Azure SQL Administration

ISM 6218

Due on October 22th

The Avengers Team

"We will avenge every problem on our way"

Aitemir Yeskenov (Team Lead)

Nagarjuna Kanneganti

Sai Suraj Argula

Vinay Kumar Reddy Baradi

Table of Contents

Business Process Supported

As part of this assignment, each team member from our group created a server on Azure as well as deployed a database to the server. Every team member created credentials for other 3 team members in their system and made accounts restricted based on the requirements.

Requirements Described

Goals:

1. Create an Azure SQL Server Instance with user accounts.

2. Deploy a local database to Azure instance.

Delivery Requirements:

1. Each individual student must create an Azure instance and create login account for each member of the team on the server and user account on the database.

2. Set user permission so that one of the members has access to one and only one schema.

3a. Set user permission so that one of the members only has access to views.

3b. Set the permission to hide columns instead (see links to walkthrough of this task).

4. Set permission so that one member (other than the admin owner of the account) has priviledge to create, alter, drop tables.

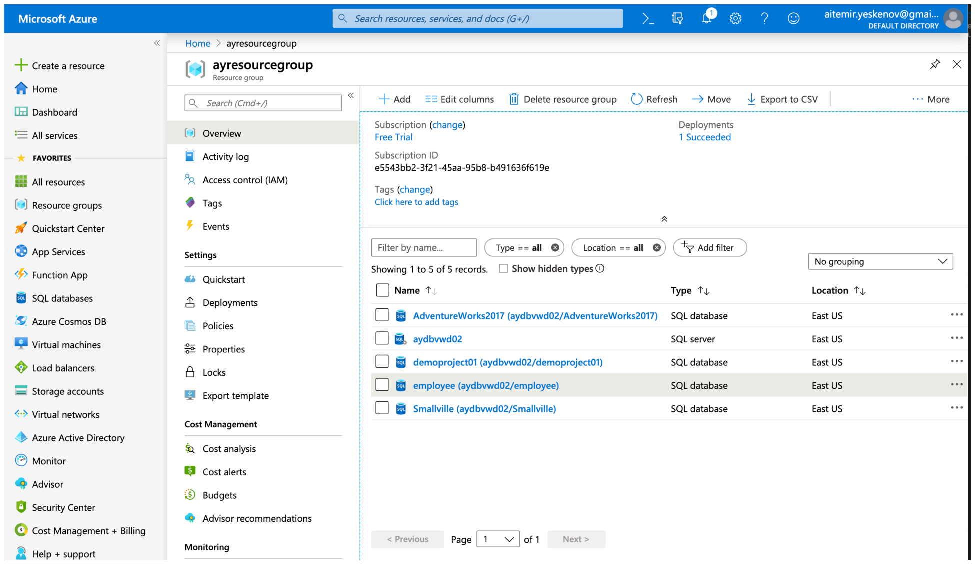
5. Deploy any local DB of your choice to the Azure instance and demonstrate tasks 1 - 4.

5.1. If your Azure instance does not allow for this, you can create the same demonstrations using your local instance.

6. There will need to be 4 reports in this Lab Notebook. One for each Instance created.

Aitemir's server:

For this assignment, each of us created a server on Azure:



Subscription (change)
Free Trial

Subscription ID
e5543bb2-3f21-45aa-95b8-b491636f619e

Tags (change)
Click here to add tags

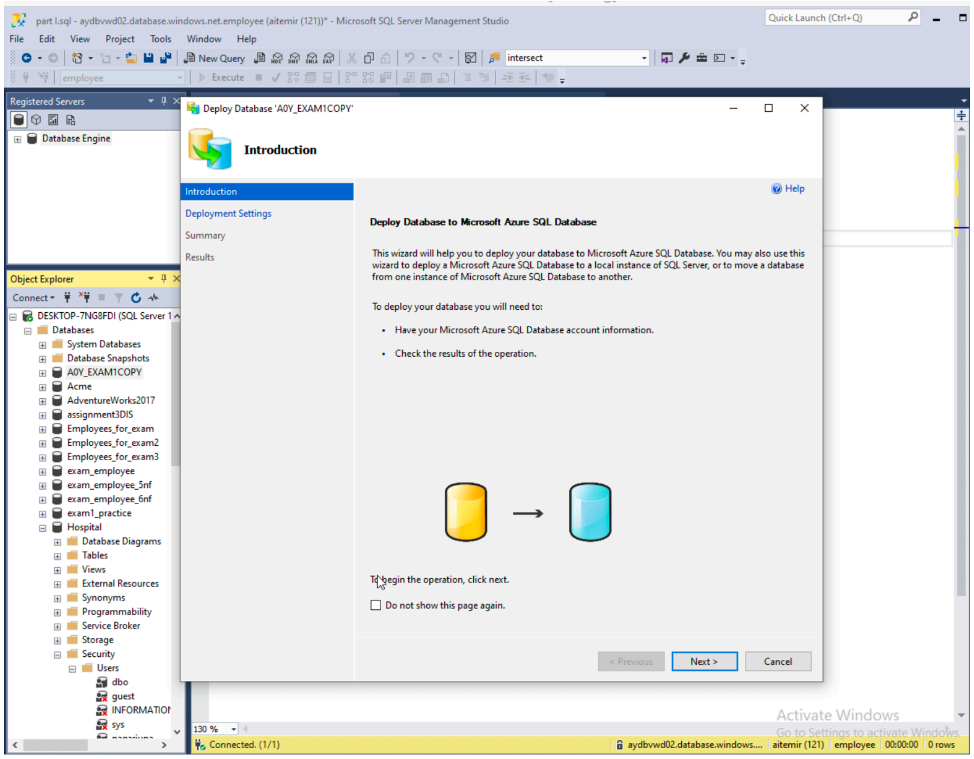
Deployments
1 Succeeded

Showing 1 to 5 of 5 records.

Name	Type	Location
AdventureWorks2017 (aydbvwd02/AdventureWorks2017)	SQL database	East US
aydbvwd02	SQL server	East US
demoproject01 (aydbvwd02/demoproject01)	SQL database	East US
employee (aydbvwd02/employee)	SQL database	East US
Smallville (aydbvwd02/Smallville)	SQL database	East US

As you can see, my SQL server is aydbvwd02.

Deployment of a local database to the Azure server is the next step. I used my first exam's db and renamed it as "employee" in the azure server.



Deploy Database to Microsoft Azure SQL Database

This wizard will help you to deploy your database to Microsoft Azure SQL Database. You may also use this wizard to deploy a Microsoft Azure SQL Database to a local instance of SQL Server, or to move a database from one instance of Microsoft Azure SQL Database to another.

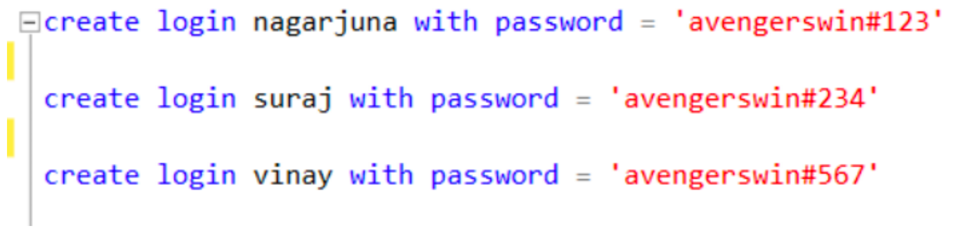
To deploy your database you will need to:

- Have your Microsoft Azure SQL Database account information.
- Check the results of the operation.

To begin the operation, click next.

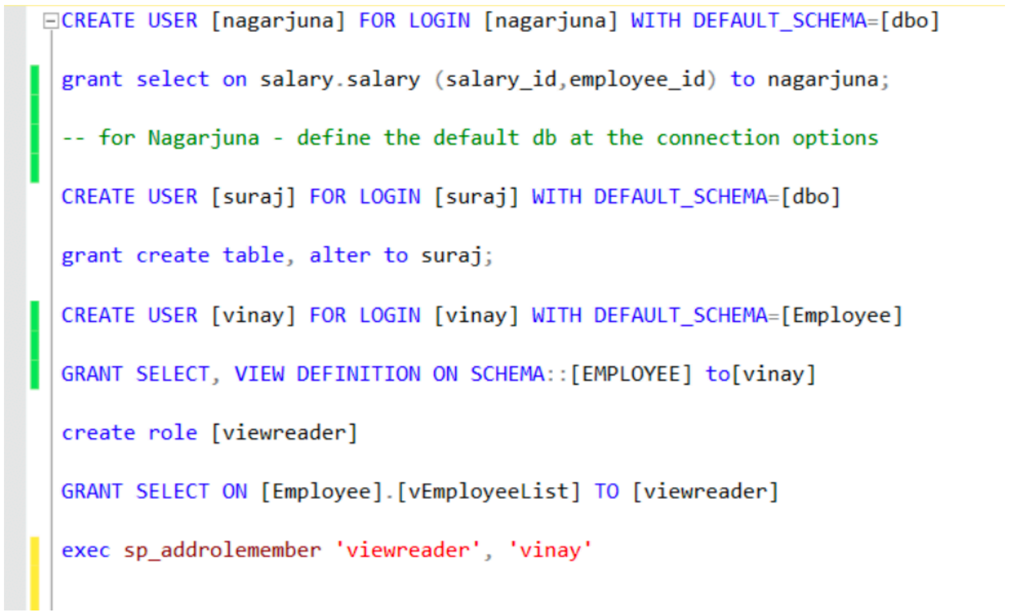
☐ Do not show this page again.

So, I created logins for all my teammates at first:



```
create login nagarjuna with password = 'avengerswin#123'

create login suraj with password = 'avengerswin#234'

create login vinay with password = 'avengerswin#567'
```

The final step was to create a user for each login in the employee db:

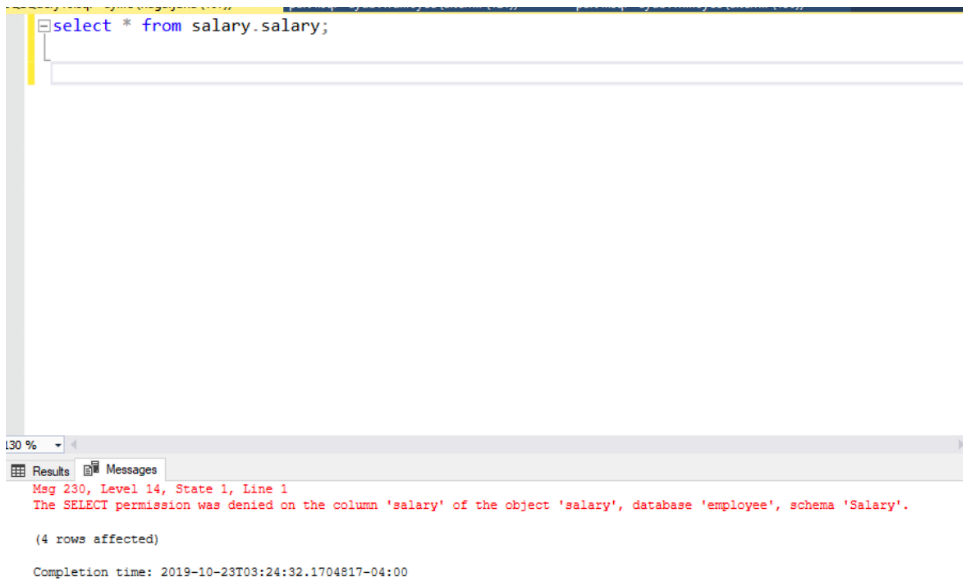


```
CREATE USER [nagarjuna] FOR LOGIN [nagarjuna] WITH DEFAULT_SCHEMA=[dbo]

grant select on salary.salary (salary_id,employee_id) to nagarjuna;

-- for Nagarjuna - define the default db at the connection options

CREATE USER [suraj] FOR LOGIN [suraj] WITH DEFAULT_SCHEMA=[dbo]

grant create table, alter to suraj;

CREATE USER [vinay] FOR LOGIN [vinay] WITH DEFAULT_SCHEMA=[Employee]

GRANT SELECT, VIEW DEFINITION ON SCHEMA::[EMPLOYEE] to[vinay]

create role [viewreader]

GRANT SELECT ON [Employee].[vEmployeeList] TO [viewreader]

exec sp_addrolemember 'viewreader', 'vinay'
```

As you can see, I created a user Nagarjuna for the corresponding login Nagarjuna. The user can only see certain columns of salary. I also created a user Suraj that can create, alter and drop tables. Finally, I created a user Vinay who can only access schema "Employee". Later, I also added him to the role of users who can only read the views but nothing else.

User permissions in action:

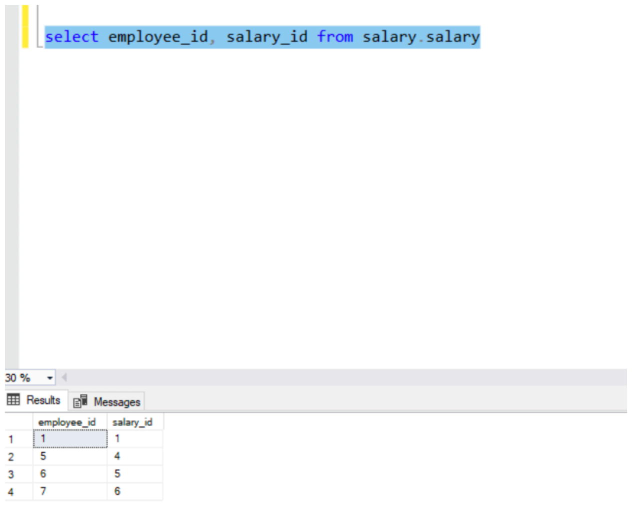
User 1 - Nagarjuna:



```
select * from salary.salary;
```

```
130 %
Results  Messages
Msg 230, Level 14, State 1, Line 1
The SELECT permission was denied on the column 'salary' of the object 'salary', database 'employee', schema 'Salary'.

(4 rows affected)

Completion time: 2019-10-23T03:24:32.1704817-04:00
```

As expected, the user can’t access the column salary in the salary table. He can only run the statement to get the rest of the columns:

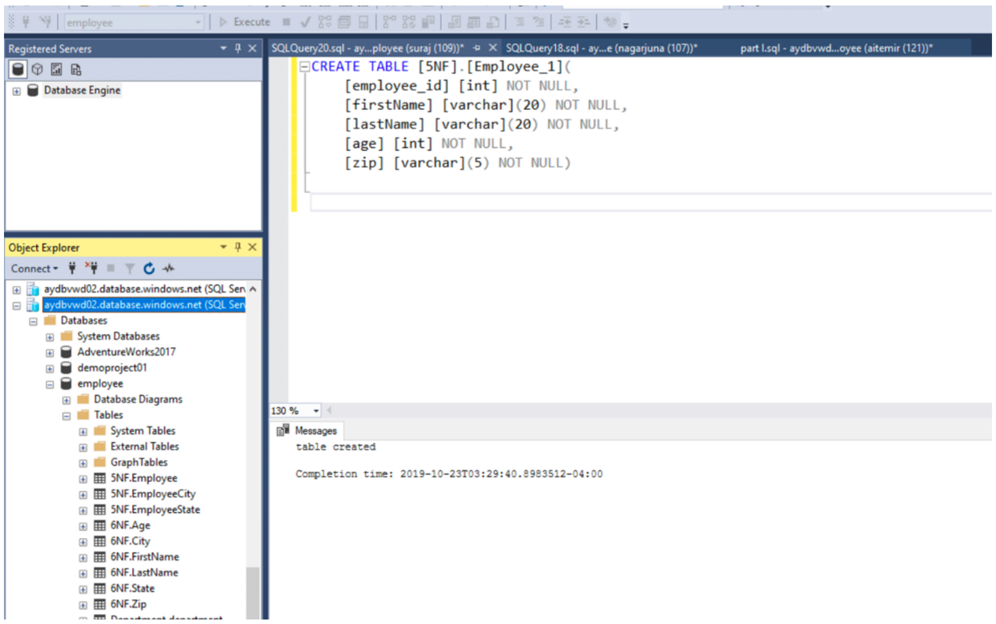


```
select employee_id, salary_id from salary.salary
```

```
30 %
Results  Messages
```

	employee_id	salary_id
1	1	1
2	5	4
3	6	5
4	7	6

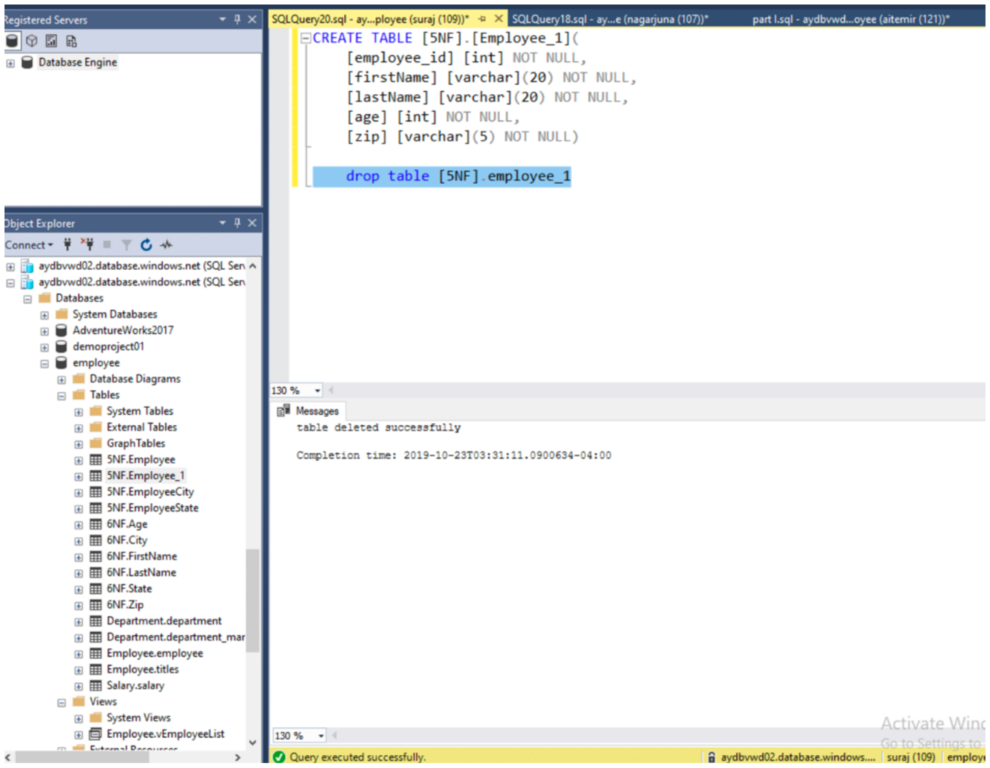
User 2 - Suraj:



```
CREATE TABLE [5NF].[Employee_1](
    [employee_id] [int] NOT NULL,
    [firstName] [varchar](20) NOT NULL,
    [lastName] [varchar](20) NOT NULL,
    [age] [int] NOT NULL,
    [zip] [varchar](5) NOT NULL)
```

```
table created

Completion time: 2019-10-23T03:29:40.8983512-04:00
```

As expected, the user can create a table with the permission that I assigned initially. Also, I successfully tested the delete as well:

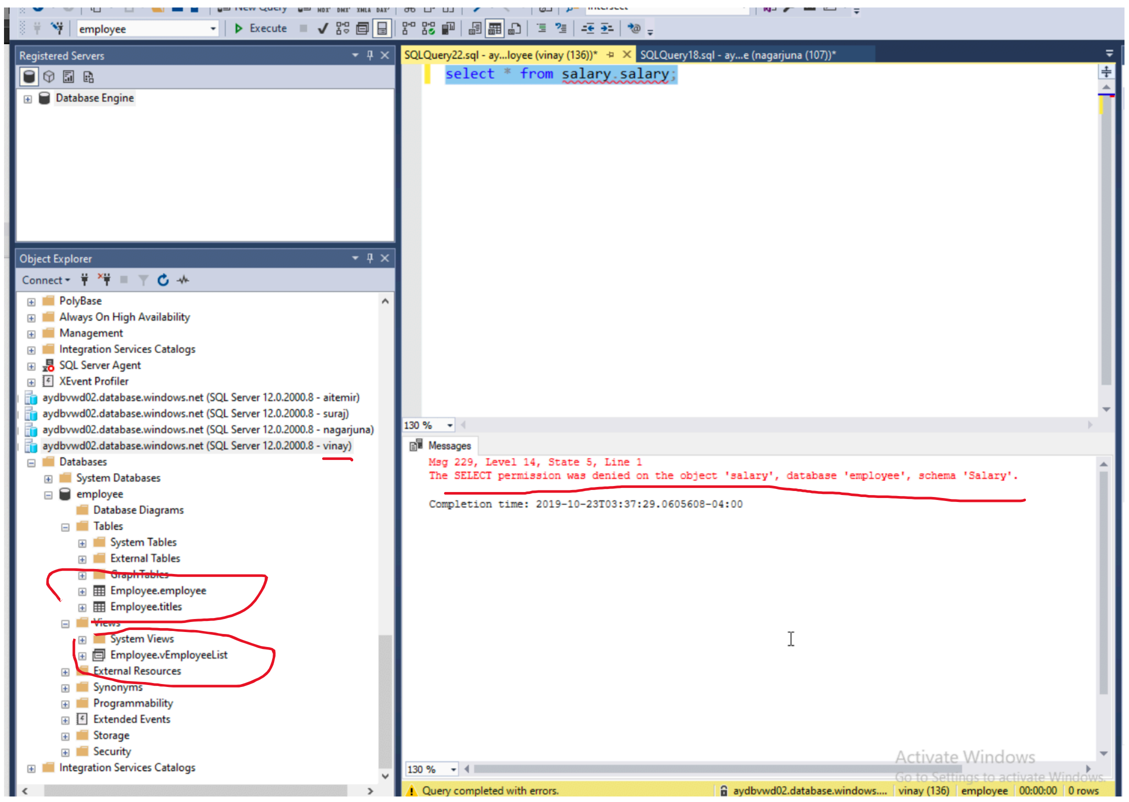


```
CREATE TABLE [5NF].[Employee_1](
    [employee_id] [int] NOT NULL,
    [firstName] [varchar](20) NOT NULL,
    [lastName] [varchar](20) NOT NULL,
    [age] [int] NOT NULL,
    [zip] [varchar](5) NOT NULL)

    drop table [5NF].employee_1
```

```
table deleted successfully

Completion time: 2019-10-23T03:31:11.0900634-04:00
```

```
Query executed successfully.
```

User 3 - Vinay:



```
select * from salary.salary;
```

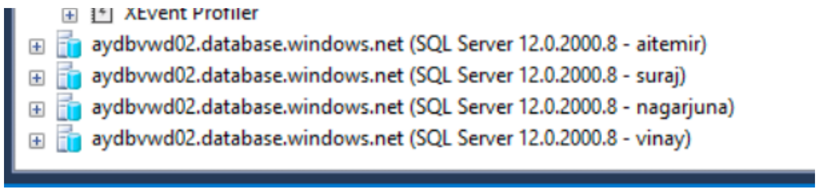
```
Msg 229, Level 14, State 5, Line 1
The SELECT permission was denied on the object 'salary', database 'employee', schema 'Salary'.

Completion time: 2019-10-23T03:37:29.0605608-04:00
```

Query completed with errors. | aydbvwd02.database.windows.... | vinay (136) | employee | 00:00:00 | 0 rows

Registered Servers
Database Engine
Object Explorer
Connect
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler
aydbvwd02.database.windows.net (SQL Server 12.0.2000.8 - aitemir)
aydbvwd02.database.windows.net (SQL Server 12.0.2000.8 - suraj)
aydbvwd02.database.windows.net (SQL Server 12.0.2000.8 - nagarjuna)
aydbvwd02.database.windows.net (SQL Server 12.0.2000.8 - vinay)
Databases
System Databases
employee
Database Diagrams
Tables
System Tables
External Tables
GraphTables
Employee.employee
Employee.titles
Views
System Views
Employee.vEmployeeList
External Resources
Synonyms
Programmability
Extended Events
Storage
Security
Integration Services Catalogs
SQLQuery22.sql - ay...loyee (vinay (136))*
SQLQuery18.sql - ay...e (nagarjuna (107))*
Execute
130 %
Messages

As you can see, I satisfied 2 requirements in this example. The user Vinay can only see schema Employee and cannot access other tables. Also, the user of the role "viewreader" and can only see the views apart from the employee schema.

All of the four connections have been successfully tested. The convenience about SSMS is that every connection has its username at the end, that way you won't be confused:

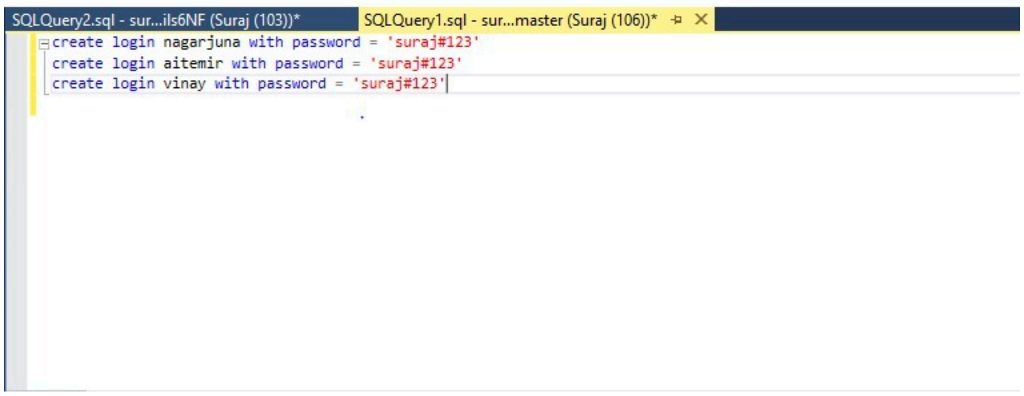


XEvent Profiler
aydbvwd02.database.windows.net (SQL Server 12.0.2000.8 - aitemir)
aydbvwd02.database.windows.net (SQL Server 12.0.2000.8 - suraj)
aydbvwd02.database.windows.net (SQL Server 12.0.2000.8 - nagarjuna)
aydbvwd02.database.windows.net (SQL Server 12.0.2000.8 - vinay)

Team members Suraj, Vinay, and Nagarjuna did the same procedures and recreated their team members in their servers as well:

Suraj's server:

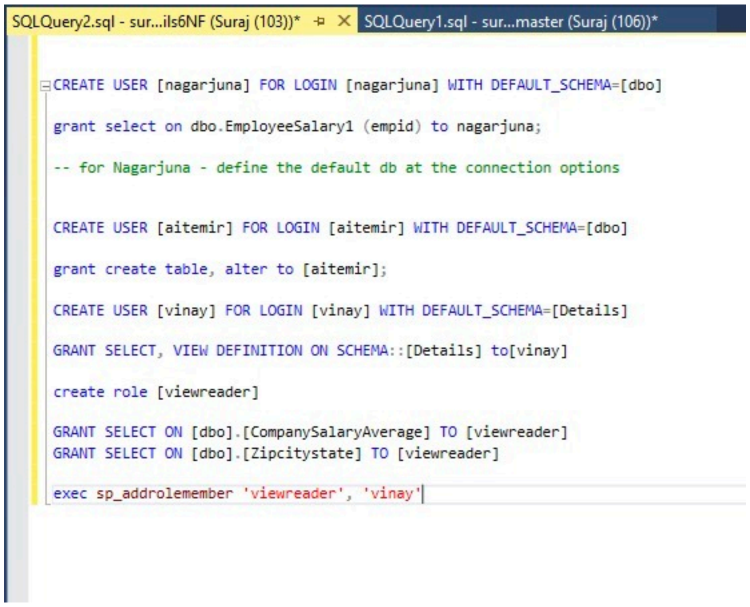
Logins created:



SQLQuery2.sql - sur...ils6NF (Suraj (103))* | SQLQuery1.sql - sur...master (Suraj (106))*

```
create login nagarjuna with password = 'suraj#123'
create login aitemir with password = 'suraj#123'
create login vinay with password = 'suraj#123'
```

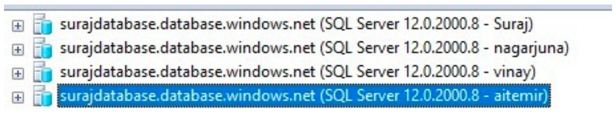
Users created with certain permissions, similar to the above examples:



SQLQuery2.sql - sur...ils6NF (Suraj (103))* | SQLQuery1.sql - sur...master (Suraj (106))*

```
CREATE USER [nagarjuna] FOR LOGIN [nagarjuna] WITH DEFAULT_SCHEMA=[dbo]

grant select on dbo.EmployeeSalary1 (empid) to nagarjuna;

-- for Nagarjuna - define the default db at the connection options


CREATE USER [aitemir] FOR LOGIN [aitemir] WITH DEFAULT_SCHEMA=[dbo]

grant create table, alter to [aitemir];

CREATE USER [vinay] FOR LOGIN [vinay] WITH DEFAULT_SCHEMA=[Details]

GRANT SELECT, VIEW DEFINITION ON SCHEMA::[Details] to[vinay]

create role [viewreader]

GRANT SELECT ON [dbo].[CompanySalaryAverage] TO [viewreader]
GRANT SELECT ON [dbo].[Zipcitystate] TO [viewreader]

exec sp_addrolemember 'viewreader', 'vinay'
```

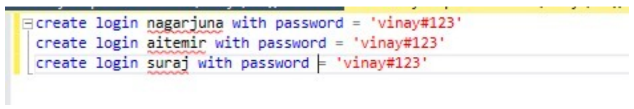
4 connections on Suraj's server:



```
surajdatabase.database.windows.net (SQL Server 12.0.2000.8 - Suraj)
surajdatabase.database.windows.net (SQL Server 12.0.2000.8 - nagarjuna)
surajdatabase.database.windows.net (SQL Server 12.0.2000.8 - vinay)
surajdatabase.database.windows.net (SQL Server 12.0.2000.8 - aitemir)
```

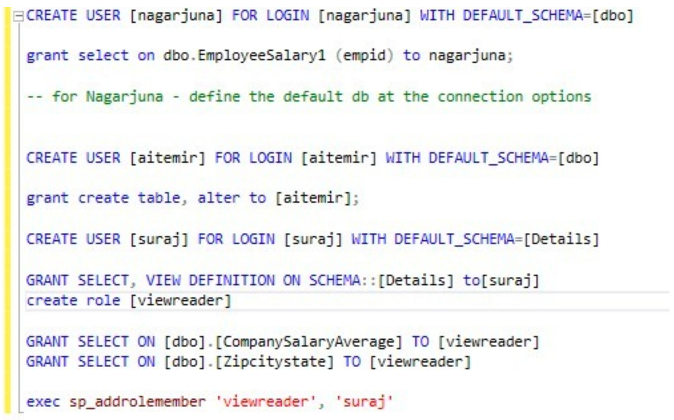
Vinay's server:

Logins created:



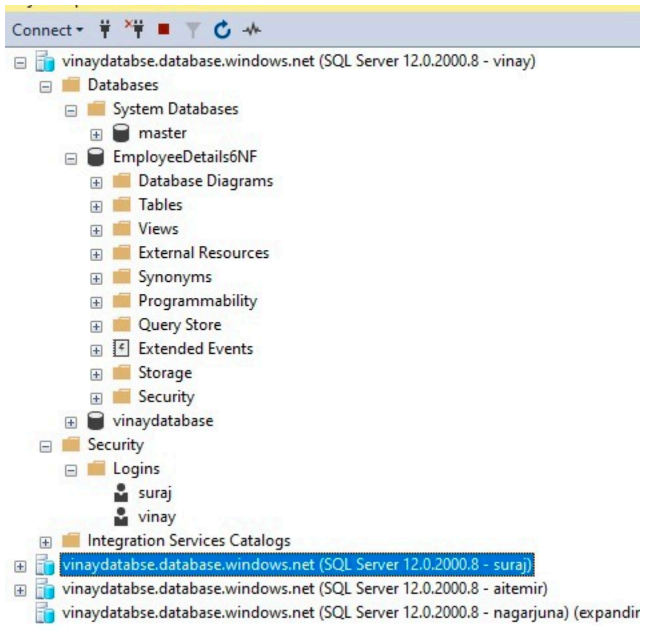
```
create login nagarjuna with password = 'vinay#123'
create login aitemir with password = 'vinay#123'
create login suraj with password = 'vinay#123'
```

Users created with certain permissions, similar to the above examples:



```
CREATE USER [nagarjuna] FOR LOGIN [nagarjuna] WITH DEFAULT_SCHEMA=[dbo]

grant select on dbo.EmployeeSalary1 (empid) to nagarjuna;

-- for Nagarjuna - define the default db at the connection options


CREATE USER [aitemir] FOR LOGIN [aitemir] WITH DEFAULT_SCHEMA=[dbo]

grant create table, alter to [aitemir];

CREATE USER [suraj] FOR LOGIN [suraj] WITH DEFAULT_SCHEMA=[Details]

GRANT SELECT, VIEW DEFINITION ON SCHEMA::[Details] to[suraj]
create role [viewreader]

GRANT SELECT ON [dbo].[CompanySalaryAverage] TO [viewreader]
GRANT SELECT ON [dbo].[Zipcitystate] TO [viewreader]

exec sp_addrolemember 'viewreader', 'suraj'
```

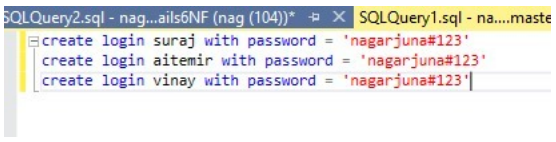
4 connections on Vinay’s server:



```
⊟ vinaydatabse.database.windows.net (SQL Server 12.0.2000.8 - vinay)
    ⊟ Databases
        ⊟ System Databases
            ⊞ master
        ⊟ EmployeeDetails6NF
            ⊞ Database Diagrams
            ⊞ Tables
            ⊞ Views
            ⊞ External Resources
            ⊞ Synonyms
            ⊞ Programmability
            ⊞ Query Store
            ⊞ Extended Events
            ⊞ Storage
            ⊞ Security
        ⊞ vinaydatabase
    ⊟ Security
        ⊟ Logins
            suraj
            vinay
    ⊞ Integration Services Catalogs
⊞ vinaydatabse.database.windows.net (SQL Server 12.0.2000.8 - suraj)
⊞ vinaydatabse.database.windows.net (SQL Server 12.0.2000.8 - aitemir)
  vinaydatabse.database.windows.net (SQL Server 12.0.2000.8 - nagarjuna) (expandir
```

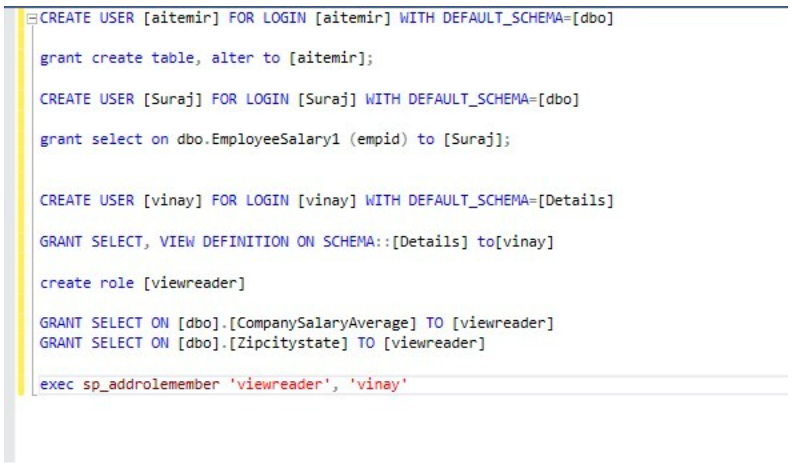
Nagarjuna’s server:

Logins created:


SQLQuery2.sql - nag...ails6NF (nag (104))*
SQLQuery1.sql - na....maste

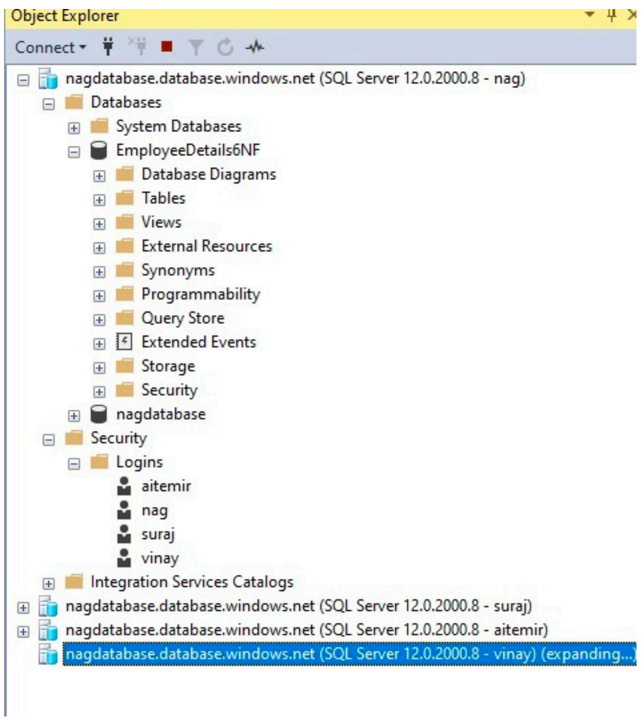
```
create login suraj with password = 'nagarjuna#123'
create login aitemir with password = 'nagarjuna#123'
create login vinay with password = 'nagarjuna#123'
```

Users created with certain permissions, similar to the above examples:



```
CREATE USER [aitemir] FOR LOGIN [aitemir] WITH DEFAULT_SCHEMA=[dbo]

grant create table, alter to [aitemir];

CREATE USER [Suraj] FOR LOGIN [Suraj] WITH DEFAULT_SCHEMA=[dbo]

grant select on dbo.EmployeeSalary1 (empid) to [Suraj];


CREATE USER [vinay] FOR LOGIN [vinay] WITH DEFAULT_SCHEMA=[Details]

GRANT SELECT, VIEW DEFINITION ON SCHEMA::[Details] to[vinay]

create role [viewreader]

GRANT SELECT ON [dbo].[CompanySalaryAverage] TO [viewreader]
GRANT SELECT ON [dbo].[Zipcitystate] TO [viewreader]

exec sp_addrolemember 'viewreader', 'vinay'
```

4 connections on Nagarjuna's server:



```
nagdatabase.database.windows.net (SQL Server 12.0.2000.8 - nag)
  Databases
    System Databases
    EmployeeDetails6NF
      Database Diagrams
      Tables
      Views
      External Resources
      Synonyms
      Programmability
      Query Store
      Extended Events
      Storage
      Security
    nagdatabase
  Security
    Logins
      aitemir
      nag
      suraj
      vinay
  Integration Services Catalogs
nagdatabase.database.windows.net (SQL Server 12.0.2000.8 - suraj)
nagdatabase.database.windows.net (SQL Server 12.0.2000.8 - aitemir)
nagdatabase.database.windows.net (SQL Server 12.0.2000.8 - vinay) (expanding...)
```

Suraj, Vinay, and Nagarjuna used EmployeeDetails6NF as the database deployed to their sql servers and it works perfectly for them.