K. Hari Teja

Extract(): Extract day, date, year from particular column.

WAQ to display f_name, l_name & hire date from Employees who was hired in month of July.

```
Select f_name, l_name, hire_date
from Employees
where Extract (month from hire_date) = 7;
```

WAQ to find all the details of Employee who was hired in 2012

07-July-2019

```
Select *
from Employees
where extract (year from hire_date) = 2012;
```

WAQ to find all the details of employee who was hired on the day 10

```
Select *
from Employees
where Extract (day from hire_date) = 10;
```

*** WAQ to fun display Employee_id, f_name, day of joining, month of joining & year of joining from Employees.

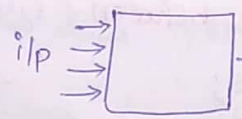
```
Select Employee_id, f_name, extract (day from hire_date) as "day",
        extract (month from hire_date) as "month" extract (year from hire_da
        as "year"
from Employees;
```

3/9/19

Multi Row Functions / Group / Aggregate :-

Multi Row functions are such functions where multi row input or single row input there is only single row output.

1. Count ()
2. Sum ()
3. Avg ()
4. Min ()
5. Max ()



Null values are not allowed.

1. Count ():- only one argument

It is used to count no. of rows in a column or table.

It can accept column or * keyword as argument.

Sum():

It is used to add the values in the column.

Avg():-

It is used to get the average of a particular column i.e

$\frac{\text{sum()}}{\text{count()}} = \text{Avg()}$

Min()

It is used to get the lowest value in the column.

Max()

It is used to get the Highest value in the Column.

WAQ to count the no. of employees in a table.

```
Select count(*) → no. of inputs giving 1 row as o/p.
from employees; // 20
```

WAQ to find the total salary

```
Select sum(salary)
from Employees; // 4,48000
```

WAQ to find avg of salary.

```
Select avg(salary)
from Employees; // 22400
```

$\frac{\text{sum()}}{\text{count()}} = \frac{4,48000}{20} = 22400$

(or)

```
Select sum(salary)/count(salary)
```

WAQ to find min and Max of a salary.

```
Select min(salary), max(salary)
from Employees; // 8500   53000
                   ↑ Min    ↑ Max
```

* Min and Max() can also gives characters → o/p will come according to alphabets.

```
Select Min(first_name), Max(first_name)
```

Brown Vinod

WAQ to count f_name from Employees table

```
Select count (f_name)
from Employees; //20
```

WAQ to count no. of employees and total salary whose working in dept_id 20.

```
Select count(*), sum(salary)
from employees
where depart_id = 20; // 2  39000
```

count Salary

WAQ to display half no. of employees working in a company.

```
Select ~~coun avg (*)~~ count(*)/2
from Employees;
```

O/P:- 20/2 = 10

WAQ to find total salary, min salary, highest salary and avg salary who is having manager_id.

```
Select sum (salary), min (salary), max (salary), avg (salary)
from Employees
where manager_id IS NOT NULL;
```

O/P: 235500
12000
53000
round(avg(salary),2) 26166.67

WAQ to display first hired employee.

```
Select min (hired_date)
from Employees; //16-Apr-07
```

WAQ to display recently hired employee

```
Select max (hired_date)
from Employees; // 07-Aug-12
```

WAQ to display number of first name & total salary who is 2 consecutive 'l' in their f_name.

```
Select count (first_name), sum(salary)
from Employees
where first_name Like '%ll%';
```

Group By Clause:-

It is used to group the identical rows in the column. group by clause is used after where clause. Group by clause can be used without where clause as well.

WAQ to display dept_id and total salary for all employees in each department.

```
Select department_id , sum (salary)
from employees
group by department_id;
```

O/p:

department_id	salary
90	60000
20	39000
110	39000
50	135000
80	95500
60	60000
10	19500

* WAQ to display lowest salary and dept_id for each department.
* display dept_id, highest salary for each dept whose dept_id is greater than 50.
* WAQ to count no. of employees working as station clerk in each dept.
* WAQ to find min and max wages given to employees in each dept.

```
* Select department_id , min (salary)
  from employees
  group by department_id;
```

```
* Select dept_id , max (salary)
  from employees
  where
  group by department_id > 50
  group by department_id;
```

O/p:

90	24000
110	19500
80	36000
60	26000

```
* Select count (*)
  from Employees
  where job_id = 'station_clerk'
  group by department_id;
```

* Select min(salary), max(salary)
  from employees
  group by dept_id;

O/P:

min	max	dept_id
17000	24000	90
19500	19500	20
:	:	:

Having Clause:-

* It is used to filter the group function.
* Having clause is used to write a condition on a group function.
* Having clause executes after group by clause.

* WAQ to find dept_id and highest salary for all the employees whose max-salary is greater than 20,000.
* WAQ to display dept_id who is working as 'ST_CLERK' in each dept make sure that no. of employees should be more than 2.
* WAQ to obtain highest salary for an employee only if the no. of employee working in the company is more than 10.

```
* Select dept_id, max(salary)
  from employees
  group by department_id
  having max(salary) > 20,000
```

difference b/w where and having

where	Having
* Executes row by row	* Executes group by group.
* It is used to check the condition before grouping on a normal column.	* It is used to check the condition after group by on a group function.
* Cannot use multirow function in where clause.	* Can use multirow function in having clause.

* Select dept_id
  from empbyees
  where Job_id = 'ST_CLERK'
  group by dept_id
  having count (*) > 2;

* Select max (salary)
  from employees
  group by dept_id
  having count (*) >10;

Order By Clause!-

It is used to arrange the result set in either ascending or descending order.

* It should always be used in the last statement of the query.
* We can order the table using multiple columns.
* By default order by clause sorts the result in ascending order.
* WAQ to display fname in ascending order
* WAQ to display f_name in ascending & l_name in descending order
* WAQ to display Salary in ascending Order.

* Select f_name
  from employees
  Order by f_name ASC;

* Select f_name, last_name
  from employees → lname is depend on f_nam
  Order by f_name ASC, last_name desc;

* Select salary
  from employees
  Order by salary Asc;

wap to display dept-id, highest salary for all employees whose dept-id is greater than 10 and having max salary greater than 20000 for each department. while displaying data display it in descending order with respect to dept-id.

Select
from
where
group by
having
order by

⑤ Select dept_id, max (salary) // columns
① from employees // Table
③ group by dept_id // group rows
② where dept_id > 10 // condition on NC
④ having max (salary) > 20000 // condition on a GF
⑥ Order by dept_id desc; // Arrange the Result.

* ③ Select f_name, salary * 12 as "Annual Salary"
① from employees
② where salary * 12 > 25000
④ Order by "Annual salary" Asc;

Sub Query / Inner / Nested :-

* Output of subquery is given as an input to the main query i.e main query is always dependent on subquery.
* Subquery is the query in another query which is embedded in where clause.
* Always subqueries should be enclosed within paranthesis ( ).
* Order by clause sh cannot be used inside a subquery.
* A subquery can have only one column in the select clause unless multiple columns are in the main query to compare with.
* If a subquery returns more than one row we can use operators such as in, not in, all, any.

= Any	Both
> Any	min
< Any	max

= all	
> all	max
< all	min

WAQ to display f-name, l-name and salary for all employees who are earning more than Tendulkar.

```
Select f-name, l-name, salary
from employees
where salary > (select salary                    24000
                from employees
                where last_name = 'Tendulkar');
```

* WAQ to display dept_id, f-name for all employees who work in the same dept in which vinod works.
* WAQ to display dept-id, f-name and job-id for all employees who work in administration dept.
* WAQ to display f-name, l-name and job-id whose job-id is similar to king and whose salary is greater than singh salary.
* WAQ to display employee_id for all employees whose dept-id in employees table is equal to dept-id in dept_table.

```
* Select department_id, f-name
  from employees
  where dept_id = (select dept_id                90
                   from employees
                   where f-name = 'Vinod');
```

O/P:

90	Sachin
90	Vinod
90	Santosh

```
* Select dept-id, f-name, job-id
  from department employees
  where dept-id = (select dept-id                10
                   from dept
                   where dept_name = 'administration');
```

O/P:

10	Mukund	AD-Assit

```
* Select f-name, l-name, job-id, salary
  from employees
  where job_id = (select job-id                  ST_CLERK
                  from employees
                  where l-name = 'king') and salary > (select salary    12000
                                                       from employees
                                                       where l-name = 'singh');
```

O/P:

Ramesh	J	ST-Clerk	18500
Marlon	Samuels	"	43000
Maureen	Jetley	"	53000

F-nam l-na Job Sala

* select employee_id
  from employees
  where dept_id IN (select dept_id
  from departments);

* WAQ to obtain all the details from employees table whose salary is greater than Adam's salary.
* WAQ to display all details of employee who were hired after Miller and salary greater than Smith's salary.
* WAQ to display f_name, L_name and job_id whose having employee_id 7521.
* WAQ to display all the details of employees who earn such amount of salary which is the smallest salary in any dept.
* WAQ to display all the details of employee whose salary is within the range of smallest salary and 50,000.