

# Relational Model

Dr. Odelu Vanga

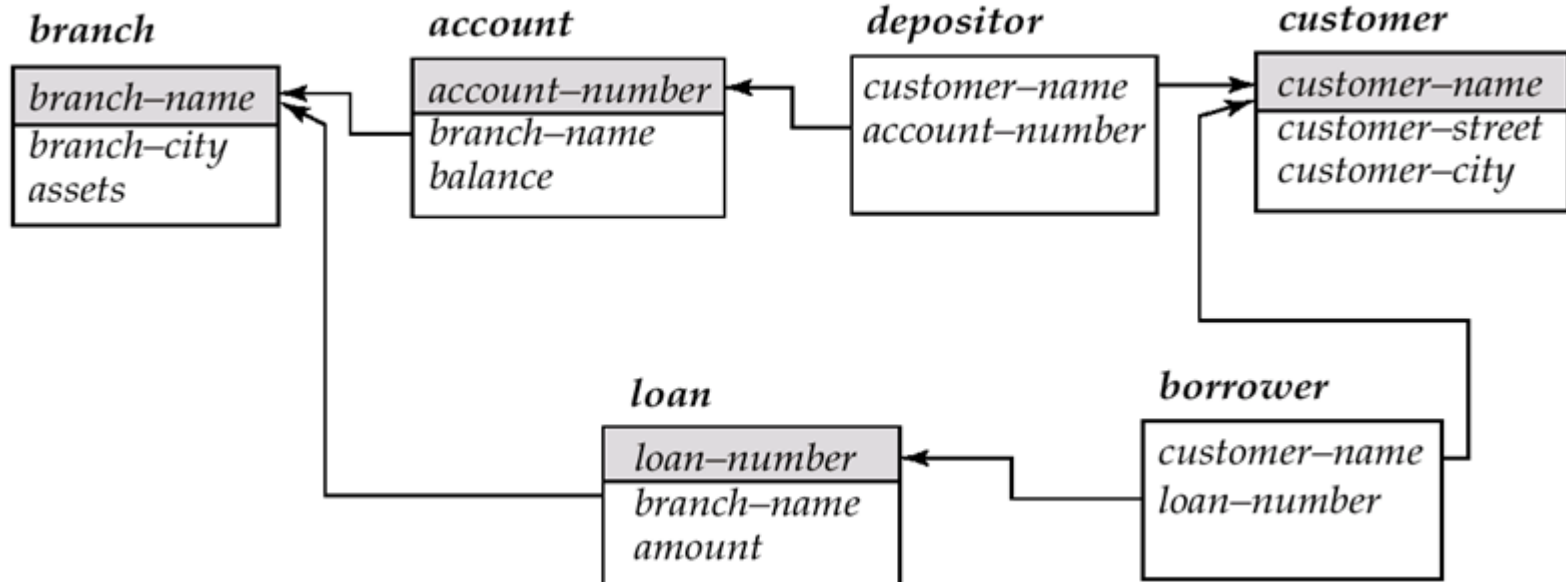
Indian Institute of Information Technology Sri City

<http://www.iiits.ac.in/people/regular-faculty/dr-odelu-vanga/>

# Nested Subqueries

- SQL provides a mechanism for the nesting of subqueries.
- A subquery is a **select-from-where** expression that is nested within another query.
- A common use of subqueries is to perform tests for
  1. set membership
  2. set comparisons
  3. set cardinality

# set membership



- Find all customers who have both an account and a loan at the bank.

```
select distinct customer-name
from borrower
where customer-name in (select customer-name
                        from depositor)
```

- Find all customers who have a loan at the bank but do not have an account at the bank

```
select distinct customer-name
from borrower
where customer-name not in (select customer-name
                               from depositor)
```

# Set Comparison

- Find all branches that have greater assets than some branch located in Brooklyn.

```
select distinct T.branch-name  
from branch as T, branch as S  
where T.assets > S.assets and  
         S.branch-city = 'Brooklyn'
```

- Same query using > **some** clause

```
select branch-name  
from branch  
where assets > some  
              (select assets  
                from branch  
                where branch-city = 'Brooklyn')
```

# Definition of Some Clause

- $F <\text{comp}> \text{some } r \Leftrightarrow \exists t \in r \text{ s.t. } (F <\text{comp}> t)$

Where  $<\text{comp}>$  can be:  $<, \leq, >, =, \neq$

$$(5 < \text{some } \begin{array}{|c|} \hline 0 \\ \hline 5 \\ \hline 6 \\ \hline \end{array}) = \text{true} \quad (\text{read: } 5 < \text{some tuple in the relation})$$

$$(5 < \text{some } \begin{array}{|c|} \hline 0 \\ \hline 5 \\ \hline \end{array}) = \text{false}$$

$$(5 = \text{some } \begin{array}{|c|} \hline 0 \\ \hline 5 \\ \hline \end{array}) = \text{true}$$

$$(5 \neq \text{some } \begin{array}{|c|} \hline 0 \\ \hline 5 \\ \hline \end{array}) = \text{true} \quad (\text{since } 0 \neq 5)$$

(= some)  $\equiv$  in  
However, ( $\neq$  some)  $\not\equiv$  not in

# Definition of all Clause

- $F <\text{comp}> \mathbf{all} \ r \Leftrightarrow \forall t \in r \ (F <\text{comp}> t)$

$$(5 < \mathbf{all} \begin{array}{|c|} \hline 0 \\ \hline 5 \\ \hline 6 \\ \hline \end{array}) = \text{false}$$

$$(5 < \mathbf{all} \begin{array}{|c|} \hline 6 \\ \hline 10 \\ \hline \end{array}) = \text{true}$$

$$(5 = \mathbf{all} \begin{array}{|c|} \hline 4 \\ \hline 5 \\ \hline \end{array}) = \text{false}$$

$$(5 \neq \mathbf{all} \begin{array}{|c|} \hline 4 \\ \hline 6 \\ \hline \end{array}) = \text{true (since } 5 \neq 4 \text{ and } 5 \neq 6)$$

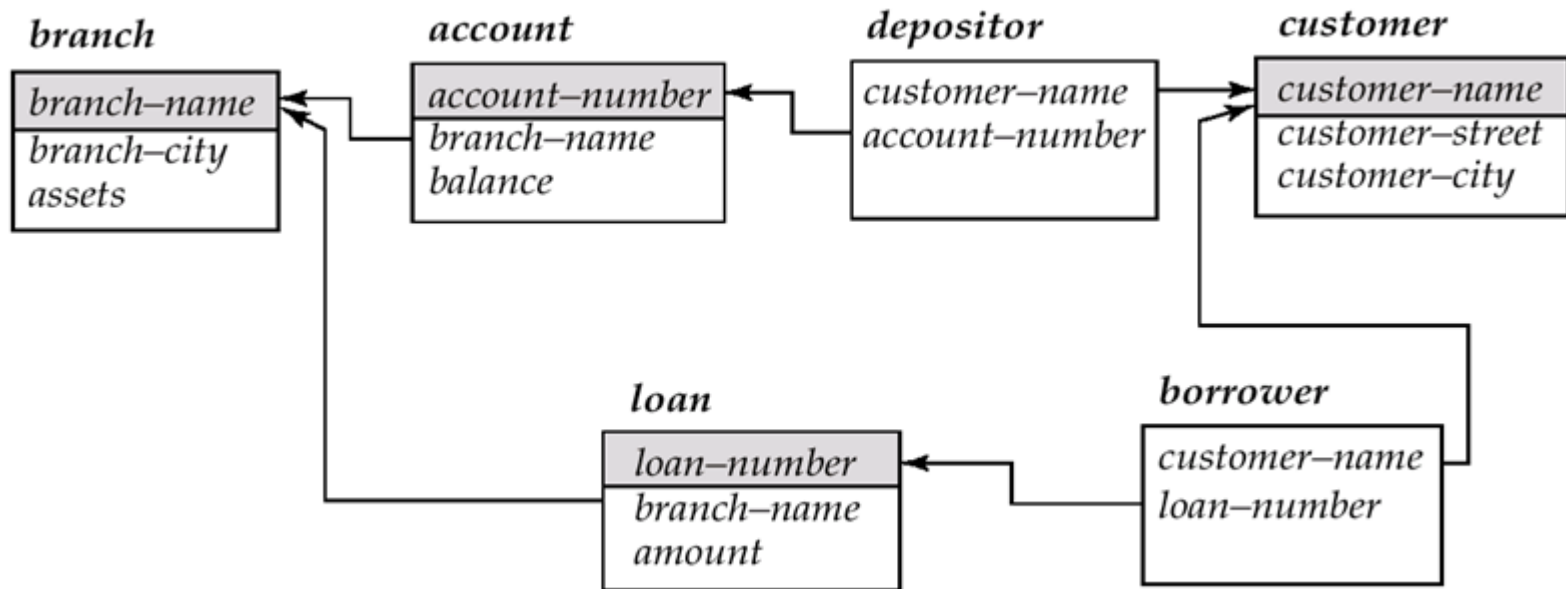
$(\neq \mathbf{all}) \equiv \text{not in}$   
However,  $(= \mathbf{all}) \not\equiv \text{in}$

# Example Query

- Find the names of all branches that have greater assets than all branches located in Brooklyn.

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

# Derived Relations



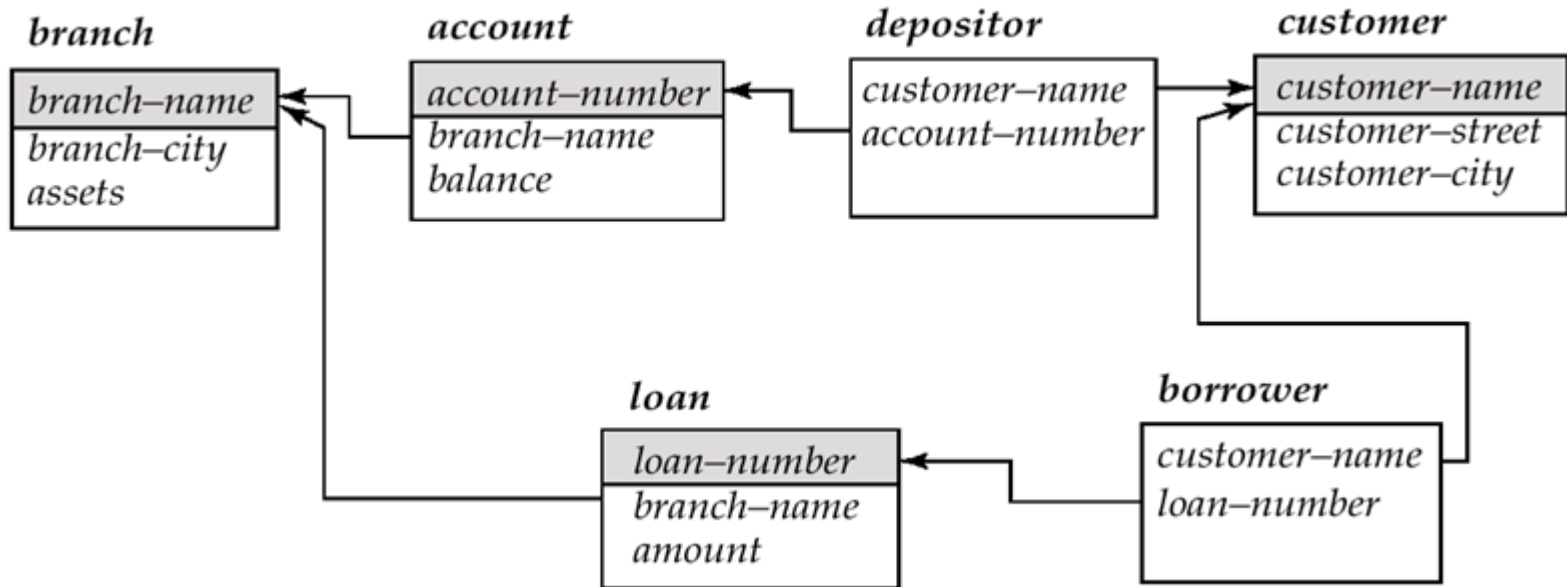
Find the names of all customers who have a loan at the Perryridge branch.

– Query 1

$$\Pi_{\text{customer-name}}(\sigma_{\text{branch-name} = \text{"Perryridge"}}(\sigma_{\text{borrower.loan-number} = \text{loan.loan-number}}(\text{borrower} \times \text{loan})))$$



# Derived Relations



Find the names of all customers who have a loan at the Perryridge branch.

– Query 2

$$\Pi_{\text{customer-name}}(\sigma_{\text{loan.loan-number} = \text{borrower.loan-number}}((\sigma_{\text{branch-name} = \text{"Perryridge"}}(\text{loan})) \times \text{borrower}))$$

# Example Queries

- Find the largest account balance and rename *account* relation as *d*

Expression:

$$\Pi_{balance}(account) - \Pi_{account.balance} (\sigma_{account.balance < d.balance} (account \times \rho_d(account)))$$

SQL query:

```
select balance
from account
where balance not in
    ( select account.balance
      from account, account as d
      where account.balance < d.balance);
```

# Example Queries

- **(Try Own)** Find the names of all customers who have a loan at the Perryridge branch but do not have an account at any branch of the bank.

$$\begin{aligned} & \Pi_{customer-name} (\sigma_{branch-name = \text{"Perryridge"}} \\ & \quad (\sigma_{borrower.loan-number = loan.loan-number} (borrower \times loan))) \\ & \quad - \Pi_{customer-name} (depositor) \end{aligned}$$

**THANK YOU**