Laboratory work 2

Please write your answers to the pdf file for defense:

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

   employee (ID, person_name, street, city)
   works (ID, person_name, company_name, salary)
   company (company_name, city)

Figure

- Find the ID and name of each employee who works for “BigBank”.

$$\Pi_{ID,\ \ person\_name}(\sigma_{company\_name=\text{"BigBank"}}(works))$$

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

$$\Pi_{ID,\ \ person\_name,city}(\sigma_{company\_name=\text{"BigBank"}}(employee * works))$$

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

$$\Pi_{ID,\ \ person\_name,street,city}(\sigma_{company\_name=\text{"BigBank"}\text{^}\text{salary} > 10000}(employee * works))$$

- 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.

$$\Pi_{ID,\ \ person\_name,}(\sigma_{employee.city=company.city}(employee * company))$$

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

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

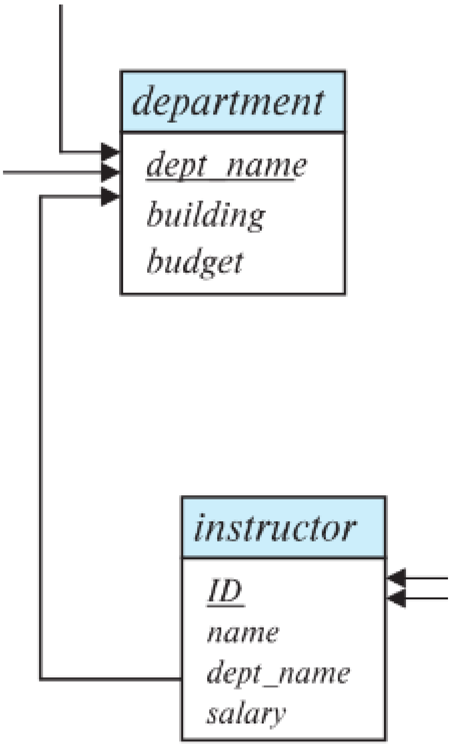
$$\Pi_{ID,\ \ person\_name}(\sigma_{company\_name\neq\text{"BigBank"}}(employee))$$

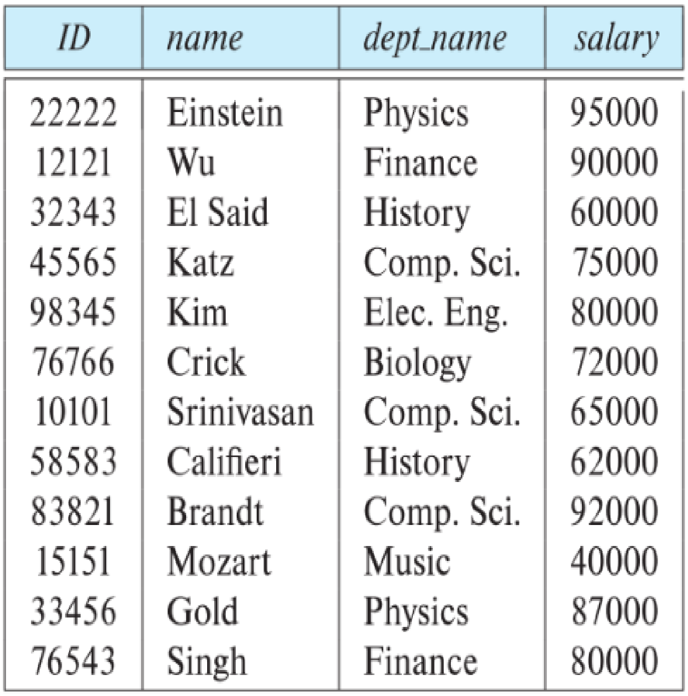
- Find the ID and name of each employee who earns at least as much as every employee in the database.

$$minimal \leftarrow \Pi_{ID,\ person_{name}}(\sigma_{works.salary \le someworks.salary}(works * \rho_{someworks}(works)))$$

$$\Pi_{ID,\ person_{name}}(works) - minimal$$

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.





ID	name	dept_name	salary
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

From this relation we can assume, that both *department* and *instructor* DB-s have had some data in **(dept_name, building, budget)** and **(ID, name, dept_name, salary)** respectively.

So, if we add some **new dept_name** to instructor's DB, it accuses error, because we don't have **the new dept_name** in our *department* DB. And reversely, if we delete **dept_name** from *department*, it accuses error, because we might have some person in *instructor* that have the **same dept_name.**

Examples:

Insert (1234, Azamat, Philosophy, 9 999 999) to instructor. There is no **dept_name** in *department* with Philosophy, So + error.

Assume, there is a person in *instructor* (ID, Aza, Music, 10 000). Delete (Music, Dom63, 123 456) from *department*. + error, because *instructor's* **dept_name** still referencing to *department* with no "Music". Paradox

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

employee (ID, person_name, street, city)
works (ID, person_name, company_name, salary)
company (company_name, city)