Example Queries

Assume the following relations:

```
BOOKS(DocId, Title, Publisher, Year)
STUDENTS(StId, StName, Major, Age)
AUTHORS(AName, Address)
borrows(DocId, StId, Date)
has-written(DocId, AName)
describes(DocId, Keyword)
```

- *List the year and title of each book.*

  $\pi_{\text{Year, Title}}(\text{BOOKS})$

- *List all information about students whose major is CS.*

  $\sigma_{\text{Major} = \text{'CS'}}(\text{STUDENTS})$

- *List all students with the books they can borrow.*

  $\text{STUDENTS} \times \text{BOOKS}$

- *List all books published by McGraw-Hill before 1990.*

  $\sigma_{\text{Publisher} = \text{'McGraw-Hill'} \wedge \text{Year} < 1990}(\text{BOOKS})$

- *List the name of those authors who are living in Davis.*

  $\pi_{\text{AName}}(\sigma_{\text{Address like '\%Davis\%'}}(\text{AUTHORS}))$

- *List the name of students who are older than 30 and who are not studying CS.*

  $\pi_{\text{StName}}(\sigma_{\text{Age}>30}(\text{STUDENTS})) -$
  $\pi_{\text{StName}}(\sigma_{\text{Major}=\text{'CS'}}(\text{STUDENTS}))$

- *Rename* `AName` *in the relation* `AUTHORS` *to* `Name`.

  $\rho_{\text{AUTHORS(Name, Address)}}(\text{AUTHORS})$

Composed Queries (formal definition)

- A *basic expression* in the relational algebra consists of either of the following:
  - A relation in the database
  - A constant relation
    (fixed set of tuples, e.g., $\{(1,2),(1,3),(2,3)\}$)
- If E_1 and E_2 are expressions of the relational algebra, then the following expressions are relational algebra expressions, too:
  - $E_1 \cup E_2$
  - $E_1 - E_2$
  - $E_1 \times E_2$
  - $\sigma_P(E_1)$ where P is a predicate on attributes in E_1
  - $\pi_A(E_1)$ where A is a list of some of the attributes in E_1
  - $\rho_x(E_1)$ where x is the new name for the result relation [and its attributes] determined by E_1

Examples of Composed Queries

1. *List the names of all students who have borrowed a book and who are CS majors.*

   $\pi_{\text{StName}}(\sigma_{\text{STUDENTS.StId=borrows.StId}}$
   $(\sigma_{\text{Major='CS'}}(\text{STUDENTS}) \times \text{borrows}))$

2. *List the title of books written by the author 'Silberschatz'.*

   $\pi_{\text{Title}}(\sigma_{\text{AName='Silberschatz'}}$
   $(\sigma_{\text{has-written.DocId=BOOKS.DocID}}(\text{has-written} \times \text{BOOKS})))$

   or

   $\pi_{\text{Title}}(\sigma_{\text{has-written.DocId=BOOKS.DocID}}$
   $(\sigma_{\text{AName='Silberschatz'}}(\text{has-written}) \times \text{BOOKS}))$

3. *As 2., but not books that have the keyword 'database'.*

   . . . as for 2. . . .

   $-\ \pi_{\text{Title}}(\sigma_{\text{describes.DocId=BOOKS.DocId}}$
   $(\sigma_{\text{Keyword='database'}}(\text{describes}) \times \text{BOOKS}))$

4. *Find the name of the youngest student.*

   $\pi_{\text{StName}}(\text{STUDENTS}) -$
   $\pi_{\text{S1.StName}}(\sigma_{\text{S1.Age>S2.Age}}(\rho_{\text{S1}}(\text{STUDENTS}) \times \rho_{\text{S2}}(\text{STUDENTS})))$

5. *Find the title of the oldest book.*

   $\pi_{\text{Title}}(\text{BOOKS}) -$
   $\pi_{\text{B1.Title}}(\sigma_{\text{B1.Year>B2.Year}}(\rho_{\text{B1}}(\text{BOOKS}) \times \rho_{\text{B2}}(\text{BOOKS})))$