CLAUSES:

- Clause Is Statement Which Is Used To Add To Sql Pre-Define Query For Providing Additional Facilities Are Like Filtering Rows, Sorting Values, Grouping Similar Values, Finding

Sub Total and Grand Total Based On the Given Values Automatically.

- Oracle Supports The Following Clauses. Those Are,

> Where - Filtering Rows (Before Grouping Data)

> Order by - Sorting Values

> Group by - Grouping Similar Data

> Having - Filtering Rows (After Grouping Data)

> Rollup - Finding Sub Total & Grand Total (Single Column) >

Cube - Finding Sub Total & Grand Total (Multiple Columns) Syntax:

<Sql Per-Define Query> + <Clauses>;

Where Clause:

> Filtering Rows in One By One Manner before Grouping Data in

Table. Syntax:

Where <Filtering Condition>

Ex:

Select * From EMP Where Empno=7788;

Update EMP Set Sal=8500 Where Job='Clerk';

Delete From EMP Where Deptno=10;

Note: "Where" Clause Can Be Used In "Select " ,"Update" And "Delete" Commands Only.

Order by Clause:

> Sorting Values Based On Columns. It Can Be Used In "Select" Command Only.

> By Default Order By Clause Arrange Values In Ascending Order But If We Want To Arrange Values In Descending Order Then We Use "Desc" Keyword.

Syntax:

Select * / <List Of Column Names> From <Tn> Order By <Column Name1> <Asc / Desc>,<Column Name2> <Asc/Desc>,.....;

Ex1:

Waq to Display Employee Salaries In Ascending Order?

Sol:

Sql> Select * From EMP Order by Sal;

(Or)

Sql> Select Sal from EMP Order by Sal;

Ex2:

Waq to Arrange Employee Names In Descending Order?

Sol:

Sql> Select Ename from EMP Order by Ename Desc;

Ex3:

Waq to Display Employee Who Are Working In The

Deptno Is 20 and Arrange Those Employee Salaries In

Descending Order?

Sol:

Sql> Select * From EMP Where Deptno=20 Order By Sal Desc;

Ex4:

Waq to Arrange Employee Deptno's In Ascending Order

And Those Employee Salaries in Descending Order From

Each Deptno Wise?

Sol:

Sql> Select * From EMP Order By Deptno, Sal Desc;

Note:

Order by Clause Not Only On Column Names Even though

We Can Apply On Position Of Column In Select Query.

Ex:

Sql> Select * From EMP Order by 6;

Sql> Select Ename, Job, Sal from EMP Order by 3;

Sql> Select Ename, Sal from EMP Order by 2;

Sql> Select Sal from EMP Order by 1;

Note:

Using Order by Clause On "Null" Values Column Then Oracle Returns "Null" Values Last In Ascending Order And "Null" Values Are Displayed First In Descending By Default.

If We Want To Change This Default Order Of "Null" Then We Use Null Clauses Are "Nulls First" And " Nulls Last ".

Ex:

Sql> Select * From EMP Order By Comm Nulls First;

Sql> Select * From Emp Order By Comm Desc Nulls Last;

Group By:

> Grouping Similar Data Based On Columns.

> When We Use "Group by "We Must Use "Aggregative Functions"

Are Sum(),Avg(),Min(),Max(),Count().

> Whenever We Implement "Group By" Clause In Select Statement Then First Grouping Similar Data Based Columns And Later An Aggregative Function/(S) Will Execute On Each Group Of Data To Produce Accurate Result.

Syntax:

Select <Column Name1>,<Column Name2>,.....,<Aggregative Function Name1>,..... From <Tn> Group By <Column Name1>,<Column Name2>,.....;

Group By

Aggregative Functions. |

Sum (), Avg (), Job (No. Of In Each Job)

Min (), Max (), Count () |

Clerk | Analyst | President | Manager | Salesman Clerk Analyst (1) Manager

Salesman Clerk (2) Manager Salesman Clerk (3) Salesman (4) (4)

Ex1:

Waq to Find Out No. Of Employee Working In Each Job?

Sol:

Sql> Select Job,Count(*) Num_Of_Employee From Emp Group By

Job; Ex2:

Waq To Calculate Department Number Wise Total Salary ?

Sol:

Sql> Select Deptno,Sum(Sal) Total_Salary From Emp

Group By Deptno Order By Deptno;

Ex3:

Waq To Display No.Of Employee Working In Each Job Along With

Deptno Wise ?

Sol:

Sql> Select Job,Deptno,Count(*) Num_Of_Employee From

Emp Group By Job,Deptno;

Ex4:

Waq To Calculate Deptno Wise Totalsalary Where Deptno's Are 10,20 ?

Sol:

Sql> Select Deptno,Sum(Sal) Total_Salary From Emp

Where Deptno In(10,20) Group By Deptno;

Ex5:

Waq To Calculate Deptno Wise Avg,Min,Max Salaries ?

Sol:

Sql> Select Deptno,Avg(Sal) Avgsal,Min(Sal) Minsal,Max(Sal) Maxsal From Emp Group By Deptno Order By Deptno;

Having:

> Filtering Rows after Grouping Data in Table. It Can Be Used Along With "Group By" Clause.

Syntax:

Select <Column Name1>,<Column Name2>,.....,<Aggregative Function Name1>,..... From <Tn> Group By <Column Name1>,<Column Name2>,.....Having <Filtering Condition>;

Ex1:

Waq To Find Out No.Of Employee Of Each Job In Which Job No.Of

Employee Are More Than 3 ?

Sol:

Sql> Select Job,Count(*) From Emp Group By Job

Having Count(*)>3;

Ex2:

Waq To Display Sum Of Salary Of Deptno's From Emp Table.If Sum Of Salary

Of Deptno Is Less Than 9000 ?

Sol:

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

Having Sum (Sal)<9000;

Diff. B/W "Where" And "Having" Clause:

Where	Having
1. Where Clause Condition On Each Row Of A Table.	1. Having Clause Condition Is Executed On Group Of Rows Of Is Executed On A Table.

2. It Can Be Apply Before 2. It Can Be Apply After Group by

Clause. Group by Clause.

3. It Cannot Support 3. It Can Supports

Aggregative Functions. Aggregative Functions. 4. without Group By

4. Without Group By We Can Use Where Clause. We Cannot Use

Having Clause. Using All Clauses In A Single Select Statement:

Syntax:

Select <Col1>,<Col2>,.....,<Aggregative Function Name1>,.....

From <Table Name> [Where <Filtering Condition>

Group By <Col1>,<Col2>,.....

Having <Filtering Condition>

Order By <Col1> [Asc/Desc],<Col2> [Asc/Desc],.....];

Ex:

```
Select Deptno,Count(*) From Emp
Where Sal>1000
Group By Deptno
Having Count(*)>3
Order By Deptno;
```

```
Deptno Count(*)
------ ----------
 20 4
 30 5
```

Order of Execution:

> From

> Where

> Group By

> Having

> Select

> Order By

Rollup & Cube:

> Special Clauses.

> To Finding Sub Total & Grand Total Based On Columns.

> Working Along With "Group By" Clause.

> Rollup Will Find Sub & Grand Total Based On a Single Column. > Cube Will Find Sub & Grand Total Based On Multiple Columns.

Syntax:

Group By Rollup (<Col1>,<Col2>,<Col3>,.....,<Col N>)

Ex. On Rollup with a Single Column:

Sql> Select Deptno, Count (*) From Emp Group By Rollup(Deptno);

Deptno	Count(*)
10	3
20	5
30	6
	14

Ex. On Rollup with Multiple Columns:

Sql> Select Deptno, Job, Count (*) From Emp Group By Rollup(Deptno,Job);

Note: In The Above Ex. Rollup Is Finding Sub & Grand Total Based On A Single

Column (Deptno).If We Want To Find Sub & Grand Total Then Use "Cube" Clause.

Syntax:

Group By Cube (<Col1>,<Col2>,.....,<Col N>)

Ex. On Cube with a Single Column:

Sql> Select Deptno, Count (*) From Emp Group By Cube(Deptno) Order By Deptno;

Ex. On Cube with Multiple Columns:

Sql> Select Deptno, Job, Count (*) From Emp Group By Cube(Deptno ,

Job) Order by Deptno;

Grouping_ID ():

> It Used More Compact Way To Identify Sub And Grand Total Rows. > Id

Number 1: To Represent Sub Total Of First Grouping Column. 2: To

Represent Sub Total Of Second Grouping Column. 3: Grand Total row.

Syntax:

Grouping_ID (<Col1>,<Col2>,.....)

Ex:

Select Deptno, Job, Count (*), Grouping_ID (Deptno, Job) From

Emp Group by Cube (Deptno, Job) Order By Deptno;