**Filter of Record:**

• The number of rows returned by a query can be limited using the WHERE clause.

• The method of restriction is the basis of the WHERE clause in SQL.

• A WHERE Clause contains a condition that must be met and should directly follow the From Clause.

Sql>SELECT \*|{[DISTINCT] column|expression [alias],...}

FROM table

[WHERE condition(s)];

• The WHERE clause can compares

o Values in Columns

o Literal Values

o Arithmetic Expressions

o Function

• The components of WHERE clause are

o Column Name

o Comparison Operator

o Column Name, constant or list of values.

• The Character strings and dates should be enclosed in single quotation marks.

• Character values are case sensitive and Date values are format sensitive (DD-MON-YY)

• The Comparison operator are used in conditions that compare one expression to another.

**Relational or Comparison operators:**

**\*\* <>,^=,!=, \*\* > \*\* >= \*\* < \*\* <= \*\* = \*\***

Example:

Sql>SELECT \* from emp WHERE deptno = 10;

**Logical Operators:**

• A logical condition combines the result of two component conditions produce a single result.

• Three logical operators are available in Oracle.

**AND OR NOT**

**AND Operator:**

• The AND operator allows creating an SQL statement based on two or more conditions being met.

• It Returns FALSE if either is FALSE, else returns unknown.

**Truth Table:**

AND

TRUE TRUE TRUE

FALSE TRUE FALSE

TRUE FALSE FALSE

FALSE FALSE FALSE

Example:

Sql> Select Ename,Sal,Job from Emp Where (Sal> = 1500 AND Sal<=5000) AND Job=’MANAGER';

**OR Operator:**

* It returns TRUE if either component conditions is TRUE.
* It returns FALSE if both are FALSE, else return unknown.

**Truth Table:**

OR

TRUE TRUE TRUE

TRUE FALSE TRUE

FALSE TRUE TRUE

FALSE FALSE FALSE

Sql> Select Empno,Ename,Sal,Deptno From Emp

Where Sal>=2000 OR Deptno=20;

Sql>Select Empno,Ename,Job,Hiredate From Emp

Where Job='MANAGER' OR Deptno=30;

Sql>Select Empno,Ename,Job,Deptno From Emp

Where (Deptno = 10 OR Deptno=20) OR JOB = 'MANAGER';

Sql> Select Ename,Job From Emp

Where (Job='CLERK' or Job='SALESMAN' or Job='ANALYST');

Sql> Select Ename,Hiredate,Deptno From Emp Where Job='MANAGER' OR Deptno=30;

Sql>Select Ename,Sal,Job From Emp

Where (Sal<=2500 OR Sal>=5000) OR Job='MANAGER';

**NOT Operator:**

• It returns TRUE if the following condition is FALSE.

• It returns FALSE if the following condition is TRUE.

• If the condition is Unknown, it returns Unknown.

**Combination of AND OR Operator:**

Sql> Select Ename,Sal From Emp

Where (Job='CLERK' or Job='PRESIDENT’ or Job='ANALYST')

And Sal>3000;

Sql> Select Empno,Ename,Job,Sal From Emp

Where (Sal>1500 OR Job='MANAGER') AND Deptno=10;

Sql> Select Empno,Ename,job,Sal From Emp

Where (Deptno=20 OR Job='MANAGER') AND Sal> = 3000;

**Not Operator Example:**

Sql> Select Empno,Ename,Job,Sal From Emp Where NOT Ename='SMITH';

Sql>Select Empno,Ename,Job,Sal From Emp Where NOT Sal>=5000;

Sql> Select Empno,Ename,Job,Sal From Emp Where NOT Job='CLERK';

Sql> Select Empno,Ename,Job,Sal From Emp Where NOT Sal<=5000;

Sql> Select Empno,Ename,Job,Hiredate From Emp Where NOT Hiredate='17-DEC-80';

Sql> Select Empno,Ename,Job,Hiredate From Emp Where NOT Job = 'CLERK' AND Deptno=20;

**Some Things To Note:**

Sql> Select Ename,Sal,Job From Emp Where Job>'CLERK';

Sql> Select Ename,Sal,Job From Emp Where Job<'CLERK';

Sql> Select Ename,Deptno,Hiredate From Emp Where Hiredate>'20-DEC-81';

Sql> Select Ename,Deptno,Hiredate From Emp Where Hiredate<'20-DEC-81';

Sql> Select Ename,Job,Hiredate From Emp Where Job^='CLERK';

Sql> Select Ename,Job,Hiredate From Emp Where NOT Job^ = 'CLERK';

Sql> Select Ename,Job,Hiredate From Emp Where NOT Job='CLERK';

Sql> Select Ename,Job,Hiredate From Emp Where NOT Hiredate='17-DEC-1980';

Sql> Select Ename,Job,Hiredate From Emp Where NOT Hiredate>'18-DEC-1981';

Rules of Precedence:

• The default Precedence order is

o All comparison operators

o NOT logical condition

o AND logical condition

o OR logical condition

Note:

• Override rules of precedence by using parentheses.

Example:

Sql> SELECT Ename,Job, Sal FROM Emp

WHERE Job = 'CLERK' OR Job = 'MANAGER' AND Sal> 1500;

Sql> SELECT Ename,Job, Sal FROM Emp

WHERE (Job = 'CLERK' OR Job = 'MANAGER') AND Sal> 1500;

SQL \*Plus Operator:

• **BETWEEN.... AND….; NOT BEWEEN ....AND....**

• BETWEEN is used to display rows based on a range of values.

• The declared range is inclusive.

• The lower limit should be declared first.

Sql>Select Empno,Ename,comm From Emp Where Comm Between 500 AND 1000;

Sql>Select Ename,Sal,Job From emp Where Sal NOT Between 1000 and 1500;

Sql>Select Ename,Sal,Job From Emp Where Job Between ‘MANAGER' AND ‘SALESMAN';

Sql> Select Ename,Sal,Job From Emp Where Job NOT Between 'MANAGER' AND 'SALESMAN';

Sql>Select Ename,Sal,Job,Hiredate From Emp Where Hiredate Between '17-FEB-1981' AND '20-JUN-1983';

Sql>Select Ename,Sal,Job,Hiredate From Emp Where Hiredate NOT Between '17-FEB-1981' AND '20-JUN-1983';

**IN Operator: NOT IN Operator;**

• The Operator is used to test for values in a specified list.

• The Operator can be used upon any datatype.

Sql>Select Ename,Sal,Job From Emp Where Ename IN(‘FORD’,’SMITH’);

Sql>Select Empno,Job,Sal From Emp Where Ename NOT IN(‘FORD’,’SMITH’);

Sql>Select Ename,Sal,Deptno From Emp Where Deptno IN(20,30);

Sql>Select Ename,Sal,Deptno From Emp Where Deptno NOT IN(20,30);

Sql>Select Ename,Sal,Deptno From Emp Where

Hiredate IN ('20-FEB-1981',’09-JUN-1981');

Sql>Select Ename,Sal,Deptno From Emp Where

Hiredate NOT IN('20-FEB-1981’,’09-JUN-1981');

**LIKE Operator : NOT LIKE Operator:**

• Use the LIKE condition to perform wildcard.

• The LIKE Operator searches of valid search string values.

• Search conditions can contain either literal characters or numbers.

• The available wild cards are

% 🡪 It is use to represent any sequence of Zero or more character.

\_ 🡪Represent any single character, only at that position only.

• The Wild Card symbols can be used in any combination with literal character.

* For finding exact match for ‘%’ and ‘\_’ the ESCAPE option has to be used, which is ‘\’ symbol with ESCAPE option.

Sql> SELECT Empno,Ename From Emp Where Ename LIKE 'M%';

Sql> SELECT Empno,Ename From Emp WHERE Ename NOT LIKE 'M%';

Sql>SELECT Empno,Ename From Emp WHERE Ename LIKE '\_O%';

Sql> SELECT Empno,Ename From Emp WHERE Ename NOT LIKE '\_O%';

Sql> SELECT Empno,Ename From Emp WHERE Ename LIKE 'SM%';

Sql> SELECT Empno,Ename,Job From Emp WHERE Job LIKE '\_\_\_\_\_';

Sql> SELECT Ename,Hiredate From Emp WHERE Hiredate LIKE '%-FEB-1981';

Sql> SELECT Ename,Hiredate From Emp WHERE Hiredate LIKE '%JAN%';

Sql> Select \*From Dept Where Dname LIKE '\_\_\\_%' ESCAPE '\';

(update dept set dname='SO\_FT\_WARE' where deptno=50;)

**IS NULL Operator: IS NOT NULL Operator:**

• The Operator tests for NULL values.

• It is the only operator that can be used to test for NULL'S.

• NULL value means the value is unavailable, unassigned, unknown, or inapplicable.

Sql>SELECT Ename,Deptno,Comm From Emp WHERE Comm IS NULL;

Sql>SELECT Ename,Deptno,Mgr,Job From Emp WHERE Mgr IS NULL;

Sql> SELECT Ename,Deptno,Comm From Emp

WHERE Comm> = 0;

Sql>SELECT Ename,Deptno,Comm From Emp WHERE Comm IS NOT NULL;

Sql>SELECT Ename,Deptno,Comm From Emp WHERE Mgr IS NOT NULL;

**ORDER BY Clause:**

• The Order of rows returned in a query result is undefined.

• The ORDER BY clause can be used to sort the rows.

• The ORDER BY clause must be the last clause of the SQL statement.

• An expression, or an alias, or column position as the sort condition.

• Default ordering of Data is ascending.

• Alias name can be used in order by clause

\*\*Number 1-999

\*\*Dates Earliest-Latest

\*\*String A-Z; NULLS->Last.

**Syntax:**

SELECT expr FROM table [WHERE condition(s) ] [ORDER BY {column, expr} [ASC|DESC]];

• The default order upon column is ascending, to change the default ordering DESC should be used after the column name.

• Sorting can be implemented on column aliases, and can also implemented upon multiple columns.

Sql>Select Ename,Job,Sal,Deptno From Emp ORDER BY Sal;

Sql>Select Ename,Job,Sal,Deptno From Emp ORDER BY Sal DESC;

Sql>Select Ename,Job,Sal,Deptno From Emp

Where Job = 'CLERK' ORDER BY Sal;

Sql>Select Ename,Job,Sal,Deptno From Emp Where Sal> = 2000 ORDER BY Deptno,Ename DESC;

Sql>Select Ename,job,Sal,Sal\*12 Annsal From Emp ORDER BY AnnSal;

Sql>Select Ename,Job,Sal,Deptno From Emp ORDER BY Deptno,Job,Sal;

Sql>Select Ename,Job,Sal,Deptno From Emp ORDER BY 2 DESC; Sql>Select \* From Emp ORDER BY 5 DESC;

**The single row functions can appear in:**

\*\*SELECT List

\*\*WHERE List

\*\*Start With Clause

\*\*CONNECT BY Clause

The types of single row functions are

\*\*CHARACTER \*\*NUMBER \*\*DATE \*\*CONVERSION

Multiple Row Functions:

• This function manipulates group of rows to give one result per group of rows.

**Single Rows Function:**

• They are used to manipulate data items.

• They accept one or more arguments and return one value for each row returned by the query.

• An argument can be:

* + User-supplied constant
  + Variable value
  + Column name
  + Expression

Syntax:

Func\_Name (Column/Expr, [Arg1, Arg2,….])

**Features of single-row functions include:**

* Acting on each row returned in the query.
* Returning one result per row.
* Returning a data value of a different type than that referenced.
* Expecting one or more arguments.
* Can be used in SELECT, WHERE, and ORDER BY clauses.
* Can be nested.

**Specification Behavior of Function:**

• **Character functions:** Accept character input and can return both character and number values.

• **Number functions:** Accept numeric input and return numeric values.

• **Date functions:** Operate on values of the DATE data type (All date functions return a value of DATE data type except the MONTHS\_BETWEEN function, which returns a number.)

• **Conversion functions:** Convert a value from one data type to another.

• **General functions:**

o NVL

o NVL2

o NULLIF

o COALSECE

o CASE

o DECODE

**Character Functions:**

• They return the data type VARCHAR2, limited to a length of 4000 Bytes.

• If the return value length exceeds, then the return value is truncate without an error.

• Functions can be divided into

o Case-manipulation functions.

o Character-manipulation functions.

**Character Function Purpose:**

**LOWER:**

• It converts alpha character values to lowercase.

• The return value has the same data type as argument char type (CHAR or VARCHAR2)

Syntax: LOWER (column | expression)

Sql>Select LOWER(‘WELCOME TO IT WORLD’) From Dual;

Sql>Select Ename,Job,LOWER(‘MY DATA’) From Emp;

Sql>Select Ename,LOWER(Ename) From Emp Where Deptno=10;

Sql> Select 'The '|| Ename || '''s Designation is '|| Job From Emp Where LOWER(Job)='manager';

**Upper Function:**

• It Converts the Alpha character values to Upper Case.

• The return value has the same data type as the argument char.

Syntax:

UPPER(Column | Expression)

Sql> Select Upper(‘welcome to it world’) From Dual;

Sql>Select Ename,Job, Upper ('My Data') From Emp;

Sql>Select Ename,Job, Upper (Ename), Upper (Job) From Emp Where Deptno=20;

Sql> Select Ename,Job From Emp Where Job=Upper('Manager');

Sql> Select Upper('E.F Codd') "Capitalised" From Dual;

Sql>Select 'The '||Ename || '''s Designation is '|| Lower(Job) From Emp Where Job=Upper(‘manager') Order by Sal;

Sql> Select Upper('The '||Ename|| ' Basic Salary is '||Sal) "Emp Salaries" ,Job From Emp

Where Job In(Upper('Manager'),Upper('clerk'))

Order by Sal Desc;

**INITCAP Function:**

* It returns a string with the first letter of each word in upper case.
* Keeping all other letters in Lower case.

Sql> Select initcap(Ename) From emp;

Sql> Select Initcap('welcome to it world') From Dual;

Sql>Select 'The Job Title for'||Inticap(Ename)||' is'|| Lower(job) Details From Emp;

Sql>Select Ename,Upper(Ename),Lower(Ename),Initcap(Ename) From Emp;

Sql>Select Empno,Initcap(Ename),Deptno From Emp

Where Ename=Upper ('ford');

**CONCAT Function:**

• It concatenates the first characters value to the second character value. Only two parameters accept.

• It return the character data type.

Syntax: CONCAT (Column1/Exp1, Column2/Exp2)

Sql>Select Concat('Oracle', 'J4A') From Emp;

Sql>Select Ename,Job,Concat(Ename,Job) From Emp Where Deptno=20;

Sql> Select Concat(Concat(Ename,Job),Sal) From Emp;

Sql> Select Concat('The Employee Name ls',initcap(Ename)) As “Employee Names " From Emp Where Deptno in(10,20);

Sql>Select Concat(Concat(Initcap(Ename),’is a’)job) Job From Emp Where Deptno in(10,20);

Sql>Select Concat(‘&Fname',’&Sname') “FulI Name" From Dual;

**SUB STRING Function:**

• Returns specified characters from character value, string from a specified position ‘m' to ‘n’ characters long.

• To extract the portion of the string it is mainly used.

Points to Remember...

• If m is 0, it is treated as 1.

• If m is positive, Oracle counts from the beginning of char to find the first character.

• If n is Omitted, Oracle returns all characters to the end of char.

• If n is less than 1 or 0, A null is returned.

Floating points numbers passed as arguments to substr are automatically converted to Integers. (Float value will be truncated)

Ex:- 1.9, 3.9

Syntax: SUBSTR(Col/Exp,m,[n])

Sql> Select Substr('SIVA RAMA KRISHNA',1,4) From Dual;

Sql> Select Substr('SIVA RAMA KRISHNA',6,4) From Dual;

Sql>Select Substr('SIVA RAMA KRISHNA',11) From Dual;

Sql>Select Substr('SIVA RAMA KRISHNA',-7) From Dual;

Sql> Select Substr('SIVA RAMA KRISHNA',-12,4) From Dual;

Sql> Select Substr('SIVA RAMA KRISHNA',-12,4) From Dual;

Sql> Select Ename,Job From Emp Where Substr(Job,6) = Upper('man');

Sql> Select Concat(Initcap(Ename),

Concat(' is a ',Concat(Initcap(Substr(Job,1,3)),' Eater.'))) From Emp Where Substr(Job,4,3) = Upper('Age');

**LENGTH Function:**

• Returns the number of characters in a value.

* If the char has data type char the length includes all trailing blanks.

• If the char is NULL, if return NULL.

Syntax:

LENGTH(Column | Expression)

Sql> Select Length('E.F CODD') From Dual;

Sql> Select Length(Ename)||' Characters exists in '||Initcap(Ename)|| '"s Name.' As "Names and Lengths " From Emp;

Sql>Select Initcap(Ename),Job From Where Length(Ename)=5;

Sql> Select Initcap(Ename),Job From Emp

Where Substr(Job,4,Length(Substr(Job,4,3))) = 'AGE';

**INSTRING Function:**

• It returns the numeric position of a named character.

Syntax: INSTR (Column | Expression, ’C’, [,m], [n] )

• Searches for Column / Expression beginning with its 'm'th character for the ‘n’th occurrences of character 'C', and return the position the character.

• 'm' can be positive or negative, if negative searches backward from the end of Column / Expression.

• The value of 'n' should be positive.

• The default value of both 'm' and ‘n' are 1.

• If search is unsuccessful, the return value is zero.

Sql> Select Instr('SIVA RAMA KRISHNA','A',1,1) From Dual;

Sql> Select Instr('SIVA RAMA KRISHNA', 'A',-20,2) From Dual;

Sql> Select Instr('SIVA RAMA KRISHNA','MA',7,1) From Dual;

Sql> Select Instr('SIVA RAMA KRISHNAVA',’V’,-1,1) From Dual;

Sql> Select Instr(Job,'A',1,2) From Emp Where Job^='MANAGER';

Sql> Select Instr(Job,'A',2) From Emp Where Job='MANAGER';

Sql> Select Instr(Job,'A') From Emp Where Job='MANAGER';