3

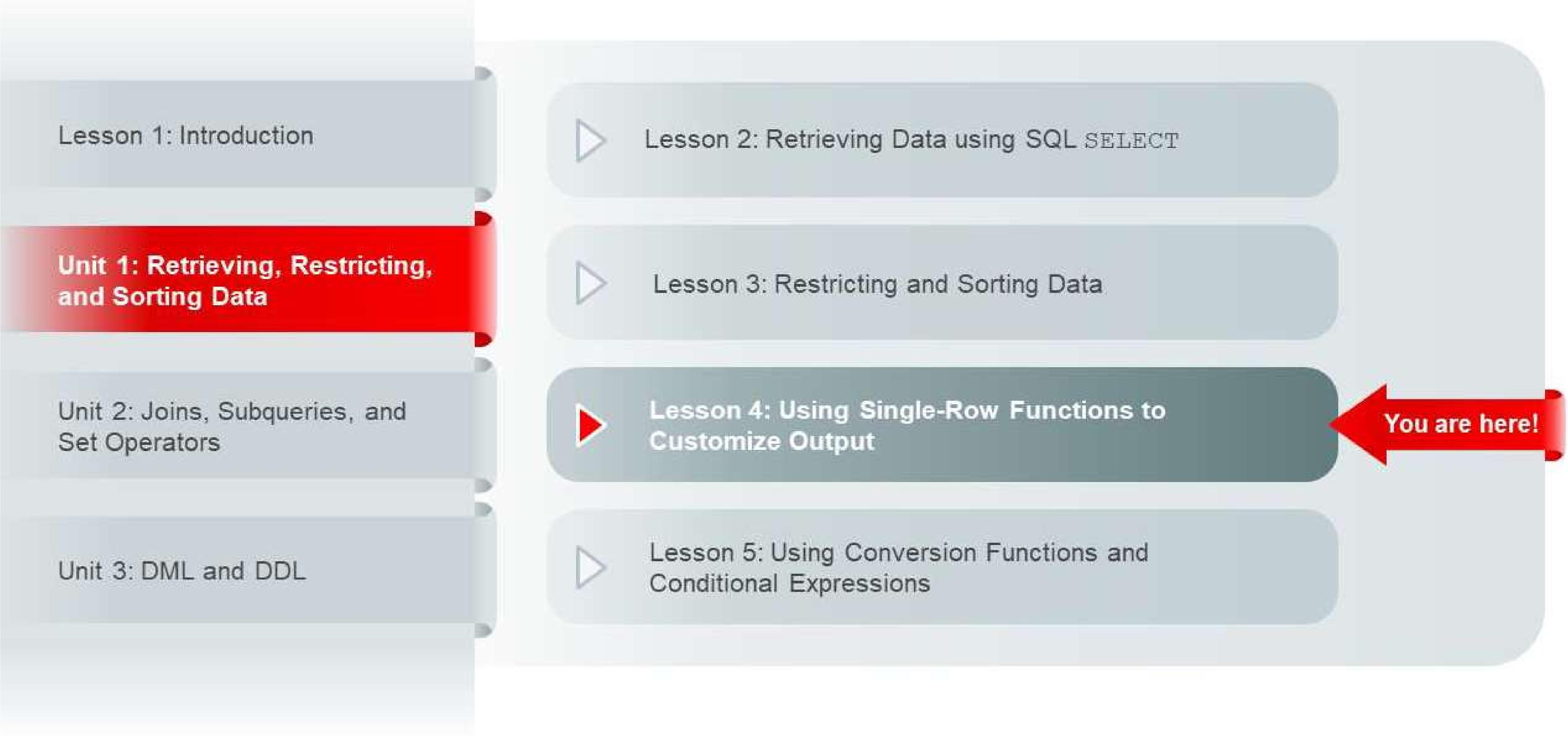
Using Single Row Functions to Customize the Output

Objectives

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

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

Course RoadMap



Lesson 1: Introduction

Unit 1: Retrieving, Restricting, and Sorting Data

Unit 2: Joins, Subqueries, and Set Operators

Unit 3: DML and DDL

Lesson 2: Retrieving Data using SQL `SELECT`

Lesson 3: Restricting and Sorting Data

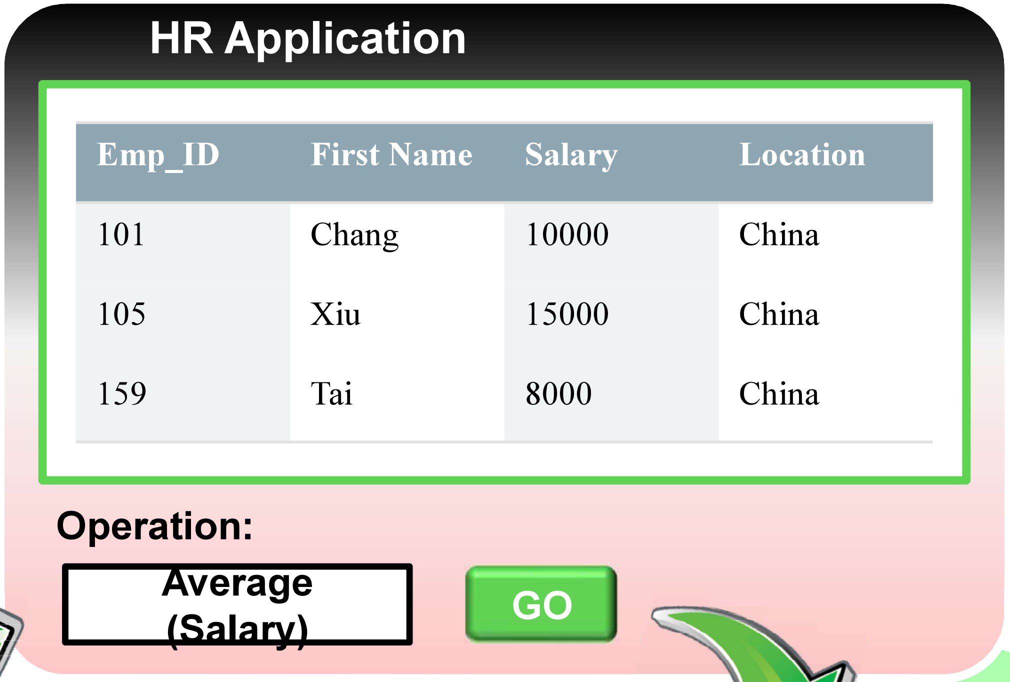
Lesson 4: Using Single-Row Functions to Customize Output

You are here!

Lesson 5: Using Conversion Functions and Conditional Expressions

HR Application Scenario

How do I calculate the average salary of all employees working in China.

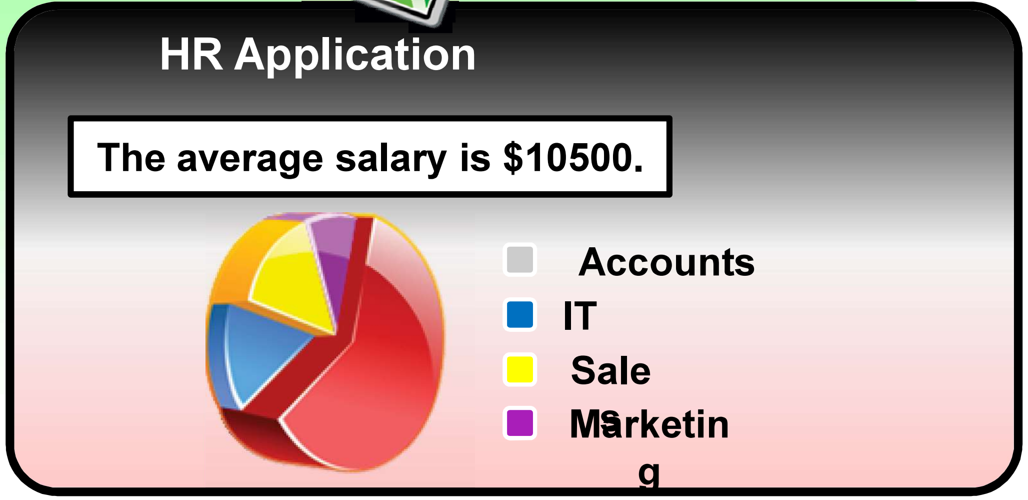


HR Application

Emp_ID	First Name	Salary	Location
101	Chang	10000	China
105	Xiu	15000	China
159	Tai	8000	China

Operation:

Average (Salary)



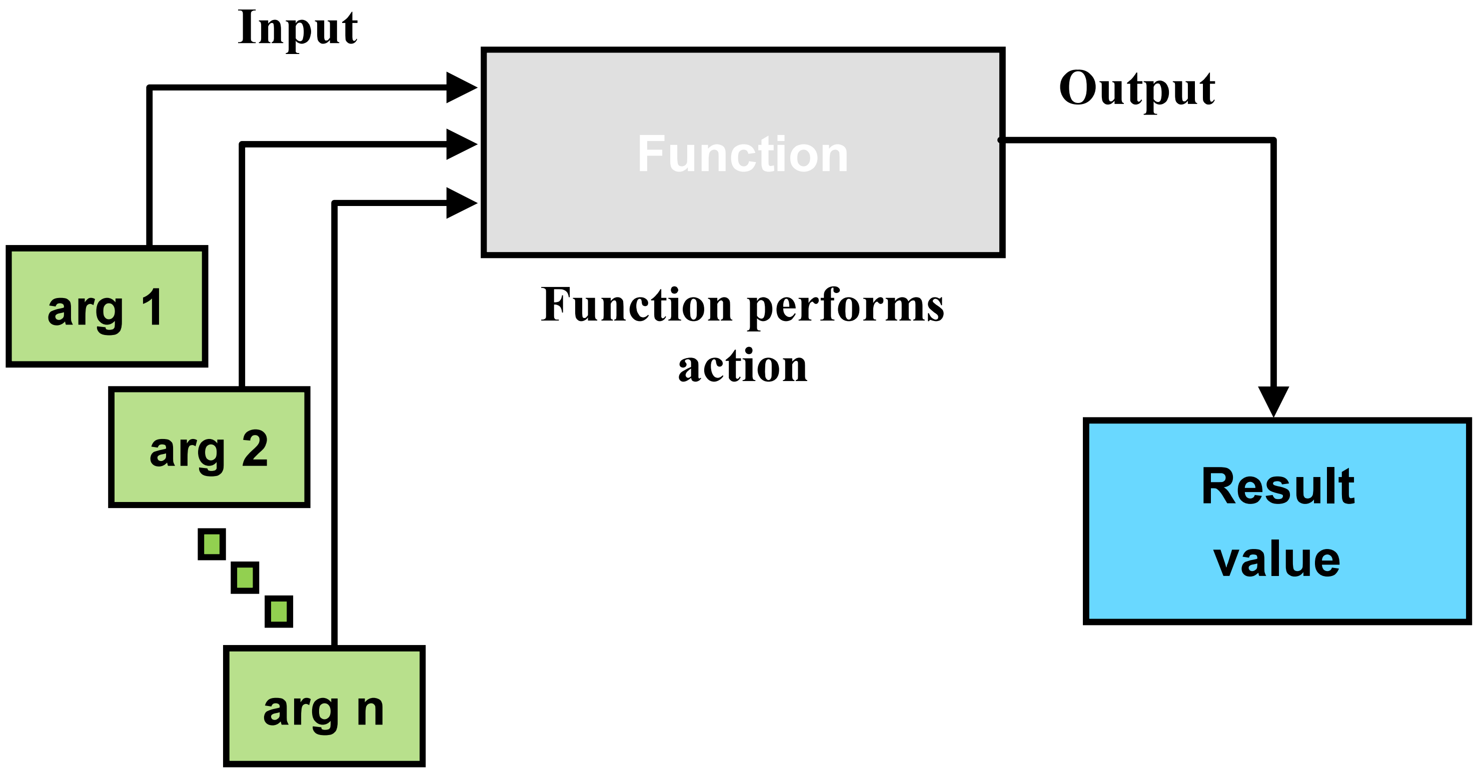
HR Application

The average salary is $10500.

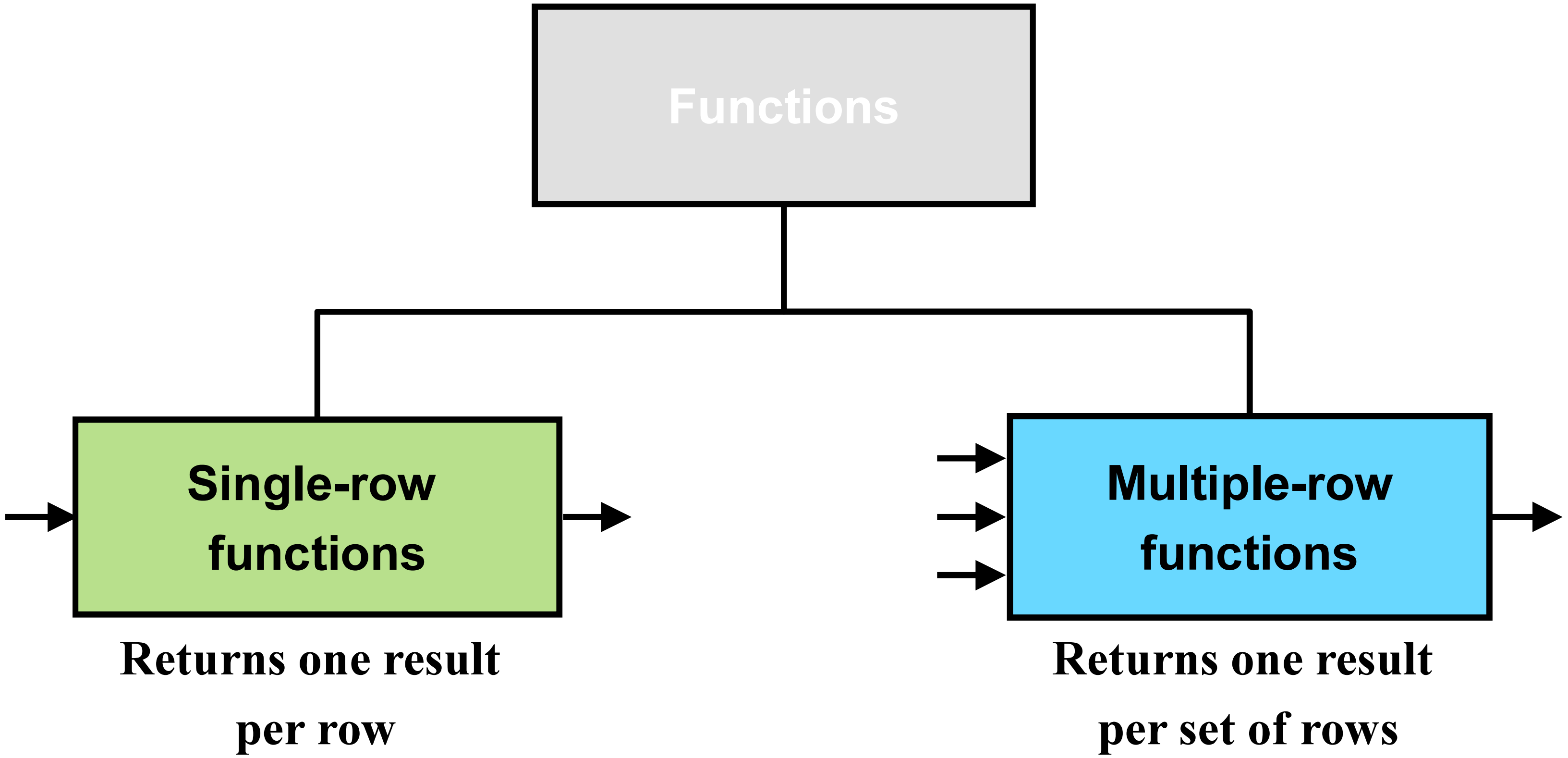
Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates
- Date functions

SQL Functions



Two Types of SQL Functions



Returns one result per row

Returns one result per set of rows

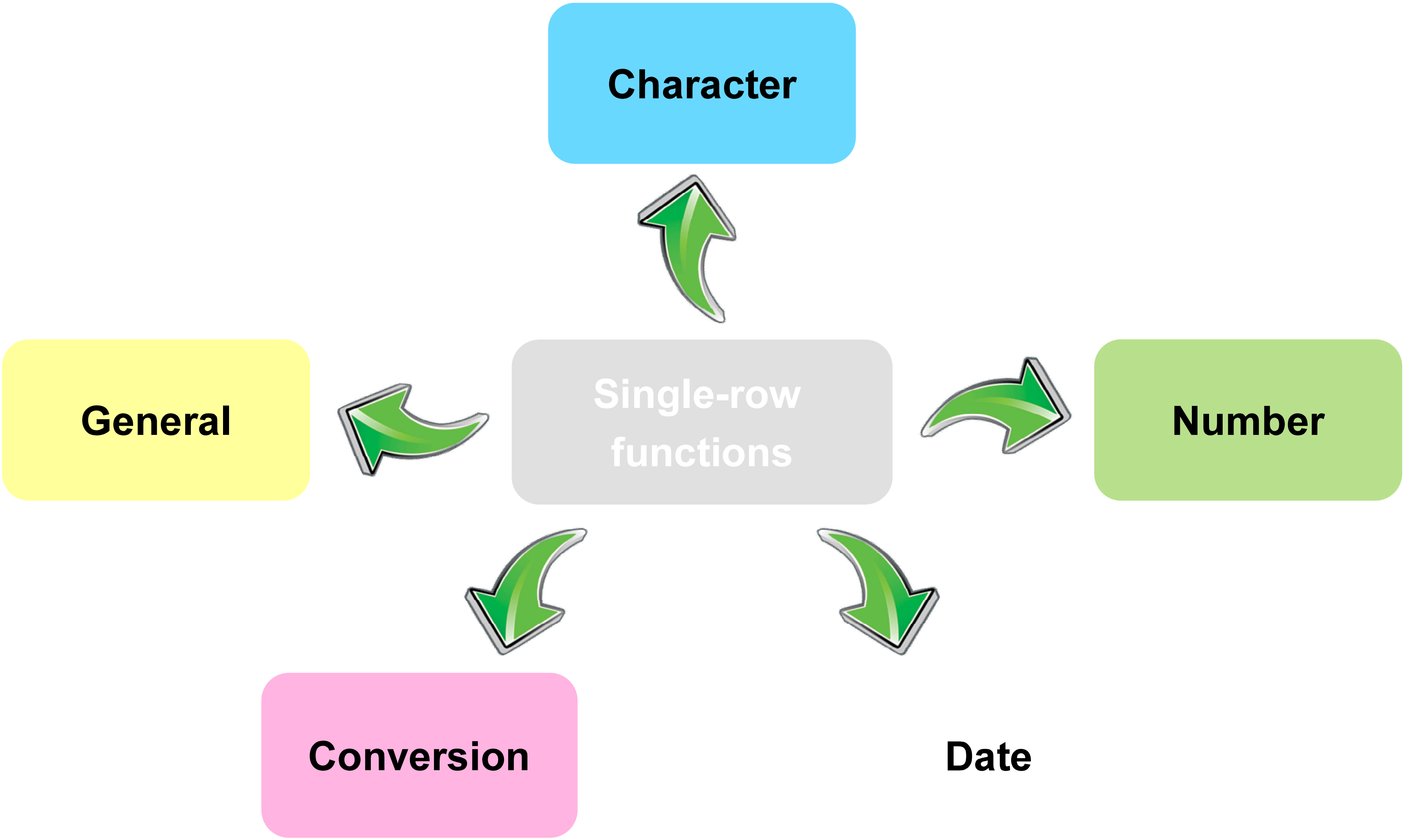
Single-Row Functions

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

```
function_name [(arg1, arg2,...)]
```

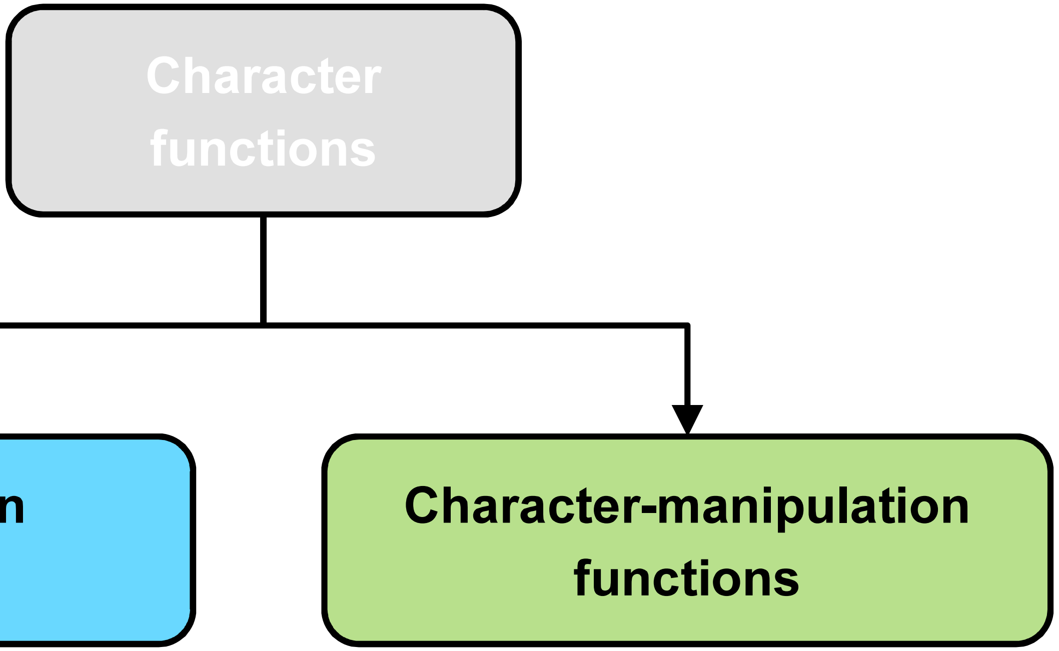
Single-Row Functions



Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates
- Date functions

Character Functions



Character functions

Case-conversion functions

Character-manipulation functions

LOWER
UPPER
INITCAP

CONCAT
SUBSTR
LENGTH
INSTR
LPAD | RPAD
TRIM
REPLACE

Case-Conversion Functions

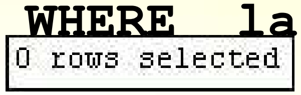
You can use these functions to convert the case of character strings:

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

Using Case-Conversion Functions

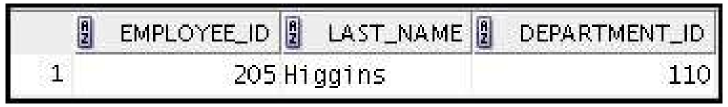
Display the employee number, name, and department number for employee Higgins:

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



```
0 rows selected
```

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



	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	205	Higgins	110

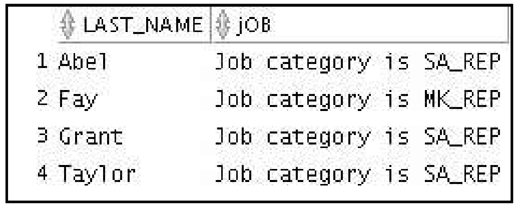
Character-Manipulation Functions

You can use these functions to manipulate character strings:

Function	Result
`CONCAT('Hello', 'World')`	`HelloWorld`
`SUBSTR('HelloWorld',1,5)`	`Hello`
`LENGTH('HelloWorld')`	`10`
`INSTR('HelloWorld', 'W')`	`6`
`LPAD(24000,10,'*')`	`*****24000`
`RPAD(24000, 10, '*')`	`24000*****`

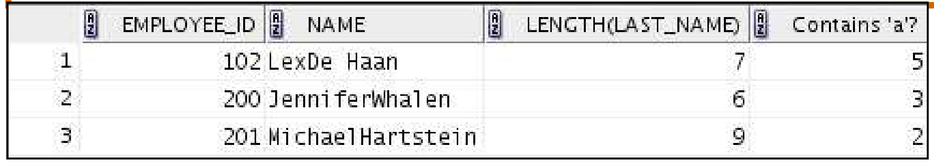
Using Character-Manipulation Functions

```
SELECT last_name, CONCAT('Job category is ', job_id)
"Job" FROM   employees
WHERE  SUBSTR(job_id, 4) = 'REP';
```



	LAST_NAME	JOB
1	Abel	Job category is SA_REP
2	Fay	Job category is MK_REP
3	Grant	Job category is SA_REP
4	Taylor	Job category is SA_REP

```
SELECT employee_id, CONCAT(first_name, last_name) NAME,
LENGTH (last_name), INSTR(last_name, 'a') "Contains 'a'?"
FROM   employees
WHERE  SUBSTR(last_name, -1, 1) = 'n';
```



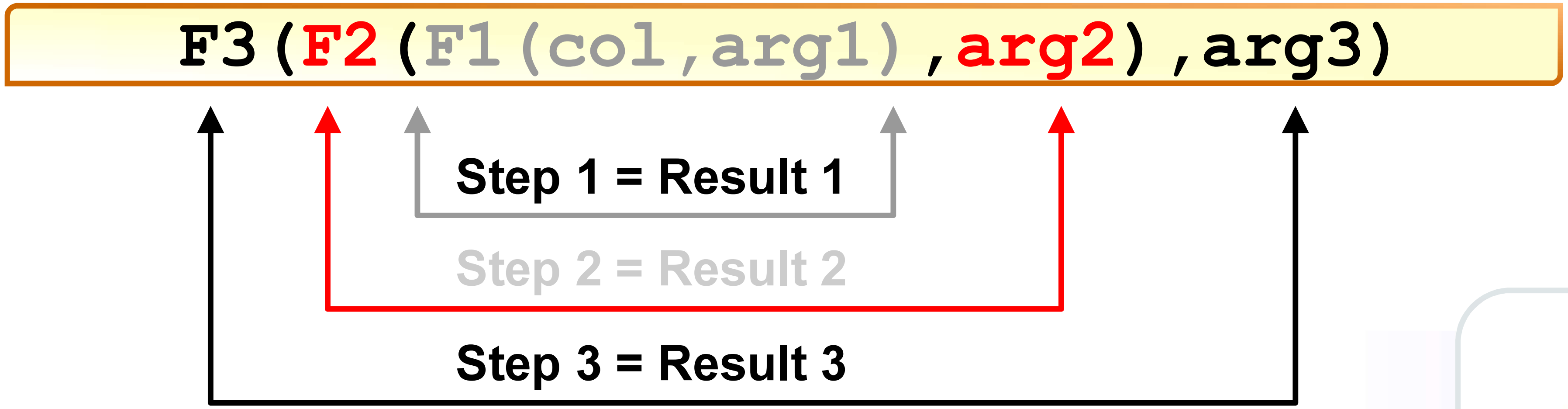
	EMPLOYEE_ID	NAME	LENGTH(LAST_NAME)	Contains 'a'?
1	102	LexDe Haan	7	5
2	200	JenniferWhalen	6	3
3	201	MichaelHartstein	9	2

Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates
- Date functions

Nesting Functions

- Single-row functions can be nested to any level.
- Nested functions are evaluated from the deepest level to the least deep level.

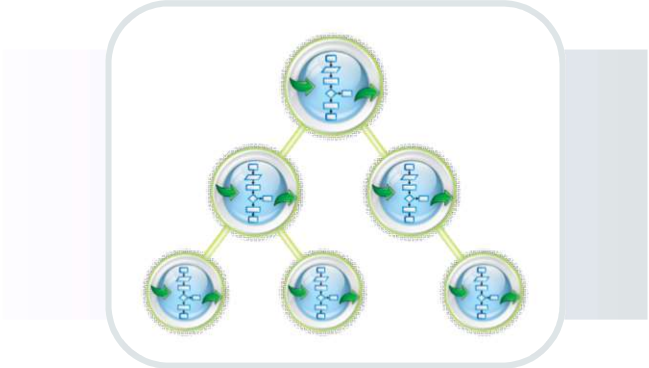


```
F3(F2(F1(col,arg1),arg2),arg3)
```

Step 1 = Result 1

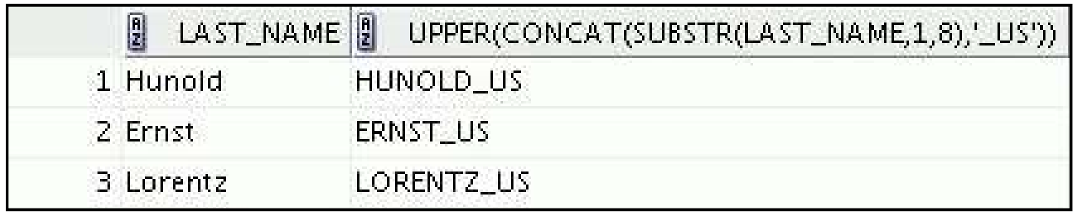
Step 2 = Result 2

Step 3 = Result 3



Nesting Functions: Example

```
SELECT last_name,
  UPPER(CONCAT(SUBSTR (LAST_NAME, 1, 8), '_US'))
FROM   employees
WHERE  department_id = 60;
```



	LAST_NAME	UPPER(CONCAT(SUBSTR(LAST_NAME,1,8),'_US'))
1	Hunold	HUNOLD_US
2	Ernst	ERNST_US
3	Lorentz	LORENTZ_US

Lesson Agenda

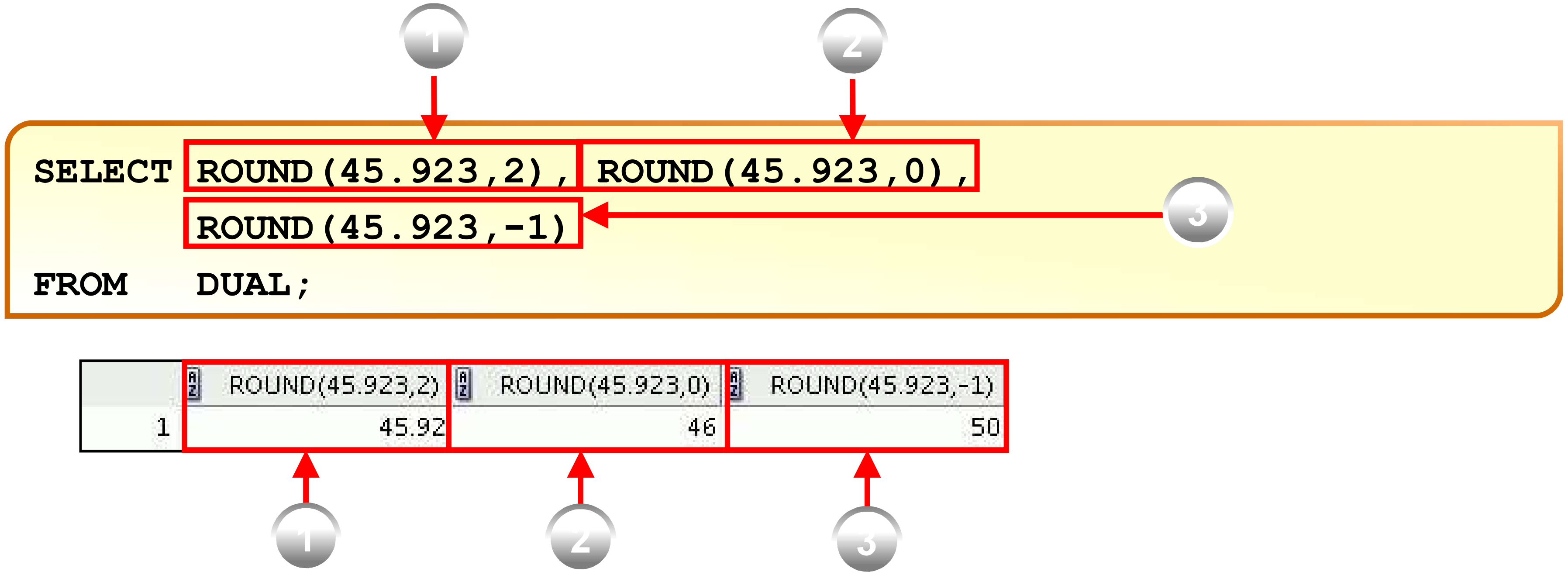
- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates
- Date Functions

Numeric Functions

- ROUND: Rounds value to a specified decimal
- TRUNC: Truncates value to a specified decimal
- CEIL: Returns the smallest whole number greater than or equal to a specified number
- FLOOR: Returns the largest whole number equal to or less than a specified number
- MOD: Returns remainder of division

Function	Result
ROUND(45.926, 2)	45.93
TRUNC(45.926, 2)	45.92
CEIL (2.83)	3
FLOOR (2.83)	2
MOD (1600, 300)	100

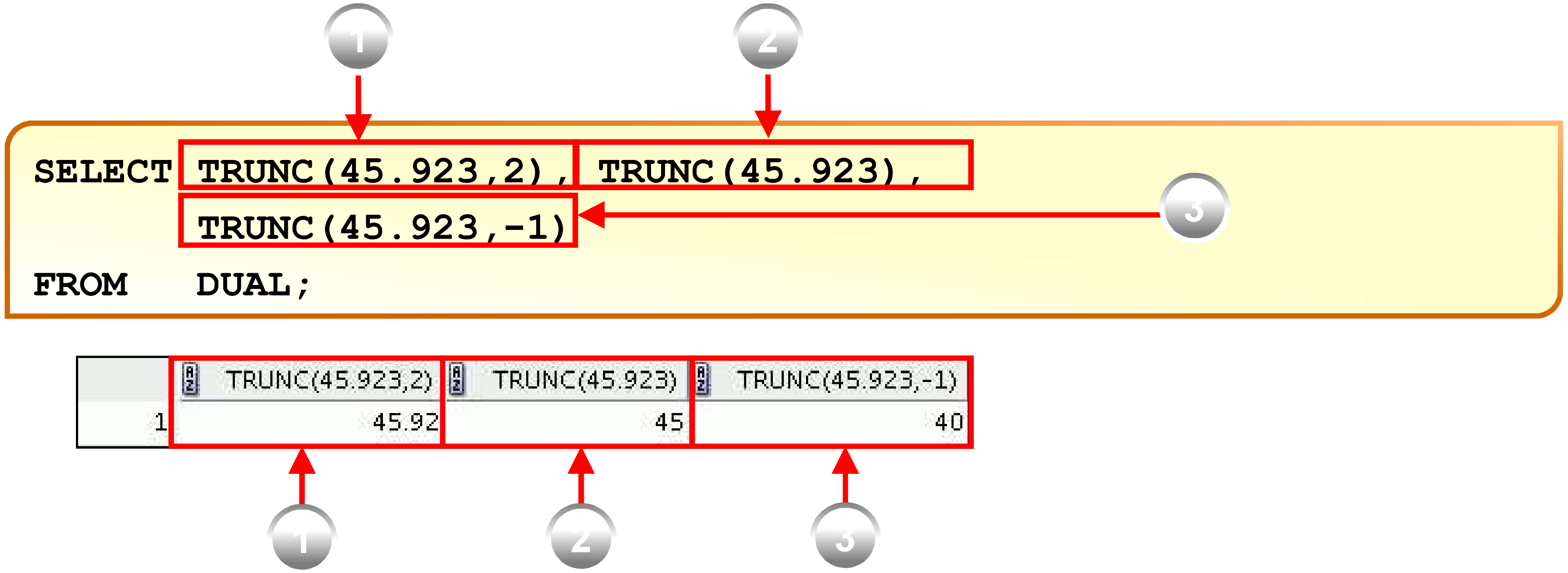
Using the ROUND Function



```
SELECT ROUND(45.923,2), ROUND(45.923,0),
       ROUND(45.923,-1)
FROM   DUAL;
```

	ROUND(45.923,2)	ROUND(45.923,0)	ROUND(45.923,-1)
1	45.92	46	50

Using the TRUNC Function



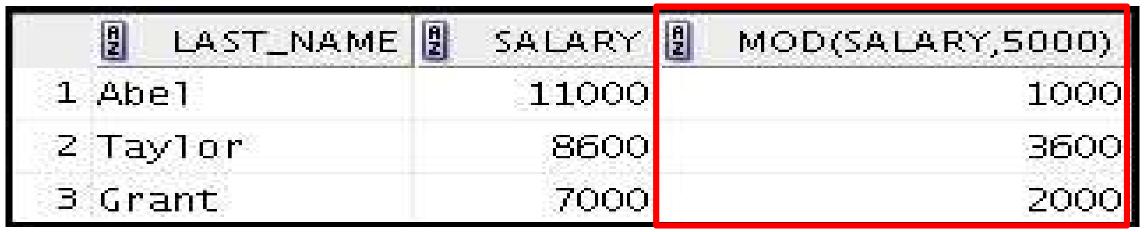
```
SELECT TRUNC(45.923,2), TRUNC(45.923),
       TRUNC(45.923,-1)
FROM   DUAL;
```

	TRUNC(45.923,2)	TRUNC(45.923)	TRUNC(45.923,-1)
1	45.92	45	40

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.

```
SELECT last_name, salary, MOD(salary, 5000)
FROM   employees
WHERE  job_id = 'SA_REP';
```



```
    LAST_NAME   SALARY   MOD(SALARY,5000)
  1 Abel         11000               1000
  2 Taylor        8600               3600
  3 Grant         7000               2000
```

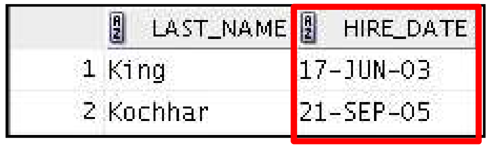
Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates
- Date functions

Working with Dates

- The Oracle Database stores dates in an internal numeric format: century, year, month, day, hours, minutes, and seconds.
- The default date display format is DD-MON-RR.
  - Enables you to store 21st-century dates in the 20th century by specifying only the last two digits of the year
  - Enables you to store 20th-century dates in the 21st century in the same way

```sql
SELECT last_name, hire_date
FROM   employees
WHERE  hire_date < '01-FEB-08';
```



```
     LAST_NAME   HIRE_DATE
   1 King        17-JUN-03
   2 Kochhar     21-SEP-05
```

...

RR Date Format

Current Year	Specified Date	RR Format	YY Format
1995	27-OCT-95	1995	1995
1995	27-OCT-17	2017	1917
2001	27-OCT-17	2017	2017
2001	27-OCT-95	1995	2095

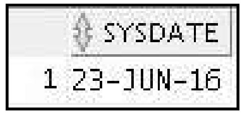
		If the specified two-digit year is:	
		0–49	**50–99**
If two digits of the current year are:	**0–49**	The return date is in the current century.	The return date is in the century before the current one.
	50–99	The return date is in the century after the current one.	The return date is in the current century.

Using the SYSDATE Function

Use the SYSDATE function to get:

- Date
- Time

```
SELECT sysdate
FROM   dual;
```

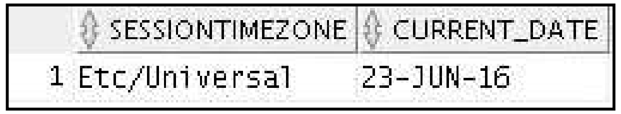


```
  SYSDATE
1 23-JUN-16
```

Using the CURRENT_ DATE and

- CURRENT_DATE returns the current date from the user session.

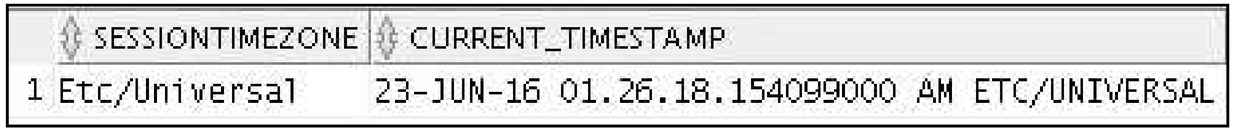
```
SELECT SESSIONTIMEZONE, CURRENT_DATE FROM DUAL;
```



```
    SESSIONTIMEZONE  CURRENT_DATE
  1 Etc/Universal    23-JUN-16
```

- CURRENT_TIMESTAMP returns the current date and time from the user session.

```
SELECT SESSIONTIMEZONE, CURRENT_TIMESTAMP FROM DUAL;
```



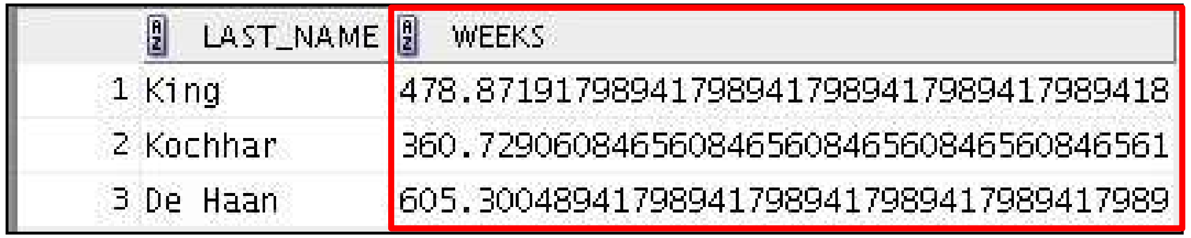
```
    SESSIONTIMEZONE  CURRENT_TIMESTAMP
  1 Etc/Universal    23-JUN-16 01.26.18.154099000 AM ETC/UNIVERSAL
```

Arithmetic with Dates

- Add to or subtract a number 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

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



	LAST_NAME	WEEKS
1	King	478.871917989417989417989417989417989418
2	Kochhar	360.729060846560846560846560846560846561
3	De Haan	605.300489417989417989417989417989417989

Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates
- Date functions

Date-Manipulation Functions

Function	Result
MONTHS_BETWEEN	Number of months between two dates
ADD_MONTHS	Add calendar months to date
NEXT_DAY	Date of the next occurrence of the specified day
LAST_DAY	Last day of the month
ROUND	Round date
TRUNC	Truncate date

Using Date Functions

Function	Result
`MONTHS_BETWEEN ('01-SEP-16','11-JAN-15')`	`19.6774194`
`ADD_MONTHS ('31-JAN-16',1)`	`'29-FEB-16'`
`NEXT_DAY ('01-JUN-16','FRIDAY')`	`'03-JUN-16'`
`LAST_DAY ('01-APR-16')`	`'30-APR-16'`

Using ROUND and TRUNC Functions with Dates

Assumption: The date when the below functions were run was **08-JUL-16**.

Function	Result
`ROUND(SYSDATE,'MONTH')`	`01-JUL-16`
`ROUND(SYSDATE,'YEAR')`	`01-JAN-17`
`TRUNC(SYSDATE,'MONTH')`	`01-JUL-16`
`TRUNC(SYSDATE,'YEAR')`	`01-JAN-16`

Quiz

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

a. Manipulate data items

b. Accept arguments and return one value per argument

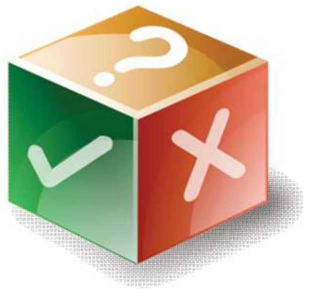
c. Act on each row that is returned

d. Return one result per set of rows

e. Never modify the data type

f. Can be nested

g. Accept arguments that can be a column or an expression



Summary

In this lesson, you should have learned how to:

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

Practice 3-Part I : Overview

This practice covers the following topics:

- Writing a query that displays the `SYSDATE`
- Creating queries that require the use of numeric, character, and date functions
- Performing calculations of years and months of service for an employee