Capgemini

ORACLE SQL

Lesson 02: Operators, Single row functions

Lesson Objectives

To understand the following topics:

- Operators – Mathematical, comparison, Logical
- Distinct clause
- SQL (single-row) functions
  - Number functions
  - Character functions
  - Date functions
  - Conversion functions
  - Miscellaneous Single-row functions

Mathematical, Comparison & Logical Operators

- Mathematical Operators:
  - Examples: +, -, *, /
- Comparison Operators:

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or Equal to
<	Less than
<=	Less than or Equal to
<>, !=, or ^=	Not Equal to

- Logical Operators:
  - Examples: AND, OR, NOT

Other Comparison Operators

Other Comparison operators	Description
[NOT] BETWEEN x AND y	Allows user to express a range. For example: Searching for numbers BETWEEN 5 and 10. The optional NOT would be used when searching for numbers that are NOT BETWEEN 5 AND 10.
[NOT] IN(x,y,…)	Is similar to the OR logical operator. Can search for records which meet at least one condition contained within the parentheses. For example: Pubid IN (1, 4, 5), only books with a publisher id of 1, 4, or 5 will be returned. The optional NOT keyword instructs Oracle to return books not published by Publisher 1, 4, or 5.

Other Comparison Operators

Other Comparison operators	Description
[NOT] LIKE	Can be used when searching for patterns if you are not certain how something is spelt. For example: title LIKE ‘TH%’. Using the optional NOT indicates that records that do contain the specified pattern should not be included in the results.
IS[NOT]NULL	Allows user to search for records which do not have an entry in the specified field. For example: Ship date IS NULL. If you include the optional NOT, it would find the records that do not have an entry in the field. For example: Ship date IS NOT NULL.

BETWEEN … AND Operator

The BETWEEN … AND operator finds values in a specified range:

```
SELECT staff_code,staff_name  FROM  staff_master
   WHERE  staff_dob
            BETWEEN '01-Jan-1980'
            AND  '31-Jan-1980';
```

IN Operator

The IN operator matches a value in a specified list.

- The List must be in parentheses.
- The Values must be separated by commas.

```
SELECT dept_code   FROM  department_master
WHERE  dept_name IN ( 'Computer Science', 'Mechanics');
```

LIKE Operator

The LIKE operator performs pattern searches.

- The LIKE operator is used with wildcard characters.
- Underscore (_) for exactly one character in the indicated position
- Percent sign (%) to represent any number of characters

```
SELECT book_code,book_name FROM  book_master
        WHERE  book_pub_author LIKE '%Kanetkar%' ;
```

Logical Operators

Logical operators are used to combine conditions.

- Logical operators are NOT, AND, OR.
  - NOT reverses meaning.
  - AND both conditions must be true.
  - OR at least one condition must be true.
- Use of AND operator

```
SELECT staff_code,staff_name,staff_sal FROM  staff_master
      WHERE  dept_code = 10
        AND  staff_dob > '01-Jan-1945';
```

Operator Precedence

- Operator precedence is decided in the following order:

Levels	Operators
1	* (Multiply), / (Division), % (Modulo)
2	+ (Positive), - (Negative), + (Add), (+ Concatenate), - (Subtract), & (Bitwise AND)
3	=, >, <, >=, <=, <>, !=, !>, !< (Comparison operators)
4	NOT
5	OR
6	AND
7	ALL, ANY, BETWEEN, IN, LIKE, OR, SOME
8	= (Assignment)

The DISTINCT clause

The SQL DISTINCT clause is used to eliminate duplicate rows.

- For example: Displays student codes from student_marks tables. the student codes are displayed without duplication

```
SELECT DISTINCT student_code
      FROM student_marks;
```

Quick Guidelines

In a WHERE clause, the various “operators” that are used, directly affect the query performance.

- Given below are the key operators used in the WHERE clause, ordered by their performance. The operators at the top produce faster results, than those listed at the bottom.
  - =
  - >, >=, <, <=
  - LIKE
  - <>
- Use “=” as much as possible, and “<>” as least as possible.

Quick Guidelines

Suppose you have a choice of using the IN or the BETWEEN clauses. In such a case use the BETWEEN clause, as it is much more efficient.

- For example: The first code is much less efficient than the second code given below.

```
SELECT customer_number, customer_name
FROM customer
WHERE customer_number in (1000, 1001, 1002, 1003, 1004)
```

```
SELECT customer_number, customer_name
FROM customer
WHERE customer_number BETWEEN 1000 and 1004
```

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

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 n^{th} 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.
ABS (arg)	Returns the absolute value of “arg”.
MOD (arg1, arg2)	Returns the remainder obtained by dividing “arg1” by “arg2”.

Number Functions - Examples

Example 1:

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

Absolute
15

Example 2:

```
SELECT POWER(3,2) "Raised"
    FROM dual;
```

Raised
9

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

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

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

Number
15.8

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.

Character Functions - Examples

Example 1:

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

Concat

Hello World

Example 2:

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

SubSt

Hello

Date Functions

- Date Functions operate on Date & Time data type values

Add_Months(date1,int1)	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

Sysdate function returns the current date and time

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- 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 ;16
```

Date Functions- Examples

Examples of Extract function

- To extract year from sysdate

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

- To extract month from date specified

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

- To extract month of issue for all books

```
SELECT EXTRACT month from book_issue_date)
    FROM book_transaction;
```

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;
```

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.
TO_TIMESTAMP(arg,fmt)	Converts character type to a value of timestamp datatype

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;
```

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;
```

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
COALESCE(arg1,arg2…argn)	Returns the first non null value in the given list
CASE	Both these functions are for conditional processing, with this IF-Then-Else logic can be applied in SQL statements
DECODE	

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 Functions - Examples

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

E sysdate

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

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

The Case Function

Case() function

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

Exan

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

```
SELECT  staff_code, staff_name,
CASE dept_code WHEN 10 then ED' ELSE ’Other’ END
     FROM staff_master;
```

The Decode Function

Decode () function:

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

Syntax

Example:

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

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

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

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

```
WHERE column_name LIKE 'b%'
```

SUMMARY

- In this lesson, you have learnt:
  - Operators - Mathematical, comparison, Logical
  - SQL (single-row) functions
    - Character functions
    - Number functions
    - Date functions
    - Conversion functions
    - Miscellaneous Single-row functions

Review Question

❖ Question 1: The LIKE operator comes under the ___ category.

- Option 1: mathematical
- Option 2: comparison
- Option 3: logical

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

❖ Question 3: 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