HW3

Andrew Plaza

McCloskey

Computer Science 340

September 28, 2017

SQL Syntax Highlighting only used because it's the most similar syntax highlighting in this particular editing program. All queries should be considered to be SRL.

Screenshots of Vim Editor are sometimes used in order for better spacing and indentation.

1.) Report the last name and department # of each employee whose first name is "Gourdhead" and whose salary is at least $70,000.

```
PROJECT Lname, Dno
FROM (RESTRICT Employee
     WHERE Fname = 'Gourdhead' AND (SALARY >= 70000));
```

2.) Report the last name and birthdate of each male employee whose middle initial is 'T' and who has a female dependent named 'Rumplestiltskin'.

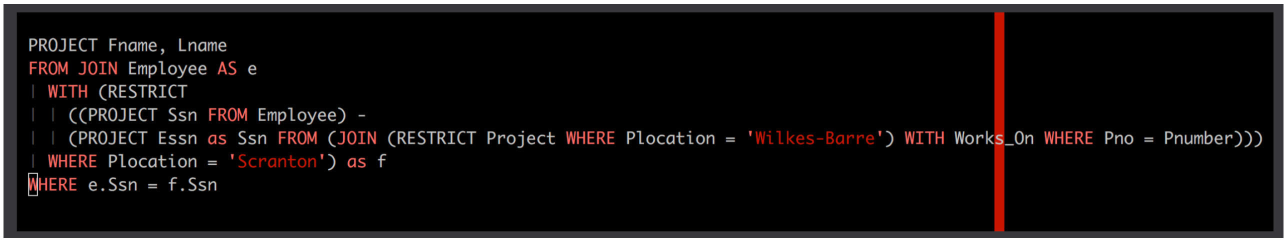
```
PROJECT Lname, Bdate
FROM JOIN (RESTRICT Employee WHERE (Minit = 'T' AND Sex = 'M')
     WITH (RESTRICT Dependent WHERE (Dependent_name = 'Rumplestiltskin' AND Sex = 'F'))
     WHERE Ssn = Essn;
```

3.) Report the first and last names of each employee who works on a project located in either Scranton or Wilkes-Barre.

```
PROJECT Fname, Lname
FROM JOIN Employee WITH (RESTRICT (JOIN Works_On WITH Project WHERE Pno = Pnumber) WHERE Plocation = 'Wilkes-Barre' OR Plocation = 'Scranton')
WHERE Ssn = Essn;
```

4.) Report the first and last names of each employee who works on a project located in Scranton but does not work on any project located in Wilkes-Barre.

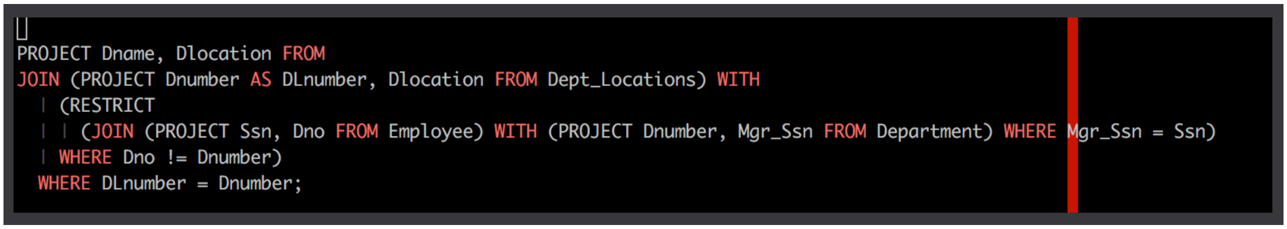
```
PROJECT Fname, Lname
FROM JOIN Employee AS e
    WITH (RESTRICT
        ((PROJECT Ssn FROM Employee) -
        (PROJECT Essn AS Ssn FROM (JOIN (RESTRICT Project WHERE Plocation = 'Wilkes-Barre') WITH Works_On WHERE Pno = Pnumber))) WHERE Plocation = 'Scranton') AS f
        WHERE e.Ssn = f.Ssn

```



```
PROJECT Fname, Lname
FROM JOIN Employee AS e
  WITH (RESTRICT
    ((PROJECT Ssn FROM Employee) -
    (PROJECT Essn as Ssn FROM (JOIN (RESTRICT Project WHERE Plocation = 'Wilkes-Barre') WITH Works_On WHERE Pno = Pnumber)))
  WHERE Plocation = 'Scranton') as f
WHERE e.Ssn = f.Ssn
```

5.) Report the name and location of each department whose manager works for a different department. (The department for which an employee works is identified by the Dno attribute in the tuple describing the employee.)

```
PROJECT Dname, Dlocation FROM
JOIN (PROJECT Dnumber AS DLnumber, Dlocation FROM Dept_Locations) WITH
    (RESTRICT
        (JOIN (PROJECT Ssn, Dno FROM Employee) WITH (PROJECT Dnumber, Mgr_Ssn FROM Department) WHERE Mgr_Ssn = Ssn)
        WHERE Dno != Dnumber)
    WHERE DLnumber = Dnumber;
```



```
PROJECT Dname, Dlocation FROM
JOIN (PROJECT Dnumber AS DLnumber, Dlocation FROM Dept_Locations) WITH
  | (RESTRICT
  | | (JOIN (PROJECT Ssn, Dno FROM Employee) WITH (PROJECT Dnumber, Mgr_Ssn FROM Department) WHERE Mgr_Ssn = Ssn)
  | WHERE Dno != Dnumber)
 WHERE DLnumber = Dnumber;
```

6.) Report, for each project, its name and number the number of employees who work on it and the sum of the hours that they work on it. (Try to formulate your query so that the result explicitly identifies projects on which no employee is working. Assume that the aggregate `SUM` function treats NULL as though it were zero.)

```
/* The number of employees who work on a project */
PROJECT Pname, Pnumber, Pcount, Sum_Hours FROM
AGGREGATE COUNT(*) AS PCount, Sum_Hours
    FROM JOIN Employee AS e
    WITH (AGGREGATE SUM(Hours) AS Sum_Hours FROM JOIN Works_On WITH Project WHERE Pno = Pnumber) AS p
    WHERE e.Ssn = p.Essn;
```

7.) Report for each project and each of the two sexes, the number of the project, how many employees of that sex work on that project, and the total number of hours that those employees devote to that project. (The resulting table should have attributes for project number, sex, # of employees, and total hours.)

Project Number	SEX	# of Employees	Total Hour
X	X	X	X

Multiple Queries for readability

```
/* One Table for Employee Info plus Works_On Info */
ProjectInfo: JOIN Employee WITH (JOIN Works_On WITH Project WHERE Pno = Pnumber) WHERE Ssn = Essn;

/* Aggregate the info we need to create the table */
AGGREGATE Pno, Sex, COUNT(*), SUM(Hours)
FROM ProjectInfo
GROUP BY Pno, Sex
```

Bonus: As above, but also include the project name in each tuple and try to formulate your query so that the result explicitly identifies project/sex combinations for which the # of employees is zero. Assume that the aggregate `SUM` function treats NULL as though it were zero.)