

CHAPTER 2



Introduction to the Relational Model

Practice Exercises

- 2.1 Consider the relational database of Figure ?? . What are the appropriate primary keys?

Answer: The answer is shown in Figure 2.1, with primary keys underlined.

- 2.2 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, which can cause a violation of the foreign key constraint.

Answer:

- Inserting a tuple:

(10111, Ostrom, Economics, 110,000)

into the *instructor* table, where the *department* table does not have the department Economics, would violate the foreign key constraint.

- Deleting the tuple:

(Biology, Watson, 90000)

from the *department* table, where at least one student or instructor tuple has *dept_name* as Biology, would violate the foreign key constraint.

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

Figure 2.1 Relational database for Practice Exercise 2.1.

- 2.3 Consider the *time_slot* relation. Given that a particular time slot can meet more than once in a week, explain why *day* and *start_time* are part of the primary key of this relation, while *end_time* is not.

Answer: The attributes *day* and *start_time* are part of the primary key since a particular class will most likely meet on several different days, and may even meet more than once in a day. However, *end_time* is not part of the primary key since a particular class that starts at a particular time on a particular day cannot end at more than one time.

- 2.4 In the instance of *instructor* shown in Figure ??, 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*?

Answer: No. For this possible instance of the instructor table the names are unique, but in general this may not be always the case (unless the university has a rule that two instructors cannot have the same name, which is a rather unlikely scenario).

- 2.5 What is the result of first performing the cross product of *student* and *advisor*, and then performing a selection operation on the result with the predicate $s_id = i_id$? (Using the symbolic notation of relational algebra, this query can be written as $\sigma_{s_id=i_id}(student \times advisor)$.)

Answer: The result attributes include all attribute values of student followed by all attributes of advisor. The tuples in the result are as follows. For each student who has an advisor, the result has a row containing that students attributes, followed by an *s_id* attribute identical to the students ID attribute, followed by the *i_id* attribute containing the ID of the students advisor.

Students who do not have an advisor will not appear in the result. A student who has more than one advisor will appear a corresponding number of times in the result.

- 2.6 Consider the following expressions, which use the result of a relational algebra operation as the input to another operation. For each expression, explain in words what the expression does.

- $\sigma_{year \geq 2009}(takes) \bowtie student$
- $\sigma_{year \geq 2009}(takes \bowtie student)$
- $\Pi_{ID, name, course_id}(student \bowtie takes)$

Answer:

- For each student who takes at least one course in 2009, display the students information along with the information about what courses the student took. The attributes in the result are:

ID, name, dept_name, tot_cred, course_id, section_id, semester, year, grade

- Same as (a); selection can be done before the join operation.
- Provide a list of consisting of

$ID, name, course_id$

of all students who took any course in the university.

2.7 Consider the relational database of Figure ?? . Give an expression in the relational algebra to express each of the following queries:

- Find the names of all employees who live in city “Miami”.
- Find the names of all employees whose salary is greater than \$100,000.
- Find the names of all employees who live in “Miami” and whose salary is greater than \$100,000.

Answer:

- $\Pi_{name} (\sigma_{city = \text{“Miami”}} (employee))$
- $\Pi_{name} (\sigma_{salary > 100000} (employee))$
- $\Pi_{name} (\sigma_{city = \text{“Miami”} \wedge salary > 100000} (employee))$

2.8 Consider the bank database of Figure ?? . Give an expression in the relational algebra for each of the following queries.

- Find the names of all branches located in “Chicago”.
- Find the names of all borrowers who have a loan in branch “Downtown”.

Answer:

- $\Pi_{branch_name} (\sigma_{branch_city = \text{“Chicago”}} (branch))$
- $\Pi_{customer_name} (\sigma_{branch_name = \text{“Downtown”}} (borrower \bowtie loan))$

