

## Database Collation Does Not Match TempDB Collation

Your database collation doesn't match the TempDB Collation.

### Recommendation

It is recommended that the collation of your database match the collation TempDB. It is possible that if your collations do not match that you may get errors, or that queries using TempDB will not run correctly. If you use temp tables and your database collation doesn't match the TempDB collation you will get an error similar to the following error when you attempt to join to Temp Tables:

Msg 468, Level 16, State 9, Line 11

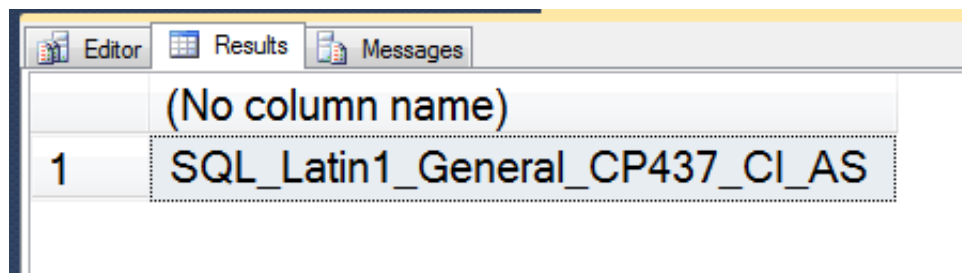
Cannot resolve the collation conflict between "SQL\_Latin1\_General\_CP437\_CI\_AS" and "Latin1\_General\_CI\_AS" in the equal to operation.

### Example Code

First to try the example, you need to check the collation of TempDB.

```
SELECT DATABASEPROPERTYEX('tempdb', 'Collation');
```

Which will look something like this (actual collation may vary).



The screenshot shows a SQL Server Enterprise Manager window with three tabs: Editor, Results, and Messages. The Results tab is active, displaying a single row of data. The first column is labeled '(No column name)' and the value is 'SQL\_Latin1\_General\_CP437\_CI\_AS'.

	(No column name)
1	SQL_Latin1_General_CP437_CI_AS

Then you need a database that has a different collation from that of TempDB.

```
CREATE DATABASE [CollationTest]  
COLLATE Latin1_General_CI_AS;
```

or if TempDB is Latin1\_General\_CI\_AS then make it different from TempDB

```
CREATE DATABASE [CollationTest]  
COLLATE SQL_Latin1_General_CP437_CI_AS;
```

Once you have a database with a different collation, you need some data to work with. In this example, I am going to use the table from my CTE presentation.

```
use [CollationTest];
CREATE TABLE [Departments] (
  id int, --would normally be an INT IDENTITY
  department VARCHAR (200),
  parent int
);

-- insert top level departments
insert into [Departments] (id, department, parent)
values (1, 'Camping', null);
insert into [Departments] (id, department, parent)
values (2, 'Cycle', null);
insert into [Departments] (id, department, parent)
values (3, 'Snowsports', null);
insert into [Departments] (id, department, parent)
values (4, 'Fitness', null);

-- now some sub-departments for camping
insert into [Departments] (id, department, parent)
values (5, 'Tents', 1);
insert into [Departments] (id, department, parent)
values (6, 'Backpacks', 1);
insert into [Departments] (id, department, parent)
values (7, 'Sleeping Bags', 1);
insert into [Departments] (id, department, parent)
values (8, 'Cooking', 1);

-- now some sub-departments for cycle
insert into [Departments] (id, department, parent)
values (9, 'Bikes', 2);
insert into [Departments] (id, department, parent)
values (10, 'Helmets', 2);
insert into [Departments] (id, department, parent)
values (11, 'Locks', 2);

-- now some sub-departments for snowsports
insert into [Departments] (id, department, parent)
values (12, 'Ski', 3);
insert into [Departments] (id, department, parent)
values (13, 'Snowboard', 3);
insert into [Departments] (id, department, parent)
values (14, 'Snowshoe', 3);

-- now some sub-departments for fitness
insert into [Departments] (id, department, parent)
values (15, 'Running', 4);
insert into [Departments] (id, department, parent)
values (16, 'Swimming', 4);
insert into [Departments] (id, department, parent)
values (17, 'Yoga', 4);
```

Now we check the collations to confirm that they are different.

```
use [CollationTest];
SELECT DATABASEPROPERTYEX('tempdb', 'Collation') as TempDBCcollation,
       DATABASEPROPERTYEX('CollationTest', 'Collation') as ThisDBCcollation;
```

Lets see where it breaks:

```
use [CollationTest];
CREATE TABLE #TempDepartments (
    id int,
    department VARCHAR (200),
    parent int
);

INSERT INTO #TempDepartments
SELECT TOP 5 * FROM [Departments];

SELECT *
    FROM [Departments] d
    INNER JOIN #TempDepartments td
        ON d.department = td.department;

DROP TABLE #TempDepartments;
```

Which will produce the following error:

**Msg 468, Level 16, State 9, Line 11**  
**Cannot resolve the collation conflict between "SQL\_Latin1\_General\_CP437\_CI\_AS" and "Latin1\_General\_CI\_AS" in the equal to operation.**

### How do we fix it?

You could change your database collation to match what TempDB is using, or you could specify the collation in the query.

```
use [CollationTest];
CREATE TABLE #TempDepartments (
    id int,
    department VARCHAR (200),
    parent int
);

INSERT INTO #TempDepartments
SELECT TOP 5 * FROM [Departments];
```

```
SELECT *
FROM [Departments] d
INNER JOIN #TempDepartments td
    ON d.department = td.department COLLATE Latin1_General_CI_AS;

DROP TABLE #TempDepartments;
```

Another option would be to use the database default collation on both.

```
use [CollationTest];
CREATE TABLE #TempDepartments (
    id int,
    department VARCHAR (200),
    parent int
);

INSERT INTO #TempDepartments
SELECT TOP 5 * FROM [Departments];

SELECT *
FROM [Departments] d
INNER JOIN #TempDepartments td
    ON d.department COLLATE DATABASE_DEFAULT
    = td.department COLLATE DATABASE_DEFAULT;

DROP TABLE #TempDepartments;
```