CHAPTER 2

Introduction to the Relational Model

Solutions for the Practice Exercises of Chapter 2

Practice Exercises

2.1

Answer:
The appropriate primary keys are shown below:

$$\begin{array}{l} \textit{employee}\ (\underline{\textit{person\_name}}, \textit{street}, \textit{city}) \\ \textit{works}\ (\underline{\textit{person\_name}}, \textit{company\_name}, \textit{salary}) \\ \textit{company}\ (\underline{\textit{company\_name}}, \textit{city}) \end{array}$$

2.2

Answer:

- Inserting a tuple:

  (10111, Ostrom, Economics, 110000)

  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.

2.3

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

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

2.5

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 student's attributes, followed by an *s_id* attribute identical to the student's 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

Answer:

a. $\Pi_{person\_name}\ (\sigma_{city\,=\,\text{“Miami”}}\ (employee))$

b. $\Pi_{person\_name}\ (\sigma_{salary\,>\,100000}\ (employee \bowtie works))$

c. $\Pi_{person\_name}\ (\sigma_{city\,=\,\text{“Miami”} \wedge salary > 100000}\ (employee \bowtie works))$

2.7

Answer:

a. $\Pi_{branch\_name}\ (\sigma_{branch\_city\,=\,\text{“Chicago”}}\ (branch))$

b. $\Pi_{ID}\ (\sigma_{branch\_name\,=\,\text{“Downtown”}}\ (borrower \bowtie_{borrower.ID=loan.ID} loan))$

2.8

Answer:

a. To find employees who do not work for BigBank, we first find all those who *do* work for BigBank. Those are exactly the employees *not* part of the desired result. We then use set difference to find the set of all employees minus those employees that should not be in the result.

$$\Pi_{ID,person\_name}(employee) -$$
$$\Pi_{ID,person\_name}$$
$$(employee \bowtie_{employee.ID=works.ID} (\sigma_{company\_name=\text{''BigBank''}}(works)))$$

b. We use the same approach as in part *a* by first finding those employess who do not earn the highest salary, or, said differently, for whom some other employee earns more. Since this involves comparing two employee salary values, we need to reference the *employee* relation twice and therefore use renaming.

$$\Pi_{ID,person\_name}(employee) -$$
$$\Pi_{A.ID,A.person\_name}(\rho_A(employee) \bowtie_{A.salary<B.salary} \rho_B(employee))$$

2.9

Answer:

a. $\Pi_{ID}(\Pi_{ID,course\_id}(takes) \div \Pi_{course\_id}(\sigma_{dept\_name=\text{'Comp. Sci'}}(course))$

b. The required expression is as follows:

$$r \leftarrow \Pi_{ID,course\_id}(takes)$$

$$s \leftarrow \Pi_{course\_id}(\sigma_{dept\_name=\text{'Comp. Sci'}}(course))$$

$$\Pi_{ID}\ (takes) - \Pi_{ID}\ ((\Pi_{ID}\ (takes) \times s) - r)$$

In general, let $r(R)$ and $s(S)$ be given, with $S \subseteq R$. Then we can express the division operation using basic relational algebra operations as follows:

$$r \div s = \Pi_{R-S}\ (r) - \Pi_{R-S}\ ((\Pi_{R-S}\ (r) \times s) - \Pi_{R-S,S}(r))$$

To see that this expression is true, we observe that $\Pi_{R-S}\ (r)$ gives us all tuples t that satisfy the first condition of the definition of division. The expression on the right side of the set difference operator

$$\Pi_{R-S}\ ((\Pi_{R-S}\ (r) \times s) - \Pi_{R-S,S}(r))$$

serves to eliminate those tuples that fail to satisfy the second condition of the definition of division. Let us see how it does so. Consider $\Pi_{R-S}\ (r) \times s$.

This relation is on schema R, and pairs every tuple in $\Pi_{R-S}(r)$ with every tuple in s. The expression $\Pi_{R-S,S}(r)$ merely reorders the attributes of r.

Thus, $(\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r)$ gives us those pairs of tuples from $\Pi_{R-S}(r)$ and s that do not appear in r. If a tuple t_j is in

$$\Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

then there is some tuple t_s in s that does not combine with tuple t_j to form a tuple in r. Thus, t_j holds a value for attributes $R - S$ that does not appear in $r \div s$. It is these values that we eliminate from $\Pi_{R-S}(r)$.