Expt. 5 10 September 2012

**Views in SQL**

**Aim:**

To study views in SQL and solve the given queries.

**Theory:**

A view is a virtual table. In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table.

The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

Views can provide advantages over tables:

* Views can represent a subset of the data contained in a table
* Views can join and simplify multiple tables into a single virtual table
* 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 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
* Depending on the SQL engine used, views can provide extra security
* Views can limit the degree of exposure of a table or tables to the outer world

**Queries:**

1) Create a view containing the account numbers and customer names for all accounts at the 'farmagudi' branch.

> create view view\_1 as select account.accno,custname from account,depositor where account.accno=depositor.accno and branchname='farmagudi';

--------------------------

2) Create a view containing the names and addresses of all the customers who have an account with the bank at the 'ponda' branch but do not have a loan at the same branch.

> create view view\_2 as select customer.custname,custaddress,custcity from customer,depositor,account where customer.custname=depositor.custname and depositor.accno=account.accno and branchname='ponda' and customer.custname not in (select custname from borrower,loan where loan.loanno=borrower.loanno and branchname='ponda');

--------------------------

3) Create a view containing the name and average account balance of every customer of the 'ponda' branch.

> create view view\_3 as select custname,avg(balance) as avg\_bal from account join depositor on account.accno=depositor.accno where branchname='ponda' group by custname;

--------------------------

4) Find all branches where the total account deposit is less than the average total account deposit at all the branches:

a) Using a nested query in the from clause

> select branchname from (select avg(balance) as avg\_bal from account) as x, account where x.avg\_bal>balance;

b) Using a nested query in the having clause

> select branchname from account group by branchname having avg(balance)<(select avg(balance) from account)

--------------------------

5) Pay 5% interest on accounts whose balance is greater than average.

> update account set balance=balance+balance\*0.05 where balance>(select avg(balance) from account)

--------------------------

6) Pay all accounts with balances over 10,000 recieve a 6% interest whereas all others recieve a 5% interest.

> update account set balance=

case

when balance>10000 then balance+0.06\*balance

else balance+0.05\*balance

end

--------------------------

7) Read 4 users 'se\_comp', 'te\_comp', 'be\_comp', 'staff'. Grant all privileges on all the relations to the user Staff with grant options. Give select privileges on all the relations to all the users. Give update privileges on 'employee' table to ‘be\_comp’ user with grant privileges. Give update privileges on address attribute of the 'employee' table to ‘te\_comp’ and they should not be able to give this privileges to any other user. Login as different newly created users and check the privileges.

create user se\_comp identified by secomp123

create user te\_comp identified by tecomp123

create user be\_comp identified by becomp123

create user staff identified by staff123

grant select on empolyee to public;

grant select on department to public;

grant select on dept\_locations to public;

grant select on project to public;

grant select on works\_on to public;

grant select on dependent to public;

grant update on employee to be\_comp with grant option;

grant update(address) on employee to te\_comp;

**Conclusion:**

Views in SQL were studied and given queries were solved.