

Using Single-Row Functions to Customize 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

HR Application Scenario

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



Zhen

HR Application

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

Operation:

Average (Salary)

GO

HR Application

The average salary is \$10500.



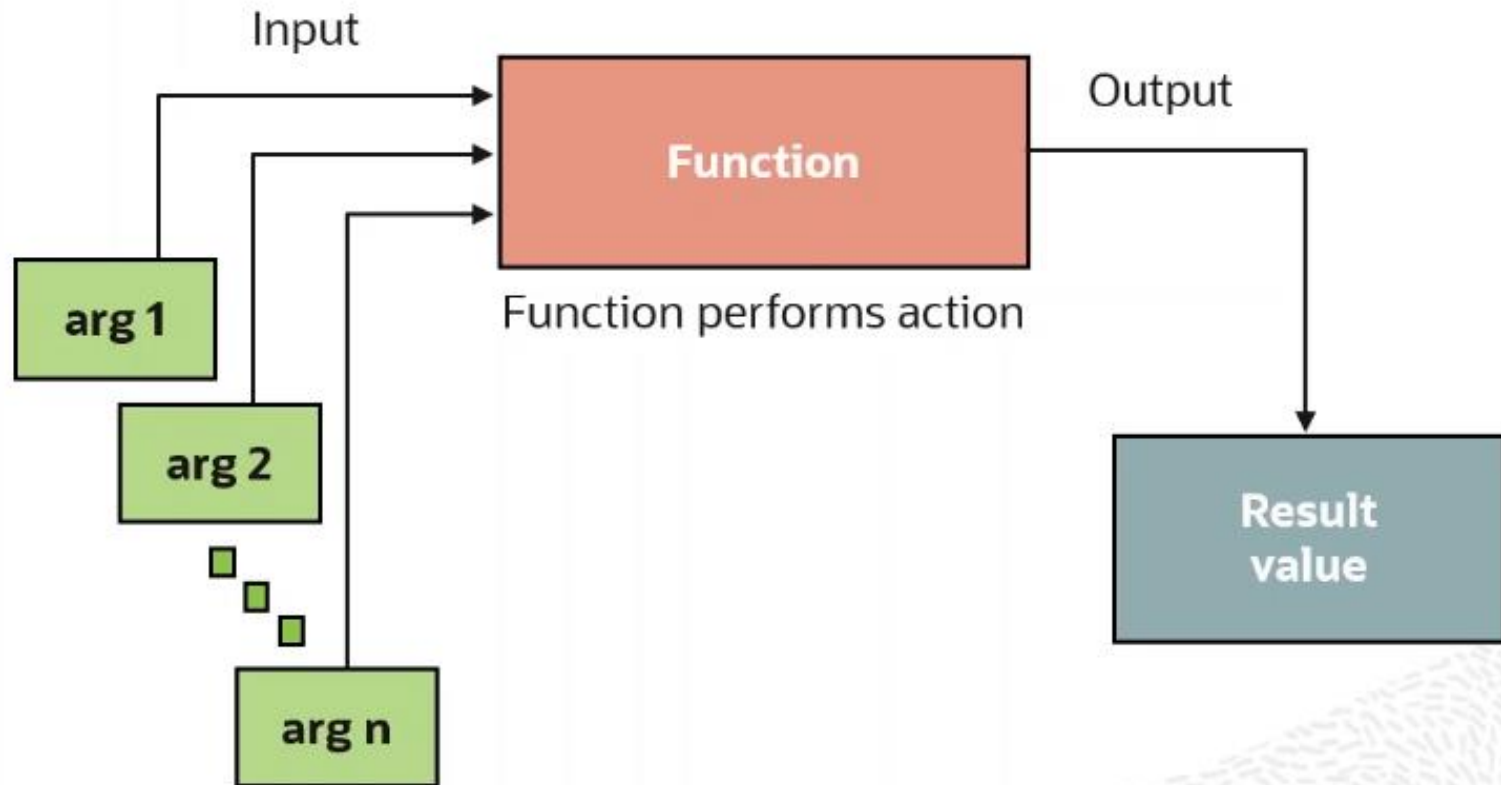
- Accounts
- IT
- Sales
- Marketing

Lesson Agenda

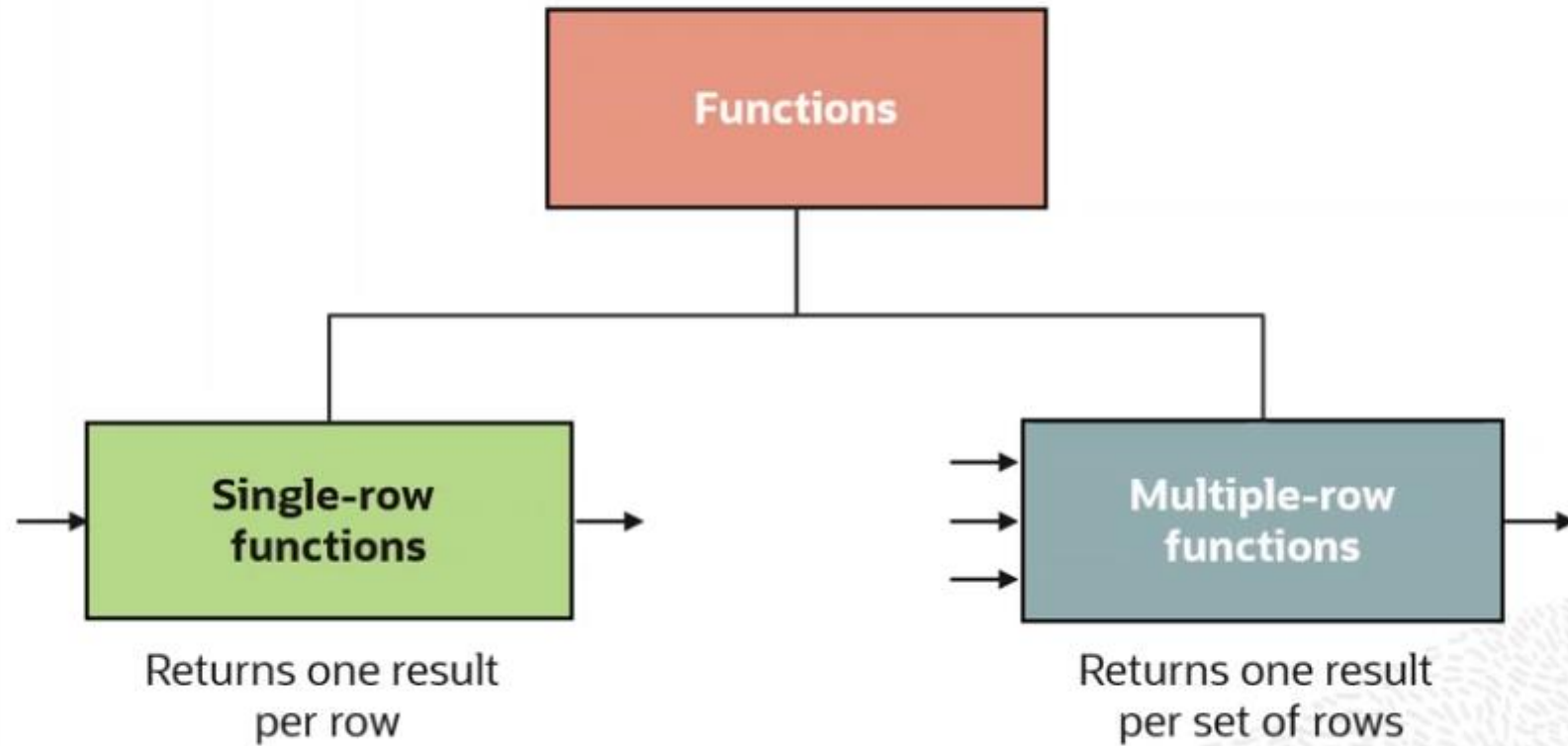


- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions

SQL Functions



Two Types of SQL Functions



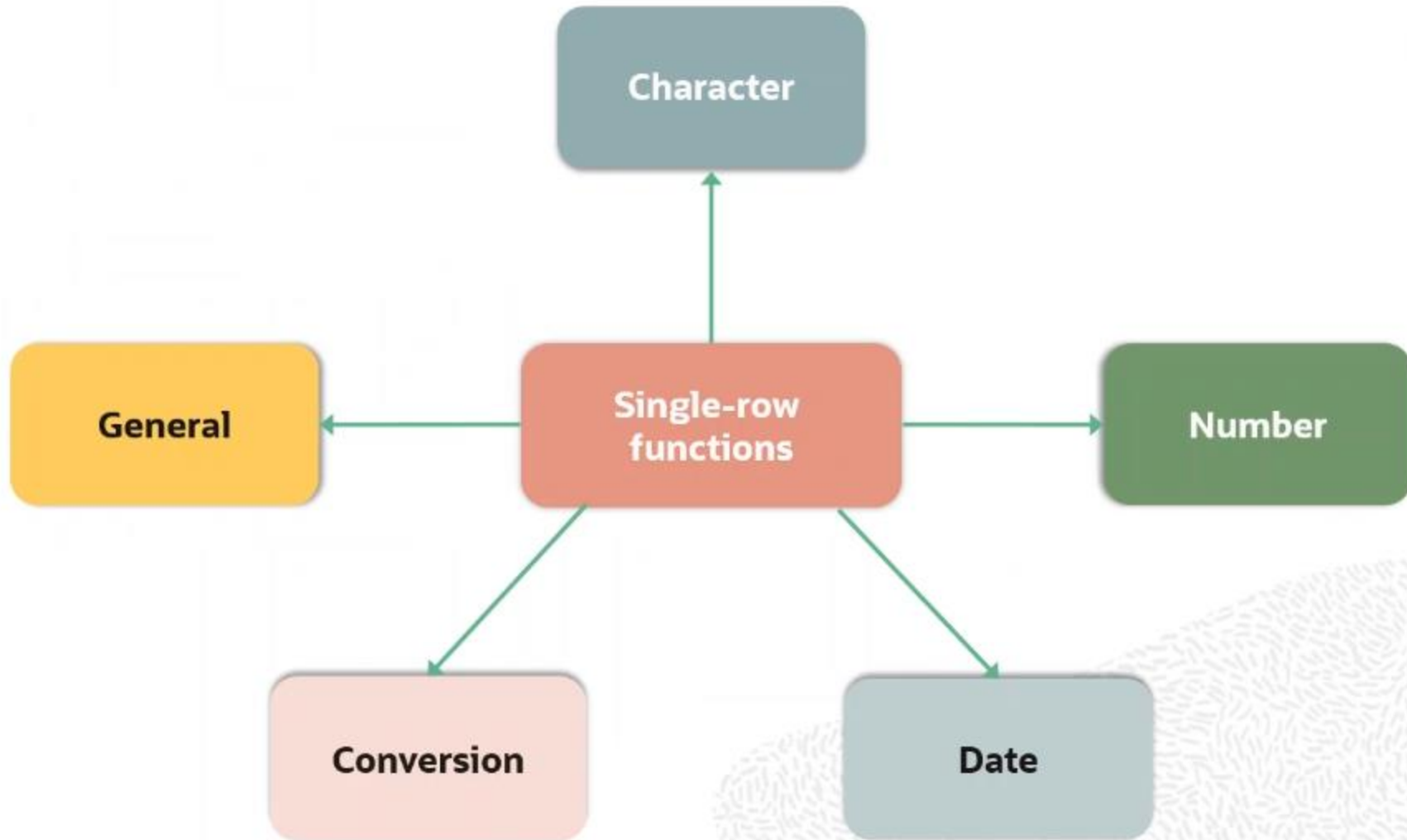
Single-Row Functions

Single-row functions:

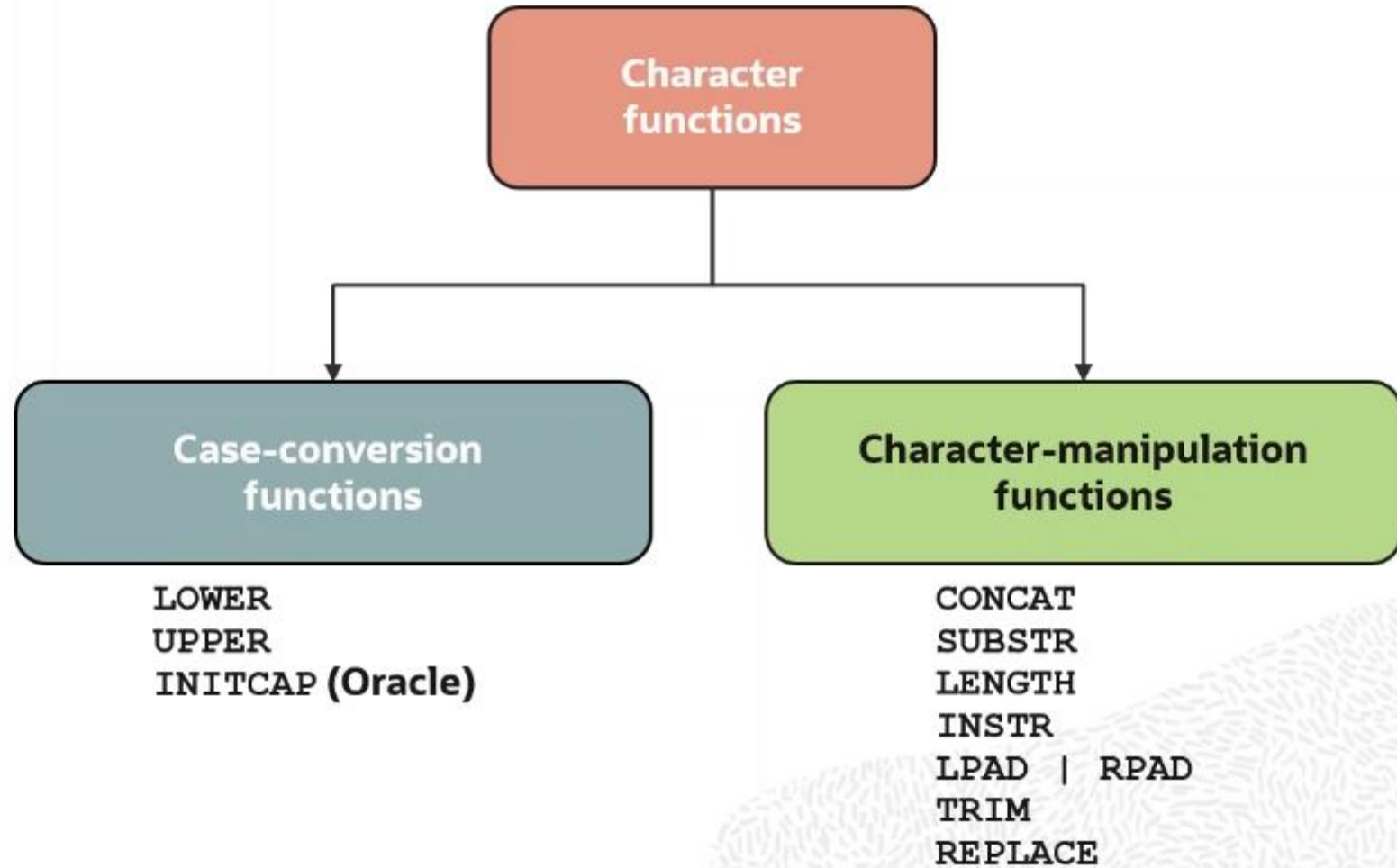
- Manipulate data items
- Accept arguments and return one value
- Act on each row that is returned
- Return one result per row
- Might modify the data type
- Can be nested
- Accept arguments that can be a column or an expression

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

Single-Row Functions



Character Functions



Case-Conversion Functions

You can use the `LOWER` and `UPPER` functions to convert the case of character strings. For example:

```
SELECT last_name, UPPER(last_name), job_id, LOWER(job_id)
FROM   employees
WHERE  department_id = 90;
```



	LAST_NAME	UPPER(LAST_NAME)	JOB_ID	LOWER(JOB_ID)
1	King	KING	AD_PRES	ad_pres
2	Kochhar	KOCHHAR	AD_VP	ad_vp
3	De Haan	DE HAAN	AD_VP	ad_vp



#	last_name	UPPER(last_name)	job_id	LOWER(job_id)
1	King	KING	AD_PRES	ad_pres
2	Kochhar	KOCHHAR	AD_VP	ad_vp
3	De Haan	DE HAAN	AD_VP	ad_vp

Using Case-Conversion Functions in WHERE Clauses in Oracle

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
<code>CONCAT('Hello', 'World')</code>	HelloWorld
<code>SUBSTR('HelloWorld',1,5)</code>	Hello
<code>LENGTH('HelloWorld')</code>	10
<code>INSTR('HelloWorld', 'W')</code>	6
<code>LPAD(24000,10,'*')</code>	*****24000
<code>RPAD(24000, 10, '*')</code>	24000*****

Using Character-Manipulation Functions

1

```
SELECT last_name, CONCAT('Job category is ', job_id)
AS 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



#	last_name	Job
1	Abel	Job category is SA_REP
2	Taylor	Job category is SA_REP
3	Grant	Job category is SA_REP
4	Fay	Job category is MK_REP

2

```
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



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

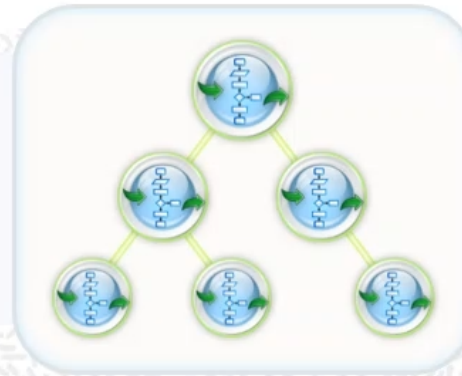
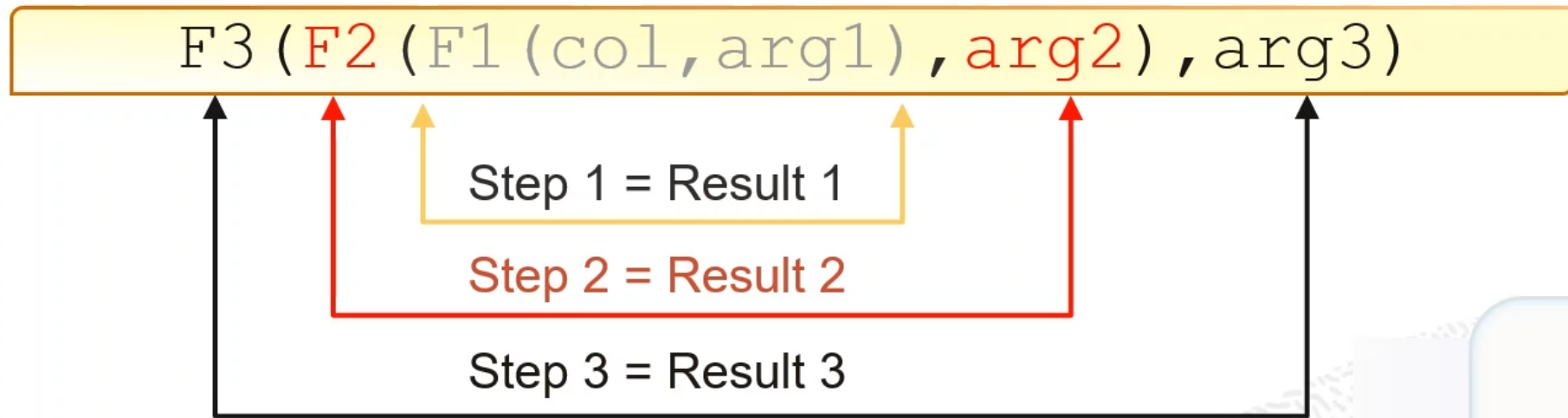
Lesson Agenda



- Single-row SQL functions
- Character functions
- Nesting 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.



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



#	last_name	UPPER(CONCAT(SUBSTR(LAST_I
1	Hunold	HUNOLD_US
2	Ernst	ERNST_US
3	Lorentz	LORENTZ_US

Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions


Numeric Functions

- **ROUND:** Rounds value to a specified decimal
- **TRUNC:** (Oracle) or **TRUNCATE:** (MySQL) 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
TRUNCATE (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 ;



	1	2	3
1	ROUND(45.923,2) 45.92	ROUND(45.923,0) 46	ROUND(45.923,-1) 50



#	1	2	3
1	ROUND(45.923,2) 45.92	ROUND(45.923,0) 46	ROUND(45.923,-1) 50



Using the TRUNC Function in Oracle

1 2

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

3



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

1

2

3

Using the MOD Function

Display the employee records where the employee_id is an even number:

```
SELECT employee_id AS Even_Numbers, last_name
FROM employees
WHERE MOD(employee_id,2) = 0;
```



	⚡ EVEN_NUMBERS	⚡ LAST_NAME
1	174	Abel
2	142	Davies
3	102	De Haan
4	104	Ernst
5	202	Fay
6	206	Gietz
7	178	Grant
8	100	King
9	124	Mourgos
10	176	Taylor
11	144	Vargas
12	200	Whalen



#	Even_Numbers	last_name
1	174	Abel
2	142	Davies
3	102	De Haan
4	104	Ernst
5	202	Fay
6	206	Gietz
7	178	Grant
8	100	King
9	124	Mourgos
10	176	Taylor
11	144	Vargas
12	200	Whalen



Working with Dates in Oracle Databases

- 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

```
SELECT last_name, hire_date
FROM employees
WHERE hire_date < '01-FEB-2013';
```



	LAST_NAME	HIRE_DATE
1	King	17-JUN-11
2	Kochhar	21-SEP-09
3	De Haan	13-JAN-09
...		