CONCATENATION OPERATORS

concatenation operator (||), which joins two distinct strings into one string value MySQL supports CONCAT as a synonym for the SQL concatenation operator and uses the || operator for logical OR.

If you want to use (||) operator sometimes it will not support always so you need to run the below commands

```
SET @old_sql_mode = @@sql_mode;
SET @@sql_mode = PIPES_AS_CONCAT;
```

1. Write a query to display first_name, last_name of all the employees as full_name

```
SELECT
   first_name || last_name
FROM
   employee;
```

Output:

first_name || last_name
kellydavis
tomtaylor
mikewhalen
andylumb
anjelnair
ramkumar

rohansharma
johnking

Here above we have used || which is known as the **Concatenation operator** which is used to link 2 or as **many** columns as you want in your select query and it is **independent of the datatype** of the column.
Here above we have linked 2 columns i.e, **first_name+last_name** as well.

2. Write a query to display the following output:
   KELLY GETS 78000 AS SALARY
   TOM GETS 84200 AS SALARY
   Like this for all employees

```
SELECT
    first_name || ' ' || 'GETS' || ' ' || salary || ' as ' ||
'salary' as details
FROM
    employee;
```

Output:

details
kelly GETS 78000 as salary
tom GETS 84200 as salary
mike GETS 98200 as salary
andy GETS 42200 as salary
anjel GETS 42200 as salary
ram GETS 64200 as salary

rohan GETS 84200 as salary
john GETS 124200 as salary

Here whenever you want to get the value that you have written as it is on the console you need to write that inside single quotes so that it will get printed.

3. Write a query to display the following output:
   EMPLOYEE KELLY IS HIRED ON 2021-01-22
   EMPLOYEE TOM IS HIRED ON 2020-09-22
   .
   .
   Like this for all employees

```
SELECT
       'EMPLOYEE ' || first_name ||  ' IS ' || ' HIRED '
|| ' ON ' || hire_date as details
FROM
       employee;
```

Output:

details
EMPLOYEE kelly IS HIRED ON 2021-01-22
EMPLOYEE tom IS HIRED ON 2020-09-22
EMPLOYEE mike IS HIRED ON 2021-06-30
EMPLOYEE andy IS HIRED ON 2021-02-27
EMPLOYEE anjel IS HIRED ON 2019-09-26
EMPLOYEE ram IS HIRED ON 2018-12-26

EMPLOYEE rohan IS HIRED ON 2021-02-09
EMPLOYEE john IS HIRED ON 2021-02-09

4. Write a query to concatenate ram and rohan

```
SELECT
   'RAM' || 'ROHAN'
FROM
    employee;
```

Output:

‘RAM’ || ‘ROHAN’
RAMROHAN
RAMROHAN
RAMROHAN
RAMROHAN
RAMROHAN
RAMROHAN
RAMROHAN
RAMROHAN

Here you are getting the concatenated result but you are getting the values as many numbers of employees are there in the record.

If you want the concatenated result one time you need to execute the below query

```
SELECT 'RAM' || 'ROHAN';
```

Output:

‘RAM’|| ‘ROHAN’
RAMROHAN

5. Write a query to display first name, last name, email id of first two employees separated by comma

```
SELECT
       first_name || ',' || last_name || ',' || email
FROM
       employee
WHERE
       emp_id < 3;
```

Output:

first_name || last_name || email
kelly,davis,davis@gmail.com
tom,taylor,tom@gmail.com

LIMIT Keyword:
The limit keyword is used to limit the number of rows returned in a query result.

6. Write a query to display first name, last name, email id of any two employees separated by comma.

   Here whenever the restriction is to number of rows that need to be displayed in result you can make use of **LIMIT Keyword**

```
SELECT
```

```
      first_name || ',' || last_name || ',' || email as
details
FROM
      employee
LIMIT 2;
```

Output:

details
kelly,davis,davis@gmail.com
tom,taylor,tom@gmail.com

OFFSET Keyword:

The **OFFSET** value is also most often used together with the LIMIT keyword. The OFF SET value allows us to specify which row to start from retrieving data

7. Write a query to display first name, last name, email id of any the employees separated by comma from 3rd row

```
SELECT
     first_name || ',' || last_name || ',' || email as
details
FROM
     employee
LIMIT 2 OFFSET 2;
```

Output:

details
mike,whalen,mike@gmail.com
andy,lumb,andy@gmail.com

Now you are getting the data from 3 rows as the offset is 2, it will start fetching the data from the 3rd row.