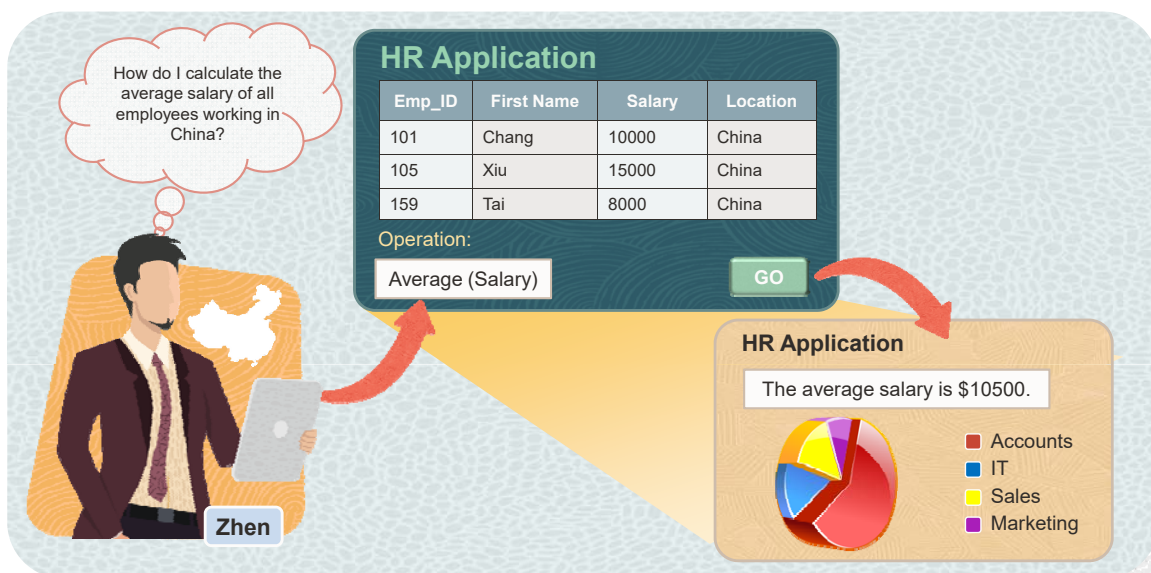


## Using Single-Row Functions to Customize Output

### HR Application Scenario



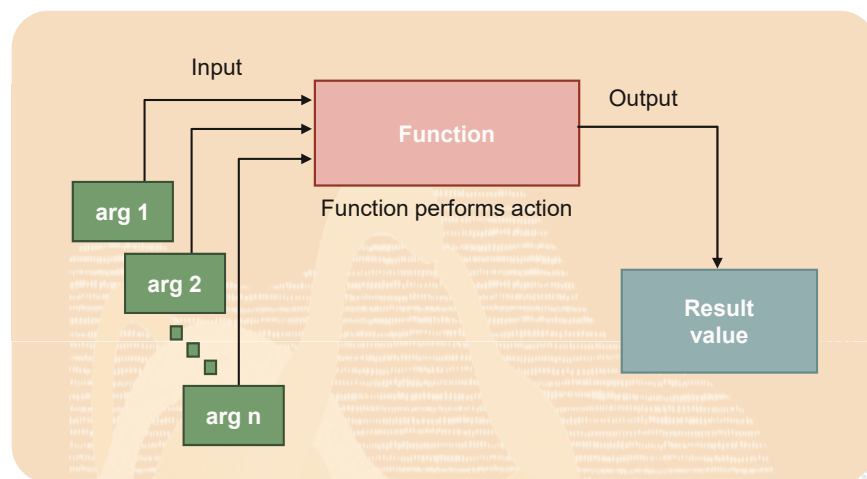
# Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates in Oracle Databases
- Working with dates in MySQL Databases
- Date functions



3

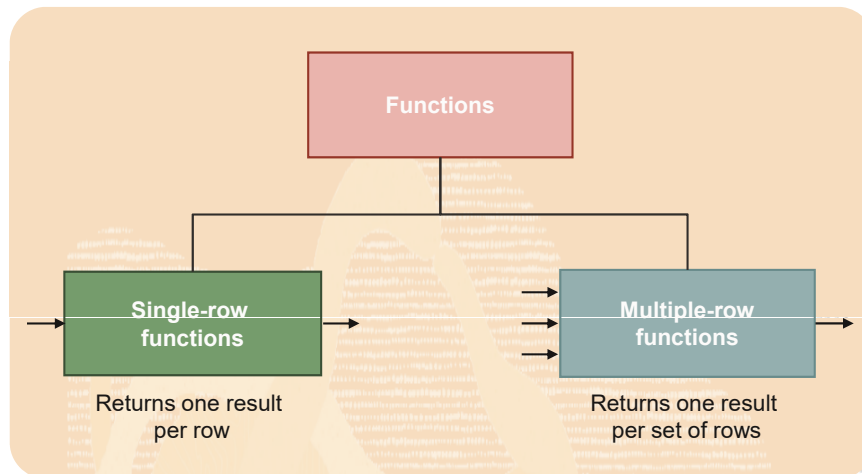
# SQL Functions



4

2

# Two Types of SQL Functions



5

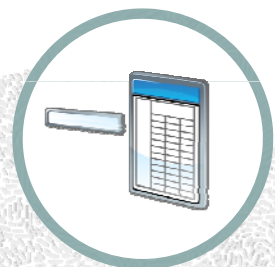


## Single-Row Functions

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

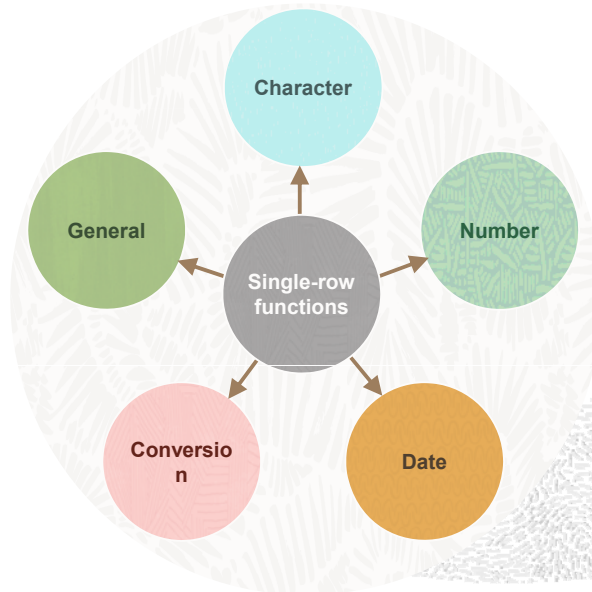
```
function_name[(arg1, arg2,...)]
```



6



# Single-Row Functions



7



# Lesson Agenda

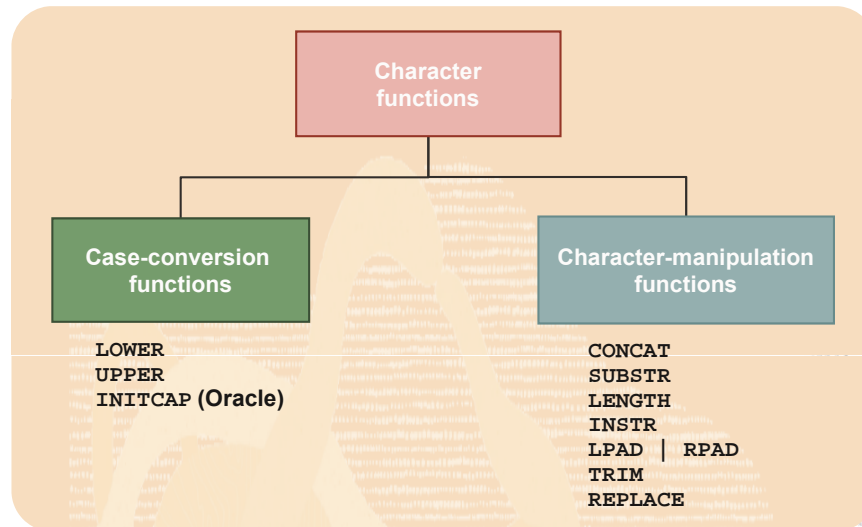
- Single-row SQL functions
- **Character functions**
- Nesting functions
- Number functions
- Working with dates in Oracle Databases
- Working with dates in MySQL Databases
- Date functions



8



# Character Functions



9



## 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

10



# 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

11



# Case-Insensitive Queries in MySQL



By default, MySQL uses a case-insensitive character set and collation. Sorting and string comparisons consider upper and lower case of the same character to be equivalent in value. To display the employee number, name, and department number for employee Higgins:

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



#	employee_id	last_name	department_id
1	205	Higgins	110
*	NULL	NULL	NULL

12





# 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*****

13



## 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

14



# Lesson Agenda

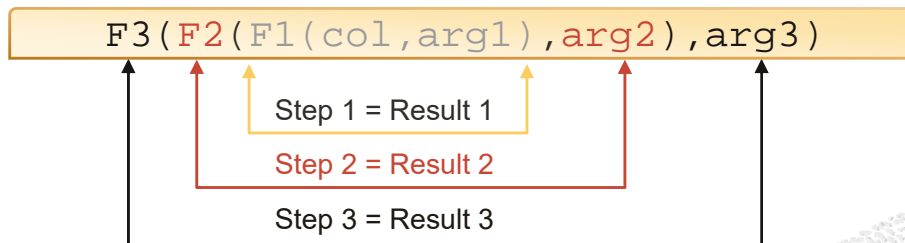
- Single-row SQL functions
- Character functions
- **Nesting functions**
- Number functions
- Working with dates in Oracle Databases
- Working with dates in MySQL Databases
- Date functions



15

# Nesting Functions

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



16



## 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

17



## Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates in Oracle Databases
- Working with dates in MySQL Databases
- Date functions



18



# 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

19




## 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

1 2 3



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

1 2 3


20



## Using the TRUNC Function in Oracle



```
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


21



## Using the TRUNCATE Function in MySQL



```
SELECT TRUNCATE(45.923,2), TRUNCATE(45.923,0),  
       TRUNCATE(45.923,-1)
```



#	TRUNCATE(45.923,2)	TRUNCATE(45.923,0)	TRUNCATE(45.923,-1)
1	45.92	45	40

22



# 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

23



## Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates in Oracle Databases
- Working with dates in MySQL Databases
- Date functions



24



# 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.

```
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

25

O

# Using the SYSDATE Function in Oracle



Use the SYSDATE function to get:

- Date
- Time

```
SELECT sysdate
FROM dual;
```



SYSDATE
1 28-JUN-18

26

O

## Using the CURRENT\_DATE and CURRENT\_TIMESTAMP Functions in Oracle



- CURRENT\_DATE returns the current date from the user session.

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



SESSIONTIMEZONE	CURRENT_DATE
1 UTC	28-JUN-18

- CURRENT\_TIMESTAMP returns the current date and time from the user session.

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



SESSIONTIMEZONE	CURRENT_TIMESTAMP
1 UTC	28-JUN-18 10.06.46.187191000 AM UTC

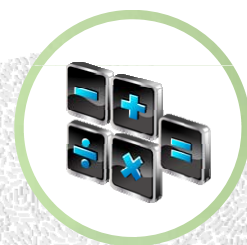
27



## Arithmetic with Dates in Oracle



- 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.



28





# Using Arithmetic Operators with Dates in Oracle



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



	LAST_NAME	WEEKS
1	King	478.871917989417989417989417989418
2	Kochhar	360.729060846560846560846560846561
3	De Haan	605.300489417989417989417989417989

29



## Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates in Oracle Databases
- Working with dates in MySQL Databases
- Date functions



30



## Working with Dates in MySQL Databases



The MySQL default date entry and display format is 'YYYY-MM-DD'. To display employees hired before February 1, 2013, enter the following query:

```
SELECT last_name, hire_date
FROM   employees
WHERE  hire_date < '2013-02-01';
```

#	last_name	hire_date
1	King	2011-06-17
2	Kochhar	2009-09-21
3	De Haan	2009-01-13
4	Rajs	2011-10-17
5	Davies	2013-01-29
6	Abel	2012-05-11
7	Whalen	2011-09-17
8	Hartstein	2012-02-17
9	Higgins	2010-06-07
10	Gietz	2010-06-07

31



## Displaying the Current Date in MySQL



The `CURDATE()` function returns the current date.

- `CURRENT_DATE()` and `CURRENT_DATE` are synonyms for `CURDATE()`.

The `NOW()` function returns the current date and time.

- `CURRENT_TIMESTAMP()` and `CURRENT_TIMESTAMP` are synonyms for `NOW()`.

The `SYSDATE()` function returns the current date and time.

```
SELECT CURDATE(), NOW(), SYSDATE();
```



#	CURDATE()	NOW()	SYSDATE()
1	2018-08-22	2018-08-22 15:33:14	2018-08-22 15:33:14

32



# Lesson Agenda

- Single-row SQL functions
- Character functions
- Nesting functions
- Number functions
- Working with dates in Oracle Databases
- Working with dates in MySQL Databases
- Date functions



33

## Date-Manipulation Functions in Oracle



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



34

## Using Date Functions in Oracle



Function	Result
MONTHS_BETWEEN ( '01-SEP-18' , '11-JAN-17' )	19.6774194
ADD_MONTHS ( '31-JAN-16' , 1 )	'29-FEB-16'
NEXT_DAY ( '01-JUN-16' , 'FRIDAY' )	'08-JUN-18'
LAST_DAY ( '01-APR-16' )	'30-APR-18'

35



## Using ROUND and TRUNC Functions with Dates in Oracle



Assumption: The below functions were run on **29-JUN-18**.

Function	Result
ROUND(SYSDATE, 'MONTH' )	01-JUL-18
ROUND(SYSDATE, 'YEAR' )	01-JAN-18
TRUNC(SYSDATE, 'MONTH' )	01-JUN-18
TRUNC(SYSDATE, 'YEAR' )	01-JAN-18

36



# Date-Manipulation Functions in MySQL



Function	Result
DATE_ADD(date, INTERVAL expr unit)	Date after an interval is added to a date
DATE_SUB(date, INTERVAL expr unit)	Date after an interval is subtracted from a date
DATEDIFF(date, date)	Difference in days between two dates
LAST_DAY(date)	Last day of the month
MONTH(date)	The month number of the date
YEAR(date)	The year of the date



37



## Using Date Functions in MySQL



The following example uses MySQL date functions in the output as well as in the WHERE clause:

```
SELECT employee_id, hire_date,  
       DATE_ADD(hire_date, INTERVAL 6 MONTH) AS Review,  
       DATEDIFF(CURDATE(), hire_date) AS Tenure  
FROM employees  
WHERE hire_date > DATE_SUB(CURDATE(), INTERVAL 4 YEAR);
```



#	employee_id	hire_date	Review	Tenure
1	104	2015-05-21	2015-11-21	1189
2	107	2015-02-07	2015-08-07	1292
3	124	2015-11-16	2016-05-16	1010
4	149	2016-01-29	2016-07-29	936
5	178	2015-05-24	2015-11-24	1186

38



# Extracting the Month or Year Portion of Dates in MySQL



Use the `MONTH()` or `YEAR()` function to extract those portions of a date. For example, the following query displays the employee number, hire date, and starting month for employees that started in 2010.

```
SELECT employee_id, hire_date,  
       MONTH(hire_date)  
FROM employees  
WHERE YEAR(hire_date) = '2010';
```



#	employee_id	hire_date	MONTH(hire_date)
1	205	2010-06-07	6
2	206	2010-06-07	6

39



## 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



40

