* 1. **Part 1**

Q-1 Consider the following employee database schema. What are the appropriate primary keys? (You may identify them by underlining them).

employee (person name, street, city)

works (person name, company name, salary)

company (company name, city)

Q-2 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.

Inserts:

1. ( 10000, Jones, Anthropology, 44000) added to the instructor table. This violates foreign-key constraint since the department table does not have an Anthropology department.
2. ( 22000, Keller, Linguistics, 55000)

Deletes:

1. (Elec. Engineering, Taylor,85000) deleted from the department table. This violates foreign-key constraint since there is Elec. Engineering under dept\_name.
2. (Biology, Watson, 90000)

Q-3 Consider the employee database described above. Give an expression in the relational algebra to express each of the following queries:

1. Find the name of each employee who lives in city “Miami”.

∏ person\_name (σ city = “Miami” (employee))

1. Find the name of each employee whose salary is greater than $100000.

∏ person\_name (σ salary > 100000 (employee ⨝ works))

1. Find the name of each employee who lives in “Miami” and whose salary is greater than $100000

∏ person\_name  (σ city = “Miami” ^ salary > 100000 ( employee ⨝ works))

Q-4 In the instance of instructor shown in Figure 1, no two instructors have the same name. From this, can we conclude that name can be used as a superkey (or primary key) of instructor?

Since super key is used to uniquely identify all attributes in a relation, it is not possible. In the instance of instructor of figure, name represents a unique value.

Q-5 Consider the bank database schema as follows.

branch (branch name, branch city, assets)

customer (ID, customer name, customer street, customer city)

loan (loan number, branch name, amount)

borrower (ID, loan number)

account (account number, branch name, balance)

depositor (ID, account number)

Give an expression in the relational algebra for each of the following queries:

1. Find the name of each branch located in “Chicago”.

∏ branch\_name (σ branch\_city = “Chicago” (branch))

1. Find the ID of each borrower who has a loan in branch “Downtown”.

∏ ID (σ branch\_name = “Downtown” ( borrower  ⨝borrow.ID = load.ID loan))

**1.2 Part 2**

Question 1 : What is a tuple?

(a) a table

(b) a row

(c) a column

(d) a specific instance

Answer: (b) a row

Question 2 : Which of these relational algebra is represented by π ?

(a) project

(b) join

(c) select

(d) rename

Answer: (a) project

Question 3 :  ΠJobTitle,Salary(jobs) yields which result?

|  |  |  |
| --- | --- | --- |
| JobID | JobTitle | Salary |
| 1 | Doctor | 200,000 |
| 2 | Lawyer | 100,000 |
| 3 | Scientist | 78,000 |
| 4 | Judge | 65,000 |

(a)

|  |
| --- |
| JobID |
| 1 |
| 2 |
| 3 |
| 4 |

(b)

|  |  |
| --- | --- |
| JobID | JobTitle |
| 1 | Doctor |
| 2 | Lawyer |
| 3 | Scientist |
| 4 | Judge |

(c)

|  |
| --- |
| JobTitle |
| Doctor |
| Lawyer |
| Scientist |
| Judge |

(d)

|  |  |
| --- | --- |
| JobTitle | Salary |
| Doctor | 200,00 |
| Lawyer | 100,000 |
| Scientist | 78,000 |
| Judge | 65,000 |

Answer: (d)

|  |  |
| --- | --- |
| JobTitle | Salary |
| Doctor | 200,00 |
| Lawyer | 100,000 |
| Scientist | 78,000 |
| Judge | 65,000 |

Question 4 : What does the cartesian product allow us to do?

(a) find s tuples that are in one relation but are not in another

(b) find tuples that are in both the input relations

(c) selects tuples that satisfy a given predicate

(d) combine information from any 2 relations

Answer: (d) combine information from any 2 relations

Question 5 : Which of the these are an example of a query language?

(a) declarative query language

(b) functional query language

(c) imperative query language

(d) all of the above

Answer: (d) all of the above