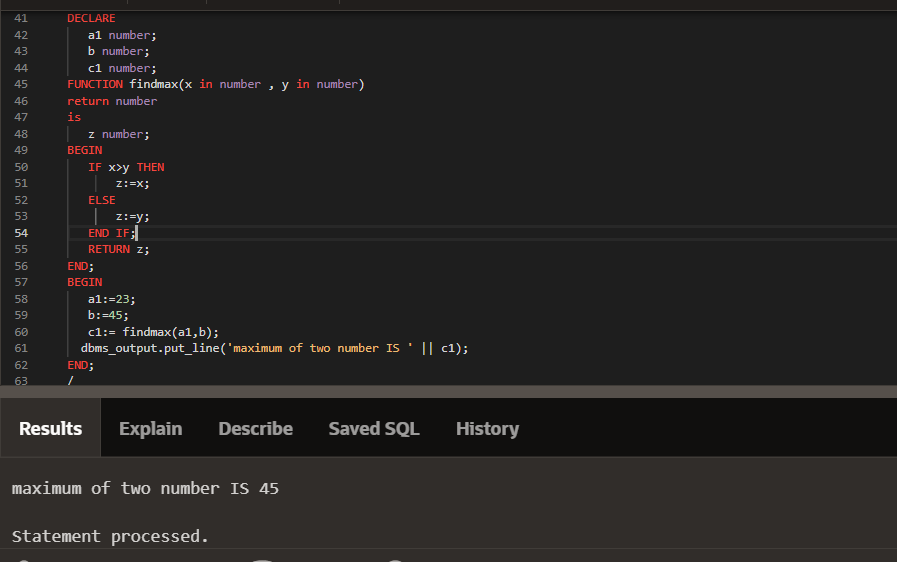
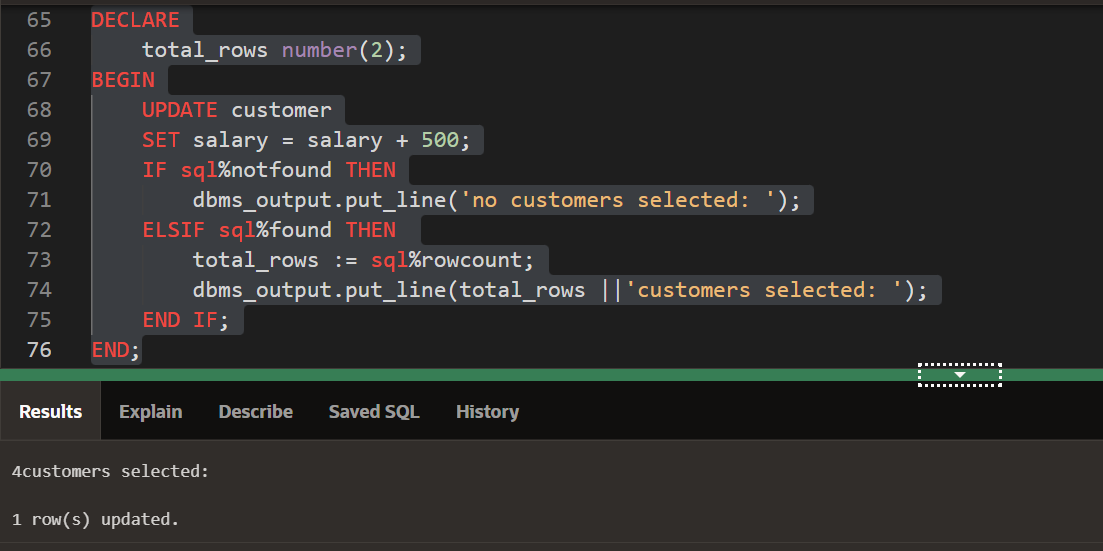


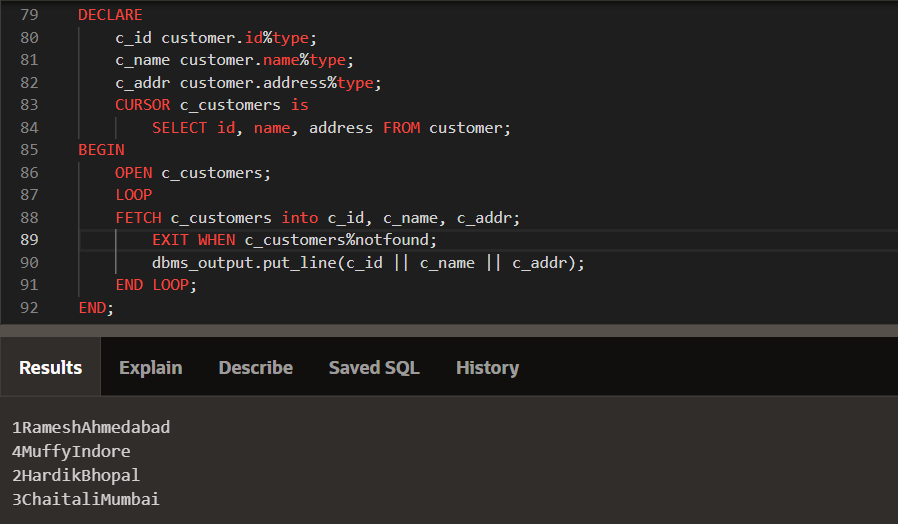
**Difference between procedures and function**

A procedure is a routine that can accept arguments but does not return any values. A function is a routine that can accept arguments and returns one or more values.

| **S.NO** | **Function** | **Procedure** |
| --- | --- | --- |
| 1. | Functions always return a value after the execution of queries. | The procedure can return a value using “IN OUT” and “OUT” arguments. |
| 2. | In SQL, those functions having a DML statement can not be called from SQL statements. But autonomous transaction functions can be called from SQL queries. | A procedure can not be called using SQL queries. |
| 3. | Each and every time functions are compiled they provide output according to the given input. | Procedures are compiled only once but they can be called many times as needed without being compiled each time. |
| 4. | A Function can not return multiple result sets. | A procedure is able to return multiple result sets. |
| 5. | The function can be called using Stored Procedure. | While procedures cannot be called from function. |
| 6. | A function used only to read data. | A procedure can be used to read and modify data. |
| 7. | The return statement of a function returns the control and function’s result value to the calling program. | While the return statement of the procedure returns control to the calling program, it can not return the result value. |
| 8. | The function does not support try-catch blocks. | Procedure supports try-catch blocks for error handling. |
| 9. | A function can be operated in the SELECT statement. | While it can’t be operated in the SELECT statement. |
| 10. | Functions do not permit transaction management. | It allows transaction management. |
| 11. | In functions, we can use only a table variable. Temporary tables can not be created in function. | In procedures, we can use temporary tables or table variables to store temporary data. |





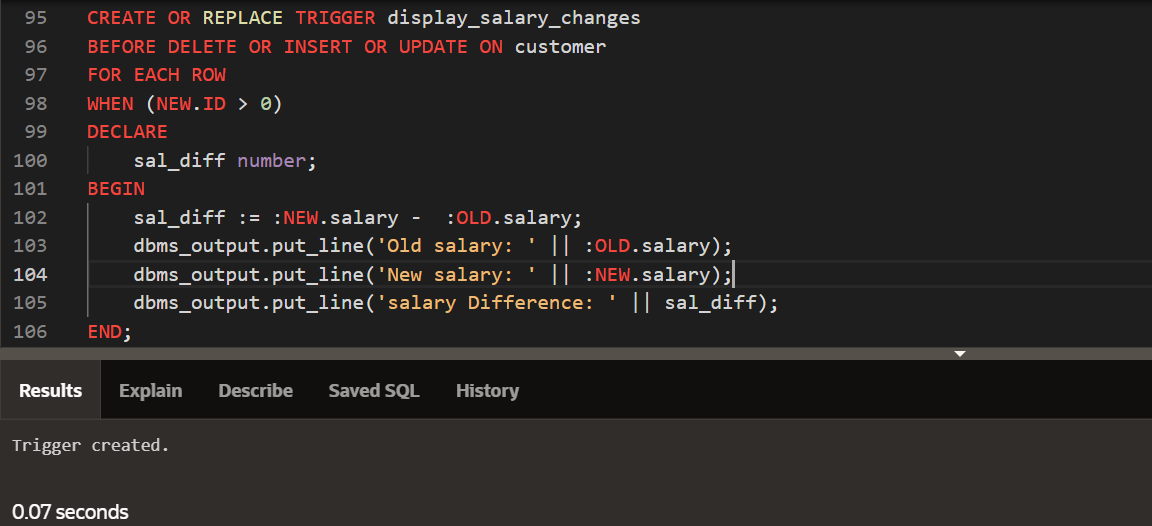


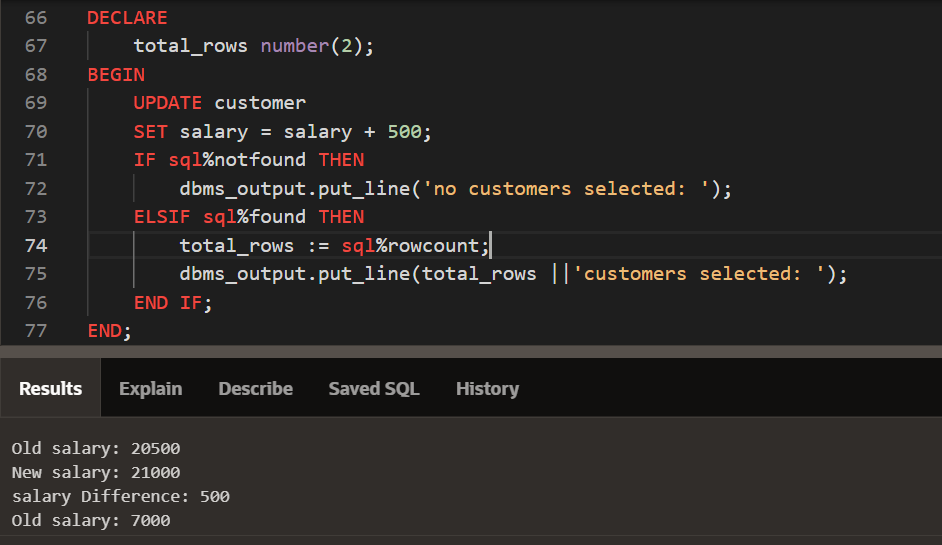
**What is Cursor in SQL ?**

**Cursor** is a Temporary Memory or Temporary Work Station. It is Allocated by [Database](https://www.geeksforgeeks.org/what-is-database/)Server at the Time of Performing [DML](https://www.geeksforgeeks.org/dml-full-form/)(Data Manipulation Language) operations on the Table by the User. Cursors are used to store Database Tables.

There are 2 types of Cursors: Implicit Cursors, and Explicit Cursors. These are explained as following below.

1. **Implicit Cursors:** Implicit Cursors are also known as Default Cursors of SQL SERVER. These Cursors are allocated by SQL SERVER when the user performs DML operations.
2. **Explicit Cursors:** Explicit Cursors are Created by Users whenever the user requires them. Explicit Cursors are used for Fetching data from Table in Row-By-Row Manner.

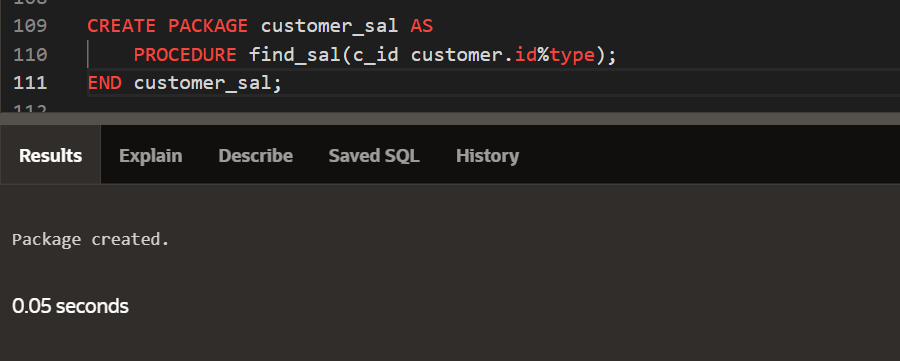


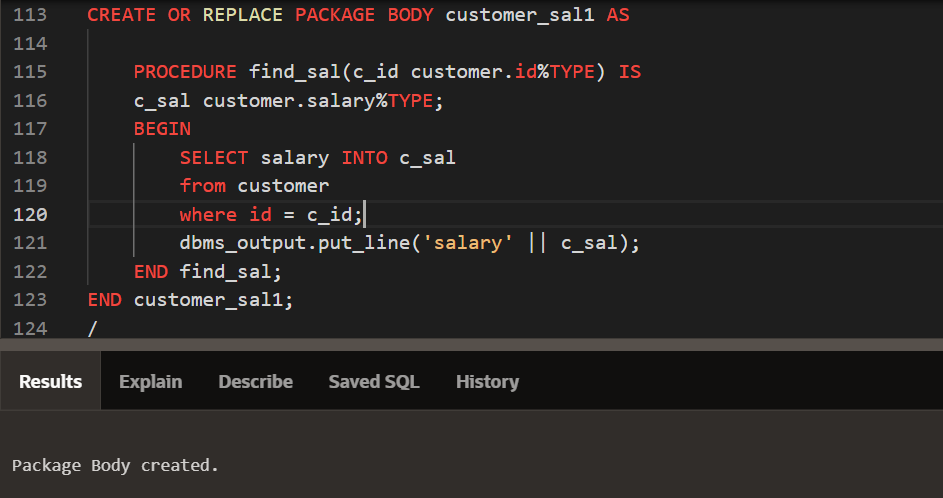


# SQL Trigger

A trigger is a stored procedure in a database that automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when specific table columns are updated. In simple words, a trigger is a collection of [SQL](https://www.geeksforgeeks.org/sql-tutorial)statements with particular names that are stored in system memory. It belongs to a specific class of stored procedures that are automatically invoked in response to database server events. Every trigger has a table attached to it.

Because a trigger cannot be called directly, unlike a stored procedure, it is referred to as a special procedure. A trigger is automatically called whenever a data modification event against a table takes place, which is the main distinction between a trigger and a procedure. On the other hand, a stored procedure must be called directly.





**What is No SQl Database ?**

When people use the term “NoSQL database,” they typically use it to refer to any non-relational database. Some say the term “NoSQL” stands for “non-SQL” while others say it stands for “not only SQL.” Either way, most agree that NoSQL databases store data in a more natural and flexible way. NoSQL, as opposed to SQL, is a database management approach, whereas SQL is just a query language, similar to the query languages of NoSQL databases.

**Advantages of No SQL Database**

1. **High scalability:** NoSQL databases use sharding for horizontal scaling. Partitioning of data and placing it on multiple machines in such a way that the order of the data is preserved is sharding. Vertical scaling means adding more resources to the existing machine whereas horizontal scaling means adding more machines to handle the data. Vertical scaling is not that easy to implement but horizontal scaling is easy to implement. Examples of horizontal scaling databases are MongoDB, Cassandra, etc. NoSQL can handle a huge amount of data because of scalability, as the data grows NoSQL scales**The auto** itself to handle that data in an efficient manner.
2. **Flexibility:** NoSQL databases are designed to handle unstructured or semi-structured data, which means that they can accommodate dynamic changes to the data model. This makes NoSQL databases a good fit for applications that need to handle changing data requirements.
3. **High availability:** The auto**,** replication feature in NoSQL databases makes it highly available because in case of any failure data replicates itself to the previous consistent state.
4. **Scalability:** NoSQL databases are highly scalable, which means that they can handle large amounts of data and traffic with ease. This makes them a good fit for applications that need to handle large amounts of data or traffic
5. **Performance:** NoSQL databases are designed to handle large amounts of data and traffic, which means that they can offer improved performance compared to traditional relational databases.
6. **Cost-effectiveness:** NoSQL databases are often more cost-effective than traditional relational databases, as they are typically less complex and do not require expensive hardware or software.
7. **Agility:**Ideal for agile development.