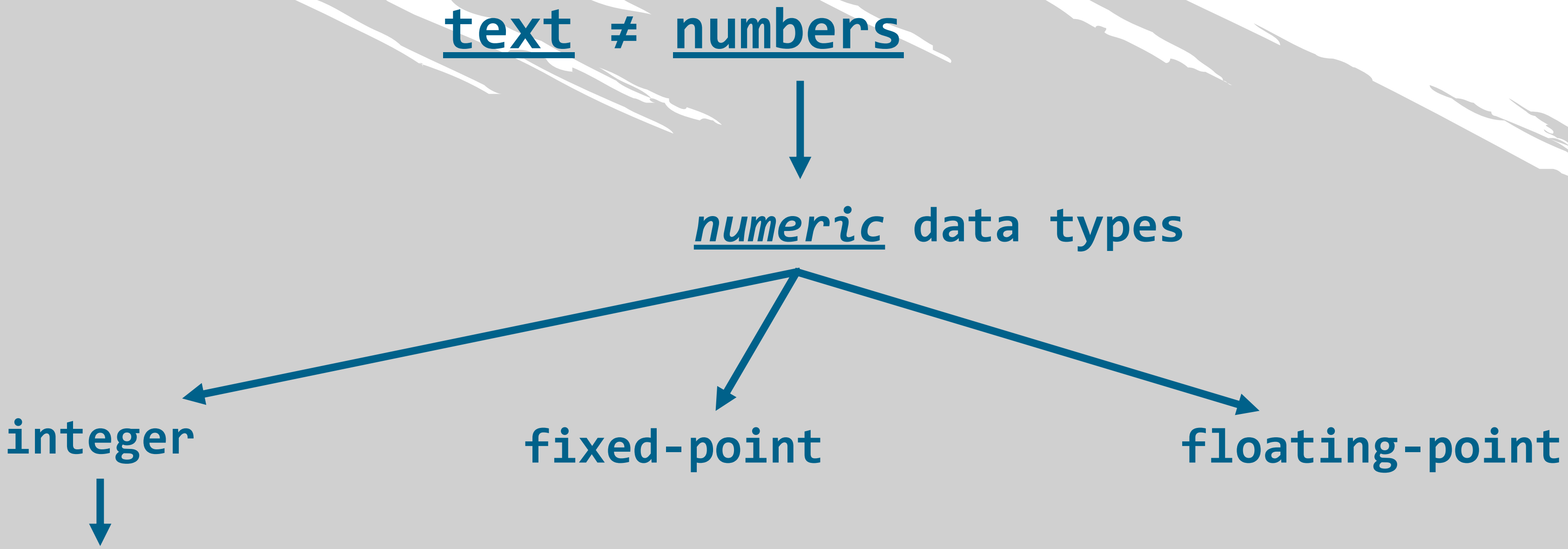
Integers

Integers



text ≠ numbers

numeric data types

integer

fixed-point

floating-point

integers

whole numbers with no decimal point

e.g. 5; 15; -200; 1,000

`INTEGER` `INT`

Integers

numeric data type	*size* (bytes)	minimum value (signed/unsigned)	maximum value (signed/unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32,768	32,767
		0	65,535
MEDIUMINT	3	-8,388,608	8,388,607
		0	16,777,215
INT	4	-2,147,483,648	2,147,483,647
		0	4,294,967,295
BIGINT	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
		0	18,446,744,073,709,551,615

Integers

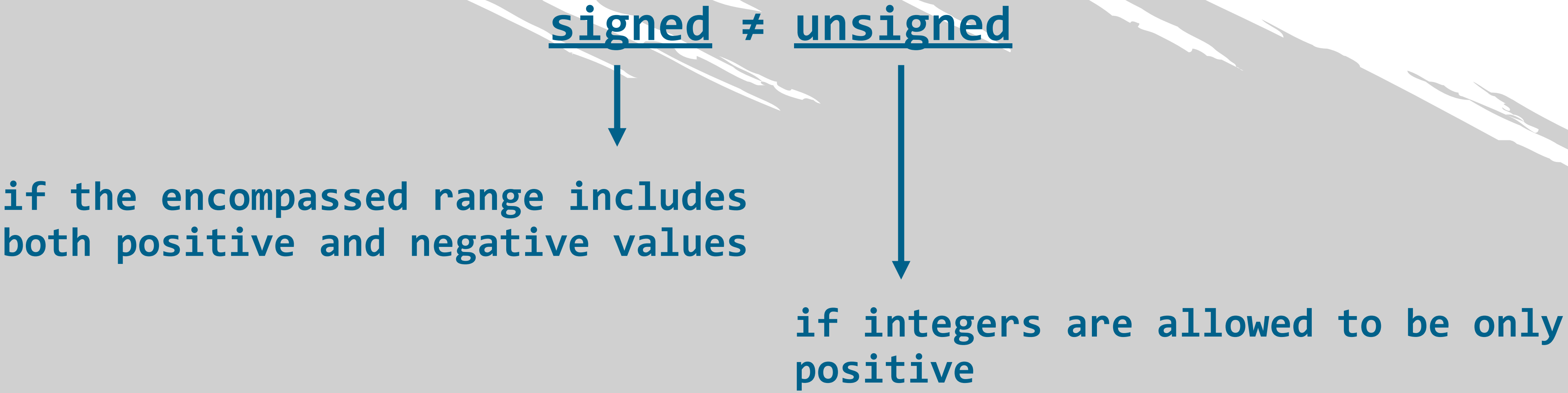
signed ≠ unsigned

if the encompassed range includes both positive and negative values

Integers

numeric data type	size (bytes)	minimum value (signed/unsigned)	maximum value (signed/unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32,768	32,767
		0	65,535
MEDIUMINT	3	-8,388,608	8,388,607
		0	16,777,215
INT	4	-2,147,483,648	2,147,483,647
		0	4,294,967,295
BIGINT	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
		0	18,446,744,073,709,551,615

Integers



signed ≠ **unsigned**

if the encompassed range includes both positive and negative values

if integers are allowed to be only positive

Integers

numeric data type	*size* (bytes)	minimum value (signed/unsigned)	maximum value (signed/unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32,768	32,767
		0	65,535
MEDIUMINT	3	-8,388,608	8,388,607
		0	16,777,215
INT	4	-2,147,483,648	2,147,483,647
		0	4,294,967,295
BIGINT	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
		0	18,446,744,073,709,551,615

Integers

- *integer* data types are ‘signed’ by default

Integers

numeric data type	size (bytes)	minimum value (signed/unsigned)	maximum value (signed/unsigned)
TINYINT	1	-128	127
		0	256
SMALLINT	2	-32,768	32,767
		0	65,535
MEDIUMINT	3	-8,388,608	8,388,607
		0	16,777,215
INT	4	-2,147,483,648	2,147,483,647
		0	4,294,967,295
BIGINT	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
		0	18,446,744,073,709,551,615

Integers

- *integer* data types are ‘signed’ by default

- if you want to use a range containing only positive, ‘unsigned’ values, you would have to specify this in your query

Integers

numeric data type	size (bytes)	minimum value (signed/unsigned)	maximum value (signed/unsigned)
TINYINT	1	-128	127
		0	256
SMALLINT	2	-32,768	32,767
		0	65,535
MEDIUMINT	3	-8,388,608	8,388,607
		0	16,777,215
INT	4	-2,147,483,648	2,147,483,647
		0	4,294,967,295
BIGINT	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
		0	18,446,744,073,709,551,615

Integers

- Why not just use `BIGINT` all the time?

Integers

- Why not just use **BIGINT** all the time?

- e.g. if you are sure that, in a certain column, you won't need an integer smaller than 0 or greater than 100, **TINYINT** would do the job perfectly and you would not need more storage space per data point

Integers

- Why not just use BIGINT all the time?

- e.g. if you are sure that, in a certain column, you won’t need an integer smaller than 0 or greater than 100, TINYINT would do the job perfectly and you would not need more storage space per data point

- a smaller integer type may increase the processing speed