**Conversion Function:**

■The Conversation functions convert a value from one data type to another.

■The Data type conversion in Oracle is of two types.

\*\*Implicit Data type Conversion

\*\*Explicit Data type Conversion

**Implicit Data type Conversion:**

■Implicit Date type conversion work according to the convention specified by oracle.

■The Assignment succeeds if the Oracle server can convert the date type of value.

■CHAR to NUMBER conversion succeed only if the character string represents a valid NUMBER.

■CHAR to DATES conversion succeed only if the character string represents the default format of DD-MON-YY.

**In Assignment Operator:**

•Varchar2/Char 🡪Number

•Varchar2/Char🡪Date

•number🡪Varchar2

•Date🡪Varchar2

**Explicit Data type Conversion:**

• SQL provided three function to convert a value from one data type to another.

•The explicit conversation function is

o TO\_CHAR --------To character conversion.

o TO\_DATE -------- To Date conversion.

o TO\_NUMBER-------To Number Conversion.

**TO\_CHAR CONVERSION:**

This function can be used in two ways.

o TO\_CHAR (Number Conversion)

o TO\_CHAR (Date Conversion)

**TO\_CHAR (NUMBER CONVERSION)**

Syntax: TO\_CHAR (NUMBER, fmt)

• Converts Number of Number data type to a value of VARCHAR2 data type.

• 'fmt' is the optional number format, that can be used.

**Decimal Indicator:D-->99D99**

• It returns the specified position of the decimal character .

• The default decimal delimiter is period '.'

• Only one decimal indicator can be specified in a number format model.

Sql> Select 1234,TO\_CHAR(1234,'9999D99') From Dual;

o/p: 1234.00

Sql> Select 1234,TO\_CHAR(1234,'999D99') From Dual;

o/p: #######

Sql> select 12345.10,TO\_CHAR(12345.10,'99999D9') from dual;

o/p:12345.1

Sql> select 12345.10, TO\_CHAR (12345.10,'999D9') from dual;

o/p: ######

**Group Separator: G---9G999**

• Returns the specified position of the Group separator

• Multiple Group separators can be specified.

Sql>Select TO\_CHAR(1234567,'99G99G9999') From Dual;

Sql> Select sal,To\_char(sal,'9G999') From Emp;

**Local Currency Indicator: L🡪L999 OR 999L**

Returns the specified position of the local currency symbol.

Sql>Select 1234,TO\_CHAR(1234,'L9999') From Dual;

Sql>Select Sal,TO\_CHAR(Sal,'L999999') Currency From Emp Where Deptno = 10;

Sql>Select Sal,TO\_CHAR(Sal,'L99G999D99','NLS\_CURRENCY=IndRupees') Sal From Emp Where Deptno = 20;

**Trailing Minus Indicator:** MI🡪9999MI

• Return negative value with a trailing minus sign'-'.

• Returns positive value with a trailing Blank.

• 'MI' Format should be declared as Trailing argument only.

Sql>select -10000,TO\_CHAR(-10000,'L99G999D99MI')From Dual;

Sql>select Sal,Comm,Comm-Sal,TO\_CHAR(Comm-Sal,'L99G999D99MI') From Emp;

**Negative Number Indicator:**

PR🡪9999PR

Returns negative number in '<>'.

It Can appear only as trailing declaration.

Sql> select TO\_CHAR(-1000,'L99G999D99PR') From Dual;

Sql> select Sal,Comm,Comm-Sal,TO\_CHAR(Comm-Sal,'L9999PR') From Emp;

**Roman Number Indicator:**

* RN->Returns Upper Roman Number.
* rn-> Returns Lower Roman Number.
* The value can be an integer between 1 and 3999.

Sql> select 12,TO\_CHAR(12,'RN'),TO\_CHAR(12,'rn') From Dual;

o/p: XII and xii

**Sign lndicator:S--S99999 OR 99999S**

* Returns Negative value with a leading Minus Sign
* Returns Positive value with a leading Plus Sign.
* Returns Negative value with trailing minus sign.
* Returns Positive value with a trailing plus sign.
* ‘S’ can appear as first or last value.

Sql> SELECT 1000,TO\_CHAR(1000,'S9999'),TO\_CHAR(-1000,'S9999') from dual;

Sql>SELECT 1000,TO\_CHAR(1000,'9999S'),TO\_CHAR(-1000,'9999S') from dual;

**Group separator:** 🡪9,999

* It returns a comma in the specified position.
* Multiple commas can be specified.

**Decimal indicator:** 🡪99.99

* It returns a decimal point, at the specified position.
* Only one period can be specified in a number format model.

Sql>SELECT 20000, TO\_CHAR (20000,'99,999.99') FROM DUAL;

Sql> SELECT ENAME,SAL,TO\_CHAR(SAL,'L99,999.99') FROM EMP;

**Dollar indicator:** $ 🡪$9999

* Return value with a leading dollar sign.

Sql> Select 20000,TO\_CHAR(20000,'$99,999.99') From Dual;

Sql> Select Ename,Sal,TO\_CHAR(Sal,'$99,999.99') From Emp;

**Zero Indicator**:0🡪0999 OR 9990

* Returns Leading OR Trailing Zeros.

Sql> SELECT 1000, TO\_CHAR (1000,'0999999'),

TO\_CHAR (1000,'09999.990') FROM DUAL;

Sql> SELECT ENAME,SAL,TO\_CHAR(SAL,'$099,999.99') FROM EMP;

**Digit place maker:9🡪9999**

* Returns value with a specified number of digits with a Leading space when positive or Leading minus when Negative.

Sql> SELECT 1000,600,TO\_CHAR(1000-600,'99999'),TO\_CHAR(600-1000,'99999'),TO\_CHAR(20.55-19,'99999'),TO\_CHAR(20.11-20,'99999') FROM DUAL;

**ISO Currency Indicator:** C🡪C9999

* Return specified position of the ISO Currency Symbol.

Sql>select 1000,TO\_CHAR(1000,'C9999.99') FROM DUAL;

Sql>select Ename,Sal, TO\_CHAR(Sal,'C9999.99') FROM Emp;

**TO \_CHAR (Date Conversion)**

**Syntax: TO\_CHAR (DATE, fmt)**

•Converts Date of Date data type to a value of VARCHAR2 data type in the format specified.

•'fmt' is the optional Date format, that can be used.

**Date Format Models:**

• The date format models can be used in the TO\_CHAR function to translate a DATE value from original format to user format.

**Date Format Elements:**

• A Date format model is composed of one or more date format elements.

• For input format models, format items cannot appear twice, and format items that represents similar information cannot be combined.

• Capitalization in a spelled word, abbreviation, or roman numeral follows capitalization in the corresponding format element.

• Punctuation such as Hyphens, Slashes, Commas, periods and Colons.

**AD or A.D./BC or B.C. Indicator**:

* Indicator AD/BC with or without periods.

Sql>Select Sysdate,TO\_CHAR(Sysdate,'AD') From Dual;

Sql>Select TO\_CHAR(Sysdate,'B.C.'),

TO\_CHAR(Sysdate, 'A.D.') From Dual;

Sql>Select Ename,Sal,Hiredate,TO\_CHAR(Hiredate,'A.D.') From Emp;

**Meridian Indicator**:

• It indicates meridian indicator with or without periods.

Sql>Select Sydate,TO\_CHAR(Sysdate,'A.M'),

TO\_CHAR(Sysdate,'PM') From Dual;

Sql>Select Ename,Sal,Hiredate,TO\_CHAR(Hiredate,'AM') From Emp;

**Century Indicator**: CC

• Indicates the century.

Sql> Select Sysdate,TO\_CHAR(Sysdate,'CC-AD') From Dual;

**Numeric Week Day Indicator**:D🡪 (l-7)

• Returns the week day number.

Sql>Select Sysdate,TO\_CHAR(Sysdate,'D') From Dual;

Sql>Select Ename,Sal,Hiredate,TO\_CHAR(Hiredate,'D') From Emp;

**Week Day Spelling Indicator: >Day**

• Pads to a length of 9 characters.

Sql> Select Sysdate,TO\_CHAR(Sysdate, 'DAY') From Dual;

Sql> Select Ename,Hiredate, TO\_CHAR(Hiredate,'DAY')

From Emp Where TO\_CHAR(Hiredate, 'DY') = 'MON';

**Month day Indicator**: DD

• It indicates the day of the Month(1-31)

Sql> Select Sysdate,TO\_CHAR(Sysdate,'DD-DAY') From Dual;

Sql> Select Ename,Hiredate,TO\_CHAR(Hiredate,'DD\_DAY') From EmP;

Sql> Select Ename,Hiredate,TO\_CHAR(Hiredate,'DY') From Emp

Where TO\_CHAR(Hiredate,'DD-DY') = '28-MON';

**Year day indicator**:DDD

• It indicates the day of the Year(1-366)

Sql> Select Sysdate,TO\_CHAR(Sysdate,'DDD') From Dual;

Sql> Select Ename,Hiredate, TO\_CHAR(Hiredate,'DDD') From Emp;

Sql> Select Ename,Hiredate,TO\_CHAR(Hiredate,'DDD') From Emp

Where TO\_CHAR(Hiredate, 'DY')= 'MON';

**Abbreviated week day**: DY

Sql> Select Sysdate,TO\_CHAR(Sysdate,'D-DY-DAY') From Dual;

Sql> Select Ename,Hiredate,TO\_CHAR(Hiredate,'D-DY-DAY') From emp Where Deptno in(10,20);

**ISO standard year week indicator**- IW

* Specifies the week of the year(1-52 or 1-53) based on the ISO standard.

Sql> Select Sysdate,TO\_CHAR(Sysdate, 'IW') From Dual;

Sql>Select Ename,Hiredate,TO\_CHAR(Hiredate,'IW') From Emp;

**Year week Indicator**: WW

* Specifies the week of the year (1-53)
* Week 1 starts on the day of the year and continues to the seventh day in that year.

Sql> Select Sysdate,TO\_CHAR (Sysdate, 'WW') From Dual;

Sql>Select Empno,Ename,Hiredate,TO\_CHAR(Hiredate, 'WW') From Emp;

**ISO standard 4 digit year indicator**: IYYY

• Specifies 4-digit year based on the ISO standard.

• It can even be used in combination of IYY,IY,I.

Sql> Select Sysdate, TO\_CHAR(Sysdate,'IYYY') From Dual;

Sql> Select Ename, Hiredate, TO\_CHAR(Hiredate,'IYYY') From Emp

Where TO\_CHAR(Hiredate,'DY’) ='MON';

**Four Digit Year Indicator: YYYY**

• It can even be used in combination of YYY or YY or Y.

• Y, YYY returns year with comma in that position.

Sql>Select to\_char(sysdate, 'yyyy') from dual;

Sql>Select Hiredate, TO\_CHAR(Hiredate,'YYYY'),

TO\_CHAR(Hiredate,'YYY') From Emp;

Sql>Select TO\_CHAR(Sysdate, 'YYYY),TO\_CHAR(Sysdate, 'YYY'),

TO\_CHAR(Sysdate,'YY') From Emp;

**Spelled Year Indicator**: YEAR

• Returns the numerical year in spelling.

Sql> select sysdate,to\_char(sysdate,'YEAR'),

TO\_CHAR(sysdate,'Year'),to\_char(sysdate,'year') from dual;

Sql>select empno,ename,hiredate,to\_char(hiredate,’YEAR’) from emp;

**Week of the month indicator**: W

• Specifies the week of the month(1-5).

• Week starts on the first day of the month and ends on the seventh day.

Sql> Select Sysdate,TO\_CHAR(Sysdate,'W') From Dual;

Sql> Select Empno,Ename,Hiredate,TO.CHAR(Hiredate, 'W') From Emp;

**Quarter of the Year Indicator**: Q

• Returns the Quarter of the year.

• Quarter starting with the month of january and ending with every three months.

Sql> Select Sysdate,TO\_CHAR(Sysdate,'Q') From Dual;

Sql> Select Empno,Ename,Hiredate,TO\_CHAR(Hiredate,'Q') from emp Where TO\_CHAR(Hiredate,'Q')=3;

**Julian Day Indicator**: J

• Returns the JULIAN Day of the given date.

• It is the number of day since January 1,4712 BC.

• Number specified with 'J' must be integers.

Sql> Select Sysdate,TO\_CHAR(Sysdate,'J') From Dual;

Sql> Select Empno,Ename,Hiredate,TO\_CHAR(Hiredate, 'J-DDD-DD-D') From Emp;

**Number Month Indicator**: MM

• Returns the numeric abbreviation of the month.

Sql> Select Sysdate,TO\_CHAR(Sysdate,'MM-YYYY') From Dual;

Sql> Select Ename, TO\_CHAR(Hiredate,'DD-MM-YYYY') From Emp Where TO\_CHAR(Hiredate,'MM') = 12;

**Abbreviated Month Indicator**: MON

• Returns the abbreviated name of the Month.

Sql> Select Sysdate,TO\_CHAR(Sysdate,'MON') From Dual;

Sql> Select Hiredate,TO\_CHAR(Sysdate,'MONTH') From Emp;

**Twelve Hour Clock Mode**: HH or HH12

* It is default clock mode.
* Returns the hour of the day in twelve-hour clock mode.

Sql> Select Sysdate,TO\_CHAR(Sysdate,'HH'),

TO\_CHAR(Sysdate,'HH12,PM') From Dual;

Sql> Select Ename,Hiredate,TO\_CHAR(Hiredate,'HH12:PM') From Emp;

**Twenty Hour Clock Mode**: HH24

• Returns the hour of the day in twenty-four-hour clock mode. (0-23)

Sql> Select Sysdate,TO\_CHAR(Sysdate,'HH24') From Dual;

**Minutes Indicator**: MI

• Returns the minutes from the given date(0-59).

Sql> Select Sysdate,TO\_CHAR(Sysdate,'MI') From Dual;

Sql> Select Ename,Hiredate,TO\_CHAR(Hiredate,'HH:MI') From Emp

Where Deptno=20;

**Roman Month Indicator**: RM

• Return the roman numeral month(I-XII).

Sql> select sysdate,to\_char(sysdate,'RM'),to\_char(sysdate,'DD-RM-YY') from dual;

Sql> Select Ename,Hiredate,TO\_CHAR(Hiredate,'DD-RM-YY')From Emp Where Deptno=20;

**Seconds Indicator**: SS

• Returns seconds from the given the dates(0-59)

Sql> Select Sysdate, TO\_CHAR(Sysdate,'SS'),

TO\_CHAR(Sysdate,'HH:MI:SS') From Dual;

Sql> Select Sysdate,TO\_CHAR(Sysdate,'RM'),

TO\_CHAR(Sysdate,'Day-Month-Year HH24:MI:SS PM.') From Dual;

Sql> Select Ename,Hiredate,TO\_CHAR(Hiredate,'HH24-MI-SS') From Emp

Where Deptno=30;

**Date Format Punctuators**:

• The punctuation marks that can be used in date formats are...

'\_', '/', '!', '.', ';', ':'

**Date Format element suffices**: TH

• TH->Suffixes the ordinal number with 'st' or 'nd' or 'rd' or 'th'.

Example:

Sql>Select Sysdate,TO\_CHAR(Sysdate,'DDth,MONTH,YYYY') From Dual;

Sql>Select Ename,Hiredate, TO\_CHAR(Hiredate,'DDth,MONTH,YYYY') From Emp;

**SP->Spells original numbers.**

Sql>Select Sysdate,TO\_CHAR(Sysdate,'DDsp,MONTH,YYYY') From Dual;

Sql>Select Ename,Hiredate,TO\_CHAR(Hiredate,'DDsp,MONTH,YYYY')

From Emp;

Sql>Select Ename,Hiredate, TO\_CHAR(Hiredate,'DDspth,MONTH,YYYY') From Emp;

Sql>Select Sysdate, TO\_CHAR(Sysdate,'DDspth Month YYYYSP') From Dual;

**Date Format Elements restrictions**:

* The suffixes when added to date return values always in English.
* Data suffixes are valid only on output, hence cannot be used to insert a data into the database.

**Format Model Modifiers**:

**Fill Mode Indicator**: FM

* It suppresses blank padding in the return value of the TO\_CHAR function.

Sql> Select Sysdate,

TO\_CHAR(Sysdate,'DDSPTH MONTH YYYYSP'),

TO\_CHAR(Sysdate,'FMDDSPTH MONTH YYYYSP') From Dual;

**TO\_\_NUMBER Function**:

Syntax:TO\_NUMBER(Char,fmt)

• It convert a char ,value of CHAR or VARCHAR2 data type containing a

NUMBER in the format specified by the optional format model 'fmt', to

a value of NUMBER data type.

Sql> Select '$10,000.00',

TO\_NUMBER('$10,000.00','L99,999.99') From Dual;

Sql> Select '$10,000.00',

TO\_NUMBER('$10,000.00','L99,999.99') + 500 From Dual;

**TO\_DATE function**:

Syntax:TO DATE(Char,'fmt')

• Converts given char of Char or Varchar2 data type to a value of DATE data type.

• The 'fmt' is optional Date format specified the format of CHAR.

Sql> Select TO\_CHAR(TO\_DATE('12-JAN-1980'),'DDSP') From Dual;

Sql> Select Ename,Hiredate,

ADD\_MONTHS(TO\_DATE ('1980-DECEMBER-17', 'YYYY-MONTH-DD'),3) FROM Emp WHERE Hiredate = '17-DEC-1980';

Sql> Select Ename,Hiredate, ADD\_MONTHS(TO\_DATE('1980-DECEMBER-17', 'YYYY-MONTH-DD'),3) FROM Emp WHERE

TO\_CHAR(Hiredate,'FMYYYY-MONTH-DD')='1980-DECEMBER-17';

**Spelling a Number**:

Sql> Select

TO\_CHAR( TO\_DATE('&GiveNumber','J'), 'JSP' ) "Spelled Number"

From Dual;

**Selecting a Date Specific to its Century**:

Sql>Select

TO\_CHAR( TO\_DATE(Hiredate,'DD-MON-RRRR'),'DD-MON-YYYY')Hiredate From Emp;

**Aggregating or Group Functions**:

* Group functions operate on sets of rows to give one result per group.
* Group functions can appear in select lists and in ORDER BY and HAVING clauses.
* The Oracle Server applies the group functions to each group of rows and returns a single result row for each group.
* The group functions to return summary information for each group.
* Group function cannot use in where clause and group by clause.

**Syntax:**

group\_function(DISTINCT/ALL Column)

**Guidelines for Using Group Functions**:

* The data types for the arguments can be CHAR, VARCHAR2, NUMBER or DATE.
* All group functions except COUNT (\*) ignore null values. To substitute a value for null values, use the NVL function. COUNT returns either a number or zero.
* The Oracle Server implicitly sorts the result set in ascending order of the grouping columns specified, when you use a GROUP BY clause. To override this default ordering, you can use DESC in an ORDER BY clause.
* When a group function is declared in a SELECT list, on single row columns should be declared.

**Average Function**:

Syntax: AVG(DISTINCT/ALL Column)

• It returns the Average value of column.

• It ignores NULL values.

Sql>Select Avg(Sal),Avg(DISTINCT Sal) From Emp;

Sal> Select Avg(Comm),Avg(DISTINCT Comm) From Emp;

**Sum Function**:

Syntax: SUM(DISTINCT/ALL Column)

• It returns the SUM value of column.

• It ignores NULL values.

Sql>Select SUM(Sal),SUM(DISTINCT Sal) From Emp;

Sql> Select SUM (Comm), SUM (DISTINCT Comm) From Emp;

**Maximum Function**:

Syntax: MAX(DISTINCT/ALL Column)

* It returns the Maximum value of column.
* It ignores NULL values.

Sql>Select MAX(Sal),MAX(DISTINCT Sal) From Emp;

Sql> Select MAX (Comm), MAX (DISTINCT Comm) From Emp;

Sql> Select MAX(Ename) From Emp;

**Minimum Function**:

Syntax: MIN(DISTINCT/ALL Column)

• It returns the Minimum value of column.

• It ignores NULL values.

Sql>Select MIN(Sal),MIN(DISTINCT Sal) From Emp;

Sql> Select MIN (Comm), MIN (DISTINCT Comm) From Emp;

Sql> Select MIN(Ename) From Emp;

Sql> Select MIN(Hiredate),Max(Hirdate) From Emp;

**Standard Deviation Function**:

Syntax: STDDEV(DISTINCT/ALL Column)

• It returns the Standard Deviation of column.

• It ignores NULL values.

Sql> Select STDDEV(Sal), STDDEV(DISTINCT Sal) From Emp;

Sql> Select STDDEV (Comm), STDDEV (DISTINCT Comm) From Emp;

**Variance Function**:

Syntax: Variance(DISTINCT/ALL Column)

• It returns the Variance of N.

• It ignores NULL values.

Sql>Select VARIANCE(Sal), VARIANCE (DISTINCT Sal) From Emp;

Sql>Select VARIANCE (Comm), VARIANCE (DISTINCT Comm) From Emp;

**Count Function**:

Syntax: COUNT(\*/DISTINCT/ALL Column)

• It Gives No of Rows in the Query.

• If '\*' used to returns all Rows, Including duplicated and NULLs.

• It can used to specify the count of all rows or only distinct values column.

Sql>Select COUNT(Empno) From Emp;

Sql> Select COUNT(Job),COUNT(DISTINCT Job) From Emp;

Sql> Select COUNT(Sal),COUNT(DISTINCT Sal) From Emp;

Sql>Select Count(\*) From Emp;

Sql> Select COUNT(Empno),COUNT(DISTINCT MGR) From Emp;

Sql>Select Select COUNT(Job),COUNT(DISTINCT MGR) From Emp Where Deptno = 20;

**Creating Groups of Data:**

• The Group by clause is used to decide the rows in table into groups.

Syntax1:

Sql>SELECT columnname1, Columnname2,……

FROM table [WHERE condition(s)] [GROUP BY Column name(s)] [ORDER BY column(s)];

Syntax2:

Sql>SELECTcolumnname1, GRP FUN(Column)

FROM table

[WHERE condition(s)]

[GROUP BY Columnname(s)]

[ORDER BY column(s)];

**Guidelines to use Group By Clause**:

* All GROUP BY CLAUSE columns list may or may not use in SELECT clause.
* The Extra non-Group functional column should be declared in the GROUP BY Clause.
* If the GROUP Function is included in a SELECT clause, we should not use individual result columns.
* Using WHERE clause, Rows can be pre excluded before dividing then into groups.
* Column aliases cannot be used in GROUP BY CLAUSE. By default, Rows are sorted by ascending order of the columns included in the GROUP BY LIST.

Example:

Sql> Select Job From Emp GROUP BY Job;

Sql> Select Deptno From Emp GROUP BY Deptno;

Sql> Select Mgr From Emp GROUP BY Mgr;

Sql> Select TO\_CHAR(Hiredate,'YYYY') YearGroup From Emp GROUP BY TO\_CHAR(Hiredate ,'YYYY');

Sql> Select TO\_CHAR(Hiredate,'Month') MonthGroup From Emp GROUP BY TO\_CHAR(Hiredate,'Month');

Sql> Select TO\_CHAR(Hiredate,'DD') DayGroup

From Emp GROUP BY TO\_CHAR(Hiredate,'DD');

Sql> Select TO\_CHAR(Hiredate,'Month') MonthGroup

From Emp Where TO\_CHAR(Hiredate,'Mon')<>'Dec'

GROUP BY TO\_CHAR(Hiredate, 'Month');

**Creating Group wise summaries**:

Sql>Select Owner,COUNT(Object\_Name)

From dba\_objects Group By Owner;

Sql> Select Deptno,COUNT(\*) From emp Group By Deptno;

Sql> Select Deptno, AVG(Sal) From Emp Group By Deptno

Order by AVG(Sal);

Sql> Select Deptno,Min(Sal),Max(Sal) From Emp Group by

Deptno;

Sql> Select Ename, Max(Sal) From Emp Group By Deptno;

**Note**: It will give error .

Sql>Select Deptno,SUM(Sal) From Emp Group By Deptno;

Sql>Select Deptno,Job,SUM(Sal) From Emp GROUP BY Deptno,Job;

Sql> Select Job,Min(Sal),Max(Sal) From Emp Where Deptno=30 GROUP BY Job;

Sql>Select Deptno,Sum(Sal),Max(Sal) From Emp Where Job='CLERK'

GROUP BY Deptno;

**OCP Questions**:

1)Examine the description of the STUDENTS table:

STD\_ID NUMBER(4)

COURSE\_ID VARCHARD2(10)

START\_DATE DATE

END\_DATE DATE

Which two aggregate functions are valid on the START\_DATE column? (Choose two)

A. SUM(start\_date) B. AVG(start\_date)

C. COUNT(start\_date) D. AVG(start\_date, end\_date)

E. MIN(start\_date) F. MAXIMUM(start\_date)

**Having Clause**:

You may not want all the summary rows returned by a GROUP BY query. You know that you can use WHERE to eliminate detail rows returned by a regular query. With summary queries, you can use the HAVING clause to eliminate summary rows.

* It used generally with GROUP BY Clause the having is useful for specifying a condition for the group.
* The Clause is used to filter data that is associated with group function.

Syntax:

SELECT [column,] group\_function(column). . .

FROM Table

[WHERE condition]

[GROUP BY group\_by\_expression]

[HAVING having\_expression];

[ORDER BY Column/Alias];

**The HAVING Clause**:

• Groups are formed and group functions are calculated before the HAVING clause is applied to the groups.

• The HAVING clause can precede the GROUP BY clause, but it is recommended that you place the GROUP BY clause first because it is more logical.

**Steps when you use the HAVING clause:**

1. Groups rows

2. Applies the group functions to the groups and displays the groups that match the criteria in the HAVING clause.

* GROUP BY Clause can used, Without a Group function in the SELECT list.
* HAVING Clause can used, Without a GROUP BY Clause in the SELECT Statement.
* If rows are restricted based on the result of a group function, we must have a GROUP BY clause as well as the HAVING Clause.
* Existence of GROUP BY clause dose not guarantees the Existence of HAVING clause.

Sql>Select Deptno,COUNT(Deptno) From Emp

GROUP BY Deptno HAVING COUNT(Deptno)>3;

Sql> Select Deptno,COUNT(Deptno)

From Emp HAVING COUNT(Deptno)>3 GROUP BY Deptno;

Sql> Select Deptno,AVG(Sal) From Emp

GROUP BY Deptno

HAVING AVG(Sal)>2500;

Sql> Select Job,SUM(Sal) Payroll From Emp

Where Job<>'SALESMAN'

GROUP BY Job

HAVING SUM(Sal)> = 5000

ORDER BY SUM(Sal) Desc;

Sql> Select Deptno,MIN(Sal),MAX(Sal),SUM(Sal) From Emp

Where Job='CLERK'

GROUP BY Deptno

HAVING MIN(Sal)<1000;

Sql> Select Deptno,AVG(Sal),SUM(Sal),MAX(Sal),MIN(Sal) From Emp GROUP BY Deptno

HAVING COUNT(Deptno)>3;

Sql>Select Deptno,AVG(Sal),SUM(Sal) From Emp

GROUP BY Deptno

HAVING AVG(Sal)>2500;

Sql>Select Deptno,Job,AVG(Sal),SUM(Sal) From Emp GROUP BY Deptno,Job

HAVING AVG(Sal)>2500;

**Nesting of Group Function**:

• Group Functions can be Nested to Depth of two levels.

Sql>Select MAX(AVG(Sal)) From Emp

GROUP BY Deptno;

Sql> Select MAX(SUM(Sal)),MIN(SUM(Sal)) From Emp

GROUP BY Deptno;

Sal> Select MAX(SUM(Sal)),MIN(AVG(Sal)) From Emp

GROUP BY Deptno;

**NOTE:**

Nested group function not accept any column name in select.

**Miscellaneous Function**:

GREATEST Function:

Syntax: GREATEST(Expr1,Expr2,…..)

• It used to pick highest values from list of values.

• All exprs after the first are implicitly converted to the data type of the first expr,before the comparison.

• Character comparison is based on the value of the character in the data base character list.

Sql>SELECT GREATEST ('SIBEL','CODD','SCOTT') FROM DUAL;

Sql>SELECT GREATEST(1000,2000,3000,4000) FROM DUAL;

Sql> SELECT GREATEST('10-JUL-05','20-JUL-05') FROM DUAL;

**LEAST Function**:

Syntax: LEAST(Expr1, Expr2,..)

* It used to pick lowest value from list of values
* All exprs after the first are implicitly converted to the data type of the first expr,before the comparison.

Sql>SELECT LEAST('M','N','O','X','Y','Z') FROM DUAL;

Sql>SELECT LEAST (1000,2000,3000,4000) FROM DUAL;

Sql> SELECT LEAST ('10-JUL-05','20-JUL-05') FROM DUAL;

**USER Function**:

Syntax: USER

* It returns the current oracle users names within the VARCHAR2 data type.
* The function cannot be used in the condition of the CHECk constraint.

Sql>Select USER from Dual;

**UlD\_Function**:

Syntax: UID

* It gjves a number that identified the user since oracle is used in a multi-user environment.
* UID is the way identifying a user.

Sql>Select USER,UID From Dual;

**USERENV Function**:

Syntax: USERENV(Option)

• Returns information of varchar2 data type above the current session.

The values in options are

• ‘ISDBA'->Returns TRUE' if DBA is enabled.

• 'LANGUAGE'🡪Returns the language and territory used in current Session.

• 'TERMINAL'🡪Returns the OS identifier for the current sessions terminal.

• 'SESSIONID'🡪 Returns the auditing session identifier.

• 'ENTRYID'🡪Returns the available auditing entry identifier.

• 'LANG'🡪Returns the ISD abbreviation for the language name.

• 'INSTANCE'🡪Returns the instance identification number of the current instance.

Sql>Select USERENV('ISDBA') From Dual;

Sql> Select USERENV('LANGUAGE') From Dual;

Sql> Select USERENV('TERMINAL') From Dual;

Sql> Select USERENV('SESSIONID') From Dual;

Sql> Select USERENV('LANG') From Dual;

Sql> Select USERENV('INSTANCE') From Dual;

**VSIZE Function**:

Syntax: VSIZE

• It returns the NUMBER of bytes in the internal representation of Expr.

• If Expr is NULL, Function returns NULL.

Sql>Select Ename,VSIZE(Ename) From Emp;

Sql>Select Deptno,VSIZE(Deptno) From Emp;

Sql> Select Ename,Hiredate,VSIZE(Hiredate) From Emp

**SOUNDEX Function**:

Syntax: SOUNDEX(CHAR)

It returns a charter string containing the representation of CHAR.

It allows comparison of words that are spelled differently, but sound alike in English.

Sql>Select Ename From emp

Where SOUNDEX(Ename)= SOUNDEX ('SMYTHE');

Sql> Select Ename,Job From Emp

Where SOUNDEX(Job)=SOUNDEX('manger');

Sql> Select Job From emp

Where SOUNDEX(Job)=SOUNDEX('CLRK');

---------------------------------------------------------------------------------------------