Tutorial 2

Relational Algebra

Relational Algebra - operations

- Given the following tables:

  employee (person_name, street, city)

  works (person_name, company_name, salary)

  company (company_name, city)

  manages (person_name, manager_name)

- **Select:** Find all people who work in “First Bank Corporation”
- **Project:** Find all person names that have a job to work.
- **Union:** Find every person who is either an employee or a manager (there is automatic duplicate elimination in all algebra operations)
- **Intersection:** Find all persons that are both employee and manager.
- **Set Difference:** Find all persons that are employee but not manager.
- **Rename:** Rename the table `employee` to `employee2`
- **Cartesian Product:** Produces all possible combinations of records from employee and works.
- **Natural Join:** Show the employee names, city, and the salary for their jobs (for employees who work in some company).
- **Division:** Find all persons that work in all companies.

Relational Algebra - operations

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

manages (person_name, manager_name)

- Select: Find all people who work in “First Bank Corporation”

  $\sigma_{company\_name = \text{“First Bank Corporation”}}(works)$

- Projection: Find all person names that have work.

  $\pi_{person\_name}(works)$

- Union: Find every person who is either an employee or a manager (there is automatic duplicate elimination in all algebra operations)

  $\pi_{person\_name}(employee) \cup \pi_{manager\_name}(manages)$

- Intersection: Find all persons that are both employee and manager.

  $\pi_{person\_name}(employee) \cap \pi_{manager\_name}(manages)$

Relational Algebra - operations

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

manages (person_name, manager_name)

- Set difference: Find all persons that are employee but NOT manager.

  $\pi_{person\_name}(employee) - \pi_{manager\_name}(manages)$

- Rename: Rename the table *employee* to *employee2*

  $\rho(employee2, employee)$

- Cartesian Product: Produces all possible combinations of records from employee and works.

  $employee \times works$

- Natural Join: Show the employee names, city, and the salary for their jobs (for employees who work in some company).

  $\pi_{person\_name,\ city,\ salary}(employee \bowtie works)$

Relational Algebra - operations

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

manages (person_naame, manager_name)

- Division: Find all persons that work in ALL companies.

$(\pi_{person\_name,\ company\_name}(works)) / (\pi_{company\_name}(company))$

Relational Algebra Exercise

- Given the following tables:

  employee (person_name, street, city)

  works (person_name, company_name, salary)

  company (company_name, city)

  manages (person_name, manager_name)

1. Find the names of all persons who work for "First Bank Corporation"
2. Find the names and cities of employees who work for "First Bank Corporation"
3. Find the name of all managers who earn more than $10,000
4. Find the worker(s) with highest salary
5. Find the names of employees who do not work for "First Bank Corporation"
6. Find the employees that hold a job in all companies located in Hong Kong
7. Find the names of employees who live in the same city and on the same street as their managers

Relational Algebra Exercise

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 persons who work for "First Bank Corporation"

$\pi_{person\_name}\,(\sigma_{company\_name} = \text{“First Bank Corporation”}\,(works))$

Relational Algebra Exercise

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 employees who work for "First Bank Corporation"

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

Relational Algebra Exercise

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

manages (person_name, manager_name)

- Find the name of all managers who earn more than $10,000

$\pi_{person\_name}(\sigma_{salary > \text{“10000”}}(works)) \cap \pi_{manager\_name}(manages)$

Relational Algebra Exercise

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

manages (person_name, manager_name)

- Find the worker(s) with highest salary

$$\pi_{person\_name}(works) - \pi_{w1.person\_name}($$

$$\sigma_{w1.salary<w2.salary}(\rho(w1, works) \times \rho(w2, works)))$$

Relational Algebra Exercise

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

manages (person_name, manager_name)

- Find the names of employees who do not work for "First Bank Corporation"

$\pi_{person\_name}(employee)$ -

$\pi_{person\_name}(\sigma_{company\_name = \text{"First Bank Corporation"}} works)$

If every employee must have a work, it can also be:

$\pi_{person\_name}(\sigma_{\text{company\_name} \neq \text{"First Bank Corporation"}} works)$

Relational Algebra Exercise

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

manages (person_name, manager_name)

- Find the employees that hold a job in all companies located in Zhuhai.

  Tip: *all*: where division always happens

$\pi_{person\_name,\ company\_name}\ works\ /$

$\pi_{company\_name}(\sigma_{city = \text{“}Zhuhai\text{”}}\ company)$

Relational Algebra Exercise

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

manages (person_name, manager_name)

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

$$\pi_{employee.person\_name}($$
$$(\sigma_{employee.city = employee2.city \wedge employee.street = employee2.street}$$
$$(employee \bowtie_{employee.person\_name = manages.person\_name} manages$$
$$\bowtie_{manages.manager\_name = employee2.person\_name} \rho_{(employee2, employee)})))$$

compound θ-Join + rename(because of comparisons)

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

$$\pi_{employee.person\_name}($$
$$(\sigma_{employee.city = employee2.city \wedge employee.street = employee2.street}$$
$$(employee \bowtie_{employee.person\_name = manages.person\_name} manages$$
$$\bowtie_{manages.manager\_name = employee2.person\_name} \rho_{(employee2, employee)})))$$

红线部分运算出来的表如下：其join目的为找出manager的地址

Manages. person_name	Manages. Manager_name	Employee2. person_name	Employee2. city	Employee2. street

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

$$\pi_{employee.person\_name}($$
$$(\sigma_{employee.city = employee2.city \wedge employee.street = employee2.street}$$
$$(employee \bowtie_{employee.person\_name = manages.person\_name} manages$$
$$\bowtie_{manages.manager\_name = employee2.person\_name} \rho_{(employee2, employee)})))$$

红线部分运算出来的表如下：其join目的为找出employee的manager

Manages. person_name	Manages. Manager_name	employee.person_name	employee.per-son_city	employee.per-son_street

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

$$\pi_{employee.person\_name}($$
$$(\sigma_{employee.city = employee2.city \wedge employee.street = employee2.street}$$
$$(employee \bowtie_{employee.person\_name = manages.person\_name} manages$$
$$\bowtie_{manages.manager\_name = employee2.person\_name} \rho_{(employee2, employee)})))$$

结合前两步，则红线部分选择运算得到表格如下：

Manages. person_name	Manages. Manager_name	employee.person_name	employee.person_city	employee.person_street	Employee2. person_name	Employee2. city	Employee2. street

第1，3，4，5列是employee属性，
第2，6，7，8是manager属性

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

$$\pi_{employee.person\_name}((\sigma_{employee.city = employee2.city \wedge employee.street = employee2.street} (employee \bowtie_{employee.person\_name = manages.person\_name} manages \bowtie_{manages.manager\_name = employee2.person\_name} \rho_{(employee2, employee)})))$$

则红线部分选择运算基于如下表格，其选择条件目的为找出employee的地址跟其manager地址一样的纪录。然后再投影 只保留employee的名字。

Manages. person_name	Manages. Manager_name	employee.person_name	employee.person_city	employee.person_street	Employee2. person_name	Employee2. city	Employee2. street