

09 / 10 – Data types and joins (date and string)

Data type precedence governs conversion. Inspect implicit conversions (especially in joins) to avoid unexpected results

- Lower precedence data types are implicitly converted to higher precedence data types
- Higher precedence data types explicitly converted to lower precedence data types will almost always result in data loss
- Not all conversions are possible

Example of conversion using date

- `date` is lower than `datetime2`, so date is ***implicitly*** converted up to `datetime2`. Illustration using cast as follows:

```
select cast(a.dt as datetime2)
from (select cast('2023-07-01' as date) dt) a; ->
2023-07-01 00:00:00.000
```

- `date` is lower than `datetime2`, so when `datetime2` is ***explicitly*** converted to `date`, data is lost

```
select cast(a.dt as date)
from (
    select cast('2023-07-01 13:15:12' as datetime2) dt
) a; ->
2023-07-01
```

Check collation and byte length when using strings

- Check database collation

```
select convert(nvarchar(128)
              , databasepropertyex('CourseDB', 'collation')); ->
SQL_Latin1_General_CP1_CI_AS
```

```
select
    case when 'biztrix.us'
           = 'BizTrix.us' collate sql_latin1_general_cp1_cs_as
    then 1
    else 0
end isEqual; -> 0
```

Note: A dashed yellow box highlights the collation 'Case Insensitive/Sensitive' in the SQL query above.

- Check byte length

```
select case when cast('biztrix.us' as char(3)) =
cast('biztrix.us' as varchar(20)) then 1 else 0 end; ->
0
```