

NUMERIC Datatypes



Numeric Datatypes

- Can be divided into two types:
 - Exact – like 42, 9.37
 - when you store a numeric value in an exact numeric column, you will always be able to retrieve exactly the same value in a query
 - Approximate – where the value you retrieve may be a different number, but very close to the original number



Exact Types

- INTEGER
 - Columns can hold both positive and negative numbers
 - Ranges from -2,147,483,648 to 2,147,483,647
 - Whole numbers



Exact Types

- **SMALL INT**
 - Supports a smaller range of integers
 - From -31,768 to 31,767
- **BIGINT**
 - Supports a larger range than INTEGER.
 - From -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807



Exact types

- There are additional non-standard integer data types
 - MEDIUMINT gives a range of -8,388,608 to 8,388,608 (MYSQL)
 - TINYINT gives a range of -128 to 127(MYSQL and SQL)
- Benefit of using say TINYINT over SMALLINT or INTEGER is for reduced disk space requirements
 - but very large number of rows needed to even save a small amount of space
- Disadvantage is that it will have to change the datatype if it is ported to a DB system that doesn't support it (but the change can be done easily)
- Calculations involving only integers are much faster than those involving decimal and floating point numbers
- **For this course and your assignments – just use INTEGER**



Exact Types

- Decimals
 - Made up of two parts: the total number of digits and the number of digits to the right of the decimal point.

EXAMPLE

9.37 has a total of 3 digits, 2 that are right of the decimal

- Two kinds of decimal
 - NUMERIC
 - DECIMAL



Exact Types

- DECIMAL

DECIMAL(p [,s])

- p represents the precision – total number of digits allowed
- s represents the scale – the total number of digits to the right of the decimal place
 - If scale (s) is omitted – default value is 0



Exact Types

- NUMERIC

NUMERIC (p [,s])

- P represents the precision – ***total number of digits allowed***
- S represents the scale – the total number of digits to the ***right*** of the decimal place
 - If scale (s) is omitted – default value is 0
 - To store 123.89

NUMERIC(5) – can store 124

NUMERIC(5,2) – can store 123.89

NUMERIC(4,1) – can store 123.9

NUMERIC(4,2) – exceeds precision



Exact Types

- Standard SQL states that the difference between NUMERIC and DECIMAL is that NUMERIC columns must have the exact precision specified while DECIMAL might have a larger precision than specified (the precision specifies the minimum).
- Both allow for positive and negative values
- MySQL treats them the same:

In MySQL, NUMERIC is implemented as DECIMAL, so the following remarks about DECIMAL apply equally to NUMERIC.



Exact Types

- To define a column that will hold a value 9.37
DECIMAL (3,2)
- Note – this would not hold a value such as
 - 12.34 – error of “arithmetic overflow”
 - 0.567 – would proceed but round to 0.57
- Choose a precision that will easily hold the maximum range of data



Approximate types

- Implemented a floating point numbers
- FLOAT

 FLOAT(size,d)

- Where the maximum number of digits (size) and the maximum number of digits to the right of the decimal point (d)
- An approximate numerical value can be –ve/+ve or zero, and is usually used to represent **very small or very large quantities (scientific, statistical, financial)**

2.5×10^{14}

