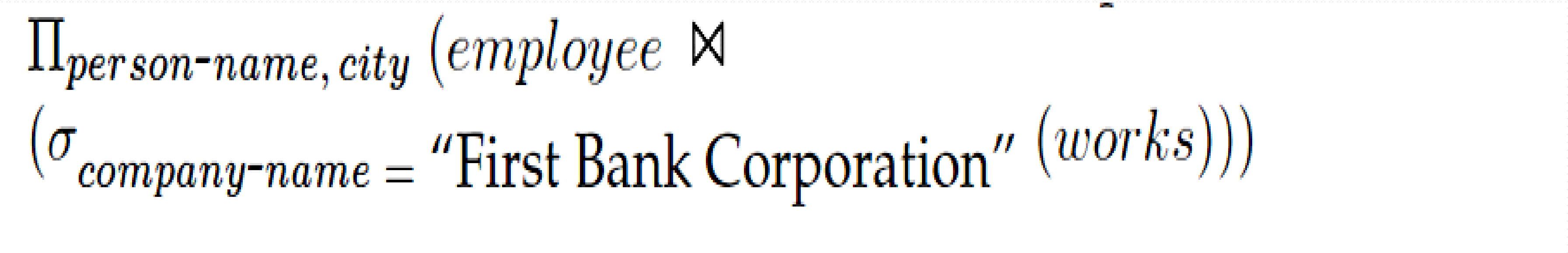
- employee (person-name, street, city)
- works (person-name, company-name, salary)
- company (company-name, city)
- manages (person-name, manager-name)

Find the names of all employees who work for First Bank Corporation.

$$\Pi_{person\text{-}name}\left(\sigma_{company\text{-}name = \text{“First Bank Corporation”}}(works)\right)$$

- employee (person-name, street, city)
- works (person-name, company-name, salary)
- company (company-name, city)
- manages (person-name, manager-name)

Find the names and cities of residence of all employees who work for First Bank Corporation.



$$\Pi_{person\text{-}name,\, city}\,(employee \bowtie (\sigma_{company\text{-}name\, =\, \text{“First Bank Corporation”}}\,(works)))$$

- employee (person-name, street, city)
- works (person-name, company-name, salary)
- company (company-name, city)
- manages (person-name, manager-name)

Find the names, street address, and cities of residence of all employees who work for First Bank Corporation and earn more than $10,000 per annum.

$$\Pi_{person\text{-}name,\ street,\ city}$$
$$(\sigma_{(company\text{-}name\,=\,\text{“First Bank Corporation”}\ \wedge\ salary\ >\ 10000)}$$
$$works \bowtie employee)$$

- employee (person-name, street, city)
- works (person-name, company-name, salary)
- company (company-name, city)
- manages (person-name, manager-name)

Find the names of all employees in this database who live in the same city as the company for which they work.

$$\Pi_{person\text{-}name}\ (employee \bowtie works \bowtie company)$$

- employee (person-name, street, city)
- works (person-name, company-name, salary)
- company (company-name, city)
- manages (person-name, manager-name)

Find the names of all employees who live in the same city and on the same street as do their managers.

$$\Pi_{person\text{-}name}\left(\left(employee \bowtie manages\right) \bowtie_{(manager\text{-}name = employee2.person\text{-}name \land employee.street = employee2.street \land employee.city = employee2.city)} \left(\rho_{employee2}\left(employee\right)\right)\right)$$

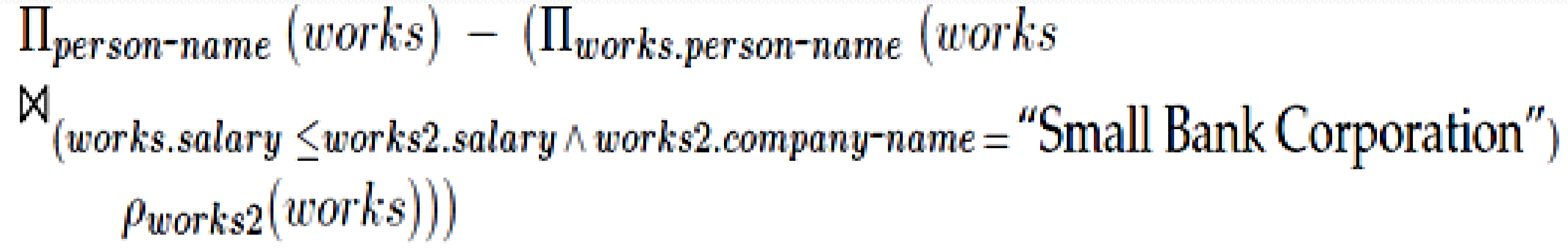
- employee (person-name, street, city)
- works (person-name, company-name, salary)
- company (company-name, city)
- manages (person-name, manager-name)

Find the names of all employees in this database who do not work for First Bank Corporation.

$$\Pi_{person\text{-}name}\left(\sigma_{company\text{-}name \neq \text{“First Bank Corporation”}}(works)\right)$$

- employee (person-name, street, city)
- works (person-name, company-name, salary)
- company (company-name, city)
- manages (person-name, manager-name)

Find the names of all employees who earn more than every employee of Small Bank Corporation.



$$\Pi_{person\text{-}name}(works) - (\Pi_{works.person\text{-}name}(works \bowtie_{(works.salary \leq works2.salary \wedge works2.company\text{-}name = \text{“Small Bank Corporation”})} \rho_{works2}(works)))$$

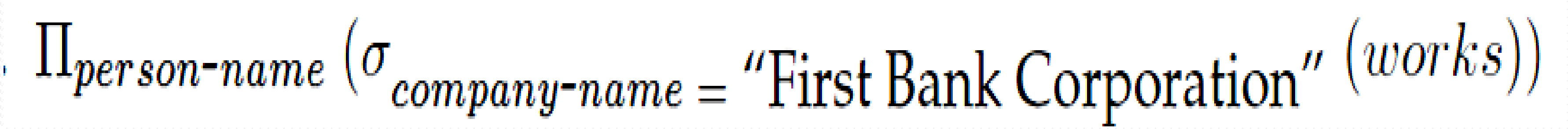
- employee (person-name, street, city)
- works (person-name, company-name, salary)
- company (company-name, city)
- manages (person-name, manager-name)

Assume the companies may be located in several cities. Find all companies located in every city in which Small Bank Corporation is located.

$$\Pi_{company\text{-}name}\,(company \div (\Pi_{city}\,(\sigma_{company\text{-}name = \text{“Small Bank Corporation”}}\,(company))))$$

- employee (person-name, street, city)
- works (person-name, company-name, salary)
- company (company-name, city)
- manages (person-name, manager-name)

Find the names of all employees who work for First Bank Corporation.



$$\Pi_{person\text{-}name}\left(\sigma_{company\text{-}name\,=\,\text{“First Bank Corporation”}}\left(works\right)\right)$$

Example : 2

- 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.

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

- 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 all information about students whose major is CS.

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

- 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 all students with the books they can borrow.

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

- 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 all books published by McGraw-Hill before 1990.

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

- 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 name of those authors who are living in Davis.

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

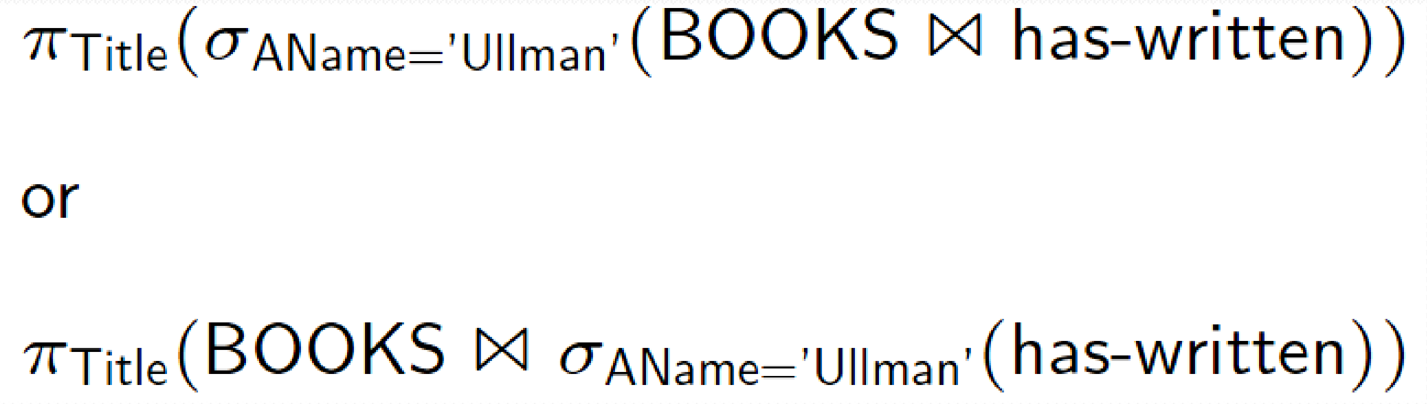
- BOOKS(DocId, Title, Publisher, Year)
- STUDENTS(StId, StName, Major, Age)
- AUTHORS(AName, Address)
- borrows(DocId, StId, Date)
- has-written(DocId, AName)
- describes(DocId, Keyword)

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

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

- 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 title of books written by the author 'Ullman'.



$$\pi_{\text{Title}}(\sigma_{\text{AName='Ullman'}}(\text{BOOKS} \bowtie \text{has-written}))$$

or

$$\pi_{\text{Title}}(\text{BOOKS} \bowtie \sigma_{\text{AName='Ullman'}}(\text{has-written}))$$

- 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 authors of the books the student 'Smith' has borrowed.

$$\pi_{\text{AName}}(\sigma_{\text{StName='Smith'}}(\text{has-written} \bowtie (\text{borrows} \bowtie \text{STUDENTS})))$$

OR

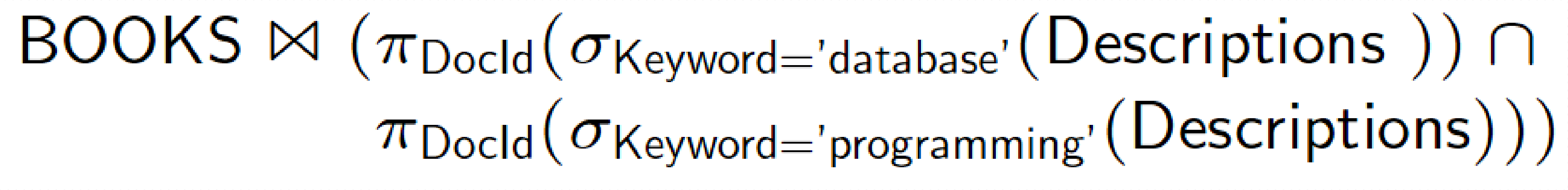
$$\text{temp1} \longleftarrow \text{borrows} \bowtie \text{STUDENTS}$$

$$\text{temp2} \longleftarrow \text{has-written} \bowtie \text{temp1}$$

$$\text{result} \longleftarrow \pi_{\text{AName}}(\sigma_{\text{StName='Smith'}}(\text{temp2}))$$

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

Which books have both keywords 'database' and 'programming‘?



$$\text{BOOKS} \bowtie (\pi_{\text{DocId}}(\sigma_{\text{Keyword}='\text{database}'}(\text{Descriptions})) \cap \pi_{\text{DocId}}(\sigma_{\text{Keyword}='\text{programming}'}(\text{Descriptions})))$$

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

Find the name of the youngest student.

$$\begin{gathered}\pi_{\mathrm{StName}}(\mathrm{STUDENTS}) - \\ \pi_{\mathrm{S1.StName}}(\sigma_{\mathrm{S1.Age}>\mathrm{S2.Age}}(\rho_{\mathrm{S1}}(\mathrm{STUDENTS}) \times \rho_{\mathrm{S2}}(\mathrm{STUDENTS})))\end{gathered}$$

Example : 3

- Sailors(sid, sname ,rating, age)
- Reserves(sid, bid, day)
- Boats(bid ,bname, color)

Find names of sailors who’ve reserved boat #103

$$\pi_{sname}((\sigma_{bid=103}\text{Reserves}) \bowtie \ Sailors)$$

OR

$$\pi_{sname}(\sigma_{bid=103}(\text{Reserves} \bowtie Sailors))$$

- Sailors(sid, sname ,rating, age)
- Reserves(sid, bid, day)
- Boats(bid ,bname, color)

Find names of sailors who’ve reserved a red boat

$$\pi_{sname}((\sigma_{color='red'} Boats) \bowtie \mathrm{Reserves} \bowtie Sailors)$$

OR

$$\pi_{sname}(\pi_{sid}((\pi_{bid}\sigma_{color='red'} Boats) \bowtie \mathrm{Res}) \bowtie Sailors)$$

- Sailors(sid, sname ,rating, age)
- Reserves(sid, bid, day)
- Boats(bid ,bname, color)

Find sailors who’ve reserved a red and a green boat

$$Tempred \longrightarrow \pi_{sid}((\sigma_{color='red'} Boats) \bowtie \text{Reserves}))$$

$$Tempgreen \longrightarrow \pi_{sid}((\sigma_{color='green'} Boats) \bowtie \text{Reserves}))$$

$$\pi_{sname}((Tempred \cap Tempgreen) \bowtie Sailors)$$

- Sailors(sid, sname ,rating, age)
- Reserves(sid, bid, day)
- Boats(bid ,bname, color)

Find the names of sailors who’ve reserved all boats

$$Tempsids \longrightarrow (\pi_{sid,bid}\mathrm{Reserves}) / (\pi_{bid} Boats))$$

$$\pi_{sname}(Tempsids \bowtie Sailors)$$

- Sailors(sid, sname ,rating, age)
- Reserves(sid, bid, day)
- Boats(bid ,bname, color)

Find sailors who've reserved a red or a green boat

$$Tempboats \longrightarrow (\sigma_{color='red' \vee color='green'} Boats))$$

$$\pi_{sname}(Tempboats \bowtie \mathrm{Reserves} \bowtie Sailors)$$

- Sailors(sid, sname ,rating, age)
- Reserves(sid, bid, day)
- Boats(bid ,bname, color)

Find names of sailors who’ve reserved boat #103

$$\pi_{sname}((\sigma_{bid=103}\text{Reserves})\bowtie \ Sailors)$$

OR

$$\pi_{sname}(\sigma_{bid=103}(\text{Reserves}\bowtie Sailors))$$