

1 Exercise 1

- (1) Find the names of IT department employees who do not work on any project that the IT department runs.

IT department employees:

$$Employee_IT \leftarrow \rho_{Employee_IT}(\pi_{(SSN,name)}(Employee \bowtie_{Employee.Dno=Department.Dnumber \wedge Department.name='IT'} Department))$$

projects that the IT department runs:

$$Project_IT \leftarrow \rho_{Project_IT}(\pi_{Pnumber}(Project \bowtie_{Project.Dno=Department.Dnumber \wedge Department.name='IT'} Department))$$

Employee who work on the projects that the IT department runs:

$$Employee_Project_IT \leftarrow \rho_{Employee_Project_IT}(\pi_{Essn}(Works_on \bowtie_{Pno=Pnumber} Project_IT))$$

the names of IT department employees who do not work on any project that the IT department runs:

$$\pi_{name}(Employee_IT) - \pi_{name}(Employee_IT \bowtie_{SSN=Essn} Employee_Project_IT)$$

- (2) Find the names of departments that are located at the same location as the IT department.

join department with its location:

$$D \leftarrow \rho_D(Department \bowtie Dept_location)$$

$$D_1 \leftarrow \rho_{D_1}(D)$$

$$D_2 \leftarrow \rho_{D_2}(D)$$

get the names of departments that are located at the same location as IT department ('IT' is not included):

$$\pi_{D_2.name}(D_1 \bowtie_{D_1.Dlocation=D_2.Dlocation \wedge D_1.name='IT' \wedge D_2.name \neq 'IT'} D_2)$$

- (3) Find the names of employees who work the most hours on projects that the IT department runs.

projects that the IT department runs:

$$Project_IT \leftarrow \rho_{Project_IT}(\pi_{Pnumber}(Project \bowtie_{Project.Dno=Department.Dnumber \wedge Department.name='IT'} Department))$$

Employee who work on the projects that the IT department runs:

$$E \leftarrow \rho_E(Works_on \bowtie_{Works_on.Pno=Project_IT.Pnumber} Project_IT)$$

$$E_1 \leftarrow \rho_{E_1}(E)$$

$$E_2 \leftarrow \rho_{E_2}(E)$$

get names of employees who work the most hours:

$$\pi_{name}((\pi_{Essn}(E) - \pi_{E_2.Essn}(E_1 \bowtie_{E_1.hours > E_2.hours} E_2)) \bowtie_{Essn=SSN} Employee)$$

- (4) Find the names of employees who work on both Project 'A' and Project 'B'.

employees who work on Project 'A':

$$E_PA \leftarrow \rho_{E_PA}(\pi_{Essn}(Project \bowtie_{Project.Pnumber=Works_on.Pno \wedge Project.name='A'} Works_on))$$

employees who work on Project 'B':

$$E_{P_B} \leftarrow \rho_{E_{P_B}}(\pi_{E_{SSN}}(Project \bowtie_{Project.Pnumber=Works_on.Pno \wedge Project.name='B'} Works_on))$$

names of employees who work on both:

$$\pi_{name}(Employee \bowtie_{SSN=E_{SSN}} (E_{P_A} \cap E_{P_B}))$$

(5) Find the name of employees who work on every project run by the IT department.

projects run by the IT department, rename Pnumber to Pno:

$$E_{IT} \leftarrow \rho_{E_{IT}}(\pi_{Pnumber}(Project \bowtie_{Project.Dno=Department.Dnumber \wedge Department.name='IT'} Department))$$

$$\rho_{Pno \leftarrow Pnumber}(E_{IT})$$

divide all projects, get the name of employees:

$$\pi_{name}((\pi_{E_{SSN}}(Works_on \div E_{IT})) \bowtie_{E_{SSN}=SSN} Employee)$$

2 Exercise 2

- (1) Find the names of customers who have booked flights at every company in the US.

$$\pi_{name}((\pi_{customer_ID}(Book \div \pi_{company_ID}(\sigma_{location='US'}(Company)))) \bowtie Customer)$$

- (2) Find the names of customers who have never booked a flight.

$$\pi_{name}((\pi_{customer_ID}(Customer) - \pi_{customer_ID}(Book)) \bowtie Customer)$$

- (3) Find the names of customers who booked the flight to New York more than once.

customers who booked the flight to New York:

$$C \leftarrow \rho_C(\pi_{(company_ID, flight_number, customer_ID)}(\sigma_{Flight.destination='NewYork'}(Book \bowtie Flight)))$$

$$C_1 \leftarrow \rho_{C_1}(C)$$

$$C_2 \leftarrow \rho_{C_2}(C)$$

two flights are booked by the same customer with different company_ID OR different flight_number:

$$\rho_{C_id}(\pi_{C_1.customer_ID}(C_1 \bowtie_{C_1.customer_ID=C_2.customer_ID \wedge (C_1.flight_number \neq C_2.flight_number \vee C_1.company_ID \neq C_2.company_ID)} C_2))$$

$$\pi_{name}(C_id \bowtie Customer)$$

- (4) Find the names of the companies that have the biggest (in terms of the number of seats) airplane.

$$F_1 \leftarrow \rho_{F_1}(Flight)$$

$$F_2 \leftarrow \rho_{F_2}(Flight)$$

get company_ID and flight_number of the biggest airplane:

$$B \leftarrow \rho_B(\pi_{(company_ID, flight_number)}(Flight) - \pi_{(F_2.company_ID, F_2.flight_number)}(F_1 \bowtie_{F_1.number_of_seats > F_2.number_of_seats} F_2))$$

$$\pi_{company_name}(Company \bowtie B)$$

- (5) Find the names of customers who booked flights to Boston and Seattle.

$$\pi_{name}(((\pi_{customer_ID}(\sigma_{Flight.destination='Boston'}(Book \bowtie Flight))) \cap (\pi_{customer_ID}(\sigma_{Flight.destination='Seattle'}(Book \bowtie Flight)))) \bowtie Customer)$$

- (6) Find all destinations (unequal to Gainesville) of cheap flights (i=500) provided by companies that also provide flights to Gainesville.

get the flights provided by companies that provide flights to Gainesville:

$$Fc \leftarrow \rho_{Fc}((\pi_{company_ID}(\sigma_{destination='Gainesville'}(Flight))) \bowtie Flight)$$

get destinations of flights have cheap prices:

$$\pi_{destination}(\sigma_{price \leq 500 \wedge destination \neq 'Gainesville'}(Fc))$$

3 Exercise 3

- (1) Find the names and GPA of CISE students who have enrolled in the DB class.

get the students who enrolled in DB class:

$$S_{DB} \leftarrow \rho_{S_{DB}}(\pi_{student_ID}((\pi_{class_ID}(\sigma_{name='DB'}(Class))) \bowtie Enroll))$$

get the names and gpa of CISE students:

$$\pi_{(name,gpa)}(\sigma_{major='CISE'}(S_{DB} \bowtie Student))$$

- (2) Find the names and major of students who have enrolled in the DB class on Wednesdays and the Algorithm class on Thursdays but have not enrolled in the OS class on Fridays.

students who enrolled in the DB class on Wednesday:

$$S_{DB} \leftarrow \rho_{S_{DB}}(\pi_{student_ID}((\pi_{class_ID}(\sigma_{name='DB' \wedge day='Wednesday'}(Class))) \bowtie Enroll))$$

students who enrolled in the Algorithm class on Thursday:

$$S_A \leftarrow \rho_{S_A}(\pi_{student_ID}((\pi_{class_ID}(\sigma_{name='Algorithm' \wedge day='Thursday'}(Class))) \bowtie Enroll))$$

students who enrolled in the OS class on Friday:

$$S_{OS} \leftarrow \rho_{S_{OS}}(\pi_{student_ID}((\pi_{class_ID}(\sigma_{name='OS' \wedge day='Friday'}(Class))) \bowtie Enroll))$$

get the results:

$$\pi_{(name,major)}((S_{DB} \cap S_A \cap (\pi_{student_ID}(Student) - S_{OS})) \bowtie Students)$$

- (3) Find the names and major of students who have enrolled in every class taught by 'James'.

students who enrolled in all class taught by 'James':

$$S_J \leftarrow \rho_{S_J}(\pi_{student_ID}(Enroll \div \pi_{class_ID}(\sigma_{lecturer='James'}(Class))))$$

get the results:

$$\pi_{(name,major)}(S_J \bowtie Students)$$

- (4) Find the names of students who have enrolled in the DB class more than twice.

students who enrolled in the DB class:

$$E \leftarrow \rho_E(\pi_{(student_ID,semester)}((\pi_{class_ID}(\sigma_{name='DB'}(Class))) \bowtie Enroll))$$

$$E_1 \leftarrow \rho_{E_1}(E)$$

$$E_2 \leftarrow \rho_{E_2}(E)$$

$$E_3 \leftarrow \rho_{E_3}(E)$$

the records of the same student:

$$T \leftarrow \rho_T(\sigma_{E_1.student_ID=E_2.student_ID \wedge E_2.student_ID=E_3.student_ID}(E_1 \times E_2 \times E_3))$$

three semester are all different, get the results:

$$\pi_{name}((\pi_{E_1.student_ID}(\sigma_{E_1.semester \neq E_2.semester \wedge E_2.semester \neq E_3.semester \wedge E_1.semester \neq E_3.semester}(T))) \bowtie Students)$$

(5) Find the names of students who have not enrolled in the DB class.

students who enrolled in the DB class:

$$S_{DB} \leftarrow \rho_{S_{DB}}(\pi_{student_ID}((\pi_{class_ID}(\sigma_{name='DB'}(Class))) \bowtie Enroll))$$

get the results:

$$\pi_{name}((\pi_{student_ID}(Student) - S_{DB}) \bowtie Students)$$

4 Exercise 4

1. The term is not correct in general. Only when $A_1 \subseteq A_2 \subseteq A_3 \dots \subseteq A_n$ the statement is true.

Proof:

We assume that the relation R have the schema $R(X_1 : D_1, X_2 : D_2, \dots, X_n : D_n)$, let two subsets A_i and $A_j (\exists i, j, 1 \leq i, j \leq n \text{ and } i < j)$. Assume that A_i has several attributes, $A_i(X_a, X_b, X_c \dots)$ and A_j also have several attributes $A_j(X_x, X_y, X_z \dots)$. Then we consider about the statement $\pi_{A_i}(\pi_{A_j}(R)) = \pi_{A_i}(R)$.

Since all of the attributes of A_j are contained in R , so the result R' of $\pi_{A_j}(R)$ is the same schema as A_j , then we have $\pi_{A_i}(R')$. However, there might be different results according to the relation between A_i and A_j .

(1) if all of the attributes of A_i are not contained in A_j , it is obvious that $\pi_{A_i}(R') = \emptyset$. So the statement won't be true.

(2) if some of the attributes of A_i are not contained in A_j , it is also obvious that there won't be any results for $\pi_{A_i}(R')$. So the statement is not true.

(3) if all of the attributes of A_i are contained in A_j , then the result of $\pi_{A_i}(R')$: R'' has the same schema as $\pi_{A_i}(R)$. So the statement is true.

To conclude, only when all of the attributes of A_i are contained in A_j , which is $A_i \subseteq A_j$, then $\pi_{A_i}(\pi_{A_j}(R)) = \pi_{A_i}(R)$.

It is also the same when we consider the term of the problem, only when A_i is contained by A_j , for $\forall i, j, 1 \leq i < j \leq n$, which is $A_1 \subseteq A_2 \subseteq A_3 \dots \subseteq A_n$, then the term will be true.

2. Only when all of the attribute names contained in F are also contain in A , the statement is true.

Proof:

We assume that relation A has the schema $A(A_i, \dots, A_j)$, in which $\forall i, j, 1 \leq i < j \leq n$, and the set of attributes which names are involved in F : $B(A_k, A_l \dots)$.

(1) if all of the attributes in B are not included in A , then there will be no result for $\sigma_F(\pi_A(R))$.

(2) if some of the attributes in B are not included in A , then there will also be no result for $\sigma_F(\pi_A(R))$.

(3) if all of the attributes in B are included in A , then the schema of $\sigma_F(\pi_A(R))$ and $\pi_A(\sigma_F(R))$ will be the same. What's more, since all of the attribute name in F are contained in A , there is no difference in tuples between them.

To conclude, only when all of the attribute names contained in F are also contain in A , the statement is true.

3. a. We consider $R(A, B) \cup T(A, B)$

Maximum: when R and T don't have duplicate tuples, which is $R \cap T = \emptyset$, then the number is $r+s$.

Minimum:

(1) $R = T$, then the result is r, s ;

(2) $R \subseteq T$, then the result is s ;

(3) $T \subseteq R$, then the result is r ;

As a result, the minimum number of tuples is $\max(r, s)$.

b. Exactly, $R \bowtie R = R$, then we can know $R \bowtie (R \bowtie R) = R$.

To conclude, Minimum is r and Maximum is also r .

c. To consider $\pi_{A,C}(R \bowtie S)$, we can consider the result of $R \bowtie S$.

Maximum: when all the values of B in either R and S are same, which means $|\pi_B(R)| = |\pi_B(S)| = 1$, then $|R \bowtie S|$ will be the maximum, so the maximum number is $r \cdot s$.

Minimum: when $R \bowtie S = \emptyset$. there will be no result for projection, so the minimum is 0.

d. To consider $\sigma_{A=C}(R \bowtie S)$, we can also consider the result of $R \bowtie S$.

Maximum: when all the values of B in either R and S are same, which means $|\pi_B(R)| = |\pi_B(S)| = 1$, then the maximum number is depend on the minimum of $|\pi_A(R)|$ and $|\pi_C(S)|$. As a result, the maximum number is $\min(r, s)$.

Minimum:

(1) when $R \bowtie S = \emptyset$, there will be no result for projection, so the minimum is 0.

(2) There is another situation, although $R \bowtie S \neq \emptyset$, there is no tuple $A = C$.

To conclude, the minimum number is 0.