Exercícios de consultas

Para cada exercício a seguir, monte uma instância contendo algumas tuplas em cada uma das relações, satisfazendo as restrições de integridade definidas, para uso no RelaX Relational Algebra Calculator. Submeta um arquivo texto contendo a definição do banco de dados (para uso no group editor), as expressões em álgebra relacional e as sentenças SQL que respondem às consultas indicadas.

Para as consultas em SQL, podem criar um banco de dados em um SGBD (sugestão: PostgreSQL) ou usar a aba SQL no RelaX. Notem que o RelaX pode ter algumas limitações no SQL, não sendo capaz de executar algumas construções que o SGBD é capaz, mas tem um funcionamento geral bom o suficiente para realizar os exercícios.

8.35. Consider the following MAILORDER relational schema describing the data for a mail order company.

Parts(Pno, Pname, Price)

Customers(Id, Name, Street, $\overline{\text{Zip}}^{\text{Zip\_Codes}}$, Phone)

Employees(Eno, Name, $\overline{\text{Zip}}^{\text{Zip\_Codes}}$, Hire_date)

ZIP_Codes(Zip, City)

Orders(Number, $\overline{\text{Cid}}^{\text{Customers}}$, $\overline{\text{Eno}}^{\text{Employees}}$, Date_Received, Date_Shipped)

Order_Details($\overline{\text{ONumber}}^{\text{Orders}}$, $\overline{\text{Pno}}^{\text{Parts}}$, Quantity)

a. Retrieve the names of parts that cost less than $20.00.
b. Retrieve the names and cities of employees who have taken orders for parts costing more than $50.00.
c. Retrieve the pairs of customer number values of customers who live in the same ZIP Code.
d. Retrieve the names of customers who have ordered parts from employees living in the city ‘Wichita’.
e. Retrieve the names of customers who have ordered parts costing less than $20.00.
f. Retrieve the names of customers who have not placed an order.
g. Retrieve the id, name and total amount spent in orders for all customers, given that the total cost of each order is the quantity multiplied by the price for every part in the order. Sort the result in descending order of total spent.

8.18. Consider the LIBRARY relational database schema in the next page, which is used to keep track of books, borrowers, and book loans. Referential integrity constraints are shown as directed arcs in the figure.

a. How many copies of the book titled ‘The Lost Tribe’ are owned by the library branch whose name is ‘Sharpstown’?
b. How many copies of the book titled ‘The Lost Tribe’ are owned by each library branch?
c. Retrieve the names of all borrowers who do not have any books checked out.
d. For each book that is loaned out from the ‘Sharpstown’ branch and whose Due_date is today, retrieve the book title, the borrower’s name, and the borrower’s address.

e. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.
f. Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.
g. For each book authored (or coauthored) by ‘Stephen King’, retrieve the title and the number of copies owned by the library branch whose name is ‘Central’.
h. Retrieve the title of the books that were co-authored by ‘James Patterson’ and ‘Mark Sullivan’ and the name of the library branches that have copies of each of these books and the number of copies the branch has.
i. Retrieve the card number, name and phone of the borrowers who have already loaned all the books authored by ‘J. K. Rowling’.
j. Retrieve the name of all borrowers who have loaned a book authored by ‘James Patterson’ but have never loaned a book authored by ‘Mark Sullivan’.