where the primary keys are underlined. Write a function *avg_salary* that takes a company name as an argument and finds the average salary of employees at that company. Then, write an SQL statement, using that function, to find companies whose employees earn a higher salary, on average, than the average salary at "First Bank".

5.16 Consider the relational schema

$$\textit{part}(\underline{\textit{part\_id}}, \textit{name}, \textit{cost})$$
$$\textit{subpart}(\underline{\textit{part\_id}}, \underline{\textit{subpart\_id}}, \textit{count})$$

where the primary-key attributes are underlined. A tuple $(p_1, p_2, 3)$ in the *subpart* relation denotes that the part with *part_id* p_2 is a direct subpart of the part with *part_id* p_1, and p_1 has 3 copies of p_2. Note that p_2 may itself have further subparts. Write a recursive SQL query that outputs the names of all subparts of the part with part-id 'P-100'.

5.17 Consider the relational schema from Exercise 5.16. Write a JDBC function using nonrecursive SQL to find the total cost of part "P-100", including the costs of all its subparts. Be sure to take into account the fact that a part may have multiple occurrences of a subpart. You may use recursion in Java if you wish.

5.18 Redo Exercise 5.12 using the language of your database system for coding stored procedures and functions. Note that you are likely to have to consult the online documentation for your system as a reference, since most systems use syntax differing from the SQL standard version followed in the text. Specifically, write a prodedure that takes an instructor *ID* as an argument and produces printed output in the format specified in Exercise 5.12, or an appropriate message if the instructor does not exist or has taught no courses. (For a simpler version of this exercise, rather than providing printed output, assume a relation with the appropriate schema and insert your answer there without worrying about testing for erroneous argument values.)

5.19 Suppose there are two relations r and s, such that the foreign key B of r references the primary key A of s. Describe how the trigger mechanism can be used to implement the **on delete cascade** option when a tuple is deleted from s.

5.20 The execution of a trigger can cause another action to be triggered. Most database systems place a limit on how deep the nesting can be. Explain why they might place such a limit.

5.21 Modify the recursive query in Figure 5.16 to define a relation

$$\textit{prereq\_depth}(\textit{course\_id}, \textit{prereq\_id}, \textit{depth})$$