



Week #1 Homework

Using SQL, solve the following queries based on the projects, employees, workson, and rate tables discussed in the class (shown below). Note that the tables contain sample records only and do not represent all the possible records that could be stored in these tables.

projects

PROJECT_NUMBER	PROJECT_NAME	PROJECT_CITY
1	Eagle	NY
2	Super Jet	LA

workon

PROJECT_NUMBER	EMP_ID
1	10
1	11
2	10
2	20
2	11

employees

EMP_ID	EMP_NAME	RATE_CATEGORY	EMP_CITY
10	Smith	B	NY
11	eSmith	C	SF
20	Smithe	C	LA
15	eSmithe	D	SD

rate

RATE_CATEGORY	RATE
A	100
B	80
C	60
D	50

1. Find the names of employees who work on the Eagle project and has a rate greater than 70.

```
SELECT emp_name
FROM projects
JOIN workon ON projects.project_number=workon.project_number
JOIN employees ON workon.emp_id=employees.emp_id
JOIN rate ON employees.rate_category=rate.rate_category
```



WHERE projects.project_name='Eagle' AND rate.rate>70;

Code/Output:

```
119 SELECT emp_name
120 FROM projects
121 JOIN workon ON projects.project_number=workon.project_number
122 JOIN employees ON workon.emp_id=employees.emp_id
123 JOIN rate ON employees.rate_category=rate.rate_category
124 WHERE projects.project_name='Eagle' AND rate.rate>70;
125
```

EMP_NAME
Smith

[Download CSV](#)

2. Find the names of the projects that have employees with rates greater or equal to 80.

```
SELECT project_name, rate
FROM projects p, rate r
WHERE r.rate>=80;
```

Code/Output:

```
127 SELECT project_name, rate
128 FROM projects p, rate r
129 WHERE r.rate>=80;
130
```

PROJECT_NAME	RATE
Eagle	100
Super Jet	100
Eagle	80
Super Jet	80

[Download CSV](#)
4 rows selected.

3. Find the names of the employees who work on projects that are located in the same city where the employees are located.

```
SELECT emp_name
FROM employees e, projects p
WHERE p.project_city=e.emp_city;
```



Code/Output:

```
131 SELECT emp_name
132 FROM employees e, projects p
133 WHERE p.project_city=e.emp_city;
134
135
```

EMP_NAME
Smith
Smithe

[Download CSV](#)
2 rows selected.

4. Find the names of employees who are not working on any projects.

```
SELECT emp_name
FROM employees
WHERE emp_id NOT IN (SELECT DISTINCT emp_id FROM workon);
```

Code/Output:

```
135 SELECT emp_name
136 FROM employees
137 WHERE emp_id NOT IN (SELECT DISTINCT emp_id FROM workon);
138
139
```

EMP_NAME
eSmithe

[Download CSV](#)

5. Find the average rate of the employees who work on project Eagle.

```
SELECT AVG(rate)
FROM projects
JOIN workon ON projects.project_number=workon.project_number
JOIN employees ON workon.emp_id=employees.emp_id
JOIN rate ON employees.rate_category=rate.rate_category
WHERE projects.project_name='Eagle';
```



Code/Output:

```
140 SELECT AVG(rate)
141 FROM projects
142 JOIN workon ON projects.project_number=workon.project_number
143 JOIN employees ON workon.emp_id=employees.emp_id
144 JOIN rate ON employees.rate_category=rate.rate_category
145 WHERE projects.project_name='Eagle';
146
147
```

AVG(RATE)
70

[Download CSV](#)

6. Find the names of the employees who work on more than one project. Display the results in descending order.

```
SELECT DISTINCT emp_name
FROM employees E, workon W1, workon W2
WHERE W1.emp_id = E.emp_id
AND W2.emp_id=E.emp_id
AND W1.project_number != W2.project_number
ORDER BY emp_name DESC;
```

Code/Output:

```
148 SELECT DISTINCT emp_name
149 FROM employees E, workon W1, workon W2
150 WHERE W1.emp_id = E.emp_id
151 AND W2.emp_id=E.emp_id
152 AND W1.project_number != W2.project_number
153 ORDER BY emp_name DESC;
154
155
```

EMP_NAME
eSmith
Smith

[Download CSV](#)

2 rows selected.