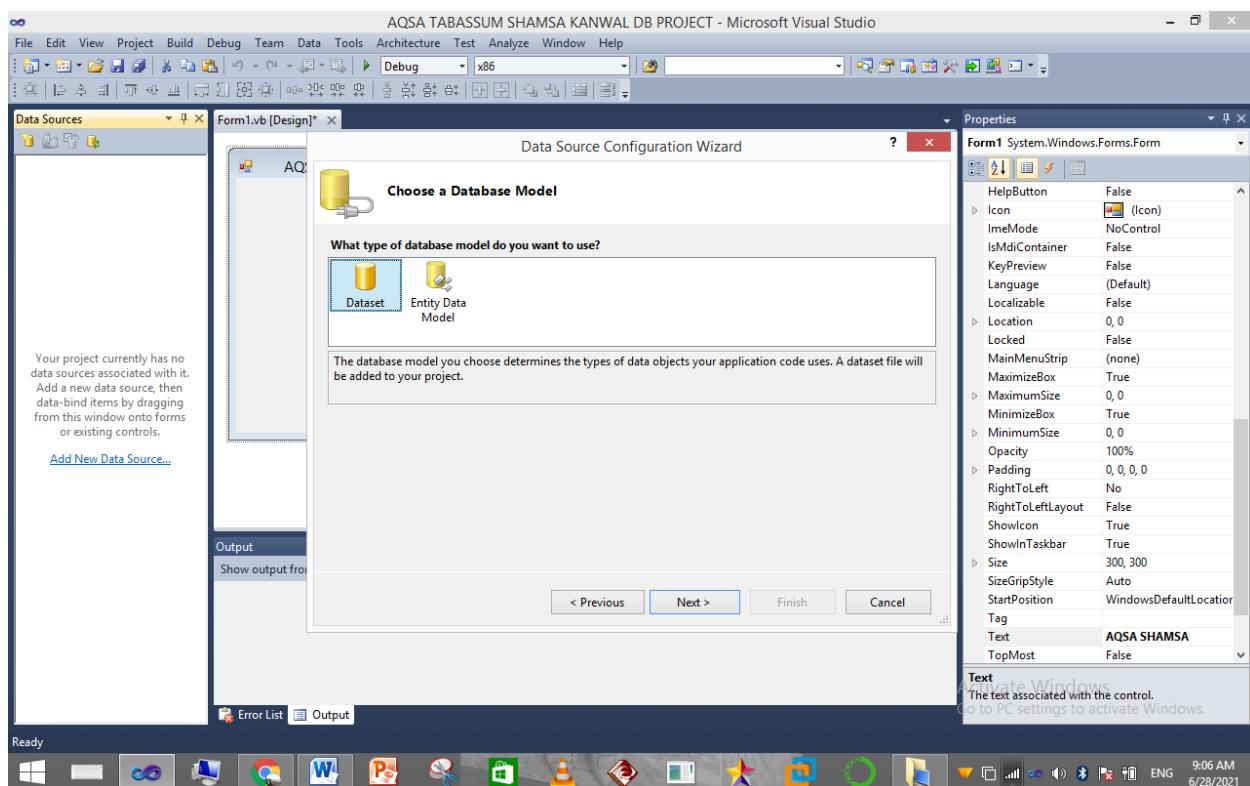
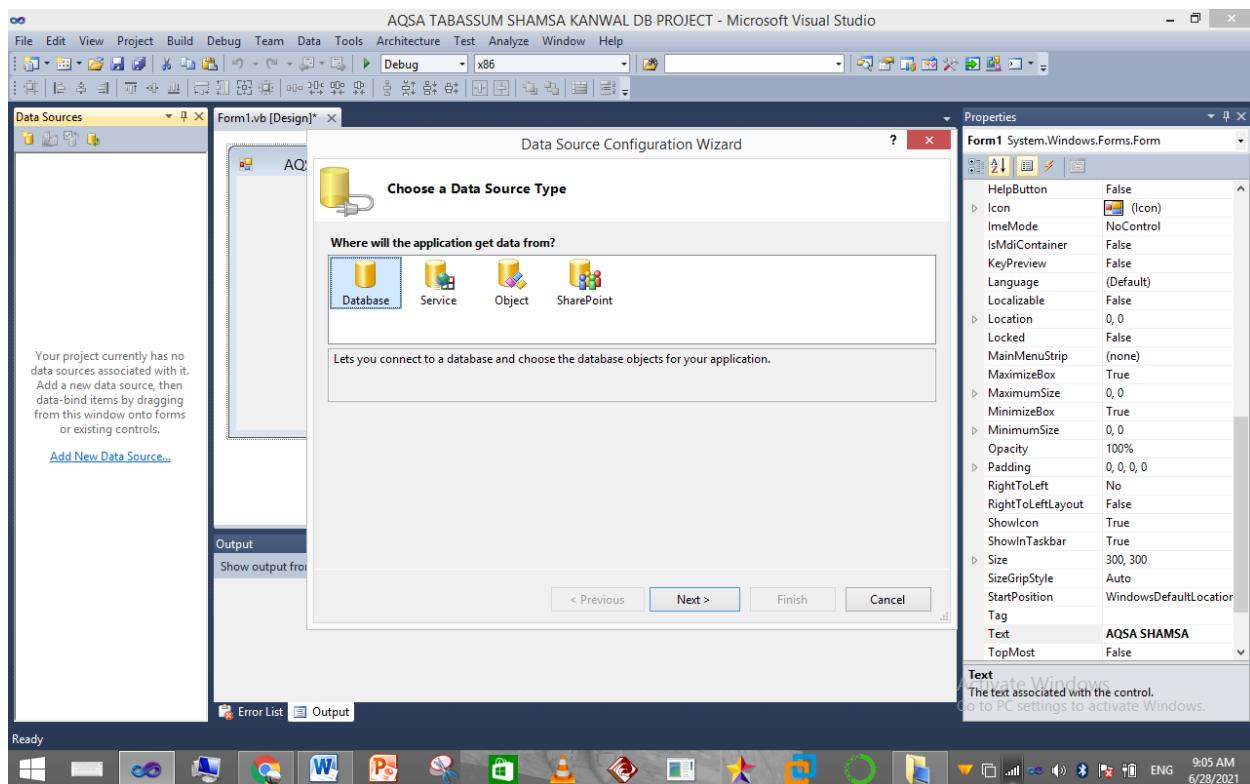
I am connecting the database file with visual studio 10 by following procedure:

Open Microsoft visual studio. Click on new project select Visual basic and Windows Form Application, name the file as AQSA SHAMSA DATABASE PROJECT and click on ok.

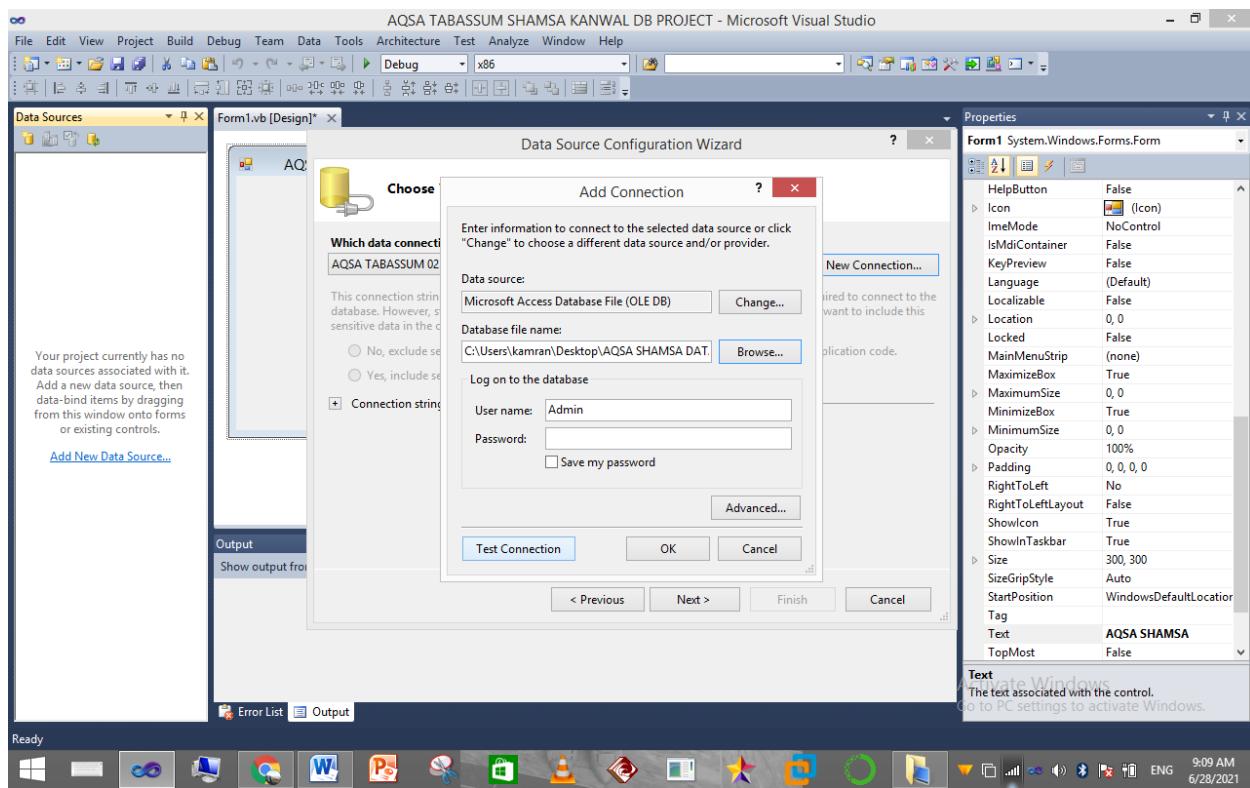


The window that appears will be, here change the name of form:

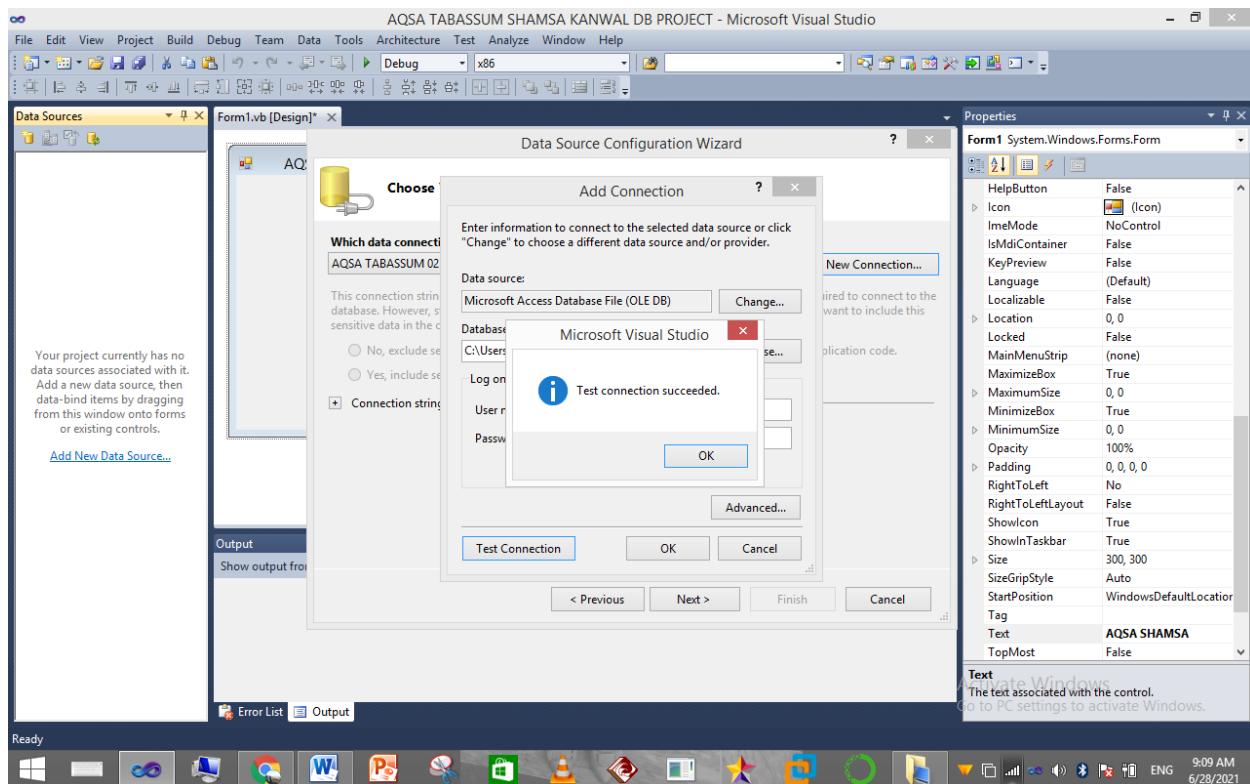


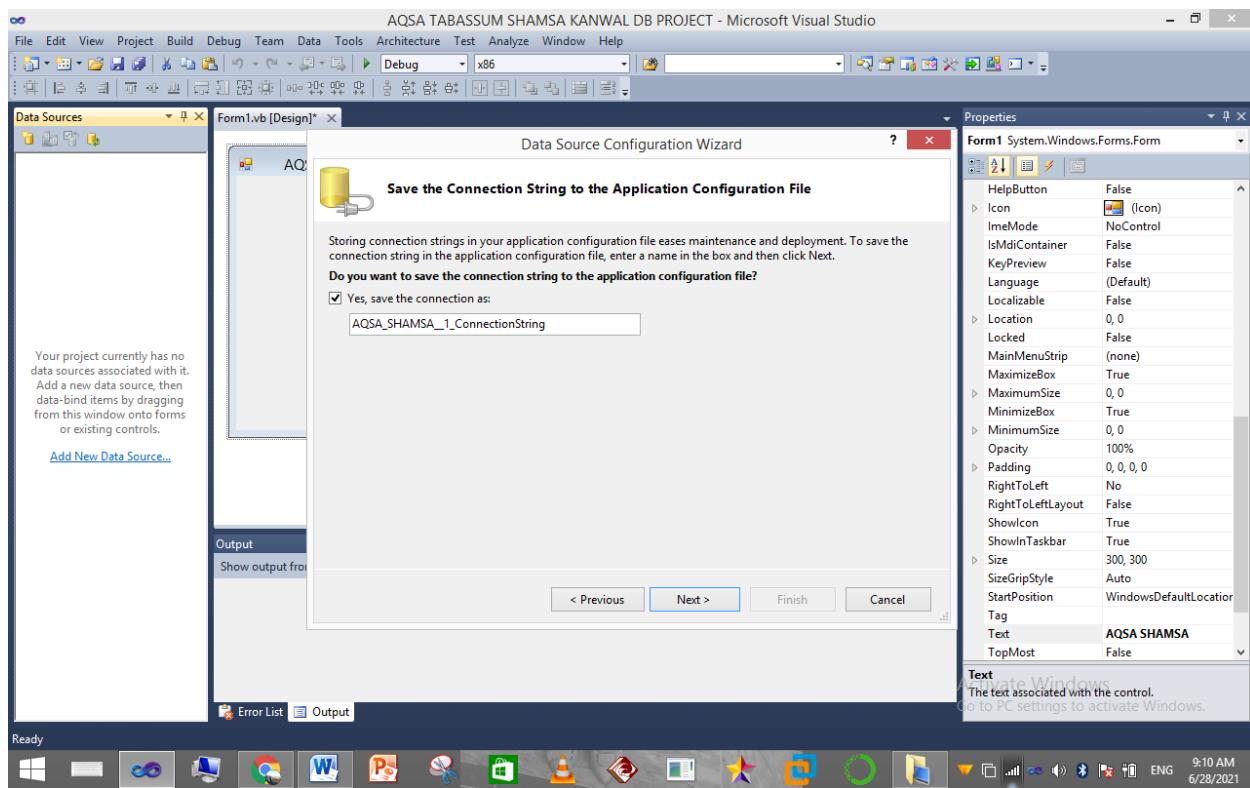


Now here you will make the connection, so select Microsoft Access Database file as the data Source and browse the file which you want to make the connection and click on test connection.

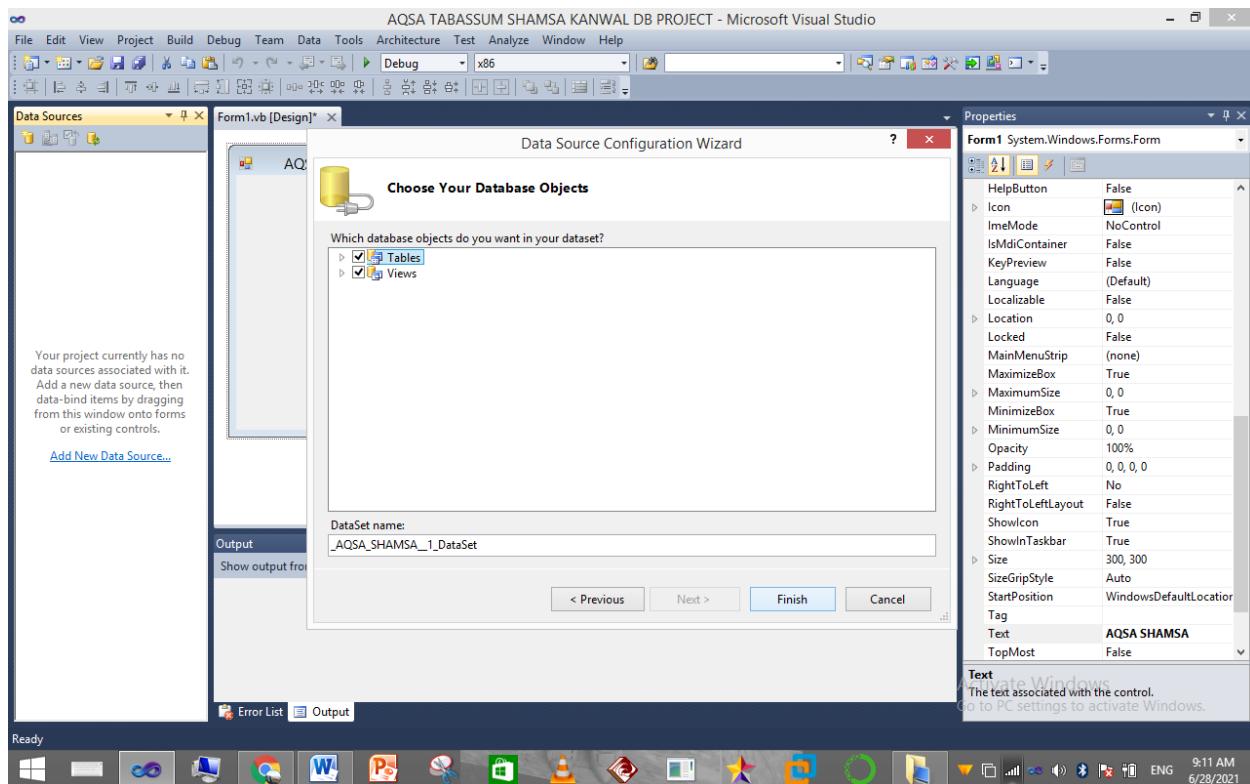


The connection is done successfully:

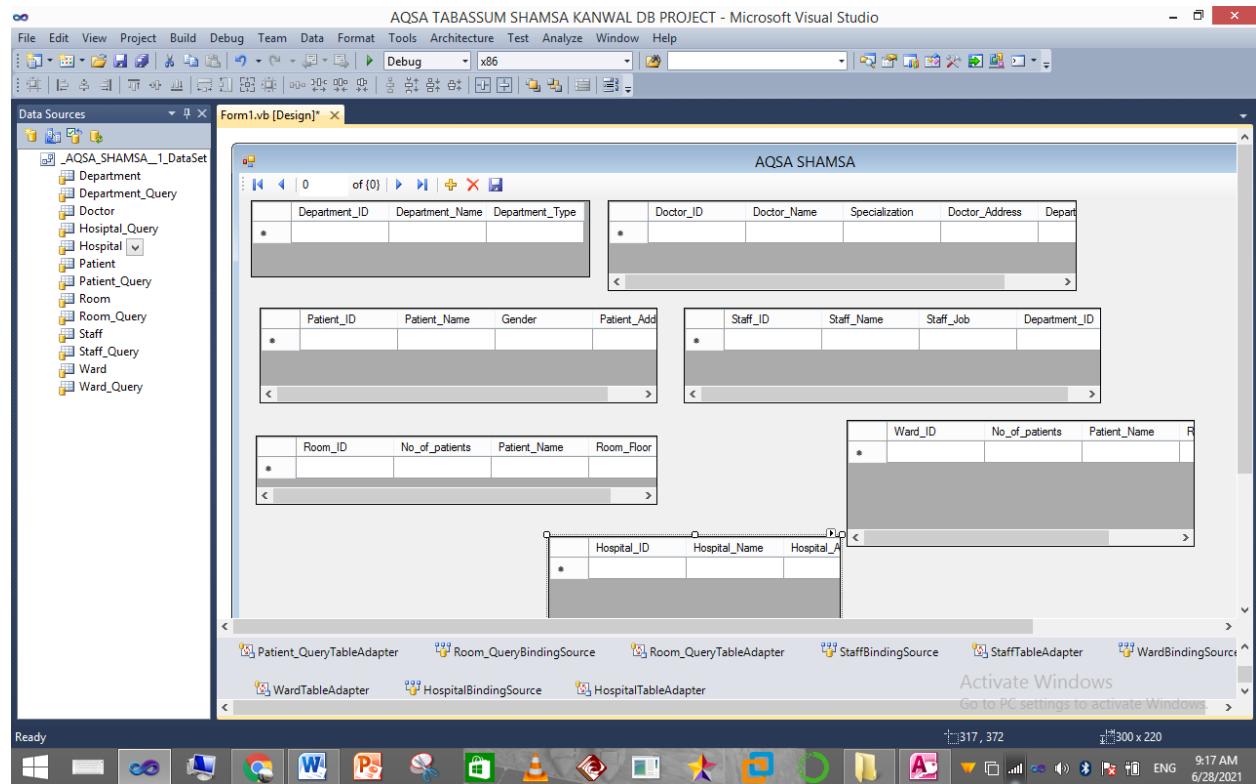




Here select both tables and views:



Following tables and views will be shown that I have created in my database file. Drag these tables to the Forms:



Now execute the form, all of the data of created queries and table will be shown:

