

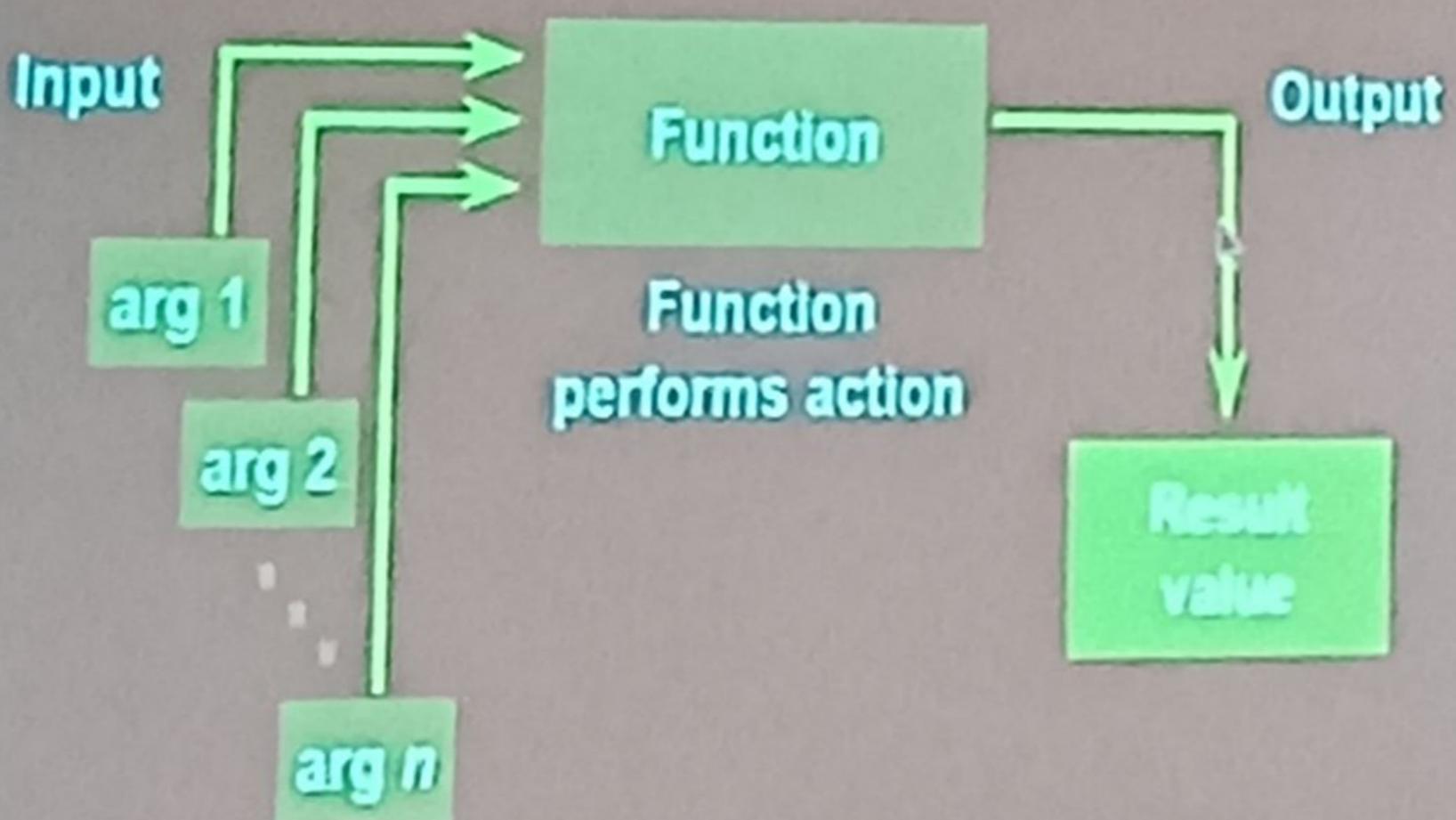
Single-Row Functions

ORACLE

Copyright © Oracle Corporation 2001. All rights reserved.



SQL Functions

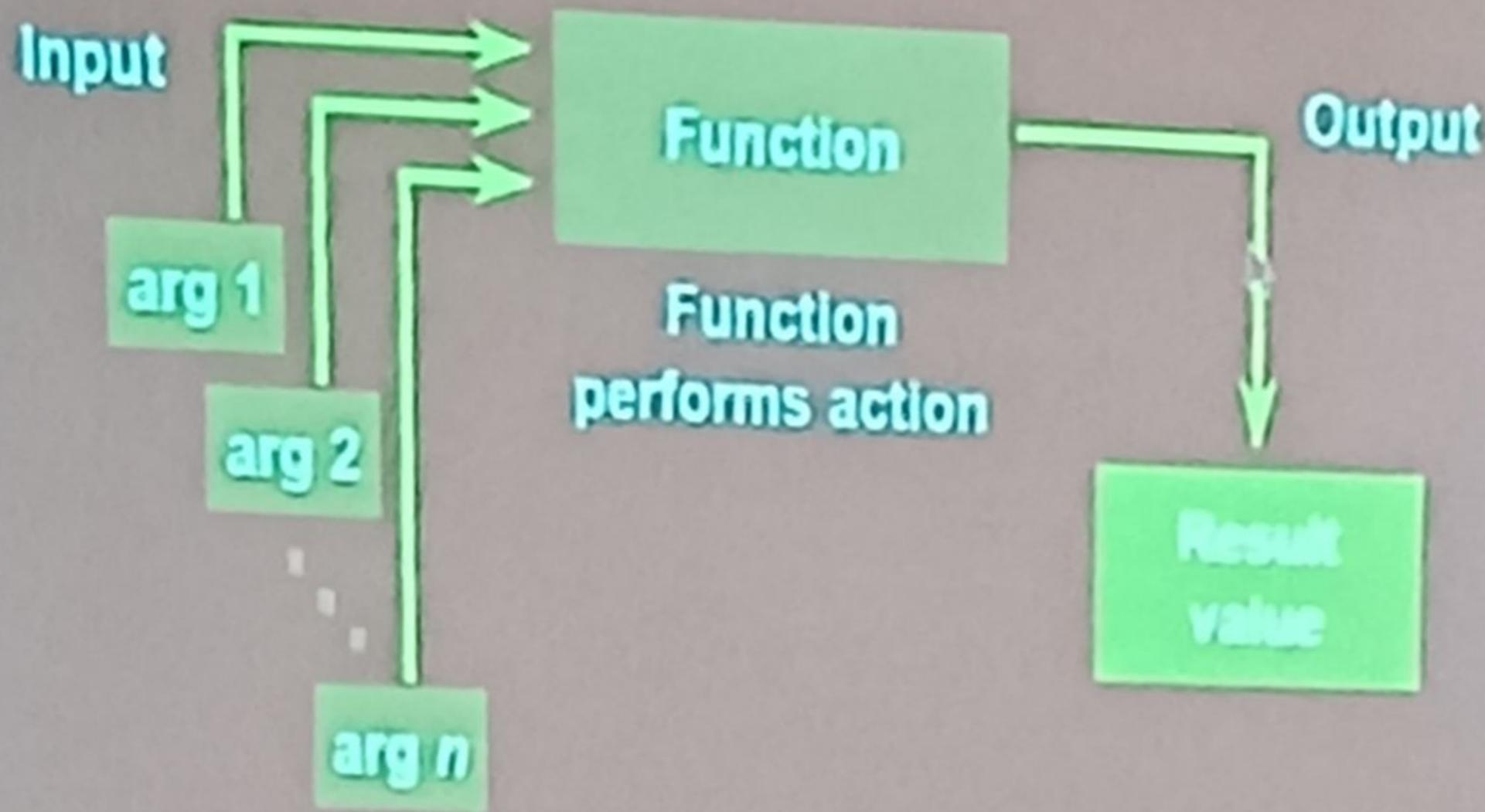


ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

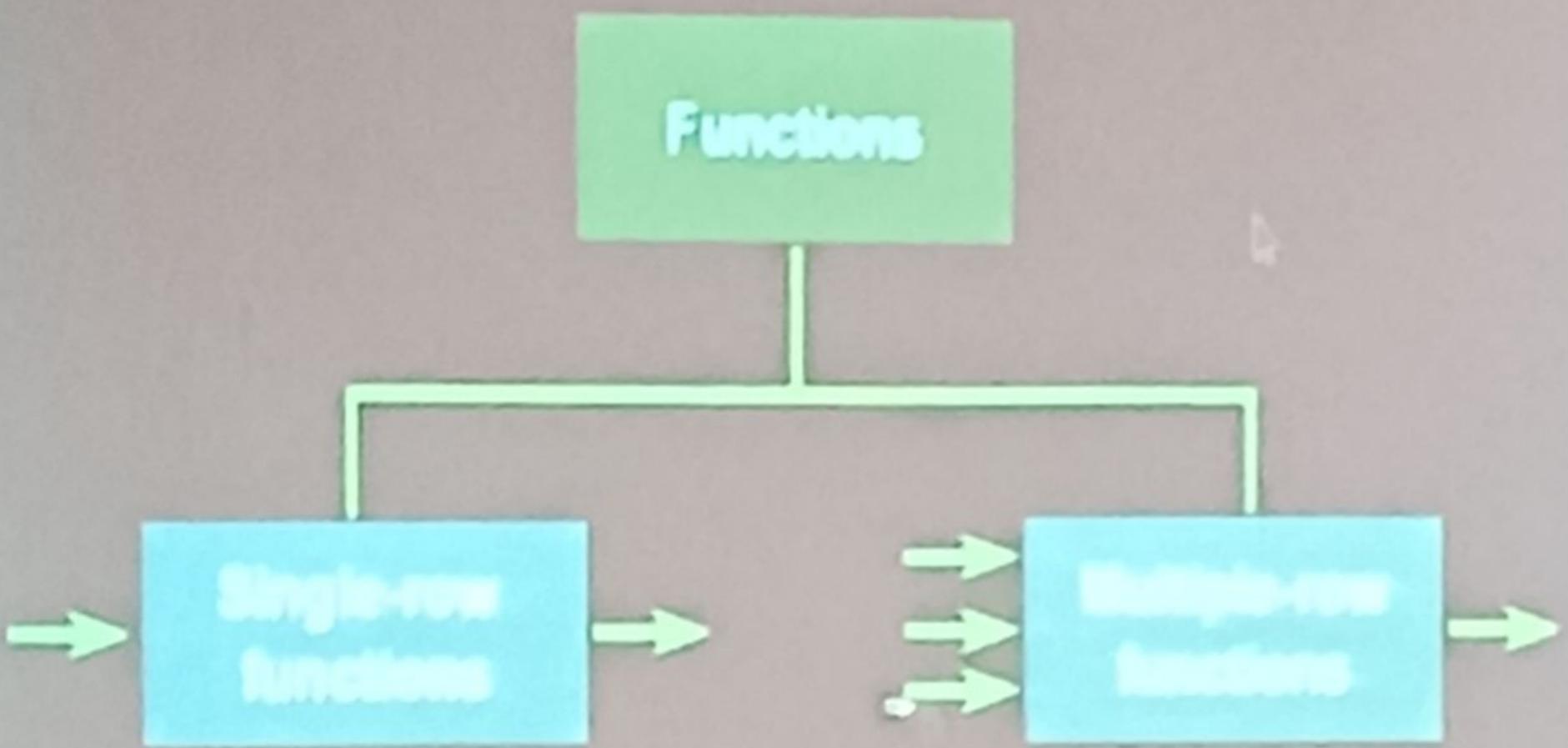
3-3

SQL Functions



ORACLE

Two Types of SQL Functions



ORACLE

Copyright © Oracle Corporation 2001. All rights reserved.

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,...)]

ORACLE

Character Functions

Character functions

Case-manipulation functions

LOWER
UPPER
INITCAP

Character-manipulation functions

CONCAT
SUBSTR
LENGTH
INSTR
LPAD | RPAD
TRIM
REPLACE

ORACLE

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

Copyright © Oracle Corporation, 2001. All rights reserved.

ORACLE

Search



10-43-2001

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

25 Higgins

Character-Manipulation Functions

These functions manipulate character strings:

Function	Result
CONCAT('Hello', 'World')	HelloWorld
SUBSTR('HelloWorld', 1, 5)	Hello
LENGTH('HelloWorld')	10
INSTR('HelloWorld', 'W')	6
LPAD(salary, 10, '*')	*****24000
RPAD(salary, 10, '*')	24000*****
TRIM('H' FROM 'HelloWorld')	elloWorld

ORACLE

Using the Character-Manipulation Functions

```
1  
SELECT employee_id, CONCAT(first_name, last_name) NAME,  
      job_id, LENGTH(last_name), ←  
      INSTR(last_name, 'a') "Contains 'a'??" ←←  
  FROM employees  
 WHERE SUBSTR(job_id, 4) = 'REP';  
2  
3
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	LENGTH(LAST_NAME)	Contains 'a'??
174	ElenAmd	SA_REP	6	0
176	ImanuelTaylor	SA_REP	11	1
179	KimberlyGrant	SA_REP	10	1
202	PuffBy	MK_REP	5	0

1

2

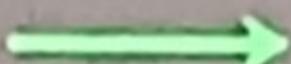
3

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

Number Functions

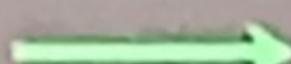
ROUND: Rounds value to specified decimal



TRUNC: Truncates value to specified decimal



MOD: Returns remainder of division



ORACLE

Using the ROUND Function

1

2

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

3

1

2

3

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

ORACLE

Using the TRUNC Function

1

2

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

3

TRUNC(45.923,2)

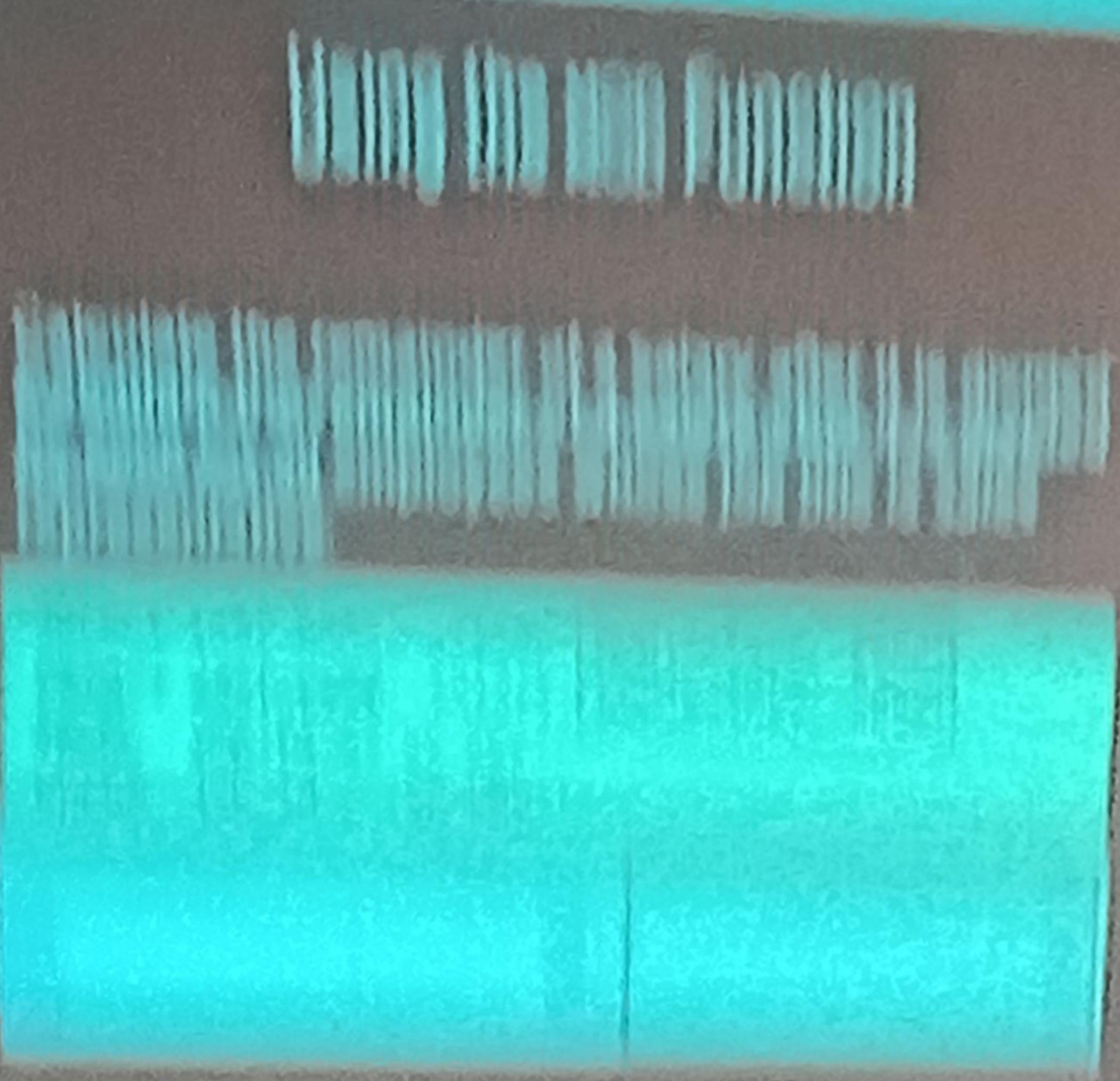
TRUNC(45.923,-2)

TRUNC(45.923,2)

结果

45

0



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.

Allows you to store 21st century dates in the 20th century by specifying only the last two digits of the year.

Allows you to store 20th century dates in the 21st century in the same way.

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

LAST NAME
Gertz
Grant

HIRE DATE
17-JUN-94
24-MAY-95

ORACLE

Working with Dates

SYSDATE is a function that returns:

Date

Time

Using Arithmetic Operators with Dates

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

LAST NAME	WEEKS
King	14.2625
Abhiram	6.357638
Cv Hall	8.357638

$$(15) = 1.6$$

ORACLE

Using Arithmetic Operators with Dates

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

LAST NAME	WEEKS
King	44.074
Partners	43.725
De man	43.374

ORACLE



Date Functions

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

Implicit Data Type Conversion

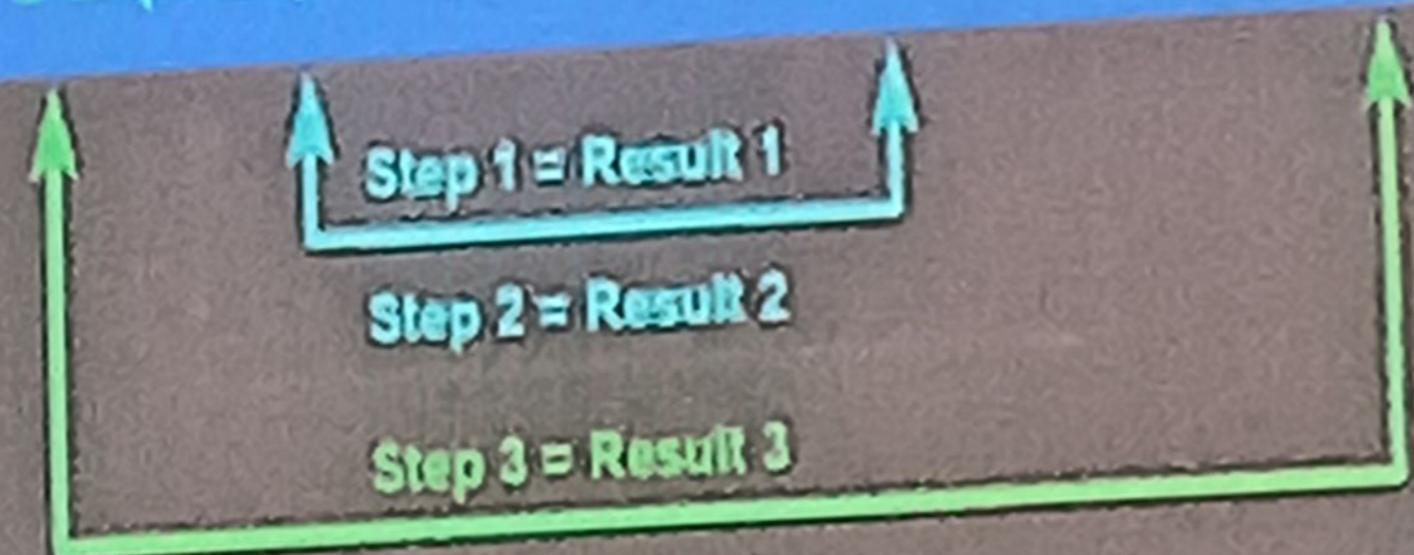
For assignments, the Oracle server can automatically convert the following:

From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

Nesting Functions

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

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



ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.



Nesting Functions

```
SELECT last_name,  
       NVL(TO_CHAR(manager_id), 'No Manager')  
FROM   employees  
WHERE  manager_id IS NULL;
```

LAWYER

FAITH.SWANSON@ACME.COM

209

No Manager

General Functions

These functions work with any data type and pertain to using nulls.

NVL (expr1, expr2)

NVL2 (expr1, expr2, expr3)

NULLIF (expr1, expr2)

COALESCE (expr1, expr2, ..., exprn)

ORACLE

NVL Function

Converts a null to an actual value.

Data types that can be used are date, character, and number.

Data types must match:

`NVL(commission_pct, 0)`

`NVL(hire_date, '01-JAN-97')`

`NVL(job_id, 'No Job Yet')`



Using the NVL Function

```
SELECT last_name, salary, NVL(commission_pct, 0) 1  
      (salary*12) + (salary*12*NVL(commission_pct, 0)) AS_SAL 2  
FROM employees;
```

LAST_NAME	SALARY	NVL(COMMISSION_PCT, 0)	AS_SAL
King	24000	0	24000
Kochhar	17000	0	17000
De Haan	15000	0	15000
Munoz	9000	0	9000
Ernst	6000	0	6000
Leverett	4000	0	4000
Murphy	3000	0	3000
Fay	2500	0	2500
...			
20 rows selected			

1

2

ORACLE

```

SELECT last_name, salary, commission_pct,
       NVL2(commission_pct,
             'SAL+COMM', 'SAL') income
FROM employees WHERE department_id IN (50, 80);

```

The diagram illustrates the execution flow of the SQL query. Box 1 highlights the reference to the 'commission_pct' column. Box 2 highlights the use of the 'NVL2' function to conditionally return either the sum of salary and commission or just the salary.

LAST_NAME	SALARY	COMMISSION_PCT	INCOME
Delaney	10500	2	SAL+COMM
Abel	11000	3	SAL+COMM
Taylor	6500	2	SAL+COMM
Murphy	5800		SAL
Patt	2500		SAL
Dawes	3100		SAL
Matos	2900		SAL
Vargas	2500		SAL

Using the NULLIF Function

1

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

2

3

FIRST NAME	MIDDLE	LAST NAME	SPY#	CLASS
Tina		Dyer	10	10
Sarah		Wichner	11	11
Lynn		De Haan	12	12
Alexander		Huang	13	13
Mike		Santos	14	14
David		Garcia	15	15
Kathy		Morgan	16	16
John		Harris	17	17
Cathy		Dowdy	18	18

1

2

2

ORACLE

Conditional Expressions

Provide the use of IF-THEN-ELSE logic within a SQL statement

Use two methods:

CASE expression

DECODE function

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

Facilitates conditional inquiries by using the WHEN clause
an IF-THEN-ELSE statement:

```
SELECT last_name, job_id, salary,  
CASE job_id WHEN 'IT_PDG' 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;
```

The DECODE Function

Facilitates conditional inquiries by doing the work of
a CASE OR IF-THEN-ELSE statement:

```
DECODE(col|expression, search1, result1  
[, search2, result2,...]  
[, default])
```

Using the DECODE Function

```
SELECT last_name, job_id, salary,  
       DECODE(job_id, 'IT_PROG', 1.10*salary,  
              'ST_CLERK', 1.15*salary,  
              'SA REP', 1.20*salary,  
              salary)  
       REVISED_SALARY  
FROM   employees;
```

LAST_NAME	JOB_ID	SALARY	REVISED_SALARY
Lorentz	IT_PROG	4300	4730
Morgon	ST_MAN	5800	6380
Rajs	ST_CLERK	2500	2875
Gertz	AC_ACCOUNT	8000	8800

20 rows selected