



Dynamic SQL

Dynamic SQL is a programming technique that could be used to write SQL queries during runtime. Dynamic SQL could be used to create general and flexible SQL queries.



Syntax for dynamic SQL is to make it string as below :

```
'SELECT statement';
```

To run a dynamic SQL statement, run the stored procedure **sp_executesql** as shown below :

```
EXEC sp_executesql N'SELECT statement';
```

Use prefix N with the sp_executesql to use dynamic SQL as a Unicode string.

Steps to use Dynamic SQL :

1. Declare two variables, @var1 for holding the name of the table and @var 2 for holding the dynamic SQL :
2. DECLARE
3. @var1 NVARCHAR (MAX),
@var2 NVARCHAR (MAX);

4. Set the value of the @var1 variable to table_name :

```
SET @var1 = N'table_name';
```

5. Create the dynamic SQL by adding the SELECT statement to the table name parameter :

6. SET @var2= N'SELECT *
FROM ' + @var1;

7. Run the sp_executesql stored procedure by using the @var2 parameter :

```
EXEC sp_executesql @var2;
```



Example –

```
SELECT *  
from geek;
```

Table – Geek

ID	NAME	CITY
1	Khushi	Jaipur
2	Neha	Noida
3	Meera	Delhi

Using Dynamic SQL :

```
DECLARE  
@tab NVARCHAR(128),  
@st NVARCHAR(MAX);  
SET @tab = N'geektable';  
SET @st = N'SELECT *  
FROM ' + @tab;  
EXEC sp_executesql @st;
```

Table – Geek

ID	NAME	CITY
1	Khushi	Jaipur
2	Neha	Noida
3	Meera	Delhi

Dynamic SQL

Not all SQL queries are run by a database administrator sitting at their desk. Each time you search Amazon, there isn't a person sitting on the other end of the Internet, keying your request into an SQL query.

Dynamic SQL allows a database administrator to write code that is flexible and adaptable. It basically means that the full SQL statement won't be known until the program runs. Think of the search example above: When a user searches Amazon, we don't know which table we should query, or what to search for until they hit Enter.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

It is a powerful tool because it lets us write programs with SQL statements that aren't necessarily known until the program runs. Dynamic SQL refers to the SQL statements that are generated at the time the program runs. As a database administrator, you necessarily know what will be executed.

Let's take the following simple example, where we want to select an artist from a table. But we're not sure which table (maybe it's from an archived table, or the live production table), and we don't know the primary key value.

1. SELECT artistID, artistName
2. FROM ????
3. WHERE artistID = ??

Dynamic SQL In Action

Let's take a look at an example of dynamic SQL in action. The code is written for SQL Server, although other database management systems are equally viable.

Below is a simple example using the scenario from above. Notice that the 'at' (@) symbol goes in front of the fields we've defined as dynamic. For example, if you have the user enter the band name, the parameter could be named @bandName. It's also in front of the variable cmd. This is because we're building the entire SELECT statement as a string.

1. DECLARE @tableName varchar(100)
2. DECLARE @artistID int
3. DECLARE @sql varchar(4000)
4. SET @cmd = "SELECT artistID, artistName FROM " + @tableName + "WHERE
artistID = " + @artistID
5. EXEC (@cmd)

Although dynamic SQL can be very powerful for creating flexible SQL queries, they can also have a negative impact on system performance. The more the system or server has to process, the slower it works. If the query has dozens of parameters, each one has to be evaluated and it takes time. It's not as easy on the system as a straightforward query with defined values.