

## SQL Practice Questions Single Table

Relational database schema:

emp (eno, ename, bdate, title, salary, supereno, dno)

proj (pno, pname, budget, dno)

dept (dno, dname, mgreno)

workson (eno, pno, resp, hours)

- 1) Return the project names that have a budget > 250000.
- 2) Return the employee numbers who make less than \$30000.
- 3) Return the list of workson responsibilities (resp) with no duplicates.
- 4) Return the employee (names) born after July 1, 1970 that have a salary > 35000 and have a title of 'SA' or 'PR'.  
◆ Write the equivalent relational algebra expression.

Page 10

- 1) SELECT pname FROM Proj WHERE budget > 250000;
- 2) SELECT eno FROM Emp where salary < 30000;
- 3) SELECT DISTINCT resp from WorksOn;
- 4) SELECT ename FROM Emp WHERE bdate > DATE '1970-07-01' and salary > 35000 and (title = 'SA' or title = 'PR'); (parenthesis needed)

$\pi_{ename}(\sigma_{bdate > '07-01-70' \text{ and } salary > 35000 \text{ and } (title='SA' \text{ or } title='PR')}(emp))$

Note that parenthesis are needed for answer to be correct.

**Note that it works well to answer questions on computer instead of on the board.**

## SQL Practice Questions

### Joins

Relational database schema:

emp (eno, ename, bdate, title, salary, supereno, dno)

proj (pno, pname, budget, dno)

dept (dno, dname, mgreno)

workson (eno, pno, resp, hours)

1) For each employee, return their name and their department name.

2) Return the list of project names for the department with name 'Consulting'.

3) Return workson records (eno, pno, resp, hours) where project budget is > \$50000 and hours worked is < 20.

4) Return a list of all department names, the names of the projects of that department, and the name of the manager of each department.

Page 14

- 1) SELECT ename, dname FROM Emp, Dept WHERE Emp.dno = Dept.dno
- 2) SELECT pname FROM Dept, Proj WHERE Dept.dno = Proj.dno AND dname = 'Consulting'
- 3) SELECT eno, WorksOn.pno, resp, hours FROM WorksOn, Proj WHERE WorksOn.pno = Proj.pno AND budget > 50000 and hours < 20
- 4) SELECT dname, pname, ename FROM Dept, Proj, Emp WHERE dept.dno = proj.dno and mgreno = eno;

## **SQL Practice Questions**

### **Expressions, *LIKE*, *IS NULL***

Relational database schema:

emp (eno, ename, bdate, title, salary, supereno, dno)

proj (pno, pname, budget, dno)

dept (dno, dname, mgreno)

workson (eno, pno, resp, hours)

- 1) Calculate the monthly salary for each employee.
- 2) List all employee names who do not have a supervisor.
- 3) List all employee names where the employee's name contains an 'S' and workson responsibility that ends in 'ER'.
- 4) Return the list of employees (names) who make less than their managers and how much less they make.

Page 27

1) SELECT ename, salary/12 FROM Emp;

2) SELECT ename FROM Emp WHERE supereno IS NULL;

3) SELECT ename FROM Emp E, WorksOn W WHERE ename LIKE '%S%' and E.eno = W.eno and resp LIKE '%ER';

4) SELECT E.ename, M.salary - E.salary FROM Emp as E, Emp as M WHERE E.supereno = M.eno and E.salary < M.salary;

## SQL Practice Questions Set Operations, ORDER BY

Relational database schema:

emp (eno, ename, bdate, title, salary, supereno, dno)

proj (pno, pname, budget, dno)

dept (dno, dname, mgreno)

workson (eno, pno, resp, hours)

1) Return the list of employees sorted by salary (desc) and then title (asc).

2) Return the employees (names) who either manage a department or manage another employee.

3) Return the employees (names) who manage an employee but do not manage a department.

4) Give a list of all employees who work on a project for the 'Management' department ordered by project number (asc).

5) **Challenge:** Return the projects (names) that have their department manager working on them.

Page 28

1) SELECT \* FROM Emp ORDER BY salary DESC, title ASC

2) (SELECT ename FROM Emp, Dept WHERE mgreno = eno)

UNION (SELECT M.ename FROM Emp E, Emp M WHERE E.supereno = M.eno);

OR:

SELECT DISTINCT M.ename From Emp M, Emp E, Dept D

WHERE M.eno = E.supereno OR M.eno = D.mgreno

3) (SELECT M.ename FROM Emp E, Emp M WHERE E.supereno = M.eno)

EXCEPT (SELECT ename FROM Emp, Dept WHERE mgreno = eno);

4) SELECT proj.pno, ename FROM Emp E, WorksOn W, Proj P, Dept D  
WHERE E.eno = W.eno and W.pno = P.pno and P.dno = D.dno and D.dname =  
'Management' ORDER BY P.pno;

5) SELECT pname FROM WorksOn W, Dept D, Proj P WHERE W.eno = D.mgreno and P.pno = W.pno and P.dno = D.dno;

More on INTERSECT and EXCEPT:

For instance, to get the *intersection* of R(a,b) and S(a,b), write:

```
SELECT DISTINCT *  
FROM R  
WHERE EXISTS (SELECT * FROM S WHERE R.a = S.a AND R.b = S.b);
```

To get the *set difference*, here is a similar approach using a sub query:

```
SELECT DISTINCT *  
FROM R  
WHERE NOT EXISTS (SELECT * FROM S WHERE R.a = S.a AND R.b = S.b);
```

**Note that MySQL does not support INTERSECT or EXCEPT. You need to use subqueries and EXISTS/NOT EXISTS to have this functionality.**

## ***GROUP BY Practice Questions***

Relational database schema:

emp (eno, ename, bdate, title, salary, supereno, dno)

proj (pno, pname, budget, dno)

dept (dno, dname, mgreno)

workson (eno, pno, resp, hours)

- 1) Return the highest salary of any employee.
- 2) Return the smallest project budget.
- 3) Return the department number and average budget for its projects.
- 4) For each project, return its name and the total number of hours employees have worked on it.
- 5) For each employee, return the total number of hours they have worked. Only show employees with more than 30 hours.

Page 42

- 1) SELECT MAX(salary) FROM Emp;
- 2) SELECT MIN(budget) FROM Proj;
- 3) SELECT dno, AVG(budget) FROM Proj GROUP BY dno;
- 4) SELECT pname, SUM(hours) FROM Proj P, WorksOn W  
WHERE W.pno = P.pno GROUP BY pname;
- 5) SELECT ename, SUM(hours) FROM Emp E, WorksOn W  
WHERE W.eno = E.eno GROUP BY ename HAVING SUM(hours) > 30;

**Challenge:** Calculate the average # of hours spent per project in each department.

Interpretation #1: Avg. hours an employee works on a project in a department

```
SELECT dname, AVG(hours) FROM WorksOn W, Proj P, Dept D
WHERE W.pno = P.pno and P.dno = D.dno
GROUP BY dname;
```

Interpretation #2: (Avg. hours per project)

```
SELECT dname, P.pno, AVG(hours) FROM WorksOn W, Proj P, Dept D
WHERE W.pno = P.pno and P.dno = D.dno
GROUP BY dname, P.pno;
```

Interpretation #3: Avg. hours per project in a department

```
SELECT dno, SUM(hours)/COUNT(DISTINCT P.pno)
FROM WorksOn W, Proj P
WHERE W.pno = P.pno
GROUP BY dno;
```

Note that DISTINCT does not work in Access in this query this way. Is it correct in general?

Interpretation #3 is different than #1 because calculates average hours a project is worked on by all employees in a department. #1 is avg. hours an employee works on a project in a department. Example:

| eno | pno | hours |
|-----|-----|-------|
| E1  | P1  | 20    |
| E2  | P1  | 30    |
| E1  | P2  | 40    |

Interpretation #1: avg. hours worked on per project in a dept = 30

Interpretation #3: avg. hours spent by all employees on a project = 45 (50 for first project, and 40 for second)

## Subquery Practice Questions

Relational database schema:

emp (eno, ename, bdate, title, salary, supereno, dno)

proj (pno, pname, budget, dno)

dept (dno, dname, mgreno)

workson (eno, pno, resp, hours)

- 1) List all departments that have at least one project.
- 2) List the employees who are not working on any project.
- 3) List the employees with title 'EE' that make more than all employees with title 'PR'.
- 4) Find all employees who work on some project that 'J. Doe' works on.

Page 57

1)

```
SELECT dname FROM Dept
WHERE dno IN (SELECT dno FROM Proj);
```

2)

```
SELECT ename
FROM Emp
WHERE eno NOT IN (SELECT eno FROM WorksOn)
```

Or correlated but not as efficient:

```
SELECT ename FROM Emp
WHERE NOT EXISTS(SELECT * FROM WorksOn WHERE Emp.eno =
WorksOn.eno);
```

3) SELECT ename FROM Emp

```
WHERE title = 'EE' and salary > ALL (SELECT salary FROM Emp WHERE
title = 'PR');
```



OR:

```
SELECT ename FROM Emp
```

```
WHERE title = 'EE' and salary > (SELECT MAX(salary) FROM Emp WHERE title = 'PR');
```

Note: SUM(salary) will work for this data set as only one record with title = 'PR' but not in general.

OR: (unique solution using HAVING)

```
SELECT eno, ename
```

```
FROM Emp
```

```
WHERE title = 'EE'
```

```
GROUP BY eno, ename
```

```
HAVING AVG(salary) > ALL (SELECT salary FROM Emp WHERE title = 'PR')
```

4)

```
SELECT ename FROM Emp WHERE eno IN
```

```
    (SELECT eno FROM WorksOn WHERE pno IN
```

```
        (SELECT pno FROM WorksOn WHERE eno =
```

```
            (SELECT eno FROM Emp WHERE ename
```

```
= 'J. Doe')));
```