|  |  |  |
| --- | --- | --- |
| **CIS 422 DBMS** | **Assignment 04** | **20 Points** |
| **ELIJAH CHONG TAY** |  |  |

1. Consider the following relational schema, and answer the following questions by writing relational algebra queries:

Employee (EmpID, City, Zipcode)

Works (EmpID, CorpID, Salary)

Corporation (CorpID, City)

Manages (EmpID, MngrID)

σσ

1. Find the ID of all employees who work for Amawig Corporation.

π EmpID σ (CorpID = “Amawig”) (Works)

1. Find the ID and City of all employees who work for Amawig.

π EmpID, City σ (C.CorpID = W.CorpID & C.CorpID = “Amawig” ^ W.EmpID = E.EmpID) (Employee Works Corporation)

1. Find the ID and City of all employees who work for Amawig corporation and earn more than $100,000.

π EmpID, City σ (C.CorpID = W.CorpID & C.CorpID = “Amawig” ^ W.EmpID = E.EmpID ^ W.Salary > 100000) (Employee Works Corporation)

1. Find the IDs of all employees who live in the same city as the corporation for which they work.

π EmpID s (E.City = C.City ^ E.EmpID = W.EmpID ^ C.CorpID = W.CorpID) (Employee Works Corporation)

1. Find the ID of all employees who live in the same area (zipcode) as do their managers.

π EmpID s (E.Zipcode = Manager.Zipcode ^ E.EimpID <> Manager.EmpID)

(Employee Manager(MngID 🡨 EmpID)Employee)

1. Find the IDs of all employees who earn more than every employee in the Amawig Corporation.

π EmpID (σ W.Salary > s ( MAX (Salary) s (C.CorpID=”Amawig”

(Works ⨝ Corporation))). Salary) Works

1. Suppose that the corporations may have more than a branch in different cities. Find all corporations located in every city in which Amawig Corporation is located.

π C.CorpID s(C.City=Amawig.City) C

(π City s (C.CorpID=”Amawig”. C)

1. Consider the following relational schema and answer the following questions.

User (email, name, address, householdSize)

1. Express in relational algebra the query that finds all pairs of users where the two people both claim to have a household size 2 and have the same address and returns their names and the common address.

π user1.name, user2.name, user1.address

s (user1.householdSize = 2 ^ user2.householdSize = 2 ^ user1.address = user2.address)

(user1 x user2)

1. Express the above query in SQL.

SELECT user1.name, user2.name, user1.address

FROM user1, user2

WHERE user1.householdSize = 2 AND user2.householdSize = 2 AND user1.address = user2.address

1. Write in SQL the query that finds the users whose household size is at least 50% more than the average household size and returns their name and household size, sorted by household size. (Hint: decompose the problem into sub-problems, and you may use a view to capture an intermediate result.)

SELECT name, householdSize

FROM User

WHERE householdSize >= 1.5\* (

SELECT AVG(householdSize)

FROM User))

ORDER BY householdSize

1. Write in SQL the query that finds all users each having a household size different from the total number of users having the same address as him or her. (Hint: again, decompose the problem into sub-problems, and you may use a view to capture a intermediate result.)

CREATE VIEW address\_count AS

SELECT address, COUNT\* AS count

FROM User

GROUP BY address

SELECT name, householdSize

FROM User

WHERE householdSize <> (SELECT count

FROM address\_count

WHERE address=User.address)

1. Consider the following relational schema and answer the following questions.

Employee (Employee\_Id, First\_Name, Last\_Name, Email, Phone\_Number, Hire\_Date, Job\_Id, Salary, Commission\_Pct, Manager\_Id, Department\_Id)

1. Write a query to list the number of jobs available in the employees table

SELECT COUNT(Job\_Id) AS numberOfJobs

FROM Employee

1. Write a query to get the total salaries payable to employees.

SELECT SUM(Salary) AS total\_salary

FROM Employee

1. Write a query to get the minimum salary from employees table

SELECT MIN(Salary) AS min\_salary

FROM Employee

1. Write a query to get the maximum salary of an employee working as a Programmer.

SELECT MAX (Salary) AS max\_salary

FROM Employee

WHERE Job\_Id = “Programmer”

1. Write a query to get the average salary and number of employees working the department 90.

SELECT AVG(Salary) AS avg\_salary, COUNT(\*) AS numberOfEmployees

FROM Employee

WHERE department\_id = 90

1. Write a query to get the number of employees with the same job

SELECT Job\_Id, COUNT(\*) numOfEmployees

FROM Employee

GROUP BY Job\_ID

1. Write a query to get the difference between the highest and lowest salaries.

SELECT MAX(Salary) – MIN(Salary) AS differenceOfSalaries

FROM Employee

1. Write a query to find the manager ID and the salary of the lowest-paid employee for that manager.

SELECT Manager\_Id, MIN(Salary) AS min\_salary

FROM Employee

GROUP BY Manager\_Id

1. Write a query to get the department ID and the total salary payable in each department.

SELECT Department\_Id, SUM(Salary) AS total\_salary

FROM Employee

GROUP BY Department\_Id

1. Write a query to get the total salary, maximum, minimum, average salary of employees (job ID wise), for department ID 90 only.

SELECT Job\_Id, SUM(Salary) , MAX (Salary), MIN (Salary), AVG (Salary)

FROM Employee

WHERE Department\_Id = 90

1. Write a query to get the average salary for all departments employing more than 10 employees.

SELECT department\_id, AVG(Salary)

FROM Employee

GROUP BY Department\_Id

WHERE COUNT(\*) > 10