29)

Relational algebra expression for above queries.

i) $\Pi_{myNum}(Tab1) \cup \Pi_{MyNum}(Tab2)$

ii) $\Pi_{MyNum}(Tab1) \cap \Pi_{MyNum}(Tab2)$

iii) $\Pi_{MyNum}(Tab1) - \Pi_{MyNum}(Tab2)$

02

$\Pi_{MyNum}(Tab2) - \Pi_{MyNum}(Tab1)$

- Union, Intersect, Minus

Rules =>
- Do not affected by columnName
- Always operate on column datatype only. (sequence of column types).

* consider following schema.
Customer (custno, custnm, contactno, balance)
Supplier (suppno, suppnm, contactno, balance)

Customer

Custno	Custnm	contactno	balance
1	Amit	121212	100
2	Vishal	343434	200
3	Naresh	696969	100

duplication of data must be avoided by writing a column name rather than * in select.

players or stack holder. or custname or suppliers

30)

Supplier.

Suppno	Suppnm	Contactno	balance
2	Vishal	343434	0
3	Naresh	696969	0
4	Yogesh	109109	100

(1) Display all players / stack holders.
or Display all customers and also suppliers

SQL> select * from customer
Union
select * from supplier;

This gives duplication of data.

or

select custNm from customer
Union
select suppnm from Supplier;

This gives us duplication of data.

(2) Display name of customer only.

SQL> select custNm from Customer
Minus
Select SuppNm from Supplier;

O/p -

CustNm
Amit

(3) Get all players customers who are customers as well as supplier.

Distinct clause

classmate
Date 10/4/13
Page

31)

SQL> select custNm from customer
Intersect
Select SuppNm from Supplier;

o/p - CustNm
Vishal
Naresh.

* Consider following database schema using SQL.
Branch (bname, bcity, assets)
Customer (cname, cstreet, ccity)
Depositor (cname, accno)
Account (accno, bname, balance)
Borrower (cname, loanno)
Loan (loanno, bname, amount)

(1) Find out customer name with account details.
(2) Find the name of all customers who have a loan at Nasik branch.
(3) Find all customer who have an account at all branches located in Dhule.
(4) Find name of all customers who live in the same street and in the same city as 'Ramesh'.
(5) Find name of all customers with both loan and account at Pune branch.

(1) select cname, accno, bname, balance.
from customer, Depositor, Account.
where customer.cname = Depositor.cname
AND Depositor.accno = Account.accno;

classmate
Date 12/4/13
Page

32)

(2) select cname
from Branch, Borrower, Loan
where Branch.bname = Loan.bname
and Borrower.loanno = Loan.loanno
and bcity = 'Nasik';

(3) select cname
from Branch, Depositor, Account
where Branch.bname = Account.bname
and Account.accno = Depositor.accno
and bcity = 'Dhule';

(4) select cname
from Customer
where ccity in (select ccity from
Customer where
cname = 'Ramesh')
AND cstreet = (select cstreet from
customer where cname = 'Ramesh'

this is called parallel nested query

only one from many use ⇒ in
when all them use ⇒ = all.
or any

33)

OR

(3) nested query

```
select cname
from Depositor
where accno = ALL ( select accno
                    from Account
                    where bname in (select
                        bname
                        from branch
                        where bcity = 'Dhule'))
```

OR

(5)

```
select cname
from Depositor, Account, Borrower, Loan
where Depositor.accno = Account.accno
  AND Borrower.loanno = Loan.loanno
  AND Account.bname = Loan.bname
  AND bname = 'Pune';
```

(5)

```
select cname
from Depositor, Account, Branch
where Depositor.accno = Account.accno
      AND
      Branch.bname = Account.bname
      AND
        bcity = 'Pune'
```

Intersect :

```
select cname
from Borrower, Loan
where Borrower.loanno = Loan.loanno
      AND
      Branch.bname = Loan.bname
      AND
      bcity = 'Pune';
```