|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Hope Foundation’s,**  **Finolex Academy of Management and Technology, Ratnagiri** | | | | | | | | | |
| **Department of Information Technology** | | | | | | | | | |
| Subject name: OLAP LAB | | | | | | | | Subject Code: ITL503 | | | |
| Class | | TE IT | | Semester – V (CBCGS) | | | | Academic year: 2018-19 | | | |
| Name of Student | | Kazi Jawwad A Rahim | | | | | **QUIZ Score :** | | | | |
| Roll No | | 32 | | | Assignment/Experiment No. | | | | | 04 | |
| Title**:** To design a distributed database for a defined problem statement. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| 1. **Course objectives applicable:**   **LOB1**- Understand the basics of distributed database environment | | | | | | | | | | | |
| 1. **Course outcomes applicable:**   **LO1**- Get familiar with the currently available models, technologies for and approaches to building distributed database systems and services. | | | | | | | | | | | |
| **3. Learning Objectives:**   * Introduce basic principles and implementation techniques of distributed database systems * Showing the need for distributed database technology to tackle deficiencies of the centralized database systems. | | | | | | | | | | | |
| 1. **Practical applications of the assignment/experiment:**  * Industry and other applications located at different locations. | | | | | | | | | | | |
| **5. Prerequisites**: | | | | | | | | | | | |
| **6. Hardware Requirements**:   1. PC with 4GB RAM, 500GB HDD,   **7. Software Requirements:**  1. SQL server 2012, SQL server Management studio 2012. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **8. Quiz Questions (if any): (Online Exam will be taken separately batch wise, attach the certificate/ Marks obtained):**  https://goo.gl/Zd4xgn | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **9. Experiment/Assignment Evaluation:** | | | | | | | | | | | |
| **Sr. No.** | **Parameters** | | | | | | | | **Marks obtained** | | **Out of** |
| **1** | Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used - | | | | | | | |  | | 6 |
| **2** | Neatness/presentation | | | | | | | |  | | 2 |
| **3** | Punctuality | | | | | | | |  | | 2 |
| **Date of performance (DOP)** | | |  | | | **Total marks obtained** | | |  | | **10** |
| **Date of checking (DOC)** | | |  | | | **Signature of teacher** | | | | | |

**Results:**

**Linked servers:**

A linked server enables you to execute distributed queries against tables stored in a Microsoft SQL Server instance and another data store. Linked servers allow submitting a T-SQL statement on a SQL Server instance, which returns data from other SQL Server instances. A linked server allows joining data from several SQL Server instances using a single T-SQL statement when data exists on multiple databases on different SQL instances. By using a linked server to retrieve data from several SQL instances, the only thing that should be done is to connect to one SQL instance.

There are two ways of configuring linked server in SSMS. One way is by using sp\_addlinkedserver system stored procedure and another is by using SQL Server Management Studio (SSMS) GUI interface.

This experiment explains how to configure a linked server using a SQL Server data source.

**Creating a Linked Server using the Object Explorer**

1. In Management Studio, connect to your Database Engine instance.
2. In the Object Explorer, expand **Server Objects**, then right-click **Linked Servers**, and then click **New Linked Server**.
3. In the New Linked Server dialog box, use the options on the General page to configure your linked server:
   1. In the **Linked Server** field, type a name for your linked server.
   2. Under **Server Type**, select **SQL SERVER**.
4. Use the options on the Security page to configure login mappings your linked server:
   1. Click on ADD. In Local Login, select the user login and check the Impersonate check box

If your data store requires authentication, then use the options on the Security page to provide your credentials:

* 1. Select **Be Made Using this Security Context**.
  2. In the **Remote Login** field, type your user name for accessing the data store.
  3. In the **With Password** field, type the password corresponding to the user name you specified above.

1. Click **OK** to save your settings and link the data store to the SQL Server instance.

The linked server is displayed as a node under the Linked Servers branch in the Object Explorer, and you can expand the node to browse the contents of the server down to the table level. You can now start executing distributed queries to work with data from your data store and the SQL Server instance.

**Querying a Linked Server**

Normally, you can query data by using scripts in Management Studio; however, scripts are not supported for linked servers. To execute queries against a linked server, use the editor.

**To query a linked server:**

In the toolbar at the top of Management Studio, click New Query. In the editor window that appears, type your query.

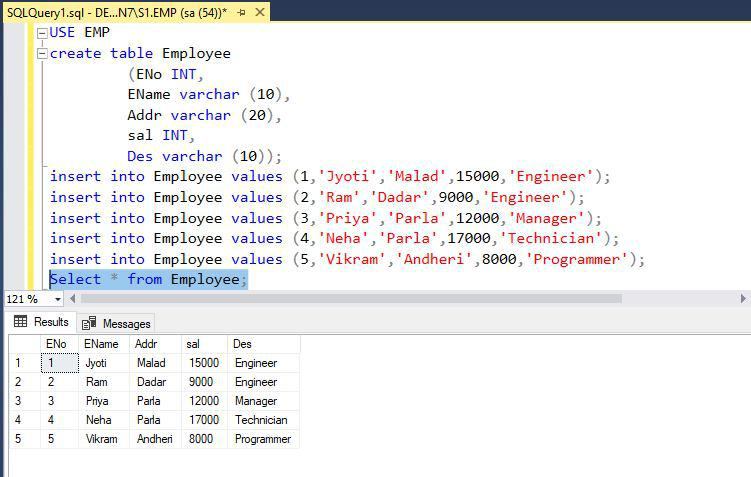
In the toolbar at the top, click Execute.

Data retrieved from your query is displayed in the Results pane.

Here to demonstrate distributed database terminology and working we have created three server instances namely S1, S2 & S3. We create an Employee database on Server instance S1. We later create linked server to create a replica using liked server on server instances S2. Later we try to operate few queries to understand the distributed database concepts.

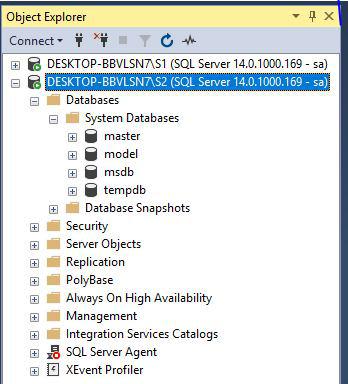
**Steps to create a distributed database environment:**

**1. In SSMS connect to server instance and create a database on S1.**



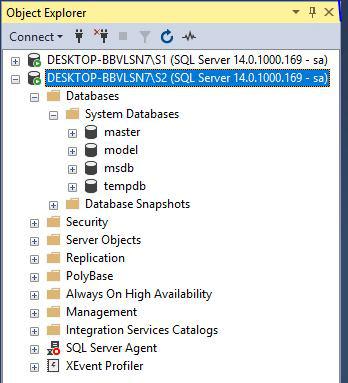
**Above is the result from Employee table on S1.**

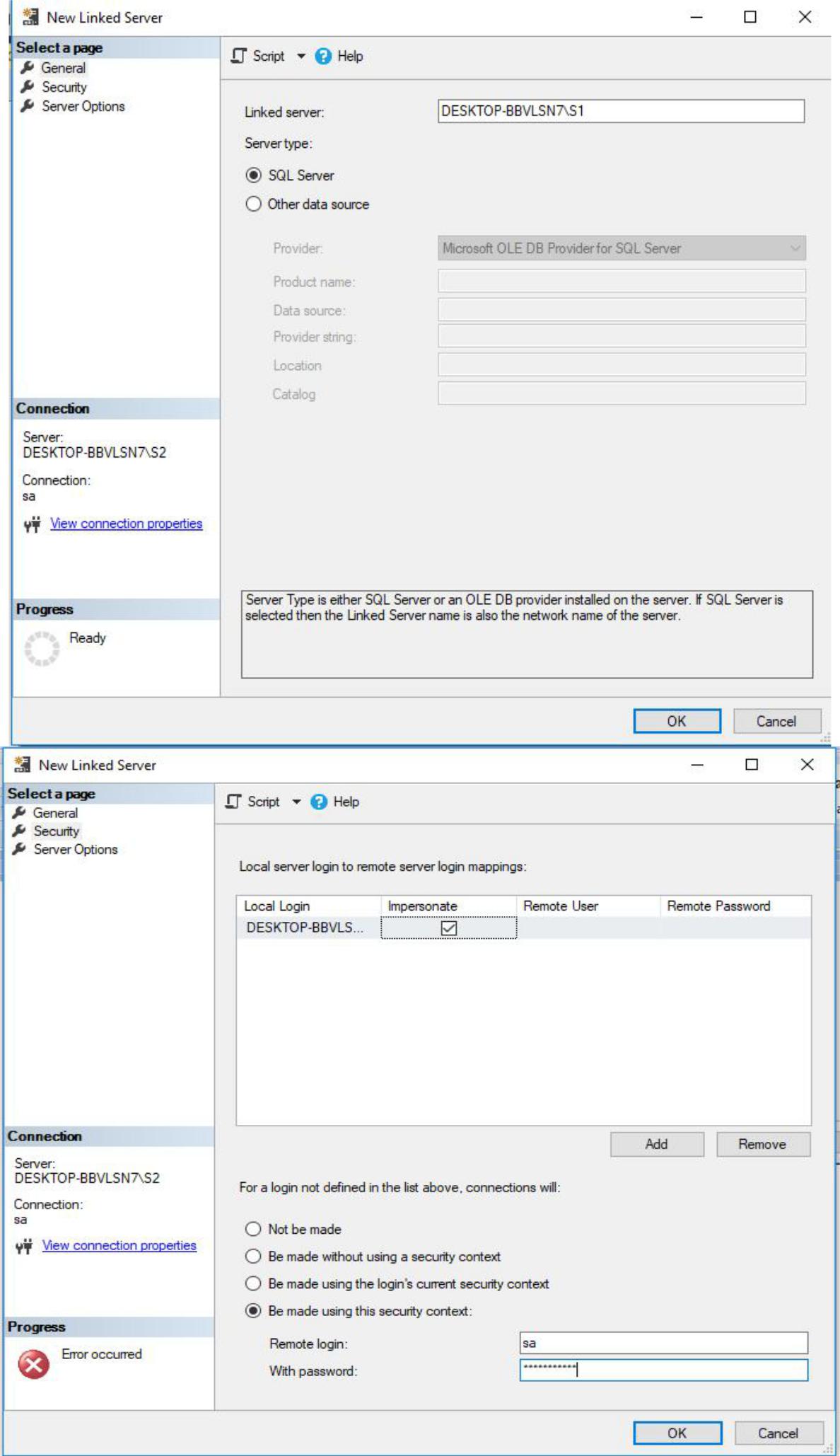
**2. Connect to server instance S2 and check for available Database**



1. Now create a linked server on server instance S2 using the steps mentioned in the **Creating** **a Linked Server using the Object Explorer** topic.

**In the server instance S2.**

**:**



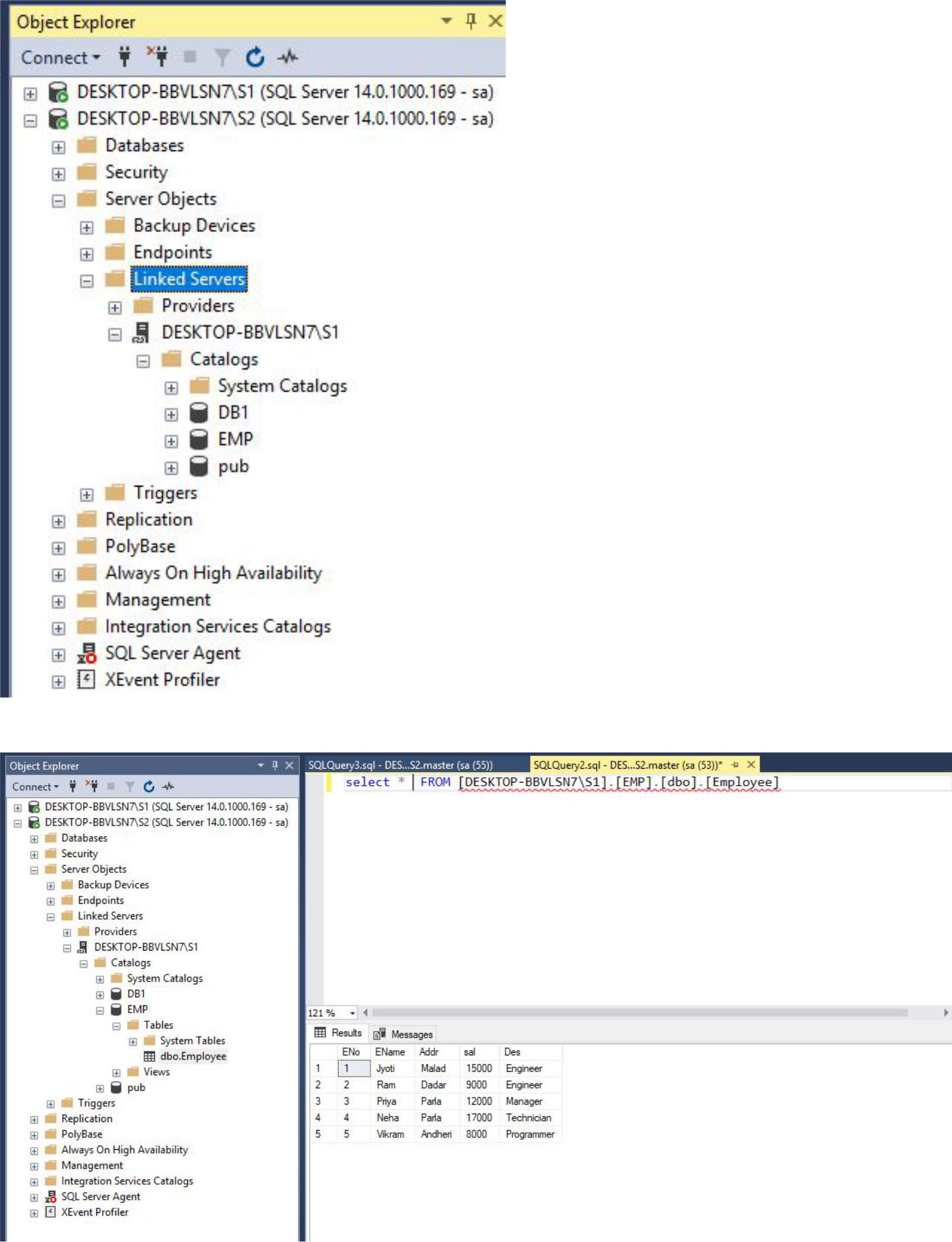


Fig. Results on server instance S2.

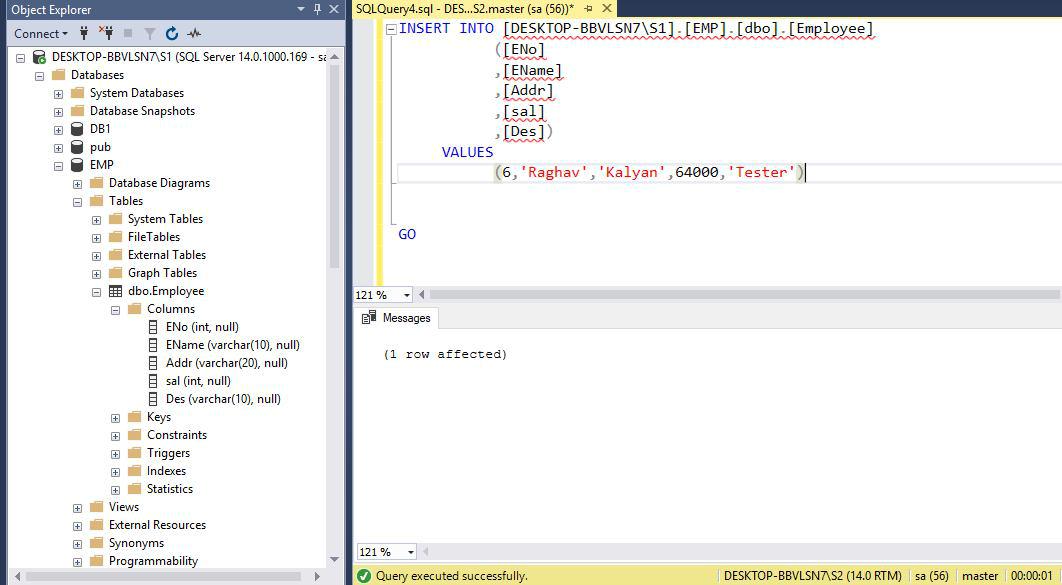


Fig. Inserting a row in Database on server instance S2

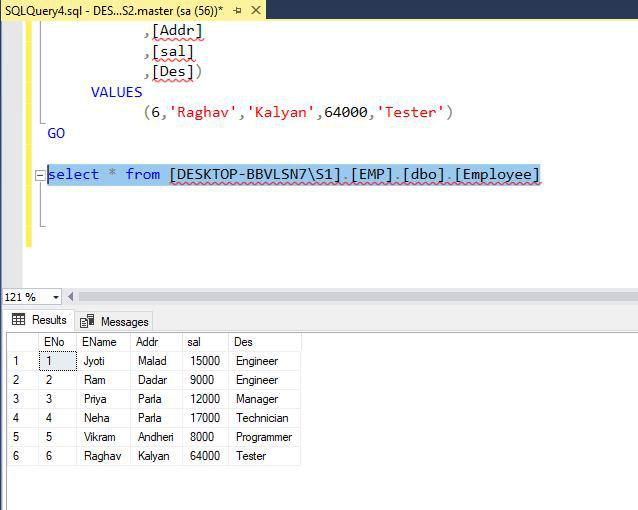


Fig. Results on server instance S2 after inserting a new row.

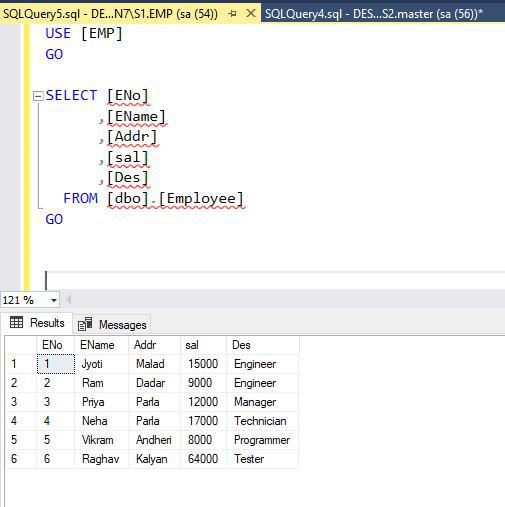


Fig. Querying the database on server instance S1. Changes made from S2 instance are reflected on database in S1.

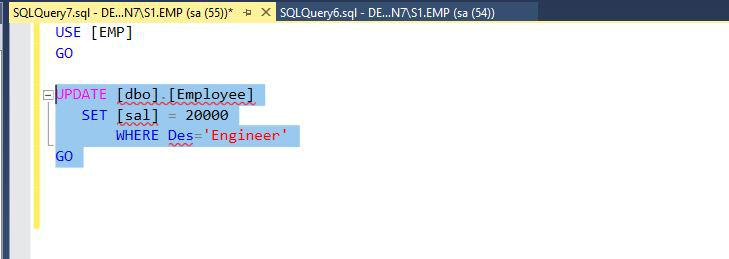


Fig. Update query on server instance S1

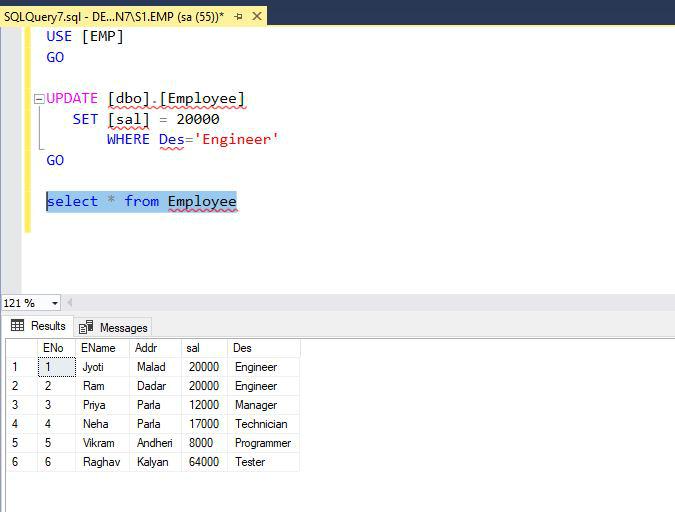


Fig. Results after update on S1.

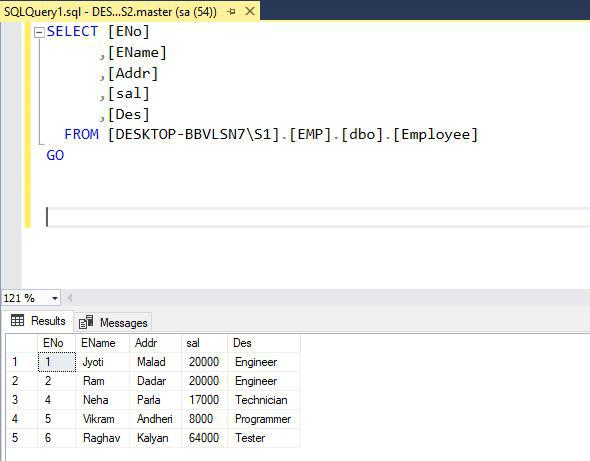


Fig. Results of update reflect on server instance S2.