

# Database Management System (DBMS)

## Lecture-26

---

Dharmendra Kumar

October 23, 2020

## Nested Sub-queries(cont.)

---

1. Find all customers who do have a loan at the bank, but do not have an account at the bank.
2. Find the names of all branches that have assets greater than those of at least one branch located in Brooklyn.

## Solution

---

1. select distinct customer-name  
from borrower  
where customer-name not in (select customer-name from  
depositor)
2. select branch-name  
from branch  
where assets > some (select assets  
from branch  
where branch-city = 'Brooklyn')

## Nested Sub-queries(cont.)

---

1. Find the names of all branches that have an asset value greater than that of each branch in Brooklyn.
2. Finds those branches for which the average balance is greater than or equal to all average balances.

## Solution

---

1. 

```
select branch-name
from branch
where assets > all (select assets
                    from branch
                    where branch-city = 'Brooklyn')
```
2. 

```
select branch-name
from account
group by branch-name
having avg (balance) ≥ all (select avg (balance)
                           from account
                           group by branch-name)
```

## Test for Empty Relations

---

The **exists** construct returns the value **true** if the argument subquery is nonempty.

**Example:** Find all customers who have both an account and a loan at the bank.

**Solution:**

```
select customer-name
from borrower
where exists (select *
              from depositor
              where depositor.customer-name = borrower.customer-name)
```

We can test for the nonexistence of tuples in a subquery by using the not exists construct. We can use the not exists construct to simulate the set containment (that is, superset) operation: We can write “relation A contains relation B” as “not exists (B except A).”

**Example:** Find all customers who have an account at all the branches located in Brooklyn.

**Solution:**

```
select distinct S.customer-name
from depositor as S
where not exists ((select branch-name
                   from branch
                   where branch-city = 'Brooklyn')
except
(select R.branch-name
 from depositor as T, account as R
 where T.account-number = R.account-number and
       S.customer-name = T.customer-name))
```

## Test for the Absence of Duplicate Tuples

---

SQL includes a feature for testing whether a subquery has any duplicate tuples in its result. The **unique** construct returns the value **true** if the argument subquery contains no duplicate tuples.

**Example:** Find all customers who have at most one account at the Perryridge branch. **Solution:**

```
select T.customer-name
from depositor as T
where unique (select R.customer-name
              from account, depositor as R
              where T.customer-name = R.customer-name and
                    R.account-number = account.account-number and
                    account.branch-name = 'Perryridge')
```



We can test for the existence of duplicate tuples in a subquery by using the **not unique** construct.

**Example:** Find all customers who have at least two accounts at the Perryridge branch. **Solution:**

```
select distinct T.customer-name
from depositor T
where not unique (select R.customer-name
                  from account, depositor as R
                  where T.customer-name = R.customer-name
and
                  R.account-number = account.account-number
and
                  account.branch-name = 'Perryridge')
```

## Some other complex queries

---

- Find the average account balance of those branches where the average account balance is greater than \$1200.

### **Solution**

```
select branch-name, avg-balance
from (select branch-name, avg (balance)
from account
group by branch-name)
as branch-avg (branch-name, avg-balance)
where avg-balance > 120
```

- Find the maximum across all branches of the total balance at each branch.

**Solution**

```
select max(tot-balance)
from (select branch-name, sum(balance)
from account
group by branch-name) as branch-total (branch-name,
tot-balance)
```