**LAPTOP SERVICE CENTRE**

Name: Naghma Khan

Student No: 10514575

Contents

[1. Background for the Development of chosen system 2](#_Toc5982305)

[2. Scope 2](#_Toc5982306)

[2.1. BUsiness Requirements 2](#_Toc5982307)

[3. Business Rules 2](#_Toc5982308)

[3.1. Implementation of business rules – screen shots 2](#_Toc5982309)

[4. Relational Schema in 3NF – showing the scope 2](#_Toc5982310)

[4.1. XML in Schema 2](#_Toc5982311)

[5. Implementation in SQL Server 2](#_Toc5982312)

[7.1. Tables with Data Diagram 2](#_Toc5982313)

[7.2. Stored PROCEDURES 2](#_Toc5982314)

[7.3. Triggers 2](#_Toc5982315)

[7.4. Views and benefits of using views 2](#_Toc5982316)

[8. Conclusions 2](#_Toc5982317)

[9. innovation 2](#_Toc5982318)

[10. Bibliography 2](#_Toc5982319)

[Appendix A – Create table Queries 2](#_Toc5982320)

[Appendix B – INSERT INTO 2](#_Toc5982321)

# 1. Background for the Development of chosen system

Laptop Service Center is a center which deals with servicing of Laptops. There are several Service centers at many locations across the city. Every center makes use of the database management where all the Services are managed. And accordingly, the Branch ID and Service ID is set. There are 2 main categories such as Standard and Premium, a unique Category ID would be created too. Every client who comes for a Service has a unique Client ID created in the system, for the registration of details and for every Service a unique Order ID will be created with the details such as date and the bill. For every Service a Service ID, Name of the Service, Price of the Service, Category ID and the Order ID will be stated. We will maintain a database record for every Service with attributes like Service ID, Order ID and the bill for the Service. There are 2 Foreign keys, Service ID and Order ID. For every Laptop there must be a unique Serial number as the Primary key. Every Client’s ID should be linked as a Foreign key with the Serial number.

# 2. Scope

The basic understanding is that, we are implementing a Laptop Service database for the details of the Service done on several locations, which belongs to several Clients. By going through the Branch details and the repair done with the category type and Order ID we can successfully retrieve information. All the details could be taken out if the client wants to.

Everything has been implemented as per my best knowledge, however, Modify XML could not be added due to Time Constraints.

## 2.1. BUsiness Requirements

1. We will modify the Email Address of the Client if the Client needs to change the Email Address at any point of time using Parametric Stored Procedures.

2. Laptop Service Centre will help the user’s get the services done on the Laptops. Here we will retrieve the information of the Client details, Laptop Serial No, Services done on the Laptop and the total service Price.

3. Using Inner Join we join 2 tables.

4.We take the details of Sum of the Price for every Order placed by the Client using GroupBy, Where.

5.Using GroupBy, having we will retrieve the Total Price for every quantity of the Service with a condition that Total Price should be greater than 1000.

6.By using the Where condition we are retrieving details of only the Premium User’s registered for the Laptop.

7.We are counting Quantity using the Service Line table and using WHERE clause.

# 3. Business Rules

1.Every time a Laptop has been given for servicing, we can update the data with the ID, Time of the Service using the Trigger function.

2.And updating data be done with any other tables too using trigger Function.

3.System will accept only one Client ID for every user and as the ClientID is the Primary Key the duplication cannot be done.

4.Every Service need to have one Service ID , Service rate and service name and we are declaring this by declaring these values not null.

5.Clients who are opting for Premium will be charged extra fees.

## 3.1. Implementation of business rules – screen shots

**TRIGGER 1: Business Rule 1:**

When a client updates anything in the Order Table then this trigger will be fired, and the details of the update will be stored in another table OrderLog.

**Code:**

SELECT \* FROM Order\_Details

Create table Order\_Details

(

ID int IDENTITY (1,1),

Details\_Data varchar(100)

)

SELECT \* FROM tbl\_Order

GO

CREATE TRIGGER Order\_ToInsert\_Ti

ON tbl\_Order

For INSERT

AS

BEGIN

Declare @ID int

Select @ID= OrderID from inserted

insert into Order\_Details

Values('New Order with ID = ' + CAST(@ID as nvarchar(10)) + ' has been added'+ CAST (GETDATE() as nvarchar(20)))

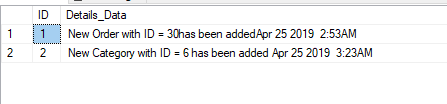
End

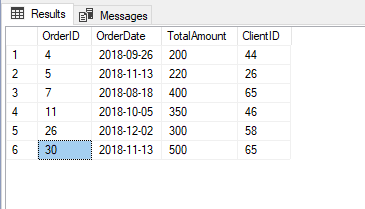
SELECT \* FROM tbl\_Order

INSERT INTO tbl\_Order

VALUES('30','2018-11-13','500','65')

SELECT \* FROM Order\_Details;





**TRIGGER 2: Business Rule 2:** When a client updates anything in the Order Table then this trigger will be fired, and the details of the update will be stored in another table OrderLog

**Code:**

CREATE TRIGGER Category\_ToInsert\_Ti

ON tbl\_Category

For INSERT

AS

BEGIN

Declare @TID int

Select @TID= CategoryID from inserted

insert into Order\_Details

Values('New Category with ID = ' + CAST(@TID as nvarchar(10)) + ' has been added '+ CAST (GETDATE() as nvarchar(20)))

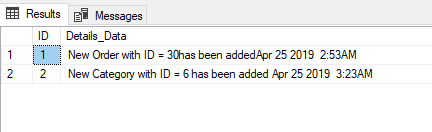
End

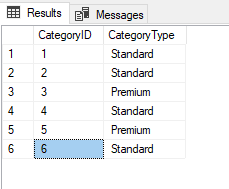
SELECT \* FROM tbl\_Category

INSERT INTO tbl\_Category

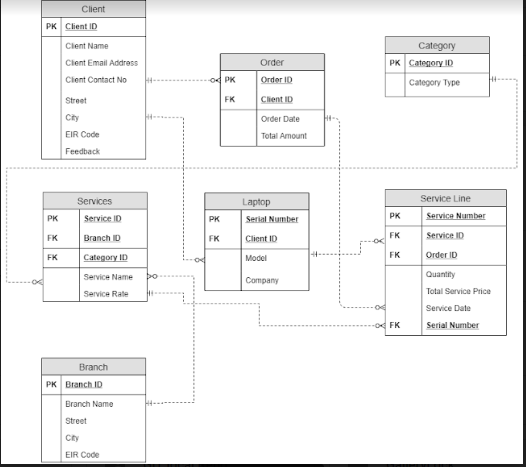
VALUES('6','Standard')

SELECT \* FROM Order\_Details;





# 4. Relational Schema in 3NF – showing the scope



This Schema explains the various relationships between the tables, One Client can have many Order(1 to many) and one Client could also have more than one laptop(1 to many), Also one Order can have many Service Lines(1 to many) where all the details of the services carried on the Laptop will be recorded.

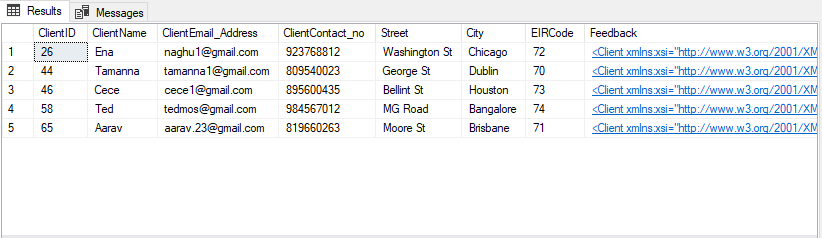
For every Service many Service Lines can be created which stores the details of the Quantity, Price and Date of Service. One Branch can have many Services and every Service can belong to only one Category (1 to 1).

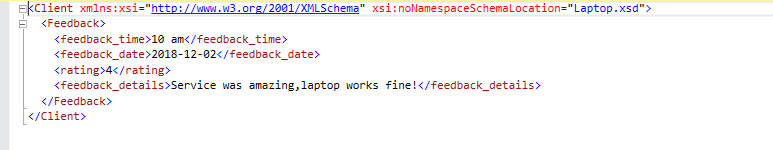
Every laptop can have n number of Service Lines where the data gets recorded for every laptop and its Servicing details.

This Schema is in 3NF form, as there are no Dependencies inside the Tables and the attributes do not depend on each other.

All the Attributes are unique and cannot be formed in common as another table.

## 4.1. XML in Schema





In the Client Table we are giving one Attribute Feedback as XML datatype, we create an XML program in Visual studio and write a DTD format program check the validation and generate XML.

We can also search and modify the XML.

# 5. Implementation in SQL Server

## 5.1. Tables with Data Diagram

**1)dbo.Client:**

As we gave some introduction about the Database the Fig 1.1 is showing a relational table, keeps all the record about different Clients of this Centre. ClientID represents the primary key (PK) of the table. Then comes the various attributes of the Client as Name, Email\_Address, Contact\_no, Street, City , EIRCode and Feedback which is XML Data Type.

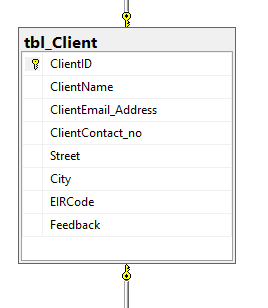
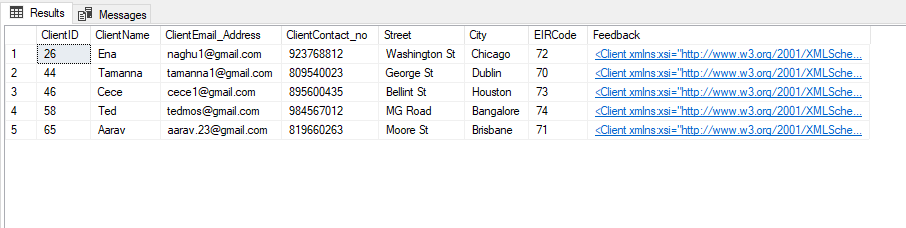


Fig 1.1



**2)dbo.Order:**

In Fig 1.2 all the Orders of the Client are being recorded. OrderID represents the primary key (PK) of the table. And the other attributes are shown in the Fig 1.2 and the Client ID is the Foreign key for this table.

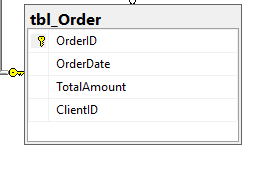
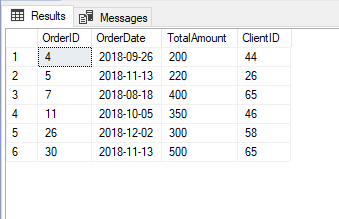
****

Fig 1.2



**3)dbo.Laptop:**

In Fig 1.3 we have Serial number as the Primary Key, from where we retrieve all the details of the Orders given by clients as the Order ID is the Foreign key here.

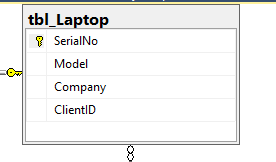
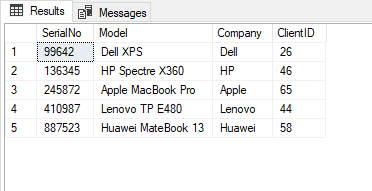
****

Fig 1.3



**4)dbo.Branch:**

Here as we show in Fig 1.4 the table has a unique key called Primary key called Branch ID every Branch will have many Services done on laptops and the other attributes are shown in the below screenshot.

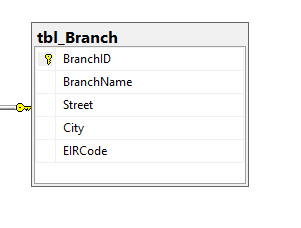
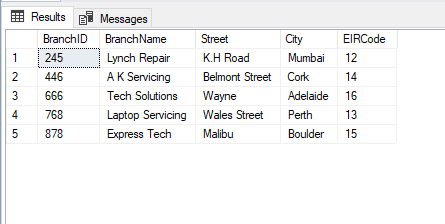
****

Fig 1.4

**5)dbo.Category:**

Here as we show in Fig 1.5 the table has a unique key called Primary key called Category ID, and we have two categories one is standard and the other is premium. Every User can be registered under one of these categories.

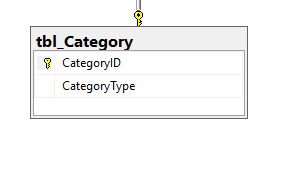
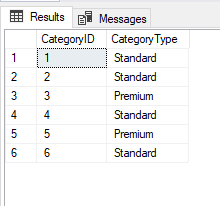
****

Fig 1.5



**6)dbo.Services:**

In this Fig 1.6 we have a Primary key called Service Id where the details about the Service name, and the Price are being stored. Branch ID and Category ID are the Foreign Keys.

Branch to know where the Laptop was given and the services asked for it and Category because every user will be of different type of categories as mentioned in Fig 1.5

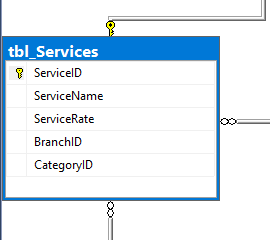
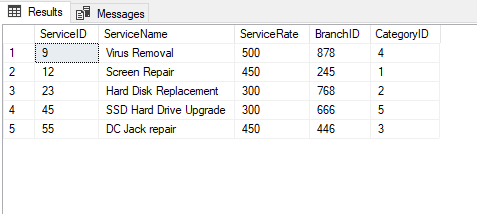
****

Fig 1.6



**7)dbo.Service\_Line:**

In this Table we record the number of Services done on the Laptop with the details as seen below.

Service Number is the Primary key and Order Id, Service ID are the Foreign keys as we retrieve details from those tables too.

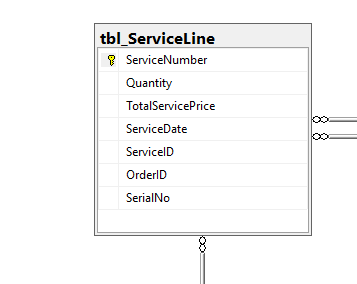
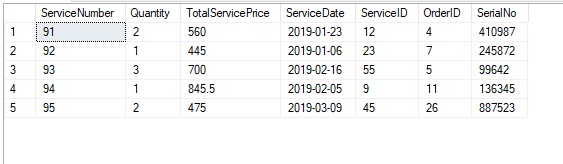
****

Fig 1.7



## 5.2. Stored PROCEDURES

**Stored Procedure 1(Business requirement 1):**

GO

CREATE Procedure [dbo].[LaptopService]

@ClientEmail\_Address varchar (40)

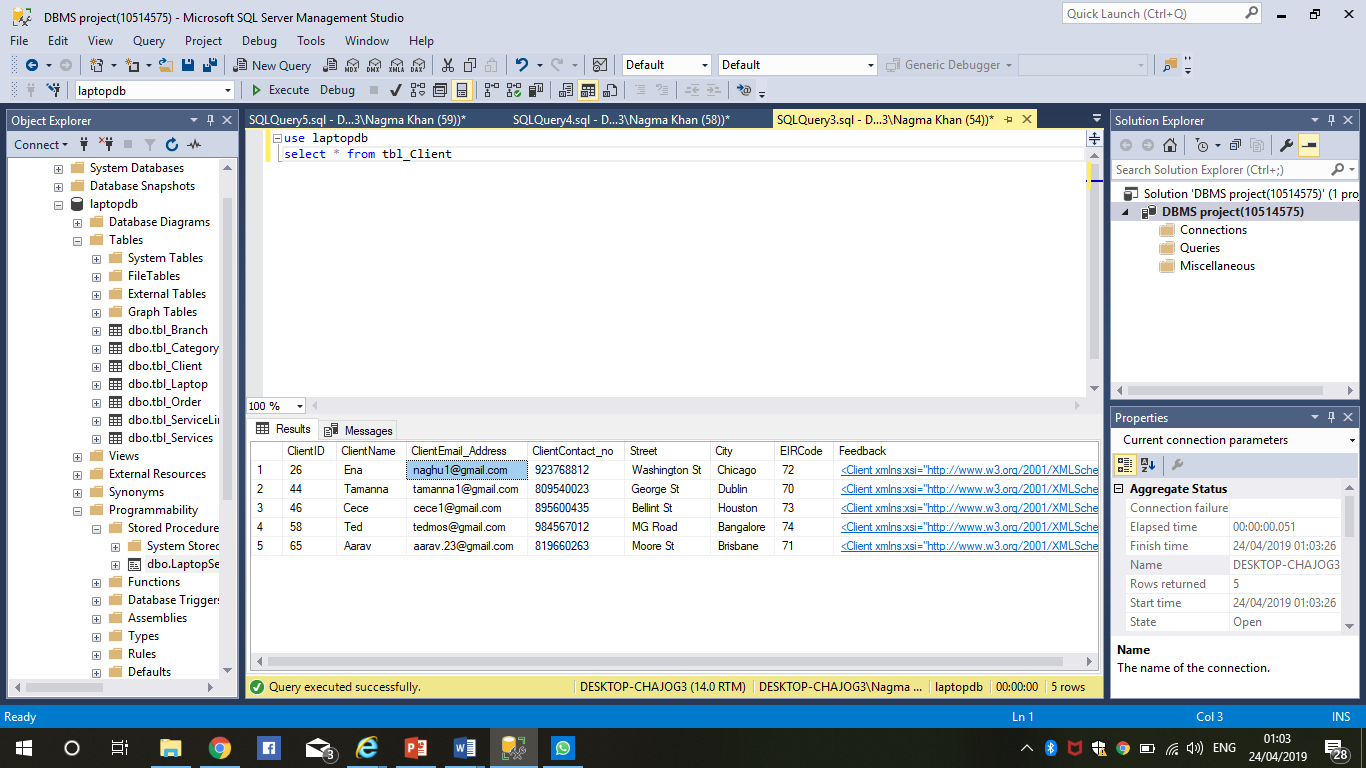
AS

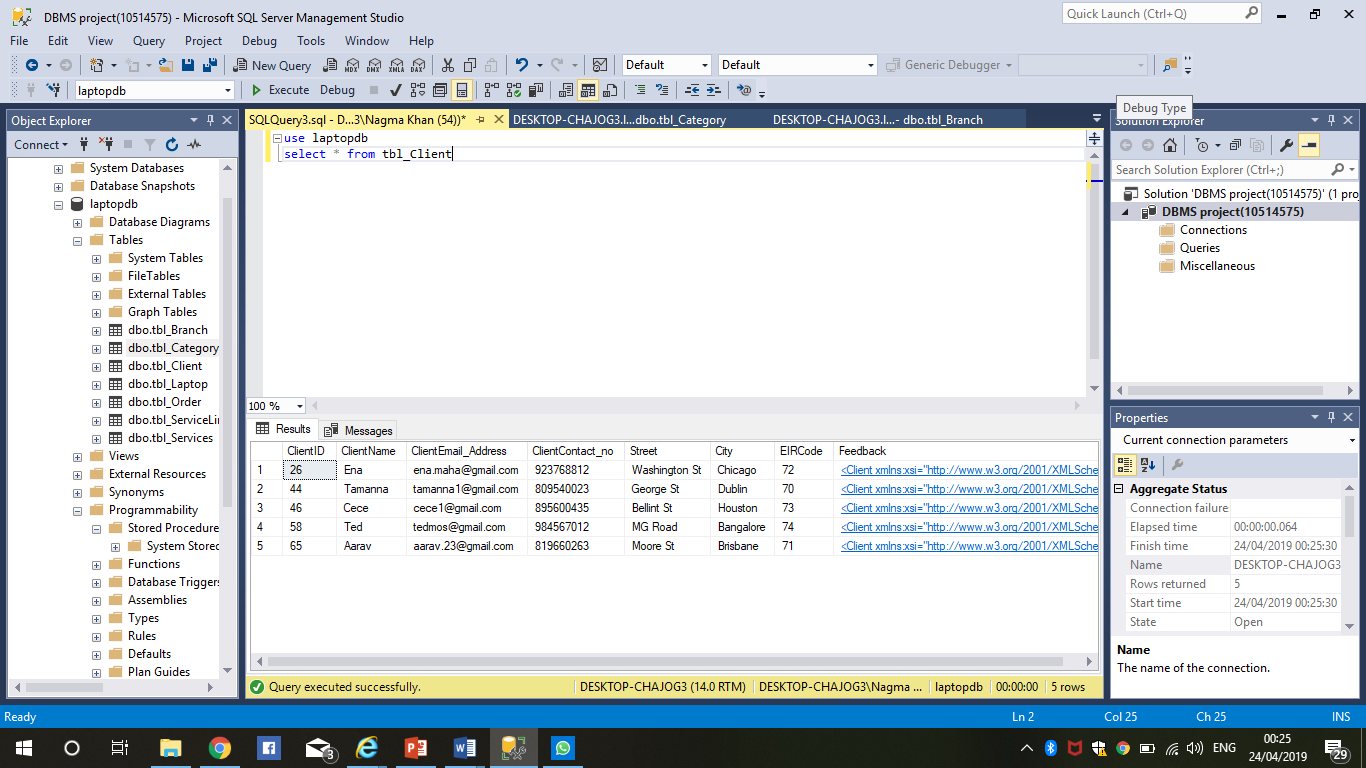
UPDATE tbl\_Client

SET ClientEmail\_Address=@ClientEmail\_Address

WHERE ClientEmail\_Address='ena.maha@gmail.com'

GO

****



**Stored Procedure 2(Business requirement 2):**

CREATE PROCEDURE [dbo].[RetrieveAllDetails]

AS

SELECT C.ClientName,C.ClientContact\_no,L.Model,SL.Quantity,SL.ServiceDate,S.ServiceName,S.ServiceRate,B.BranchName,B.EIRCode,CT.CategoryType

FROM tbl\_Client C with (nolock)

left JOIN tbl\_Laptop L (NOLOCK) ON C.ClientID = L.ClientID

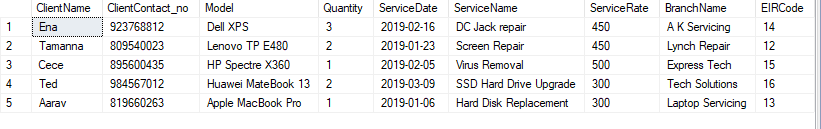
left Join tbl\_ServiceLine SL on L.SerialNo = SL.SerialNo

left Join tbl\_Services S on SL.ServiceID = S.ServiceID

left Join tbl\_Branch B on S.BranchID = B.BranchID

left Join tbl\_Category CT on CT.CategoryID = S.CategoryID

GO



**Stored Procedure 3(Business requirement 3):**

CREATE Procedure [dbo].[GetLaptopServiceDetails]

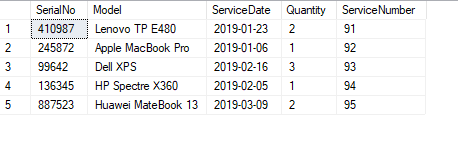
AS

BEGIN

SELECT t1.SerialNo,t1.Model,t2.ServiceDate,t2.Quantity,t2.ServiceNumber FROM tbl\_Laptop t1 inner join tbl\_ServiceLine t2 on t1.SerialNo=t2.SerialNo

END

GO



**Stored Procedure 4(Business requirement 4):**

CREATE PROCEDURE [dbo].[Groupby]

AS

BEGIN

SELECT OrderDate O,

SUM(TotalServicePrice)

AS TotalSpent

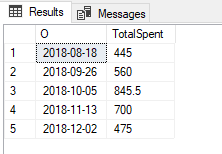
FROM tbl\_Order O , tbl\_ServiceLine S

WHERE O.OrderID = S.OrderID

GROUP BY O.OrderDate

END

GO



**Stored Procedure 5(Business requirement 5):**

CREATE PROCEDURE [dbo].[GroupbyHaving]

AS

BEGIN

Select Quantity, SUM(TotalServicePrice)

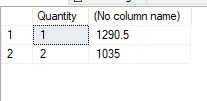
From tbl\_ServiceLine

GROUP by Quantity

having SUM(TotalServicePrice)>1000

END

GO



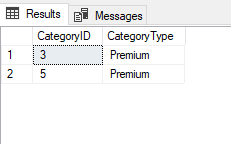
**Stored Procedure 6(Business requirement 6):**

CREATE PROCEDURE [dbo].[SelectPremiumUsers] @CategoryType varchar(20)

AS

SELECT \* FROM tbl\_Category WHERE CategoryType = @CategoryType;

GO



**Stored Procedure 7(Business requirement 7):**

CREATE PROCEDURE [dbo].[Countquantity] @quantity int

AS

BEGIN

SELECT Quantity,COUNT(\*) as [counting quantity for service date]

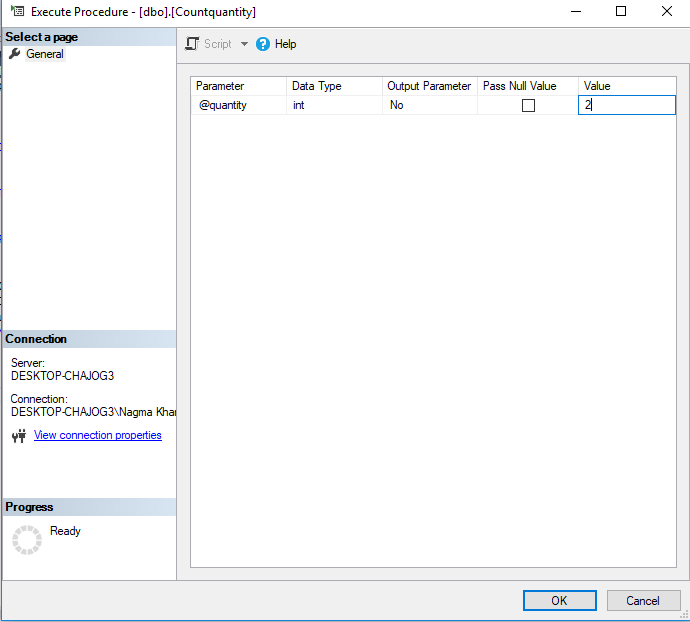
from tbl\_ServiceLine

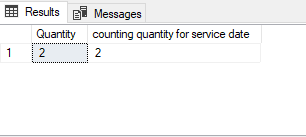
WHERE Quantity=@quantity

GROUP BY Quantity

END

GO





## 5.3. Triggers

**TRIGGER 1:** When a client updates anything in the Order Table then this trigger will be fired, and the details of the update will be stored in another table OrderLog.

**CODE:**

SELECT \* FROM Order\_Details

Create table Order\_Details

(

ID int IDENTITY (1,1),

Details\_Data varchar(100)

)

SELECT \* FROM tbl\_Order

GO

CREATE TRIGGER Order\_ToInsert\_Ti

ON tbl\_Order

For INSERT

AS

BEGIN

Declare @ID int

Select @ID= OrderID from inserted

insert into Order\_Details

Values('New Order with ID = ' + CAST(@ID as nvarchar(10)) + ' has been added'+ CAST (GETDATE() as nvarchar(20)))

End

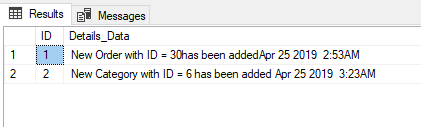
SELECT \* FROM tbl\_Order

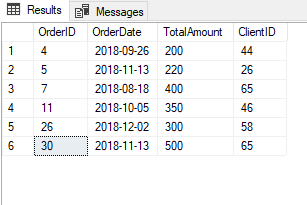
INSERT INTO tbl\_Order

VALUES('30','2018-11-13','500','65')

SELECT \* FROM Order\_Details;

TRIGGER INITIATED:





**TRIGGER 2: Business Rule 2:** When a client updates anything in the Category Table then this trigger will be fired, and the details of the update will be stored in another table CategoryLog.

**Code:**

CREATE TRIGGER Category\_ToInsert\_Ti

ON tbl\_Category

For INSERT

AS

BEGIN

Declare @TID int

Select @TID= CategoryID from inserted

insert into Order\_Details

Values('New Category with ID = ' + CAST(@TID as nvarchar(10)) + ' has been added '+ CAST (GETDATE() as nvarchar(20)))

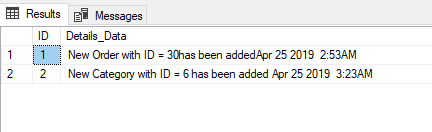
End

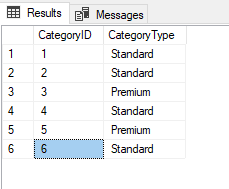
SELECT \* FROM tbl\_Category

INSERT INTO tbl\_Category

VALUES('6','Standard')

SELECT \* FROM Order\_Details;





## 5.4. Views and benefits of using views

**VIEW 1:**

**CODE:**

CREATE VIEW [dbo].[Service\_Price]

AS

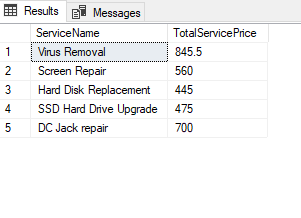
SELECT ServiceName, TotalServicePrice

from tbl\_Services s left join tbl\_ServiceLine l

on

s.ServiceID = l.ServiceID

GO



In this View, we are trying to retrieve ServiceName from tbl\_Services and TotalServicePrice from tbl\_ServiceLine by using the left join function and give the view of the data as shown in the screenshot above.

**VIEW 2:**

**Code:**

CREATE VIEW [dbo].[tablesview]

AS

SELECT

Model, ServiceName, OrderDate

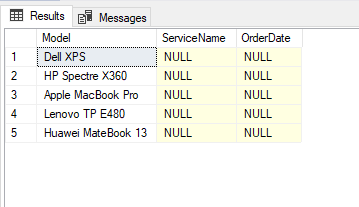
from tbl\_Laptop s left join tbl\_Services t

on s.SerialNo = t.ServiceID

left join tbl\_Order c

on t.ServiceID = c.OrderID

GO



In this View, we are trying to retrieve Model from tbl\_Laptop and ServiceName from tbl\_Services and OrderDate from tbl\_Order by using the left join function and give the view of the data as shown in the screenshot above.

# 6. Conclusions

The database system is completed by using SQL commands of how to create tables, datatypes, insert values, alter columns.

The Laptop Servicing is being done and all the Service Details and Laptop Details and Order Details and other Details related to Client and the Laptop are being displayed. Also stored procedures are being implemented for Business Requirements and Views are being implemented and Business Rules are being implemented using Triggers and SQL.

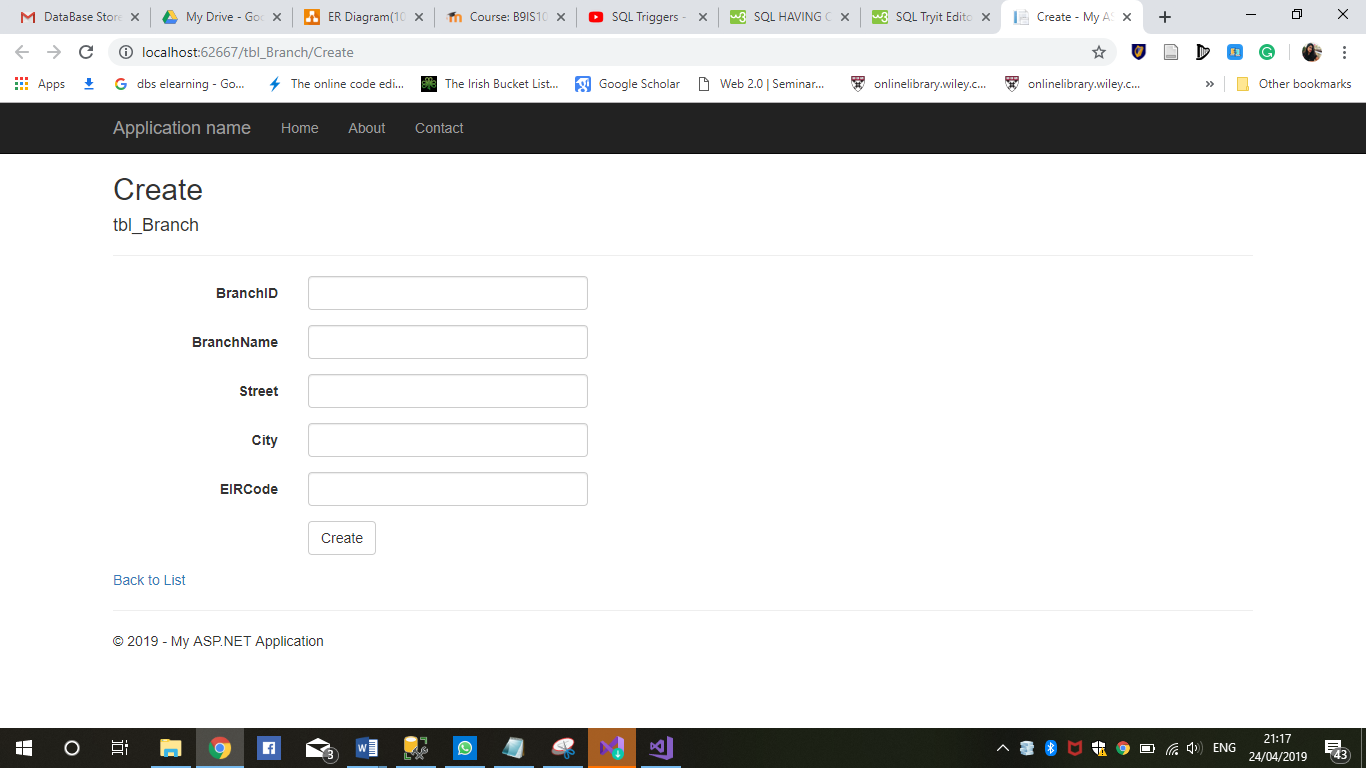
XML has also been implemented and executed using SQL commands

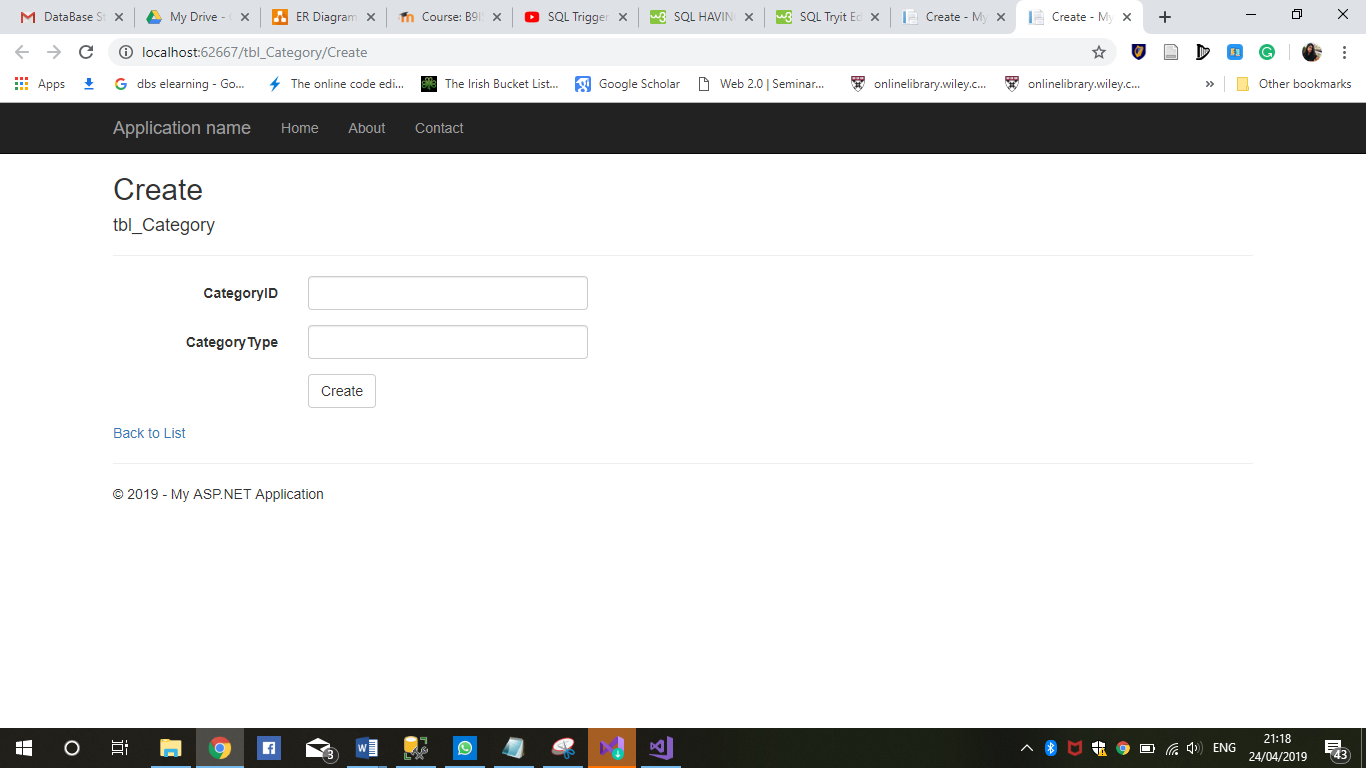
# 7. innovation

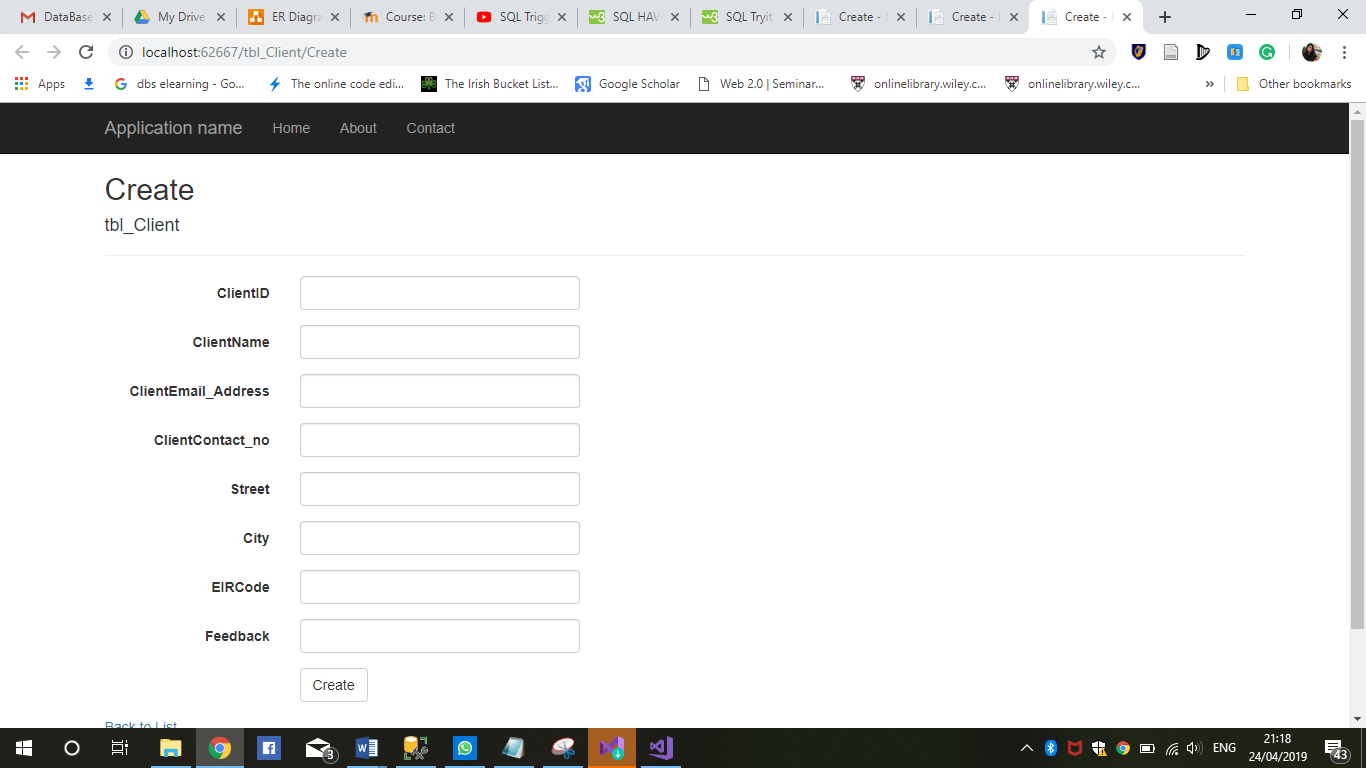
Front End is being designed with MVC Web Application with the use of ASP.NET where we select individual tables and scaffold tables to be executed on a website, this will act as an Interface to provide the Database

For every table we have created on our Database and the SQL Server by linking the Server name of the SQL Project and the Project Name.

All tables have been designed and saved on a file attached with the project folder and few Screenshots are attached below to show the Functionality.

****

****

****

# 8. Bibliography

1) <https://stackoverflow.com/>

2) <https://www.xmlvalidation.com/>

3) <https://docs.microsoft.com/en-us/sql/t-sql/functions/functions?view=sql-server-2017>

4) <https://www.w3schools.com/sql/>

5) <https://www.dofactory.com/sql/join>

# Appendix A – Create table Queries

**1)tbl\_Client:**

GO

CREATE TABLE tbl\_Client

(

ClientID int not null PRIMARY KEY,

ClientName varchar(100) not null UNIQUE,

ClientEmail\_Address varchar(40) not null,

ClientContact\_no int not null,

Street varchar(100) not null,

City varchar(30) not null,

EIRCode int not null,

Feedback xml

)

ALTER TABLE [dbo].[tbl\_Client]

ALTER COLUMN ClientEmail\_Address varchar(100)

**2)tbl\_Order:**

GO

CREATE TABLE tbl\_Order

(

OrderID int not null PRIMARY KEY,

OrderDate date not null,

TotalAmount int not null,

ClientID int FOREIGN KEY(ClientID) References tbl\_Client(ClientID),

)

**3)tbl\_Branch:**

GO

CREATE TABLE tbl\_Branch

(

BranchID int not null PRIMARY KEY,

BranchName varchar(50) not null,

Street varchar(100) not null,

City varchar(50) not null,

EIRCode int not null

)

**4)tbl\_Laptop:**

GO

CREATE TABLE tbl\_Laptop

(

SerialNo int PRIMARY KEY not null,

Model varchar(50) not null,

Company varchar(50) not null,

ClientID int FOREIGN KEY(ClientID) References tbl\_Client(ClientID),

)

**5)tbl\_Category:**

CREATE TABLE tbl\_Category

(

CategoryID int not null PRIMARY KEY,

CategoryType varchar(20))

**6)tbl\_Services:**

GO

CREATE TABLE tbl\_Services

(

ServiceID int not null PRIMARY KEY,

ServiceName varchar(50) not null,

ServiceRate float(24) not null,

BranchID int FOREIGN KEY(BranchID) References tbl\_Branch(BranchID),

CategoryID int FOREIGN KEY(CategoryID) References tbl\_Category(CategoryID),

)

**7)tbl\_ServiceLine:**

GO

CREATE TABLE tbl\_ServiceLine

(

ServiceNumber int not null PRIMARY KEY,

Quantity int not null,

TotalServicePrice float(24) not null,

ServiceDate date not null,

ServiceID int FOREIGN KEY(ServiceID) References tbl\_Services(ServiceID),

OrderID int FOREIGN KEY(OrderID) References tbl\_Order(OrderID),

SerialNo int FOREIGN KEY(SerialNo) References tbl\_Laptop(SerialNo),

)

**OR**

**DER BY Customers.CustomerID;**

# Appendix B – INSERT INTO

**1)tbl\_Client:**

INSERT INTO tbl\_Client VALUES

('44','Tamanna','tamanna1@gmail.com','809540023','George St','Dublin','70','<?xml version="1.0"?>

<Client xmlns:xsi="http://www.w3.org/2001/XMLSchema" xsi:noNamespaceSchemaLocation="Laptop.xsd">

<Feedback>

<feedback\_time>10 am</feedback\_time>

<feedback\_date>2018-12-02</feedback\_date>

<rating>4</rating>

<feedback\_details>Service was amazing,laptop works fine!</feedback\_details>

</Feedback>

</Client>')

INSERT INTO tbl\_Client VALUES

('65','Aarav','aarav.23@gmail.com','819660263','Moore St','Brisbane','71','<?xml version="1.0"?>

<Client xmlns:xsi="http://www.w3.org/2001/XMLSchema" xsi:noNamespaceSchemaLocation="Laptop.xsd">

<Feedback>

<feedback\_time>12 am</feedback\_time>

<feedback\_date>2019-01-26</feedback\_date>

<rating>4.5</rating>

<feedback\_details>Issue was fixed. Great work.</feedback\_details>

</Feedback>

</Client>')

NSERT INTO tbl\_Client VALUES

('26','Ena','ena.maha@gmail.com','923768812','Washington St','Chicago','72','<?xml version="1.0"?>

<Client xmlns:xsi="http://www.w3.org/2001/XMLSchema" xsi:noNamespaceSchemaLocation="Laptop.xsd">

<Feedback>

<feedback\_time>10 am</feedback\_time>

<feedback\_date>2018-12-02</feedback\_date>

<rating>4</rating>

<feedback\_details>Service was amazing,laptop works fine!</feedback\_details>

</Feedback>

</Client>')

INSERT INTO tbl\_Client VALUES

('46','Cece','cece1@gmail.com','895600435','Bellint St','Houston','73','<?xml version="1.0"?>

<Client xmlns:xsi="http://www.w3.org/2001/XMLSchema" xsi:noNamespaceSchemaLocation="Laptop.xsd">

<Feedback>

<feedback\_time>10 am</feedback\_time>

<feedback\_date>2018-12-02</feedback\_date>

<rating>4</rating>

<feedback\_details>Service was amazing,laptop works fine!</feedback\_details>

</Feedback>

</Client>')

INSERT INTO tbl\_Client VALUES

('58','Ted','tedmos@gmail.com','984567012','MG Road','Bangalore','74','<?xml version="1.0"?>

<Client xmlns:xsi="http://www.w3.org/2001/XMLSchema" xsi:noNamespaceSchemaLocation="Laptop.xsd">

<Feedback>

<feedback\_time>10 am</feedback\_time>

<feedback\_date>2018-12-02</feedback\_date>

<rating>4</rating>

<feedback\_details>Service was amazing,laptop works fine!</feedback\_details>

</Feedback>

</Client>')

SELECT \*FROM tbl\_Client.

**2)tbl\_Order:**

INSERT INTO tbl\_Order VALUES

('04','2018-09-26','200','44')

INSERT INTO tbl\_Order VALUES

('07','2018-08-18','400','65')

INSERT INTO tbl\_Order VALUES

('05','2018-11-13','220','26')

INSERT INTO tbl\_Order VALUES

('11','2018-10-05','350','46')

INSERT INTO tbl\_Order VALUES

('26','2018-12-02','300','58')

SELECT \*FROM tbl\_Order.

**3)tbl\_Branch:**

INSERT INTO tbl\_Branch VALUES

('245','Lynch Repair','K.H Road','Mumbai','12')

INSERT INTO tbl\_Branch VALUES

('768','Laptop Servicing','Wales Street','Perth','13')

INSERT INTO tbl\_Branch VALUES

('446','A K Servicing','Belmont Street','Cork','14')

INSERT INTO tbl\_Branch VALUES

('878','Express Tech','Malibu','Boulder','15')

INSERT INTO tbl\_Branch VALUES

('666','Tech Solutions','Wayne','Adelaide','16')

SELECT \*FROM tbl\_Branch.

**4)tbl\_Laptop:**

INSERT INTO tbl\_Laptop VALUES

('410987','Lenovo TP E480','Lenovo','44')

INSERT INTO tbl\_Laptop VALUES

('245872','Apple MacBook Pro','Apple','65')

INSERT INTO tbl\_Laptop VALUES

('099642','Dell XPS','Dell','26')

INSERT INTO tbl\_Laptop VALUES

('136345','HP Spectre X360','HP','46')

INSERT INTO tbl\_Laptop VALUES

('887523','Huawei MateBook 13','Huawei','58')

SELECT \*FROM tbl\_Laptop.

**5)tbl\_Category:**

INSERT INTO tbl\_Category VALUES

('1','Standard')

INSERT INTO tbl\_Category VALUES

('2','Standard')

INSERT INTO tbl\_Category VALUES

('3','Premium')

INSERT INTO tbl\_Category VALUES

('4','Standard')

INSERT INTO tbl\_Category VALUES

('5','Premium')

INTO tbl\_Services VALUES

**5)tbl\_Services:**

('12','Screen Repair','450','245','1'),

('23','Hard Disk Replacement','300','768','2'),

('55','DC Jack repair','450','446','3'),

('09','Virus Removal','500','878','4'),

('45','SSD Hard Drive Upgrade','300','666','5')

SELECT \*FROM tbl\_Services.

**7)tbl\_ServiceLine:**

INSERT INTO tbl\_ServiceLine VALUES

('91','2','560.0','2019-01-23','12','04','410987'),

('92','1','445','2019-01-06','23','07','245872'),

('93','3','700','2019-02-16','55','05','099642'),

('94','1','845.5','2019-02-05','09','11','136345'),

('95','2','475','2019-03-09','45','26','887523')

SELECT \*FROM tbl\_ServiceLine.