Using Conversion Functions and Conditional Expressions

Lesson Objectives

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

- Use TO_CHAR conversion functions
- Apply conditional expressions in a SELECT statement

Lesson Agenda

- TO_CHAR function
- Nesting functions
- General functions:
  - NVL
  - NVL2
  - NULLIF
- Conditional expressions:
  - CASE
  - DECODE

Lesson Agenda

- **TO_CHAR function**
- Nesting functions
- General functions:
  - NVL
  - NVL2
  - NULLIF
- Conditional expressions:
  - CASE
  - DECODE

Using the TO_CHAR Function with Dates

```
TO_CHAR(date, 'format_model')
```

The format model:

- Must be enclosed with **single-quotation** marks.
- Is case-sensitive.
- Can include any valid date format element.
- Has an `fm` element to **remove padded blanks or suppress leading zeros**.
- Is separated from the date value by a comma.

Elements of the Date Format Model

Function	Description
YYYY	Full year in numbers
YEAR	Year spelled out (in English)
MM	Two-digit value for the month
MONTH	Full name of the month
MON	Three-letter abbreviation of the month
DY	Three-letter abbreviation of the day of the week
DAY	Full name of the day of the week
DD	Numeric day of the month

Format Date : DD-MON-RR

Using the TO_CHAR Function with Dates

In the example in the slide display employee number, hire date format model for 'MM/YY'.

```
SELECT    employee_id,
          TO_CHAR(hire_date, 'MM/YY') Month_Hired
FROM      employees
WHERE     last_name = 'Higgins';
```

	EMPLOYEE_ID	MONTH_HIRED
1	205	06/94

Elements of the Date Format Model

Description	Format	Result
Time elements format the time portion of the date	HH24:MI:SS AM	15:45:32 PM
Add character strings by enclosing them with double quotation marks	DD "of" MONTH	12 of OCTOBER
Number suffixes spell out numbers	ddspth	fourteenth

Elements of the Date Format Model

Element	Description
AM or PM	Meridian indicator
A.M. or P.M.	Meridian indicator with periods
HH or HH12 or HH24	Hour of day, or hour(1-12), or hour(0-23)
MI	Minute(0-59)
SS	Second(0-59)
SSSSSW	Seconds past midnight(0-86399)

Use the formats that are listed in the following tables to display time information and literals, and o change numerals to spelled numbers.

Elements of the Date Format Model

Other Format

Element	Description
/ . ,	Punctuation is reproduced in the result.
"of the"	Quoted string is reproduced in the result.

Elements of the Date Format Model

Specifying Suffixes to Influence Number Display

Element	Description
TH	Ordinal number (for example, DDTH for 4TH)
SP	Spelled-out number (for example, DDSP for FOUR
SPTH OR THSP	Spelled-out ordinal numbers (for example, DDSPTH for FOURTH)

Using the TO_CHAR Function with Dates

The SQL statement in the slide displays the last names and hire dates for all employees. The hire date appears as 17 June 1987.

	LAST_NAME	HIREDATE
1	King	17 June 1987
2	Kochhar	21 September 1989
3	De Haan	13 January 1993
4	Hunold	3 January 1990
5	Ernst	21 May 1991

```
SELECT  last_name,
        TO_CHAR(hire_date,  'fmDD Month  YYYY')
        AS HIREDATE
FROM    employees;
```

Using the TO_CHAR Function with Dates

PRACTICE:

Modify the example in the slide to display the dates in a format that appears as "Seventeenth of June 1987 12:00:00 AM."

	LAST_NAME	HIREDATE
1	King	Seventeenth of June 1987 12:00:00 AM
2	Kochhar	Twenty-First of September 1989 12:00:00 AM
3	De Haan	Thirteenth of January 1993 12:00:00 AM
4	Hunold	Third of January 1990 12:00:00 AM
5	Ernst	Twenty-First of May 1991 12:00:00 AM

Using the TO_CHAR Function with Numbers

```
TO_CHAR(number, 'format_model')
```

These are some of the format elements that you can use with the TO_CHAR function to display a number value as a character:

Element	Description
9	Represents a number
0	Forces a zero to be displayed
$	Places a floating dollar sign
L	Uses the floating local currency symbol
.	Prints a decimal point
,	Prints a comma as a thousands indicator

Using the TO_CHAR Function with Numbers

- Number Format Elements
- If you are converting a number to the character data type, you can use the following format elements:

Element	Example	Result
9	999999	1234
0	099999	001234
$	$999999	$1234
D	99D99	99.99
.	999999.99	1234.00
,	999,999	1,234

Using the TO_CHAR Function with Numbers

The Oracle server displays a string of number signs in place of a whole number whose digits exceed the number of digits provided in the format model.

```
SELECT  TO_CHAR(salary, '$99,999.00')  SALARY
FROM    employees
WHERE   last_name =  'Ernst';
```

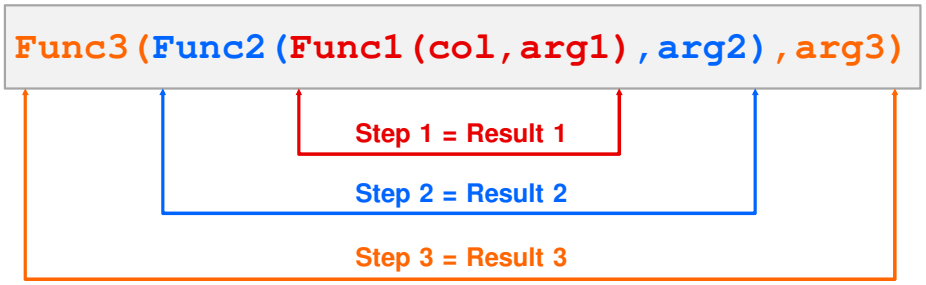
	SALARY
1	$6,000.00

Lesson Agenda

- TO_CHAR function
- **Nesting functions**
- General functions:
  - NVL
  - NVL2
  - NULLIF
- Conditional expressions:
  - CASE
  - DECODE

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

Require	Statement
1. The inner function retrieves the first eight characters of the last name.	`Result1 = SUBSTR(last_name,1,8)`
2. The outer function concatenates the result with _US.	`Result2 = CONCAT(Result1,'_US')`
3. The outermost function converts the results to upper-case.	`Result3 = UPPER(Result2)`

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

	LAST_NAME	Last name
1	Hunold	HUNOLD_US
2	Ernst	ERNST_US
3	Austin	AUSTIN_US
4	Pataballa	PATABALL_US
5	Lorentz	LORENTZ_US

Nesting Functions

PRACTICE:

Display the date of the next Friday that is six months from the hire date. The resulting date should appear as `Friday,August 13th,1999` Order the results by hire date.

	Next 6 Month Review
1	Friday,December 18th,1987
2	Friday,March 18th,1988
3	Friday,March 23rd,1990
4	Friday,July 6th,1990

Lesson Agenda

- TO_CHAR function
- Nesting functions
- **General functions:**
  - **NVL**
  - **NVL2**
  - **NULLIF**
- Conditional expressions:
  - CASE
  - DECODE

General Functions

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

Function	Description
NVL(expr1,expr2)	Converts a null value to an actual value.
NVL2(expr1,expr2,expr3)	If expr1 is not null, NVL2 returns expr2, If expr1 is null, NVL2 returns expr3. The argument expr1 can have any data type.
NULLIF(expr1,expr2)	Compares two expressions and returns null if they are equal; returns the first expression if they are not equal.
COALESCE(expr1,…,exprN)	Return the first non-null expression in the expression list.

NVL Function

To Converts a null value to an actual value:

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

```
NVL(expr1,  expr2)
```

Data Type	Conversion Example
NUMBER	NVL(commission_pct, **0**)
DATE	NVL(hire_date, **'01-JAN-97'**)
CHAR OR VARCHAR2	NVL(job_id, **'Unavailable'**)

Using NVL Function

- Display column commission if null value actual value is zero.
- Calculate annual salary by salary per year multiply with commission per year.

LAST_NAME	SALARY	COMMISSION_PCT	NVL_COMMISSION_PCT	AN_SAL
Davies	3100	(null)	0	37200
Matos	2600	(null)	0	31200
Vargas	2500	(null)	0	30000
Russell	14000	0.4	0.4	235200
Partners	13500	0.3	0.3	210600

```
SELECT   last_name, salary,  commission_pct,
         NVL(commission_pct,0) NVL_COMMISSION_PCT,
         (salary*12)  +  (salary*12*NVL(commission_pct,0)) AN_SAL
FROM     employees;
```

NVL2 Function

If expr1 is not null,NVL2 returns expr2,If expr1 is null,NVL2 returns expr3.The argument expr1 can have any data type.

```
NVL2(expr1,  expr2,  expr3)
```

expression	Description
expr1	is the source value or expression that may contain a null
expr2	is the value that is returned if *expr1* is not null
expr3	is the value that is returned if *expr1* is null

Using NVL2 Function

Display last name, salary, and commission. By commission is not null value display text are 'SAL+COM' but commission is null value display text are 'SAL' with department number in 50 or 80.

LAST_NAME	SALARY	COMMISSION_PCT	INCOME
Taylor	8600	0.2	SAL+COM
Livingston	8400	0.2	SAL+COM
Johnson	6200	0.1	SAL+COM
Taylor	3200	(null)	SAL
Fleaur	3100	(null)	SAL
Sullivan	2500	(null)	SAL

```
SELECT   last_name, salary,  commission_pct,
         NVL2(commission_pct,  'SAL+COM',  'SAL') AS INCOME
FROM     employees
WHERE    department_id IN  (50,  80);
```


NULLIF Function

The NULLIF function compares two expressions. If they are equal, the function returns a null. If they are not equal, the function returns the first expression.

```
NULLIF(expr1,  expr2)
```

NOTE:

NULLIF compares *expr1* and *expr2*.

- If they are equal, then the function returns null.
- If they are not, then the function returns *expr1*.
- However, you cannot specify the literal NULL for *expr1*.

Using NULLIF Function

Display first name, last name and using function length of first name and last name. If they are equal, the function returns a null. If they are not equal, the function returns the length or first name.

FIRST_NAME	expr1	LAST_NAME	expr2	result
Gerald	6	Cambrault	9	6
Nanette	7	Cambrault	9	7
John	4	Chen	4	(null)
Kelly	5	Chung	5	(null)
Karen	5	Colmenares	10	5

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

COALESCE Function

The advantage of the COALESCE function over the NVL function is that the COALESCE function can take multiple alternate values.

```
COALESCE(expr1, expr2, ..., exprN)
```

expression	Description
expr1	returns this expression if it is not null
expr2	*expr2* returns this expression if the first expression is null and this expression is not null
exprN	returns this expression if the preceding expressions are null and this expression is not null

Note that all expressions must be of the same data type.

Using COALESCE Function

In the example shown in the slide, if the manager_id value is not null, it is displayed, If the manager_id value is null, then the commission_pct is displayed. If the manager_id and commission_pct values are null, then "No commission and no manager" is displayed.

Note: TO_CHAR function is applied so that all expressions are of the same data type.

	LAST_NAME	MANAGER_ID	COMMISSION_PCT	EXPRESSION
1	King	(null)	(null)	No commission and No manager
2	Kochhar	100	(null)	100
3	De Haan	100	(null)	100
...				
49	Cambrault	100	0.3	0.3
50	Zlotkey	100	0.2	0.2
51	Tucker	145	0.3	0.3

```
SELECT   last_name, manager_id, commission_pct,
         COALESCE(TO_CHAR(commission_pct), TO_CHAR(manager_id),
                  'No commission and No manager') AS EXPRESSION
FROM     employees;
```


Using COALESCE Function

PRACTICE:

Display employees who do not get any commission, the new salary column shows the salary increment by $2,000 and for employees who get commission and for employees who get commission, the new salary column shows the computed commission amount added to the salary.

	LAST_NAME	SALARY	COMMISSION_PCT	New Salary
44	Matos	2600	(null)	4600
45	Vargas	2500	(null)	4500
46	Russell	14000	0.4	19600
47	Partners	13500	0.3	17550
48	Errazuriz	12000	0.3	15600

Lesson Agenda

- TO_CHAR function
- Nesting functions
- General functions:
  - NVL
  - NVL2
  - NULLIF
- **Conditional expressions:**
  - **CASE**
  - **DECODE**

Conditional Expressions

- Provide the use of the IF–THEN–ELSE logic within a SQL statement
- Use two methods:
  - CASE expression
  - DECODE function

CASE Expression

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

```
CASE expr WHEN comparison_expr1 THEN retrun_expr1
          [WHEN comparison_expr2 THEN retrun_expr2
           WHEN comparison_exprN THEN retrun_exprN
           ELSE else_expr]
END
```

All of the expressions(expr, comparison_expr and return_expr) must be of the **same data type**, which can be CHAR or VARCHAR2.

Using CASE Expression

```
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 AS "REVISED SALARY"
FROM   employees;
```

In the SQL statement in the slide, the value of JOB_ID is decoded. If JOB_ID is IT_PROG, the salary increase is 10%;if JOB_ID is ST_CLERK, the salary increase is 15%;if JOB_ID is SA_REP, the salary increase is 20%. For all other job roles, there is no increase in salary.

	LAST_NAME	JOB_ID	SALARY	REVISED SALARY
1	King	AD_PRES	24000	24000
2	Kochhar	AD_VP	17000	17000
3	De Haan	AD_VP	17000	17000
4	Hunold	IT_PROG	9000	9900
5	Ernst	IT_PROG	6000	6600
6	Austin	IT_PROG	4800	5280
7	Pataballa	IT_PROG	4800	5280
8	Lorentz	IT_PROG	4200	4620
9	Greenberg	FI_MGR	12000	12000
10	Faviet	FI_ACCOUNT	9000	9000

Using CASE Expression

PRACTICE:

Display the last name, salaries. If salaries less than 5,000 then "Low" is displayed, if salaries less than 10,000 then "Medium" is displayed, if salaries less than 20,000 then "Good" is displayed.

	LAST_NAME	SALARY	QUALIFIED_SALARY
1	King	24000	(null)
2	Kochhar	17000	Good
3	De Haan	17000	Good
4	Hunold	9000	Medium
5	Ernst	6000	Medium
6	Austin	4800	Low
7	Pataballa	4800	Low

DECODE Expression

Facilitates conditional inquiries by doing the work of a CASE expression or an IF–THEN–ELSE statement:

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

- The DECODE function decodes *expression* after comparing it to each *search* value. If the expression is the same as *search*, *result* is returned.
- If the default value is omitted, a null value is returned where a search value does not match any of the result values.

Using DECODE Expression

```
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) AS revised_salary
FROM    employees;
```

The same statement can be expressed in pseudocode as an IF–THEN–ELSE statement:

- IF job_id='IT_PROG' THEN salary=salary * 1.10
- IF job_id='ST_CLERK' THEN salary=salary * 1.15
- IF job_id='SA_REP' THEN salary=salary * 1.20
- ELSE salary=salary

Using Conditional Expression

PRACTICE:

Display the tax rate for each employee in department 80 based on the monthly salary. The tax rates are as follows.

	LAST_NAME	SALARY	TAX_RATE
1	Russell	14000	0.45
2	Partners	13500	0.44
3	Errazuriz	12000	0.44
4	Cambrault	11000	0.42
5	Zlotkey	10500	0.42
6	Tucker	10000	0.42
7	Bernstein	9500	0.4
8	Hall	9000	0.4
9	Olsen	8000	0.4

Display 34 rows.

Using Conditional Expression

PRACTICE (CASE):

Using Conditional Expression

PRACTICE (DECODE):

Using Conditional Expression

*ต้องการแสดงข้อมูล รหัสงาน ชื่องาน และเงินเดือนขั้นต่ำ โดยให้ปรับเงินเดือนขั้นต่ำ (ชื่อคอลัมน์ใหม่ **NEW_MIN**) ตามเงื่อนไขดังนี้*

- *ถ้า ชื่องาน ขึ้นต้นด้วย **A** เพิ่มเงินเดือนขั้นต่ำให้ **10%***
- *ถ้า ชื่องาน ขึ้นต้นด้วย **P** เพิ่มเงินเดือนขั้นต่ำให้ **20%***
- *ชื่องานอื่น ไม่มีการเพิ่มเงินเดือนขั้นต่ำ*

	JOB_ID	JOB_TITLE	MIN_SALARY	NEW_MIN
1	AD_PRES	President	20000	24,000.00
2	AD_VP	Administration Vice President	15000	16,500.00
3	AD_ASST	Administration Assistant	3000	3,300.00
4	FI_MGR	Finance Manager	8200	8,200.00
5	FI_ACCOUNT	Accountant	4200	4,620.00
6	AC_MGR	Accounting Manager	8200	9,020.00
7	AC_ACCOUNT	Public Accountant	4200	5,040.00
8	SA_MAN	Sales Manager	10000	10,000.00
9	SA_REP	Sales Representative	6000	6,000.00
10	PU_MAN	Purchasing Manager	8000	9,600.00
11	PU_CLERK	Purchasing Clerk	2500	3,000.00
12	ST_MAN	Stock Manager	5500	5,500.00

Using Conditional Expression

Summary

In this lesson, you should have learned how to:

- Alter date formats for displays using functions
- Convert column data types using functions
- Use `NVL` functions
- Use `IF-THEN-ELSE` logic and other conditional expressions in a `SELECT` statement