

Name: Rohan Vinayak Chaudhari

Batch: Data Engineering

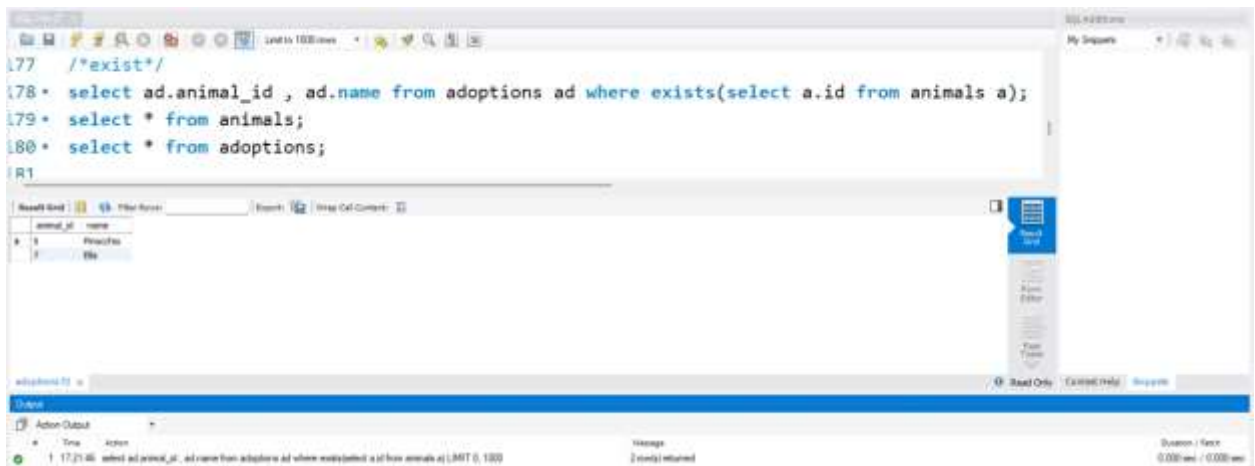
Date:22/01/2024

Topic: MYSQL(EXIST,ANY,STRING FUNCTION)

Solution:

1. MYSQL:

EXIST:



The screenshot shows a MySQL IDE interface. The top pane contains the following SQL query:

```
177 /*exist*/
178 * select ad.animal_id , ad.name from adoptions ad where exists(select a.id from animals a);
179 * select * from animals;
180 * select * from adoptions;
R1
```

The bottom pane displays the results of the query. It shows a table with two columns: 'animal_id' and 'name'. The first row contains the values '1' and 'Pussycat'.

animal_id	name
1	Pussycat

The bottom status bar indicates that the query was executed successfully, returning 2 rows and 0.000 seconds.

ANY

The screenshot shows a SQL IDE window with a query editor and a results pane. The query is as follows:

```
L82  /*any*/
L83  * SELECT a.id,a.name
L84  FROM animals a
L85  WHERE a.id = ANY
L86  (SELECT ad.animal_id
L87  FROM adoptions ad
L88  where name='Pinocchio');
```

The results pane shows a single row with the following data:

id	name
5	Cricket

The output pane shows the following message:

```
1 17:21:46 select ad.animal_id , ad.name from adoptions ad where exists(select a.id from animals a) LIMIT 0, 1000 2 row(s) returned
2 17:24:29 SELECT a.id,a.name FROM animals a WHERE a.id = ANY (SELECT ad.animal_id FROM adoptions ad where name='Pinocchi... 1 row(s) returned
```

STRING FUNCTIONS:

ASCII:

The screenshot shows a SQL IDE window with a query editor and a results pane. The query is as follows:

```
90  /*string functions*/
91
92  /*gives the ascii value*/
93  * SELECT name,ASCII(Name) AS Ascii_Code
94  FROM animals; /*RETURNS ASCII OF FIRST CHAR OF A STR*/
95
```

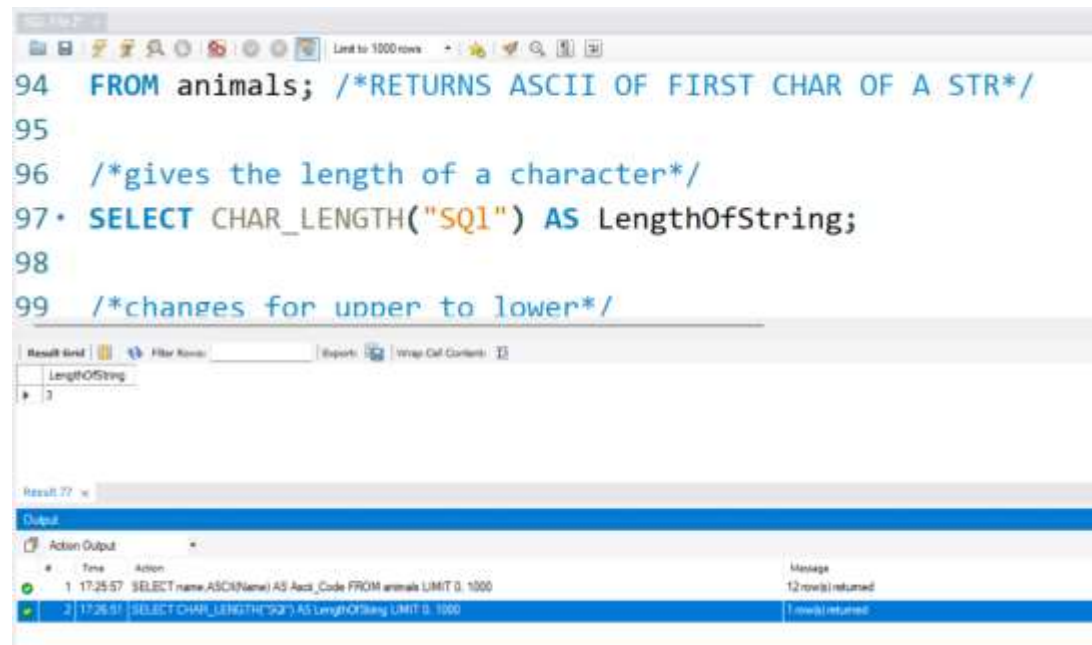
The results pane shows the following data:

name	Ascii_Code
Bella	66
Bella	66
Snowy	83
Princess	80
Cricket	67

The output pane shows the following message:

```
1 17:25:57 SELECT name,ASCII(Name) AS Ascii_Code FROM animals LIMIT 0, 1000 12 row(s) returned
```

CHAR:



The screenshot shows a SQL IDE window with a script editor and a results pane. The script editor contains the following SQL code:

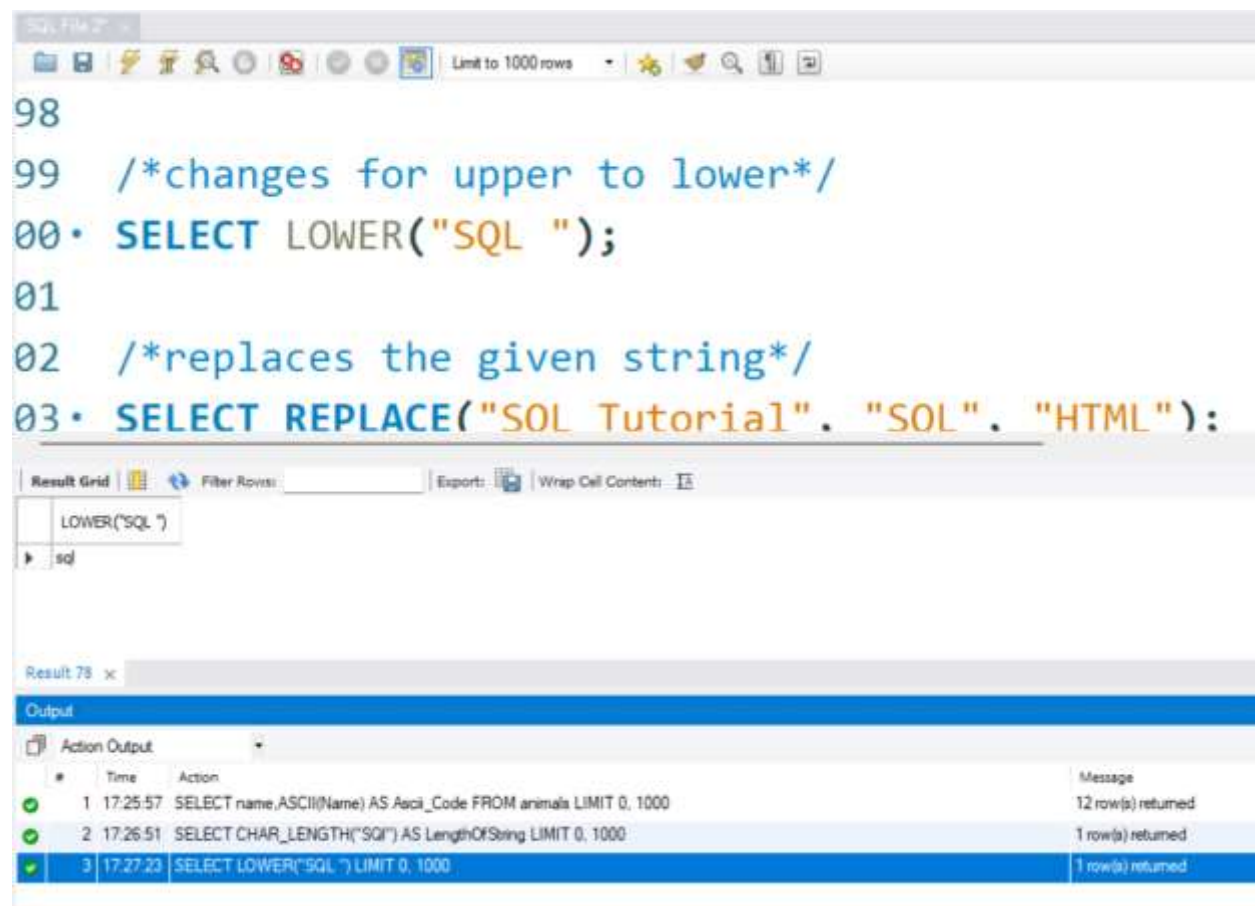
```
94 FROM animals; /*RETURNS ASCII OF FIRST CHAR OF A STR*/
95
96 /*gives the length of a character*/
97 • SELECT CHAR_LENGTH("SQL") AS LengthOfString;
98
99 /*changes for upper to lower*/
```

The results pane shows a single row with the value '3' for the column 'LengthOfString'.

Below the results pane, the 'Action Output' tab is selected, showing a log of executed actions:

#	Time	Action	Message
1	17:25:57	SELECT name,ASCII(Name) AS Ascii_Code FROM animals LIMIT 0, 1000	12 row(s) returned
2	17:26:51	SELECT CHAR_LENGTH("SQL") AS LengthOfString LIMIT 0, 1000	1 row(s) returned

LOWER:



The screenshot shows a SQL IDE window with a script editor and a results pane. The script editor contains the following SQL code:

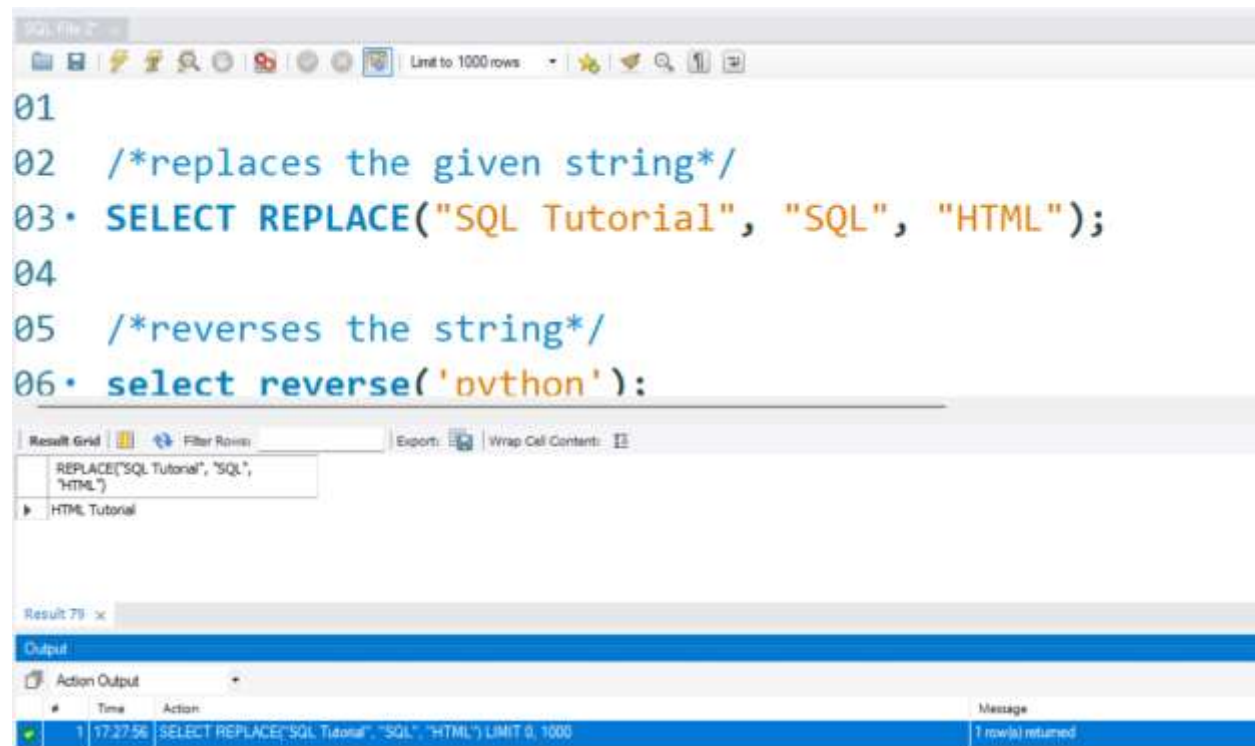
```
98
99 /*changes for upper to lower*/
00 • SELECT LOWER("SQL ");
01
02 /*replaces the given string*/
03 • SELECT REPLACE("SQL Tutorial", "SQL", "HTML"):
```

The results pane shows a single row with the value 'sql' for the column 'LOWER("SQL ")'.

Below the results pane, the 'Action Output' tab is selected, showing a log of executed actions:

#	Time	Action	Message
1	17:25:57	SELECT name,ASCII(Name) AS Ascii_Code FROM animals LIMIT 0, 1000	12 row(s) returned
2	17:26:51	SELECT CHAR_LENGTH("SQL") AS LengthOfString LIMIT 0, 1000	1 row(s) returned
3	17:27:23	SELECT LOWER("SQL ") LIMIT 0, 1000	1 row(s) returned

REPLACE:



SQL File 2

```
01
02  /*replaces the given string*/
03 • SELECT REPLACE("SQL Tutorial", "SQL", "HTML");
04
05  /*reverses the string*/
06 • select reverse('python');
```

Result Grid | Filter Rows | Export | Wrap Cell Contents

	REPLACE("SQL Tutorial", "SQL", "HTML")
▶	HTML Tutorial

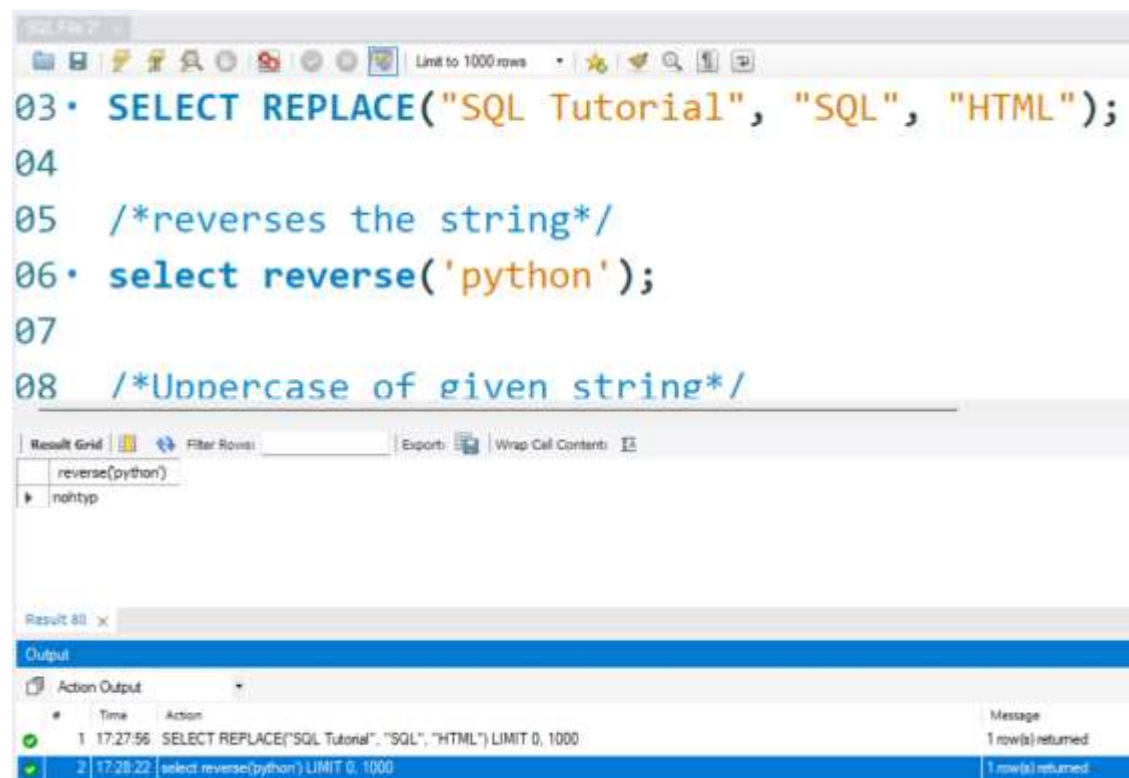
Result 79 x

Output

Action Output

#	Time	Action	Message
1	17:27:56	SELECT REPLACE("SQL Tutorial", "SQL", "HTML") LIMIT 0, 1000	1 row(s) returned

REVERSE:



SQL File 2

```
03 • SELECT REPLACE("SQL Tutorial", "SQL", "HTML");
04
05  /*reverses the string*/
06 • select reverse('python');
07
08  /*Uppercase of given string*/
```

Result Grid | Filter Rows | Export | Wrap Cell Contents

	reverse(python)
▶	nohtyp

Result 80 x

Output

Action Output

#	Time	Action	Message
1	17:27:56	SELECT REPLACE("SQL Tutorial", "SQL", "HTML") LIMIT 0, 1000	1 row(s) returned
2	17:28:22	select reverse(python) LIMIT 0, 1000	1 row(s) returned

UPPERCASE:

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
08
09  /*Uppercase of given string*/
10  select upper('Rohan');
11
12
13
```

Below the editor, the 'Result Grid' shows the query result:

upper('Rohan')
ROHAN

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message
1	17:27:56	SELECT REPLACE("SQL Tutorial", "SQL", "HTML") LIMIT 0, 1000	1 row(s) returned
2	17:28:22	select reverse(python) LIMIT 0, 1000	1 row(s) returned
3	17:29:00	select upper(Rohan) LIMIT 0, 1000	1 row(s) returned