

# **Nested Subqueries in SQL**



# Terms

For a query,

- **in-** tests for set membership
- **select-** chooses values
- **not in** - absence of set membership

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

Start with the bank account

- (**select** *customer\_name*  
**from** *depositor*)

Find the list of customers who also have a loan account. combine using the outer select.

- **select distinct** *customer\_name*  
**from** *borrower*  
**where** *customer\_name* **in** (**select**  
*customer\_name*  
**from** *depositor*)

**Find all customers who have a loan from the bank but not an account.**

- **select distinct** *customer\_name*  
**from** *borrower*  
**where** *customer\_name* **not in** (**select**  
    *customer\_name*  
**from** *depositor*)

## Find the names of all customers who are not Smith nor Jones

**not in** can also be used for enumerated sets.

- **select distinct** *customer\_name*  
**from** *borrower*  
**where** *customer\_name* **not in** ('Smith', 'Jones')

## Find the names of all branches that have assets greater than those of at least one in Brooklyn

- Set comparison using comparing operators
- Use the rename operator **as** to compare elements in the same table

- **select distinct** *T.branch\_name*

**from** *branch* **as** *T*, *branch* **as** *S*

**where** *T.assets* > *S.assets* **and** *S.branch\_city* =  
'Brooklyn'

## Another way using the “some” operator

**some-** at least one

- **select** *branch\_name*

**from** *branch*

**where** *assets* > **some** (**select** *assets*

**from** *branch*

**where** *branch\_city* = 'Brooklyn')



**Find all branches that have an asset value greater than that of each branch in Brooklyn.**

= **some** - the same as **in**

<> **all** - the same as **not in**

- **select** *branch\_name*

**from** *branch*

**where** *assets* > **all** (**select** *assets*

**from** *branch*

**where** *branch\_city* = 'Brooklyn')

# Find the branch with the highest average balance.

- SQL cannot compose of aggregate functions (function of a function), so it must be written in a certain way.

- **select** *branch\_name*  
**from** *account*  
**group by** *branch\_name*  
**having** **avg** (*balance*) **>= all** (**select** **avg** (*balance*)  
**from** *account*  
**group by** *branch\_name*)

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

**exists** - returns true if the argument in it is not empty

- **select** *customer\_name*

**from** *borrower*

**where exists (select \***

**from** *depositor*

**where** *depositor.customer\_name =  
borrower.customer\_name)*

**not exists**- superset operation.

- To show that relation A contains relation B, use **not exists (B except A)**