**DBMS LAB**

**Lab Assignment number 07**

**Name:** Aamir Ansari **Batch:** A **Roll no.** 01

**Aim:** Experiment to study views.

**Theory:**

View:-

In SQL view is a table that is derived from other existing base table or existing views.

Unlike ordinary tables (base tables) in a relational datatbase a view does not form part of the physical schema

A view does not exists in physical form it is considered as a virtual table.

Views are nothing but saved SQL statements, and are sometimes referred as “Virtual Tables”

Specification of view in SQL:-

1. CREATE VIEW View\_Name AS query statement

2. CREATE VIEW View\_Name(list of columns) AS query

Statemen

create view deptavg as select dno,avg(salary) as avg\_sal

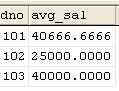
from emp group by dno

or

create view deptavg(dno,avgsal) as select dno,avg(salary)

from emp group by dno

select \*from deptavg



--Retrieve employee details of employees those are earning

Salary more than average salary of dno 101.

-- By using view

select \*From emp where salary>(select avg\_sal from deptavg

where dno=101)

40666.6666

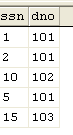
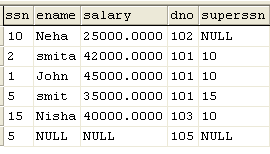
-- Without using view

select \*From emp where salary>(select avg(salary) from emp where  dno=101)

Eg. create view empinfo as select ssn,dno from emp

select \*From empinfo

Insert into empinfo(20,103)

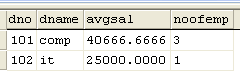
Delete from empinfo where ssn=10

--Create view deptinfo which includes columnsdno,dname,

No\_of\_emp,avgsal of every dept.

create view deptinfo as select e.dno,dname,avg(salary) as avgsal,count(\*) as noofemp from emp as e,dept as d where e.dno=d.dno group by e.dno,dname

select \*from deptinfo



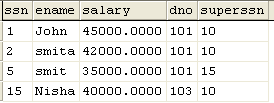
Retrieve employee details of employees those are earning

Salary more than average salary of IT department.

--By using view

select \*From emp where salary>(select avgSal from deptinfo where dname='it')

25000.0000



--without using view

select \*From emp where salary>(select avg(salary) from emp where dno=(Select dno from dept where dname='it'))

Advantages of view:-

View can provide extra security.

View can provide abstraction so database users can create abstraction by using view

Views can hide the complexity of data.

Views take very little space to store; the database contains only the definition of a view, not a copy of all the data it presents(view saves memory)

Views can act as aggregated tables, where the database engine aggregates data (sum, average, etc) and presents the calculated results as part of the data(views simplifies certain queries)

How views are stored?

When we define a view the database system stores the definition of view itself rather than the result of the evaluation of query expression.

Whenever a view definition appears in a query it is replaced by stored query expression and gets recomputed.

Types of views.

1.Updatable Views

2. Read Only Views

Updatable View:-

]If the database system can determine the reverse mapping from the view schema to the schema of the underlying base tables, then the view is updatable.

INSERT, UPDATE and DELETE operations can be performed on updatable views

Read-only view:-

Read-only views do not support such operations because the DBMS cannot map the changes to the underlying base tables.

Updation on view:-

A view with single defining table is updatable if the view attribute contains primary key of base relation as well as attributes with not null constraints that do not have default value.

A view defined on more than one table is not generally updatable.

If it is insertion delete operation then it is not possible on more than one table.

Update operation is possible on the view defined on more than one table provided that it does not violates integrity constraints defined on the table.

If view contains aggregate functions then also view is not updatable.

**// CODE**

-- (1) Write a query to create a view to display average salary of every department

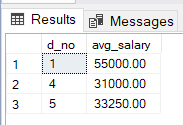
CREATE VIEW average(d\_no, avg\_salary) AS

SELECT d\_no, AVG(salary)

FROM Employee

GROUP BY d\_no;

SELECT \* FROM average;



-- (2a) Write a query to retrieve employee details of employees those are earning salary more than the average salary of department no. 5 without using view

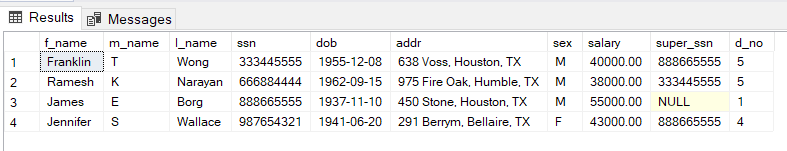
SELECT \*

FROM Employee

WHERE salary > (SELECT AVG(salary)

FROM Employee

Where d\_no = 5);

.

-- (2b) Write a query to retrieve employee details of employees those are earning salary more than the average salary of department no. 5 using view

CREATE VIEW salary\_greater\_avg\_5 AS

SELECT \*

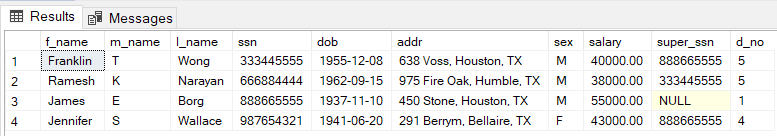
FROM Employee

WHERE salary > (SELECT AVG(salary)

FROM Employee

Where d\_no = 5);

SELECT \* FROM salary\_greater\_avg\_5;



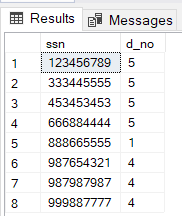
-- (3) Write a query to create a view of employee info which shows the ssn and department no. of every employee

CREATE VIEW ssn\_and\_dno AS

SELECT ssn, d\_no

FROM Employee

SELECT \* FROM ssn\_and\_dno;



-- (4) Write a query to create a view of department info which shows dept name, dept id, average salary and no. of employees

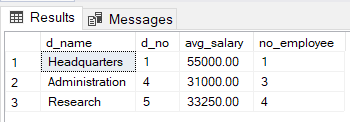
CREATE VIEW department\_info(d\_name, d\_no, avg\_salary, no\_employee) AS

SELECT d\_name, d.d\_no, AVG(salary), COUNT(\*)

FROM Employee e JOIN Department d ON e.d\_no = d.d\_no

GROUP BY d.d\_name, d.d\_no;

SELECT \* FROM department\_info;



-- (5a) Write a query to retrieve employee details of employee those are earning salary more than the average salary of Research dept without using view

SELECT \*

FROM Employee

WHERE salary > (SELECT AVG(salary)

FROM Employee

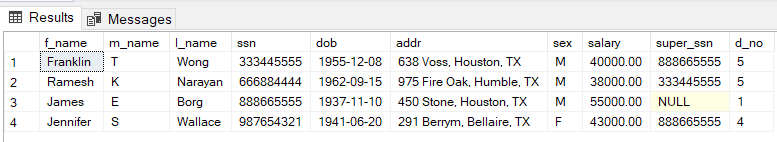
WHERE d\_no = (SELECT d\_no

FROM Department

WHERE d\_name = 'Research'

)

);



-- (5b) Write a query to retrieve employee details of employee those are earning salary more than the average salary of Research dept using view

CREATE VIEW salary\_greater\_avg\_research AS

SELECT \*

FROM Employee

WHERE salary > (SELECT AVG(salary)

FROM Employee

WHERE d\_no = (SELECT d\_no

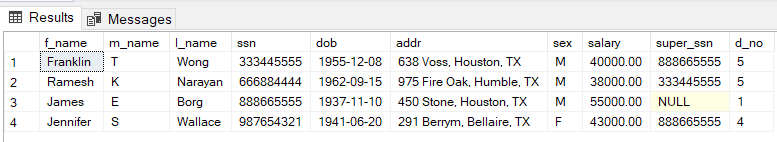
FROM Department

WHERE d\_name = 'Re search'

)

);

SELECT \* FROM salary\_greater\_avg\_research;



**Conclusion** : Hence we have successfully studied and implemented Views in DBMS.