

## COLUMN ALIASING

Mysql column headings in the output depends on the case that is being typed.

```
mysql> select "test";
```

```
+-----+
| test  |
+-----+
| test  |
+-----+
```

```
mysql> select "TEST";
```

```
+-----+
| TEST  |
+-----+
| TEST  |
+-----+
```

```
mysql> select "Test";
```

```
+-----+
| Test  |
+-----+
| Test  |
+-----+
```

We can change the headings by column aliasing. Following are the ways. Suppose we want to display ename, sal and additionally sal incremented by 100 to be displayed.

sal+100 "NEWSAL"

sal+100 as NEWSAL

sal+100 NEWSAL

```
mysql> SELECT ename,Sal,SAL,SAL+100,sal+100 "NEWSAL",
->sal+100 as NEWSAL,sal+100 NEWSAL FROM EMP
-> WHERE ENAME='smith';
```

```
+-----+-----+-----+-----+-----+-----+
| ename | Sal  | SAL+100 | NEWSAL | NEWSAL | NEWSAL |
+-----+-----+-----+-----+-----+-----+
| SMITH | 800  | 900    | 900    | 900    | 900    |
+-----+-----+-----+-----+-----+-----+
```

## DISTINCT

Queries sometimes produces duplicate rows which can be eliminated by using distinct.

```
mysql> select job from emp;
```

job
CLERK
SALESMAN
SALESMAN
MANAGER
SALESMAN
MANAGER
MANAGER
ANALYST
PRESIDENT
SALESMAN
CLERK
CLERK
ANALYST
CLERK

```
mysql> select distinct(job) from emp;
```

job
CLERK
SALESMAN
MANAGER
ANALYST
PRESIDENT

### **ORDER BY CLAUSE**

Data in table usually it will not be in order. We have to order by clause to sort the data.

ORDER BY has the following general characteristics:

- Sort using one or more column or expression values.
- Sort columns independently in ascending order (the default) or descending order.
- You can refer to sort columns by name or by using an alias or column number.

```
mysql> select * from emp order by ename;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7876	ADAMS	CLERK	7788	1987-07-13	1100	NULL	20

7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450	NULL	10
7902	FORD	ANALYST	7566	1981-12-03	3000	NULL	20
7900	JAMES	CLERK	7698	1981-12-03	950	NULL	30
7566	JONES	MANAGER	7839	1981-04-02	2975	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17	5000	NULL	10
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7934	MILLER	CLERK	7782	1982-01-23	1300	NULL	10
7788	SCOTT	ANALYST	7566	1987-07-13	3000	NULL	20
7369	SMITH	CLERK	7902	1980-12-17	800	NULL	20
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30

mysql> select \* from emp order by 3;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	1987-07-13	3000	NULL	20
7902	FORD	ANALYST	7566	1981-12-03	3000	NULL	20
7369	SMITH	CLERK	7902	1980-12-17	800	NULL	20
7876	ADAMS	CLERK	7788	1987-07-13	1100	NULL	20
7900	JAMES	CLERK	7698	1981-12-03	950	NULL	30
7934	MILLER	CLERK	7782	1982-01-23	1300	NULL	10
7566	JONES	MANAGER	7839	1981-04-02	2975	NULL	20
7698	BLAKE	MANAGER	7839	1981-05-01	2850	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450	NULL	10
7839	KING	PRESIDENT	NULL	1981-11-17	5000	NULL	10
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30

mysql> select ename,deptno,job from emp order by deptno desc;

ename	deptno	job
ALLEN	30	SALESMAN
WARD	30	SALESMAN
MARTIN	30	SALESMAN
BLAKE	30	MANAGER
TURNER	30	SALESMAN
JAMES	30	CLERK

	SMITH		20		CLERK	
	JONES		20		MANAGER	
	SCOTT		20		ANALYST	
	ADAMS		20		CLERK	
	FORD		20		ANALYST	
	CLARK		10		MANAGER	
	KING		10		PRESIDENT	
	MILLER		10		CLERK	
+-----+-----+-----+-----+						

## **LIMIT CLAUSE**

LIMIT clause that tells the server to return only part of a result set. It answers questions about first or last, largest or smallest, newest or oldest, least or most expensive, and so forth.

**mysql> select \* from emp;**

+-----+-----+-----+-----+-----+-----+-----+-----+								
	EMPNO		ENAME		JOB		MGR	
	HIREDATE		SAL		COMM		DEPTNO	
+-----+-----+-----+-----+-----+-----+-----+-----+								
	7369		SMITH		CLERK		7902	
	7499		ALLEN		SALESMAN		7698	
	7521		WARD		SALESMAN		7698	
	7566		JONES		MANAGER		7839	
	7654		MARTIN		SALESMAN		7698	
	7698		BLAKE		MANAGER		7839	
	7782		CLARK		MANAGER		7839	
	7788		SCOTT		ANALYST		7566	
	7839		KING		PRESIDENT		NULL	
	7844		TURNER		SALESMAN		7698	
	7876		ADAMS		CLERK		7788	
	7900		JAMES		CLERK		7698	
	7902		FORD		ANALYST		7566	
	7934		MILLER		CLERK		7782	
+-----+-----+-----+-----+-----+-----+-----+-----+								

**mysql> select \* from emp LIMIT 1;**

+-----+-----+-----+-----+-----+-----+-----+-----+								
	EMPNO		ENAME		JOB		MGR	
	HIREDATE		SAL		COMM		DEPTNO	
+-----+-----+-----+-----+-----+-----+-----+-----+								
	7369		SMITH		CLERK		7902	
+-----+-----+-----+-----+-----+-----+-----+-----+								

**--find toppest salary person**

**mysql> select \* from emp order by sal desc limit 1;**

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	NULL	1981-11-17	5000	NULL	10

**--skipping first 2 and displaying further 3 rows**

```
mysql> select * from emp limit 2,3;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02	2975	NULL	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30

**SET**

To assign a value to a variable

```
mysql> set @v1=10;
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> select * from dept where deptno=@v1;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK

```
mysql> set @v2="turner";
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> select * from emp where ename=@v2;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30

MYSQL provides lot of built in functions which can be used for transformation to meet our requirements.

SINGLE ROW FUNCTION	MULTIPLE ROW FUNCTION
It operates on a single row at a time	It operates on groups of rows.
It returns one result per row.	It returns one result for a group of rows.

#### SINGLE ROW FUNCTIONS

*CHARACTER FUNCTIONS*

*NUMERIC FUNCTIONS*

*DATE FUNCTIONS*

*CONTROL FUNCTIONS*

#### **CHARACTER FUNCTIONS**

Allow us to manipulate string values for display. Some of the manipulations are:

- Concatenating strings, adding text,
- Extracting text from a string
- Position of a string
- Changing part of string
- Transforming a string

```
mysql> select concat(ename,job) from emp where deptno=10;
```

```
+-----+
| concat (ename, job) |
+-----+
| CLARKMANAGER        |
| KINGPRESIDENT       |
| MILLERCLERK         |
```

```
+-----+
```

Let us make it fair display by adding a text to it.

```
mysql> select concat(ename,' is working as ',job) "employee and his job" from emp;
```

```
+-----+
| employee and his job |
+-----+
| SMITH is working as CLERK |
| ALLEN is working as SALESMAN |
| WARD is working as SALESMAN |
| JONES is working as MANAGER |
| MARTIN is working as SALESMAN |
| BLAKE is working as MANAGER |
| CLARK is working as MANAGER |
| SCOTT is working as ANALYST |
| KING is working as PRESIDENT |
| TURNER is working as SALESMAN |
| ADAMS is working as CLERK |
| JAMES is working as CLERK |
| FORD is working as ANALYST |
| MILLER is working as CLERK |
+-----+
```

### LENGTH

```
mysql> SELECT LENGTH('mary had a little lamb');
```

```
+-----+
| LENGTH('mary had a little lamb') |
+-----+
| 22 |
+-----+
```

### LEFT/RIGHT

Both of these functions we specify string, length of the string to be kept, relative to right or left.

```
mysql> select left('hello world',5),right('helloworld',5);
```

```
+-----+-----+
| left('hello world',5) | right('helloworld',5) |
+-----+-----+
| hello | world |
+-----+-----+
```

+-----+-----+

### SUBSTR(SUBSTRING)

It can return specific character or specific length of characters from the specified position. Can take 2 or 3 arguments.

```
mysql> select substr('helloworld',5),substr('helloworld',1,1);
```

+-----+-----+	
substring('helloworld',5)	substr('helloworld',1,1)
+-----+-----+	
oworld	h
+-----+-----+	

### INSTR

It may be required to find position of a character. (it gives only 1st occurrence). Look at the following example.

```
mysql> select instr('helloworld','l') as lposition
```

+-----+	
lposition	
+-----+	
3	
+-----+	

### UPPER(UCASE)/LOWER(LCASE)

We can transform the string to different cases.

```
mysql> select upper('happiest mind') "UCASE",lower('HAPPIEST MIND')  
"LCASE";
```

+-----+-----+	
UCASE	LCASE
+-----+-----+	
HAPPIEST MIND	happiest mind
+-----+-----+	

### LPAD/RPAD

In Microsoft Word, Excel we have left justify, right justify. LPAD/RPAD will be fulfilling that. Difference is that we have to specify padding characters.



```
mysql> select lpad(dname,15,'*') "LPAD",rpad(dname,15,'#') from dept;
```

LPAD	rpad(dname,15,'#')
*****ACCOUNTING	ACCOUNTING#####
*****RESEARCH	RESEARCH#####
*****SALES	SALES#####
*****OPERATIONS	OPERATIONS#####

## REPLACE

A portion of the string to be changed.

```
mysql> select replace('helloworld','world','globe') "replace";
```

replace
helloglobe

## REPEAT

Repeats a string as many times as specified.

```
mysql> select repeat('happy',5);
```

repeat('happy',5)
happyhappyhappyhappyhappy

## TRIM

Removes the space character OR other specified characters from the start or end of a string.

```
mysql> select trim(' RICHARDS ') "trim",  
trim('x' from 'xxxAAAxxxBBBxxx');
```

trim	trim('x' from 'xxxAAAxxxBBBxxx')
RICHARDS	AAAxxxBBB

```
+-----+-----+
```

```
mysql> select rtrim(' RICHARDS ') "rtrim", ltrim(' RICHARDS ') "ltrim";
```

```
+-----+-----+
| rtrim      | ltrim      |
+-----+-----+
|  RICHARDS  | RICHARDS   |
+-----+-----+
```

FUNCTIONS	RESULT
Upper(' hello world')	HELLOWORLD
Lower('HELLO WORLD')	helloworld
Substring('hello wold',1,5)	hello
Left('hello world',3)	Hel
right('hello world',3)	rld
Ltrim(' data ')	data
Rtrim(' data ')	data
Replace('SREE','RE','WI')	SWIE
Concat('james','bond')	jamesbond
Repeat('data',3)	datadatadata
Lpad('data',10,'*')	*****data
Rpad('data',10,'*')	data*****
Instr('mary','a')	2

## **NUMBER FUNCTIONS**

### **MOD**

Remainder can be found using mod function. I want to find out odd empno, following query can be written.

```
mysql> select empno from emp where mod(empno,2)=1;
```

```
+-----+
| empno |
+-----+
|  7839 |
|  7369 |
+-----+
```

7499
7521

### SIGN

To compare 2 numbers.

```
mysql> set @v1=10;
mysql> set @v2=20;
mysql> set @v3=20;
mysql> select sign(@v1-@v2),sign(@v2-@v3),sign(@v3-@v1);
```

sign(@v1-@v2)	sign(@v2-@v3)	sign(@v3-@v1)
-1	0	1

### ROUND/TRUNCATE/CEIL/FLOOR

```
mysql> select ceil(91.1),floor(91.9),round(100.218,2),
-> truncate(100.218,2);
```

ceil(91.1)	floor(91.9)	round(100.218,2)	truncate(100.218,2)
92	91	100.22	100.21

### ASCII/CHAR/ABS/POWER/SQRT

```
mysql> select sqrt(625),power(7,3),ascii('a'),char(65 using ascii);
```

sqrt(625)	power(7,3)	ascii('a')	char(65 using ascii)
25	343	97	A

# Date Functions

FUNCTION	DESCRIPTION
CURDATE() CURRENT_DATE()	SHOWS CURRENT DATE WITHOUT TIME
SYSDATE() NOW()	SHOWS CURRENT DATE WITH TIME
DATEDIFF()	NUMBER OF DAYS BETWEEN 2 DATES
ADDDATE(),DATE_ADD()	ADDS DAYS,MONTHS,YEARS
SUBDATE(),DATE_SUB()	DEDUCTS DAYS,MONTHS,YEARS
LAST_DAY()	LAST DAY OF MONTH SPECIFIED
DAYNAME()	SHOWS DAYS NAME
MONTHNAME	SHOWS MONTH NAME
DATE()	SHOWS ONLY THE DATE PORTION
TIME()	SHOWS ONLY THE TIME PORTION

## GETTING CURRENT DATE AND TIME

```
mysql> select current_date,curdate();
```

```
+-----+-----+
| current_date | curdate() |
+-----+-----+
| 2022-08-12   | 2022-08-12 |
+-----+-----+
```

```
mysql> select sysdate(),now();
```

```
+-----+-----+
| sysdate()      | now()      |
+-----+-----+
| 2022-08-12 20:14:14 | 2022-08-12 20:14:14 |
+-----+-----+
```

```
mysql> select date(sysdate()),time(sysdate()),date(now()),time(now());
```

```
+-----+-----+-----+-----+
| date(sysdate()) | time(sysdate()) | date(now()) | time(now()) |
+-----+-----+-----+-----+
```

2022-08-12	20:17:28	2022-08-12	20:17:28
------------	----------	------------	----------

## **CHANGING DATE VALUES**

```
mysql> select date_add(curdate(),interval '1' day) "add 1 day",
date_add(curdate(),interval '1' month) "add 1 month",
date_add(curdate(),interval '1' year) "add 1 year";
```

add 1 day	add 1 month	add 1 year
2022-08-13	2022-09-12	2023-08-12

```
mysql> select date_sub(curdate(),interval '1' day) "deduct 1 day",
date_sub(curdate(),interval '1' month) "deduct 1 month",
date_sub(curdate(),interval '1' year) "deduct 1 year";
```

deduct 1 day	deduct 1 month	deduct 1 year
2022-08-11	2022-07-12	2021-08-12

```
mysql> select last_day(curdate());
```

last_day(curdate())
2022-08-31

## **DIFFERENCE BETWEEN DATES**

```
mysql> select timestampdiff(month,'2020-08-12',curdate());
```

timestampdiff(month, '2020-08-12', curdate())
24

```
mysql> select timestampdiff(year,'2020-08-12',curdate());
```

```
+-----+
| timestampdiff(year, '2020-08-12', curdate()) |
+-----+
|                                             2 |
+-----+
```

```
mysql> select timestampdiff(day,'2020-08-12',curdate());
```

```
+-----+
| timestampdiff(day, '2020-08-12', curdate()) |
+-----+
|                                             730 |
+-----+
```

```
mysql> SELECT DATEDIFF(curdate(),'2020-08-12');
```

```
+-----+
| DATEDIFF (curdate() , '2020-08-12') |
+-----+
|                                     730 |
+-----+
```

## **RETURNING DATE TIME PARTS**

DATE\_FORMAT()

This function allows to format date by specifying a sequence of format strings. Format should be prefixed with % enclosed inside quote ('%format'). Some options are shown below.

# PARAMETERS WITH DATE\_FORMAT

FORMAT	DESCRIPTION	EXAMPLE
%b	Number of month	12
%m	Three letter abbreviation	DEC
%M	Month fully spelt out	DECEMBER
%j	Number of days since Jan 1	347
%d	Number of the day of month	13
%D	DAY NUMBER WITH th,rd etc	13TH
%a	Three-letter abbreviation of day	WED
%W	Day fully spelt out	WEDNESDAY
%Y	4 digit year	1995
%y	2 digit year	95

```
mysql> select dayname(sysdate()),monthname(sysdate()),year(sysdate());
```

```
+-----+-----+-----+
| dayname(sysdate()) | monthname(sysdate()) | year(sysdate()) |
+-----+-----+-----+
| Friday             | August               | 2022             |
+-----+-----+-----+
```

## CONTROL FUNCTIONS

# CASE

## IF

## IFNULL

## NULLIF

### CASE

There are two versions of using CASE statement:

#### **Version 1:**

CASE value WHEN [compare\_value] THEN result [WHEN [compare\_value] THEN result ...] [ELSE result] END

It has been decided to pay bonus based upon jobs.

For clerk job 1.5 times the salary, analyst 1.75 times the salary, salesman 2.0 times the salary and others only salary as bonus. Let us use CASE to arrive at the bonus.

```
mysql> select ename,job,sal,  
-> (case when job='clerk' then 1.5*sal  
->    when job='analyst' then 1.75*sal  
->    when job='salesman' then 2.0*sal  
->    else  
->    sal  
->    end) "bonus"  
-> from emp  
-> order by 2;
```

ename	job	sal	bonus
SCOTT	ANALYST	3000	5250
FORD	ANALYST	3000	5250
SMITH	CLERK	800	1200
ADAMS	CLERK	1100	1650



JAMES	CLERK	950	1425
MILLER	CLERK	1300	1950
JONES	MANAGER	2975	2975
BLAKE	MANAGER	2850	2850
CLARK	MANAGER	2450	2450
KING	PRESIDENT	5000	5000
ALLEN	SALESMAN	1600	3200
WARD	SALESMAN	1250	2500
MARTIN	SALESMAN	1250	2500
TURNER	SALESMAN	1500	3000

### Version 2:

CASE WHEN [condition] THEN result [WHEN [condition] THEN result ...] [ELSE result]  
END

```
mysql> select ename,job,sal,
-> (case job when 'clerk' then 1.5*sal
->      when 'analyst' then 1.75*sal
->      when 'salesman' then 2.0*sal
->      else
->      sal
->      end) "bonus"
-> from emp
-> order by 2;
```

ename	job	sal	bonus
SCOTT	ANALYST	3000	5250
FORD	ANALYST	3000	5250
SMITH	CLERK	800	1200
ADAMS	CLERK	1100	1650
JAMES	CLERK	950	1425
MILLER	CLERK	1300	1950
JONES	MANAGER	2975	2975
BLAKE	MANAGER	2850	2850
CLARK	MANAGER	2450	2450
KING	PRESIDENT	5000	5000
ALLEN	SALESMAN	1600	3200
WARD	SALESMAN	1250	2500
MARTIN	SALESMAN	1250	2500
TURNER	SALESMAN	1500	3000

### **IF(expr1,expr2,expr3)**

IF function accepts three arguments and the result is returned based on if expr1 is TRUE.

If expr1 is evaluated to TRUE, the function returns expr2. Otherwise, expr3 is returned.

```
mysql> select ename,sal,if(sal>3000,'high','low') as comments from emp;
```

ename	sal	comments
SMITH	800	low
ALLEN	1600	low
WARD	1250	low
JONES	2975	low
MARTIN	1250	low
BLAKE	2850	low
CLARK	2450	low
SCOTT	3000	low
KING	5000	high
TURNER	1500	low
ADAMS	1100	low
JAMES	950	low
FORD	3000	low
MILLER	1300	low

### **IFNULL**

#### **IFNULL(expr1,expr2)**

If expr1 is not NULL, the function returns expr1. Otherwise it returns expr2.

```
mysql> select ename,sal,comm,sal+comm,ifnull((sal+comm),sal) "ifnull" from emp;
```

ename	sal	comm	sal+comm	ifnull
SMITH	800	NULL	NULL	800
ALLEN	1600	300	1900	1900
WARD	1250	500	1750	1750

JONES	2975	NULL	NULL	2975
MARTIN	1250	1400	2650	2650
BLAKE	2850	NULL	NULL	2850
CLARK	2450	NULL	NULL	2450
SCOTT	3000	NULL	NULL	3000
KING	5000	NULL	NULL	5000
TURNER	1500	0	1500	1500
ADAMS	1100	NULL	NULL	1100
JAMES	950	NULL	NULL	950
FORD	3000	NULL	NULL	3000
MILLER	1300	NULL	NULL	1300

### **NULLIF**

**NULLIF(expr1, expr2)**

Returns NULL if expr1 = expr2 is true, otherwise returns expr1.

**mysql> SELECT ename,sal,length(sal),length(ename),  
>NULLIF(LENGTH(ENAME), LENGTH(SAL)) "nullif"  
-> from emp;**

ename	sal	length(sal)	length(ename)	nullif
SMITH	800	3	5	5
ALLEN	1600	4	5	5
WARD	1250	4	4	NULL
JONES	2975	4	5	5
MARTIN	1250	4	6	6
BLAKE	2850	4	5	5
CLARK	2450	4	5	5
SCOTT	3000	4	5	5
KING	5000	4	4	NULL
TURNER	1500	4	6	6
ADAMS	1100	4	5	5
JAMES	950	3	5	5
FORD	3000	4	4	NULL
MILLER	1300	4	6	6

