Instructor Notes:

DBMS/SQL

Lesson 05 SQL (Single-row) Functions

Capgemini

Instructor Notes:

None

Lesson Objectives

To understand the following topics:
- SQL (single-row) functions
  - Number functions
  - Character functions
  - Date functions
  - Conversion functions
  - Miscellaneous Single-row functions

Instructor Notes:

None

5.1: Types of Single Row Functions

Single Row Functions

Single-row functions return a single result row for every row of a queried Table or View.

- Single-row functions can appear in SELECT lists, WHERE clauses, START WITH and CONNECT BY clauses, and HAVING clauses.
- Different categories of single valued functions are:
  - Numerical functions
  - Character functions
  - Date and Time functions
  - Conversion functions
  - Miscellaneous Single-row functions

SQL Functions:
- So far, we have seen "Aggregate functions", which operate against a "collection of values", however return a "single value".
- Now we shall see "scalar functions" which operate against a "single value", and return a "single value" based on the input value.
- The functions can be broadly classified into:
  - Numerical functions- Accept numeric input and return numeric values
  - For example: ABS, CEIL, TRUNC, ROUND, POWER, etc.
  - Character functions - Accept character input and can return both characters and number values
  - For example: CONCAT, LPAD, RPAD, TRIM, SUBSTR, etc.
  - Date and Time functions- Operate on values of the DATE data type
  - For example: SYSDATE, ADD_MONTHS, LAST_DAY, etc.
  - Conversion functions- Convert a value from one data type to another
  - For example: CAST, ASCIISTR, ROWIDTOCHAR, etc.
  - Miscellaneous Single-row functions
  - For example: BFILENAME, DECODE, NVL, NULLIF, USERENV, etc.

Instructor Notes:

5.1: Types of Single Row Functions

Single Row Functions - Characteristics

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, …)]

Can be used in SELECT, WHERE, and ORDER BY clauses

Let us now have a look at each of these function categories

Instructor Notes:

None

5.2: Types of Single Row Functions

Number Functions

Number functions accept "numeric data" as argument, and returns "numeric values".

TRUNC(arg,n)	Returns a number "arg" truncated to a "n" number of decimal places.
ROUND (arg,n)	Returns "arg" rounded to "n" decimal places. If "n" is omitted, then "arg" is rounded as an integer.
CEIL (arg)	Returns the smallest integer greater than or equal to "arg".
FLOOR (arg)	Returns the largest integer less than or equal to "arg".
ABS (arg)	Returns the absolute value of "arg".
POWER (arg, n)	Returns the argument "arg" raised to the nth power.
SQRT (arg)	Returns the square root of "arg".
SIGN (arg)	Returns −1, 0, or +1 according to "arg" which is negative, zero, or positive respectively.
MOD (arg1, arg2)	Returns the remainder obtained by dividing "arg1" by "arg2".

Numeric Functions:
TRUNC(n,m)
The trunc function returns a number truncated to a certain number of decimal places. The syntax for the trunc function is:

> trunc(number, [decimal_places])

where:
number = the number to truncate.
decimal_places = the number of decimal places to truncate to. This value must be an integer. If this parameter is omitted, the trunc function will truncate the number to 0 decimal places.
For example:
trunc(125.815) would return 125
trunc(125.815, 1) would return 125.8
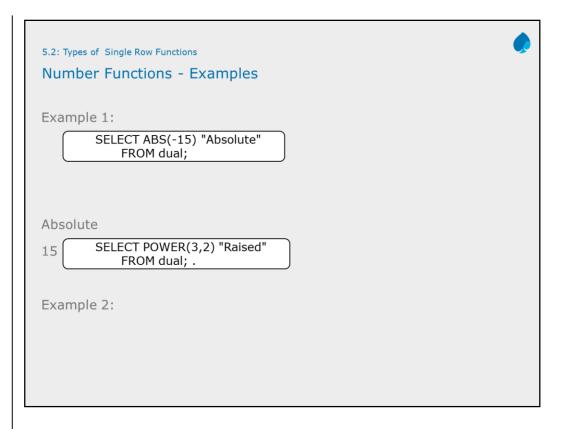trunc(125.815, -1) would return 120
trunc(125.815, -2) would return 100

ROUND(n,m)
The round function returns a number rounded to a certain number of decimal places. The syntax for the round function is:

> round(number, [decimal_places])

Instructor Notes:

None

5.2: Types of Single Row Functions

Number Functions - Examples

Example 1:

```
SELECT ABS(-15) "Absolute"
         FROM dual;
```

Absolute

15
```
SELECT POWER(3,2) "Raised"
         FROM dual; .
```
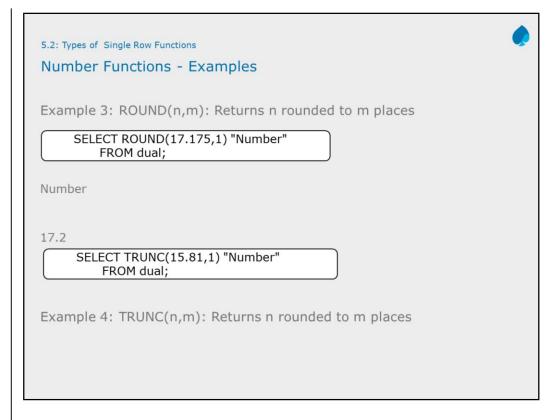
Example 2:

Examples of Number Functions:
- DUAL table, which is shown in the slide, is a table owned by SYS.
    - ➢ SYS owns the "data dictionary", and DUAL is part of the data dictionary.
    - ➢ DUAL is a small work-table, which consists of only one row and one column, and contains the value "x" in that column. Besides arithmetic calculations, it also supports "date retrieval" and it's "formatting".
    - ➢ Often a simple calculation needs to be done. A SELECT must have a table name in it's FROM clause, else it fails.
    - ➢ To facilitate such calculations via a SELECT, the DUAL dummy table is provided.
    - ➢ The structure of the DUAL table can be viewed by using the SQL statement:

```
DESC DUAL;
```

None

5.2: Types of Single Row Functions

Number Functions - Examples

Example 3: ROUND(n,m): Returns n rounded to m places

```
SELECT ROUND(17.175,1) "Number"
     FROM dual;
```

Number

17.2

```
SELECT TRUNC(15.81,1) "Number"
     FROM dual;
```


Example 4: TRUNC(n,m): Returns n rounded to m places

Examples of Number (numeric) Functions (contd.):
- **Round(n,m)**:

```
SELECT ROUND(17.175,-1) "Number"
      FROM dual;
```

O/P : 20

- **TRUNC(n,m)**:

```
SELECT TRUNC(15.81,-1) "Number"
      FROM dual;
```

O/P : 10

Instructor Notes:

None

5.3: Types of Single Row Functions

Character Functions

Character functions accept "character data" as argument, and returns "character" or "number" values.

LOWER (arg)	Converts alphabetic character values to lowercase.
UPPER (arg)	Converts alphabetic character values to uppercase.
INITCAP (arg)	Capitalizes first letter of each word in the argument string.
CONCAT (arg1, arg2)	Concatenates the character strings "arg1" and "arg2".
SUBSTR (arg, pos, n)	Extracts a substring from "arg", "n" characters long, and starting at position "pos".
LTRIM (arg)	Removes any leading blanks in the string "arg".
RTRIM (arg)	Removes any trailing blanks in the string "arg".
LENGTH (arg)	Returns the number of characters in the string "arg".
REPLACE (arg, str1, str2)	Returns the string "arg" with all occurrences of the string "str1" replaced by "str2".
LPAD (arg, n, ch)	Pads the string "arg" on the left with the character "ch", to a total width of "n" characters.
RPAD (arg, n, ch)	Pads the string "arg" on the right with the character "ch", to a total width of "n" characters.
INSTR(string, pattern[, start [,occurrence]])	It returns the location of a character in a string.

Character Functions:
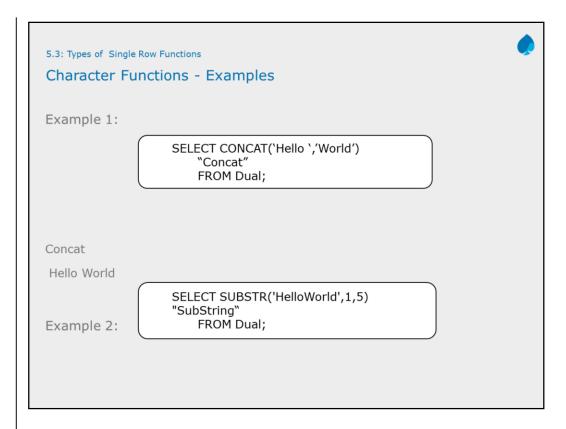**Example1:**

```
SELECT Upper('Hello'), Lower('WORLD')
FROM Dual;
```

Note: Functions can be nested to any depth. Evaluation starts with the "inner most functions", and proceeds outwards.
For example: LENGTH(LTRIM(RTRIM(name)))

Instructor Notes:

None

5.3: Types of Single Row Functions

Character Functions - Examples

Example 1:

```
SELECT CONCAT('Hello ','World')
        "Concat"
        FROM Dual;
```

Concat

 Hello World

```
SELECT SUBSTR('HelloWorld',1,5)
"SubString"
    FROM Dual;
```

Example 2:

Examples of Character Functions:
- **REPLACE function** returns char with every occurrence of search_string replaced with replacement_string.
    - ➢ If replacement_string is omitted or NULL, then all occurrences of search_string are removed.
    - ➢ If search_string is NULL, then char is returned.

```
SELECT REPLACE('JACK and JUE','J','BL') "Changes"
        FROM DUAL;
```

Changes
BLACK and BLUE

- **CONCAT function** returns "char1" concatenated with "char2". Both "char1" and "char2" can be any of the datatypes CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, or NCLOB. The string returned is in the same character set as char1. It's datatype depends on the datatypes of the arguments.
    - ➢ In concatenations of two different datatypes, the Oracle Database returns the datatype that results in a "lossless conversion". Therefore:
        - ▪ If one of the arguments is a LOB, then the returned value is a LOB.
        - ▪ If one of the arguments is a national datatype, then the returned value is a national datatype.

Instructor Notes:

5.4: Types of Single Row Functions
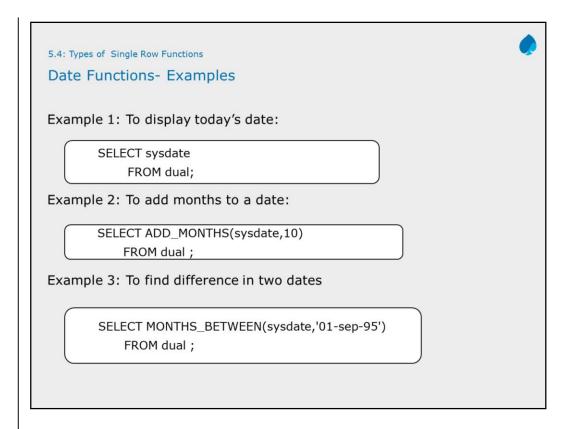
Date Functions

Date Functions operate on Date & Time datatype values

Add_Months(date1,int 1)	Returns a DATE, int1 times added, int1 can be a negative integer
Months_Between(date1,date2)	Returns number of months between two dates
Last_Day(date1)	Returns the date of the last day of the month that contains the date
Next_Day(date1,char)	Returns the date of the first weekday specified as char that is later the given date
Current_Date()	Returns the current date in the session time zone. The value is in Gregorian Calendar type
Current_Timestamp	Returns the current date and time in the session time zone. The value returned is of TimeStamp with TimeZone.
Extract(datetime)	Extracts and returns the value of a specified datetime field
Round(date,[fmt])	Returns date rounded to the unit specified . The format will be according to format model fmt
Trunc(date,[fmt])	Returns date truncated to the unit specified . The format will be according to format model fmt

DATE Functions
- Datetime functions operate on date (DATE), timestamp, and interval values.
- Some of the datetime functions were designed for the Oracle DATE datatype (ADD_MONTHS, CURRENT_DATE, LAST_DAY, and NEXT_DAY).

> If you provide a timestamp value as their argument, Oracle Database internally converts the input type to a DATE value and returns a DATE value.

> The exceptions are:
  - the MONTHS_BETWEEN function, which returns a number, and
  - the ROUND and TRUNC functions, which do not accept timestamp or interval values at all.

- The remaining datetime functions are designed to accept any of the three types of data (date, timestamp, and interval), and to return a value of one of these types.

Instructor Notes:

None

5.4: Types of Single Row Functions

Date Functions- Examples

Example 1: To display today's date:

```
SELECT sysdate
    FROM dual;
```

Example 2: To add months to a date:

```
SELECT ADD_MONTHS(sysdate,10)
    FROM dual ;
```

Example 3: To find difference in two dates

```
SELECT MONTHS_BETWEEN(sysdate,'01-sep-95')
    FROM dual ;
```
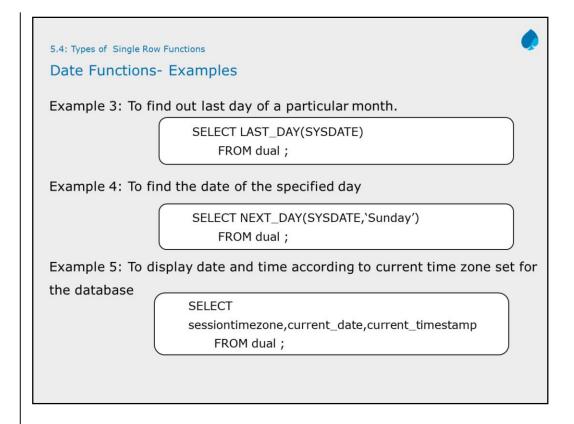

Date Functions:

ADD_MONTHS(DATE1,int1) returns the date as addition of "date" and "integer" months. The first argument can be a datetime value or any value that can be implicitly converted to DATE. The second argument can be an integer or any value that can be implicitly converted to an integer. The return type is always DATE.

```
SELECT book_code,ADD_MONTHS(book_issue_date,1)
FROM book_transaction;
```

MONTHS_BETWEEN (DATE1,DATE2) This function returns number of months between the two DATEs. The result is positive if date1 is later than date2 and result is negative if date2 is later than date2.

```
SELECT staff_code,
MONTHS_BETWEEN(TRUNC(sysdate),hiredate)
FROM staff_master;
```

Instructor Notes:

None

5.4: Types of Single Row Functions

Date Functions- Examples

Example 3: To find out last day of a particular month.

> SELECT LAST_DAY(SYSDATE)
>   FROM dual ;

Example 4: To find the date of the specified day

> SELECT NEXT_DAY(SYSDATE,'Sunday')
>   FROM dual ;

Example 5: To display date and time according to current time zone set for the database

> SELECT
> sessiontimezone,current_date,current_timestamp
>   FROM dual ;

Date Functions (contd.):

LASTDAY(date1) returns the date of the last day of the month that contains the date. The return type is always DATE.
- To display the date of the last day in the month the books were issued

> SELECT book_issue_date, LAST_DAY(book_issue_date)
>   FROM book_transaction;

NEXT_DAY(date1, char) Returns the date of the first weekday specified as char that is later the given date
- To display the date on coming Friday in the week that books were issued
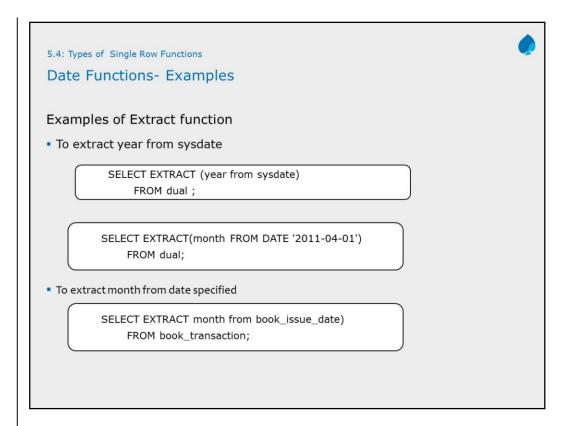
> SELECT book_issue_date,
> NEXT_DAY(book_issue_date,'Friday')
>   FROM book_transaction

CURRENT_DATE & CURRENT_TIMESTAMP return current date and time respectively base on the timezone set for the database.
The query given on the slide above shows the date and time based on the offset time base on GMT which is according to the timezone set for the database

5.4: Types of Single Row Functions

Date Functions- Examples

Examples of Extract function

- To extract year from sysdate

> SELECT EXTRACT (year from sysdate)
>   FROM dual ;

> SELECT EXTRACT(month FROM DATE '2011-04-01')
>   FROM dual;

- To extract month from date specified

> SELECT EXTRACT month from book_issue_date)
>   FROM book_transaction;

Date Functions (contd.):

EXTRACT(datetime) extracts the value of a specified datetime field. This
function is useful for manipulating datetime field values. For example you
can extract only year, month or day from a given date value.

• To display year of birth for all students

> SELECT EXTRACT(year from student_dob)
>   FROM student_master

TRUNC (DATE1)

• This function truncates the time part from the DATE. This is required
when we do DATE calculations.

> SELECT staff_name, TRUNC(sysdate) -
> TRUNC(HIREDATE) FROM staff_master;

Instructor Notes:

None

5.4: Types of Single Row Functions

Arithmetic with Dates

Use '+' operator to Add and '-' operator Subtract number of days to/from a date for a resultant date value

```
SELECT Student_code , (Book_actual_return_date –
       Book_expected_return_date) AS Payable_Days
            FROM  Book_Transaction
            WHERE  Book_Code = 1000;
```

• Given below are formats of Date Functions

Format	Meaning
YYYY	Four digit year
YY	Last two digits of the year
MM	Month (01-12 where JAN = 01 …)
MONTH	Name of the month stored as a length of nine characters.
MON	Name of the month in three letter format.
DD	Day of the month (01-31).
D	Day of the week.
DAY	Name of the day stored as a length of nine characters.
DY	Name of the day in three letter format.
FM	This prefix can be added to suppress blank padding.
HH	Hour of the day.
HH24	Hour in the 24 hour format.
MI	Minutes of the hour.
SS	Seconds of the minute.
TH	The suffix used with the day.

Instructor Notes:

In the class you can give examples on implicit datatype conversion.
Varchar2 or Char to Number
Varchar2 or Char to Date
Number to Varchar2
Date to Varchar2

5.5: Types of Single Row Functions

Conversion Functions

Conversion functions facilitate the conversion of values from one datatype to another.

TO_CHAR (arg,fmt)	Converts a number or date "arg" to a specific character format.
TO_DATE (arg,fmt)	Converts a date value stored as string to date datatype
TO_NUMBER (arg)	Converts a number stored as a character to number datatype.

Conversion Functions

Conversion functions convert values of one datatype to another. Usually the conversion function accepts two arguments wherein first is the input type and second is the output type.
Oracle takes care of implicit datatype conversion. Explicit datatype conversions are done using the conversion functions.
Although Oracle does provide implicit type conversion to ensure reliability of SQL statements you should use conversion functions

Instructor Notes:

None

5.5: Types of Single Row Functions

Conversion Functions - Examples

Example 1: To display system date in format as 29 November, 1999.

SELECT TO_CHAR(SYSDATE,'DD month, YYYY') FROM dual ;

Example 2: To display system date in the format as 29th November, 1999.

SELECT TO_CHAR (SYSDATE,'DDth month,YYYY') FROM dual ;

Example 3: To display a number in currency format.

select to_char(17000,'$99,999.00')
    FROM dual;

TO_CHAR(DATE1,format)
- This function converts the DATE given to the format specified.

SELECT TO_CHAR(TRUNC(sysdate), 'ddth fmMonth yy')
'DATE'
FROM dual;

DATE
01st January 95

SELECT TO_CHAR(TRUNC(sysdate),'fmMonth') "DATE"
        FROM dual;

DATE
July

- Example: To display the quarter which has the specified date.

SELECT to_char (sysdate , 'Q')
        FROM dual ;

Instructor Notes:

5.5: Types of Single Row Functions

Conversion Functions - Examples

Example 3: To display employees whose hire date is September 08, 1981.

```
SELECT staff_code, hiredate
    FROM staff_master
    WHERE
    hiredate = TO_DATE ('September 08,1981','Month DD, YYYY');
```

Example 4: To display the value in timestamp format

```
SELECT TO_TIMESTAMP(sysdate,'DD-MM-YY')
    from dual;
```

Instructor Notes:

5.6: Types of Single Row Functions

Miscellaneous Functions

Some functions do not fall under any specific category and hence listed as miscellaneous functions

NVL (arg1,arg2)	Replaces and returns a null value with specified actual value
NVL2(arg1,arg2,arg3	If arg1 is not null then it returns arg2. If arg1 is null then arg3 is returned
NULLIF(arg1,arg2)	Compares both the arguments, returns null if both are equal or first argument if both are unequal
CASE	Both these functions are for conditional processing, with this IF-Then-Else logic can be applied in SQL statements
DECODE	

<u>Miscellaneous Functions</u>
These functions are sometimes also called as General Functions. These functions work with any datatype values

Instructor Notes:

None

5.6: Types of Single Row Functions

Miscellaneous Functions - Examples

Example 1: To display the return date of books and if not returned

it should display today's date

```
SELECT book_code,
NVL(book_actual_return_date,sysdate)
    FROM book_transaction;
```

Example 2: To examine expected return date of book, and if null

return today's date else return the actual return date

```
SELECT book_code,
NVL2(book_expected_return_date,book_actual_return_date,
sysdate)
    FROM book_transaction;
```

Miscellaneous Single-row Functions:
NVL ()
- Many times there are records holding NULL values in a table. When an output of such a table is displayed, it is difficult to understand the reason of NULL or BLANK values shown in the output.
- The only way to overcome this problem is to replace NULL values with some other meaningful value while computing the records. This can be done using the NVL function.

NVL2()
- The NVL2 function can be thought of as an extension to NVL function. But this function examines the first value. If the value is not null then returns the second value and if null then returns the third value.

Examples for both functions are shown on the slide

Instructor Notes:

5.6: Types of Single Row Functions
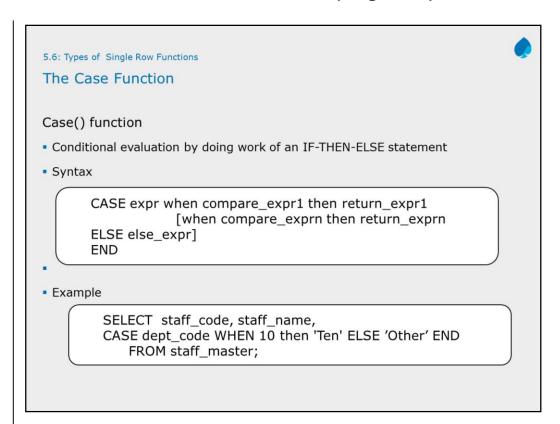
Miscellaneous Functions - Examples

Example 3:To check if the actual return date of the book is same

as the expected return date of the book

```
SELECT book_code,
NULLIF(book_expected_return_date,
book_actual_return_date)
    FROM book_transaction;
```

NullIF ()
- This function compares the arguments provided. If they are equal, the function returns null and if not then it returns the first argument.
- You cannot specify the literal NULL as the first argument.
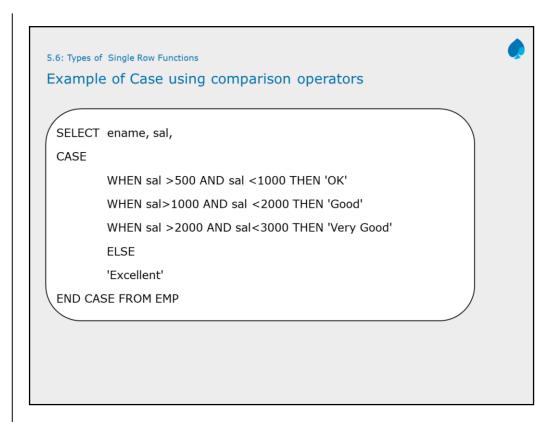
Example shown on the slide

Instructor Notes:

5.6: Types of Single Row Functions

The Case Function

Case() function

- Conditional evaluation by doing work of an IF-THEN-ELSE statement
- Syntax

> CASE expr when compare_expr1 then return_expr1
>   [when compare_exprn then return_exprn
> ELSE else_expr]
> END

-

- Example

> SELECT staff_code, staff_name,
> CASE dept_code WHEN 10 then 'Ten' ELSE 'Other' END
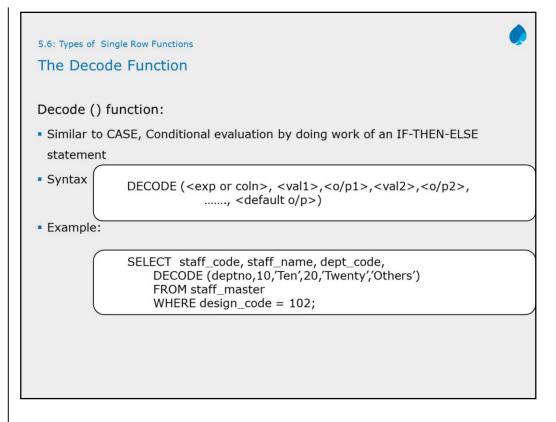>   FROM staff_master;

CASE()
In a simple case expression, Oracle searches for the first WHEN…THEN
pair for which expr is equal to comparison expr and returns return_expr.
The expressions used should be of same datatype.
CASE is capable of more logical comparisons like <> etc..
Also CASE can work with predicates and subqueries

Instructor Notes:

5.6: Types of Single Row Functions

Example of Case using comparison operators

```
SELECT  ename, sal,
 CASE
          WHEN sal >500 AND sal <1000 THEN 'OK'
          WHEN sal>1000 AND sal <2000 THEN 'Good'
          WHEN sal >2000 AND sal<3000 THEN 'Very Good'
          ELSE
          'Excellent'
 END CASE FROM EMP
```

5.6: Types of Single Row Functions

The Decode Function

Decode () function:

- Similar to CASE, Conditional evaluation by doing work of an IF-THEN-ELSE statement

- Syntax

  DECODE (<exp or coln>, <val1>,<o/p1>,<val2>,<o/p2>,
              ...., <default o/p>)

- Example:

  SELECT staff_code, staff_name, dept_code,
      DECODE (deptno,10,'Ten',20,'Twenty','Others')
      FROM staff_master
      WHERE design_code = 102;

DECODE () function
- This function decodes the 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.
- As compared to CASE, DECODE can do an equality check only. The expressions in DECODE can work only on scalar values.
- DECODE can work as a function inside SQL only but CASE can be more efficient substitute in PL/SQL blocks.

Instructor Notes:

None

5.7: Tips and Tricks

Quick Guidelines

If possible, try avoiding the SUBSTRING function in the WHERE clauses.

- Depending on how it is constructed, using the SUBSTRING function can force a table scan instead of allowing the Optimizer to use an Index (assuming there is one).

- Instead, use the LIKE condition, for better performance.

> WHERE SUBSTRING(column_name,1,1) = 'b'

- For example: Use the second query instead of using the first query.

> WHERE column_name LIKE 'b%'

Tips and Tricks:
- If possible, try avoiding the SUBSTRING function in your WHERE clauses.
- Depending on how it is constructed, using the SUBSTRING function can force a table scan instead of allowing the Optimizer to use an Index (assuming there is one).
  - ➢ If the substring you are searching for does not include the first character of the column you are searching for, then a table scan is performed.
- ➢ In case of conversion functions, If value to be converted is not in right format, then Oracle will throw an error

Instructor Notes:

None

Summary

In this lesson, you have learnt:

- SQL (single-row) functions
- Character functions
- Number functions
- Date functions
- Conversion functions
- Miscellaneous functions

Summary

Instructor Notes:

**Answers to Review
Questions:
Question 1:**
Answer: Option 4

Question 2:
Answer: initcap

Review – Questions

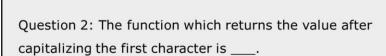Question 1: Single row functions can be broadly classified
as _____.

- Option 1: character functions
- Option 2: numeric functions
- Option 3: Date functions
- Option 4: all the above

Question 2: The function which returns the value after
capitalizing the first character is ____.

Instructor Notes:

**Answers to Review
Questions:
Question 3:**
Answer: $17,000.00

Question 4:
Answer: False

Review – Questions

Question 3: The output of the following function will be

____.
▪ select to_char(17000,'$99,999.00') 'Amount' from dual;

Question 4: The decode function can convert NULL values
to required type.
▪ True / False

Instructor Notes:

Answers to Review Questions:

Question 5:
Answer: Option 2

Review – Questions

Question 5: The function which returns the last date of the month is ____.

- Option 1: LAST_DATE

- Option 2: LAST_DAY

- Option 3: MONTH_LAST_DATE

- Option 4: MONTH_LAST_DAY