

# Shea CS442 HW4

Ryan Shea

11/2/2022

I pledge my honor that I have abided by the Stevens Honor System.

- Emp(eid, ename, age, salary)
- Works(eid, did)
- Dept(did, budget, managerid, dname)

The key fields are underlined.

The three tables have the following reference relationships and cardinality constraints:

- The managerid value of the managers in Dept table refers to the eid value in Emp table.
- A manager may control multiple departments, and an employee can work in multiple departments. Thus the Works relation associates the employees with their departments.

Write the following queries in SQL.

- (1) Find the name of the employees who are older than 40 and are working in Software department [10pts];

```
SELECT E.ename
FROM Emp E NATURAL JOIN Works NATURAL JOIN Dept D
WHERE E.age > 40
AND D.dname = 'Software';
```

- (2) Find the name of the employee(s) in Hardware department that has the highest salary [10pts];

```
SELECT E.ename
FROM Emp E NATURAL JOIN Works W NATURAL JOIN Dept D
WHERE D.dname = 'Hardware'
EXCEPT
SELECT E.ename
FROM Emp E NATURAL JOIN Works W NATURAL JOIN Dept D
GROUP BY E.ename
HAVING D.dname = 'Hardware'
AND E.salary = MAX(E.salary);
```

This will return just the name. If you want the salary as well, you can do this:

```
SELECT E.ename, MAX(E.Salary)
FROM Emp E NATURAL JOIN Works W NATURAL JOIN Dept D
WHERE D.dname = 'Hardware';
```

- (3) Find the salary of the oldest employee in Hardware department [10pts];

```
SELECT E.salary, MAX(E.age)
FROM Emp E NATURAL JOIN Works W NATURAL JOIN Dept D
WHERE D.dname = 'Hardware';
```

The same principals from Question (2) apply here as well.

```

SELECT E.salary
FROM Emp E NATURAL JOIN Works W NATURAL JOIN Dept D
WHERE D.dname = 'Hardware'
EXCEPT
SELECT E.salary
FROM Emp E NATURAL JOIN Works W NATURAL JOIN Dept D
GROUP BY E.ename
HAVING D.dname = 'Hardware'
AND E.age = MAX(E.age);

```

(4) Find the name of all employees who work in both Hardware and Software departments [10pts];

```

SELECT E.ename
FROM Emp E
WHERE EXISTS (SELECT *
              FROM Works W NATURAL JOIN Dept D
              WHERE W.eid = E.eid
              AND D.dname = 'Hardware')
AND EXISTS (
  SELECT *
  FROM Works W NATURAL JOIN Dept D
  WHERE W.eid = E.eid
  AND D.dname = 'Software');

```

(5) Find the name of the employees who work in more than 3 departments [10pts].

```

SELECT E.ename
FROM Emp E NATURAL JOIN Works W
GROUP BY E.eid
HAVING COUNT(DISTINCT W.did) > 3;

```

(6) Find the name of the employees (managers) who manages more than one department; [10pts];

```

SELECT ename
FROM Dept NATURAL JOIN WORKS NATURAL JOIN Emp
WHERE eid = managerid
AND dname NOT IN(
  SELECT D.dname
  FROM Emp E NATURAL JOIN Works W NATURAL JOIN Dept D
  GROUP BY D.managerid);

```

(7) Find the ID of managers who control the largest total amounts of budget [10pts];

Note: As a manager can manage multiple departments, the budget that he/she can control should be the total amounts of budget of all the departments that he/she is the manager.

```

SELECT MAX(b), managerid
FROM (
  SELECT SUM(budget) b, managerid
  FROM Dept
  GROUP BY managerid);

```

- (8) Find the ID of the managers who manages at least one department of budget over 1 million dollars [10pts];

```
SELECT managerid
FROM Dept
GROUP BY did
HAVING SUM(budget) > 1000000;
```

- (9) Find the ID of the managers who only manages department(s) of budget over 1 million dollars [10pts] (i.e., for all the departments of the returned managers, each department has the budget over 1 million dollars);

```
SELECT managerid
FROM Dept
WHERE managerid NOT IN (SELECT managerid
                        FROM Dept
                        GROUP BY did
                        HAVING budget <= 1000000);
```

- (10) For each department, return the average salary of all its employees [10pts].

```
SELECT AVG(E.salary) average_salary, D.dname -- or D.did
FROM Emp E NATURAL JOIN Works W NATURAL JOIN Dept D
GROUP BY D.did;
```