Using Single-Row Function to Customize Output

Lesson Objectives

After completing this lesson, you should be able to do the following:

- Describe various types of functions available in SQL
- Use character, number and date functions in `SELECT` statements

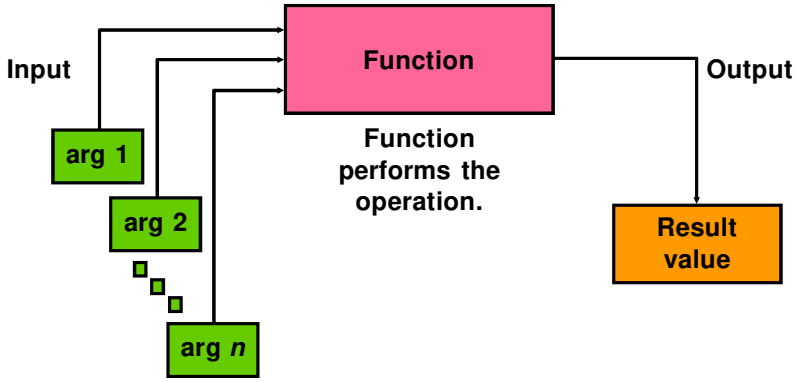
Lesson Agenda

- Single-row SQL Functions
- Character Functions
- Number Functions
- Working with Dates
- Date Functions

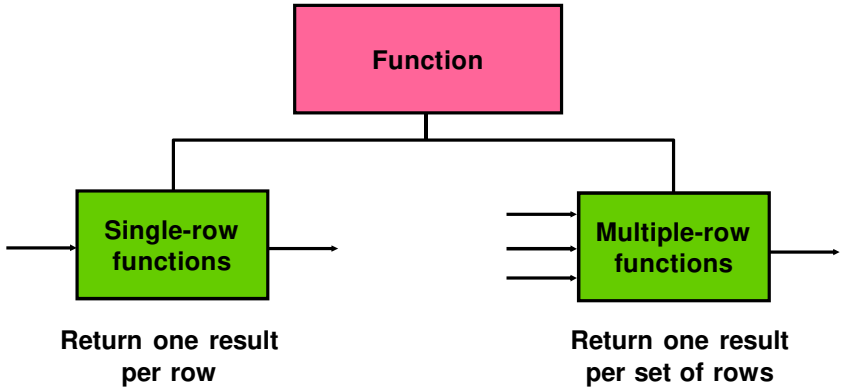
Lesson Agenda

- **Single-row SQL Functions**
- Character Functions
- Number Functions
- Working with Dates
- Date Functions

SQL Functions



Two Types of SQL Functions



Single-Row Functions

Single-row functions:

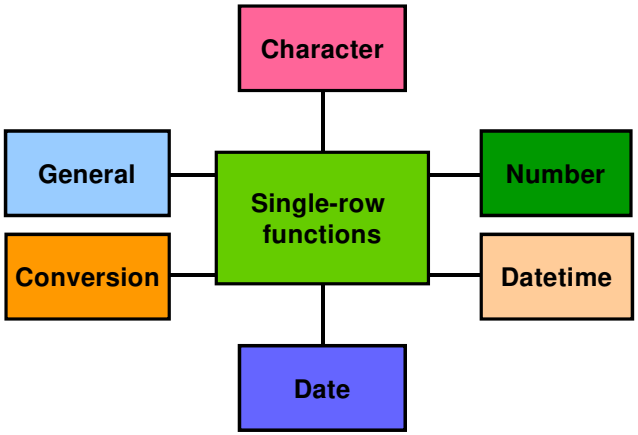
- Manipulate data items
- Accept arguments and return one value
- Act on each row that is returned
- Return one result per row
- May modify the data type
- Can be nested
- Accept arguments that can be a column or an expression

Single-Row Functions

```
function_name(column|expression|[,arg1,arg2,...])
```

function_name	the name of the function
column	any named database column
expression	any character string or calculated expression
arg1, arg2, …	any argument to be used by the function

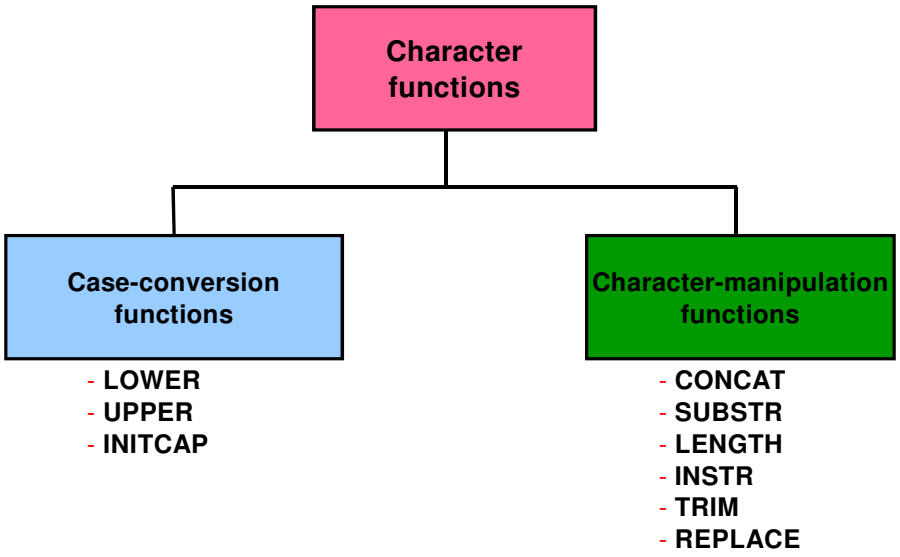
Single-Row Functions



Lesson Agenda

- Single-row SQL Functions
- **Character Functions**
- Number Functions
- Working with Dates
- Date Functions

Character Functions



Character Functions

Case-Conversion Functions

Function	Purpose
`LOWER(column|expression)`	Converts alpha character values to lower-case.
`UPPER(column|expression)`	Converts alpha character values to upper-case.
`INITCAP(column|expression)`	Converts alpha character values to uppercase for the first letter of each word; all other letters in lowercase.

Character Functions

Character-Manipulation Functions

Function	Purpose
CONCAT(column1|expression1, column2|expression2)	Concatenates the first character value to the second character value; equivalent to concatenation operator (||).
SUBSTR(column|expression, position_m[,n])	Returns specified characters from character value starting at character *position_m*, *n* characters long (If *m* is negative, the count starts from the end of the character value, If *n* is omitted, all characters to the end of the string are returned.)

Character Functions

Character-Manipulation Functions (continue)

Function	Purpose
LENGTH(column|expression)	Returns the number of characters in the expression.
INSTR(column|expression, 'string' [,m] [,n])	Return the numeric position of a named string. Optionally, you can provide a position *m* to start searching, and the occurrence n of the string. *m* and *n* default to 1, meaning start the search at the beginning of the string and report the first occurrence.

Character Functions

Character-Manipulation Functions (continue)

Function	Purpose
TRIM(leading|trailing|both, trim_character FROM trim_source)	Enables you to trim leading or trailing characters(or both) from a character string. If *trim_character* or *trim_source* is a character literal, you must enclose it in single quotation marks.
REPLACE(text, search_string, replacement_string)	Searches a text expression for a character string and, if found replaces it with a specified replacement string.

Case-Conversion Functions

String	Function	Result
Sql Course	LOWER('Sql Course')	sql course
Sql Course	UPPER('Sql Course')	SQL COURSE
sql course	INITCAP('sql course')	Sql Course

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

Case-Conversion Functions

	EMPLOYEE DETAILS
1	The job id for KING is ad_pres
2	The job id for KOCHHAR is ad_vp
3	The job id for DE HAAN is ad_vp
4	The job id for HUNOLD is it_prog
5	The job id for ERNST is it_prog
6	The job id for AUSTIN is it_prog
7	The job id for PATABALLA is it_prog
8	The job id for LORENTZ is it_prog
9	The job id for GREENBERG is fi_mgr
10	The job id for FAVIET is fi_account
11	The job id for CHEN is fi_account
12	The job id for SCIARRA is fi_account

Use column last_name and job_id from employees table.

```
SELECT  'The job id for ' || UPPER(last_name) ||
        ' is ' || LOWER(job_id) AS "EMPLOYEE DETAILS"
FROM    employees;
```


Using Case-Conversion Functions

For example, Display the employee number, last name and department number for last name is ***higgins***

```
SELECT  employee_id, last_name,  department_id
FROM    employees
WHERE   last_name =  'higgins';
```

EMPLOY...	LAST_N...	DEPART...

Display 0 row.

```
SELECT  employee_id, last_name,  department_id
FROM    employees
WHERE   LOWER(last_name) = 'higgins';
```

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	205	Higgins	110

Using Case-Conversion Functions

PRACTICE:

Display the employee number, last name and department number for last name is *Higgins*.

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	205	Higgins	110

Character-Manipulation Functions

Function	Result
CONCAT('Hello', 'World')	HelloWorld
SUBSTR('HelloWorld', 1, 5)	Hello
LENGTH('HelloWorld')	10
INSTR('HelloWorld', 'W')	6
TRIM('H' FROM 'HelloWorld')	elloWorld
REPLACE('JACK and JUE', 'J', 'BL')	BLACK and BLUE

Using Character-Manipulation Functions

```
SELECT  employee_id,
        _____________________________________ NAME,
        job_id,
        _____________________________,
        _____________________ "Contains  'a'?"
FROM    employees
WHERE   SUBSTR(job_id,4)  =  'REP';
```

	EMPLOYEE_ID	NAME	JOB_ID	LENGTH(LAST_NAME)	Contains 'a'?
1	150	PeterTucker	SA_REP	6	0
2	151	DavidBernstein	SA_REP	9	0
3	152	PeterHall	SA_REP	4	2
4	153	ChristopherOlsen	SA_REP	5	0
5	154	NanetteCambrault	SA_REP	9	2

Using Character-Manipulation Functions

Modify the SQL statement in the previous slide to display the data for those employees whose last names end with the letter '**n**'.

	EMPLOYEE_ID	NAME	JOB_ID	LENGTH(LAST_NAME)	Contains 'a'?
1	102	LexDe Haan	AD_VP	7	5
2	105	DavidAustin	IT_PROG	6	0
3	110	JohnChen	FI_ACCOUNT	4	0
4	112	Jose ManuelUrman	FI_ACCOUNT	5	4

Display 19 rows.

```
SELECT  employee_id,
        _____________________________________ NAME,
        job_id,
        _____________________________,
        _____________________ "Contains  'a'?"
FROM    employees
WHERE   _____________________________________;
```


Character-Manipulation Functions

These functions manipulate character strings:

Function	Purpose
`LPAD(column|expression, n, 'string')`	Return an expression left-padded to length of *n* characters with a character expression.
`RPAD(column|expression, n, 'string')`	Return an expression right-padded to length of *n* characters with a character expression.

Character-Manipulation Functions

Example of LPAD

```
SELECT  LPAD('5000', 10, '*')
FROM    dual;
```

	LPAD('5000',10,'*')
1	******5000

Example of RPAD

```
SELECT  RPAD('5000', 10, '*')
FROM    dual;
```

	RPAD('5000',10,'*')
1	5000******

LPAD and RPAD Functions

In this slide, name is right padded to its length of 20 characters with the ' ' and left padded with * to its length of 20.

	FIRST_NAME	FIRST_NAME1	FIRST_NAME2
1	Ellen	Ellen	**********Ellen
2	Sundar	Sundar	**********Sundar
3	Mozhe	Mozhe	**********Mozhe
4	David	David	**********David
5	Hermann	Hermann	**********Hermann
6	Shelli	Shelli	**********Shelli
7	Amit	Amit	**********Amit

Lesson Agenda

- Single-row SQL Functions
- Character Functions
- **Number Functions**
- Working with Dates
- Date Functions

Number Functions

Function	Purpose
ROUND(column|expression, n)	Rounds the column, expression, or value to *n* decimal place or, if *n* is omitted, no decimal places. (If *n* is negative, numbers to the left of decimal point are rounded.)
TRUNC(column|expression, n)	Truncates the column, expression or value to *n* decimal places or, if *n* is omitted, *n* defaults to zero.
MOD(m, n)	Returns the remainder of *m* divided by *n*.

- ROUND: Rounds value to specified decimal.
- TRUNC: Truncates value to specified decimal.
- MOD: Returns remainder of division.

Number Functions

Function	Result
ROUND(45.926, 2)	45.93
TRUNC(45.926, 2)	45.92
MOD(1600, 300)	100

- ROUND: Rounds value to specified decimal.
- TRUNC: Truncates value to specified decimal.
- MOD: Returns remainder of division.

Using the ROUND Function

```
SELECT   ROUND(67830.4557,  3),
         ROUND(67830.4557, 2),
         ROUND(67830.4557, -3)
FROM     dual;
```

	ROUND(67830.4557,3)	ROUND(67830.4557,2)	ROUND(67830.4557,-3)
1	67830.456	67830.46	68000

- DUAL is a dummy table that you can use to view results from functions and calculations.
- If the second argument is 0 or is missing, the value is rounded to zero decimal places. If the second argument is 2, the value is rounded to two decimal places. Conversely, if the second argument is -2, the value is rounded to two decimal places to the left(rounded to the nearest unit of 100).

Using the TRUNC Function

```
SELECT   TRUNC(67830.4557,  3),
         TRUNC(67830.4557, 2),
         TRUNC(67830.4557, -3)
FROM     dual;
```

	TRUNC(67830.4557,3)	TRUNC(67830.4557,2)	TRUNC(67830.4557,-3)
1	67830.455	67830.45	67000

- The TRUNC function works with arguments similar to those of the ROUND function.
- If the second argument is 0 or is missing, the value is truncated to zero decimal places. If the second argument is 2, the value is truncated to two decimal places. Conversely, if the second argument is -2, the value is truncated to two decimal places to the left.

Using the MOD Function

For all employees with the job title of Sales Representative, calculate the remainder of the salary after it is divided by 5,000.

	LAST_NAME	SALARY	Mod Salary
1	Tucker	10000	0
2	Bernstein	9500	4500
3	Hall	9000	4000
4	Olsen	8000	3000
5	Cambrault	7500	2500

Display 30 rows.

```
SELECT   last_name,  salary,
         _____________________________ "Mod  Salary"
FROM     dual;
```


Lesson Agenda

- Single-row SQL Functions
- Character Functions
- Number Functions
- **Working with Dates**
- Date Functions

Working with Dates

- Oracle database stores dates in an internal numeric format: century, year, month, day, hours, minutes, seconds.
- The default date display format is DD-MON-RR.

Working with Dates

In the example in the slide, the HIRE_DATE column output is displayed in the default format DD-MON-RR.

	LAST_NAME	HIRE_DATE
1	King	17-JUN-87
2	Whalen	17-SEP-87

```
SELECT  last_name, hire_date
FROM    employees
WHERE   _____________________;
```


Using the SYSDATE Function

SYSDATE is a function that returns:

- Date
- Time

	SYSDATE
1	06-JUN-18

```
SELECT  SYSDATE
FROM    dual;
```

- sysdate function will return the US date and time.
- sysdate returns the current date and time set for the operating system on which the database resides.

Arithmetic with Dates

Operation	Result	Description
date+number	Date	Adds a number of days to a date.
date-number	Date	Subtracts a number of days from a date.
date-date	Number of days	Subtracts one date from another.
date+number /24	Date	Adds a number of hours to a date.

- Add or subtract a number to or from a date for a resultant date value.
- Subtract two dates to find the number of days between those dates.
- Add hours to a date by dividing the number of hours by 24.

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.

```
SELECT  last_name, (SYSDATE-hire_date)/7 AS WEEKS
FROM    employees
WHERE   department_id = 90;
```

	LAST_NAME	WEEKS
1	King	1616.085292658730158730158730158730158729
2	Kochhar	1497.942435515873015873015873015873015871
3	De Haan	1325.08529265873015873015873015873015873

NOTE: Executed program at 06-Jun-18 (SYSDATE = 06-Jun-18)

Lesson Agenda

- Single-row SQL Functions
- Character Functions
- Number Functions
- Working with Dates
- **Date Functions**

Date Functions

Function	Result	Format
MONTHS_BETWEEN	Number of months between two dates.	MONTHS_BETWEEN(date1, date2)
ADD_MONTHS	Add calendar months to date.	ADD_MONTHS(date, n)
NEXT_DAY	Next day of the date specified.	NEXT_DAY(date, 'char')
LAST_DAY	Last day of the month.	LAST_DAY(date)

Using Date Functions

Function	Result
MONTHS_BETWEEN('01-SEP-95', '11-JAN-94')	19.6774194
ADD_MONTHS('31-JAN-96', 1)	29-FEB-96
NEXT_DAY('01-SEP-95', 'FRIDAY')	08-SEP-95
LAST_DAY('01-FEB-95')	28-FEB-95

Using Date Functions

The HR department wants to find the duration of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which the employee was hired. Label the column MONTHS_WORKED. Round the number of months up to the closest whole number.

	LAST_NAME	MONTHS_WORKED
1	King	372
2	Kochhar	345
3	De Haan	305
4	Hunold	341
5	Ernst	325

```
SELECT   last_name,
         ROUND(MONTHS_BETWEEN(SYSDATE,  hire_date))
         AS  MONTHS_WORKED
FROM     employees;
```


Using ROUND and TRUNC Functions with Date

Assume SYSDATE = '25-JUL-08';

Function	Result
ROUND(SYSDATE, 'MONTH')	01-AUG-03
ROUND(SYSDATE, 'YEAR')	01-JAN-04
TRUNC(SYSDATE, 'MONTH')	01-JUL-03
TRUNC(SYSDATE, 'YEAR')	01-JAN-03

- Round dates to the nearest year or month. If the format model is month, dates 1-15 result in the first day of the current month. Date 16-31 result in the first day of the next month.
- If the format model is year months 1-6 result in January 1 of the current year. Months 7-12 result in January 1 of the next year.

Using ROUND and TRUNC Functions with Date

Compare the hire dates for all employees who started in 1977. Display the employee number, hire date, and starting month using the ROUND and TRUNC functions.

	EMPLOYEE_ID	HIRE_DATE	ROUNDMONTH	TRUNCMONTH
1	105	25-JUN-97	01-JUL-97	01-JUN-97
2	110	28-SEP-97	01-OCT-97	01-SEP-97
3	111	30-SEP-97	01-OCT-97	01-SEP-97
4	116	24-DEC-97	01-JAN-98	01-DEC-97

```
SELECT   employee_id,  hire_date,
         ROUND(hire_date,  'MONTH') AS  ROUNDMONTH,
         TRUNC(hire_date,  'MONTH') AS  TRUNCMONTH
FROM     employees
WHERE    hire_date LIKE  '%97';
```


Using Date Functions

PRACTICE:

Displays the employee number, hire date, six-month review date, first Friday after hire date, the last day of the hire month, and number of months employed for all employees who have been employed for more than 250 months.

	EMPLOYEE_ID	HIRE_DATE	REVIEW	Next Hiredate	Last Hiredate	TENURE
1	100	17-JUN-87	17-DEC-87	19-JUN-87	30-JUN-87	371.665038829151732377538829151732377539
2	101	21-SEP-89	21-MAR-90	22-SEP-89	30-SEP-89	344.536006571087216248506571087216248507
3	102	13-JAN-93	13-JUL-93	15-JAN-93	31-JAN-93	304.794071087216248506571087216248506571
4	103	03-JAN-90	03-JUL-90	05-JAN-90	31-JAN-90	341.116651732377538829151732377538829152
5	104	21-MAY-91	21-NOV-91	24-MAY-91	31-MAY-91	324.536006571087216248506571087216248507

Quiz

Which of the following statements are true about single-row functions?

- [ ] Manipulate data times.
- [ ] Accept arguments and return one value per argument.
- [ ] Act on each row that is returned.
- [ ] Return one result per set of rows.
- [ ] May not modify the data type.
- [ ] Can be nested.
- [ ] Accept arguments that can be a column or an expression.

Summary

In this lesson, you should have learned how to:

- Perform calculations on data using functions
- Modify individual data items using functions

Summary

- Single-row functions can be nested to any level.Single-row functions can manipulate the following:
- Character date:LOWER,UPPER,INITCAP,CONCAT,SUBSTR,INSTR,LENGTH
- Number data:ROUND,TRUNC,MOD
- Date values:SYSDATE,MONTHS_BETWEEN,ADD_MONTHS,NEXT_DAY, LAST_DAY

SYSDATE and DUAL

- SYSDATE is a date function that returns the current date and time. It is customary to select
- SYSDATE from a dummy table called DUAL