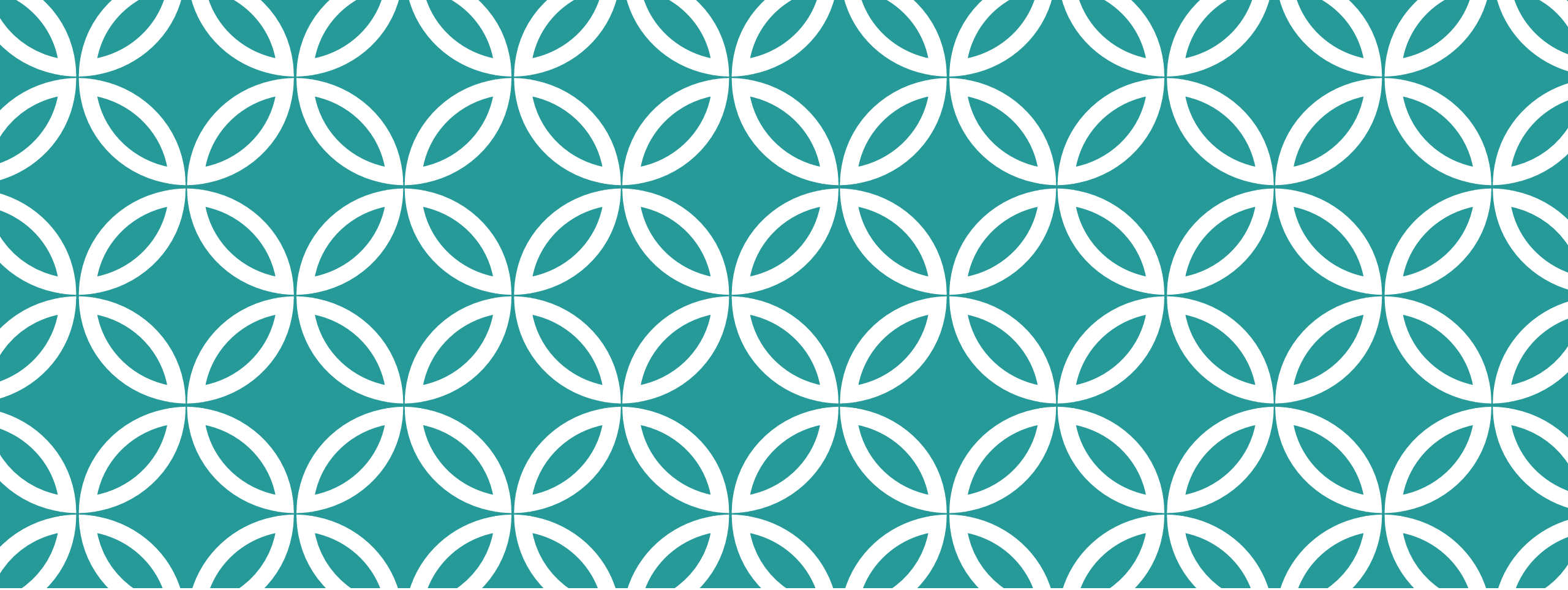

# CHAPTER 16: NUMBER FUNCTIONS

2023

# CHAPTER 16: NUMBER FUNCTIONS

Contents:

1. Ceiling function
2. Floor function
3. Round function
4. Cast function

# CEILING AND FLOOR ROUNDING

Floor() function returns the integer value less than or equal to the value passed in.

Ceiling() function  on the other hand  is the opposite. It returns the same data type as floor but returns the integer equal to or higher than the value passed in. See example below:

```
select list_price , round(list_price,1,1) as rounded_listprice,
ceiling(list_price) as ceiling_price,
floor(list_price) as floor_price
from production.products
```

100 %

Results | Messages

| | list_price | rounded_listprice | ceiling_price | floor_price |
|---|---|---|---|---|
| 1 | 379.99 | 379.90 | 380 | 379 |
| 2 | 749.99 | 749.90 | 750 | 749 |
| 3 | 999.99 | 999.90 | 1000 | 999 |
| 4 | 2899.99 | 2899.90 | 2900 | 2899 |
| 5 | 1320.99 | 1320.90 | 1321 | 1320 |
| 6 | 469.99 | 469.90 | 470 | 469 |
| 7 | 3999.99 | 3999.90 | 4000 | 3999 |
| 8 | 1799.99 | 1799.90 | 1800 | 1799 |
| 9 | 2999.99 | 2999.90 | 3000 | 2999 |
| 10 | 1549.00 | 1549.00 | 1549 | 1549 |

# ROUND

The ROUND() function rounds a number or value to a specific number of decimal places that you wish.

Example: It has been rounded to 0 decimal. Pay attention to the red highlighted blocks.

```
select * from production.products
```

100 %

Results | Messages

| | product_id | product_name | brand_id | category_id | model_year | list_price |
|---|---|---|---|---|---|---|
| 1 | 1 | Trek 820 - 2016 | 9 | 6 | 2016 | 379.99 |
| 2 | 2 | Ritchey Timberwolf Frameset - 2016 | 5 | 6 | 2016 | 749.99 |
| 3 | 3 | Surly Wednesday Frameset - 2016 | 8 | 6 | 2016 | 999.99 |
| 4 | 4 | Trek Fuel EX 8 29 - 2016 | 9 | 6 | 2016 | 2899.99 |
| 5 | 5 | Heller Shagamaw Frame - 2016 | 3 | 6 | 2016 | 1320.99 |
| 6 | 6 | Surly Ice Cream Truck Frameset - 2016 | 8 | 6 | 2016 | 469.99 |
| 7 | 7 | Trek Slash 8 27.5 - 2016 | 9 | 6 | 2016 | 3999.99 |
| 8 | 8 | Trek Remedy 29 Carbon Frameset - 2016 | 9 | 6 | 2016 | 1799.99 |
| 9 | 9 | Trek Conduit+ - 2016 | 9 | 5 | 2016 | 2999.99 |

```
select * , round(list_price,0) as rounded_listprice from production.products
```

100 %

Results | Messages

| | product_id | product_name | brand_id | category_id | model_year | list_price | rounded_listprice |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Trek 820 - 2016 | 9 | 6 | 2016 | 379.99 | 380.00 |
| 2 | 2 | Ritchey Timberwolf Frameset - 2016 | 5 | 6 | 2016 | 749.99 | 750.00 |
| 3 | 3 | Surly Wednesday Frameset - 2016 | 8 | 6 | 2016 | 999.99 | 1000.00 |
| 4 | 4 | Trek Fuel EX 8 29 - 2016 | 9 | 6 | 2016 | 2899.99 | 2900.00 |
| 5 | 5 | Heller Shagamaw Frame - 2016 | 3 | 6 | 2016 | 1320.99 | 1321.00 |
| 6 | 6 | Surly Ice Cream Truck Frameset - 2016 | 8 | 6 | 2016 | 469.99 | 470.00 |
| 7 | 7 | Trek Slash 8 27.5 - 2016 | 9 | 6 | 2016 | 3999.99 | 4000.00 |
| 8 | 8 | Trek Remedy 29 Carbon Frameset - 2016 | 9 | 6 | 2016 | 1799.99 | 1800.00 |
| 9 | 9 | Trek Conduit+ - 2016 | 9 | 5 | 2016 | 2999.99 | 3000.00 |

# ROUND CONTINUED

Round(value -1) round the value to the nearest tens

Round(value -2) round the value to the nearest hundreds

Round(value -3) round the value to the nearest thousands and you can move in that order.

```
select * from production.products
```

100 %

Results | Messages

| | product_id | product_name | brand_id | category_id | model_year | list_price |
|---|---|---|---|---|---|---|
| 1 | 1 | Trek 820 - 2016 | 9 | 6 | 2016 | 379.99 |
| 2 | 2 | Ritchey Timberwolf Frameset - 2016 | 5 | 6 | 2016 | 749.99 |
| 3 | 3 | Surly Wednesday Frameset - 2016 | 8 | 6 | 2016 | 999.99 |
| 4 | 4 | Trek Fuel EX 8 29 - 2016 | 9 | 6 | 2016 | 2899.99 |
| 5 | 5 | Heller Shagamaw Frame - 2016 | 3 | 6 | 2016 | 1320.99 |
| 6 | 6 | Surly Ice Cream Truck Frameset - 2016 | 8 | 6 | 2016 | 469.99 |
| 7 | 7 | Trek Slash 8 27.5 - 2016 | 9 | 6 | 2016 | 3999.99 |
| 8 | 8 | Trek Remedy 29 Carbon Frameset - 2016 | 9 | 6 | 2016 | 1799.99 |
| 9 | 9 | Trek Conduit+ - 2016 | 9 | 5 | 2016 | 2999.99 |

```
select * , round(list_price,-1) as rounded_listprice from production.products
```

100 %

Results | Messages

| | product_id | product_name | brand_id | category_id | model_year | list_price | rounded_listprice |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Trek 820 - 2016 | 9 | 6 | 2016 | 379.99 | 380.00 |
| 2 | 2 | Ritchey Timberwolf Frameset - 2016 | 5 | 6 | 2016 | 749.99 | 750.00 |
| 3 | 3 | Surly Wednesday Frameset - 2016 | 8 | 6 | 2016 | 999.99 | 1000.00 |
| 4 | 4 | Trek Fuel EX 8 29 - 2016 | 9 | 6 | 2016 | 2899.99 | 2900.00 |
| 5 | 5 | Heller Shagamaw Frame - 2016 | 3 | 6 | 2016 | 1320.99 | 1320.00 |
| 6 | 6 | Surly Ice Cream Truck Frameset - 2016 | 8 | 6 | 2016 | 469.99 | 470.00 |
| 7 | 7 | Trek Slash 8 27.5 - 2016 | 9 | 6 | 2016 | 3999.99 | 4000.00 |
| 8 | 8 | Trek Remedy 29 Carbon Frameset - 2016 | 9 | 6 | 2016 | 1799.99 | 1800.00 |
| 9 | 9 | Trek Conduit+ - 2016 | 9 | 5 | 2016 | 2999.99 | 3000.00 |

# CAST

The CAST() function returns the expression converted to the target data type.

The syntax:

```
CAST ( expression AS target_type[length])
```

In the above syntax:

1. The expression can be a literal value or a valid expression of any type that will be converted.

2. The target_type is the target data type to which you want to convert the expression.

Note that it cannot be an alias data type.

3. The length is an *optional* integer that specifies the length of the target type, eg varchar(*20*).

The length defaults to 30.

# CAST EXAMPLE

```
SELECT CAST(5.95 AS INT) result;
```

100 %

Results | Messages

| | result |
|---|---|
| 1 | 5 |

CAST() function to convert the monthly sales in 2017 to integer values.

```
SELECT
    MONTH(order_date) month,
    CAST(SUM(quantity * list_price * (1 - discount)) AS INT) amount
FROM sales.orders o
    INNER JOIN sales.order_items i ON o.order_id = i.order_id
WHERE
    YEAR(order_date) = 2017
GROUP BY
    MONTH(order_date)
ORDER BY
    month;
```

100 %

Results | Messages

| | month | amount |
|---|---|---|
| 1 | 1 | 285616 |
| 2 | 2 | 312923 |
| 3 | 3 | 308911 |
| 4 | 4 | 227290 |
| 5 | 5 | 268233 |
| 6 | 6 | 378865 |
| 7 | 7 | 229995 |
| 8 | 8 | 290553 |
| 9 | 9 | 293405 |
| 10 | 10 | 310328 |
| 11 | 11 | 281577 |
| 12 | 12 | 259505 |

# CAST EXAMPLES

```
SELECT CAST(14.85 AS int);
Result: 14          (result is truncated)

SELECT CAST(14.85 AS float);
Result: 14.85       (result is not truncated)

SELECT CAST(15.6 AS varchar);
Result: '15.6'

SELECT CAST(15.6 AS varchar(4));
Result: '15.6'

SELECT CAST('15.6' AS float);
Result: 15.6

SELECT CAST('2014-05-02' AS datetime);
Result: '2014-05-02 00:00:00.000'
```

# Practice ROUND / CEILING / FLOOR … FUNCTIONS

1. Round the following number 235.415 , to 2 decimal label it as RoundValue
2. Round the following number 748.58 , to the nearest thousand label it as RoundValue
3. Calculate the ceil value for the following number 25.75 label it ‘CeilValue’
4. Calculate the ceil value for the following number 43.1 label it ‘CeilValue’
5. Calculate the floor value for the following number 25.75 label it ‘FloorValue’
6. Calculate the floor value for the following number -13.5 label it ‘FloorValue’
7. Calculate the ceiling of the target column of the longitude data

# Practice CAST

1. Convert number of reviews into an integer.