# CSCI4333 Database Design & Implement

## Lecture 14 – Relational Algebra 4

Instructor: Dr. Yifeng Gao

# Schedule

- Today:  Relational Algebra
- Tuesday:  Review for HW4 & 5 and midterm exam
- Next Thursday: **Midterm**

# Midterm

- Close book and close note
- All the content until today:
  - Week 1: Intro to Database to Week 7: Today Class
- Total Question:  5

# Relational Algebra

- selection $\sigma$
- Projection $\pi$
- Union $\cup$
- Set difference $-$
- Cross product $\times$
- Rename $\rho$

# Example Queries 1

- Find all loans over $1200

"select from the relation *loan*, only the rows which have a *amount* greater than 1200"

*loan*

| ***loan-number*** | ***branch-id*** | ***amount*** |
|---|---|---|
| 1234 | 001 | 1,923.03 |
| 3421 | 002 | 123.00 |
| 2342 | 004 | 56.25 |
| 4531 | 005 | 120.03 |

$\sigma_{amount > 1200}(loan)$

| | | |
|---|---|---|
| 1234 | 001 | 1,923.03 |

# Example Queries 2

- Find the loan number for each loan of an amount greater than $1200

"select from the relation *loan*, only the rows which have a *amount* greater than 1200, then project out just the *loan_number*"

*loan*

| *loan-number* | *branch-id* | *amount* |
|---|---|---|
| 1234 | 001 | 1,923.03 |
| 3421 | 002 | 123.00 |
| 2342 | 004 | 56.25 |
| 4531 | 005 | 120.03 |

$\sigma_{amount > 1200}\ (loan)$

| | | |
|---|---|---|
| 1234 | 001 | 1,923.03 |

$\pi_{loan\text{-}number}\ (\sigma_{amount > 1200}\ (loan))$

| |
|---|
| 1234 |

# Example Queries 3

- Find all loans greater than $1200 or less than $75

"select from the relation *loan*, only the rows which have a *amount* greater than 1200 or an *amount* less than 75

*loan*

| *loan-number* | *branch-id* | *amount* |
|---|---|---|
| 1234 | 001 | 1,923.03 |
| 3421 | 002 | 123.00 |
| 2342 | 004 | 56.25 |
| 4531 | 005 | 120.03 |

$\sigma_{amount > 1200 \vee amount < 75}(loan)$

| | | |
|---|---|---|
| 1234 | 001 | 1,923.03 |
| 2342 | 004 | 56.25 |

# Example Queries 4

- Find the IDs of all customers who have a loan, an account, or both, from the bank

*borrower*

| ***customer-id*** | ***loan-number*** |
|---|---|
| 201 | 1234 |
| 304 | 3421 |
| 425 | 2342 |
| 109 | 4531 |

*depositor*

| ***customer-id*** | ***account-number*** |
|---|---|
| 333 | 3467 |
| 304 | 2312 |
| 201 | 9999 |
| 492 | 3423 |

$\pi_{customer\text{-}id}\,(borrower)$

| |
|---|
| 201 |
| 304 |
| 425 |
| 109 |

| |
|---|
| 201 |
| 304 |
| 425 |
| 109 |
| 333 |
| 492 |

$\pi_{customer\text{-}id}\,(depositor)$

| |
|---|
| 333 |
| 304 |
| 201 |
| 492 |

$$\pi_{customer\text{-}id}\,(borrower) \cup \pi_{customer\text{-}id}\,(depositor)$$

# Example Queries 5

Note this example is split over two slides!

Find the IDs of all customers who have a loan at branch 001.

We retrieve *borrower* and *loan*…

*borrower*

| ***customer-id*** | ***loan-number*** |
|---|---|
| 201 | 1234 |
| 304 | 3421 |

*loan*

| ***loan-number*** | ***branch-id*** | ***amount*** |
|---|---|---|
| 1234 | 001 | 1,923.03 |
| 3421 | 002 | 123.00 |

…we calculate their cross product…

| ***customer-id*** | ***borrower.loan-number*** | ***loan.loan-number*** | ***branch-id*** | ***amount*** |
|---|---|---|---|---|
| 201 | 1234 | 1234 | 001 | 1,923.03 |
| 201 | 1234 | 3421 | 002 | 123.00 |
| 304 | 3421 | 1234 | 001 | 1,923.03 |
| 304 | 3421 | 3421 | 002 | 123.00 |

…we calculate their cross product…

| *customer-id* | *borrower.loan-number* | *loan.loan-number* | *branch-id* | *amount* |
|---|---|---|---|---|
| 201 | 1234 | 1234 | 001 | 1,923.03 |
| 201 | 1234 | 3421 | 002 | 123.00 |
| 304 | 3421 | 1234 | 001 | 1,923.03 |
| 304 | 3421 | 3421 | 002 | 123.00 |

…we select the rows where *borrower.loan-number* is equal to *loan.loan-number*…

| *customer-id* | *borrower.loan-number* | *loan.loan-number* | *branch-id* | *amount* |
|---|---|---|---|---|
| 201 | 1234 | 1234 | 001 | 1,923.03 |
| 304 | 3421 | 3421 | 002 | 123.00 |

…we select the rows where *branch-id* is equal to *"001"*

| *customer-id* | *borrower.loan-number* | *loan.loan-number* | *branch-id* | *amount* |
|---|---|---|---|---|
| 201 | 1234 | 1234 | 001 | 1,923.03 |

…we project out the *customer-id.*

| 201 |
|---|

$$\pi_{customer\text{-}id}\,(\sigma_{branch\text{-}id='001'}\,(\sigma_{borrower.loan\text{-}number\,=\,loan.loan\text{-}number}(borrower \times loan)))$$

# Now Using Natural Join

Find the IDs of all customers who have a loan at branch 001.

We retrieve *borrower* and *loan*…

*borrower*

| *customer-id* | *loan-number* |
|---|---|
| 201 | 1234 |
| 304 | 3421 |

*loan*

| *loan-number* | *branch-id* | *amount* |
|---|---|---|
| 1234 | 001 | 1,923.03 |
| 3421 | 002 | 123.00 |

1234 in *borrower* is matched with 1234 in *loan*…

3421 in *borrower* is matched with 3421 in *loan*…

| *customer-id* | *loan-number* | *branch-id* | *amount* |
|---|---|---|---|
| 201 | 1234 | 001 | 1,923.03 |
| 304 | 3421 | 002 | 123.00 |

The rest is the same.

| *customer-id* | *loan-number* | *branch-id* | *amount* |
|---|---|---|---|
| 201 | 1234 | 001 | 1,923.03 |

$\pi_{customer\text{-}id}\,(\sigma_{branch\text{-}id='001'}\,(\sigma_{borrower.loan\text{-}number\,=\,loan.loan\text{-}number}(borrower \times loan)))$

$= \quad \pi_{customer\text{-}id}\,(\sigma_{branch\text{-}id='001'}\,(borrower \bowtie loan))$

# Example Queries 6

Note this example is split over two slides!

- Find the *names* of all customers who have a loan, an account, or both, from the bank

*borrower*

| ***customer-id*** | ***loan-number*** |
|---|---|
| 201 | 1234 |
| 304 | 3421 |
| 425 | 2342 |
| 109 | 4531 |

*depositor*

| ***customer-id*** | ***account-number*** |
|---|---|
| 333 | 3467 |
| 304 | 2312 |
| 201 | 9999 |
| 492 | 3423 |

customer

| ***customer-id*** | ***customer-name*** | ***customer-city*** |
|---|---|---|
| 101 | Carol | Edinburg |
| 109 | David | Edinburg |
| 201 | John | Pharr |
| 304 | Mary | McAllen |
| 333 | Ben | Mission |
| 425 | David | Mission |
| 492 | Jason | Edinburg |
| 501 | Adam | Harlingen |

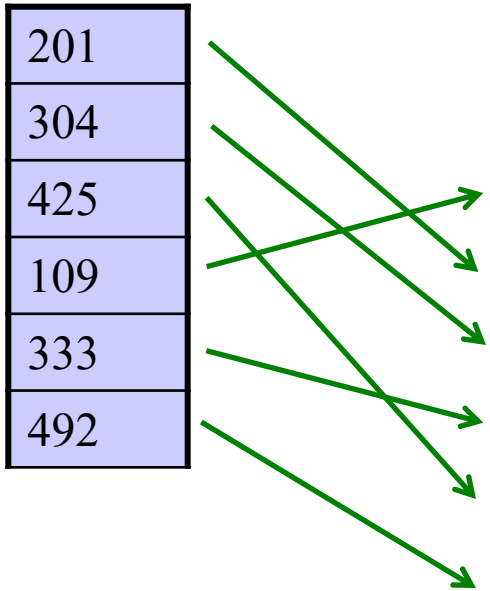
# Example Queries 6

Note this example is split over two slides!

- Find the *names* of all customers who have a loan, an account, or both, from the bank

$\pi_{customer\text{-}id}(borrower) \cup \pi_{customer\text{-}id}(depositor)$

customer




| *customer-id* | *customer-name* | *customer-city* |
|---|---|---|
| 101 | Carol | Edinburg |
| 109 | David | Edinburg |
| 201 | John | Pharr |
| 304 | Mary | McAllen |
| 333 | Ben | Mission |
| 425 | David | Mission |
| 492 | Jason | Edinburg |
| 501 | Adam | Harlingen |

# Example Queries 7

- Find the *names* of all customers who have a loan, an account, or both, from the bank

| *customer-id* | *customer-name* | *customer-city* |
|---|---|---|
| 109 | David | Edinburg |
| 201 | John | Pharr |
| 304 | Mary | McAllen |
| 333 | Ben | Mission |
| 425 | David | Mission |
| 492 | Jason | Edinburg |

| *customer-name* |
|---|
| David |
| John |
| Mary |
| Ben |
| David |
| Jason |

| *customer-name* |
|---|
| David |
| John |
| Mary |
| Ben |
| Jason |

$\Pi_{customer\text{-}name}( ( \Pi_{customer\text{-}id} (borrower) \cup \Pi_{customer\text{-}id} (depositor) ) \bowtie customer)$

# Example Queries 8

- Find the name of the sailor having the highest rating.

| sid | sname | rating | age |
|---|---|---|---|
| 28 | yuppy | 9 | 35.0 |
| 31 | lubber | 8 | 55.5 |
| 44 | guppy | 5 | 35.0 |
| 58 | rusty | 10 | 35.0 |

S2

With only six operators:
**selection, projection, union, difference, cross-product, rename**
can we solve it?

# Example Queries 8

- Find the name of the sailor having the highest rating.

$$\text{AllR} = \pi_{ratingA}\, \rho(rating\text{->}ratingA, S2)$$

$$\text{Result} = \pi_{Sname}(\sigma_{rating<ratingA}(S2 \times \text{AllR}))$$

What's in "Result"?

Does it answer our query?

S2

| sid | sname | rating | age |
|---|---|---|---|
| 28 | yuppy | 9 | 35.0 |
| 31 | lubber | 8 | 55.5 |
| 44 | guppy | 5 | 35.0 |
| 58 | rusty | 10 | 35.0 |

S2

| sid | sname | rating | age |
|---|---|---|---|
| 28 | yuppy | 9 | 35.0 |
| 31 | lubber | 8 | 55.5 |
| 44 | guppy | 5 | 35.0 |
| 58 | rusty | 10 | 35.0 |

×

AllR

| ratingA |
|---|
| 9 |
| 8 |
| 5 |
| 10 |

=

| sid | sname | rating | age | ratingA |
|---|---|---|---|---|
| 28 | yuppy | 9 | 35.0 | 9 |
| 28 | yuppy | 9 | 35.0 | 8 |
| 28 | yuppy | 9 | 35.0 | 5 |
| **28** | **yuppy** | **9** | **35.0** | **10** |
| **31** | **lubber** | **8** | **55.5** | **9** |
| 31 | lubber | 8 | 55.5 | 8 |
| 31 | lubber | 8 | 55.5 | 5 |
| **31** | **lubber** | **8** | **55.5** | **10** |
| **44** | **guppy** | **5** | **35.0** | **9** |
| **44** | **guppy** | **5** | **35.0** | **8** |
| 44 | guppy | 5 | 35.0 | 5 |
| **44** | **guppy** | **5** | **35.0** | **10** |
| 58 | rusty | 10 | 35.0 | 9 |
| 58 | rusty | 10 | 35.0 | 8 |
| 58 | rusty | 10 | 35.0 | 5 |
| 58 | rusty | 10 | 35.0 | 10 |

$$\text{AllR} = \pi_{ratingA}\, \rho(rating\text{->}ratingA, S2)$$

$$\text{Result} = \pi_{Sname}(\sigma_{rating<ratingA}(S2 \times \text{AllR}))$$

# Back to our query

- Find the name of the sailor having the highest rating.

$$\text{AllR} = \pi_{ratingA}\, \rho(rating{->}ratingA,\ S2)$$

$$\text{Tmp} = \pi_{Sid,Sname}(\sigma_{rating<ratingA}(S2 \times \text{AllR}))$$

$$\text{Result} = \pi_{Sname}\,(\pi_{Sid,Sname}(S2) - \text{Tmp})$$

# Example Queries 9

- Find all pairs of the customers who live in the same city (name1,name2, city)

*customers*

| *customer-id* | *customer-name* | *customer-city* |
|---|---|---|
| 101 | Carol | Edinburg |
| 109 | David | Edinburg |
| 201 | John | Pharr |
| 304 | Mary | McAllen |
| 333 | Ben | Mission |
| 425 | David | Mission |
| 492 | Jason | Edinburg |
| 501 | Adam | Harlingen |