causes the cast to use no more than *N* bytes of the argument. Values shorter than *N* bytes are padded with `0x00` bytes to a length of *N*.

- `CHAR[(N)] [charset_info]`

  Produces a string with the `CHAR` data type. If the optional length *N* is given, `CHAR(N)` causes the cast to use no more than *N* characters of the argument. No padding occurs for values shorter than *N* characters.

  With no *charset_info* clause, `CHAR` produces a string with the default character set. To specify the character set explicitly, these *charset_info* values are permitted:

  - `CHARACTER SET charset_name`: Produces a string with the given character set.
  - `ASCII`: Shorthand for `CHARACTER SET latin1`.
  - `UNICODE`: Shorthand for `CHARACTER SET ucs2`.

  In all cases, the string has the character set default collation.

- `DATE`

  Produces a `DATE` value.

- `DATETIME`

  Produces a `DATETIME` value.

- `DECIMAL[(M[,D])]`

  Produces a `DECIMAL` value. If the optional *M* and *D* values are given, they specify the maximum number of digits (the precision) and the number of digits following the decimal point (the scale).

- `DOUBLE`

  Produces a `DOUBLE` result. Added in MySQL 8.0.17.

- `FLOAT[(p)]`

  If the precision *p* is not specified, produces a result of type `FLOAT`. If *p* is provided and 0 <= < *p* <= 24, the result is of type `FLOAT`. If 25 <= *p* <= 53, the result is of type `DOUBLE`. If *p* < 0 or *p* > 53, an error is returned. Added in MySQL 8.0.17.

- `JSON`

  Produces a `JSON` value. For details on the rules for conversion of values between `JSON` and other types, see Comparison and Ordering of JSON Values.

- `NCHAR[(N)]`

  Like `CHAR`, but produces a string with the national character set. See Section 10.3.7, “The National Character Set”.

  Unlike `CHAR`, `NCHAR` does not permit trailing character set information to be specified.

- `REAL`