DATA AND
ARTIFICIAL INTELLIGENCE

Functions in SQL

Learning Objectives

By the end of this lesson, you will be able to:

- Illustrate SQL functions
- Identify aggregate functions
- Outline date and time, numeric, and advance functions
- List general, duplicate, and inline functions

Understanding SQL Functions

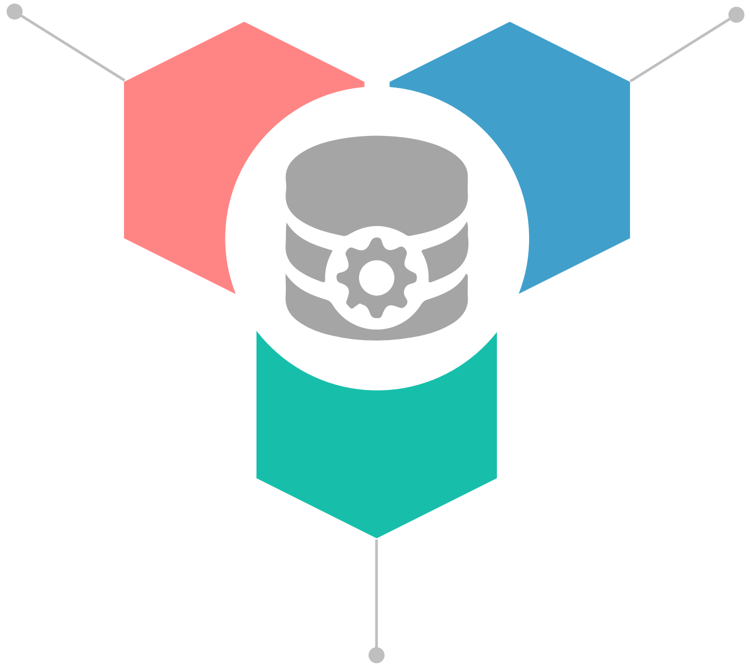
Understanding SQL Functions

- SQL functions are basic subprograms used extensively to handle or manipulate data.

- SQL functions enhance database speed and performance.

- SQL functions are short programs with one or more input parameters but just one output value.

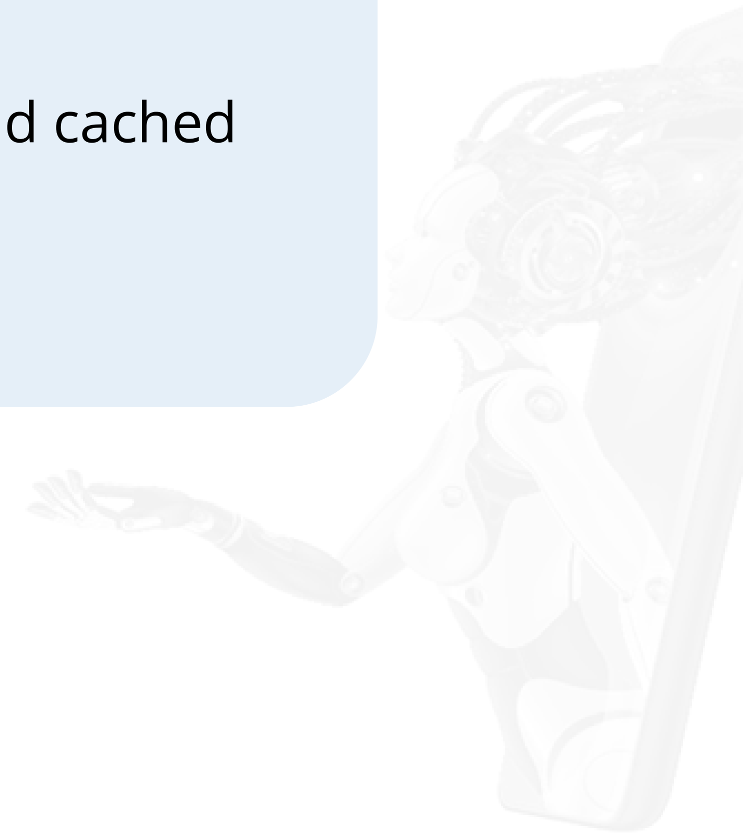
Advantages of SQL Functions



Boost the database's efficiency and productivity

Are compiled and cached

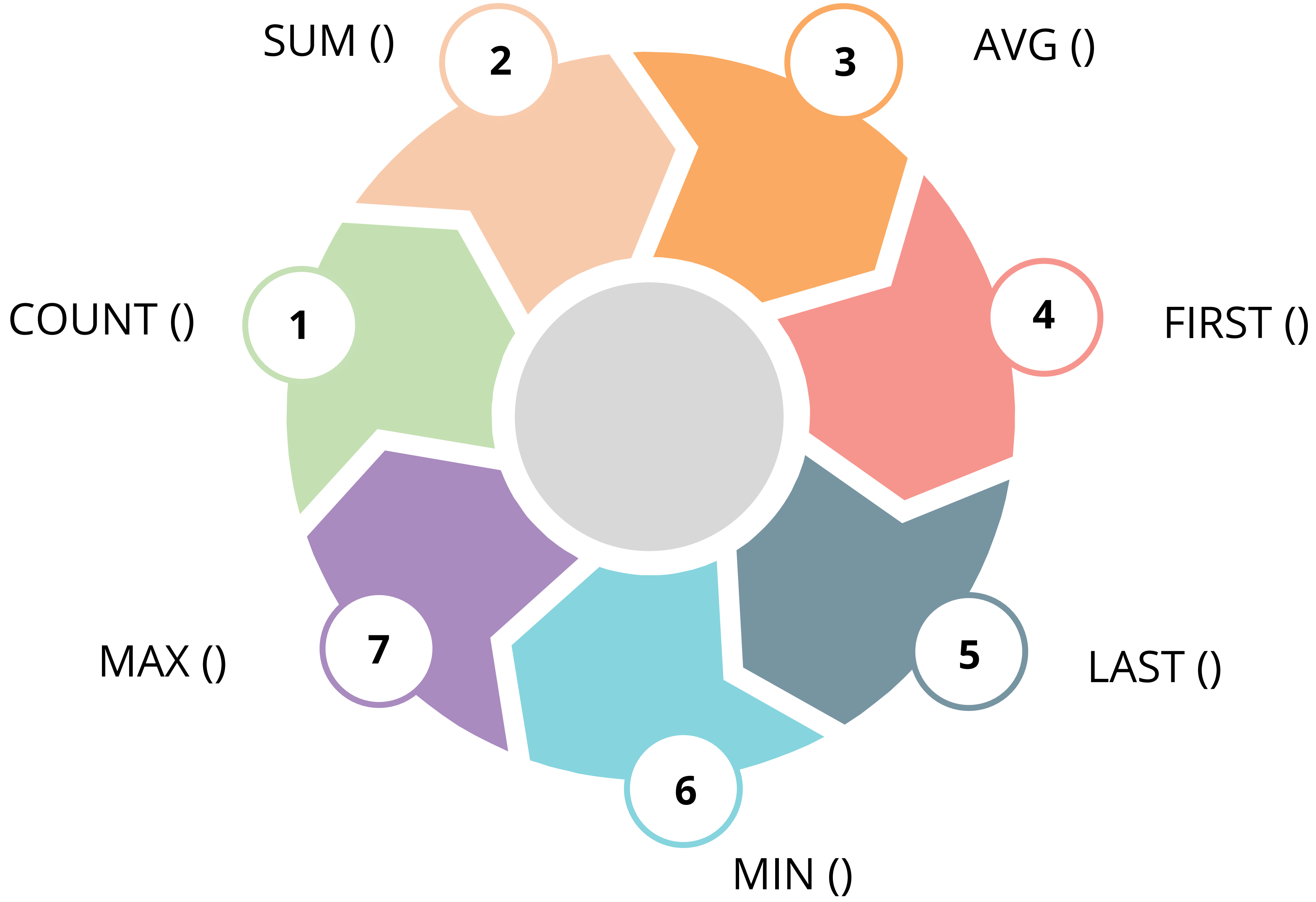
Are complicated mathematical logic that can be broken down into simpler functions.



Aggregate Functions

Aggregate Functions and Its Types

The aggregate functions allow performing the calculation on a set of values to return a single scalar value.



Count Function

Definition

Count function returns the total number of rows in a specified column or a table.

Syntax

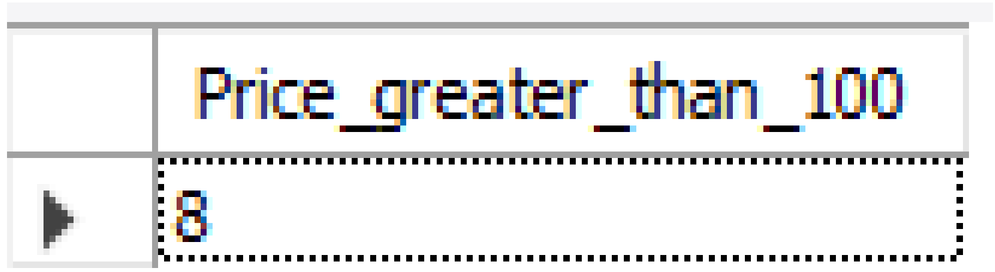
```
SELECT COUNT (column name)
FROM table name
WHERE condition;
```

Count Example

Example

```
SELECT COUNT(price) as Price_greater_than_100
FROM product WHERE price > 100;
```

Output



	Price_greater_than_100
▶	8

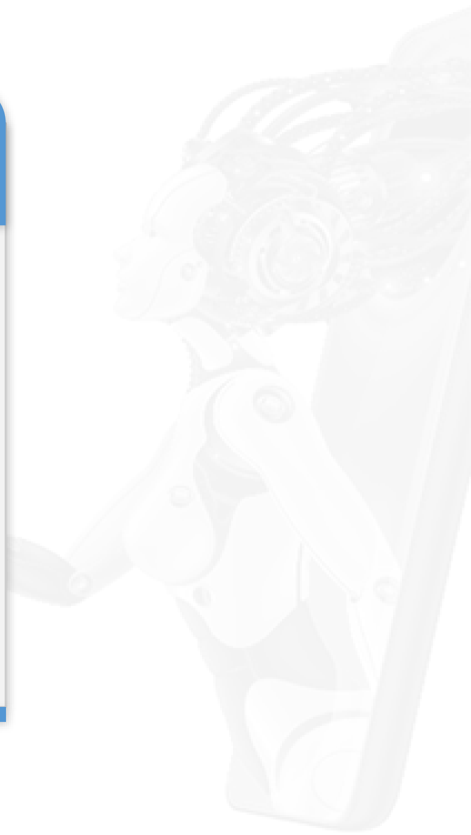
Sum Function

Definition

Sum function returns the sum of values from a particular column.

Syntax

```
SELECT SUM (column name)
FROM table name;
```

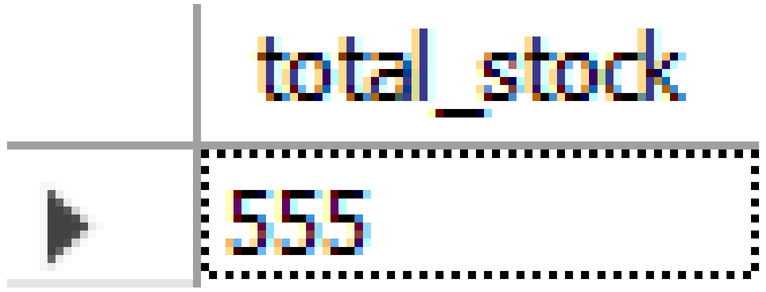


Sum Example

Example

```
SELECT SUM(stock) as total_stock FROM product;
```

Output



```
total_stock
555
```

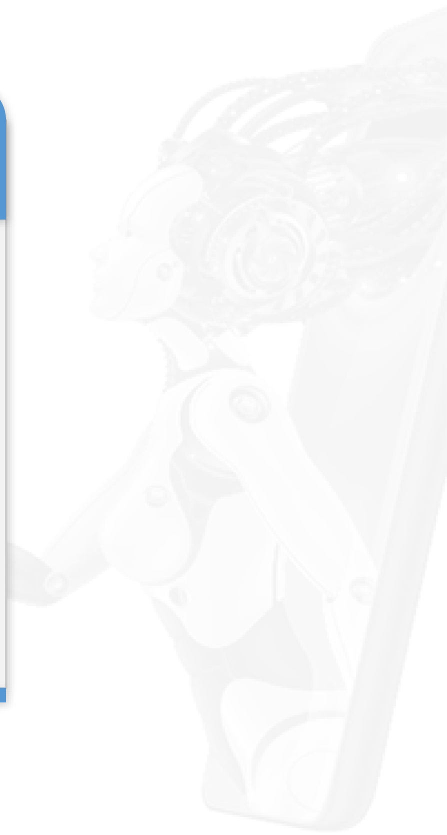
Average Function

Definition

Average function returns the average value of a particular column.

Syntax

```
SELECT AVG (Column name)
FROM table name;
```

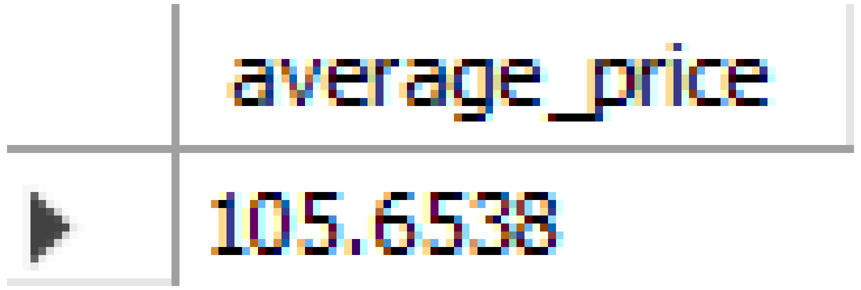


Average Example

Example

```
SELECT AVG(price) as average_price FROM product;
```

Output



```
average_price
105.6538
```

First Function

Definition

First function returns the first field value of the given column.

Syntax

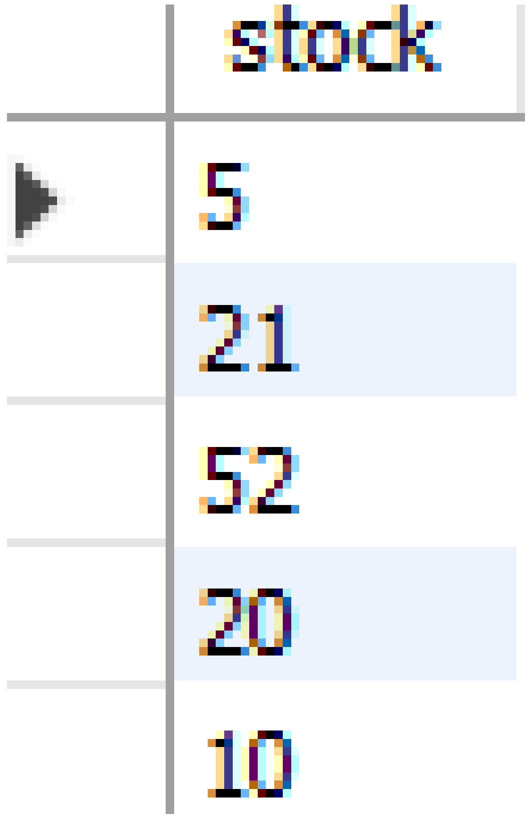
```
SELECT column_name FROM table_name
LIMIT value;
```

First Example

Example

```
SELECT stock FROM product LIMIT 5;
```

Output



stock
5
21
52
20
10

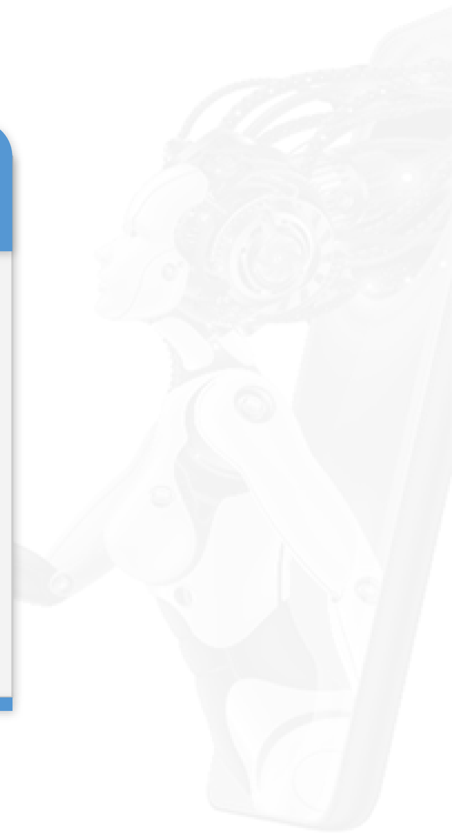
Last Function

Definition

Last function returns the last field value of the given column.

Syntax

```
SELECT column_name FROM table_name
ORDER BY column name DESC LIMIT
value;
```

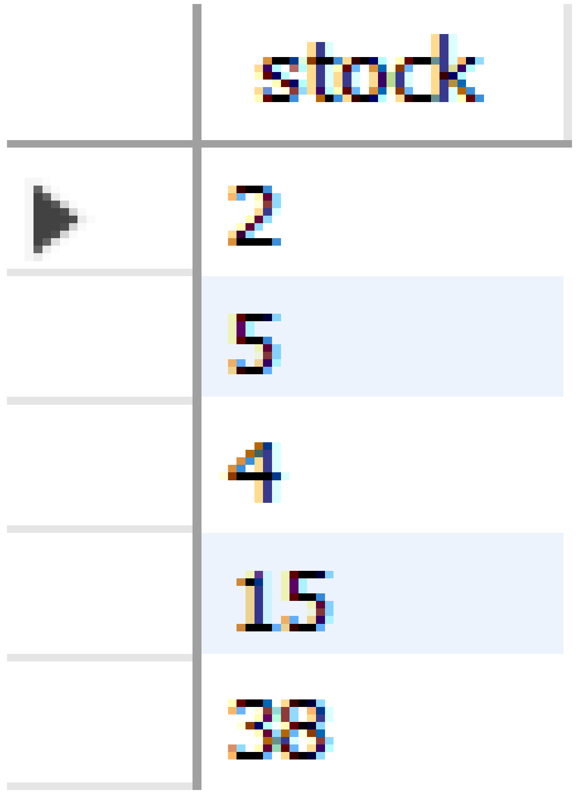


Last Example

Example

```
SELECT stock FROM product ORDER BY p_code
DESC LIMIT 5;
```

Output



	stock
▶	2
	5
	4
	15
	38

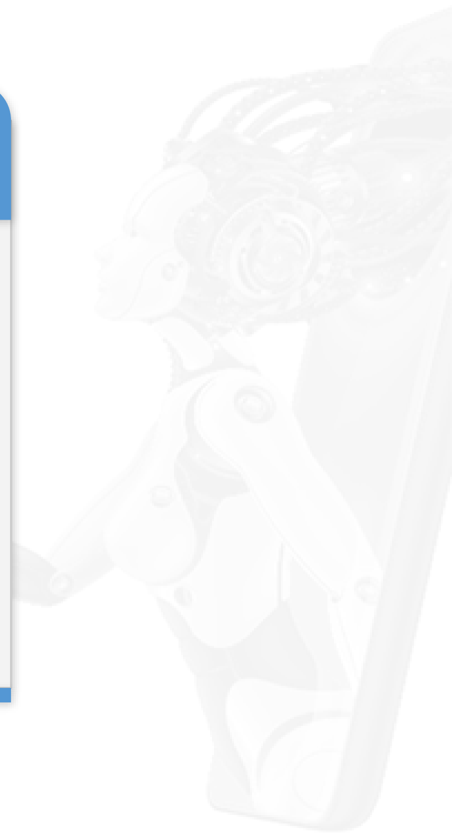
Min Function

Definition

Min function returns the minimum value of the given column.

Syntax

```
SELECT MIN(column_name) FROM
table_name;
```

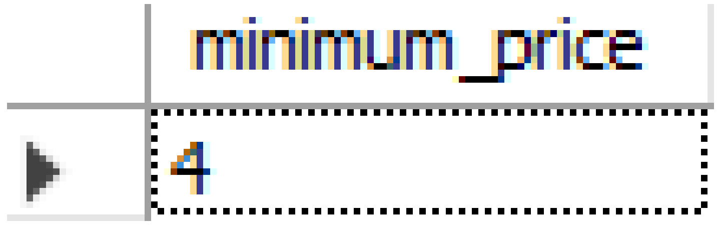


Min Example

Example

```
SELECT MIN(price) as minimum_price FROM
product;
```

Output



```
minimum_price
4
```

Max Function

Definition

Max function returns the maximum value of the given column.

Syntax

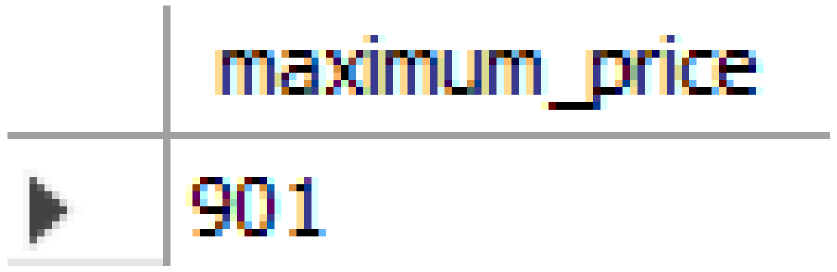
```
SELECT MAX(column_name) FROM
table_name;
```

Max Example

Example

```
SELECT MAX(price) as maximum_price FROM
product;
```

Output



	maximum_price
▶	901

Problem Statement

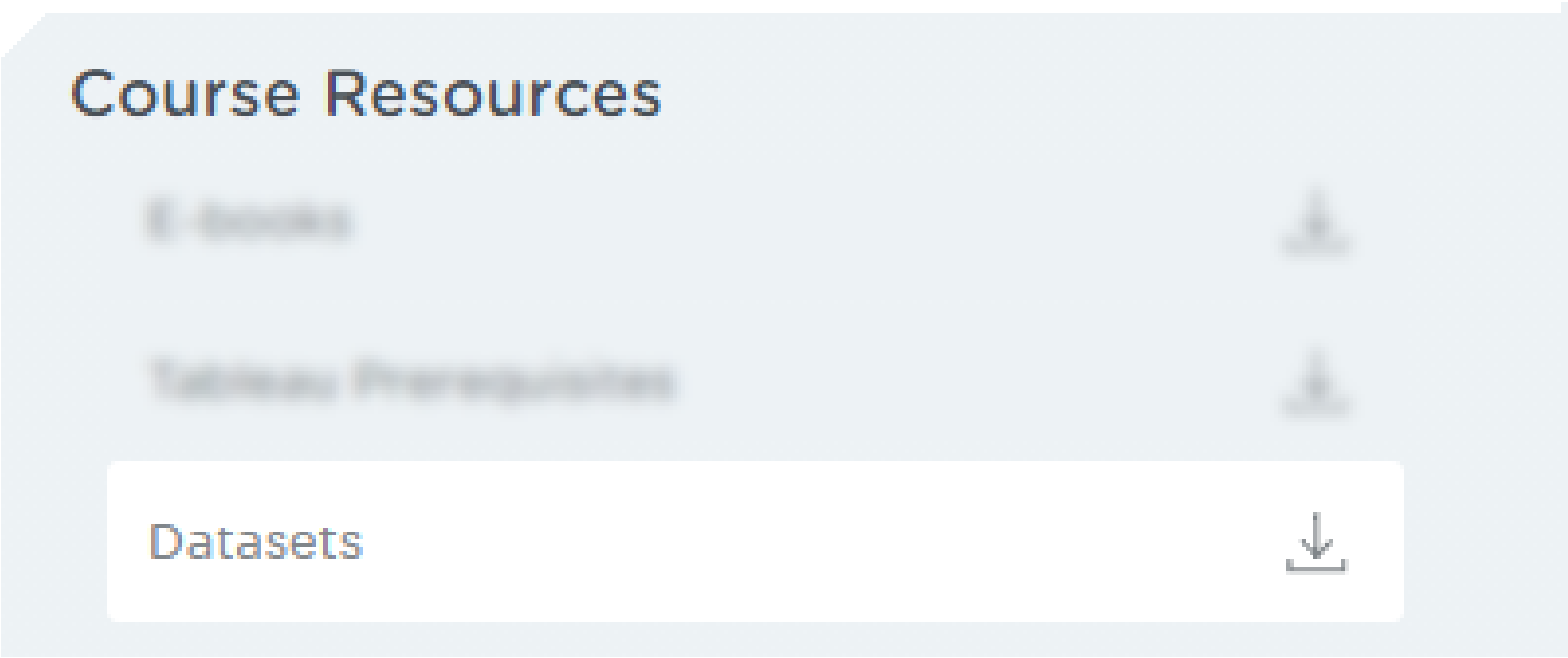
Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you to collect data from the superstore's table with the schema named as **example** to check and improve the sales records and growth of your store by performing a queried operation on the database.

Objective: You should determine the sum of the sales and profit columns, calculate the average profit, count the total number of products with a price greater than 100, and calculate the maximum profit and loss from the superstore table.

Problem Statement

Steps to be performed:

1. Download the **superstore** table from the course resources, and import it in MySQL workbench.



Course Resources

E-books

Tableau Prerequisites

Datasets

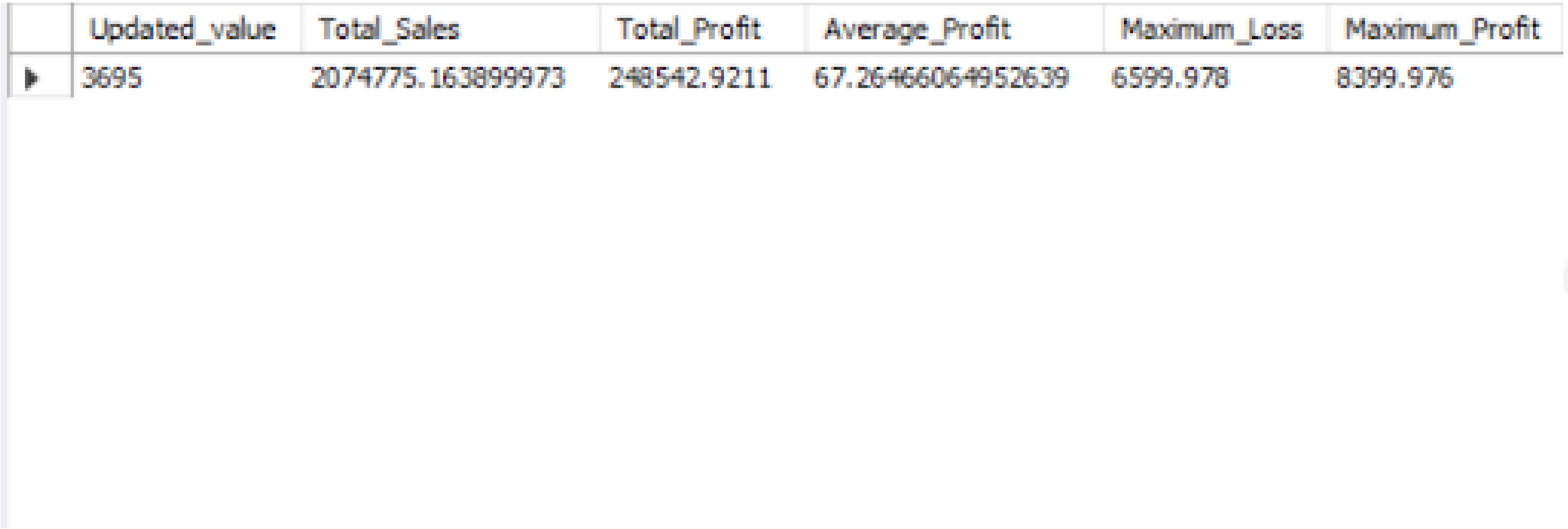
Solution

Query

```
SELECT COUNT(Sales) as Updated_value, sum(Sales) as Total_Sales, Sum(Profit) as
Total_Profit, avg(Profit) as Average_Profit, ABS(min(Profit)) as Maximum_Loss, max(Profit)
as Maximum_Profit

FROM example.superstore

WHERE Sales > 100;
```

Output

After executing the query, we get the updated value of the sales, profit, and average profit columns.

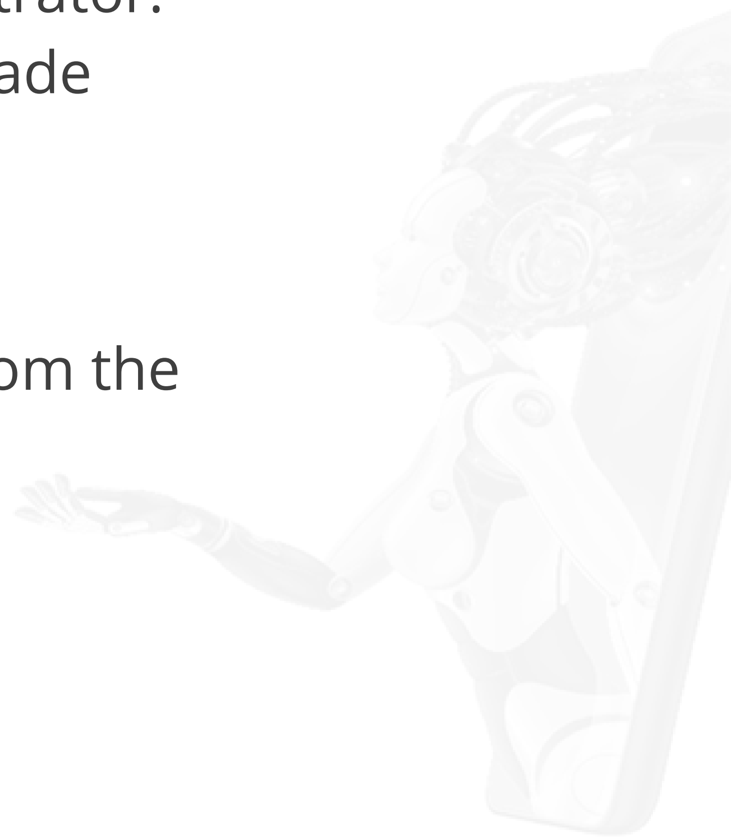


	Updated_value	Total_Sales	Total_Profit	Average_Profit	Maximum_Loss	Maximum_Profit
▶	3695	2074775.163899973	248542.9211	67.26466064952639	6599.978	8399.976

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you to retrieve the first ten records of sales that were made during the opening of the store.

Objective: You are required to extract the first ten records of the sales column from the superstore table.



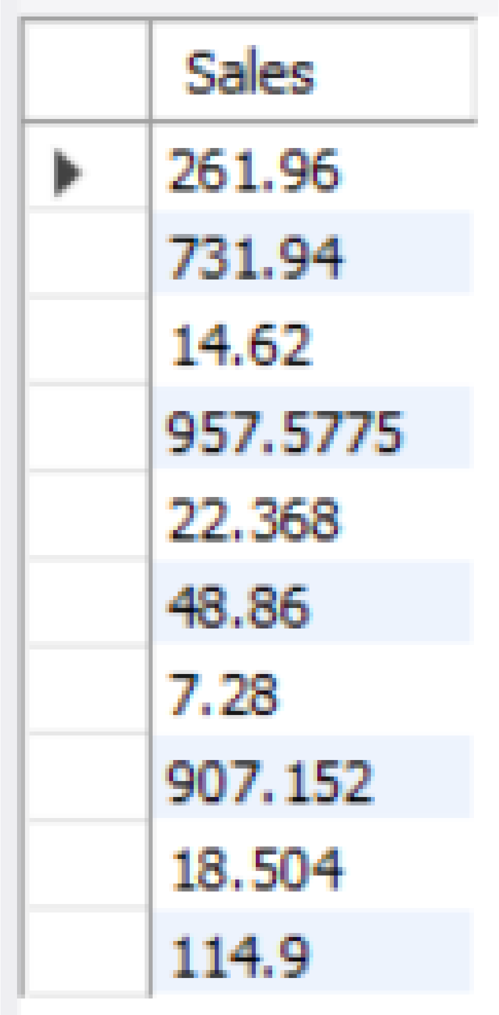
Solution

Query for FIRST ten records

```
SELECT Sales
FROM superstore limit 10;
```

Output

After executing the query, the first ten records of the database are shown as the following output:

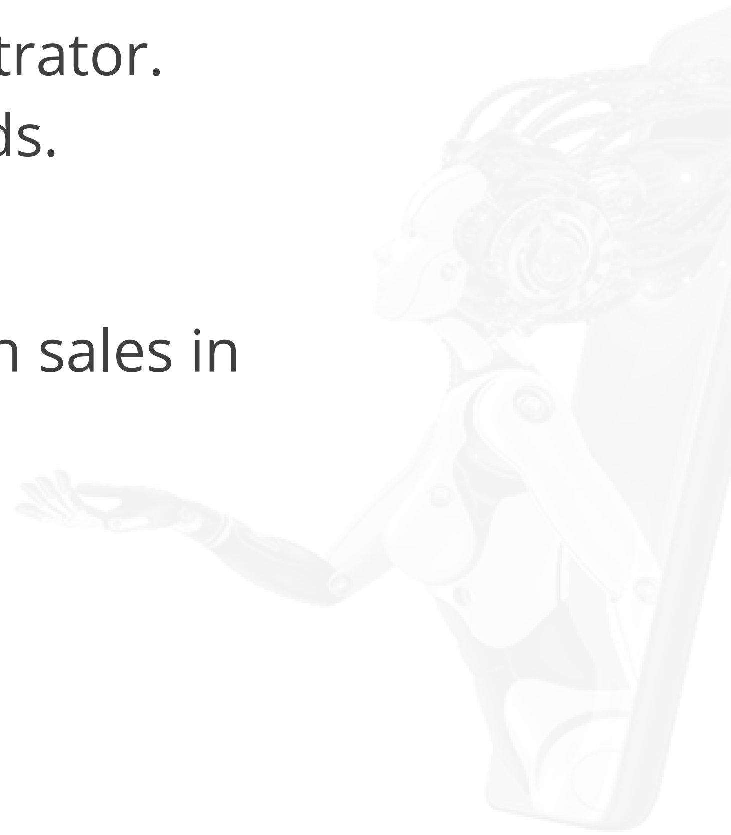


Sales
261.96
731.94
14.62
957.5775
22.368
48.86
7.28
907.152
18.504
114.9

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has assigned you the task of identifying the top twenty sales records.

Objective: You are required to analyze the superstore table by sorting the column sales in descending order and finding the first twenty records.



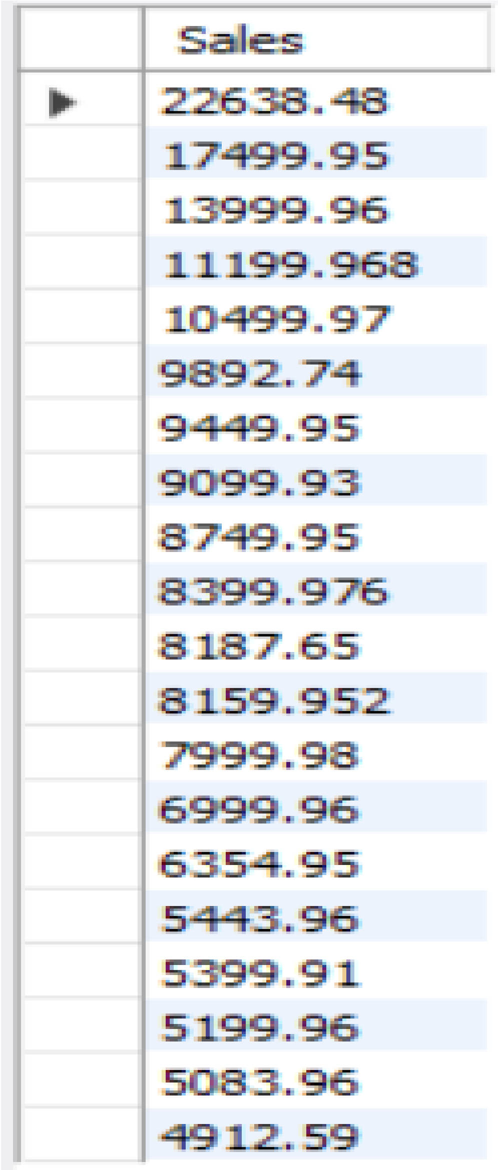
Solution

Query for LAST twenty records

```
SELECT Sales
FROM superstore
ORDER BY Sales DESC limit 20;
```

Output

After executing the query, the first twenty records sorted in descending order will be shown as the following output:

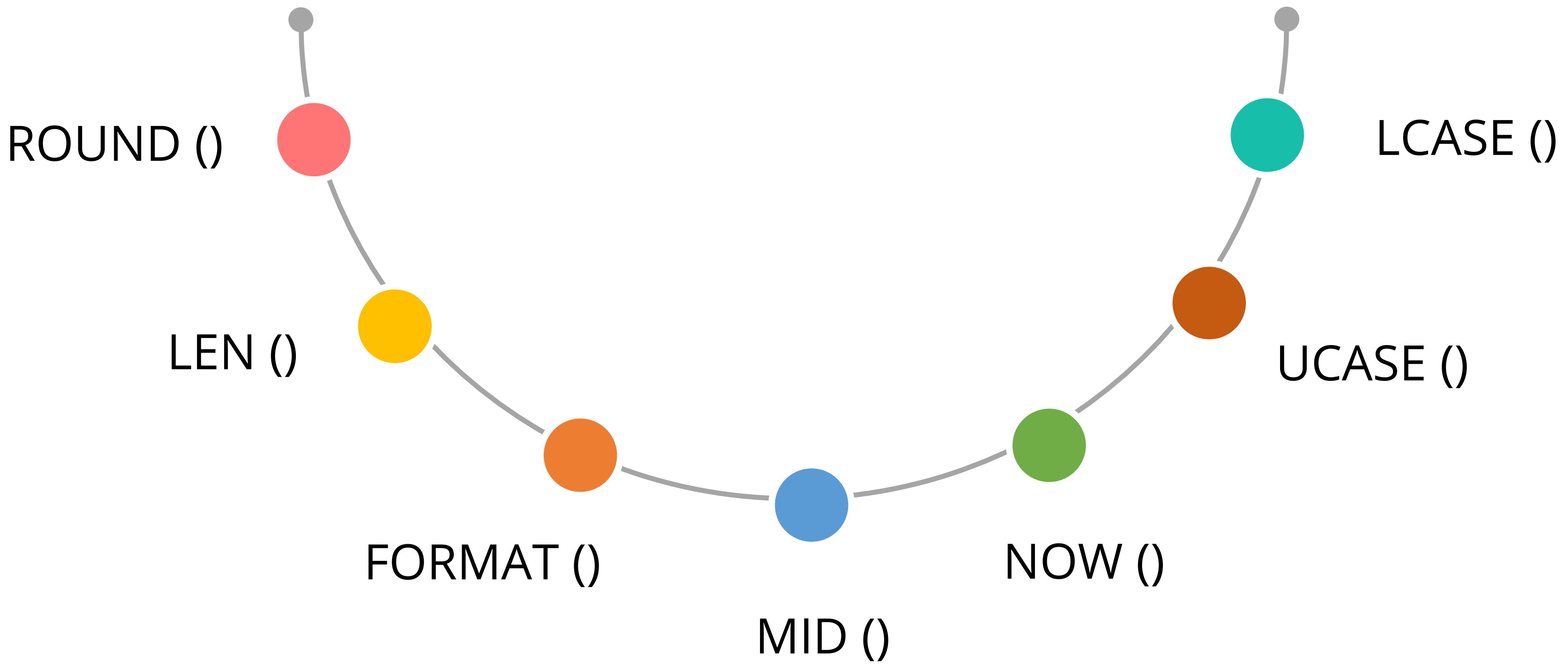


	Sales
▶	22638.48
	17499.95
	13999.96
	11199.968
	10499.97
	9892.74
	9449.95
	9099.93
	8749.95
	8399.976
	8187.65
	8159.952
	7999.98
	6999.96
	6354.95
	5443.96
	5399.91
	5199.96
	5083.96
	4912.59

Scalar Functions

Scalar Functions

The scalar functions return a single value from an input value. It works on each record independently.



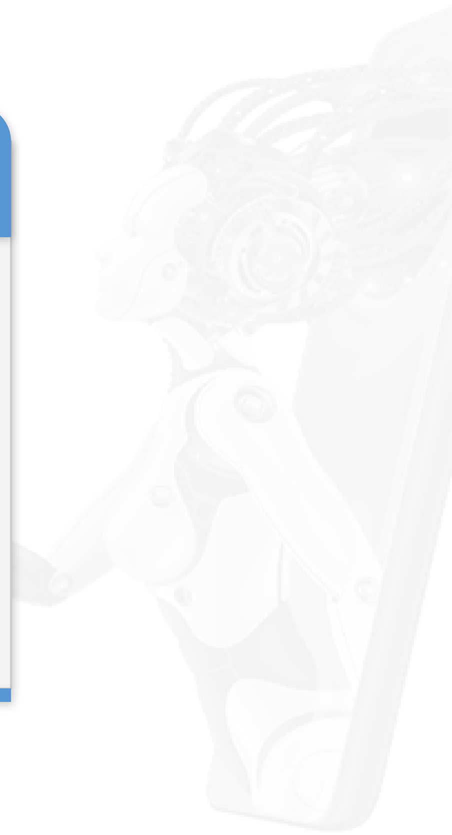
Round Function

Definition

Round function helps to round a value to a specified number of places.

Syntax

```
ROUND(column_name, decimals)
```

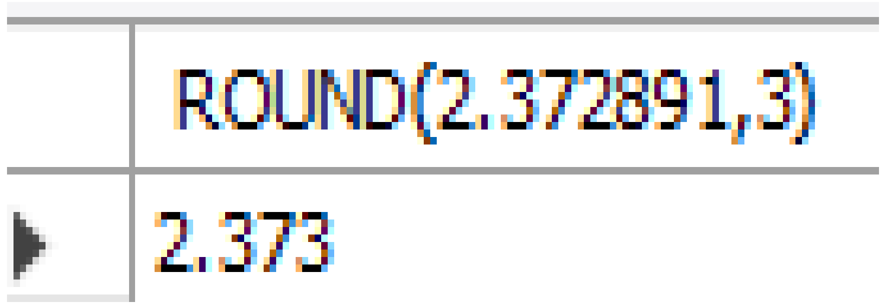


Round Example

Example

```
SELECT ROUND(2.372891,3)
```

Output



```
ROUND(2.372891,3)
2.373
```

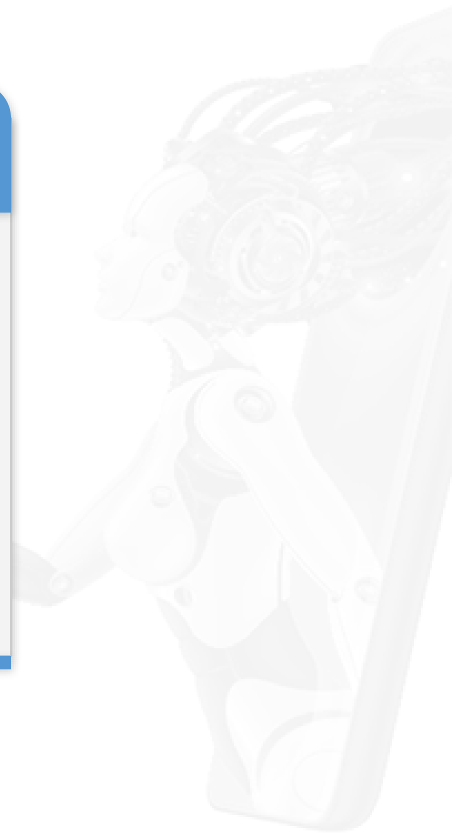
Length Function

Definition

Length function returns the total length of the given column.

Syntax

```
SELECT LENGTH(column_name) FROM
table_name;
```

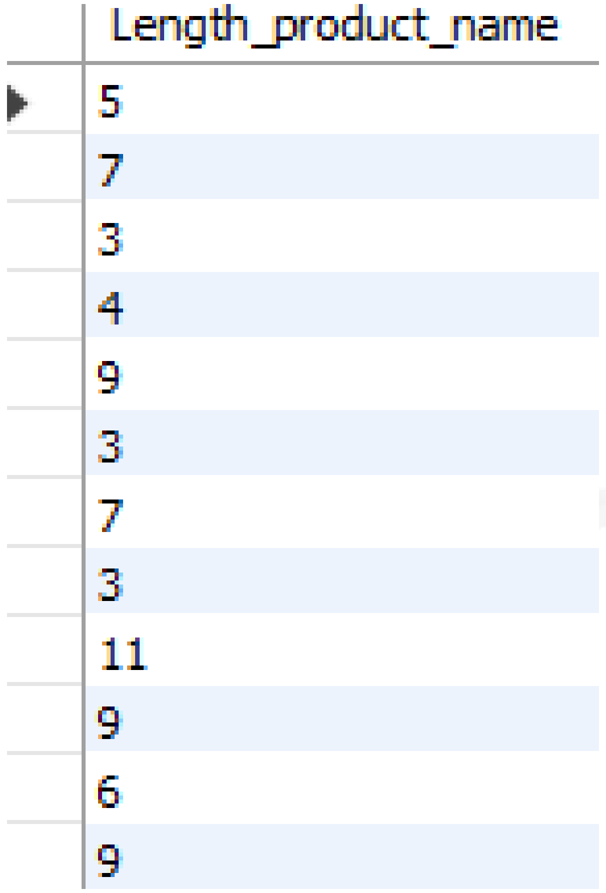


Length Example

Example

```
SELECT length(p_name) as Length_product_name
FROM product;
```

Output



```
Length_product_name
5
7
3
4
9
3
7
3
11
9
6
9
```

Format Function

Definition

Format function is used to format field value in the specified format.

Syntax

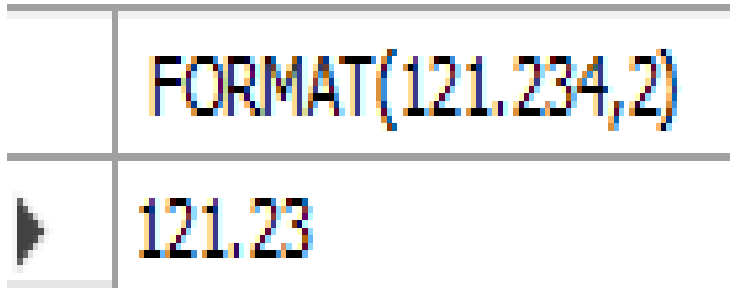
```
SELECT FORMAT(column_name, format) FROM table_name;
```

Format Example

Example

```
SELECT FORMAT(121.234,2)
```

Output



FORMAT(121.234,2)
121.23

MID Function

Definition

MID function is used to retrieve the specified characters from the text field.

Syntax

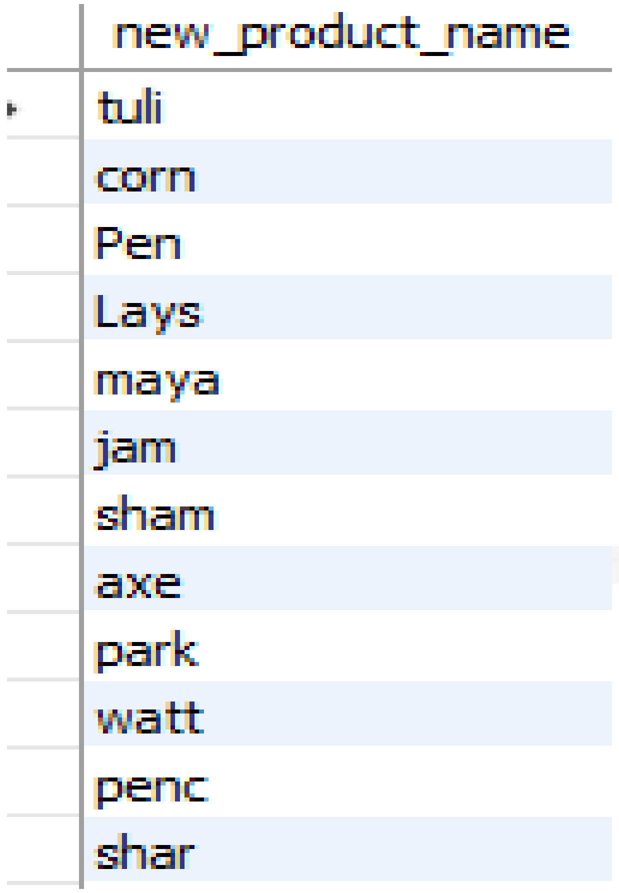
SELECT MID(column_name, start, length) FROM table_name;

MID Example

Example

```
SELECT MID(p_name,1,4) as new_product_name
FROM product;
```

Output



new_product_name
tuli
corn
Pen
Lays
maya
jam
sham
axe
park
watt
penc
shar

NOW Function

Definition

NOW function is used to retrieve the system’s current date and time.

Syntax

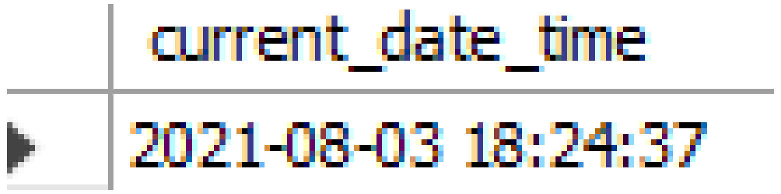
```
SELECT NOW()
```

NOW Example

Example

```
SELECT NOW() AS current_date_time
```

Output



```
current_date_time
2021-08-03 18:24:37
```

UCASE Function

Definition

UCASE function converts the given column to uppercase.

Syntax

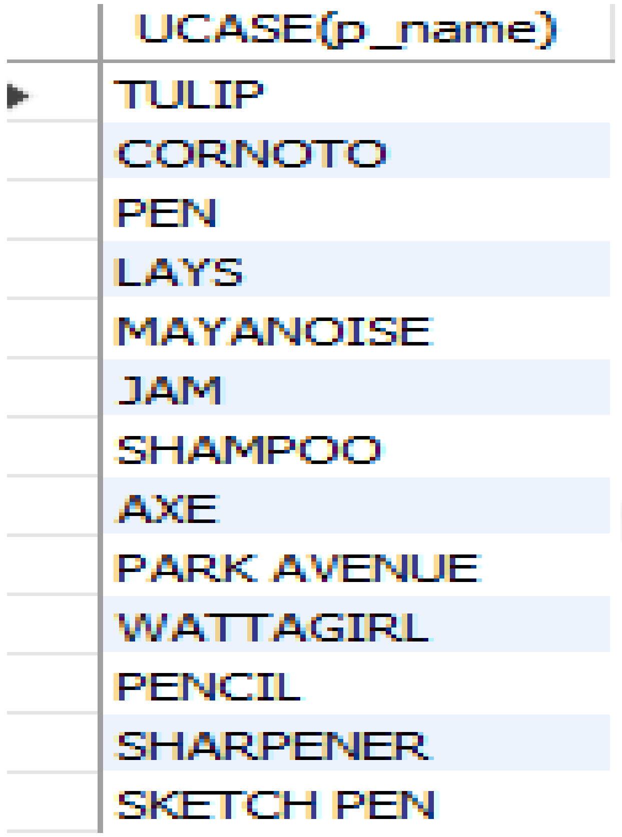
```
SELECT UCASE(column_name) FROM
table_name;
```

UCASE Example

Example

```
SELECT UCASE(p_name) FROM product
```

Output



UCASE(p_name)
TULIP
CORNOTO
PEN
LAYS
MAYANOISE
JAM
SHAMPOO
AXE
PARK AVENUE
WATTAGIRL
PENCIL
SHARPENER
SKETCH PEN

LCASE Function

Definition

LCASE function converts the given column to lowercase.

Syntax

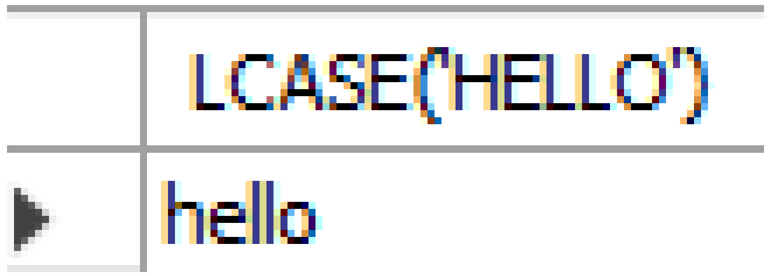
```
SELECT LCASE(column_name) FROM
table_name;
```

LCASE Example

Example

```
SELECT LCASE('HELLO')
```

Output



```
LCASE('HELLO')
hello
```

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you to find the order number from the order ID column for the better functionality of your store and to compare the order shipping and delivery dates.

Objective: You are required to extract the order number from the column **order ID** and list the shipping and delivery dates. Also, compare these dates with the present date.

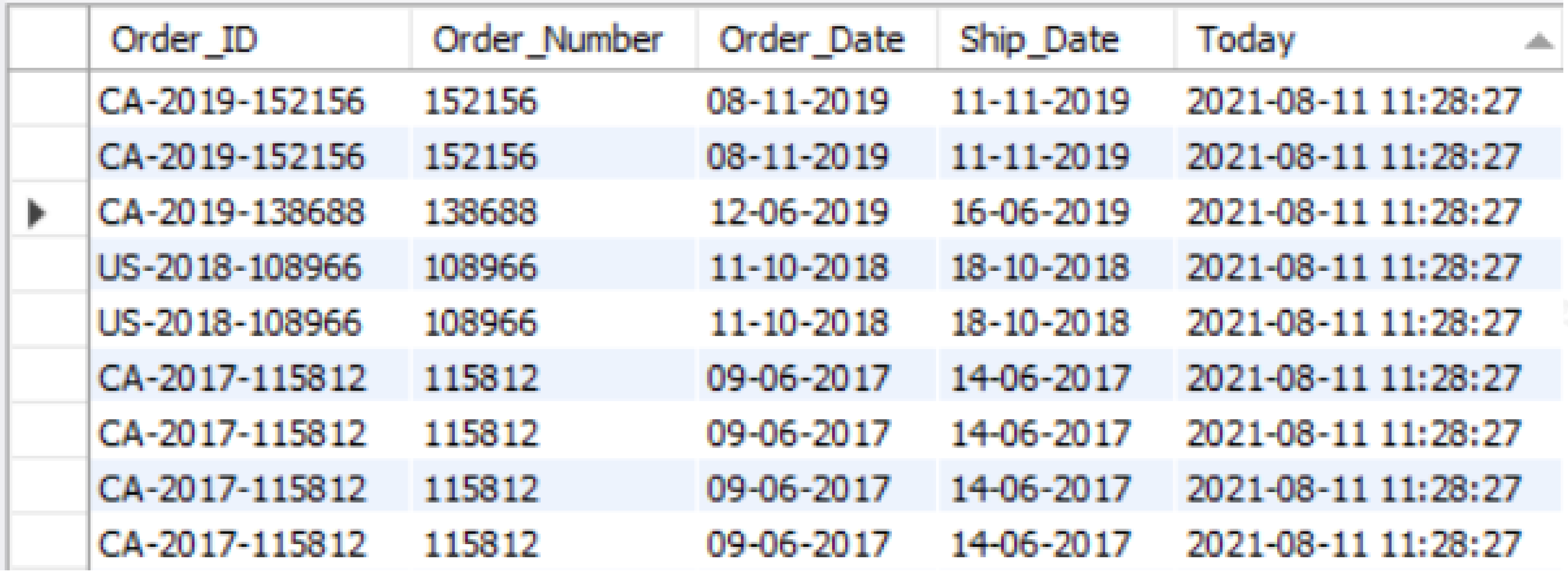
Solution

Query

```
SELECT Order_ID, mid(Order_ID,9,14) as Order_Number , Order_Date, Ship_Date, Now()
as Today

FROM example.superstore;
```

Output

After executing the query, the order number from order ID, order date, ship date, and current date is being displayed.

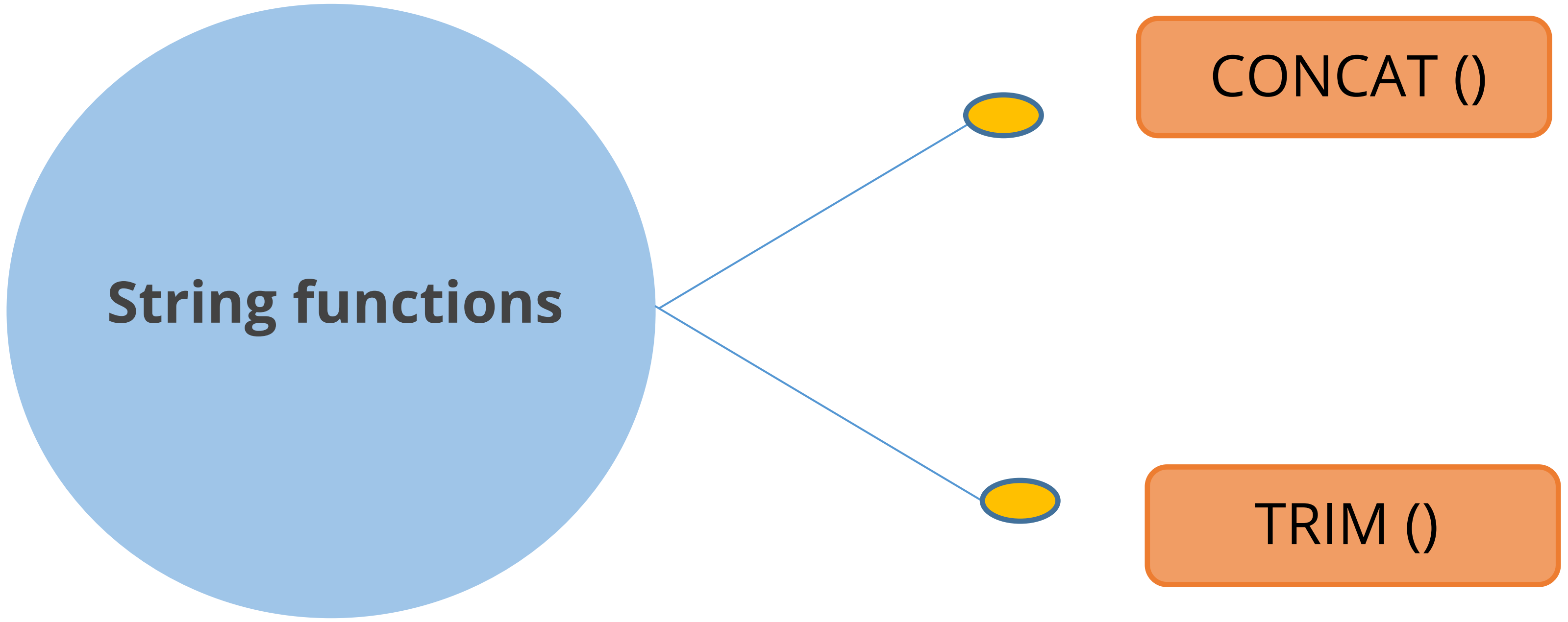


	Order_ID	Order_Number	Order_Date	Ship_Date	Today
	CA-2019-152156	152156	08-11-2019	11-11-2019	2021-08-11 11:28:27
	CA-2019-152156	152156	08-11-2019	11-11-2019	2021-08-11 11:28:27
▶	CA-2019-138688	138688	12-06-2019	16-06-2019	2021-08-11 11:28:27
	US-2018-108966	108966	11-10-2018	18-10-2018	2021-08-11 11:28:27
	US-2018-108966	108966	11-10-2018	18-10-2018	2021-08-11 11:28:27
	CA-2017-115812	115812	09-06-2017	14-06-2017	2021-08-11 11:28:27
	CA-2017-115812	115812	09-06-2017	14-06-2017	2021-08-11 11:28:27
	CA-2017-115812	115812	09-06-2017	14-06-2017	2021-08-11 11:28:27
	CA-2017-115812	115812	09-06-2017	14-06-2017	2021-08-11 11:28:27

String Functions

String Functions

The string functions are used for string manipulation.



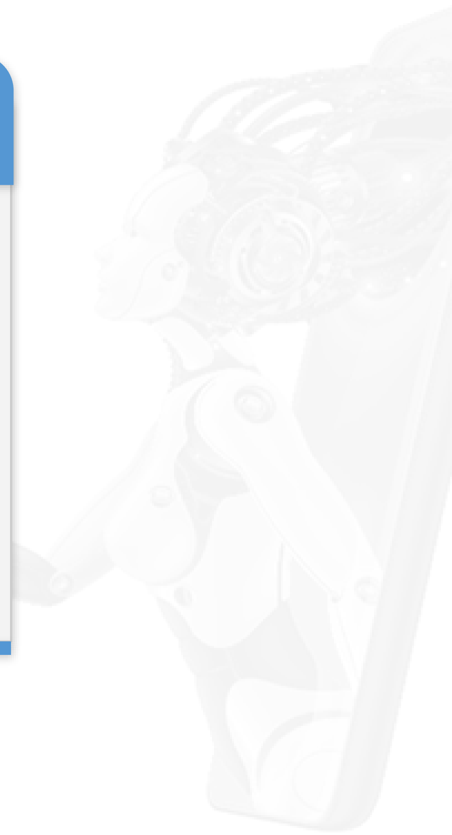
Concat Function

Definition

Concat function is used to combine one or more characters into a single string.

Syntax

```
SELECT CONCAT (String 1, String 2, String3.., Str
ing N)  FROM  table name;
```

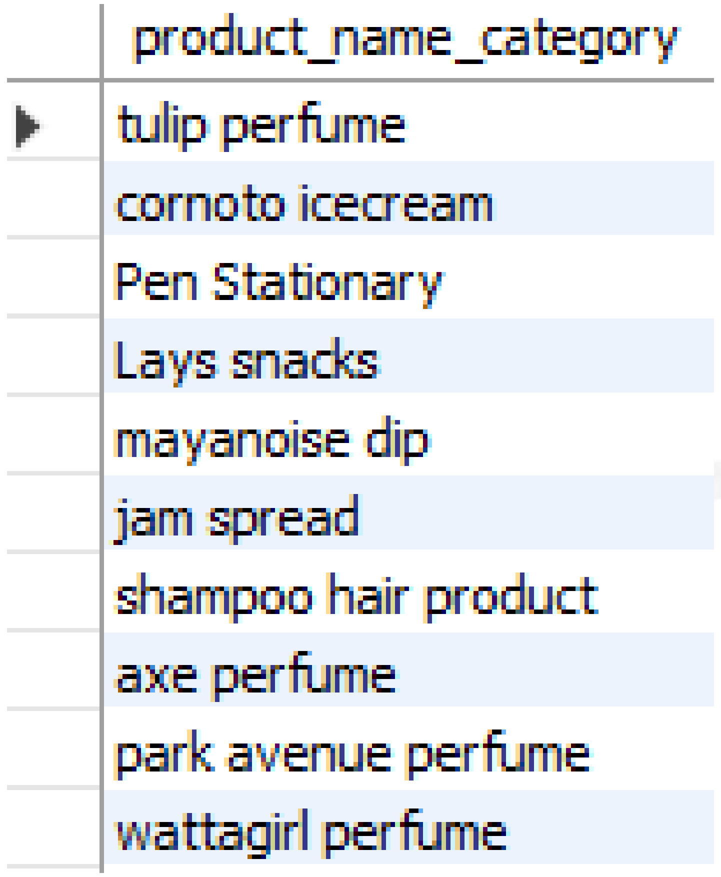


Concat Example

Example

```
SELECT CONCAT(p_name,' ',category) AS
product_name_category FROM product
```

Output



product_name_category
tulip perfume
cornoto icecream
Pen Stationary
Lays snacks
mayanoise dip
jam spread
shampoo hair product
axe perfume
park avenue perfume
wattagirl perfume

Trim Function

Definition

Trim function is used to remove the spaces from both sides of the given string.

Syntax

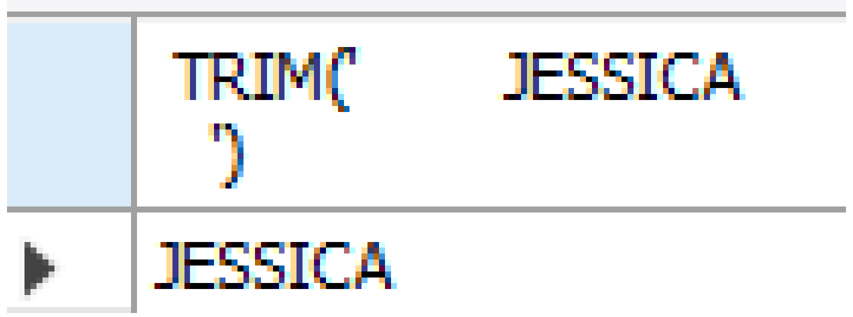
```
SELECT TRIM (String 1)  FROM  table name;
```

Trim Example

Example

```
SELECT TRIM('       JESSICA     ')
```

Output



```
TRIM('       JESSICA
 ')
JESSICA
```

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you to retrieve the list of all the customer addresses to send them a personalized invite as a marketing strategy for an upcoming sale in the store.

Objective: You are required to display the customer's name, city, state, and postal code from the superstore table in a single column **address**. Also, count the length of the customer’s **name** and convert it into lowercase and **state** into uppercase, respectively.

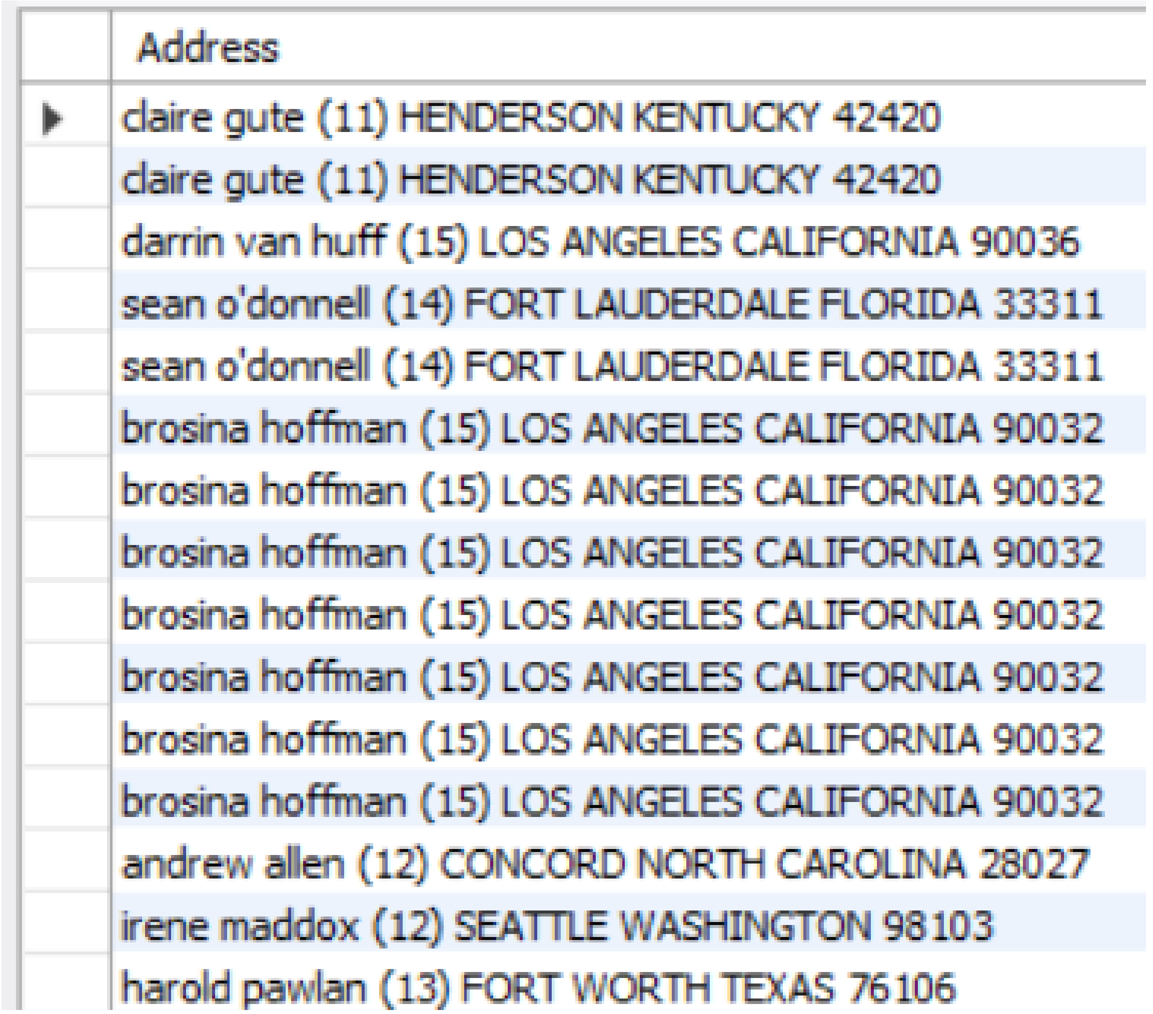
Solution

Query

```
SELECT Concat(lcase(Customer_Name),' ','(' , length(Customer_Name), ')', ' ', ucase(City),' ',
ucase(State),' ', Postal_Code) as Address
FROM example.superstore;
```

Output

After executing the query, the customer's name, city, state, and postal code are collectively shown as an address.

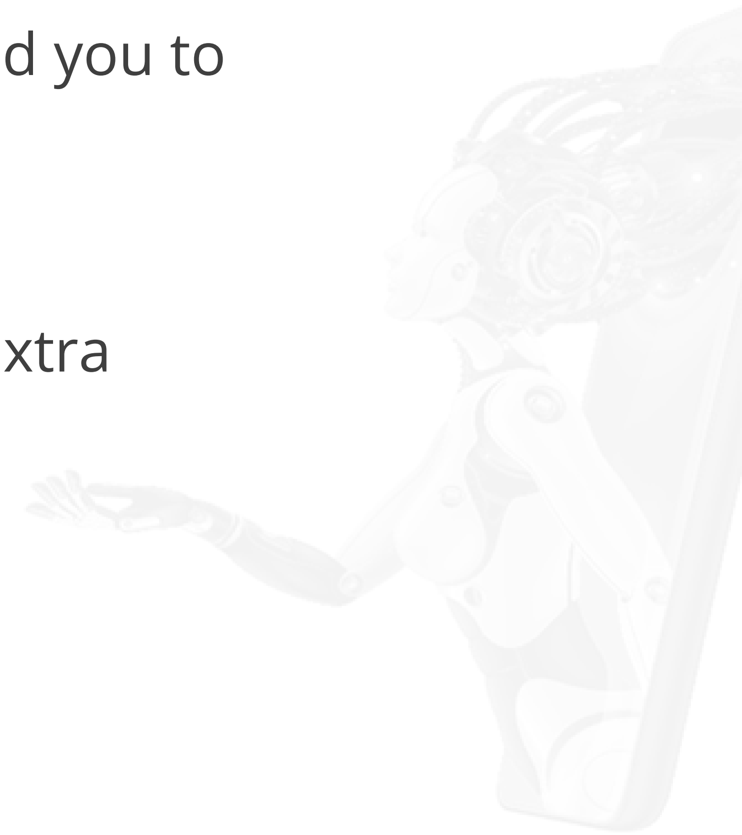


	Address
▶	claire gute (11) HENDERSON KENTUCKY 42420
	claire gute (11) HENDERSON KENTUCKY 42420
	darrin van huff (15) LOS ANGELES CALIFORNIA 90036
	sean o'donnell (14) FORT LAUDERDALE FLORIDA 33311
	sean o'donnell (14) FORT LAUDERDALE FLORIDA 33311
	brosina hoffman (15) LOS ANGELES CALIFORNIA 90032
	brosina hoffman (15) LOS ANGELES CALIFORNIA 90032
	brosina hoffman (15) LOS ANGELES CALIFORNIA 90032
	brosina hoffman (15) LOS ANGELES CALIFORNIA 90032
	brosina hoffman (15) LOS ANGELES CALIFORNIA 90032
	brosina hoffman (15) LOS ANGELES CALIFORNIA 90032
	brosina hoffman (15) LOS ANGELES CALIFORNIA 90032
	andrew allen (12) CONCORD NORTH CAROLINA 28027
	irene maddox (12) SEATTLE WASHINGTON 98103
	harold pawlan (13) FORT WORTH TEXAS 76106

Problem Statement

Problem Scenario: As the junior database administrator, your manager has asked you to format the customer ID column and remove the extra spaces.

Objective: You are required to format the customer ID column and remove the extra spaces.



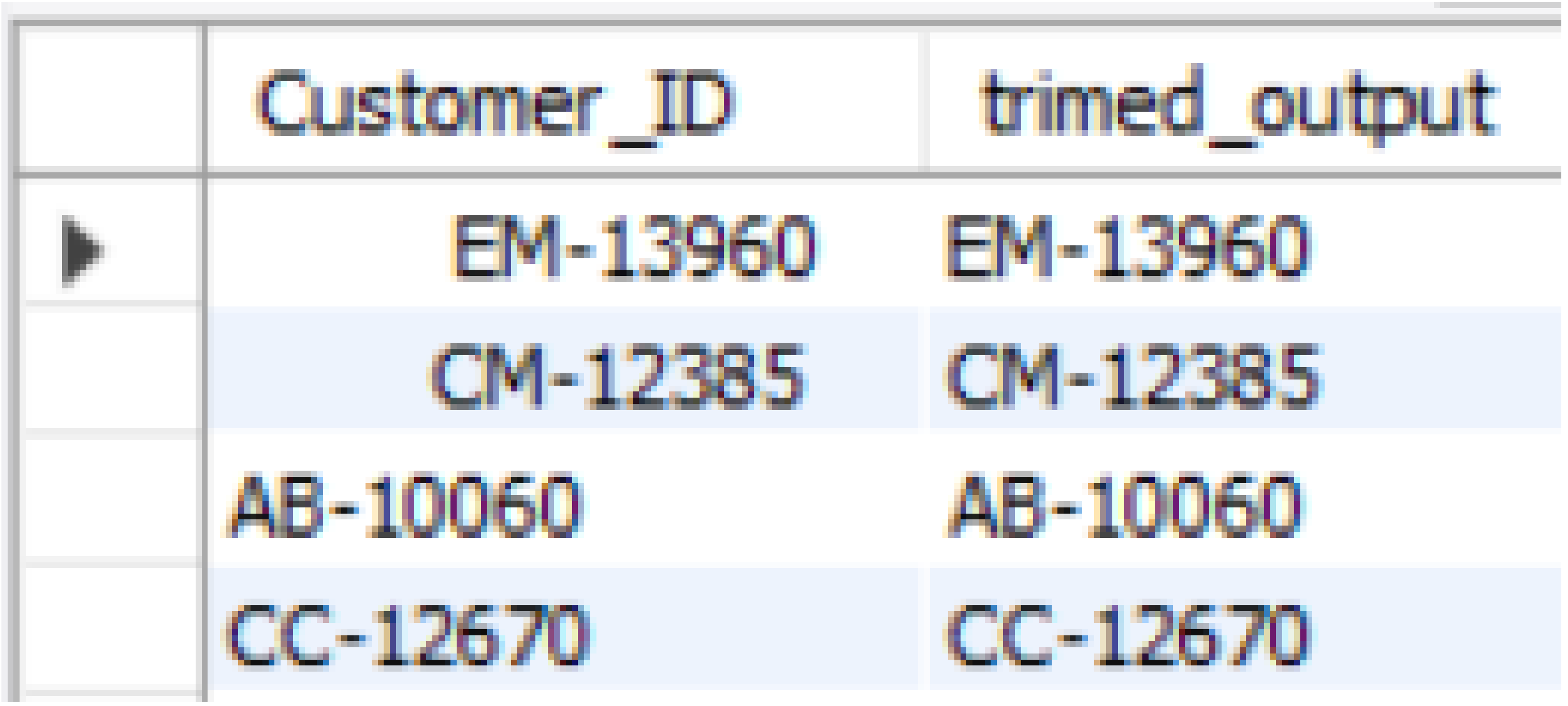
Solution

Query

```
SELECT Customer_ID, TRIM(Customer_ID) as trimed_output
FROM example.return_products;
```

Output

After executing the query, we can eliminate the excess white spaces from the customer ID column.



	Customer_ID	trimed_output
▶	EM-13960	EM-13960
	CM-12385	CM-12385
	AB-10060	AB-10060
	CC-12670	CC-12670

Assisted Practice: String Function

Duration: 15 min

Problem Statement: As the HR of your organization, you are expected to wish Merry Christmas to everyone. List down the **full names** of all the employees in uppercase using string functions.

Assisted Practice: String Function

Steps to be performed:

1. Create a database named **example**, then make a table named **candidates**, that has a column named **FirstName** and **LastName**.

TABLE CREATION

```
CREATE TABLE `example`.`candidates` (
`FirstName` VARCHAR(255) NOT NULL,
`LastName` VARCHAR(255) NOT NULL);
```

Assisted Practice: String Function

2. Insert values in the **candidates** table.

VALUE INSERTION

```
INSERT INTO `example`.`candidates` (`FirstName`, `LastName`)
VALUES ('James', 'Smith'),
('Maria ', 'Gracia'),
('Michael ', 'Rodriguez'),
('Robert ', 'Johnson'),
('David', 'Hernandez');
```

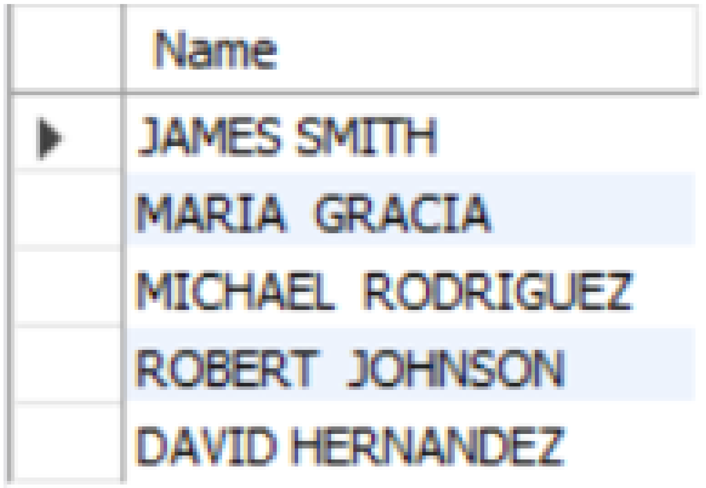
Assisted Practice: String Function

3. Write a query to combine **FirstName** and **LastName** into a single string in a new column named **Name**.

QUERY

```
SELECT CONCAT(UCASE(FirstName)," ",UCASE(LastName)) AS Name
FROM example.candidates;
```

Assisted Practice: Lab Output

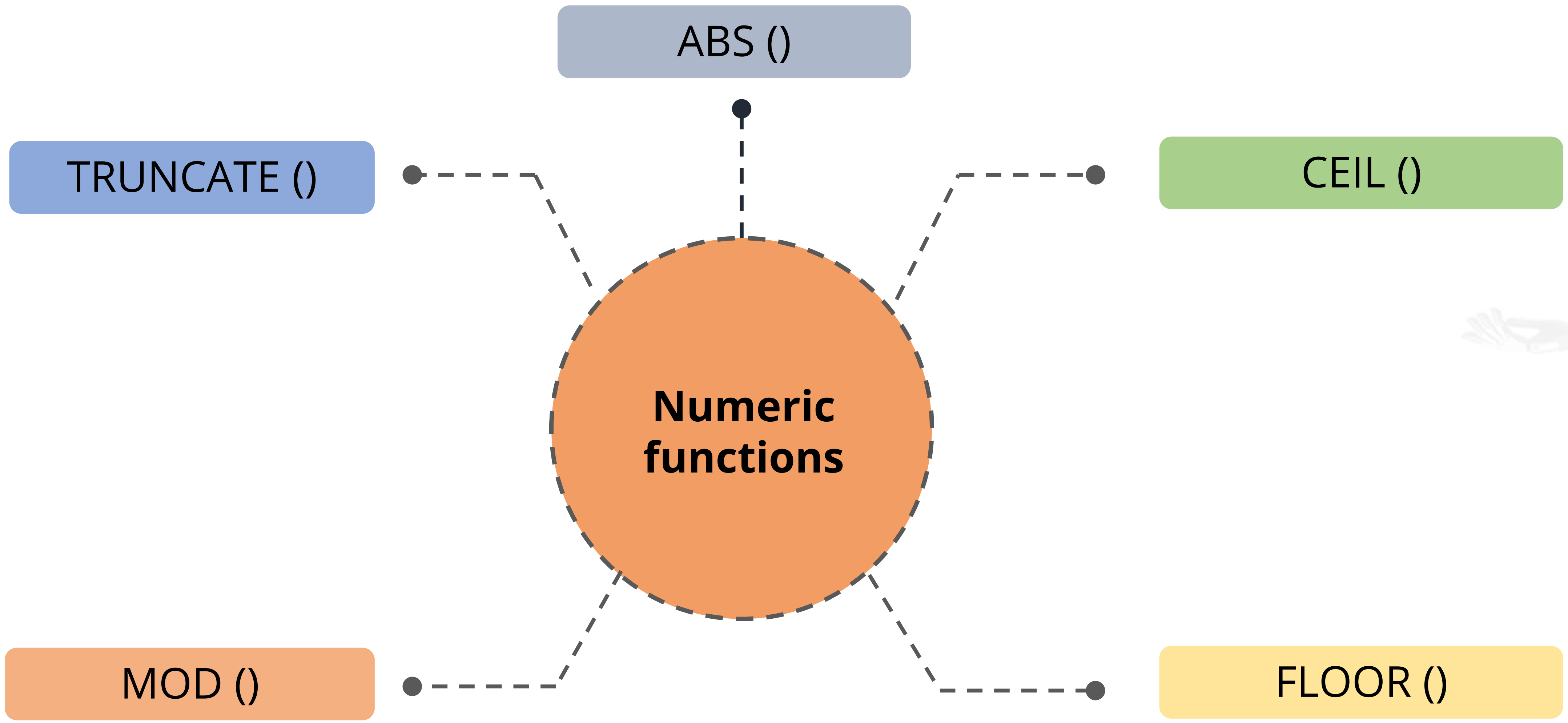


	Name
▶	JAMES SMITH
	MARIA GRACIA
	MICHAEL RODRIGUEZ
	ROBERT JOHNSON
	DAVID HERNANDEZ

Numeric Functions

Numeric Functions

The numeric functions are used to perform numeric manipulation or mathematical operations.



ABS ()

TRUNCATE ()

CEIL ()

Numeric functions

MOD ()

FLOOR ()

ABS Function

Definition

ABS function is used to return the absolute value of a given number.

Syntax

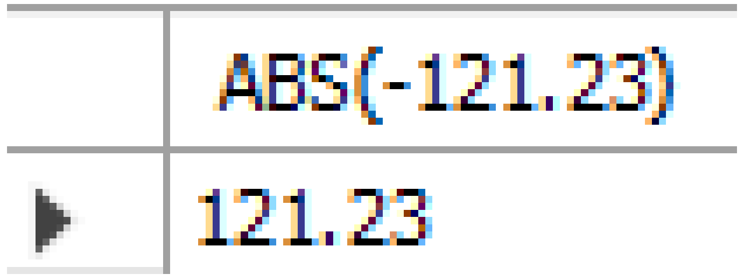
```
SELECT ABS (VALUE);
```

ABS Example

Example

```
SELECT ABS(-121.23)
```

Output



	ABS(-121.23)
▶	121.23

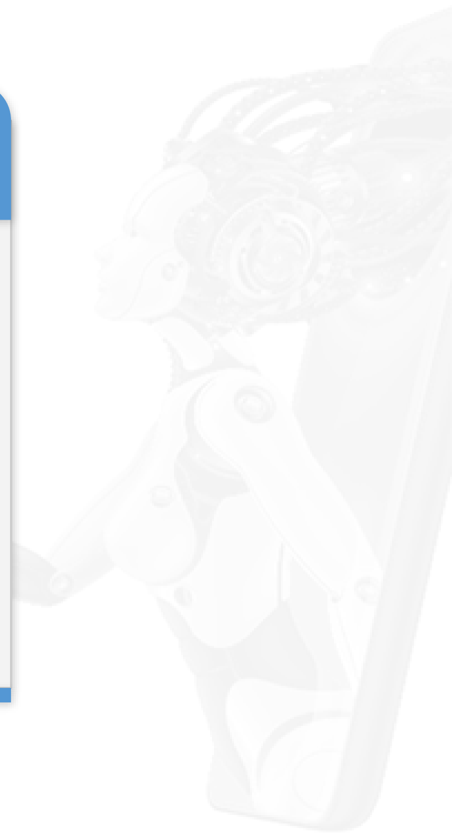
Ceil Function

Definition

Ceil function returns the smallest integer value that is greater than or equal to the given number.

Syntax

```
SELECT CEIL(VALUE);
```

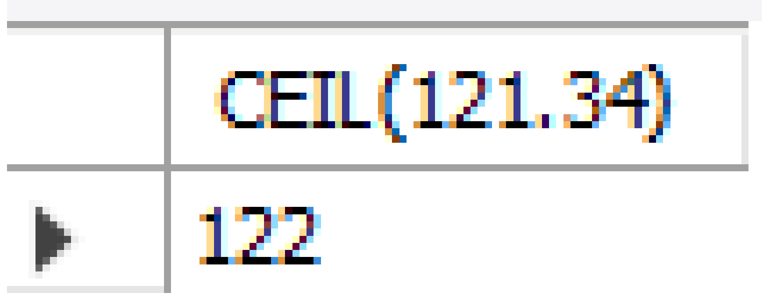


Ceil Example

Example

```
SELECT CEIL(121.34)
```

Output



```
CEIL(121.34)
122
```

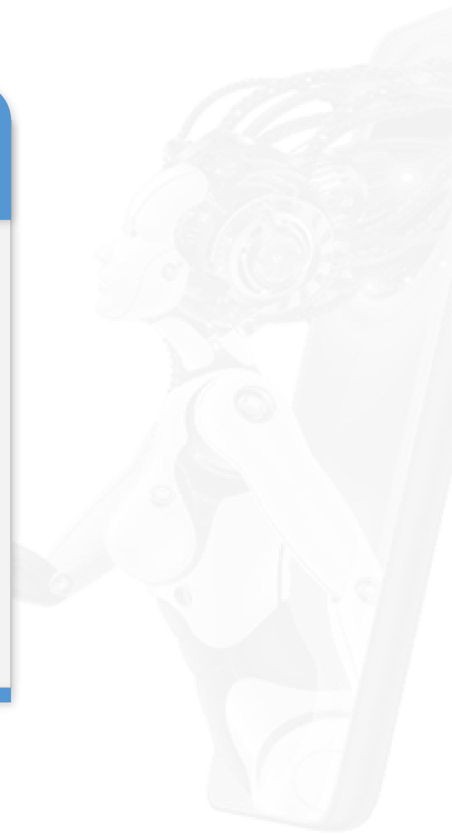
Floor Function

Definition

Floor function returns the largest integer value that is less than or equal to the given number.

Syntax

```
SELECT FLOOR(VALUE);
```

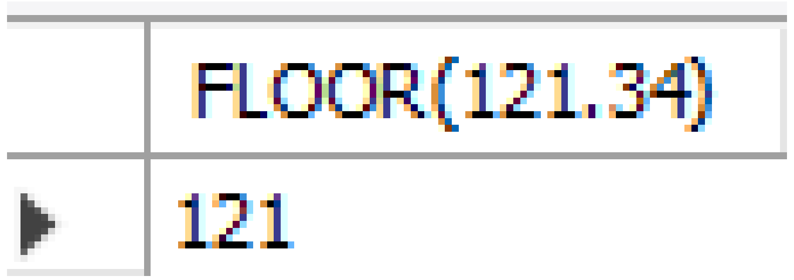


Floor Example

Example

```
SELECT FLOOR(121.34)
```

Output



```
FLOOR(121.34)
121
```

Truncate Function

Definition

Truncate function is used to truncate a number to the specified number of decimal places.

Syntax

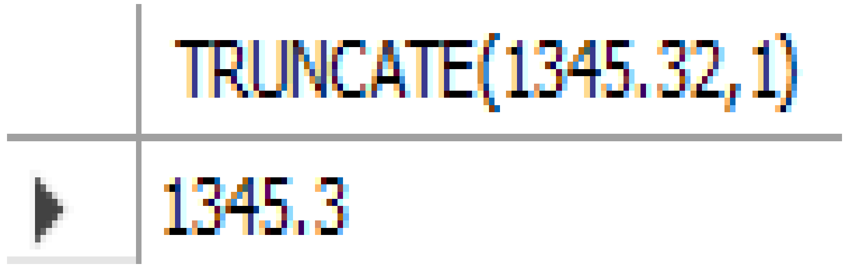
```
SELECT TRUNCATE (VALUE,DECIMALS);
```

Truncate Example

Example

```
SELECT TRUNCATE(1345.32,1)
```

Output



```
TRUNCATE(1345.32,1)
1345.3
```

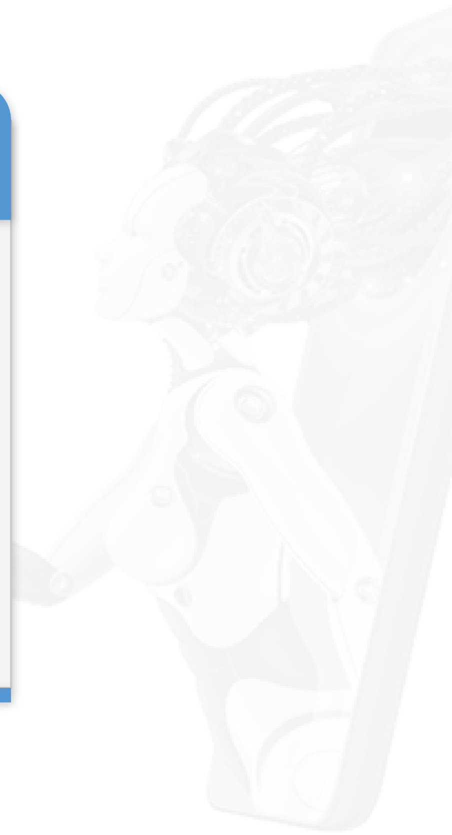
MOD Function

Definition

MOD function returns the remainder of a number by dividing it with another number.

Syntax

```
SELECT MOD (VALUE1 , VALUE2);
```

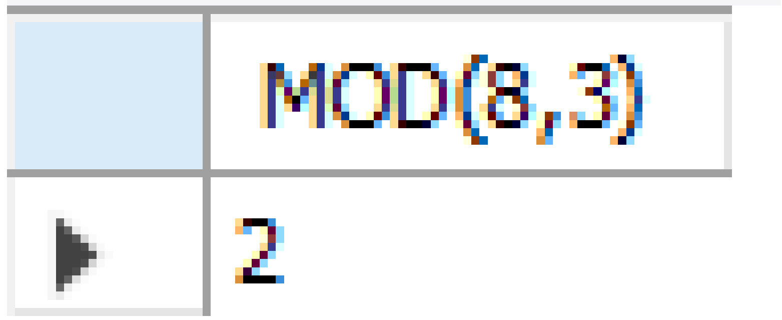


MOD Example

Example

```
SELECT MOD(8,3)
```

Output

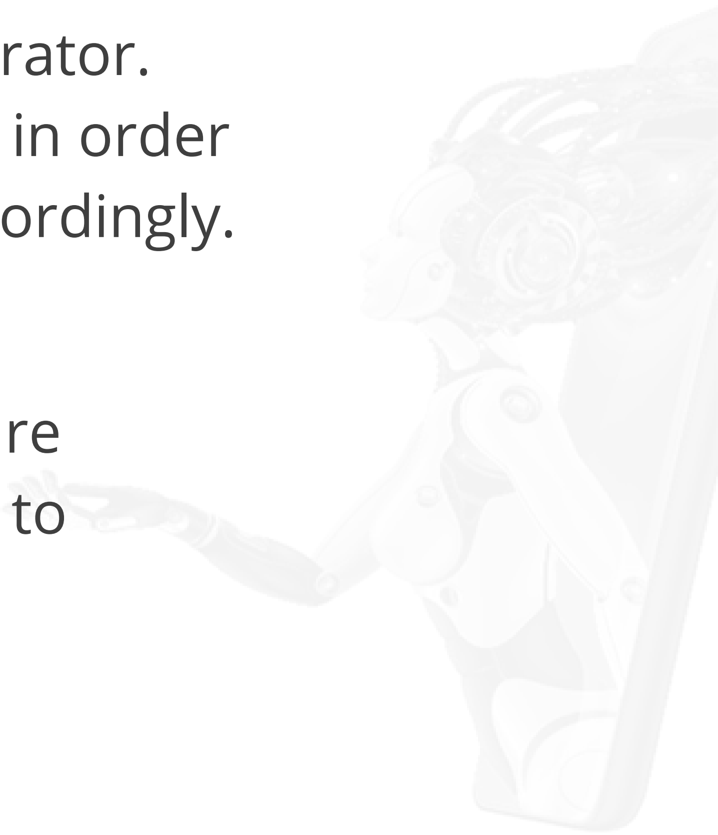


	MOD(8,3)
▶	2

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you to perform different operations on the sales column in order to obtain the highest profit so that the management can plan the next quarter accordingly.

Objective: The data that you received from the profit column is in decimals. You are required to perform mathematical and scaler operations using different functions to manipulate and compare the profit generated.



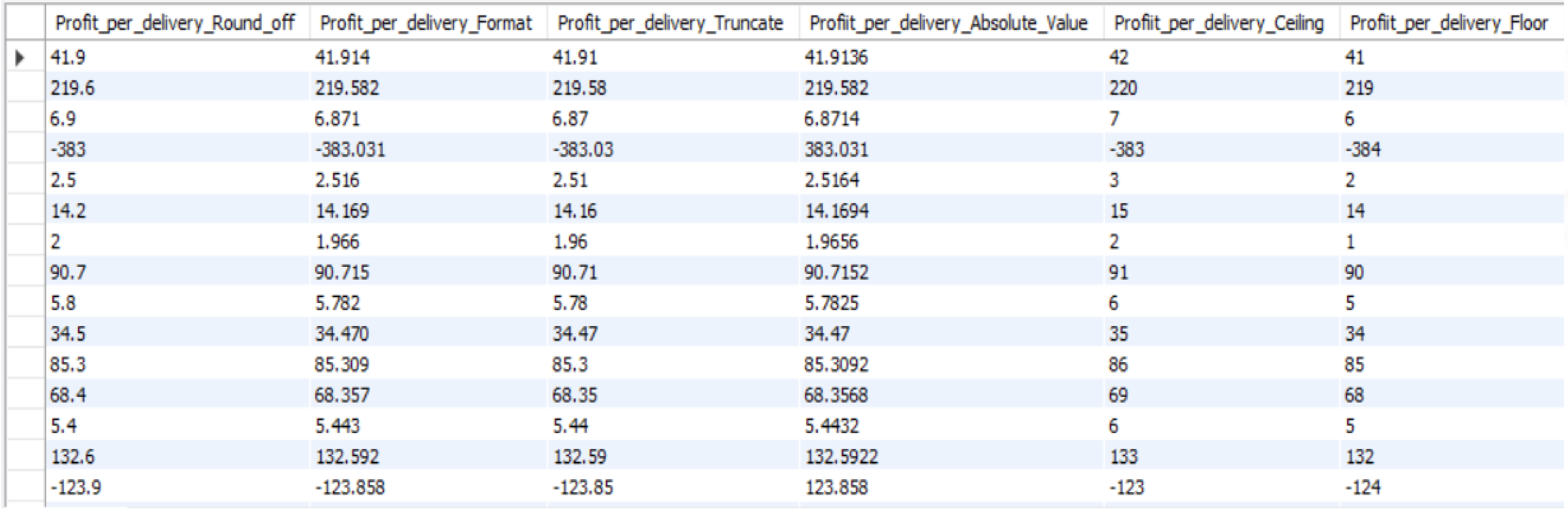
Solution

Query

```
SELECT Round(Profit, 1) as Profit_per_delivery_Round_off, Format(Profit, 3) as
Profit_per_delivery_Format, Truncate(Profit,2) as Profit_per_delivery_Truncate, ABS(Profit)
as Profiit_per_delivery_Absolute_Value, Ceil(Profit) as Profiit_per_delivery_Ceiling,
Floor(Profit) as Profiit_per_delivery_Floor

FROM example.superstore;
```

Output

The following output is generated after executing the query:

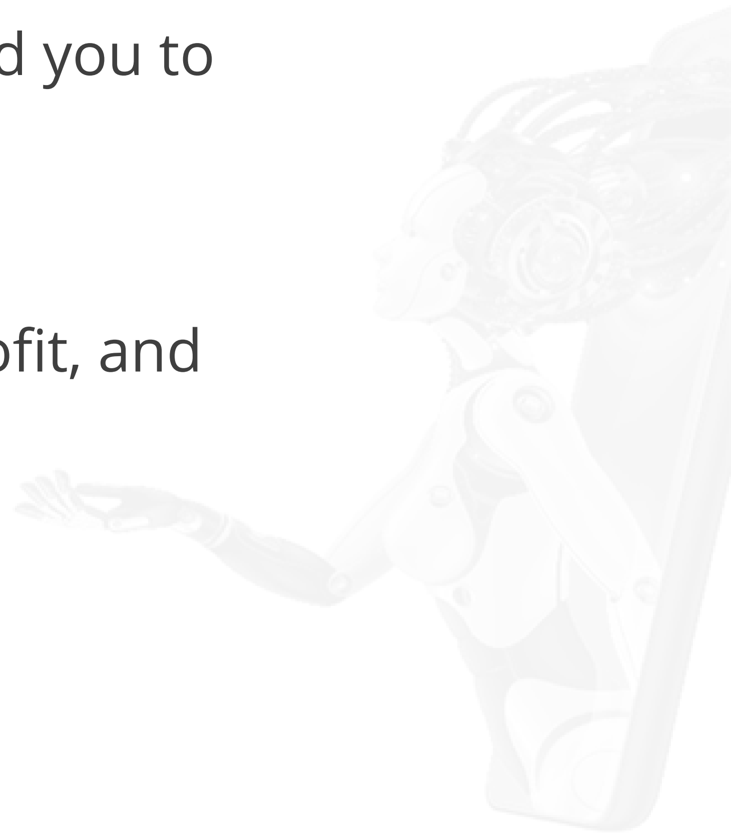


Profit_per_delivery_Round_off	Profit_per_delivery_Format	Profit_per_delivery_Truncate	Profit_per_delivery_Absolute_Value	Profiit_per_delivery_Ceiling	Profiit_per_delivery_Floor
41.9	41.914	41.91	41.9136	42	41
219.6	219.582	219.58	219.582	220	219
6.9	6.871	6.87	6.8714	7	6
-383	-383.031	-383.03	383.031	-383	-384
2.5	2.516	2.51	2.5164	3	2
14.2	14.169	14.16	14.1694	15	14
2	1.966	1.96	1.9656	2	1
90.7	90.715	90.71	90.7152	91	90
5.8	5.782	5.78	5.7825	6	5
34.5	34.470	34.47	34.47	35	34
85.3	85.309	85.3	85.3092	86	85
68.4	68.357	68.35	68.3568	69	68
5.4	5.443	5.44	5.4432	6	5
132.6	132.592	132.59	132.5922	133	132
-123.9	-123.858	-123.85	123.858	-123	-124

Problem Statement

Problem Scenario: As the junior database administrator, your manager has asked you to calculate the profit percentage in order to review your sales.

Objective: You are required to calculate the profit percentage using the sales, profit, and discount columns.



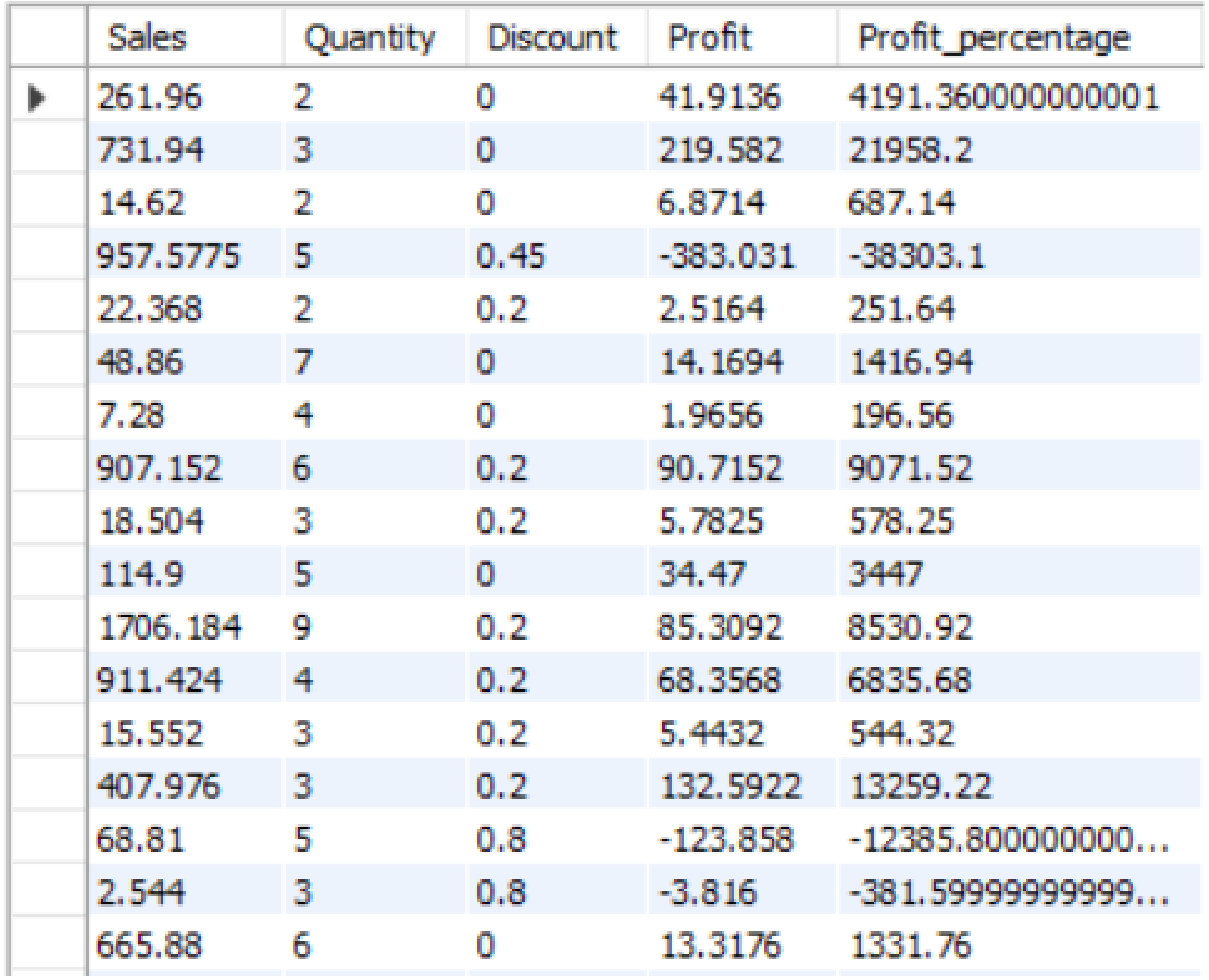
Solution

Query

```
SELECT Sales, Quantity, Discount, Profit,(MOD (profit,(sales-profit+Discount))*100) as
Profit_percentage

FROM example.superstore;
```

Output

After executing the query, we can calculate the profit percentage for the superstore table.



Sales	Quantity	Discount	Profit	Profit_percentage
261.96	2	0	41.9136	4191.360000000001
731.94	3	0	219.582	21958.2
14.62	2	0	6.8714	687.14
957.5775	5	0.45	-383.031	-38303.1
22.368	2	0.2	2.5164	251.64
48.86	7	0	14.1694	1416.94
7.28	4	0	1.9656	196.56
907.152	6	0.2	90.7152	9071.52
18.504	3	0.2	5.7825	578.25
114.9	5	0	34.47	3447
1706.184	9	0.2	85.3092	8530.92
911.424	4	0.2	68.3568	6835.68
15.552	3	0.2	5.4432	544.32
407.976	3	0.2	132.5922	13259.22
68.81	5	0.8	-123.858	-12385.800000000...
2.544	3	0.8	-3.816	-381.59999999999...
665.88	6	0	13.3176	1331.76

Assisted Practice: Numeric Function

Duration: 20 min

Problem Statement: You need to understand the approximate and actual profit from your shop's daily transaction ledger and decide to *round off* the **Amount** up to 0 and 2 decimal places. Also, apply *ceiling* and *floor* on the **Amount** respectively to understand the differences.

Assisted Practice: Numeric Function

Steps to be performed:

1. Create a database named **example** and then make a table named **bill**, that has a column named **S.no.**, **Name** and **Amount.** Also, assign **S.no.** as the **primary key**.

TABLE CREATION

```
CREATE TABLE `example'.' bill` (
  `S.no.` INT NOT NULL,
  `Name` VARCHAR(255) NOT NULL,
  `Amount` DECIMAL NOT NULL,
  PRIMARY KEY (`S.no.`));
```

Assisted Practice: Numeric Function

2. Insert values in the **bill** table.

VALUE INSERTION

```
INSERT INTO `example`.`bill` (`S.no.`, `Name`, `Amount`)
VALUES ('1', 'Oliver', '2753.3491’),
('2', 'George', '2532.4082’),
('3', 'Arthur', '2021.5541’),
('4', 'Muhammad', '1934.9436’),
('5', 'Leo', '1846.2651’),
('6', 'Jack', '1244.0034’),
('7', 'Harry', '1187.0017');
```

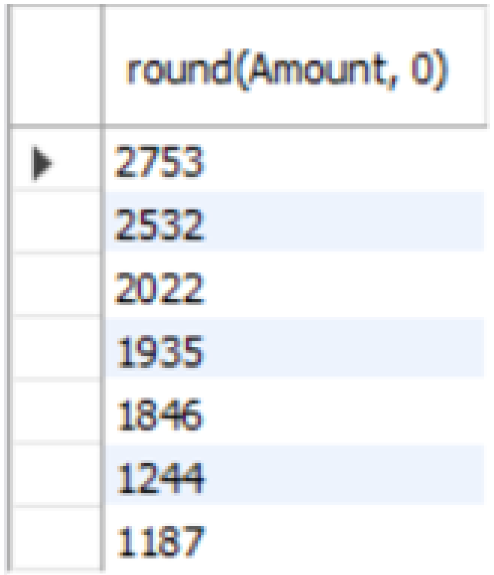
Assisted Practice: Numeric Function

3. Write a query to perform **round()** function up to 0 and 2 decimal places and perform **ceil()** and **floor()** functions.

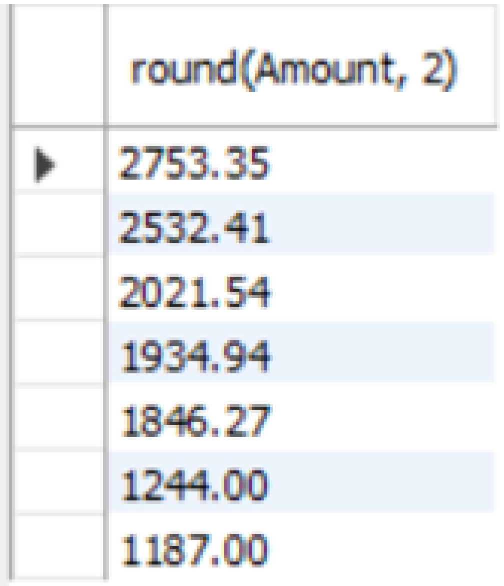
QUERY

```
1.SELECT round(Amount, 0)
FROM example.bill;

2.SELECT round(Amount, 2)
FROM example.bill;

3.SELECT ceil(Amount)
FROM example.bill;

4.SELECT floor(Amount)
FROM example.bill;
```

Assisted Practice: Lab Output

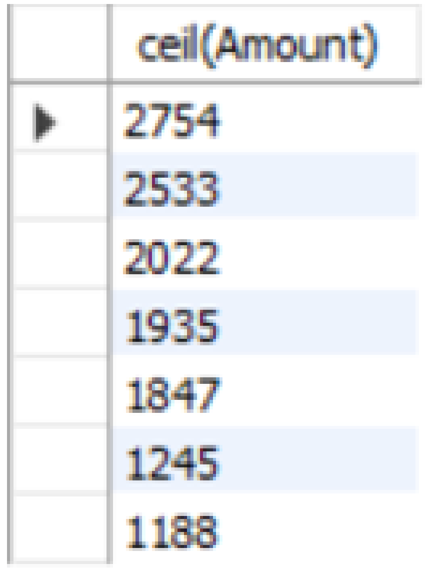


	round(Amount, 0)
▶	2753
	2532
	2022
	1935
	1846
	1244
	1187

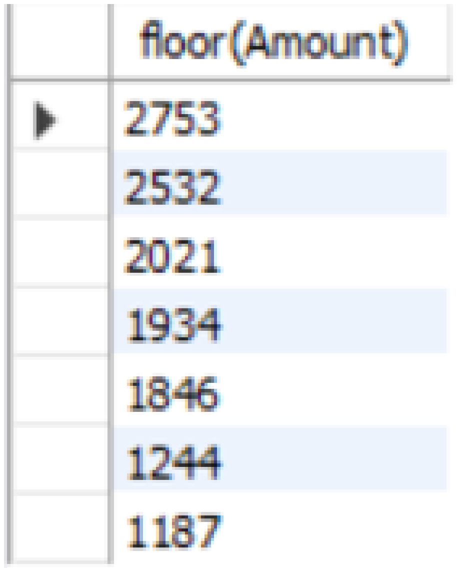


	round(Amount, 2)
▶	2753.35
	2532.41
	2021.54
	1934.94
	1846.27
	1244.00
	1187.00

Assisted Practice: Lab Output



ceil(Amount)
2754
2533
2022
1935
1847
1245
1188



floor(Amount)
2753
2532
2021
1934
1846
1244
1187

Date and Time Functions

Date and Time Functions

It helps to extract the time, date, and year as per the requirement.

DATE ()
EXTRACT ()
TIME ()
DATE FORMAT ()

Date Function

Definition

Date function extracts the date part from the given expression.

Syntax

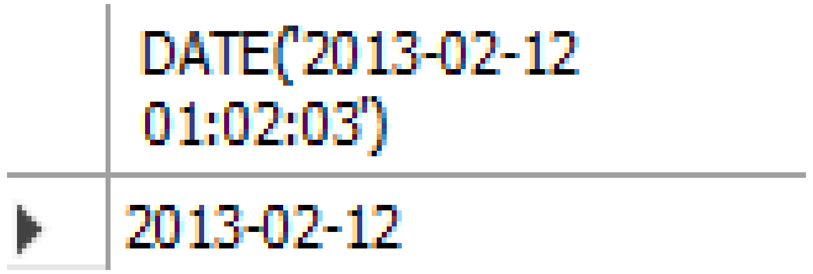
```
select date('expression');
```

Date Example

Example

```
SELECT DATE('2013-02-12 01:02:03')
```

Output



```
  | DATE('2013-02-12
  | 01:02:03')
▶ | 2013-02-12
```

Time Function

Definition

Time function extracts the time from the given expression.

Syntax

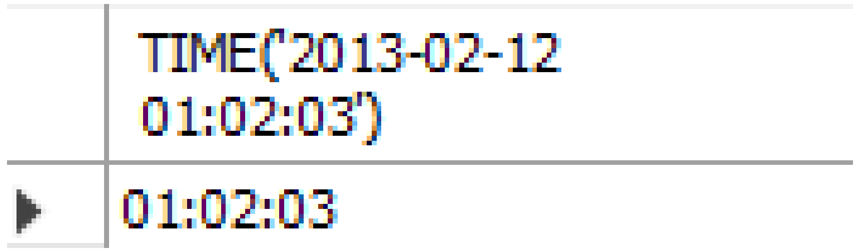
```
select time(expression);
```

Time Example

Example

```
SELECT TIME('2013-02-12 01:02:03')
```

Output



```
TIME('2013-02-12
01:02:03')
01:02:03
```

Extract Function

Definition

Extract function extracts the date, month, year, and time from the given expression.

Syntax

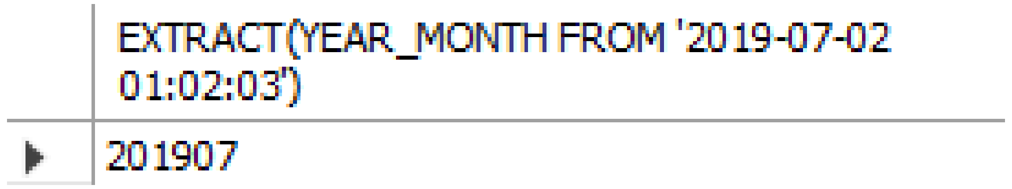
```
EXTRACT(part FROM expression)
```

Extract Example

Example

```
SELECT EXTRACT(YEAR_MONTH FROM '2019-07-02
01:02:03');
```

Output



```
EXTRACT(YEAR_MONTH FROM '2019-07-02
01:02:03')
201907
```

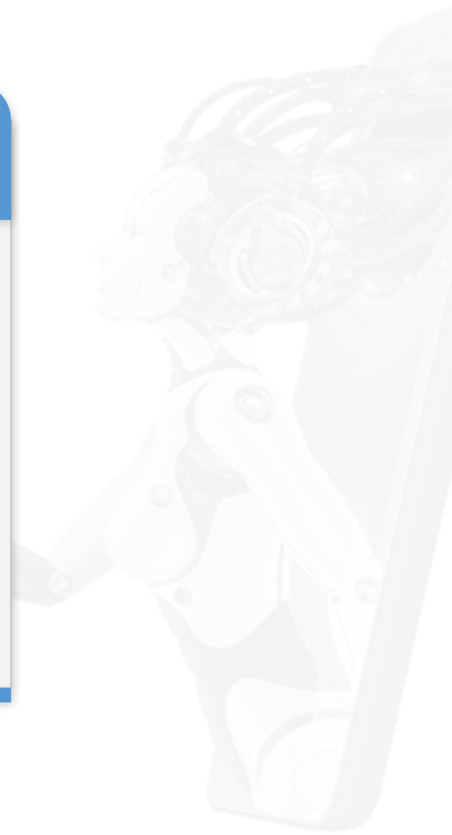
Date Format Function

Definition

Date format function returns the date in a specified format.

Syntax

```
select date_format(date, format_mask)
```

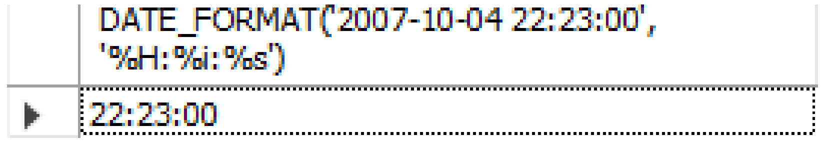


Date Format Example

Example

```
SELECT DATE_FORMAT('2007-10-04 22:23:00',
'%H:%i:%s');
```

Output

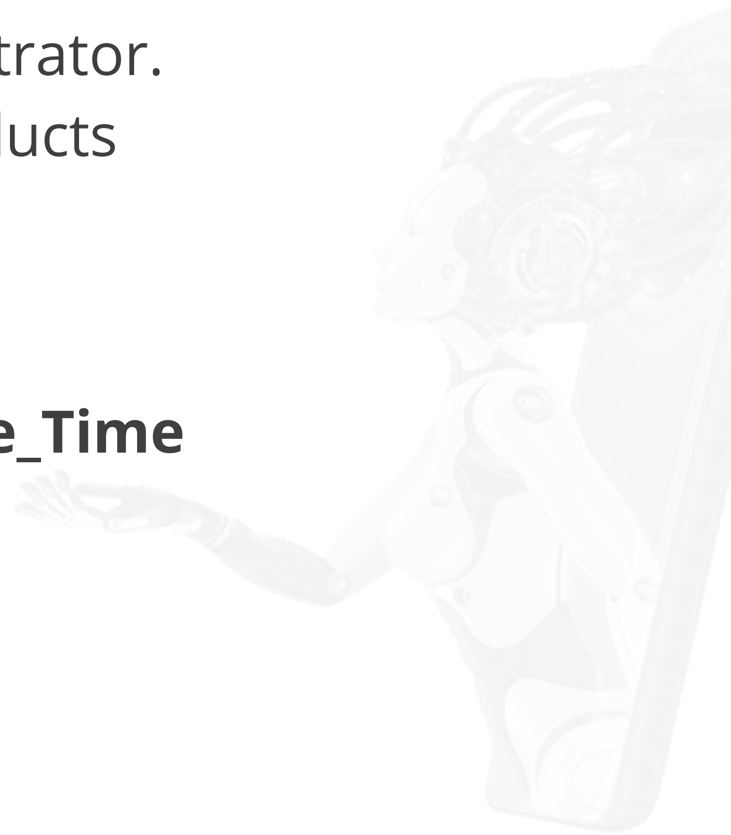


```
DATE_FORMAT('2007-10-04 22:23:00',
'%H:%i:%s')
22:23:00
```

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you to find the date, time, and year of the returned products while listing them in the American standard format.

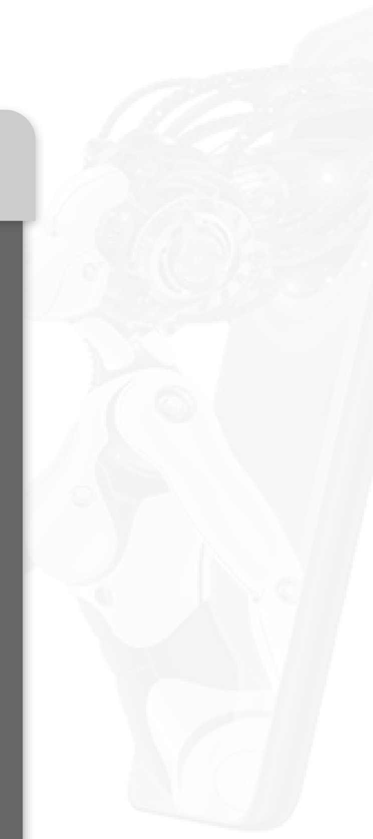
Objective: You are required to extract date, time, and year from the **Return_Date_Time** column of the table **Return product** and list the date in American format.



Solution

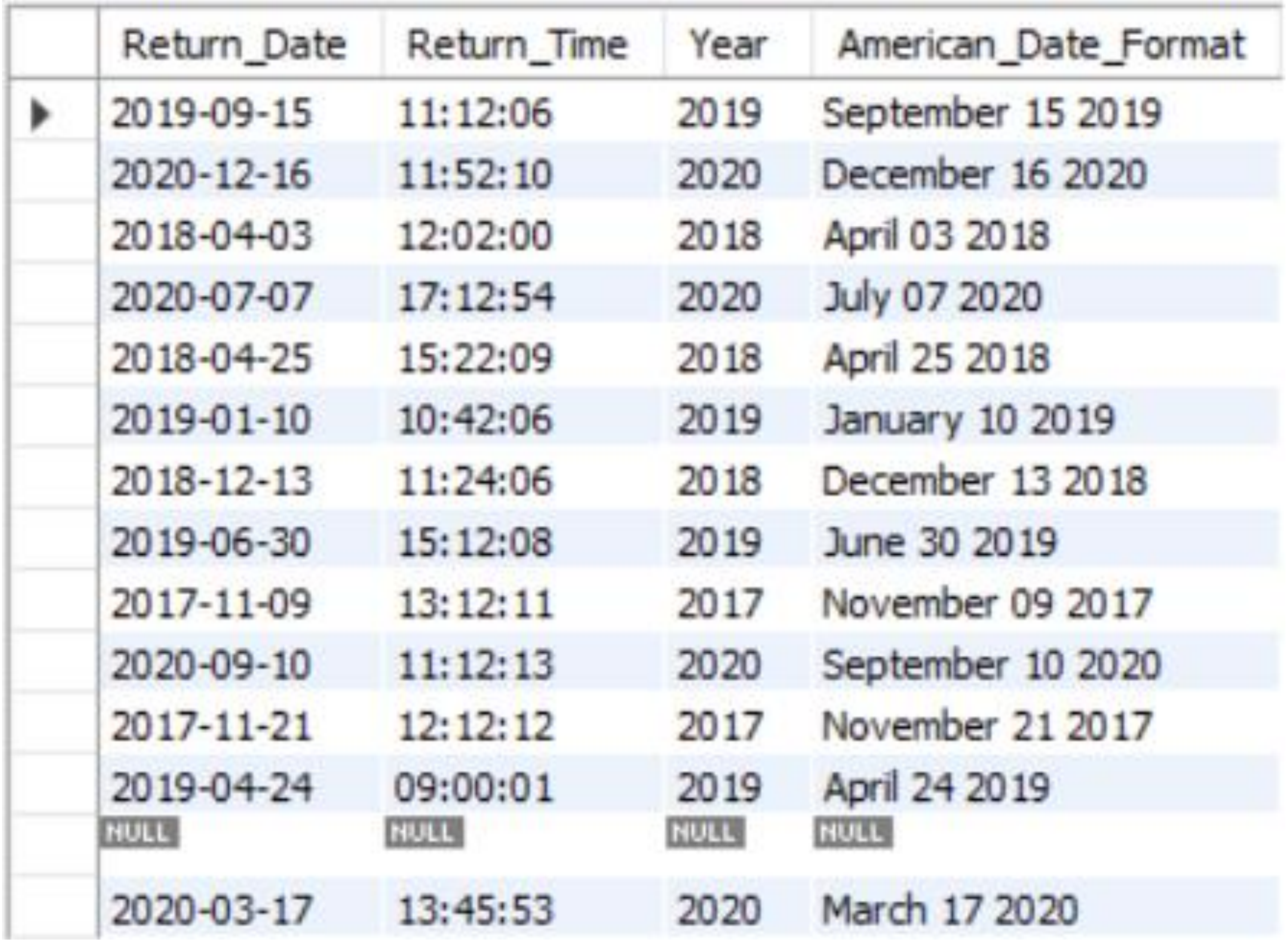
Query

```
SELECT Date(Return_Date_Time) as Return_Date, Time(Return_Date_Time) as
Return_Time, EXTRACT(YEAR FROM Return_Date_Time) AS Year,
DATE_FORMAT(Return_Date_Time, '%M %d %Y') as American_Date_Format

FROM example.return_products;
```



Output

After executing the query, return date is converted into standard American date format.



Return_Date	Return_Time	Year	American_Date_Format
2019-09-15	11:12:06	2019	September 15 2019
2020-12-16	11:52:10	2020	December 16 2020
2018-04-03	12:02:00	2018	April 03 2018
2020-07-07	17:12:54	2020	July 07 2020
2018-04-25	15:22:09	2018	April 25 2018
2019-01-10	10:42:06	2019	January 10 2019
2018-12-13	11:24:06	2018	December 13 2018
2019-06-30	15:12:08	2019	June 30 2019
2017-11-09	13:12:11	2017	November 09 2017
2020-09-10	11:12:13	2020	September 10 2020
2017-11-21	12:12:12	2017	November 21 2017
2019-04-24	09:00:01	2019	April 24 2019
NULL	NULL	NULL	NULL
2020-03-17	13:45:53	2020	March 17 2020

Handling Duplicate Records

Handling Duplicate Records

The duplicate records can be handled in two ways:

- Using DISTINCT and COUNT keywords to fetch the number of unique records.
- Using COUNT and GROUP BY keywords to eliminate the duplicate records.

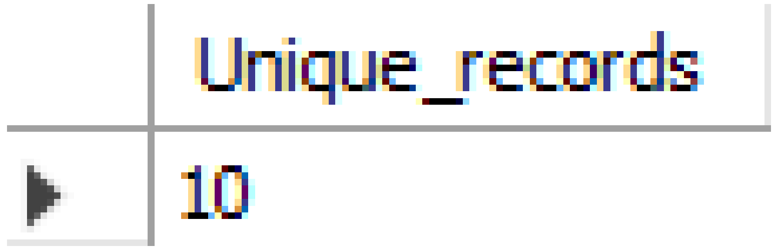
Handling Duplicate Records

Using DISTINCT and COUNT keywords to fetch the number of unique records.

Example

```
SELECT COUNT(DISTINCT(category)) AS
Unique_records FROM product;
```

Output



	Unique_records
▶	10

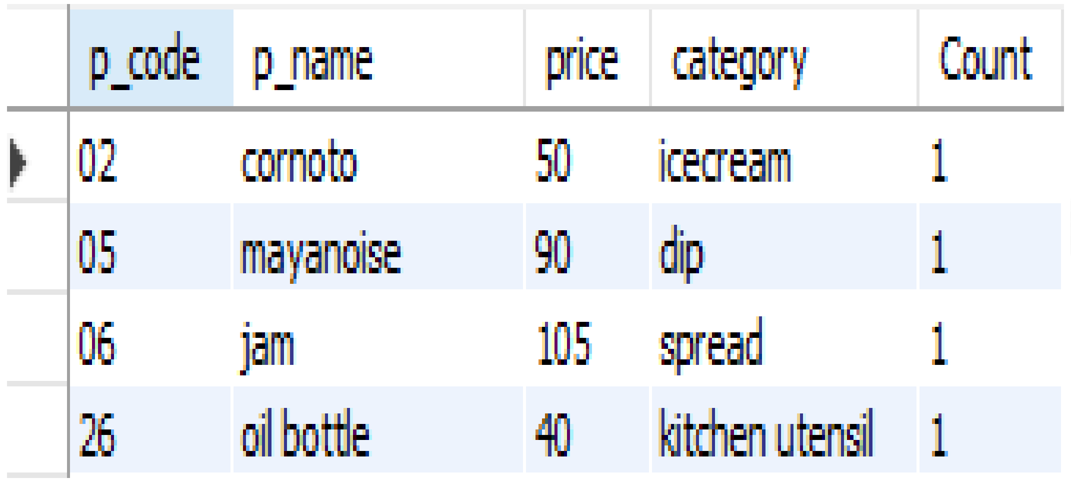
Handling Duplicate Records

Using COUNT and GROUP BY keywords to eliminate the duplicate records.

Example

```
SELECT p_code,p_name,price,category,
COUNT(*) as CountFROM productGROUP
BY categoryHAVING COUNT(*)  = 1
```

Output



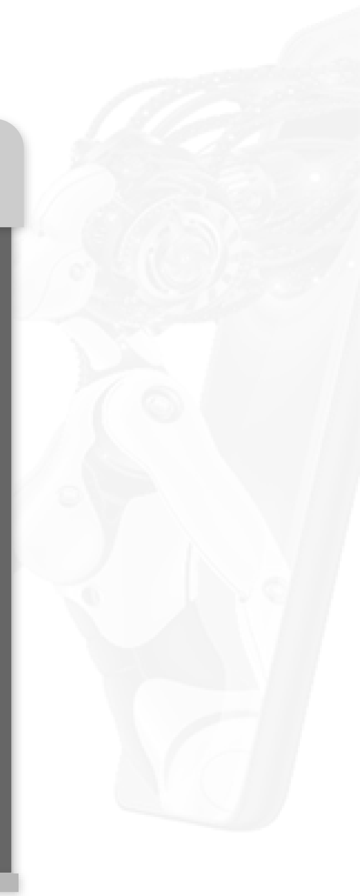
p_code	p_name	price	category	Count
02	cornoto	50	icecream	1
05	mayanoise	90	dip	1
06	jam	105	spread	1
26	oil bottle	40	kitchen utensil	1

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager informed you that the table of your superstore has duplicate customer IDs due to multiple orders from the same customer.

Objective: You are required to filter all the duplicate values and display the list of unique customers.

Solution

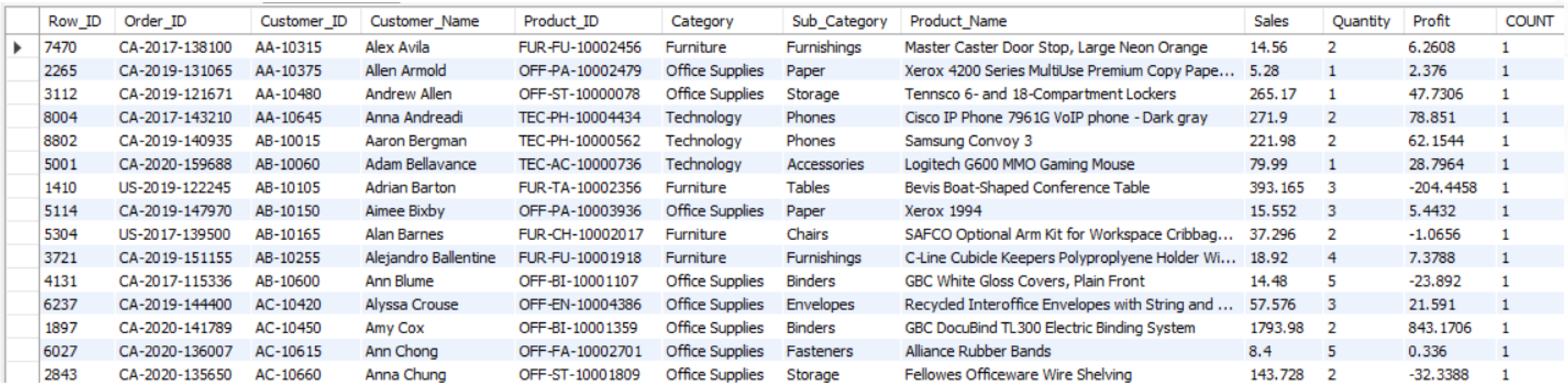


Query

```
SELECT * , COUNT(DISTINCT Customer_ID) as Count
FROM example.superstore
GROUP BY Customer_ID;
```

Output

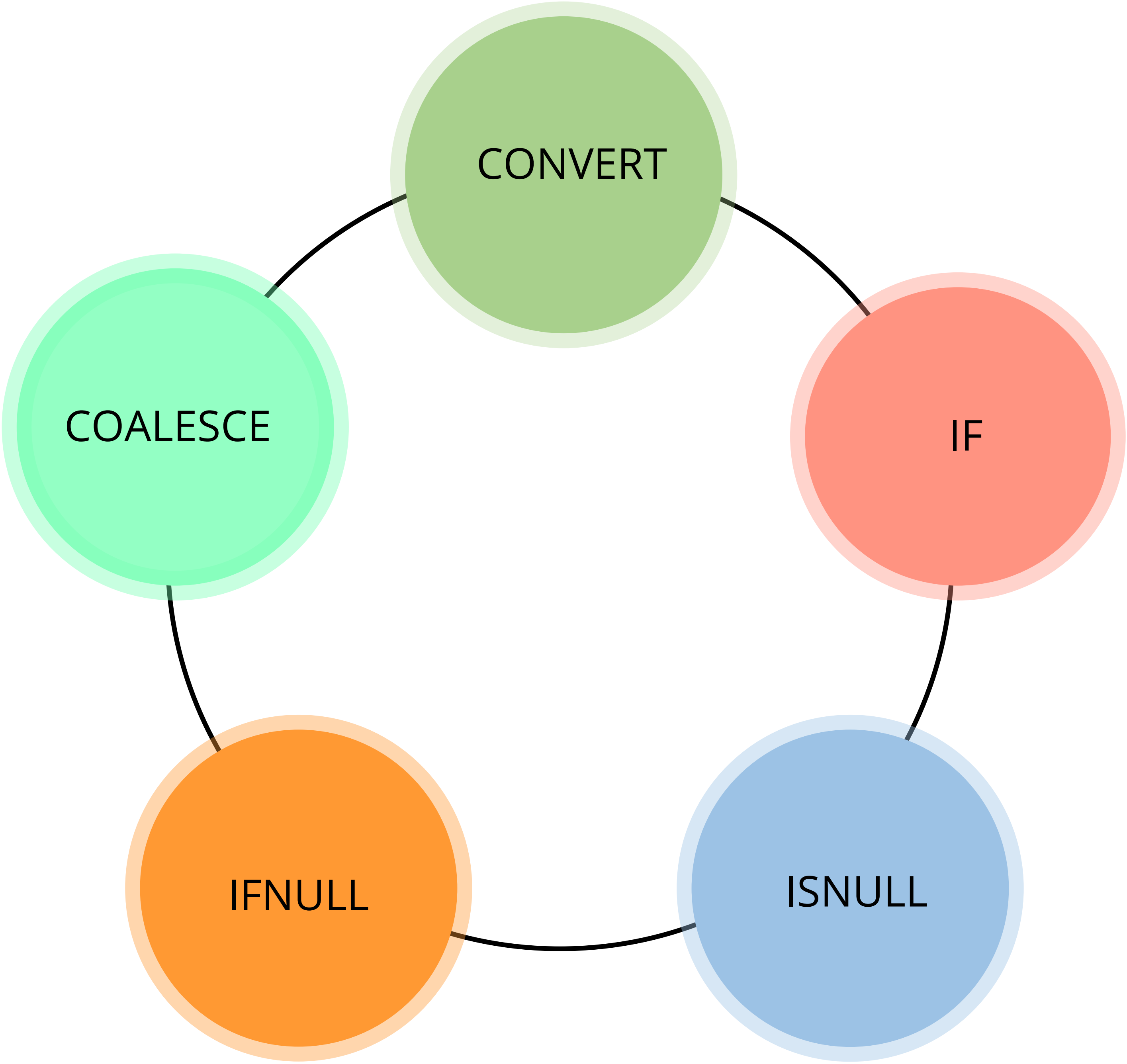
After executing the query, we get the list of unique customers.



	Row_ID	Order_ID	Customer_ID	Customer_Name	Product_ID	Category	Sub_Category	Product_Name	Sales	Quantity	Profit	COUNT
▶	7470	CA-2017-138100	AA-10315	Alex Avila	FUR-FU-10002456	Furniture	Furnishings	Master Caster Door Stop, Large Neon Orange	14.56	2	6.2608	1
	2265	CA-2019-131065	AA-10375	Allen Armold	OFF-PA-10002479	Office Supplies	Paper	Xerox 4200 Series MultiUse Premium Copy Pape...	5.28	1	2.376	1
	3112	CA-2019-121671	AA-10480	Andrew Allen	OFF-ST-10000078	Office Supplies	Storage	Tennsco 6- and 18-Compartment Lockers	265.17	1	47.7306	1
	8004	CA-2017-143210	AA-10645	Anna Andreadi	TEC-PH-10004434	Technology	Phones	Cisco IP Phone 7961G VoIP phone - Dark gray	271.9	2	78.851	1
	8802	CA-2019-140935	AB-10015	Aaron Bergman	TEC-PH-10000562	Technology	Phones	Samsung Convoy 3	221.98	2	62.1544	1
	5001	CA-2020-159688	AB-10060	Adam Bellavance	TEC-AC-10000736	Technology	Accessories	Logitech G600 MMO Gaming Mouse	79.99	1	28.7964	1
	1410	US-2019-122245	AB-10105	Adrian Barton	FUR-TA-10002356	Furniture	Tables	Bevis Boat-Shaped Conference Table	393.165	3	-204.4458	1
	5114	CA-2019-147970	AB-10150	Aimee Bixby	OFF-PA-10003936	Office Supplies	Paper	Xerox 1994	15.552	3	5.4432	1
	5304	US-2017-139500	AB-10165	Alan Barnes	FUR-CH-10002017	Furniture	Chairs	SAFCO Optional Arm Kit for Workspace Cribbag...	37.296	2	-1.0656	1
	3721	CA-2019-151155	AB-10255	Alejandro Ballentine	FUR-FU-10001918	Furniture	Furnishings	C-Line Cubicle Keepers Polyproplyene Holder Wi...	18.92	4	7.3788	1
	4131	CA-2017-115336	AB-10600	Ann Blume	OFF-BI-10001107	Office Supplies	Binders	GBC White Gloss Covers, Plain Front	14.48	5	-23.892	1
	6237	CA-2019-144400	AC-10420	Alyssa Crouse	OFF-EN-10004386	Office Supplies	Envelopes	Recycled Interoffice Envelopes with String and ...	57.576	3	21.591	1
	1897	CA-2020-141789	AC-10450	Amy Cox	OFF-BI-10001359	Office Supplies	Binders	GBC DocuBind TL300 Electric Binding System	1793.98	2	843.1706	1
	6027	CA-2020-136007	AC-10615	Ann Chong	OFF-FA-10002701	Office Supplies	Fasteners	Alliance Rubber Bands	8.4	5	0.336	1
	2843	CA-2020-135650	AC-10660	Anna Chung	OFF-ST-10001809	Office Supplies	Storage	Fellowes Officeware Wire Shelving	143.728	2	-32.3388	1

Miscellaneous Functions

Miscellaneous Functions and Its Types



Convert Function

Definition

Convert function converts a value into a specified data type.

Syntax

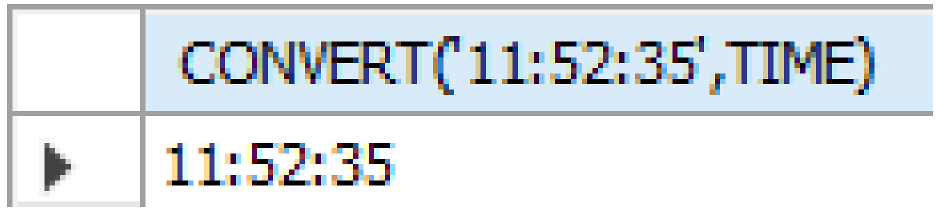
```
select CONVERT(value,datatype);
```

Convert Example

Example

```
SELECT CONVERT('11:52:35',TIME)
```

Output



	CONVERT('11:52:35',TIME)
▶	11:52:35

IF Function

Definition

IF function returns value1 if the expression is TRUE, or value2 if the expression is FALSE.

Syntax

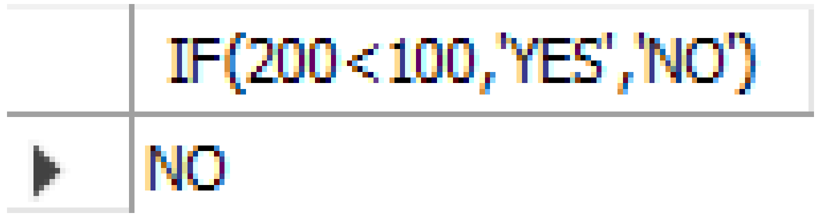
```
select  IF(expression,VALUE1,VALUE2);
```

IF Example

Example

```
SELECT IF(200<100,'YES','NO')
```

Output



```
IF(200<100,'YES','NO')
NO
```

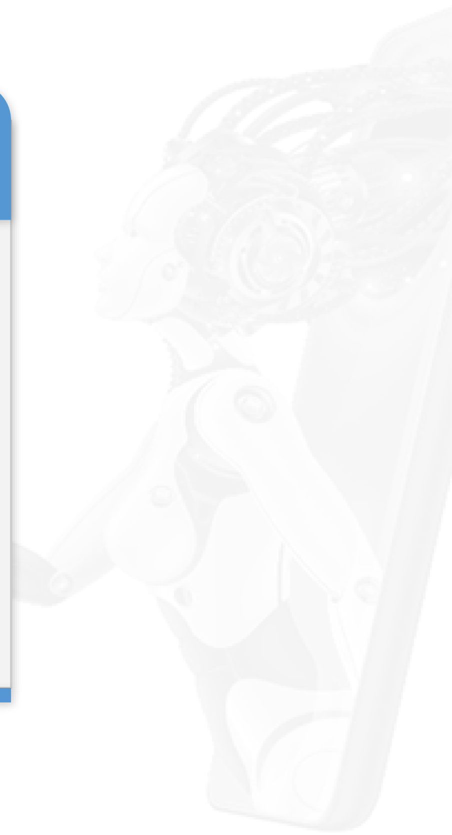
ISNULL Function

Definition

ISNULL function returns 1 if the expression is NULL or else 0 if the expression is NOT NULL.

Syntax

```
select  ISNULL(expression)
```

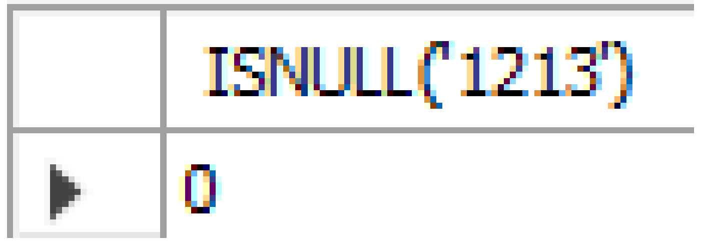


ISNULL Example

Example

```
SELECT ISNULL('1213')
```

Output



	ISNULL('1213')
▶	0

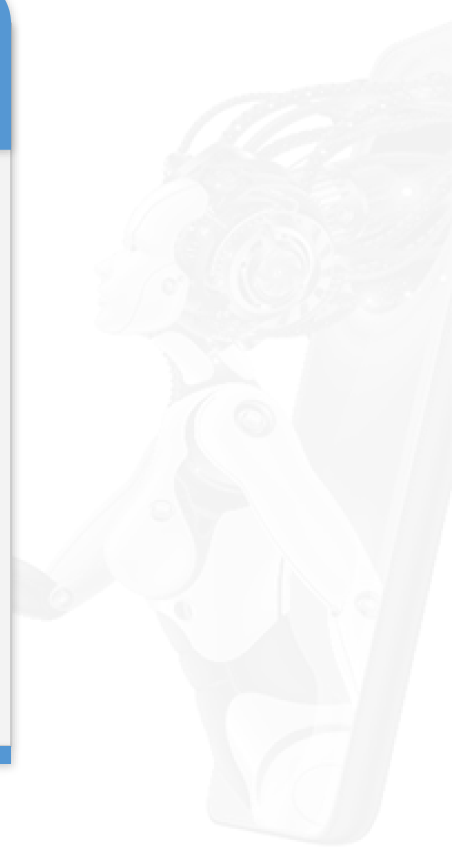
IFNULL Function

Definition

- IFNULL function takes two expression.
- It returns the first expression if the first expression is NOT NULL otherwise returns the second expression.

Syntax

```
select  IFNULL(expression1,expression2)
```

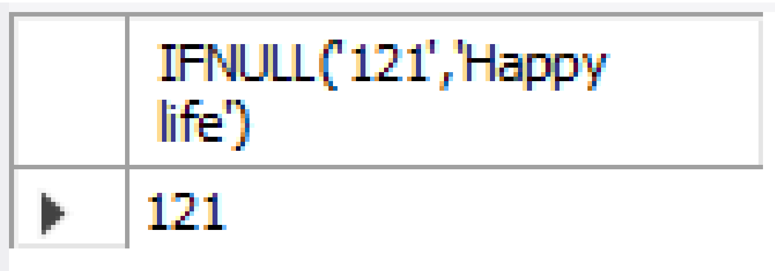


IFNULL Example

Example

```
SELECT IFNULL('121','Happy life')
```

Output



	IFNULL('121','Happy life')
▶	121

Coalesce Function

Definition

Coalesce function returns the first non-null value from a list of expressions.

Syntax

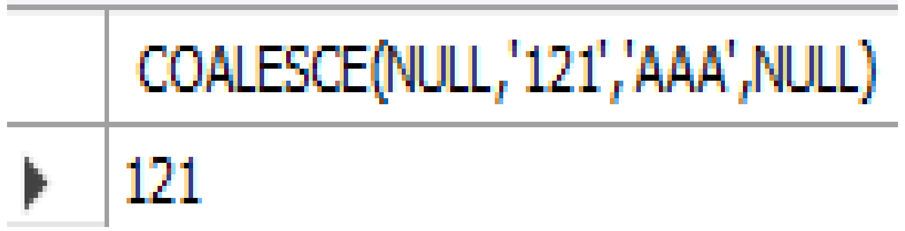
```
select  COALESCE(expression1,expression2,.....)
```

Coalesce Example

Example

```
SELECT COALESCE(NULL,'121','AAA',NULL)
```

Output



```
COALESCE(NULL,'121','AAA',NULL)
121
```

Problem Statement

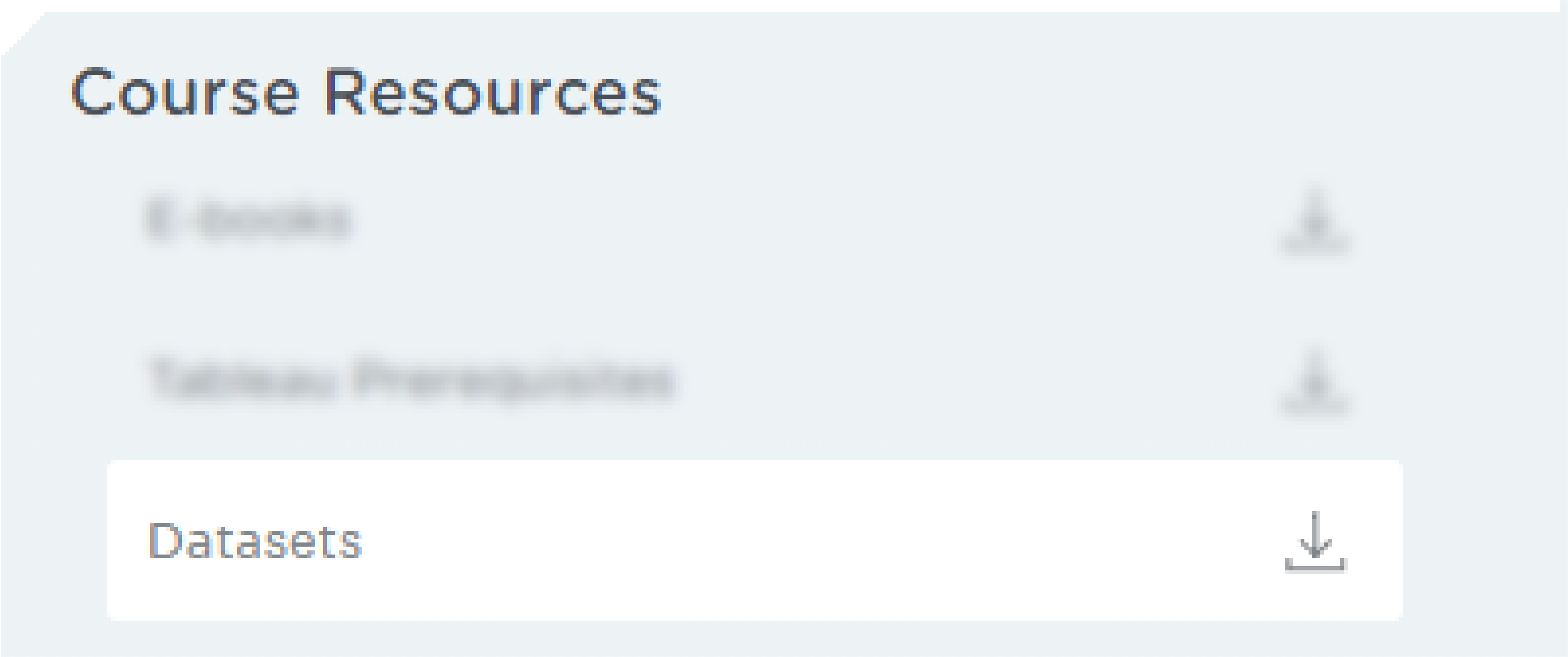
Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you to cross-check the database for any NULL value.

Objective: You are required to check for NULL value in the database and display the output message as **problem in the record** if any NULL value is found in the table.

Problem Statement

Steps to be performed:

1. Download the **return_products** table from course resources and import it in MySQL workbench.



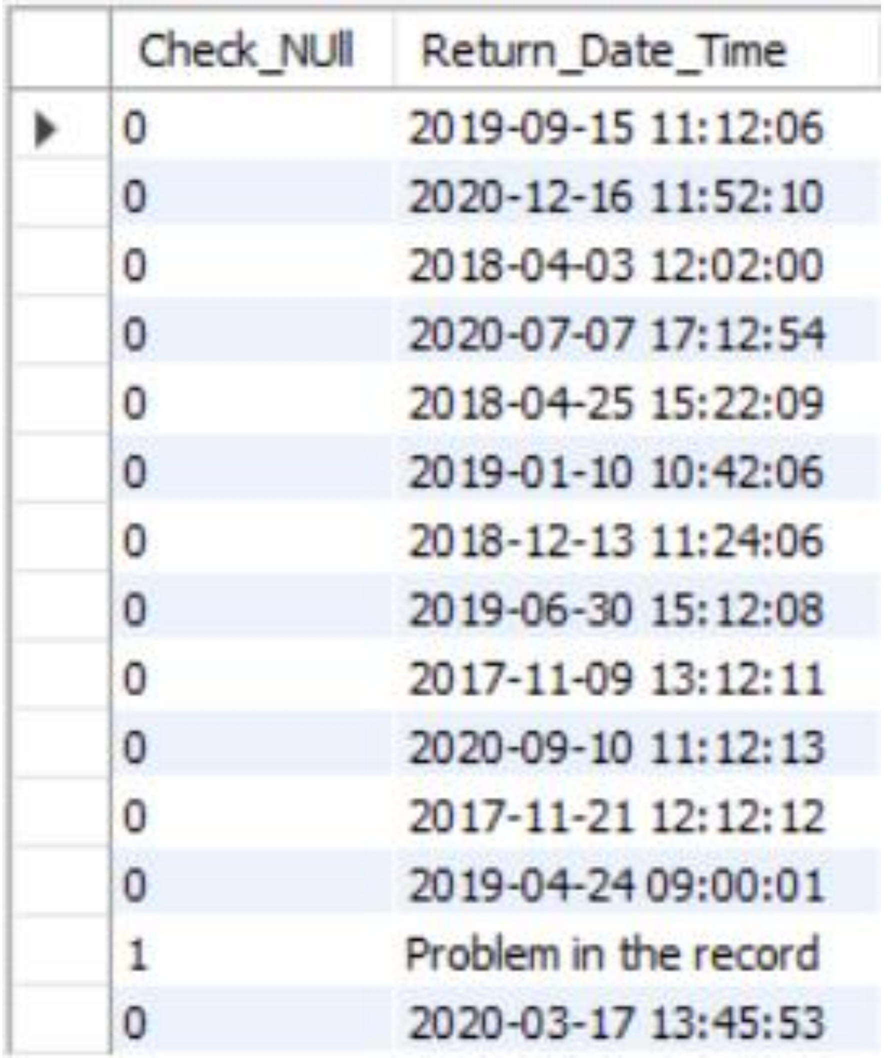
Solution

Query

```
SELECT ISNULL(Return_Date_Time) as Check_NUll, IFNULL(Return_Date_Time,'Probem in
the record') as Return_Date_Time

FROM example.return_products;
```

Output

After executing the query, a message is displayed in the table when it encounters a NULL value.

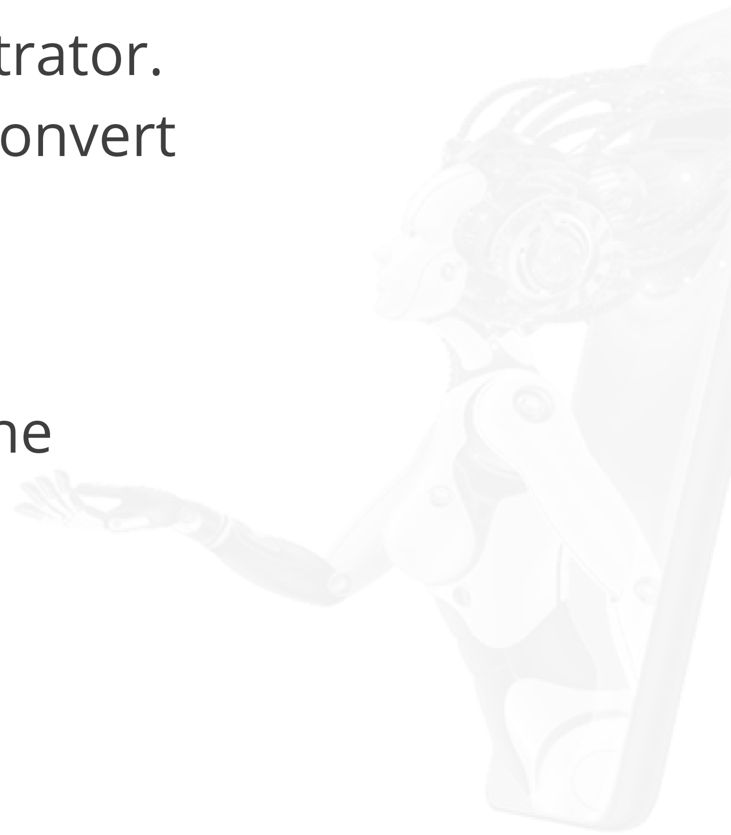


	Check_NUll	Return_Date_Time
▶	0	2019-09-15 11:12:06
	0	2020-12-16 11:52:10
	0	2018-04-03 12:02:00
	0	2020-07-07 17:12:54
	0	2018-04-25 15:22:09
	0	2019-01-10 10:42:06
	0	2018-12-13 11:24:06
	0	2019-06-30 15:12:08
	0	2017-11-09 13:12:11
	0	2020-09-10 11:12:13
	0	2017-11-21 12:12:12
	0	2019-04-24 09:00:01
	1	Problem in the record
	0	2020-03-17 13:45:53

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you to check the profit or loss in the profit column and convert the datatype of the quantity column to decimal.

Objective: You are required to check for profit in the profit column and convert the datatype of the quantity column to decimal.



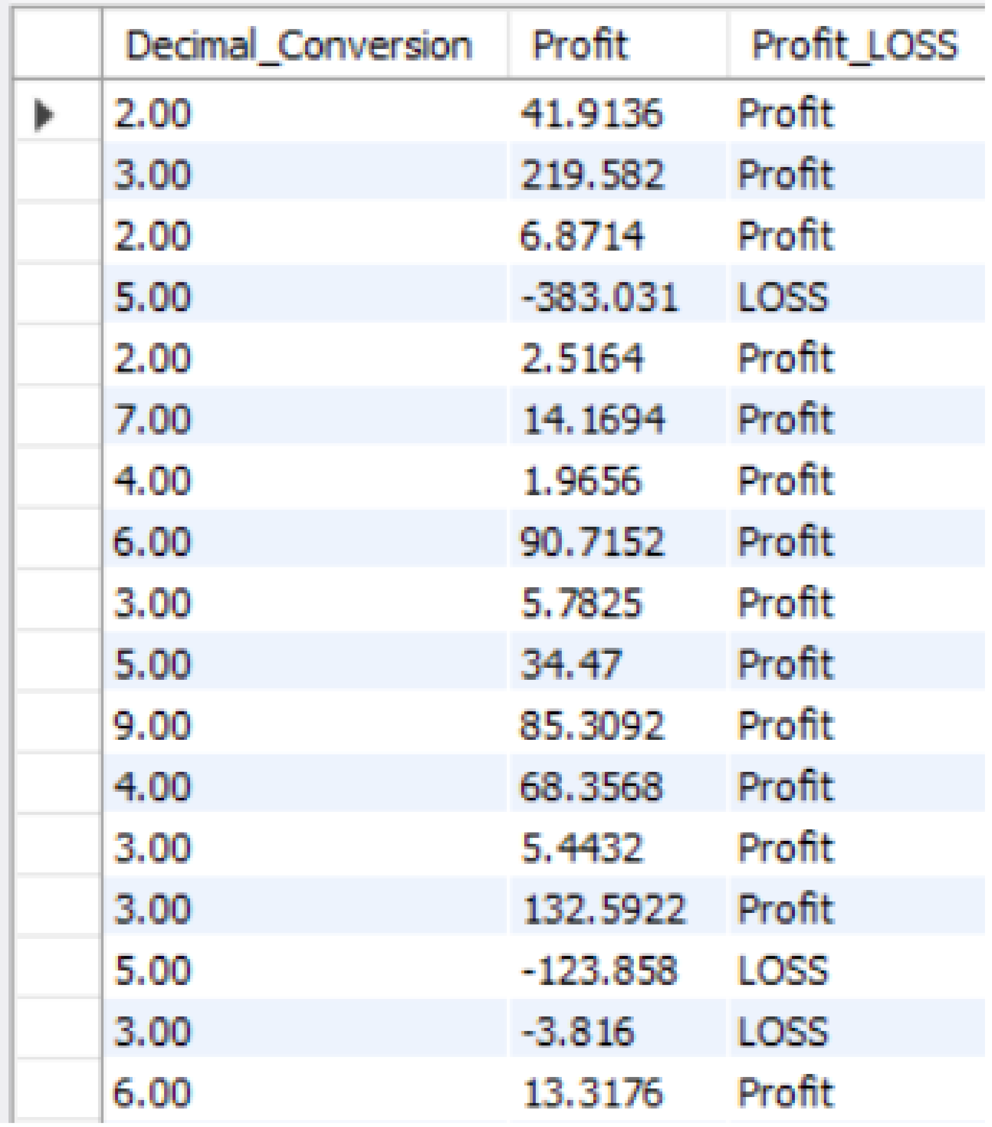
Solution

Query

```
SELECT Convert(Quantity, Decimal(10,2)) as Decimal_Conversion, Profit,
IF((ABS(Profit))!=profit, 'LOSS', 'Profit') as Profit_LOSS

FROM example.superstore;
```

Output

After executing the query, we can check the profit or loss.

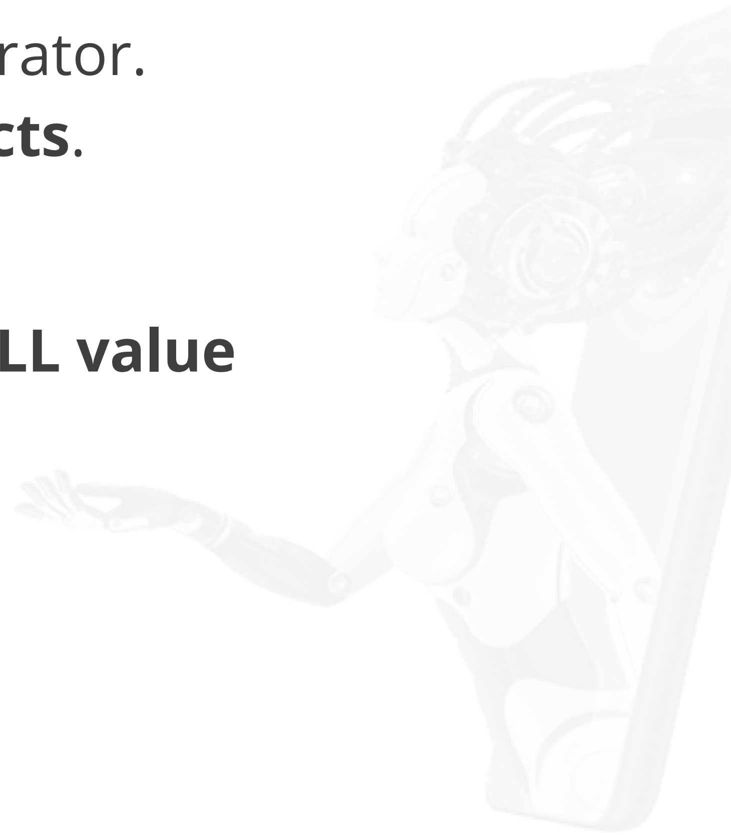


Decimal_Conversion	Profit	Profit_LOSS
2.00	41.9136	Profit
3.00	219.582	Profit
2.00	6.8714	Profit
5.00	-383.031	LOSS
2.00	2.5164	Profit
7.00	14.1694	Profit
4.00	1.9656	Profit
6.00	90.7152	Profit
3.00	5.7825	Profit
5.00	34.47	Profit
9.00	85.3092	Profit
4.00	68.3568	Profit
3.00	5.4432	Profit
3.00	132.5922	Profit
5.00	-123.858	LOSS
3.00	-3.816	LOSS
6.00	13.3176	Profit

Problem Statement

Problem Scenario You are working in a superstore as a junior database administrator. Your manager has asked you to check for NULL values in the table **return_products**.

Objective: You are required to check for NULL values in the table and display **NULL value** as a message if any **NULL value** exists in the table.



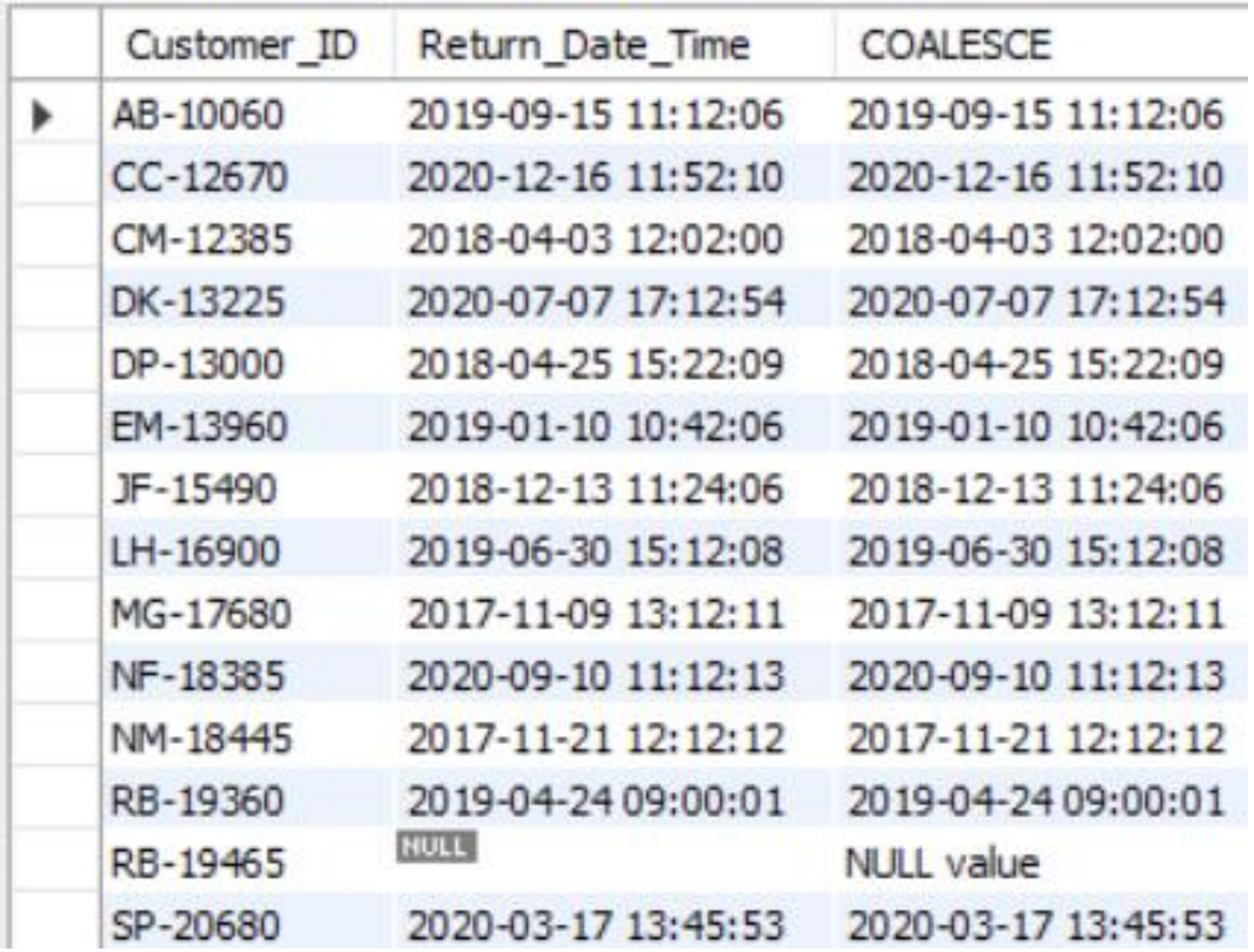
Solution

Query

```
SELECT *, COALESCE(Return_Date_Time,NULL,'NULL value',NULL,NULL,5) as COALESCE
FROM example.return_products;
```

Output

After executing the query, the message is displayed as NULL value when we encounter the first NON-NULL value in the table.



Customer_ID	Return_Date_Time	COALESCE
AB-10060	2019-09-15 11:12:06	2019-09-15 11:12:06
CC-12670	2020-12-16 11:52:10	2020-12-16 11:52:10
CM-12385	2018-04-03 12:02:00	2018-04-03 12:02:00
DK-13225	2020-07-07 17:12:54	2020-07-07 17:12:54
DP-13000	2018-04-25 15:22:09	2018-04-25 15:22:09
EM-13960	2019-01-10 10:42:06	2019-01-10 10:42:06
JF-15490	2018-12-13 11:24:06	2018-12-13 11:24:06
LH-16900	2019-06-30 15:12:08	2019-06-30 15:12:08
MG-17680	2017-11-09 13:12:11	2017-11-09 13:12:11
NF-18385	2020-09-10 11:12:13	2020-09-10 11:12:13
NM-18445	2017-11-21 12:12:12	2017-11-21 12:12:12
RB-19360	2019-04-24 09:00:01	2019-04-24 09:00:01
RB-19465	NULL	NULL value
SP-20680	2020-03-17 13:45:53	2020-03-17 13:45:53

General Functions

General Functions and Its Types

Session User

ASCII

Version

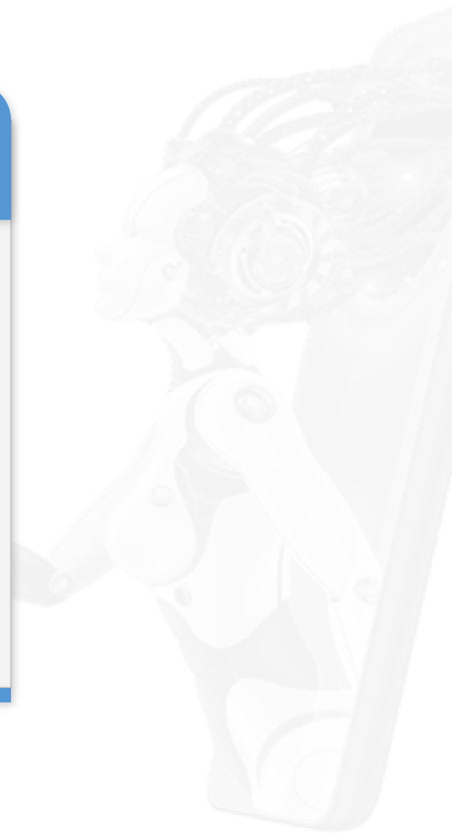
ASCII Function

Definition

ASCII function returns the ASCII value of the specified character.

Syntax

```
select ASCII(expression);
```

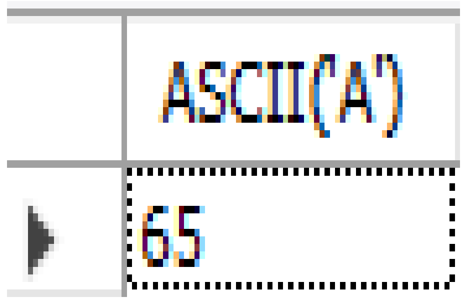


ASCII Example

Example

```
SELECT ASCII('A')
```

Output



```
ASCII('A')
65
```

Version Function

Definition

Version function returns the current version of the MySQL database.

Syntax

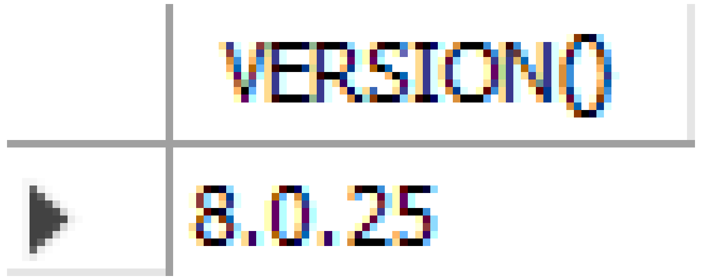
```
select VERSION()
```

Version Example

Example

```
SELECT VERSION()
```

Output



```
VERSION()
8.0.25
```

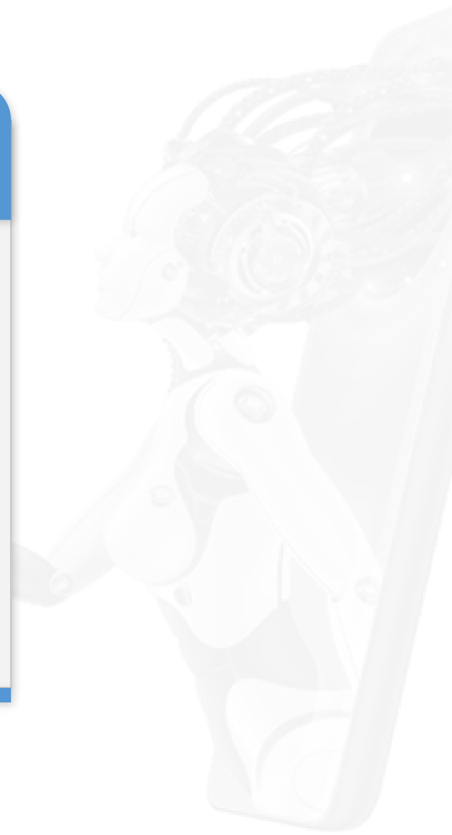
Session User Function

Definition

Session user function returns the current username and host name for the MySQL connection.

Syntax

```
select  SESSION_USER()
```

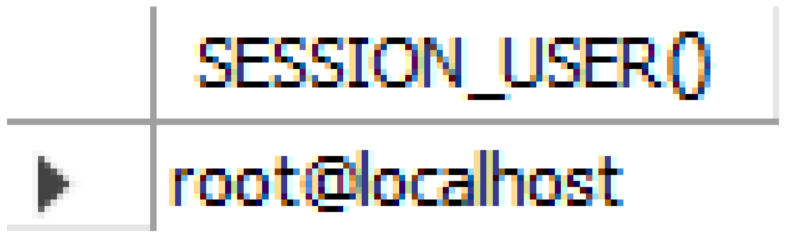


Session User Example

Example

```
SELECT SESSION_USER()
```

Output



	SESSION_USER()
▶	root@localhost

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you about the current MySQL version installed on your system.

Objective: You are required to find the current version of MySQL installed on your system.

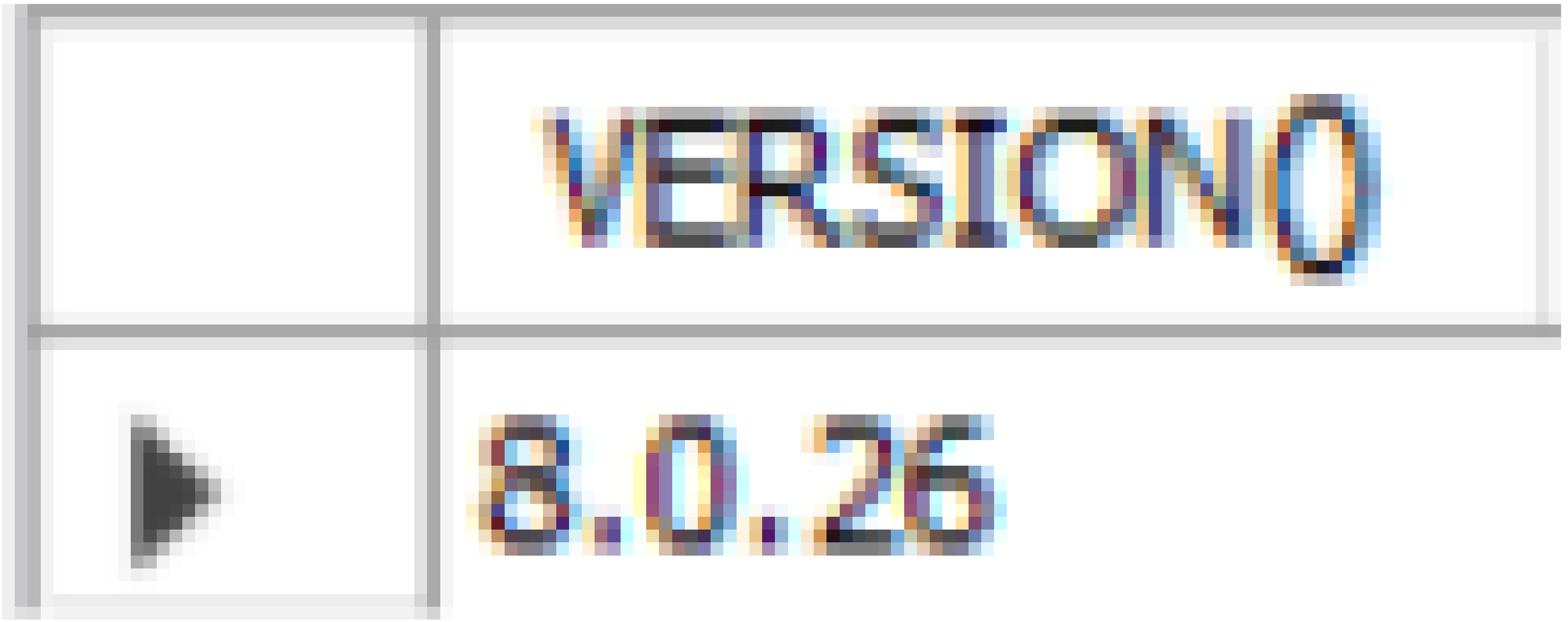
Solution

Query

```
SELECT VERSION();
```

Output

After executing the query, current version of MySQL is shown in the output.

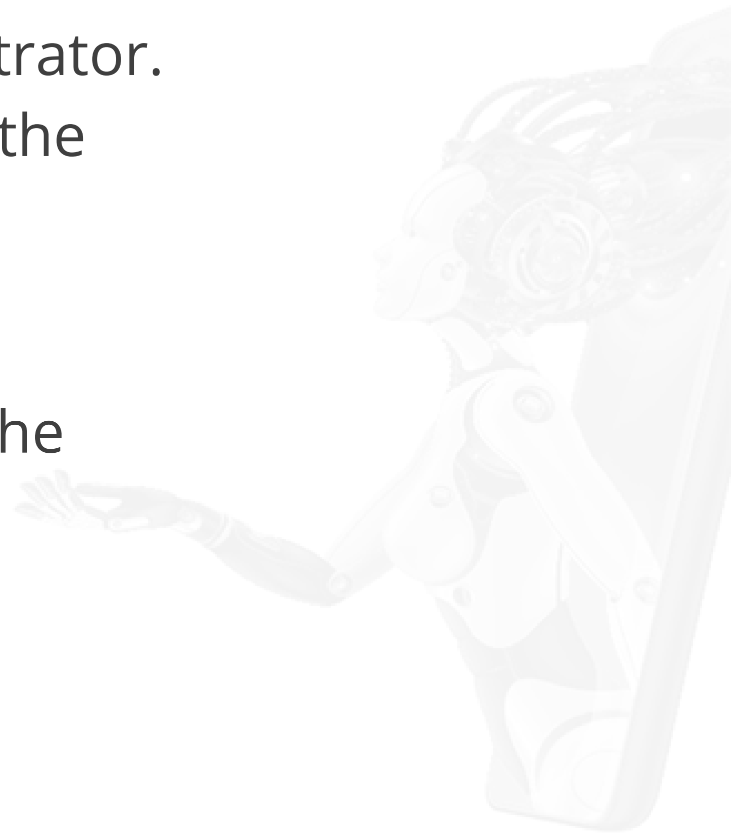


	VERSION()
▶	8.0.26

Problem Statement

Problem Scenario: You are working in a superstore as a junior database administrator. Your manager has asked you to find the ASCII value of the first character from all the records in column **Customer_Name**.

Objective: You are required to find the ASCII value of the first character from all the records in column **Customer_Name**.



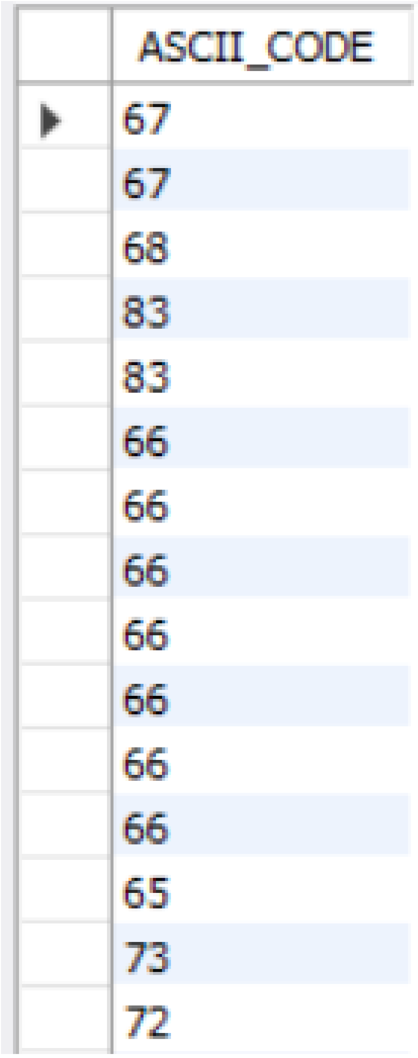
Solution

Query

```
SELECT ASCII(Customer_Name) as ASCII_CODE
From example.superstore;
```

Output

After executing the query, the ASCII value of the first character from all the records in **Customer_Name** column is shown as the following output:



	ASCII_CODE
▶	67
	67
	68
	83
	83
	66
	66
	66
	66
	66
	66
	66
	65
	73
	72

Key Takeaways

- SQL functions are basic subprograms used extensively to handle or manipulate data.
- Aggregate functions allow performing the calculation on a set of values to return a single scalar value.
- Scalar functions return a single value from an input value. It works on each record independently.
- String functions are used for string manipulation.
- Duplicate records can be handled by using the keywords- DISTINCT, COUNT, and GROUP BY.

Knowledge Check

Knowledge Check

1

Which one of the following is an aggregate function?

A. Sum ( )

B. Date ( )

C. Concat ( )

D. Trim ( )

Knowledge Check

1

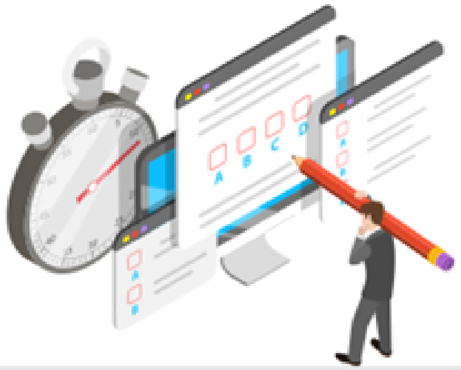
Which one of the following is an aggregate function?

A. Sum ( )

B. Date ( )

C. Concat ( )

D. Trim ( )



The correct answer is **A**

Sum function is an aggregate function.

Knowledge Check

2

Which of the following works on each record independently?

A. Aggregate function

B. Scalar function

C. Date and time function

D. Numeric function

Knowledge Check

2

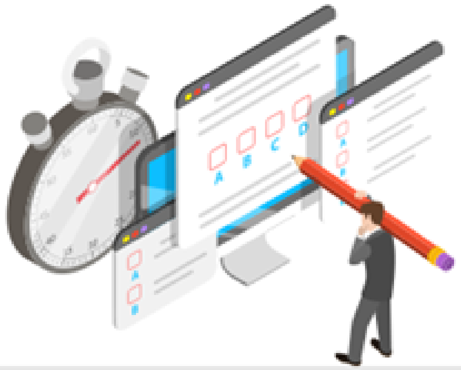
Which of the following works on each record independently?

A. Aggregate function

B. Scalar function

C. Date and time function

D. Numeric function



The correct answer is **B**

Scalar function works in each record independently.

Knowledge Check

3

Which of the following function returns largest integer value which is less than or equal to the given number ?

A. Ceil ( )

B. Floor ( )

C. Round ( )

D. MOD ( )

Knowledge Check

3

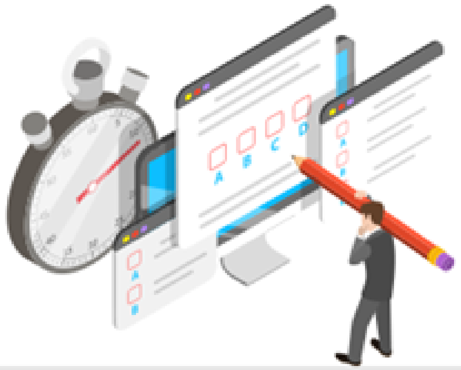
Which of the following function returns largest integer value which is less than or equal to the given number ?

A. Ceil ( )

B. Floor ( )

C. Round ( )

D. MOD ( )



The correct answer is **B**

Floor function returns largest integer value which is less than or equal to the given number.

Knowledge Check
4

Which of the following function helps to change a value into specific data type?

A. Convert ( )

B. IFNULL ( )

C. Coalesce ( )

D. ISNULL ( )

Knowledge Check

4

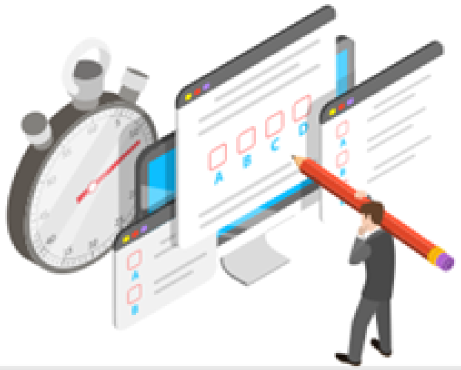
Which of the following function helps to change a value into specific data type?

A. Convert ( )

B. IFNULL ( )

C. Coalesce ( )

D. ISNULL ( )



The correct answer is **A**

Convert function helps to convert a value into specific data type.

Lesson-End Project: Patient Diagnosis Report

Problem statement:

You are a data analyst working in a hospital and you have been asked to store the patients’ diagnosis reports as a best practice.

Objective:

The objective is to design a database to retrieve, update, and modify the patients’ details to keep track of the patients’ health.

Note: Download the **patients_datasets.csv** file from **Course Resources** to perform the required tasks

Lesson-End Project: Patient Diagnosis Report

Tasks to be performed:

1. Write a query to create a **patients** table with the date, patient ID, patient name, age, weight, gender, location, phone number, disease, doctor name, and doctor ID fields
2. Write a query to insert values into the **patients** table
3. Write a query to display the total number of patients in the table
4. Write a query to display the patient ID, patient name, gender, and disease of the oldest (age) patient

LESSON-END PROJECT

Lesson-End Project: Patient Diagnosis Report

Tasks to be performed:

5. Write a query to display the patient ID and patient name of all entries with the current date
6. Write a query to display the old patient name and new patient name in uppercase
7. Write a query to display the patients' names along with the total number of characters in their name
8. Write a query to display the gender of the patient as M or F along with the patient's name

Lesson-End Project: Patient Diagnosis Report

Tasks to be performed:

9. Write a query to combine the patient's name and doctor's name in a new column
10. Write a query to display the patients' age along with the logarithmic value (base 10) of their age
11. Write a query to extract the year for a given date and place it in a separate column
12. Write a query to check the patient's name and doctor's name are similar and display **NULL**, else return the patient's name

Lesson-End Project: Patient Diagnosis Report

Tasks to be performed:

13. Write a query to check if a patient's age is greater than 40 and display **Yes** if it is and **No** if it isn't
14. Write a query to display duplicate entries in the doctor name column

Note: Download the solution document from the **Course Resources** section and follow the steps given in the document