Week #2 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	В	NY
11	eSmith	С	SF
20	Smithe	С	LA
15	eSmithe	D	SD

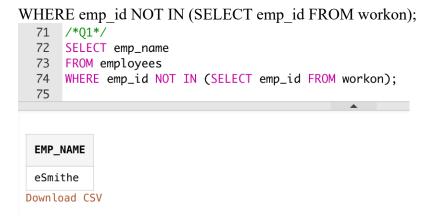
rate

RATE_CATEGORY	RATE
Α	100
В	80
С	60
D	50

1. Use non-correlated sub-query, find the names of employees who are not working on any projects.

SELECT emp_name FROM employees





2. Use correlated sub-query, find the names of employees who are not working on any projects.

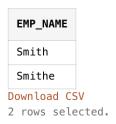
```
SELECT emp name
FROM employees e
WHERE NOT EXISTS
  (SELECT emp id FROM workon w
  WHERE w.emp id=e.emp id);
 76 /*Q2*/
    SELECT emp_name
 77
 78 FROM employees e
 79 WHERE NOT EXISTS
 80
         (SELECT emp_id FROM workon w
 81
         WHERE w.emp_id=e.emp_id);
 82
 EMP_NAME
 eSmithe
Download CSV
```

3. Use non-correlated sub-query, 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
WHERE e.emp_city IN
(SELECT project_city
FROM projects);
```



```
83 /*Q3*/
84 SELECT emp_name
85 FROM employees e
86 WHERE e.emp_city IN
87 (SELECT project_city
88 FROM projects);
```



4. Use correlated sub-query, 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
WHERE EXISTS
  (SELECT project city
  FROM projects p
  WHERE p.project city=e.emp city);
  90 /* Q4*/
  91 SELECT emp_name
  92 FROM employees e
  93 WHERE EXISTS
  94
          (SELECT project_city
  95
          FROM projects p
  96
          WHERE p.project_city=e.emp_city);
  97
  EMP_NAME
```

5. Use sub-query, find the names of the employees with the highest rate.

```
SELECT emp_name
FROM employees
WHERE rate_category IN
```

Smith

Smithe

Download CSV
2 rows selected.



```
(SELECT MIN(rate_category)
FROM employees);

98  /* Q5 */
99  SELECT emp_name
100  FROM employees
101  WHERE rate_category IN
102  (SELECT MIN(rate_category)
103  FROM employees);

EMP_NAME
Smith
Download CSV
```

6. Use sub-query and the ALL operator, find the names of the employees with the highest rate.

```
select emp name
from employees
where rate_category=ALL
  (select MIN(rate category)
  from employees);
  105 /* Q6 */
       select emp_name
  106
  107
       from employees
  108
       where rate_category=ALL
  109
            (select MIN(rate_category)
            from employees);
  110
  111
  117
```



7. Use inline views and sub-query, find the names of employees with the highest rate.

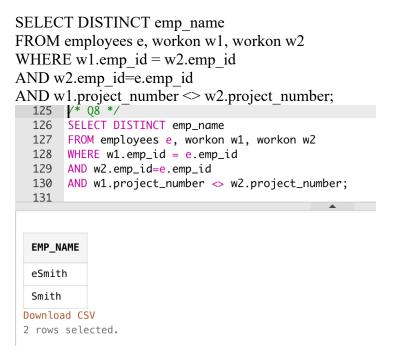
```
SELECT *
FROM

(
SELECT emp_name
FROM employees
ORDER BY rate category
```



```
)
WHERE
  ROWNUM = 1;
 113 /* Q7 */
       SELECT *
 114
 115 FROM
 116
 117
              SELECT emp_name
 118
              FROM employees
 119
              ORDER BY rate_category
 120
      WHERE
 121
 122
          ROWNUM = 1;
 EMP_NAME
 Smith
```

8. Use self-join, find the names of the employees who work on more than one project.



9. Use non-correlated sub-query, find the names of the employees who work on more than one project.

SELECT emp name

Download CSV

```
FROM employees
WHERE emp id IN
  (SELECT emp id
 FROM workon
  GROUP BY emp id
  HAVING COUNT(emp_id)>1);
  132 /* Q9 */
  133
      SELECT emp_name
  134
       FROM employees
  135
       WHERE emp_id IN
  136
           (SELECT emp_id
  137
           FROM workon
  138
           GROUP BY emp_id
  139
           HAVING COUNT(emp_id)>1);
  140
```

```
EMP_NAME

Smith

eSmith

Download CSV
2 rows selected.
```

10. Use correlated sub-query, find the names of the employees who work on more than one project.

```
SELECT emp_name
FROM employees e
WHERE EXISTS
(SELECT emp_id
FROM workon w
WHERE w.emp_id=e.emp_id
GROUP BY emp_id
HAVING COUNT(emp_id)>1);
```



National University School of Engineering, Technology, & Media

ANA 660 Advanced SQL

```
141 /* Q10*/
142 SELECT emp_name
143 FROM employees e
144 WHERE EXISTS
145 (SELECT emp_id
146 FROM workon w
147 WHERE w.emp_id=e.emp_id
148 GROUP BY emp_id
149 HAVING COUNT(emp_id)>1);
```

EMP_NAME

Smith

eSmith

Download CSV

2 rows selected.