Derived Tables Lab Problems

1) Will the following query execute properly? Why or why not?

```
SELECT *
FROM (
       SELECT P.BusinessEntityID, P.FirstName, P.LastName, YEAR(E.HireDate)
       FROM HumanResources.Employee E
       INNER JOIN Person.Person P
       ON P.BusinessEntityID = E.BusinessEntityID
       ) AS EmployeeDetails
```

2) Will the following query execute properly? Why or why not?

```
SELECT *
FROM (
       SELECT TOP 100 P.BusinessEntityID, P.FirstName, P.LastName
       FROM HumanResources.Employee E
       INNER JOIN Person.Person P
       ON P.BusinessEntityID = E.BusinessEntityID
       ORDER BY 1
       ) AS EmployeeDetails
```

3) Using a derived table so that no functions will appear in the WHERE clause, find all employees from the HumanResources.Employee table who were hired in the year 2006 or greater and who were born in the year 1968 or less.
4) Using a derived table, find the total sales revenue generated for the year 2005 and 2006. The WHERE and GROUP BY clause of your SELECT statement should have no functions.

Using Derived Tables Lab Answers

Question 1:
The query will not execute because a required condition of a table expression is that each column must have a name specified. The "YEAR(HireDate)" column does not have an alias specified, therefore the table expression cannot be evaluated and the query will fail.

Question 2:
This query will execute without an issue. ORDER BY clauses are not allowed in table expressions **unless** the TOP operator is used within the expression. Since the TOP operator is included in this table expression the ORDER BY clause is allowed.

Question 3:
```
SELECT *
FROM (
       SELECT *, YEAR(HireDate) AS HireYear, YEAR(BirthDate) AS BirthYear
       FROM HumanResources.Employee
       ) AS Emp
WHERE HireYear >= 2006 AND BirthYear <= 1968
```

Question 4:
```
SELECT
       OrderYear,
       SUM(TotalDue) AS TotalSales
FROM (
       SELECT *, YEAR(OrderDate) AS OrderYear
       FROM Sales.SalesOrderHeader
       ) AS SalesOrders
WHERE OrderYear IN (2005, 2006)
GROUP BY OrderYear
```