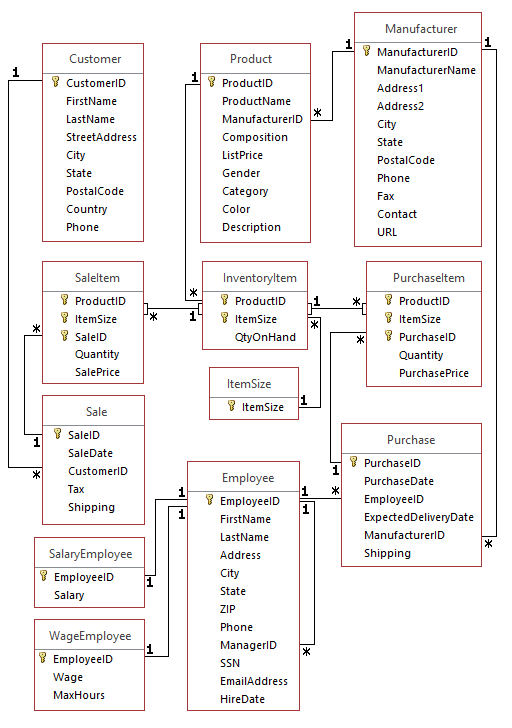
[MyEducator Home Page](https://www.myeducator.com/)

* [BMIS 441 02: Database Manag...](https://app.myeducator.com/course/activity/617eba/2701101308412010497/#)
* [Contents](https://app.myeducator.com/course/activity/617eba/2701101308412010497/#)
* Course
* Nathan Yoon

# **BMIS 441 02: Database Management [Fall 2021]**

## **SQL Assignment #10-Sub Queries**

For each information request below, formulate a single SQL query to produce the required information. 

| Due Date: | Nov 02 at 23:59 |
| --- | --- |
| Points: | 100 |

Graded on Nov 07 at 16:38

Your Submission:

| Submission Score: | 100 / 100 (100.00%) |
| --- | --- |
| Grade Time: | Nov 07 at 16:38 |
| Submitted On: | Nov 02 at 18:02 |

1. List the DISTINCT names of manufacturers of black sneakers that cost over $70. Use a subquery in the Where clause.
2. SELECT DISTINCT M.ManufacturerName
3. FROM Manufacturer M
4. WHERE M.ManufacturerID IN(
5. SELECT DISTINCT P.ManufacturerID
6. FROM Product P
7. WHERE P.Category = 'sneakers'
8. AND P.Color = 'Black'
9. AND P.ListPrice > 70);

**FEEDBACK**Ungraded / 10

1. List the name and list price of products that were sold on a sale for which the shipping cost was over $60. (use two levels of subqueries in where clauses.)
2. SELECT P.ProductName, P.ListPrice
3. FROM Product P
4. WHERE ProductID IN(
5. SELECT ProductID
6. FROM SaleItem
7. WHERE SaleID IN(
8. SELECT SaleID
9. FROM Sale
10. WHERE Shipping > 60));

**FEEDBACK**Ungraded / 10

1. List Employees (first and last names) and their hire dates of employees who were hired before their managers. (whose hire date is earlier than their managers hire date). Use a subquery in the where clause.
2. SELECT E.FirstName AS 'EmployeeFirstName', E.LastName AS 'EmployeeLastName', E.HireDate
3. FROM Employee E
4. WHERE E.HireDate < (
5. SELECT M.HireDate
6. FROM Employee M
7. WHERE E.ManagerID = M.EmployeeID);
8. -- M is a new "manager table"

**FEEDBACK**Ungraded / 10

1. List the name and list price of products that sold in quantity of 3. Use a subquery in the where clause.
2. SELECT P.ProductName AS 'ProductName', P.ListPrice
3. FROM Product P
4. WHERE P.ProductID IN (
5. SELECT S.ProductID
6. FROM SaleItem S
7. WHERE S.Quantity = 3);

**FEEDBACK**Ungraded / 10

1. Which employees receive a higher-than-average salary?
2. SELECT E.FirstName AS 'EmployeeFirstName', E.LastName AS 'EmployeeLastName', SE.Salary
3. FROM Employee E
4. JOIN SalaryEmployee SE
5. ON E.EmployeeID = SE.EmployeeID
6. WHERE SE.Salary > (
7. SELECT AVG(Salary)
8. FROM SalaryEmployee)

**FEEDBACK**Ungraded / 10

1. What is the total amount of purchases made in 2015 made by each employee. Include EmployeeID, First and Last names, and total amount. (Use a select subquery in the SELECT clause). Name the column TotalPurchases. When you create it, use both the round and sum functions, i.e. Round(Sum( ... ) , 2).
2. SELECT E.EmployeeID, E.FirstName AS 'EmployeeFirstName', E.LastName AS 'EmployeeLastName',
3. (SELECT ROUND(SUM(PI.PurchasePrice \* PI.Quantity),2)
4. FROM PurchaseItem PI
5. JOIN Purchase P
6. ON PI.PurchaseID = P.PurchaseID
7. WHERE E.EmployeeID = P.EmployeeID
8. AND YEAR(PurchaseDate) = 2015) AS 'TotalPurchases'
9. FROM Employee E;

**FEEDBACK**Ungraded / 10

1. List the product names of boots made in New Hampshire. Use a subquery in the where clause.
2. SELECT P.ProductName
3. FROM Product P
4. WHERE P.ManufacturerID IN (
5. SELECT M.ManufacturerID
6. FROM Manufacturer M
7. WHERE P.Category = 'boots'
8. AND M.State = 'NH');

**FEEDBACK**Ungraded / 10

1. Count the number of unique values of the set of values of PurchaseDate and EmployeeID from the Purchase table. (Use a subquery in FROM) (name column NumbOfEmpPurch).
2. SELECT COUNT(\*) AS 'NumberOfEmployeePurchases'
3. FROM (SELECT DISTINCT PurchaseDate, EmployeeID FROM Purchase)
4. AS AliasQuery;

**FEEDBACK**Ungraded / 10

1. What is the largest purchase made by each employee? Include EmployeeID, FirstName, LastName and amount of the largest purchase. (Name it MaxAmt). (Hint, use a subquery in the FROM clause to total each Purchase. To create a total use the Sum ( ) function. Use a MAX function in the main select query to get the max on the total for each Purchase. Use joins in main query and subquery. ) Note: Due to an idiosyncracy of SQL Server, to evaluate the problem correctly, you must also use the round function to elimnate extraneous decimal digits, i.e. to create the total use Round(Sum(....), 2).
2. SELECT E.EmployeeID, FirstName, LastName, Max(PurchaseAmt) as 'MaxAmount'
3. FROM Employee E
4. JOIN (SELECT EmployeeID, P.PurchaseID, ROUND(SUM(PurchasePrice \* Quantity), 2) AS 'PurchaseAmt'
5. FROM Purchase P
6. JOIN PurchaseItem PI ON P.PurchaseID = PI.PurchaseID
7. GROUP BY P.PurchaseID, P.EmployeeID) AS SumQuery
8. ON E.EmployeeID = SumQuery.EmployeeID
9. GROUP BY E.EmployeeID, FirstName, LastName;

**FEEDBACK**Ungraded / 10

1. What is the total amount of purchases made in 2015 made by each employee? Include EmployeeID, First and Last names and total amount. (Use a join with a select subquery in the FROM clause. Remember to name the subquery.) Name the column TotalPurchases. When you create it, use both the round and sum functions, i.e. Round(Sum( ... ) , 2). Remember to remove any extra spaces between the function name and the left parenthesis of its argument.
2. SELECT E.EmployeeID, FirstName, LastName, TotalPurchases
3. FROM Employee E
4. JOIN (SELECT P.EmployeeID, ROUND(SUM(PurchasePrice\*Quantity),2) AS 'TotalPurchases'
5. FROM Purchase P
6. JOIN PurchaseItem PI ON P.PurchaseID = PI.PurchaseID
7. WHERE YEAR(PurchaseDate) = 2015
8. GROUP BY EmployeeID) AS SumQuery
9. ON E.EmployeeID=SumQuery.EmployeeID;

**FEEDBACK**Ungraded / 10

Copyright © 2022 MyEducator. All rights reserved.

[Need Help?](https://myeducator.freshdesk.com/support/home)