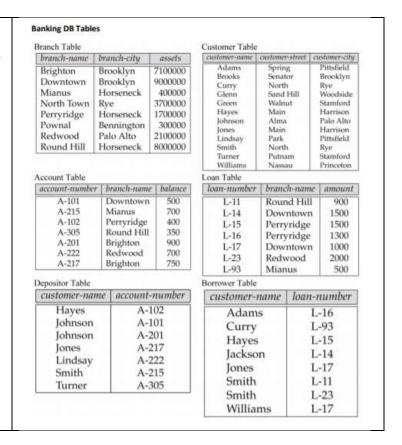
## Module Exercise 7.5: CS 5012: RA Banking

## H. Diana McSpadden (hdm5s)

## Banking DB Schema

- branch (<u>branch-name</u>, branch-city, assets)
- customer (<u>customer-name</u>, customer-street, customercity)
- account (account-number, branch-name (FK), balance)
- loan (loan-number, branch-name (FK), amount)
- depositor (<u>customer-name</u>, account-number (FK))
- borrower (customer-name, loan-number (FK))



## Questions

- 1. Find all loans over \$1200.
  - a.  $\sigma_{(amount>1200)}Loan$
  - b. Results in:

| loan-number | branch-name | amount |
|-------------|-------------|--------|
| L-14        | Downtown    | 1500   |
| L-15        | Perryridge  | 1500   |
| L-16        | Perryridge  | 1300   |
| L-23        | Redwood     | 2000   |

- 2. Find the loan number for each loan of an amount greater than \$1200.
  - a.  $\pi_{loan-number}(\sigma_{(amount>1200)}Loan)$
  - b. Results in:

| loan-number |
|-------------|
| L-14        |
| L-15        |
| L-16        |
| L-23        |

- 3. Find the names of all customers who have a loan, an account, or both from the bank.
  - a.  $\pi_{(customer-name)}((Customer \bowtie Depositor) = \bowtie = (Customer \bowtie Borrower))$
  - b. Results in:

| customer-name |
|---------------|
| Hayes         |
| Johnson       |
| Jones         |
| Lindsay       |
| Smith         |
| Turner        |
| Adams         |
| Curry         |
| Jackson       |
| Williams      |

- ⋈ enforces equality on shared attributes (customer-name)
- =⋈=pads with nulls any accounts-number or loan-number that isn't available for the join on customer-name
- $\pi$  removes duplicates
- 4. Find the names of all customers who have a loan and an account at the bank.
  - a.  $\pi_{(customer-name)}(Depositor \bowtie Borrower)$
  - b. Results in:

| customer-name |  |
|---------------|--|
| Hayes         |  |
| Jones         |  |
| Smith         |  |

- c. ⋈ enforces equality on shared attributes (customer-name)
- 5. Find the names of all customers who have a loan at the Perryridge branch.
  - a.  $\pi_{(customer-name)}(Borrower \bowtie \sigma_{(branch-name="Perryridge")}Loan)$
  - b. Results in:

| customer-name |  |
|---------------|--|
| Hayes         |  |
| Adams         |  |

- c. ⋈ enforces equality on shared attributes (loan-number)
- 6. Find the names of all customers who have a loan at the Perryridge branch, but no account at any branch of the bank.
  - a.  $\pi_{(customer-name)}(Borrower \bowtie \sigma_{(branch-name=Perryridge)}Loan) \pi_{(customer-name)}(Depositor)$
  - b. Results in:

| customer-name |  |
|---------------|--|
| Adams         |  |

c. — finds tuples in the first projection but not the second.

- 7. Find the names of all customers who have an account at the Downtown and Mianus branches.
  - a.  $\pi_{(customer-na)}((Depositor \bowtie \sigma_{(branch-name="Downtown")}Account) \bowtie (Depositor \bowtie \sigma_{(branch-name="Mianus")}Account))$
  - b. Results in an empty tuple.
- 8. Find the total amount each branch has in accounts.
  - a.  $\pi_{(branch-nam,assets)}(Branch)$
  - b. The above assumes that total amount the branch has in its accounts is the amount in the assets column of the relation.
- 9. Find the average loan amount of each customer.
  - a. customer-name  $\mathcal{G}$   $avg_{(amount)}(\pi_{(customer-name,amount)}(\textit{Borrower} \bowtie \pi_{(loa})(Loan))$
  - b. Left join the Borrower relation on the projection of loan-number and amount from the Loan relation. From that join, project customer-name and amount. Then, by customer-name, aggregate the average loan amount.
- 10. Find the names of all customers who have an account at every branch located in Brooklyn.
  - a.  $\pi_{(customer-name)}(\pi_{(customer-name,branch-nam)}(Depositor \bowtie Account)\% \left(\pi_{(branch-nam)}(\sigma_{(branch-city=Brooklyn)}Branch)\right))$
  - b. Use the division operator to only return records that match both the branch-names in branches from Brooklyn.