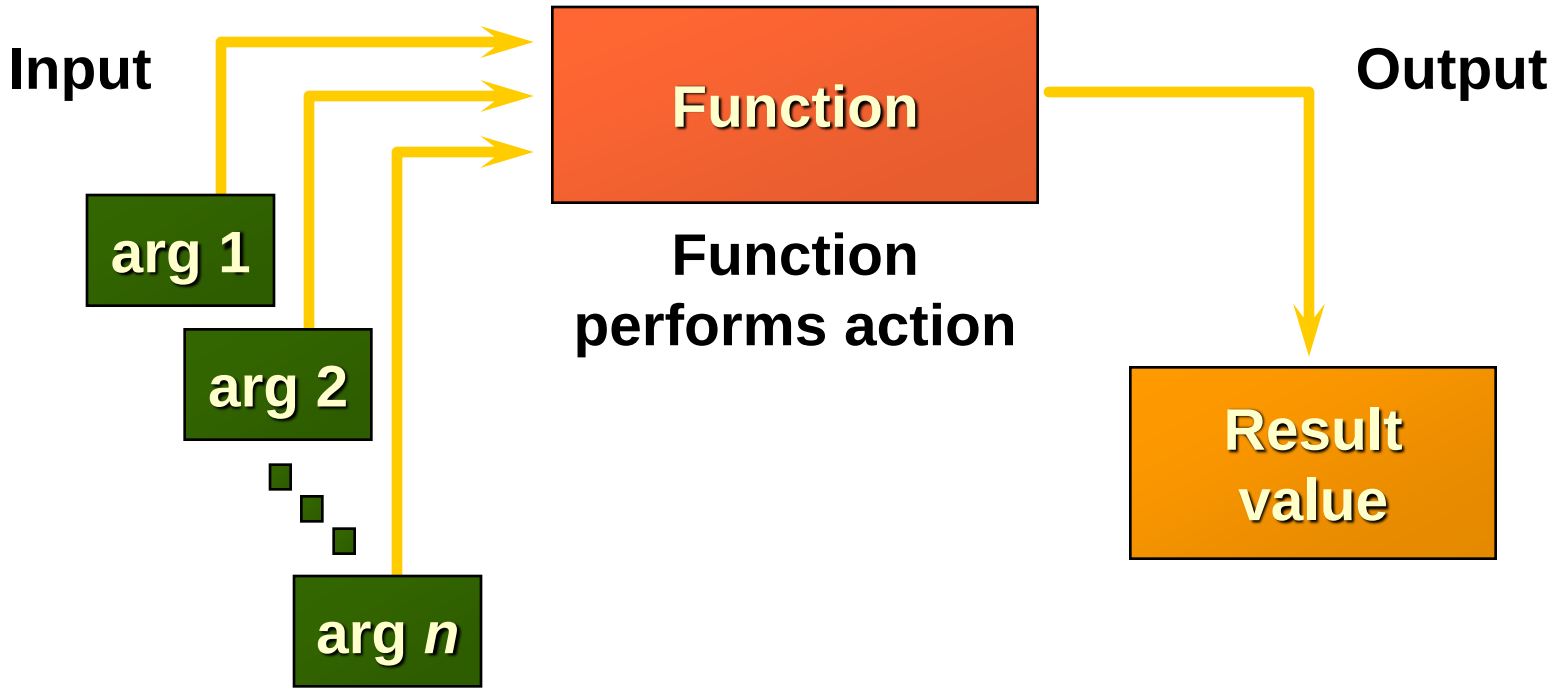
Single-Row Functions

Objectives

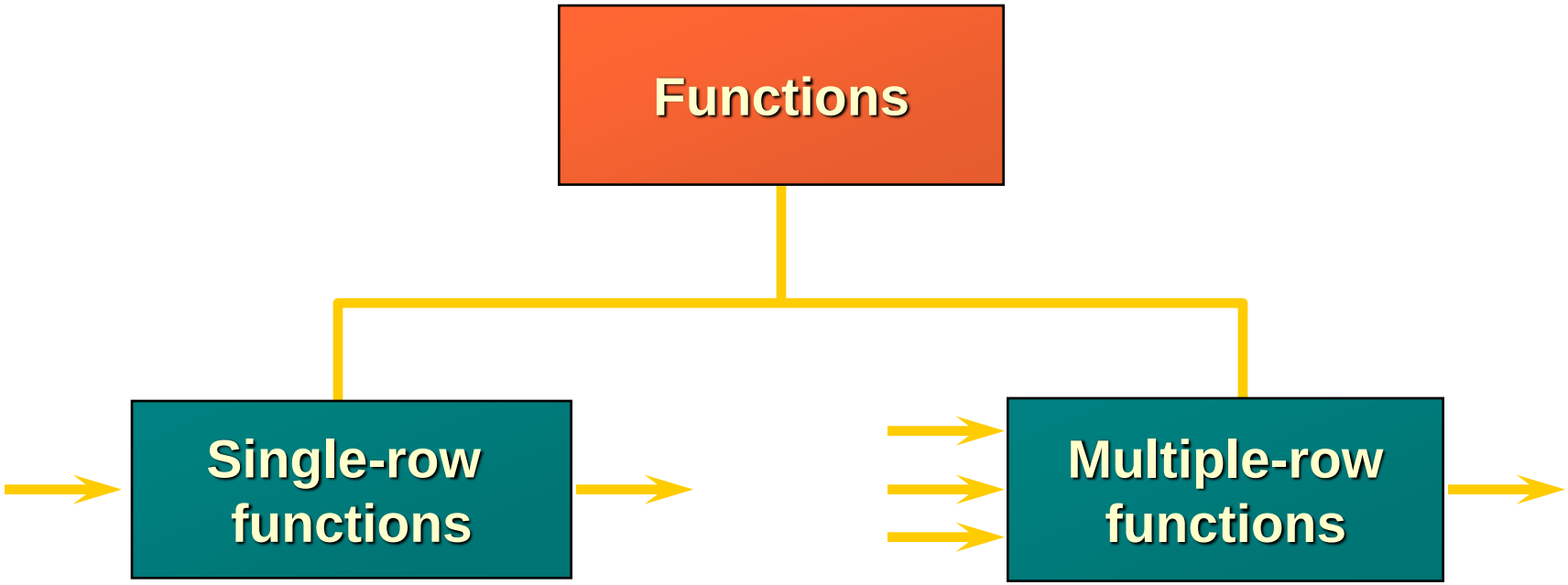
After completing this lesson, you should be able to do the following:

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

SQL Functions



Two Types of SQL Functions



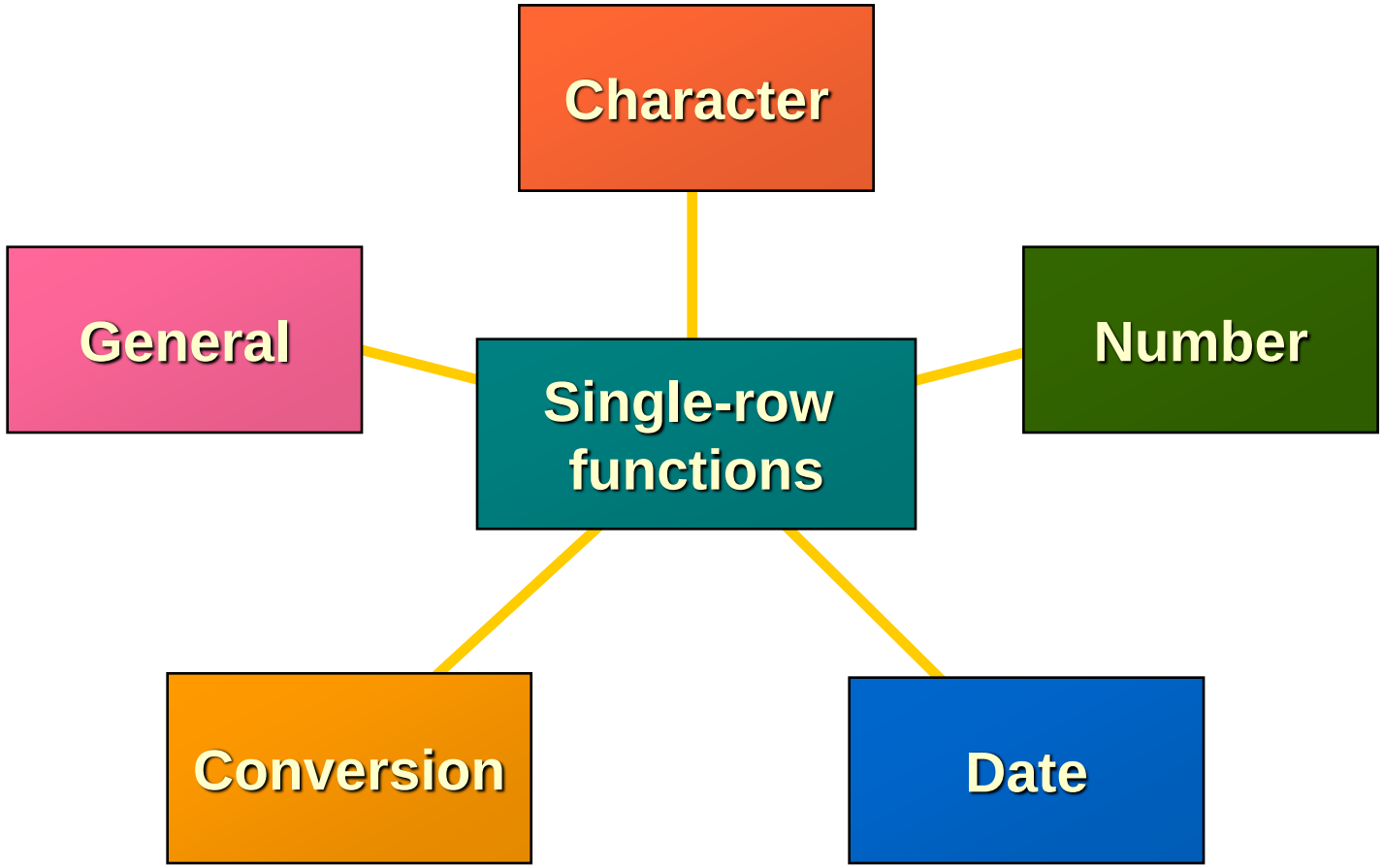
Single-Row Functions

Single row functions:

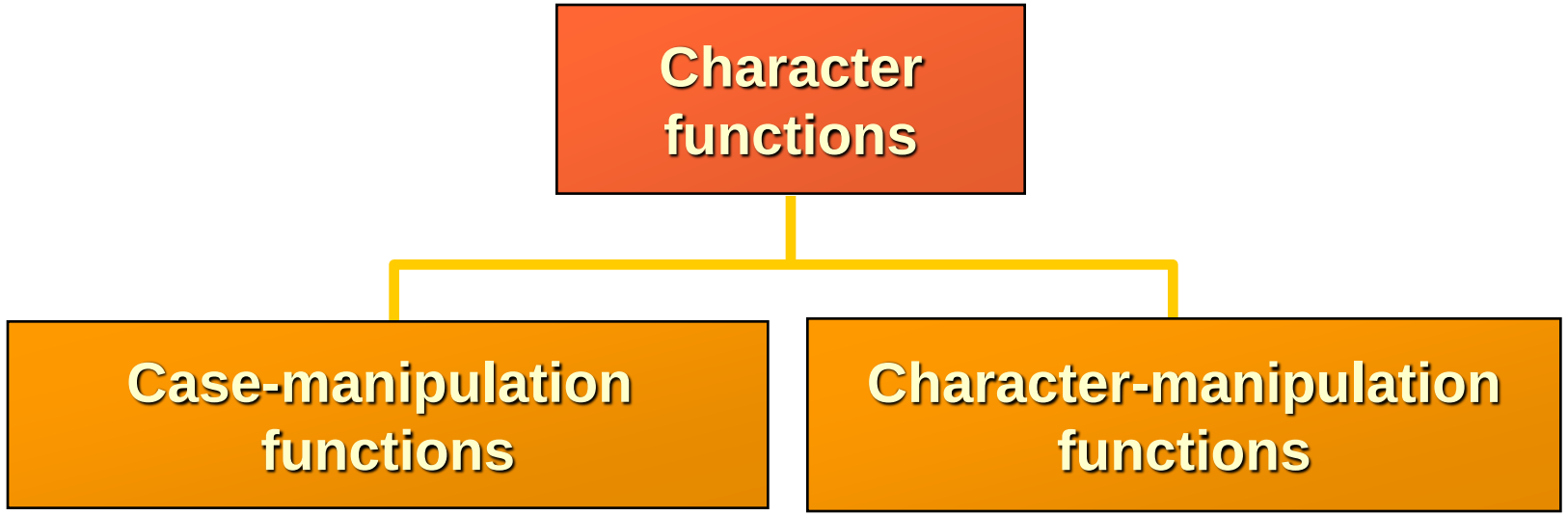
- Manipulate data items
- Accept arguments and return one value
- Act on each row returned
- Return one result per row
- May modify the data type
- Can be nested
- Accept arguments which can be a column or an expression

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

Single-Row Functions



Character Functions



Case Manipulation Functions

These functions convert case for character strings.

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

Using Case Manipulation 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';
no rows selected
```

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

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
205	Higgins	110

Character-Manipulation Functions

These functions manipulate character strings:

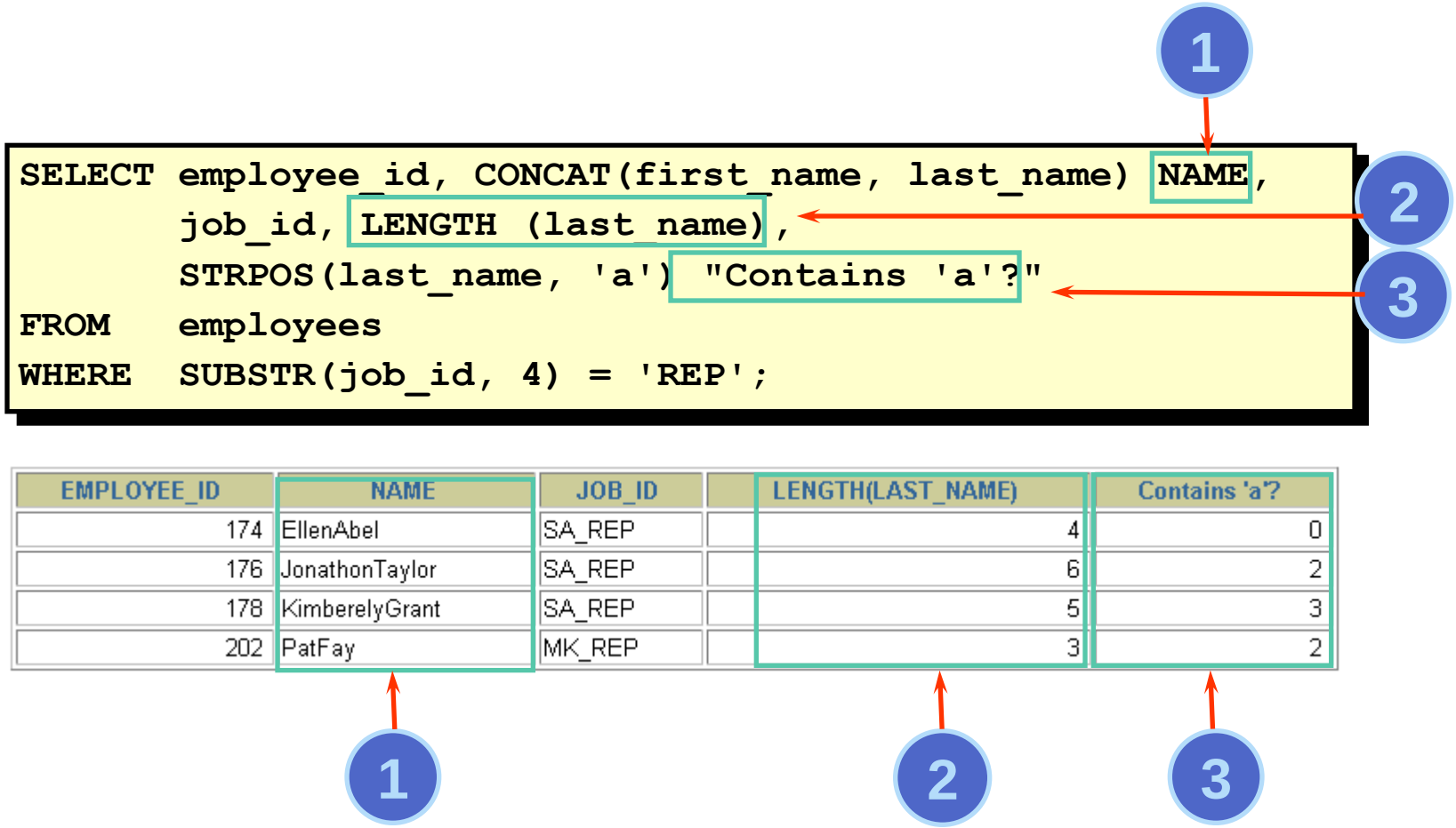
Function	Result
`CONCAT('Hello', 'World')`	`HelloWorld`
`SUBSTR('HelloWorld',1,5)`	`Hello`
`LENGTH('HelloWorld')`	`10`
`STRPOS('HelloWorld', 'W')`	`6`
`TRIM('H' FROM 'HelloWorld')`	`elloWorld`

Character-Manipulation Functions-
TRIM - for data cleaning

```
TRIM([LEADING | TRAILING | BOTH] [characters FROM] string)
```

- **LEADING:** Removes characters from the beginning of the string.
- **TRAILING**: Removes characters from the end of the string.
- **BOTH**: Removes characters from both the beginning and the end of the string (this is the default if no direction is specified).
- **characters**: Specifies which characters to remove (e.g., a specific character like 'x' or default whitespace).
- **string**: The string from which you want to remove characters

Using the Character-Manipulation Functions



Number Functions

- `ROUND`: Rounds value to specified decimal

```
ROUND(45.926, 2) ⟶ 45.93
```

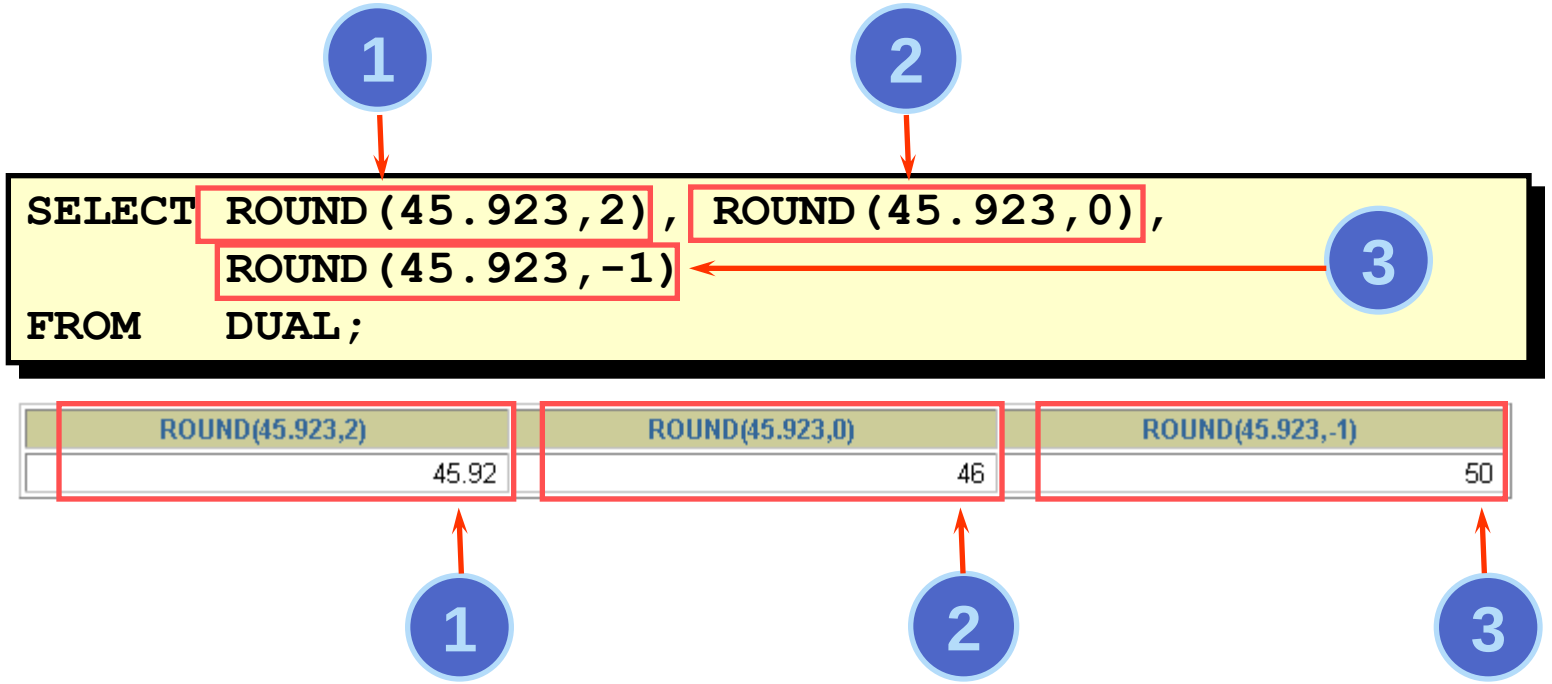
- `TRUNC`: Truncates value to specified decimal

```
TRUNC(45.926, 2) ⟶ 45.92
```

- `MOD`: Returns remainder of division

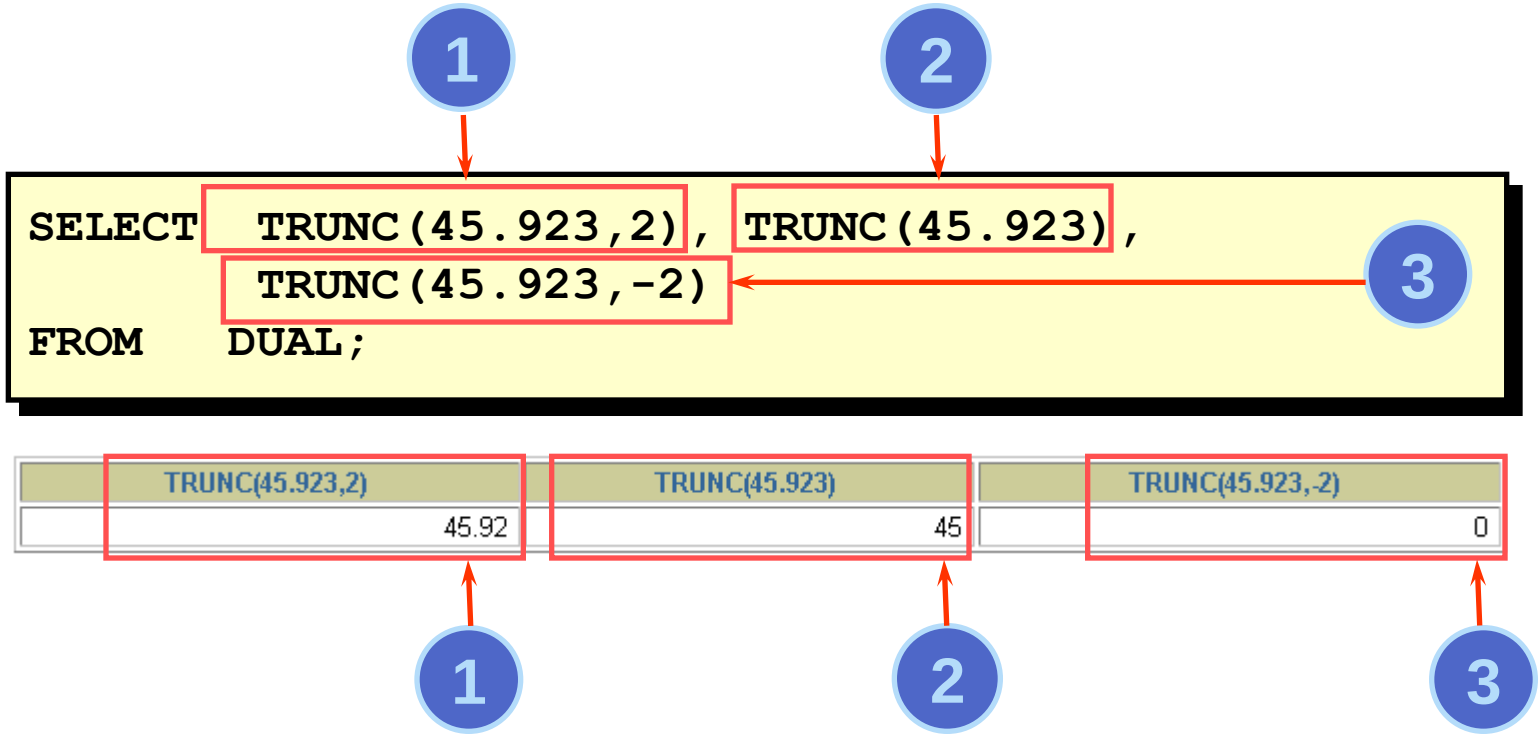
```
MOD(1600, 300) ⟶ 100
```

Using the ROUND Function



DUAL is a dummy table you can use to view results from functions and calculations.

Using the TRUNC Function



Using the MOD Function

Calculate the remainder of a salary after it is divided by 5000 for all employees whose job title is sales representative.

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

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

Working with Dates

- Format ‘yyyy-mm-dd’
- CURRENT_TIMESTAMP
- NOW() (only for pg)

```
SELECT last_name, hire_date
FROM   employees
WHERE  last_name like 'G%';
```

	last_name character varying (25)	hire_date timestamp without time zone
1	Greenberg	1994-08-17 00:00:00
2	Gee	1999-12-12 00:00:00
3	Greene	1999-03-19 00:00:00
4	Grant	1999-05-24 00:00:00
5	Geoni	2000-02-03 00:00:00
6	Gates	1998-07-11 00:00:00
7	Grant	2000-01-13 00:00:00
8	Gietz	1994-06-07 00:00:00

Using Arithmetic Operators with Dates

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

EXTRACT

```
EXTRACT(part FROM date_column)
```

Part- YEAR, MONTH, DAY, HOUR, MINUTE, SECOND,
QUARTER: Returns the quarter of the year (1 to 4)
WEEK, DOW (Day of Week

```
SELECT  last_name, hire_date,
        extract(year from hire_date) as year,
        extract(month from hire_date) as month,
        extract(day from hire_date) as day
from employees;
```

Output | Messages | Notifications

SQL

last_name character varying (25)	hire_date timestamp without time zone	year numeric	month numeric	day numeric
King	1987-06-17 00:00:00	1987	6	17
Kochhar	1989-09-21 00:00:00	1989	9	21
De Haan	1993-01-13 00:00:00	1993	1	13

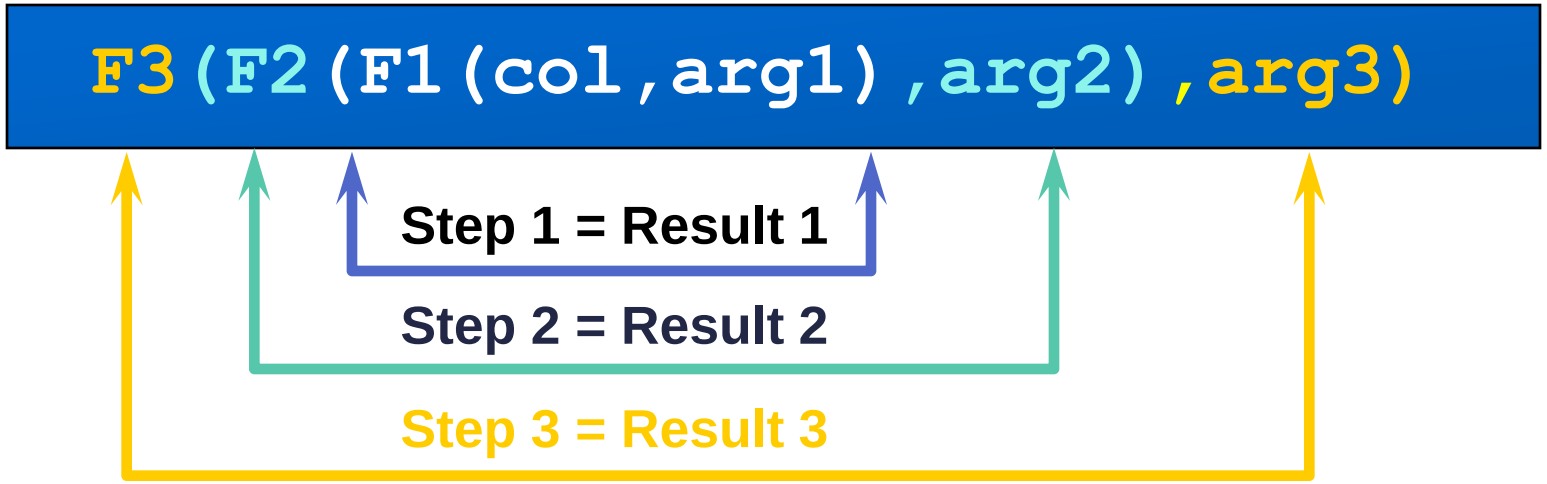
Conversion Functions

SQL Data Types

Numeric (int\integer, smallint, bigint, decimal(p,s)/numeric(p,s), float/real
Strings- char(n)-fixed length string, varchar(n)- variable-length string, text- variable length long string
Dates date, time, timestamp, datetime, interval
Boolean
Others binary(binary date as files or images), blob (binary large object like image, video or audio), UUID , xml. Json, jasonB

Nesting Functions

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



COALESCE Function

```
COALESCE(value1, value2, ..., valueN)
```

A posstgraSQL-Only function

Converts a null to an actual value.

- Works best with simple data types(such as date, character, and number)
- Data types must match:
  - `coalesce(commission_pct,0)`
  - `coalesce(hire_date,'01-JAN-97')`
  - `coalesce(job_id,'No Job Yet')`

Using the NULLIF Function

```
SELECT  first_name, LENGTH(first_name) "expr1",
        last_name,  LENGTH(last_name)  "expr2",
        NULLIF(LENGTH(first_name), LENGTH(last_name)) result
FROM    employees;
```

FIRST_NAME	expr1	LAST_NAME	expr2	RESULT
Steven	6	King	4	6
Neena	5	Kochhar	7	5
Lex	3	De Haan	7	3
Alexander	9	Hunold	6	9
Bruce	5	Ernst	5	
Diana	5	Lorentz	7	5
Kevin	5	Mourgos	7	5
Trenna	6	Rajs	4	6
Curtis	6	Davies	6	

...

20 rows selected.

The CASE Expression

Facilitates conditional inquiries by doing the work of an IF-THEN-ELSE statement:

```
CASE expr WHEN comparison_expr1 THEN return_expr1
          [WHEN comparison_expr2 THEN return_expr2
           WHEN comparison_exprn THEN return_exprn
           ELSE else_expr]
END
```

Using the CASE Expression

Facilitates conditional inquiries by doing the work of an IF-THEN-ELSE statement:

```
SELECT last_name, job_id, salary,
       CASE job_id WHEN 'IT_PROG'  THEN  1.10*salary
                   WHEN 'ST_CLERK' THEN  1.15*salary
                   WHEN 'SA_REP'   THEN  1.20*salary
       ELSE       salary END     "REVISED_SALARY"
FROM   employees;
```

LAST_NAME	JOB_ID	SALARY	REVISED_SALARY
...			
Lorentz	IT_PROG	4200	4620
Mourgos	ST_MAN	5800	5800
Rajs	ST_CLERK	3500	4025
...			
Gietz	AC_ACCOUNT	8300	8300

20 rows selected.

Summary

In this lesson, you should have learned how to:

- Perform calculations on data using functions
- Modify individual data items using functions
- Convert column data types using functions
- Use COALEASE functions
- Use IF-THEN-ELSE logic