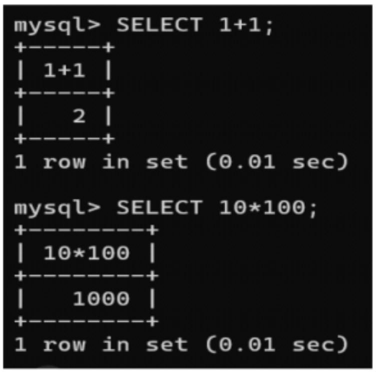
CN-Lec8-String Operations + Aggregate Functions

String Operations + Aggregate Functions

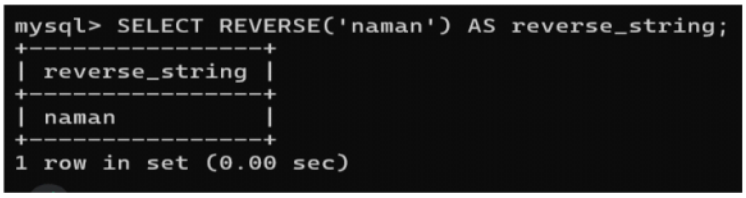
Session Overview

- ☑ Understand different types of aggregate functions.
- ☑ Use the GROUP BY function effectively.
- ☑ Apply the HAVING clause for advanced filtering.
- ☑ Utilise scalar functions like ROUND and ABS.

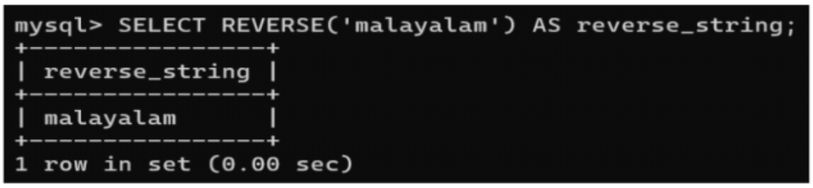


```
mysql> SELECT 1+1;
+------+
| 1+1 |
+------+
|    2 |
+------+
1 row in set (0.01 sec)

mysql> SELECT 10*100;
+---------+
| 10*100 |
+---------+
|    1000 |
+---------+
1 row in set (0.01 sec)
```

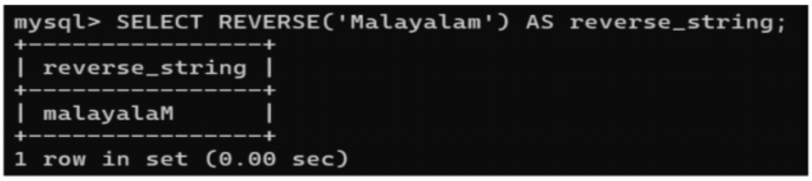
REVERSE



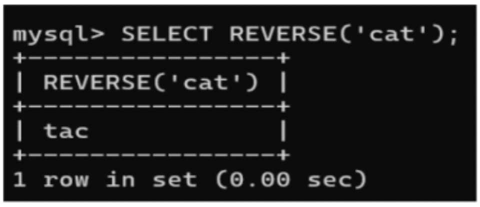
```
mysql> SELECT REVERSE('naman') AS reverse_string;
+----------------+
| reverse_string |
+----------------+
| naman          |
+----------------+
1 row in set (0.00 sec)
```



```
mysql> SELECT REVERSE('malayalam') AS reverse_string;
+----------------+
| reverse_string |
+----------------+
| malayalam      |
+----------------+
1 row in set (0.00 sec)
```



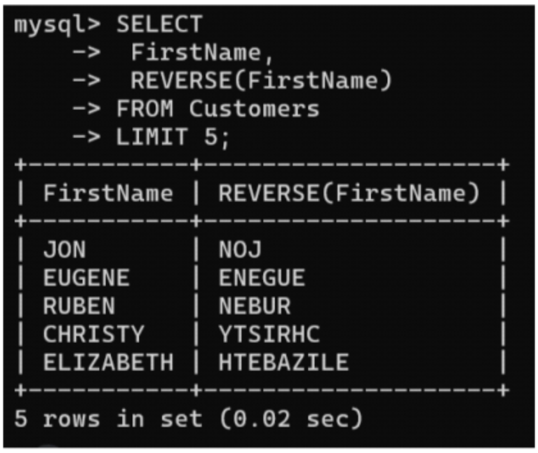
```
mysql> SELECT REVERSE('Malayalam') AS reverse_string;
+----------------+
| reverse_string |
+----------------+
| malayalaM      |
+----------------+
1 row in set (0.00 sec)
```



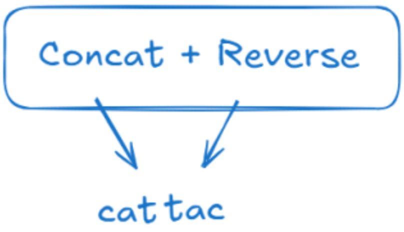
```
mysql> SELECT REVERSE('cat');
+----------------+
| REVERSE('cat') |
+----------------+
| tac            |
+----------------+
1 row in set (0.00 sec)
```

What is Palindrome

"racecar", "mom", "nitin", "dad", "level", "madam", "wow", "rotator".

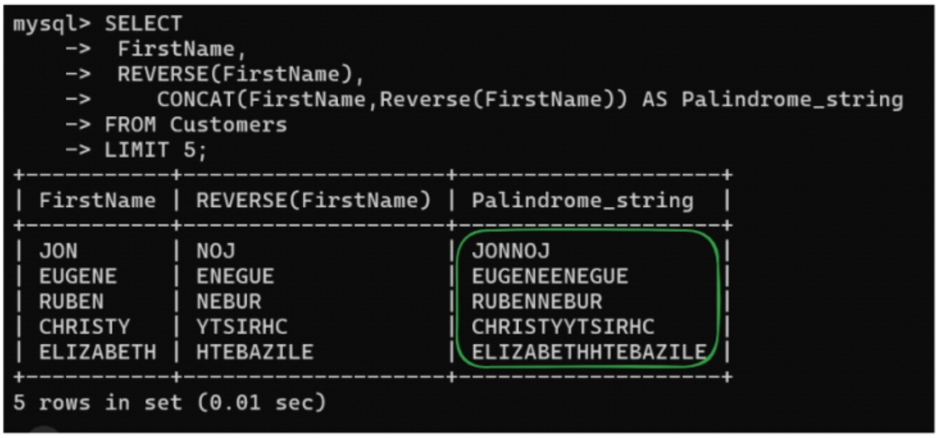


```
mysql> SELECT
    ->  FirstName,
    ->  REVERSE(FirstName)
    -> FROM Customers
    -> LIMIT 5;
+-----------+--------------------+
| FirstName | REVERSE(FirstName) |
+-----------+--------------------+
| JON       | NOJ                |
| EUGENE    | ENEGUE             |
| RUBEN     | NEBUR              |
| CHRISTY   | YTSIRHC            |
| ELIZABETH | HTEBAZILE          |
+-----------+--------------------+
5 rows in set (0.02 sec)
```



Concat + Reverse

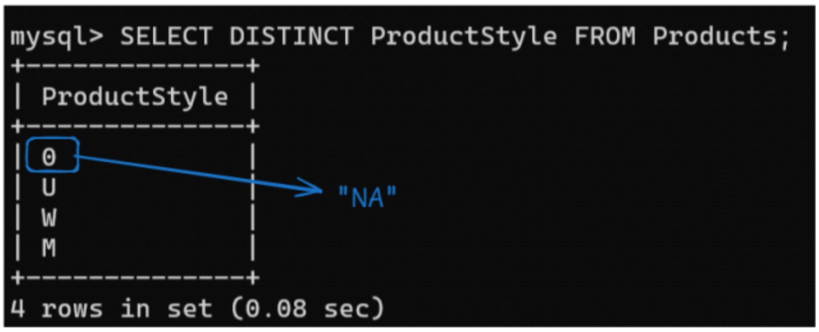
cattac



```
mysql> SELECT
    ->  FirstName,
    ->  REVERSE(FirstName),
    ->     CONCAT(FirstName,Reverse(FirstName)) AS Palindrome_string
    -> FROM Customers
    -> LIMIT 5;
+-----------+--------------------+--------------------+
| FirstName | REVERSE(FirstName) | Palindrome_string  |
+-----------+--------------------+--------------------+
| JON       | NOJ                | JONNOJ             |
| EUGENE    | ENEGUE             | EUGENEENEGUE       |
| RUBEN     | NEBUR              | RUBENNEBUR         |
| CHRISTY   | YTSIRHC            | CHRISTYYTSIRHC     |
| ELIZABETH | HTEBAZILE          | ELIZABETHHTEBAZILE |
+-----------+--------------------+--------------------+
5 rows in set (0.01 sec)
```

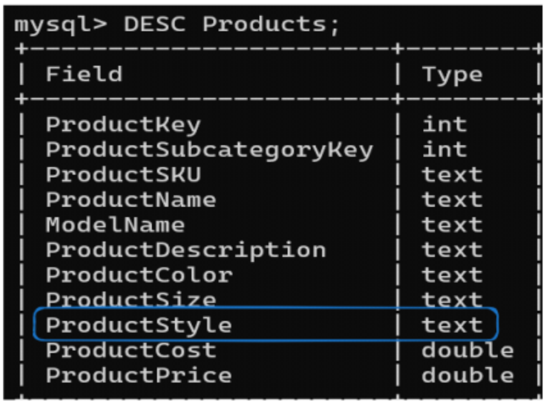
Replace()

```
Replace(column,what value you want to replace,
whom you want to replace with)
```



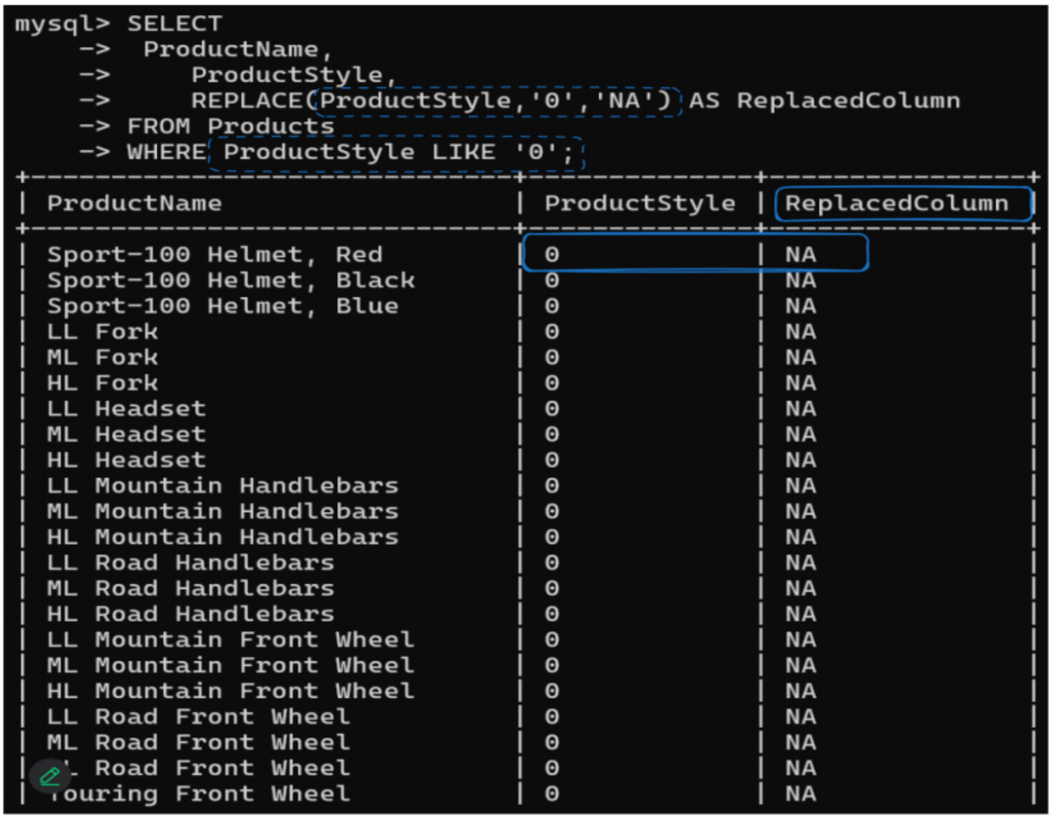
```
mysql> SELECT DISTINCT ProductStyle FROM Products;
+--------------+
| ProductStyle |
+--------------+
| 0            |
| U            |
| W            |
| M            |
+--------------+
4 rows in set (0.08 sec)
```

"NA"

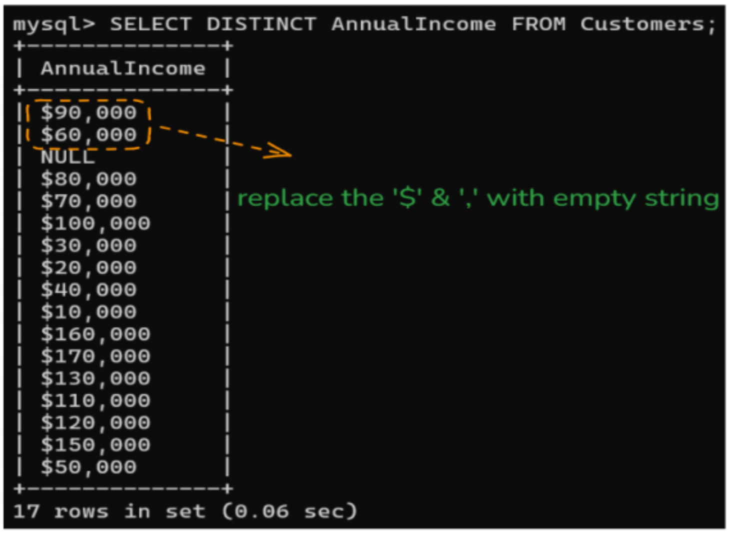


```
mysql> DESC Products;
+-------------------------+--------+
| Field                   | Type   |
+-------------------------+--------+
| ProductKey              | int    |
| ProductSubcategoryKey   | int    |
| ProductSKU              | text   |
| ProductName             | text   |
| ModelName               | text   |
| ProductDescription      | text   |
| ProductColor            | text   |
| ProductSize             | text   |
| ProductStyle            | text   |
| ProductCost             | double |
| ProductPrice            | double |
```

`'Pl0nt'   LIKE "%0%"`

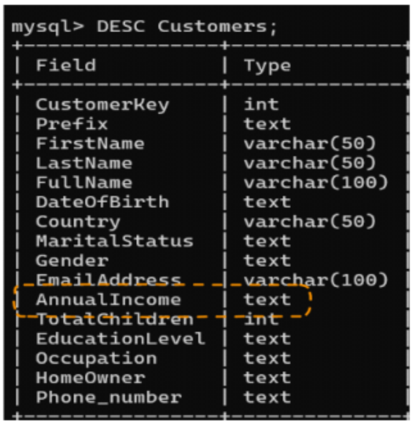


```
mysql> SELECT
    ->   ProductName,
    ->      ProductStyle,
    ->      REPLACE(ProductStyle,'0','NA') AS ReplacedColumn
    -> FROM Products
    -> WHERE ProductStyle LIKE '0';
+--------------------------------+--------------+----------------+
| ProductName                    | ProductStyle | ReplacedColumn |
+--------------------------------+--------------+----------------+
| Sport-100 Helmet, Red          | 0            | NA             |
| Sport-100 Helmet, Black        | 0            | NA             |
| Sport-100 Helmet, Blue         | 0            | NA             |
| LL Fork                        | 0            | NA             |
| ML Fork                        | 0            | NA             |
| HL Fork                        | 0            | NA             |
| LL Headset                     | 0            | NA             |
| ML Headset                     | 0            | NA             |
| HL Headset                     | 0            | NA             |
| LL Mountain Handlebars         | 0            | NA             |
| ML Mountain Handlebars         | 0            | NA             |
| HL Mountain Handlebars         | 0            | NA             |
| LL Road Handlebars             | 0            | NA             |
| ML Road Handlebars             | 0            | NA             |
| HL Road Handlebars             | 0            | NA             |
| LL Mountain Front Wheel        | 0            | NA             |
| ML Mountain Front Wheel        | 0            | NA             |
| HL Mountain Front Wheel        | 0            | NA             |
| LL Road Front Wheel            | 0            | NA             |
| ML Road Front Wheel            | 0            | NA             |
| HL Road Front Wheel            | 0            | NA             |
| Touring Front Wheel            | 0            | NA             |
```



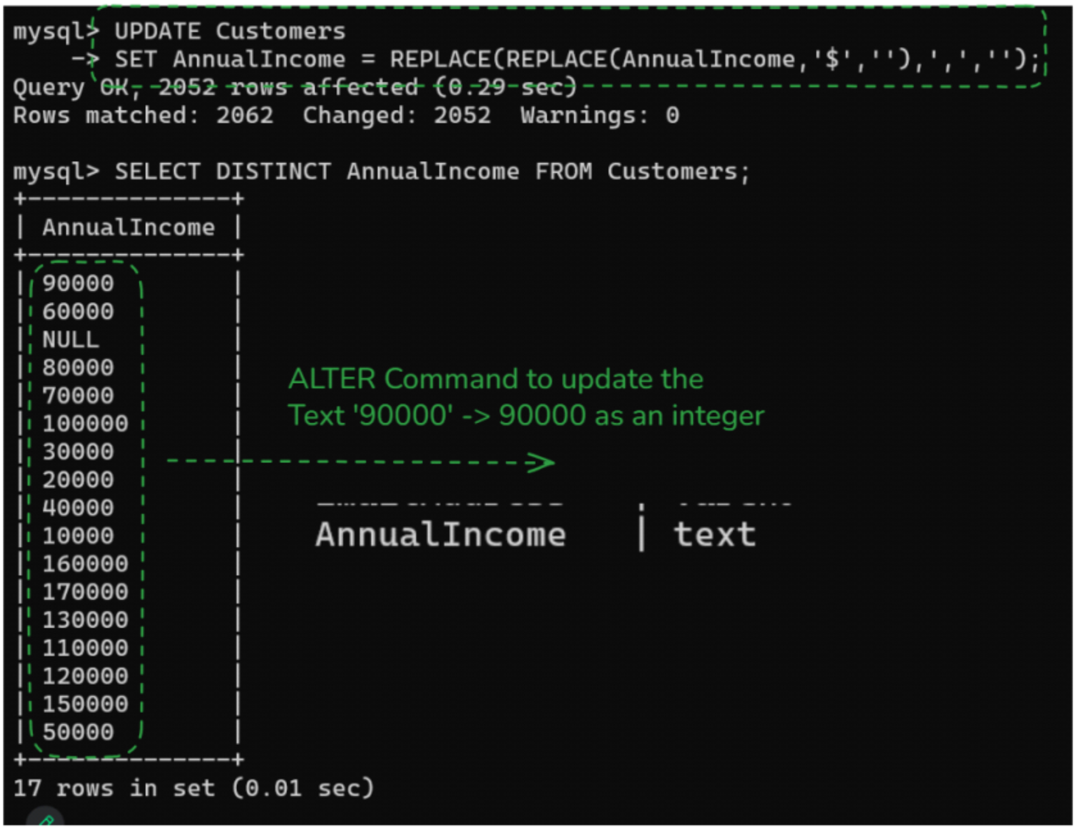
```
mysql> SELECT DISTINCT AnnualIncome FROM Customers;
+--------------+
| AnnualIncome |
+--------------+
| $90,000      |
| $60,000      |
| NULL         |
| $80,000      |
| $70,000      |
| $100,000     |
| $30,000      |
| $20,000      |
| $40,000      |
| $10,000      |
| $160,000     |
| $170,000     |
| $130,000     |
| $110,000     |
| $120,000     |
| $150,000     |
| $50,000      |
+--------------+
17 rows in set (0.06 sec)
```

replace the '$' & ',' with empty string



```
mysql> DESC Customers;
+----------------+--------------+
| Field          | Type         |
+----------------+--------------+
| CustomerKey    | int          |
| Prefix         | text         |
| FirstName      | varchar(50)  |
| LastName       | varchar(50)  |
| FullName       | varchar(100) |
| DateOfBirth    | text         |
| Country        | varchar(50)  |
| MaritalStatus  | text         |
| Gender         | text         |
| EmailAddress   | varchar(100) |
| AnnualIncome   | text         |
| TotalChildren  | int          |
| EducationLevel | text         |
| Occupation     | text         |
| HomeOwner      | text         |
| Phone_number   | text         |
+----------------+--------------+
```

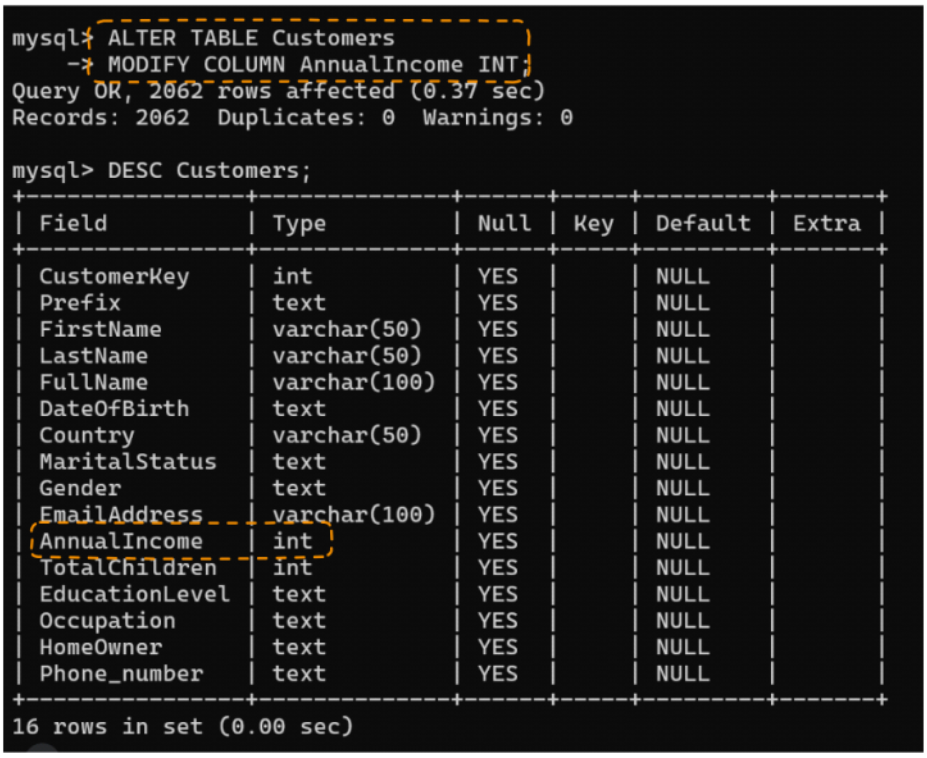
'1' -> 1 [Int]
'Abc123' -> X [can't convert it into integer]
12234 -> '12234' [Numeric values can easily casted in a string]



```
mysql> UPDATE Customers
    -> SET AnnualIncome = REPLACE(REPLACE(AnnualIncome,'$',''),',','');
Query OK, 2052 rows affected (0.29 sec)
Rows matched: 2062  Changed: 2052  Warnings: 0

mysql> SELECT DISTINCT AnnualIncome FROM Customers;
+--------------+
| AnnualIncome |
+--------------+
| 90000        |
| 60000        |
| NULL         |
| 80000        |
| 70000        |
| 100000       |
| 30000        |
| 20000        |
| 40000        |
| 10000        |
| 160000       |
| 170000       |
| 130000       |
| 110000       |
| 120000       |
| 150000       |
| 50000        |
+--------------+
17 rows in set (0.01 sec)
```

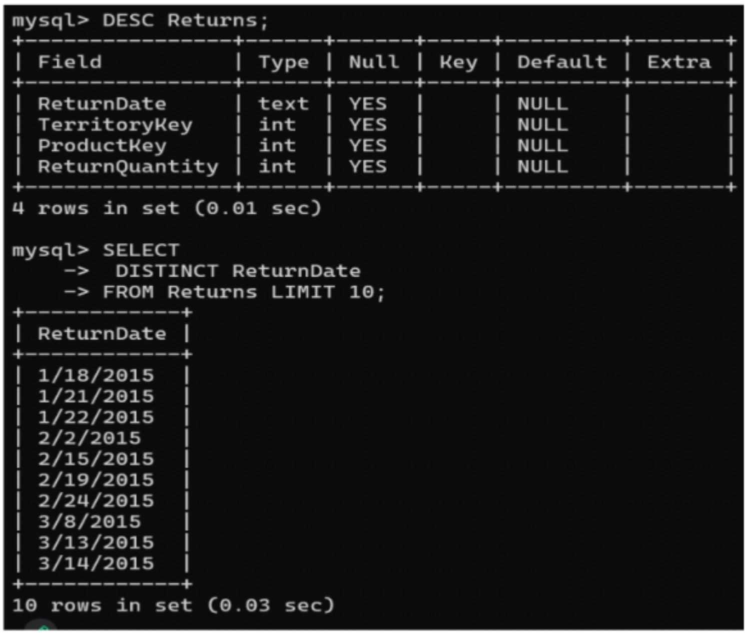
ALTER Command to update the Text '90000' -> 90000 as an integer

```
AnnualIncome     | text
```

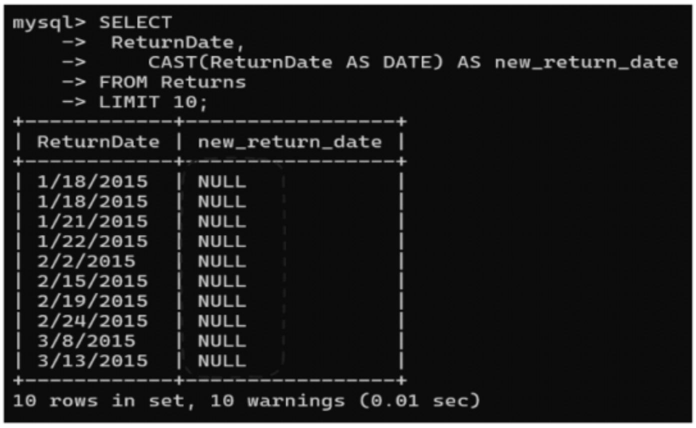


```
mysql> ALTER TABLE Customers
    -> MODIFY COLUMN AnnualIncome INT;
Query OK, 2062 rows affected (0.37 sec)
Records: 2062  Duplicates: 0  Warnings: 0

mysql> DESC Customers;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| CustomerKey     | int          | YES  |     | NULL    |       |
| Prefix          | text         | YES  |     | NULL    |       |
| FirstName       | varchar(50)  | YES  |     | NULL    |       |
| LastName        | varchar(50)  | YES  |     | NULL    |       |
| FullName        | varchar(100) | YES  |     | NULL    |       |
| DateOfBirth     | text         | YES  |     | NULL    |       |
| Country         | varchar(50)  | YES  |     | NULL    |       |
| MaritalStatus   | text         | YES  |     | NULL    |       |
| Gender          | text         | YES  |     | NULL    |       |
| EmailAddress    | varchar(100) | YES  |     | NULL    |       |
| AnnualIncome    | int          | YES  |     | NULL    |       |
| TotalChildren   | int          | YES  |     | NULL    |       |
| EducationLevel  | text         | YES  |     | NULL    |       |
| Occupation      | text         | YES  |     | NULL    |       |
| HomeOwner       | text         | YES  |     | NULL    |       |
| Phone_number    | text         | YES  |     | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
16 rows in set (0.00 sec)
```

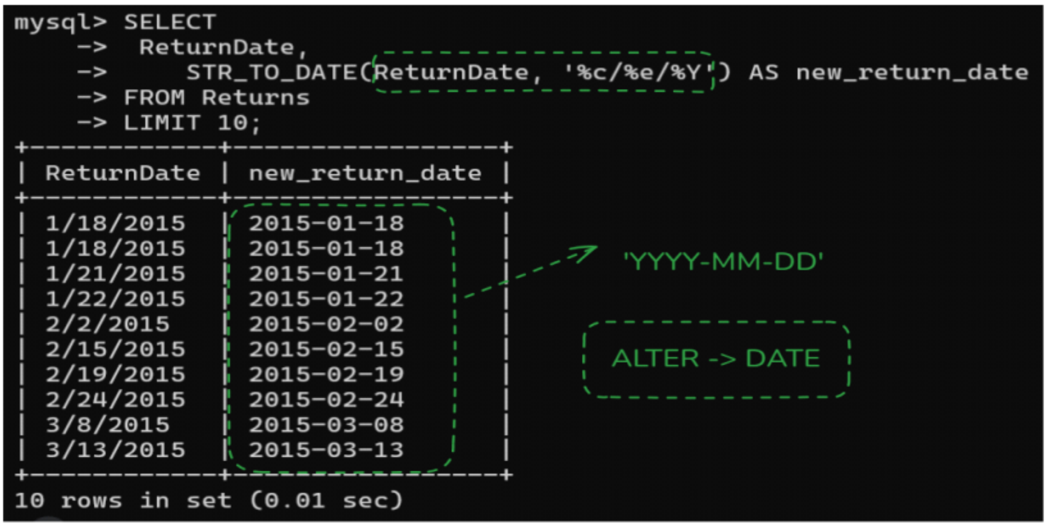
CAST() -> TEXT -> DATE



```
mysql> DESC Returns;
+----------------+------+------+-----+---------+-------+
| Field          | Type | Null | Key | Default | Extra |
+----------------+------+------+-----+---------+-------+
| ReturnDate     | text | YES  |     | NULL    |       |
| TerritoryKey   | int  | YES  |     | NULL    |       |
| ProductKey     | int  | YES  |     | NULL    |       |
| ReturnQuantity | int  | YES  |     | NULL    |       |
+----------------+------+------+-----+---------+-------+
4 rows in set (0.01 sec)

mysql> SELECT
    ->  DISTINCT ReturnDate
    -> FROM Returns LIMIT 10;
+------------+
| ReturnDate |
+------------+
| 1/18/2015  |
| 1/21/2015  |
| 1/22/2015  |
| 2/2/2015   |
| 2/15/2015  |
| 2/19/2015  |
| 2/24/2015  |
| 3/8/2015   |
| 3/13/2015  |
| 3/14/2015  |
+------------+
10 rows in set (0.03 sec)
```



```
mysql> SELECT
    ->  ReturnDate,
    ->     CAST(ReturnDate AS DATE) AS new_return_date
    -> FROM Returns
    -> LIMIT 10;
+------------+-----------------+
| ReturnDate | new_return_date |
+------------+-----------------+
| 1/18/2015  | NULL            |
| 1/18/2015  | NULL            |
| 1/21/2015  | NULL            |
| 1/22/2015  | NULL            |
| 2/2/2015   | NULL            |
| 2/15/2015  | NULL            |
| 2/19/2015  | NULL            |
| 2/24/2015  | NULL            |
| 3/8/2015   | NULL            |
| 3/13/2015  | NULL            |
+------------+-----------------+
10 rows in set, 10 warnings (0.01 sec)
```



```
mysql> SELECT
    ->  ReturnDate,
    ->     STR_TO_DATE(ReturnDate, '%c/%e/%Y') AS new_return_date
    -> FROM Returns
    -> LIMIT 10;
+------------+-----------------+
| ReturnDate | new_return_date |
+------------+-----------------+
| 1/18/2015  | 2015-01-18      |
| 1/18/2015  | 2015-01-18      |
| 1/21/2015  | 2015-01-21      |
| 1/22/2015  | 2015-01-22      |
| 2/2/2015   | 2015-02-02      |
| 2/15/2015  | 2015-02-15      |
| 2/19/2015  | 2015-02-19      |
| 2/24/2015  | 2015-02-24      |
| 3/8/2015   | 2015-03-08      |
| 3/13/2015  | 2015-03-13      |
+------------+-----------------+
10 rows in set (0.01 sec)
```

'YYYY-MM-DD'

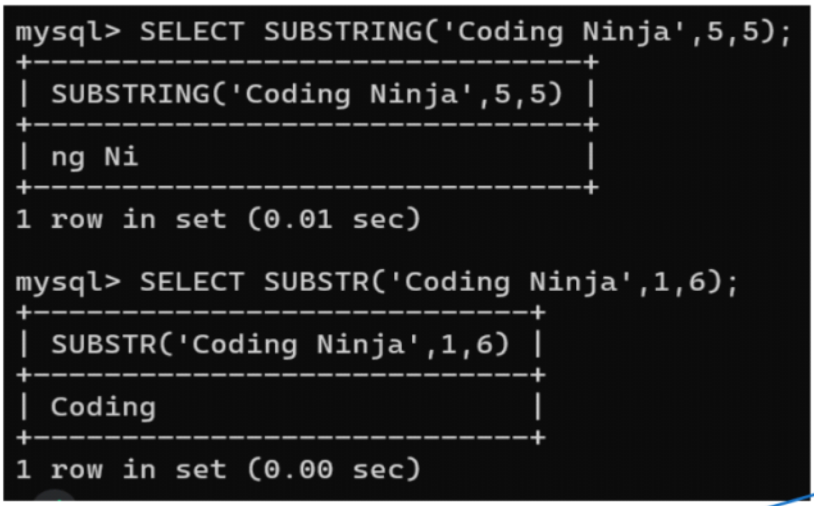
ALTER -> DATE

https://dev.mysql.com/doc/refman/8.4/en/date-and-time-functions.html#function_str-to-date

Substring()/
Substr() → Part of Substring

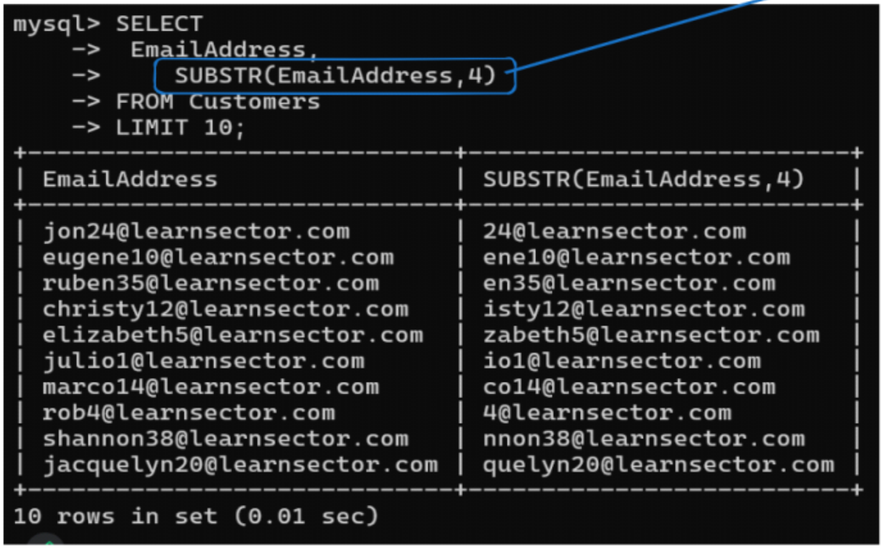
- Substring(str, starting position, length)

Index - starts with
1 [default]

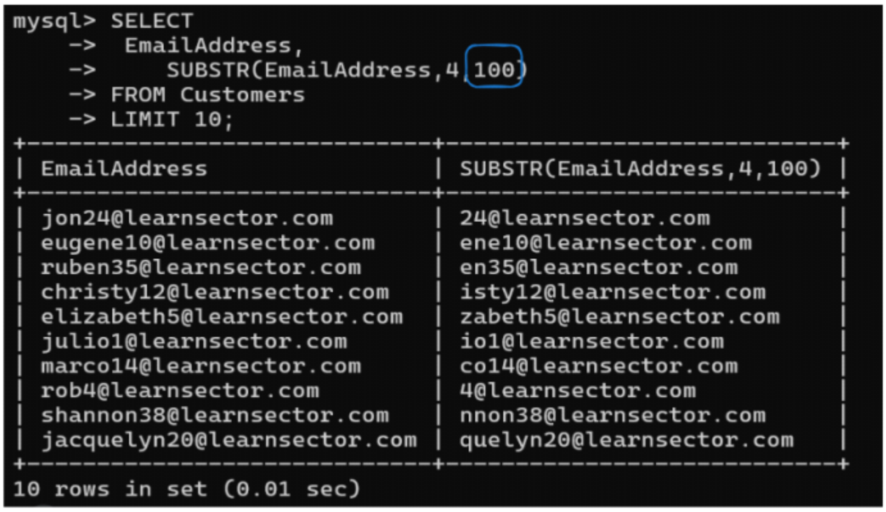


```
mysql> SELECT SUBSTRING('Coding Ninja',5,5);
+-------------------------------+
| SUBSTRING('Coding Ninja',5,5) |
+-------------------------------+
| ng Ni                         |
+-------------------------------+
1 row in set (0.01 sec)

mysql> SELECT SUBSTR('Coding Ninja',1,6);
+----------------------------+
| SUBSTR('Coding Ninja',1,6) |
+----------------------------+
| Coding                     |
+----------------------------+
1 row in set (0.00 sec)
```

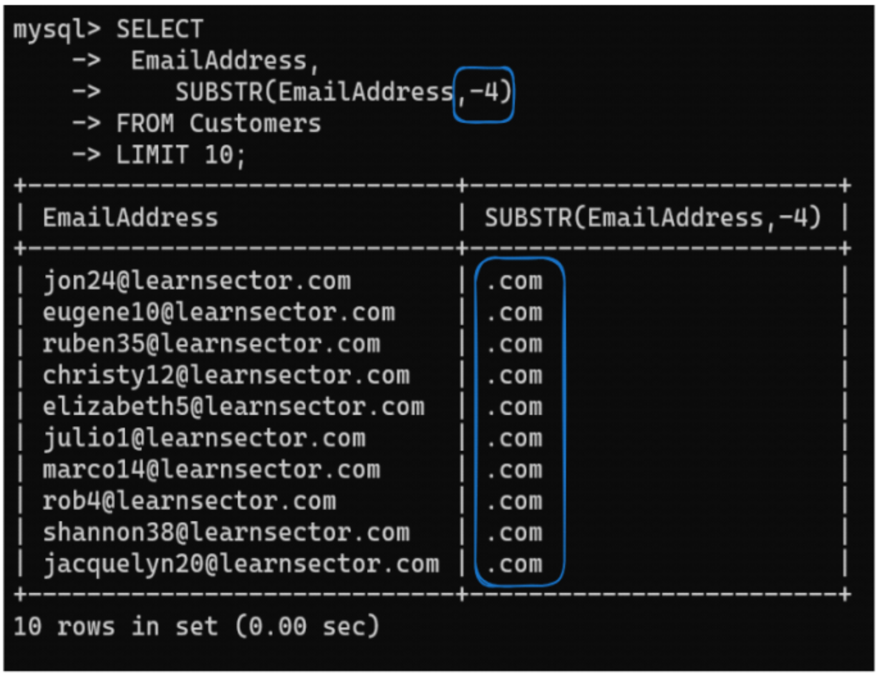
skipping the 3rd parameter,
will always extract the string
till the end.



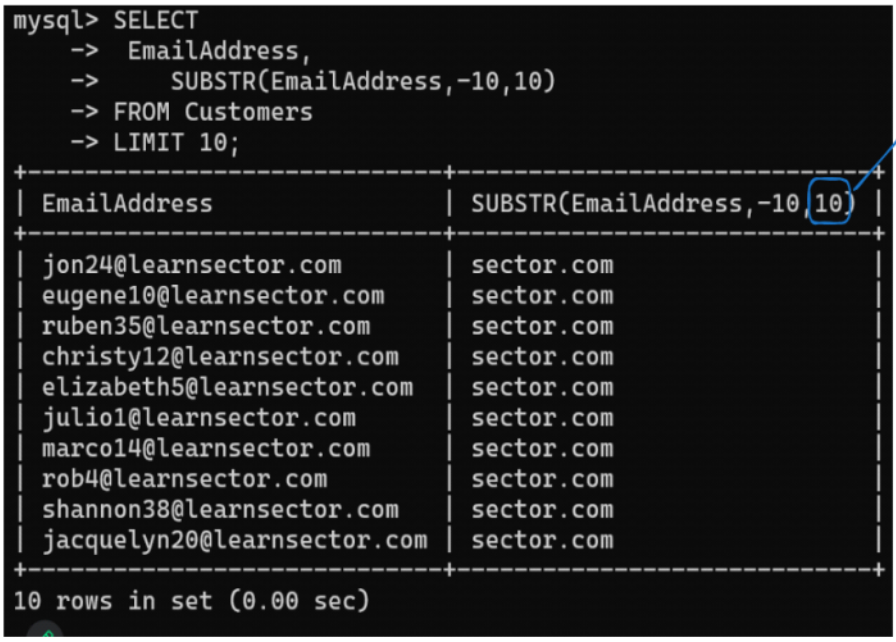
```
mysql> SELECT
    ->  EmailAddress,
    ->     SUBSTR(EmailAddress,4)
    -> FROM Customers
    -> LIMIT 10;
+------------------------------+------------------------+
| EmailAddress                 | SUBSTR(EmailAddress,4) |
+------------------------------+------------------------+
| jon24@learnsector.com        | 24@learnsector.com     |
| eugene10@learnsector.com     | ene10@learnsector.com  |
| ruben35@learnsector.com      | en35@learnsector.com   |
| christy12@learnsector.com    | isty12@learnsector.com |
| elizabeth5@learnsector.com   | zabeth5@learnsector.com |
| julio1@learnsector.com       | io1@learnsector.com    |
| marco14@learnsector.com      | co14@learnsector.com   |
| rob4@learnsector.com         | 4@learnsector.com      |
| shannon38@learnsector.com    | nnon38@learnsector.com |
| jacquelyn20@learnsector.com  | quelyn20@learnsector.com |
+------------------------------+------------------------+
10 rows in set (0.01 sec)
```



```
mysql> SELECT
    ->  EmailAddress,
    ->     SUBSTR(EmailAddress,4,100)
    -> FROM Customers
    -> LIMIT 10;
+------------------------------+----------------------------+
| EmailAddress                 | SUBSTR(EmailAddress,4,100) |
+------------------------------+----------------------------+
| jon24@learnsector.com        | 24@learnsector.com         |
| eugene10@learnsector.com     | ene10@learnsector.com      |
| ruben35@learnsector.com      | en35@learnsector.com       |
| christy12@learnsector.com    | isty12@learnsector.com     |
| elizabeth5@learnsector.com   | zabeth5@learnsector.com    |
| julio1@learnsector.com       | io1@learnsector.com        |
| marco14@learnsector.com      | co14@learnsector.com       |
| rob4@learnsector.com         | 4@learnsector.com          |
| shannon38@learnsector.com    | nnon38@learnsector.com     |
| jacquelyn20@learnsector.com  | quelyn20@learnsector.com   |
+------------------------------+----------------------------+
10 rows in set (0.01 sec)
```

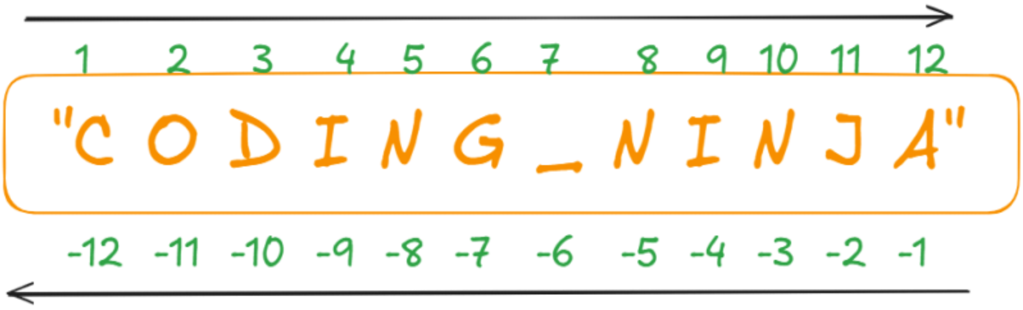


```
mysql> SELECT
    ->  EmailAddress,
    ->     SUBSTR(EmailAddress,-4)
    -> FROM Customers
    -> LIMIT 10;
+-----------------------------+-------------------------+
| EmailAddress                | SUBSTR(EmailAddress,-4) |
+-----------------------------+-------------------------+
| jon24@learnsector.com       | .com                    |
| eugene10@learnsector.com    | .com                    |
| ruben35@learnsector.com     | .com                    |
| christy12@learnsector.com   | .com                    |
| elizabeth5@learnsector.com  | .com                    |
| julio1@learnsector.com      | .com                    |
| marco14@learnsector.com     | .com                    |
| rob4@learnsector.com        | .com                    |
| shannon38@learnsector.com   | .com                    |
| jacquelyn20@learnsector.com | .com                    |
+-----------------------------+-------------------------+
10 rows in set (0.00 sec)
```

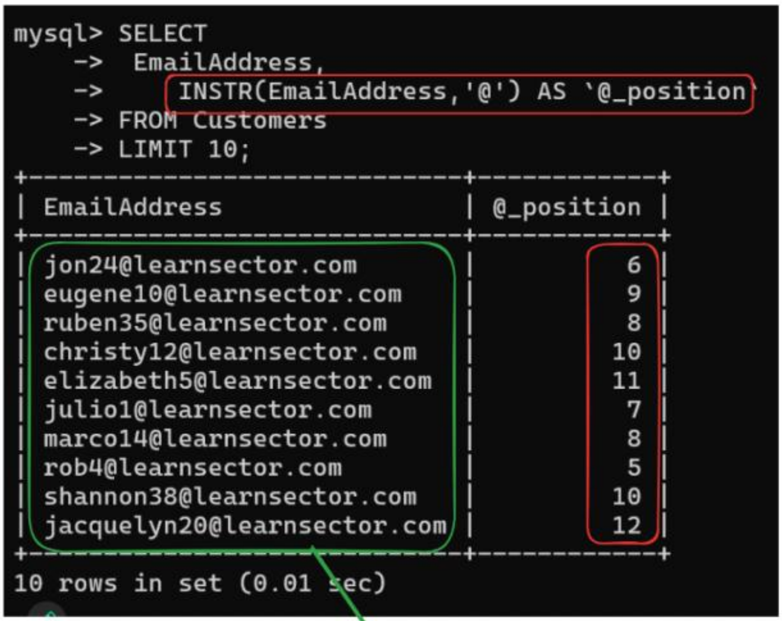


```
mysql> SELECT
    ->  EmailAddress,
    ->     SUBSTR(EmailAddress,-10,10)
    -> FROM Customers
    -> LIMIT 10;
+-----------------------------+-----------------------------+
| EmailAddress                | SUBSTR(EmailAddress,-10,10) |
+-----------------------------+-----------------------------+
| jon24@learnsector.com       | sector.com                  |
| eugene10@learnsector.com    | sector.com                  |
| ruben35@learnsector.com     | sector.com                  |
| christy12@learnsector.com   | sector.com                  |
| elizabeth5@learnsector.com  | sector.com                  |
| julio1@learnsector.com      | sector.com                  |
| marco14@learnsector.com     | sector.com                  |
| rob4@learnsector.com        | sector.com                  |
| shannon38@learnsector.com   | sector.com                  |
| jacquelyn20@learnsector.com | sector.com                  |
+-----------------------------+-----------------------------+
10 rows in set (0.00 sec)
```

length can't be -ve.



INSTR() → Find the substring index.



```
mysql> SELECT
    ->  EmailAddress,
    ->     INSTR(EmailAddress,'@') AS `@_position`
    -> FROM Customers
    -> LIMIT 10;
+------------------------------+------------+
| EmailAddress                 | @_position |
+------------------------------+------------+
| jon24@learnsector.com        |          6 |
| eugene10@learnsector.com     |          9 |
| ruben35@learnsector.com      |          8 |
| christy12@learnsector.com    |         10 |
| elizabeth5@learnsector.com   |         11 |
| julio1@learnsector.com       |          7 |
| marco14@learnsector.com      |          8 |
| rob4@learnsector.com         |          5 |
| shannon38@learnsector.com    |         10 |
| jacquelyn20@learnsector.com  |         12 |
+------------------------------+------------+
10 rows in set (0.01 sec)
```

INSTR(str,substr)

Extract the username & domain name from email address by finding the position of '@' using instr.
HINT: LEFT() & RIGHT()

18-7-1=10

Mumbai#Maharastra#India

Aggregate Functions.

Numerical Column

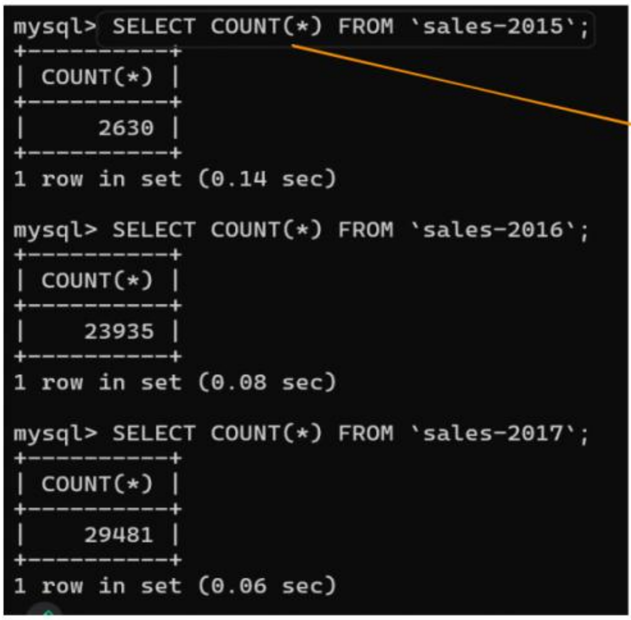
Count - To count the number of values.

Sum - To take sum of values.

Avg - To calculate the average of a column.

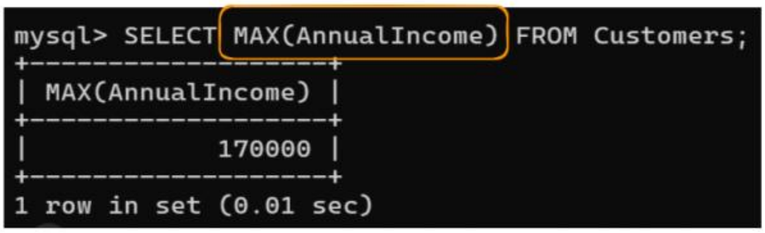
Min - Finding the minimum value by comparing all the values from a particular table.

Max - Finding the maximum value by comparing all the values from a particular table.



```
mysql> SELECT COUNT(*) FROM `sales-2015`;
+----------+
| COUNT(*) |
+----------+
|     2630 |
+----------+
1 row in set (0.14 sec)

mysql> SELECT COUNT(*) FROM `sales-2016`;
+----------+
| COUNT(*) |
+----------+
|    23935 |
+----------+
1 row in set (0.08 sec)

mysql> SELECT COUNT(*) FROM `sales-2017`;
+----------+
| COUNT(*) |
+----------+
|    29481 |
+----------+
1 row in set (0.06 sec)
```

Counting the number of records from a particular table.



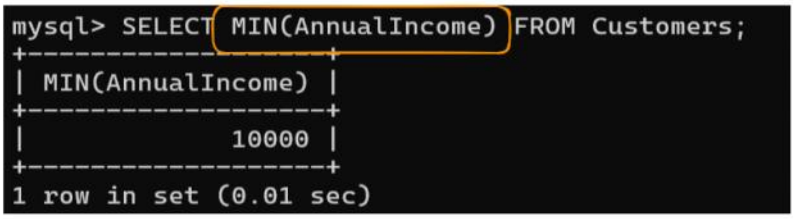
```
mysql> SELECT MAX(AnnualIncome) FROM Customers;
+-------------------+
| MAX(AnnualIncome) |
+-------------------+
|            170000 |
+-------------------+
1 row in set (0.01 sec)
```

Memory

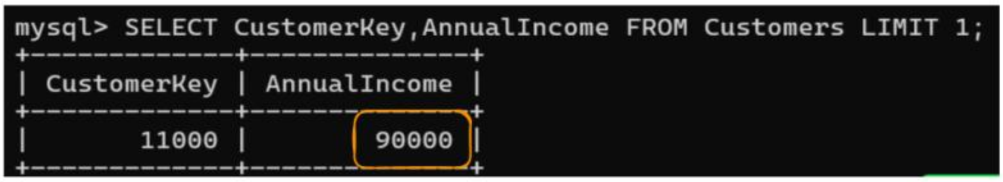
MIN - ~~90000~~ 10000
MAX - ~~90000~~ 170000

= MIN(x,y) = x, if x is smaller else y

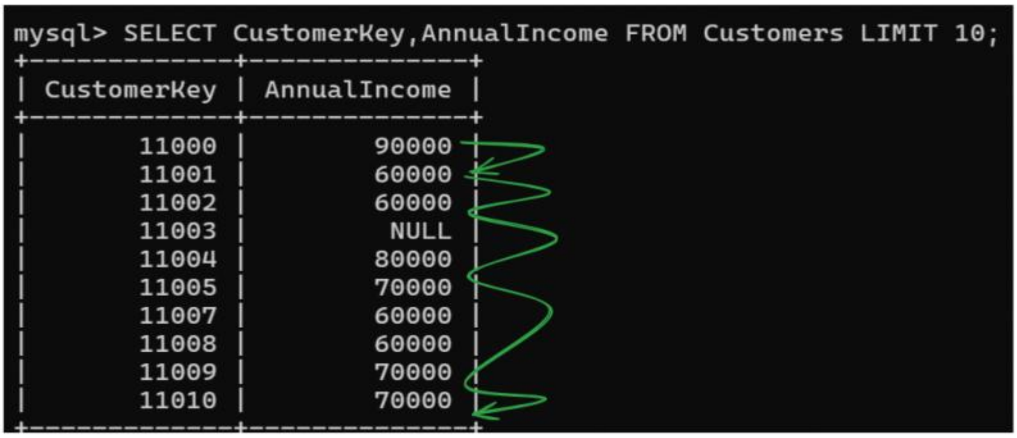
= MAX(x,y) = x, if x is larger else y



```
mysql> SELECT MIN(AnnualIncome) FROM Customers;
+-------------------+
| MIN(AnnualIncome) |
+-------------------+
|             10000 |
+-------------------+
1 row in set (0.01 sec)
```

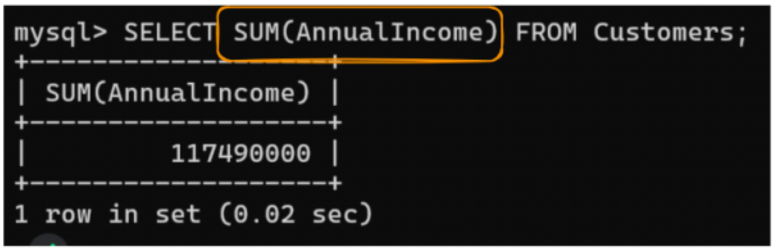


```
mysql> SELECT CustomerKey,AnnualIncome FROM Customers LIMIT 1;
+-------------+--------------+
| CustomerKey | AnnualIncome |
+-------------+--------------+
|       11000 |        90000 |
+-------------+--------------+
```

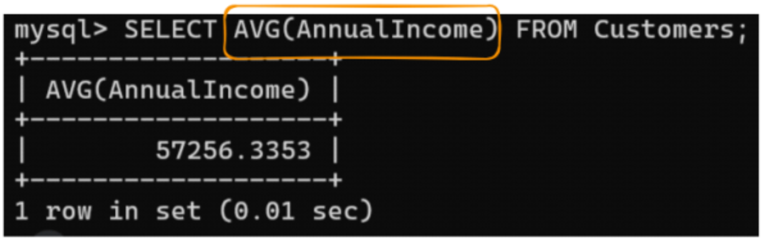


```
mysql> SELECT CustomerKey,AnnualIncome FROM Customers LIMIT 10;
+-------------+--------------+
| CustomerKey | AnnualIncome |
+-------------+--------------+
|       11000 |        90000 |
|       11001 |        60000 |
|       11002 |        60000 |
|       11003 |         NULL |
|       11004 |        80000 |
|       11005 |        70000 |
|       11007 |        60000 |
|       11008 |        60000 |
|       11009 |        70000 |
|       11010 |        70000 |
+-------------+--------------+
```

Min = 60000
Max = 90000



```
mysql> SELECT SUM(AnnualIncome) FROM Customers;
+-------------------+
| SUM(AnnualIncome) |
+-------------------+
|         117490000 |
+-------------------+
1 row in set (0.02 sec)
```



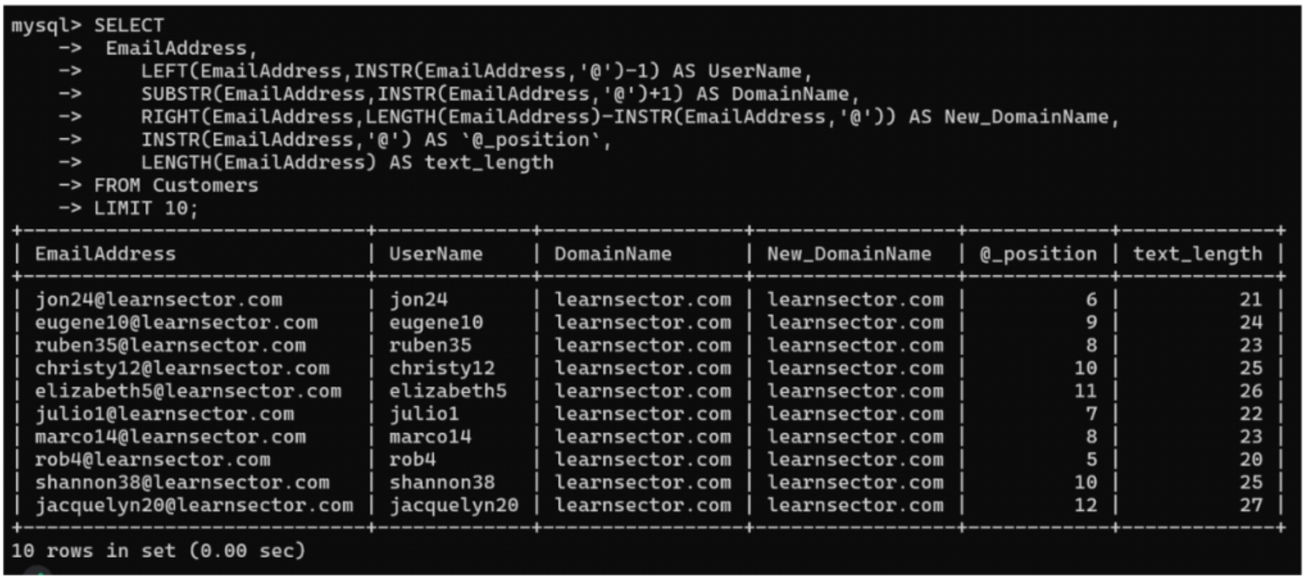
```
mysql> SELECT AVG(AnnualIncome) FROM Customers;
+-------------------+
| AVG(AnnualIncome) |
+-------------------+
|        57256.3353 |
+-------------------+
1 row in set (0.01 sec)
```

Memory

SUM = 150000 + ...

x + 0 = x [0 is the identity property]

x * 1 = x [1 is the identity property]



```
mysql> SELECT
    -> EmailAddress,
    ->    LEFT(EmailAddress,INSTR(EmailAddress,'@')-1) AS UserName,
    ->    SUBSTR(EmailAddress,INSTR(EmailAddress,'@')+1) AS DomainName,
    ->    RIGHT(EmailAddress,LENGTH(EmailAddress)-INSTR(EmailAddress,'@')) AS New_DomainName,
    ->    INSTR(EmailAddress,'@') AS `@_position`,
    ->    LENGTH(EmailAddress) AS text_length
    -> FROM Customers
    -> LIMIT 10;
+-----------------------------+-------------+-----------------+-----------------+------------+-------------+
| EmailAddress                | UserName    | DomainName      | New_DomainName  | @_position | text_length |
+-----------------------------+-------------+-----------------+-----------------+------------+-------------+
| jon24@learnsector.com       | jon24       | learnsector.com | learnsector.com |          6 |          21 |
| eugene10@learnsector.com    | eugene10    | learnsector.com | learnsector.com |          9 |          24 |
| ruben35@learnsector.com     | ruben35     | learnsector.com | learnsector.com |          8 |          23 |
| christy12@learnsector.com   | christy12   | learnsector.com | learnsector.com |         10 |          25 |
| elizabeth5@learnsector.com  | elizabeth5  | learnsector.com | learnsector.com |         11 |          26 |
| julio1@learnsector.com      | julio1      | learnsector.com | learnsector.com |          7 |          22 |
| marco14@learnsector.com     | marco14     | learnsector.com | learnsector.com |          8 |          23 |
| rob4@learnsector.com        | rob4        | learnsector.com | learnsector.com |          5 |          20 |
| shannon38@learnsector.com   | shannon38   | learnsector.com | learnsector.com |         10 |          25 |
| jacquelyn20@learnsector.com | jacquelyn20 | learnsector.com | learnsector.com |         12 |          27 |
+-----------------------------+-------------+-----------------+-----------------+------------+-------------+
10 rows in set (0.00 sec)
```