COSC210
Database Mangement Sytems

Lab Session 5 - Relational Algebra

By Dr Edmund Sadgrove

University of New England

Reading

- Chapter 8 from ***Fundamentals of Database Systems*** by Elmazri and Navathe

Summary

Introduction

- This workshop is an optional workshop with some examples from the banking database we have utilised in our PSQL lectures. Recall that Relational Algebra (RA) is another way to express our SQL, for this reason we will be looking at some of the examples from the lectures and recreating them in RA. For convenience we will go through some RA revision. Just like the lectures, we will have the option to do the keyboard form or standard notation.

Cheat Sheet

- Recall that Relational Algebra can express all operations from SQL, there are some notable differences however, including the ability to use `DIVIDEDBY` and `MINUS` that can both be used for universal quantifiers. Cross products and natural joins can also be expressed in PSQL. These have been provided in both forms (symbol and keyboard).

  Full cheat sheet:

Operation	Symbol	Keyboard Form
Assignment	$\leftarrow$	`:=`
Select	σ	`R WHERE C`
Project	π	`R[ ]`
Inner join	$\bowtie$	`JOIN`
Left outer join	□	`LEFT OUTER JOIN`
Right outer join	□	`RIGHT OUTER JOIN`
Full outer join	□	`FULL OUTER JOIN`
Natural join	$*$	`*`
Cross product	$\times$	`TIMES`
Union	$\cup$	`UNION`
Intersection	$\cap$	`INTERSECTION`
Minus	$-$	`MINUS`
Divided by	$\div$	`DIVIDEBY`
Aggregation	$\Im$	`R GROUP BY A`
Rename	ρ	`R( )`
Relational	$<, >, \leq, \geq, =, \neq$	`<, >, ≤, ≥, =, <>`
Logical	$\wedge, \vee, \neg$	`and, or, not`

The Select Operation

- The select operation is similar to the WHERE clause in PSQL, this refines our search to a select number of tuples based on a set criteria (or condition).
- Syntax:
  - $\sigma_{attribute=value}\ (Table\_Name)$
  - Keyboard form: `Table_Name WHERE attribute = value`
  - $\sigma_{attribute>value}\ (Tablename)$
  - Keyboard form: `Table_Name WHERE attribute > value`

The Project Operation

- The project operation is similar to the SELECT operation in PSQL, this is used to refine the number of attributes returned.
- Syntax:
  - $\pi_{attribute_1,attribute_2,attribue_3}(Table\_Name)$
  - Keyboard form:
  - `Table_Name[attribute_1, attribute_2, attribue_3]`

Assigning and Renaming

- The assignment operator allows us to store results for later use, experessed $\leftarrow$ and := in the keyboard form. Renaming utilises the `\rho` or `R` operators respectively.
- Syntax:
  1. $TEMP \leftarrow \sigma_{attribue=value}\ (Table\_Name)$
  2. $\rho_{(new\_name_1,new\_name_2,new\_name_3)} \leftarrow \pi_{attribute_1,attribute_2,attribue_3}\ (TEMP)$
- Keyboard form:
  1. `TEMP := Table_Name WHERE attribue = value`
  2. `R(new_name_1, new_name_2, new_name_3) := TEMP[attribute_1, attribute_2, attribue_3]`

Set operatons

- Relational algebra includes the set operations:
  - **Union**:
    - Denoted by $R \cup S$ or `R UNION S`.
    - Includes tuples from *R* and *S* - duplicates are removed.
  - **Intersection**:
    - Denoted by $R \cap S$ or `R INTERSECTION S`.
    - Includes tuples that are in both *R* and *S*.
  - **Minus**:
    - Denoted by $R - S$ or `R MINUS S`.
    - Includes tuples that are in *R* but not in *S*.

Joins and Cartesian Product Operations

- The **join** operation:
  - **Inner Join**:
    - $R \bowtie_{<join\ condition>} S$ or `R JOIN<condition>S`
  - **Equijoin**:
    - Cartesian product with select operaton.
    - E.g. $\sigma_{a_1=a_2}(R \times S)$ or `(R TIMES S) WHERE a_1=a_2`
  - **Natural join**:
    - Join on attributes of the same name.
    - Denoted with a $*$ symbol:
    - E.g. $R * \rho_{(a_1,a_2,\ldots,a_n)}(S)$ or `R * S(a_1, a_2, ..., a_n)`

Divided by Operation

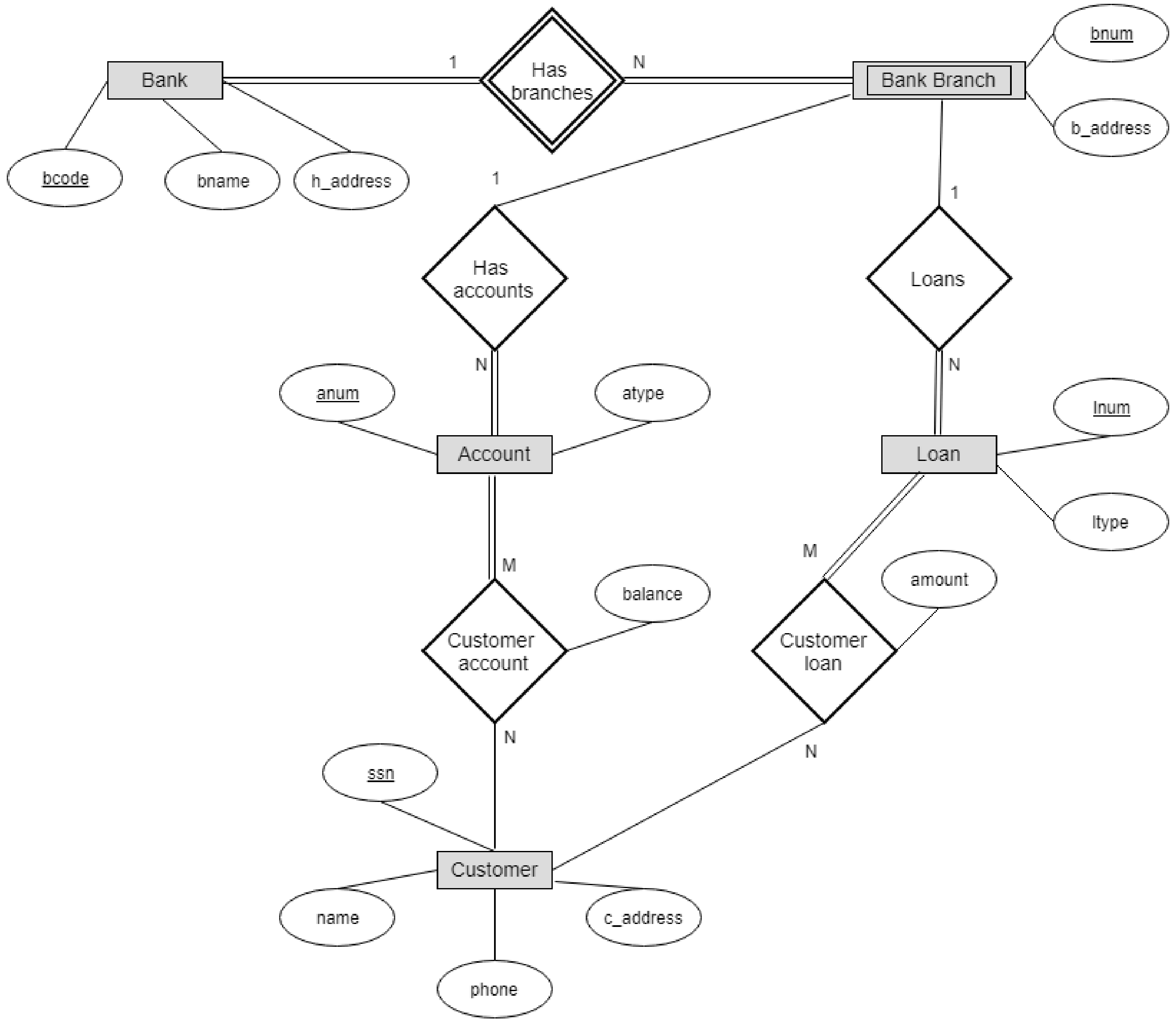
- The Divided by operation allows us to do universal quantifiers that rely on matching a particular condition for all attributes.
- The result of $T \leftarrow R \div S$ as stored in T.
- The results where all values for A in S occur for attribute B in R.
- Keyboard form: `T := R DIVIDEDBY S`

Aggregation

- Aggregation allows us to summarise data and just like PSQL, requires a group attribute.
- Includes the following funcitons:
  - **SUM**
  - **COUNT**
  - **MAXIMUM**
  - **MINIMUM**
  - **AVERAGE**
- The general form of an aggregate operation:
  - $_{<grouping\ attributes>}\Im_{<function\ list>}(R)$
  - Keyboard form:
    - `<function list> R GROUP BY <grouping attributes>`

The Banking Database

- Recall the banking database from the PSQL lectures, we will go through some of the examples questions from the slides in RA.



Exercises for You

For these questions, feel free to make assumptions about the key attributes for joins in each relation. RA is implementation independent, but with that said, the key attribute definitions can be found in the database definitions lecture of week 2 (lecture 4).

1. Display customers with a student account in Armidale branch.
2. Lets display average, total and count of balances per branch.
3. List the number of accounts and names of customers who have more than one bank account.
4. Use a UNION to display customer accounts and loans in one table. This should display customer name in the first column, balances and amounts in the next column and account number or loan number in the last column.
5. Display customers with all accounts NOT in Armidale branch.
6. Get Customers whos loan types are all general.
7. Return a sole customer with general loan types and if another customer with loan type general exists return no result.