**IS 441 SQL Demo 2019 0424 Self JOIN and Correlated Subquery**

1. Self Join

Task 1: Find employees who were hired before their managers, list the names of both

Comment:

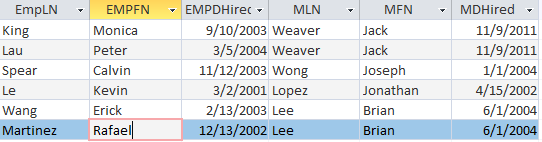
1. We’ll use the same table Employee as two tables EMP and MGR;
2. Since the fields are of the same name in both tables, we must indicate the table source of the fields

SELECT EMP.[Last Name] AS EmpLN, EMP.[First Name] AS EMPFN, EMP.DateHired AS EMPDHired, MGR.[Last Name] AS MLN, MGR.[First Name] AS MFN, MGR.DateHired AS MDHired

FROM Employee EMP, Employee MGR

WHERE EMP.[Manager ID] = MGR.[Employee ID]

AND EMP.DateHired<MGR.DateHired



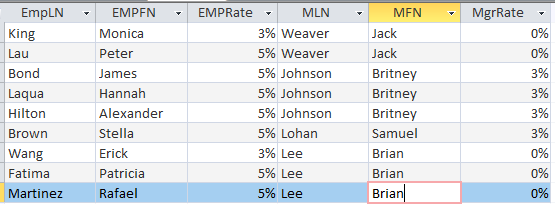
Task 2: List employees who had higher merit (raise %) than their manager’s, list BOTH set of data.

SELECT EMP.[Last Name] AS EmpLN, EMP.[First Name] AS EMPFN, EMP.[Merit Rate] AS EMPRate, MGR.[Last Name] AS MLN, MGR.[First Name] AS MFN, MGR.[Merit Rate] AS MgrRate

FROM Employee EMP, Employee MGR

WHERE EMP.[Manager ID] = MGR.[Employee ID]

AND MGR.[Merit Rate] < EMP.[Merit Rate]



1. Correlated Subquery

Task 1: List restaurants whose sales are higher than the average sales of its own city.

Comment:

1. To accomplish this task, SQL actually go thru each and every row (each restaurant),
2. For each restaurant (each row), the city of THAT row would be identified,
3. Then the AVG sales of THAT city would be calculated in the subquery, and
4. THE AVG would then be returned to the outer query to be compared with the sales of row in “1”
5. – the subquery needs a piece of critical info from the outer query, which is (in this case) the City from the outer query. We call that a “parameter” passed into the subquery from the outer query.
6. To play this “little trick” of passing a parameter from outer query into the subquery (inner query), we resort to table alias, i.e.,
   1. Although the outer query and inner query can have different tables, we specifically force the parameter into inner query using the table alias of the outer query [WATCH!!!]
   2. The inner query would be operating on the same table as the outer query (of course), but while, in the example of the Restaurants DB, the inner query is based on the Restaurants table, when it needs to retrieve rows of a SPECIFIC city, THAT city is passed from the outer table.

SELECT RestaurantID, AnnualSales, City,

(SELECT AVG(AnnualSales) FROM Restaurants

GROUP BY City

HAVING City = R\_Out.City) AS CityAVG

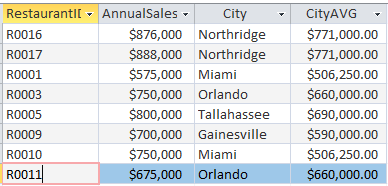
FROM Restaurants R\_out

WHERE AnnualSales >

(SELECT AVG(AnnualSales) FROM Restaurants

GROUP BY City

HAVING City = R\_Out.City)



Comment: We do need to pass the parameter (City here) to force the subquery to return **only** the AVG **of THE city** that the current row of the outer query is being examined.