Database-Specific Mappings

This page lists the mappings of database-specific types to the E2E Bridge base types.

Oracle

These mappings are valid for Oracle version 8 or higher.

Data Type	Description	E2E Bridge Base Type
`BFILE`	Contains a locator to a large binary file stored outside the database. Enables byte stream I/O access to external LOBs residing on the database server. Maximum size is 4 gigabytes.	Blob
`BIGINT`	An integer data type with a precision of 19 decimal digits.	Integer
`BINARY`	Enables storage of binary data up to 4,096 bytes.	Blob
`BINARY_FLOAT`	32-bit floating point number. This data type requires 5 bytes, including the length byte.	Float
`BINARY_DOUBLE`	64-bit floating point number. This data type requires 9 bytes, including the length byte.	Float
`BIT`	Enables your application to store a bit unconstrained by character semantics.	Boolean
`BLOB`	A binary large object. Maximum size is 2 gigabytes.	Blob
`CHAR`	Fixed length character data of length size bytes. Maximum size is 4,096 bytes. Default and minimum size is 1 byte.	String
`CLOB`	A character large object containing single-byte characters. Both fixed-width and variable-width character sets are supported, both using the CHAR database character set. Maximum size is 2 gigabytes. When specifying **String** objects as **CLOB** parameters on stored procedures, you have to mark them with tag **nativeType=CLOB**. Otherwise the xUML Runtime might fail on doing the conversion (see Troubleshooting the SQL Adapter).	String
`DATE`	Valid date range from January 1, 4712 BC to December 31, 4712 AD.	DateTime
`DECIMAL`	A number that can be measured in terms of precision (decimal value) or scale (fractional value). You can measure precision by using DECIMAL (*p*). You can measure scale by using NUMERIC (*p*, *s*). Precisions larger than the one you specify are acceptable, but smaller ones are not.	Float
`DOUBLE PRECISION`	Contains a precision defined during implementation, which must be greater than the precision of REAL.	Float
`FLOAT`	Enables you to specify the precision. The resulting precision must be at least as large as the precision you request. You can specify a precision of some value by typing FLOAT (*p*). For example, a portable application, may use a single precision on one platform and double precision on another.	Float
`INTEGER`	An integer value whose precision (the number of decimal values or bits that can be stored) is defined upon implementation.	Integer
`INTERVAL YEAR [(year_precision)] TO MONTH`	Stores a period of time in years and months, where *year_precision* is the number of digits in the YEAR datetime field. Accepted values are 0 to 9. The default is 2. The size is fixed at 5 bytes.	**Not supported**

On this Page:

- Oracle
- MySQL
- SQL Server

Related Pages:

- Troubleshooting the SQL Adapter
- MySQL documentation

`INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds)]`	Stores a period of time in days, hours, minutes, and seconds, where *day_precision* is the maximum number of digits in the DAY datetime field. Accepted values are 0 to 9. The default is 2. *fractional_seconds_precision* is the number of digits in the fractional part of the SECOND field. Accepted values are 0 to 9. The default is 6. The size is fixed at 11 bytes.	**Not supported**
`LONG`	Character data of variable length up to 2 gigabytes, or 231 -1 bytes.	String
`LONG RAW`	Raw binary data of variable length up to 2 gigabytes.	Blob
`LONG VARBINARY`	Stores but does not interpret up to 2 gigabytes of variable binary data.	Blob
`LONG VARCHAR`	Variable-length character string having maximum length size bytes. Maximum size is 2 gigabytes, and minimum is 1. You must specify size for a VARCHAR2.	String
`NCHAR [(size)]`	Fixed-length character data of length *size* characters. The number of bytes can be up to two times *size* for AL16UTF16 encoding and three times *size* for UTF8 encoding. Maximum *size* is determined by the national character set definition, with an upper limit of 2000 bytes. Default and minimum *size* is 1 character.	String
`NCLOB`	A character large object containing Unicode characters. Both fixed-width and variable-width character sets are supported, both using the database national character set. Maximum size is (4 gigabytes - 1) * (database block size). Stores national character set data.	String
`NUMBER`	Number having precision *p* and scale *s*. The precision *p* can range from 1 to 38. The scale *s* can range from -84 to 127. A number with no precision now maps to DOUBLE PRECISION in Oracle compatibility mode.	Integer
`NUMERIC`	A number that can be measured in terms of precision (decimal value) or scale (fractional value). You can measure precision by using DECIMIAL (*p*). You can measure scale by using NUMERIC (*p, s*). The scale cannot be negative and cannot be larger than the number itself.	Float
`NVARCHAR2 (size)`	Variable-length Unicode character string having maximum length *size* characters. The number of bytes can be up to two times *size* for AL16UTF16 encoding and three times *size* for UTF8 encoding. Maximum *size* is determined by the national character set definition, with an upper limit of 4000 bytes. You must specify *size* for NVARCHAR2.	String
`RAW`	Raw binary data of length size bytes. Maximum size is 4,096 bytes. You must specify size for a RAW value.	Blob
`REAL`	Enables you to request a single-precision floating point with no options. The precision is chosen by the implementation and is normally the default single-precision data type on the hardware platform.	Float
`ROWID`	A 16-byte hexadecimal string representing the unique address of a row in its table. ROWID is primarily for values returned by the ROWID pseudo-column.	String
`SMALLINT`	An integer value whose precision is defined upon implementation but whose value is no greater than the implementation of INTEGER.	Integer
`TIME`	Stores a time value in terms of hours minutes and seconds. Hours are represented by two digits ranging from 00 through 23. Minutes are also represented by two digits ranging from 00 through 59. The seconds value ranges from 00 through 60.	DateTime
`TIMESTAMP`	Stores the year, month, and day values of a date and the hour, minute, second value of time. TIMESTAMP length and restrictions correspond to DATE and TIME values, except that in TIME the default is 0 and in TIMESTAMP it is 6.	DateTime
`TINYINT`	An integer with a precision of 1 byte (-128 to +127).	Integer
`UROWID [(size)]`	Base 64 string representing the logical address of a row of an index-organized table. The optional *size* is the size of a column of type UROWID. The maximum size and default is 4000 bytes.	**Not supported**
`VARBINARY`	Stores but does not interpret variable binary data.	Blob

`VARCHAR`	See VARCHAR2	String
`VARCHAR2`	Variable-length character string with a maximum length size of 4,096 bytes (minimum is 1). You must specify size for a VARCHAR2.	String

MySQL

Refer to the MySQL documentation for more information on specific types. You can click on the MySQL type to jump directly to the corresponding MySQL documentation page.

Data Type	Description	E2E Bridge Base Type
`BIGINT`	A large integer. The signed range is **-9223372036854775808** to **9223372036854775807**. The unsigned range is **0** to **18446744073709551615**.	Integer
`BINARY(M)`	The `BINARY` type is similar to the `CHAR` type, but stores binary byte strings rather than non-binary character strings. **`M`** represents the column length in bytes.	Blob
`BIT`	A bit-field type. **`M`** indicates the number of bits per value, from 1 to 64. The default is 1 if **`M`** is omitted.	**Not supported**
`BLOB`	A `BLOB` column with a maximum length of 65,535 (2^{16} 1) bytes. Each `BLOB` value is stored using a 2-byte length prefix that indicates the number of bytes in the value. An optional length **`M`** can be given for this type. If this is done, MySQL creates the column as the smallest `BLOB` type large enough to hold values **`M`** bytes long.	Blob
`CHAR`	The length of a `CHAR` column is fixed to the length that you declare when you create the table. The length can be any value from 0 to 255. When `CHAR` values are stored, they are right-padded with spaces to the specified length. When `CHAR` values are retrieved, trailing spaces are removed unless the `PAD_CHAR_TO_FULL_LENGTH` SQL mode is enabled.	String
`DATE`	A date. The supported range is **'1000-01-01'** to **'9999-12-31'**. MySQL displays `DATE` values in **'YYYY-MM-DD'** format, but permits assignment of values to `DATE` columns using either strings or numbers.	DateTime
`DATETIME`	A date and time combination. The supported range is **'1000-01-01 00:00:00.000000'** to **'9999-12-31 23:59:59.999999'**. MySQL displays `DATETIME` values in **'YYYY-MM-DD HH:MM:SS[.fraction]'** format, but permits assignment of values to `DATETIME` columns using either strings or numbers. An optional ***fsp*** value in the range from 0 to 6 may be given to specify fractional seconds precision. A value of 0 signifies that there is no fractional part. If omitted, the default precision is 0.	DateTime
`DEC` `DECIMAL`	A packed "exact" fixed-point number. **`M`** is the total number of digits (the precision) and **`D`** is the number of digits after the decimal point (the scale). The decimal point and (for negative numbers) the "**-**" sign are not counted in **`M`** . If **`D`** is 0, values have no decimal point or fractional part. The maximum number of digits (**`M`**) for `DECIMAL` is 65. The maximum number of supported decimals (**`D`**) is 30. If **`D`** is omitted, the default is 0. If **`M`** is omitted, the default is 10. `UNSIGNED`, if specified, disallows negative values. All basic calculations (**+, -, *, /**) with `DECIMAL` columns are done with a precision of 65 digits.	Float
`DOUBLE`	A normal-size (double-precision) floating-point number. Permissible values are **-1.7976931348623157E+308** to**-2.2250738585072014E-308**, **0**, and **2.2250738585072014E-308** to **1.7976931348623157E+308**. These are the theoretical limits, based on the IEEE standard. The actual range might be slightly smaller depending on your hardware or operating system. **`M`** is the total number of digits and **`D`** is the number of digits following the decimal point. If **`M`** and **`D`** are omitted, values are stored to the limits permitted by the hardware. A double-precision floating-point number is accurate to approximately 15 decimal places. `UNSIGNED`, if specified, disallows negative values.	Float

`ENUM`	An enumeration. A string object that can have only one value, chosen from the list of values **'** ***value1*** **','** ***value2*** **'**, **...**, **NULL** or the special **''** error value. `ENUM` values are represented internally as integers. An `ENUM` column can have a maximum of 65,535 distinct elements. (The practical limit is less than 3000.) A table can have no more than 255 unique element list definitions among its `ENUM` and `SET` columns considered as a group.	String
`FLOAT`	A small (single-precision) floating-point number. Permissible values are **-3.402823466E+38** to **-1.175494351E-38**, **0**, and **1.175494351E-38** to **3.402823466E+38**. These are the theoretical limits, based on the IEEE standard. The actual range might be slightly smaller depending on your hardware or operating system. **`M`** is the total number of digits and **`D`** is the number of digits following the decimal point. If **`M`** and **`D`** are omitted, values are stored to the limits permitted by the hardware. A single-precision floating-point number is accurate to approximately 7 decimal places. `UNSIGNED`, if specified, disallows negative values. Using `FLOAT` might give you some unexpected problems because all calculations in MySQL are done with double precision.	Float
`INT` `INTEGER`	A normal-size integer. The signed range is **-2147483648** to **2147483647**. The unsigned range is **0** to**4294967295**.	Integer
`LONGBLOB`	A `BLOB` column with a maximum length of 4,294,967,295 or 4GB (2^{32} 1) bytes. The effective maximum length of `LONGBLOB` columns depends on the configured maximum packet size in the client/server protocol and available memory. Each `LONGBLOB` value is stored using a 4-byte length prefix that indicates the number of bytes in the value	Blob
`LONGTEXT`	A `TEXT` column with a maximum length of 4,294,967,295 or 4GB (2^{32} 1) characters. The effective maximum length is less if the value contains multibyte characters. The effective maximum length of `LONGTEXT` columns also depends on the configured maximum packet size in the client/server protocol and available memory. Each `LONGTEXT` value is stored using a 4-byte length prefix that indicates the number of bytes in the value.	String
`MEDIUMBLOB`	A `BLOB` column with a maximum length of 16,777,215 (2^{24} 1) bytes. Each `MEDIUMBLOB` value is stored using a 3-byte length prefix that indicates the number of bytes in the value.	Blob
`MEDIUMINT`	A medium-sized integer. The signed range is **-8388608** to **8388607**. The unsigned range is **0** to **16777215**.	Integer
`MEDIUMTEXT`	A `TEXT` column with a maximum length of 16,777,215 (2^{24} 1) characters. The effective maximum length is less if the value contains multibyte characters. Each `MEDIUMTEXT` value is stored using a 3-byte length prefix that indicates the number of bytes in the value.	String
`SET`	A set. A string object that can have zero or more values, each of which must be chosen from the list of values**'** ***value1*** **'**, **'** ***value2*** **'**, **...** `SET` values are represented internally as integers. A `SET` column can have a maximum of 64 distinct members. A table can have no more than 255 unique element list definitions among its `ENUM` and `SET` columns considered as a group.	String
`SMALLINT`	A small integer. The signed range is **-32768** to **32767**. The unsigned range is **0** to **65535**.	Integer
`TEXT`	A `TEXT` column with a maximum length of 65,535 (2^{16} 1) characters. The effective maximum length is less if the value contains multibyte characters. Each `TEXT` value is stored using a 2-byte length prefix that indicates the number of bytes in the value. An optional length **`M`** can be given for this type. If this is done, MySQL creates the column as the smallest `TEXT` type large enough to hold values **`M`** characters long.	String

TIME	A time. The range is **'-838:59:59.000000'** to **'838:59:59.000000'**. MySQL displays TIME values in**'HH:MM:SS[.fraction]'** format, but permits assignment of values to TIME columns using either strings or numbers. An optional ***fsp*** value in the range from 0 to 6 may be given to specify fractional seconds precision. A value of 0 signifies that there is no fractional part. If omitted, the default precision is 0.	**Not supported**
TIMESTAMP	A timestamp. The range is **'1970-01-01 00:00:01.000000'** UTC to **'2038-01-19 03:14:07.999999'** UTC. TIMESTAMP values are stored as the number of seconds since the epoch (**'1970-01-01 00:00:00'** UTC). A TIMESTAMP cannot represent the value **'1970-01-01 00:00:00'** because that is equivalent to 0 seconds from the epoch and the value 0 is reserved for representing **'0000-00-00 00:00:00'**, the “zero” TIMESTAMP value. An optional ***fsp*** value in the range from 0 to 6 may be given to specify fractional seconds precision. A value of 0 signifies that there is no fractional part. If omitted, the default precision is 0.	DateTime
TINYBLOB	A BLOB column with a maximum length of 255 (2^{8} 1) bytes. Each TINYBLOB value is stored using a 1-byte length prefix that indicates the number of bytes in the value.	Blob
TINYINT	A very small integer. The signed range is **-128** to **127**. The unsigned range is **0** to **255**.	Integer
TINYTEXT	A TEXT column with a maximum length of 255 (2^{8} 1) characters. The effective maximum length is less if the value contains multi-byte characters. Each TINYTEXT value is stored using a 1-byte length prefix that indicates the number of bytes in the value.	String
VARBINARY	The VARBINARY type is similar to the VARCHAR type, but stores binary byte strings rather than non-binary character strings. **M** represents the maximum column length in bytes.	Blob
VARCHAR	Values in VARCHAR columns are variable-length strings. The length can be specified as a value from 0 to 65,535. The effective maximum length of a VARCHAR is subject to the maximum row size (65,535 bytes, which is shared among all columns) and the character set used.	String
YEAR	A year in four-digit format. MySQL displays YEAR values in **YYYY** format, but permits assignment of values to YEAR columns using either strings or numbers. Values display as **1901** to **2155**, and **0000**. **Note:** The **YEAR(2)** data type is deprecated and support for it is removed in MySQL 5.7.5.	String

SQL Server

Refer to the SQL Server Language documentation for more information on specific Transact-SQL types. You can click on the type names to jump directly to the corresponding documentation page.

Data Type	Description	E2E Bridge Base Type
BIGINT	A large integer. The signed range is from **-9223372036854775808 (-2 ^{63} 1)** to **9223372036854775807 (2 ^{63} 1)**.	Integer
BINARY	The BINARY type stores fixed-length binary data with a maximum length of 8,000 bytes.	Blob
BIT	An integer data type that can take a value of 1, 0, or NULL.	Boolean
CHAR	CHAR is a fixed width character string. It stores fixed-length non-Unicode character data with a maximum length of 8,000 characters.	String
DATETIME	A date and time combination. DATETIME contains date and time data from **January 1, 1753**, to **December 31, 9999**, with an accuracy of three-hundredths of a second, or **3.33 milliseconds**.	DateTime
DECIMAL	Fixed precision and scale numeric data, which allows numbers from **(-10 ^{38} + 1)** to **(10 ^{38} - 1)**.	Float

`FLOAT`	The `FLOAT` type contains floating precision number data with the valid values **-1.79E+308** to **-2.23E-308**, **0** and **2.23E+308 to 1.79E+308**.	Float
`INT`	Integer data from **-2147483648 (-2 ^{31})** to **2147483647 (2 ^{31} -1)**.	Integer
`NCHAR`	`NCHAR` stores fixed-length Unicode data with a maximum length of 4,000 characters.	String
`NUMERIC`	Functionally equivalent to `DECIMAL`.	Float
`NVARCHAR`	The `NVARCHAR` type contains variable-length Unicode data with a maximum size of 2 gigabytes (`NVARCHAR(MAX)`). If using SQLNCLI/ODBC API for SQL Adapter with MS SQL Server, you may need to mark the field as CLOB in your model, when long character data is inserted into a `NVARCHAR(MAX)` field.	String / CLOB
`REAL`	The `REAL` type stores floating precision number data with the valid values **-3.40E+38** to **-1.18E-38**, **0** and **1.18E-38** to **3.40E+38**.	Float
`SMALLDATETIME`	A date and time combination. `SMALLDATETIME` contains date and time data from **January 1, 1900**, to **June 6, 2079**, with an accuracy of **one minute**.	DateTime
`SMALLINT`	A small integer. The signed range is from **-32,768 (-2 ^{15})** to **32,767 (2 ^{15} - 1)**.	Integer
`TEXT`	The `TEXT` type stores variable-length non-Unicode data with a maximum length of **2,147,483,647 (2^{31} 1)** characters. If using SQLNCLI/ODBC API for SQL Adapter with MS SQL Server, you may need to mark the field as CLOB in your model, when long character data is inserted into a `TEXT` field.	String / CLOB
`TINYINT`	A very small integer. The range is **0** to **255**.	Integer
`VARBINARY`	The `VARBINARY` type stores variable-length binary data with a maximum size of 2 gigabytes (`VARBINARY(MAX)`).	Blob
`VARCHAR`	Similar to `VARBINARY`, but stores variable-length non-Unicode data with a maximum of 8,000 characters.	String