Lab-05

# Single Rows Function including Date Functions

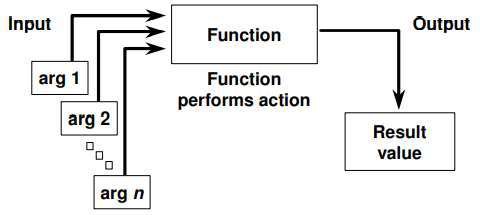
**Objective:**

Ԅ Students will learn how to use built-in functions and retrieve data using those functions

# SQL Functions

Functions are a very powerful feature of SQL and can be used to do the following:

* Perform calculations on data
* Modify individual data items
* Manipulate output for groups of rows
* Format dates and numbers for display
* Convert column data types

SQL functions sometimes take arguments and always return a value.

# Types of SQL Functions

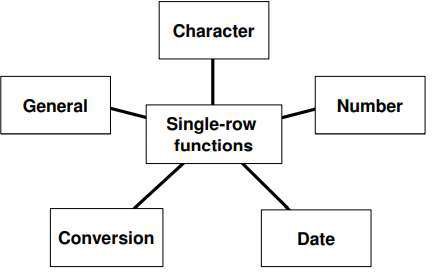
There are two distinct types of functions:

* Single-row functions
* Multiple-row functions

# Single-Row Functions

These functions operate on single rows only and return one result per row. There are different types of single-row functions. This lesson covers the following ones:

* Character
* Number
* Date
* Conversion



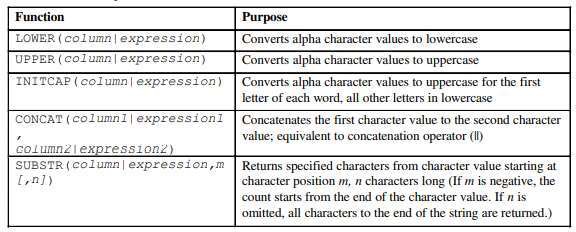
* Character functions: Accept character input and can return both character and number values
* Number functions: Accept numeric input and return numeric values
* Date functions: Operate on values of the DATE data type (All date functions return a value of DATE data type except the MONTHS\_BETWEEN function, which returns a number.)
* Conversion functions: Convert a value from one data type to another
* General functions:

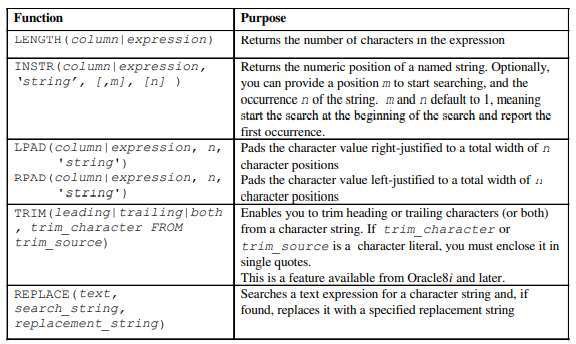
- NVL - NVL2 - NULLIF - COALSECE - CASE - DECODE

# Character Functions

Single-row character functions accept character data as input and can return both character and numeric values. Character functions can be divided into the following:

* Case-manipulation functions
* Character-manipulation functions

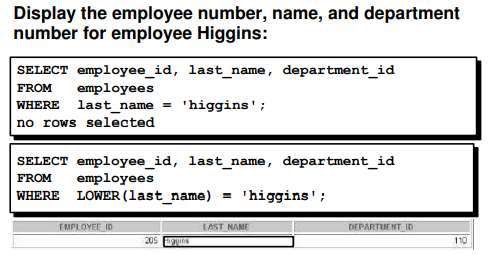




Case Manipulation Functions

LOWER, UPPER, and INITCAP are the three case-conversion functions.

* LOWER: Converts mixed case or uppercase character strings to lowercase
* UPPER: Converts mixed case or lowercase character strings to uppercase
* INITCAP: Converts the first letter of each word to uppercase and remaining letters to lowercase

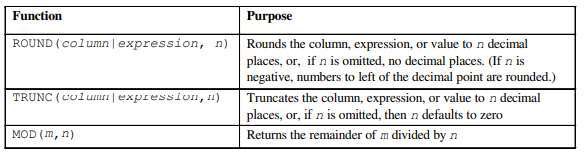


# Character Manipulation Functions

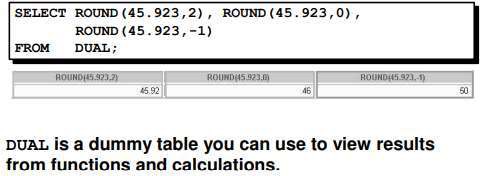
CONCAT, SUBSTR, LENGTH, INSTR, LPAD, RPAD, and TRIM are the character manipulation functions covered in this lesson.

* CONCAT: Joins values together (you are limited to using two parameters with CONCAT
* SUBSTR: Extracts a string of determined length
* LENGTH: Shows the length of a string as a numeric value
* INSTR: Finds numeric position of a named character
* LPAD: Pads the character value right-justified
* RPAD: Pads the character value left-justified
* TRIM: Trims heading or trailing characters (or both) from a character string (Iftrim\_character or trim\_source is a character literal, you must enclose it in single quotes.)

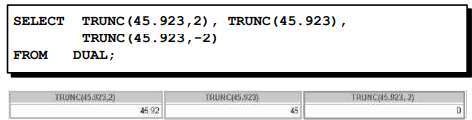
# Number Functions

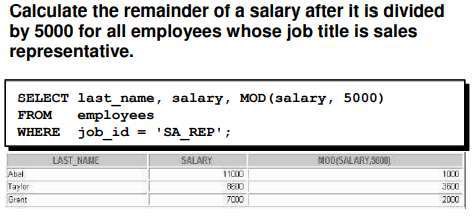
Number functions accept numeric input and return numeric values. This section describes some of the number functions.

**ROUND Function**



**TRUNC Function**





**MOD Function**

**Multiple-Row Functions**

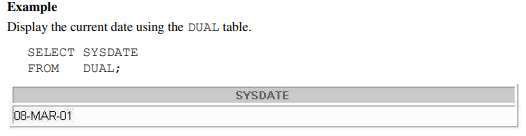
Functions can manipulate groups of rows to give one result per group of rows. These functions are known as group functions. This is covered in a later lesson.

DATE FUNCTIONS

# Oracle Date Format

**The SYSDATE Function**

SYSDATE is a date function that returns the current database server date and time. You can use SYSDATE just as you would use any other column name. For example, you can display the current date by selecting SYSDATE from a table. It is customary to select SYSDATE from a dummy table called DUAL.

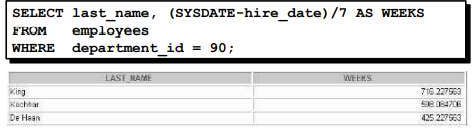


# Using Arithmetic Operators with Dates

The example in the slide displays the last name and the number of weeks employed for all employees in department 90. It subtracts the date on which the employee was hired from the current date (SYSDATE) and divides the result by 7 to calculate the number of weeks that a worker has been employed.

Note: SYSDATE is a SQL function that returns the current date and time. Your results may differ from the example.

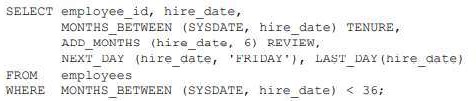
If a more current date is subtracted from an older date, the difference is a negative number

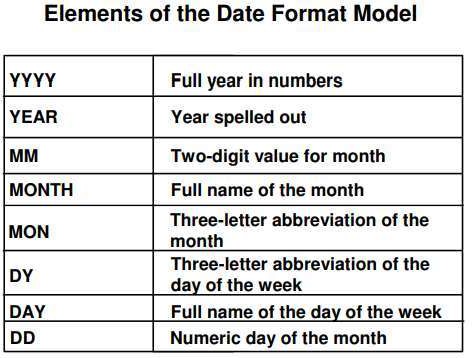


# Date Functions

**Using Date Functions**

For example, display the employee number, hire date, number of months employed, six-month review date, first Friday after hire date, and last day of the month when hired for all employees employed for fewer than 36 months.





**Lab Tasks:**

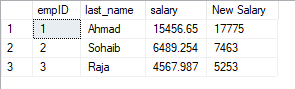
1. Write a query to display the current date. Label the column Date.

SELECT CURRENT\_TIMESTAMP AS Date;



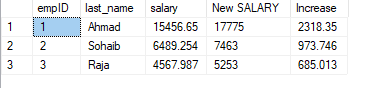
1. For each employee, display the employee ID number, last\_name, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary. Place your SQL statement in a text file named lab3\_2.sql. Run your query

SELECT empID,last\_name,salary,ROUND((salary\*0.15)+salary,0) AS 'New Salary' FROM Employee;



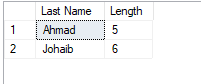
1. Modify your query lab3\_2.sql to add a column that subtracts the old salary from the new salary. Label the column Increase. Save the contents of the file as lab3\_4.sql. Run the revised query.

SELECT empID,last\_name,salary,ROUND((salary\*0.15+salary),0)AS 'New SALARY',(ROUND((salary\*0.15+salary),0)-salary) AS 'Increase' FROM Employee;



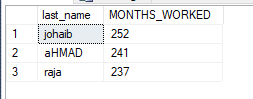
1. Write a query that displays the employee's last names with the first letter capitalized and all other letters lowercase and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.

SELECT CONCAT(UPPER(SUBSTRING(last\_name,1,1)),LOWER(SUBSTRING(last\_name,2,LEN(last\_name)))) AS 'Last Name',LEN(last\_name) AS 'Length' FROM employee WHERE SUBSTRING(last\_name,1,1) = 'A' OR SUBSTRING(last\_name,1,1) = 'J' OR SUBSTRING(last\_name,1,1) = 'M' ORDER BY last\_name ASC ;



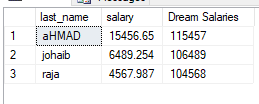
1. For each employee, display the employee's last name, and calculate the number of months between today and the date the employee was hired. Label the column MONTHS\_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number. Note: Your results will differ.

SELECT last\_name ,ROUND(DATEDIFF(month,hire\_date,CURRENT\_TIMESTAMP),0) AS 'MONTHS\_WORKED' FROM Employee ORDER BY hire\_date;



1. Write a query that produces the following for each employee: earns monthly but wants . Label the column Dream Salaries.

SELECT last\_name, salary,ROUND((salary+100000),0) AS'Dream Salaries' FROM Employee;



1. Create a query to display the last name and salary for all employees. Format the salary to be 15 For Example:

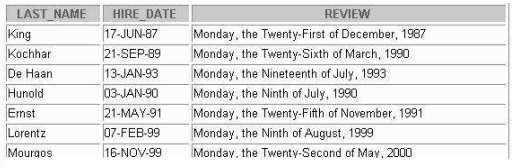
SELECT last\_name,CONCAT(REPLICATE('$',8),salary) AS 'Salary' FROM Employee;

A screenshot of a graph

Description automatically generated

1. Display each employee's last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear similar to "Monday, the Thirty-First of July, 2000."

For Example:



1. Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week starting with Monday.

SELECT last\_name,hire\_date,DATENAME(Weekday,hire\_date) AS 'DAY' FROM Employee ORDER BY DATENAME(Weekday,hire\_date) DESC;

A close-up of a number

Description automatically generated

1. Create a query that displays the employees' last names and commission amounts. If an employee does not earn commission, put "No Commission." Label the column COMM.
2. SELECT last\_name,comission AS COMM FROM Employee;

A screenshot of a computer

Description automatically generated

1. Create a query that displays the employees' last names and indicates the amounts of their annual salaries with asterisks. Each asterisk signifies a thousand doll ars. Sort the data in descending order of salary. Label the column EMPLOYEES\_AND\_THEIR\_SALARIES.