

1 Question 1

Consider the following relation:

employee_id	employee_name	job_title	years_exp	department
1	Nick	HTA	1	CS
2	Desmond	HTA	2	CS
3	Stan	Professor	30	CS
4	Junchi	UTA	1	CS
5	Huiyuan	UTA	2	CS

Given the above relation, provide the following:

1. A maximal super key (a super key with the largest possible size).
2. All possible candidate keys.
3. Which candidate key should we choose as our primary key (justify your response).

1.1 Solution

1. A maximal super key (a super key with the largest possible size).

The entire schema: (employee_id, employee_name, job_title, years_exp, department). A superkey just has to be a set of attributes such that every row can be uniquely determined by those attributes; since every row is unique, the maximal such set is the whole set.

2. All possible candidate keys.

The following: (employee_id), (employee_name), (job_title, years_exp). The first two sets are singletons with an attribute that is unique across all tables. The third set is minimal, since either one alone isn't unique across rows.

3. Which candidate key should we choose as our primary key (justify your response).

The (employee_id) candidate key is most suitable, since we never know when another employee will come along with the same name as an existing employee, but we can presumably control an employee's id, guaranteeing nonduplication.

2 Question 2

Consider the following relational schema:

```
student (student_id, name, age, year)
student_account (student_id, tuition, course)
```

Use relational algebra to answer the following queries:

1. Find names of students who have a tuition charge greater than or equal to \$10,000.

2. Find student_ids of students whose age is over 20, graduate in 2025, and will take the “Databases” course
3. Find all of pairs of ids of students who have the same name.
4. Find all of the students that pay more in tuition than another student with the same name as them.
5. Find the sum of tuition paid for each school year.

2.1 Solution

1. Find names of students who have a tuition charge greater than or equal to \$10,000.

$$\sigma_{tuition \geq 10,000}(student_id \mathbin{g}_{sum(tuition)}((student_account) \bowtie_{student_id} (student)))$$

2. Find student_ids of students whose age is over 20, graduate in 2025, and will take the “Databases” course

$$(\sigma_{course = \text{“Databases”}}(student_account)) \bowtie_{student_id} (\sigma_{age > 20 \wedge year = 2025}(student))$$

3. Find all of pairs of ids of distinct students who have the same name.

$$\sigma_{i.student_name = j.student_name}(\sigma_{i.student_id \neq j.student_id}(\rho_i(student) \times \rho_j(student)))$$

4. Find all of the students that pay more in tuition than another student with the same name as them.

$$i \leftarrow \rho_i(student_id \mathbin{g}_{sum(tuition)}(student \bowtie student_account))$$

$$j \leftarrow \rho_j(student_id \mathbin{g}_{sum(tuition)}(student \bowtie student_account))$$

$$\sigma_{i.student_name = j.student_name}(\sigma_{i.student_id \neq j.student_id}(\sigma_{i.tuition > j.tuition}(i \times j)))$$

5. Find the sum of tuition paid for each school year.

$$year \mathbin{g}_{SUM(tuition)}((student_account) \bowtie_{student_id} (student))$$

3 Question 3

The division (\div) relational operator identifies attribute values from a relation that can be paired with all of the values from the other relation. One way to think about it is that it is the inverse of the Cartesian Product (\times) relational operator: given relations R and S , $(R \times S) \div R = S$. Another way to think about it is that the tuples returned are the elements of R that are associated with every element of S . For another definition, see page 61 in the textbook.

Given this operator, consider the following relational schema:

```

student (student_id, name, age, year)
course (course_id, course_name, department)
takes (student_id, course_id)

```

Use relational algebra to answer the following queries:

1. Which courses were taken by every student?
2. Which courses were taking by every student graduating in 2023?
3. Which students took every course in the CS department?
4. Which students took every course in the CS department except for “Databases”? Note that you should not return any students that have taken “Databases”.

3.1 Solution

1. Which courses were taken by every student?

$$\pi_{course_id}(takes \div \pi_{student_id}(student))$$

2. Which courses were taking by every student graduating in 2023?

$$\pi_{course_id}(takes \div \pi_{student_id}(\sigma_{year=2023}(student)))$$

3. Which students took every course in the CS department?

$$takes \div \pi_{course_id}(\sigma_{department=CS}(course))$$

4. Which students took every course in the CS department except for “Databases”?

$$all_but_db \leftarrow takes \div \pi_{course_id}(\sigma_{department=CS \wedge course_name \neq "Databases"}(course))$$

$$took_db \leftarrow \pi_{student_id}((takes) \bowtie_{course_id} (\sigma_{department=CS \wedge course_name="Databases"}(course)))$$

$$all_but_db - took_db$$

Here, we’re getting all of the students that have taken all of the courses in the CS department except for Databases. This includes students that have taken all of the classes including databases as well. So, we get all of the students that have taken databases, and subtract them from the first result.

4 Question 4

Consider the following selected tables from some schema:

Bees:

name	cluster_no	job
Barry Benson	24601	7
Adam Flayman	47	7
Buzzwell	24601	7
Lou Lo Duca	42	7
Bob Bumble	42	7
Janet Benson	24601	8

Clusters:

cluster_no	established
42	2147483647
47	1630026000
24601	1658797200

In this question, we will explore how we can leverage properties of relational algebra to be more efficient.

- For each relational query, write out the intermediate tables after each calculation step.
 - $\sigma_{cluster_no=24601}(\sigma_{job=7}(bees)) \bowtie clusters$
 - $\sigma_{cluster_no=24601}(\sigma_{job=7}(bees \bowtie clusters))$
- How do the final tables for both queries compare?
- Sum the number of rows for each query that you counted in (2). Which query required you to write the least number of rows along the way in total?
- What qualities about your “least-expensive” query made it select less rows in total? Which operators were located where? Is there something particularly special about that?

1. Relational Queries

i. First query

- $\sigma_{Job=7}(Bees)$

Name	ClusterNo	Job
Barry Benson	24601	7
Adam Flayman	47	7
Janet Benson	24601	7
Lou Lo Duca	42	7
Bob Bumble	42	7

- $\sigma_{ClusterNo=24601}(\sigma_{Job=7}(Bees))$

Name	ClusterNo	Job
Barry Benson	24601	7
Buzzwell	24601	7

c. $\sigma_{ClusterNo=24601}(\sigma_{Job=7}(Bees)) \bowtie Clusters$

Name	ClusterNo	Job	Established
Barry Benson	24601	7	1658797200
Buzzwell	24601	7	1658797200

ii. $\sigma_{ClusterNo=24601}(\sigma_{Job=7}(Bees \bowtie Clusters))$

a. $Bees \bowtie Clusters$

Name	ClusterNo	Job	Established
Barry Benson	24601	7	1658797200
Adam Flayman	47	7	1630026000
Buzzwell	24601	7	1658797200
Lou Lo Duca	42	7	2147483647
Bob Bumble	42	7	2147483647
Janet Benson	24601	8	1658797200

b. $\sigma_{Job=7}(Bees \bowtie Clusters)$

Name	ClusterNo	Job	Established
Barry Benson	24601	7	1658797200
Adam Flayman	47	7	1630026000
Buzzwell	24601	7	1658797200
Lou Lo Duca	42	7	2147483647
Bob Bumble	42	7	2147483647

c. $\sigma_{ClusterNo=24601}(\sigma_{Job=7}(Bees \bowtie Clusters))$

Name	ClusterNo	Job	Established
Barry Benson	24601	7	1658797200
Buzzwell	24601	7	1658797200

2. The final tables are the same!

3. Dimensions

i. A table of 5 x 3 table, a 2 x 3 table, and then a 2 x 3 table. ($5 \times 3 + 2 \times 3 + 2 \times 3 = 27$)

- ii. A table of 5 x 4 table, a 5 x 4 table, and then a 2 x 3 table. ($5 \times 4 + 5 \times 4 + 2 \times 3 = 46$)

The first query uses less total cells!

4. In the first query, the selection came BEFORE the cross.

4.1 Note

For any tabular data in this assignment, you may want to consider using [an online table tool](#) to help you save time when you type up your answers.

Feedback

As this is a new course, we appreciate any feedback you have for us! If you enjoyed this assignment, hated this assignment, or have other thoughts to share, please do so [here](#)!