Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Structured Query Language

Session 5

Coverage

- Set Operators
- Working with SQL Functions

Set Operators

Set Operators

- Union
- Intersection
- Minus
- Multiple queries are joined using the Set Operators.
- Queries should match in terms of number and data-type of of columns

Union

- The data returned by the queries is “AND” .
- Duplicate values are not repeated
- Returns unique match from both the queries
- Union All will retain duplicate rows

- **List the employees working in Pune and Hyderabad**

```
 select kin_no,emp_name from emp_pune
 union
 select kin_no,emp_name from emp_hyd
```

- **List the depts with or without employees**

```
select  deptno from emp
union
select  deptno from dept
```

Intersect

- Query returns the common values returned by the multiple queries.
- Helps to find duplicate rows

- **Display employees who have worked in both the locations**

```
select kin_no,emp_name from emp_pune
intersect
select kin_no,emp_name from emp_hyd
```

- **List the depts with employees**

```
select deptno from emp
Intersect
select deptno from dept
```

Minus

- The query list the rows not present in the preceding queries

- **List the employees who have worked only in Pune**

```
select kin_no,emp_name from emp_pune
minus
select kin_no,emp_name from emp_hyd
```

- **List the dept without employees**

```
select deptno from dept
 minus
select deptno from emp
```

Working with SQL Functions

What is SQL Function

- Functions work on the arguments provided to manipulate data value and return a result
- SQL Functions are built-in functions provided by Oracle to be used by SQL statements
- SQL functions are used for :
  - Perform calculations
  - Modify individual data
  - Manipulate output for groups of rows
  - Format dates and numbers
  - Convert column data types

Types of Functions

- Single Row Functions :
  - String/Character
  - Date and Time
  - Number
  - Conversion
  - Common
- Multi-row functions
  - Aggregate - Group/ Columnar Functions

Single-Row Function

- Single-row functions return a single result row for every row of a queried table.
- The function can be used in select lists , where and order by clause

Single- row String Functions

Lower(str)	Converts the string into lower case
Upper(str)	Converts the string into upper case
Initcap(str)	Converts the string into proper case
Length(str)	Returns the number of characters in the string
Lpad(str,length,char_set)	Pads the character specified by character set upto the length of the string on the left side
Rpad(str,length,char_set)	Pads the character specified by character set upto the length of the string on the right side

String functions..

Substr(str,starting_position, no_of_char)	Returns the characters from the position specified by starting position upto the number of characters specified
Replace(str,search_str, replace_str)	Replaces the characters in string specified in the search string with the replace string
Rtrim(str,characters)	Trims the specified characters from the right side of the string.By Default trims blank characters/spaces
Ltrim(str,characters)	Trims the specified characters from the left side of the string.By Default trims blank characters/spaces
Trim(str)	Trims spaces from the string

Self Study

- Instr
- Concat
- RR Date format

Examples of Single-row functions

```
select ename, lower(ename), upper(ename),
  initcap(ename),length(ename) from emp;
select  rpad(job,20,'*') from emp;
select ename, ltrim(ename), ltrim(ename,'SM'),
  rtrim(ename,'NS') from emp;
select ename,substr(ename,2,5) from emp;
select ename, replace(ename,'AM','*#')  from emp;
```

Date Functions

Add_months(date, no_of_months_to_add)	Returns a date after adding number of specified months to the date
Last_day(date)	Returns the last date of the month for the given date
Months_between(date1,date2)	Returns the number of months between two dates
Next_day(date,'day')	Returns the next date after the specified date for the specified day.

Dual table

- Is owned by the user SYS
- Can be accessed by all users.
- Contains one column, DUMMY, and one row with the value X.
- Is used when the data is not required to be selected from the table

Examples of Date functions

```
select sysdate, last_day(sysdate) from dual;

select ename,hiredate,
  months_between(sysdate,hiredate) "worked",
  add_months(hiredate,6) "confirm"
  from emp

select sysdate,next_day(sysdate,'MON')
  from dual;
```

Numeric functions

ABS(number)	Returns the absolute value of the number
Mod(number,divisor)	Returns modules of the number divided by the divisor
Power(number,exponent)	Returns the value raised to an exponent power
Sqrt(number)	Returns the square root of the number
Sign(number)	Returns 1 if number is positive or negative or 0 if zero

Examples on Numeric functions

- select **abs(-100),mod(36,5),power(2,4),sqrt(121)** from dual;

Common functions

Round(value,format_specifier) For – date, number	Rounds the value of number/date to the specified precision/format in the format specifier
Truncate(value,format_specifier) For - date,number	Truncates the value of number/date to the specified precision/format in the format specifier
NVL(value,substitute_value) For – date,number,character	Substitutes the NULL value with the specified substitute value
Greatest(value1,value2,value3..) For – date,number	Returns the greatest value from the list specified
Least(value1,valu2,value3…) For – date,number	Returns the least value from the list specified

If.. then logic

Decode(value,condition1,value1, condition2,valu2,……. default_value) For – date,number,character	For the value, checks the conditions and substitutes the corresponding value specified if the condition is true Condition is compared for equality only
Case <expression> when <cond_1> then <Return_value1> when <cond_2> then <Return_value2> else <Return_value_n> end	For the expression, checks the conditions and substitutes the corresponding value specified if the condition is true Condition specified can have relational and logical operators

Examples for common functions

```
select sysdate ,round(sysdate,'month') from dual;
select sysdate, trunc(sysdate,'month') from dual;
select round(186.956, 2),round(186.956,-2) from dual;
select trunc(186.956, 2),trunc(186.956,-2) from dual;
 select ename,sal,comm, greatest(sal,comm), least(sal,comm)  from emp
 select ename,sal,nvl(comm,0)  from emp;
 select ename,sal,comm, greatest(sal,nvl(comm,0)) from emp;
select greatest('01-JAN-85','31-MAR-99','01-APR-95') from dual;
select nvl(hiredate,'01-JAN-00') from emp;
select nvl(to_char(hiredate),'Not Entered') from emp;
select empno,ename, decode(job,'MANAGER','M','CLERK','C','O') from emp;
```

Case examples

```
select empno,ename, job,
case job
when 'MANAGER' then 'M'
when'CLERK' then 'C'
else 'O' end  from emp

select empno,ename, job,sal,
case
when sal > 4000 then 'High Salary'
when sal > 2000 then 'Meduim Salary'
else 'Low salary' end  from emp
```

Conversion functions

To_char(number,format_specifier)	Converts a numeric value into character value according to the format specifier
To_char(date,format_specifier)	Converts the date into character value according to the format specifier
To_Number(character_number, format_specifier)	Converts a numeric value of character type as specified by format specifier to number
To_date(character_date, format-specifier)	Converts a character date as specified by the format sepecifier to date in default format

Implicit Conversion

From	To
Varchar2 or Char	Number
Varchar2 or Char	Date
Number	Varchar2
Date	Varchar2

Examples of Conversion functions

```
Select to_char(123,'$999.999') from dual;

Select ename, to_char(hiredate,'dd/mm/yyyy ') from
  emp;

 Select to_date('01/03/1999','dd/mm/yyyy ') from
  dual;

 select to_number('$1000','$9999999') from dual;
```

Number format

,	9,9999	Puts comma at the specified position
.	9999.999	Puts decimal at the specified position
$	$9999	Puts currency symbol at the specified position
0	0999	Returns leading zeros
9	9999	Returns the number with specified digits
B	B9999	Returns blank when zero
MI	9999MI	Returns a negative number with trailing minus sign
PR	9999PR	Returns a negative number in <angular_brackets>
S	S9999 9999S	Returns a negative number with leading minus sign Returns a positive number with leading positive sign Returns a negative number with trailing minus sign Returns a positive number with trailing positive sign
L	L9999	Puts local currency symbol set by NLS_CURRENCY

Date Formats

- / , . ; : “text”	Puts the specified symbols or text in the date
AD or A.D.	Specifies AD or A.D. of the century
AM or A.M.	Specifies AM or A.M. of time zone
BC or B.C.	Specifies BC or B.C. of the century
CC , SCC	Specifies first 2 digits of the Century of a four digit year. S prefixes BC dates with -
D	Day of week (1 to 7)
DAY	Name of the Day - MONDAY
DD	Day of month (1 to 31)
DDD	Day of year (1- 366)
DY	Abbreviated name of day – mon,tue

Date Formats …

HH or HH12	Hour of the day (1-12)
HH24	Hour of day (0-23)
MI	Minute (0-59)
MM	Two-digit numeric abbreviated of month(1,2,3)
MON	Abbreviated name of month (Jan,Feb)
MONTH	Name of month
PM or P.M.	Meridian indicator
Q	Quarter of year
SS	Second (0-59)

Date Formats ….

WW	Week of year (1-53)
W	Week of month (1 –5)
YYYY, SYYYY	4-digit year . S prefixes BC dates with -
YEAR , SYEAR	Year spelled out. S prefixes BC dates with -
YYY , YY, Y	Last 3,2 or 1 digit of year
TH	Ordinal number e.g. 4th
SP	Spelled number FOUR
SPTH or THSP	Spelled, ordinal number - FOURTH

Examples of Date formats

```
select ename, to_char(hiredate,'dd/mm/yyy') from
  emp;
```

```
select to_char(sysdate,'"today is " ddth  "of "
  Month YEAR ,   WW "week of the year"') from dual;
```

```
select to_char(sysdate,'"today is " ddspth  Mon
  YYYY ,   " and time "  HH24:MI:SS') from dual;
```

Group/ Aggregate Functions

- Aggregate functions return a single result row based on groups of rows
- They are commonly used with the GROUP BY clause in a SELECT statement

Count(*/column)	Counts the number of rows for the specified column for group of rows
Max(column)	Returns the maximum value for the specified column for group of rows
Min(column)	Returns the minimum value for the specified column for group of rows
Sum(column)	Returns the total of all values for the specified column for group of rows
Avg(column)	Returns the average value for the specified column for group of rows

Examples

- **Display total number of employees, total , maximum , minimum and average salaries paid**

```
select count(*),sum(sal),max(sal),min(sal),avg(sal) from emp;
```

- **Display maximum , minimum and average salaries paid in department 30 / SALES**

```
select max(sal),min(sal),avg(sal) from emp
where deptno = 30


select max(sal),min(sal),avg(sal)
from emp e, dept d
where e.deptno = d.deptno
and dname = 'SALES‘
```

Examples

- **Find out the difference between maximum and minimum salaries paid**

```
select max(sal)-min(sal) "Difference" from emp;
```

- **Find out how many distinct jobs are held**

```
select count(distinct job) from emp
```

All work described was performed by Capgemini or a Capgemini affiliate

Nulls in SQL Functions

- All scalar functions (except REPLACE, NVL, and CONCAT) return null when given a null argument
- Most aggregate functions ignore nulls.

NULLIF function

- NULLIF is used for comparison between two expressions.

e.g. NULLIF(exp1,exp2)

If exp1=exp2 the function will return null.
Else function will return exp1.

```
select e.empno,e.ename, e.job, b.job , nullif(e.job,b.job)
from emp e, bonus b
where e.ename = b.ename;
```

NVL2 Function

- NVL2 function takes 3 parameters. If 1st parameter is not null function returns 2nd parameter else it returns 3rd parameter.

NVL2 (pr1,pr2,pr3)

If pr1 is not null returns pr2

Else returns pr3.

e.g.

select empno,ename, sal, comm, nvl2(comm, comm,sal) from emp;

COALESCE Function

- COALESCE function take multiple parameters and returns 1st non null parameter.

e.g. COALESCE (pr1,pr2,…….prn)

If pr1 is not null function will return pr1

Else it performs COALESCE of remaining parameters.

e.g.

select empno,ename, sal, comm, COALESCE(comm,sal,1000)

from emp

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Thank You !!!