ORDER BY in DBMS

ORDER BY  in DBMS

* Sometimes the user may be interested in***arranging the data in the table in some increasing or decreasing order of values***
* **Example**: If you want to display the details of all students based on descending order of their attendance or marks etc

ORDER BY clause

The order by clause is used to arrange the fetched data from the database table in ascending or descending order of data values based on one or more columns

Syntax

SELECT column1, column2, ...  
FROM table\_name  
***ORDER BY***column1, column2, ... ***ASC|DESC***;

***ASC***: Displays ***data based on increasing order of values in the column***  
***DESC***: Displays***data based on decreasing order of values in the column***

**Consider the following sample *EMP*table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **DEPTNO** |
| 7839 | KING | PRESIDENT | – | 17-NOV-81 | 5000 | 10 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | 10 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | 20 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | 20 |

Sorting according to a single column

Here we ***use only a single column for data arrangement***

Display the names, salaries of all employees based on decreasing order of their salaries

select ename,sal from emp

***ORDER BY*** sal ***DESC***;

O/P

7 ROWS SELECTED

|  |  |
| --- | --- |
| **ENAME** | **SAL** |
| KING | 5000 |
| FORD | 3000 |
| SCOTT | 3000 |
| JONES | 2975 |
| BLAKE | 2850 |
| CLARK | 2450 |
| SMITH | 800 |

The row which contains the highest salary is displayed first, the row with the second highest salary is displayed next and so on

Display the department number and employee name as per increasing order of department numbers

select deptno,ename from emp

***ORDER BY*** deptno;

**O/P**

7 rows selected

|  |  |
| --- | --- |
| **DEPTNO** | **ENAME** |
| 10 | KING |
| 10 | CLARK |
| 20 | JONES |
| 20 | SMITH |
| 20 | SCOTT |
| 20 | FORD |
| 30 | BLAKE |

As we have 3 departments in the employee table, first the rows of 10th department followed by 20 and 30 departments are displayed i.e increasing order of department numbers

Note:

***ASC is optional and the Default value for order by***, If you don’t specify ASC or DSC by default data is arranged in ascending order, but for descending arrangement of data you must specify DESC in order by explicitly

Display all employees for working in a company based on seniority level

select ename,job,sal,hiredate

from emp

***order by*** hiredate ***desc***;

7 rows selected

|  |  |  |  |
| --- | --- | --- | --- |
| **ENAME** | **JOB** | **SAL** | **HIREDATE** |
| SCOTT | ANALYST | 3000 | 19-APR-87 |
| FORD | ANALYST | 3000 | 03-DEC-81 |
| KING | PRESIDENT | 5000 | 17-NOV-81 |
| CLARK | MANAGER | 2450 | 09-JUN-81 |
| BLAKE | MANAGER | 2850 | 01-MAY-81 |
| JONES | MANAGER | 2975 | 02-APR-81 |
| SMITH | CLERK | 800 | 17-DEC-80 |

Employees who are having an older hire date is displayed first followed by the second older hire date and so on

Sorting according to more than one column

You can***specify more than one column in the order by clause this is done when you further need sorting internally based on a certain column***

Display the details of employees based on increasing order of Departments and in each department salary should be further arranged in highest to lowest order

select ename,sal,deptno from emp

***ORDER BY*** deptno,sal ***DESC***;

O/P

7 rows selected

|  |  |  |
| --- | --- | --- |
| **ENAME** | **SAL** | **DEPTNO** |
| KING | 5000 | 10 |
| CLARK | 2450 | 10 |
| FORD | 3000 | 20 |
| SCOTT | 3000 | 20 |
| JONES | 2975 | 20 |
| SMITH | 800 | 20 |
| BLAKE | 2850 | 30 |

Before displaying the rows of30th department, rows or sorted based on the salaries of30 department in decreasing order i.e that is first highest salary in the 30th department and the second highest salary in 30th department followed by 20th department first highest salary followed by 20th department second highest salary and so on

Using where clause in ORDER BY

* You can also***specify some conditions and filter the data by using where class and then sort the data***
* Here first ***where***clause  is to be followed by***order by*** class

Display the names,sal,jobs of employees who are working as manager in highest to lowest order

select ename,job,sal from emp

where job='MANAGER'

***order by*** sal ***desc*** ;

O/P

3 rows selected

|  |  |  |
| --- | --- | --- |
| **ENAME** | **JOB** | **SAL** |
| JONES | MANAGER | 2975 |
| BLAKE | MANAGER | 2850 |
| CLARK | MANAGER | 2450 |

Here only records of managers are displayed as per decreasing order of their salaries

Specifying column numbers in order by

Instead of column names, ***you can also use the position number of columns are specified in the select statement***

select ename, sal, job

from emp

***order by*** 2 ***DESC***;

O/P

7 rows selected

|  |  |  |
| --- | --- | --- |
| **ENAME** | **SAL** | **JOB** |
| KING | 5000 | PRESIDENT |
| FORD | 3000 | ANALYST |
| SCOTT | 3000 | ANALYST |
| JONES | 2975 | MANAGER |
| BLAKE | 2850 | MANAGER |
| CLARK | 2450 | MANAGER |
| SMITH | 800 | CLERK |

The second column in the select statement is ‘sal’ hence in above query sorting of rows is done as  per increasing order of salaries

**Sorting data based on expressions**

User can specify basic arithmetic expressions and  sort the rows based on this column generated by expression

Display the names, job, annual salary of all employees based on decreasing order of their annual salary

select ename ,sal\*12 annsal ,job,hiredate

from emp

***order by*** annsal ***desc***;

O/P

7 ROWS SELECTED

|  |  |  |  |
| --- | --- | --- | --- |
| **ENAME** | **ANNSAL** | **JOB** | **HIREDATE** |
| KING | 60000 | PRESIDENT | 17-NOV-81 |
| FORD | 36000 | ANALYST | 03-DEC-81 |
| SCOTT | 36000 | ANALYST | 19-APR-87 |
| JONES | 35700 | MANAGER | 02-APR-81 |
| BLAKE | 34200 | MANAGER | 01-MAY-81 |
| CLARK | 29400 | MANAGER | 09-JUN-81 |
| SMITH | 9600 | CLERK | 17-DEC-80 |

Here we have performed operations on the existing***sal column*** and the rows are  sorted based on this newly formed display purpose column

**Group by in DBMS**

Group by clause

If you want to display the total salaries of each department in a company or else to display the highest paid employees of each branch of that company. this purpose is also called a by using a  
group by clause

Group by clause in SQL used to arrange logically related data into groups with help of some functions i.e  if a particular column has the same type of data in different rows then they can be organized this into a logical groups

General Syntax for GROUP BY clause

SELECT column\_name(s),function(column\_name)  
FROM table\_name  
WHERE condition  
***GROUP BY*** column\_name(s)  
ORDER BY column\_name(s);

Simple example for GROUP BY

**Consider a sample table*’emp’***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 7839 | KING | PRESIDENT | – | 17-NOV-81 | 5000 | – | 10 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | – | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | – | 10 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | – | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | – | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | – | 20 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | – | 20 |

Display the number of employees present in each department

select deptno,count(\*)

from emp

***group by***deptno;

O/P

3 rows selected.

|  |  |
| --- | --- |
| **DEPTNO** | **COUNT(\*)** |
| 30 | 1 |
| 10 | 2 |
| 20 | 4 |

The above query display group wise count of each department

Display the highest played employees in each department

select deptno,**max**(sal)

from emp

***group by*** deptno

O/P

3 rows selected.

|  |  |
| --- | --- |
| **DEPTNO** | **MAX(SAL)** |
| 30 | 2850 |
| 10 | 5000 |
| 20 | 3000 |

Department wise  highest salary i.e logically categorizing category employees into department wise

Using  WHERE  clause in the GROUP BY

Using ***where clause  rows  can be pre excluded before dividing them into groups***,***where***clause  must be specified before the ***group by***clause  when it is used in the query

Example

select deptno,max(sal)

from emp

***where*** deptno!=30

***group by*** deptno;

O/P

2 rows selected.

|  |  |
| --- | --- |
| **DEPTNO** | **MAX(SAL)** |
| 10 | 5000 |
| 20 | 3000 |

Highest salaries of all departments expect deptno 30 are displayed

Using ORDER BY with GROUP BY

We can also***display the rows in sorted order after logical organizing into groups using order by clause*** along with group by clause

Example

**select** deptno,max(sal)

***from*** emp

***group by*** deptno

***order by*** deptno;

O/P

3 ROWS SELECTED

|  |  |
| --- | --- |
| **DEPTNO** | **MAX(SAL)** |
| 10 | 5000 |
| 20 | 3000 |
| 30 | 2850 |

The above query displays the highest salaries in each department but first deptno 30 details followed by deptno 20 and deptno 30 i.e  increasing order of department numbers

Points to note about GROUP BY  clause

* ***GROUP BY clause is used only with the SELECT statement.***
* ***Where******class is placed before group by class*** if it is used in the query.
* ***Order by class is placed after the group by class***if it is used in the query group by class
* ***All the columns that are used in the select statement must be specified by using group by clause***
* ***If a group function is included in the select clause then we cannot use individual result columns***

**HAVING  clause**

* Having Clause is ***used to place conditions and decide which group will be part of the final result***
* You cannot use aggregate functions like sum() count() etc with  where clause
* Hence we need to use having clause if you want to specify conditions using this aggregate functions

Example

SELECT ename , SUM(sal) FROM emp

***GROUP BY*** ename

***HAVING*** SUM(sal)>2000;

O/P

6 rows selected.

|  |  |
| --- | --- |
| **ENAME** | **SUM(SAL)** |
| JONES | 2975 |
| KING | 5000 |
| CLARK | 2450 |
| SCOTT | 3000 |
| BLAKE | 2850 |
| FORD | 3000 |

UPDATE Query in DBMS

UPDATE in DBMS

In the Real world scenario, facebook, Gmail will  give an option to update your name or  Profile picture, etc, how does it work, at the backend SQL UPDATE  is executed internally

* The UPDATE  statement is used to modify or change the data of the existing table in the database
* We can update a single column as well as multiple columns as per our requirement

The general syntax of UPDATE

***UPDATE*** table\_name  
***SET*** column1 = value1, column2 = value2, ...  
WHERE condition;

* The ***SET*** is Is used to set new values to the required column and the where Clause is used to filter the rows for which rows of data are needed to be updated

Updating a single column

Any row in the database table can be updated using update  statement

**Consider a sample table EMP as shown below**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 7839 | KING | PRESIDENT | – | 17-NOV-81 | 5000 | – | 10 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | – | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | – | 10 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | – | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | – | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | – | 20 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | – | 20 |

Update the commission of all employees to 200/-  whose salary is less than 3000/-

***UPDATE*** emp ***SET*** comm=200

WHERE sal<3000;

**O/P**

7 ROWS SELECTED

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 7839 | KING | PRESIDENT | – | 17-NOV-81 | 5000 | – | 10 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | 200 | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | 200 | 10 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | 200 | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | – | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | – | 20 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | 200 | 20 |

Initially, in the EMP table, values are NULLfor comm column for all the rows but using update statement we have changed  the commission to 200 from NULL  for those employees who are having salaries less than 3000

Updating multiple columns

Using a single update statement with can***parallelly update any number of column***

Update salary by 500 and change the commission to 1000 for all analysts working in the company

***update*** emp ***set*** sal=sal+500 ,comm=1000

where job='ANALYST';

O/P

2 rows selected.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3500 | 1000 | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3500 | 1000 | 20 |

We have two analysts, hence two rows will be updated i.e salary will be increased from 3000 to 3500 and commission will be changed to 1000 i.e both salary and commission columns get updated

Updating without where clause

If you don’t specify where clause, it means that no condition is required and no data filtering happens and ***all the rows present in the table will be updated for that column as specified***

Make the salary of all employees working in the company to 500

***update***emp

set sal=500;

O/P

7 rows selected

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | ***SAL*** | **COMM** | **DEPTNO** |
| 7839 | KING | PRESIDENT | – | 17-NOV-81 | 500 | – | 10 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 500 | 200 | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 500 | 200 | 10 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 500 | 200 | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 500 | – | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 500 | – | 20 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 500 | 200 | 20 |

We have seven rows present in the table, the value of a salary in every row becomes 500 because there is no condition for data change, hence all the rows will be updated

LIKE in DBMS

LIKE in DBMS

Suppose you need to search some records based on certain patterns from the database tables then we will make use of this **LIKE** operator

* If you want to ***search all employees starting with letter P or names of all products which which consists of exactly 4 letter***s etc
* LIKE  operator used to search specified pattern in the data and retrieve the record when there is a pattern match as required

General Syntax for LIKE

SELECT column1, column2, ...  
FROM table\_name  
WHERE column ***LIKE*** pattern;

Wildcard operators

**%**: ***Percent(%)***represents 0,1 or multiple characters **\_**: ***Underscore(\_)*** is used to match exactly a single character This wild card operator is used in conjunction with where clause and filter the records based on specified patterns as required

Different ways that we can use this LIKE  clause

The following are the rules for pattern matching with the LIKE Clause:

|  |  |
| --- | --- |
| **PATTERN** | **MEANING** |
| **‘r%’** | Matches strings which start with ‘r’ |
| ‘%r’ | Matches strings with end with ‘r’ |
| ‘r%t’ | Matches strings which contain the start with ‘r’ and end with ‘t’. |
| ‘%tri%’ | Matches strings which contain the substring ‘tri’ in them at any position. |
| ‘\_tri%’ | Matches strings which contain the substring ‘tri’ in them at the second position. |
| ‘\_r%’ | Matches strings which contain ‘r’ at the second position. |
| ‘r\_%\_%’ | Matches strings which start with ‘r’ and contain at least 2 more characters. |

**Sample table *EMP***

|  |  |
| --- | --- |
| **ENAME** | **HIREDATE** |
| KING | 17-NOV-81 |
| BLAKE | 01-MAY-81 |
| CLARK | 09-JUN-81 |
| JONES | 02-APR-81 |
| SCOTT | 19-APR-87 |
| FORD | 03-DEC-81 |
| SMITH | 17-DEC-80 |
| ALLEN | 20-FEB-81 |
| WARD | 22-FEB-81 |
| MARTIN | 28-SEP-81 |
| TURNER | 08-SEP-81 |
| ADAMS | 23-MAY-87 |
| JAMES | 03-DEC-81 |
| MILLER | 23-JAN-82 |

 Display the employees whose name start with ‘M’

select ename from emp

where ename ***like*** 'M%';

**O/P**

2 rows selected.

|  |
| --- |
| **ENAME** |
| MARTIN |
| MILLER |

All employee names that start with letter **M** are displayed

Display the names of all employees having M  in any position in their name

select ename from emp

where ename ***like*** '%M%';

**O/P**

5 rows selected.

|  |
| --- |
| **ENAME** |
| SMITH |
| MARTIN |
| ADAMS |
| JAMES |
| MILLER |

The above query discuss the employee name that contains one or more M anywhere in their names

Display the names of employees whose name contain the second letter as L

select ename from emp

where ename ***like*** '\_L%';

**O/P**

3 rows selected.

|  |
| --- |
| **ENAME** |
| BLAKE |
| CLARK |
| ALLEN |

All string second letter as L are displayed

Display the names of employees which contains the fourth letter as M

select ename from

emp where ename ***like*** '\_\_\_M%';

O/P

1 ROW SELECTED

|  |
| --- |
| **ENAME** |
| ADAMS |

In the above query, we have used 3 underscores(\_\_\_) followed by letter M which mean that fourth letter should be M, like this, you can search for a specific character at any position based upon the number of underscores(\_)

Display the employee names and hire dates for the employees joined in the month of December

select ename,hiredate from emp

where hiredate ***LIKE*** '%DEC%';

O/P

3 ROWS SELECTED

|  |  |
| --- | --- |
| **ENAME** | **HIREDATE** |
| FORD | 03-DEC-81 |
| SMITH | 17-DEC-80 |
| JAMES | 03-DEC-81 |

For all the values of ***hire date*** wherever it found a string with ‘DEC’, all those  records are displayed

Display names of all employees whose name contains exactly 4 letters

select ename from emp

where ename like '\_\_\_\_';

**O/P**

3 rows selected

|  |
| --- |
| **ENAME** |
| KING |
| FORD |
| WARD |

All employee names which contains four letters are displayed. Here we have used 4 underscores, you can display required length strings based on number of underscores (\_)

Display the names of all employees whose name does not contain ‘A’ anywhere

select ename from emp

where ename ***not like*** '%A%';

**O/P**

7 rows selected

|  |
| --- |
| **ENAME** |
| KING |
| JONES |
| SCOTT |
| FORD |
| SMITH |
| TURNER |
| MILLER |

Here all the names that are free from letter ‘A’ displayed

DISTINCT in DBMS

**DISTINCT in DBMS**

* Inside table columns may contain many duplicate values and sometimes we require to list only unique values, this is done by using DISTINCT clause along with a select statement
* DISTINCT statement is used to return only unique values present in a column or combination of columns
* The DISTINCT clause is only ***for display purpose and will not affect the original database table***

**DISTINCT Syntax**

SELECT ***DISTINCT*** column1, column2,

FROM table name;

Consider a sample table EMP

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 7839 | KING | PRESIDENT | – | 17-NOV-81 | 5000 | – | 10 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | – | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | – | 10 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | – | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | – | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | – | 20 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | – | 20 |

**DISTINCT on a single column**

select ***DISTINCT*** job from emp;

Output

4 rows selected.

|  |
| --- |
| **JOB** |
| ANALYST |
| CLERK |
| MANAGER |
| PRESIDENT |

The above query Returns unique values present in the job column i.e. ***even if a value is appeared for more than once but it is displayed only for one time***

**DISTINCT on more than one column**

Whenever ***DISTINCT is applied*** on the select statement which contains ***more than one column then the combination of values of all the columns is considered as a single value*** even if ***any of the column value is different in that row it is considered as a DISTINCT value***

**Example**

select ***DISTINCT*** job,sal

from emp;

o/p

6 rows selected.

|  |  |
| --- | --- |
| **JOB** | **SAL** |
| MANAGER | 2850 |
| CLERK | 800 |
| MANAGER | 2975 |
| PRESIDENT | 5000 |
| ANALYST | 3000 |
| MANAGER | 2450 |

The above query Returns 6 values because each value is considered as a ***key combination of two values and [ANALYST, 3000] combination would appear twice*** and hence it is eliminated during display

**Another example for DISTINCT**

select ***DISTINCT*** ename, job

from emp;

Output

6 rows selected

|  |  |
| --- | --- |
| **ENAME** | **JOB** |
| FORD | ANALYST |
| CLARK | MANAGER |
| JONES | MANAGER |
| SMITH | CLERK |
| KING | PRESIDENT |
| SCOTT | ANALYST |
| BLAKE | MANAGER |

The above query results in all the seven rows in the table because ***even if job column contains duplicate values name column contain only unique values hence the key combination both ename and job is considered as one entity*** and all values become unique

AND & OR in DBMS

AND & OR DBMS in DBMS

* Sometimes user requires***more than one condition for filtering the data***, this purpose is served by AND / OR clauses
* The AND  and  OR operators are used with the where clause for precise filtration of data from the database tables by combining more than one condition along with select, update and delete queries

AND clause

**Definition**: The AND results***true*** only when ***all the conjunction of conditions specified after the where clause are satisfied***

Syntax:

SELECT column1, column2, ...  
FROM table\_name  
WHERE condition1 ***AND*** condition2 ***AND*** condition3 ...;

**Consider a sample table *emp***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **DEPTNO** |
| 7839 | KING | PRESIDENT | – | 17-NOV-81 | 5000 | 10 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | 10 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | 20 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | 20 |

Display the records of those employees who are working as a manager and getting salary greater than 2500 /-

select \* from emp

where job='MANAGER' ***AND***sal>2500;

**O/P**

2 rows selected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **DEPTNO** |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | 30 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | 20 |

Here two conditions first employee must be a manager at the same time he must be getting salary greater than 2500 then only condition becomes true and the record gets displayed

OR clause

**Definition**: Among multiple conditions specified in the ***where*** clause ***the transaction is performed if any of the condition becomes true***

Syntax

SELECT column1, column2, ...

FROM table\_name

WHERE condition1 ***OR*** condition2 ***OR*** condition3 ...

Display all employees records who are working as analyst and managers

select \* from emp

where JOB='ANALYST' ***OR*** JOB='MANAGER';

O/P

**5 rows selected**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **DEPTNO** |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | 10 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | 20 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | 20 |

In the example above, the data is fetched if  the employee is an analyst or a manager i.e data is fetched even if any one condition is satisfied

Combining AND and OR

In the where clause it is possible that a ***condition is specified as a conjunction of both AND  & OR***

Syntax:

SELECT \* FROM table\_name

WHERE condition1 ***AND|OR*** (condition2 ***AND|OR*** condition3....);

Example

select \* from emp

where (sal>1500 ***OR*** job='MANAGER') ***AND*** (deptno=10 **OR** deptno=30);

**O/P**

3 rows selected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **DEPTNO** |
| 7839 | KING | PRESIDENT | – | 17-NOV-81 | 5000 | 10 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | 10 |

The above query displays the details of employees  having a salary greater than 1500/- or if he is a manager which is exclusively from 10th dept and 30th departments