Databases with *MYSQL*

8. *SQL* Functions *Querys*

Dante – Digital Area for Networking Teachers and Educators

Funded by the Erasmus+ Programme of the European Union

Learning Outcomes

After this lesson, the learner will be able to use all the following *SQL* Function query commands to retrieve data from database tables:

- CONCAT
- LOWER
- UPPER
- LPAD
- RPAD
- SUBSTR
- INSTR
- LTRIM
- RTRIM
- LENGTH
- REPLACE
- Math operators
- ROUND
- CEIL
- FLOOR
- SIGN
- MOD
- DATEFORMAT
- CURDATE
- DATEDIF
- ADDTIME
- IFNULL

Character Manipulation Functions

CONCAT (*col*|*value*)

The concatenation function lets you connect columns to other columns, arithmetic expressions, or constant values to create a character expression.

To combine the EMPLOYEE name and job:

```
SELECT CONCAT(ENAME , JOB) FROM EMP;
```

CONCAT(ENAME , JOB)
SMITHCLERK
ALLENSALESMAN
WARDSALESMAN
JONESMANAGER
MARTINSALESMAN
BLAKEMANAGER
CLARKMANAGER
SCOTTANALYST
KINGPRESIDENT
TURNERSALESMAN
ADAMSCLERK
JAMESCLERK
FORDANALYST
MILLERCLERK

Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions

CONCAT (*col*|*value*)

To combine the EMPLOYEE name and job with a space between strings:

```
SELECT CONCAT( ENAME ," ", JOB) FROM EMP;
```

	CONCAT(ENAME ," ", JOB)
▶	SMITH CLERK
	ALLEN SALESMAN
	WARD SALESMAN
	JONES MANAGER
	MARTIN SALESMAN
	BLAKE MANAGER
	CLARK MANAGER
	SCOTT ANALYST
	KING PRESIDENT
	TURNER SALESMAN
	ADAMS CLERK
	JAMES CLERK
	FORD ANALYST
	MILLER CLERK

Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions

LOWER (*col*|*value*)

Force alphanumeric values of type character that are in uppercase to lowercase.

```
SELECT LOWER(ENAME) FROM EMP;
```

LOWER(ENAME)
smith
allen
ward
jones
martin
blake
clark
scott
king
turner
adams
james
ford
miller

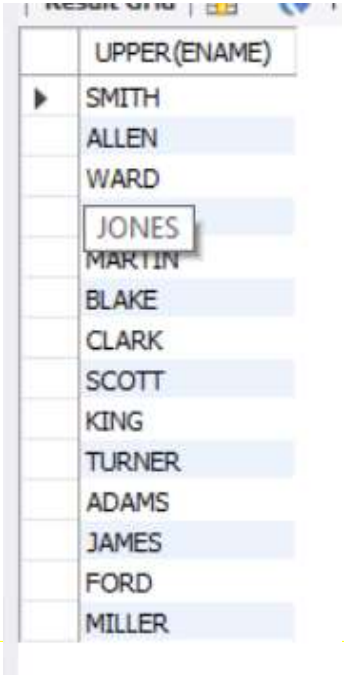
Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions

UPPER(*col*|*value*)

Force alphanumeric values of type char or varchar that are in lowercase to uppercase.

```
SELECT UPPER(ENAME) FROM EMP;
```



UPPER(ENAME)
SMITH
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
KING
TURNER
ADAMS
JAMES
FORD
MILLER

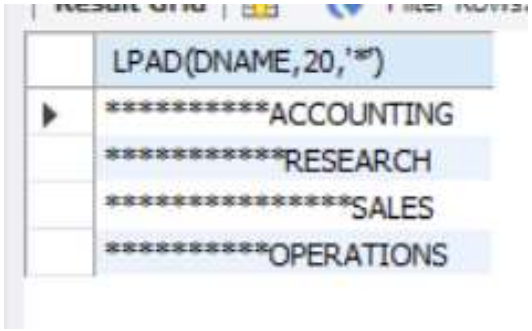
Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions

LPAD(*col|value,n,'string'*)

Fills the column or literal value from the left, up to a total of n character positions. If the value is omitted, it is padded with spaces.

```
SELECT LPAD(DNAME,20,'*') FROM DEPT;
```



	LPAD(DNAME,20,'*')
▶	**********ACCOUNTING
	**********RESEARCH
	**********SALES
	**********OPERATIONS

```
SELECT LPAD(DNAME,20,'   ') FROM DEPT;
```

Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions

RPAD(*col|value,n,'string'*)

Fills the column or literal value from the right, up to a total of n character positions. If the value is omitted, it is padded with spaces.

```
SELECT RPAD(DNAME,20,'*') FROM DEPT;
```

RPAD(DNAME,20,'*')
ACCOUNTING**********
RESEARCH **********
SALES**********
OPERATIONS**********

```
SELECT RPAD(DNAME,20,' ') FROM DEPT;
```

RPAD(DNAME,20,' ')
ACCOUNTING
RESEARCH
SALES
OPERATIONS

Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions

SUBSTR(*col|value*, *pos*,*n*)

Fills the column or literal value from the right, up to a total of n character positions. If the value is omitted, it is padded with spaces.

```
SELECT DNAME, SUBSTR(DNAME,2,3) FROM DEPT;
```

	DNAME	SUBSTR(DNAME,2,3)
▶	ACCOUNTING	CCO
	RESEARCH	ESE
	SALES	ALE
	OPERATIONS	PER

```
SELECT DNAME, SUBSTR(DNAME,2) FROM DEPT;
```

	DNAME	SUBSTR(DNAME,2)
▶	ACCOUNTING	CCOUNTING
	RESEARCH	ESEARCH
	SALES	ALES
	OPERATIONS	PERATIONS

Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions

INSTR(*col*|*value*, '*string*')

Finds the position of the first occurrence of 'string'.

```
SELECT DNAME, INSTR(DNAME,'A') FROM DEPT;
```

DNAME	INSTR(DNAME,'A')
ACCOUNTING	1
RESEARCH	5
SALES	2
OPERATIONS	5

Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions

LTRIM(*col|value, 'characters'*)

Removes occurrences of spaces that precede the left of other characters.

```
SELECT '     ACCOUNTING' , LTRIM ('     ACCOUNTING');
```

	ACCOUNTING	LTRIM (' ACCOUNTING')
▶	ACCOUNTING	ACCOUNTING

Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions

RTRIM(*col|value, 'characters'*)

Strips occurrences of trailing spaces from other characters.

```
SELECT 'ACCOUNTING      ' , RTRIM
('ACCOUNTING    ');
```

	ACCOUNTING	RTRIM ('ACCOUNTING ')
▶	ACCOUNTING	ACCOUNTING

Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions
LENGTH(*col*|*value*)

Outputs the number of characters (or digits) in the column or literal value.

```
SELECT DNAME, LENGTH(DNAME) "STRING SIZE" FROM DEPT;
```

	DNAME	STRING SIZE
▶	ACCOUNTING	10
	RESEARCH	9
	SALES	5
	OPERATIONS	10

Funded by the Erasmus+ Programme of the European Union

Character Manipulation Functions
REPLACE(*col*|*value*, *from*, *to*)

Replaces the string from (from) with the string from (to). Replacement will be performed on all occurrences.

```
SELECT ENAME, REPLACE(ENAME, 'ER','XY')    FROM EMP;
```

ENAME	REPLACE(ENAME, 'ER','XY')
SMITH	SMITH
ALLEN	ALLEN
WARD	WARD
JONES	JONES
MARTIN	MARTIN
BLAKE	BLAKE
CLARK	CLARK
SCOTT	SCOTT
KING	KING
TURNER	TURNXY
ADAMS	ADAMS
JAMES	JAMES
FORD	FORD
MILLER	MILLXY

Funded by the Erasmus+ Programme of the European Union

Math Operators

+	Addition
-	Subtrction
*	Multiplication
/	Division
**	Exponentiation

Funded by the Erasmus+ Programme of the European Union

Math Functions

ROUND(*col*|*value*, *n*)

Rounds to n decimal places. If n is omitted, round to units. If n is negative, digits to the left of the decimal point are rounded.

```
SELECT SAL*1.035, ROUND(SAL*1.035,1), ROUND(SAL*1.035,0) FROM EMP;
```

	SAL*1.035	ROUND(SAL*1.035,1)	ROUND(SAL*1.035,0)
▶	828.00000	828.0	828
	1656.00000	1656.0	1656
	1293.75000	1293.8	1294
	3079.12500	3079.1	3079
	1293.7500	1293.75000	1294
	2949.75000	2949.8	2950
	2535.75000	2535.8	2536
	3105.00000	3105.0	3105
	5175.00000	5175.0	5175
	1552.50000	1552.5	1553
	1138.50000	1138.5	1139
	983.25000	983.3	983
	3105.00000	3105.0	3105
	1345.50000	1345.5	1346

Funded by the Erasmus+ Programme of the European Union

Math Functions

CEIL(*col*|*value*)

Determines the smallest integer greater than or equal to the column value, expression, or value.

```
SELECT 1000-SAL*1.035, CEIL(1000-SAL*1.035) FROM EMP;
```

1000-SAL*1.035	CEIL(1000-SAL*1.035)
172.00000	172
-656.00000	-656
-293.75000	-293
-2079.12500	-2079
-293.75000	-293
-1949.75000	-1949
-1535.75000	
-2105.00000	
-4175.00000	-4175
-552.50000	-552
-138.50000	-138
16.75000	17
-2105.00000	-2105
-345.50000	-345

Funded by the Erasmus+ Programme of the European Union

Math Functions

FLOOR(*col*|*value*)

Determines the largest integer less than or equal to the column value, expression, or value.

```
SELECT 1000-SAL*1.035, FLOOR(1000-SAL*1.035) FROM EMP;
```

	1000-SAL*1.035	FLOOR(1000-SAL*1.035)
▶	172.00000	172
	-656.00000	-656
	-293.75000	-294
	-2079.12500	-2080
	-293.75000	-294
	-1949.75000	-1950
	-1535.75000	-1536
	-2105.00000	-2105
	-4175.00000	-4175
	-552.50000	-553
	-138.50000	-139
	16.75000	16
	-2105.00000	-2105
	-345.50000	-346

Funded by the Erasmus+ Programme of the European Union

Math Functions

SIGN(*col*|*value*)

Returns –1 if the column, value, or expression is negative, 0 if it is zero, and +1 if it is positive.

(Used to compare two values Ex. SIGN(x-y)).

```
SELECT 1000-SAL, SIGN(1000-SAL)
FROM EMP;
```

1000-SAL	SIGN(1000-SAL)
200.00	1
-600.00	-1
-250.00	-1
-197 -1975.00	
-250.00	-1
-1850.00	-1
-1450.00	-1
-2000.00	-1
-4000.00	-1
-500.00	-1
-100.00	-1
50.00	1
-2000.00	-1
-300.00	-1

Funded by the Erasmus+ Programme of the European Union

Math Functions

MOD(*value1*,*value2*)

Determines the remainder of dividing *value1* by *value2*.

```
SELECT SAL, MOD(SAL,3) FROM EMP;
```

SAL	MOD(SAL,3)
800.00	2.00
1600.00	1.00
1250.00	2.00
2975.00	2.00
1250.00	2.00
2850.00	0.00
2450.00	2.00
3000.00	0.00
5000.00	2.00
1500.00	0.00
1100.00	2.00
950.00	2.00
3000.00	0.00
1300.00	1.00

Funded by the Erasmus+ Programme of the European Union

Operations with dates

To view the current date

```
SELECT CURDATE();
```

	CURDATE()
▶	2023-03-12

Funded by the Erasmus+ Programme of the European Union

Date Formats
DATE_FORMAT()

%a Abbreviated Week Name (Mon to Sun)
%b Abbreviated month name (Jan to Dec)
%c Month numerically (1 to 12)
%D Day of month with English suffix (1st, 2nd, 3rd, …)
%d Day of the Month numerically (01 to 31)
%e Day of the Month numerically (1 to 31)
%f Micro seconds (000000..999999)
% H Hours (00 to 23)
%h Hours (01 to 12)
%I Hours (01 to 12)
%i Numeric minutes (00 to 59)
%j Day of Year (001 to 366)
%k Hours (0 to 23)
%l Hours (1 to 12)
%M Month name (January to December)
%m Month numerically (01 to 12)
%p AM or PM
%r Hours, 12-hours (hh:mm:ss followed by AM or PM)
%S Seconds (00 to 59)
%s Seconds (00 to 59)
%T Hours, 24-hour (hh:mm:ss)
%U Week (00 to 53), where Sunday is the first day of the week
%u Week (00 to 53), where Monday is the first day of the week
%V Week (00 to 53), where Sunday is the first day of the week; used with %X
%v Week (00 to 53), where Monday is the first day of the week; used with %x
%W Name of the day of the week (Monday to Sunday)
%w D ay of the week (0=Sunday to 6=Saturday)
%X Day of the week where Sunday is the first day of the week, 4-digit numerical form, used with %V
%x Year of the week, where Monday is the first day of the week, numerically, 4 digits, used with %v
%Y 4-digit numeric year
%y Numeric 2-digit year
%% A single “%” character
%x x, for any “x” not listed above

Funded by the Erasmus+ Programme of the European Union

Current Date
CURDATE()

```
SELECT date_format(CURDATE(),'%Y-%m-%d') "Current Date" ;
```

	Current Date
▶	2023-03-12

Funded by the Erasmus+ Programme of the European Union

Operations with Dates Date+number

Adds a number of days to the date, producing a date.

```
SELECT CURDATE() "Today", CURDATE()+5 "5 Days from Today";
```

	Today	5 Days from Today
▶	2023-03-12	20230317

Funded by the Erasmus+ Programme of the European Union

Operations with Dates Date-number

Subtracts a number of days from the date, producing a date.

```
SELECT CURDATE() "Today", CURDATE() -5 "5 Days ago";
```

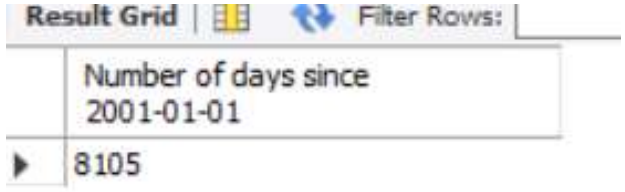
	Today	5 Days ago
▶	2023-03-12	20230307

Funded by the Erasmus+ Programme of the European Union

Subtract Dates Datediff()

Number of days between two dates.

```
Select datediff(sysdate(), '2001-01-01') "Number of days
since 2001-01-01";
```

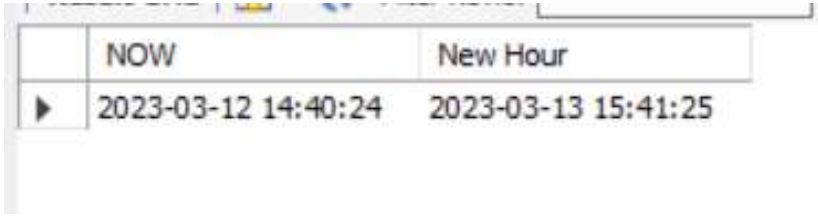


Funded by the Erasmus+ Programme of the European Union

Operations with Dates ADDTIME()

Adds a number of hours to the date, producing a date.

```
SELECT DATE_FORMAT(SYSDATE(), '%Y-%m-%d %T') "NOW",
DATE_FORMAT(ADDTIME(SYSDATE(),'1 1:1:1.000002'), '%Y-%m-%d
%T') "New Hour";
```



	NOW	New Hour
▶	2023-03-12 14:40:24	2023-03-13 15:41:25

Funded by the Erasmus+ Programme of the European Union

Date Functions

ADDDATE()	Add time values (intervals) to a date value
ADDTIME()	Add time
CONVERT_TZ()	Convert from one timezone to another
CURDATE()	Return the current date
CURRENT_DATE(), CURRENT_DATE	Synonyms for CURDATE()
CURRENT_TIME(), CURRENT_TIME	Synonyms for CURTIME()
CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP	Synonyms for NOW()
CURTIME()	Return the current time
DATE_ADD()	Add time values (intervals) to a date value
DATE_FORMAT()	Format date as specified
DATE_SUB()	Subtract a time value (interval) from a date
DATE()	Extract the date part of a date or datetime expression
DATEDIFF()	Subtract two dates
DAY()	Synonym for DAYOFMONTH()
DAYNAME()	Return the name of the weekday
DAYOFMONTH()	Return the day of the month (0-31)
DAYOFWEEK()	Return the weekday index of the argument
DAYOFYEAR()	Return the day of the year (1-366)
EXTRACT()	Extract part of a date
FROM_DAYS()	Convert a day number to a date
FROM_UNIXTIME()	Format UNIX timestamp as a date
GET_FORMAT()	Return a date format string
HOUR()	Extract the hour
LAST_DAY	Return the last day of the month for the argument
LOCALTIME(), LOCALTIME	Synonym for NOW()
LOCALTIMESTAMP, LOCALTIMESTAMP()	Synonym for NOW()

Funded by the Erasmus+ Programme of the European Union

Date Functions

MAKEDATE()	Create a date from the year and day of year
MAKETIME	MAKETIME()
MICROSECOND()	Return the microseconds from argument
MINUTE()	Return the minute from the argument
MONTH()	Return the month from the date passed
MONTHNAME()	Return the name of the month
NOW()	Return the current date and time
PERIOD_ADD()	Add a period to a year-month
PERIOD_DIFF()	Return the number of months between periods
QUARTER()	Return the quarter from a date argument
SEC_TO_TIME()	Converts seconds to 'HH:MM:SS' format
SECOND()	Return the second (0-59)
STR_TO_DATE()	Convert a string to a date
SUBDATE()	A synonym for DATE_SUB() when invoked with three arguments
SUBTIME()	Subtract times
SYSDATE()	Return the time at which the function executes
TIME_FORMAT()	Format as time
TIME_TO_SEC()	Return the argument converted to seconds
TIME()	Extract the time portion of the expression passed
TIMEDIFF()	Subtract time
TIMESTAMP()	With a single argument, this function returns the date or datetime expression; with two arguments, the sum of the arguments
TIMESTAMPADD()	Add an interval to a datetime expression
TIMESTAMPDIFF()	Subtract an interval from a datetime expression
TO_DAYS()	Return the date argument converted to days
TO_SECONDS()	Return the date or datetime argument converted to seconds since Year 0
UNIX_TIMESTAMP()	Return a UNIX timestamp
UTC_DATE()	Return the current UTC date
UTC_TIME()	Return the current UTC time
UTC_TIMESTAMP()	Return the current UTC date and time
WEEK()	Return the week number
WEEKDAY()	Return the weekday index
WEEKOFYEAR()	Return the calendar week of the date (0-53)
YEAR()	Return the year
YEARWEEK()	Return the year and week

Funded by the Erasmus+ Programme of the European Union

Functions that accept any datatype

IFNULL(*col|valor*, *val*)

Converts a null into a value.

```
SELECT COMM, IFNULL(COMM,0) FROM EMP;
```

	COMM	IFNULL(COMM,0)
▶	NULL	0.00
	300.00	300.00
	500.00	500.00
	NULL	0.00
	1400.00	1400.00
	NULL	0.00
	NULL	0.00

Funded by the Erasmus+ Programme of the European Union

Functions that accept any datatype

IFNULL(*col|valor*, *val*)

```
SELECT JOB, IFNULL(JOB, '******') FROM EMP;
```

JOB	IFNULL(JOB, '******')
CLERK	CLERK
SALESMAN	SALESMAN
SALESMAN	SALESMAN
MANAGER	MANAGER
SALESMAN	SALESMAN
MANAGER	MANAGER