employee (id, person name, street, city)
works (id, person name, company name, salary)
company (id, company name, city)

1. Consider the employee database of figure below. Give an expression in the relational algebra to express each of the following queries:

1.1 Find the ID, name of each employee who works for “BigBank”.

Answer -> $\Pi_{(ID, person\_name)}( \sigma_{(company\_name = \text{“BigBank”}}(works)))$

1.2 Find the ID, name, and city of residence of each employee who works for “BigBank”.

Answer -> $\Pi_{(ID, person\_name, city)}$ $(employee \bowtie_{(employee.id = works.id)} (\sigma_{(company\_name = \text{“BigBank”}}(works)))$

1.3 Find the ID, name, street address, and city of residence of each employee who works for “BigBank” and earns more than $10000.

Answer - > $\Pi_{(ID, person\_name, street\ address, city)}$ $(\sigma_{(company\_name = \text{“BigBank”} \text{ and } (salary > 10000}$ $(works \bowtie_{(employee.id = works.id)} employee ))$

1.4 Find the ID and name of each employee in this database who lives in the same city as the company for which she or he works.

Answer - > $\Pi_{(ID, person\_name)}$ $\sigma_{(employee.city = company.city}(employee_{(employee.id = works.id} \bowtie works \bowtie_{(employee.id = company.id)} company ))$

2. Consider the employee database of figure above. Give an expression in the relational algebra to express each of the following queries:

2.1 Find the ID and name of each employee who does not work for “BigBank”.

Answer - > $\Pi_{(ID, person\_name)}$ $\sigma_{(company\_name \neq \text{“BigBank”}}(works))$

2.2 Find the ID and name of each employee who earns at least as much as every employee in the database.

Answer -> $\Pi_{(ID, person\_name)}$ $\sigma_{(salary = aveSalary(employee}$ $\bowtie_{(employee.id = works.id)}$ works $\bowtie_{(works.id = company.id)}$ company))

3. Consider the foreign-key constraint from the dept_name attribute of instructor to the department relation. Give examples of inserts and deletes to these relations that can cause a violation of the foreign-key constraint.

1.Inserting the tuple(02344, Hello, algebra, 120000) into table instructor, where doesn’t have the Department algebra, would violate the foreign key constaint.

2.Deleting the tuple(Finance, Painter, 120000) from the department table, where at least one student or instructor tuple has dept_name as Finance, would violate the foreign key constraint.

4. Consider the employee database of figure above. What are the appropriate primary keys?

Answer -> 1. id, 2. id, 3. company_name and id are foreign key