A list of SQL best practices

Here are some SQL programming guidelines and best practices we collected, keeping quality, performan and maintainability in mind. This list is not complete at this moment, and will be constantly updated.

- Do not use SELECT * in your queries.
- Always use table aliases when your SQL statement involves more than one source.
- Use the more readable ANSI-Standard Join clauses instead of the old style joins.
- Do not use column numbers in the ORDER BY clause.
- Always use a column list in your INSERT statements.
- Don't ever use double quotes in your T-SQL code.
- Do not prefix your stored procedure names with "sp_".
- Always use a SQL formatter to format your sql like Instant SQL Formatter(Free and Online)

Do not use SELECT * in your queries, write out the full syntax.

Always write the required column names after the SELECT statement, like:

```
SELECT CustomerID, CustomerFirstName, City from Emp;
```

This technique results in reduced disk I/O and better performance.

Always use table aliases when your SQL statement involves more than one sou

If more than one table is involved in a from clause, each column name must be qualified using either the complete table name or an alias. The alias is preferred. It is more human readable to use aliases instead writing columns with no table information.

Use the more readable ANSI-Standard Join clauses instead of the old style join

With ANSI joins, the WHERE clause is used only for filtering data. Where as with older style joins, the W clause handles both the join condition and filtering data. Furthermore ANSI join syntax supports the full join. The first of the following two queries shows the old style join, while the second one shows the ne ANSI join syntax:

```
-- old style join
SELECT a.Au_id,
       t.Title
FROM   TITLES t,
       AUTHORS a,
       TITLEAUTHOR ta
WHERE  a.Au_id = ta.Au_id
       AND ta.Title_id = t.Title_id
       AND t.Title LIKE '%Computer%'


--ANSI join syntax
SELECT a.Au_id,
       t.Title
FROM   AUTHORS a
       INNER JOIN TITLEAUTHOR ta
         ON a.Au_id = ta.Au_id
       INNER JOIN TITLES t
         ON ta.Title_id = t.Title_id
WHERE  t.Title LIKE '%Computer%'
```

Do not use column numbers in the ORDER BY clause

Always use column names in an order by clause. Avoid positional references. Consider the following exa in which the second query is more readable than the first one:

```
SELECT OrderID, OrderDate
FROM Orders
ORDER BY 2
```

```
SELECT OrderID, OrderDate
FROM Orders
ORDER BY OrderDate
```

Always use a column list in your INSERT statements

Always specify the target columns when executing an insert command. This helps in avoiding problems the table structure changes (like adding or dropping a column). Consider the following table:

```
CREATE TABLE EUROPEANCOUNTRIES
  (
     Countryid   INT PRIMARY KEY,
     Countryname VARCHAR(25)
  )
```

Here's an INSERT statement without a column list , that works perfectly:

```
INSERT INTO EuropeanCountries
VALUES (1, 'Ireland')
```

Now, let's add a new column to this table:

```
ALTER TABLE EuropeanCountries
ADD EuroSupport bit
```

Now run the above INSERT statement. You get the following error from SQL Server: Server: Msg 213, L 16, State 4, Line 1 Insert Error: Column name or number of supplied values does not match table definit This problem can be avoided by writing an INSERT statement with a column list as shown below:

```
INSERT INTO EuropeanCountries
(CountryID, CountryName)
VALUES (1, 'England')
```

Don't ever use double quotes in your T-SQL code

Use single quotes for string constants. If it's necessary to qualify an object name, use (non-ANSI SQL standard) brackets around the name, like table name: ORDER DETAILS in this SQL.

```
SELECT od.[Discount],
       od.[Quantity],
       od.[Unitprice]
FROM   [northwind].[dbo].[ORDER DETAILS] AS od
```

Do not prefix your stored procedure names with "sp_"

The prefix sp_ is reserved for system stored procedure that ship with SQL Server. Whenever SQL Serve encounters a procedure name starting with sp_, it first tries to locate the procedure in the master data then it looks for any qualifiers (database, owner) provided, then it tries dbo as the owner. So you can r save time in locating the stored procedure by avoiding the "sp_" prefix.

Always use a SQL formatter to format your sql like Instant SQL Formatter(Free and Online)

The formatting of SQL code may not seem that important,but consistent formatting makes it easier for others to scan and understand your code. SQL statements have a structure, and having that structure visually evident makes it much easier to locate and verify various parts of the statements. Uniform form also makes it much easier to add sections to and remove them from complex T-SQL statements for debugging purposes. Instant SQL Formatter is a free online SQL tidy tool that makes your SQL script re instantly.

Related Links

Datatypes translation between Oracle and SQL Server part 1: character, binary strings (7/28/2010)
Datatypes translation between Oracle and SQL Server part 2: number (8/4/2010)