

SQLSERVER 2005: SET ANSI_PADDING

[Stampa](#)
(10) » [ORACLE: Sql Ansi](#) » [SQLSERVER 2005/2008: Integration Services](#) » **[SQLSERVER 2005: SET ANSI_PADDING](#)**

Indice dei Contenuti [[Nascondi/Mostra](#)]

[Scenario](#)
[SET ANSI_PADDING \(Transact-SQL\)](#)
[Importante](#)
[Sintassi](#)
[Esempi](#)
[Riferimenti](#)

Scenario

- If you don't enable SET ANSI_PADDING ON, SQL Server trims trailing blanks for variable-length columns and for fixed-length columns that allow NULLs. The reason seems to be compatibility with legacy (pre 6.5) versions of SQL Server.

SET ANSI_PADDING (Transact-SQL)

- This setting controls the way the column stores values shorter than the defined size of the column, and the way the column stores values that have trailing blanks in char, varchar, binary, and varbinary data.
- When set to ON, trailing blanks in character values inserted into varchar columns and trailing zeros in binary values inserted into varbinary columns are not trimmed. Values are not padded to the length of the column. When set to OFF, the trailing blanks (for varchar) and zeros (for varbinary) are trimmed. This setting affects only the definition of new columns.
- **It is recommended that ANSI_PADDING always be set to ON.** SET ANSI_PADDING must be ON when creating or manipulating indexes on computed columns or indexed views.
- The status of this option can be determined by examining the IsAnsiPaddingEnabled property of the DATABASEPROPERTYEX function.
- This setting affects only the definition of new columns. **After the column is created, SQL Server 2005 stores the values based on the setting when the column was created.** Existing columns are not affected by a later change to this setting.

Importante

- In a future version of SQL Server **ANSI_PADDING will always be ON and any applications that explicitly set the option to OFF will produce an error.** Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

Sintassi

SET ANSI_PADDING { ON | OFF }

Esempi

- The following example shows how the setting affects each of these data types.

```

1  <b>With ANSI_PADDING ON</b>:
2  PRINT 'Testing with ANSI_PADDING ON'
3  SET ANSI_PADDING ON;
4  GO
5
6  CREATE TABLE t1 (
7      charcol CHAR(16) NULL,
8      varcharcol VARCHAR(16) NULL,
9      varbinarycol VARBINARY(8)
10 );
11 GO
12 INSERT INTO t1 VALUES ('No blanks', 'No blanks', 0x00ee);
13 INSERT INTO t1 VALUES ('Trailing blank ', 'Trailing blank ', 0x00ee00);
14
15 SELECT 'CHAR' = '>' + charcol + '<', 'VARCHAR'='>' + varcharcol + '<',
16        varbinarycol
17 FROM t1;
18 GO
19
20 <b>With ANSI_PADDING OFF</b>:
21 PRINT 'Testing with ANSI_PADDING OFF';
22 SET ANSI_PADDING OFF;
23 GO
24
25 CREATE TABLE t2 (
26     charcol CHAR(16) NULL,
27     varcharcol VARCHAR(16) NULL,
28     varbinarycol VARBINARY(8)
29 );
30 GO
31 INSERT INTO t2 VALUES ('No blanks', 'No blanks', 0x00ee);
32 INSERT INTO t2 VALUES ('Trailing blank ', 'Trailing blank ', 0x00ee00);
33
34 SELECT 'CHAR' = '>' + charcol + '<', 'VARCHAR'='>' + varcharcol + '<',
35        varbinarycol
36 FROM t2;
37 GO
38
39 DROP TABLE t1
40 DROP TABLE t2

```

Riferimenti

- [SQL Server 2005 Books Online: SET ANSI_PADDING \(Transact-SQL\)](#)
- [The Joel on Software Discussion Group: Why does SQL Server do this?](#)