**Code for creating table**

CREATE TABLE Child

(

ChildNo tinyint PRIMARY KEY Identity (1,1),

FirstName varchar (20),

LastName varchar (20),

Address varchar (30),

MedicalCondition varchar (20),

Allergy varchar (20),

DateOfBirth Date,

Fees smallmoney,

DueFees char (1) CHECK ( DueFees IN ('Y', 'N')),

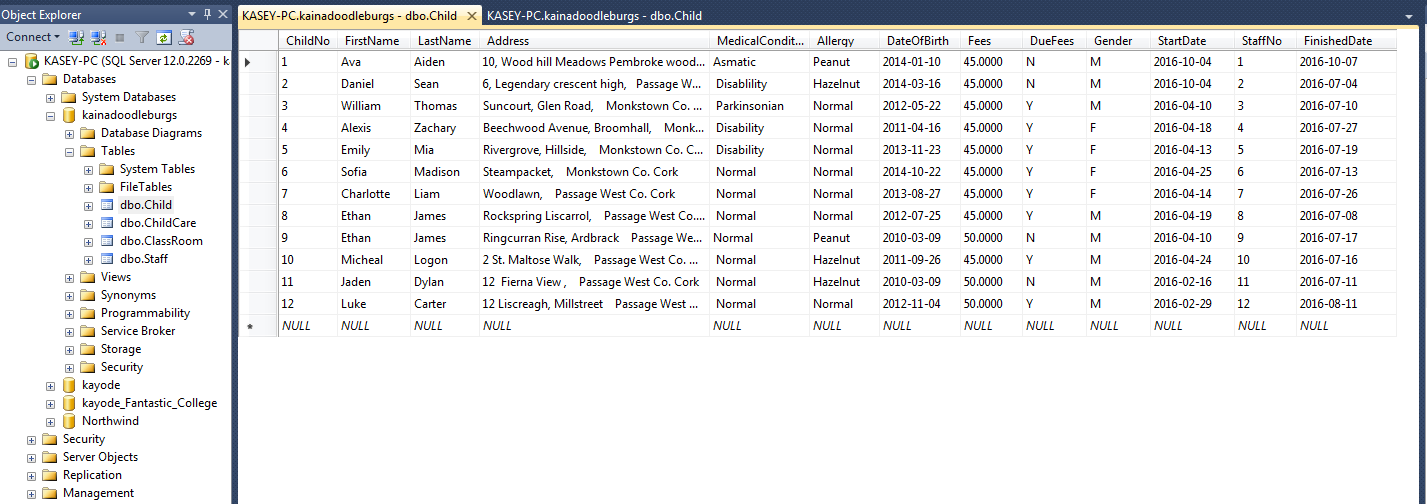
Gender char (1) CHECK (Gender IN ('F', 'M' )),

StartDate Date ,

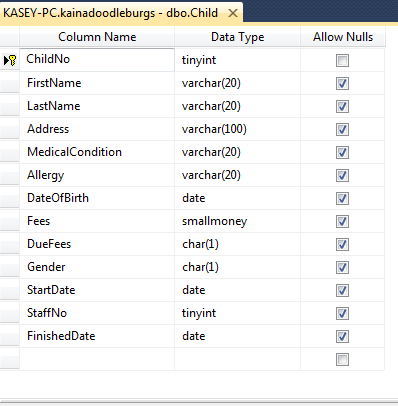
FinishedDate Date

)

**Result of table**

****

**After change**



**Changes and reason**

I decided to make changes because address space is too small for some addresses then I increase it to varchar (100) and also add staff Id as tinyint to link relationship between child and staff table.

**Code for creating table**

CREATE TABLE Staff

(

StaffNo tinyint PRIMARY KEY NOT NULL,

FirstName varchar(20),

LastName varchar(20),

Address varchar(30),

Shift varchar(9),

Qualification varchar(30),

JobTitle varchar(15),

Phone varchar(10),

WorkHours tinyint,

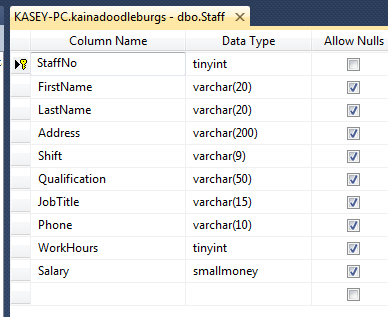
Salary smallmoney,

)

**Result of table**



**After change**



**Changes and reason**

I decided to make changes because address space is too small for some addresses then I increase it to varchar (200)

**Code for creating table**

CREATE TABLE ChildCare

(

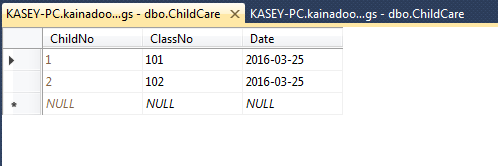
ChildNo tinyint Identity (1,1),

ClassNo tinyint Identity (1,1),

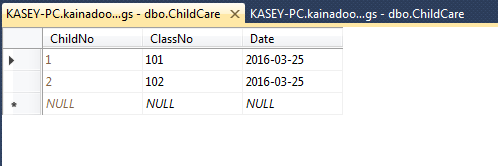
CONSTRAINT PK\_ChildCare PRIMARY KEY (ChildNo, ClassNo)

)

**Result of table**



**After change**



**Changes and reason**

I decided to add date to track when each child added to the childcare table

**Code for creating table**

CREATE TABLE ClassRoom

(

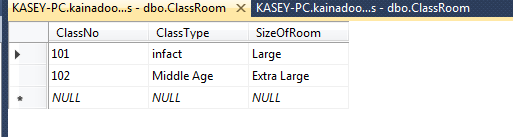
ClassNo tinyint PRIMARY KEY,

ClassType varchar (15),

SizeOfRoom varchar (15)

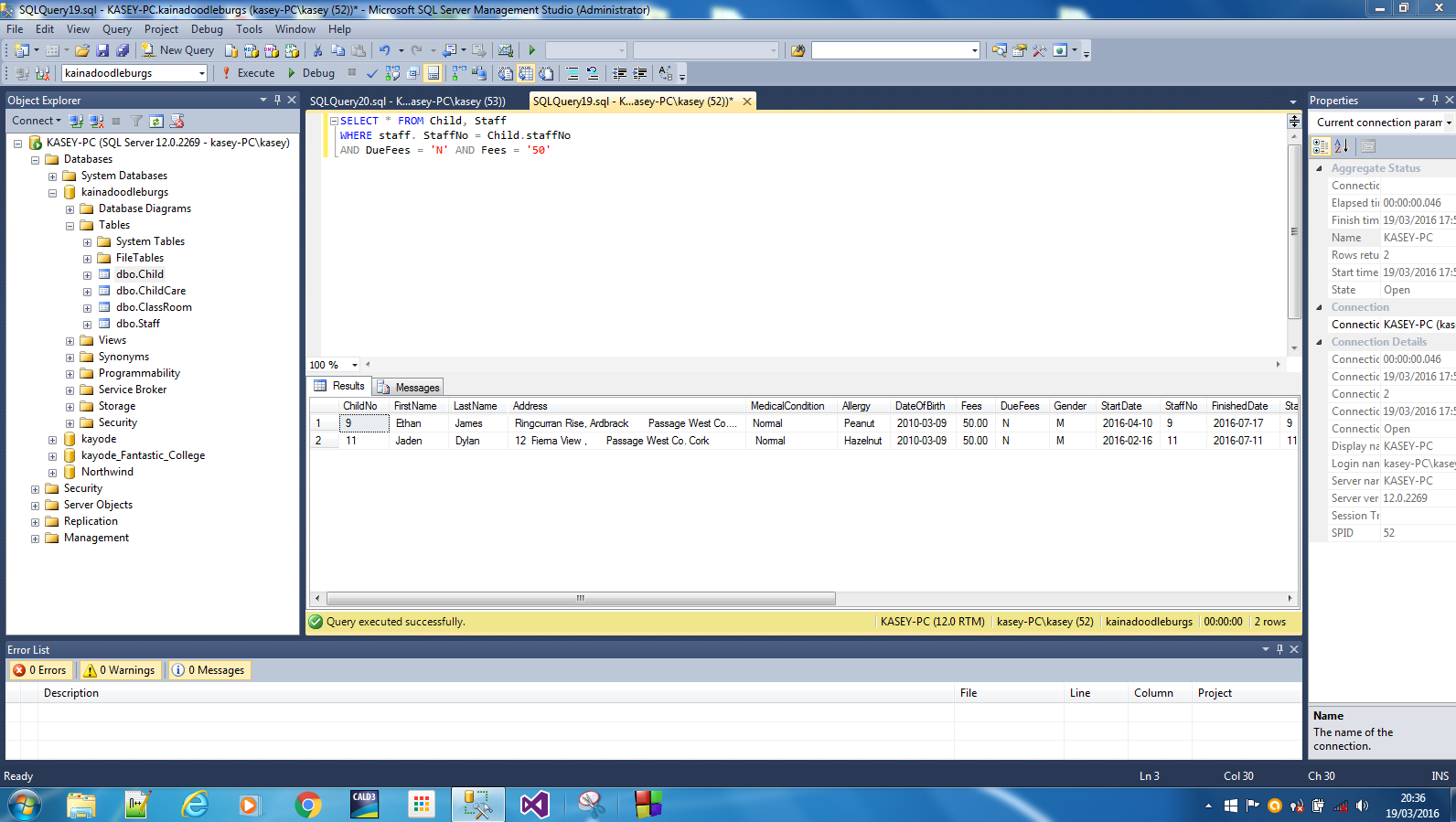
)

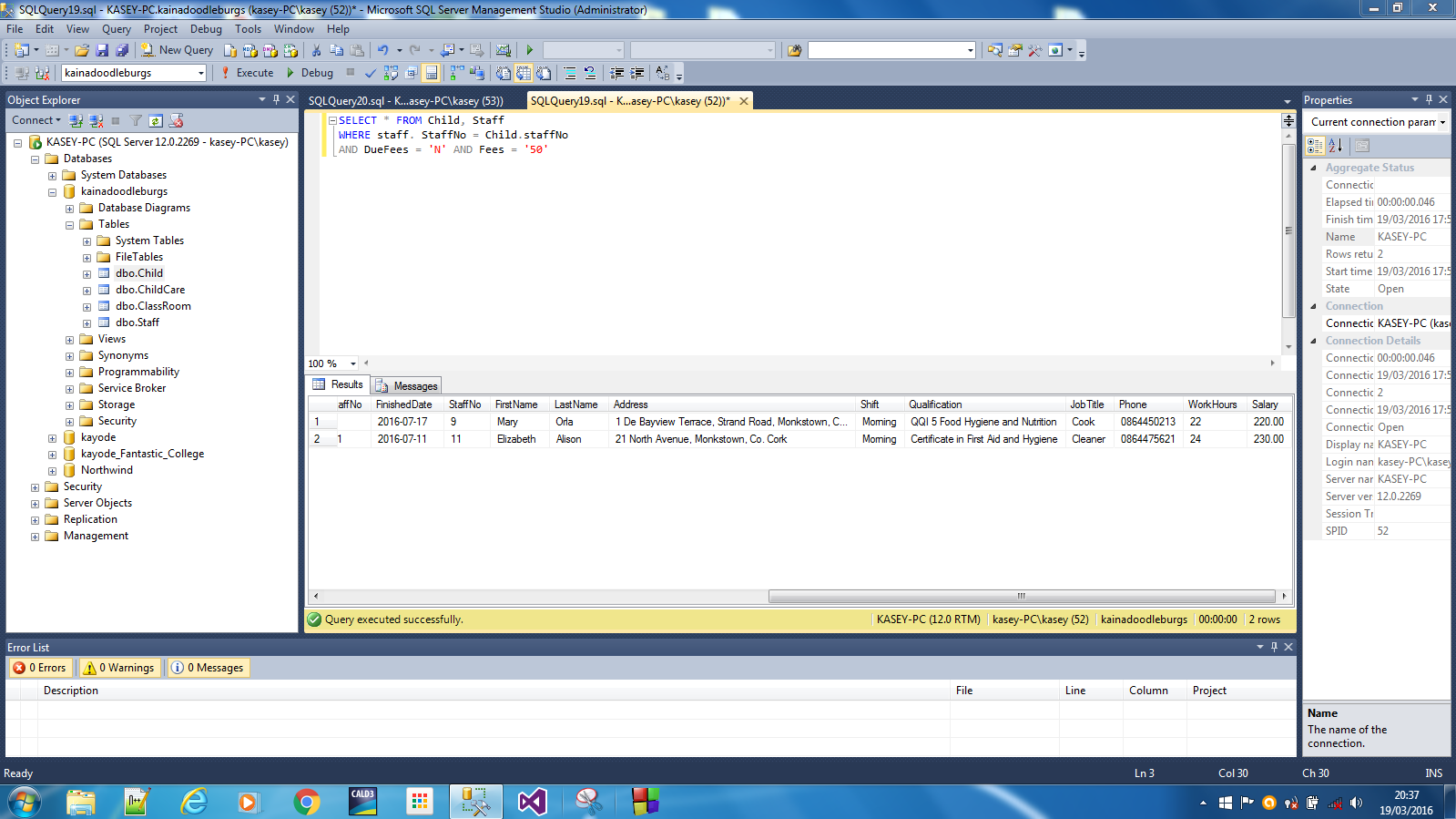
**Result of table**



**Queries 1**

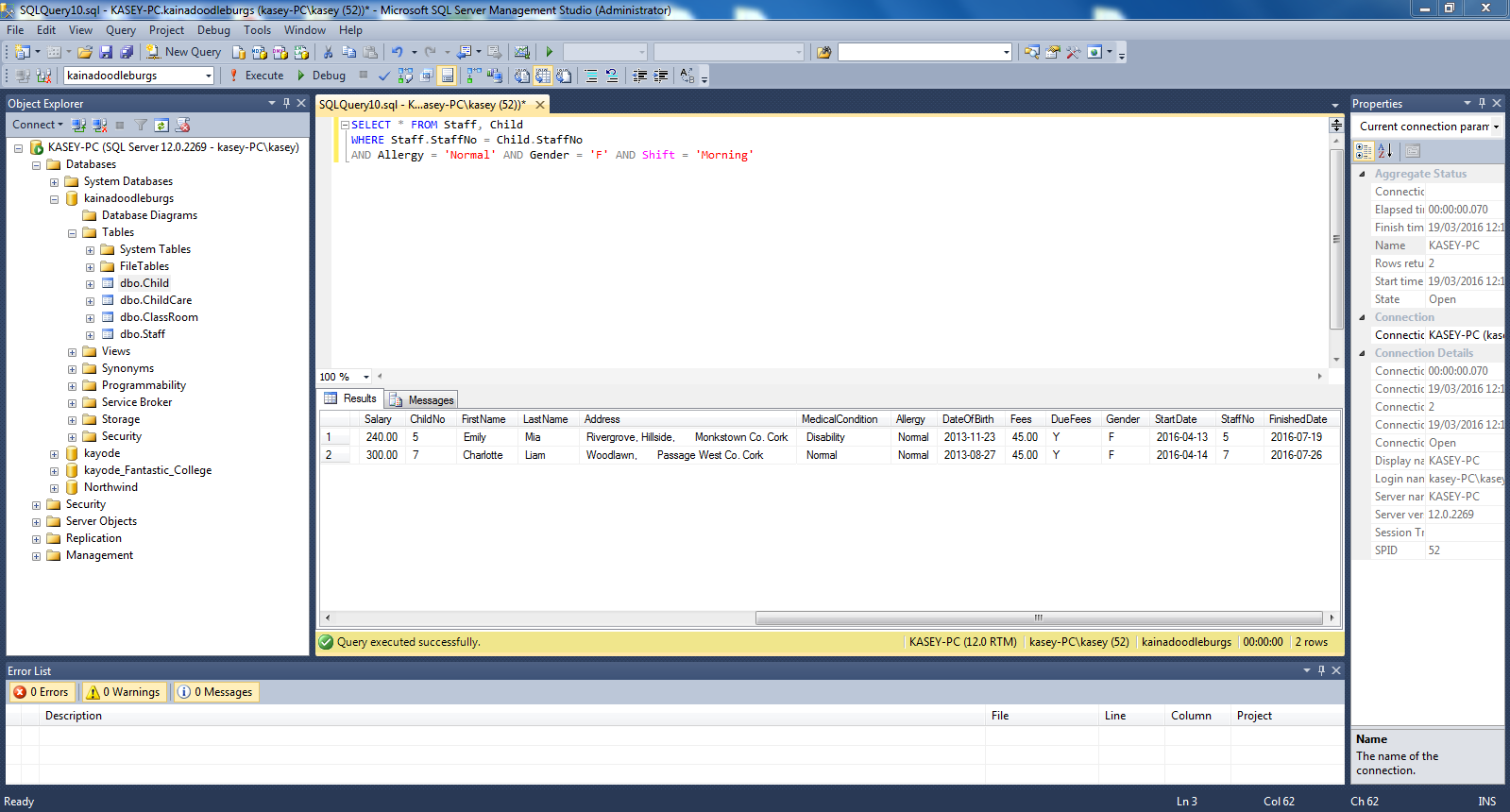
**Joint two table to show fees not pay and fees**

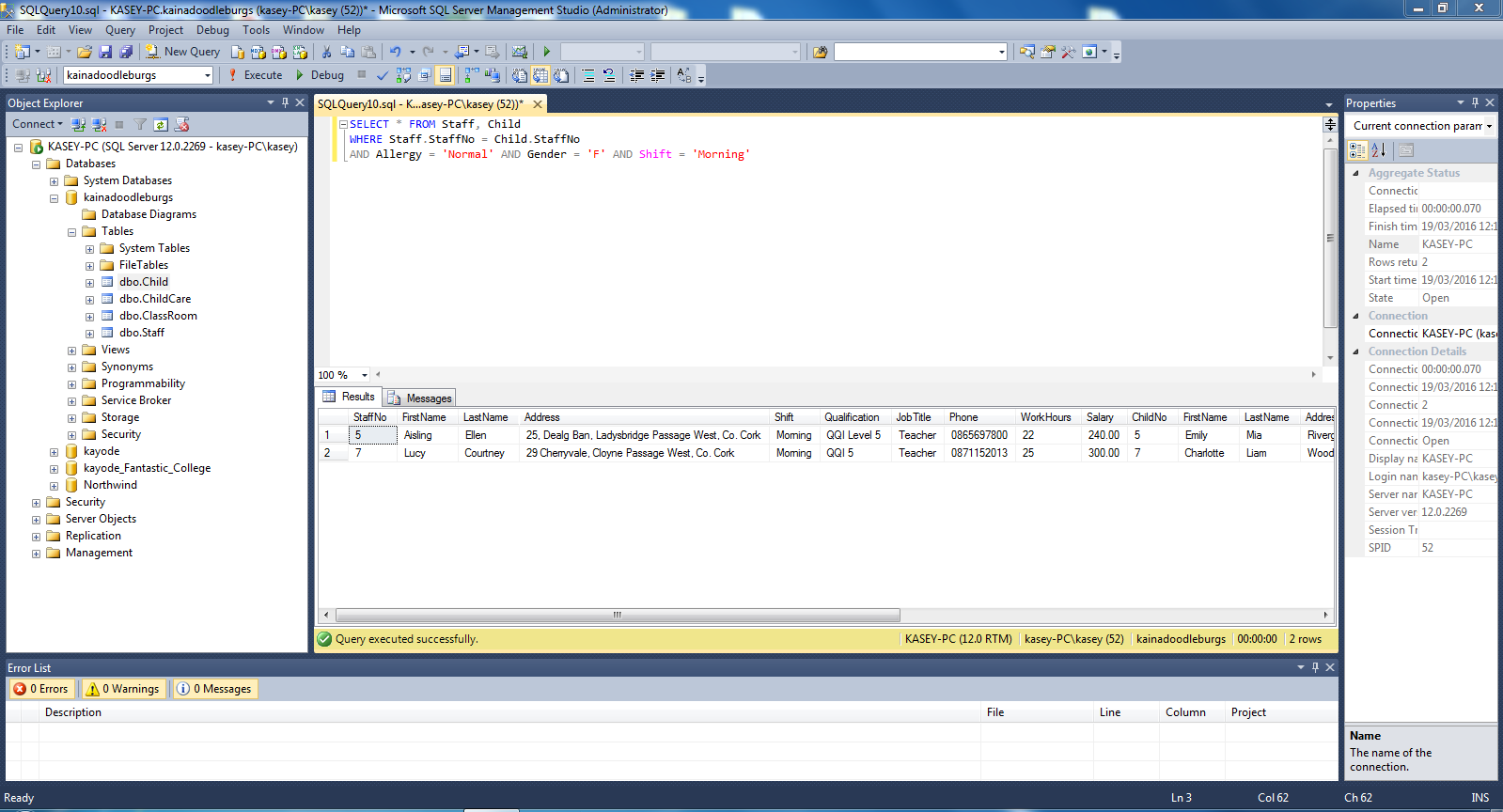
****



**Queries 2**

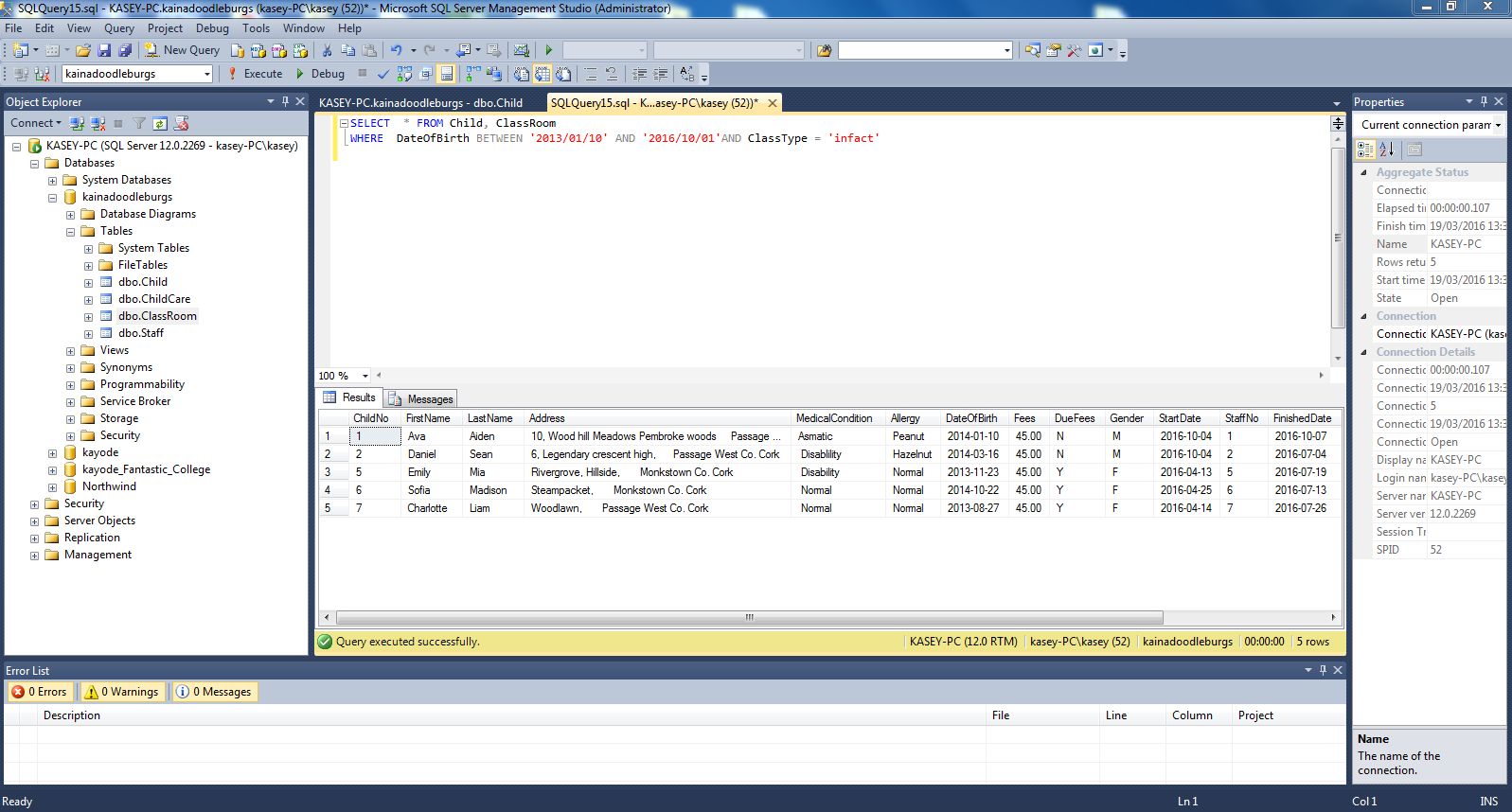
**Joint two table to show all allergy children, gender and staff working in morning shift**





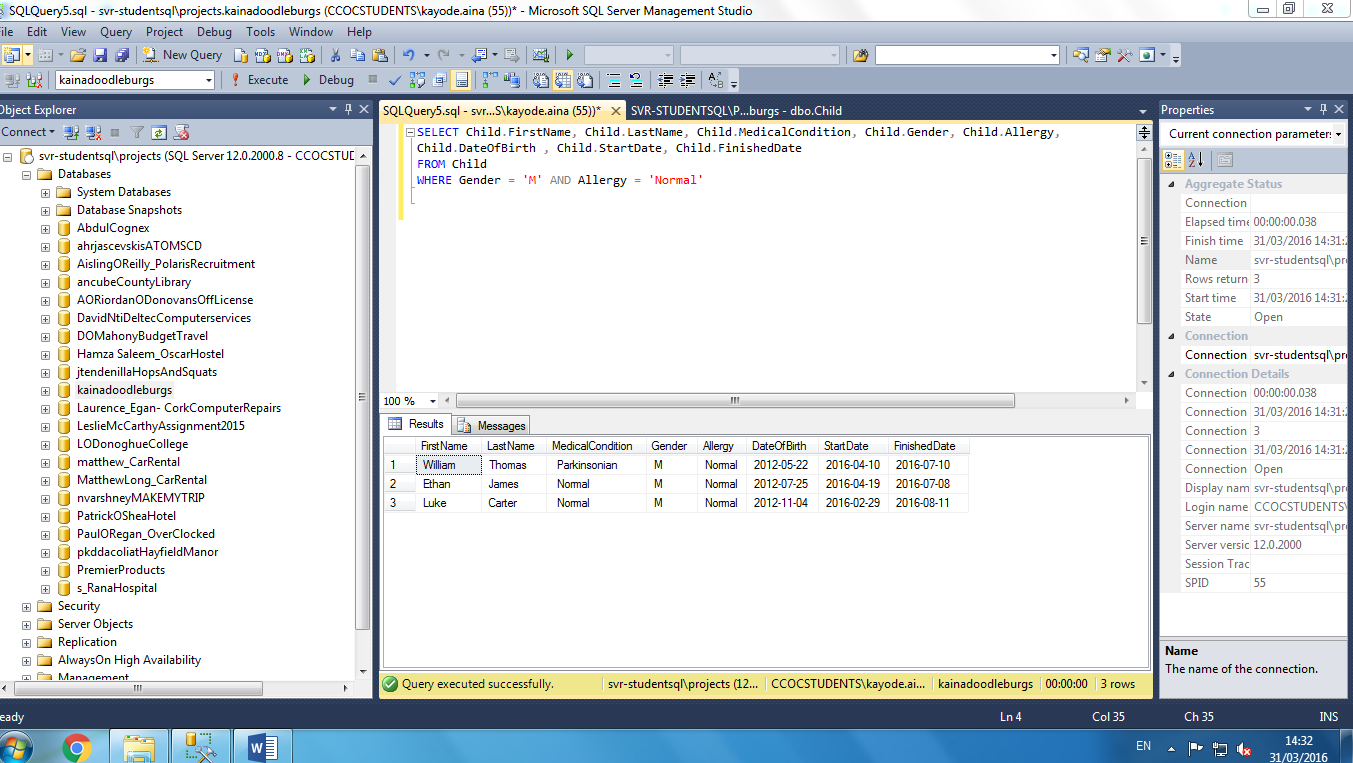
**Queries 3**

**Joint two table to show all children that are less four years and their class type**



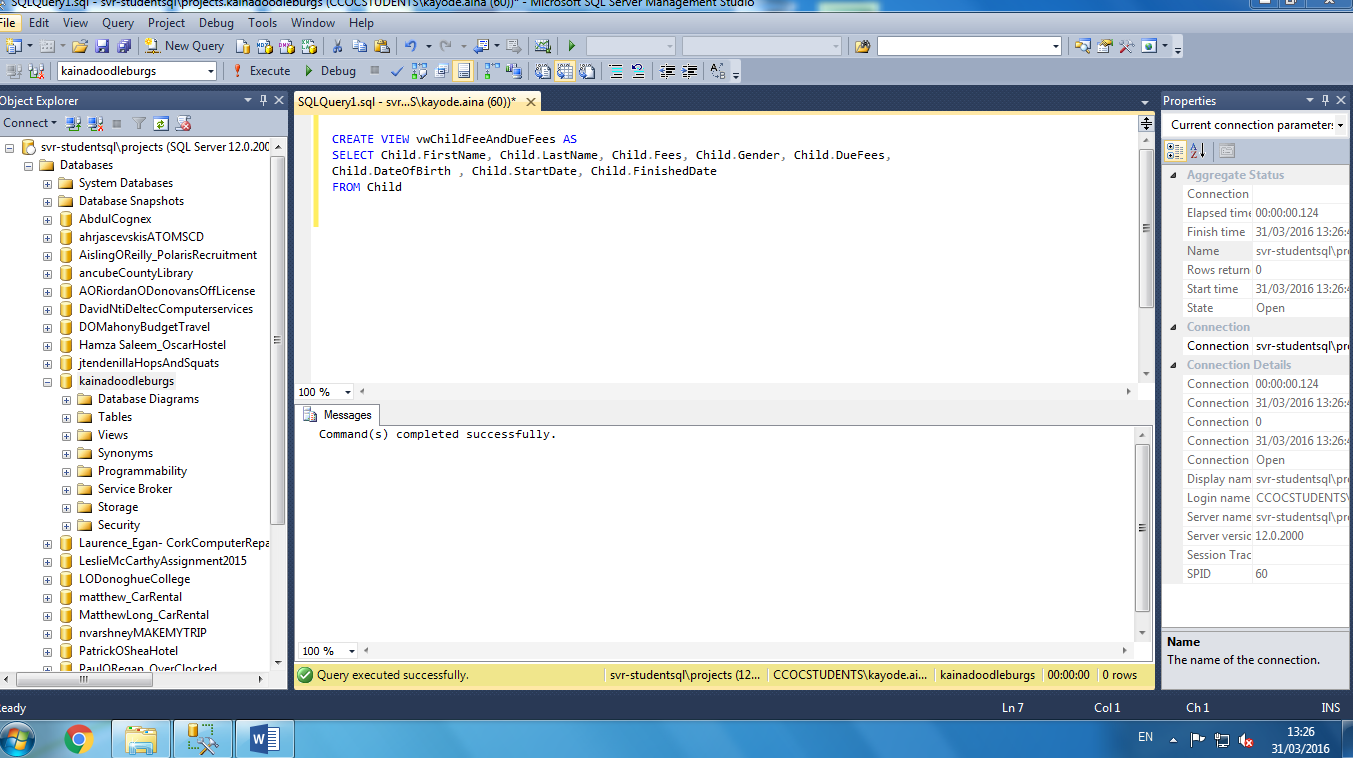
**Queries 4**

**All male children without allergy**

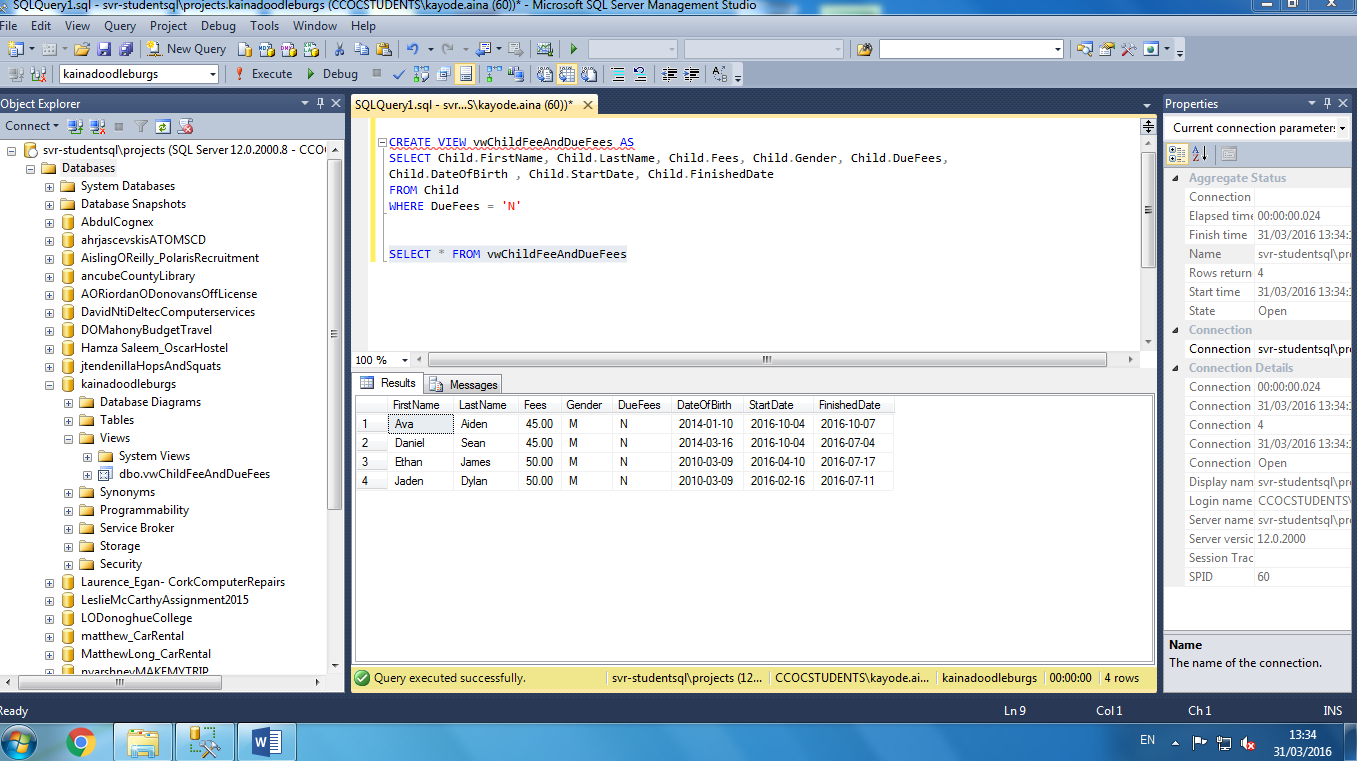


**First view**

**Creating Views**

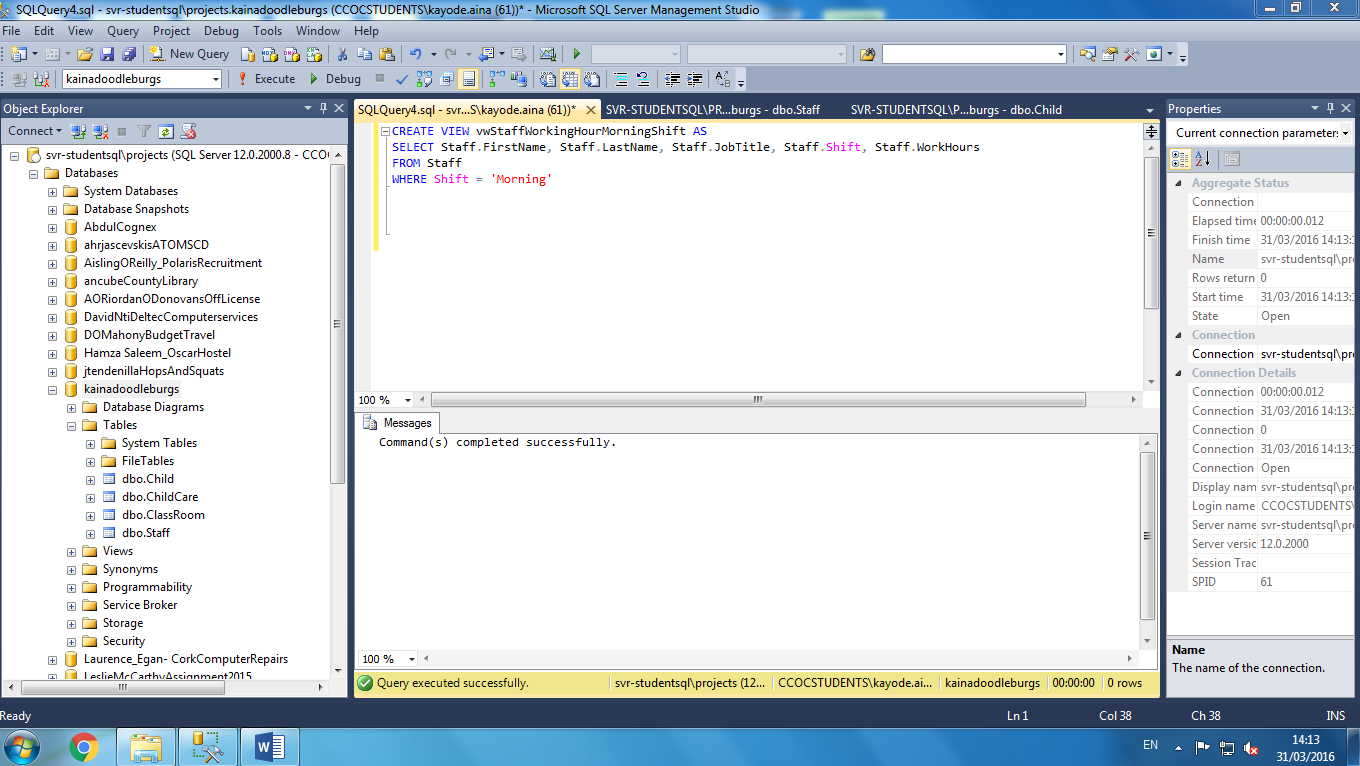


**Views result on non-pay fees with their basic information**

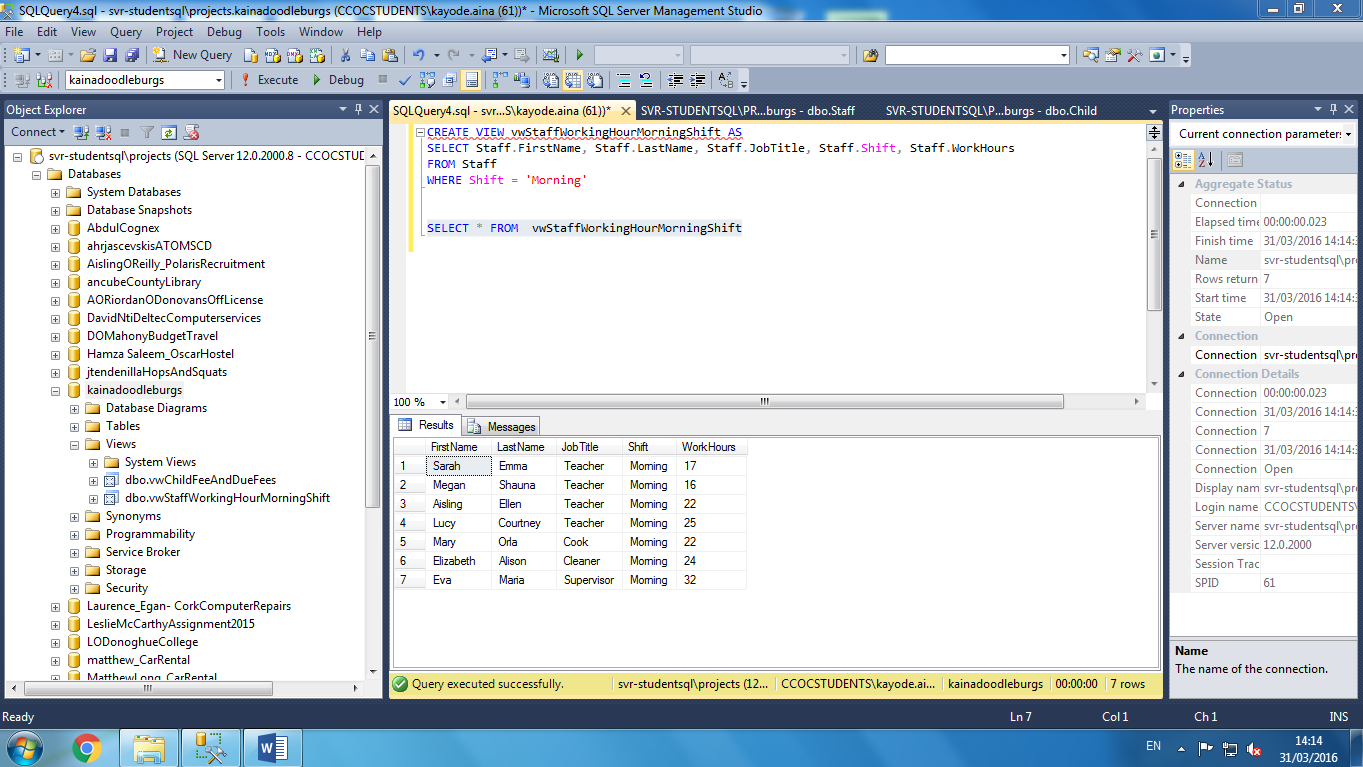


**Second view**

**Creating Views**

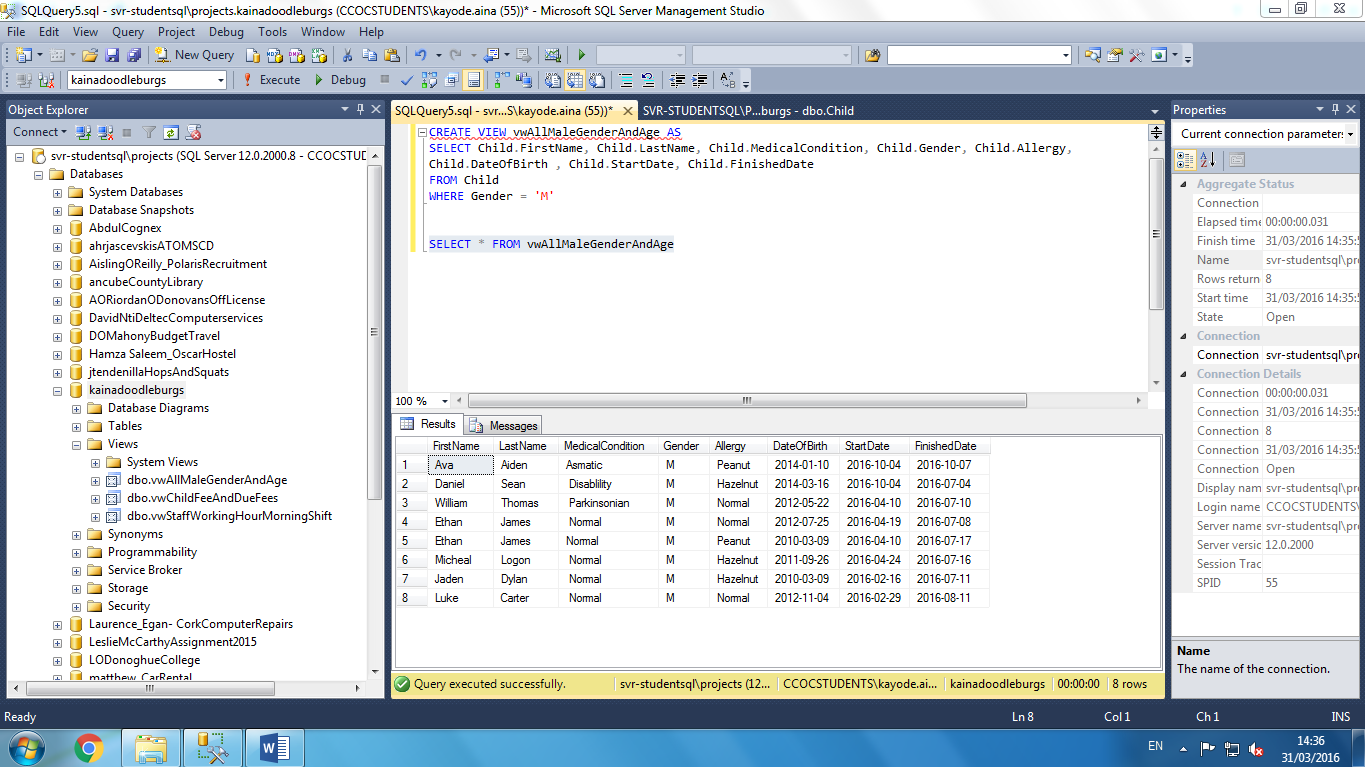


**Views result on basic information of morning shift staff**



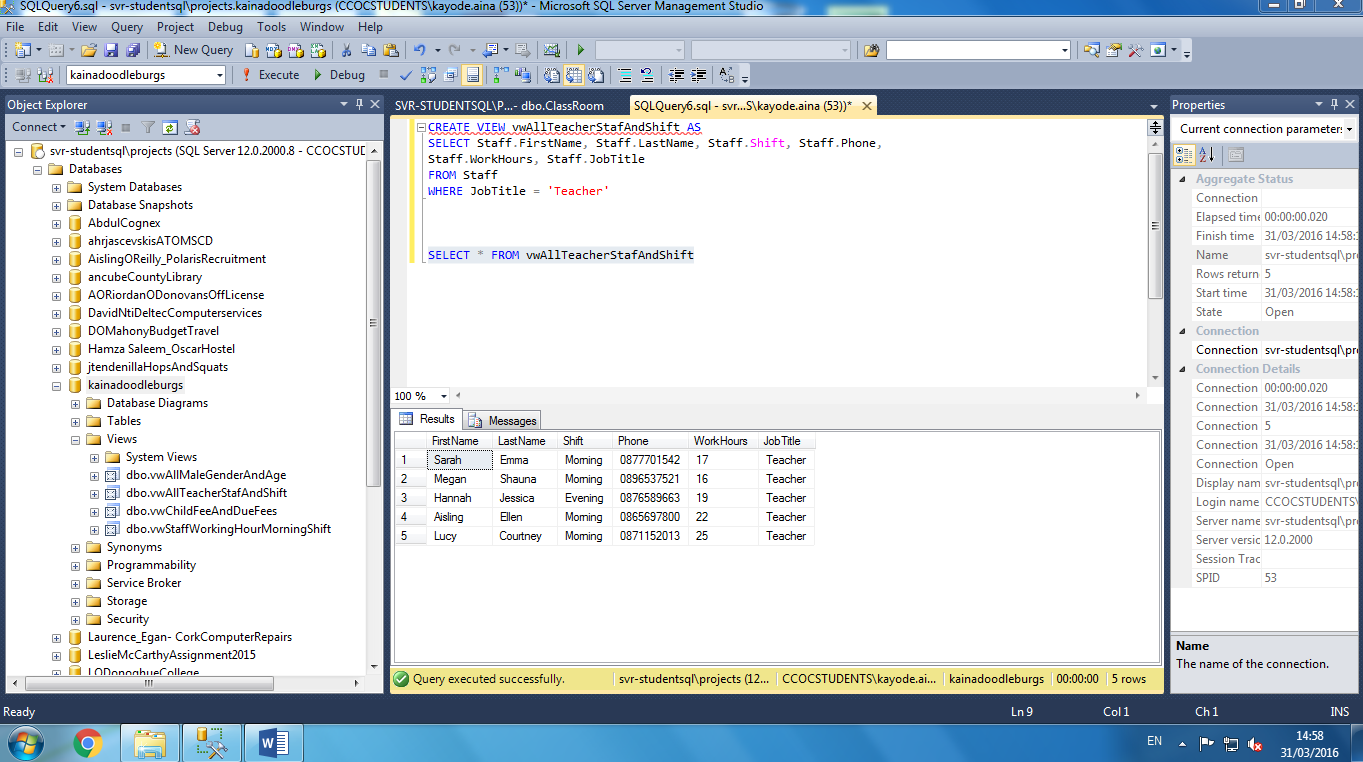
**Third view**

**Creating views result on male child and age**

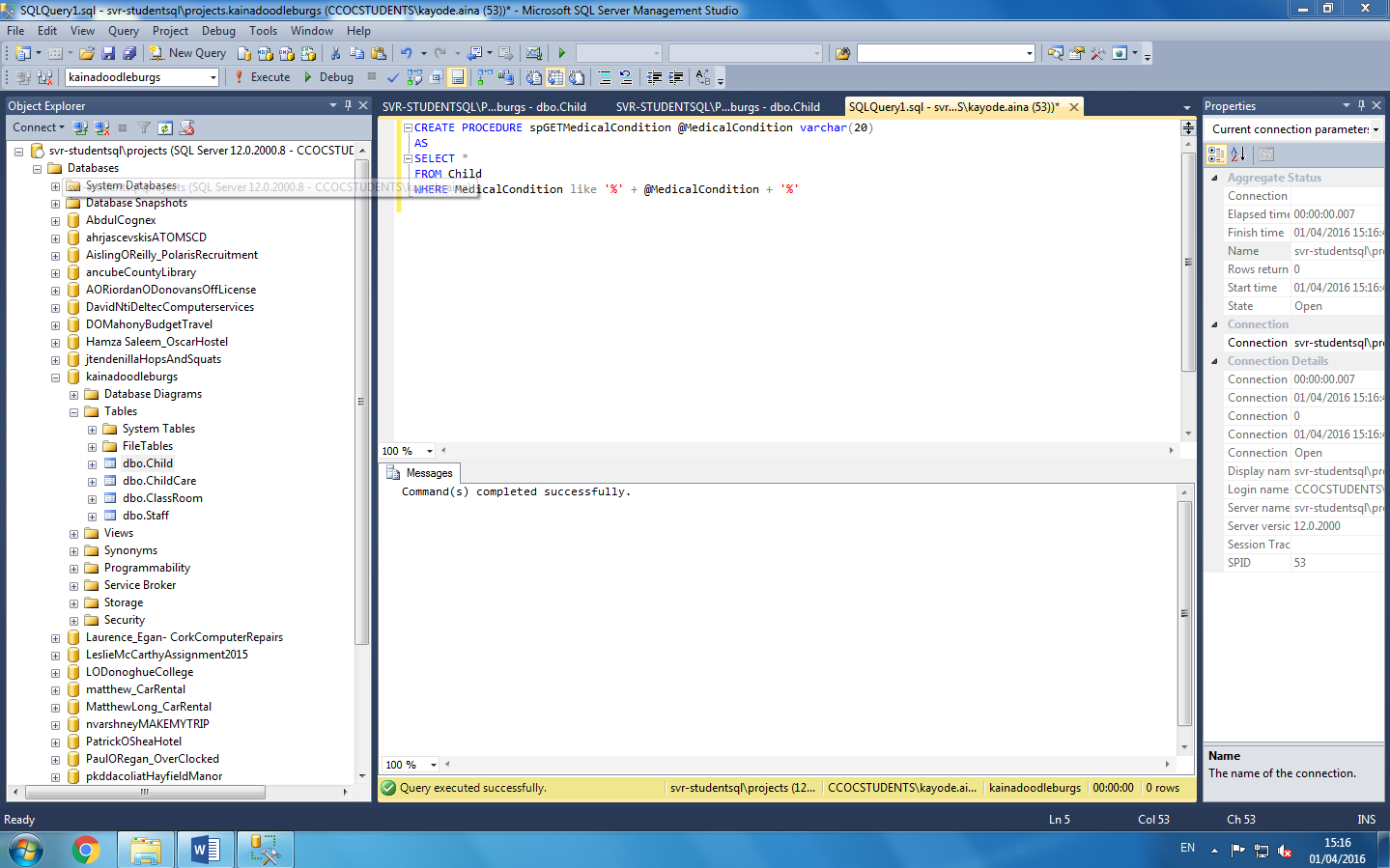


**Fourth view**

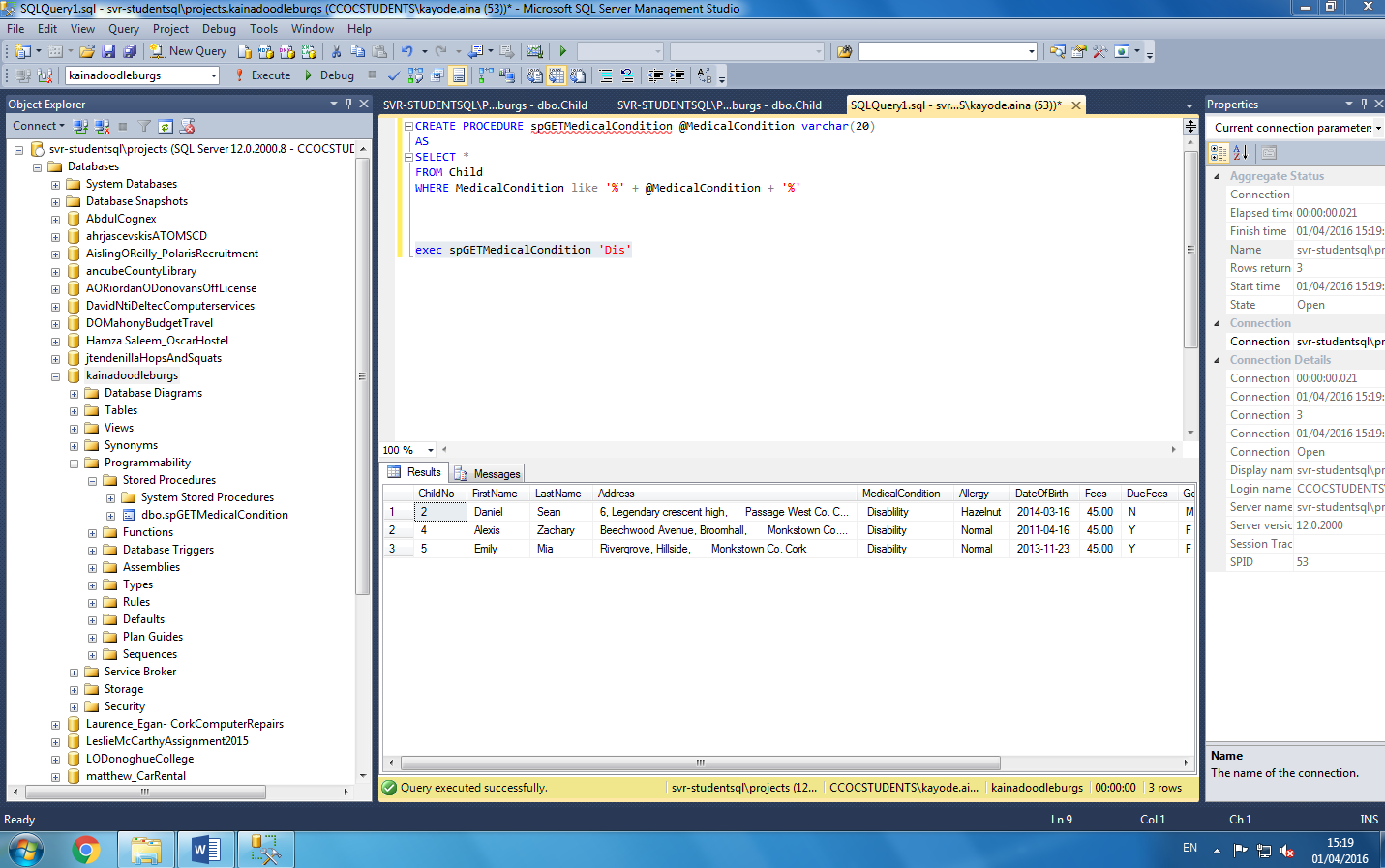
**Creating views and result to get basic information on numbers of the teacher**



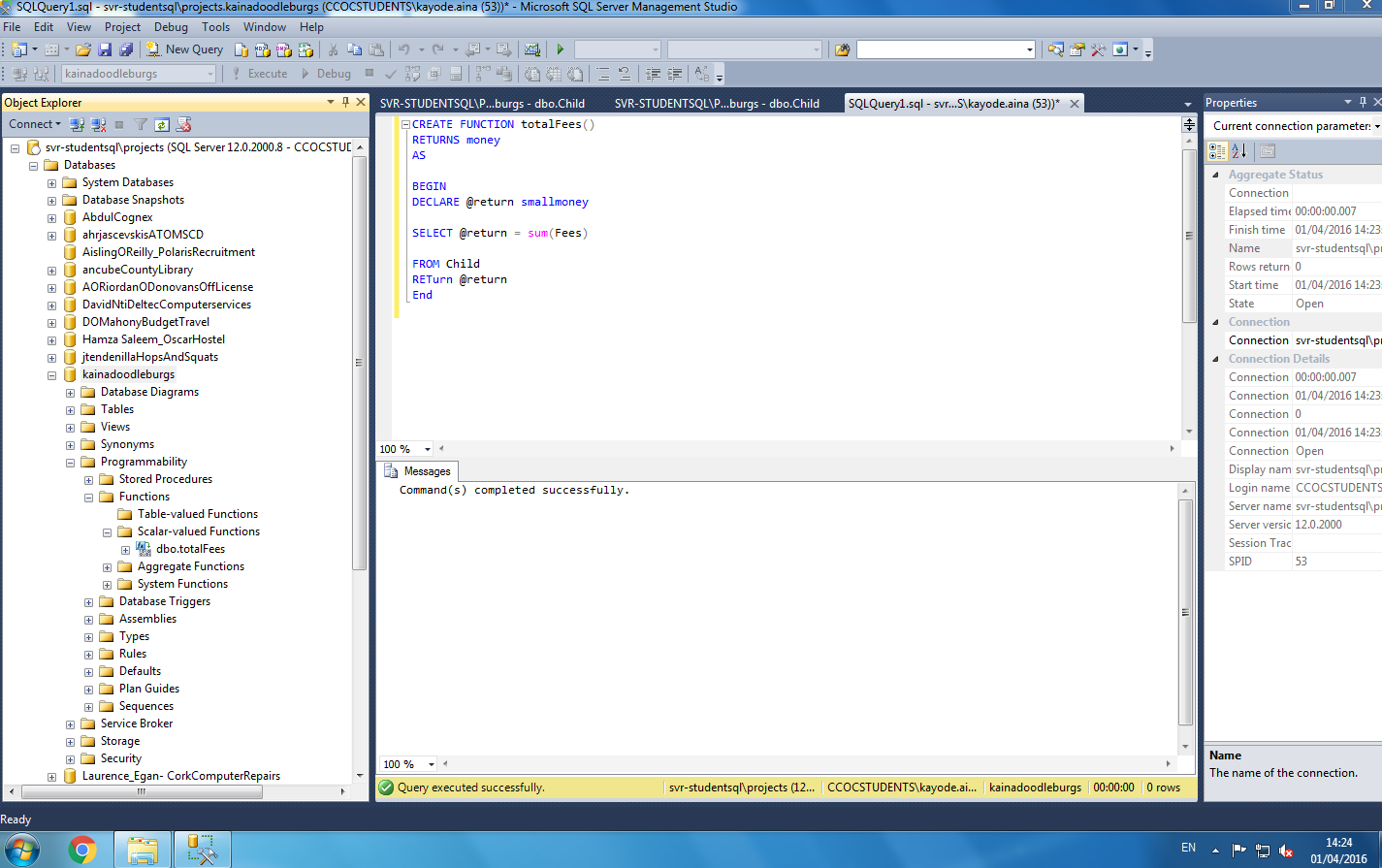
**Store procedure**



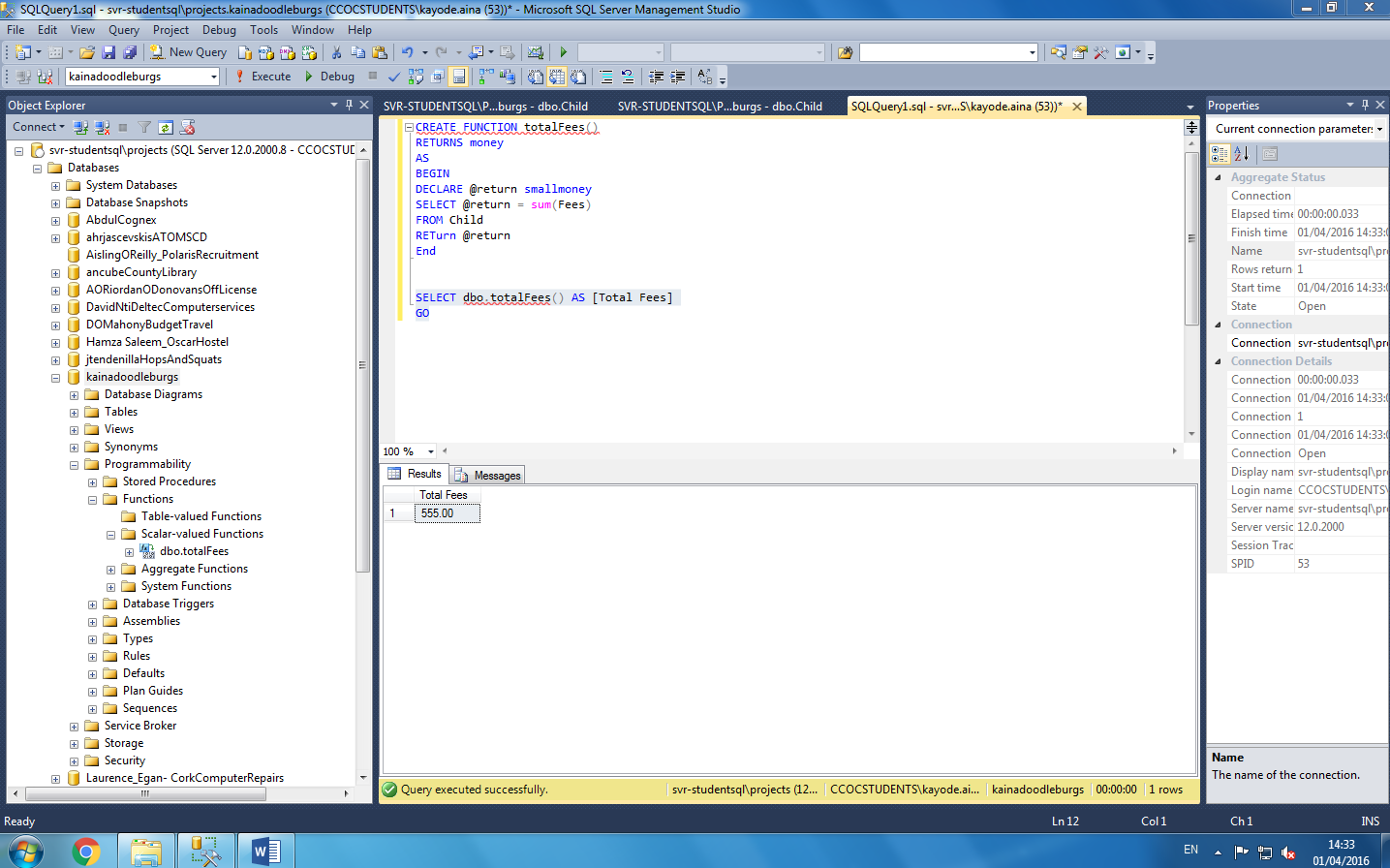
**Result on store procedure on disability**



**Function to calculate fees pay in child table**



**Result of the function calculate fees**

****

**Trigger table**

****

**Trigger code**

CREATE TRIGGER trChildForInsert

ON Child

FOR INSERT

AS

BEGIN

Declare @ChildNo tinyint

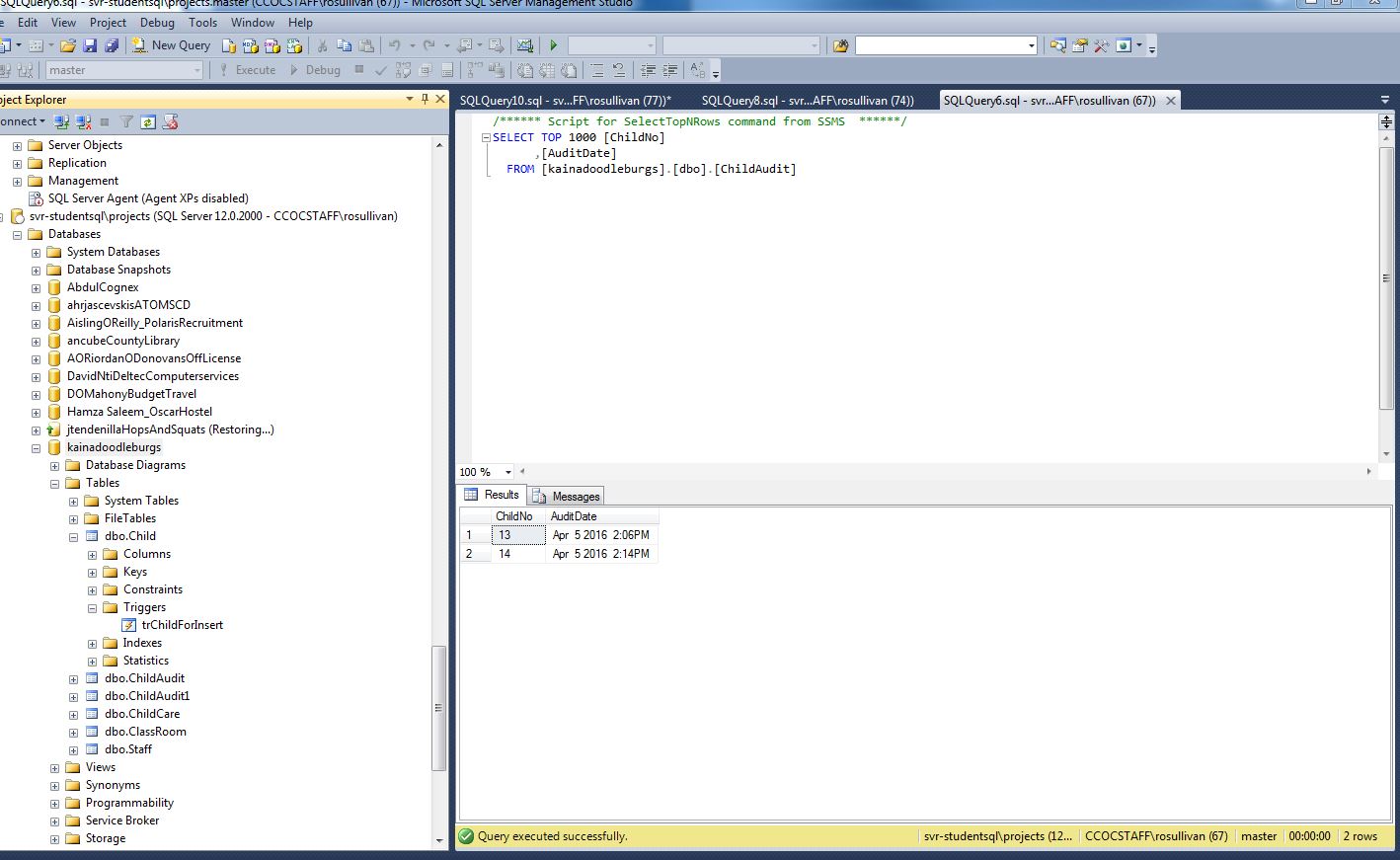
Set @ChildNo = (Select ChildNo from inserted)

Insert into ChildAudit

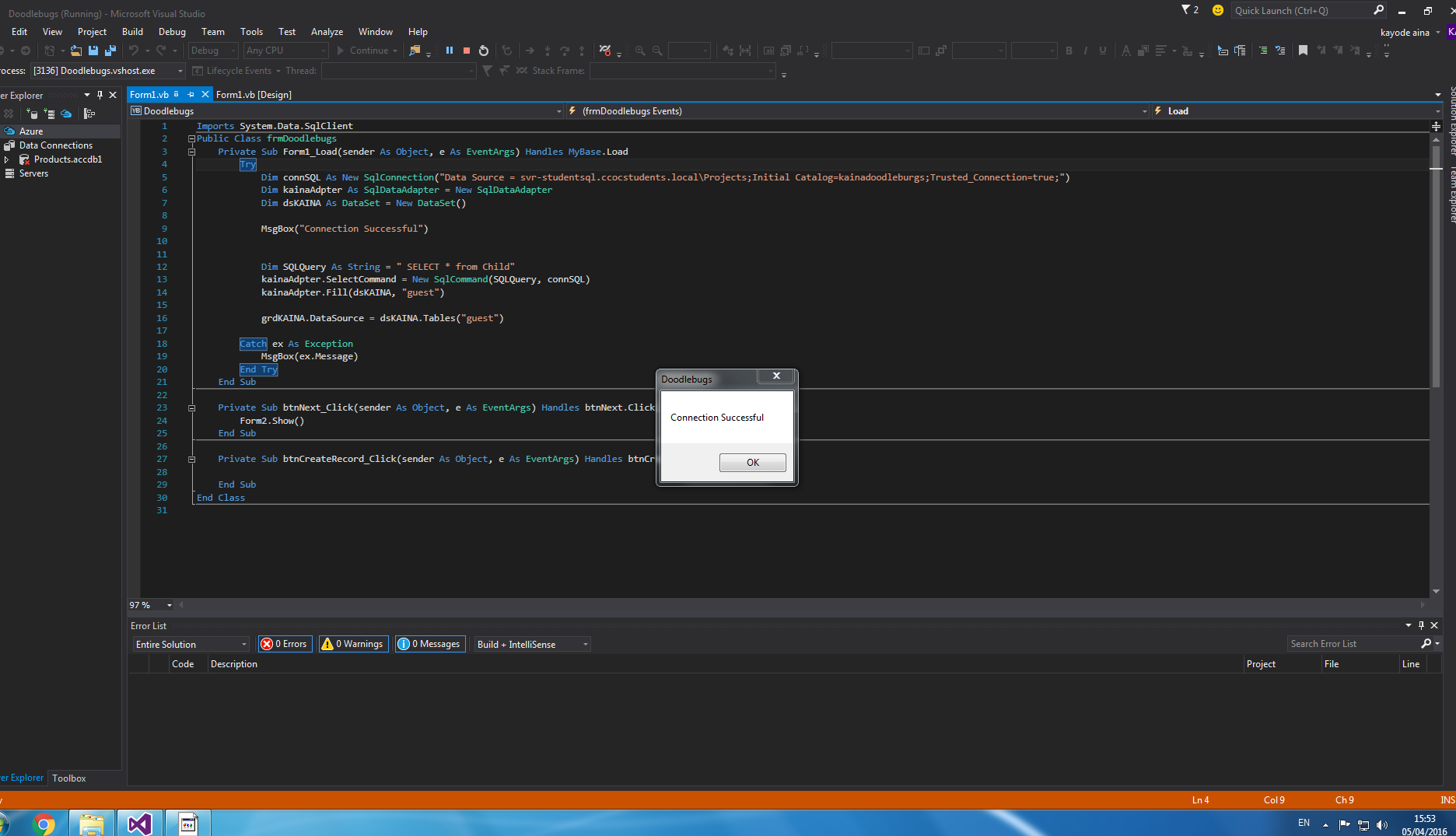
Values (Cast(@ChildNo as tinyint), Cast(Getdate () as nvarchar (20)))

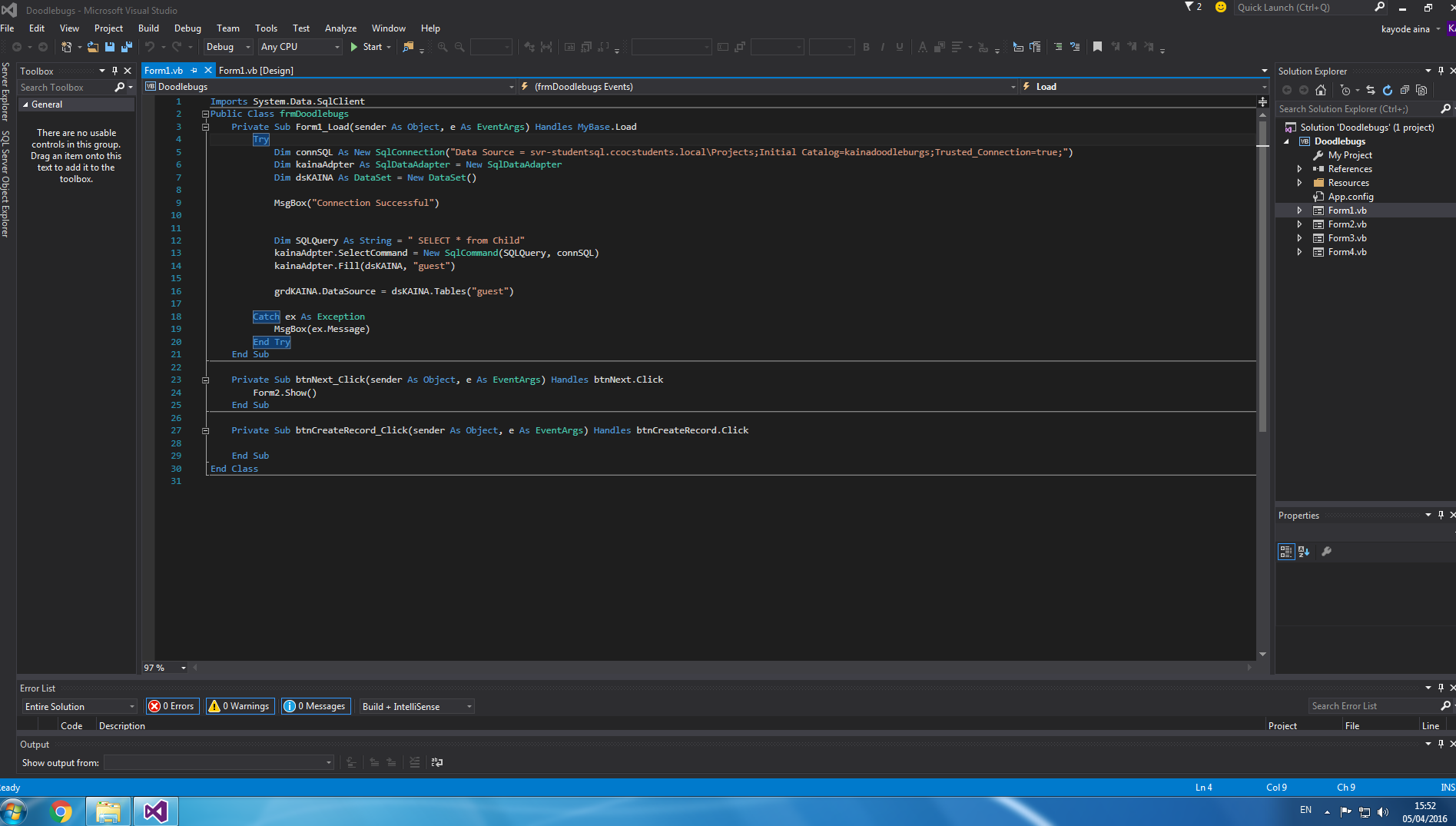
END

**Trigger result to actual added row, date and time of event**

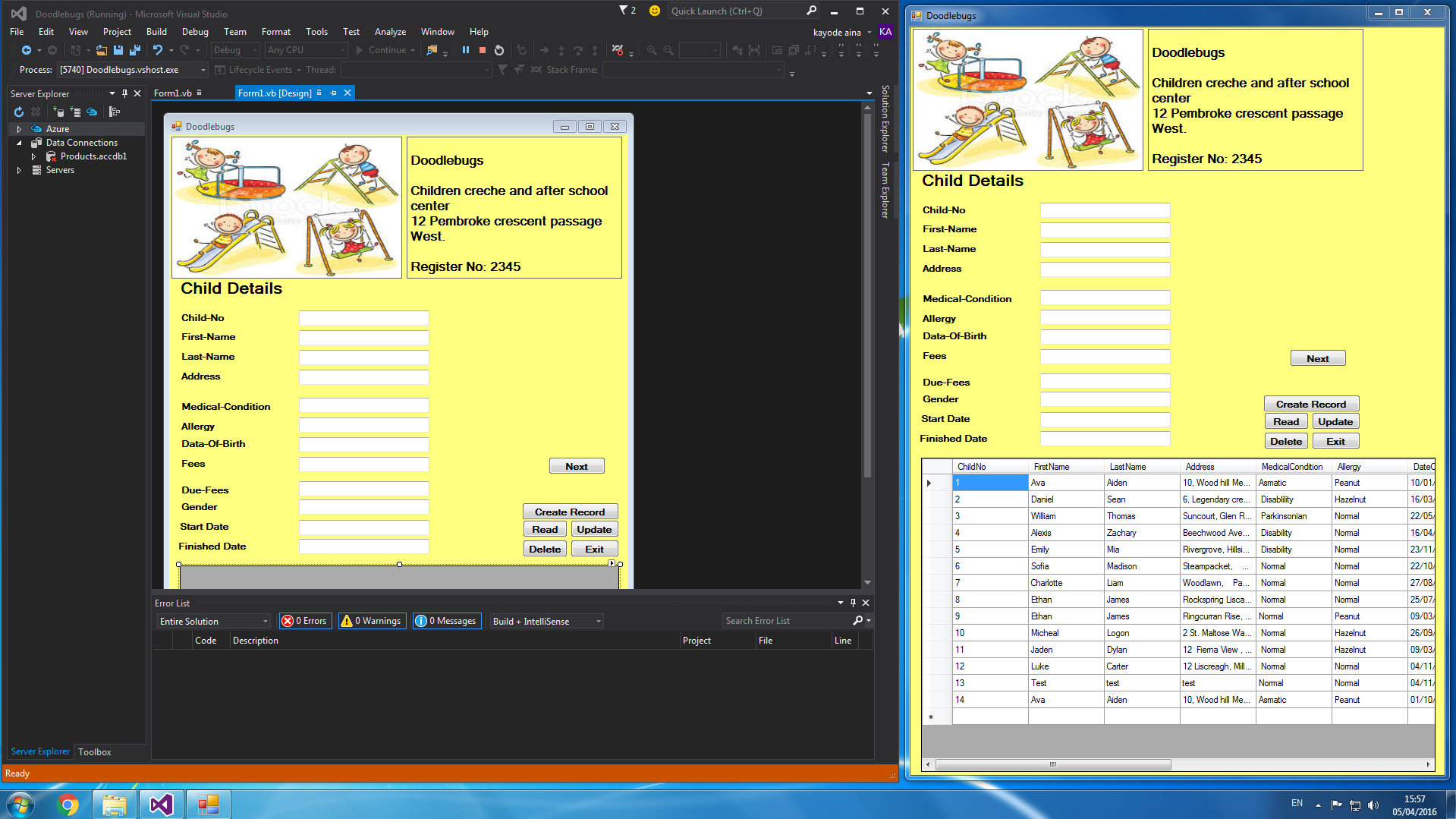
****

**Running code behind child form and connected to sql**

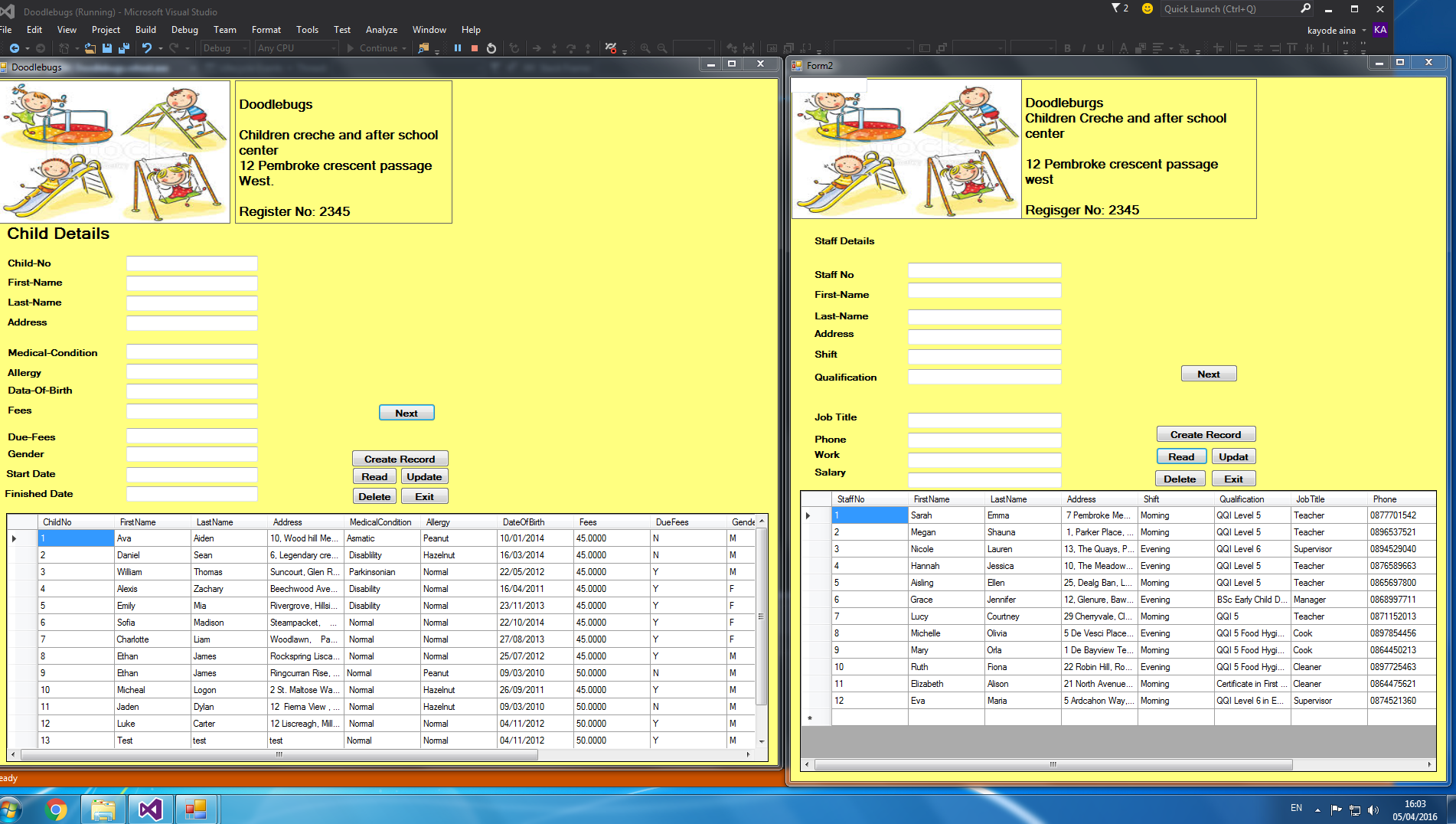




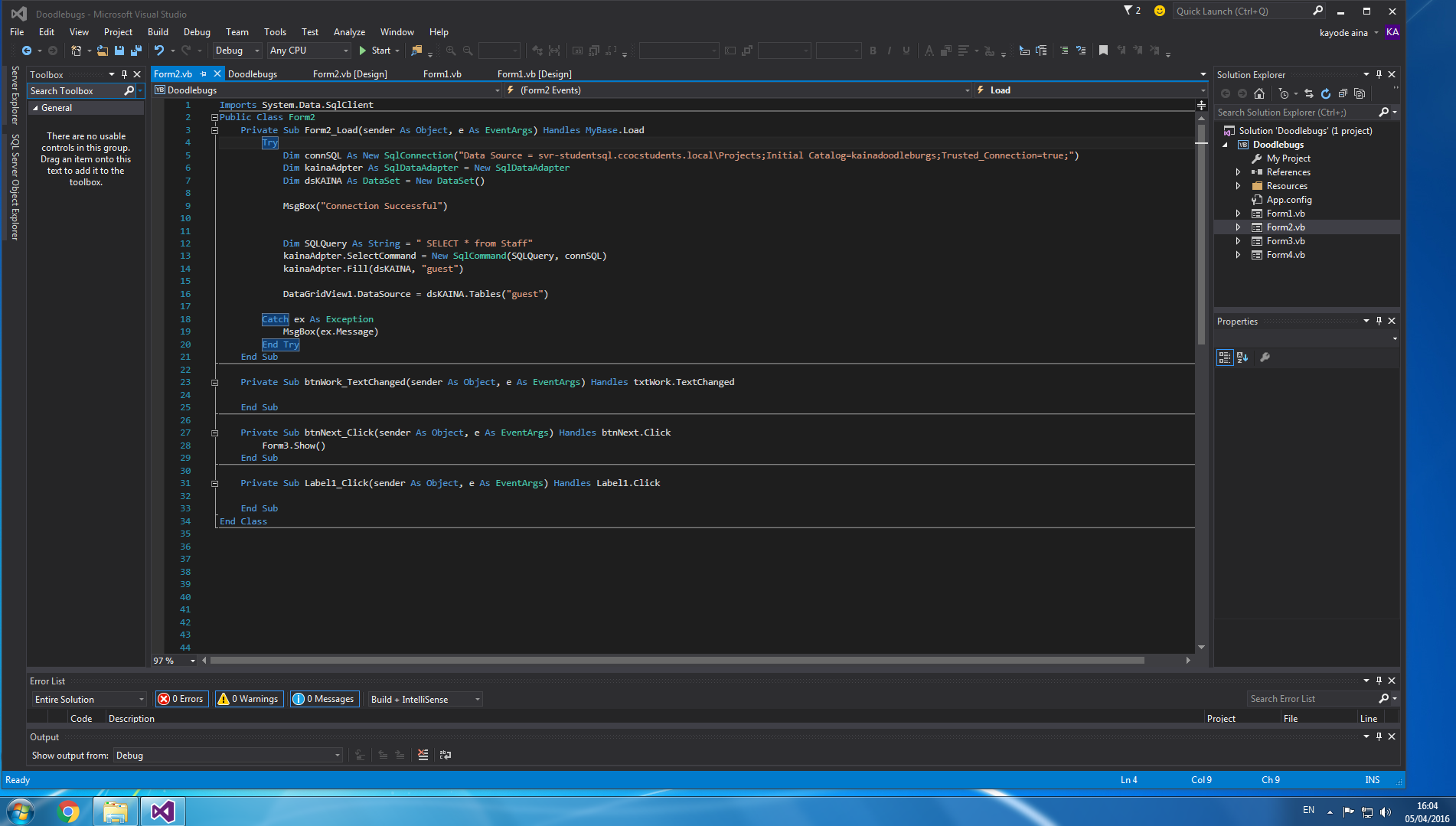
**Running data grid child form**



**Both data grid form running child and staff**



**Running code behind staff form**



**Modification**

* Added another field for next button.
* Address field size to accommodate much longer client and staff address
* Change the font size from 10 to 12
* Added one field to child table to established relation with staff table

**Conclusion**

I think the tests I did went smoothly and I had no problems, except for the fact that the data grid button is not coded but yet the form is still running. Therefore, In future I would have love to re-arrange field button horizontally for great view through visual studio.