CHAPTER 4
OBSERVATION, RESULT AND DISCUSSION

This chapter covers observation, result and discuss various issues related to security of cloud computing in special reference to the banking system as Cloud computing receives significant attention recently. Public clouds are available from Amazon, Google, Yahoo!, Microsoft, Salesforce.com and others. Private cloud technologies, in which the cloud software is loaded locally, are available from VMware, Eucalyptus, Citrix, and there are thousands of vendors offering “cloud solutions”. However, storing valuable business data online create a situation similar to storing “money”, attracting frequent assaults by malicious attackers. As a result, security is a high main alarm issue in clouds. Several interesting concerns are often embedded in customers’ mind, such as: Can cloud employees/administrators be trusted to not to look at private data or change it? Can other customers of the cloud access private data by any means including hacking? The security and privacy violations of business data can be devastating. Several cloud security accidents had already happened. One of the notable security incidents occurred in March 2009 with Google Docs, when a system failure allowed the content of private documents to be exposed to everyone for a brief period of time. As a result of this security breakdown, The Electronic Privacy Information Center (EPIC), filed a detailed complaint with the Federal Trade Commission to request an injunction against Google offering their cloud service until “safeguards are verifiably established” claiming. Google’s inadequate security is a deceptive business practice. Cloud security and vulnerability are similar to the traditional issues in networking and applications. In a cloud environment, security mostly depends on the security mechanisms supplied by cloud providers. They control the hardware and the hypervisors on which data are stored and applications are run. A cloud and conventional data center share many characteristics. On the other hand, in the cloud, due to multi-tenancy architecture, the same software can store and manage the data from multiple clients. A mistake is made by software; private data can be
accessed of other clients. In addition, data stored in a cloud may be available to cloud manager or adapt data can be accessed for their own profit. The MTA danger has been greater than before due to the distribution of software, information and data schemas by numerous tenants. As these collocated tenants become competitors, if anyone try to break blockade between tenants, one occupant may access another tenant’s data. The cloud providers be supposed to be responsible for ensuring that one client will not be able to smash into another client data and applications. To conquer all this role-based access manage (RBAC) be supposed to be provided. A RBAC system has two phases in assigning a privilege to a user: in the first phase, the user will be assigned one or more roles; and in the subsequent phase, the roles will be checkered against the requested operations. In RBAC, permissions be able to be associated with roles rather than users, Users be able to obtain access rights by their roles, and they be able to be animatedly re-assigned or removed from roles without changing the permissions linked with roles. The number of roles will be much smaller than the figure of users. Roles may contain a hierarchical structure, and organization’s lines of authority will be reflected by it and responsibility. Such as CEO, CTO, and Vp are arranged in the diagram, where junior roles come into view at the underneath and older roles at the top. On the other hand, we do not how to describe roles for a specific application area. For instance, if its own access manages model is being tried to be complete for its application, from scrape and define role hierarchy and related policies will be difficult to start. Our strategy RBAC model to control MTA in cloud be supposed to be used. Intuitively, it be able to be found that ontology in sequence will be used to map users to roles and build up the role hierarchy. It proposes a reference ontology framework, ontology database be able to be searched in given a specific sphere of influence to find out relevant applicant role hierarchy templates, further get the matching policies linked with the templates to help with their own designs. We have to face several challenges to apply RBAC model in this problem. For example, how to define and manage the roles in a cloud? Given a specific domain, there are more than one ontology systems will be provided by different applications, how to compare the similarity between ontology systems? How to transfer one oncology to another? How to define the policies associated with different roles? These issues should be investigated and designs a reference ontology
framework using ontology, that good recommendations and ease the security management process can be generated for RBC.

- A reference ontology structure is future for access manages in a cloud to make easy the design of safety system and be able to decrease the difficulty of system plan and completion.

- This is exploits the possibility of RBAC to support MTA in a cloud. Ontology information will be used while building up the role hierarchy. Ontology transformation operations algorithm will be used to compare the similarity of different ontology. “Ontology Comparator” model can be implemented to compare the similarity of its own ontology with other candidates. He can refer to the most similar can be referred one with highest score, and reuse the policy template from the existing ones. On the further hand, if the tenant is totally new, a default ontology template, as well as policy template in his specific domain will be provided for reference. When a try is being done by tenant to access to a protected service/data, the various contextual information will be collected from both the environment and tenants. The Role Evaluator module will use context information to quantify these values and interacts with role databases and policy databases to the security level can be determined. According to the security level, role, and access policy, the Policy Controller the appropriate security services can be determined, includes granting, denying or revoking access.

4.1 Ontology for rbac

A. Roles is defined with Semantic Information

The first question is how we should define roles given a specific domain. Ontology, a conceptual structure, knowledge is contained in a domain and their relationships, useful and valuable information are used for cloud computing. A conceptualization of a area in conditions of idea and their family will be particular, which can be used to produce a frequently agreed expressions for information switch over with no
vagueness. In general, according to the semantic in a exact area, roles will be distinct as a mixture of the bureaucrat positions, job purpose, and etc. The user's daily duties will be stand for by purpose such as being a developer, testing wangle and etc. furthermore the managerial unit to which a user belongs is used as an access manage criterion for certain applications. All these data is defined and maintained in the human resources (HR) database. Thus, a RBAC system has an accurate image of the current organizational status and existing roles. A sample role definition is shown in Table I, e.g., CEO has a unique roleID = 3. Note that each employee can be assigned to one or more roles.

- Manage Roles Hierarchy with Ontology
In a domain, multiple possible role hierarchies are defined by ontology systems from different communities. Examples of possible role hierarchies in an IT company are shown in Existing research have discussed how to define an ontology in a specific domain, compare and integrate ontology systems between different communities. In practices, for a given domain, multiple reference ontology systems from various communities may in that domain. Solve Role Hierarchy using Ontology Trees Role hierarchies impose restrictions which can generate a simpler tree structure (i.e., a role may have one or more immediate ascendants, but is restricted to a single immediate descendant). To extend the traditional definition of trees for an ontology in a specific domain,

4.2 Windows Azure –Cloud Service From Microsoft
Windows Azure is the cloud service of Microsoft. It comes with high security and easy access with lower price rates. Following are the features of the Microsoft Azure. Getting started with development on Windows Azure The WindowsAzure.com Developer Center provides tutorials, downloads, and how-to guides that help we get started developing on Windows Azure.

- Windows Azure technical documentation
In this section we have outlined the MSDN documentation available for Windows Azure. In which it include Local development environment which state that the Windows Azure SDKs for .NET, Node.js, Java, and PHP provide common tools and
resources that we use to package, test and deploy our application. The Windows Azure SDK for .NET includes the Windows Azure Tools for Microsoft Visual Studio, which extends Visual Studio to enable the creation, building, packaging, running, and debugging of scalable web applications and services on Windows Azure. The following learning resources are available:

- Windows Azure SDK Tools
- Windows Azure Tools for Visual Studio
- Management Portal

The Windows Azure Management Portal provides access to hosted service deployment and management tasks as well as at-a-glance status information that lets us know the overall health of our deployments and accounts. The Management Portal organizes the components of our Windows Azure deployments with constantly refreshed information that is easy to discover and understand.

- To learn more about the portal, see The Management Portal.
- To learn about managing hosted services, see Managing Hosted Services in Windows Azure.

- Applications – Cloud Services, Web Sites, and Virtual Machines
  We can run applications on Windows Azure using Windows Azure Web Sites, Cloud Services (formerly hosted services), and Virtual Machines (which support IaaS)
  - Web Sites
  - Cloud Services
  - Virtual Machines

- Data management services
  The Windows Azure Table, Blob, and SQL Database services provide storage for binary and text data, messages, structured data and relational data. Benefits include manageability, high availability, high scalability, and a familiar development model. We can also use SQL Data Sync to synchronize relational data to other SQL Database instances or to on-premises SQL Server databases. The following learning resources are available:
• Blobs, Queues, and Tables
• SQL Database
• SQL Data Sync

1. **Identity - Windows Azure Active Directory Access Control**

   Access Control (formerly ACS) is a cloud-based service that provides an easy way of authenticating and authorizing users to gain access to our web applications and services while allowing the features of authentication and authorization to be factored out of our code.

   ➢ **Messaging – Service Bus**

   Windows Azure Service Bus provides a hosted, secure, and widely available infrastructure for widespread communication, large-scale event distribution, naming, and service publishing. Service Bus provides connectivity options for Windows Communication Foundation (WCF) and other service endpoints, which includes REST endpoints. To learn about integrating Service Bus into applications, review the EAI and EDI labs. These labs provide common integration capabilities (e.g. bridges, transforms, B2B messaging) using Windows Azure Service Bus.

   ➢ **Networking – Connect**

   With Windows Azure Connect, we can use a simple user interface to configure IPsec protected connections between individual computers or virtual machines in the network of our organization, and role instances running in Windows Azure.

   ➢ **Networking – Virtual Network**

   With Windows Azure Virtual Network, we can create secure site-to-site connectivity as well as protected private virtual networks in the cloud. We can specify the address space that will be used for both our virtual network and for the virtual network gateway that we create. Additionally, new name resolution features allow us to connect directly to role instances and to virtual machines by hostname. These features allow us to use Windows Azure as we would a branch office, or as a protected private virtual network in the cloud.
Networking – Traffic Manager
Windows Azure Traffic Manager allows us to control the distribution of user traffic to Windows Azure applications (Web Sites, Cloud Services, Virtual Machines). The applications can be running in the same data center or in different centers across the world. Traffic Manager works by applying an intelligent policy engine to the Domain Name Service (DNS) queries on our domain names.

Performance - Content Delivery Network (CDN)
The Windows Azure Content Delivery Network (CDN) caches Windows Azure blobs and the static content output of compute instances at strategically placed locations to provide maximum bandwidth for delivering content to users.

Performance - Caching
Windows Azure Caching enables to easily provision a cache in the cloud to be used from any applications or services that could benefit from caching. This includes a very common scenario of session state and output caching in ASP.NET. Caching increases performance by temporarily storing information from other backend sources.

Performance – Media Services
Windows Azure Media Services form an extensible cloud-based platform that enables developers to build scalable media management and delivery applications.

Business Analytics – SQL Reporting
Windows Azure SQL Reporting is a cloud-based reporting service for Windows Azure built on SQL Server Reporting Services technologies. SQL Reporting provides many benefits, including rapid provisioning, cost-effective scalability, high availability, reduced management overhead, and secure access for viewing and managing reports.

Marketplace – Datamarket and Applications
The Windows Azure Marketplace provides a place to buy or sell Windows Azure applications and premium data subscriptions.
To achieve the secure and scalable prepared copy in genuine time banking system using cloud computing, we have design the following algorithms with their specific objectives:

### 4.3 Algorithm 1:

**Objective:**
- To Generating XML file and filling up the data on cloud(AZURE) according to the XML
- SAXM Meta Xml Model
- Algorithm Specifications: Counting the total no of fields generated by user, generating the xml accordingly.

**Algo formula:**

```java
Int counter =0;
Int k=1;//for the fileds specified by user;
If(model.Is Selected ) // If the client has selected the specific model for his entry
{// to detect the rbac(role based access control )
For(int i=0;i<counterfiled;i++)// this loop will check the no of access provided to each user
{if(counterfield.checked==true)//the the displayed role in check box is true
    Rbsc rbc=new rbac(); // created a new object for the rbac class defined in the dll
    Rbc.roles.add(counterfiled.text);//add role to the rbc class object
    If(user.confirms.rbc.roles.added==true)
    {Put (xml.schema.action);//generates the tag according to the script
     }
    }
Else
    {Move.next(); // check the next filed ........
    }
For(int i=0;i<filedcount;i++)
{for(k!=null)
```
Draw(xmlschema.xml.rbc()); // To generate the schema of Xml
Xmlschema.rbc.fieldcount=filedcount; // Total no of tags in XML =Total no of fields
selected by user
Xmlschema.xmltag=new xmltag("<"+filedname+">"));
Xmlschema.xmltag=new xmltag("<"+filedname+"/>"));
Xmlschema.show(); // it shows we the generated XML
Put.Azure(spcefilled.rest.databaseschema);//to put the schema on Azure
Exit
Count++; //To increase the counter
}

} Else
{
This .close(); //to close current schema ;
Goto whileback; // to return to main program
}

}

4.4 Algorithm 2

- **Objective** :
  - Generate XML file the front end users
  - **Steps** : We have to generate the XML in such a format that it includes the
tables and roles and the No of tables are getting defined by the user
himself

Algo :
start countertb ;// To count the no of tables
start counterfields;
START counterrole // To count no of roles
IMPORT TO XML START TAG("<XML>")
END TAG ("</XML>")
LOOP
RUN TILL
STARTINGVALUE < = USERAREAVALUE
\{ 
Generate.tag("<table">).xml(name);
Countertb ++;
Counterfiled=USERDATA.MAX.DATA;
Countefiled ++;
Loopwhileback (i!=max)
GENERATE OPENING TABLE TAG(<"tbname">).xml(name).roottag("<table>");
GENERATE CLOSING TABLE TAG(<"/tbname">)(xmlname) roottag("<table>");
MERGEXML (name).roottag("<tbname">).data(boxdata); // From here whatever the
data will be put from drop box will be inserted into the root tag tbname;
While(!counterfield=max)
Generate.tag("<field">)(xmlname) roottag("<tbname>");
Generate.tag("</field">)(xmlname) roottag("<tbname>");
Merge.xml(name).roottag("<field">).data(boxdata);
} 
START COUNTERROLE;
GENERATE ROLES TAG ("<roles>")
GENERATE LOOP 
RUN TILL COUNTERROLE!=MAX 
GENRATE Tag ("<rolename>")
GENRATE Tag ("</rolename>")
GENERATE Tag("<AccessRights>”).BaseTag("<RoleName.”);
GENERATE Tag(”</AccessRights>”).BaseTag("<RoleName.”);
LOOP RUN TILL FIELDCOUT!=MAX PER TABLE 
GENERATE Tag(“,<RIGHT1><TABLE NAME ><FIELD NAME><READ><ALLOWED>VALUE from 
USER</ALLOWED></READ></FIELD NAME></TABLE NAME ><RIGHT1> 
<RIGHT 2><TABLE NAME ><FIELD NAME><WRITE><ALLOWED>VALUE from 
USER</ALLOWED></WRITE></FIELD NAME></TABLE NAME ><RIGHT 2>
4.5 Algorithm 3

- **Objective**: To put the file into the Azure cloud from the XML File

**ALGO**:

```plaintext
SET.Microsoft.Cloud(Azure.Cloud.DomainName(“Name”));
Import.Type=ImportType.Xml;
Draw.Into.DomainName(“Name”).Xml;// To draw a basic XML Lawet in Azure
Int countertable=0;
Int tagcounter=0;
Countertable=XML.FileName(“xmlfilename”.tagcount);
XSLT:foreach(int x in tagcount)
{
  Puttoazure(tag.basic.element);// to put the basic element of the tag
  Tagcounter++;
  If(!found.lastelement)
  {
    Puttoazure(tag.child.element.name)//to put the child element name
  }
  Else
  {
    Goto xslt;
  }
If(tagcounter==tagcounter.max)
{
  Exit.Azure.Cloud(save.saveas.Azure.sqlilitefile);
}
```
On top of talk about algorithms be able to be association with the banking system and it is able to create the safe and Scalable prepared replica in actual time banking system by means of cloud Computing.

4.6 CODE:

4.6.1 Code without Azure
Following code has been written to implement the security without implementing the azure

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Web;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Xml;

public partial class Default3 : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection();
    static  String[] fields = new String[100];
    static  String[] datatypes = new String[100];
    static  String[] size = new String[100];
    static int tablecounter = 0;
static String tablename = "";
static int f = 0;
static int d = 0;
static int s = 0;
static String finalquery = "";
static String thequery = "";
static String myfinalquery = "";
static String[] alltables = new String[100];
protected void Page_Load(object sender, EventArgs e)
{
    ButtonCONT.Visible = false;
    Labelnotification.Visible = false;
    CheckBoxList1.Visible = false;
    con.ConnectionString =
    ConfigurationManager.ConnectionStrings["sau"].ConnectionString;
    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }

    //fillrole();

    // Response.Write(myfinalquery.ToString());
    if (Page.IsPostBack == false)
    {
        filltable();
        fillrole();
        try
        {
            thequery = "";
            myfinalquery = "";
            tablecounter = 0;
        }
f = 0; d = 0; s = 0;

Label2.Text = "You have opted to draw " + Session["notables"].ToString() + " tables ";

if (con.State == ConnectionState.Closed)
{
    con.Open();
}

catch
{
    Label2.Text = "You have opted to draw" + "0 tables";
}

if (Page.IsPostBack == false)
{
    Label4.Text = "welcome to table generation ";
    DropDownList1.Items.Add("--select--");
    DropDownList2.Items.Add("--select--");
    LinkButton3.Visible = false;

    DropDownList1.Items.Add("int");
    DropDownList1.Items.Add("bigint");
    DropDownList1.Items.Add("varchar");
    DropDownList1.Items.Add("nchar");

    for (int i = 50; i <= 500; i = i + 50)
    {
        DropDownList2.Items.Add(i.ToString());
    }

}
private void fillrole()
{

    //if (Page.IsPostBack == false)
    //{
    //    SqlCommand cmd1 = new SqlCommand("select * from roles", con);
    //    SqlDataReader dr1 = cmd1.ExecuteReader();
    //    //if (dr1.HasRows)
    //    //{
    //    //    while (dr1.Read())
    //    //    {
    //    //        CheckBoxList1.Items.Add(dr1[0].ToString());
    //    //    //}
    //    //}
    //    //dr1.Close();
    //cmd1.Dispose();
    if (DropDownListROLETYPE.Items.Count > 0)
    {
        DropDownListROLETYPE.Items.Clear();
    }
    DropDownListROLETYPE.Items.Add("--select--");
    try
    {
        SqlCommand cmd = new SqlCommand("select * from roles", con);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.HasRows)
        {
            while (dr.Read())
            {

            }
        }
    }
}
Chapter 4 Observation, Result and Discussion

```csharp
{  
    DropDownListROLETYPE.Items.Add(dr[0].ToString());
}
}
dr.Close();
cmd.Dispose();
}  
catch (Exception err)
{
    Response.Write("no data present for roles ");
}
}
//
protected void LinkButton2_Click(object sender, EventArgs e)
{
    LinkButton2.Text = "insert another field";

tablename = TextBox1.Text;
if (TextBox1.Text != "]")
{
    if (TextBox2.Text != "]")
    {
        fields[f] = TextBox2.Text;
        datatypes[d] = DropDownList1.SelectedItem.Text;
        if (DropDownList2.SelectedIndex != 0)
        {
            size[s] = DropDownList2.SelectedItem.Text;
        }
        else
        {
            size[s] = "]";
        }
    }
    f++;
```
s++;  
d++;  
}
else  
{  
    Label4.Text = "insert field name ";  
}
}
else  
{  
    Label4.Text = "insert table name ";  
}
TextBox2.Text = String.Empty;  
DropDownList1.SelectedIndex = 0;  
DropDownList2.SelectedIndex = 0;  
}
protected void LinkButton4_Click(object sender, EventArgs e)
{
    for (int i = 0; i < f; i++)
    {
        if (size[i].ToString() != "")
        {
            thequery += fields[i].ToString() + " " + datatypes[i].ToString() + "(" + size[i].ToString() + ")" + ",";  
        }
        else  
        {
            thequery += fields[i].ToString() + " " + datatypes[i].ToString() + ",";  
        }
    }
}
try {
    thequery = thequery.Substring(0, (thequery.Length) - 1);
    Label4.Text = "table name :=> " + tablename + " the fields will be : " +
        "fields :=> " + thequery;
}
catch (Exception err)
{
    Response.Write("INVALID ATTEMPT TO MINIMISE");
}

f = 0;
s = 0;
d = 0;
LinkButton3.Visible = true;
//Label4.Text = thequery.ToString();

protected void LinkButton1_Click(object sender, EventArgs e)
{
    DropDownList2.Visible = true;
}
protected void LinkButton5_Click(object sender, EventArgs e)
{
    DropDownList2.SelectedIndex = 0;
}
protected void LinkButton3_Click(object sender, EventArgs e)
{
    LinkButton3.Visible = false;
    TextBox1.Text = String.Empty;
    try
    {
        if (thequery != "")
{ finalquery = "create table " + tablename + "(" + thequery + ", roletype varchar(50)" + " + ")";
Label4.Text = finalquery;
alltables[tablecounter] = tablename;
tablecounter++;
SqlCommand cmd = new SqlCommand(finalquery, con);
cmd.ExecuteNonQuery();
cmd.Dispose();
tablename = ""
; d = 0;
f = 0;
s = 0;

Label4.Text = "table created successfully";
finalquery = ""
; thequery = ""
;
} else {
Label4.Text = "kindly generate a table structure";
}
}
catch {
Response.Write("table exists already");
};
}
protected void LinkButton6_Click(object sender, EventArgs e)
{
TextBox3.Text = String.Empty;
int mytablecouter;
Session["tablecounter"] = tablecounter;

String filename = ";

for (mytablecouter = 0; mytablecouter < tablecounter; mytablecouter++)
{
    myfinalquery += "select * from " + alltables[mytablecouter]+" " +";
}

myfinalquery = myfinalquery.Substring(0, myfinalquery.Length - 2);
//Response.Write(myfinalquery.ToString());
Session["finalquery"] = myfinalquery;
SqlDataAdapter adp1 = new SqlDataAdapter(myfinalquery ,
ConfigurationManager.ConnectionStrings["sau"].ConnectionString);
DataSet ds1 = new DataSet();
adp1.Fill(ds1);
if (ds1.Tables.Count > 0)
{
    for (int i = 0; i < ds1.Tables.Count; i++)
    {
        filename = "~/xml/test" + i + ".xml";
        ds1.WriteXmlSchema(Server.MapPath(filename));
    }
    XmlDocument doc = new XmlDocument();
doc.Load(Server.MapPath(filename));
    //TextBox3.Text = doc.InnerXml.ToString();
    doc.RemoveAll();
    ds1.Clear();
    
    XMLDocument doc = new XmlDocument();
doc.Load(Server.MapPath(filename));
    //TextBox3.Text = doc.InnerXml.ToString();
    doc.RemoveAll();
    ds1.Clear();
    
    //} the following line is to generate the roles cml
    SqlDataAdapter adp = new SqlDataAdapter("select * from tbaccess" ,
ConfigurationManager.ConnectionStrings["sau"].ConnectionString);
DataSet ds = new DataSet();
adp.Fill(ds);
String anotherfile = "";
anotherfile = "~/xml/roles99.xml";
ds.WriteXml(Server.MapPath(anotherfile));
/// MERGING THE TWO XML FILES
///

XmlDocument doc1 = new XmlDocument();
XmlNode cont1 = doc1.CreateElement("Container");
doc1.AppendChild(cont1);

// first document to merge
XmlDocument stuff = new XmlDocument();
stuff.Load( Server.MapPath( filename));
XmlNode imported =
    doc1.ImportNode(stuff.DocumentElement, true);
doc1.DocumentElement.AppendChild(imported);

// second document to merge
stuff.Load(Server .MapPath(anotherfile));
imported = doc1.ImportNode(stuff.DocumentElement, true);
doc1.DocumentElement.AppendChild(imported);

TextBox3.Text = doc1.Inne


doc1.RemoveAll();
ds.Clear();

/////////////////////////////////////////////////////////////////XML FILES MERGED

///

}
protected void LinkButton7_Click(object sender, EventArgs e)
{
    Response.Redirect("default6.aspx");
}
protected void Button1_Click(object sender, EventArgs e)
{
    SqlCommand cmdin = new SqlCommand("insert roles values (@role)", con);
    cmdin.Parameters.Add("@role", SqlDbType.VarChar, 50).Value = TextBox4.Text;
    cmdin.ExecuteNonQuery();
    cmdin.Dispose();
    Label5.Text = "ROLE INSERTED : " + TextBox4.Text;
    TextBox4.Text = String.Empty;
    fillrole();
}
protected void Button1_Click1(object sender, EventArgs e)
{
    Labelnotification.Text = " ";
    if (DropDownListROLETYPE.SelectedIndex != 0)
    {
        for (int i = 0; i < CheckBoxList1.Items.Count; i++)
        {
            if (CheckBoxList1.Items[i].Selected == true)
            {
                SqlCommand cmdcheckprev = new SqlCommand("select * from tbroles
where roletype=@type and access=@access", con);
                cmdcheckprev.Parameters.Add("@type", SqlDbType.VarChar, 50).Value = DropDownListROLETYPE.SelectedItem.Text;
                cmdcheckprev.Parameters.Add("@access", SqlDbType.VarChar, 50).Value = CheckBoxList1.Items[i].Text;
            }
        }
SqlDataReader drcheck = cmdcheckprev.ExecuteReader();
if (!drcheck.HasRows)
{
    drcheck.Close();
    cmdcheckprev.Dispose();
    SqlCommand cmd = new SqlCommand("insert tbroles values(@type, @access)", con);
    cmd.Parameters.Add("@access", SqlDbType.VarChar, 50).Value = CheckBoxList1.Items[i].Text;
    cmd.Parameters.Add("@type", SqlDbType.VarChar, 50).Value = DropDownListROLETYPE.SelectedItem.Text;
    cmd.ExecuteNonQuery();
    cmd.Dispose();
}
else
{
    drcheck.Close();
    cmdcheckprev.Dispose();

    Labelnotification.Text += " ACCESS OF " + CheckBoxList1.SelectedItem.Text + " IS ALREADY AVAILABLE WITH THE ROLE " + DropDownListROLETYPE.SelectedItem.Text + ", ";
}

for (int j = 0; j < CheckBoxList1.Items.Count; j++)
{
    if (CheckBoxList1.Items[j].Selected == true)
    {
        CheckBoxList1.Items[j].Selected = false;
    }
}
protected void LinkButton8_Click(object sender, EventArgs e)
{
    try
    {
        if (DropDownList3.Items.Count > 0)
        {
            DropDownList3.Items.Clear();
        }
        DropDownList3.Items.Add("--select--");
        SqlCommand cmd = new SqlCommand("select * from information_schema.tables", con);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.HasRows)
        {
            while (dr.Read())
            {
                DropDownList3.Items.Add(dr[2].ToString());
            }
        }
        dr.Close();
        cmd.Dispose();
    }
    catch (Exception err)
    {
    }
protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
{
    //CheckBoxList2.Visible = true;
    //String query = "";
    //try
    //{
    //    if (CheckBoxList2.Items.Count > 0)
    //    {
    //        CheckBoxList2.Items.Clear();
    //    }
    //}
    // query = "select * from information_schema.columns where table_name = " + 
    "" + DropDownList3.SelectedItem.Text + "";
    // SqlCommand cmd = new SqlCommand(query, con);
    // SqlDataReader dr = cmd.ExecuteReader();
    // if (dr.HasRows)
    // {
    //     // dr.Read();
    //     while (dr.Read())
    //     {
    //         CheckBoxList2.Items.Add(dr[3].ToString());
    //     }
    // }
    // dr.Close();
    // cmd.Dispose();
private void filltable()
{
    if (DropDownList3.Items.Count > 0)
    {
        DropDownList3.Items.Clear();
    }
    DropDownList3.Items.Add("--select--");
    SqlCommand cmd = new SqlCommand("select * from information_schema.tables", con);
    SqlDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows)
    {
        while (dr.Read())
        {
            DropDownList3.Items.Add(dr[2].ToString());
        }
    }
    dr.Close();
    cmd.Dispose();
}
protected void LinkButton10_Click(object sender, EventArgs e)
{
    //CheckBoxList2.Visible = true;
    String query = "";
    try
{  
    if (DropDownList4.Items.Count > 0)  
    {  
        DropDownList4.Items.Clear();  
    }  

    query = "select * from information_schema.columns where table_name = " + 
    """ + DropDownList3.SelectedItem.Text + """;  

    SqlCommand cmd = new SqlCommand(query, con);  
    SqlDataReader dr = cmd.ExecuteReader();  
    if (dr.HasRows)  
    {  
        //dr.Read();  
        while (dr.Read())  
        {  
            DropDownList4.Items.Add(dr[3].ToString());  
        }  
    }  
    dr.Close();  
    cmd.Dispose();  

}  
catch (Exception err)  
{  
    Response.Write(err.ToString());  
}  
}  
protected void LinkButton9_Click(object sender, EventArgs e)  
{  
    String table, col, role, type;
for (int i = 0; i < CheckBoxList3.Items.Count; i++)
{
    if (CheckBoxList3.Items[i].Selected == true)
    {
        SqlCommand cmdCheckPrev = new SqlCommand("select * from tbaccess where tablename=@name and columnname=@cname and roletype=@type and access=@access", con);
        cmdCheckPrev.Parameters.Add (@"@type", SqlDbType.VarChar, 50).Value = DropDownListROLETYPE.SelectedItem.Text;
        cmdCheckPrev.Parameters.Add (@"@access", SqlDbType.VarChar, 50).Value = CheckBoxList3.Items[i].Text;
        SqlDataReader drCheck = cmdCheckPrev.ExecuteReader();
        if (!drCheck.HasRows)
        {
            drCheck.Close();
            cmdCheckPrev.Dispose();
        }
        SqlCommand cmd = new SqlCommand("insert into tbaccess values(@tablename, @colname, @roletype, @access)", con);
        cmd.Parameters.Add (@"@roletype", SqlDbType.VarChar, 50).Value = DropDownListROLETYPE.SelectedItem.Text;
        cmd.Parameters.Add (@"@access", SqlDbType.VarChar, 50).Value = CheckBoxList3.Items[i].Text;
        cmd.ExecuteNonQuery();
        labelNote.Text = "ACCESS GRANTED ";
        cmd.Dispose();
    }
}
else
{
  drcheck.Close();
  cmdcheckprev.Dispose();
  }
}

//CheckBoxList3.Items.Clear();
// i have made this lines as comment today
//DropDownList4.SelectedIndex = 0;
//if (DropDownList4.Items.Count > 0)
//{
//  DropDownList4.Items.Clear();
//}
/////DropDownList3.SelectedIndex = 0;
/////DropDownListROLETYPE.SelectedIndex = 0;
}

protected void Button2_Click(object sender, EventArgs e)
{

}

4.6.2 Code With Azure:

Following code represent the security issues with the azure:-

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Web;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Xml;
using System.IO;

public partial class Default3 : System.Web.UI.Page
{
    static  String AZURESCRIPT = "";
    // StreamWriter sw = new StreamWriter();
    SqlConnection con = new SqlConnection();
    static  String[] fields = new String[100];
    static  String[] datatypes = new String[100];
    static  String[] size = new String[100];
    static int tablecounter = 0;
    static  String tablename = "";
    static  int f = 0;
    static  int d = 0;
    static  int s = 0;
    static int fsimilar = 0;
    static String finalquery = "";
    static  String thequery = "";
    static String myfinalquery = "";
    static String[] alltables = new String[100];
    protected void Page_Load(object sender, EventArgs e)
if (Page.IsPostBack == false )
{
    AZURESCRIPT += "CREATE DATABASE MYCLOUD ";
    AZURESCRIPT += Environment.NewLine;
    AZURESCRIPT += " GO ";
    AZURESCRIPT += Environment.NewLine;
    AZURESCRIPT += "USE MYCLOUD ";
    AZURESCRIPT += Environment.NewLine;
    AZURESCRIPT += " GO ";
    AZURESCRIPT += Environment.NewLine;
    // AZURESCRIPT +="SET ANSI_NULLS ON  GO   SET QUOTED_IDENTIFIER ON  GO IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[tbroles]') AND type in (N'U')) BEGIN";
}

ButtonCONT.Visible = false;
Labelnotification.Visible = false;
CheckBoxList1.Visible = false;
try
{
    con.ConnectionString = ConfigurationManager.ConnectionStrings["sau"].ConnectionString;
    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }
} //fillrole();
// Response.Write(myfinalquery.ToString());
if (Page.IsPostBack == false)
{
    filltable();
    fillrole();
    try
    {
        thequery = "";
        myfinalquery = "";
tablecounter = 0;
f = 0; d = 0; s = 0;
Label2.Text = "You have opted to draw " + Session["notables"].ToString() + " tables ";
if (con.State == ConnectionState.Closed)
{
    con.Open();
}
}

catch
{
    Label2.Text = " You have opted to draw" + "0 tables";
}
}
if (Page.IsPostBack == false)
{
    Label4.Text = "welcome to table generation ";
    DropDownList1.Items.Add("--select--");
    DropDownList2.Items.Add("--select--");
    LinkButton3.Visible = false;
    DropDownList1.Items.Add("int");
    DropDownList1.Items.Add("bigint");
    DropDownList1.Items.Add("varchar");
    DropDownList1.Items.Add("nchar");
    for (int i = 50; i <= 500; i = i + 50)
    {
        DropDownList2.Items.Add(i.ToString());
    }
}
}
private void fillrole()
{
    //if (Page.IsPostBack == false)
    //{ 
        //SqlCommand cmd1 = new SqlCommand("select * from roles", con);
        //SqlDataReader dr1 = cmd1.ExecuteReader();
        //if (dr1.HasRows)
        //{ 
            //while (dr1.Read())
            //  {
                //    CheckBoxList1.Items.Add(dr1[0].ToString());
            // }
        //} 
        //dr1.Close();
        //cmd1.Dispose();
    
    if (DropDownListROLETYPE.Items.Count > 0)
    {
        DropDownListROLETYPE.Items.Clear();
    }
    DropDownListROLETYPE.Items.Add("--select--");
    try
    {
        SqlCommand cmd = new SqlCommand("select * from roles", con);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.HasRows)
        {
            while (dr.Read())
            {
                DropDownListROLETYPE.Items.Add(dr[0].ToString());
            }
        }
    }
    dr.Close();
    cmd.Dispose();
catch (Exception err)
{
    Response.Write("no data present for roles ");
}
}
//}
protected void LinkButton2_Click(object sender, EventArgs e)
{
    LinkButton2.Text = "insert another field";
    tablename = TextBox1.Text;
    if (TextBox1.Text != "")
    {
        if (TextBox2.Text != "")
        {
            fields[f] = TextBox2.Text;
            datatypes[d] = DropDownList1.SelectedItem.Text;
            if (DropDownList2.SelectedIndex != 0)
            {
                size[s] = DropDownList2.SelectedItem.Text;
            }
            else
            {
                size[s] = ";
            }
        }
        else
        {
            size[s] = ";
        }
    }
    f++;
    fsimilar++;
    s++;
    d++;
}
else
{

Label4.Text = "insert field name ";
}
}

else
{
    Label4.Text = "insert table name ";
}
TextBox2.Text = String.Empty;
DropDownList1.SelectedIndex = 0;
DropDownList2.SelectedIndex = 0;
}
protected void LinkButton4_Click(object sender, EventArgs e)
{
    for (int i = 0; i < f; i++)
    {
        if (size[i].ToString() != "")
        {
            thequery += fields[i].ToString() + " " + datatypes[i].ToString() + "(" + size[i].ToString() + ")" + ",";
        }
        else
        {
            thequery += fields[i].ToString() + " " + datatypes[i].ToString() + ",";
        }
    }
    try
    {
        thequery = thequery.Substring(0, (thequery.Length) - 1);
        Label4.Text = "table name :=> " + tablename + " the fields will be : " + thequery;
    }
    catch (Exception err)
Response.Write("INVALID ATTEMPT TO MINIMISE");
}
f = 0;
s = 0;
d = 0;
LinkButton3.Visible = true;
//Label4.Text = thequery.ToString();
}
protected void LinkButton1_Click(object sender, EventArgs e)
{
    DropDownList2.Visible = true;
}
protected void LinkButton5_Click(object sender, EventArgs e)
{
    DropDownList2.SelectedIndex = 0;
}
protected void LinkButton3_Click(object sender, EventArgs e)
{
    LinkButton3.Visible = false;
    TextBox1.Text = String.Empty;
    try
    {
        if (thequery != "")
        {
            AZURESCRIPT += " IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'\[dbo\].[" + tablename + "]') AND type in (N'U')) BEGIN";
            AZURESCRIPT += " CREATE TABLE [dbo]. [" + tablename + "](";
            for (int i = 0; i < fsimilar; i++)
            {
                if (size[i].ToString() != "")
                {
//thequery += fields[i].ToString() + " " + datatypes[i].ToString() + "(" + size[i].ToString() + ")" + ",";

AZURESCRIPT += " [" + fields[i].ToString() + "]" + " [" + datatypes[i].ToString() + "]" + "(" + size[i] + ")" + " NULL , ";
} else {

//thequery += fields[i].ToString() + " " + datatypes[i].ToString() + ",";

AZURESCRIPT += " [" + fields[i].ToString() + "]" + " [" + datatypes[i].ToString() + "]" + " NULL , ";
} }

AZURESCRIPT = AZURESCRIPT.Substring(0, AZURESCRIPT.Length - 2);

AZURESCRIPT += ") ON [PRIMARY] END ";
AZURESCRIPT += " SET ANSI_NULLS ON SET QUOTED_IDENTIFIER ON ";

Session["azscript"] = AZURESCRIPT;
fsimilar = 0;
finalquery = "create table " + tablename + "( " + thequery + ", roletype varchar(50)" + ", " + ");
Label4.Text = finalquery;
alltables[tablecounter] = tablename;
tablecounter++;
SqlCommand cmd = new SqlCommand(finalquery, con);
cmd.ExecuteNonQuery();
cmd.Dispose();
tablename = "";
d = 0;
f = 0;
s = 0;
// HERE WE NEED TO GENERATE THE SCRIPT FOR THE AZURE
Label4.Text = "table created successfully";
finalquery = "";
thequery = "";
}
else
{
    Label4.Text = "kindly generate a table structure";
}
}
}
catch
{
    Response.Write("table exists already");
}
}
protected void LinkButton6_Click(object sender, EventArgs e)
{
    TextBox3.Text = String.Empty;
    int mytablecouter;
    Session["tablecounter"] = tablecounter;
    String filename = "";
    for (mytablecouter = 0; mytablecouter < tablecounter; mytablecouter++)
    {
        myfinalquery += "select * from " + alltables[mytablecouter]+" ";
    }
    myfinalquery = myfinalquery.Substring(0, myfinalquery.Length - 2);
    //Response.Write(myfinalquery.ToString());
    Session["finalquery"] = myfinalquery;
    SqlDataAdapter adp1 = new SqlDataAdapter(myfinalquery, ConfigurationManager.ConnectionStrings["sau"].ConnectionString);
    DataSet ds1 = new DataSet();
adp1.Fill(ds1);
    if (ds1.Tables.Count > 0)
for (int i = 0; i < ds1.Tables.Count; i++)
{
    filename = "~/xml/test" + i + ".xml";
    ds1.WriteXmlSchema(Server.MapPath(filename));

    XmlDocument doc = new XmlDocument();
    doc.Load(Server.MapPath(filename));
    //TextBox3.Text = doc.InnerXml.ToString();
    doc.RemoveAll();
    ds1.Clear();

    SqlDataAdapter adp = new SqlDataAdapter("select * from tbaccess",
    ConfigurationManager.ConnectionStrings["sau"].ConnectionString);
    DataSet ds = new DataSet();
adp.Fill(ds);
    anotherfile = "~/xml/roles99.xml";
    ds.WriteXml(Server.MapPath(anotherfile));

    XmlDocument doc1 = new XmlDocument();
    XmlNode cont1 = doc1.CreateElement("Container");
    doc1.AppendChild(cont1);

    // first document to merge
    XmlDocument stuff = new XmlDocument();
    stuff.Load(Server.MapPath(filename));
    XmlNode imported =
        doc1.ImportNode(stuff.DocumentElement, true);
    doc1.DocumentElement.AppendChild(imported);

    // second document to merge
Observation, Result and Discussion

stuff.Load(Server.MapPath(anotherfile));
imported = doc1.ImportNode(stuff.DocumentElement, true);
doc1.DocumentElement.AppendChild(imported);
TextBox3.Text = doc1.InnerXml;

doc1.RemoveAll();
ds.Clear();

/XML FILES MERGED

protected void LinkButton7_Click(object sender, EventArgs e)
{
    Response.Redirect("default6.aspx");
}
protected void Button1_Click(object sender, EventArgs e)
{
    SqlCommand cmdin = new SqlCommand("insert roles values (@role)", con);
    cmdin.Parameters.Add("@role", SqlDbType.VarChar, 50).Value = TextBox4.Text;
    cmdin.ExecuteNonQuery();
    cmdin.Dispose();
    Label5.Text = "ROLE INSERTED : " + TextBox4.Text;
    fillrole();
}
protected void Button1_Click1(object sender, EventArgs e)
{
    Labelnotification.Text = " ";
    if (DropDownListROLETYPE.SelectedIndex != 0)
    {
        for (int i = 0; i < CheckBoxList1.Items.Count; i++)
if (CheckBoxList1.Items[i].Selected == true)
{
    SqlCommand cmdcheckprev = new SqlCommand("select * from tbroles where roletype=@type and access=@access", con);
    cmdcheckprev.Parameters.Add("@type", SqlDbType.VarChar, 50).Value = DropDownListROLETYPE.SelectedItem.Text;
    cmdcheckprev.Parameters.Add("@access", SqlDbType.VarChar, 50).Value = CheckBoxList1.Items[i].Text;
    SqlDataReader drcheck = cmdcheckprev.ExecuteReader();
    if (!drcheck.HasRows)
    {
        drcheck.Close();
        cmdcheckprev.Dispose();
        SqlCommand cmd = new SqlCommand("insert tbroles values(@type,@access)", con);
        cmd.Parameters.Add("@access", SqlDbType.VarChar, 50).Value = CheckBoxList1.Items[i].Text;
        cmd.Parameters.Add("@type", SqlDbType.VarChar, 50).Value = DropDownListROLETYPE.SelectedItem.Text;
        cmd.ExecuteNonQuery();
        cmd.Dispose();
    }
    else
    {
        drcheck.Close();
        cmdcheckprev.Dispose();
        Labelnotification.Text += " ACCESS OF " + CheckBoxList1.SelectedItem.Text + " IS ALREADY AVAILABLE WITH THE ROLE " + DropDownListROLETYPE.SelectedItem.Text + ", ";
    }
}
for (int j = 0; j < CheckBoxList1.Items.Count; j++)
{
    if (CheckBoxList1.Items[j].Selected == true)
    {
        CheckBoxList1.Items[j].Selected = false;
    }
}
else
{
    Response.Write("Kindly select a value from the drop down");
}

protected void LinkButton8_Click(object sender, EventArgs e)
{
    try
    {
        if (DropDownList3.Items.Count > 0)
        {
            DropDownList3.Items.Clear();
        }
        DropDownList3.Items.Add("--select--");
        SqlCommand cmd = new SqlCommand("select * from information_schema.tables", con);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.HasRows)
        {
            while (dr.Read())
            {
                DropDownList3.Items.Add(dr[2].ToString());
            }
        }
    }
}
dr.Close();
cmd.Dispose();
}
catch (Exception err)
{

}
}
protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
{
    //CheckBoxList2.Visible = true;
    //String query = "";
    //try
    //{
    //    if (CheckBoxList2.Items.Count > 0)
    //    {
    //        CheckBoxList2.Items.Clear();
    //    }
    //}
    // query = "select * from information_schema.columns where table_name = " + "" + DropDownList3.SelectedItem.Text + ";"

    // SqlCommand cmd = new SqlCommand(query, con);
    // SqlDataReader dr = cmd.ExecuteReader();
    // if (dr.HasRows)
    // {
    //     //dr.Read();
    //     while (dr.Read())
    //     {
    //         CheckBoxList2.Items.Add(dr[3].ToString());
    //     }
    // }
    // }
//
private void filltable()
{
    if (DropDownList3.Items.Count > 0)
    {
        DropDownList3.Items.Clear();
    }
    DropDownList3.Items.Add("--select--");
    SqlCommand cmd = new SqlCommand("select * from information_schema.tables", con);
    SqlDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows)
    {
        while (dr.Read())
        {
            DropDownList3.Items.Add(dr[2].ToString());
        }
    }
    dr.Close();
    cmd.Dispose();
}
protected void LinkButton10_Click(object sender, EventArgs e)
{
    //CheckBoxList2.Visible = true;
    String query = "";
    try
if (DropDownList4.Items.Count > 0)
{
    DropDownList4.Items.Clear();
}
query = "select * from information_schema.columns where table_name = " +
"" + DropDownList3.SelectedItem.Text + ";
SqlCommand cmd = new SqlCommand(query, con);
SqlDataReader dr = cmd.ExecuteReader();
if (dr.HasRows)
{
    //dr.Read ();
    while (dr.Read())
    {
        DropDownList4.Items.Add(dr[3].ToString());
    }
}
    dr.Close();
    cmd.Dispose();
}
catch (Exception err)
{
    Response.Write(err.ToString());
}
protected void LinkButton9_Click(object sender, EventArgs e)
{
    String table, col, role, type;
    for (int i = 0; i < CheckBoxList3.Items.Count; i++)
    {
        if (CheckBoxList3.Items[i].Selected == true)
SqlCommand cmdcheckprev = new SqlCommand("select * from tbaccess where tablename=@name and columnname=@cname and roletype=@type and access=@access", con);


cmdcheckprev.Parameters.Add("@type", SqlDbType.VarChar, 50).Value = DropDownListROLETYPE.SelectedItem.Text;

cmdcheckprev.Parameters.Add("@access", SqlDbType.VarChar, 50).Value = CheckBoxList3.Items[i].Text;

SqlDataReader drcheck = cmdcheckprev.ExecuteReader();

if (!drcheck.HasRows)
{
    drcheck.Close();
    cmdcheckprev.Dispose();
    SqlCommand cmd = new SqlCommand("insert into tbaccess values(@tablename,@colname,@roletype,@access)", con);

    cmd.Parameters.Add("@roletype", SqlDbType.VarChar, 50).Value = DropDownListROLETYPE.SelectedItem.Text;
    cmd.Parameters.Add("@access", SqlDbType.VarChar, 50).Value = CheckBoxList3.Items[i].Text;

    cmd.ExecuteNonQuery();
    labelnote.Text = "ACCESS GRANTED ";
    cmd.Dispose();
}
else
{
    drcheck.Close();
    cmdcheckprev.Dispose();
}
Chapter 4 Observation, Result and Discussion


//CheckBoxList3.Items.Clear();
// i have made this lines as comment today
//DropDownList4.SelectedIndex = 0;
//if (DropDownList4.Items.Count > 0)
//{
//    DropDownList4.Items.Clear();
//}
/////DropDownList3.SelectedIndex = 0;
//DropDownListROLETYPETXT.SelectedIndex = 0;

protected void Button2_Click(object sender, EventArgs e)
{
    String sb = Session["azscript"].ToString();
    using (StreamWriter outfile = new
        StreamWriter(Server.MapPath("AllTxtFiles.sql")))
    {
        outfile.Write(sb.ToString());
    }
    // String FileName = "filename";
    // String FilePath = "";
    response.ClearContent();
    response.Clear();
    response.ContentType = "text/plain";
response.AddHeader("Content-Disposition", "attachment; filename=" + @"\AllTxtFiles.sql" + ",");
response.TransmitFile(@"AllTxtFiles.sql");
response.Flush();
response.End();
}