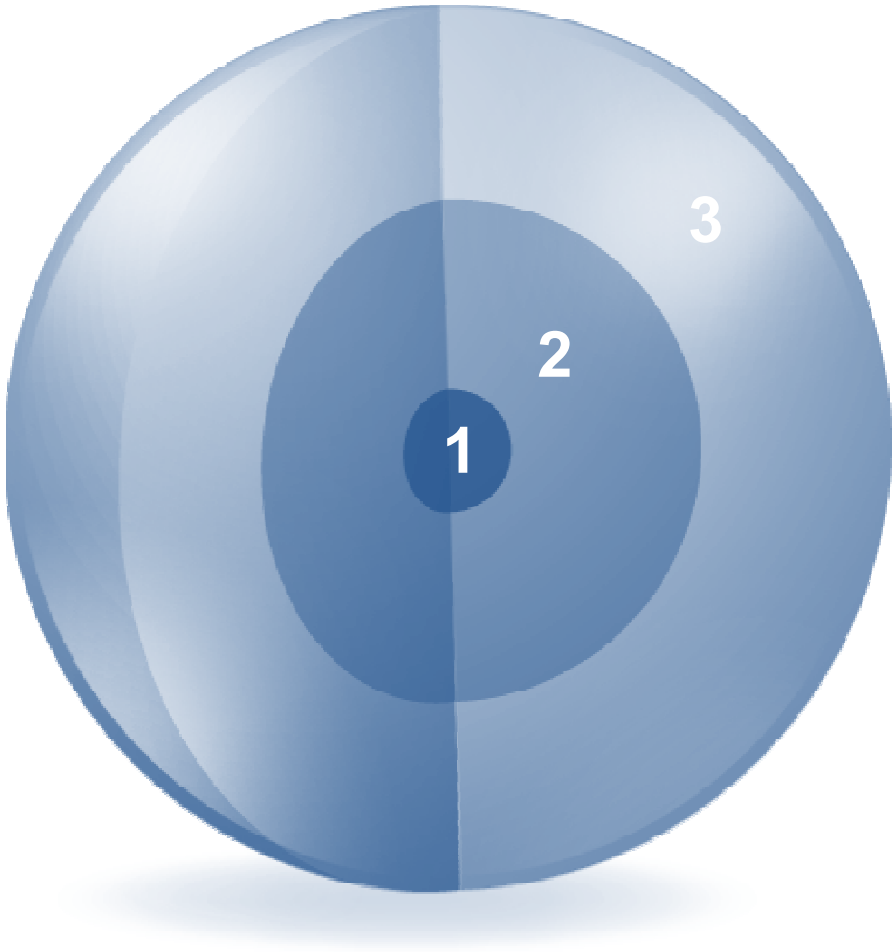
Lesson 4

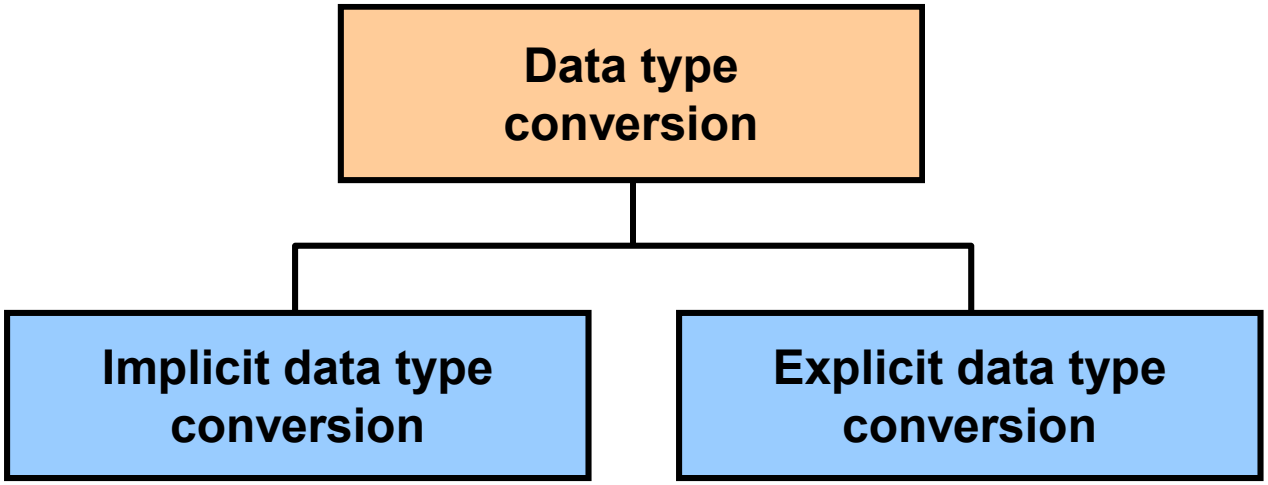
Using Conversion Functions and Conditional Expressions

What You will learn at the end of this Session?



1. **Describe the various types of conversion functions that are available in SQL**
2. **Use the TO_CHAR, TO_NUMBER, and TO_DATE conversion functions**
3. **Apply conditional expressions in a SELECT statement**

Conversion Functions



Implicit Data Type Conversion

In expressions, the Oracle server can automatically convert the following:

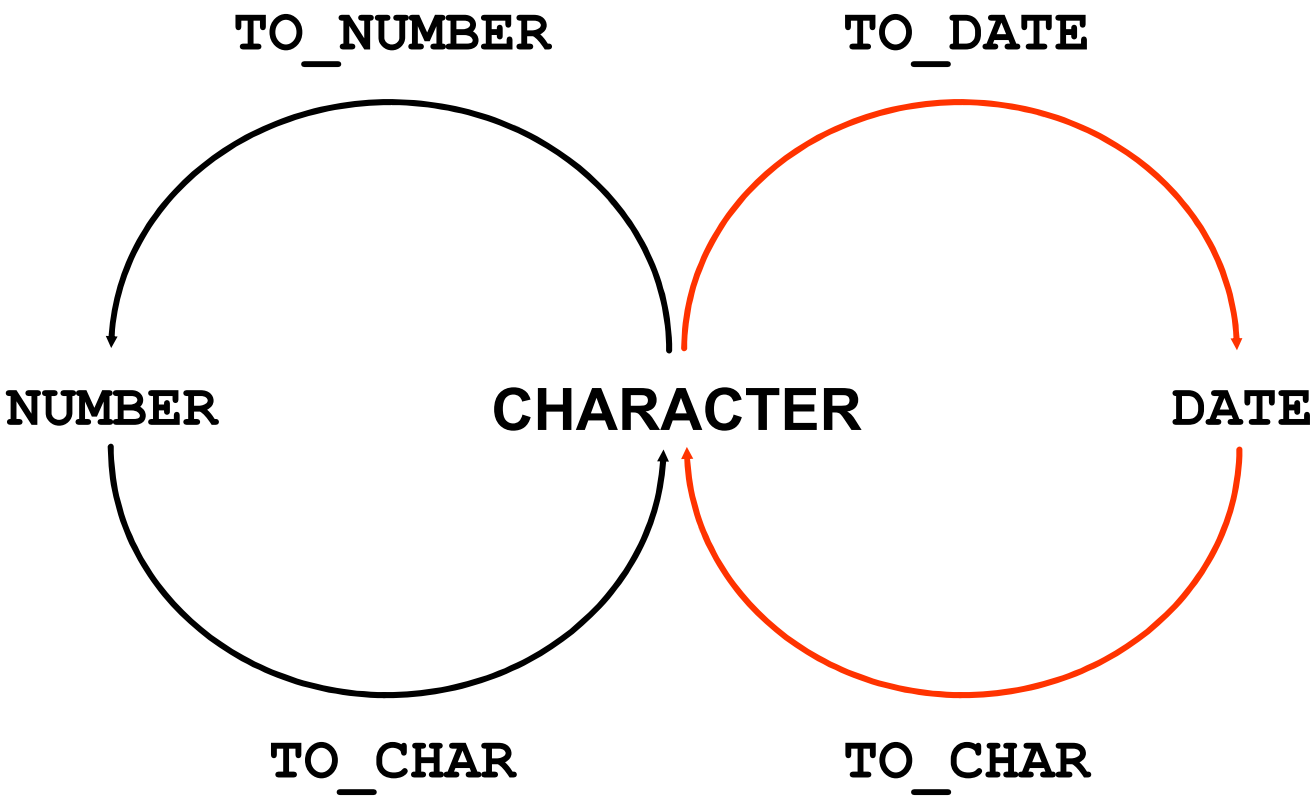
From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE

Implicit Data Type Conversion

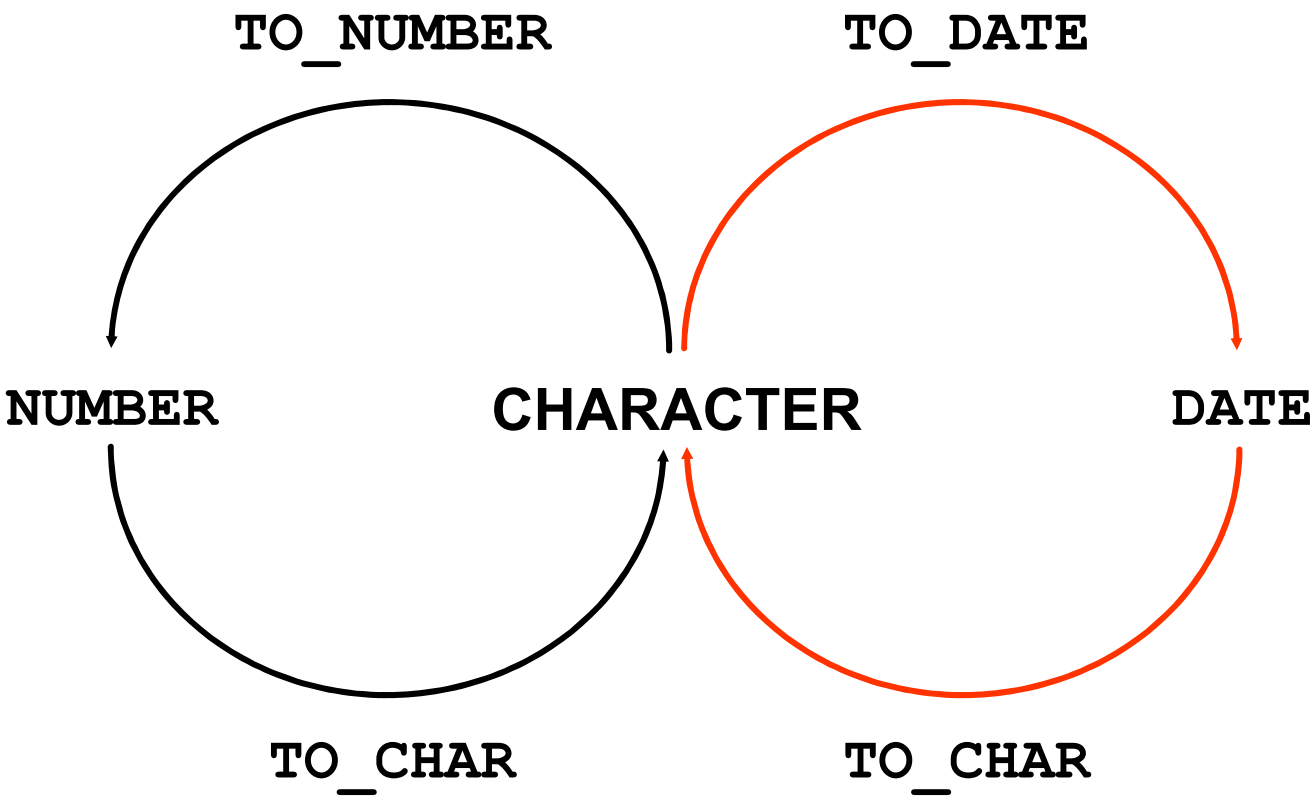
For expression evaluation, the Oracle server can automatically convert the following:

From	To
NUMBER	VARCHAR2 or CHAR
DATE	VARCHAR2 or CHAR

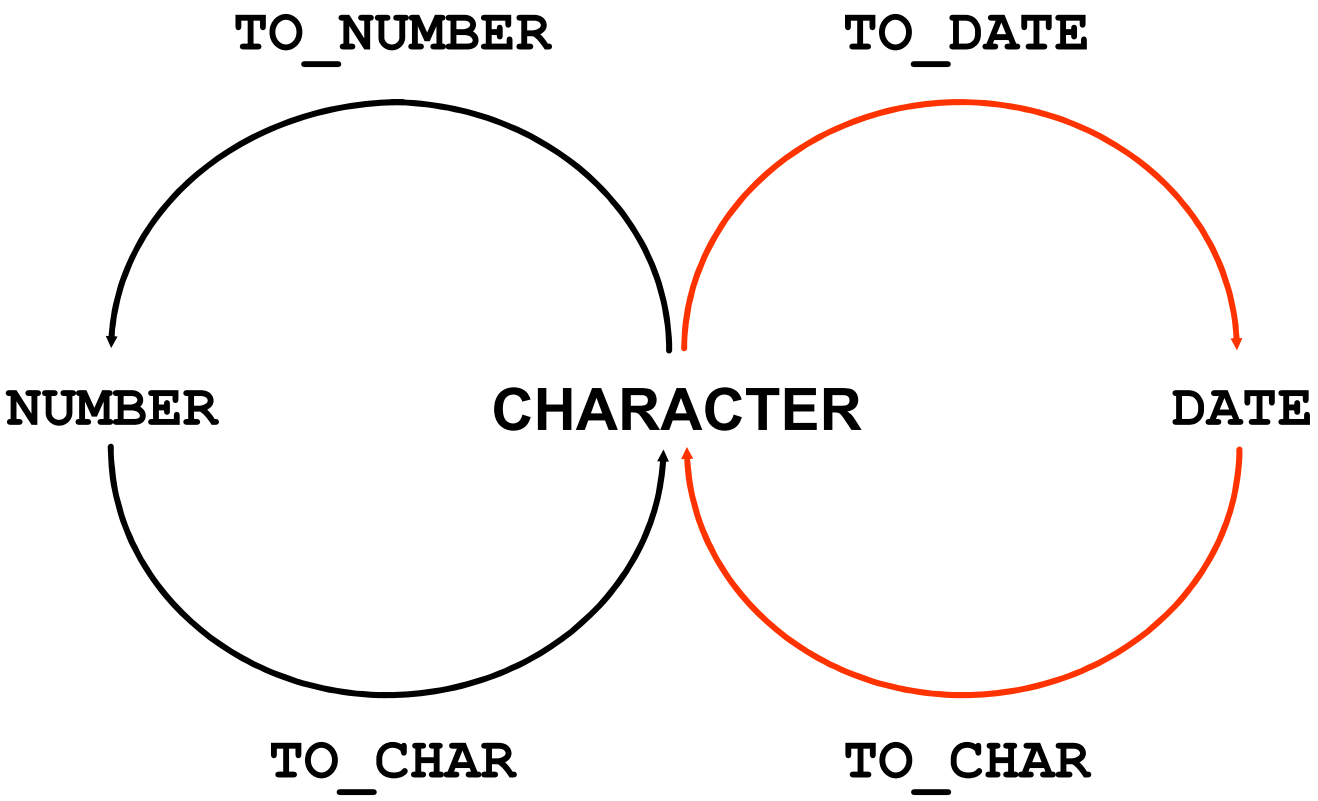
Explicit Data Type Conversion



Explicit Data Type Conversion



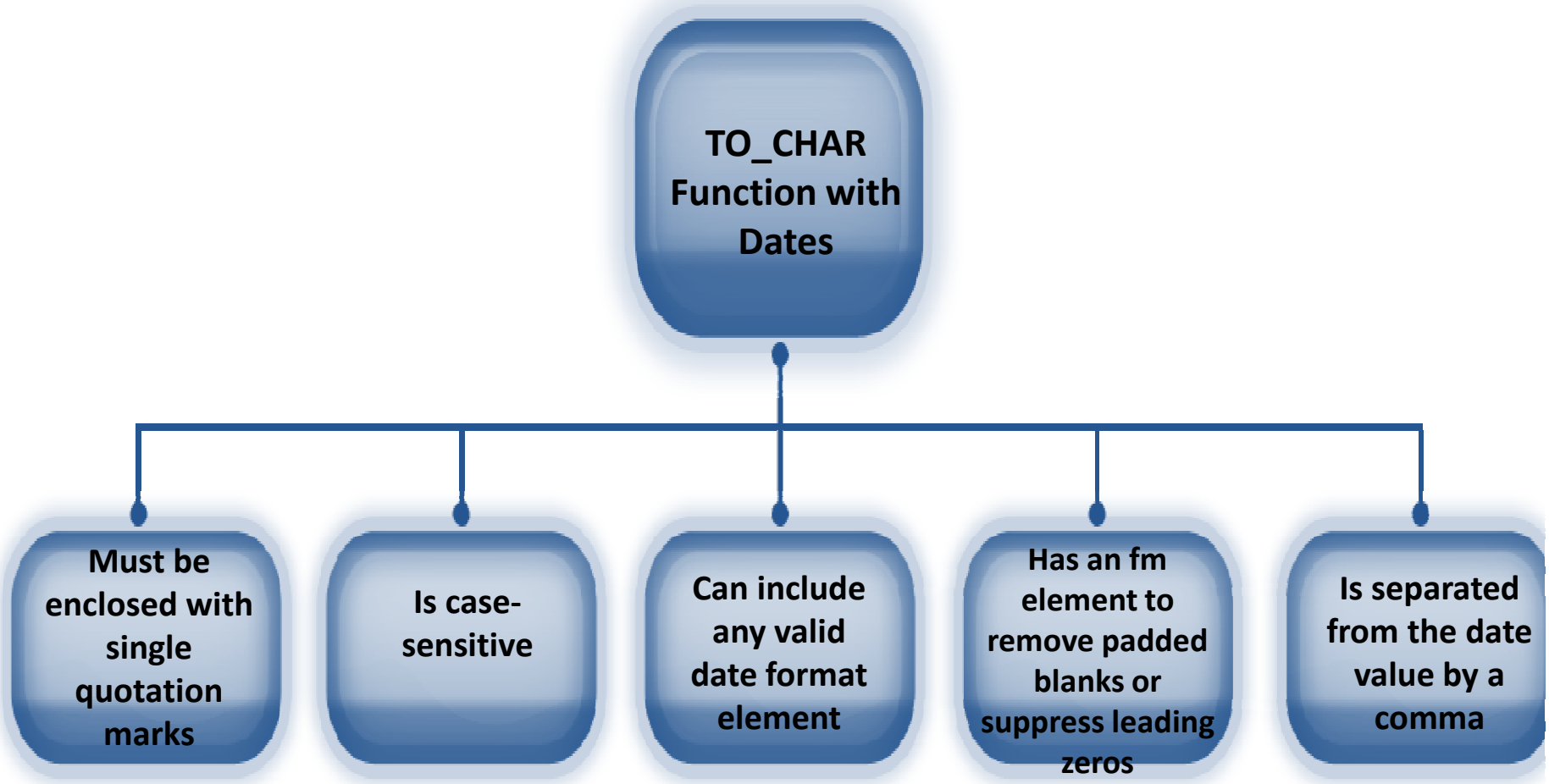
Explicit Data Type Conversion



Using the TO_CHAR Function with Dates

The format model:

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



Elements of the Date Format Model

Element	Result
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

Elements of the Date Format Model

– **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**

`DD "of" MONTH`	`12 of OCTOBER`

– **Number suffixes spell out numbers:**

`ddspth`	`fourteenth`

Using the TO_CHAR Function with Dates

```
SELECT  order_id,  to_char (order_date, ' fmDD Month YYYY ' )
AS   " Date of Order "
FROM   orders ;
```

	ORDER_ID	Date of Order
1	2458	17 August 1999
2	2397	20 November 1999
3	2454	3 October 1999
4	2354	15 July 2000
5	2358	9 January 2000
6	2381	15 May 2000
7	2440	1 September 1999
8	2357	9 January 1998
9	2394	11 February 2000
10	2435	3 September 1999

...

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	Result
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

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

	SALARY
1	$6,000.00

Using the TO_NUMBER and TO_DATE Functions

- Convert a character string to a number format using the TO_NUMBER function:

```
TO_NUMBER(char[, 'format_model'])
```

- Convert a character string to a date format using the TO_DATE function:

```
TO_DATE(char[, 'format_model'])
```

- These functions have an fx modifier. This modifier specifies the exact match for the character argument and date format model of a TO_DATE function.

Using the TO_CHAR and TO_DATE Function with the RR Date Format

To find employees hired before 1990, use the RR date format, which produces the same results whether the command is run in 1999 or now:

```
SELECT last_name, TO_CHAR(hire_date, 'DD-Mon-YYYY')
FROM  employees
WHERE hire_date < TO_DATE('01-Jan-90','DD-Mon-RR');
```

	LAST_NAME	TO_CHAR(HIRE_DATE,'DD-MON-YYYY')
1	Whalen	17-Sep-1987
2	King	17-Jun-1987
3	Kochhar	21-Sep-1989