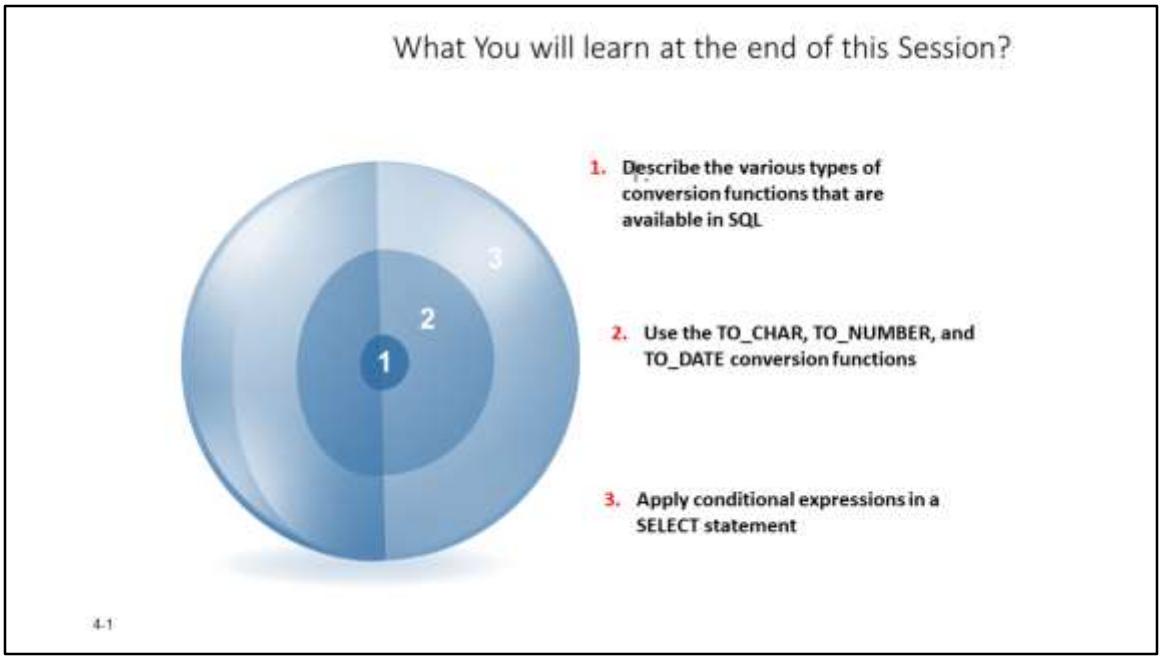
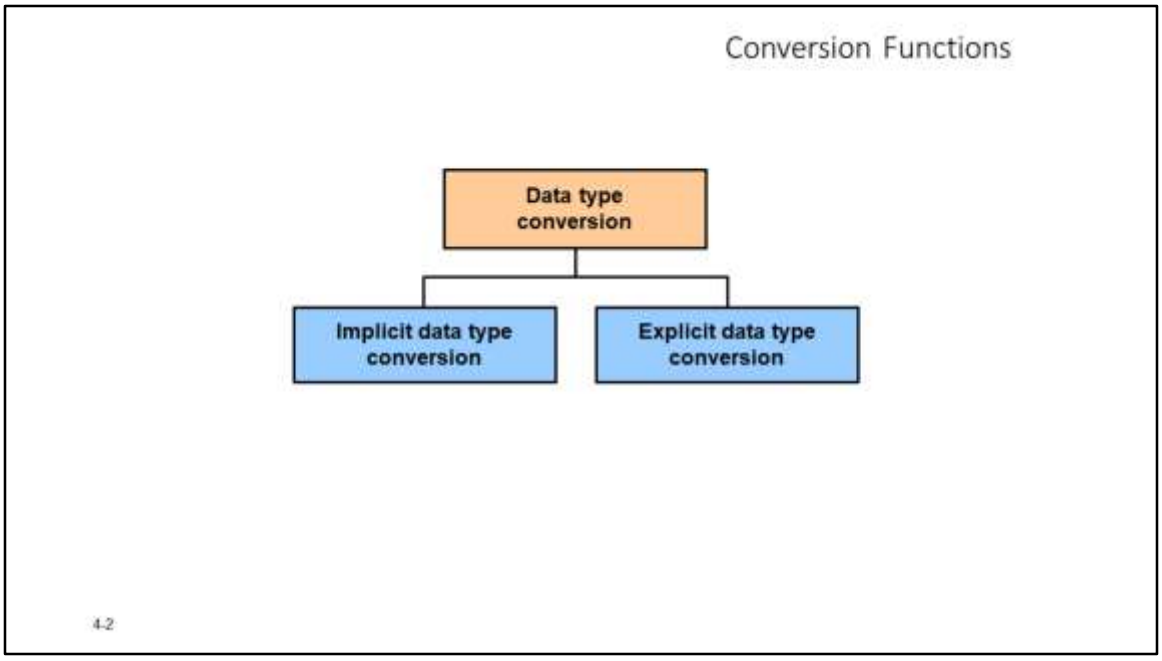


What You will learn at the end of this Session?

This lesson focuses on functions that convert data from one type to another (for example, conversion from character data to numeric data) and discusses the conditional expressions in SQL `SELECT` statements.



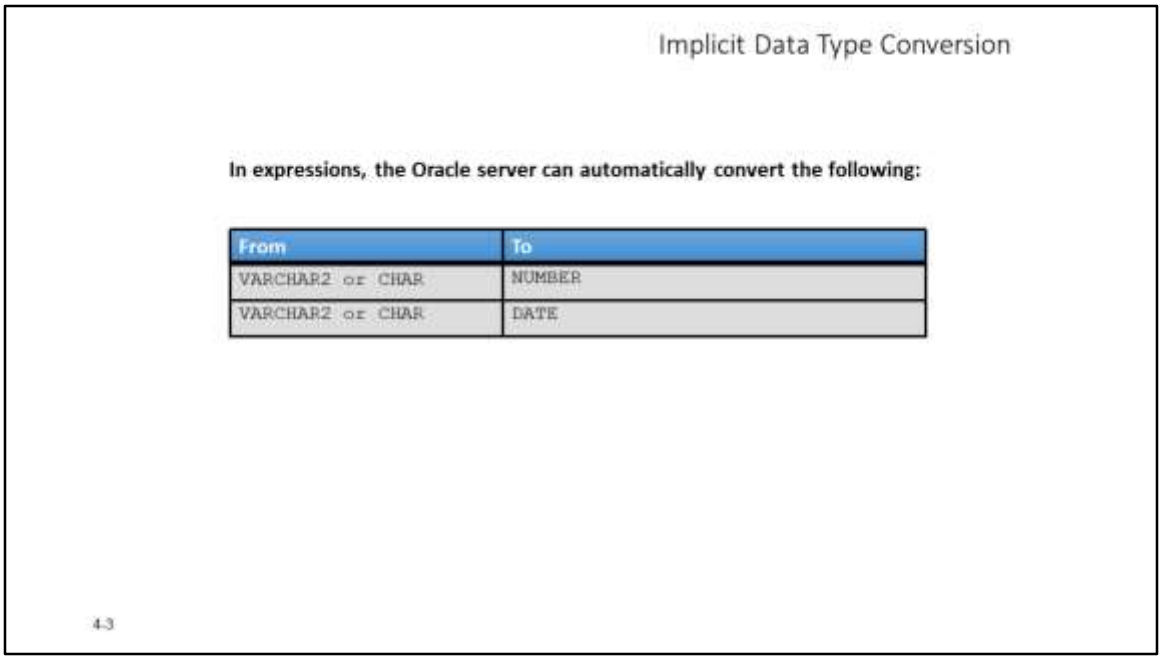
Conversion Functions

In addition to Oracle data types, columns of tables in an Oracle Database can be defined by using the American National Standards Institute (ANSI), DB2, and SQL/DS data types. However, the Oracle server internally converts such data types to Oracle data types.

In some cases, the Oracle server receives data of one data type where it expects data of a different data type. When this happens, the Oracle server can automatically convert the data to the expected data type. This data type conversion can be done *implicitly* by the Oracle server or *explicitly* by the user. Implicit data type conversions work according to the rules explained in the following slides.

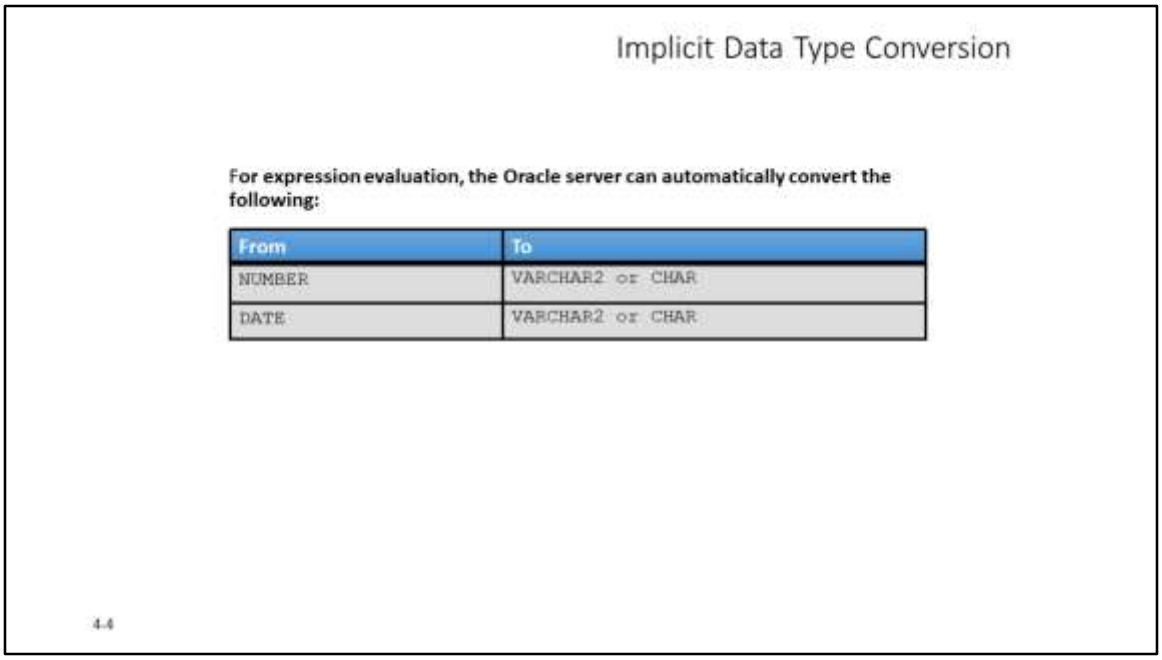
Explicit data type conversions are performed by using the conversion functions. Conversion functions convert a value from one data type to another. Generally, the form of the function names follows the convention *data type* `TO` *data type*. The first data type is the input data type and the second data type is the output.

Note: Although implicit data type conversion is available, it is recommended that you do the explicit data type conversion to ensure the reliability of your SQL statements.



Implicit Data Type Conversion

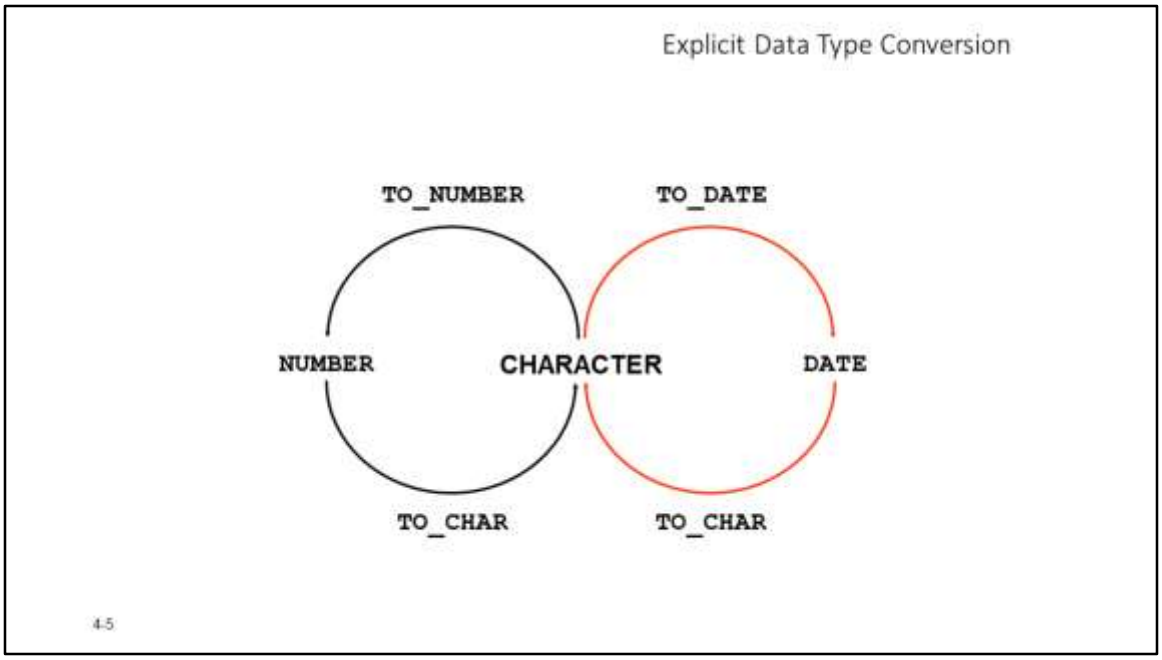
Oracle server can automatically perform data type conversion in an expression. For example, the expression `hire_date > '01-JAN-90'` results in the implicit conversion from the string `'01-JAN-90'` to a date. Therefore, a `VARCHAR2` or `CHAR` value can be implicitly converted to a number or date data type in an expression.



Implicit Data Type Conversion (continued)

In general, the Oracle server uses the rule for expressions when a data type conversion is needed. For example, the expression `grade = 2` results in the implicit conversion of the number `2` to the string "2" because grade is a `CHAR(2)` column.

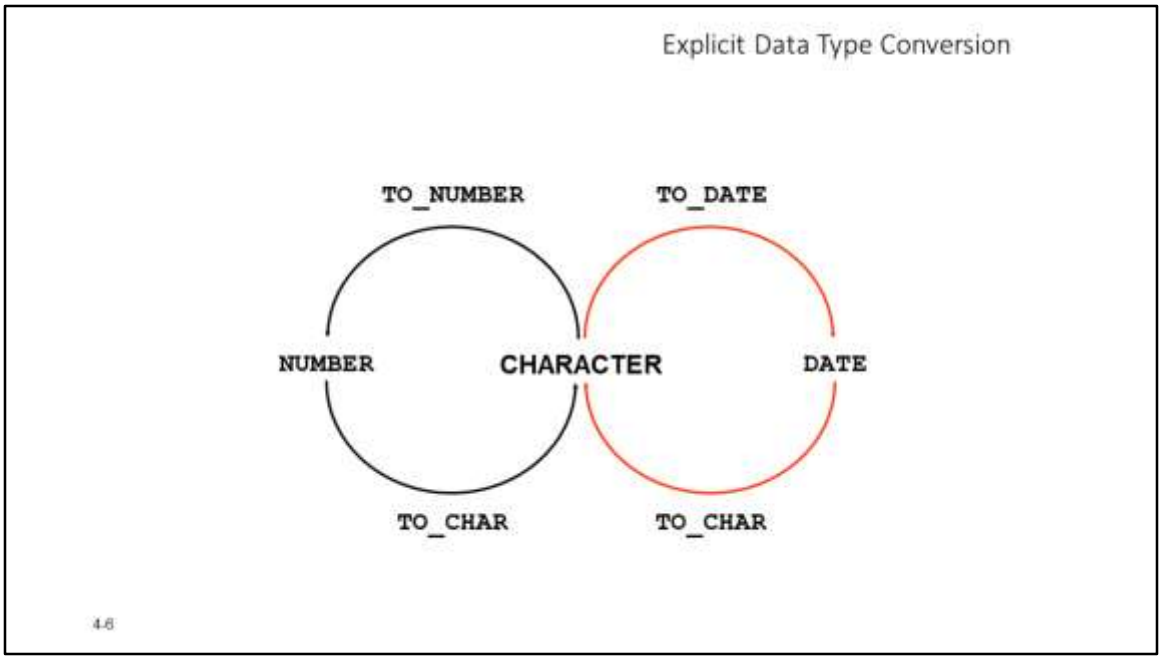
Note: `CHAR` to `NUMBER` conversions succeed only if the character string represents a valid number.



Explicit Data Type Conversion

SQL provides three functions to convert a value from one data type to another:

Function	Purpose
TO_CHAR(*number*|*date*,[*fmt*], *[nlsparams*])	Converts a number or date value to a VARCHAR2 character string with the format model *fmt* **Number conversion:** The nlsparams parameter specifies the following characters, which are returned by number format elements: • Decimal character • Group separator • Local currency symbol • International currency symbol If nlsparams or any other parameter is omitted, this function uses the default parameter values for the session.



Explicit Data Type Conversion (continued)

Function	Purpose
TO_CHAR(*number*|*date*,[*fmt*], [*nlsparams*])	**Date conversion:** The nlsparams parameter specifies the language in which the month and day names, and abbreviations are returned. If this parameter is omitted, this function uses the default date languages for the session.
TO_NUMBER(*char*,*[fmt]*, *[nlsparams]*)	Converts a character string containing digits to a number in the format specified by the optional format model *fmt*. The nlsparams parameter has the same purpose in this function as in the TO_CHAR function for number conversion.
TO_DATE(*char*,[*fmt*],[*nlsparams*])	Converts a character string representing a date to a date value according to *fmt* that is specified. If *fmt* is omitted, the format is DD-MON-YY. The nlsparams parameter has the same purpose in this function as in the TO_CHAR function for date conversion.

Explicit Data Type Conversion (continued)

Note: The list of functions mentioned in this lesson includes only some of the available conversion functions.

For more information, see the "Conversion Functions" section in *Oracle Database SQL Language Reference* for 10*g* or 11*g* database.