LAB - Integer Data Types

in this lab, you will learn about the SQL Server `DECIMAL` data type and how to use it to store exact numeric values.

To store numbers that have fixed precision and scale, you use the `DECIMAL` data type.

The following shows the syntax of the `DECIMAL` data type:

```
DECIMAL(p,s)
```

In this syntax:

- p is the precision which is the maximum total number of decimal digits that will be stored, both to the left and to the right of the decimal point. The precision has a range from 1 to 38. The default precision is 38.
- s is the scale which is the number of decimal digits that will be stored to the right of the decimal point. The scale has a range from 0 to p (precision). The scale can be specified only if the precision is specified. By default, the scale is zero.

The maximum storage sizes vary, depending on the precision as illustrated in the following table:

Precision	Storage bytes
1 – 9	5
10-19	9
20-28	13
29-38	17

The `NUMERIC` and `DECIMAL` are synonyms, therefore, you can use them interchangeably.

The following declarations are equivalent:

```
DECIMAL(10,2)
NUMERIC(10,2)
```

Because the ISO synonyms for `DECIMAL` are `DEC` and `DEC(p,s)`, you can use either `DECIMAL` or `DEC`:

```
DECIMAL(10,2)
DEC(10,2)
```

Example

Let's take an example of using the `DECIMAL` and `NUMERIC` data types.

First, create a new table that consists of two columns: one decimal and one numeric:

```
CREATE TABLE sql_server_decimal (
    dec_col DECIMAL (4, 2),
    num_col NUMERIC (4, 2)
);
```

Second, insert a new row into the `sql_server_decimal` table:

```
INSERT INTO sql_server_decimal (dec_col, num_col)
VALUES
    (10.05, 20.05);
```

Third, query data from the table:

```
SELECT
    dec_col,
    num_col
FROM
    test.sql_server_decimal;
```

Fourth, the following example attempts to insert a new row into the table with values that exceed the precision and scale specified in the column definition:

```
INSERT INTO test.sql_server_decimal (dec_col, num_col)
VALUES
    (99.999, 12.345);
```

SQL Server issued an error and terminated the statement:

```
Arithmetic overflow error converting numeric to data type numeric.
The statement has been terminated
```

In this lab, you have learned how to use the SQL Server `DECIMAL` data type to store exact numeric values.