SQL Exercises

Data Science

United International College

Exercise 1

Consider the following database:

Project (pid, pname, dept_no)

Works_on (emp_id, pid, hours)

Employee (emp_id, ename, address, salary)

Department (dept_no, dname)

We are given the following three tasks:

A. For each employee working on a project with pname of ‘231Project’, retrieve the name of the employee and his/her salary. (Both relational algebra and SQL statement)

Exercise 1

Consider the following database:

Project (pid, pname, dept_no)

Works_on (emp_id, pid, hours)

Employee (emp_id, ename, address, salary)

Department (dept_no, dname)

We are given the following three tasks:

B. Retrieve the name of each employee who works on all the projects controlled by department number 5. (Both relational algebra and SQL statement)

Exercise 1

Consider the following database:

Project (pid, pname, dept_no)

Works_on (emp_id, pid, hours)

Employee (emp_id, ename, address, salary)

Department (dept_no, dname)

We are given the following three tasks:

C. For each project on which more than two employees work, retrieve the project number, the project name and the number of employees who work on the project. (SQL statement)

Exercise 2

Consider the following database:

employee (person_name, street, city)

works (person_name, company_name, salary)

company (company_name, city)

manages (person_name, manager_name)

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

- Find the names of employees who work for some company in Boston.

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

- Find the streets of employees who work for all companies in Boston.

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

- Find the names of employees who earn more than $10,000 and live in Hong Kong.

- Alternative solutions

employee (person_name, street, city)
works (person_name, company_name, alary)
company (company_name, city)
manages (person_name, manager_name)

- Find the names of the employees who are not managers.

- Alternative solutions

employee (person_name, street, city)
works (person_name, company_name, alary)
company (company_name, city)
manages (person_name, manager_name)

employee (person_name, street, city)
works (person_name, company_name, alary)
company (company_name, city)
manages (person_name, manager_name)

- Find the names of all persons who work for “First Bank Corporation” and live in the city where the company is located.

- Find the names, cities of employees who work for exactly ONE company

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

- Find the names of all employees who earn more than SOME employee of Small Bank Corporation.

- Alternative solution

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_naame, manager_name)

- Find the company located in Hong Kong that has the largest number of employees

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

- Find all companies located in Hong Kong and have total payroll less than 100,000

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

- Find the names of the employees whose salaries are higher than those of all employees living in Los Angeles. (One employee may have several works.)

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

- Find the names of the managers whose salaries are higher than that of at least one employee that they manage. (One employee may have several works.)

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

Alternative solution:

Exercise 3

Consider the following schemas.
CUST (*cust-id*, *name*), and
WITHDRAW (*w-id*, *cust-id*, *acc-id*, *date*, *amount*)

- Write an SQL query to retrieve all the names of the customers who have withdrawn more than 1k dollars in a single withdrawal. If a customer made several such withdrawals, her/his name should be reported only once.

Exercise 3

Consider the following schemas.
CUST (*cust-id*, *name*), and
WITHDRAW (*w-id*, *cust-id*, *acc-id*, *date*, *amount*)

- Sometimes there may be a “shared” account, namely, an account with multiple owners.
- Write an SQL query to return the *acc-id* of all the shared accounts.You may assume that all the owners of a shared account have made withdrawals from the account.

Exercise 3

Consider the following schemas.
CUST (*cust-id*, *name*), and
WITHDRAW (*w-id*, *cust-id*, *acc-id*, *date*, *amount*)

- We want to retrieve the *cust-id* of the customers who withdraw from the account with *acc-id* = ‘A1’ or ‘A2’ but not both.

Exercise 3

Consider the following schemas.
CUST (*cust-id*, *name*), and
WITHDRAW (*w-id*, *cust-id*, *acc-id*, *date*, *amount*)

- Retrieve the *cust-id* of the customer who withdraw the largest number of times.

Exercise 3

Consider the following schemas.
CUST (*cust-id*, *name*), and
WITHDRAW (*w-id*, *cust-id*, *acc-id*, *date*, *amount*)

- Let us use the name “interesting account” to refer to the account from which the withdrawal with smallest amount was made.
- Retrieve the *acc-id* of accounts from which withdrawals have been made, except the interesting account.

- Question: As with natural join, outer joins are not compulsory operators. That is, we can implement an outer join using "conventional" SQL.

- Let us verify this for left outer join.

- CS-PROF (*prof-id*, *name*)
- SUPERVISION (*prof-id*, *stu-id*)

- Write an alternative query that returns the same information as

- Answer:
- CS-PROF (*prof-id*, *name*)
- SUPERVISION (*prof-id*, *stu-id*)

- Question: Consider MARKS(*stu-id*, *course-id*, *score*)
- Write a query to retrieve the *stu-id* of every student who scored at least 80 in all the courses s/he took, but scored less than 90 in at least one course.

- Try to write your query with
  - 2 **select**
  - Only 1 **select**

- Answer: Consider MARKS(*stu-id*, *course-id*, *score*).
- Write a query to retrieve the *stu-id* of every student who scored at least 80 in all the courses s/he took, but scored less than 90 in at least one course.

- Answer: Consider MARKS (*stu-id*, *course-id*, *score*).
- Write a query to retrieve the *stu-id* of every student who scored at least 80 in all the courses s/he took, but scored less than 90 in at least one course.