smartdata

Effective Dynamic SQL

Presenter:
Chris St Amand, chris.stamand@wearesmartdata.com

1. ORM/auto-generated SQL
2. Build fully-expanded SQL string at client
3. Build parameterised SQL string at client
4. Use direct static SQL in stored procedures
5. Build SQL string in stored procedures and execute as SQL

Common ways to access SQL Server

Items 1, 2, 3, 5 are effectively dynamic SQL

In C#:

```
string Query="SELECT * FROM Table1 "  +
"WHERE 1=1 ";

if (condition1)

    Query += "AND Col1=0 ";

if (condition2)

    Query += "AND Col2=1 ";

if (condition3)

    Query += "AND Col3=2 ";
```

Examples of fully expanded string

In SQL:

```
EXEC('SELECT col1, col2, col3
      FROM ' + @tblname + '
      WHERE keycol = ''' + @key + '''')
```

Examples of fully expanded string

For cool people:

Examples of fully expanded string

```
con.query('SELECT * FROM ' + table, (err,rows) => {
  if(err) throw err;

  console.log('Data received from Db:\n');
  console.log(rows);
});
```

In C#:

@parameterised

dynamic SQL

```
string sql = "SELECT empSalary from employee where salary = @salary";

using (SqlConnection connection = new SqlConnection(/* connection info */))
using (SqlCommand command = new SqlCommand(sql, connection))
{
    var salaryParam = new SqlParameter("salary", SqlDbType.Money);
    salaryParam.Value = txtMoney.Text;

    command.Parameters.Add(salaryParam);
    var results = command.ExecuteReader();
}
```

In SQL:

@parameterised dynamic SQL

```
DECLARE @sql nvarchar(4000)
SELECT @sql = ' SELECT col1, col2, col3 ' +
              ' FROM dbo.' + quotename(@tblname) +
              ' WHERE keycol = @key'
EXEC sp_executesql @sql, N'@key varchar(10)', @key
```

For cool people:

:parameterised

dynamic SQL

```
const query = 'INSERT INTO artists (id, name) VALUES (:id,
:name)';
// Parameters by marker name
const params = { id: 'krichards', name: 'Keith Richards' };
client.execute(query, params, { prepare: true }, callback);
```

Why use Dynamic SQL?

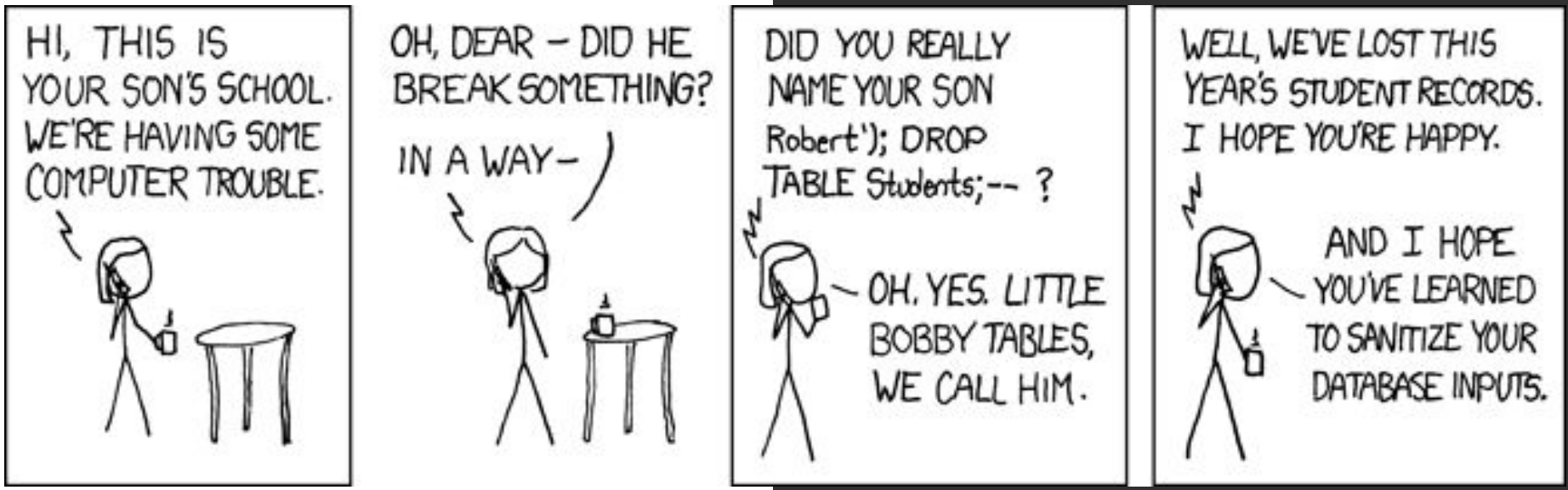
1. Customizable everything!
2. Optimize SQL performance by reducing SQL stmts
3. SQL execution across servers, databases
4. Database management

Some issues with Dynamic SQL

1. No syntax checking
2. It’s hard to read
3. Security implications
4. SQL injection risk
5. Performance implications
6. String implications
7. Increased network traffic (maybe)

- With dynamic SQL (in stored procedure or code), user executing SQL needs access to tables directly*
- With static SQL in stored procedure, they do not (ownership chaining)
- *Not entirely true for SQL 2005 and later (you can sign a stored procedure with dynamic SQL with a certificate or use Execute As clause [caution])

Security Implications

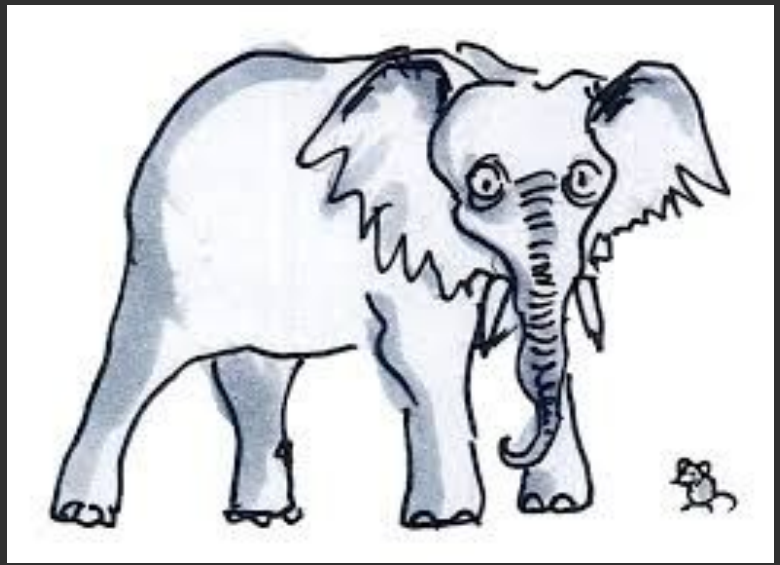

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Query Plans

- What's a query plan and why do I care about it?
- Query plan hashing
- With SQL Server 7 and later, query plans are cached for all SQL thrown at the DB

Query Plans

- For longer executing queries, recompiling a query may not be an issue and may even be a benefit (option recompile)
  - Parameter sniffing
  - Hints
- For fast executing, repeatedly called queries, recompiling each time could be a killer



- Sql IN typically has poor performance for large lists
  - Remember, the slightest difference in a query uses a different query plan
- Starting with SQL 2008 you can use TVP
  - IEnumerable<SqlDataRecord>
  - DataTable, DBDataReader
  - SqlServerDataTable for JAVA

IN (@list)

- Some people enjoy cooking, their families, and their dogs.

- Some people enjoy cooking their families and their dogs.

```sql
/* Create a table type. */
CREATE TYPE LocationTableType AS TABLE
( LocationName VARCHAR(50)
, CostRate INT );
GO

/* Create a procedure to receive data for the table-valued parameter. */
CREATE PROCEDURE dbo. usp_InsertProductionLocation
    @TVP LocationTableType READONLY
    AS
    SET NOCOUNT ON
    INSERT INTO AdventureWorks2012.Production.Location
            (Name
            ,CostRate
            ,Availability
            ,ModifiedDate)
        SELECT *, 0, GETDATE()
        FROM  @TVP;
        GO
```

IN (@list)

- For a quick hack or if you cannot use TVP for some reason, in SQL 2016 you can also use string_split

```
SELECT ...
FROM   tbl
WHERE  col IN (
    SELECT convert(int, value)
    FROM string_split('1,2,3,4', ',')
)
```

Other tips

- NULL values can bite you too
- Order By, LIKE aren’t going to work the way you think, use non-parameterized SQL
- Can’t use in it MS SQL functions
- For large SQL statements, use NVARCHAR(MAX) for params (8192 concatenation issue)

Google: drew sql execution plans youtube

Exploring Microsoft SQL Server Query Execution Plans - Drew ...

I want to learn about (SQL) execution plans!


AND THE
WINNER IS

SMART DATA SYSTEMS

Presenter Details