Transform data with SQL

Data analysts usually need to convert data from one format to another to complete an analysis. But what if you are using SQL rather than a spreadsheet? Just like spreadsheets, SQL uses standard rules to convert one type of data to another. If you are wondering why data transformation is an important skill to have as a data analyst, think of it like being a driver who is able to change a flat tire. Being able to convert data to the right format speeds you along in your analysis. You don't have to wait for someone else to convert the data for you.

In this reading, you will go over the conversions that can be done using the `CAST` function. There are also more specialized functions like `COERCION` to work with big numbers, and `UNIX_DATE` to work with dates. `UNIX_DATE` returns the number of days that have passed since January 1, 1970 and is used to compare and work with dates across multiple time zones. You will likely use `CAST` most often.

Common conversions

The following table summarizes some of the more common conversions made with the `CAST` function. Refer to Conversion Rules in Standard SQL for a full list of functions and associated rules.

Starting with	`CAST` function can convert to:
Numeric (number)	- Integer - Numeric (number) - Big number - Floating integer - String
String	- Boolean - Integer - Numeric (number) - Big number - Floating integer - String - Bytes - Date - Date time - Time - Timestamp
Date	- String - Date - Date time - Timestamp

The `CAST` function (syntax and examples)

`CAST` is an American National Standards Institute (ANSI) function used in lots of programming languages, including BigQuery. This section provides the BigQuery syntax and examples of converting the data types in the first column of the previous table. The syntax for the `CAST` function is as follows:

```
CAST(expression AS typename)
```

Where **expression** is the data to be converted and **typename** is the data type to be returned.

Converting a number to a string

The following `CAST` statement returns a string from a numeric identified by the variable `MyCount` in the table called `MyTable`.

```
SELECT CAST(MyCount AS STRING) FROM MyTable
```

In the above SQL statement, the following occurs:

- `SELECT` indicates that you will be selecting data from a table
- `CAST` indicates that you will be converting the data you select to a different data type
- `AS` comes before and identifies the data type which you are casting to
- `STRING` indicates that you are converting the data to a string
- `FROM` indicates which table you are selecting the data from

Converting a string to a number

The following `CAST` statement returns an integer from a string identified by the variable `MyVarcharCol` in the table called `MyTable`. (An integer is any whole number.)

```
SELECT CAST(MyVarcharCol AS INT) FROM MyTable
```

In the above SQL statement, the following occurs:

- `SELECT` indicates that you will be selecting data from a table
- `CAST` indicates that you will be converting the data you select to a different data type
- `AS` comes before and identifies the data type which you are casting to
- `INT` indicates that you are converting the data to an integer
- `FROM` indicates which table you are selecting the data from

Converting a date to a string

The following `CAST` statement returns a string from a date identified by the variable `MyDate` in the table called `MyTable`.

```
SELECT CAST(MyDate AS STRING) FROM MyTable
```

In the above SQL statement, the following occurs:

- `SELECT` indicates that you will be selecting data from a table
- `CAST` indicates that you will be converting the data you select to a different data type
- `AS` comes before and identifies the data type which you are casting to
- `STRING` indicates that you are converting the data to a string
- `FROM` indicates which table you are selecting the data from

Converting a date to a datetime

Datetime values have the format of YYYY-MM-DD hh: mm: ss format, so date and time are retained together. The following `CAST` statement returns a datetime value from a date.

```
SELECT CAST (MyDate AS DATETIME) FROM MyTable
```

In the above SQL statement, the following occurs:

- `SELECT` indicates that you will be selecting data from a table
- `CAST` indicates that you will be converting the data you select to a different data type
- `AS` comes before and identifies the data type which you are casting to
- `DATETIME` indicates that you are converting the data to a datetime value
- `FROM` indicates which table you are selecting the data from

The `SAFE_CAST` function

Using the `CAST` function in a query that fails returns an error in BigQuery. To avoid errors in the event of a failed query, use the `SAFE_CAST` function instead. The `SAFE_CAST` function returns a value of Null instead of an error when a query fails.

The syntax for `SAFE_CAST` is the same as for `CAST`. Simply substitute the function directly in your queries. The following `SAFE_CAST` statement returns a string from a date.

```
SELECT SAFE_CAST(MyDate AS STRING) FROM MyTable
```

More information

Browse these resources for more information about data conversion using other SQL dialects (instead of BigQuery):

- CAST and CONVERT: SQL Server reference documentation
- MySQL CAST Functions and Operators: MySQL reference documentation
- How to: SQL Type Casting: Blog about type casting that has links to other SQL short guides

Mark as completed

Like Dislike Report an issue