SQL Demo: Self Join and Correlated Subquery (Passing of Parameters into Subquery)

**Self Join:**

List the employees (together with their date-hired) who were hired before their supervisor.

|  |  |
| --- | --- |
| SELECT E.Name, E.DateHired, S.Name, S.DateHired  FROM [Loan Officers] E, [Loan Officers] S  WHERE E.Supervisor=S.OfficerID  AND E.DateHired < S.DateHired; |  |

List the employees (together with their salary) who have salaries higher than their supervisor.

|  |  |
| --- | --- |
| SELECT E.Name, E.Salary, S.Name, S.Salary  FROM [Loan Officers] E, [Loan Officers] S  WHERE E.Supervisor = S.OfficerID  AND E.Salary > S.Salary; |  |

**Passing parameter (Correlated subquery):**

List the loan officers who has a salary higher than the average salary of his/her branch.

We want to find out individual officers (so we go down the list of rows one by one),

To find those who has salary higher than AVG of his/her branch

– while we go down the list of the officers, we need to watch and pass his/her branch into the subquery,

Then we calculate the avg of THAT branch,

Then we return THE avg to the outer query for the comparison:

Individual salary > AVG salary of THE branch?

|  |  |
| --- | --- |
|  | SELECT Name, Salary FROM [Loan Officers] L\_Outside  WHERE Salary >  (SELECT AVG(Salary) FROM [Loan Officers]  GROUP BY Branchlocation  HAVING Branchlocation = L\_Outside. Branchlocation) |

Now we also want to display that branch average for a check:

|  |  |
| --- | --- |
| SELECT Name, Salary,  (SELECT AVG(Salary) FROM [Loan Officers]  GROUP BY Branchlocation  HAVING Branchlocation = L\_Outside.Branchlocation)  AS BranchAVG  FROM [Loan Officers] L\_Outside  WHERE Salary >  (SELECT AVG(Salary) FROM [Loan Officers]  GROUP BY Branchlocation  HAVING Branchlocation = L\_Outside.Branchlocation) |  |